



Σχολή Θετικών Επιστημών και Τεχνολογίας
Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα

Διπλωματική Εργασία
«Σχεδιαμός και Δημιουργία Συστήματος Διαχείρισης Συνεδρίων»

Αλεξάνδρα Σδρίμα

Επιβλέπων καθηγητής: Δρ. Παπαμιχαήλ Γεώργιος

Πάτρα, Σεπτέμβριος 2026

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία της φοιτήτριας («συγγραφέας/δημιουργός») που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο ΕΑΠ, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.



«Σχεδιαμός και Δημιουργία Συστήματος Διαχείρισης Συνεδρίων»

Αλεξάνδρα Σδρίμα

Επιτροπή Επίβλεψης Διπλωματικής Εργασίας

Επιβλέπων Καθηγητής:
Δρ. Παπαμιχαήλ Γεώργιος

Συν-Επιβλέπων Καθηγητής:
Δρ. Μπεληγιάννης Γρηγόριος

Πάτρα, Σεπτέμβριος 2026

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους όσοι συνέβαλαν με οποιονδήποτε τρόπο στην ολοκλήρωση της παρούσας διπλωματικής εργασίας.

Ευχαριστώ επίσης όλους τους συμμετέχοντες που αφιέρωσαν χρόνο για να χρησιμοποιήσουν και να αξιολογήσουν την εφαρμογή, καθώς και για να συμπληρώσουν το ερωτηματολόγιο. Οι απαντήσεις, οι παρατηρήσεις και οι προτάσεις τους συνέβαλαν ουσιαστικά στην αξιολόγηση της ευχρηστίας και της λειτουργικότητας της εφαρμογής και στον εντοπισμό σημείων που μπορούν να βελτιωθούν.

Ιδιαίτερα, θα ήθελα να ευχαριστήσω τον Βασίλη για την απίστευτη βοήθεια, την αμέριστη συμπαράσταση και τη συνεχή ενθάρρυνση που μου προσέφερε σε όλη τη διάρκεια αυτής της προσπάθειας. Η στήριξή του υπήρξε καθοριστική, τόσο στις εύκολες όσο και στις δύσκολες στιγμές, δίνοντάς μου δύναμη και κίνητρο να συνεχίσω μέχρι την ολοκλήρωση της εργασίας.

Επίσης, θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες στον επιβλέποντα καθηγητή κ. Παπαμιχαήλ Γεώργιο για την πολύτιμη καθοδήγη του καθ' όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας.

Περίληψη

Η παρούσα εργασία παρουσιάζει τη μελέτη, το σχεδιασμό και την υλοποίηση ενός ολοκληρωμένου συστήματος διαχείρισης συνεδρίων. Το σύστημα έχει ως κύριο σκοπό την υποστήριξη όλων των θεμελιωδών διαδικασιών που εμπλέκονται στην οργάνωση ενός συνεδρίου για δημοσίευση: εγγραφή και συμμετοχή, υποβολή επιστημονικών εργασιών, αξιολόγησή τους, ενσωμάτωσή τους σε θεματικές ενότητες και τελική τους διαμόρφωση για παρουσίαση.

Ακολουθώντας ένα δυναμικό προφίλ χρήστη βάσει ρόλων, κάθε χρήστης δεν αποκτά μόνιμο ρόλο κατά την εγγραφή, αλλά αποκτά ρόλο ανάλογα με κάθε συνέδριο στο οποίο συμμετέχει. Ένας χρήστης μπορεί έτσι να γίνει απλός επισκέπτης ή παρατηρητής, ή παρουσιαστής, αξιολογητής ή διοργανωτής, ανάλογα με τη δραστηριότητα που εκτελεί και τα δικαιώματα που του χορηγούνται. Το σύστημα υποστηρίζει την εγγραφή και τη σύνδεση χρηστών, την προβολή της δημοσίευσης του συνεδρίου, την προσθήκη συνεδρίων σε λίστα παρακολούθησης, την υποβολή επιστημονικής εργασίας και την παρακολούθηση της κατάστασης κάθε υποβολής, προκειμένου να λαμβάνει ειδοποιήσεις για σημαντικές αλλαγές ή αναθέσεις.

Μεγάλη σημασία αποδίδεται στη φάση αξιολόγησης των εργασιών, κατά την οποία οι διοργανωτές μπορούν να αναθέσουν εργασίες σε αξιολογητές. Το σύστημα επαληθεύει περιορισμούς κατά τη διάρκεια αυτής της διαδικασίας, όπως η ανάθεση ενός αξιολογητή στην ίδια εργασία την οποία έχει ήδη αξιολογήσει, ή η ανάθεση της ίδιας εργασίας σε δύο διαφορετικούς αξιολογητές. Οι αξιολογητές λαμβάνουν πρόσβαση στις εργασίες για αξιολόγηση. Στη συνέχεια, οι αξιολογητές μπορούν να αξιολογήσουν κάθε υποβολή και να εισάγουν μια βαθμολογία, σχόλια και μια σύσταση (αποδοχή/απόρριψη). Μετά τη λήψη αυτής της αξιολόγησης, η εργασία μπορεί να απορριφθεί ή να γίνει αποδεκτή ή να ζητηθεί η αξιολόγηση από τρίτο αξιολογητή. Μόλις γίνει αποδεκτή, οι διοργανωτές μπορούν να ενσωματώσουν την επιστημονική εργασία σε μια συγκεκριμένη ενότητα και να ενεργοποιήσουν τη λειτουργία πληρωμής μέσω του εξωτερικού περιβάλλοντος εφαρμογής Stripe Checkout. Μετά την επιτυχή ολοκλήρωση της πληρωμής, η κατάσταση του άρθρου αλλάζει σε "έτοιμο για παρουσίαση".

Επιπλέον, το σύστημα παρέχει λειτουργικότητα διαχείρισης μέσω ενός χρήστη προσωπικού που μπορεί να δημιουργήσει νέα συνέδρια. Ο χρήστης προσωπικού μπορεί να αποφασίσει τις ημερομηνίες υποβολής και εκδήλωσης, τον αριθμό των θεματικών ενότητων, τα τέλη του συνεδρίου, το νόμισμα πληρωμής και την κατάσταση δημοσίευσης του συνεδρίου. Στην προσωπική ενότητα κάθε χρήστη, το σύστημα παρέχει μια ενοποιημένη προβολή των ρόλων του, των υποβολών του, των ειδοποιήσεων και των συνεδρίων που παρακολουθεί, καθώς και των υποβολών που έχουν ανατεθεί στην αξιολόγηση κάθε χρήστη.

Η υλοποίηση μιας τέτοιας εφαρμογής οδηγεί στην αυτοματοποίηση και την οργάνωση της ροής εργασιών δημοσίευσης ενός συνεδρίου, ελαχιστοποιώντας τη χειροκίνητη παρέμβαση, βελτιώνοντας τη διαφάνεια της διαδικασίας αξιολόγησης και διευκολύνοντας την αλληλεπίδραση αξιολογητών, διοργανωτών, διαχειριστών και χρηστών μέσω μιας κοινής πλατφόρμας. Με βάση τα αποτελέσματα των δοκιμών, το σύστημα αποδεικνύεται ένα λειτουργικό και επεκτάσιμο εργαλείο για τη διαχείριση συνεδρίων και την υποβολή, αξιολόγηση και παρουσίαση επιστημονικών εργασιών.

Λέξεις – Κλειδιά

Σύστημα Πληροφοριών Συνεδρίων, Διαχείριση Συνεδρίων, Υποβολή Εργασιών, Αξιολόγηση Εργασιών, Αξιολογητές, Ενότητες, Django, Stripe Checkout

«Design and implementation of a conference management system»

Alexandra Sdrima

Abstract

The goal of this thesis is the study, design, and implementation of an integrated information system for scientific conference management. The application primarily aims to assist in the processes related to: publishing and managing conferences, registration and management of users, paper submission, review assignment, paper evaluation, track organization for accepted papers, and preparation of the final paper presentation.

In the role-based model of the implemented solution, the user does not take on a fixed role. Rather, each user's role is specific to each conference they are involved in and depends on their specific actions and/or responsibilities within the context of that conference. This means that a user may take part as a guest, an observer, a presenter, a reviewer, or an organizer, depending on the conference. Additionally, the application allows users to register and authenticate themselves, browse through published conferences, add conferences to their watchlist, submit a paper, track the status of paper submission, and receive notifications in case of significant events, assignments, and decisions.

The focus was given to the paper review process. This process includes the organizer's assignment of reviewers to specific papers, which is restricted by certain limitations like not being able to assign the same reviewer twice for a paper. Reviewers, upon seeing the papers assigned to them, can download the papers, submit a score for the paper, add comments, and express their opinion on the final decision (i.e., recommend acceptance, rejection, and third review). Based on the reviewer decision, the paper can be rejected, accepted, or marked as awaiting further review. Accepted papers are organized into tracks by the organizer. In

this case, the participant (reviewers, organizers) is able to proceed to Stripe Checkout for the payment. After a successful payment, a paper can be presented.

A staff user is also granted administrative functionality where they can create a new conference, set submission and conference dates, set the number of available tracks, set a participation fee, select a payment currency, and set the publication status of each conference. A user can access their personal dashboard for an overview of roles they hold for a specific conference, papers they have submitted, notifications, the conference watchlist, and pending assignments of papers to be reviewed. The implementation of a system of this type is an effort to organize and automate conference processes, eliminate unnecessary human intervention, ensure the transparency of the process, and establish a collaborative platform for the participants. A functional and extensible paper submission and review system for scientific conferences is demonstrated with an example of organizing paper tracks, payment, and final paper presentation.

Keywords

conference management system, scientific conferences, paper submission, paper review, reviewers, tracks, Django, Stripe Checkout

Περιεχόμενα

ΠΕΡΙΛΗΨΗ.....	V
ABSTRACT	VII
ΠΕΡΙΕΧΟΜΕΝΑ	IX
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ / ΣΧΗΜΑΤΩΝ.....	X
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	XII
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ & ΑΚΡΩΝΥΜΙΑ	XIII
1. ΕΙΣΑΓΩΓΗ	1
1.1 ΣΚΟΠΟΣ ΚΑΙ ΣΤΟΧΟΙ ΤΗΣ ΕΡΓΑΣΙΑΣ	1
1.2 ΥΦΙΣΤΑΜΕΝΕΣ ΕΦΑΡΜΟΓΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΣΥΝΕΔΡΙΩΝ	2
1.3 ΔΟΜΗ ΕΡΓΑΣΙΑΣ.....	6
2. ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ	9
2.1 ΛΟΓΙΣΜΙΚΑ ΚΑΙ ΕΡΓΑΛΕΙΑ	9
2.1.1 Η γλώσσα προγραμματισμού Python.....	10
2.1.2 Το πλαίσιο ανάπτυξης εφαρμογών ιστού Django	12
2.1.3 Η βάση δεδομένων SQLite.....	14
2.1.4 JSON API	16
2.1.4 HTML & Django Template Language.....	19
2.1.5 CSS	21
2.1.6 Stripe.....	23
2.1.7 Mailpit	24
2.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ	25
3. ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΜΟΣ ΕΦΑΡΜΟΓΗΣ	28
3.2 ΡΟΛΟΙ, ΔΙΚΑΙΩΜΑΤΑ ΚΑΙ ΣΕΝΑΡΙΑ ΧΡΗΣΗΣ	29
3.3 ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ	29
3.4 ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ	30
3.4.1 Τίτλος περίπτωσης χρήσης : Διαχείριση πίνακα και watchlist.....	33
3.4.2 Τίτλος περίπτωσης χρήσης: Συμμετοχή σε συνέδριο και υποβολή paper	35
3.4.3 Τίτλος περίπτωσης χρήσης: Διαχείριση υποβολών από Presenter	37
3.4.4 Τίτλος περίπτωσης χρήσης: Αξιολόγηση εργασιών από Reviewer.....	39
3.4.5 Τίτλος περίπτωσης χρήσης: Διαχείριση συνεδρίου από Organizer	41
3.4.6 Τίτλος περίπτωσης χρήσης: Διαχείριση συστήματος από Superuser / Staff.....	44
3.5 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΔΕΔΟΜΕΝΩΝ	46
4. ΥΛΟΠΟΙΗΣΗ ΛΕΙΤΟΥΡΓΙΩΝ.....	60
4.1 ΔΙΑΧΕΙΡΙΣΗ ΣΥΝΕΔΡΙΟΥ ΚΑΙ ΡΟΛΩΝ.....	60
4.2 ΥΠΟΒΟΛΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ PAPER ΜΕ ΚΑΝΟΝΑ ΟΜΟΦΩΝΙΑΣ.....	60
4.3 ΑΝΑΘΕΣΗ TRACK ΚΑΙ ΟΛΟΚΛΗΡΩΣΗ ΠΛΗΡΩΜΗΣ.....	61
4.3.1 Δημιουργία Checkout Session και επιστροφή στο σύστημα	62
4.3.2 Email ειδοποιήσεις και περιβάλλον ανάπτυξης.....	63
4.4 ΕΙΔΟΠΟΙΗΣΕΙΣ ΚΑΙ ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΣΥΝΕΔΡΙΩΝ.....	63
4.5 ΚΑΤΑΣΤΑΣΕΙΣ, ΜΕΤΑΒΑΣΕΙΣ ΚΑΙ ΕΛΕΓΧΟΙ ΑΚΕΡΑΙΟΤΗΤΑΣ	63
4.6 ΠΛΟΗΓΗΣΗ ΣΤΗΝ ΕΦΑΡΜΟΓΗ	64
4.6.1 Αρχική πρόσβαση και προβολή συνεδρίων	64
4.6.2 Πίνακας ελέγχου χρήστη	69
4.6.3Υποβολή και παρακολούθηση εργασίας.....	72
4.6.4 Λειτουργίες Staff και Organizer	72
4.6.5 Διαδικασία αξιολόγησης από Reviewer	77

4.6.6 Πληρωμή και Κατάσταση Presentation Ready	79
4.6.7 Διαχειριστικό περιβάλλον	80
4.6.8 Περιβάλλον Mailpit	81
5. ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	82
5.1 ΣΚΟΠΟΣ ΚΑΙ ΕΡΕΥΝΗΤΙΚΑ ΕΡΩΤΗΜΑΤΑ	82
5.2 ΜΕΘΟΔΟΛΟΓΙΑ ΑΞΙΟΛΟΓΗΣΗΣ	83
5.2.1 Επιλογή Συμμετεχόντων	84
5.2.2 Περιβάλλον Δοκιμών	85
5.2.3 Συλλογή Δεδομένων	85
5.3 ΣΕΝΑΡΙΑ ΔΟΚΙΜΩΝ ΑΝΑ ΡΟΛΟ	85
5.3.1 Μη Συνδεδεμένος Χρήστης	85
5.3.2 Συνδεδεμένος Χρήστης και Κοινές Λειτουργίες	86
5.3.3 Λίστα Παρακολούθησης και Ρόλος Παρατηρητή	86
5.3.4 Υποβολή και Παρακολούθηση Υποβολής	87
5.3.5 Αξιολόγηση Υποβολής από Αξιολογητή	88
5.3.7 Δημιουργία και Διαχείριση Συνεδρίου από Διαχειριστή/Superuser	89
5.4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΞΙΟΛΟΓΗΣΗΣ	90
5.4.1 Γενική Αξιολόγηση	90
5.4.2 Αξιολόγηση από μη Συνδεδεμένο Επισκέπτη	91
5.4.3 Προσωπικός Πίνακας ελέγχου και Λίστα Παρακολούθησης	91
5.4.4 Αξιολόγηση Λειτουργιών Παρατηρητή	92
5.4.5 Αξιολόγηση Λειτουργιών Παρουσιαστή	93
5.4.6 Αξιολόγηση Λειτουργιών Αξιολογητή	94
5.4.8 Αξιολόγηση Λειτουργιών Διαχειριστή – Superuser	95
5.4.9 Συνολική Βαθμολογία Ευχρηστίας – Αποτελεσματικότητας	96
5.5 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΝΟΙΚΤΩΝ ΕΡΩΤΗΣΕΩΝ	96
5.6 ΧΡΗΣΙΜΟΤΕΡΕΣ ΛΕΙΤΟΥΡΓΙΕΣ	97
5.7 ΚΥΡΙΟΤΕΡΕΣ ΔΥΣΚΟΛΙΕΣ	98
5.8 ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΒΕΛΤΙΩΣΕΙΣ	99
5.9 ΠΕΡΙΟΡΙΣΜΟΙ ΤΗΣ ΑΞΙΟΛΟΓΗΣΗΣ	100
5.10 ΣΥΜΠΕΡΑΣΜΑΤΑ ΑΞΙΟΛΟΓΗΣΗΣ	100
6. ΣΥΜΠΕΡΑΣΜΑΤΑ	102
ΒΙΒΛΙΟΓΡΑΦΙΑ	103
ΠΑΡΑΡΤΗΜΑ Α: ΕΝΔΕΙΚΤΙΚΟΣ ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ	108
ΠΑΡΑΡΤΗΜΑ Β: ΣΥΝΟΠΤΙΚΕΣ ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ	140

Κατάλογος Εικόνων / Σχημάτων

Εικόνα 1 Διάγραμμα Περιπτώσεων Χρήσης	31
Εικόνα 2 Διάγραμμα Διαχείρισης Πίνακα Χρήστη και Watchlist	35
Εικόνα 3 Διάγραμμα Συμμετοχής σε Συνέδριο και Υποβολής Paper	37
Εικόνα 4 Διάγραμμα Διαχείρισης Υποβολών από Presente	39
Εικόνα 5 Διάγραμμα Αξιολόγησης Εργασιών από Reviewer	41
Εικόνα 6 Διάγραμμα Διαχείρισης Συνεδρίου από Organizer	43
Εικόνα 7 Διάγραμμα Διαχείρισης Συστήματος από Superuser / Staff	45
Εικόνα 8 Διάγραμμα Οντοτήτων Συσχετίσεων	59
Εικόνα 9 Ροή Υποβολής και Αξιολόγησης Paper	61
Εικόνα 10 Ακολουθία Πληρωμής με Stripe Checkout και Webhooks	62
Εικόνα 11 Αρχική μη Συνδεδεμένου Χρήστη	65

Εικόνα 12 Σελίδα Συνεδρίου μη Συνδεδεμένου Χρήστη	66
Εικόνα 13 Αρχική Οθόνη Συνδεδεμένου Χρήστη	67
Εικόνα 14 Σελίδα Συνεδρίου Συνδεδεμένου Χρήστη	68
Εικόνα 15 Πίνακας Ελέγχου Απλού Χρήστη	69
Εικόνα 16 Πίνακας Ελέγχου μετά την Εγγραφή ως Observer	70
Εικόνα 17 Πίνακας Ελέγχου Presenter με Υποβολές	71
Εικόνα 18 Φόρμα Νέας Υποβολής Paper	72
Εικόνα 19 Staff Panel: Δημιουργία Συνεδρίου	73
Εικόνα 20 Πίνακας Ελέγχου Organizer	74
Εικόνα 21 Organizer View: Ρόλοι και Tracks, Μέλη και Ανάθεση Reviewers, Αποτελέσματα Reviews και Καταστάσεις Papers, Ανάθεση Τρίτου Reviewer, Ανάθεση Accepted Paper σε Track	76
Εικόνα 22 Reviewer Dashboard με Review Queue	77
Εικόνα 23 Φόρμα Αξιολόγησης Paper	78
Εικόνα 24 Πληρωμή και Κατάσταση Presentation Ready	79
Εικόνα 25 Διαχειριστικό Περιβάλλον	80
Εικόνα 26 Περιβάλλον Mailpit	81

Κατάλογος Πινάκων

Πίνακας 1 Ενδεικτική Σύγκριση Υφιστάμενων Λύσεων.....	6
Πίνακας 2 Πεδία του Πίνακα Χρήστηs	51
Πίνακας 3 Πεδία του Πίνακα Συνέδριο	52
Πίνακας 4 Πεδία του Πίνακα Θεματική Ενότητα	52
Πίνακας 5 Πεδία του Πίνακα Συμμετοχή σε Συνέδριο	53
Πίνακας 6 Πεδία του Πίνακα Λίστα Παρακολούθησης	53
Πίνακας 7 Πεδία του Πίνακα Ειδοποίηση	54
Πίνακας 8 Πεδία του Πίνακα Πληρωμή	55
Πίνακας 9 Πεδία του Πίνακα Συμβάν Πληρωμής	56
Πίνακας 10 Πεδία του Πίνακα Ανάθεση Αξιολόγησης	57
Πίνακας 11 Πεδία του Πίνακα Αξιολόγηση	58
Πίνακας 12 Πεδία του Πίνακα Υποβολή	58
Πίνακας 13 Προφίλ συμμετεχόντων και ρόλοι που αξιολογήθηκαν κατά τη δοκιμή της εφαρμογής.....	84
Πίνακας 14 Αποτελέσματα Γενικής Αξιολόγησης της Εφαρμογής.....	90
Πίνακας 15 Αποτελέσματα Αξιολόγησης από Μη Συνδεδεμένο Χρήστη.....	91
Πίνακας 16 Αποτελέσματα Αξιολόγησης Προσωπικού Πίνακα Ελέγχου & Λίστας Παρακολούθησης.....	92
Πίνακας 17 Αποτελέσματα Αξιολόγησης Λειτουργιών Παρατηρητή.....	92
Πίνακας 18 Αποτελέσματα Αξιολόγησης Λειτουργιών Παρουσιαστή.....	93
Πίνακας 19 Αποτελέσματα Αξιολόγησης Λειτουργιών Αξιολογητή.....	94
Πίνακας 20 Αποτελέσματα Αξιολόγησης Λειτουργιών Οργανωτικής Επιτροπής.....	95
Πίνακας 21 Αποτελέσματα Αξιολόγησης Λειτουργιών Διαχειριστή/Superuser.....	95
Πίνακας 22 Συνολική Αξιολόγηση Ευχρηστίας & Αποτελεσματικότητας Εφαρμογής...	96

Συντομογραφίες & Ακρωνύμια

ΕΑΠ	Ελληνικό Ανοικτό Πανεπιστήμιο
API	Διεπαφή Προγραμματισμού Εφαρμογών
ΠΧ	Περίπτωση Χρήσης
DTL	Django Template Language
HTML	Hypertext Markup Language
UI	User Interface

1. Εισαγωγή

Τα συνέδρια αποτελούν πρωταρχικό τρόπο διάδοσης της γνώσης, καθώς δημιουργούν ένα περιβάλλον για την παρουσίαση, συζήτηση και ανασκόπηση της έρευνας με επίσημο, συστηματικό τρόπο. Η λειτουργικότητα ενός συνεδρίου προκύπτει από μια σειρά ρόλων που πρέπει να εκπληρωθούν. Για τη δημιουργία και δημοσίευση του συνεδρίου, την εγγραφή σε ένα συνέδριο, την υποβολή εργασιών, την ανάθεση κριτών, την αποδοχή ή απόρριψη εργασιών, ή τη δημιουργία του προγράμματος του συνεδρίου, απαιτούνται κανόνες.

Συνήθως, στην πραγματική πρακτική, οι παραπάνω ροές υλοποιούνται μερικώς χειροκίνητα. Η χρήση κατακερματισμένων εργαλείων όπως email, υπολογιστικά φύλλα ή άσχετα έγγραφα οδηγεί σε αύξηση του λειτουργικού κόστους, ασυνέπεια της κατάστασης και μειωμένη ιχνηλασιμότητα των πληροφοριών. Η έλλειψη μιας μοναδικής πηγής πληροφοριών καθιστά δύσκολη την παρακολούθηση της διαδικασίας λήψης αποφάσεων και την ανίχνευσή της σε σχετικές ενέργειες όταν υπάρχουν πολλαπλοί ρόλοι και πολλαπλές εργασίες που επεξεργάζονται.

Στόχος της εργασίας είναι ο σχεδιασμός και η ανάπτυξη μιας διαδικτυακής εφαρμογής σε Django που καλύπτει τον πλήρη κύκλο ζωής ενός ακαδημαϊκού συνεδρίου από την οπτική γωνία των βασικών ενδιαφερομένων. Η υλοποίηση περιλαμβάνει έλεγχο πρόσβασης βάσει ρόλων, σύστημα αξιολόγησης από ομοτίμους βασισμένο σε κανόνα συναίνεσης, διαχείριση διαφορετικών θεματικών ενοτήτων και επιβολή συμμετοχής μέσω πληρωμής στην περίπτωση των αποδεκτών εργασιών. Επιπλέον, η εφαρμογή έχει σχεδιαστεί λαμβάνοντας υπόψη τις προδιαγραφές παραγωγής λογισμικού, συμπεριλαμβανομένων πρακτικών ασφαλείας, καταγραφής ελέγχου και εργαλείων παρατηρησιμότητας.

1.1 Σκοπός και Στόχοι της Εργασίας

Ο κύριος στόχος είναι η δημιουργία ενός λογισμικού διαχείρισης συνεδρίων που μπορεί να υποστηρίξει πραγματικές περιπτώσεις χρήσης και να αποφύγει κοινά λάθη στη λειτουργία του συστήματος, χρησιμοποιώντας σαφώς καθορισμένες καταστάσεις και μεταβάσεις καταστάσεων. Από λειτουργική άποψη, επιτρέπει τη δημιουργία και δημοσίευση συνεδρίων από το προσωπικό του συνεδρίου. Παρέχει επίσης σε κάθε χρήστη ένα ενιαίο πίνακα ελέγχου που εμφανίζει όλα τα συνέδρια που παρακολουθεί αυτός ο χρήστης (λίστα

παρακολούθησης), τα συνέδρια στα οποία ο χρήστης έχει θεσμικούς ρόλους και τα εκκρεμή στοιχεία που πρέπει να διαχειριστεί ο χρήστης.

Οι κύριες ροές αφορούν την υποβολή και την αξιολόγηση εργασιών. Στην περίπτωση της διαδικασίας αξιολόγησης, χρησιμοποιείται ένας κανόνας συναίνεσης βασισμένος σε δύο κριτές. Μόλις οι απόψεις συγκλίνουν, η εργασία λαμβάνει την αντίστοιχη κατάσταση. Εάν προκύψει διαφωνία, εμπλέκεται ένας τρίτος κριτής και η εργασία λαμβάνει ένα κατάλληλο αποτέλεσμα. Μετά την αποδοχή μιας εργασίας, το λογισμικό επιτρέπει επίσης την προσθήκη της εργασίας σε μια θεματική ενότητα από την Οργανωτική Επιτροπή και την τελική επιβεβαίωση με την πληρωμή του τέλους εγγραφής, εάν το συνέδριο έχει οριστεί ως επί πληρωμή.

Στην περίπτωση των απαιτήσεων για ετοιμότητα παραγωγής, εφαρμόζονται τα ακόλουθα. Η εφαρμογή υλοποιεί προστασία από Cross Site Request Forgery (CSRF) και η επιφάνεια Cross Site Scripting (CSP) ελαχιστοποιείται χρησιμοποιώντας μια πολιτική ασφάλειας περιεχομένου. Επιπλέον υλοποιείται καταγραφή ελέγχου για τις πιο σημαντικές ενέργειες.

1.2 Υφιστάμενες Εφαρμογές Διαχείρισης Συνεδρίων

Η διαχείριση ενός συνεδρίου περιλαμβάνει πολλαπλά στάδια και διαφορετικούς τύπους φορέων. Σε γενικές γραμμές, μπορεί να χαρακτηριστεί μέσω της δημιουργίας των μεταδεδομένων και των θεματικών ενοτήτων του συνεδρίου, της υποβολής άρθρων/περιλήψεων από τους συγγραφείς, της αξιολόγησης, της αποδοχής ή απόρριψης υποβολών, της δημιουργίας προγράμματος και της εγγραφής συμμετεχόντων. Λαμβάνοντας υπόψη αυτή την ανάγκη, υπάρχουν αρκετές διαδικτυακές εφαρμογές διαθέσιμες που καλύπτουν μερικώς ή πλήρως τις αναφερόμενες απαιτήσεις.

Ένα από τα πιο διάσημα συστήματα στον τομέα των ακαδημαϊκών συνεδρίων είναι το EasyChair. Το EasyChair προσφέρει ένα ολοκληρωμένο περιβάλλον για τη διαχείριση ακαδημαϊκών συνεδρίων με λειτουργίες όπως υποβολή περιλήψεων, αξιολόγηση, πρόγραμμα, εγγραφή συμμετεχόντων, πρακτικά και δημιουργία πρόσκλησης για υποβολή εργασιών. Σύμφωνα με το εγχειρίδιο του EasyChair, το σύστημα χρησιμοποιείται ευρέως από πολλά συνέδρια και υποστηρίζει όλους τους τύπους διοργάνωσης συνεδρίων, από μικρές συναντήσεις έως τεράστια συνέδρια.

Το Microsoft Conference Management Toolkit (CMT) είναι επίσης ένα φιλοξενούμενο σύστημα, το οποίο επιτρέπει τη διαχείριση ενός ακαδημαϊκού συνεδρίου. Υποστηρίζει απεριόριστες υποβολές, διαφορετικά ατομικά χρονοδιαγράμματα και προθεσμίες, παρακολούθηση, μονή και διπλή τυφλή αξιολόγηση, προσαρμόσιμες φόρμες υποβολής και αξιολόγησης, και ανάθεση εργασιών σε αξιολογητές. Υποστηρίζει επίσης γύρους αξιολόγησης, συγκρούσεις συμφερόντων και αυτόματη ανάθεση εργασιών.

Το Indico προσφέρει μια διαφορετική προσέγγιση από τα προηγούμενα παραδείγματα, καθώς είναι λογισμικό ανοιχτού κώδικα που αναπτύχθηκε από το CERN και είναι αφιερωμένο στη διοργάνωση εκδηλώσεων γενικά, χωρίς ιδιαίτερη εστίαση στα ακαδημαϊκά συνέδρια. Το Indico επιτρέπει τη διοργάνωση συνεδρίων, εργαστηρίων και άλλων εκδηλώσεων όπως διαλέξεις και συναντήσεις. Στην περίπτωση των συνεδρίων, το εγχειρίδιο του Indico αναφέρει λειτουργίες όπως οργάνωση συνεδρίου, δημιουργία και υποβολή περιλήψεων, εγγραφή, πληρωμή μέσω διαφόρων μέσων και διαδικασία αξιολόγησης δημοσιεύσεων.

Οι προαναφερθείσες εφαρμογές συγκρίθηκαν με το OpenConf. Το OpenConf παρέχει παρόμοιες λειτουργίες, αλλά επικεντρώνεται στη διαχείριση υποβολών και αξιολογήσεων. Προσφέρει υποβολή εργασιών, μεταφόρτωση αρχείων εργασιών, διαδικασία αξιολόγησης, αποδοχή ή απόρριψη εργασιών, αυτόματες ειδοποιήσεις, εξαγωγή δεδομένων, υποβολή προσφορών για εργασίες, ανάθεση, συζητήσεις, βαθμολογία, διαχείριση εργασιών, πρόγραμμα και διαδικτυακά πρακτικά. Το OpenConf έχει την επιλογή να είναι τόσο μια φιλοξενούμενη υπηρεσία όσο και να εγκαθίσταται στον διακομιστή ενός οργανισμού, καθιστώντας το ένα ελκυστικό σύστημα για έναν οργανισμό που θέλει έλεγχο στο λειτουργικό του σύστημα.

Το HotCRP είναι ένα άλλο σύστημα με εστίαση στη διαχείριση υποβολών και αξιολογήσεων, με έμφαση στα συνέδρια επιστήμης υπολογιστών. Οι λειτουργίες του HotCRP περιλαμβάνουν ευχρηστία στη διαδικασία αξιολόγησης, αναζήτηση και φιλτράρισμα εργασιών, προσαρμοσμένες ετικέτες, προσαρμοσμένες φόρμες αξιολόγησης, πολλαπλούς γύρους αξιολόγησης, ανάθεση σε εξωτερικούς αξιολογητές, συζητήσεις και διαχείριση επιτροπής προγράμματος. Στην αρχική του σελίδα, ο δημιουργός του HotCRP εξηγεί ότι το σύστημα είναι ένα "πακέτο υποβολής και αξιολόγησης συνεδρίων μέσω διαδικτύου. Μπορεί να διαμορφωθεί για να λειτουργεί σε έναν υπάρχοντα ιστότοπο και

εστιάζει στην ευκολία χρήσης κατά τη διαδικασία αξιολόγησης και στη μέγιστη ευελιξία στον τρόπο διεξαγωγής των συνεδρίων."

Το ConfTool είναι ένα άλλο ολοκληρωμένο σύστημα που υποστηρίζει τη διαχείριση υποβολών και αξιολογήσεων, καθώς και την εγγραφή συνεδρίων και συμμετεχόντων. Προσφέρει λειτουργίες όπως προσαρμόσιμη διαδικασία υποβολής και αξιολόγησης, μεταφόρτωση αρχείων εργασιών, υποβολή προσφορών, προσαρμοσμένες φόρμες αξιολόγησης, ανοιχτή και τυφλή αξιολόγηση, φόρουμ επιτροπής, πολλαπλές παράλληλες συνεδρίες, πρόγραμμα, προσωπικό χρονοδιάγραμμα, διαδικτυακή εγγραφή, τιμολόγια και πληρωμή με διαφορετικούς τρόπους. Το ConfTool καλύπτει τόσο την ακαδημαϊκή όσο και την οργανωτική/διοικητική πτυχή του συνεδρίου.

Στην ίδια κατηγορία, υπάρχει επίσης το ακαδημαϊκό σύστημα EDAS που καλύπτει τη διαδικασία υποβολής και αξιολόγησης εργασιών και την εγγραφή για συνέδρια, εργαστήρια και περιοδικά. Το EDAS είναι φιλοξενούμενο και οι πρόεδροι/διαχειριστές του μπορούν να δημιουργήσουν ένα συνέδριο χρησιμοποιώντας φόρμες στο διαδίκτυο, ενώ οι χρήστες αλληλεπιδρούν μέσω προγράμματος περιήγησης στο διαδίκτυο. Παρέχει λειτουργίες όπως ιστοσελίδα συνεδρίου, δημοσιότητα, προσωπικός λογαριασμός μέσω ηλεκτρονικού ταχυδρομείου, υποβολή αρχείων, παράλληλες συνεδρίες με προθεσμίες, έλεγχος υποβολών.

Επιπλέον, εκτός από την ακαδημαϊκή εστίαση στις υποβολές, υπάρχουν συστήματα που χρησιμοποιούνται για τη διαχείριση εκδηλώσεων, όπως το Whova και το Eventbrite, τα οποία εστιάζουν στην εγγραφή συμμετεχόντων και στο πρόγραμμα, στη διαχείριση ομιλητών και στο check-in, στις ανακοινώσεις και στη δικτύωση συμμετεχόντων και στην εμπειρία του κοινού. Το Whova για παράδειγμα παρέχει λειτουργίες όπως εγγραφή, πρόσκληση ομιλητών, διαχείριση κονκάρδων, διαχείριση ατζέντας, περίπτερο check-in (αυτοεξυπηρέτηση check-in), δικτύωση συμμετεχόντων και ζωντανές ανακοινώσεις. Το Eventbrite, από την άλλη πλευρά, καλύπτει κυρίως τη δημιουργία μιας εκδήλωσης, την εγγραφή συμμετεχόντων, τη δημιουργία εισιτηρίων, τις πληρωμές και την προώθηση μιας εκδήλωσης.

Από την προηγούμενη παράγραφο, συνάγεται ότι υπάρχουν πολλά ακαδημαϊκά συστήματα, που καλύπτουν σε μεγάλο βαθμό την οργάνωση συνεδρίων. Υπάρχουν συστήματα που εστιάζουν στη διαχείριση υποβολών και αξιολογήσεων (EasyChair, CMT, OpenConf, HotCRP, EDAS), συστήματα που εστιάζουν στην εγγραφή συμμετεχόντων, στη διαχείριση

προγράμματος και στην προώθηση, και στην εμπειρία του κοινού (Whova, Eventbrite) και συστήματα που προσπαθούν να καλύψουν το μεγαλύτερο μέρος των φάσεων ενός συνεδρίου, όπως το περιεχόμενο, η διαχείριση συμμετεχόντων και προγράμματος (Indico, ConfTool).

Ωστόσο, ορισμένα από τα παραπάνω συστήματα είναι εμπορικά, φιλοξενούνται και προορίζονται να καλύψουν τις κοινές απαιτήσεις μεγάλων συνεδρίων, γεγονός που τα καθιστά ακατάλληλα για άλλες, μικρότερες οντότητες ή πανεπιστημιακά τμήματα ή για όσους ενδιαφέρονται για ένα απλούστερο, προσαρμόσιμο και πλήρως ελεγχόμενο σύστημα διαχείρισης. Επιπλέον, σε πολλές περιπτώσεις, η προσαρμογή της ροής εργασιών, της γλώσσας διεπαφής, των ρόλων χρηστών, των φορμών ή των αναφορών ενδέχεται να μην είναι δυνατή, ή το πολύ να απαιτεί τη χρήση άλλων πακέτων, υπηρεσιών ή προϊόντων. Γι' αυτό υπάρχει χώρος για τη δημιουργία μιας τέτοιας ελαφριάς, προσαρμόσιμης διαδικτυακής εφαρμογής ικανής να υποστηρίξει βασική λειτουργικότητα διαχείρισης συνεδρίων.

Για το σκοπό αυτό, η παρούσα εργασία περιγράφει το σχεδιασμό και την υλοποίηση ενός συστήματος διαχείρισης συνεδρίων χρησιμοποιώντας το Django. Το Django είναι ένα πλαίσιο διαδικτυακών εφαρμογών υψηλού επιπέδου της Python που ενθαρρύνει την ταχεία ανάπτυξη και έναν καθαρό, πραγματιστικό σχεδιασμό, και περιλαμβάνει ενσωματωμένες λύσεις, οι οποίες μπορεί να είναι ιδιαίτερα χρήσιμες για τέτοια συστήματα, π.χ., ένας αντικειμενο-σχεσιακός αντιστοιχιστής, σύστημα ελέγχου ταυτότητας, διεπαφή διαχειριστή, σύστημα προτύπων και βασικοί μηχανισμοί ασφαλείας.

Η εφαρμογή που περιγράφεται στην παρούσα εργασία διαφέρει από τις υπάρχουσες λύσεις που αναφέρθηκαν παραπάνω, καθώς προορίζεται να είναι ένα ανοιχτό προς χρήση, προσαρμόσιμο σύστημα ικανό να χρησιμοποιηθεί για ακαδημαϊκούς σκοπούς. Ο στόχος δεν είναι ο σχεδιασμός ενός συστήματος που θα είναι μια ολοκληρωμένη, εμπορική εναλλακτική λύση, αλλά μάλλον η ανάπτυξη ενός απλού, κατανοητού πυρήνα συστήματος διαχείρισης συνεδρίων με τυπικούς ρόλους συμμετεχόντων στη διαδικασία (π.χ. διαχειριστής, διοργανωτής, συγγραφέας, κριτής και συμμετέχων), και βασικά λειτουργικά μπλοκ (π.χ. δημιουργία συνεδρίου, δημιουργία θεματικών συνεδρίων, υποβολή εργασιών, ανάθεσή τους σε κριτές, συγγραφή των εκθέσεων αξιολόγησής τους, καθορισμός της απόφασης, καθορισμός του προγράμματος του συνεδρίου, δημοσίευση των πληροφοριών του συνεδρίου στους συμμετέχοντες).

Μπορεί να εξελιχθεί στο μέλλον με πρόσθετες λειτουργίες (π.χ. διαχείριση online πληρωμών, έκδοση πιστοποιητικών συνεδρίων, αυτοματοποιημένη διαχείριση email για τους χρήστες, δημιουργία στατιστικών και εξαγωγή δεδομένων, υποστήριξη πολλαπλών γλωσσών και διεθνοποίησης, υβριδικά συνέδρια).

Στον παρακάτω πίνακα παρουσιάζεται μια ενδεικτική σύγκριση των υφιστάμενων λύσεων.

Εφαρμογή	Κύρια στόχευση	Ενδεικτικές λειτουργίες
EasyChair	Ακαδημαϊκά συνέδρια	Υποβολές, reviews, πρόγραμμα, εγγραφές, proceedings
Microsoft CMT	Ακαδημαϊκά συνέδρια μεγάλης κλίμακας	Tracks, blind review, conflict management, reviewer assignment
Indico	Γενική διαχείριση εκδηλώσεων και συνεδρίων	Call for abstracts, registration, e-payments, programme
OpenConf	Υποβολές και peer review	Submission, review, bidding, automated assignments, proceedings
HotCRP	Review επιστημονικών εργασιών	Search, tagging, review forms, rebuttals, PC meeting
ConfTool	Ολοκληρωμένη διαχείριση συνεδρίου	Submission, reviewing, scheduling, registration, payments
EDAS	Συνέδρια, workshops, journals	Submission, review, registration, conference pages
Whova / Eventbrite	Γενική διαχείριση εκδηλώσεων	Εγγραφές, agenda, check-in, networking, εισιτήρια

Πίνακας 1 Ενδεικτική Σύγκριση Υφιστάμενων Λύσεων

1.3 Δομή Εργασίας

Η παρούσα εργασία οργανώνεται σε πέντε βασικά κεφάλαια, με σκοπό την ομαλή παρουσίαση του θεωρητικού υποβάθρου, της ανάλυσης, του σχεδιασμού και της υλοποίησης της εφαρμογής διαχείρισης συνεδρίων.

Στο πρώτο κεφάλαιο παρουσιάζεται η εισαγωγή της εργασίας. Γίνεται αναφορά στο αντικείμενο της εφαρμογής, στην ανάγκη ύπαρξης συστημάτων διαχείρισης συνεδρίων, καθώς και σε υφιστάμενες εφαρμογές που χρησιμοποιούνται για αντίστοιχους σκοπούς. Μέσα από την αναφορά αυτή αναδεικνύονται βασικές λειτουργίες που συναντώνται σε τέτοιου είδους συστήματα, όπως η δημιουργία συνεδρίων, η υποβολή εργασιών, η αξιολόγηση από reviewers και η διαχείριση συμμετοχών.

Στο δεύτερο κεφάλαιο αναλύονται οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Συγκεκριμένα, παρουσιάζεται η γλώσσα προγραμματισμού Python, το πλαίσιο ανάπτυξης εφαρμογών ιστού Django, η βάση δεδομένων SQLite, καθώς και τεχνολογίες που σχετίζονται με την ανταλλαγή δεδομένων, τη δημιουργία διεπαφών χρήστη και την υποστήριξη επιμέρους λειτουργιών. Επιπλέον, γίνεται αναφορά στο JSON API, στην HTML και στη Django Template Language, στο CSS, στο Stripe για την προσομοίωση πληρωμών και στο Mailpit για τη διαχείριση email ειδοποιήσεων σε περιβάλλον ανάπτυξης. Τέλος, παρουσιάζεται η γενική αρχιτεκτονική του συστήματος.

Στο τρίτο κεφάλαιο πραγματοποιείται η ανάλυση και ο σχεδιασμός της εφαρμογής. Περιγράφονται οι λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος, καθώς και οι βασικοί ρόλοι των χρηστών, όπως ο Presenter, ο Reviewer, ο Organizer και ο Superuser ή Staff. Επιπλέον, παρουσιάζονται αναλυτικά οι περιπτώσεις χρήσης της εφαρμογής, οι οποίες καλύπτουν διαδικασίες όπως η συμμετοχή σε συνέδριο, η υποβολή εργασίας, η αξιολόγηση εργασιών, η διαχείριση συνεδρίου και η διαχείριση του συστήματος. Στο ίδιο κεφάλαιο περιλαμβάνεται και η μοντελοποίηση των δεδομένων, όπου αποτυπώνονται οι βασικές οντότητες και οι σχέσεις μεταξύ τους.

Στο τέταρτο κεφάλαιο παρουσιάζεται η υλοποίηση των βασικών λειτουργιών της εφαρμογής. Αναλύεται η διαχείριση συνεδρίων και ρόλων, η διαδικασία υποβολής και αξιολόγησης paper με βάση τον κανόνα ομοφωνίας, καθώς και η ανάθεση track και η ολοκλήρωση πληρωμής. Περιγράφεται επίσης η δημιουργία Checkout Session μέσω Stripe, η αποστολή email ειδοποιήσεων και η χρήση του Mailpit στο περιβάλλον ανάπτυξης. Επιπλέον, παρουσιάζονται οι καταστάσεις και οι μεταβάσεις των βασικών αντικειμένων του συστήματος, καθώς και η πλοήγηση στην εφαρμογή μέσα από τις βασικές οθόνες και λειτουργίες της.

Στο πέμπτο κεφάλαιο διατυπώνονται τα συμπεράσματα της εργασίας. Συνοψίζονται τα αποτελέσματα της ανάπτυξης της εφαρμογής, αξιολογείται ο βαθμός επίτευξης των στόχων που τέθηκαν και αναφέρονται πιθανά σημεία βελτίωσης ή μελλοντικής επέκτασης του συστήματος.

Τέλος, ακολουθεί η βιβλιογραφία, στην οποία παρατίθενται οι πηγές που χρησιμοποιήθηκαν κατά την εκπόνηση της εργασίας, καθώς και το Παράρτημα Α, όπου παρουσιάζεται ενδεικτικός κώδικας της εφαρμογής.

2. Τεχνολογίες και Εργαλεία

Σε αυτό το κεφάλαιο παρουσιάζονται οι τεχνολογίες, τα λογισμικά ανάπτυξης, η βάση δεδομένων, τα frameworks, τα εργαλεία και οι αρχιτεκτονικές προσεγγίσεις που χρησιμοποιήθηκαν κατά τον σχεδιασμό και την ανάπτυξη του Πληροφοριακού Συστήματος Διαχείρισης Συνεδρίων. Η εφαρμογή υλοποιήθηκε με στόχο την υποστήριξη βασικών διαδικασιών ενός ακαδημαϊκού συνεδρίου, όπως η δημιουργία και δημοσίευση συνεδρίων, η υποβολή εργασιών, η ανάθεση αξιολογητών, η καταχώρηση reviews, η ένταξη αποδεκτών εργασιών σε θεματικά tracks, η παρακολούθηση ρόλων χρηστών και η διαχείριση πληρωμών.

2.1 Λογισμικά και Εργαλεία

Σε αυτή την ενότητα, διερευνούμε τα εργαλεία λογισμικού και τις τεχνολογίες που χρησιμοποιούνται για την ανάπτυξη του πληροφοριακού συστήματος, μαζί με τους λόγους πίσω από την επιλογή τους. Η επιλογή των τεχνολογιών που χρησιμοποιήθηκαν σε αυτό το έργο βασίστηκε κυρίως στη σταθερότητά τους, τη διαθεσιμότητα λεπτομερούς τεκμηρίωσης και υποστήριξης από την κοινότητα, και τη δυνατότητα γρήγορης ανάπτυξης μιας δυναμικής διαδικτυακής εφαρμογής που επιτρέπει την ύπαρξη διάφορων ρόλων χρηστών και σύνθετων ροών εργασίας. Οι επιλεγμένες τεχνολογίες είναι πολύ δημοφιλείς στην ανάπτυξη διαδικτυακών εφαρμογών και η χρήση τους εγγυάται επεκτασιμότητα, ασφάλεια, καθώς και ευκολία συντήρησης του κώδικα.

Η γλώσσα προγραμματισμού Python και το διαδικτυακό πλαίσιο Django χρησιμοποιήθηκαν στην ανάπτυξη του συστήματος. Για την ανάπτυξη του συστήματος, χρησιμοποιήθηκε μια σχεσιακή βάση δεδομένων όπου αποθηκεύονται πληροφορίες για χρήστες, συνέδρια, ρόλους ειδικούς για το συνέδριο, θεματικές ενότητες, υποβολές εργασιών, αναθέσεις κριτών, κριτικές, ειδοποιήσεις και πληρωμές. HTML, CSS και JavaScript χρησιμοποιήθηκαν στο επίπεδο παρουσίασης και ένα κατάλληλο πλαίσιο frontend για τις ανάγκες χρησιμοποιήθηκε στη μορφοποίηση και οργάνωση των σελίδων, έτσι ώστε η εφαρμογή να παρέχει ένα περιβάλλον εύκολης και κατανοητής αλληλεπίδρασης για τους χρήστες της.

Σε αυτό το μέρος, κάθε εργαλείο θα παρουσιαστεί με μια σύντομη περιγραφή των βασικών του χαρακτηριστικών, τα πλεονεκτήματα που προσφέρει το εργαλείο στο έργο και τη χρήση του στην εφαρμογή. Η σύνδεση των τεχνολογιών με την υλοποίηση του συστήματος και ο

τρόπος εφαρμογής τους στις διάφορες λειτουργίες της εφαρμογής θα παρουσιαστούν αναλυτικότερα στο επόμενο κεφάλαιο.

