



Πανεπιστήμιο Αιγαίου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών
Συστημάτων

321-4002 – Τεχνολογία Λογισμικού

Διδάσκων: Κυριάκος Κρητικός

Project: Festival Managment System

Εργαστηριακός Συνεργάτης: Αλέξανδρος Φακής

321/2019157 Στυλιανός Νικολόπουλος

321/2020010 Γεώργιος Αναγνωστόπουλος

321/2020077 Γεώργιος Καμτσικλής

Σάμος, 17 Σεπτεμβρίου 2025



Πίνακας περιεχομένων

1	Περίληψη	3
2	Οργάνωση	4
2.1	Ρόλοι ομάδας.....	4
2.2	Στάδια Ανάπτυξης.....	4
2.3	Χρονοδιάγραμμα και Διάρκεια	5
3	Απαιτήσεις Συστήματος.....	6
3.1	Λειτουργικές Απαιτήσεις.....	6
3.2	Μη Λειτουργικές Απαιτήσεις.....	6
4	Σχεδιασμός Συστήματος	8
4.1	Διαγράμματα Αρχιτεκτονικής.....	8
5	Υλοποίηση Συστήματος.....	9
5.1	Υλοποίηση Βάσης Δεδομένων.....	9
5.2	Υλοποίηση Back – End	9
6	Συμπεράσματα.....	10
6.1	Αξιολόγηση Έργου.....	10
6.2	Μελλοντική Επέκταση και Βελτιώσεις	10



1 Περίληψη

Η παρούσα εργασία έχει ως αντικείμενο την ανάπτυξη ενός πληροφοριακού συστήματος για τη διαχείριση μουσικών φεστιβάλ. Στόχος είναι ο σχεδιασμός και η υλοποίηση ενός backend συστήματος που θα παρέχει RESTful υπηρεσίες για τη διαχείριση χρηστών, φεστιβάλ και μουσικών εμφανίσεων, αξιοποιώντας τεχνολογίες που υποστηρίζουν συνεργατική ανάπτυξη λογισμικού, αυτοματοποίηση, έλεγχο εκδόσεων και μονάδων.

Το έργο επικεντρώνεται στη δημιουργία ενός backend πληροφοριακού συστήματος που θα υποστηρίζει την πλήρη ροή διαδικασιών, από τη δημιουργία και οργάνωση ενός φεστιβάλ έως την υποβολή, αξιολόγηση και τελική αποδοχή ή απόρριψη μουσικών εμφανίσεων. Μέσω της υλοποίησης RESTful υπηρεσιών και της αξιοποίησης βάσης δεδομένων, επιδιώκεται να προσφερθεί μια αξιόπιστη, ασφαλής και επεκτάσιμη λύση, η οποία θα διασφαλίζει τη διαφάνεια των ρόλων, την ακεραιότητα των δεδομένων και την ταχύτητα εκτέλεσης των λειτουργιών.

Το σύστημα βασίζεται σε RESTful web services που αλληλεπιδρούν με υποκείμενη βάση δεδομένων για την αποθήκευση και διαχείριση όλων των σχετικών οντοτήτων, όπως χρήστες, φεστιβάλ και εμφανίσεις (performances). Υποστηρίζει διαφορετικούς ρόλους χρηστών (Visitor, User, Artist, Organizer, Staff, Admin), στους οποίους αντιστοιχούν συγκεκριμένα δικαιώματα και λειτουργίες. Η λειτουργικότητα περιλαμβάνει, μεταξύ άλλων, τη δημιουργία και ενημέρωση φεστιβάλ, την υποβολή και αξιολόγηση εμφανίσεων, την ανάθεση ρόλων και τη διαχείριση χρηστών, ακολουθώντας προκαθορισμένες καταστάσεις (states) για κάθε φεστιβάλ και εμφάνιση. Με τον τρόπο αυτό διασφαλίζεται η ορθή ροή εργασιών, η συνεργασία μεταξύ διαφορετικών εμπλεκόμενων ρόλων και η αξιοπιστία της πληροφορίας.