2.1.1 Η γλώσσα προγραμματισμού Python

Η Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου κατάλληλη για την ανάπτυξη εφαρμογών, αυτοματισμών, διαδικτυακών συστημάτων, ανάλυσης δεδομένων, καθώς και εκπαιδευτικών εφαρμογών, καθώς είναι μια διερμηνευόμενη γλώσσα, πράγμα που σημαίνει ότι έχει διαδραστική υποστήριξη, και είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που περιλαμβάνει επίσης άλλα μοντέλα προγραμματισμού όπως δομικά και λειτουργικά μοντέλα. (Python Software Foundation, n.d.-a)

Επιπλέον, η Python είναι απλή και εύκολη στην εκμάθηση, καθώς η επίσημη τεκμηρίωση αναφέρει ότι η Python συνδυάζει μεγάλη δύναμη με πολύ σαφές σύνταξη, πράγμα που σημαίνει ότι η Python είναι επίσης κατάλληλη για μικρά σενάρια που λειτουργούν σε μικροκλίμακα, καθώς και σε συστήματα πληροφοριών που λειτουργούν σε μακροκλίμακα. Οι δομές δεδομένων υψηλού επιπέδου, οι μονάδες, οι εξαιρέσεις, η δυναμική πληκτρολόγηση, η δυναμική σύνδεση και οι κλάσεις (Python Software Foundation, n.d.-a) είναι χαρακτηριστικά που βοηθούν τον προγραμματιστή να γράφει εύκολα κώδικα που είναι πιο ευανάγνωστος και μπορεί να γίνει κατανοητός από άλλους. Αυτά είναι ιδιαίτερα χρήσιμα σε μια εφαρμογή όπως αυτή που εξετάζεται σε αυτή την εργασία με πολλαπλά υποσυστήματα όπως αυτό που χρησιμοποιείται από τους χρήστες, τις διασκέψεις, την υποβολή εργασιών από τους κριτές και το πρόγραμμα της διάσκεψης.

Η Python διαθέτει πολύ καλή και εκτενή τεκμηρίωση που περιλαμβάνει οδηγούς και τεκμηρίωση της γλώσσας, βιβλιοθήκης, εγχειρίδιο εγκατάστασης, HOWTOs (ένα έγγραφο που προορίζεται να είναι "ΠΩΣ ΝΑ κάνετε κάτι"), οδηγούς για την εγκατάσταση πακέτων τρίτων, καθώς και την τεκμηρίωση του API Python/C (Python Software Foundation, n.d.-b) που μπορεί να βοηθήσει έναν προγραμματιστή να γράψει τον κώδικα πιο γρήγορα. Επιπλέον, υπάρχει η τυπική βιβλιοθήκη της Python που έχει πολλές μονάδες με προ-προγραμματισμένη και έτοιμη λειτουργικότητα, όπως η είσοδος/έξοδος αρχείων (Python Software Foundation, n.d.-c), η χειραγώγηση δεδομένων, το διαδίκτυο και το HTTP, τα email, οι βάσεις δεδομένων, οι δοκιμές, η καταγραφή κ.λπ. Η διαθεσιμότητα τέτοιων τυπικών βιβλιοθηκών βοηθά τον προγραμματιστή, ο οποίος δεν χρειάζεται να γράψει

κώδικα για να καλύψει βασικές λειτουργίες που μπορούν να βρεθούν στην προαναφερθείσα βιβλιοθήκη.

Η Python συχνά συνδυάζεται με ένα από τα δημοφιλή της πλαίσια ιστού, όπως το Django, για την ανάπτυξη διαδικτυακών εφαρμογών. Το Django είναι ένα πλαίσιο ιστού Python υψηλού επιπέδου που επιτρέπει την ταχεία ανάπτυξη ισχυρού και καθαρού λογισμικού (Django Software Foundation, n.d.). Οι προαναφερθέντες λόγοι και η άμεση χρήση με το Django καθιστούν την Python την πιο κατάλληλη γλώσσα για την ανάπτυξη ενός συστήματος διαχείρισης συνεδρίων, το οποίο έχει ένα σαφές backend για τους χρήστες, τους ρόλους, την υποβολή, τις κριτικές και τη διαχείριση.

Η χρήση εικονικών περιβαλλόντων είναι ένα άλλο χαρακτηριστικό που βοηθά τον προγραμματιστή. Αυτό σημαίνει ότι κάθε έργο θα εγκατασταθεί ξεχωριστά σε έναν φάκελο, ώστε να μην υπάρχει σύγκρουση μεταξύ δύο διαφορετικών πακέτων για διαφορετικές εκδόσεις σε ένα έργο (Python Software Foundation, n.d.-d), και επίσης όταν θέλετε να χρησιμοποιήσετε έναν άλλο υπολογιστή ή μηχανή, θα μπορείτε να αναδημιουργήσετε εύκολα το περιβάλλον ανάπτυξης.

Η γλώσσα Python έχει σχεδιαστεί για να ενθαρρύνει την απλότητα και την αναγνωσιμότητα του πηγαίου κώδικα. Αυτές οι αρχές αντικατοπτρίζονται καλά στη φιλοσοφία της Python, ιδίως στο PEP 20 The Zen of Python, το οποίο τονίζει μια σειρά αρχών όπως: "Το απλό είναι καλύτερο από το πολύπλοκο", "Η αναγνωσιμότητα μετράει", "Το ρητό είναι καλύτερο από το υπονοούμενο" (Peters, 2004). Είναι, επομένως, μια προφανής επιλογή για τη γλώσσα προγραμματισμού Python σε μια εκπαιδευτική ή ακαδημαϊκή εφαρμογή λογισμικού, καθώς ο κώδικας του λογισμικού πρέπει να γίνεται όσο το δυνατόν περισσότερο κατανοητός για να επιτευχθεί η επιθυμητή λειτουργικότητά του, η οποία, σε αυτή την περίπτωση, είναι το σύστημα εγγραφής συνεδρίων.

Για την ανάπτυξη αυτής της εργασίας, η γλώσσα προγραμματισμού Python επιλέγεται ως η πιο κατάλληλη. Συγκεκριμένα, λόγω της απλής σύνταξής της και της μεγάλης και συνεχώς εξελισσόμενης κοινότητας της Python, της παρουσίας λεπτομερούς τεκμηρίωσης και, τελευταίο αλλά όχι λιγότερο σημαντικό, του συνδυασμού με το πλαίσιο ιστού Django, η Python είναι η ιδανική επιλογή για την ανάπτυξη μιας εφαρμογής της προαναφερθείσας δομής. Τέλος, με τη γλώσσα προγραμματισμού Python, υλοποιούνται όλες οι κύριες οντότητες και λειτουργίες της εφαρμογής, δηλαδή: η διαχείριση χρηστών και η εγγραφή συνεδρίων των εργασιών τους, η υποβολή εργασιών, η ανάθεση κριτικών και η παραγωγή κατάλληλων αποτελεσμάτων για τους διαχειριστές συστήματος και τους χρήστες.

2.1.2 Το πλαίσιο ανάπτυξης εφαρμογών ιστού Django

Το Django είναι ένα πλαίσιο ιστού βασισμένο στην Python. Ο ιστότοπος του Django αναφέρει ότι στοχεύει στο "να βοηθήσει τους προγραμματιστές να δημιουργούν γρήγορα διαδικτυακές εφαρμογές, πράγμα που σημαίνει λιγότερο κώδικα, καλύτερες διαδικτυακές εφαρμογές και ταχύτερη ανάπτυξη και πραγματιστικό σχεδιασμό." Το Django είναι δωρεάν και ανοιχτού κώδικα, πράγμα που επιτρέπει ερευνητικά και ακαδημαϊκά έργα, καθώς και την ανάπτυξη νέων, δωρεάν εφαρμογών. Αυτό το πλαίσιο επιτρέπει σε φοιτητές και ερευνητές να μελετήσουν, να προσαρμόσουν και να επεκτείνουν το Django.

Ο επίσημος ιστότοπος του Django παραθέτει έτοιμα στοιχεία λειτουργικότητας του Django, όπως έλεγχο ταυτότητας, διεπαφή διαχειριστή και διαχείριση περιεχομένου. Το Django είναι χτισμένο με την παραδοχή ότι ένας προγραμματιστής ιστοτόπων έχει πολλά κοινά χαρακτηριστικά και μηχανισμούς που έχουν ήδη υλοποιηθεί σε ένα πλαίσιο. Είναι σημαντικό για τους προγραμματιστές του Django να επικεντρωθούν στα συγκεκριμένα χαρακτηριστικά και τη λογική της εφαρμογής αντί για τα γενικά και επαναλαμβανόμενα ζητήματα ανάπτυξης ιστού, και το Django έχει σχεδιαστεί ακριβώς για αυτό. Ένα από τα βασικά στοιχεία αυτού του έργου είναι ότι υπάρχουν πολλά κοινά χαρακτηριστικά και λειτουργίες, που υπάρχουν ήδη στο Django και δεν χρειάζονται επιπλέον προσπάθεια προγραμματισμού: σύνδεση και αποσύνδεση χρήστη, ρόλοι, διαχείριση χρηστών, υποβολές, κριτικές, διαχείριση περιεχομένου, κ.λπ.

Το Django ακολουθεί την αρχιτεκτονική MVC. Η επίσημη τεκμηρίωσή του εξηγεί ότι το Django ακολουθεί το μοντέλο MTV. ενώ είναι παρόμοιο με το μοντέλο MVC, στο MTV, το μοντέλο αφορά τα δεδομένα, το πρότυπο είναι ένας μηχανισμός εμφάνισης δεδομένων στον χρήστη, και η προβολή είναι μια συνάρτηση ή κλάση Python που καθορίζει τα δεδομένα που θα εμφανιστούν. Για ένα σύστημα πληροφοριών, είναι σημαντικό να είναι δυνατή η διαχωρισμός της λογικής του συστήματος από την παρουσίασή του, πράγμα που διευκολύνει τη συντήρηση και την περαιτέρω επέκταση του συστήματος.

Ένα μοντέλο βάσης δεδομένων (μοντέλο Django) είναι μια κλάση Python που κληρονομεί από το `django.db.models.Model`, και κάθε χαρακτηριστικό της κλάσης είναι ένα πεδίο βάσης δεδομένων (Django Software Foundation, n.d.-d). Για παράδειγμα, μια διάσκεψη μπορεί να έχει ένα χαρακτηριστικό τίτλο. Μέσω των μοντέλων Django, οι οντότητες δεδομένων εφαρμογής μπορούν να περιγραφούν σε Python, όπως Διάσκεψη, Εργασία,

Κριτής, Κριτική και Συμμετέχων. Επιπλέον, το Django ORM παρέχει ένα API αντιστοίχισης αντικειμένων-σχεσιακών για πρόσβαση σε βάσεις δεδομένων που δεν απαιτεί άμεσες εντολές SQL και επιτρέπει την άμεση χειραγώγηση οντοτήτων στο μοντέλο μέσω των αντικειμένων Python τους.

Το Django υλοποιεί ένα σύστημα προτύπων για το επίπεδο παρουσίασης του πλαισίου. Το Django παρέχει μια ειδική γλώσσα σήμανσης για την περιγραφή δεδομένων εφαρμογής στον χρήστη και ενθαρρύνει το διαχωρισμό της επιχειρηματικής λογικής και της παρουσίασης. Τα πρότυπα είναι ένα σύνολο δεδομένων που μπορούν να διαχειριστούν από κάποιον που δεν χρειάζεται απαραίτητα να έχει γνώσεις Python. Μπορούν να χρησιμοποιήσουν μόνο HTML. Για να αποδώσει ο μηχανισμός προτύπων τον κώδικά του, ο προγραμματιστής μπορεί να γράψει κώδικα σε HTML που χρησιμοποιεί πρότυπα με τα αντίστοιχα δεδομένα τους (Django Software Foundation, n.d.-e). Ο προγραμματιστής απλώς πρέπει να περιγράψει τον κώδικα HTML που χρησιμοποιεί τα πρότυπα, χωρίς να χρειάζεται να περιγράψει τα ίδια τα πρότυπα.

Σε μια προβολή, λαμβάνεται ένα αίτημα ιστού και επιστρέφεται μια απάντηση ιστού (Django Software Foundation, n.d.-f). Σύμφωνα με την τεκμηρίωση, μια προβολή μπορεί να επιστρέψει HTML, ανακατεύθυνση, μήνυμα σφάλματος, XML, εικόνα ή οποιοδήποτε άλλο είδος απάντησης. Σε μια εφαρμογή διαχείρισης συνεδρίων, μια προβολή μπορεί να εμφανίζει τη λίστα των συνεδρίων, μια φόρμα υποβολής, κριτικές και ένα πρόγραμμα.

Το Django δημιουργεί αυτόματα μια διεπαφή διαχειριστή με βάση τα μοντέλα. Μέσω αυτής της διεπαφής, ένας προγραμματιστής ή διαχειριστής ιστότοπου μπορεί να διαχειριστεί το περιεχόμενο του συστήματος (Django Software Foundation, n.d.-g). Σε αυτήν την εργασία, η διεπαφή διαχειριστή μπορεί να χρησιμοποιηθεί για τη διαχείριση δεδομένων όπως Συνέδρια, Θέματα, Χρήστες, Υποβολές, Κριτικές, Συμμετέχοντες, κ.λπ. Ωστόσο, το Django δεν προορίζεται για αυτό, σύμφωνα με την επίσημη τεκμηρίωση, καθώς ο διαχειριστής δεν είναι μια κατάλληλη διεπαφή εφαρμογής (Django Software Foundation, n.d.-g).

Το Django έρχεται με ένα σύστημα ελέγχου ταυτότητας και εξουσιοδότησης. Με άλλα λόγια, το σύστημα περιέχει λειτουργίες για την εκτέλεση βασικών λειτουργιών διαχείρισης χρηστών, όπως εγγραφή λογαριασμών, σύνδεση και αποσύνδεση, έλεγχος δικαιωμάτων και φόρμα επαναφοράς κωδικού πρόσβασης (Django Software Foundation, n.d.-h). Επιπλέον, το Django δεν αποθηκεύει κωδικούς πρόσβασης σε απλό κείμενο, μόνο ως κατακερματισμούς (Django Software Foundation, n.d.-h), κάτι που αποτελεί σημαντικό χαρακτηριστικό ασφαλείας για εφαρμογές που επιτρέπουν στους χρήστες να συνδέονται

και να εγγράφονται. Είναι σημαντικό να σημειωθεί ότι σε ένα σύστημα διαχείρισης συνεδρίων, οι χρήστες μπορεί να έχουν διαφορετικά δικαιώματα, όπως αυτά του διαχειριστή, του διοργανωτή, του συγγραφέα, του κριτή και του συμμετέχοντα.

Το Django δίνει μεγάλη προσοχή στην ασφάλεια των διαδικτυακών εφαρμογών. Σύμφωνα με την επίσημη τεκμηρίωση, η ασφάλεια είναι ένα πολύ σημαντικό μέρος της ανάπτυξης διαδικτυακών εφαρμογών (Django Software Foundation, n.d.-i). Για το λόγο αυτό, το Django έρχεται με εργαλεία και μηχανισμούς που προσφέρουν προστασία από κοινές επιθέσεις, όπως SQL injection, cross-site scripting, cross-site request forgery, clickjacking (Django Software Foundation, n.d.-i). Για την εφαρμογή που αναπτύχθηκε σε αυτήν την εργασία, αυτό το χαρακτηριστικό είναι πολύ σημαντικό, καθώς το σύστημα διαχειρίζεται προσωπικά δεδομένα, αρχεία υποβολής, κριτικές, αποδοχή ή απόρριψη υποβολών, μεταξύ άλλων πληροφοριών.

Το Django είναι η επιλογή για την ανάπτυξη του έργου σε αυτήν την εργασία, καθώς προσφέρει ταχεία ανάπτυξη, δομή παρόμοια με MVC, ORM, πρότυπα, διεπαφή διαχειριστή, διαχείριση χρηστών, μηχανισμούς ασφαλείας και δωρεάν άδεια ανοιχτού κώδικα. Λόγω των χαρακτηριστικών του, είναι κατάλληλο για την ανάπτυξη ενός συστήματος όπως ένα σύστημα διαχείρισης συνεδρίων που απαιτεί οργανωμένη διαχείριση δεδομένων και πολλαπλούς ρόλους για τους χρήστες. Επιπλέον, τα χαρακτηριστικά του Django επιτρέπουν την επέκταση και βελτίωση του συστήματος στο μέλλον.

2.1.3 Η βάση δεδομένων SQLite

Το SQLite είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που ξεχωρίζει από τις παραδοσιακές βάσεις δεδομένων πελάτη-εξυπηρετητή, όπως η MySQL και η PostgreSQL. Ο επίσημος οδηγός ορίζει το SQLite ως μια βιβλιοθήκη εντός της διαδικασίας που αποτελεί μια αυτόνομη, χωρίς εξυπηρετητή, μηδενικής ρύθμισης και συναλλακτική μηχανή βάσης δεδομένων SQL (SQLite, n.d.-a). Με άλλα λόγια, εκτελείται ως βιβλιοθήκη απευθείας στη διαδικασία της εφαρμογής. Δεν χρειάζεστε ξεχωριστό διακομιστή βάσης δεδομένων, εγκατάσταση υπηρεσίας ή οποιαδήποτε ειδική ρύθμιση.

Το SQLite διατηρεί επίσης ολόκληρη τη βάση δεδομένων σε ένα μόνο αρχείο δίσκου. Αυτό το αρχείο μπορεί να περιέχει τους πίνακες, τους δείκτες, τις ενεργοποιήσεις και τις προβολές της βάσης δεδομένων και αποθηκεύεται σε μια ανεξάρτητη από την πλατφόρμα μορφή

αρχείου, η οποία μπορεί να αντιγραφεί από έναν υπολογιστή σε άλλον (SQLite, n.d.-a). Λόγω αυτού, το SQLite χρησιμοποιείται συχνά για μικρές και μεσαίες εφαρμογές, για εκπαιδευτικούς σκοπούς, ως μέρος πρωτοτύπων συστημάτων και σε εφαρμογές που απαιτούν εύκολη εγκατάσταση και απλές λειτουργίες αντιγραφής δεδομένων μεταξύ περιβαλλόντων.

Η κύρια διαφορά σε σύγκριση με άλλα συστήματα διαχείρισης βάσεων δεδομένων είναι ότι το SQLite δεν έχει ξεχωριστή διαδικασία εγκατάστασης. Δεν χρειάζεται να δημιουργήσετε διακομιστή βάσης δεδομένων, λογαριασμό χρήστη ή ρύθμιση ελέγχου πρόσβασης. Ο επίσημος οδηγός τονίζει ρητά τον όρο "μηδενική ρύθμιση" για το SQLite, επειδή δεν υπάρχει ξεχωριστός διακομιστής βάσης δεδομένων που να χρειάζεται να ξεκινήσει, να σταματήσει ή να ρυθμιστεί (SQLite, n.d.-b). Αυτό το καθιστά ιδανικό για την τοπική ανάπτυξη εφαρμογών όταν απαιτείται μια βάση δεδομένων έτοιμη προς χρήση χωρίς πρόσθετη διαχειριστική ρύθμιση.

Το SQLite υποστηρίζει τις τυπικές λειτουργίες ενός συστήματος σχεσιακών βάσεων δεδομένων: συναλλαγές ACID, δηλώσεις SQL, δείκτες, προβολές και ενεργοποιήσεις. Η σελίδα "χαρακτηριστικά" επιβεβαιώνει τις συναλλαγές ACID του συστήματος βάσης δεδομένων: ατομικές, συνεπείς, απομονωμένες και ανθεκτικές συναλλαγές (SQLite, n.d.-c). Το SQLite διαθέτει επίσης μια πλήρη υλοποίηση SQL με επεκτάσεις όπως μερικοί δείκτες, δείκτες σε εκφράσεις, JSON, κοινές εκφράσεις πινάκων και συναρτήσεις παραθύρου.

Είναι σύνηθες να χρησιμοποιείται το SQLite σε εκπαιδευτικές εφαρμογές και σε συστήματα δοκιμών ή επίδειξης. Σύμφωνα με τον επίσημο οδηγό, συνιστάται ως κατάλληλη λύση για διδασκαλία και εκπαίδευση, επειδή μπορεί να χρησιμοποιηθεί εύκολα και επιτρέπει την εύκολη δημιουργία πολλαπλών βάσεων δεδομένων. Επιπλέον, θεωρείται μια καλή επιλογή για δοκιμές ή επιδείξεις, επειδή δεν απαιτεί πρόσθετη εγκατάσταση και η βάση δεδομένων είναι άμεσα διαθέσιμη με ένα μόνο αρχείο βάσης δεδομένων (SQLite, n.d.-d).

Η βάση δεδομένων SQLite μπορεί επίσης να προσπελαστεί χρησιμοποιώντας την ενότητα `sqlite3` στην τυπική βιβλιοθήκη της Python. Η τεκμηρίωση της Python αναφέρει ότι η ενότητα `sqlite3` παρέχει μια διεπαφή SQL συμβατή με την Προδιαγραφή Διεπαφής Βάσης Δεδομένων Python v2.0 και επιτρέπει τη χρήση βάσεων δεδομένων SQLite σε εφαρμογές Python (Python Software Foundation, n.d.). Έτσι, η βάση δεδομένων μπορεί να χρησιμοποιηθεί στην εφαρμογή απευθείας χωρίς την εγκατάσταση οποιουδήποτε ξεχωριστού προγράμματος οδήγησης βάσης δεδομένων.

Το SQLite λειτουργεί επίσης με το Django. Η τεκμηρίωση δείχνει την υποστήριξή του για τη βάση δεδομένων στο πλαίσιο και παρέχει τη χρήση του backend `django.db.backends.sqlite3` για τη σύνδεση της εφαρμογής με τη βάση δεδομένων SQLite (Django Software Foundation, n.d.-a). Έτσι, η βάση δεδομένων μπορεί να χρησιμοποιηθεί βολικά για την ανάπτυξη μιας εφαρμογής Django, καθώς είναι εύκολη στη ρύθμιση και χρήση αμέσως, χωρίς την εγκατάσταση ενός πρόσθετου διακομιστή βάσης δεδομένων. Μπορείτε να δημιουργήσετε μοντέλα, μεταναστεύσεις και βασικές λειτουργίες ως μέρος των πρώιμων σταδίων ανάπτυξης.

Ωστόσο, η βάση δεδομένων έχει ορισμένους περιορισμούς που πρέπει επίσης να ληφθούν υπόψη. Η τεκμηρίωση δηλώνει ρητά ότι το SQLite είναι μια ελαφριά βάση δεδομένων και ότι η ταυτοχρονισμός δεν κλιμακώνεται καλά σε αυτήν τη βάση δεδομένων. Το σφάλμα λειτουργίας: η βάση δεδομένων είναι κλειδωμένη, προκύπτει σε περίπτωση υψηλού ταυτοχρονισμού, όταν ένα συγκεκριμένο νήμα ή διαδικασία έχει αποκλειστική κλειδαριά βάσης δεδομένων και μια άλλη μπλοκάρεται, προσπαθώντας να την αποκτήσει (Django Software Foundation, n.d.-b). Εάν η εφαρμογή προορίζεται για χρήση από πολλούς χρήστες ή πρόκειται να εκτελεστεί σε παραγωγή υπό σημαντικά αυξημένο φορτίο, τότε η επιλογή θα ήταν μια πιο σύνθετη βάση δεδομένων πελάτη-εξυπηρετητή, όπως η PostgreSQL.

Για τους σκοπούς αυτής της μελέτης, το SQLite μπορεί να χρησιμοποιηθεί ως βάση δεδομένων κατά την ανάπτυξη και δοκιμή του συστήματος διαχείρισης συνεδρίων. Η επιλογή αυτής της βάσης δεδομένων είναι κατάλληλη για ένα εκπαιδευτικό και δοκιμαστικό περιβάλλον λόγω της απλής εγκατάστασής της, της ενσωμάτωσής της με το Django και του γεγονότος ότι η διαχείριση των δεδομένων της γίνεται μέσω συστήματος αρχείων. Με το SQLite, μπορούν να αποθηκευτούν οι κύριες οντότητες του συστήματος—συνέδρια, χρήστες, υποβολές, κριτικές, θεματικές συνεδρίες και εγγραφή συμμετεχόντων. Όταν η ανάπτυξη ή η χρήση του συστήματος γίνει αρκετά σημαντική, αυτή είναι μια ευκαιρία να μεταβείτε σε μια πιο σύνθετη βάση δεδομένων, αξιοποιώντας τις δυνατότητες που προσφέρει το πλαίσιο Django.

2.1.4 JSON API

Οι διεπαφές JSON API είναι μια μορφή επικοινωνίας μεταξύ των στοιχείων μιας διαδικτυακής εφαρμογής ή μεταξύ διαφορετικών συστημάτων. Πρακτικά, ένα API επιτρέπει σε μια εφαρμογή να εκθέτει δεδομένα και λειτουργικότητα σε άλλες εφαρμογές

ή σε διαφορετικά περιβάλλοντα για χρήση, όπως διαδικτυακά frontends, εφαρμογές για κινητά ή υπηρεσίες τρίτων. Η επικοινωνία δεδομένων σε μορφή JSON σημαίνει την ανταλλαγή δεδομένων σε μια μορφή που είναι απλή, ευανάγνωστη και ελαφριά, και κατανοητή τόσο από υπολογιστές όσο και από ανθρώπους.

Το JSON σημαίνει JavaScript Object Notation και περιγράφεται ως μια ελαφριά, βασισμένη σε κείμενο, και ανεξάρτητη από γλώσσα προγραμματισμού μορφή ανταλλαγής δεδομένων. Περιγράφεται ως μια δομημένη μορφή δεδομένων για ανταλλαγή δεδομένων από το RFC 8259, ενώ το ECMA-404 το περιγράφει ως μια ελαφριά σύνταξη για μορφή ανταλλαγής δεδομένων (Bray, 2017; Ecma International, 2017). Παρόλο που το JSON έχει την προέλευσή του στη JavaScript, σήμερα υποστηρίζεται από μια σειρά γλωσσών προγραμματισμού όπως Python, Java, PHP, C#, Go και η ίδια η Javascript, και επομένως είναι κατάλληλο για διαδικτυακές εφαρμογές και APIs.

Στο JSON, οι πληροφορίες οργανώνονται με τη μορφή ζευγών ονόματος-τιμής και μιας διατεταγμένης λίστας τιμών, η οποία επιτρέπει την αναπαράσταση τόσο απλών όσο και πιο σύνθετων δομών πληροφοριών, όπως χρήστες, συνέδρια, υποβολές εργασιών, κριτικές ή εγγραφές. Έτσι, μια υποβληθείσα εργασία για ένα συνέδριο μπορεί να αναπαρασταθεί ως ένα αντικείμενο JSON που περιέχει ιδιότητες όπως τίτλος, περίληψη, συγγραφείς, θέμα και κατάσταση κριτικής, και η μορφή μπορεί να είναι κατάλληλη για επικοινωνία δεδομένων με το frontend, για παράδειγμα για την αμφίδρομη επικοινωνία μεταξύ backend και frontend, και με μελλοντικές εξωτερικές υπηρεσίες.

Η επικοινωνία ενός JSON API γίνεται συχνά μέσω HTTP. Το HTTP ορίζει τη σημασιολογία για το αίτημα και την απόκριση, καθώς και τις μεθόδους που μπορούν να χρησιμοποιηθούν για την ανάγνωση, δημιουργία, ενημέρωση ή διαγραφή των δεδομένων. Το πρότυπο ορίζει αυτές ως GET, POST, PUT, PATCH και DELETE, αντίστοιχα (Fielding et al., 2022). Επιπλέον, οι κωδικοί κατάστασης HTTP μπορούν να οριστούν για διαφορετικά αποτελέσματα, για παράδειγμα στην περίπτωση επιτυχημένης εκτέλεσης, δημιουργίας πόρου, κακού αιτήματος ή μη εξουσιοδοτημένης πρόσβασης.

Η επιστροφή δεδομένων JSON είναι δυνατή μέσω του JsonResponse σε περιβάλλον Django, και αυτή η κλάση απόκρισης JSON είναι μια υποκλάση του HttpResponse, «σχεδιασμένη να επιστρέφει δεδομένα κωδικοποιημένα σε JSON ως μέρος της απόκρισης στον καλούντα» (Django Software Foundation, n.d.-a). Οι αποκρίσεις JSON όπως λίστες συνεδρίων, πληροφορίες υποβολής εργασιών ή αποτελέσματα αναζήτησης μπορεί να είναι

κατάλληλες για τη χρήση ενός αντικειμένου απόκρισης σε ένα backend Django, όπου μπορεί να χρησιμοποιηθεί το JsonResponse.

Για πιο προηγμένη ανάπτυξη APIs στο Django, χρησιμοποιείται συχνά το Django REST Framework. Το Django REST Framework παρέχει πολλά εργαλεία για τη δημιουργία Web APIs, όπως σειριοποιητές, views API, πιστοποιήσεις, δικαιώματα, κλάσεις απόκρισης και κωδικούς κατάστασης. Συγκεκριμένα, οι σειριοποιητές μπορούν να χρησιμοποιηθούν για τη μετατροπή σύνθετων αντικειμένων, π.χ., στιγμιότυπα μοντέλων Django και querysets, σε μια αναπαράσταση που μπορεί να μετατραπεί σε JSON, και μπορούν να χρησιμοποιηθούν για τη μετατροπή των εισερχόμενων δεδομένων σε αντικείμενα Python αφού ελεγχθεί ότι τα δεδομένα είναι έγκυρα (Django REST Framework, n.d.-a). Το αίτημα και η απόκριση είναι επίσης κρίσιμες πτυχές ενός JSON API. Το Django REST Framework παρέχει αντικείμενα Request και Response, τα οποία βοηθούν στην επεξεργασία των δεδομένων που αποστέλλονται από τον χρήστη και στην απόκριση στο αίτημα σε διάφορες μορφές ανάλογα με το περιεχόμενο του αιτήματος από τον πελάτη (Django REST Framework, n.d.-b, Django REST Framework, n.d.-c). Από αυτή την άποψη, το backend μιας εφαρμογής θα λαμβάνει δεδομένα από μια φόρμα ή αιτήματα JSON, και στη συνέχεια θα παρέχει μια δομημένη απόκριση στο frontend ή σε μια εφαρμογή.

Στο πλαίσιο μιας εφαρμογής διαχείρισης συνεδρίων, ένα JSON API θα μπορούσε να χρησιμοποιηθεί για διάφορους σκοπούς, συμπεριλαμβανομένης της παροχής endpoints API για προβολή συνεδρίων, λήψη θέματος, υποβολή εργασιών, διανομή εργασιών σε κριτές, αποθήκευση κριτικών, παρουσίαση του προγράμματος του συνεδρίου ή λήψη αναφοράς. Έτσι, ένα σύστημα δεν περιορίζεται απαραίτητα στην απόδοση από την πλευρά του διακομιστή, αλλά θα μπορούσε να εξελιχθεί για να υποστηρίξει δυναμική λειτουργικότητα frontend ή μια εφαρμογή για κινητά στο μέλλον.

Μια περαιτέρω ανησυχία θα ήταν η πτυχή ασφάλειας του API, καθώς ένα JSON API είναι ικανό να παρέχει τις λειτουργίες για τη δημιουργία, λήψη ή τροποποίηση δεδομένων, οπότε πρέπει να υπάρχει ένας τρόπος να προσδιοριστεί ποιος χρήστης μπορεί να λάβει ή να τροποποιήσει δεδομένα σε ένα συγκεκριμένο endpoint, χρησιμοποιώντας μια λειτουργία πιστοποίησης και δικαιωμάτων για την έγκριση ή την απόρριψη ενός αιτήματος (Django REST Framework, n.d.-d, Django REST Framework, n.d.-e). Αυτό είναι ιδιαίτερα σημαντικό σε ένα σύστημα διαχείρισης συνεδρίων όπου υπάρχουν διαφορετικοί τύποι χρηστών (διαχειριστής, διοργανωτής, συγγραφέας, κριτής, συμμετέχων) και πρέπει να μπορούν να έχουν πρόσβαση στην εφαρμογή με διαφορετικούς τρόπους.

Απαιτείται περαιτέρω συζήτηση για μια πιο συγκεκριμένη μορφή, το JSON:API. Είναι μια προδιαγραφή για τον καθορισμό του πώς ο πελάτης θα λάβει έναν πόρο και πώς ο διακομιστής θα του απαντήσει σε JSON. Η προδιαγραφή JSON:API (JSON:API, n.d.) μπορεί να είναι επωφελής για μια πιο τυποποιημένη και προβλέψιμη απόκριση. Ωστόσο, σε μια μικρή και εκπαιδευτική ή ακαδημαϊκή εφαρμογή, μια πιο απλοϊκή απόκριση JSON μπορεί να προτιμηθεί εφόσον είναι συνεπής και εύκολα κατανοητή.

Τέλος, η τεκμηρίωση API είναι ένα πολύ σημαντικό πράγμα, ειδικά σε μεγαλύτερα έργα. και η προδιαγραφή OpenAPI μπορεί να χρησιμοποιηθεί για τη δημιουργία αυτής της τεκμηρίωσης. Το έγγραφο προδιαγραφών OpenAPI παραθέτει όλα τα μονοπάτια, τις μεθόδους, τις παραμέτρους, το σώμα των αιτημάτων και τις αποκρίσεις ενός API που μπορεί να είναι σε μορφή JSON ή YAML (OpenAPI Initiative, n.d.). Η πλήρης τεκμηρίωση OpenAPI ενδέχεται να μην είναι αυστηρά απαραίτητη σε κάθε μικρή εφαρμογή, αλλά θα μπορούσε να είναι μια χρήσιμη επέκταση στο μέλλον, ειδικά σε ένα πλαίσιο όπου η εφαρμογή θα μπορούσε να χρησιμοποιηθεί από εξωτερικές εφαρμογές ή προγραμματιστές τρίτων.

Στο πλαίσιο αυτής της εργασίας, το JSON API χρησιμοποιείται για να χειριστεί την επικοινωνία μεταξύ του backend και frontend. Η εφαρμογή διαχείρισης συνεδρίων θα μπορούσε ακόμα να βασίζεται στα πρότυπα Django, αλλά, χρησιμοποιώντας JSON API, το σύστημα γίνεται πιο ευέλικτο και επεκτάσιμο καθώς θα μπορεί να παρέχει δυναμικές σελίδες, ένα ταμπλό για τους συγγραφείς, τους διοργανωτές και τους κριτές, θα μπορεί να αποτελέσει τη βάση μιας εφαρμογής για κινητά, ή να επιτρέψει την ενσωμάτωση με άλλες εφαρμογές στο μέλλον, όλα αυτά χωρίς να χρειάζεται να αλλάξει πλήρως το σχήμα της βάσης δεδομένων και την γενική λογική του συστήματος.

2.1.4 HTML & Django Template Language

Η HTML (HyperText Markup Language) είναι η κύρια γλώσσα σήμανσης που προορίζεται για τη δημιουργία ιστοσελίδων. Με αυτήν, είναι δυνατόν να οριστεί η δομή της σελίδας και να προστεθεί περιεχόμενο όπως επικεφαλίδες, μπλοκ παραγράφων, υπερσύνδεσμοι, φόρμες, πίνακες, εικόνες και κουμπιά στη δομή. Ωστόσο, από την άποψη της λογικής εκτέλεσης, είναι μια μη-γλώσσα προγραμματισμού και χρησιμοποιείται μόνο για τον ορισμό και τη διάταξη του περιεχομένου της σελίδας όπως εμφανίζεται στο πρόγραμμα περιήγησης του πελάτη (MDN Web Docs, n.d.).

Για αυτήν την εφαρμογή, η γλώσσα HTML χρησιμοποιήθηκε για την υλοποίηση μιας βασικής δομής σελίδας του CMIS. Πιο συγκεκριμένα, η γλώσσα χρησιμοποιήθηκε για την ανάπτυξη της δομής των ακόλουθων μερών: της αρχικής σελίδας, των φορμών σύνδεσης και εγγραφής, των σελίδων που εμφανίζουν συνέδρια και των φορμών υποβολής εργασιών (περίληψη, τίτλος κ.λπ.), των ταμπλό για τους χρήστες μετά την εγγραφή, των λιστών των υποβληθεισών εργασιών, των φορμών που χρησιμοποιούνται για την αξιολόγηση εργασιών από ανατεθειμένους αξιολογητές, των σελίδων που ορίζουν την ανάθεση αξιολογητών για κάθε εργασία σε θέματα και των σελίδων ενός χρήστη που εμφανίζουν την πρόοδο των πληρωμών.

Αξίζει να αναφερθεί ότι η HTML λειτουργεί ως το επίπεδο παρουσίασης για τα δεδομένα που μεταδίδονται στον χρήστη από την ιστοσελίδα μέσω του backend. Για παράδειγμα, οι πληροφορίες του συνεδρίου, όπως το όνομα του συνεδρίου, η περιγραφή, οι ημερομηνίες, η προθεσμία υποβολής, τα αποδεκτά θέματα και οι υποβληθείσες εργασίες εμφανίζονται στους χρήστες μέσω κατάλληλων ετικετών HTML για κάθε στοιχείο. Επίσης, τα πεδία εισαγωγής για τις φόρμες που χρησιμοποιούνται για την υποβολή, όπως τίτλος, περίληψη, επιλεγμένα θέματα, λέξεις-κλειδιά και αρχείο, είναι ετικέτες HTML.

Σε συνδυασμό με την HTML, χρησιμοποιείται η Django Template Language (DTL), η οποία είναι ένας ενσωματωμένος μηχανισμός για τη δημιουργία δυναμικών προτύπων. Με αυτήν, επιτρέπεται η χρήση μεταβλητών, βρόχων, συνθηκών και φίλτρων στον κώδικα μιας σελίδας που είναι γραμμένη σε HTML. Στην πράξη, με τη βοήθεια της DTL, κάθε σελίδα της εφαρμογής εμφανίζει δεδομένα δυναμικά με βάση το περιεχόμενο της βάσης δεδομένων (Django Software Foundation, n.d.). Κατά συνέπεια, κάθε χρήστης, ανάλογα με τον τύπο του συνεδρίου του και τον ρόλο του σε αυτό, καθώς και την κατάσταση των υποβολών σε κάθε ένα από τα συνέδρια, βλέπει διαφορετικό περιεχόμενο για κάθε σελίδα.

Η υλοποίηση κάθε ενός από τους ρόλους στην εφαρμογή καθίσταται δυνατή με τη βοήθεια της DTL, καθώς παρέχει έναν τρόπο για την εμφάνιση περιεχομένου υπό συνθήκη σε κάθε ρόλο. Για παράδειγμα, οι μη συνδεδεμένοι χρήστες βλέπουν μόνο τα δημοσιευμένα συνέδρια και τις πληροφορίες σχετικά με αυτά, ενώ οι συνδεδεμένοι χρήστες βλέπουν επιπλέον επιλογές ενεργειών που τους ταιριάζουν, όπως η ενέργεια προσθήκης του συνεδρίου σε μια λίστα παρακολούθησης, η δυνατότητα εγγραφής τους ως παρατηρητές

στο συνέδριο και η δυνατότητα υποβολής μιας εργασίας για αξιολόγηση. Ως συνέπεια, ένας διοργανωτής βλέπει μια επιλογή για τον ορισμό των θεμάτων και των αξιολογητών αυτών των θεμάτων, ενώ ένας αξιολογητής βλέπει τη λίστα των εργασιών που του έχουν ανατεθεί για αξιολόγηση μέσω του κατάλληλου ρόλου. Αυτές οι συνθήκες στα πρότυπα επιτρέπουν τη δημιουργία μιας μοναδικής εμπειρίας ιστού προσαρμοσμένης σε κάθε ρόλο σε κάθε συνέδριο.

Ένας χρήσιμος μηχανισμός της γλώσσας προτύπων ήταν η χρήση της για τη δημιουργία επαναχρησιμοποιήσιμου περιεχομένου για κάθε σελίδα στην εφαρμογή. Για παράδειγμα, το πρότυπο βάσης της εφαρμογής, οι επικεφαλίδες σελίδων, οι γραμμές πλοήγησης και ορισμένες επαναχρησιμοποιήσιμες ενότητες σελίδων περιλαμβάνονται ως περιεχόμενο σε κάθε ένα από τα πρότυπα της εφαρμογής. Αυτό, με τη σειρά του, έχει το πλεονέκτημα της διατήρησης του κώδικα ελάχιστου και συντηρήσιμου, καθώς εάν υπάρξει μια αλλαγή σε ένα από αυτά τα στοιχεία, αλλάζει μόνο στην κατάλληλη ενότητα.

Συμπερασματικά, οι δύο γλώσσες που συνέβαλαν στη δημιουργία της εφαρμογής ως σύνολο είναι η HTML, που χρησιμοποιήθηκε για τη βασική της δομή και οργάνωση του περιεχομένου της, και η γλώσσα προτύπων της Django που χρησιμοποιήθηκε για τη δημιουργία μιας δυναμικής εμπειρίας περιεχομένου που εμφανίζει σε κάθε χρήστη, κάθε συνέδριο και κάθε εργασία συνεδρίου ένα μοναδικό περιβάλλον προσαρμοσμένο σε κάθε ρόλο.

2.1.5 CSS

Η γλώσσα CSS είναι αυτή που χρησιμοποιείται για τη διαμόρφωση μιας ιστοσελίδας και ενός ιστότοπου. Ενώ η δομή HTML καθορίζει το περιεχόμενο μιας σελίδας, με τη βοήθεια της CSS ορίζεται πώς αυτό θα φαίνεται στον χρήστη. Η γλώσσα CSS χρησιμοποιείται για τον καθορισμό της εμφάνισης όσον αφορά τα χρώματα, τις γραμματοσειρές, την απόσταση, τα περιγράμματα, τις στοιχίσεις, τα μεγέθη και τη διάταξη των στοιχείων, καθώς και πώς η εμφάνιση προσαρμόζεται σε διάφορα μεγέθη οθόνης (MDN Web Docs, n.d.).

Σε αυτήν την εφαρμογή δεν χρησιμοποιήσαμε κανένα έτοιμο front end framework, αλλά δημιουργήσαμε προσαρμοσμένα αρχεία CSS για να καλύψουμε τις ανάγκες μας. Αυτό μας επέτρεψε να ελέγξουμε καλύτερα πώς φαίνεται η εφαρμογή και πώς προσαρμόζεται στις υλοποιημένες λειτουργίες μας. Ήμασταν σε θέση να προσαρμόσουμε το γραφικό

περιβάλλον στις ανάγκες μας για ένα σύστημα συνεδρίων χωρίς να χρειαστεί να βασιστούμε σε έτοιμες κλάσεις και στοιχεία από πηγές τρίτων.

Χρησιμοποιήσαμε CSS για να διαμορφώσουμε την αρχική σελίδα, τις κάρτες συνεδρίων, τη φόρμα σύνδεσης και εγγραφής, τους πίνακες ελέγχου, τις λίστες υποβολών, τα πλαίσια ειδοποιήσεων, τα κουμπιά ενεργειών, τη φόρμα αξιολόγησης κ.λπ. Δώσαμε μεγάλη έμφαση στην οπτική διαχώριση των διαφορετικών λειτουργικών περιοχών της εφαρμογής, ώστε ο χρήστης να γνωρίζει ποιες επιλογές είναι διαθέσιμες, πώς προχωρά μια υποβολή, ποιος είναι ο ρόλος ενός χρήστη σε κάθε συνέδριο κ.λπ. Για παράδειγμα, οι διαφορετικές ενότητες που αποτελούν την εφαρμογή, η λίστα παρακολούθησης, οι ειδοποιήσεις, οι υποβολές μου και η ουρά αναθεώρησης, διαμορφώθηκαν σε ξεχωριστά πλαίσια, ώστε να μπορούν να παρουσιαστούν εύκολα και με κατανοητό τρόπο.

Ομοίως, ήταν δυνατόν να διαμορφωθούν και να διαφοροποιηθούν οπτικά οι διάφορες καταστάσεις που μπορεί να περάσει μια εργασία, π.χ. υποβλήθηκε, υπό αναθεώρηση, αποδεκτή, απορριφθείσα, χρειάζεται 3ο αναθεωρητή, αναμένεται πληρωμή, έτοιμη για παρουσίαση κ.λπ. Αυτό διευκόλυνε την άμεση κατανόηση σε ποιο στάδιο βρισκόταν κάθε υποβολή. Η χρήση της CSS συνέβαλε επίσης στη δημιουργία ενός ενιαίου σχεδιασμού σε ολόκληρη την εφαρμογή. Στοιχεία όπως η γραμμή πλοήγησης, τα κουμπιά, οι φόρμες, οι πίνακες και το μήνυμα για την ενημέρωση ενός προφίλ, αποκτήσαν μια κοινή εμφάνιση μεταξύ των διαφορετικών σελίδων της εφαρμογής, παρέχοντας έτσι μια ενοποιημένη εμπειρία στον χρήστη.

Η ύπαρξη ενός μεγάλου αρχείου CSS ή λίγων οργανωμένων και καλά ονομασμένων αρχείων CSS καθιστά ευκολότερη τη συντήρηση της εφαρμογής στο μέλλον, καθώς οι αλλαγές στην εμφάνιση και την αίσθηση της εφαρμογής μπορούν να εφαρμοστούν εύκολα χωρίς να χρειαστεί να αλλάξει η διάταξη του HTML template. Χρησιμοποιήσαμε CSS για να προσαρμοστούμε σε οθόνες διαφορετικών μεγεθών. Χρησιμοποιώντας κατάλληλα στυλ διάταξης, καταφέραμε να διασφαλίσουμε ότι τα σημαντικά περιεχόμενα της εφαρμογής, τα πεδία φόρμας κ.λπ., είναι ευανάγνωστα ακόμη και σε μικρότερα μεγέθη οθόνης. Αυτό είναι απαραίτητο για μια διαδικτυακή εφαρμογή όπου ο χρήστης μπορεί να έχει πρόσβαση στην διαδικτυακή εφαρμογή σε διαφορετικούς υπολογιστές και συσκευές.

Συνοψίζοντας, η γλώσσα CSS ήταν ένα σημαντικό εργαλείο στο σχεδιασμό της εμφάνισης και της χρηστικότητας της εφαρμογής μας. Η χρήση της CSS από μόνοι μας, χωρίς να

βασιστούμε σε έτοιμα frameworks, μας επέτρεψε να δημιουργήσουμε ένα προσαρμοσμένο περιβάλλον που ταιριάζει στις ανάγκες μας, μια καθαρή και εύκολη στην κατανόηση διάταξη, μια ενιαία εμφάνιση των στοιχείων, μια ενοποιημένη εμπειρία σε διαφορετικά μεγέθη οθόνης και έναν επεκτάσιμο σχεδιασμό.

2.1.6 Stripe

Το Stripe είναι μια διαδικτυακή πλατφόρμα πληρωμών που χρησιμοποιείται ευρέως για την ενσωμάτωση ηλεκτρονικών πληρωμών σε ιστοσελίδες και εφαρμογές ηλεκτρονικού εμπορίου. Προσφέρει εργαλεία και API (Διεπαφή Προγραμματισμού Εφαρμογών) για τους προγραμματιστές, ώστε να ρυθμίζουν διαδικασίες πληρωμής, να επεξεργάζονται και να παρακολουθούν πληρωμές, καθώς και να συνδέουν μια ολοκληρωμένη συναλλαγή με την εφαρμογή (Stripe, n.d.).

Στην παρούσα εφαρμογή, το Stripe χρησιμοποιήθηκε για την υποστήριξη της διαδικασίας πληρωμής του τέλους συνεδρίου μετά την αποδοχή μιας εργασίας και μετά την ανάθεσή της σε ένα συγκεκριμένο τμήμα του προγράμματος. Αυτή η πληρωμή δεν είναι διαθέσιμη αμέσως μόλις γίνει αποδεκτή μια εργασία· αντίθετα, γίνεται διαθέσιμη μόνο αφού ο διοργανωτής έχει αναθέσει την αποδεκτή εργασία σε ένα τμήμα. Αυτό διασφαλίζει ότι η εφαρμογή δεν προτρέπει έναν χρήστη να πληρώσει για μια εργασία που δεν έχει περάσει από τα απαραίτητα στάδια αποδοχής και δεν έχει γίνει μέρος του προγράμματος. Μέσω της ενσωμάτωσης του Stripe, οι χρήστες ανακατευθύνονται σε ένα περιβάλλον πληρωμής όπου μπορούν να ολοκληρώσουν τη διαδικασία πληρωμής και, μετά την ολοκλήρωση, να ενημερώσουν την εφαρμογή και να τροποποιήσουν την κατάσταση μιας υποβολής σε "έτοιμη για παρουσίαση" ή "αναμονή πληρωμής". Με την επιτυχή πληρωμή, η εργασία λαμβάνει την κατάσταση "έτοιμη για παρουσίαση", οπότε μπορεί να συμπεριληφθεί στο τμήμα παρουσιάσεων του προγράμματος. Σε περίπτωση αποτυχίας πληρωμής, η εργασία θα παραμείνει στην κατάσταση "αναμονή πληρωμής" και ο χρήστης θα έχει την ευκαιρία να δοκιμάσει ξανά τη διαδικασία.

Η χρήση του Stripe βοήθησε στην υλοποίηση μιας σαφούς διαδικασίας αποδοχής μιας εργασίας, ανάθεσής της σε ένα τμήμα και στη συνέχεια στην τελική πληρωμή του τέλους για αυτήν την εργασία. Βοήθησε επίσης στην υλοποίηση της διαδικασίας πληρωμής, η οποία είναι αρκετά ρεαλιστική με αυτήν μιας πραγματικής εγγραφής σε συνέδριο και εκδήλωση. Η υπηρεσία Stripe περιλαμβάνει μια δοκιμαστική λειτουργία για πληρωμές, η

οποία επιτρέπει στους χρήστες να δοκιμάζουν συναλλαγές χωρίς να πραγματοποιούν πραγματικές πληρωμές. Συνοπτικά, η χρήση του Stripe ως εξωτερικής υπηρεσίας πληρωμών για την ολοκλήρωση της διαδικασίας πληρωμής ήταν αρκετά χρήσιμη στην υλοποίηση της συμμετοχής των παρουσιαστών.

2.1.7 Mailpit

Το Mailpit λειτουργεί ως ένα τοπικό εργαλείο για τη δοκιμή και την παρακολούθηση των email που αποστέλλονται σε ένα περιβάλλον ανάπτυξης, επιτρέποντας στους προγραμματιστές να καταγράφουν αυτά τα email από μια εφαρμογή χωρίς τα πραγματικά μηνύματα να παραδίδονται στις διευθύνσεις email των χρηστών. Ουσιαστικά, αυτό επιτρέπει τη δοκιμή της διαδικασίας αποστολής εντός του περιβάλλοντος ανάπτυξης χωρίς την ανάγκη αποστολής δοκιμαστικών email σε πραγματικούς λογαριασμούς Mailpit, (v.δ.).

Η εφαρμογή χρησιμοποίησε το Mailpit για τη δοκιμή ειδοποιήσεων μέσω email. Αυτές οι ειδοποιήσεις θα μπορούσαν να αφορούν την εγγραφή χρήστη, την ανάθεση ρόλου σε συνέδριο, την ανάθεση εργασίας αναθεώρησης, την ενημέρωση για μια ενημέρωση υποβολής, την αποδοχή ή απόρριψη μιας εργασίας ή την επιτυχή ή αποτυχημένη πληρωμή μιας εργασίας. Η εφαρμογή χρησιμοποίησε το mailpit για να επιτρέψει την επιβεβαίωση ότι μια ειδοποίηση μέσω email δημιουργείται σωστά και περιλαμβάνει τις σχετικές πληροφορίες στην εφαρμογή πριν συνδεθεί ένας πραγματικός διακομιστής αλληλογραφίας για την αποστολή τέτοιων ειδοποιήσεων.

Το Mailpit λειτουργεί ως ένας τοπικός διακομιστής SMTP, στον οποίο η εφαρμογή στέλνει εξερχόμενα email, χωρίς τα email να αποστέλλονται εκτός του τοπικού δικτύου και τα email μπορούν να προβληθούν σε μια διαδικτυακή διεπαφή προσβάσιμη από έναν προγραμματιστή για να δει ποιος αποστολέας έστειλε ένα email, σε ποια διεύθυνση email στάλθηκε το email, τι αφορούσε το email, το περιεχόμενο του email και τις τεχνικές λεπτομέρειες ενός email, όπως η έκδοση του πρωτοκόλλου email. Αυτό σημαίνει ότι ήταν δυνατόν να γίνει εύκολα αποσφαλμάτωση και να διαπιστωθεί εάν οι ειδοποιήσεις μέσω email λειτουργούσαν όπως αναμενόταν.

Το Mailpit αποδείχθηκε ιδιαίτερα χρήσιμο κατά την ανάπτυξη της λειτουργικότητας αναθεώρησης εργασιών. Για παράδειγμα, μια ειδοποίηση μέσω email θα μπορούσε να σταλεί στους αναθεωρητές ότι τους έχει ανατεθεί η αναθεώρηση μιας εργασίας από τον

διοργανωτή του συνεδρίου. Επιπλέον, μια ειδοποίηση μέσω email μπορεί να σταλεί στον παρουσιαστή ενημερώνοντάς τον ότι η εργασία του έγινε αποδεκτή, απορρίφθηκε ή χρειάζεται έναν τρίτο αναθεωρητή. Αυτό επέτρεψε να επιβεβαιωθεί ότι η ειδοποίηση ήταν σωστή και ότι οι χρήστες μπορούσαν να λάβουν την ειδοποίηση εάν όλα πήγαιναν όπως αναμενόταν.

Το πλεονέκτημα της χρήσης του Mailpit κατά την ανάπτυξη εφαρμογών με πολλαπλούς ρόλους και διαφορετικούς τύπους αυτοματοποιημένων ειδοποιήσεων είναι ότι μπορεί κανείς να δει όλα τα εξερχόμενα email χωρίς να χρειάζεται να δημιουργήσει λογαριασμό σε καμία υπηρεσία email. Αυτό είναι πλεονέκτημα για έναν προγραμματιστή κατά την ανάπτυξη, επειδή δεν χρειάζεται να δημιουργήσει λογαριασμό για δοκιμές και δεν διατρέχει τον κίνδυνο αποστολής ανεπιθύμητων δοκιμαστικών email σε χρήστες. Επιπλέον, το Mailpit χρησιμοποιήθηκε ως εργαλείο για τη δοκιμή της αποστολής email εντός του περιβάλλοντος ανάπτυξης εφαρμογών και, χρησιμοποιώντας το, η ομάδα ανάπτυξης μπόρεσε να δοκιμάσει με επιτυχία και ασφάλεια τις δημιουργημένες ειδοποιήσεις στην εφαρμογή, γεγονός που με τη σειρά του επέτρεψε τη βελτίωση και τη δοκιμή της αποστολής ειδοποιήσεων, καθώς και την επιβεβαίωση ότι στους χρήστες αποστέλλονται οι σωστές ειδοποιήσεις, ανάλογα με τον ρόλο και την κατάσταση υποβολής.

2.2 Αρχιτεκτονική Συστήματος

Ένα κρίσιμο στοιχείο στην ανάπτυξη, υλοποίηση και επεκτασιμότητα ενός συστήματος πληροφοριών είναι η αρχιτεκτονική του. Αυτό το λογισμικό διαχείρισης συνεδρίων έχει αναπτυχθεί ακολουθώντας τον αρχιτεκτονικό σχεδιασμό MVT (Model View Template) του Django, ο οποίος διευκολύνει την οργανωμένη και δομημένη ανάπτυξη διαδικτυακών εφαρμογών.

Με απλά λόγια, οι χρήστες αλληλεπιδρούν με την εφαρμογή μέσω του ιστότοπου, ή Frontend. Οι σελίδες χρησιμοποιούνται για την εμφάνιση δημοσιευμένων συνεδρίων, την εγγραφή ενός χρήστη (sign up), τη σύνδεση στον ιστότοπο, την προσθήκη ενός συνεδρίου στη λίστα παρακολούθησης, ή την υποβολή μιας εργασίας, ή για την πρόσβαση στο Προσωπικό τους Πίνακα Ελέγχου και την εκτέλεση πρόσθετων ενεργειών ανάλογα με τον ρόλο τους στο συγκεκριμένο συνέδριο. Μεταξύ των διαθέσιμων ρόλων στην εφαρμογή περιλαμβάνονται ο Οργανωτής (αναθέτει κριτή, κρίνει εργασίες, δημιουργεί θεματικές ενότητες), ο Κριτής (κρίνει εργασίες), ο Παρουσιαστής (βλέπει υποβληθείσες εργασίες και

την κατάστασή τους) και το Μέλος (βλέπει δημοσιευμένα συνέδρια). Αυτές οι ενέργειες μεταβιβάζονται από τον ιστότοπο στο backend ως αιτήματα HTTP. Εντός του backend, προωθούνται σε συγκεκριμένες Προβολές (Views) από το σύστημα δρομολόγησης URL. Οι Προβολές είναι το σημείο στην ανάπτυξη της εφαρμογής όπου εκτελείται η βασική λογική, όπως ο έλεγχος εάν ένας χρήστης είναι συνδεδεμένος, ή ο έλεγχος για συγκεκριμένους ρόλους χρήστη για συγκεκριμένα συνέδρια (π.χ., ρόλοι οργανωτή ή κριτή για συγκεκριμένα συνέδρια), ο έλεγχος εάν τα συνέδρια έχουν ανοιχτές υποβολές για μια συγκεκριμένη εργασία, ο προσδιορισμός εάν μια εργασία μπορεί να κριθεί, ή εάν η διαδικασία πληρωμής μπορεί να ενεργοποιηθεί. Τα μοντέλα Django χρησιμοποιούνται στις Προβολές για τη λήψη, αποθήκευση και επεξεργασία δεδομένων από τη βάση δεδομένων της εφαρμογής. Τα μοντέλα είναι κεντρικά αντικείμενα που αναπαριστούν τα κύρια στοιχεία της εφαρμογής, συμπεριλαμβανομένων των χρηστών, των συνεδρίων, των ρόλων ανά συνέδριο, των θεματικών ενότητων, των υποβολών εργασιών, της επιλογής κριτών, των βαθμολογιών κριτών, των ειδοποιήσεων και των στοιχείων πληρωμής. Η ανταλλαγή δεδομένων μεταξύ της βάσης δεδομένων και του συστήματος πραγματοποιείται μέσω του ORM (Object Relational Mapper) του Django, το οποίο εξαλείφει την ανάγκη για άμεση γραφή σύνθετων ερωτημάτων SQL και διαχειρίζεται τα δεδομένα με αντικειμενοστραφή τρόπο. Όταν οι Προβολές λαμβάνουν τα επεξεργασμένα δεδομένα, αυτά επιστρέφονται στο Πρότυπο (Template). Οι σελίδες HTML δημιουργούνται με βάση τα δεδομένα από τα Πρότυπα και εμφανίζονται στον χρήστη σύμφωνα με τη Γλώσσα Προτύπων Django (Django Template Language). Ως αποτέλεσμα, οι μη συνδεδεμένοι χρήστες βλέπουν λεπτομέρειες συνεδρίων στη δημόσια σελίδα, αλλά οι εγγεγραμμένοι χρήστες μπορούν να έχουν πρόσβαση σε διαφορετικές πληροφορίες ανάλογα με το συνέδριο και τους ανατεθειμένους ρόλους τους. Ένας Οργανωτής έχει την επιπλέον επιλογή να αναθέτει ρόλους κριτών, να δημιουργεί θεματικές ενότητες και να αναθέτει κριτές σε συγκεκριμένες εργασίες προς κρίση. Ένας Κριτής βλέπει μια λίστα εργασιών προς κρίση, ενώ ένας Παρουσιαστής μπορεί να παρακολουθεί την κατάσταση των υποβληθεισών εργασιών του.

Η αρχιτεκτονική που χρησιμοποιείται σε αυτήν την εργασία είναι επίσης ικανή για ενσωμάτωση με εξωτερικές υπηρεσίες. Στο πλαίσιο της επεξεργασίας πληρωμών, χρησιμοποιείται το Stripe για την επεξεργασία της πληρωμής όταν ένα συνέδριο ανατίθεται σε μια θεματική ενότητα και ο παρουσιαστής απαιτείται να πληρώσει το τέλος του συνεδρίου μετά την αποδοχή της εργασίας του. Επιπλέον, για τη δοκιμή ειδοποιήσεων μέσω

email, χρησιμοποιείται το Mailpit για όλες τις εφαρμογές, επιτρέποντας στον χρήστη να επιθεωρεί τα email σχετικά με την ανάθεση ρόλων, την κρίση εργασιών και την κατάσταση υποβολής εργασιών. Η αρχιτεκτονική MVT του Django επιτρέπει την ευκολία ανάπτυξης, καθώς κάθε στοιχείο έχει συγκεκριμένες ευθύνες για τη διαχείριση του τμήματος των λειτουργιών της εφαρμογής. Αυτό οδηγεί σε εύκολη συντήρηση κώδικα για Μοντέλα, Προβολές και Πρότυπα, καθώς και σε μελλοντική επεκτασιμότητα, καθώς η εφαρμογή μπορεί να χωριστεί σε μικρότερες, διαχειρίσιμες εφαρμογές Django, καθεμία υπεύθυνη για ένα μόνο χαρακτηριστικό ή ένα μικρό σύνολο σχετικών λειτουργιών, όπως η δημιουργία πρόσθετων αναφορών για ένα συνέδριο, ένα καλύτερο πρόγραμμα παρουσιάσεων ή βελτιωμένοι μηχανισμοί αξιολόγησης. Επιπλέον, το Django ORM απλοποιεί τη διαχείριση της βάσης δεδομένων και παρέχει μηχανισμούς ασφαλείας για την αποφυγή κοινών ευπαθειών διαδικτυακών εφαρμογών, όπως επιθέσεις SQL injection ή Cross-Site Scripting. Το πλαίσιο παρέχει έναν ενσωματωμένο μηχανισμό για τη διαχείριση της πιστοποίησης, των συνεδρίων και των δικαιωμάτων, τα οποία είναι κλειδιά για αυτήν τη συγκεκριμένη εφαρμογή δεδομένης της έννοιας ρόλου ανά συνέδριο που υλοποιείται σε αυτήν. Με λίγα λόγια, βασιζόμενο στην αρχιτεκτονική MVT του Django, το σύστημα αναπτύχθηκε ακολουθώντας τις βασικές αρχές διαχωρισμού της διαχείρισης δεδομένων από την επιχειρηματική λογική και την παρουσίαση πληροφοριών, παρέχοντας μια λειτουργική, επεκτάσιμη και ευέλικτη εφαρμογή διαχείρισης συνεδρίων.

3. Ανάλυση και Σχεδιασμός Εφαρμογής

Οι απαιτήσεις του συστήματος προκύπτουν από τις αρχικές προδιαγραφές του έργου, οι οποίες περιγράφουν ένα ολοκληρωμένο workflow διαχείρισης συνεδρίων. Η παρούσα ανάλυση μεταφράζει τις αρχικές προδιαγραφές σε σαφέστερες ροές χρήσης και τις συμπληρώνει με μη λειτουργικά κριτήρια ποιότητας, ώστε η υλοποίηση να είναι συμβατή με πραγματικές συνθήκες λειτουργίας. Η συμπλήρωση δεν μεταβάλλει τον πυρήνα του προβλήματος, αλλά αποσαφηνίζει ιδιότητες που είναι αναγκαίες για ασφαλή, αξιόπιστη και επαληθεύσιμη λειτουργία.

3.1 Λειτουργικές Απαιτήσεις

Η λειτουργική προδιαγραφή ξεκινά από την ανάγκη ελεγχόμενης δημιουργίας συνεδρίων από διαχειριστές. Η ύπαρξη κεντρικής σελίδας διαχείρισης επιτρέπει να ορίζονται οι βασικές παράμετροι κάθε συνεδρίου, όπως χρονικά όρια υποβολών, περίοδοι εγγραφών και ρυθμίσεις παραβόλου. Η επιλογή staff-only δημιουργίας συνδέεται με την ανάγκη αποφυγής αυθαίρετης δημοσίευσης από μη εξουσιοδοτημένους χρήστες και με την ανάγκη ύπαρξης ενός σαφούς σημείου ευθύνης για τις βασικές ρυθμίσεις. Δεύτερο κρίσιμο στοιχείο είναι η σελίδα χρήστη (dashboard), η οποία συγκεντρώνει πληροφορία ανά ρόλο και ανά σχέση με συνέδρια. Το dashboard οφείλει να εμφανίζει, σε ενιαία επιφάνεια, συνέδρια που ο χρήστης παρακολουθεί, συνέδρια στα οποία συμμετέχει ως μέλος οργανωτικής ή κριτικής επιτροπής, συνέδρια στα οποία έχει εγγραφεί ως παρατηρητής, και συνέδρια στα οποία συμμετέχει ως παρουσιαστής. Η λειτουργία αυτή μειώνει την ανάγκη πλοήγησης σε πολλαπλά σημεία και επιτρέπει στον χρήστη να εντοπίζει γρήγορα εκκρεμείς ενέργειες. Η τρίτη βασική απαίτηση αφορά τη διαφοροποίηση λειτουργικότητας ανά ρόλο στη σελίδα του 7 εκάστοτε συνεδρίου. Η ίδια σελίδα αποτελεί σημείο σύγκλισης πληροφοριών και actions, αλλά το σύνολο των actions πρέπει να εξαρτάται από τον ρόλο του χρήστη. Η οργανωτική επιτροπή πρέπει να μπορεί να ορίζει tracks και να εκτελεί ενέργειες ανάθεσης, η κριτική επιτροπή να μπορεί να καταχωρεί αξιολογήσεις μόνο για εργασίες που της έχουν ανατεθεί, ενώ οι παρουσιαστές και παρατηρητές να έχουν περιορισμένες δυνατότητες που δεν αλλοιώνουν κρίσιμα δεδομένα. Στον πυρήνα του workflow βρίσκεται η υποβολή εργασιών και η αξιολόγησή τους. Κάθε χρήστης μπορεί να καταθέσει paper προς ένα συνέδριο, υπό προϋποθέσεις που ορίζονται από την κατάσταση του συνεδρίου. Η οργανωτική επιτροπή αναθέτει reviewers από τα μέλη της κριτικής επιτροπής. Η αποδοχή

μιας εργασίας απαιτεί είτε ομοφωνία από δύο reviewers είτε, σε περίπτωση διαφωνίας, ανάθεση τρίτου reviewer και τελική απόφαση βάσει πλειοψηφίας. Η απαίτηση αυτή επιβάλλει σαφή αποτύπωση καταστάσεων και transitions ώστε να αποφεύγονται αμφίσημες αποφάσεις ή ελλυπείς καταχωρίσεις. Μετά την αποδοχή, απαιτείται ένταξη της εργασίας σε track και ενημέρωση του χρήστη. Όταν το συνέδριο έχει οριστεί ως επί πληρωμή, ο χρήστης πρέπει να έχει δυνατότητα πληρωμής παραβόλου για την τελική επιβεβαίωση συμμετοχής. Η πληρωμή δεν λειτουργεί ως ανεξάρτητη λειτουργία, αλλά ως βήμα που συνδέεται με σαφή κατάσταση της υποβολής και με κανόνες που οδηγούν στην τελική κατάσταση παρουσίας.

3.2 Ρόλοι, Δικαιώματα και Σενάρια Χρήσης

Η προδιαγραφή επιβάλλει μοντέλο πρόσβασης βασισμένο σε ρόλους (RBAC), σε επίπεδο συνεδρίου. Ο ίδιος χρήστης μπορεί να έχει διαφορετικό ρόλο σε διαφορετικά συνέδρια. Η διάκριση αυτή είναι κρίσιμη, διότι επιτρέπει στο σύστημα να υποστηρίζει πολλαπλές κοινότητες και πολλαπλές διαδικασίες χωρίς να αποδίδει μόνιμα προνόμια σε χρήστες. Επιπλέον, επιτρέπει να οριστούν σε φή σενάρια χρήσης, στα οποία κάθε ρόλος εκτελεί περιορισμένες ενέργειες που αντιστοιχούν στις ευθύνες του. Ο ρόλος staff αφορά την ανάρτηση και διαχείριση συνεδρίων σε επίπεδο συστήματος. Ο ρόλος organizer αφορά κρίσιμες αποφάσεις εντός συνεδρίου, όπως ανάθεση reviewers και tracks. Ο ρόλος reviewer αφορά την αξιολόγηση υποβολών που έχουν ανατεθεί. Ο ρόλος presenter αφορά την υποβολή εργασιών και την ολοκλήρωση βημάτων συμμετοχής, συμπεριλαμβανομένης της πληρωμής όταν απαιτείται. Ο ρόλος observer αφορά πρόσβαση σε πληροφορία και παρακολούθηση χωρίς δυνατότητα αλλοίωσης κρίσιμων δεδομένων. Η αποτύπωση των ρόλων ως ρητές συσχετίσεις στο μοντέλο δεδομένων διευκολύνει την επιβολή πολιτικών πρόσβασης σε όλα τα endpoints, ανεξάρτητα από το UI.

3.3 Μη Λειτουργικές Απαιτήσεις

Η λειτουργική πληρότητα δεν αρκεί για ένα σύστημα που αποθηκεύει δεδομένα χρηστών και υποβολές. Σε επίπεδο ασφάλειας, απαιτείται προστασία έναντι CSRF μέσω των μηχανισμών του Django, καθώς και μείωση της επιφανείας XSS με αυστηρές πολιτικές CSP και ασφαλείς προεπιλογές cookie (Django Software Foundation, χ.χ.-c, χ.χ.-a; OWASP Cheat Sheet Series, χ.χ.-a). Επιπλέον, απαιτείται περιορισμός ρυθμού αιτημάτων σε ευαίσθητα endpoints, ώστε να μειώνεται ο κίνδυνος brute force ή κατάχρησης (OWASP

Cheat Sheet Series, χ.χ.-b). Για εξωτερικές ολοκληρώσεις, όπως πληρωμές, απαιτείται επαλήθευση webhooks και idempotency, ώστε να αποφεύγεται διπλή επεξεργασία γεγονότων (Stripe, χ.χ.-c, χ.χ.-b). Σε επίπεδο auditability, απαιτείται καταγραφή κρίσιμων ενεργειών σε σταθερό σχήμα, ώστε να εί ναι δυνατή η διερεύνηση συμβάντων.

3.4 Περιπτώσεις Χρήσης

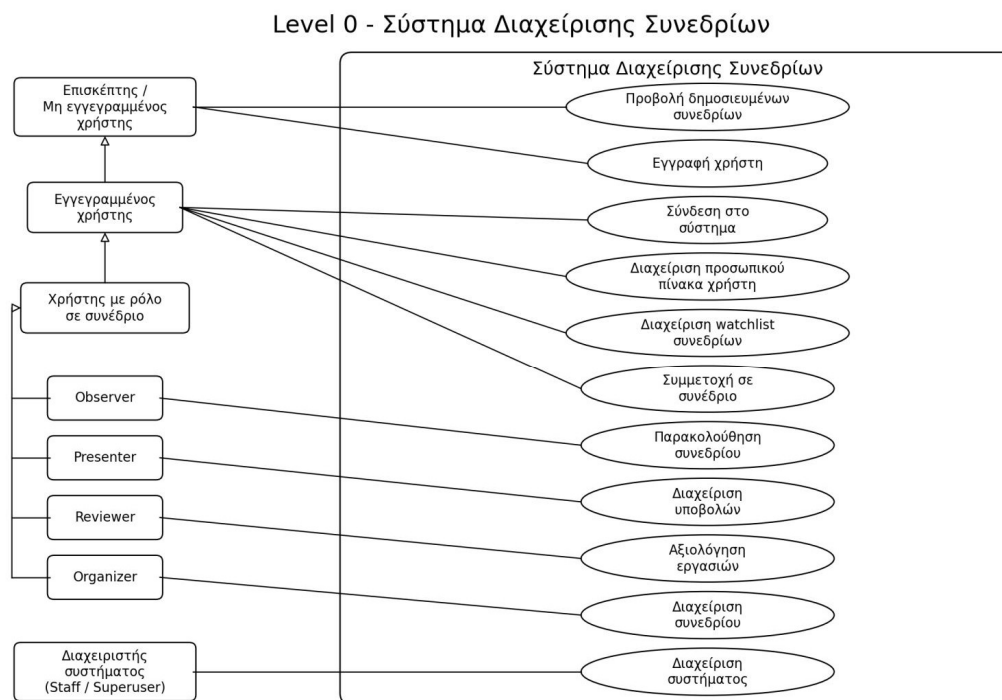
Στην παρούσα ενότητα παρουσιάζονται οι βασικές περιπτώσεις χρήσης της εφαρμογής, όπως αυτές προκύπτουν από τις λειτουργίες που παρέχει το Πληροφοριακό Σύστημα Διαχείρισης Συνεδρίων. Οι περιπτώσεις χρήσης αποτυπώνουν τον τρόπο με τον οποίο οι διαφορετικές κατηγορίες χρηστών αλληλεπιδρούν με το σύστημα, καθώς και τις λειτουργίες στις οποίες έχουν πρόσβαση ανάλογα με την κατάστασή τους και τον ρόλο τους σε κάθε συνέδριο.

Η εφαρμογή υποστηρίζει τόσο μη συνδεδεμένους όσο και συνδεδεμένους χρήστες. Ο μη εγγεγραμμένος ή μη συνδεδεμένος χρήστης έχει πρόσβαση μόνο σε δημόσιες λειτουργίες, όπως η προβολή των δημοσιευμένων συνεδρίων και η δυνατότητα εγγραφής στο σύστημα. Μετά την εγγραφή και σύνδεση, ο χρήστης μπορεί να διαχειρίζεται τον προσωπικό του πίνακα, να προσθέτει συνέδρια στη watchlist, να εγγράφεται ως observer σε συνέδρια και να υποβάλλει εργασίες, εφόσον οι υποβολές είναι ανοιχτές.

Ιδιαίτερο χαρακτηριστικό της εφαρμογής είναι ότι οι ρόλοι των χρηστών δεν αποδίδονται γενικά κατά την εγγραφή, αλλά συνδέονται με συγκεκριμένο συνέδριο. Έτσι, ένας χρήστης μπορεί να είναι απλός συνδεδεμένος χρήστης στο σύστημα, observer σε ένα συνέδριο, presenter σε άλλο συνέδριο και reviewer σε κάποιο τρίτο. Η διαφοροποίηση αυτή αποτυπώνεται στο διάγραμμα μέσω της κατηγορίας «Χρήστης με ρόλο σε συνέδριο», από την οποία εξειδικεύονται οι ρόλοι Observer, Presenter, Reviewer και Organizer.

Ο Observer έχει τη δυνατότητα παρακολούθησης ενός συνεδρίου, ο Presenter μπορεί να διαχειρίζεται τις υποβολές του, ο Reviewer μπορεί να αξιολογεί εργασίες που του έχουν ανατεθεί, ενώ ο Organizer είναι υπεύθυνος για τη διαχείριση του συνεδρίου, την ανάθεση ρόλων, τον ορισμό tracks και την ανάθεση reviewers σε υποβολές. Επιπλέον, ο Διαχειριστής Συστήματος ή Staff/Superuser έχει πρόσβαση στη διαχείριση του συστήματος, συμπεριλαμβανομένης της δημιουργίας συνεδρίων και της πρόσβασης στο διαχειριστικό περιβάλλον.

Η Εικόνα 1 παρουσιάζει το γενικό διάγραμμα περιπτώσεων χρήσης επιπέδου 0 της εφαρμογής. Το διάγραμμα αυτό λειτουργεί ως συνοπτική απεικόνιση των βασικών αλληλεπιδράσεων των χρηστών με το σύστημα και αποτελεί τη βάση για την αναλυτική περιγραφή των επιμέρους περιπτώσεων χρήσης που ακολουθούν.



Εικόνα 1 Διάγραμμα Περιπτώσεων Χρήσης

Στην Εικόνα παρατηρείται ότι οι βασικές λειτουργίες του συστήματος οργανώνονται γύρω από τις κατηγορίες χρηστών. Ο μη συνδεδεμένος χρήστης μπορεί να δει τα δημοσιευμένα συνέδρια και να πραγματοποιήσει εγγραφή. Ο εγγεγραμμένος χρήστης μπορεί να συνδεθεί, να διαχειριστεί τον προσωπικό του πίνακα, να προσθέσει ή να αφαιρέσει συνέδρια από τη watchlist και να συμμετάσχει σε συνέδριο. Οι χρήστες που αποκτούν ρόλο σε ένα συνέδριο έχουν πρόσβαση σε πρόσθετες λειτουργίες ανάλογα με τον ρόλο τους. Συγκεκριμένα, ο Observer μπορεί να παρακολουθεί το συνέδριο, ο Presenter να διαχειρίζεται τις υποβολές του, ο Reviewer να αξιολογεί εργασίες και ο Organizer να διαχειρίζεται το συνέδριο. Τέλος, ο Διαχειριστής Συστήματος έχει πρόσβαση στη συνολική διαχείριση του συστήματος.

Οι βασικές περιπτώσεις χρήσης που προκύπτουν από το διάγραμμα είναι οι εξής:

ΠΧ1: Προβολή δημοσιευμένων συνεδρίων

Η περίπτωση χρήσης περιγράφει τη δυνατότητα του επισκέπτη ή οποιουδήποτε χρήστη να δει τα συνέδρια που έχουν δημοσιευθεί στο σύστημα. Τα συνέδρια που βρίσκονται σε κατάσταση draft δεν εμφανίζονται στη δημόσια λίστα.

ΠΧ2: Εγγραφή χρήστη

Η περίπτωση χρήσης περιγράφει τη διαδικασία δημιουργίας νέου λογαριασμού χρήστη. Με την εγγραφή ο χρήστης αποκτά πρόσβαση στο σύστημα, χωρίς όμως να αποκτά αυτόματα κάποιον ρόλο σε συνέδριο.

ΠΧ3: Σύνδεση στο σύστημα

Η περίπτωση χρήσης περιγράφει τη διαδικασία αυθεντικοποίησης του χρήστη με χρήση των στοιχείων σύνδεσης. Μετά τη σύνδεση, ο χρήστης μπορεί να έχει πρόσβαση σε προσωπικές λειτουργίες και στον πίνακα ελέγχου του.

ΠΧ4: Διαχείριση προσωπικού πίνακα χρήστη

Η περίπτωση χρήσης περιγράφει την πρόσβαση του συνδεδεμένου χρήστη στον προσωπικό του πίνακα ελέγχου. Εκεί εμφανίζονται οι ειδοποιήσεις, οι ρόλοι του ανά συνέδριο, οι υποβολές του, η watchlist και οι εκκρεμότητες αξιολόγησης, εφόσον υπάρχουν.

ΠΧ5: Διαχείριση watchlist συνεδρίων

Η περίπτωση χρήσης περιγράφει τη δυνατότητα του συνδεδεμένου χρήστη να προσθέτει ή να αφαιρεί συνέδρια από τη λίστα παρακολούθησης. Η προσθήκη ενός συνεδρίου στη watchlist δεν αποδίδει κάποιον ρόλο στον χρήστη.

ΠΧ6: Συμμετοχή σε συνέδριο

Η περίπτωση χρήσης περιγράφει τη δυνατότητα του χρήστη να αποκτήσει ρόλο σε ένα συνέδριο. Για παράδειγμα, μέσω της εγγραφής ως observer αποκτά τον ρόλο του παρατηρητή, ενώ μέσω της υποβολής εργασίας αποκτά τον ρόλο του presenter.

ΠΧ7: Παρακολούθηση συνεδρίου

Η περίπτωση χρήσης αφορά τον χρήστη με ρόλο Observer. Ο observer μπορεί να παρακολουθεί το συνέδριο και να έχει πρόσβαση στις σχετικές πληροφορίες του, χωρίς να συμμετέχει απαραίτητα στη διαδικασία υποβολής ή αξιολόγησης εργασιών.

ΠΧ8: Διαχείριση υποβολών

Η περίπτωση χρήσης αφορά τον Presenter και περιγράφει τις ενέργειες υποβολής και παρακολούθησης εργασιών. Ο presenter μπορεί να υποβάλει εργασία σε συνέδριο με ανοιχτές υποβολές και να παρακολουθεί την κατάστασή της, όπως submitted, under review, accepted, rejected, needs third reviewer, paymentpending ή presentationready.

ΠΧ9: Αξιολόγηση εργασιών

Η περίπτωση χρήσης αφορά τον Reviewer και περιγράφει τη διαδικασία αξιολόγησης εργασιών που του έχουν ανατεθεί. Ο reviewer μπορεί να δει τις εκκρεμείς αξιολογήσεις του, να κατεβάσει το αρχείο της εργασίας, να καταχωρήσει πρόταση αξιολόγησης, βαθμολογία και σχόλια.

ΠΧ10: Διαχείριση συνεδρίου

Η περίπτωση χρήσης αφορά τον Organizer και περιγράφει τις ενέργειες διαχείρισης ενός συνεδρίου. Ο organizer μπορεί να αναθέτει ρόλους σε χρήστες, να ορίζει tracks, να βλέπει τα μέλη του συνεδρίου, να αναθέτει reviewers σε υποβολές και να εντάσσει αποδεκτές εργασίες σε tracks.

ΠΧ11: Διαχείριση συστήματος

Η περίπτωση χρήσης αφορά τον Διαχειριστή Συστήματος ή Staff/Superuser. Ο διαχειριστής μπορεί να δημιουργεί νέα συνέδρια, να ορίζει την κατάστασή τους ως draft, published ή archived, να παρακολουθεί τα υπάρχοντα συνέδρια και να έχει πρόσβαση στο διαχειριστικό περιβάλλον του Django.

Το διάγραμμα περιπτώσεων χρήσης επιπέδου 0 παρέχει μια συνολική εικόνα της λειτουργικότητας της εφαρμογής και βοηθά στην κατανόηση της σχέσης ανάμεσα στους χρήστες, τους ρόλους και τις διαθέσιμες λειτουργίες του συστήματος. Στις επόμενες υποενότητες περιγράφονται αναλυτικά οι σημαντικότερες περιπτώσεις χρήσης, με αναφορά στις συνθήκες εισόδου και εξόδου, στη βασική ροή γεγονότων και στις εναλλακτικές ροές όπου αυτές απαιτούνται.

3.4.1 Τίτλος περίπτωσης χρήσης : Διαχείριση πίνακα και watchlist

Αναγνωριστικό: ΠΧ2

Σύντομη περιγραφή: Η παρούσα περίπτωση χρήσης περιγράφει τις λειτουργίες που είναι διαθέσιμες στον εγγεγραμμένο χρήστη μέσα από τον προσωπικό του πίνακα ελέγχου. Ο

πίνακας χρήστη αποτελεί το κεντρικό σημείο ενημέρωσης του χρήστη, καθώς συγκεντρώνει τις ειδοποιήσεις του, τους ρόλους που έχει αποκτήσει ανά συνέδριο, τις υποβολές του, τα συνέδρια που έχει προσθέσει στη watchlist και τις εκκρεμότητες αξιολόγησης, εφόσον ο χρήστης έχει ρόλο reviewer.

Βασικός χειριστής: Εγγεγραμμένος χρήστης

Δευτερεύοντες χειριστές: Presenter, Reviewer, Organizer

Συνθήκη εισόδου: Ο χρήστης έχει πραγματοποιήσει σύνδεση στο σύστημα.

Συνθήκη εξόδου: Ο χρήστης έχει προβάλει τον προσωπικό του πίνακα ή έχει εκτελέσει κάποια από τις διαθέσιμες ενέργειες, όπως προσθήκη ή κατάργηση συνεδρίου από τη watchlist, προβολή ειδοποιήσεων ή προβολή των υποβολών του.

Κανονική ροή γεγονότων:

1. Ο εγγεγραμμένος χρήστης επιλέγει την προβολή του προσωπικού του πίνακα.
2. Το σύστημα εμφανίζει τις βασικές ενότητες του dashboard.
3. Ο χρήστης μπορεί να δει τα συνέδρια που έχει προσθέσει στη watchlist.
4. Ο χρήστης μπορεί να δει τους ρόλους που έχει αποκτήσει ανά συνέδριο.
5. Ο χρήστης μπορεί να δει τις ειδοποιήσεις που έχουν δημιουργηθεί από το σύστημα.
6. Ο χρήστης μπορεί να σημειώσει μια ειδοποίηση ως αναγνωσμένη ή να σημειώσει όλες τις ειδοποιήσεις ως αναγνωσμένες.
7. Ο χρήστης μπορεί να δει τις υποβολές που έχει πραγματοποιήσει σε συνέδρια.
8. Αν ο χρήστης έχει ρόλο reviewer, μπορεί να δει το review queue με τις εκκρεμείς αξιολογήσεις.
9. Αν ο χρήστης έχει προσθέσει συνέδρια στη watchlist, μπορεί να τα αφαιρέσει από αυτή.

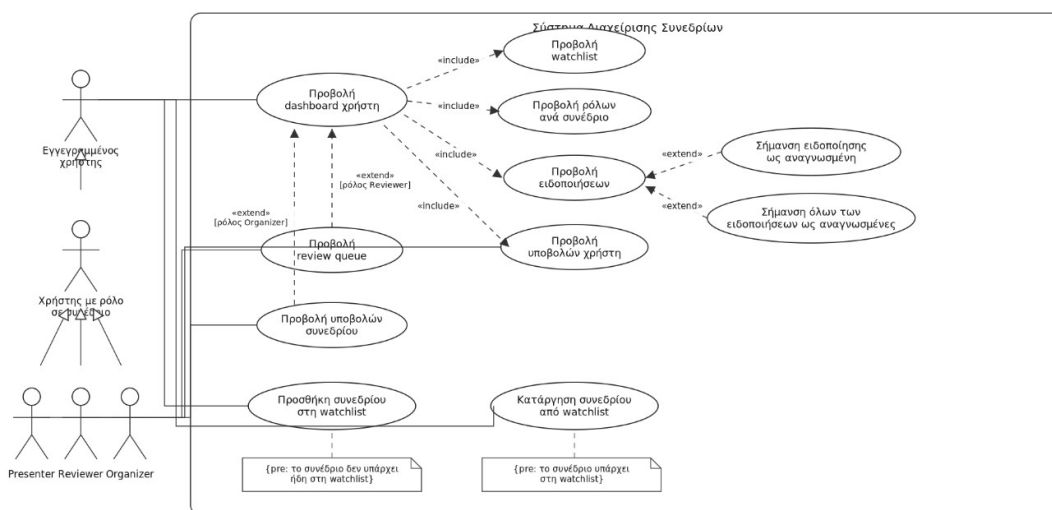
Εναλλακτική ροή γεγονότων:

- Αν ο χρήστης δεν έχει προσθέσει συνέδρια στη watchlist, το αντίστοιχο πλαίσιο εμφανίζεται κενό

- Αν ο χρήστης δεν έχει αποκτήσει ρόλο σε κάποιο συνέδριο, δεν εμφανίζεται κάποια σχετική ιδιότητα.
- Αν ο χρήστης δεν έχει υποβάλει εργασία, η λίστα υποβολών εμφανίζεται κενή.
- Αν ο χρήστης δεν έχει ρόλο reviewer, δεν εμφανίζονται εκκρεμότητες αξιολόγησης.

Σημείωση:

Ο επισκέπτης ή μη συνδεδεμένος χρήστης δεν έχει πρόσβαση σε προσωπικό πίνακα ελέγχου ή σε watchlist. Οι λειτουργίες αυτές απαιτούν σύνδεση στο σύστημα.



Εικόνα 2 — Διάγραμμα Διαχείρισης Πίνακα Χρήστη και Watchlist

3.4.2 Τίτλος περίπτωσης χρήσης: Συμμετοχή σε συνέδριο και υποβολή paper

Αναγνωριστικό: PX3

Σύντομη περιγραφή: Η συγκεκριμένη περίπτωση χρήσης περιγράφει τους τρόπους με τους οποίους ένας εγγεγραμμένος χρήστης μπορεί να συμμετάσχει σε ένα συνέδριο. Η συμμετοχή μπορεί να πραγματοποιηθεί είτε με εγγραφή ως παρατηρητής, οπότε ο χρήστης αποκτά τον ρόλο Observer, είτε μέσω υποβολής εργασίας, οπότε αποκτά τον ρόλο Presenter για το συγκεκριμένο συνέδριο.

Βασικός χειριστής: Εγγεγραμμένος χρήστης

Συνθήκη εισόδου: Ο χρήστης έχει συνδεθεί στο σύστημα και έχει μεταβεί στη σελίδα πληροφοριών ενός δημοσιευμένου συνεδρίου.

Συνθήκη εξόδου: Ο χρήστης έχει εγγραφεί ως observer ή έχει υποβάλει paper και έχει αποκτήσει τον αντίστοιχο ρόλο στο συγκεκριμένο συνέδριο.

Κανονική ροή γεγονότων για εγγραφή ως παρατηρητής:

1. Ο χρήστης μεταβαίνει στη σελίδα ενός δημοσιευμένου συνεδρίου.
2. Το σύστημα εμφανίζει τις διαθέσιμες ενέργειες συμμετοχής.
3. Ο χρήστης επιλέγει την εγγραφή ως παρατηρητής.
4. Το σύστημα καταχωρεί τη συμμετοχή του χρήστη στο συνέδριο.
5. Ο χρήστης αποκτά τον ρόλο Observer για το συγκεκριμένο συνέδριο.
6. Ο ρόλος εμφανίζεται στον προσωπικό πίνακα ελέγχου του χρήστη.

Κανονική ροή γεγονότων για υποβολή paper:

1. Ο χρήστης μεταβαίνει στη σελίδα ενός συνεδρίου.
2. Το σύστημα ελέγχει αν οι υποβολές του συνεδρίου είναι ανοιχτές.
3. Ο χρήστης επιλέγει τη λειτουργία υποβολής paper.
4. Το σύστημα εμφανίζει τη φόρμα υποβολής.
5. Ο χρήστης συμπληρώνει τα στοιχεία της εργασίας, όπως τίτλο, περίληψη και λέξεις-κλειδιά.
6. Ο χρήστης μπορεί προαιρετικά να μεταφορτώσει αρχείο PDF.
7. Το σύστημα ελέγχει ότι δεν υπάρχει ήδη εργασία του ίδιου χρήστη με τον ίδιο τίτλο στο ίδιο συνέδριο.
8. Το σύστημα καταχωρεί την υποβολή.
9. Ο χρήστης αποκτά τον ρόλο Presenter για το συγκεκριμένο συνέδριο.
10. Η υποβολή εμφανίζεται στον πίνακα του χρήστη και στη λίστα υποβολών του συνεδρίου.

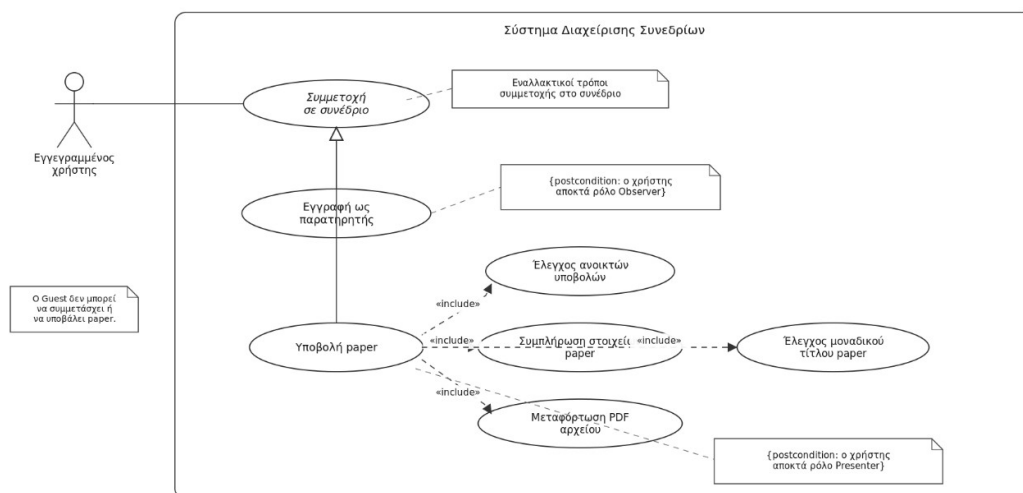
Εναλλακτική ροή γεγονότων:

- Αν ο χρήστης δεν είναι συνδεδεμένος, δεν μπορεί να συμμετάσχει σε συνέδριο ή να υποβάλει paper.

- Αν οι υποβολές του συνεδρίου είναι κλειστές, η δυνατότητα υποβολής δεν είναι διαθέσιμη.
- Αν υπάρχει ήδη paper με τον ίδιο τίτλο από τον ίδιο χρήστη στο ίδιο συνέδριο, το σύστημα δεν επιτρέπει την καταχώρηση.
- Αν ο χρήστης δεν επιλέξει αρχείο PDF, η υποβολή μπορεί να ολοκληρωθεί χωρίς αρχείο, εφόσον η μεταφόρτωση είναι προαιρετική.

Σημείωση:

Η συμμετοχή σε συνέδριο δεν είναι ενιαία λειτουργία με έναν μόνο τρόπο. Ο χρήστης μπορεί να συμμετάσχει είτε ως observer είτε ως presenter μέσω υποβολής paper.



Εικόνα 3 — Διάγραμμα Συμμετοχής σε Συνέδριο και Υποβολής Paper

3.4.3 Τίτλος περίπτωσης χρήσης: Διαχείριση υποβολών από Presenter

Αναγνωριστικό: ΠΧ4

Σύντομη περιγραφή: Η παρούσα περίπτωση χρήσης περιγράφει τις ενέργειες που μπορεί να εκτελέσει ο χρήστης με ρόλο Presenter σε σχέση με τις υποβολές του. Ο presenter μπορεί να δει τις εργασίες που έχει υποβάλει, να παρακολουθήσει την κατάσταση κάθε paper, να λάβει ενημέρωση για αποδοχή ή απόρριψη, να κατεβάσει το αρχείο της υποβολής του και, εφόσον η εργασία έχει ανατεθεί σε track, να προχωρήσει σε πληρωμή του παραβόλου.

Βασικός χειριστής: Presenter

Δευτερεύων χειριστής: Stripe / Πάροχος πληρωμών

Συνθήκη εισόδου: Ο χρήστης έχει συνδεθεί στο σύστημα και έχει υποβάλει τουλάχιστον ένα paper σε συνέδριο.

Συνθήκη εξόδου: Ο presenter έχει προβάλει τις υποβολές του, έχει ενημερωθεί για την κατάστασή τους ή έχει ολοκληρώσει επιτυχώς τη διαδικασία πληρωμής για εργασία που έχει ανατεθεί σε track.

Κανονική ροή γεγονότων για προβολή υποβολών:

1. Ο presenter μεταβαίνει στον προσωπικό του πίνακα ή στη σελίδα του συνεδρίου.
2. Το σύστημα εμφανίζει τις υποβολές που έχει πραγματοποιήσει.
3. Για κάθε υποβολή εμφανίζεται ο τίτλος, το συνέδριο, η κατάσταση και, όπου υπάρχει, το track.
4. Ο presenter μπορεί να δει την τρέχουσα κατάσταση του paper.
5. Αν έχει επισυναφθεί αρχείο, ο presenter μπορεί να το κατεβάσει.
6. Αν η εργασία έχει γίνει accepted ή rejected, το σύστημα εμφανίζει σχετική ειδοποίηση.

Κανονική ροή γεγονότων για πληρωμή παραβόλου:

1. Το paper έχει γίνει accepted από τη διαδικασία αξιολόγησης.
2. Ο organizer έχει αναθέσει την εργασία σε track.
3. Το σύστημα εμφανίζει τη δυνατότητα πληρωμής.
4. Ο presenter επιλέγει την πληρωμή παραβόλου.
5. Το σύστημα τον μεταφέρει στο περιβάλλον Stripe Checkout.
6. Ο presenter ολοκληρώνει τη διαδικασία πληρωμής.
7. Το σύστημα επιβεβαιώνει την επιτυχή πληρωμή.
8. Η κατάσταση της εργασίας μεταβάλλεται σε presentationready.
9. Η εργασία μπορεί πλέον να εμφανιστεί στο πρόγραμμα παρουσιάσεων.

Εναλλακτική ροή γεγονότων:

- Αν το paper δεν έχει ανατεθεί σε track, δεν εμφανίζεται δυνατότητα πληρωμής.

- Αν η πληρωμή αποτύχει ή δεν ολοκληρωθεί, το paper μεταβαίνει ή παραμένει σε κατάσταση paymentpending.
- Αν το paper βρίσκεται ακόμη σε κατάσταση submitted ή under review, ο presenter μπορεί μόνο να παρακολουθεί την κατάστασή του.
- Αν το paper έχει απορριφθεί, δεν ενεργοποιείται διαδικασία πληρωμής.

Σημείωση:

Η λήψη αρχείου, η πληρωμή και η επιβεβαίωση επιτυχούς πληρωμής είναι λειτουργίες που εκτελούνται υπό προϋποθέσεις και δεν είναι υποχρεωτικά βήματα σε κάθε υποβολή.



Εικόνα 4 — Διάγραμμα Διαχείρισης υποβολών από Presenter

3.4.4 Τίτλος περίπτωσης χρήσης: Αξιολόγηση εργασιών από Reviewer

Αναγνωριστικό: PX5

Σύντομη περιγραφή: Η παρούσα περίπτωση χρήσης περιγράφει τη διαδικασία αξιολόγησης εργασιών από χρήστη που έχει οριστεί ως Reviewer σε συγκεκριμένο συνέδριο. Ο reviewer μπορεί να δει τις εργασίες που του έχουν ανατεθεί, να επιλέξει paper προς αξιολόγηση, να κατεβάσει το αρχείο της εργασίας και να καταχωρήσει αξιολόγηση με πρόταση, βαθμολογία και σχόλια.

Βασικός χειριστής: Reviewer

Συνθήκη εισόδου: Ο χρήστης έχει συνδεθεί στο σύστημα, έχει ρόλο Reviewer στο συγκεκριμένο συνέδριο και του έχει ανατεθεί τουλάχιστον ένα paper προς αξιολόγηση.