2 Οργάνωση

2.1 Ρόλοι ομάδας

Η εργασία υλοποιήθηκε από τρία μέλη ομάδας, καθένα από τα οποία ανέλαβε έναν κύριο ρόλο, ενώ παράλληλα συνέβαλε και στα υπόλοιπα στάδια της ανάπτυξης ώστε να εξασφαλιστεί η ομαλή συνεργασία και η συνολική κατανόηση του έργου από όλα τα μέλη.

- **Μέλος Α:** Είχε ως βασική ευθύνη τη συλλογή και ανάλυση των απαιτήσεων του συστήματος, την τεκμηρίωση των λειτουργικών και μη λειτουργικών απαιτήσεων, καθώς και τη διαμόρφωση της σχετικής ενότητας στην αναφορά. Παράλληλα, συμμετείχε στον σχεδιασμό των use case διαγραμμάτων και συνέβαλε στην υλοποίηση μικρών τμημάτων κώδικα.
- **Μέλος Β:** Ανέλαβε κυρίως τον σχεδιασμό του συστήματος και την παραγωγή των βασικών διαγραμμάτων (use case, component, activity, sequence, ER), καθώς και την τεκμηρίωσή τους στην αναφορά. Επιπλέον, βοήθησε στη συλλογή απαιτήσεων, συμμετείχε στη συγγραφή τμημάτων του κώδικα backend και συνέβαλε στις δοκιμές του συστήματος.
- **Μέλος Γ:** Είχε ως κύριο ρόλο την υλοποίηση του backend συστήματος, την ανάπτυξη των RESTful υπηρεσιών, τη διασύνδεση με τη βάση δεδομένων και την ανάπτυξη unit tests. Παράλληλα, συνέβαλε στον σχεδιασμό της βάσης δεδομένων, στη δημιουργία της τεκμηρίωσης υλοποίησης και στη συγγραφή της τελικής αναφοράς.

Και τα τρία μέλη συνεργάστηκαν ενεργά σε όλα τα στάδια του έργου, από τη συλλογή απαιτήσεων έως τις δοκιμές και τη συγγραφή της αναφοράς, διατηρώντας κοινή παρουσία στο GitHub repository και φροντίζοντας για τη συνεχή ανασκόπηση και βελτίωση του κώδικα και της τεκμηρίωσης.

- Μέλος Α → Γεώργιος Αναγνωστόπουλος
- Μέλος Β → Γεώργιος Καμτσικλής
- Μέλος Γ → Στυλιανός Νικολόπουλος

2.2 Στάδια Ανάπτυξης

Προκειμένου να ολοκληρωθεί και να είναι λειτουργικό το έργο ακολουθήθηκαν τα παρακάτω στάδια ανάπτυξης:

1. **Ανάλυση Απαιτήσεων (15%)** – Καταγραφή και οργάνωση λειτουργικών και μη λειτουργικών απαιτήσεων.
2. **Σχεδιασμός (30%)** – Δημιουργία διαγραμμάτων για την αρχιτεκτονική, τις χρήσεις και τη συμπεριφορά του συστήματος, καθώς και για τη βάση δεδομένων.
3. **Υλοποίηση (50%)** – Ανάπτυξη του backend με RESTful υπηρεσίες, χρήση GitHub για συνεργασία και υλοποίηση unit tests.
4. **Συγγραφή Αναφοράς (5%)** – Ενοποίηση παραδοτέων και τεκμηρίωση της εργασίας.



2.3 Χρονοδιάγραμμα και Διάρκεια

Η συνολική διάρκεια ανάπτυξης της εργασίας εκτιμάται σε 252.8 ώρες. Η κατανομή του χρόνου στα στάδια ήταν περίπου η εξής:

- Ανάλυση απαιτήσεων: ~ 3 ημέρες
- Σχεδιασμός: ~ 7 ημέρες
- Υλοποίηση: ~ 20 ημέρες
- Αναφορά: ~ 1.6 ημέρες



3 Απαιτήσεις Συστήματος

Στην ενότητα αυτή παρουσιάζονται οι απαιτήσεις του συστήματος, οι οποίες καθορίζουν τις λειτουργίες που πρέπει να υποστηρίξει, καθώς και τα χαρακτηριστικά που διασφαλίζουν την ορθή και αξιόπιστη λειτουργία του. Οι απαιτήσεις διαχωρίζονται σε λειτουργικές, που αφορούν τις δυνατότητες και τις ενέργειες των χρηστών και του συστήματος, και σε μη λειτουργικές, που καθορίζουν την απόδοση, την ασφάλεια και τη δομή του συστήματος.