Συνθήκη εξόδου: Ο reviewer έχει υποβάλει αξιολόγηση για την εργασία ή έχει επιστρέψει στο review queue χωρίς να πραγματοποιήσει ενέργεια.

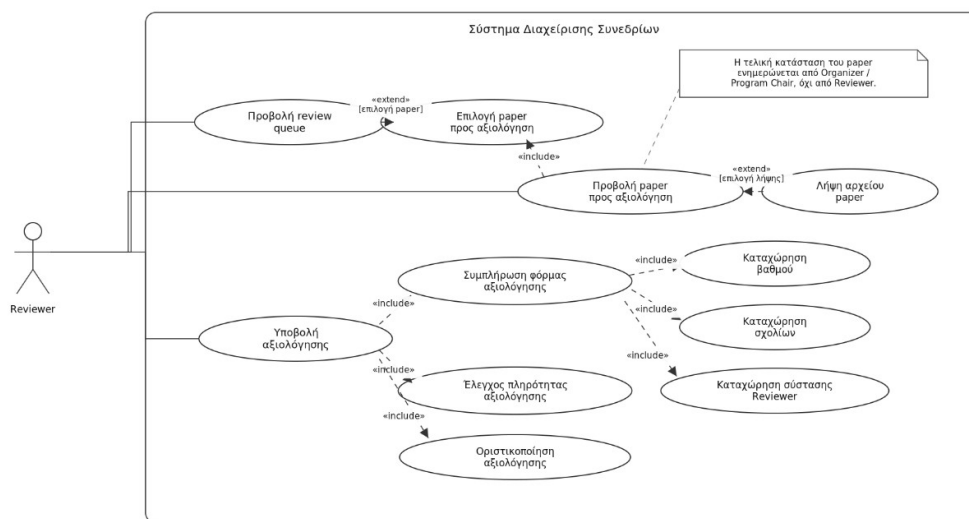
Κανονική ροή γεγονότων:

1. Ο reviewer συνδέεται στο σύστημα.
2. Μεταβαίνει στον προσωπικό του πίνακα ή στη σελίδα του συνεδρίου.
3. Το σύστημα εμφανίζει το review queue με τις εκκρεμείς αξιολογήσεις.
4. Ο reviewer επιλέγει paper προς αξιολόγηση.
5. Το σύστημα εμφανίζει τα στοιχεία της υποβολής.
6. Ο reviewer μπορεί να κατεβάσει το αρχείο του paper.
7. Ο reviewer συμπληρώνει τη φόρμα αξιολόγησης.
8. Καταχωρεί τη σύσταση αξιολόγησης, όπως αποδοχή, απόρριψη ή ανάγκη αλλαγών.
9. Καταχωρεί βαθμολογία.
10. Καταχωρεί σχόλια.
11. Το σύστημα ελέγχει την πληρότητα της αξιολόγησης.
12. Ο reviewer οριστικοποιεί την αξιολόγηση.
13. Η εκκρεμότητα αφαιρείται από το review queue του reviewer.

Εναλλακτική ροή γεγονότων:

- Αν ο reviewer δεν έχει εκκρεμείς αναθέσεις, το review queue εμφανίζεται κενό.
- Αν ο reviewer δεν συμπληρώσει όλα τα υποχρεωτικά πεδία της φόρμας, το σύστημα δεν επιτρέπει την υποβολή της αξιολόγησης.
- Αν ο reviewer επιλέξει επιστροφή, το σύστημα τον μεταφέρει στην προηγούμενη σελίδα χωρίς να αποθηκεύσει αξιολόγηση.
- Αν ο reviewer επιλέξει λήψη αρχείου, το σύστημα του παρέχει πρόσβαση μόνο στα papers που του έχουν ανατεθεί.

Σημείωση: Η τελική κατάσταση του paper δεν ενημερώνεται απευθείας από τον reviewer. Η τελική απόφαση προκύπτει από τη συνολική διαδικασία αξιολόγησης και ενημερώνεται από τον organizer ή από τη λογική του συστήματος.



Εικόνα 5 — Διάγραμμα Αξιολόγησης Εργασιών από Reviewer

3.4.5 Τίτλος περίπτωσης χρήσης: Διαχείριση συνεδρίου από Organizer

Αναγνωριστικό: ΠΧ6

Σύντομη περιγραφή: Η παρούσα περίπτωση χρήσης περιγράφει τις λειτουργίες που είναι διαθέσιμες στον Organizer ενός συνεδρίου. Ο organizer έχει αυξημένα δικαιώματα στο συγκεκριμένο συνέδριο και μπορεί να παρακολουθεί τις υποβολές, να διαχειρίζεται τα μέλη, να αναθέτει και να απενεργοποιεί ρόλους, να ορίζει tracks, να αναθέτει reviewers σε papers, να παρακολουθεί τα αποτελέσματα αξιολόγησης και να εντάσσει αποδεκτές εργασίες σε tracks.

Βασικός χειριστής: Organizer

Συνθήκη εισόδου: Ο χρήστης έχει συνδεθεί στο σύστημα και έχει ρόλο Organizer στο συγκεκριμένο συνέδριο.

Συνθήκη εξόδου: Ο organizer έχει πραγματοποιήσει μία ενέργεια διαχείρισης, όπως ανάθεση ρόλου, απενεργοποίηση ρόλου, ορισμό tracks, ανάθεση reviewers ή ανάθεση αποδεκτού paper σε track.

Κανονική ροή γεγονότων για διαχείριση μελών:

1. Ο organizer μεταβαίνει στη σελίδα του συνεδρίου.
2. Το σύστημα εμφανίζει τα μέλη του συνεδρίου και τους ρόλους τους.
3. Ο organizer μπορεί να αναθέσει ρόλο σε χρήστη με βάση το email του.
4. Το σύστημα ελέγχει αν ο χρήστης υπάρχει.
5. Το σύστημα καταχωρεί τον ρόλο για το συγκεκριμένο συνέδριο.
6. Ο organizer μπορεί να απενεργοποιήσει ρόλο μέλους, αν χρειάζεται.

Κανονική ροή γεγονότων για διαχείριση tracks:

1. Ο organizer επιλέγει τον ορισμό tracks.
2. Το σύστημα εμφανίζει τα υπάρχοντα tracks.
3. Ο organizer συμπληρώνει τα στοιχεία νέου track.
4. Το σύστημα ελέγχει ότι δεν ξεπερνιέται ο μέγιστος αριθμός tracks του συνεδρίου.
5. Το track καταχωρείται και εμφανίζεται στη σελίδα του συνεδρίου.

Κανονική ροή γεγονότων για ανάθεση reviewers:

1. Ο organizer βλέπει τις υποβολές του συνεδρίου.
2. Επιλέγει paper προς αξιολόγηση.
3. Το σύστημα εμφανίζει τους διαθέσιμους reviewers.
4. Ο organizer επιλέγει δύο διαφορετικούς reviewers.
5. Το σύστημα ελέγχει ότι οι reviewers είναι διαφορετικοί.
6. Το σύστημα καταχωρεί την ανάθεση.
7. Οι reviewers ενημερώνονται και το paper εμφανίζεται στο review queue τους.

Κανονική ροή γεγονότων για ανάθεση accepted paper σε track:

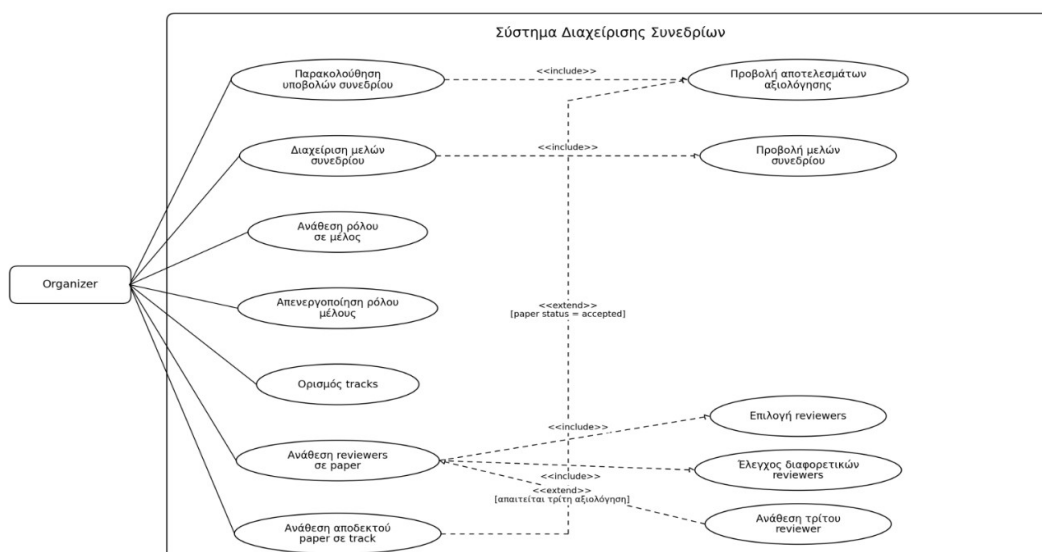
1. Το paper έχει ολοκληρώσει επιτυχώς τη διαδικασία αξιολόγησης και βρίσκεται σε κατάσταση accepted.
2. Ο organizer επιλέγει την ανάθεση σε track.
3. Το σύστημα εμφανίζει τα διαθέσιμα tracks του συνεδρίου.
4. Ο organizer επιλέγει track.

5. Το σύστημα ελέγχει τη χωρητικότητα του track.
6. Το paper ανατίθεται στο track.
7. Η κατάσταση του paper μεταβάλλεται σε track assigned.
8. Ενεργοποιείται η δυνατότητα πληρωμής από τον presenter.

Εναλλακτική ροή γεγονότων:

- Αν ο χρήστης δεν έχει ρόλο organizer στο συνέδριο, δεν εμφανίζονται οι λειτουργίες διαχείρισης.
- Αν το email χρήστη δεν αντιστοιχεί σε εγγεγραμμένο χρήστη, η ανάθεση ρόλου δεν ολοκληρώνεται.
- Αν επιλεγεί ο ίδιος reviewer δύο φορές για το ίδιο paper, το σύστημα δεν επιτρέπει την ανάθεση.
- Αν οι δύο reviewers διαφωνούν ως προς την αξιολόγηση, το paper μεταβαίνει σε κατάσταση needs third reviewer και ενεργοποιείται η ανάθεση τρίτου reviewer.
- Αν το paper δεν είναι accepted, δεν μπορεί να ανατεθεί σε track.
- Αν το track έχει συμπληρώσει τη χωρητικότητά του, δεν μπορεί να δεχτεί νέα εργασία.

Σημείωση: Ο organizer έχει δικαιώματα μόνο στα συνέδρια στα οποία έχει οριστεί με αυτόν τον ρόλο. Δεν μπορεί να διαχειριστεί συνολικά το σύστημα, εκτός αν διαθέτει και δικαιώματα staff ή superuser.



Εικόνα 6 — Διάγραμμα Διαχείρισης Συνεδρίου από Organizer

3.4.6 Τίτλος περίπτωσης χρήσης: Διαχείριση συστήματος από Superuser / Staff

Αναγνωριστικό: ΠΧ7

Σύντομη περιγραφή: Η παρούσα περίπτωση χρήσης περιγράφει τις λειτουργίες που είναι διαθέσιμες στον διαχειριστή συστήματος ή staff/superuser. Ο συγκεκριμένος χρήστης έχει τη δυνατότητα πρόσβασης στο περιβάλλον διαχείρισης, δημιουργίας νέων συνεδρίων, προβολής των υπαρχόντων συνεδρίων, αλλαγής της κατάστασής τους και πρόσβασης στο Django Administration.

Βασικός χειριστής: Διαχειριστής συστήματος / Staff / Superuser

Συνθήκη εισόδου: Ο χρήστης έχει συνδεθεί στο σύστημα και διαθέτει δικαιώματα staff ή superuser.

Συνθήκη εξόδου: Έχει δημιουργηθεί νέο συνέδριο, έχει προβληθεί ή ενημερωθεί υπάρχον συνέδριο, ή ο χρήστης έχει μεταβεί στο περιβάλλον διαχείρισης.

Κανονική ροή γεγονότων για δημιουργία συνεδρίου:

1. Ο διαχειριστής συνδέεται στο σύστημα.
2. Επιλέγει την πρόσβαση στο περιβάλλον διαχείρισης.
3. Το σύστημα εμφανίζει τη φόρμα δημιουργίας νέου συνεδρίου.
4. Ο διαχειριστής συμπληρώνει τον τίτλο του συνεδρίου.
5. Συμπληρώνει την περιγραφή και την τοποθεσία.
6. Ορίζει τις ημερομηνίες διεξαγωγής του συνεδρίου.
7. Ορίζει την περίοδο υποβολών.
8. Ορίζει τον μέγιστο αριθμό tracks.
9. Ορίζει το ποσό παραβόλου και το νόμισμα.
10. Επιλέγει την αρχική κατάσταση του συνεδρίου.
11. Το σύστημα ελέγχει την εγκυρότητα των στοιχείων.
12. Το συνέδριο δημιουργείται με αρχική κατάσταση draft ή με την κατάσταση που έχει επιλεγεί.

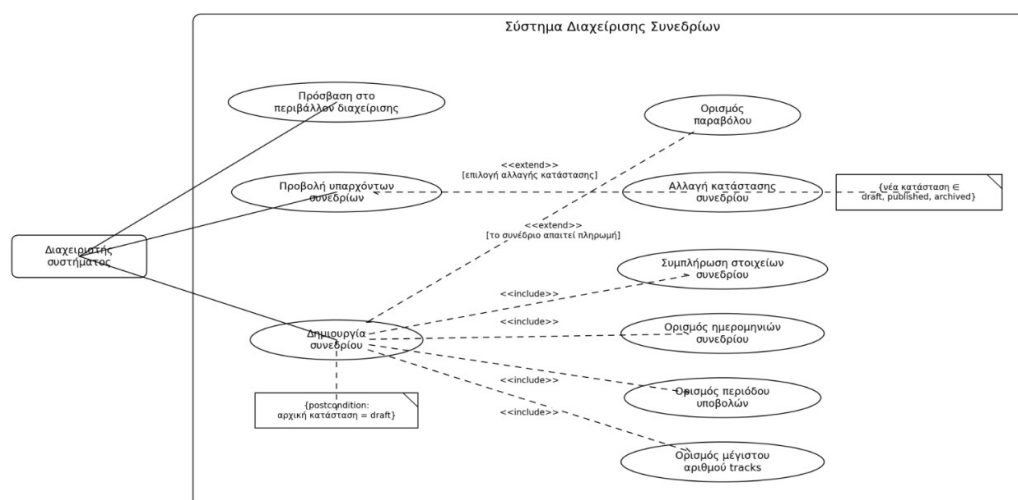
Κανονική ροή γεγονότων για διαχείριση υπάρχοντος συνεδρίου:

1. Ο διαχειριστής μεταβαίνει στη λίστα υπάρχόντων συνεδρίων.
2. Το σύστημα εμφανίζει τα συνέδρια, την κατάστασή τους, τον αριθμό tracks και τον αριθμό papers.
3. Ο διαχειριστής επιλέγει αλλαγή κατάστασης συνεδρίου.
4. Επιλέγει νέα κατάσταση, όπως draft, published ή archived.
5. Το σύστημα αποθηκεύει την αλλαγή.
6. Η δημόσια προβολή του συνεδρίου ενημερώνεται ανάλογα.

Εναλλακτική ροή γεγονότων:

- Αν η ημερομηνία λήξης υποβολών είναι μεταγενέστερη από την έναρξη του συνεδρίου, το σύστημα δεν επιτρέπει την καταχώρηση.
- Αν δεν συμπληρωθούν υποχρεωτικά πεδία, το σύστημα εμφανίζει μήνυμα σφάλματος.
- Αν το συνέδριο παραμένει σε κατάσταση draft, δεν εμφανίζεται στη δημόσια αρχική σελίδα.
- Αν το συνέδριο αλλάξει σε published, εμφανίζεται στη λίστα δημοσιευμένων συνεδρίων.
- Αν το συνέδριο αρχειοθετηθεί, μπορεί να παύσει να εμφανίζεται ως ενεργό.

Σημείωση: Η δημιουργία συνεδρίων πραγματοποιείται μόνο από χρήστες με δικαιώματα staff ή superuser. Ο organizer μπορεί να διαχειριστεί ένα συνέδριο στο οποίο έχει οριστεί, αλλά δεν έχει απαραίτητα δικαίωμα δημιουργίας νέων συνεδρίων.



Εικόνα 7 — Διάγραμμα Διαχείρισης Συστήματος από Superuser/Staff

3.5 Μοντελοποίηση Δεδομένων

Το μοντέλο δεδομένων σχεδιάστηκε με σαφείς συσχετίσεις μεταξύ οντοτήτων και αξιοποίηση περιορισμών/δεικτών για την επιβολή invariants. Ενδεικτικά παραδείγματα αποτελούν ο μοναδικός ρόλος ανά χρήστη/συνέδριο, η μοναδική ανάθεση slotreview ανά paper και η μοναδική συσχέτιση πληρωμής με υποβολή.

Οντότητες και περιγραφή

- **Χρήστης (User):** Εκπροσωπεί κάθε φυσικό πρόσωπο που χρησιμοποιεί το σύστημα. Οι χρήστες ταυτοποιούνται μέσω email (όχι username), έχουν εμφανιζόμενο όνομα (display_name), και μπορούν να έχουν διάφορους ρόλους ανά συνέδριο (organizer, reviewer, observer, presenter).
- **Συνέδριο (Conference):** Εκπροσωπεί κάποιο επιστημονικό συνέδριο ή ακαδημαϊκή διοργάνωση με τίτλο, ημερομηνίες, τοποθεσία, και κόστος υποβολής εργασίας για παρουσίαση. Κάθε συνέδριο έχει έναν διοργανωτή (organizer).
- **Track:** Υποκατηγορία / θεματική ενότητα ενός συνεδρίου. Περιλαμβάνει όνομα, περιγραφή και χωρητικότητα. Σχετίζεται με ένα συνέδριο και μπορεί να έχει πολλές υποβολές εργασιών/παρουσιάσεων.
- **Συμμετοχή σε Συνέδριο (ConferenceMembership):** Συνδέει έναν χρήστη με ένα συνέδριο και του αποδίδει έναν ρόλο (rganizer, reviewer, observer, presenter). Επιτρέπει την ύπαρξη του ίδιου χρήστη με διαφορετικούς ρόλους στο ίδιο συνέδριο.
- **Λίστα Παρακολούθησης (WatchList):** Επιτρέπει σε έναν χρήστη να παρακολουθεί (watch) ένα συνέδριο για να λαμβάνει ενημερώσεις. Μοναδικός συνδυασμός ανά ζεύγος χρήστη-συνεδρίου.
- **Υποβολή Εργασίας (PaperSubmission):** Μια επιστημονική εργασία που υποβάλλεται σε ένα συνέδριο από έναν χρήστη (presenter). Περιλαμβάνει τίτλο, περίληψη, λέξεις-κλειδιά, αρχείο, track, status και track_note.
- **Ανάθεση Κρίσης (PaperReviewAssignment):** Αναθέτει έναν κριτή (reviewer) σε μια συγκεκριμένη υποβολή εργασίας, με συγκεκριμένη θέση (slot: 1ος, 2ος, 3ος κριτής). Περιλαμβάνει ποιος έκανε την ανάθεση (assigned_by) και αν είναι ενεργή.
- **Αξιολόγηση (PaperReview):** Η πραγματική κρίση που γράφει ένας κριτής για μια υποβολή. Περιλαμβάνει απόφαση (accept/reject/request_changes), βαθμολογία (score) και σχόλια (comments).

- **Πληρωμή (Payment):** Καταγράφει μια πληρωμή για μια υποβολή εργασίας. Περιλαμβάνει ποσό, νόμισμα, πάροχο (provider), κατάσταση (pending/processing/succeeded/failed/canceled) και metadata από τον πάροχο.
- **Webhook Πληρωμής (PaymentWebhookEvent):** Χρησιμοποιείται για idempotency, ώστε να μην μπορεί να γίνει η ίδια πληρωμή παραπάνω από μια φορές, και καταγραφή εισερχόμενων webhooks από παρόχους πληρωμών. Δεν συνδέεται άμεσα με άλλες οντότητες μέσω FK, αλλά λειτουργεί ως log/audit.
- **Ειδοποίηση (Notification):** Μήνυμα που αποστέλλεται σε έναν χρήστη (recipient) για γεγονότα όπως κρίση, πληρωμή, ανάθεση παρουσίασης σε track, κ.λπ. Περιλαμβάνει κατηγορία, τίτλο, μήνυμα, κατάσταση ανάγνωσης (is_read) και προαιρετική αναφορά (GenericForeignKey) σε οποιοδήποτε μοντέλο.

Σχέσεις

User (Χρήστης)

- **User ↔ Conference (ως διοργανωτής):** Σχέση 1-προς-N. Ένας χρήστης μπορεί να είναι διοργανωτής πολλών συνεδρίων, ενώ ένα συνέδριο έχει ακριβώς έναν διοργανωτή (on_delete=PROTECT).
- **User ↔ Conference (μέσω ConferenceMembership):** Σχέση N-προς-N (με ρόλο). Ένας χρήστης μπορεί να έχει πολλαπλούς ρόλους (organizer, reviewer, observer, presenter) σε πολλά συνέδρια, και ένα συνέδριο μπορεί να έχει πολλούς χρήστες με διαφορετικούς ρόλους. Μοναδικός συνδυασμός ανά (conference, user, role).
- **User ↔ Conference (μέσω WatchList):** Σχέση N-προς-N. Ένας χρήστης μπορεί να παρακολουθεί πολλά συνέδρια, και ένα συνέδριο μπορεί να παρακολουθείται από πολλούς χρήστες. Μοναδικός συνδυασμός ανά (user, conference).
- **User ↔ PaperSubmission (ως presenter):** Σχέση 1-προς-N. Ένας χρήστης μπορεί να υποβάλει πολλές εργασίες, ενώ κάθε υποβολή έχει έναν μοναδικό παρουσιαστή (on_delete=PROTECT).
- **User ↔ Payment (ως payer):** Σχέση 1-προς-N. Ένας χρήστης μπορεί να πραγματοποιήσει πολλές πληρωμές, ενώ κάθε πληρωμή αντιστοιχεί σε έναν μοναδικό χρήστη (on_delete=PROTECT).

- **User ↔ PaperReviewAssignment (ως reviewer):** Σχέση 1-προς-N. Ένας χρήστης μπορεί να έχει ανατεθεί ως κριτής σε πολλές υποβολές (on_delete=PROTECT).
- **User ↔ PaperReviewAssignment (ως assigned_by):** Σχέση 1-προς-N. Ένας χρήστης (π.χ. διοργανωτής) μπορεί να κάνει πολλές αναθέσεις κρίσεων (on_delete=PROTECT).
- **User ↔ PaperReview (ως reviewer):** Σχέση 1-προς-N. Ένας χρήστης μπορεί να γράψει πολλές αξιολογήσεις (μία ανά υποβολή που κρίνει) (on_delete=CASCADE).
- **User ↔ Notification:** Σχέση 1-προς-N. Ένας χρήστης μπορεί να λάβει πολλές ειδοποιήσεις, ενώ κάθε ειδοποίηση απευθύνεται σε έναν μοναδικό παραλήπτη (on_delete=CASCADE).

Conference (Συνέδριο)

- **Conference ↔ User (ως διοργανωτής):** Σχέση N-προς-1. Κάθε συνέδριο έχει έναν διοργανωτή, ενώ ένας χρήστης μπορεί να διοργανώσει πολλά συνέδρια (on_delete=PROTECT).
- **Conference ↔ Track:** Σχέση 1-προς-N. Ένα συνέδριο μπορεί να έχει πολλά tracks, ενώ κάθε track ανήκει σε ένα μόνο συνέδριο (on_delete=CASCADE).
- **Conference ↔ ConferenceMembership:** Σχέση 1-προς-N. Ένα συνέδριο μπορεί να έχει πολλές εγγραφές μελών (ρόλους), ενώ κάθε εγγραφή αντιστοιχεί σε ένα μόνο συνέδριο (on_delete=CASCADE).
- **Conference ↔ WatchList:** Σχέση 1-προς-N. Ένα συνέδριο μπορεί να βρίσκεται στη λίστα παρακολούθησης πολλών χρηστών, ενώ κάθε εγγραφή watchlist αντιστοιχεί σε ένα μόνο συνέδριο (on_delete=CASCADE).
- **Conference ↔ PaperSubmission:** Σχέση 1-προς-N. Ένα συνέδριο μπορεί να έχει πολλές υποβολές εργασιών, ενώ κάθε υποβολή ανήκει σε ένα μόνο συνέδριο (on_delete=CASCADE).

Track

- **Track ↔ Conference:** Σχέση N-προς-1. Κάθε track ανήκει σε ένα συνέδριο, ενώ ένα συνέδριο μπορεί να έχει πολλά tracks (on_delete=CASCADE).
- **Track ↔ PaperSubmission:** Σχέση 1-προς-N (προαιρετική). Ένα track μπορεί να έχει πολλές υποβολές, ενώ κάθε υποβολή μπορεί να αντιστοιχεί σε ένα track (ή να μην έχει track, on_delete=SET_NULL).

PaperSubmission (Υποβολή)

- **PaperSubmission ↔ Conference:** Σχέση N-προς-1. Κάθε υποβολή ανήκει σε ένα συνέδριο, ενώ ένα συνέδριο μπορεί να έχει πολλές υποβολές (on_delete=CASCADE).
- **PaperSubmission ↔ User (presenter):** Σχέση N-προς-1. Κάθε υποβολή έχει έναν παρουσιαστή, ενώ ένας χρήστης μπορεί να έχει πολλές υποβολές (on_delete=PROTECT).
- **PaperSubmission ↔ Track:** Σχέση N-προς-1 (προαιρετική). Κάθε υποβολή μπορεί να αντιστοιχεί σε ένα track, ενώ ένα track μπορεί να έχει πολλές υποβολές (on_delete=SET_NULL).
- **PaperSubmission ↔ Payment:** Σχέση 1-προς-1. Κάθε υποβολή μπορεί να έχει το πολύ μία πληρωμή, και κάθε πληρωμή αντιστοιχεί σε μία μοναδική υποβολή (OneToOneField, on_delete=CASCADE).
- **PaperSubmission ↔ PaperReviewAssignment:** Σχέση 1-προς-N. Μία υποβολή μπορεί να έχει πολλές αναθέσεις κρίσης (π.χ. 1ος, 2ος, 3ος κριτής), ενώ κάθε ανάθεση αφορά μία συγκεκριμένη υποβολή (on_delete=CASCADE).
- **PaperSubmission ↔ PaperReview:** Σχέση 1-προς-N. Μία υποβολή μπορεί να έχει πολλές αξιολογήσεις (από διαφορετικούς κριτές), ενώ κάθε αξιολόγηση αντιστοιχεί σε μία υποβολή (on_delete=CASCADE).

Payment (Πληρωμή)

- **Payment ↔ PaperSubmission:** Σχέση 1-προς-1. Κάθε πληρωμή αντιστοιχεί σε μία μοναδική υποβολή, και κάθε υποβολή μπορεί να έχει το πολύ μία πληρωμή (OneToOneField, on_delete=CASCADE).
- **Payment ↔ User (payer):** Σχέση N-προς-1. Κάθε πληρωμή αντιστοιχεί σε έναν χρήστη, ενώ ένας χρήστης μπορεί να πραγματοποιήσει πολλές πληρωμές (on_delete=PROTECT).

PaperReviewAssignment (Ανάθεση κρίσης)

- **PaperReviewAssignment ↔ PaperSubmission:** Σχέση N-προς-1. Κάθε ανάθεση αφορά μία υποβολή, ενώ μία υποβολή μπορεί να έχει πολλές αναθέσεις (on_delete=CASCADE).
- **PaperReviewAssignment ↔ User (reviewer):** Σχέση N-προς-1. Κάθε ανάθεση αντιστοιχεί σε έναν κριτή, ενώ ένας χρήστης μπορεί να έχει πολλές αναθέσεις ως κριτής (on_delete=PROTECT).
- **PaperReviewAssignment ↔ User (assigned_by):** Σχέση N-προς-1. Κάθε ανάθεση γίνεται από έναν χρήστη (π.χ. διοργανωτή), ενώ ένας χρήστης μπορεί να κάνει πολλές αναθέσεις (on_delete=PROTECT).

PaperReview (Αξιολόγηση)

- **PaperReview ↔ PaperSubmission:** Σχέση N-προς-1. Κάθε αξιολόγηση αντιστοιχεί σε μία υποβολή, ενώ μία υποβολή μπορεί να έχει πολλές αξιολογήσεις (on_delete=CASCADE).
- **PaperReview ↔ User (reviewer):** Σχέση N-προς-1. Κάθε αξιολόγηση γράφεται από έναν κριτή, ενώ ένας χρήστης μπορεί να γράψει πολλές αξιολογήσεις (on_delete=CASCADE).

Notification (Ειδοποίηση)

- **Notification ↔ User (recipient):** Σχέση N-προς-1. Κάθε ειδοποίηση απευθύνεται σε έναν παραλήπτη, ενώ ένας χρήστης μπορεί να λάβει πολλές ειδοποιήσεις (on_delete=CASCADE).

PaymentWebhookEvent

- Το μοντέλο PaymentWebhookEvent είναι ανεξάρτητο (δεν έχει ξένα κλειδιά προς άλλες οντότητες). Χρησιμοποιείται για idempotency και καταγραφή εισερχόμενων webhooks από παρόχους πληρωμών.

Πίνακας user (Οντότητα: Χρήστης)

Πεδίο	Τύπος Δεδομένων	Περιγραφή	Σχόλια
email	EmailField (unique=True)	Διεύθυνση ηλεκτρονικού ταχυδρομείου	Μοναδικό, χρησιμοποιείται ως USERNAME_FIELD
display_name	CharField (max_length=150, blank=True)	Εμφανιζόμενο όνομα	Προαιρετικό (blank=True)
username	None	Απενεργοποιημένο πεδίο	Ορίζεται ρητά ως None για απενεργοποίηση του default username

Πίνακας 2 Πεδία του πίνακα Χρήστης

Πίνακας Conference (Οντότητα: Συνέδριο)

Πεδίο	Τύπος Δεδομένων	Περιγραφή	Σχόλια
title	CharField (max_length=255)	Τίτλος συνεδρίου	—
slug	SlugField (max_length=255, unique=True)	Φιλική προς URL εκδοχή του τίτλου	Μοναδικό, δημιουργείται αυτόματα από τον τίτλο αν δεν δοθεί
description	TextField (blank=True)	Περιγραφή συνεδρίου	Προαιρετικό
venue	CharField (max_length=255, blank=True)	Τοποθεσία διεξαγωγής	Προαιρετικό
status	CharField (max_length=20, choices=ConferenceStatus.choices)	Κατάσταση συνεδρίου	Επιλογές: draft (default), published, archived
registration_start	DateTimeField	Ημερομηνία έναρξης εγγραφών	Υποχρεωτικό
registration_end	DateTimeField	Ημερομηνία λήξης εγγραφών	Υποχρεωτικό
start_date	DateTimeField	Ημερομηνία έναρξης συνεδρίου	Υποχρεωτικό

end_date	DateTimeField	Ημερομηνία λήξης συνεδρίου	Υποχρεωτικό
max_tracks	PositiveSmallIntegerField (default=5)	Μέγιστος αριθμός tracks	Προεπιλογή 5
submission_fee	DecimalField (max_digits=10, decimal_places=2)	Κόστος υποβολής εργασίας	Προεπιλογή 0.00
currency	CharField (max_length=3, default="EUR")	Νόμισμα	Προεπιλογή EUR
organizer	ForeignKey (AUTH_USER_MODEL, on_delete=PROTECT)	Διοργανωτής συνεδρίου	related_name="organizing_conferences"
created_at	DateTimeField (auto_now_add=True)	Ημερομηνία δημιουργίας	Κληρονομείται από TimeStampedModel
updated_at	DateTimeField (auto_now=True)	Ημερομηνία τελευταίας ενημέρωσης	Κληρονομείται από TimeStampedModel

Πίνακας 3 Πεδία του πίνακα Συνέδριο

Πίνακας Track (Οντότητα: Θεματική Ενότητα)

Πεδίο	Τύπος Δεδομένων	Περιγραφή	Σχόλια
conference	ForeignKey (Conference, on_delete=CASCADE)	Συνέδριο στο οποίο ανήκει το track	related_name="tracks"
name	CharField (max_length=120)	Όνομα track	—
description	TextField (blank=True)	Περιγραφή track	Προαιρετικό
capacity	PositiveIntegerField (default=0)	Μέγιστη χωρητικότητα	0 = χωρίς περιορισμό
created_at	DateTimeField (auto_now_add=True)	Ημερομηνία δημιουργίας	Κληρονομείται από TimeStampedModel
updated_at	DateTimeField (auto_now=True)	Ημερομηνία τελευταίας ενημέρωσης	Κληρονομείται από TimeStampedModel

Πίνακας 4 Πεδία του πίνακα Θεματική Ενότητα

Πίνακας ConferenceMembership (Οντότητα: Συμμετοχή σε Συνέδριο)

Πεδίο	Τύπος Δεδομένων	Περιγραφή	Σχόλια
conference	ForeignKey (Conference, on_delete=CASCADE)	Συνέδριο	related_name="memberships"
user	ForeignKey (AUTH_USER_MODEL, on_delete=CASCADE)	Χρήστης	related_name="conference_roles"
role	CharField (max_length=20, choices=ConferenceMembershipRole.choices)	Ρόλος χρήστη στο συνέδριο	Επιλογές: organizer, reviewer, observer, presenter
is_active	BooleanField (default=True)	Ενεργή ιδιότητα μέλους	—
notes	TextField (blank=True)	Σημειώσεις	Προαιρετικό
created_at	DateTimeField (auto_now_add=True)	Ημερομηνία δημιουργίας	Κληρονομείται από TimeStampedModel
updated_at	DateTimeField (auto_now=True)	Ημερομηνία τελευταίας ενημέρωσης	Κληρονομείται από TimeStampedModel

Πίνακας 5 Πεδία του πίνακα Συμμετοχή σε Συνέδριο

Πίνακας Watchlist (Οντότητα: Λίστα Παρακολούθησης)

Πεδίο	Τύπος Δεδομένων	Περιγραφή	Σχόλια
user	ForeignKey (AUTH_USER_MODEL, on_delete=CASCADE)	Χρήστης που παρακολουθεί	related_name="watchlist_items"
conference	ForeignKey (Conference, on_delete=CASCADE)	Συνέδριο που παρακολουθείται	related_name="watchers"
created_at	DateTimeField (auto_now_add=True)	Ημερομηνία δημιουργίας	Κληρονομείται από TimeStampedModel
updated_at	DateTimeField (auto_now=True)	Ημερομηνία τελευταίας ενημέρωσης	Κληρονομείται από TimeStampedModel

Πίνακας 6 Πεδία του πίνακα Λίστα Παρακολούθηση

Πίνακας Notificationbn (Οντότητα: Ειδοποίηση)

Πεδίο	Τύπος Δεδομένων	Περιγραφή	Σχόλια
recipient	ForeignKey (AUTH_USER_MODEL, on_delete=CASCADE)	Παραλήπτης της ειδοποίησης	related_name="notifications"
category	CharField (max_length=20, choices=NotificationCategory.choices)	Κατηγορία ειδοποίησης	Επιλογές: info (default), review, payment, track
title	CharField (max_length=120)	Τίτλος ειδοποίησης	Υποχρεωτικό
message	TextField	Περιεχόμενο ειδοποίησης	Υποχρεωτικό
is_read	BooleanField (default=False)	Ένδειξη αν έχει διαβαστεί	—
target_content_type	ForeignKey (ContentType, on_delete=CASCADE, null=True, blank=True)	Τύπος περιεχομένου του σχετιζόμενου αντικειμένου	Για GenericForeignKey
target_object_id	PositiveBigIntegerField (null=True, blank=True)	ID του σχετιζόμενου αντικειμένου	Για GenericForeignKey
target_object	GenericForeignKey	Γενική αναφορά σε οποιοδήποτε μοντέλο	Δεν αποθηκεύεται ως στήλη στη βάση
created_at	DateTimeField (auto_now_add=True)	Ημερομηνία/ώρα δημιουργίας ειδοποίησης	Ορίζεται αυτόματα
read_at	DateTimeField (null=True, blank=True)	Ημερομηνία/ώρα ανάγνωσης	Null αν δεν έχει διαβαστεί ακόμα

Πίνακας 7 Πεδία του πίνακα Έιδοποίηση

Πίνακας Payment (Οντότητα: Πληρωμή)

Πεδίο	Τύπος Δεδομένων	Περιγραφή	Σχόλια
paper	OneToOneField (PaperSubmission, on_delete=CASCADE)	Η υποβολή εργασίας για την οποία γίνεται η πληρωμή	related_name="payment"
payer	ForeignKey (AUTH_USER_MODEL, on_delete=PROTECT)	Χρήστης που πληρώνει	related_name="payments"
amount	DecimalField (max_digits=10, decimal_places=2)	Ποσό πληρωμής	Υποχρεωτικό
currency	CharField (max_length=3, default="EUR")	Νόμισμα	Προεπιλογή EUR
provider	CharField (max_length=30, default="sandbox")	Πάροχος πληρωμών	Π.χ. stripe, paypal, sandbox
provider_payment_id	CharField (max_length=180, unique=True, blank=True, null=True)	Μοναδικό ID πληρωμής από τον πάροχο	Επιτρέπει null/κενό
status	CharField (max_length=20, choices=PaymentStatus.choices)	Κατάσταση πληρωμής	Επιλογές: pending (default), processing, succeeded, failed, canceled
failure_reason	TextField (blank=True)	Αιτία αποτυχίας	Προαιρετικό
provider_metadata	JSONField (default=dict, blank=True)	Πρόσθετα μεταδεδομένα από τον πάροχο	Default = κενό dict
created_at	DateTimeField (auto_now_add=True)	Ημερομηνία/ώρα δημιουργίας	—
updated_at	DateTimeField (auto_now=True)	Ημερομηνία/ώρα τελευταίας ενημέρωσης	—

Πίνακας 8 Πεδία του πίνακα Πληρωμή

Πίνακας PaymentWebhookEvent (Οντότητα: Συμβάν Πληρωμής)

Πεδίο	Τύπος Δεδομένων	Περιγραφή	Σχόλια
provider	CharField (max_length=30)	Πάροχος πληρωμών που έστειλε το webhook	Υποχρεωτικό
event_id	CharField (max_length=255, unique=True)	Μοναδικό ID του συμβάντος από τον πάροχο	Για idempotency
event_type	CharField (max_length=120, blank=True)	Τύπος συμβάντος (π.χ. payment.success)	Προαιρετικό
status	CharField (max_length=20, choices=PaymentWebhookEventStatus.choices)	Κατάσταση επεξεργασίας	Επιλογές: received (default), processed, ignored, failed
processed_at	DateTimeField (null=True, blank=True)	Πότε επεξεργάστηκε το webhook	Null αν δεν έχει επεξεργαστεί
error	TextField (blank=True)	Μήνυμα σφάλματος (αν αποτυχία)	—
created_at	DateTimeField (auto_now_add=True)	Πότε ελήφθη το webhook	—

Πίνακας 9 Πεδία του πίνακα Συμβάν Πληρωμής

Πίνακας PaperReviewAssignment (Οντότητα: Ανάθεση Αξιολόγησης)

Πεδίο	Τύπος Δεδομένων	Περιγραφή	Σχόλια
paper	ForeignKey (PaperSubmission, on_delete=CASCADE)	Η υποβολή εργασίας που ανατίθεται για κριτική	related_name="review_assignments"

reviewer	ForeignKey (AUTH_USER_MODEL, on_delete=PROTECT)	Ο κριτής στον οποίο ανατίθεται η εργασία	related_name="review_assignments"
slot	PositiveSmallIntegerField (choices=ReviewSlot.choices)	Θέση κριτικής	Επιλογές: 1=First, 2=Second, 3=Third
assigned_by	ForeignKey (AUTH_USER_MODEL, on_delete=PROTECT)	Χρήστης που έκανε την ανάθεση	related_name="assigned_review_assignments"
is_active	BooleanField (default=True)	Ένδειξη αν η ανάθεση είναι ενεργή	—
created_at	DateTimeField (auto_now_add=True)	Ημερομηνία/ώρα δημιουργίας	Κληρονομείται από TimeStampedModel
updated_at	DateTimeField (auto_now=True)	Ημερομηνία/ώρα τελευταίας ενημέρωσης	Κληρονομείται από TimeStampedModel

Πίνακας 10 Πεδία του πίνακα Ανάθεση Αξιολόγησης

Πίνακας (Οντότητα: Αξιολόγηση)

Πεδίο	Τύπος Δεδομένων	Περιγραφή	Σχόλια
paper	ForeignKey (PaperSubmission, on_delete=CASCADE)	Η υποβολή εργασίας που αξιολογείται	related_name="reviews"
reviewer	ForeignKey (AUTH_USER_MODEL, on_delete=CASCADE)	Ο κριτής που έγραψε την αξιολόγηση	related_name="paper_reviews"
decision	CharField (max_length=20, choices=ReviewDecision.choices)	Απόφαση κριτή	Επιλογές: accept, reject, request_changes
score	PositiveSmallIntegerField (null=True, blank=True)	Βαθμολογία (π.χ. 1-10)	Προαιρετικό
comments	TextField (blank=True)	Σχόλια κριτή	Προαιρετικό
created_at	DateTimeField (auto_now_add=True)	Ημερομηνία/ώρα δημιουργίας	Κληρονομείται από TimeStampedModel

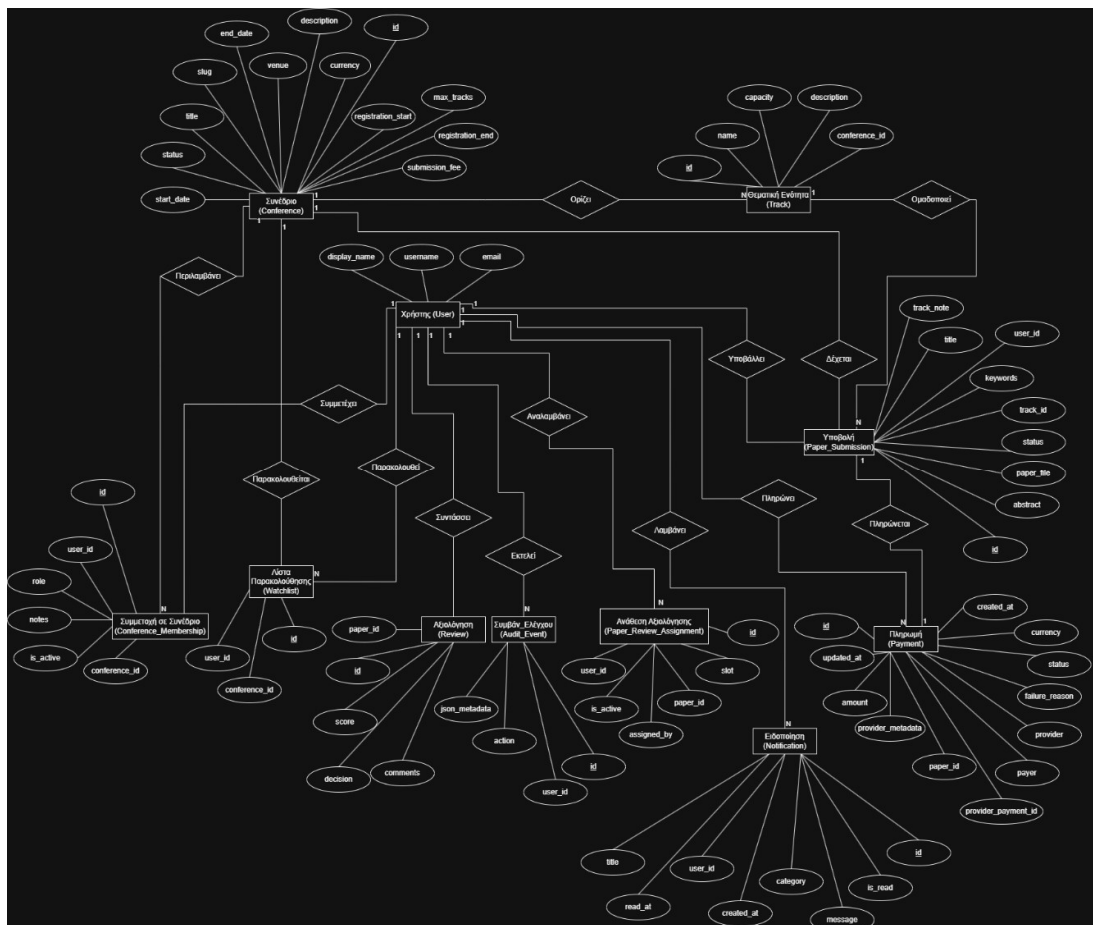
updated_at	DateTimeField (auto_now=True)	Ημερομηνία/ώρα τελευταίας ενημέρωσης	Κληρονομείται από TimeStampedModel
------------	-------------------------------	--	---------------------------------------

Πίνακας 11 Πεδία του πίνακα Αξιολόγηση

Πίνακας Submission (Οντότητα: Υποβολή)

Πεδίο	Τύπος Δεδομένων	Περιγραφή	Σχόλια
conference	ForeignKey (Conference, on_delete=CASCADE)	Συνέδριο στο οποίο υποβάλλεται η εργασία	related_name="submissions"
presenter	ForeignKey (AUTH_USER_MODEL, on_delete=PROTECT)	Χρήστης που υποβάλλει/παρουσιάζει την εργασία	related_name="submissions"
title	CharField (max_length=255)	Τίτλος εργασίας	Υποχρεωτικό
abstract	TextField	Περίληψη εργασίας	Υποχρεωτικό
keywords	CharField (max_length=255, blank=True)	Λέξεις-κλειδιά	Προαιρετικό
paper_file	FileField (upload_to="papers/%Y/%m/%d/", blank=True, null=True)	Αρχείο PDF της εργασίας	Προαιρετικό, αποθηκεύεται σε ημερομηνιακό φάκελο
track	ForeignKey (Track, on_delete=SET_NULL, null=True, blank=True)	Track στο οποίο αντιστοιχείται η εργασία	related_name="submissions", μπορεί να είναι null
status	CharField (max_length=25, choices=SubmissionStatus.choices)	Κατάσταση υποβολής	Επιλογές: draft, submitted (default), under_review, needs_third_review, accepted, track_assigned, payment_pending, presentation_ready, rejected
track_note	TextField (blank=True)	Σημειώσεις για το track	Προαιρετικό
created_at	DateTimeField (auto_now_add=True)	Ημερομηνία/ώρα δημιουργίας	Κληρονομείται από TimeStampedModel
updated_at	DateTimeField (auto_now=True)	Ημερομηνία/ώρα τελευταίας ενημέρωσης	Κληρονομείται από TimeStampedModel

Πίνακας 12 Πεδία του πίνακα Υποβολή



Εικόνα 8 Διάγραμμα Οντοτήτων Συσχετίσεων

4. Υλοποίηση Λειτουργιών

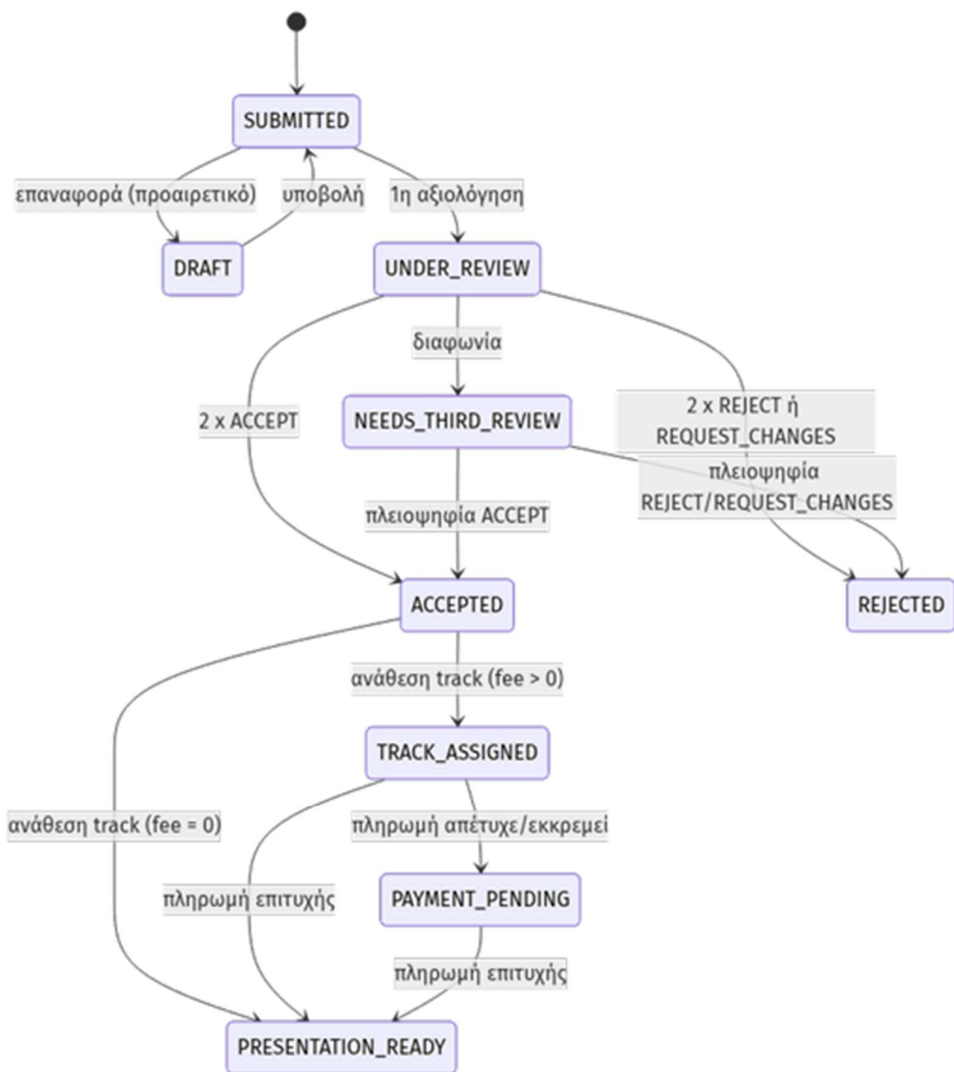
Το παρόν κεφάλαιο περιγράφει τις βασικές ροές που καλύπτουν τις λειτουργικές απαιτήσεις. Οι ροές σχεδιάστηκαν ώστε να είναι role-aware, να ακολουθούν συγκεκριμένες καταστάσεις (states) και να υποστηρίζονται από έλεγχο ακεραιότητας και καταγραφή κρίσιμων ενεργειών.

4.1 Διαχείριση Συνεδρίου και Ρόλων

Η δημιουργία συνεδρίου υλοποιείται μέσω staff-only διεπαφής και αντίστοιχου API, ώστε να αποτρέπεται η αυθαίρετη δημιουργία συνεδρίων από μη εξουσιοδοτημένους χρήστες. Ο ρόλος του organizer ορίζεται ως ιδιοκτησία του συνεδρίου και υποστηρίζεται επιπλέον μέσω memberships, ώστε να είναι δυνατή η ανάθεση πολλαπλών ρόλων και η εναλλαγή τους ανά συνέδριο. Η ανάθεση ρόλων (organizer, reviewer, observer, presenter) βασίζεται σε email χρήστη και αποτυπώνεται σε ConferenceMembership, με μοναδικότητα (conference, user, role). Η προσέγγιση αυτή αποτρέπει διπλοεγγραφές και διευκολύνει την ανάκτηση λιστών ανά ρόλο.

4.2 Υποβολή και Αξιολόγηση paper με Κανόνα Ομοφωνίας

Η ροή υποβολής ξεκινά με τη δημιουργία υποβολής σε κατάσταση submitted. Με την πρώτη ενέργεια αξιολόγησης, η κατάσταση μεταβαίνει σε under_review. Μετά την καταχώριση δύο αξιολογήσεων, το σύστημα υπολογίζει το αποτέλεσμα: αν υπάρχουν δύο αποφάσεις αποδοχής, η υποβολή γίνεται accepted. Αν υπάρχουν δύο αποφάσεις απόρριψης ή αιτήματος αλλαγών, η υποβολή γίνεται rejected. Σε αντίθετη περίπτωση, ορίζεται η κατάσταση needs_third_review και απαιτείται ανάθεση τρίτου reviewer. Μετά την τρίτη αξιολόγηση, εφαρμόζεται πλειοψηφικός κανόνας μεταξύ αποδοχής και απόρριψης/αιτήματος αλλαγών.



Εικόνα 9 Ροή Υποβολής και Αξιολόγησης Paper

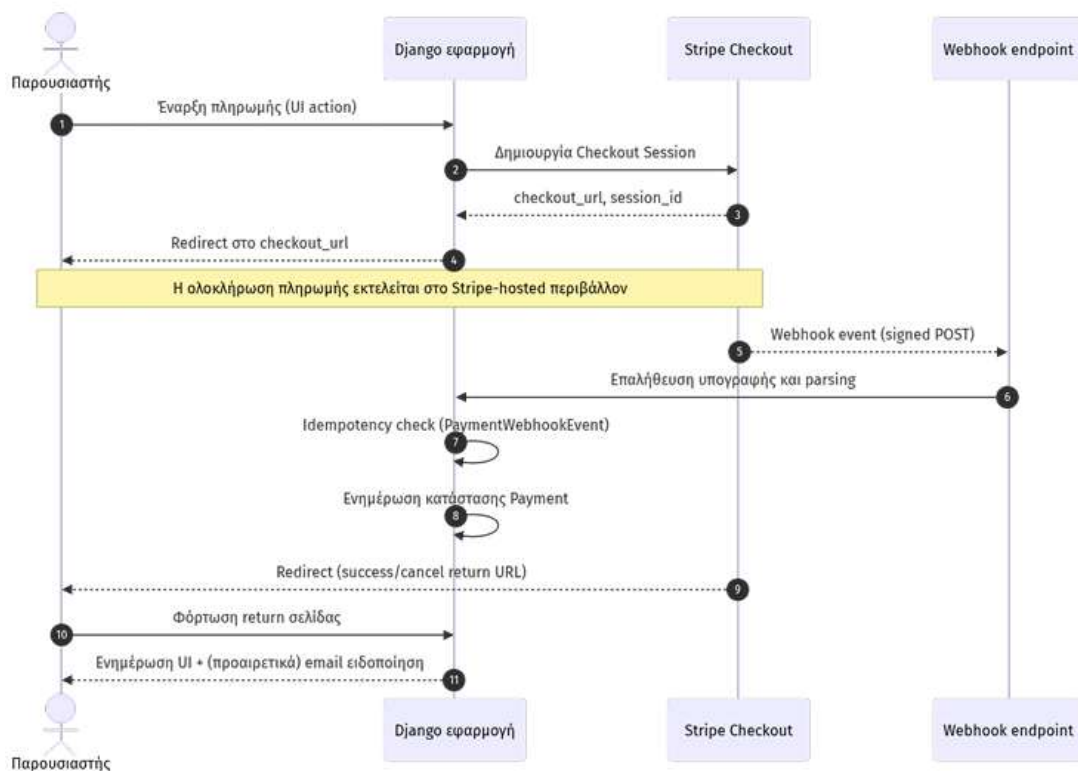
4.3 Ανάθεση track και ολοκλήρωση πληρωμής

Μετά την αποδοχή, το paper εντάσσεται σε track. Αν το συνέδριο έχει μηδενικό παράβολο, η ανάθεση track οδηγεί άμεσα σε presentation_ready. Σε περίπτωση θετικού παραβόλου, η ανάθεση track οδηγεί σε track_assigned και απαιτείται πληρωμή. Η πληρωμή καταγράφεται σε οντότητα Payment (1-προς-1 με υποβολή). Σε αποτυχία πληρωμής, το paper μεταβαίνει σε payment_pending, ενώ σε επιτυχία μεταβαίνει σε presentation_ready. Η υλοποίηση της πληρωμής σχεδιάστηκε με διάκριση μεταξύ της επιχειρησιακής οντότητας (Payment) και

του παρόχου πληρωμών. Για παραγωγικό σενάριο, ενσωματώθηκε πάροχος Stripe, αξιοποιώντας το Stripe Checkout ως hosted ροή πληρωμής, ώστε να μειώνεται η επιφάνεια ευθύνης της εφαρμογής ως προς την επεξεργασία στοιχείων κάρτας και να υιοθετείται τυποποιημένη ροή εξόφλησης (Stripe, χ.χ.-a).

4.3.1 Δημιουργία Checkout Session και επιστροφή στο σύστημα

Με την έναρξη της πληρωμής, ο server δημιουργεί Checkout Session μέσω API, αποθηκεύει στο `Payment.provider_metadata` το `checkout_url` και το `checkout_session_id`, και επιστρέφει στο UI το URL για ανακατεύθυνση. Με την ολοκλήρωση ή ακύρωση, ο χρήστης επιστρέφει σε σελίδες επιβεβαίωσης (return URLs) της εφαρμογής, όπου ενημερώνεται για την τρέχουσα κατάσταση της πληρωμής.



Εικόνα 10 Ακολουθία Πληρωμής με Stripe Checkout και webhooks

Η οριστικοποίηση της πληρωμής δεν βασίζεται αποκλειστικά στην ανακατεύθυνση του χρήστη. Για την αξιόπιστη ενημέρωση της κατάστασης, χρησιμοποιούνται webhooks από τον πάροχο (Stripe, χ.χ.-c). Κάθε webhook event επαληθεύεται με έλεγχο υπογραφής (signature verification) βάσει μυστικού (webhook secret), ώστε να απορρίπτονται πλαστά ή

αλλοιωμένα αιτήματα (Stripe, χ.χ. b). Για την αντιμετώπιση επαναλήψεων (retries) ή διπλών παραδόσεων, υλοποιήθηκε μηχανισμός idempotency σε επίπεδο δεδομένων μέσω της οντότητας PaymentWebhookEvent, η οποία αποθηκεύει το event_id του παρόχου με μοναδικότητα. Με τον τρόπο αυτό, το ίδιο event δεν επεξεργάζεται δεύτερη φορά, ακόμη και αν παραδοθεί πολλαπλές φορές.

4.3.2 Email ειδοποιήσεις και περιβάλλον ανάπτυξης

Η ροή πληρωμής συνδέθηκε με το υποσύστημα ειδοποιήσεων, με δυνατότητα αποστολής email για κρίσιμες μεταβάσεις (π.χ. ανάθεση track που απαιτεί πληρωμή, επιτυχία/αποτυχία πληρωμής). Η αποστολή είναι ρυθμιζόμενη μέσω μεταβλητών περιβάλλοντος (ενεργοποίηση/απενεργοποίηση, SMTP ρυθμίσεις) σύμφωνα με την τεκμηρίωση του Django (Django Software Foundation, χ.χ. b). Για τοπική ανάπτυξη χρησιμοποιείται Mailpit ως SMTP sink, ώστε τα email να επιθεωρούνται χωρίς εξωτερική παράδοση (Mailpit, χ.χ.)

4.4 Ειδοποιήσεις και παρακολούθηση συνεδρίων

Κάθε κρίσιμη αλλαγή κατάστασης (υποβολή, καταχώριση review, ανάγκη τρίτου review, αποδοχή/απόρριψη, ανάθεση track, αλλαγές πληρωμής) συνοδεύεται από δημιουργία ειδοποίησης προς τον εμπλεκόμενο χρήστη ή τους organizers. Επιπλέον, υποστηρίζεται watchlist μέσω συσχέτισης 20 χρήστη-συνεδρίου, ώστε ο χρήστης να παρακολουθεί επιλεγμένα συνέδρια ανεξάρτητα από ρόλους.

4.5 Καταστάσεις, μεταβάσεις και έλεγχοι ακεραιότητας

Η υλοποίηση αντιμετωπίζει τις βασικές οντότητες ως στοιχεία με ρητά ορισμένες καταστάσεις. Η υποβολή εργασίας δεν είναι απλή εγγραφή δεδομένων, αλλά οντότητα με κύκλο ζωής που περιλαμβάνει μεταβάσεις, οι οποίες επιτρέπονται μόνο όταν ικανοποιούνται προϋποθέσεις. Για παράδειγμα, μια υποβολή δεν μπορεί να θεωρηθεί αποδεκτή χωρίς να έχουν καταχωριστεί οι απαιτούμενες αξιολογήσεις, ενώ η ανάθεση track σε αποδεκτή υποβολή επιτρέπεται μόνο όταν η υποβολή βρίσκεται σε κατάσταση που το επιτρέπει. Η ρητή αποτύπωση των καταστάσεων λειτουργεί ως μηχανισμός μείωσης ασάφειας, διότι οι κανόνες μετατρέπονται σε ελέγξιμες προϋποθέσεις. Η επιβολή ακεραιότητας πραγματοποιείται σε δύο επίπεδα. Στο επίπεδο βάσης δεδομένων, εφαρμόζονται μοναδικότητες και δείκτες, ώστε να αποτρέπονται διπλοεγγραφές και αντιφατικές συσχετίσεις. Στο επίπεδο επιχειρησιακής λογικής, οι services εφαρμόζουν ελέγχους που βασίζονται στην τρέχουσα κατάσταση και στον ρόλο του χρήστη. Ο συνδυασμός των δύο

επιπέδων μειώνει τον κίνδυνο να παραχθούν μη έγκυρες καταστάσεις είτε από bug σε view είτε από μη αναμενόμενη ακολουθία αιτημάτων. Ειδική έμφαση δόθηκε στην αντιστοίχιση των transitions με γεγονότα που έχουν νόημα σε επίπεδο πεδίου. Η καταχώριση αξιολόγησης αποτελεί γεγονός που επηρεάζει την απόφαση αποδοχής, ενώ η ανάθεση track αποτελεί γεγονός που μεταφέρει την υποβολή σε φάση προετοιμασίας παρουσίασης ή σε φάση πληρωμής. Η σχεδίαση αυτή επιτρέπει πιο καθαρή τεκμηρίωση, διότι κάθε μετάβαση συνδέεται με αντίστοιχη ενέργεια και αντίστοιχες εγγραφές ειδοποιήσεων και audit events.

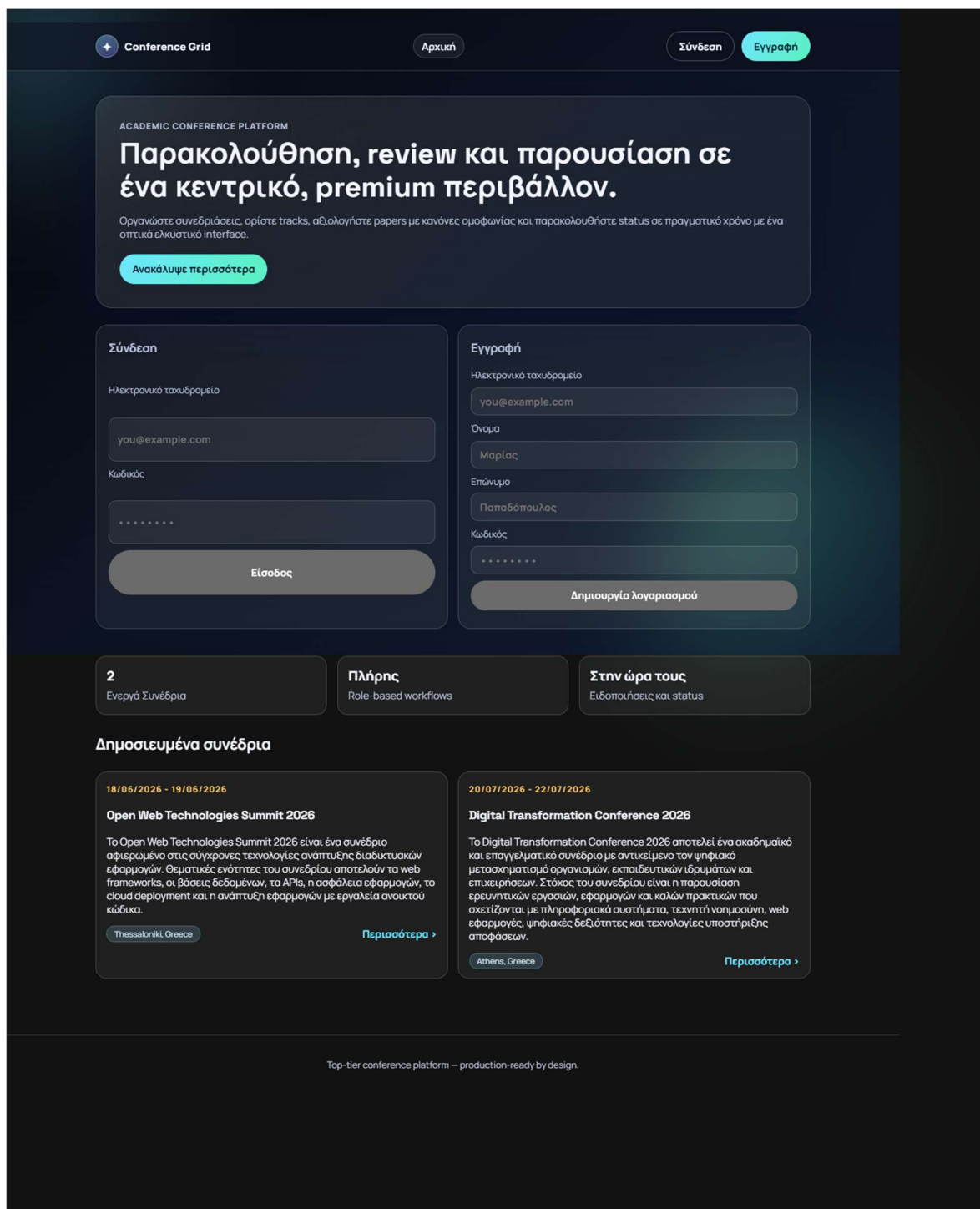
4.6 Πλοήγηση στην Εφαρμογή

Σε αυτή την ενότητα, θα παρουσιαστεί αναλυτικά:

- Η ροή πλοήγησης από οθόνη σε οθόνη.
- Η δομή της εφαρμογής και τα βασικά αρχεία.

4.6.1 Αρχική πρόσβαση και προβολή συνεδρίων

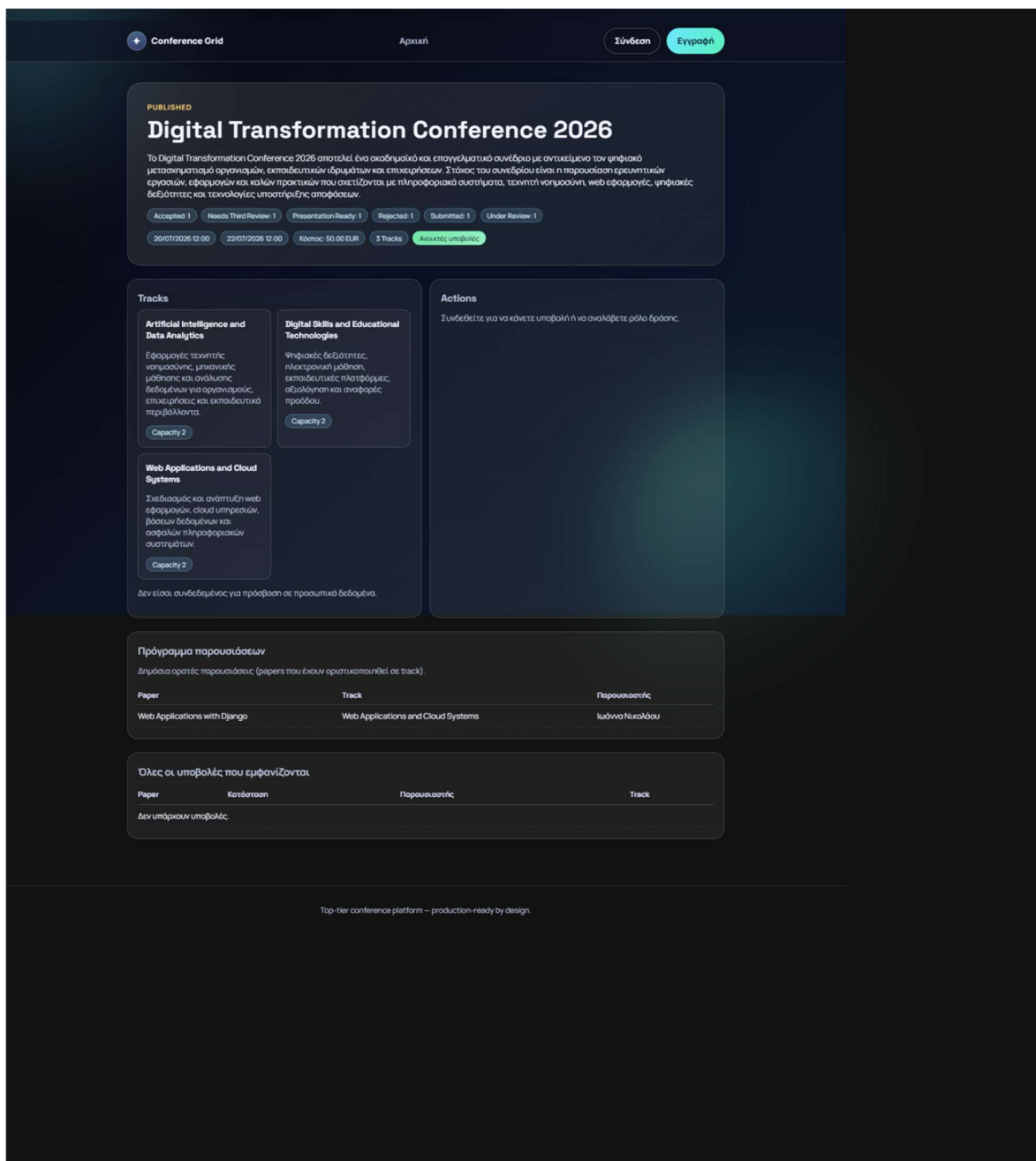
Στην αρχική οθόνη της εφαρμογής εμφανίζονται τα δημοσιευμένα συνέδρια που είναι διαθέσιμα στους επισκέπτες. Ο χρήστης που δεν έχει συνδεθεί μπορεί να δει βασικές πληροφορίες, να μεταβεί στη σελίδα λεπτομερειών ενός συνεδρίου ή να χρησιμοποιήσει τις επιλογές σύνδεσης και εγγραφής. Συνέδρια που βρίσκονται σε κατάσταση draft δεν εμφανίζονται στη δημόσια λίστα.



Εικόνα 11 Αρχική μη συνδεδεμένου χρήστη

Μετά τη σύνδεση του χρήστη η αρχική οθόνη προσαρμόζεται ανάλογα με την κατάσταση σύνδεσης. Οι επιλογές σύνδεσης και εγγραφής αντικαθίστανται από την ένδειξη του

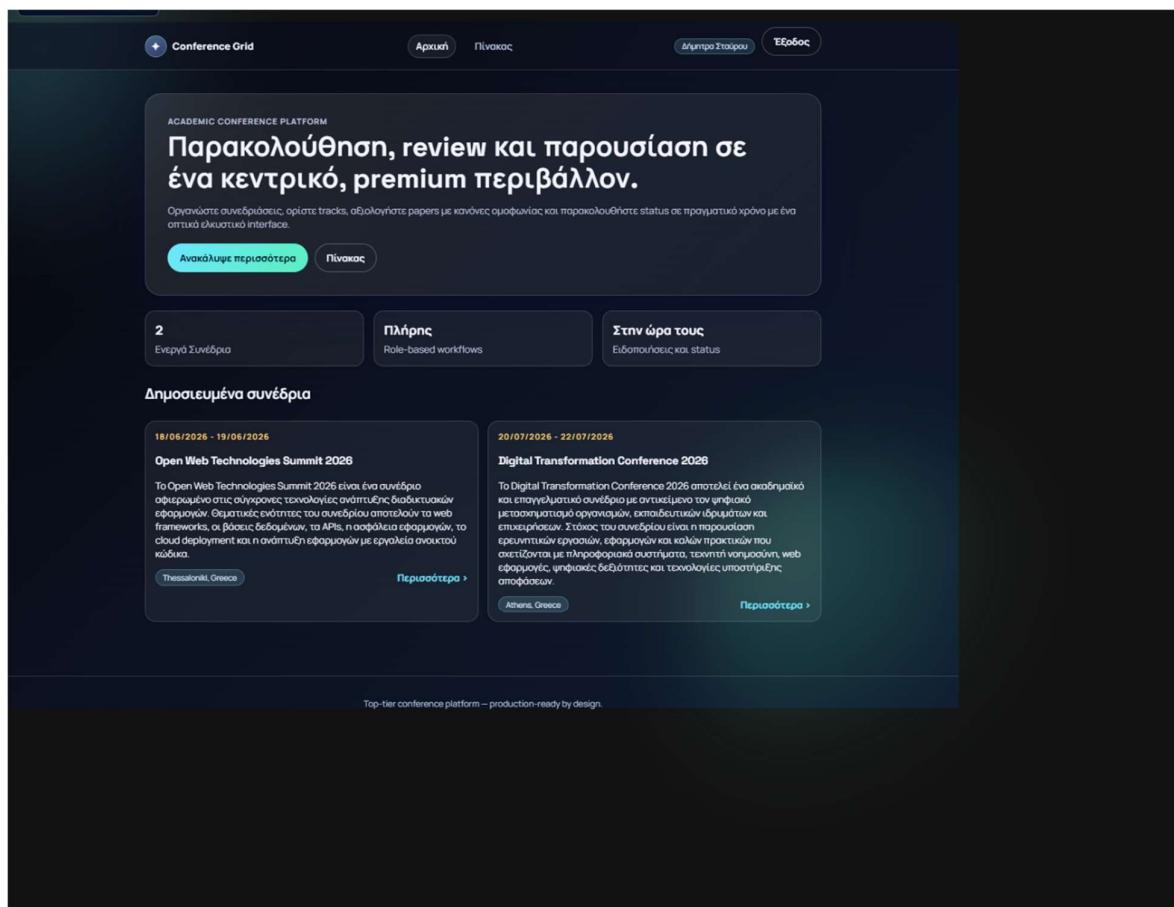
συνδεδεμένου χρήστη, την επιλογή εξόδου και το κουμπί μετάβασης στον προσωπικό πίνακα ελέγχου. Ο χρήστης παραμένει αρχικά χωρίς ειδικό ρόλο σε κάποιο συνέδριο.



Εικόνα 12 Σελίδα Συνεδρίου μη συνδεδεμένου χρήστη

Στη σελίδα πληροφοριών του συνεδρίου εμφανίζονται τα βασικά στοιχεία του, όπως η τοποθεσία, οι ημερομηνίες διεξαγωγής, το χρονικό διάστημα υποβολών, το κόστος συμμετοχής, τα διαθέσιμα tracks και τα στατιστικά των υποβολών. Επειδή ο χρήστης δεν

είναι συνδεδεμένος, δεν έχει πρόσβαση σε προσωπικές ενέργειες, όπως υποβολή εργασίας, εγγραφή ως observer ή προσθήκη στη watchlist.



Εικόνα 13 Αρχική Οθόνη συνδεδεμένου χρήστη

Όταν ο χρήστης είναι συνδεδεμένος, στη σελίδα του συνεδρίου εμφανίζονται πρόσθετες ενέργειες. Ο χρήστης μπορεί να προσθέσει το συνέδριο στη watchlist, χωρίς αυτό να του αποδίδει κάποιον ρόλο, να εγγραφεί ως observer ή να προχωρήσει σε νέα υποβολή εργασίας, εφόσον οι υποβολές του συνεδρίου είναι ανοιχτές.

Conference Grid

Αρχική Πίνακας

Δήμιτρα Στάυρου

Έξοδος

PUBLISHED

Digital Transformation Conference 2026

Το Digital Transformation Conference 2026 αποτελεί ένα ακαδημαϊκό και επαγγελματικό συνέδριο με αντικείμενο τον ψηφιακό μετασχηματισμό οργανισμών, εκπαιδευτικών ιδρυμάτων και επιχειρήσεων. Στόχος του συνεδρίου είναι η παρουσίαση ερευνητικών εργασιών, εφαρμογών και καλών πρακτικών που αφορούν με πληροφοριακά συστήματα, τεχνητή νοημοσύνη, web εφαρμογές, ψηφιακές δεξιότητες και τεχνολογίες υποστήριξης αποφάσεων.

Accepted: 1
Needs Third Review: 1
Presentation Ready: 1
Rejected: 1
Submitted: 1
Under Review: 1

Κατάργηση από Watchlist
20/07/2026 12:00
22/07/2026 12:00
Κόστος: 50.00 EUR
3 Tracks
Άνοιχτες υποβολές

Tracks

Artificial Intelligence and Data Analytics

Εφαρμογές τεχνητής νοημοσύνης, μηχανικής μάθησης και ανάλυσης δεδομένων για οργανισμούς, επιχειρήσεις και εκπαιδευτικά περιβάλλοντα.

Capacity 2

Digital Skills and Educational Technologies

Ψηφιακές δεξιότητες, ηλεκτρονική μάθηση, εκπαιδευτικές πλατφόρμες, αξιολόγηση και αναφορές προόδου.

Capacity 2

Web Applications and Cloud Systems

Σχεδιασμός και ανάπτυξη web εφαρμογών, cloud υπηρεσιών, βάσεων δεδομένων και ασφαλών πληροφοριακών συστημάτων.

Capacity 2

Οι συμμετέχοντες βλέπουν μόνο τις δικές σου υποβολές.

Actions

Νέα υποβολή paper

Τίτλος

Τίτλος paper

Abstract

Περίληψη έως 2-3 παραγράφων

Λέξεις-κλειδιά

machine learning, ai, vision

PDF (προαιρετικό)

Choose File

No file chosen

Καταχώρηση paper

Εγγραφή ως παρατηρητής

Εγγράψου ως observer για να παρακολουθείς το συνέδριο.

Εγγραφή Observer

Πρόγραμμα παρουσιάσεων

Δημόσια ορατές παρουσιάσεις (papers που έχουν οριστικοποιηθεί σε track).

Paper	Track	Παρουσιαστής
Web Applications with Django	Web Applications and Cloud Systems	Ιωάννα Νικολάου

Όλες οι υποβολές που εμφανίζονται

Paper	Κατάσταση	Παρουσιαστής	Track
Δεν υπάρχουν υποβολές.			

Top-tier conference platform – production-ready by design.

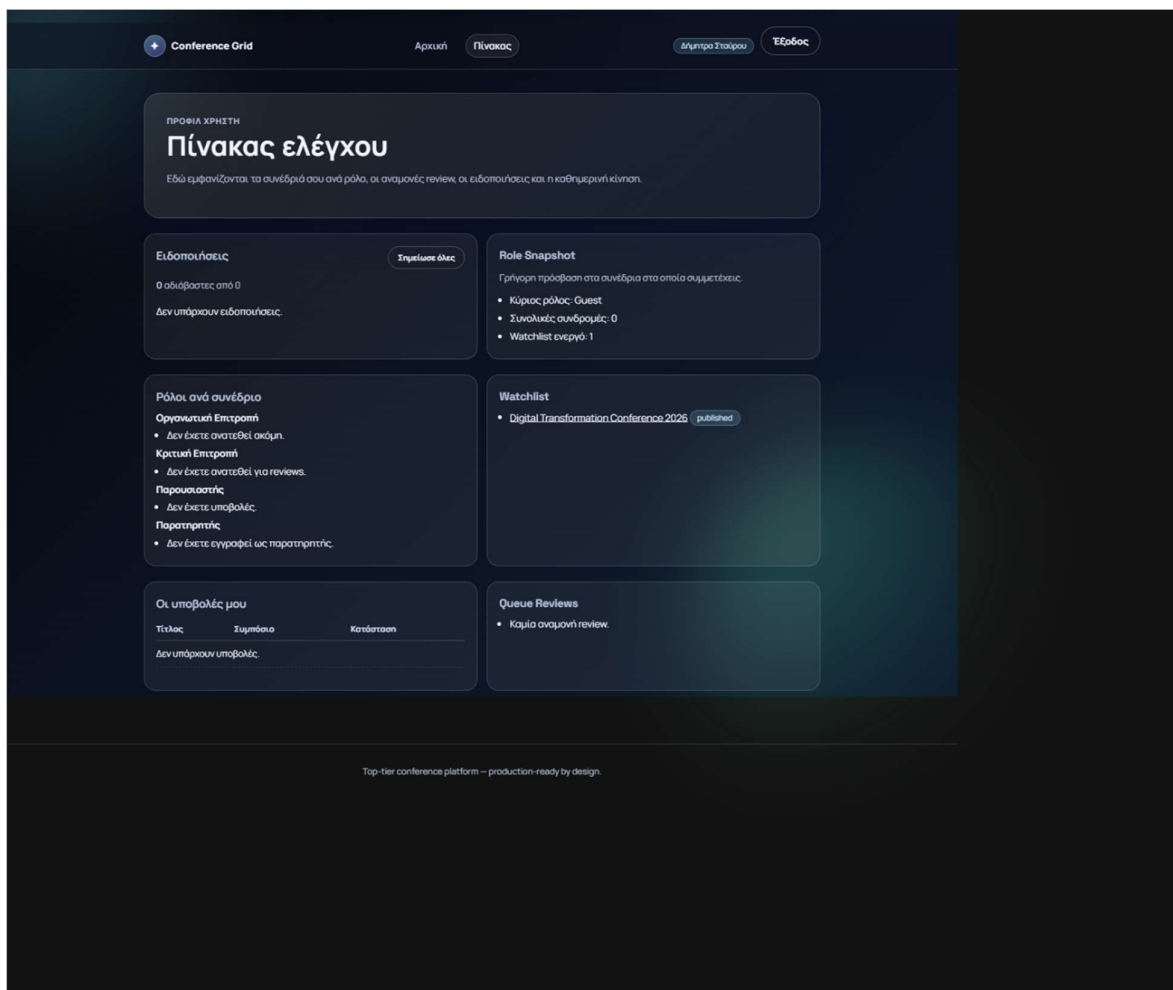
Εικόνα 14 Σελίδα Συνεδρίου Συνδεδεμένου Χρήστη

Διπλωματική Εργασία

68

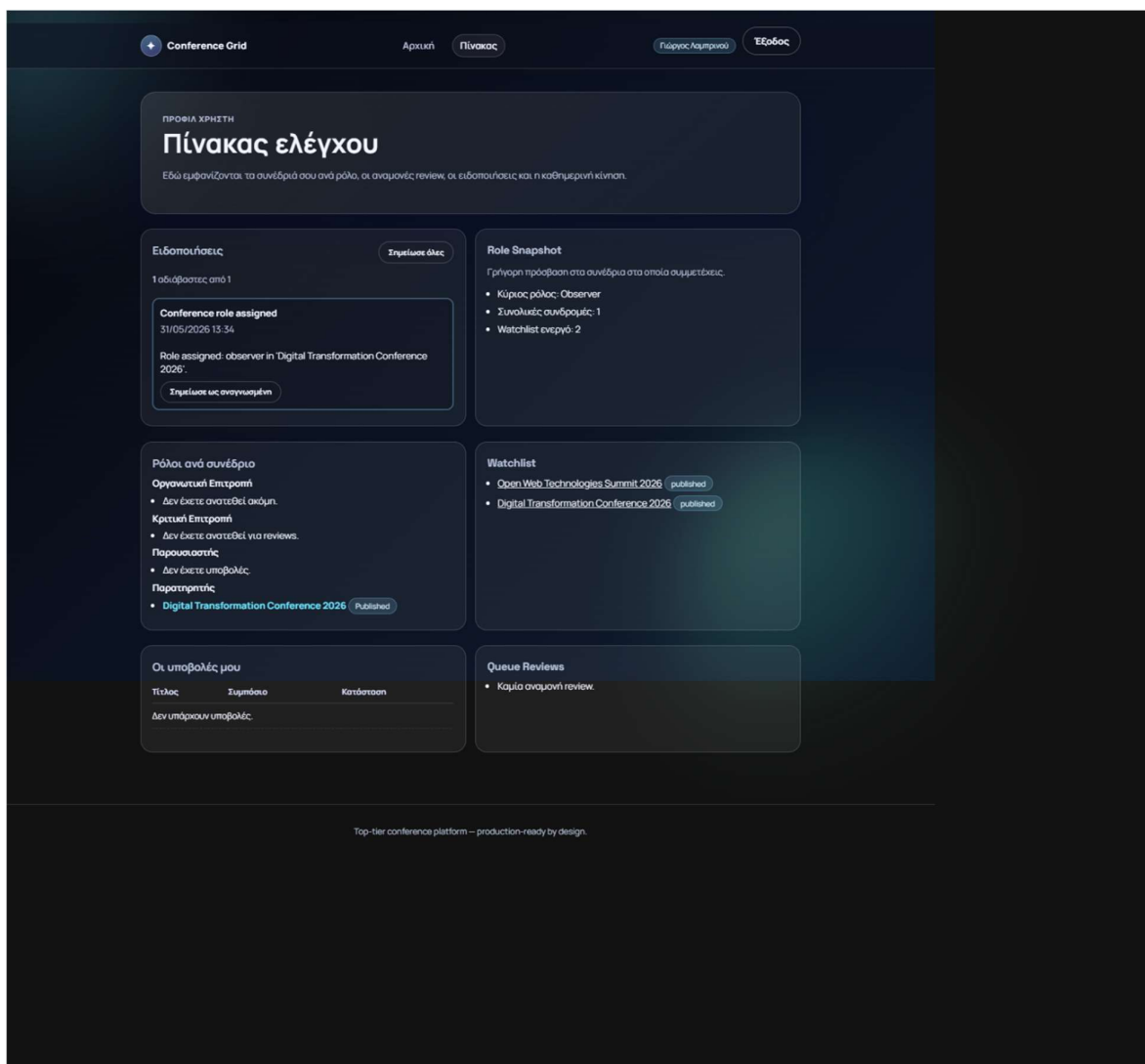
4.6.2 Πίνακας ελέγχου χρήστη

Ο πίνακας ελέγχου συγκεντρώνει τις προσωπικές πληροφορίες του χρήστη. Εμφανίζονται οι ειδοποιήσεις, οι ρόλοι που έχει αποκτήσει ανά συνέδριο, οι υποβολές του, η λίστα παρακολούθησης και τυχόν εκκρεμότητες αξιολόγησης. Στο συγκεκριμένο παράδειγμα ο χρήστης δεν έχει ακόμη κάποιον ειδικό ρόλο σε συνέδριο, γεγονός που δείχνει ότι η απλή εγγραφή δεν αποδίδει αυτόματα ρόλο.



Εικόνα 15 Πίνακας Ελέγχου Απλού Χρήστη

Μετά την εγγραφή του χρήστη ως observer, ο αντίστοιχος ρόλος εμφανίζεται στον πίνακα ελέγχου του, κάτω από το συνέδριο στο οποίο αφορά. Η λειτουργία αυτή διαφοροποιείται από τη watchlist, καθώς η watchlist αποτελεί απλή παρακολούθηση ενδιαφέροντος, ενώ η εγγραφή ως observer αποδίδει συγκεκριμένο ρόλο στον χρήστη για το επιλεγμένο συνέδριο.



Εικόνα 16 Πίνακας Ελέγχου μετά την Εγγραφή ως Observer

Στον πίνακα ελέγχου του presenter εμφανίζονται οι εργασίες που έχει υποβάλει, το συνέδριο στο οποίο ανήκουν και η τρέχουσα κατάστασή τους. Με αυτόν τον τρόπο ο χρήστης μπορεί να παρακολουθεί την πορεία κάθε υποβολής, από την αρχική καταχώριση μέχρι την αξιολόγηση, την αποδοχή, την ανάθεση σε track και την τελική ετοιμότητα για παρουσίαση.

Conference Grid

Αρχική Πίνακας

Αιτήσεις Παροχής Έξοδος

PUBLISHED

Digital Transformation Conference 2026

Το Digital Transformation Conference 2026 αποτελεί ένα ακαδημαϊκό και επαγγελματικό συνέδριο με αντικείμενο τον ψηφιακό μετασχηματισμό οργανισμών, εκπαιδευτικών ιδρυμάτων και επιχειρήσεων. Στόχος του συνεδρίου είναι η παρουσίαση ερευνητικών εργασιών, εφαρμογών και ελλών προκτιών που αετίζονται με πληροφορικά συστήματα, τεκνιτή νοημοσύνη, web εφαρμογές, ψηφιακές δεξιότητες και τεχνολογίες υποστήριξης αποφάσεων.

Accepted 1 Needs Third Review 1 Presentation Ready 1 Rejected 1 Submitted 1 Under Review 1

Προσθήκη στην Watchlist 20/07/2026 12:00 22/07/2026 12:00 Κόστος: 50.00 EUR 3 Tracks Ανοικτός υποβολές

Tracks

Artificial Intelligence and Data Analytics

Εφαρμογές τεκνιτής νοημοσύνης, μηχανικής μάθησης και ανάλυσης δεδομένων για οργανισμούς, επιχειρήσεις και εκπαιδευτικά περιβάλλοντα.

Capacity 2

Digital Skills and Educational Technologies

Ψηφιακές δεξιότητες, ηλεκτρονική μάθηση, εκπαιδευτικές πλατφόρμες, αξιολόγηση και αναφοράς προόδου.

Capacity 2

Web Applications and Cloud Systems

Σχεδιασμός και ανάπτυξη web εφαρμογών, cloud υπηρεσιών, βέλους δεδομένων και ασφαλών πληροφοριακών συστημάτων.

Capacity 2

Ος συμμετέων βλέπεις μόνο τις δικές σου υποβολές.

Actions

Νέα υποβολή paper

Τίτλος

Τίτλος paper

Abstract

Περίληψη έως 2-3 παραγράφοις.

Λέξεις-κλειδιά

machine learning, ai, vision

PDF (προαιρετικό)

Choose File

No file chosen

Καταχώρηση paper

Εγγραφή ως παρατηρητής

Εγγραφού ως observer για να παρακολουθείς το συνέδριο.

Εγγραφή Observer

Πρόγραμμα παρουσιάσεων

Δηλώσει ορατές παρουσιάσεις (papers που έχουν οριστικοποιηθεί σε track).

Paper	Track	Παρουσιαστής
Web Applications with Django	Web Applications and Cloud Systems	Ιωάννα Νικολάου

Οι υποβολές μου σε αυτό το συνέδριο

Paper	Κατάσταση	Track	Ενέργει
Adaptive Learning and Workshops /Λήψη PDF	Needs Third Review	—	—
Digital Skills Assessment Platform /Λήψη PDF	Accepted	—	—
AI Tools in Higher Education /Λήψη PDF	Submitted	—	—

Όλες οι υποβολές που εμφανίζονται

Paper	Κατάσταση	Παρουσιαστής	T
Adaptive Learning and Workshops /Λήψη PDF	Needs Third Review	Αιρίδος Παπαδόκης	—
Digital Skills Assessment Platform /Λήψη PDF	Accepted	Αιρίδος Παπαδόκης	—
AI Tools in Higher Education /Λήψη PDF	Submitted	Αιρίδος Παπαδόκης	—

Top-tier conference platform – production-ready by design.

Εικόνα 17 Πίνακας Ελέγχου Presenter με Υποβολές

Διπλωματική Εργασία

71

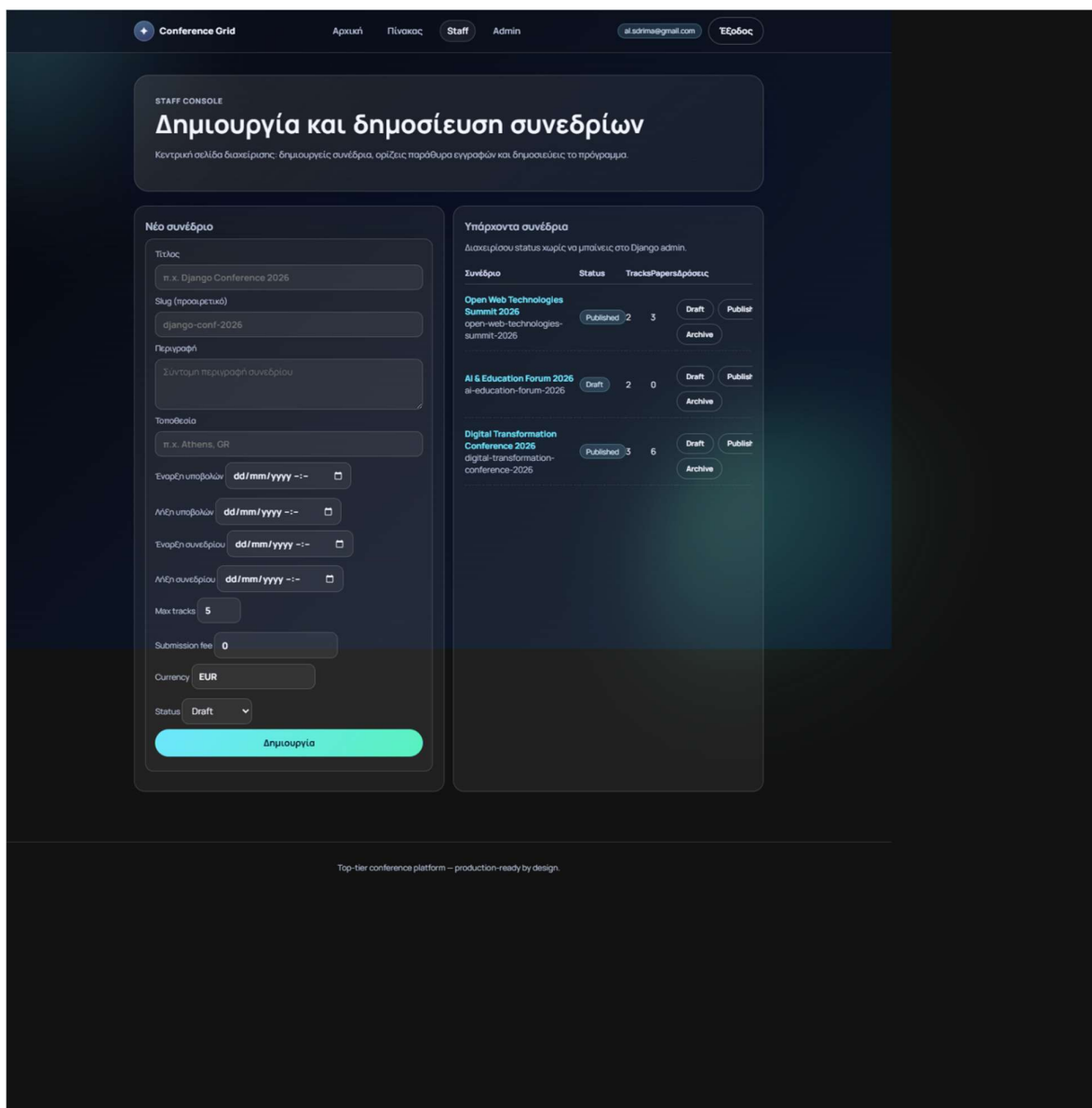
4.6.3 Υποβολή και παρακολούθηση εργασίας

Η φόρμα υποβολής εργασίας είναι διαθέσιμη μόνο σε συνδεδεμένους χρήστες και μόνο όταν οι υποβολές του συνεδρίου είναι ανοιχτές. Ο χρήστης συμπληρώνει τον τίτλο, την περίληψη, τις λέξεις-κλειδιά και προαιρετικά επισυνάπτει αρχείο PDF. Με την ολοκλήρωση της υποβολής, ο χρήστης αποκτά τον ρόλο του presenter για το συγκεκριμένο συνέδριο.

Εικόνα 18 Φόρμα Νέας Υποβολής Paper

4.6.4 Λειτουργίες Staff και Organizer

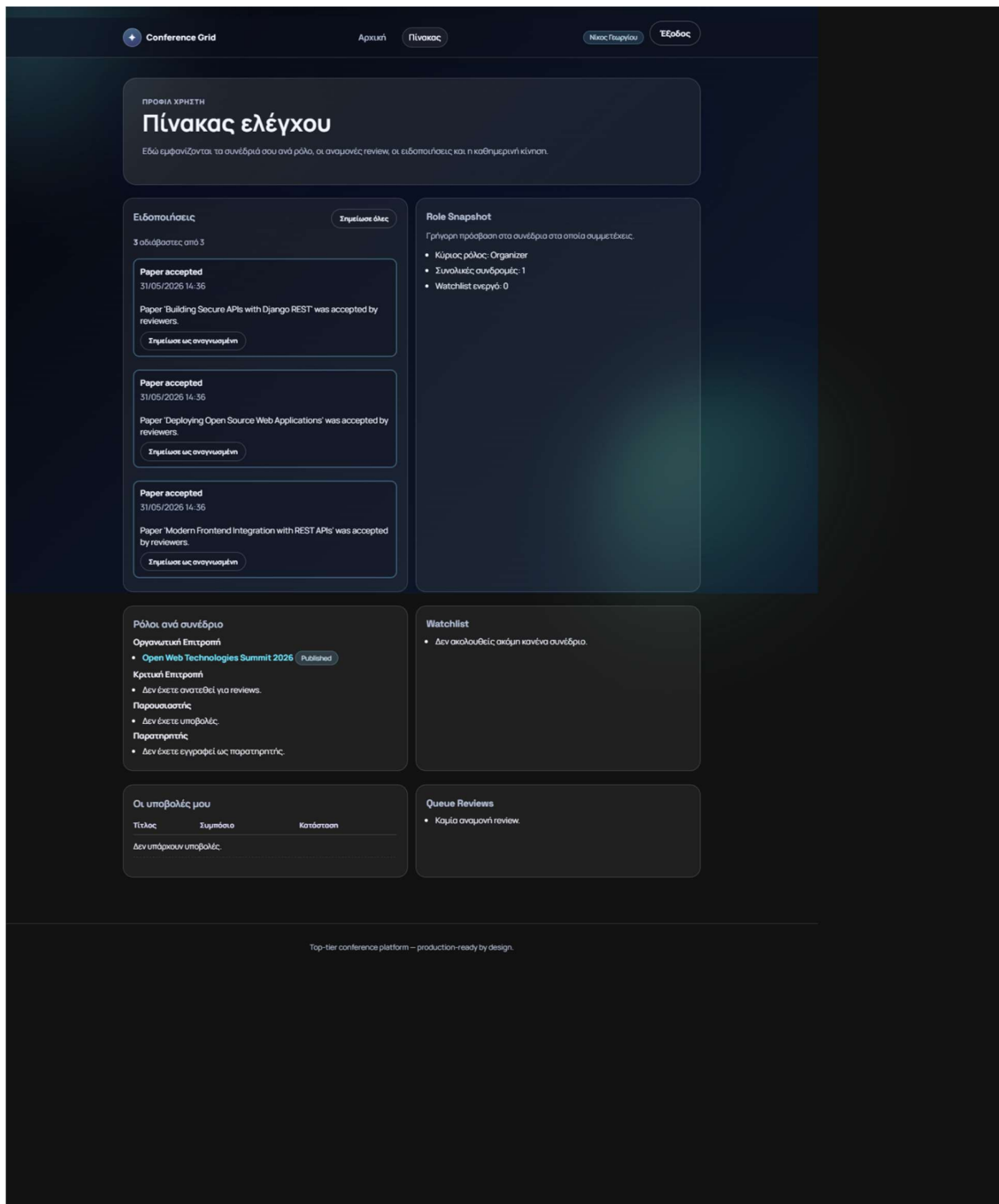
Ο staff χρήστης έχει πρόσβαση στη σελίδα διαχείρισης συνεδρίων, από όπου μπορεί να δημιουργήσει νέο συνέδριο. Κατά τη δημιουργία ορίζονται τα βασικά στοιχεία του συνεδρίου, οι ημερομηνίες υποβολών και διεξαγωγής, ο μέγιστος αριθμός tracks, το ποσό συμμετοχής, το νόμισμα και η κατάσταση δημοσίευσης. Αν το συνέδριο αποθηκευτεί ως draft, δεν εμφανίζεται στη δημόσια αρχική σελίδα.