3.1 Λειτουργικές Απαιτήσεις

i. Διαχείριση χρηστών

- Εγγραφή χρήστη με μοναδικό username και ισχυρό password.
- Login / authentication με username + password.
- Δημιουργία και χρήση token για τις υπόλοιπες λειτουργίες.
- Ενημέρωση προσωπικών στοιχείων (εκτός password).
- Αλλαγή password με παλιό + νέο.
- Διαγραφή λογαριασμού από τον χρήστη.
- Ενεργοποίηση / απενεργοποίηση λογαριασμών από admin.

ii. Διαχείριση φεστιβάλ

- Αναζήτηση και προβολή φεστιβάλ με διαφορετικά επίπεδα πρόσβασης ανά role.
- Δημιουργία φεστιβάλ με μοναδικό όνομα και όλα τα υποχρεωτικά στοιχεία.
- Ενημέρωση στοιχείων φεστιβάλ (π.χ. περιγραφή, ημερομηνίες, stages, vendors, budget).
- Διαχείριση καταστάσεων φεστιβάλ: CREATED → SUBMISSION → ASSIGNMENT → REVIEW → SCHEDULING → FINAL_SUBMISSION → DECISION → ANNOUNCED.
- Προσθήκη Organizers και Staff σε φεστιβάλ.
- Διαγραφή φεστιβάλ μόνο στην κατάσταση CREATED.

iii. Διαχείριση εμφανίσεων (Performances)

- Δημιουργία και ενημέρωση εμφάνισης από Artist (περιγραφή, genre, duration, band members, setlist, time slots).
- Προσθήκη band members από Artist.
- Υποβολή ή ανάκληση Performance πριν την υποβολή.
- Ανάθεση Staff από Organizer κατά τη φάση ASSIGNMENT.
- Αξιολόγηση Performance (score + comments) σε κατάσταση REVIEW.
- Έγκριση, απόρριψη ή οριστική αποδοχή Performance ανάλογα με την κατάσταση (SCHEDULING / FINAL_SUBMISSION / DECISION).

3.2 Μη Λειτουργικές Απαιτήσεις

i. Απόδοση και αξιοπιστία

- Κάθε αίτημα να εκτελείται σε 5–10 δευτερόλεπτα.
- Το σύστημα να είναι αξιόπιστο, χωρίς εσωτερικά σφάλματα.
- Οι αλλαγές στη βάση να είναι συνεπείς (transactional).



ii. Διαχείριση σφαλμάτων και ασφάλεια

- Κατάλληλα μηνύματα λάθους για λανθασμένες εισόδους.
- Αποφυγή διπλών εκτελέσεων μη idempotent ενεργειών.
- Απενεργοποίηση λογαριασμού μετά από 3 αποτυχημένα login ή password update.
- Error messages για invalid ή expired tokens.
- Απενεργοποίηση λογαριασμών σε περίπτωση σύγκρουσης tokens.
- Ακύρωση tokens όταν αλλάζουν στοιχεία χρήστη ή password.

iii. Πρόσβαση και εξουσιοδότηση

- Το σύστημα να δέχεται μόνο authenticated χρήστες με σωστά roles.
- Κάθε λειτουργία να εκτελείται μόνο αν ο χρήστης έχει τα κατάλληλα δικαιώματα.

iv. Σχεδίαση και δομή συστήματος

- Ασφαλής διαχείριση passwords (pattern check, strong password rules).
- Modular design: separation of concerns (Controllers, Services, DAO).
- Validation token σε κάθε αίτημα.



4 Σχεδιασμός Συστήματος

4.1 Διαγράμματα Αρχιτεκτονικής



5 Υλοποίηση Συστήματος

5.1 Υλοποίηση Βάσης Δεδομένων

5.2 Υλοποίηση Back – End



6 Συμπεράσματα

6.1 Αξιολόγηση Έργου

6.2 Μελλοντική Επέκταση και Βελτιώσεις