Εικόνα 19 Staff Panel: Δημιουργία Συνεδρίου

Στο ίδιο περιβάλλον ο staff χρήστης μπορεί να δει τα συνέδρια που έχουν καταχωρηθεί στο σύστημα, την κατάστασή τους, τον αριθμό των tracks και τον αριθμό των υποβολών. Παρέχεται επίσης η δυνατότητα αλλαγής της κατάστασης ενός συνεδρίου, ώστε αυτό να παραμείνει draft, να δημοσιευθεί ή να αρχειοθετηθεί.

Ο organizer έχει πρόσβαση σε πρόσθετες λειτουργίες στη σελίδα του συνεδρίου. Μπορεί να αναθέσει ρόλους σε χρήστες με βάση το email τους και να ορίσει τα tracks του συνεδρίου, σύμφωνα με το μέγιστο πλήθος που έχει δηλωθεί κατά τη δημιουργία του. Οι ρόλοι που αποδίδονται ισχύουν μόνο για το συγκεκριμένο συνέδριο.



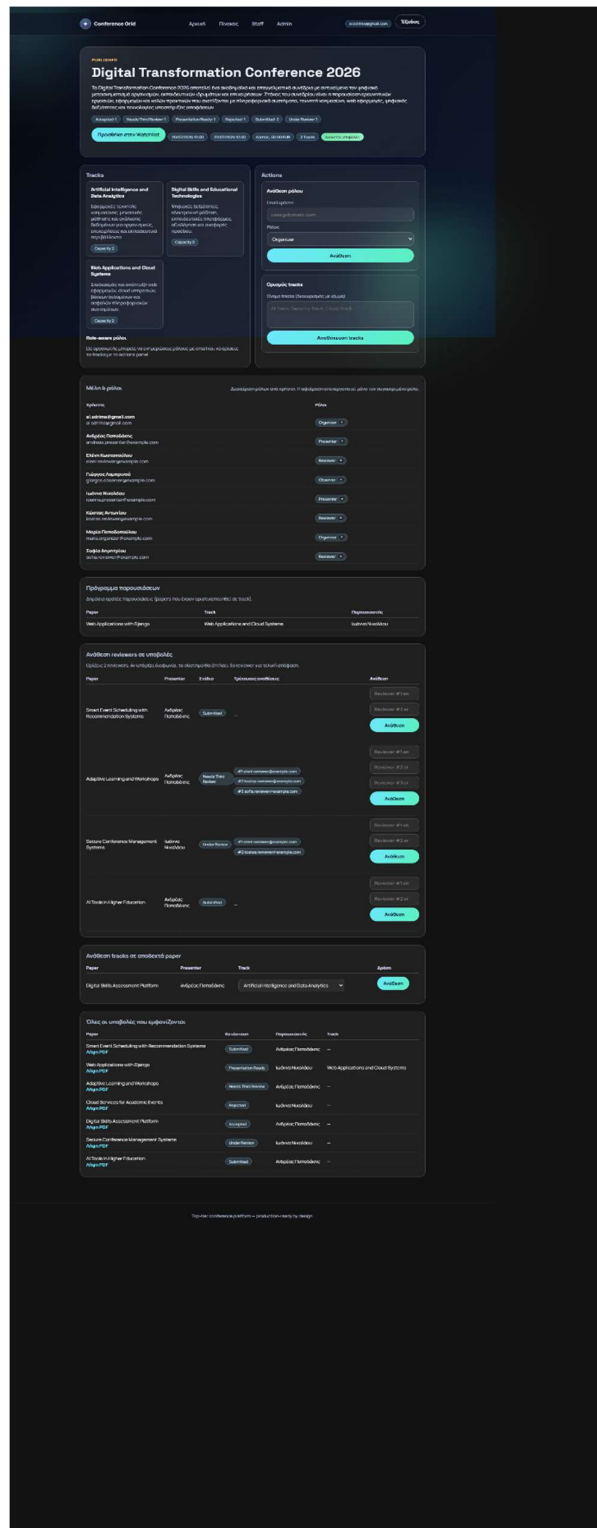
Εικόνα 20 Πίνακας Ελέγχου Organizer

Ο organizer μπορεί να παρακολουθεί τα μέλη του συνεδρίου και τους ρόλους τους, καθώς και να αναθέτει reviewers σε υποβολές. Για κάθε paper μπορούν να οριστούν δύο reviewers, ενώ σε περίπτωση διαφωνίας των αξιολογήσεων ενεργοποιείται η δυνατότητα ανάθεσης τρίτου reviewer. Το σύστημα δεν επιτρέπει την ανάθεση του ίδιου reviewer δύο φορές στο ίδιο paper.

Με βάση τα αποτελέσματα των αξιολογήσεων, το σύστημα ενημερώνει την κατάσταση κάθε υποβολής. Όταν και οι δύο reviewers προτείνουν αποδοχή, η εργασία μεταβαίνει σε κατάσταση accepted. Αν και οι δύο προτείνουν απόρριψη, η εργασία γίνεται rejected. Σε περίπτωση διαφωνίας, η εργασία μεταβαίνει σε κατάσταση needs third reviewer, ώστε ο organizer να αναθέσει τρίτο αξιολογητή.

Όταν οι δύο αρχικοί reviewers δεν συμφωνούν ως προς την τελική πρόταση αξιολόγησης, το σύστημα επισημαίνει την υποβολή ως needs third reviewer. Στην περίπτωση αυτή ο organizer μπορεί να ορίσει τρίτο reviewer, ο οποίος θα πραγματοποιήσει επιπλέον αξιολόγηση για την υποβοήθηση της τελικής απόφασης.

Μετά την αποδοχή μιας εργασίας, ο organizer μπορεί να την εντάξει σε ένα από τα διαθέσιμα tracks του συνεδρίου, εφόσον υπάρχει διαθέσιμη χωρητικότητα. Με την ανάθεση σε track, η κατάσταση της εργασίας μεταβάλλεται σε track assigned και στη συνέχεια ενεργοποιείται η δυνατότητα πληρωμής από τον presenter.

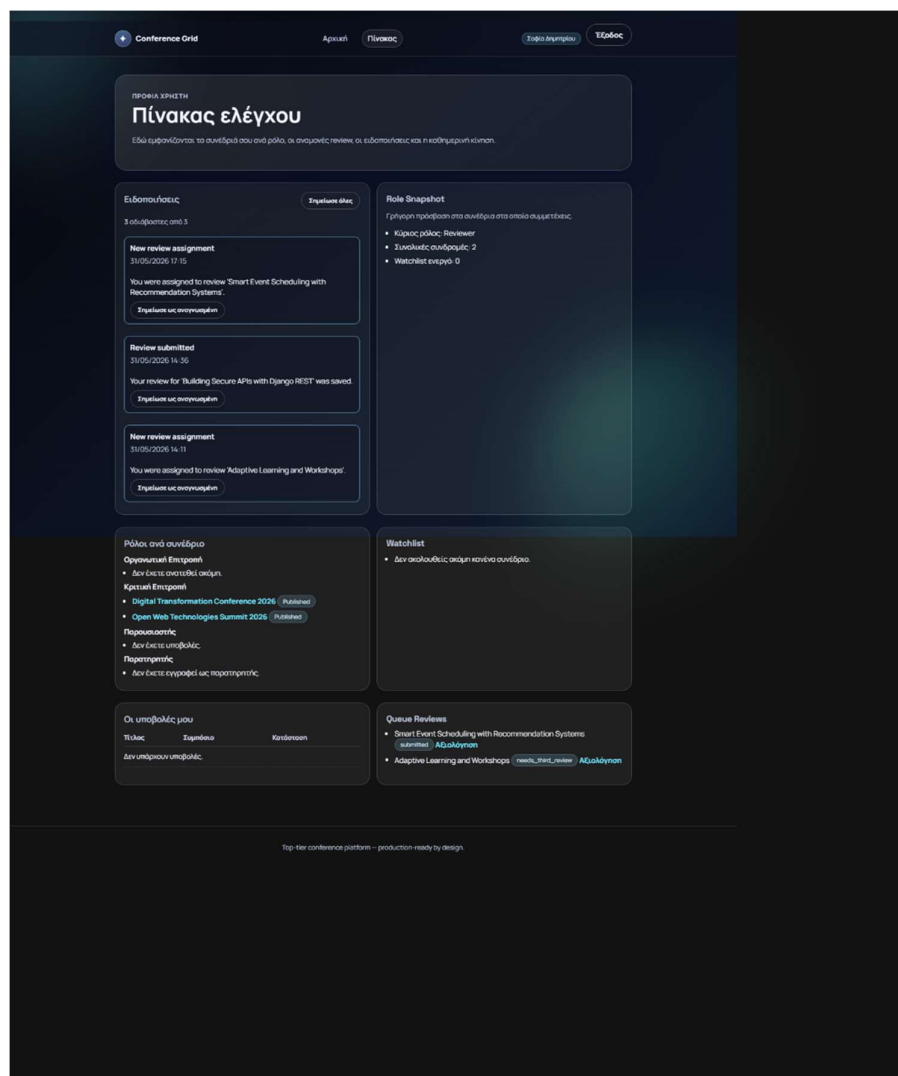


Εικόνα 21 — Organizer View: Ρόλοι και Tracks, Μέλη και Ανάθεση Reviewers, Αποτελέσματα Reviews και Καταστάσεις Papers, Ανάθεση Τρίτου Reviewer, Ανάθεση Accepted Paper σε Track

4.6.5 Διαδικασία αξιολόγησης από Reviewer

Όταν ένας χρήστης οριστεί ως reviewer σε κάποιο συνέδριο και του ανατεθεί εργασία προς αξιολόγηση, η σχετική πληροφορία εμφανίζεται στον πίνακα ελέγχου του. Το review queue περιλαμβάνει τις εκκρεμείς αξιολογήσεις και παρέχει σύνδεσμο προς τη σελίδα του συνεδρίου, ώστε ο reviewer να πραγματοποιήσει την αξιολόγηση.

Στη φόρμα αξιολόγησης ο reviewer έχει πρόσβαση στα στοιχεία της υποβολής και μπορεί να κατεβάσει το αρχείο PDF που του έχει ανατεθεί. Στη συνέχεια καταχωρεί την πρότασή του, τη βαθμολογία και τα σχόλιά του. Μετά την υποβολή της αξιολόγησης, η εκκρεμότητα αφαιρείται από το review queue του reviewer.



Εικόνα 22 — Reviewer Dashboard με Review Queue

PUBLISHED

Digital Transformation Conference 2026

To Digital Transformation Conference 2026 αποτελεί ένα πολυμέσικο και επηρεαζόμενο από το περιβάλλον των ψηφιακών μετασχηματισμών οργανισμό, εκπαιδευτικό ιδρυματικό και επαγγελματικό. Στόχος του συνεδρίου είναι η παρουσίαση καινοτομικών εργαλείων, εφαρμογών και νέων προκλήσεων που σχετίζονται με πληροφοριακά συστήματα, τεχνητή νοημοσύνη, web εφαρμογές, ψηφιακές μετασχηματιστικές τεχνολογίες, υποδομές.

Accepted 1 Needs Third Review 1 Presentation Ready 1 Reported 1 Submitted 2 Under Review 1

Προσέβαση στην Watchlist 20/07/2026 12:00 20/07/2026 12:00 Κλειστό 06/09/2026 3 Tracks **Κεντρικός υπολογιστής**

Tracks

Artificial Intelligence and Data Analytics

Εφαρμογές τεχνητής νοημοσύνης, μηχανικής μάθησης και ανάλυσης δεδομένων για οργανισμούς, επιχειρήσεις και εκπαιδευτικά περιβάλλοντα.

Capacity 1

Digital Skills and Educational Technologies

Ψηφιακές δεξιότητες, ηλεκτρονικά μάθημα, εκπαιδευτικές πλατφόρμες, εκπαίδευση και αναπτυξιακά προγράμματα.

Capacity 2

Web Applications and Cloud Systems

Συνολικό και αναπτυξιακό web εφαρμογών, cloud υπηρεσιών, βάσεων δεδομένων και υπηρεσιών πληροφοριακών συστημάτων.

Capacity 3

Οι κριτικές (Reviews) αφορούν το review για αυτό το συνέδριο.

Actions

Νέα υποβληθέντα paper

Τίτλος

Τίτλος paper

Abstract

Παράτημα έως 2-3 παραγράφους

Αξιολογία

machine learning, AI, vision

PDF (προαιρετικό)

Choose File No file chosen

Καταχώριση paper

Εγγραφή ως παρατηρητής

Εγγραφή ως observer για να παρακολουθείς το συνέδριο.

Εγγραφή Observer

Πρόγραμμα παρουσιάσεων

Λίστα με τις παρουσιάσεις (papers που έχουν αποσταλεί) σε tracks.

Paper	Track	Παρουσιαστής
Web Applications with Django	Web Applications and Cloud Systems	Καθώς Νικολάου

Μεταβολές στο περιεχόμενο

Enterprise Review Queue

Paper

Παρουσιαστής 1 τάξη Αξιολογητής

Apriori Πλαίσιο

Ταξινόμηση Υπόδειξη

Conference Grid

Smart Event Scheduling with Recommendation Systems
Align PDF

Αυτοματ. Πλαίσιο

Submitted

Προσέβαση ανάθεση

Υποβληθεί

Adaptive Learning and Workshops
Align PDF

Αυτοματ. Πλαίσιο

Needs Third Review

Προσέβαση ανάθεση

Υποβληθεί

Τίτλος

Λίστα οι υποβληθείς που εμφανίζονται

Paper	Κατάσταση	Παρουσιαστής	Tracks
Smart Event Scheduling with Recommendation Systems Align PDF	Submitted	Αυτοματ. Πλαίσιο	-
Web Applications with Django Align PDF	Presentation Ready	Καθώς Νικολάου	Web Applications and Cloud Systems
Adaptive Learning and Workshops Align PDF	Needs Third Review	Αυτοματ. Πλαίσιο	-
Cloud Services for Academic Events Align PDF	Reported	Καθώς Νικολάου	-
Digital Skills Assessment Platform Align PDF	Accepted	Αυτοματ. Πλαίσιο	-
Secure Conference Management Systems Align PDF	Under Review	Καθώς Νικολάου	-
AI Tools in Higher Education Align PDF	Submitted	Αυτοματ. Πλαίσιο	-

Top tier conference platform - production ready by design

Εικόνα 23 — Φόρμα Αξιολόγησης Paper

4.6.6 Πληρωμή και Κατάσταση Presentation Ready

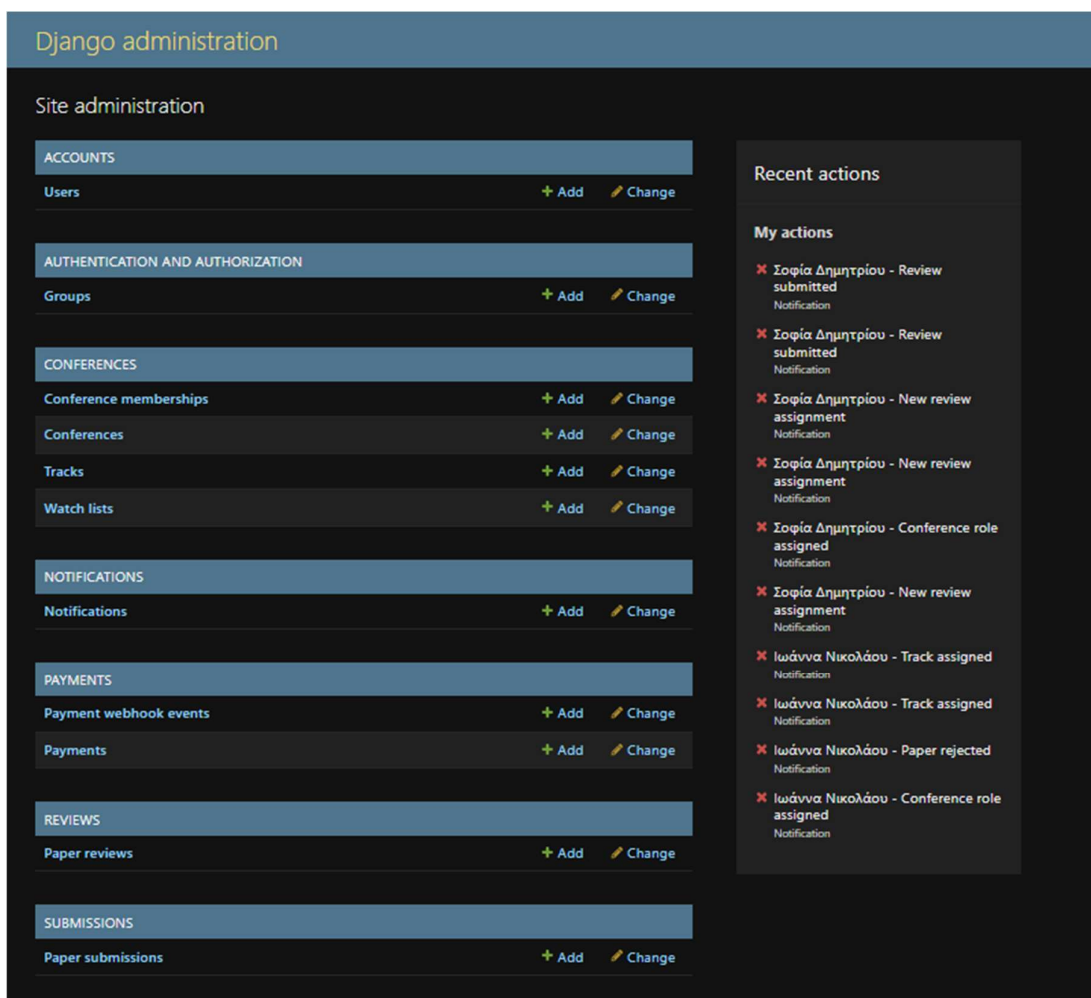
Η δυνατότητα πληρωμής εμφανίζεται μόνο όταν μια αποδεκτή εργασία έχει ανατεθεί σε track. Μετά την επιτυχή ολοκλήρωση της πληρωμής, η κατάσταση της υποβολής μεταβάλλεται σε presentationready και η εργασία μπορεί πλέον να εμφανιστεί στο πρόγραμμα παρουσιάσεων του συνεδρίου. Αν η πληρωμή δεν ολοκληρωθεί, η υποβολή παραμένει σε κατάσταση paymentpending και ο χρήστης μπορεί να επαναλάβει τη διαδικασία.

The screenshot shows a payment interface for a submission fee. On the left, there is a header with a back arrow, a document icon, and the text "New business sandbox" next to a "Sandbox" button. Below this, the text "Submission fee: Web Applications with Django" is displayed, followed by the amount "€50.00". On the right, there are two prominent buttons: "Pay with link" (green) and "Pay with Klarna" (black). Below these is a horizontal line with "OR" in the center. Under "Contact information", there is a text input field for "Email" containing "ioanna.presenter@example.com". The "Payment method" section lists five options: "Card" (with Visa, Mastercard, and American Express logos), "Klarna", "MB WAY", "Bancontact", and "EPS". Below the payment methods is a checkbox labeled "Save my information for faster checkout" with the text "Pay securely at New business sandbox and everywhere link is accepted." at the bottom. A large blue "Pay" button is at the bottom right. At the very bottom, it says "Powered by stripe" followed by "Terms" and "Privacy" links.

Εικόνα 24 — Πληρωμή και Κατάσταση Presentation Ready

4.6.7 Διαχειριστικό περιβάλλον

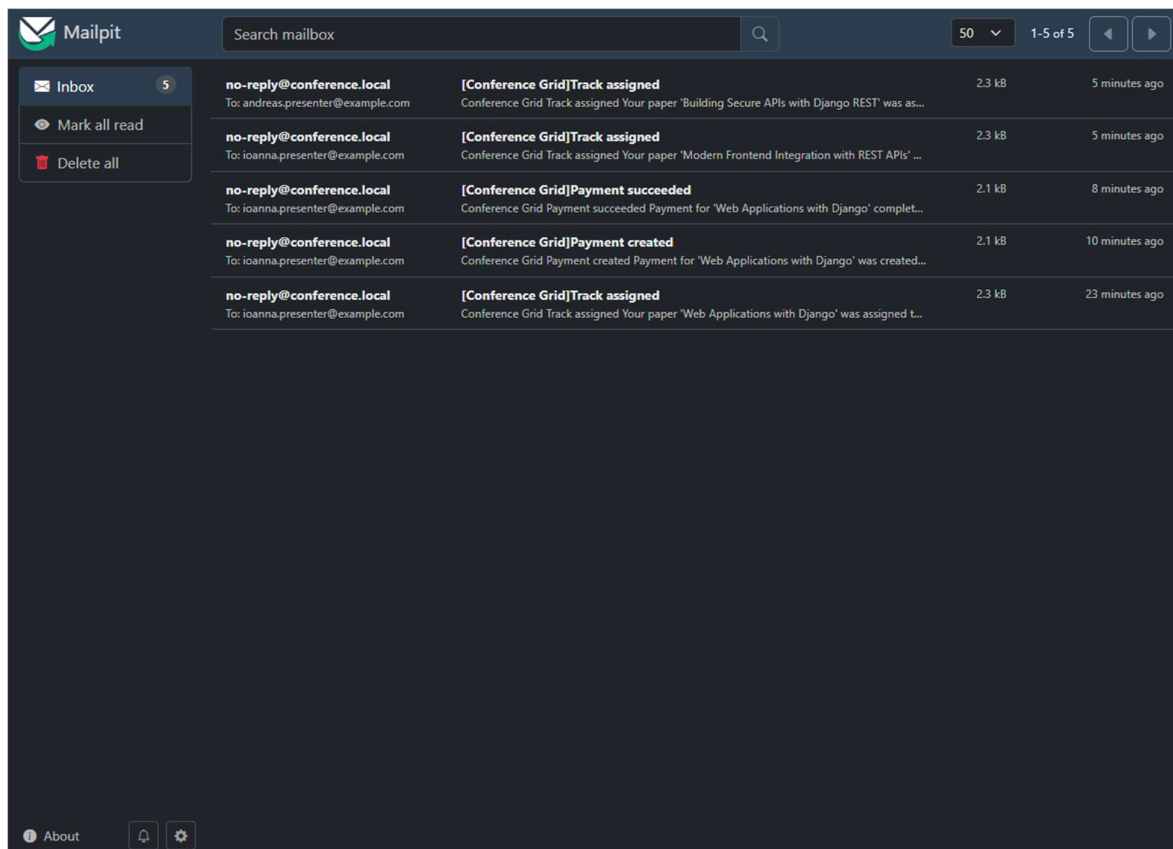
Εκτός από τις σελίδες της εφαρμογής, ο staff χρήστης έχει πρόσβαση και στο διαχειριστικό περιβάλλον του Django. Μέσω αυτού μπορεί να ελέγχει και να διαχειρίζεται απευθείας τις βασικές οντότητες της εφαρμογής, όπως χρήστες, συνέδρια, tracks, υποβολές, αξιολογήσεις, ειδοποιήσεις και πληρωμές.



Εικόνα 25 — Διαχειριστικό Περιβάλλον

4.6.8 Περιβάλλον Mailpit

Οι ειδοποιήσεις για κάθε εναλλαγή κατάστασης αποστέλλονται και με mail μέσω Mailpit.



Εικόνα 26 — Περιβάλλον Mailpit

5. Αξιολόγηση της Εφαρμογής

Η αξιολόγηση αποτελεί σημαντικό στάδιο της ανάπτυξης ενός πληροφοριακού συστήματος, καθώς επιτρέπει να εξεταστεί όχι μόνο αν οι επιμέρους λειτουργίες εκτελούνται τεχνικά με ορθό τρόπο, αλλά και αν οι χρήστες μπορούν να τις εντοπίσουν, να τις κατανοήσουν και να τις χρησιμοποιήσουν αποτελεσματικά. Στην περίπτωση της παρούσας εφαρμογής, η αξιολόγηση αποκτά ιδιαίτερη σημασία λόγω της ύπαρξης διαφορετικών ρόλων, δικαιωμάτων και ροών εργασίας ανά συνέδριο.

Ο ρόλος ενός χρήστη δεν αποδίδεται κατά την απλή εγγραφή του στην εφαρμογή, αλλά διαμορφώνεται ανάλογα με τις ενέργειες που πραγματοποιεί και τη συμμετοχή του σε κάθε συνέδριο. Ένας συνδεδεμένος χρήστης λειτουργεί αρχικά ως guest, μπορεί να αποκτήσει τον ρόλο observer μετά την εγγραφή του ως παρατηρητής, presenter μετά την υποβολή εργασίας, reviewer μετά από σχετική ανάθεση, organizer όταν δημιουργήσει ή οργανώνει ένα συνέδριο και staff ή superuser όταν διαθέτει τα αντίστοιχα δικαιώματα διαχείρισης. Κατά συνέπεια, η αξιολόγηση πρέπει να εξετάσει τόσο τη γενική ευχρηστία της εφαρμογής όσο και την ορθότητα της μετάβασης μεταξύ ρόλων και της εφαρμογής των αντίστοιχων περιορισμών πρόσβασης.

Λόγω αυτής της διαφοροποίησης, η αξιολόγηση οργανώθηκε ανά ρόλο χρήστη. Κάθε συμμετέχων εκτέλεσε τα κοινά σενάρια και μόνο τα ειδικά σενάρια που αντιστοιχούσαν στους ρόλους που του ανατέθηκαν κατά τη δοκιμή. Με αντίστοιχο τρόπο οργανώθηκε και το ερωτηματολόγιο. Όλοι οι συμμετέχοντες συμπλήρωσαν μία κοινή ενότητα, ενώ οι ειδικές ενότητες συμπληρώθηκαν μόνο από όσους είχαν χρησιμοποιήσει τον αντίστοιχο ρόλο.

Το κεφάλαιο περιγράφει τον σκοπό και τη μεθοδολογία της αξιολόγησης, τα σενάρια δοκιμών, τη δομή του ερωτηματολογίου και τον τρόπο ανάλυσης της ποσοτικής και ποιοτικής ανατροφοδότησης.

5.1 Σκοπός και Ερευνητικά Ερωτήματα

Κύριος σκοπός της αξιολόγησης ήταν να εξεταστεί κατά πόσο η εφαρμογή υποστηρίζει με κατανοητό, συνεπή και αποτελεσματικό τρόπο τον συνολικό κύκλο λειτουργίας ενός συνεδρίου. Ο κύκλος αυτός περιλαμβάνει τον εντοπισμό των δημοσιευμένων συνεδρίων, την εγγραφή και τη σύνδεση των χρηστών, την παρακολούθηση συνεδρίων, την υποβολή

επιστημονικών εργασιών, την ανάθεση ρόλων, τη διαδικασία αξιολόγησης των εργασιών και τη διαχείριση των συνεδρίων.

Η αξιολόγηση επικεντρώνεται στα ακόλουθα ερευνητικά ερωτήματα:

1. Σε ποιο βαθμό η εφαρμογή διαχείρισης συνεδρίων είναι εύχρηστη για τους διαφορετικούς ρόλους χρηστών;
2. Σε ποιο βαθμό η εφαρμογή υποστηρίζει αποτελεσματικά τις λειτουργίες και τις εργασίες που αντιστοιχούν σε κάθε ρόλο χρήστη;
3. Πόσο σαφείς και κατανοητές είναι οι πληροφορίες, οι ειδοποιήσεις και οι καταστάσεις που εμφανίζονται στην εφαρμογή;
4. Παρουσιάζει η εφαρμογή συνέπεια και ορθή λειτουργία κατά την εκτέλεση των διαφορετικών σεναρίων χρήσης;
5. Σε ποιο βαθμό οι χρήστες θεωρούν ότι η εφαρμογή προστατεύει τα προσωπικά δεδομένα, τα papers και τις λειτουργίες που περιορίζονται ανά ρόλο;
6. Ποιες δυσκολίες και ανάγκες βελτίωσης εντοπίζουν οι χρήστες κατά την αλληλεπίδρασή τους με την εφαρμογή;
7. Σε ποιο βαθμό η εφαρμογή θεωρείται κατάλληλη για τη διαχείριση ενός πραγματικού συνεδρίου;
8. Διαφέρει η αξιολόγηση της ευχρηστίας και της λειτουργικότητας ανάλογα με τον ρόλο και την πολυπλοκότητα των εργασιών του χρήστη;

5.2 Μεθοδολογία Αξιολόγησης

Για την αξιολόγηση της εφαρμογής χρησιμοποιείται μικτή μεθοδολογική προσέγγιση, η οποία συνδυάζει ποσοτικά και ποιοτικά δεδομένα. Η ποσοτική αξιολόγηση βασίζεται στην καταγραφή των απαντήσεων σε ερωτηματολόγιο κλίμακας Likert. Η ποιοτική αξιολόγηση βασίζεται στην παρατήρηση της συμπεριφοράς των χρηστών και στην καταγραφή σχολίων, δυσκολιών και προτάσεων βελτίωσης.

Η διαδικασία περιλαμβάνει τρία στάδια:

1. επιλογή συμμετεχόντων
2. εκτέλεση προκαθορισμένων σεναρίων, ανά ρόλο
3. συμπλήρωση ερωτηματολογίου και συλλογή σχολίων μέσω ανοικτών ερωτήσεων.

5.2.1 Επιλογή Συμμετεχόντων

Στην αξιολόγηση συμμετείχαν επτά άτομα με διαφορετικό επίπεδο εμπειρίας στη χρήση διαδικτυακών εφαρμογών. Η επιλογή τους έγινε με σκοπό να καλυφθούν οι βασικοί ρόλοι και τα σημαντικότερα σενάρια χρήσης της εφαρμογής διαχείρισης συνεδρίων.

Δεν εκτέλεσαν όλοι οι συμμετέχοντες τις ίδιες εργασίες. Κάθε συμμετέχων ακολούθησε συγκεκριμένο σενάριο χρήσης, ανάλογα με τον ρόλο που του ανατέθηκε. Με αυτόν τον τρόπο εξετάστηκαν τόσο οι απλές λειτουργίες περιήγησης όσο και πιο σύνθετες διαδικασίες, όπως η υποβολή και αξιολόγηση paper, η ανάθεση σε track, η πληρωμή και η διαχείριση ενός συνεδρίου.

Ο Σ1 αξιολόγησε την εφαρμογή ως μη συνδεδεμένος επισκέπτης, ενώ οι συμμετέχοντες Σ2 έως Σ7 χρησιμοποίησαν συνδεδεμένους δοκιμαστικούς λογαριασμούς και αξιολόγησαν τον προσωπικό πίνακα ελέγχου και τις λειτουργίες των ρόλων που τους είχαν ανατεθεί, όπως παρουσιάζεται και στον παρακάτω πίνακα.

Συμμετέχων	Εμπειρία σε διαδικτυακές εφαρμογές	Κατάσταση σύνδεσης	Ρόλοι που αξιολογήθηκαν
Σ1	Μέτρια	Μη συνδεδεμένος χρήστης	Μη συνδεδεμένος επισκέπτης
Σ2	Μέτρια	Συνδεδεμένος χρήστης	Guest, Observer
Σ3	Μεγάλη	Συνδεδεμένος χρήστης	Presenter
Σ4	Μεγάλη	Συνδεδεμένος χρήστης	Presenter
Σ5	Μεγάλη	Συνδεδεμένος χρήστης	Reviewer
Σ6	Μεγάλη	Συνδεδεμένος χρήστης	Organizer
Σ7	Πολύ μεγάλη	Συνδεδεμένος χρήστης	Organizer, Staff/Superuser

Πίνακας 13 Προφίλ συμμετεχόντων και ρόλοι που αξιολογήθηκαν κατά τη δοκιμή της εφαρμογής

5.2.2 Περιβάλλον Δοκιμών

Οι συμμετέχοντες χρησιμοποίησαν την εφαρμογή για την εκτέλεση διαφορετικών δραστηριοτήτων σε ελεγχόμενο περιβάλλον δοκιμών. Η εφαρμογή είχε εγκατασταθεί τοπικά και ήταν πλήρως λειτουργική κατά τη διάρκεια της αξιολόγησης. Το περιβάλλον αυτό επέτρεπε την παρακολούθηση της χρήσης της εφαρμογής και την καταγραφή τυχόν δυσκολιών ή προβλημάτων που προέκυπταν κατά την εκτέλεση των σεναρίων.

5.2.3 Συλλογή Δεδομένων

Μετά την ολοκλήρωση των σεναρίων, οι συμμετέχοντες συμπλήρωσαν ψηφιακό ερωτηματολόγιο μέσω του εξής συνδέσμου (<https://forms.gle/vLNHxjNr8bPMB4Tw9>). Οι ερωτήσεις κλειστού τύπου βαθμολογήθηκαν σε πενταβάθμια κλίμακα Likert:

- 1: διαφωνώ απόλυτα,
- 2: διαφωνώ,
- 3: ούτε συμφωνώ ούτε διαφωνώ,
- 4: συμφωνώ,
- 5: συμφωνώ απόλυτα.

Στο τέλος του ερωτηματολογίου υπήρχαν ερωτήσεις ανοικτού τύπου για την καταγραφή προβλημάτων και προτάσεων.

5.3 Σενάρια Δοκιμών ανά Ρόλο

5.3.1 Μη Συνδεδεμένος Χρήστης

Το σενάριο του μη συνδεδεμένου χρήστη εκτελέστηκε από τον συμμετέχοντα Σ1.

Ο συμμετέχων κλήθηκε να:

1. εντοπίσει τα δημοσιευμένα συνέδρια στην αρχική σελίδα,
2. χρησιμοποιήσει το κουμπί «Ανακάλυψε περισσότερα»,
3. επιλέξει ένα δημοσιευμένο συνέδριο και να μεταβεί στη σελίδα πληροφοριών του,
4. εντοπίσει τις ημερομηνίες υποβολών και διεξαγωγής του συνεδρίου,

5. εντοπίζει το κόστος συμμετοχής και τον αριθμό των tracks,
6. εξετάσει τη χωρητικότητα των διαθέσιμων tracks,
7. εντοπίζει τα υποβληθέντα papers και τις καταστάσεις τους,
8. εξετάσει το πρόγραμμα παρουσιάσεων,
9. αναγνωρίζει ποιες πληροφορίες ήταν δημόσια διαθέσιμες,
10. αναγνωρίζει ποιες ενέργειες απαιτούσαν σύνδεση στην εφαρμογή.

5.3.2 Συνδεδεμένος Χρήστης και Κοινές Λειτουργίες

Οι κοινές λειτουργίες του συνδεδεμένου χρήστη εκτελέστηκαν από τους συμμετέχοντες Σ2 έως Σ7.

Οι συμμετέχοντες κλήθηκαν να:

1. συνδεθούν στην εφαρμογή με τον προκαθορισμένο δοκιμαστικό λογαριασμό τους,
2. επιβεβαιώσουν ότι εμφανιζόταν σωστά το ονοματεπώνυμό τους,
3. μεταβούν στον προσωπικό πίνακα ελέγχου,
4. εξετάσουν τις ειδοποιήσεις τους,
5. σημειώσουν μία ειδοποίηση ως αναγνωσμένη,
6. χρησιμοποιήσουν τη λειτουργία επισήμανσης όλων των ειδοποιήσεων ως αναγνωσμένων,
7. εντοπίσουν τους ρόλους που κατείχαν ανά συνέδριο,
8. εξετάσουν τις υποβολές ή τις εκκρεμείς εργασίες που αντιστοιχούσαν στον λογαριασμό τους,
9. χρησιμοποιήσουν τους συνδέσμους του πίνακα ελέγχου για να μεταβούν στο αντίστοιχο συνέδριο,
10. αποσυνδεθούν από την εφαρμογή.

5.3.3 Λίστα Παρακολούθησης και Ρόλος Παρατηρητή

Το σενάριο εκτελέστηκε από τον συμμετέχοντα Σ2.

Ο συμμετέχων κλήθηκε να:

1. επιλέξει ένα δημοσιευμένο συνέδριο,
2. το προσθέσει στη λίστα παρακολούθησης,

3. επιβεβαιώσει ότι το συνέδριο εμφανίστηκε στη watchlist του πίνακα ελέγχου,
4. αφαιρέσει το συνέδριο από τη λίστα παρακολούθησης,
5. το προσθέσει ξανά στη λίστα,
6. εγγραφεί στο συνέδριο ως παρατηρητής,
7. επιβεβαιώσει ότι η επιλογή εγγραφής ως παρατηρητής δεν ήταν πλέον διαθέσιμη,
8. ελέγξει ότι ο ρόλος observer εμφανίστηκε στον πίνακα ελέγχου,
9. διακρίνει τη διαφορά μεταξύ της προσθήκης στη watchlist και της εγγραφής ως observer,
10. εξετάσει τις πληροφορίες και το πρόγραμμα του συνεδρίου ως παρατηρητής.

5.3.4 Υποβολή και Παρακολούθηση Υποβολής

Το σενάριο του presenter εκτελέστηκε από τους συμμετέχοντες Σ3 και Σ4.

Οι συμμετέχοντες κλήθηκαν να:

1. επιλέξουν ένα συνέδριο με ανοικτές υποβολές,
2. εντοπίσουν τη δυνατότητα νέας υποβολής,
3. συμπληρώσουν τον τίτλο, την περίληψη και τις λέξεις-κλειδιά του paper,
4. μεταφορτώσουν προαιρετικά αρχείο PDF,
5. ολοκληρώσουν την υποβολή,
6. επιβεβαιώσουν ότι το paper εμφανίστηκε στις προσωπικές υποβολές τους,
7. επιβεβαιώσουν ότι το paper εμφανίστηκε στη σελίδα του συνεδρίου,
8. ελέγξουν ότι απέκτησαν τον ρόλο presenter για το συγκεκριμένο συνέδριο,
9. κατεβάσουν το αρχείο που είχαν υποβάλει,
10. παρακολουθήσουν την κατάσταση του paper από τον πίνακα ελέγχου και τη σελίδα του συνεδρίου,
11. εξετάσουν τις σχετικές ειδοποιήσεις.

Ο Σ3 ακολούθησε το σενάριο κατά το οποίο το paper απορρίφθηκε και κλήθηκε να επιβεβαιώσει την εμφάνιση της κατάστασης rejected και της σχετικής ειδοποίησης.

Ο Σ4 ακολούθησε το σενάριο αποδοχής και κλήθηκε επιπλέον να:

12. επιβεβαιώσει την εμφάνιση της κατάστασης accepted,
13. περιμένει την ανάθεση του paper σε track από τον organizer,

14. επιβεβαιώσει τη μετάβαση στην κατάσταση track assigned,
15. εντοπίσει τη δυνατότητα πληρωμής,
16. μεταβεί στο δοκιμαστικό περιβάλλον Stripe Checkout,
17. ολοκληρώσει τη δοκιμαστική πληρωμή,
18. επιβεβαιώσει ότι το paper μεταφέρθηκε στην κατάσταση presentation ready.

5.3.5 Αξιολόγηση Υποβολής από Αξιολογητή

Το σενάριο εκτελέστηκε από τον συμμετέχοντα Σ5.

Ο συμμετέχων κλήθηκε να:

1. εντοπίσει την ειδοποίηση ανάθεσης του ρόλου reviewer,
2. εντοπίσει την ειδοποίηση ανάθεσης paper προς αξιολόγηση,
3. μεταβεί στην ουρά εκκρεμών αξιολογήσεων του πίνακα ελέγχου,
4. χρησιμοποιήσει τον σύνδεσμο για να μεταβεί στη σελίδα του συνεδρίου,
5. εντοπίσει το paper που του είχε ανατεθεί,
6. κατεβάσει το αρχείο του paper,
7. εξετάσει τα διαθέσιμα στοιχεία της υποβολής,
8. επιλέξει το προτεινόμενο αποτέλεσμα της αξιολόγησης,
9. καταχωρίσει βαθμολογία,
10. προσθέσει σχόλια αξιολόγησης,
11. υποβάλει την αξιολόγηση,
12. επιβεβαιώσει ότι το paper αφαιρέθηκε από την ουρά εκκρεμών αξιολογήσεων,
13. επιβεβαιώσει ότι η κατάσταση του paper ενημερώθηκε μετά την υποβολή της αξιολόγησης.

5.3.6 Διαχείριση συνεδρίου από Οργανωτική Επιτροπή

Το κύριο σενάριο του organizer εκτελέστηκε από τον συμμετέχοντα Σ6. Ο Σ7 αξιολόγησε επίσης ορισμένες από τις λειτουργίες του organizer μετά τη δημιουργία συνεδρίου.

Οι συμμετέχοντες κλήθηκαν, ανάλογα με το σενάριό τους, να:εντοπίσουν τα υπάρχοντα μέλη και τους ρόλους τους,

1. αναθέσουν ρόλο σε χρήστη μέσω της διεύθυνσης ηλεκτρονικού ταχυδρομείου του,

2. επιβεβαιώσουν ότι ο νέος ρόλος εμφανίστηκε στη λίστα μελών,
3. απενεργοποιήσουν έναν ρόλο χρήστη,
4. δημιουργήσουν tracks και να ορίσουν τη χωρητικότητά τους,
5. εντοπίσουν papers που χρειάζονταν ανάθεση reviewers,
6. αναθέσουν πρώτο και δεύτερο reviewer σε paper,
7. επιχειρήσουν να επιλέξουν τον ίδιο χρήστη ως πρώτο και δεύτερο reviewer, ώστε να ελεγχθεί ο σχετικός περιορισμός,
8. παρακολουθήσουν την εξέλιξη των αξιολογήσεων,
9. εντοπίσουν paper για το οποίο οι δύο reviewers είχαν διαφορετική πρόταση,
10. αναθέσουν τρίτο reviewer στο συγκεκριμένο paper,
11. εντοπίσουν accepted paper που δεν είχε ακόμη ενταχθεί σε track,
12. τοποθετήσουν το paper σε κατάλληλο track,
13. επιβεβαιώσουν ότι η κατάσταση του paper άλλαξε σε track assigned,
14. ελέγξουν ότι το paper εμφανίστηκε στο αντίστοιχο track και στο πρόγραμμα του συνεδρίου.

5.3.7 Δημιουργία και Διαχείριση Συνεδρίου από Διαχειριστή/Superuser

Το σενάριο εκτελέστηκε από τον συμμετέχοντα Σ7.

Ο συμμετέχων κλήθηκε να:

1. μεταβεί στην καρτέλα Staff,
2. συμπληρώσει τη φόρμα δημιουργίας νέου συνεδρίου,
3. ορίσει τίτλο, περιγραφή και τοποθεσία,
4. ορίσει τις ημερομηνίες έναρξης και λήξης των υποβολών,
5. ορίσει τις ημερομηνίες διεξαγωγής του συνεδρίου,
6. καταχωρίσει τον μέγιστο αριθμό tracks,
7. ορίσει το κόστος συμμετοχής και το νόμισμα πληρωμής,
8. δημιουργήσει το συνέδριο αρχικά σε κατάσταση draft,
9. επιβεβαιώσει ότι το draft συνέδριο δεν εμφανιζόταν στη δημόσια αρχική σελίδα,
10. αλλάξει την κατάσταση του συνεδρίου σε published,
11. επιβεβαιώσει ότι το συνέδριο εμφανίστηκε στη δημόσια λίστα,
12. αλλάξει την κατάσταση του συνεδρίου σε archived,

13. ελέγξει τον περιορισμό σύμφωνα με τον οποίο η λήξη των υποβολών πρέπει να προηγείται ή να συμπίπτει με την έναρξη του συνεδρίου,
14. επιβεβαιώσει ότι μετά τη δημιουργία του συνεδρίου απέκτησε τον ρόλο organizer,
15. μεταβεί στο περιβάλλον Django Administration.

5.4 Αποτελέσματα Αξιολόγησης

Οι συμμετέχοντες εμφανίζονται στους πίνακες ως Σ1–Σ7. Επειδή οι ενότητες του ερωτηματολογίου εμφανίζονταν δυναμικά ανάλογα με τον ρόλο και τις ενέργειες κάθε χρήστη, στις ειδικές ενότητες υπολογίστηκε ο μέσος όρος μόνο από τις διαθέσιμες απαντήσεις.

5.4.1 Γενική Αξιολόγηση

Η γενική αξιολόγηση έγινε και από τους επτά συμμετέχοντες στην αξιολόγηση.

Ερώτηση	Σ1	Σ2	Σ3	Σ4	Σ5	Σ6	Σ7	Μέση βαθμολογία
A1	5	4	4	5	4	4	5	4,43
A2	4	4	5	4	4	4	5	4,29
A3	4	4	4	5	4	4	4	4,14
A4	4	4	4	4	3	4	5	4,00
A5	4	4	5	4	4	4	4	4,14
A6	5	4	4	5	4	4	5	4,43
A7	4	4	5	4	4	4	5	4,29
A8	4	4	5	4	4	5	4	4,29
A9	4	4	5	4	4	4	4	4,14
A10	4	4	5	5	4	4	4	4,29
A11	5	4	5	4	5	5	5	4,71
A12	4	4	5	4	4	4	5	4,29
A13	4	4	4	5	4	4	5	4,29
A14	4	4	5	4	4	4	5	4,29
Μέση βαθμολογία συμμετέχοντα	4,21	4,00	4,64	4,36	4,00	4,14	4,64	4,29

Πίνακας 14 Αποτελέσματα Γενικής Αξιολόγησης της Εφαρμογής

Η γενική αξιολόγηση της εφαρμογής ήταν θετική, με συνολική μέση βαθμολογία 4,29/5. Η υψηλότερη βαθμολογία καταγράφηκε στην ερώτηση A11, η οποία αφορούσε την

προστασία των papers, των προσωπικών δεδομένων και των λειτουργιών που προορίζονται για συγκεκριμένους ρόλους, με μέση τιμή 4,71/5.

Θετικά αξιολογήθηκαν επίσης η σαφήνεια του ρόλου του χρήστη και η ενημέρωση μετά την ολοκλήρωση σημαντικών ενεργειών, με μέση βαθμολογία 4,43/5. Η χαμηλότερη βαθμολογία καταγράφηκε στην ερώτηση A4, σχετικά με τη σαφήνεια των διαφορετικών καταστάσεων των συνεδρίων και των papers, με μέση τιμή 4,00/5.

Οι ερωτήσεις σχετικά με την υποστήριξη ενός πραγματικού συνεδρίου και την πρόθεση χρήσης ή πρότασης της εφαρμογής συγκέντρωσαν μέση βαθμολογία 4,29/5. Το αποτέλεσμα δείχνει ότι οι συμμετέχοντες θεωρούν την εφαρμογή κατάλληλη για πραγματικές συνθήκες χρήσης και θα ήταν θετικοί στη μελλοντική χρήση ή πρότασή της.

5.4.2 Αξιολόγηση από μη Συνδεδεμένο Επισκέπτη

Η ενότητα απαντήθηκε από τον Σ1, ο οποίος χρησιμοποίησε την εφαρμογή χωρίς να συνδεθεί.

Ερώτηση	Σ1	Μέση βαθμολογία
B1	5	5,00
B2	4	4,00
B3	5	5,00
B4	4	4,00
B5	4	4,00
Μέση βαθμολογία συμμετέχοντα	4,40	4,40

Πίνακας 15 Αποτελέσματα Αξιολόγησης από Μη Συνδεδεμένο Χρήστη

Ο μη συνδεδεμένος επισκέπτης αξιολόγησε θετικά τη δυνατότητα εντοπισμού των δημοσιευμένων συνεδρίων και τη σαφή διάκριση μεταξύ των δημόσιων πληροφοριών και των ενεργειών που απαιτούν σύνδεση. Η μέση βαθμολογία της ενότητας ήταν 4,40/5.

Οι χαμηλότερες βαθμολογίες αφορούσαν τη σαφήνεια ορισμένων πληροφοριών του συνεδρίου, όπως οι καταστάσεις των υποβολών και τα επόμενα στάδια της διαδικασίας.

5.4.3 Προσωπικός Πίνακας ελέγχου και Λίστα Παρακολούθησης

Οι ερωτήσεις Γ1–Γ6 απαντήθηκαν από τους έξι συνδεδεμένους συμμετέχοντες. Οι Γ7–Γ9, οι οποίες αφορούσαν τη watchlist, απαντήθηκαν μόνο από τον Σ2.

Ερώτηση	Σ2	Σ3	Σ4	Σ5	Σ6	Σ7	Μέση βαθμολογία
Γ1	4	4	5	5	4	4	4,33
Γ2	5	5	5	5	4	5	4,83
Γ3	4	5	5	5	5	4	4,67
Γ4	5	5	5	4	5	5	4,83
Γ5	5	5	5	5	4	4	4,67
Γ6	4	4	5	5	4	4	4,33
Γ7	5	—	—	—	—	—	5,00
Γ8	5	—	—	—	—	—	5,00
Γ9	5	—	—	—	—	—	5,00
Μέση βαθμολογία συμμετέχοντα	4,67	4,67	5,00	4,83	4,33	4,33	4,64

Πίνακας 16 Αποτελέσματα Αξιολόγησης Προσωπικού Πίνακα Ελέγχου & Λίστας Παρακολούθησης

Η ενότητα του προσωπικού πίνακα ελέγχου συγκέντρωσε συνολική μέση βαθμολογία 4,64/5. Ιδιαίτερα θετικά αξιολογήθηκαν η σύνδεση κάθε ειδοποίησης, ρόλου ή υποβολής με το αντίστοιχο συνέδριο και η λειτουργία επισήμανσης των ειδοποιήσεων ως αναγνωσμένων, με μέση βαθμολογία 4,83/5.

Η λειτουργία της λίστας παρακολούθησης αξιολογήθηκε με 5,00/5 από τον συμμετέχοντα που τη χρησιμοποίησε. Ο χρήστης μπόρεσε να προσθέσει και να αφαιρέσει συνέδρια με προβλέσιμο τρόπο και κατανόησε ότι η watchlist δεν του απέδιδε κάποιον ρόλο στο συνέδριο.

5.4.4 Αξιολόγηση Λειτουργιών Παρατηρητή

Η ενότητα απαντήθηκε από τον Σ2, ο οποίος εγγράφηκε ως παρατηρητής σε ένα συνέδριο.

Ερώτηση	Σ2	Μέση βαθμολογία
Δ1	4	4,00
Δ2	5	5,00
Δ3	4	4,00
Δ4	4	4,00
Μέση βαθμολογία συμμετέχοντα	4,25	4,25

Πίνακας 17 Αποτελέσματα Αξιολόγησης Λειτουργιών Παρατηρητή

Η λειτουργία εγγραφής ως παρατηρητή συγκέντρωσε μέση βαθμολογία 4,25/5. Η νέα ιδιότητα εμφανίστηκε σωστά στη σελίδα του συνεδρίου και στον πίνακα ελέγχου.

Ωστόσο, ο συμμετέχων ανέφερε ότι θα ήταν χρήσιμη μια σαφέστερη επεξήγηση της διαφοράς μεταξύ της προσθήκης ενός συνεδρίου στη watchlist και της εγγραφής σε αυτό ως observer.

5.4.5 Αξιολόγηση Λειτουργιών Παρουσιαστή

Η ενότητα απαντήθηκε από τους Σ3 και Σ4, οι οποίοι πραγματοποίησαν υποβολή paper. Οι ερωτήσεις E10–E13 σχετικά με την πληρωμή απαντήθηκαν μόνο από τον Σ4, του οποίου η υποβολή ανατέθηκε σε θεματική ενότητα και έφθασε στην κατάσταση presentation ready.

Ερώτηση	Σ3	Σ4	Μέση βαθμολογία
E1	5	5	5,00
E2	4	5	4,50
E3	4	4	4,00
E4	5	5	5,00
E5	5	5	5,00
E6	5	5	5,00
E7	3	4	3,50
E8	5	5	5,00
E9	–	4	4,00
E10	–	5	5,00
E11	–	4	4,00
E12	–	5	5,00
E13	–	4	4,00
Μέση βαθμολογία συμμετέχοντα	4,50	4,62	4,57

Πίνακας 18 Αποτελέσματα Αξιολόγησης Λειτουργιών Παρουσιαστή

Η ενότητα των presenters συγκέντρωσε συνολική μέση βαθμολογία 4,57/5. Οι συμμετέχοντες αξιολόγησαν ιδιαίτερα θετικά τη σαφήνεια της περιόδου υποβολών, την αποτροπή μη επιτρεπτών υποβολών, την άμεση εμφάνιση της υποβολής και τη δυνατότητα λήψης του υποβληθέντος αρχείου.

Η χαμηλότερη βαθμολογία της ενότητας καταγράφηκε στην ερώτηση E7, με μέση τιμή 3,50/5, και αφορούσε την κατανόηση του σταδίου στο οποίο βρισκόταν η υποβολή. Οι συμμετέχοντες πρότειναν την εμφάνιση μιας γραμμής προόδου ή ενός διαγράμματος με τα στάδια της διαδικασίας.

Η διαδικασία πληρωμής αξιολογήθηκε θετικά. Ωστόσο, επισημάνθηκε ότι πρέπει να εξηγείται σαφέστερα πως η πληρωμή ενεργοποιείται μόνο μετά την ανάθεση του accepted paper σε track.

5.4.6 Αξιολόγηση Λειτουργιών Αξιολογητή

Η ενότητα απαντήθηκε από τον Σ5, ο οποίος αξιολόγησε υποβολές που του είχαν ανατεθεί.

Ερώτηση	Σ5	Μέση βαθμολογία
ΣΤ1	5	5,00
ΣΤ2	5	5,00
ΣΤ3	4	4,00
ΣΤ4	5	5,00
ΣΤ5	4	4,00
ΣΤ6	4	4,00
ΣΤ7	4	4,00
ΣΤ8	5	5,00
ΣΤ9	5	5,00
ΣΤ10	4	4,00
ΣΤ11	5	5,00
Μέση βαθμολογία συμμετέχοντα	4,55	4,55

Πίνακας 19 Αποτελέσματα Αξιολόγησης Λειτουργιών Αξιολογητή

Οι λειτουργίες του reviewer συγκέντρωσαν μέση βαθμολογία 4,55/5. Θετικά αξιολογήθηκαν οι ειδοποιήσεις ανάθεσης, η εμφάνιση των εκκρεμών αξιολογήσεων, η δυνατότητα λήψης του paper και η αφαίρεσή του από την ουρά μετά την ολοκλήρωση της αξιολόγησης.

Χαμηλότερες βαθμολογίες καταγράφηκαν στη σαφήνεια των πληροφοριών της φόρμας αξιολόγησης και στη διάκριση μεταξύ των πιθανών αποτελεσμάτων, όπως «απόρριψη» και «χρειάζεται αλλαγές». Προτάθηκε η προσθήκη σύντομων επεξηγήσεων για την επίδραση κάθε επιλογής στην κατάσταση του paper.

5.4.7 Αξιολόγηση Λειτουργιών Οργανωτικής Επιτροπής

Η ενότητα απαντήθηκε από τους Σ6 και Σ7, οι οποίοι χρησιμοποίησαν λειτουργίες διαχείρισης μελών, reviewers, tracks και papers.

Ερώτηση	Σ6	Σ7	Μέση βαθμολογία
Z1	4	5	4,50
Z2	5	5	5,00
Z3	4	4	4,00
Z4	3	4	3,50
Z5	4	5	4,50
Z6	4	5	4,50
Z7	4	5	4,50

Z8	5	5	5,00
Z9	3	4	3,50
Z10	4	5	4,50
Z11	5	5	5,00
Z12	4	5	4,50
Z13	5	5	5,00
Z14	4	5	4,50
Z15	5	5	5,00
Z16	4	5	4,50
Μέση βαθμολογία συμμετέχοντα	4,19	4,81	4,50

Πίνακας 20 Αποτελέσματα Αξιολόγησης Λειτουργιών Οργανωτικής Επιτροπής

Η ενότητα των organizers συγκέντρωσε συνολική μέση βαθμολογία 4,50/5. Υψηλά αξιολογήθηκαν η εμφάνιση των νέων μελών και των ρόλων τους, η αποτροπή ανάθεσης του ίδιου χρήστη ως πρώτου και δεύτερου reviewer, η εμφάνιση της ανάγκης τρίτου reviewer και η ανάθεση accepted papers σε tracks.

Οι χαμηλότερες βαθμολογίες καταγράφηκαν στις ερωτήσεις Z4 και Z9, με μέση τιμή 3,50/5. Οι ερωτήσεις αφορούσαν την αποτροπή λανθασμένων ή διπλών αναθέσεων ρόλων και την παρακολούθηση της προόδου των reviews. Προτάθηκε η δημιουργία ενός συγκεντρωτικού πίνακα με φίλτρα ανά paper, κατάσταση, reviewer και track.

5.4.8 Αξιολόγηση Λειτουργιών Διαχειριστή – Superuser

Η ενότητα απαντήθηκε από τον Σ7, ο οποίος δημιούργησε και διαχειρίστηκε συνέδριο.

Ερώτηση	Σ7	Μέση βαθμολογία
H1	5	5,00
H2	4	4,00
H3	5	5,00
H4	4	4,00
H5	5	5,00
H6	5	5,00
H7	5	5,00
H8	5	5,00
H9	4	4,00
H10	4	4,00
H11	4	4,00
Μέση βαθμολογία συμμετέχοντα	4,55	4,55

Πίνακας 21 Αποτελέσματα Αξιολόγησης Λειτουργιών Διαχειριστή/Superuser

Οι λειτουργίες του διαχειριστή – superuser συγκέντρωσαν μέση βαθμολογία 4,55/5. Θετικά αξιολογήθηκαν η πληρότητα της φόρμας δημιουργίας συνεδρίου, ο έλεγχος των

ημερομηνιών, η αυτόματη απόδοση του ρόλου organizer και η διαχείριση της κατάστασης draft, published ή archived.

Περιθώρια βελτίωσης εντοπίστηκαν στη σαφήνεια των μηνυμάτων σφάλματος για μη έγκυρες ημερομηνίες και στη διάκριση μεταξύ της καρτέλας Staff και του Django Administration.

5.4.9 Συνολική Βαθμολογία Ευχρηστίας – Αποτελεσματικότητας

Οι ερωτήσεις για την ευχρηστία και την αποτελεσματικότητα αξιολογήθηκαν σε κλίμακα από 1 έως 10.

Ερώτηση	Σ1	Σ2	Σ3	Σ4	Σ5	Σ6	Σ7	Μέση βαθμολογία
Θ1 — Συνολική ευχρηστία	8	9	8	9	8	7	8	8,14
Θ2 — Αποτελεσματική υποστήριξη εργασιών	8	9	8	9	9	8	9	8,57
Μέση βαθμολογία συμμετέχοντα	8,00	9,00	8,00	9,00	8,50	7,50	8,50	8,36

Πίνακας 22 Συνολική Αξιολόγηση Ευχρηστίας & Αποτελεσματικότητας Εφαρμογής

Η συνολική ευχρηστία της εφαρμογής αξιολογήθηκε με 8,14/10, ενώ η αποτελεσματικότητα υποστήριξης των εργασιών των χρηστών με 8,57/10. Η συνολική μέση βαθμολογία των δύο διαστάσεων ήταν 8,36/10.

Ο υψηλότερος μέσος όρος συμμετέχοντα ήταν 9,00/10 και καταγράφηκε στους Σ2 και Σ4. Ο χαμηλότερος ήταν 7,50/10 και καταγράφηκε στον Σ6, ο οποίος χρησιμοποίησε τις πιο σύνθετες λειτουργίες διαχείρισης ως organizer.

5.5 Αποτελέσματα Ανοικτών Ερωτήσεων

Από τις ανοικτές απαντήσεις προέκυψαν τα ακόλουθα βασικά σημεία:

- Οι καταστάσεις των papers δεν ήταν πάντοτε πλήρως κατανοητές, ιδιαίτερα ως προς το επόμενο στάδιο της διαδικασίας.
- Χρειάζεται σαφέστερη διάκριση μεταξύ της watchlist και της εγγραφής ως observer.
- Η εφαρμογή πρέπει να εξηγεί ότι η πληρωμή ενεργοποιείται μετά την ανάθεση του paper σε track και όχι αμέσως μετά την αποδοχή.

- Για τους reviewers θα ήταν χρήσιμη μια ένδειξη προόδου, όπως «1 από 2 αξιολογήσεις ολοκληρώθηκαν».
- Οι organizers χρειάζονται συγκεντρωτική προβολή και φίλτρα για τη διαχείριση πολλών papers, reviewers και tracks.
- Τα μηνύματα ελέγχου ημερομηνιών κατά τη δημιουργία συνεδρίου μπορούν να υποδεικνύουν με μεγαλύτερη ακρίβεια το πεδίο που πρέπει να διορθωθεί.
- Προτάθηκε η προσθήκη προεπισκόπησης του συνεδρίου πριν από τη δημοσίευσή του.

Συνολικά, οι συμμετέχοντες θεώρησαν ότι η εφαρμογή καλύπτει αποτελεσματικά τις βασικές λειτουργίες διαχείρισης ενός συνεδρίου. Οι κυριότερες προτάσεις βελτίωσης δεν αφορούσαν την απουσία βασικών λειτουργιών, αλλά τη σαφέστερη παρουσίαση της κατάστασης των διαδικασιών, την καλύτερη πληροφόρηση για τα επόμενα βήματα και την αποτελεσματικότερη διαχείριση μεγάλου αριθμού δεδομένων.

5.6 Χρησιμότερες Λειτουργίες

Οι συμμετέχοντες ανέδειξαν ως ιδιαίτερα χρήσιμες τις λειτουργίες που παρείχαν συγκεντρωμένη πληροφόρηση και άμεση πρόσβαση στις εκκρεμείς ενέργειές τους.

Ο προσωπικός πίνακας ελέγχου αξιολογήθηκε θετικά, καθώς επέτρεπε στους συνδεδεμένους χρήστες να βλέπουν σε ένα σημείο τις ειδοποιήσεις, τους ρόλους τους ανά συνέδριο, τις υποβολές, τη λίστα παρακολούθησης και τις εκκρεμείς αξιολογήσεις. Χρήσιμοι θεωρήθηκαν επίσης οι σύνδεσμοι που οδηγούσαν απευθείας στο αντίστοιχο συνέδριο ή στην εργασία που έπρεπε να ολοκληρωθεί.

Η watchlist κρίθηκε χρήσιμη για τη γρήγορη πρόσβαση στα συνέδρια που ενδιέφεραν τον χρήστη. Παράλληλα, η εγγραφή ως observer επέτρεπε τη σαφή καταγραφή της συμμετοχής ενός χρήστη σε συγκεκριμένο συνέδριο.

Οι presenters αξιολόγησαν θετικά την άμεση εμφάνιση του paper μετά την υποβολή, τη δυνατότητα λήψης του αρχείου και τις ειδοποιήσεις που αφορούσαν την αποδοχή ή την απόρριψή του. Η σύνδεση της ανάθεσης σε track με την ενεργοποίηση της πληρωμής υποστήριζε επίσης την ορθή ακολουθία της διαδικασίας.

Για τους reviewers, ιδιαίτερα χρήσιμη ήταν η ουρά εκκρεμών αξιολογήσεων, καθώς εμφάνιζε τα papers που απαιτούσαν ενέργεια και παρείχε άμεση πρόσβαση στη φόρμα αξιολόγησης. Η αφαίρεση ενός paper από την ουρά μετά την υποβολή της αξιολόγησης λειτουργούσε ως επιβεβαίωση ολοκλήρωσης της εργασίας.

Οι organizers θεώρησαν χρήσιμες τις λειτουργίες ανάθεσης ρόλων, reviewers και τρίτου reviewer, καθώς και τη δυνατότητα δημιουργίας tracks και ένταξης accepted papers σε αυτά. Για τον staff/superuser χρήστη, σημαντική ήταν η συγκεντρωτική διαχείριση των συνεδρίων και η δυνατότητα αλλαγής της κατάστασής τους μεταξύ draft, published και archived.

Συνολικά, οι χρησιμότερες λειτουργίες ήταν εκείνες που υποστήριζαν την παρακολούθηση της προόδου, την ενημέρωση των χρηστών και τη γρήγορη πρόσβαση στις ενέργειες που αντιστοιχούσαν στον ρόλο τους.

5.7 Κυριότερες Δυσκολίες

Η βασικότερη δυσκολία που εντοπίστηκε αφορούσε την κατανόηση των διαφορετικών καταστάσεων των papers και της σειράς με την οποία εξελίσσεται η διαδικασία. Όροι όπως submitted, under review, accepted, track assigned και presentation ready δεν καθιστούσαν πάντοτε σαφές ποιο στάδιο είχε ολοκληρωθεί και ποια ήταν η επόμενη ενέργεια.

Οι presenters δυσκολεύτηκαν κυρίως να κατανοήσουν γιατί η πληρωμή δεν ήταν διαθέσιμη αμέσως μετά την αποδοχή του paper. Δεν ήταν εξαρχής εμφανές ότι έπρεπε να προηγηθεί η ανάθεση σε track.

Στην περίπτωση του observer, παρατηρήθηκε αρχική ασάφεια ως προς τη διαφορά μεταξύ της προσθήκης ενός συνεδρίου στη watchlist και της εγγραφής ως παρατηρητής. Παρότι οι δύο λειτουργίες εκτελούν διαφορετικό σκοπό, η διαφορά τους θα μπορούσε να παρουσιάζεται με σαφέστερο τρόπο.

Ο reviewer αντιμετώπισε δυσκολία στη διάκριση μεταξύ ορισμένων αποτελεσμάτων αξιολόγησης, όπως η απόρριψη και η ανάγκη αλλαγών. Επιπλέον, δεν ήταν πάντοτε σαφές αν είχε ολοκληρωθεί μόνο η δική του αξιολόγηση ή και η συνολική διαδικασία review του paper.

Οι organizers αντιμετώπισαν μεγαλύτερη πολυπλοκότητα κατά τη διαχείριση πολλών papers, reviewers και tracks. Η παρακολούθηση των εκκρεμών αναθέσεων απαιτούσε τη

μετάβαση μεταξύ διαφορετικών πλαισίων, ενώ δεν υπήρχε ενιαία συγκεντρωτική προβολή με φίλτρα.

Ο staff/superuser χρήστης εντόπισε μικρότερη σαφήνεια στα μηνύματα ελέγχου των ημερομηνιών κατά τη δημιουργία συνεδρίου. Το σύστημα απέτρεπε τις μη έγκυρες τιμές, αλλά δεν υπεδείκνυε πάντοτε με ακρίβεια ποιο πεδίο έπρεπε να διορθωθεί.

Οι δυσκολίες που καταγράφηκαν δεν αφορούσαν κυρίως την απουσία λειτουργιών, αλλά τη σαφήνεια της πληροφόρησης, την κατανόηση των επόμενων βημάτων και τη διαχείριση πιο σύνθετων σεναρίων.

5.8 Προτεινόμενες Βελτιώσεις

Με βάση τις παρατηρήσεις των συμμετεχόντων, προτείνονται οι ακόλουθες βελτιώσεις:

- προσθήκη σύντομων επεξηγήσεων για τις καταστάσεις των papers,
- εμφάνιση γραμμής προόδου με το τρέχον και το επόμενο στάδιο κάθε υποβολής,
- σαφέστερη ενημέρωση ότι η πληρωμή ενεργοποιείται μετά την ανάθεση του paper σε track,
- εμφανέστερη διάκριση μεταξύ watchlist και εγγραφής ως observer,
- προσθήκη επεξηγήσεων στις διαθέσιμες επιλογές αξιολόγησης,
- εμφάνιση ένδειξης προόδου των reviews, όπως «1 από 2 αξιολογήσεις ολοκληρώθηκαν»,
- δημιουργία συγκεντρωτικού πίνακα για organizers με φίλτρα ανά paper, reviewer, κατάσταση και track,
- βελτίωση των μηνυμάτων σφάλματος και ελέγχου εγκυρότητας κατά τη δημιουργία συνεδρίου,
- προσθήκη προεπισκόπησης πριν από τη δημοσίευση ενός συνεδρίου,
- πιο εμφανής επιβεβαίωση της επιτυχούς πληρωμής και της μετάβασης σε presentation ready.

Οι συγκεκριμένες προτάσεις στοχεύουν κυρίως στη βελτίωση της πληροφόρησης και όχι στην αλλαγή της βασικής λειτουργικότητας της εφαρμογής.

5.9 Περιορισμοί της Αξιολόγησης

Η αξιολόγηση πραγματοποιήθηκε με μικρό δείγμα επτά συμμετεχόντων. Για τον λόγο αυτό, τα αποτελέσματα δεν μπορούν να γενικευθούν στο σύνολο των πιθανών χρηστών της εφαρμογής.

Επιπλέον, ορισμένοι ρόλοι αξιολογήθηκαν από έναν μόνο συμμετέχοντα. Επομένως, οι διαφορές μεταξύ των ρόλων δεν μπορούν να θεωρηθούν στατιστικά τεκμηριωμένες, αλλά αποτελούν ενδεικτικές παρατηρήσεις.

Η αξιολόγηση πραγματοποιήθηκε σε ελεγχόμενο τοπικό περιβάλλον και όχι κατά τη διαχείριση ενός πραγματικού συνεδρίου. Κατά συνέπεια, δεν εξετάστηκαν πλήρως συνθήκες όπως ο μεγάλος αριθμός ταυτόχρονων χρηστών, ο μεγάλος όγκος υποβολών και οι πραγματικές χρονικές πιέσεις μιας διοργάνωσης.

Τα σενάρια χρήσης ήταν προκαθορισμένα και δεν κάλυψαν κάθε πιθανή εξαίρεση ή μη αναμενόμενη συμπεριφορά. Επίσης, η εξοικείωση των συμμετεχόντων με διαδικτυακές εφαρμογές ενδέχεται να επηρέασε τον τρόπο με τον οποίο αξιολόγησαν την ευχρηστία.

Τέλος, οι απαντήσεις βασίστηκαν στην προσωπική αντίληψη των συμμετεχόντων. Επομένως, οι βαθμολογίες αποτυπώνουν την αντιλαμβανόμενη ευχρηστία και λειτουργικότητα και όχι αποκλειστικά αντικειμενικές μετρήσεις απόδοσης.

5.10 Συμπεράσματα Αξιολόγησης

Τα αποτελέσματα δείχνουν ότι η εφαρμογή υποστηρίζει σε ικανοποιητικό βαθμό τις βασικές λειτουργίες διαχείρισης συνεδρίων και τις εργασίες των διαφορετικών ρόλων χρηστών.

Οι συμμετέχοντες μπόρεσαν να ολοκληρώσουν τα κύρια σενάρια, όπως την αναζήτηση συνεδρίων, την παρακολούθηση μέσω watchlist, την υποβολή paper, την αξιολόγηση, την ανάθεση reviewers, τη διαχείριση tracks και τη δημιουργία συνεδρίου. Θετικά αξιολογήθηκαν επίσης η προστασία των λειτουργιών ανά ρόλο, η εμφάνιση των σχετικών ειδοποιήσεων και η συγκέντρωση των προσωπικών πληροφοριών στον πίνακα ελέγχου.

Οι χαμηλότερες αξιολογήσεις αφορούσαν κυρίως τη σαφήνεια των καταστάσεων και των επόμενων βημάτων. Οι χρήστες χρειάζονταν περισσότερη ενημέρωση σχετικά με την

εξέλιξη μιας υποβολής, τις προϋποθέσεις της πληρωμής και την πρόοδο της διαδικασίας αξιολόγησης.

Οι πιο σύνθετοι ρόλοι, ιδιαίτερα ο organizer, παρουσίασαν μεγαλύτερες απαιτήσεις ως προς την οργάνωση και την εποπτεία των δεδομένων. Το εύρημα αυτό δείχνει ότι η εφαρμογή θα μπορούσε να ενισχυθεί με συγκεντρωτικές προβολές και εργαλεία φιλτραρίσματος.

Συνολικά, η εφαρμογή θεωρήθηκε κατάλληλη για την υποστήριξη ενός πραγματικού συνεδρίου, υπό την προϋπόθεση ότι θα βελτιωθούν ορισμένα στοιχεία παρουσίασης και καθοδήγησης. Οι προτεινόμενες αλλαγές αφορούν κυρίως τη σαφέστερη επικοινωνία της κατάστασης των διαδικασιών και την αποτελεσματικότερη διαχείριση σύνθετων εργασιών.

6. Συμπεράσματα

Η εφαρμογή καλύπτει τις βασικές απαιτήσεις διαχείρισης συνεδρίων: δημιουργία και δημοσίευση, παρακολούθηση μέσω watchlist, ροές υποβολής και αξιολόγησης με κανόνα ομοφωνίας και κλιμάκωση σε τρίτο reviewer, ανάθεση track και ολοκλήρωση πληρωμής για τελική επιβεβαίωση παρουσίας. Παράλληλα, ενσωματώθηκαν πρακτικές παραγωγικής ετοιμότητας με έμφαση σε ασφάλεια και audit. Η εργασία καταδεικνύει ότι η παραγωγική ωρίμανση δεν αποτελεί δραστηριότητα που ακολουθεί την υλοποίηση λειτουργιών, αλλά σύνολο πρακτικών που πρέπει να συνδέονται με τις αποφάσεις σχεδιασμού. Η υποστήριξη πληρωμών με υπογεγραμμένα webhooks και idempotency αναδεικνύει την ανάγκη οι εξωτερικές ολοκληρώσεις να αντιμετωπίζονται ως σημεία υψηλού ρίσκου, με ρητό έλεγχο ακεραιότητας και με μηχανισμούς αποφυγής διπλής επεξεργασίας. Η τεκμηρίωση μέσω διαγραμμάτων και στιγμιotypών οθόνης λειτουργεί συμπληρωματικά προς τον κώδικα, καθώς καθιστά σαφείς τις αποφάσεις σχεδιασμού, τις αλλαγές και τα αποτέλεσματα επαλήθευσης. Η αναπαραγωγικότητα των artifacts ενισχύει την ακαδημαϊκή αξία της εργασίας, διότι επιτρέπει να ελεγχθούν τα βήματα που οδήγησαν στα παραδοτέα και να αξιολογηθεί η συνέπεια μεταξύ περιγραφής και υλοποίησης. Με βάση τα παραπάνω, η υλοποίηση αποτελεί συνεκτική βάση για περαιτέρω επέκταση σε πλήρη πλατφόρμα διαχείρισης ακαδημαϊκών συνεδρίων. Η επέκταση μπορεί να κινηθεί προς τον εμπλουτισμό του μοντέλου (πρόγραμμα, slots, speakers), προς την περαιτέρω αυτοματοποίηση λειτουργιών (ασύγχρονα tasks, αναφορές) και προς την ενίσχυση διαδικασιών λειτουργίας (deployment automation, incident response), αξιοποιώντας τα θεμέλια ασφάλειας και παρατηρησιμότητας που έχουν ήδη ενσωματωθεί.

Βιβλιογραφία

Ακολουθούν οι βιβλιογραφικές αναφορές (πηγές) της Εργασίας.

EasyChair. (χ.χ.). *General Information*. Ανακτήθηκε από <https://easychair.org/docs/general>

Microsoft Conference Management Toolkit. (χ.χ.). *Features*. Ανακτήθηκε από <https://cmt3.research.microsoft.com/docs/help/overview/features.html>

Indico. (χ.χ.). *Learning Indico*. Ανακτήθηκε από <https://learn.getindico.io/>

Indico – CERN. (χ.χ.). *Home – Indico*. Ανακτήθηκε από <https://indico.cern.ch/>

OpenConf. (χ.χ.). *Peer-Review Software System*. Ανακτήθηκε από <https://www.openconf.com/peer-review/>

HotCRP. (χ.χ.). *HotCRP Conference Management Software*. Ανακτήθηκε από <https://www.read.seas.harvard.edu/~kohler/hotcrp/>

HotCRP. (χ.χ.). *HotCRP.com*. Ανακτήθηκε από <https://hotcrp.com/>

Kohler E., (2008). *Hot Crap!*. Proceedings of the Workshop on Organizing Workshops, Conferences, and Symposia for Computer Systems. USENIX Association. Ανακτήθηκε από https://www.usenix.org/legacy/event/wowcs08/tech/full_papers/kohler/kohler.pdf

ConfTool. (χ.χ.). *Submission, Reviewing, Scheduling, Participant Registration – Main Features at a Glance*. Ανακτήθηκε από <https://www.conftool.net/en/features.html>

EDAS. (2026, 15 Μαρτίου). *Features and Pricing*. Ανακτήθηκε από <https://edas.info/doc/features.html>

Whova. (χ.χ.). *Event Apps & Event Management*. Ανακτήθηκε από <https://whova.com/>

Whova. (χ.χ.). *All-in-One Event Management Software*. Ανακτήθηκε από <https://whova.com/event-management-software/>

Eventbrite. (χ.χ.). *Event Management Software*. Ανακτήθηκε από <https://www.eventbrite.com/organizer/features/event-management-software/>

Eventbrite. (χ.χ.). *The Best Event Registration Platform*. Ανακτήθηκε από <https://www.eventbrite.com/organizer/features/registration-online/>

Django Software Foundation. (χ.χ.). *Django: The web framework for perfectionists with deadlines*. Ανακτήθηκε από <https://www.djangoproject.com/>

Django Software Foundation. (χ.χ.). *Django documentation*. Ανακτήθηκε από <https://docs.djangoproject.com/en/6.0/>

Python Software Foundation. (n.d.-a). *General Python FAQ*. Ανακτήθηκε από <https://docs.python.org/3/faq/general.html>

Python Software Foundation. (n.d.-b). *Python documentation*. Ανακτήθηκε από <https://docs.python.org/3/>

Python Software Foundation. (n.d.-c). *The Python Standard Library*. Ανακτήθηκε από <https://docs.python.org/3/library/index.html>

Python Software Foundation. (n.d.-d). *venv — Creation of virtual environments*. Ανακτήθηκε από <https://docs.python.org/3/library/venv.html>

Peters, T. (2004). *PEP 20 – The Zen of Python*. Python Enhancement Proposals. Ανακτήθηκε από <https://peps.python.org/pep-0020/>

Django Software Foundation. (n.d.). *The web framework for perfectionists with deadlines*. Ανακτήθηκε από <https://www.djangoproject.com/>

Django Software Foundation. (n.d.-a). *The web framework for perfectionists with deadlines*. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.djangoproject.com/>

Django Software Foundation. (n.d.-b). *Django overview*. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.djangoproject.com/start/overview/>

Django Software Foundation. (n.d.-c). *FAQ: General*. Ανακτήθηκε 20 Μαΐου 2026, από <https://docs.djangoproject.com/en/6.0/faq/general/>

Django Software Foundation. (n.d.-d). *Models*. Ανακτήθηκε 20 Μαΐου 2026, από <https://docs.djangoproject.com/en/6.0/topics/db/models/>

Django Software Foundation. (n.d.-e). *Templates*. Ανακτήθηκε 20 Μαΐου 2026, από <https://docs.djangoproject.com/en/6.0/ref/templates/>

Django Software Foundation. (n.d.-f). *Writing views*. Ανακτήθηκε 20 Μαΐου 2026, από <https://docs.djangoproject.com/en/6.0/topics/http/views/>

Django Software Foundation. (n.d.-g). *The Django admin site*. Ανακτήθηκε 20 Μαΐου 2026, από <https://docs.djangoproject.com/en/6.0/ref/contrib/admin/>

- Django Software Foundation. (n.d.-h). *Using the Django authentication system*. Ανακτήθηκε 20 Μαΐου 2026, από <https://docs.djangoproject.com/en/6.0/topics/auth/default/>
- Django Software Foundation. (n.d.-i). *Security in Django*. Ανακτήθηκε 20 Μαΐου 2026, από <https://docs.djangoproject.com/en/6.0/topics/security/>
- SQLite. (n.d.-a). *About SQLite*. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.sqlite.org/about.html>
- SQLite. (n.d.-b). *SQLite is a zero-configuration database*. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.sqlite.org/zeroconf.html>
- SQLite. (n.d.-c). *Features of SQLite*. Ανακτήθηκε 20 Μαΐου 2026, από <https://sqlite.org/features.html>
- SQLite. (n.d.-d). *Appropriate uses for SQLite*. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.sqlite.org/whentouse.html>
- SQLite. (n.d.-e). *SQLite is serverless*. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.sqlite.org/serverless.html>
- Python Software Foundation. (n.d.). *sqlite3 — DB-API 2.0 interface for SQLite databases*. Ανακτήθηκε 20 Μαΐου 2026, από <https://docs.python.org/3/library/sqlite3.html>
- Django Software Foundation. (n.d.-a). *Databases: SQLite notes*. Ανακτήθηκε 20 Μαΐου 2026, από <https://docs.djangoproject.com/en/dev/ref/databases/#sqlite-notes>
- Django Software Foundation. (n.d.-b). *Databases: “Database is locked” errors*. Ανακτήθηκε 20 Μαΐου 2026, από <https://docs.djangoproject.com/en/dev/ref/databases/#database-is-locked-errors>
- Bray, T. (2017). *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 8259. Internet Engineering Task Force. Ανακτήθηκε 20 Μαΐου 2026, από <https://datatracker.ietf.org/doc/rfc8259/>
- Ecma International. (2017). *ECMA-404: The JSON data interchange syntax*. Ανακτήθηκε 20 Μαΐου 2026, από <https://ecma-international.org/publications-and-standards/standards/ecma-404/>
- Fielding, R., Nottingham, M., & Reschke, J. (2022). *HTTP Semantics*. RFC 9110. Internet Engineering Task Force. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.ietf.org/rfc/rfc9110.html>
- MDN Web Docs. (n.d.). *HTTP response status codes*. Ανακτήθηκε 20 Μαΐου 2026, από <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

Django Software Foundation. (n.d.-a). *Request and response objects*. Ανακτήθηκε 20 Μαΐου 2026, από <https://docs.djangoproject.com/en/6.0/ref/request-response/>

Django REST Framework. (n.d.-a). *Serializers*. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.django-rest-framework.org/api-guide/serializers/>

Django REST Framework. (n.d.-b). *Requests*. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.django-rest-framework.org/api-guide/requests/>

Django REST Framework. (n.d.-c). *Responses*. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.django-rest-framework.org/api-guide/responses/>

Django REST Framework. (n.d.-d). *Authentication*. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.django-rest-framework.org/api-guide/authentication/>

Django REST Framework. (n.d.-e). *Permissions*. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.django-rest-framework.org/api-guide/permissions/>

Django REST Framework. (n.d.-f). *Status codes*. Ανακτήθηκε 20 Μαΐου 2026, από <https://www.django-rest-framework.org/api-guide/status-codes/>

JSON:API. (n.d.). *A specification for building APIs in JSON*. Ανακτήθηκε 20 Μαΐου 2026, από <https://jsonapi.org/>

JSON:API. (n.d.). *JSON:API specification*. Ανακτήθηκε 20 Μαΐου 2026, από <https://jsonapi.org/format/>

OpenAPI Initiative. (n.d.). *OpenAPI Specification*. Ανακτήθηκε 20 Μαΐου 2026, από <https://spec.openapis.org/oas/>

Mozilla Developer Network. (χ.χ.). *HTML: HyperText Markup Language*. Ανακτήθηκε από <https://developer.mozilla.org/en-US/docs/Web/HTML>

Mozilla Developer Network. (χ.χ.). *Structuring content with HTML*. Ανακτήθηκε από https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Structuring_content

Django Software Foundation. (χ.χ.). *The Django template language*. Ανακτήθηκε από <https://docs.djangoproject.com/en/stable/ref/templates/language/>

Mozilla Developer Network. (χ.χ.). *CSS: Cascading Style Sheets*. Ανακτήθηκε από <https://developer.mozilla.org/en-US/docs/Web/CSS>

Mozilla Developer Network. (χ.χ.). *CSS styling basics*. Ανακτήθηκε από https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics

World Wide Web Consortium. (χ.χ.). *Web Standards*. Ανακτήθηκε από <https://www.w3.org/standards/>

World Wide Web Consortium. (χ.χ.). *HTML*. Ανακτήθηκε από <https://www.w3.org/html/>

World Wide Web Consortium. (χ.χ.). *CSS Specifications*. Ανακτήθηκε από <https://www.w3.org/Style/CSS/specs.en.html>

Stripe. (χ.χ.). *Stripe documentation*. Ανακτήθηκε από <https://docs.stripe.com/>

Stripe. (χ.χ.). *Checkout overview*. Ανακτήθηκε από <https://docs.stripe.com/payments/checkout>

Stripe. (χ.χ.). *Stripe API reference*. Ανακτήθηκε από <https://docs.stripe.com/api>

Mailpit. (χ.χ.). *Mailpit*. Ανακτήθηκε από <https://mailpit.axllent.org/>

Mailpit. (χ.χ.). *Mailpit GitHub Repository*. Ανακτήθηκε από <https://github.com/axllent/mailpit>

OWASP Cheat Sheet Series. (χ.χ.-a). *Content Security Policy Cheat Sheet*. Ανακτήθηκε 24
Φεβρουάριος 2026, από https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html

OWASP Cheat Sheet Series. (χ.χ.-b). *Rate Limiting Cheat Sheet*. Ανακτήθηκε 24
Φεβρουάριος 2026, από https://cheatsheetseries.owasp.org/cheatsheets/Rate_Limiting_Cheat_Sheet.html

OWASP Foundation. (2021). *OWASP Top 10*. <https://owasp.org/www-project-top-ten/>

Παράρτημα Α: Ενδεικτικός Κώδικας Εφαρμογής

Παρακάτω παρουσιάζονται επιλεγμένα αποσπάσματα του κώδικα της εφαρμογής.

Core App

Ο πυρήνας της εφαρμογής περιλαμβάνει κρίσιμες λειτουργίες υποδομής.

Email (email.py)

Το module email.py είναι υπεύθυνο για την αποστολή ειδοποιήσεων και και αναφορών στους χρήστες.

```
"""Email helpers"""

from __future__ import annotations

import logging

from django.conf import settings
from django.core.mail import EmailMultiAlternatives
from django.template.loader import render_to_string

logger = logging.getLogger(__name__)

def send_notification_email(
    *,
    to_email: str,
    title: str,
    message: str,
    cta_url: str | None = None,
    cta_label: str | None = None,
) -> bool:
    """Send a branded notification email (text + HTML).

    Designed to be safe to call from request/worker paths: failures are
    logged and do not raise.
    """
    enabled = bool(getattr(settings, "EMAIL_NOTIFICATIONS_ENABLED", False))
    if not enabled:
        return False

    to_email = (to_email or "").strip()
    if not to_email:
        return False

    subject_prefix = getattr(settings, "EMAIL_SUBJECT_PREFIX", "") or ""
    subject = f"{subject_prefix}{title}".strip()

    context = {
        "title": title,
        "message": message,
        "cta_url": cta_url,
        "cta_label": cta_label,
        "site_base_url": getattr(settings, "SITE_BASE_URL", "") or "",
    }

    try:
        text_body = render_to_string("emails/notification.txt",
```

```
context).strip()
    html_body = render_to_string("emails/notification.html",
context).strip()

    email = EmailMultiAlternatives(
        subject=subject,
        body=text_body,
        from_email=getattr(settings, "DEFAULT_FROM_EMAIL", None),
        to=[to_email],
    )
    if html_body:
        email.attach_alternative(html_body, "text/html")
    email.send(fail_silently=False)
    return True
except Exception:
    logger.exception("Failed to send notification email to %s", to_email)
    return False
```

Βοηθητικά εργαλεία (utils.py)

Σε αυτό το module υπάρχουν μια μέθοδος που προσπαθεί να εξάγει τα δεδομένα του request από μορφή JSON αν αυτά είναι σε τέτοια μορφή και μια κλάση σφαλμάτων για Validation Errors.

```
"""Core API helpers."""

import json
from dataclasses import dataclass

from django.http import JsonResponse

@dataclass(frozen=True)
class ValidationError(Exception):
    message: str

def extract_payload(request):
    """Read JSON body first, then fallback to form payload."""
    if request.content_type and "application/json" in request.content_type:
        try:
            return json.loads(request.body.decode("utf-8") or "{}")
        except (ValueError, TypeError, UnicodeDecodeError):
            return {}
    return request.POST.dict()

def response_error(message: str, status: int = 400):
    return JsonResponse({"ok": False, "error": message}, status=status)
```

Accounts App

Μοντέλο Χρήστη (models.py)

Το μοντέλο χρήστη είναι ένα αφηρημένο μοντέλο από το οποίο κληρονομούν τις κοινές ιδιότητες οι υπόλοιποι χρήστες.

```
"""User model."""

from django.contrib.auth.models import AbstractUser
from django.db import models
from django.urls import reverse

from .managers import UserManager

class User(AbstractUser):
    username = None
    email = models.EmailField(unique=True)
    display_name = models.CharField(max_length=150, blank=True)

    objects = UserManager()

    USERNAME_FIELD = "email"
    REQUIRED_FIELDS = []

    def __str__(self):
        return self.get_display_name()

    def get_display_name(self) -> str:
        return self.display_name or self.get_full_name() or self.email

    def get_absolute_url(self):
        return reverse("admin:index")
```

Views Λογαριασμού Χρηστών (views.py)

Εδώ γίνεται η δημιουργία και αυθεντικοποίηση χρηστών

```
"""Account API views."""

from django.contrib.auth import authenticate, get_user_model, login, logout
from django.contrib.auth.password_validation import validate_password
from django.core.exceptions import ValidationError
from django.http import JsonResponse
from django.views.decorators.http import require_http_methods

from core.decorators import api_login_required
from core.rate_limits import RateLimitRule, enforce_rate_limits,
get_client_ip
from core.utils import extract_payload, response_error

User = get_user_model()

def _serialize_user(user: User):
    return {
        "id": user.id,
        "email": user.email,
        "first_name": user.first_name,
        "last_name": user.last_name,
```

```
        "display_name": user.display_name,
    }

@require_http_methods(["POST"])
def register(request):
    payload = extract_payload(request)
    email = payload.get("email", "").strip().lower()
    password = payload.get("password", "")

    limited = enforce_rate_limits(
        request,
        prefix="accounts:register",
        rules=[
            RateLimitRule(
                name="ip", limit=20, period_seconds=60 * 15,
key_func=get_client_ip
            ),
            RateLimitRule(
                name="email",
                limit=5,
                period_seconds=60 * 15,
                key_func=lambda r, email=email: email,
            ),
        ],
    )
    if limited:
        return limited

    if not email or not password:
        return response_error("email and password are required.")

    if User.objects.filter(email=email).exists():
        return response_error("Registration failed.", status=400)

    try:
        validate_password(password)
    except ValidationError as exc:
        return response_error(" ".join(exc.messages))

    user = User.objects.create_user(
        email=email,
        password=password,
        first_name=(payload.get("first_name", "") or "").strip(),
        last_name=(payload.get("last_name", "") or "").strip(),
        display_name=(payload.get("display_name", "") or "").strip(),
        is_active=True,
    )
    login(request, user)
    return JsonResponse({"ok": True, "user": _serialize_user(user)})

@require_http_methods(["POST"])
def login_view(request):
    payload = extract_payload(request)
    email = payload.get("email", "").strip().lower()
    password = payload.get("password", "")

    limited = enforce_rate_limits(
        request,
        prefix="accounts:login",
        rules=[
            RateLimitRule(
                name="ip", limit=60, period_seconds=60 * 10,
key_func=get_client_ip
            ),
        ],
    )
```

```
        RateLimitRule(
            name="email",
            limit=10,
            period_seconds=60 * 10,
            key_func=lambda r, email=email: email,
        ),
    ],
)
if limited:
    return limited

user = authenticate(request, username=email, password=password)
if not user:
    return response_error("Invalid credentials.", status=401)
login(request, user)
return JsonResponse({"ok": True, "user": _serialize_user(user)})

@require_http_methods(["POST"])
@api_login_required
def logout_view(request):
    logout(request)
    return JsonResponse({"ok": True})

@require_http_methods(["GET"])
@api_login_required
def me(request):
    return JsonResponse({"ok": True, "user": _serialize_user(request.user)})
```

Conferences App

Μοντέλα Συνεδρίων (models.py)

Εδώ περιγράφονται τα μοντέλα της βάσης για τα συνέδρια.

```
"""Conference domain models."""

from decimal import Decimal

from django.conf import settings
from django.db import models
from django.utils.text import slugify

from core.models import TimeStampedModel

class ConferenceStatus(models.TextChoices):
    DRAFT = "draft", "Draft"
    PUBLISHED = "published", "Published"
    ARCHIVED = "archived", "Archived"

class ConferenceMembershipRole(models.TextChoices):
    ORGANIZER = "organizer", "Organizer"
    REVIEWER = "reviewer", "Reviewer"
    OBSERVER = "observer", "Observer"
    PRESENTER = "presenter", "Presenter"

class Conference(TimeStampedModel):
    title = models.CharField(max_length=255)
```

```
slug = models.SlugField(max_length=255, unique=True)
description = models.TextField(blank=True)
venue = models.CharField(max_length=255, blank=True)
status = models.CharField(
    max_length=20,
    choices=ConferenceStatus.choices,
    default=ConferenceStatus.DRAFT,
)
registration_start = models.DateTimeField()
registration_end = models.DateTimeField()
start_date = models.DateTimeField()
end_date = models.DateTimeField()
max_tracks = models.PositiveSmallIntegerField(default=5)
submission_fee = models.DecimalField(
    max_digits=10, decimal_places=2, default=Decimal("0.00")
)
currency = models.CharField(max_length=3, default="EUR")
organizer = models.ForeignKey(
    settings.AUTH_USER_MODEL,
    on_delete=models.PROTECT,
    related_name="organizing_conferences",
)

class Meta:
    ordering = ["start_date"]

def __str__(self):
    return self.title

def save(self, *args, **kwargs):
    if not self.slug:
        self.slug = slugify(self.title)[:255]
    super().save(*args, **kwargs)

class Track(TimeStampedModel):
    conference = models.ForeignKey(
        Conference, on_delete=models.CASCADE, related_name="tracks"
    )
    name = models.CharField(max_length=120)
    description = models.TextField(blank=True)
    capacity = models.PositiveIntegerField(default=0)

    class Meta:
        unique_together = ("conference", "name")
        ordering = ["conference__start_date", "name"]

    def __str__(self):
        return f"{self.conference.title} - {self.name}"

class ConferenceMembership(TimeStampedModel):
    conference = models.ForeignKey(
        Conference, on_delete=models.CASCADE, related_name="memberships"
    )
    user = models.ForeignKey(
        settings.AUTH_USER_MODEL,
        on_delete=models.CASCADE,
        related_name="conference_roles",
    )
    role = models.CharField(max_length=20,
choices=ConferenceMembershipRole.choices)
    is_active = models.BooleanField(default=True)
    notes = models.TextField(blank=True)

    class Meta:
```

```
constraints = [
    models.UniqueConstraint(
        fields=["conference", "user", "role"],
        name="unique_role_per_user_per_conference",
    )
]
indexes = [
    models.Index(fields=["conference", "role"]),
    models.Index(fields=["conference", "user"]),
]

def __str__(self):
    return f"{self.user} - {self.conference} ({self.role})"

class WatchList(TimeStampedModel):
    user = models.ForeignKey(
        settings.AUTH_USER_MODEL,
        on_delete=models.CASCADE,
        related_name="watchlist_items",
    )
    conference = models.ForeignKey(
        Conference, on_delete=models.CASCADE, related_name="watchers"
    )

    class Meta:
        unique_together = ("user", "conference")

    def __str__(self):
        return f"{self.user} watches {self.conference}"
```

Conferences URLs (urls.py)

Το urls.py περιέχει τα URL της εφαρμογής που σχετίζονται με τη διαχείριση των συνεδρίων.

```
"""Conference routes."""

from django.urls import path

from . import views

urlpatterns = [
    path("", views.list_conferences, name="conference-index"),
    path("create/", views.create_conference, name="conference-create"),
    path("<slug:slug>/", views.conference_detail, name="conference-detail"),
    path("<slug:slug>/watchlist/", views.watchlist_toggle, name="conference-watchlist"),
    path("<slug:slug>/role/", views.add_role, name="conference-role"),
    path("<slug:slug>/tracks/", views.set_tracks, name="conference-tracks"),
    path("<slug:slug>/status/", views.update_status, name="conference-status"),
    path("<slug:slug>/observe/", views.register_as_observer,
        name="conference-observe"),
]
```

Conferences Admin models (admin.py)

Το admin.py περιέχει τα μοντέλα διαχείρισης συνεδρίων για τον Admin.

```
"""Admin registrations for conferences."""

from django.contrib import admin

from .models import Conference, ConferenceMembership, Track, WatchList

@admin.register(Conference)
class ConferenceAdmin(admin.ModelAdmin):
    list_display = ("title", "status", "start_date", "end_date", "organizer")
    list_filter = ("status",)
    search_fields = ("title", "slug", "description")
    prepopulated_fields = {"slug": ("title",)}

@admin.register(Track)
class TrackAdmin(admin.ModelAdmin):
    list_display = ("conference", "name", "capacity")
    list_filter = ("conference",)
    search_fields = ("conference__title", "name")

@admin.register(ConferenceMembership)
class ConferenceMembershipAdmin(admin.ModelAdmin):
    list_display = ("conference", "user", "role", "is_active")
    list_filter = ("conference", "role", "is_active")
    search_fields = ("user__email", "conference__title")

@admin.register(WatchList)
class WatchListAdmin(admin.ModelAdmin):
    list_display = ("user", "conference")
```

Submissions App

Services Υποβολής (services.py)

Αυτό το module περιέχει τις μεθόδους υποβολής των paper.

```
"""Workflow helpers for submissions."""

from decimal import Decimal

from conferences.models import Conference, Track
from core.services import atomic_update
from notifications.services import create_notification

from .models import PaperSubmission, SubmissionStatus

def create_submission(
    presenter,
    conference: Conference,
    *,
    title: str,
    abstract: str,
    keywords: str = "",
    paper_file=None,
```

```
) -> PaperSubmission:
    paper = PaperSubmission.objects.create(
        conference=conference,
        presenter=presenter,
        title=title,
        abstract=abstract,
        keywords=keywords,
        paper_file=paper_file,
        status=SubmissionStatus.SUBMITTED,
    )
    create_notification(
        recipient=presenter,
        title="Submission received",
        message=f"Your paper '{paper.title}' was submitted to
{conference.title}.",
        category="info",
        target_object=paper,
    )
    return paper

def assign_track(paper: PaperSubmission, track: Track) -> PaperSubmission:
    """Assign a track and move the submission forward."""
    if track.conference_id != paper.conference_id:
        raise ValueError("Track and paper must belong to same conference.")
    if track.capacity and track.capacity > 0:
        current = PaperSubmission.objects.filter(track=track).count()
        if current >= track.capacity:
            raise ValueError("Track has reached max capacity.")

    next_status = SubmissionStatus.TRACK_ASSIGNED
    if paper.fee <= Decimal("0"):
        next_status = SubmissionStatus.PRESENTATION_READY

    paper = atomic_update(paper, track=track, status=next_status)
    if next_status == SubmissionStatus.PRESENTATION_READY:
        create_notification(
            recipient=paper.presenter,
            title="Track assigned",
            message=f"Your paper '{paper.title}' was assigned to track
'{track.name}'. No payment is required.",
            category="track",
            target_object=paper,
            send_email=True,
        )
        return paper

    create_notification(
        recipient=paper.presenter,
        title="Track assigned",
        message=(
            f"Your paper '{paper.title}' was assigned to track
'{track.name}'. "
            f"Payment is required to finalize your presentation."
        ),
        category="track",
        target_object=paper,
        send_email=True,
    )
    return paper

def mark_payment_not_required(paper: PaperSubmission) -> PaperSubmission:
    return atomic_update(paper, status=SubmissionStatus.PRESENTATION_READY)
```

```
def reset_to_draft(paper: PaperSubmission) -> PaperSubmission:  
    return atomic_update(paper, status=SubmissionStatus.DRAFT, track=None)
```

Reviews App

Λογική Αξιολόγησης (services.py)

Αυτό το module περιέχει τη λογική της αξιολόγησης των paper από τους reviewers.

```
"""Submission review logic."""  
  
from collections import Counter  
  
from conferences.models import ConferenceMembership, ConferenceMembershipRole  
from notifications.services import create_notification  
from submissions.models import SubmissionStatus  
  
from .models import PaperReview, ReviewDecision  
  
FINAL_REVIEW_STATUSES = {  
    SubmissionStatus.ACCEPTED,  
    SubmissionStatus.REJECTED,  
    SubmissionStatus.TRACK_ASSIGNED,  
    SubmissionStatus.PAYMENT_PENDING,  
    SubmissionStatus.PRESENTATION_READY,  
}  
  
def _accepted_reviews(paper):  
    return list(  
        PaperReview.objects.filter(  
            paper=paper,  
            decision__in=[  
                ReviewDecision.ACCEPT,  
                ReviewDecision.REJECT,  
                ReviewDecision.REQUEST_CHANGES,  
            ],  
        ).order_by("created_at")  
    )  
  
def submit_review(*, paper, reviewer, decision: str, score=None, comments:  
str = ""):  
    """Create or update a review and re-evaluate consensus."""  
    review, created = PaperReview.objects.update_or_create(  
        paper=paper,  
        reviewer=reviewer,  
        defaults={  
            "decision": decision,  
            "score": score,  
            "comments": comments,  
        },  
    )  
  
    # Once at least one reviewer starts, status is in review loop.  
    if paper.status == SubmissionStatus.SUBMITTED:  
        paper.status = SubmissionStatus.UNDER_REVIEW  
        paper.save(update_fields=["status"])  
  
    evaluate_submission_review_status(paper)  
  
    create_notification(
```

```
        recipient=reviewer,
        title="Review submitted",
        message=f"Your review for '{paper.title}' was saved.",
        category="review",
        target_object=paper,
    )

    return review

def _notify_organizers(paper, *, title: str, message: str, category: str =
"review"):
    conference = paper.conference
    organizer_ids = set(
        ConferenceMembership.objects.filter(
            conference=conference,
            role=ConferenceMembershipRole.ORGANIZER,
            is_active=True,
        ).values_list("user_id", flat=True)
    )
    if getattr(conference, "organizer_id", None):
        organizer_ids.add(conference.organizer_id)

    for user_id in organizer_ids:
        create_notification(
            recipient_id=user_id,
            title=title,
            message=message,
            category=category,
            target_object=paper,
        )

def evaluate_submission_review_status(paper):
    """Evaluate consensus after each review action."""
    if paper.status in FINAL_REVIEW_STATUSES:
        return paper

    decisions = [review.decision for review in _accepted_reviews(paper)]
    if len(decisions) < 2:
        return paper

    accept_count = decisions.count(ReviewDecision.ACCEPT)
    reject_count = decisions.count(ReviewDecision.REJECT) + decisions.count(
        ReviewDecision.REQUEST_CHANGES
    )

    new_status = paper.status
    if len(decisions) >= 3:
        new_status = (
            SubmissionStatus.ACCEPTED
            if accept_count > reject_count
            else SubmissionStatus.REJECTED
        )
    elif accept_count >= 2:
        new_status = SubmissionStatus.ACCEPTED
    elif reject_count >= 2:
        new_status = SubmissionStatus.REJECTED
    else:
        new_status = SubmissionStatus.NEEDS_THIRD_REVIEW

    if new_status == paper.status:
        return paper

    paper.status = new_status
    paper.save(update_fields=["status"])
```

```
if new_status == SubmissionStatus.NEEDS_THIRD_REVIEW:
    _notify_organizers(
        paper,
        title="Third review required",
        message=f"Paper '{paper.title}' needs a third reviewer due to
disagreement.",
    )
    return paper

if new_status == SubmissionStatus.ACCEPTED:
    create_notification(
        recipient=paper.presenter,
        title="Paper accepted",
        message=f"Your paper '{paper.title}' was accepted. Await track
assignment.",
        category="review",
        target_object=paper,
    )
    _notify_organizers(
        paper,
        title="Paper accepted",
        message=f"Paper '{paper.title}' was accepted by reviewers.",
    )
    return paper

if new_status == SubmissionStatus.REJECTED:
    create_notification(
        recipient=paper.presenter,
        title="Paper rejected",
        message=f"Your paper '{paper.title}' was rejected.",
        category="review",
        target_object=paper,
    )
    _notify_organizers(
        paper,
        title="Paper rejected",
        message=f"Paper '{paper.title}' was rejected by reviewers.",
    )
    return paper

return paper

def get_status_snapshot(paper) -> dict:
    """Return the latest aggregated review status of a submission."""
    decisions = [review.decision for review in _accepted_reviews(paper)]
    counter = Counter(decisions)
    return {
        "total": len(decisions),
        "accept_count": counter[ReviewDecision.ACCEPT],
        "reject_count": counter[ReviewDecision.REJECT]
+ counter[ReviewDecision.REQUEST_CHANGES],
        "latest": decisions[-1] if decisions else None,
    }
```

Payments App

Διαχείριση Πληρωμών (views.py)

Αυτό το module περιέχει τη γενική λογική πληρωμών.

```
"""Payment API views + provider webhooks + UI return endpoints."""

from __future__ import annotations

from django.conf import settings
from django.contrib import messages
from django.contrib.auth.decorators import login_required
from django.http import HttpResponseForbidden, JsonResponse
from django.shortcuts import redirect
from django.urls import reverse
from django.views.decorators.csrf import csrf_exempt
from django.views.decorators.http import require_http_methods

from conferences.models import ConferenceMembershipRole
from core.audit import log_event
from core.decorators import api_login_required
from core.permissions import user_has_role
from core.utils import extract_payload, response_error
from submissions.models import PaperSubmission, SubmissionStatus

from .models import Payment, PaymentStatus
from .services import (
    create_payment_request,
    handle_stripe_event,
    mark_payment_as_failed,
    mark_payment_as_succeeded,
    mark_webhook_event_failed,
    mark_webhook_event_ignored,
    mark_webhook_event_processed,
    record_webhook_event,
    start_checkout_for_payment,
)
from .stripe_gateway import construct_event

@require_http_methods(["POST"])
@api_login_required
def request_payment(request, paper_id: int):
    paper = (
        PaperSubmission.objects.filter(pk=paper_id)
        .select_related("presenter", "conference")
        .first()
    )
    if not paper:
        return response_error("paper not found.", status=404)
    if paper.presenter_id != request.user.id and not user_has_role(
        request.user, paper.conference, ConferenceMembershipRole.ORGANIZER
    ):
        return response_error("permission denied.", status=403)
    if paper.status == SubmissionStatus.PRESENTATION_READY:
        return response_error("payment already completed for this
submission.")
    if paper.status not in {
        SubmissionStatus.TRACK_ASSIGNED,
        SubmissionStatus.PAYMENT_PENDING,
    }:
        return response_error("payment can be requested only after track
assignment.")

    existing_payment = None
    if hasattr(paper, "payment"):
        existing_payment: Payment = getattr(paper, "payment")
        if existing_payment.status == PaymentStatus.SUCCEEDED:
            return response_error("payment already completed.")
```

```

        if existing_payment.status in {PaymentStatus.PENDING,
PaymentStatus.PROCESSING}:
            payload = {
                "ok": True,
                "provider": existing_payment.provider,
                "payment_id": existing_payment.id,
                "amount": str(existing_payment.amount),
                "currency": existing_payment.currency,
                "status": existing_payment.status,
            }
            if existing_payment.provider == "stripe":
                checkout_url = (existing_payment.provider_metadata or
{}).get(
                    "checkout_url", ""
                )
                if not checkout_url:
                    success_url =
request.build_absolute_uri(reverse("payment-success"))
                    cancel_url = request.build_absolute_uri(reverse("payment-
cancel"))
                    success_url =
f"{success_url}?payment_id={existing_payment.id}&session_id={{CHECKOUT_SESSIO
N_ID}}"
                    cancel_url =
f"{cancel_url}?payment_id={existing_payment.id}"
                    checkout_url = start_checkout_for_payment(
                        payment=existing_payment,
                        success_url=success_url,
                        cancel_url=cancel_url,
                    )
                    payload["checkout_url"] = checkout_url
                return JsonResponse(payload)
            if existing_payment.status in {PaymentStatus.FAILED,
PaymentStatus.CANCELED}:
                existing_payment.status = PaymentStatus.PENDING
                existing_payment.failure_reason = ""
                existing_payment.provider_payment_id = None
                existing_payment.provider_metadata = {}
                existing_payment.save(
                    update_fields=[
                        "status",
                        "failure_reason",
                        "provider_payment_id",
                        "provider_metadata",
                    ]
                )
                log_event(
                    action="payment.retry",
                    actor=request.user,
                    target=existing_payment,
                    request=request,
                    metadata={"paper_id": paper.id},
                )
                return JsonResponse(
                    {
                        "ok": True,
                        "provider": existing_payment.provider,
                        "payment_id": existing_payment.id,
                        "amount": str(existing_payment.amount),
                        "currency": existing_payment.currency,
                        "status": existing_payment.status,
                    }
                )

        payment = create_payment_request(paper, payer=paper.presenter)
        if paper.status != SubmissionStatus.PAYMENT_PENDING:

```

```

        paper.status = SubmissionStatus.PAYMENT_PENDING
        paper.save(update_fields=["status"])
    log_event(
        action="payment.request",
        actor=request.user,
        target=payment,
        request=request,
        metadata={
            "paper_id": paper.id,
            "amount": str(payment.amount),
            "currency": payment.currency,
        },
    )
    response_payload = {
        "ok": True,
        "provider": payment.provider,
        "payment_id": payment.id,
        "amount": str(payment.amount),
        "currency": payment.currency,
    }
    if payment.provider == "stripe":
        success_url = request.build_absolute_uri(reverse("payment-success"))
        cancel_url = request.build_absolute_uri(reverse("payment-cancel"))
        success_url = (
            f"{success_url}?payment_id={payment.id}&session_id={{CHECKOUT_SESSION_ID}}"
        )
        cancel_url = f"{cancel_url}?payment_id={payment.id}"
        checkout_url = start_checkout_for_payment(
            payment=payment, success_url=success_url, cancel_url=cancel_url
        )
        response_payload["checkout_url"] = checkout_url
    return JsonResponse(response_payload)

@require_http_methods(["POST"])
def webhook_fake(request):
    if (
        getattr(settings, "PAYMENTS_PROVIDER", "sandbox") or "sandbox"
    ) == "stripe" and not settings.DEBUG:
        return response_error("not found.", status=404)

    expected = (getattr(settings, "PAYMENT_WEBHOOK_SECRET", "") or
        "").strip()
    if not expected and not settings.DEBUG:
        return response_error("webhook not configured.", status=503)

    provided = (request.headers.get("X-Webhook-Secret", "") or "").strip()
    if expected and provided != expected:
        return response_error("unauthorized.", status=401)

    payload = extract_payload(request)
    payment_id = payload.get("payment_id")
    status = payload.get("status")
    reason = payload.get("reason", "")
    provider_id = payload.get("provider_payment_id", "")

    payment = (
        Payment.objects.select_related("paper",
        "payer").filter(pk=payment_id).first()
    )
    if not payment:
        return response_error("payment not found.", status=404)

    if payment.status == PaymentStatus.SUCCEEDED:
        log_event(

```

```
        action="payment.webhook",
        actor=None,
        target=payment,
        request=request,
        metadata={"incoming_status": status, "ignored": True},
    )
    return JsonResponse({"ok": True, "status": payment.status})

    if status == PaymentStatus.SUCCEEDED:
        payment = mark_payment_as_succeeded(payment,
        provider_payment_id=provider_id)
    elif status == PaymentStatus.FAILED:
        payment = mark_payment_as_failed(payment, reason=reason)
    else:
        return response_error("unsupported status.")

    log_event(
        action="payment.webhook",
        actor=None,
        target=payment,
        request=request,
        metadata={
            "incoming_status": status,
            "result_status": payment.status,
            "provider_payment_id": provider_id,
        },
    )
    return JsonResponse({"ok": True, "status": payment.status})

@csrf_exempt
@require_http_methods(["POST"])
def webhook_stripe(request):
    if (getattr(settings, "PAYMENTS_PROVIDER", "sandbox") or "sandbox") !=
    "stripe":
        return response_error("not found.", status=404)

    signature = (request.headers.get("Stripe-Signature", "") or "").strip()
    if not signature:
        return response_error("missing signature.", status=400)

    try:
        event = construct_event(payload=request.body,
        signature_header=signature)
    except Exception:
        return response_error("invalid webhook signature.", status=400)

    event_id = event.get("id", "")
    event_type = event.get("type", "")
    if not event_id:
        return response_error("invalid event.", status=400)

    row, created = record_webhook_event(
        provider="stripe", event_id=event_id, event_type=event_type
    )
    if not created:
        return JsonResponse({"ok": True, "status": "duplicate"})

    try:
        outcome = handle_stripe_event(event)
        if outcome.startswith("processed:"):
            mark_webhook_event_processed(row)
        elif outcome.startswith("ignored:"):
            mark_webhook_event_ignored(row, reason=outcome)
        else:
            mark_webhook_event_ignored(row, reason="unknown")
```

```
except Exception as exc:
    mark_webhook_event_failed(row, error=str(exc))
    return response_error("webhook processing failed.", status=500)

log_event(
    action="payment.webhook",
    actor=None,
    target=row,
    request=request,
    metadata={"provider": "stripe", "event_id": event_id, "event_type":
event_type},
)
return JsonResponse({"ok": True, "status": outcome})

@login_required
def payment_success(request):
    payment_id = (request.GET.get("payment_id") or "").strip()
    if not payment_id.isdigit():
        messages.info(request, "Η πληρωμή καταχωρήθηκε. Επιβεβαίωση σε
εξέλιξη.")
        return redirect("dashboard")

    payment = (
        Payment.objects.select_related("paper__conference")
        .filter(pk=int(payment_id))
        .first()
    )
    if not payment:
        messages.info(request, "Η πληρωμή καταχωρήθηκε. Επιβεβαίωση σε
εξέλιξη.")
        return redirect("dashboard")

    paper = payment.paper
    if paper.presenter_id != request.user.id and not user_has_role(
        request.user, paper.conference, ConferenceMembershipRole.ORGANIZER
    ):
        return HttpResponseForbidden("Forbidden")

    messages.success(
        request,
        "Η πληρωμή καταχωρήθηκε. Αν δεν εμφανιστεί ως ολοκληρωμένη, ανανέωσε
σε λίγο.",
    )
    return redirect("ui-conference-detail", slug=paper.conference.slug)

@login_required
def payment_cancel(request):
    payment_id = (request.GET.get("payment_id") or "").strip()
    redirect_to = "dashboard"
    redirect_kwargs = {}

    if payment_id.isdigit():
        payment = (
            Payment.objects.select_related("paper__conference")
            .filter(pk=int(payment_id))
            .first()
        )
        if payment:
            paper = payment.paper
            allowed = paper.presenter_id == request.user.id or user_has_role(
                request.user, paper.conference,
                ConferenceMembershipRole.ORGANIZER
            )
            if not allowed:
```

```
return HttpResponseForbidden("Forbidden")

if payment.status != PaymentStatus.SUCCEEDED:
    payment.status = PaymentStatus.CANCELED
    payment.save(update_fields=["status"])

redirect_to = "ui-conference-detail"
redirect_kwargs = {"slug": paper.conference.slug}

messages.error(request, "Η πληρωμή ακυρώθηκε.")
return redirect(redirect_to, **redirect_kwargs)
```

Εργαλεία για πληρωμές Stripe(stripe_gateway.py)

Αυτό το module περιέχει τη λογική πληρωμών μέσω Stripe.

```
"""Stripe gateway wrapper (Checkout + webhook verification).

This module is intentionally small and isolated to keep Stripe-specific logic
out of the HTTP layer and to keep it easy to mock in unit tests.
"""

from __future__ import annotations
from decimal import Decimal

import stripe
from django.conf import settings

def _require_stripe_enabled() -> None:
    if (getattr(settings, "PAYMENTS_PROVIDER", "sandbox") or "sandbox") !=
    "stripe":
        raise RuntimeError("Stripe provider is not enabled.")

    if not getattr(settings, "STRIPE_SECRET_KEY", ""):
        raise RuntimeError("STRIPE_SECRET_KEY is not configured.")

def create_checkout_session(
    *,
    payment_id: int,
    amount: Decimal,
    currency: str,
    title: str,
    customer_email: str,
    success_url: str,
    cancel_url: str,
) -> dict:
    _require_stripe_enabled()

    stripe.api_key = settings.STRIPE_SECRET_KEY

    cents = int((amount * 100).to_integral_value())
    currency = (currency or "EUR").lower()

    session = stripe.checkout.Session.create(
        mode="payment",
        customer_email=customer_email,
        success_url=success_url,
        cancel_url=cancel_url,
        metadata={"payment_id": str(payment_id)},
        line_items=[
```

```
{
    "price_data": {
        "currency": currency,
        "product_data": {"name": title},
        "unit_amount": cents,
    },
    "quantity": 1,
}

],
)

# Stripe returns a StripeObject (dict-like); we normalize to a plain dict
so
# call sites remain test-friendly.
return {"id": session["id"], "url": session.get("url", "")}

def construct_event(*, payload: bytes, signature_header: str) -> dict:
    _require_stripe_enabled()
    secret = getattr(settings, "STRIPE_WEBHOOK_SECRET", "")
    if not secret:
        raise RuntimeError("STRIPE_WEBHOOK_SECRET is not configured.")

    stripe.api_key = settings.STRIPE_SECRET_KEY
    return stripe.Webhook.construct_event(
        payload=payload,
        sig_header=signature_header,
        secret=secret,
    )
```

Notifications App

Υπηρεσία Ειδοποιήσεων (services.py)

Αυτό το module περιέχει τις μεθόδους για τη δημιουργία ειδοποιήσεων για τους χρήστες.

```
"""Notification dispatch helpers."""

import logging

from django.conf import settings
from django.contrib.contenttypes.models import ContentType
from django.utils import timezone

from core.email import send_notification_email

from .models import Notification

logger = logging.getLogger(__name__)

def create_notification(
    recipient=None,
    *,
    recipient_id: int | None = None,
    title: str,
    message: str,
    category: str = "info",
    target_object=None,
    send_email: bool = False,
    email_cta_url: str | None = None,
    email_cta_label: str | None = None,
```

```
) :
    if recipient is None and recipient_id is None:
        raise ValueError("recipient or recipient_id is required.")
    if recipient is not None and recipient_id is not None:
        raise ValueError("only one of recipient or recipient_id can be
provided.")

    content_type = None
    object_id = None
    if target_object is not None:
        content_type = ContentType.objects.get_for_model(target_object)
        object_id = target_object.pk

    base_fields = (
        {"recipient": recipient}
        if recipient is not None
        else {"recipient_id": recipient_id}
    )
    notification = Notification.objects.create(
        title=title,
        message=message,
        category=category,
        target_content_type=content_type,
        target_object_id=object_id,
        **base_fields,
    )
    if send_email and bool(getattr(settings, "EMAIL_NOTIFICATIONS_ENABLED",
False)):
        try:
            to_email = getattr(notification.recipient, "email", "")
            send_notification_email(
                to_email=to_email,
                title=title,
                message=message,
                cta_url=email_cta_url,
                cta_label=email_cta_label,
            )
        except Exception:
            logger.exception(
                "Failed to send notification email (notification_id=%s)",
                notification.id,
            )
    return notification

def mark_all_read(recipient):
    now = timezone.now()
    return Notification.objects.filter(recipient=recipient,
is_read=False).update(
        is_read=True, read_at=now
    )

def unread_count(recipient):
    return Notification.objects.filter(recipient=recipient,
is_read=False).count()
```

Frontend

Conference Detail Template(conference_detail.html)

Εδώ υπάρχει το frontend Template με τις λεπτομερείς πληροφορίες συνεδρίου.

```
{% extends "core/base.html" %}

{% block title %}{{ conference.title }} · Conference Hub{% endblock %}

{% block content %}
<input id="conference-id" type="hidden" value="{{ conference.id }}" />
<input id="conference-slug" type="hidden" value="{{ conference.slug }}" />

<section class="conference-hero reveal">
  <div>
    <p class="eyebrow">{{ conference.status|capfirst }}</p>
    <h1>{{ conference.title }}</h1>
    <p>{{ conference.description }}</p>
    {% if submission_status_summary %}
      <div class="chip-row">
        {% for item in submission_status_summary %}
          <span class="chip">{{ item.status }}: {{ item.total }}</span>
        {% endfor %}
      </div>
    {% endif %}
  </div>
  <div class="chip-row">
    {% if user.is_authenticated and conference.status == 'published' %}
      <form method="post" action="{% url 'ui-conference-watch'
conference.slug %}">
        {% csrf_token %}
        <button class="solid-btn" type="submit">
          {% if is_watching %}Κατάργηση από Watchlist{% else %}Προσθήκη
στην Watchlist{% endif %}
        </button>
      </form>
    {% endif %}
    <span class="chip">{{ conference.start_date|date:"d/m/Y H:i" }}</span>
    <span class="chip">{{ conference.end_date|date:"d/m/Y H:i" }}</span>
    <span class="chip">Κόστος: {{ conference.submission_fee }} {{
conference.currency }}</span>
    <span class="chip">{{ tracks_count }} Tracks</span>
    {% if is_registration_open %}
      <span class="chip chip-ok">Ανοικτές υποβολές</span>
    {% else %}
      <span class="chip chip-warn">Κλειδωμένες υποβολές</span>
    {% endif %}
  </div>
</section>

<section class="split-grid">
  <article class="card reveal">
    <h3>Tracks</h3>
    <div class="track-grid">
      {% for t in tracks %}
        <div class="track-item">
          <h4>{{ t.name }}</h4>
          <p>{{ t.description|default:"-" }}</p>
          <span class="chip">Capacity {{ t.capacity }}</span>
        </div>
      {% empty %}
        <p>Δεν έχουν οριστεί tracks.</p>
      {% endfor %}
    </div>
    {% if flags.is_organizer %}
      <div class="data-table-wrap section-spacer">
        <h4>Role-aware ρόλοι</h4>
        <p class="small">
          Ως οργανωτής μπορείς να ενημερώσεις ρόλους με email και να ορίσεις
```

```

τα tracks με το actions panel.
    </p>
  </div>
  {% elif flags.is_reviewer %}
  <p class="small">Ως reviewer βλέπεις αιτήματα review για αυτό το
συνέδριο.</p>
  {% elif is_authenticated %}
  <p class="small">Ως συμμετέχων βλέπεις μόνο τις δικές σου υποβολές.</p>
  {% else %}
  <p class="small">Δεν είσαι συνδεδεμένος για πρόσβαση σε προσωπικά
δεδομένα.</p>
  {% endif %}
</article>

<article class="card reveal">
  <h3>Actions</h3>

  {% if flags.is_organizer %}
  <form class="card form-card" id="assign-role-form">
    <h4>Ανάθεση ρόλου</h4>
    <label for="role-email">Email χρήστη</label>
    <input id="role-email" type="email" placeholder="user@domain.com"
/>

    <label for="role-select">Ρόλος</label>
    <select id="role-select">
      <option value="organizer">Organizer</option>
      <option value="reviewer">Reviewer</option>
      <option value="observer">Observer</option>
      <option value="presenter">Presenter</option>
    </select>
    <button class="solid-btn" id="assign-role-btn"
type="button">Ανάθεση</button>
    <small id="assign-role-message" aria-live="polite"></small>
  </form>

  <form class="card form-card" id="set-tracks-form">
    <h4>Ορισμός tracks</h4>
    <label for="set-tracks-text">Όνομα tracks (διαχωρισμός με
κόμμα)</label>
    <textarea id="set-tracks-text" rows="3" placeholder="AI Track,
Security Track, Cloud Track"></textarea>
    <button class="solid-btn" id="set-tracks-btn"
type="button">Αποθήκευση tracks</button>
    <small id="set-tracks-message" aria-live="polite"></small>
  </form>
  {% elif is_authenticated and can_submit %}
  <form class="card form-card" id="submit-paper-form">
    <h4>Νέα υποβολή paper</h4>
    <label for="paper-title">Τίτλος</label>
    <input id="paper-title" type="text" placeholder="Τίτλος paper" />
    <label for="paper-abstract">Abstract</label>
    <textarea id="paper-abstract" rows="4" placeholder="Περίληψη έως 2-
3 παραγράφους"></textarea>
    <label for="paper-keywords">Λέξεις-κλειδιά</label>
    <input id="paper-keywords" type="text" placeholder="machine
learning, ai, vision" />
    <label for="paper-file">PDF (προαιρετικό)</label>
    <input id="paper-file" type="file" accept="application/pdf" />
    <button class="solid-btn" id="paper-submit-btn"
type="button">Καταχώρηση paper</button>
    <small id="paper-submit-message" aria-live="polite"></small>
  </form>

  {% if conference.status == 'published' and not flags.is_observer %}
  <form class="card form-card" id="observer-register-form">
    <h4>Εγγραφή ως παρατηρητής</h4>

```

```

        <p class="small">Εγγράψου ως observer για να παρακολουθείς το
        συνέδριο.</p>
        <button class="ghost-btn" id="observer-register-btn"
        type="button">Εγγραφή Observer</button>
        <small id="observer-register-message" aria-live="polite"></small>
    </form>
    {% endif %}
    {% elif is_authenticated %}
    <article class="card form-card">
    <h4>Υποβολή paper</h4>
    <p class="small">Η περίοδος υποβολών για αυτό το συνέδριο είναι
    κλειστή.</p>
    </article>

    {% if conference.status == 'published' and not flags.is_observer %}
    <form class="card form-card" id="observer-register-form">
    <h4>Εγγραφή ως παρατηρητής</h4>
    <p class="small">Εγγράψου ως observer για να παρακολουθείς το
    συνέδριο.</p>
    <button class="ghost-btn" id="observer-register-btn"
    type="button">Εγγραφή Observer</button>
    <small id="observer-register-message" aria-live="polite"></small>
    </form>
    {% endif %}
    {% else %}
    <p>Συνδεθείτε για να κάνετε υποβολή ή να αναλάβετε ρόλο δράσης.</p>
    {% endif %}
    </article>
</section>

{% if flags.is_organizer %}
<section class="card reveal">
<div class="card-toolbar">
<h3>Μέλη & ρόλοι</h3>
<p class="small">Διαχείριση ρόλων ανά χρήστη. Η αφαίρεση απενεργοποιεί
μόνο τον συγκεκριμένο ρόλο.</p>
</div>
<small id="role-directory-message" class="small"></small>
<div class="data-table-wrap">
<table>
<thead>
<tr>
<th>Χρήστης</th>
<th>Ρόλοι</th>
</tr>
</thead>
<tbody>
{% for m in membership_directory %}
<tr>
<td>
<strong>{{ m.name }}</strong><br />
<span class="small">{{ m.email }}</span>
</td>
<td>
<div class="chip-row">
{% for r in m.roles %}
<span class="chip role-chip">
{{ r.label }}
<button
class="role-remove js-remove-role"
type="button"
data-email="{{ m.email }}"
data-role="{{ r.value }}"
title="Αφαίρεση ρόλου"
aria-label="Αφαίρεση ρόλου {{ r.label }} από {{
m.email }}"

```

```

        >x</button>
    </span>
    {% endfor %}
</div>
</td>
</tr>
{% empty %}
<tr><td colspan="2">Δεν υπάρχουν ενεργά memberships.</td></tr>
{% endfor %}
</tbody>
</table>
</div>
</section>
{% endif %}

<section class="card reveal">
    <h3>Πρόγραμμα παρουσιάσεων</h3>
    <p class="small">Δημόσια ορατές παρουσιάσεις (papers που έχουν
οριστικοποιηθεί σε track).</p>
    <div class="data-table-wrap">
        <table>
            <thead>
                <tr>
                    <th>Paper</th>
                    <th>Track</th>
                    <th>Παρουσιαστής</th>
                </tr>
            </thead>
            <tbody>
                {% for p in public_program %}
                <tr>
                    <td>{{ p.title }}</td>
                    <td>{{ p.track.name }}</td>
                    <td>{{
p.presenter.get_full_name|default:p.presenter.display_name|default:"Presenter
" }}</td>
                </tr>
                {% empty %}
                <tr><td colspan="3">Δεν υπάρχουν ακόμη οριστικοποιημένες
παρουσιάσεις.</td></tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
</section>

{% if flags.is_reviewer %}
<section class="card reveal">
    <h3>Εκκρεμή Review Queue</h3>
    <div class="data-table-wrap">
        <table>
            <thead>
                <tr>
                    <th>Paper</th>
                    <th>Παρουσιαστής</th>
                    <th>Στάδιο</th>
                    <th>Αξιολόγηση</th>
                </tr>
            </thead>
            <tbody>
                {% for p in reviewer_queue %}
                <tr>
                    <td>
                        {{ p.title }}
                        {% if p.paper_file %}
                        <br /><a class="text-link" href="{% url 'submission-file'

```

```

p.id %}">Λήψη PDF</a>
    {% endif %}
    </td>
    <td>{{ p.presenter.get_full_name|default:p.presenter.email
}}</td>
    <td><span class="chip">{{ p.get_status_display }}</span></td>
    <td>
        <form class="js-review-form" data-paper-id="{{ p.id }}">
            <select class="review-decision">
                <option value="accept">Προτείνω αποδοχή</option>
                <option value="reject">Απόρριψη</option>
                <option value="request_changes">Αλλαγές</option>
            </select>
            <input class="review-score" type="number" min="0" max="10"
value="7" />
            <textarea class="review-comment" rows="2"
placeholder="Σχόλια"></textarea>
            <button class="solid-btn js-review-submit"
type="button">Υποβολή</button>
            <small class="review-message" data-paper-id="{{ p.id
}}"></small>
        </form>
    </td>
</tr>
{% empty %}
<tr>
    <td colspan="4">Δεν υπάρχουν εργασίες για αξιολόγηση.</td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
</section>
{% endif %}

{% if flags.is_organizer and review_assignment_targets %}
<section class="card reveal">
    <h3>Ανάθεση reviewers σε υποβολές</h3>
    <p class="small">
        Ορίζεις 2 reviewers. Αν υπάρξει διαφωνία, το σύστημα θα ζητήσει 3ο
reviewer για τελική απόφαση.
    </p>

    <datalist id="reviewer-directory">
        {% for m in reviewer_directory %}
            <option value="{{ m.user.email }}">{{ m.user.get_display_name
}}</option>
        {% endfor %}
    </datalist>

    <div class="data-table-wrap">
        <table>
            <thead>
                <tr>
                    <th>Paper</th>
                    <th>Presenter</th>
                    <th>Στάδιο</th>
                    <th>Τρέχουσες αναθέσεις</th>
                    <th>Ανάθεση</th>
                </tr>
            </thead>
            <tbody>
                {% for p in review_assignment_targets %}
                    <tr>
                        <td>{{ p.title }}</td>
                        <td>{{ p.presenter.get_full_name|default:p.presenter.email

```

```
}}</td>

<td><span class="chip">{{ p.get_status_display }}</span></td>
<td>
  <div class="chip-row">
    {% for a in p.review_assignments.all %}
      {% if a.is_active %}
```

Frontend UI(ui.js)

Εδώ υπάρχει ο κώδικας javascript που διαχειρίζεται τις λειτουργίες του frontend.

```
((() => {
  const getCookie = (name) => {
    const row = document.cookie
      .split('; ')
      .find((entry) => entry.startsWith(`${name}=`));
    return row ? decodeURIComponent(row.split('=')[1]) : '';
  };

  const getCsrftoken = () => {
    const tokenFromMeta = document
      .querySelector('meta[name="csrf-token"]')
      ?.getAttribute('content');
    if (tokenFromMeta && tokenFromMeta !== 'NOTPROVIDED') {
      return tokenFromMeta;
    }
    return getCookie('csrftoken');
  };

  const parseResponse = async (response) => {
    const raw = await response.text();
    let data = {};

    if (raw) {
      try {
        data = JSON.parse(raw);
      } catch {
        data = {};
      }
    }

    if (!response.ok || data?.ok === false) {
      const message = data?.error || `Η ενέργεια απέτυχε (HTTP ${response.status}).`;
      throw new Error(message);
    }

    return data;
  };

  const request = async (url, method, payload) => {
    const headers = {
      'X-CSRFToken': getCsrftoken(),
    };

    const isForm = typeof FormData !== 'undefined' && payload instanceof
    FormData;
    const body = isForm ? payload : JSON.stringify(payload ?? {});
    if (!isForm) {
      headers['Content-Type'] = 'application/json';
    }
  };
})
```

```
}

const response = await fetch(url, {
  method,
  headers,
  credentials: 'same-origin',
  body,
});

return parseResponse(response);
};

const setMessage = (selector, text, isError = false) => {
  const target = document.querySelector(selector);
  if (!target) return;
  target.textContent = text;
  target.classList.remove('msg-error', 'msg-success');
  target.classList.add(isError ? 'msg-error' : 'msg-success');
};

const setBusy = (element, busy, workingText = 'Σε εξέλιξη...') => {
  if (!element) return;
  if (busy) {
    element.dataset.originalText = element.textContent;
    element.textContent = workingText;
    element.disabled = true;
    return;
  }

  element.disabled = false;
  if (element.dataset.originalText) {
    element.textContent = element.dataset.originalText;
    delete element.dataset.originalText;
  }
};

const reloadSoon = (ms = 700) => setTimeout(() => window.location.reload(),
ms);
const to = (id) => document.getElementById(id)?.value || '';

const bindButton = (buttonId, builder, urlBuilder, messageSelector, method
= 'POST', onSuccess) => {
  const button = document.getElementById(buttonId);
  if (!button) return;

  button.addEventListener('click', async () => {
    try {
      const payload = builder();
      const url = typeof urlBuilder === 'function' ? urlBuilder(payload) :
urlBuilder;
      setBusy(button, true, 'Αποθήκευση...');
      const data = await request(url, method, payload);
      setBusy(button, false);
      if (typeof onSuccess === 'function') {
        onSuccess(data);
      }
      setMessage(messageSelector, data.message || 'Επιτυχία', false);
    } catch (err) {
      setBusy(button, false);
      setMessage(messageSelector, err.message || 'Αποτυχία', true);
    }
  });
};

const bindMarkNotification = () => {
  const markAllBtn = document.getElementById('mark-all-notifications-btn');
```

```
if (markAllBtn) {
  markAllBtn.addEventListener('click', async () => {
    try {
      setBusy(markAllBtn, true, 'Ενημέρωση...');
      await request('/notifications/mark-all/', 'POST', {});
      setBusy(markAllBtn, false);
      setMessage('#mark-all-notifications-btn', 'Επιτυχία', false);
      reloadSoon(250);
    } catch (err) {
      setBusy(markAllBtn, false);
      setMessage('#mark-all-notifications-btn', err.message ||
'Αποτυχία', true);
    }
  });
}

document.querySelectorAll('.js-mark-notification').forEach((button) => {
  button.addEventListener('click', async () => {
    const id = button.dataset.notificationId;
    try {
      setBusy(button, true, 'Ενημέρωση...');
      await request(`/notifications/mark-read/${id}/`, 'POST', {});
      button.closest('.notification-item').classList.remove('unread');
      button.remove();
      reloadSoon(250);
    } catch (err) {
      setBusy(button, false);
      button.textContent = err.message || 'Αποτυχία';
    }
  });
});

const loginBtn = document.getElementById('login-btn');
if (loginBtn) {
  loginBtn.addEventListener('click', async () => {
    try {
      const email = to('login-email').trim();
      const password = to('login-password');
      if (!email || !password) {
        throw new Error('Συμπλήρωσε email και κωδικό.');
```

```

    if (!email || !password || !first || !last) {
      throw new Error('Συμπλήρωσε όλα τα στοιχεία εγγραφής.');
```

```
    }

    setBusy(registerBtn, true, 'Εγγραφή...');
    const result = await request('/accounts/register/', 'POST', {
      email,
      first_name: first,
      last_name: last,
      password,
      display_name: `${first} ${last}`.trim(),
    });
    if (result.ok) {
      window.location.reload();
    }
  } catch (err) {
    setBusy(registerBtn, false);
    setMessage('#register-message', err.message || 'Αποτυχία', true);
  }
});
}

bindButton(
  'assign-role-btn',
  () => ({
    email: to('role-email').trim(),
    role: document.getElementById('role-select')?.value || 'presenter',
  }),
  () => `/api/conferences/${document.getElementById('conference-
slug')?.value || ''}/role/`,
  '#assign-role-message',
  'POST',
  () => window.location.reload(),
);

bindButton(
  'set-tracks-btn',
  () => ({
    tracks: document
      .getElementById('set-tracks-text')
      ?.value.split(/\n/)
      .map((name) => name.trim())
      .filter(Boolean),
  }),
  () => `/api/conferences/${document.getElementById('conference-
slug')?.value || ''}/tracks/`,
  '#set-tracks-message',
  'POST',
  () => window.location.reload(),
);

bindButton(
  'create-conference-btn',
  () => {
    const title = to('conf-title').trim();
    if (!title) {
      throw new Error('Ο τίτλος είναι υποχρεωτικός.');
```

```
    }

    const regStart = document.getElementById('conf-reg-start')?.value ||
    '';
    const regEnd = document.getElementById('conf-reg-end')?.value || '';
    const startDate = document.getElementById('conf-start')?.value || '';
    const endDate = document.getElementById('conf-end')?.value || '';
  }
);

```

```

    if (!regStart || !regEnd || !startDate || !endDate) {
      throw new Error('Συμπλήρωσε όλα τα πεδία ημερομηνιών.');
```

```

    }

    return {
      title,
      slug: to('conf-slug').trim(),
      description: document.getElementById('conf-description')?.value ||
'',
      venue: to('conf-venue').trim(),
      registration_start: regStart,
      registration_end: regEnd,
      start_date: startDate,
      end_date: endDate,
      max_tracks: to('conf-max-tracks') || 5,
      submission_fee: to('conf-fee') || 0,
      currency: (to('conf-currency') || 'EUR').trim(),
      status: document.getElementById('conf-status')?.value || 'draft',
    };
  },
  '/api/conferences/create/',
  '#create-conference-message',
  'POST',
  () => reloadSoon(350),
);

document.querySelectorAll('.js-conference-status').forEach((button) => {
  button.addEventListener('click', async () => {
    const slug = button.dataset.slug;
    const status = button.dataset.status;
    const message = document.getElementById('conference-status-message');
    if (!slug || !status) return;

    try {
      setBusy(button, true, 'Ενημέρωση...');
      await request(`/api/conferences/${slug}/status/`, 'POST', { status
});
      setBusy(button, false);
      if (message) {
        message.textContent = 'Ενημερώθηκε';
        message.classList.remove('msg-error', 'msg-success');
        message.classList.add('msg-success');
      }
      reloadSoon(450);
    } catch (err) {
      setBusy(button, false);
      if (message) {
        message.textContent = err.message || 'Αποτυχία';
        message.classList.remove('msg-error', 'msg-success');
        message.classList.add('msg-error');
      }
    }
  });
});

bindButton(
  'observer-register-btn',
  () => ({}),
  () => `/api/conferences/${document.getElementById('conference-
slug')?.value || ''}/observe/`,
  '#observer-register-message',
  'POST',
  () => window.location.reload(),
);

bindButton(
```

```
'paper-submit-btn',
() => {
  const title = to('paper-title').trim();
  const abstract = to('paper-abstract').trim();
  if (!title || !abstract) {
    throw new Error('Το title και το abstract είναι υποχρεωτικά.');
```

'';

```
const conferenceId = document.getElementById('conference-id')?.value ||
const fileInput = document.getElementById('paper-file');
const file = fileInput?.files?.[0];

if (file) {
  const filename = (file.name || '').toLowerCase();
  if (!filename.endsWith('.pdf')) {
    throw new Error('Το αρχείο πρέπει να είναι PDF.');
```

'';

```
if (file.size > 10 * 1024 * 1024) {
  throw new Error('Το PDF είναι πολύ μεγάλο (max 10MB).');
```

'';

```
const form = new FormData();
form.append('conference_id', conferenceId);
form.append('title', title);
form.append('abstract', abstract);
form.append('keywords', to('paper-keywords') || '');
form.append('paper_file', file);
return form;
}

return {
  conference_id: conferenceId,
  title,
  abstract,
  keywords: to('paper-keywords'),
};
},
'/submissions/create/',
'#paper-submit-message',
'POST',
() => window.location.reload(),
);

document.querySelectorAll('.js-review-form').forEach((form) => {
  const submit = form.querySelector('.js-review-submit');
  if (!submit) return;

  submit.addEventListener('click', async () => {
    const paperId = form.dataset.paperId;
    const decision = form.querySelector('.review-decision')?.value ||
'accept';
    const scoreRaw = form.querySelector('.review-score')?.value;
    const score = scoreRaw === '' ? null : Number(scoreRaw);
    const comments = form.querySelector('.review-comment')?.value || '';
    const messageEl = form.querySelector(`.review-message[data-paper-
id="${paperId}"]`);

    if (score !== null && Number.isNaN(score)) {
      if (messageEl) {
        messageEl.textContent = 'Score must be a number.';
        messageEl.classList.remove('msg-error', 'msg-success');
        messageEl.classList.add('msg-error');
      }
      return;
    }
  });
});
```

```
try {
  setBusy(submit, true, 'Αποθήκευση...');
  await request(`/reviews/${paperId}/submit/`, 'POST', {
    decision,
    score,
    comments,
  });
  if (messageEl) {
    messageEl.textContent = 'Αποθηκεύτηκε';
    messageEl.classList.remove('msg-error', 'msg-success');
    messageEl.classList.add('msg-success');
  }
  reloadSoon(900);
} catch (err) {
  setBusy(submit, false);
  if (messageEl) {
    messageEl.textContent = err.message || 'Αποτυχία';
    messageEl.classList.remove('msg-error', 'msg-success');
    messageEl.classList.add('msg-error');
  }
}
});
});

document.querySelectorAll('.js-assign-reviewers-form').forEach((form) => {
  const submit = form.querySelector('.js-assign-reviewers-btn');
  if (!submit) return;

  submit.addEventListener('click', async () => {
    const paperId = form.dataset.paperId;
    const messageEl = form.querySelector(`.assign-reviewers-message[data-
paper-id="${paperId}"]`);
    const reviewer1 = form.querySelector('.reviewer-email-
1')?.value.trim();
    const reviewer2 = form.querySelector('.reviewer-email-
2')?.value.trim();
    const reviewer3 = form.querySelector('.reviewer-email-
3')?.value.trim();
```

Παράρτημα Β: Συνοπτικές Οδηγίες Εγκατάστασης

- Κατεβάζουμε και εγκαθιστούμε την Python έκδοσης 3.12 ή νεότερης.
- Κάνουμε unzip το αρχείο που περιέχει το project.
- Ανοίγουμε το project με το εργαλείο ή IDE της επιλογής μας.
- Δημιουργούμε ένα virtual environment για την εφαρμογή και το ενεργοποιούμε.
- Εγκαθιστούμε τις απαραίτητες βιβλιοθήκες του project με την εντολή:

```
pip install -r requirements.txt
```

- Ρυθμίζουμε το Stripe για τη λειτουργία των πληρωμών. Συγκεκριμένα, δημιουργούμε ή χρησιμοποιούμε έναν Stripe λογαριασμό σε test mode και προσθέτουμε στο project τα απαραίτητα test keys, όπως το publishable key και το secret key. Ανοίγουμε το Stripe CLI και εκτελούμε την εντολή ακρόασης, ώστε τα Stripe events να προωθούνται στο αντίστοιχο endpoint της εφαρμογής,:

```
stripe listen --forward-to http://127.0.0.1:8000/payments/webhook/stripe
```

- Ρυθμίζουμε και ανοίγουμε το Mailpit για τον έλεγχο των email ειδοποιήσεων σε περιβάλλον ανάπτυξης. Το Mailpit πρέπει να εκτελείται τοπικά και η εφαρμογή να χρησιμοποιεί ως SMTP server το localhost στη θύρα 1025. Για να δούμε τα email που αποστέλλονται από την εφαρμογή, ανοίγουμε στον browser τη διεύθυνση:

```
http://localhost:8025/
```

- Δημιουργούμε τους πίνακες στη βάση δεδομένων με την εντολή:

```
python .\manage.py migrate
```

- Δημιουργούμε έναν admin χρήστη με την εντολή:

```
python .\manage.py createsuperuser
```

- Τρέχουμε την εφαρμογή με την εντολή:

```
python .\manage.py runserver
```

- Ανοίγουμε την εφαρμογή από το link που εμφανίζεται στο terminal, συνήθως:

```
http://127.0.0.1:8000/
```

- Ελέγχουμε ότι η εφαρμογή λειτουργεί σωστά, ότι οι πληρωμές εκτελούνται δοκιμαστικά μέσω Stripe και ότι οι email ειδοποιήσεις εμφανίζονται στο Mailpit.
- Η εφαρμογή είναι έτοιμη προς χρήση.

Υπεύθυνη Δήλωση Συγγραφέα:

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν.1599/1986, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης.