

Traineeship Application Sprint Report

WEB_APP_ANY%

ΚΙΟΣΣΕΣ ΑΘΑΝΑΣΙΟΣ ΑΜ: 5254

ΣΙΟΥΛΑΣ ΑΛΕΞΑΝΔΡΟΣ ΑΜ: 5349

VERSIONS HISTORY

Date	Version	Description	Author
10/05/2025	1.0	Initial Draft of the Final Sprint Report	WEB_APP_ANY%
26/05/2025	1.1	Final Version after review and additions	WEB_APP_ANY%

1 Introduction

Η αναφορά συνοψίζει τα τρία sprints που πραγματοποιήθηκαν στο πλαίσιο του project **"Traineeship Management Application"**. Παρουσιάζει την πρόοδο που σημειώθηκε στην υλοποίηση, τις σχεδιαστικές έννοιες που εφαρμόστηκαν, τις περιπτώσεις χρήσης που προέκυψαν από τις ιστορίες χρηστών, τη συνολική αρχιτεκτονική της web app, καθώς και τις CRC κάρτες για τις βασικές κλάσεις του συστήματος.

1.1 Purpose

Ο σκοπός της αναφοράς είναι να τεκμηριώσει τη δουλειά που πραγματοποιήθηκε κατά τη διάρκεια των development sprints για την «εφαρμογή διαχείρισης πρακτικής άσκησης». Περιλαμβάνει τις υλοποιημένες user stories, τις σχεδιαστικές αποφάσεις που λήφθηκαν, και προσφέρει μια ολοκληρωμένη εικόνα της κατάστασης του project στο τέλος των sprints, σύμφωνα με τις απαιτήσεις που προσδιορίστηκαν στο Product Backlog Specification.

1.2 Document Structure

Η συνέχεια της αναφοράς οργανώνεται ως εξής:

- Η **Ενότητα 2** περιγράφει την ομάδα Scrum και συνοψίζει το backlog από όλα τα sprints.
- Η **Ενότητα 3** παρουσιάζει τις κύριες περιπτώσεις χρήσης που προκύπτουν από τις user stories.
- Η **Ενότητα 4** αναλύει την αρχιτεκτονική της web app, τις βασικές σχεδιαστικές έννοιες και τις CRC κάρτες της τρέχουσας έκδοσης του project.
- Η **Ενότητα 5** περιγράφει τη στρατηγική δοκιμών που εφαρμόστηκε.

2 Scrum team and Sprint Backlog

2.1 Scrum team

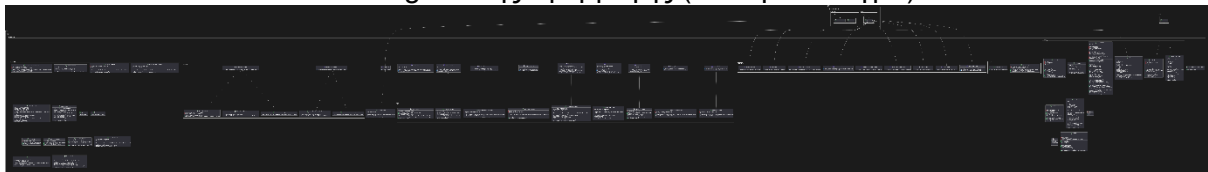
Role	Name
Product Owner	ΚΙΟΣΣΕΣ ΑΘΑΝΑΣΙΟΣ ΣΙΟΥΛΑΣ ΑΛΕΞΑΝΔΡΟΣ
Scrum Master	ΚΙΟΣΣΕΣ ΑΘΑΝΑΣΙΟΣ ΣΙΟΥΛΑΣ ΑΛΕΞΑΝΔΡΟΣ
Development Team	ΚΙΟΣΣΕΣ ΑΘΑΝΑΣΙΟΣ ΣΙΟΥΛΑΣ ΑΛΕΞΑΝΔΡΟΣ

2.2 Sprints

Sprint No	Begin Date	End Date	Number of weeks	User stories Implemented
1	01/04/2025	14/04/2025	2	US1, US2, US3, US4, US7, US13
2	15/04/2025	28/03/2025	2	US5, US6, US8, US10, US14, US16, US17
3	29/04/2025	14/05/2025	2	US9, US11, US12, US15, US18, US19, US20, US21

3 USE CASES

UML Diagram της εφαρμογής (click για άνοιγμα):



3.1 UC01 - User Account Management

- **Use case ID:** UC01 (Αντιστοιχεί σε US1, US2, US3)
- **Actors:** User (Generic user, can be Student, Company representative, Professor, Committee member)
- **Preconditions:** Ο χρήστης δεν έχει λογαριασμό ή είναι αποσυνδεδεμένος.
- **Main flow of events:**
 1. Ο χρήστης επισκέπτεται τη σελίδα εγγραφής.
 2. Συμπληρώνει στοιχεία (username, password, role) και δημιουργεί λογαριασμό.
 3. Ο χρήστης επισκέπτεται τη σελίδα login.
 4. Ο χρήστης εισάγει τα στοιχεία του (username, password) και κάνει login.
 5. Ο χρήστης, αφού συνδεθεί, επιλέγει να κάνει logout από την εφαρμογή.
- **Post conditions:** Ο λογαριασμός έχει δημιουργηθεί. Ο χρήστης έχει συνδεθεί επιτυχώς. Ο χρήστης έχει αποσυνδεθεί επιτυχώς.

3.2 UC02 - Student Profile Management

- **Use case ID:** UC02 (Αντιστοιχεί σε US4)
- **Actors:** Student

- **Preconditions:** Ο φοιτητής είναι εγγεγραμμένος και έχει κάνει login.
- **Main flow of events:**
 1. Ο φοιτητής πλοηγείται στη σελίδα διαχείρισης του προφίλ του.
 2. Ο φοιτητής συμπληρώνει ή τροποποιεί το προφίλ του με στοιχεία (full name, university ID, list of interests, list of skills, preferred location).
 3. Αποθηκεύει τις πληροφορίες του.
- **Post conditions:** Το προφίλ έχει αποθηκευτεί/ενημερωθεί επιτυχώς.

3.3 UC03 - Apply for Traineeship

- **Use case ID:** UC03 (Αντιστοιχεί σε US5)
- **Actors:** Student, Committee (implicity, as they receive the application)
- **Preconditions:** Ο φοιτητής έχει συνδεθεί και έχει πλήρες προφίλ. Υπάρχουν διαθέσιμες θέσεις πρακτικής.
- **Main flow of events:**
 1. Ο φοιτητής επιλέγει "Αναζήτηση Θέσεων" πρακτικής άσκησης.
 2. Το σύστημα εμφανίζει λίστα διαθέσιμων θέσεων.
 3. Ο φοιτητής επιλέγει μια θέση που τον ενδιαφέρει.
 4. Ο φοιτητής υποβάλλει αίτηση για τη συγκεκριμένη θέση.
- **Post conditions:** Η αίτηση έχει καταχωρηθεί και αποσταλεί στην επιτροπή πρακτικής άσκησης.

3.4 UC04 - Student Logbook Submission

- **Use case ID:** UC04 (Αντιστοιχεί σε US6)
- **Actors:** Student
- **Preconditions:** Ο φοιτητής έχει τοποθετηθεί σε θέση πρακτικής άσκησης.
- **Main flow of events:**
 1. Ο φοιτητής πλοηγείται στην ενότητα του logbook για την πρακτική του.
 2. Ο φοιτητής συμπληρώνει αναφορά δραστηριότητας.
 3. Υποβάλλει την αναφορά.
- **Post conditions:** Η αναφορά του ημερολογίου έχει αποθηκευτεί.

3.5 UC05 - Company Profile & Position Management

- **Use case ID:** UC05 (Αντιστοιχεί σε US7, US8, US10, US11)
- **Actors:** Company
- **Preconditions:** Η εταιρεία έχει λογαριασμό και ο εκπρόσωπός της έχει συνδεθεί.
- **Main flow of events:**

1. Η εταιρεία δημιουργεί/ενημερώνει το προφίλ της (company name, location).
 2. Η εταιρεία πλοηγείται στη διαχείριση θέσεων πρακτικής.
 3. Δημιουργεί μια νέα αγγελία θέσης πρακτικής (start/end dates, description, required skills, topics of interest).
 4. Η εταιρεία βλέπει τη λίστα των διαθέσιμων/ενεργών θέσεων που έχει αναρτήσει.
 5. Η εταιρεία επιλέγει μια θέση που δεν είναι πλέον διαθέσιμη και τη διαγράφει/απενεργοποιεί.
- **Post conditions:** Το προφίλ της εταιρείας έχει δημιουργηθεί/ενημερωθεί. Οι αγγελίες έχουν δημιουργηθεί / διαγραφεί.

3.6 UC06 - Company Evaluation of Student

- **Use case ID:** UC06 (Αντιστοιχεί σε US12)
- **Actors:** Company
- **Preconditions:** Η πρακτική ενός φοιτητή στην εταιρεία είναι σε εξέλιξη.
- **Main flow of events:**
 1. Η εταιρεία επιλέγει τον φοιτητή/πρακτική που θέλει να αξιολογήσει.
 2. Η εταιρεία συμπληρώνει φόρμα αξιολόγησης για τον φοιτητή (motivation, effectiveness, efficiency on a scale 1-5).
 3. Υποβάλλει την αξιολόγηση.
- **Post conditions:** Η αξιολόγηση του φοιτητή από την εταιρεία έχει καταγραφεί.

3.7 UC07 - Professor Profile & Supervision Management

- **Use case ID:** UC07 (Αντιστοιχεί σε US13, US14)
- **Actors:** Professor
- **Preconditions:** Ο καθηγητής είναι μέλος της πλατφόρμας και έχει συνδεθεί.
- **Main flow of events:**
 1. Ο καθηγητής δημιουργεί/ενημερώνει το προφίλ του (full name, list of interests).
 2. Ο καθηγητής βλέπει τη λίστα των πρακτικών ασκήσεων που εποπτεύει.
- **Post conditions:** Το προφίλ του καθηγητή έχει δημιουργηθεί/ενημερωθεί. Τα δεδομένα των εποπτευόμενων πρακτικών έχουν εμφανιστεί.

3.8 UC08 - Professor Evaluation of Student and Company

- **Use case ID:** UC08 (Αντιστοιχεί σε US15)
- **Actors:** Professor
- **Preconditions:** Ο φοιτητής έχει πρακτική υπό την επίβλεψή του καθηγητή. Η πρακτική είναι σε εξέλιξη.

- **Main flow of events:**
 1. Ο καθηγητής επιλέγει την πρακτική/φοιτητή που θέλει να αξιολογήσει.
 2. Ο καθηγητής συμπληρώνει φόρμα αξιολόγησης για τον φοιτητή (motivation, effectiveness, efficiency on a scale 1-5).
 3. Ο καθηγητής συμπληρώνει φόρμα αξιολόγησης για την εταιρεία υποδοχής (facilities, guidance on a scale 1-5).
 4. Υποβάλλει τις αξιολογήσεις.
- **Post conditions:** Οι αξιολογήσεις του φοιτητή και της εταιρείας από τον καθηγητή έχουν καταγραφεί.

3.9 UC09 - Committee Reviews Student Applications

- **Use case ID:** UC09 (Αντιστοιχεί σε US16)
- **Actors:** Committee
- **Preconditions:** Υπάρχουν αιτήσεις φοιτητών για θέσεις πρακτικής. Το μέλος της επιτροπής έχει συνδεθεί.
- **Main flow of events:**
 1. Η επιτροπή πλοηγείται στην ενότητα διαχείρισης αιτήσεων.
 2. Η επιτροπή βλέπει τη λίστα των φοιτητών που έχουν υποβάλει αίτηση για πρακτική άσκηση.
- **Post conditions:** Η λίστα των αιτούντων φοιτητών έχει εμφανιστεί.

3.10 UC10 - Committee Assigns Position to Student

- **Use case ID:** UC10 (Αντιστοιχεί σε US17, US18)
- **Actors:** Committee
- **Preconditions:** Ο φοιτητής έχει επιλεγεί από την επιτροπή ως υποψήφιος για τοποθέτηση. Υπάρχουν διαθέσιμες θέσεις.
- **Main flow of events:**
 1. Η επιτροπή επιλέγει έναν φοιτητή από τη λίστα των αιτούντων.
 2. Η επιτροπή εφαρμόζει στρατηγική αναζήτησης θέσεων για τον φοιτητή (βάσει ενδιαφερόντων, τοποθεσίας ή και των δύο, με ταίριασμα δεξιοτήτων). (Βλ. UC14, UC15)
 3. Το σύστημα εμφανίζει τις κατάλληλες διαθέσιμες θέσεις.
 4. Η επιτροπή επιλέγει μια θέση και την αναθέτει στον φοιτητή.
- **Post conditions:** Η θέση έχει ανατεθεί στον φοιτητή.

3.11 UC11 - Committee Assigns Supervisor to Traineeship

- **Use case ID:** UC11 (Αντιστοιχεί σε US19)
- **Actors:** Committee

- **Preconditions:** Υπάρχει ανατεθειμένη θέση πρακτικής σε φοιτητή, η οποία δεν έχει ακόμη επιβλέποντα καθηγητή.
- **Main flow of events:**
 1. Η επιτροπή επιλέγει μια πρακτική άσκηση που χρειάζεται ανάθεση επιβλέποντα.
 2. Η επιτροπή εφαρμόζει στρατηγική ανάθεσης επιβλέποντα (βάσει ενδιαφερόντων καθηγητή ή φόρτου εργασίας). (Βλ. UC15)
 3. Το σύστημα προτείνει κατάλληλους καθηγητές.
 4. Η επιτροπή επιλέγει αυτόματα έναν καθηγητή και τον αναθέτει ως επιβλέποντα στην πρακτική.
- **Post conditions:** Ο καθηγητής έχει συνδεθεί με τη θέση ως επιβλέπων.

3.12 UC12 - Committee Reviews Traineeship Progress

- **Use case ID:** UC12 (Αντιστοιχεί σε US20)
- **Actors:** Committee
- **Preconditions:** Πρακτικές ασκήσεις φοιτητών βρίσκονται σε εξέλιξη.
- **Main flow of events:**
 1. Η επιτροπή πλοηγείται στη λίστα των εν εξελίξει πρακτικών ασκήσεων.
 2. Η επιτροπή επιλέγει μια συγκεκριμένη πρακτική.
 3. Η επιτροπή βλέπει τις αξιολογήσεις (αν υπάρχουν) από την εταιρεία και τον επιβλέποντα καθηγητή.
- **Post conditions:** Η πρόοδος της πρακτικής έχει προβληθεί.

3.13 UC13 - Committee Completes Traineeship

- **Use case ID:** UC13 (Αντιστοιχεί σε US21)
- **Actors:** Committee
- **Preconditions:** Η πρακτική άσκηση ενός φοιτητή έχει αξιολογηθεί από την εταιρεία και τον επιβλέποντα καθηγητή.
- **Main flow of events:**
 1. Η επιτροπή επιλέγει μια πρακτική που είναι έτοιμη για οριστικοποίηση.
 2. Η επιτροπή καταχωρεί τελικό χαρακτηρισμό (pass/fail) για την πρακτική άσκηση.
- **Post conditions:** Η πρακτική άσκηση έχει ολοκληρωθεί και ο τελικός χαρακτηρισμός έχει καταχωρηθεί.

3.14 UC14 - Search by Interests/Location (Strategy for Traineeship Position Search)

- **Use case ID:** UC14 (Εσωτερικό του συστήματος, καλείται από UC10, σχετίζεται με US17 και NF1.2)

- **Actors:** Committee (μέσω του συστήματος)
- **Preconditions:** Υπάρχει επιλεγμένος φοιτητής για αναζήτηση θέσης. Η επιτροπή έχει επιλέξει στρατηγική αναζήτησης.
- **Main flow of events (παραδείγματα στρατηγικών):**
 1. *Strategy 1 (Interests):* Το σύστημα φιλτράρει τις διαθέσιμες θέσεις βάσει των ενδιαφερόντων του φοιτητή, χρησιμοποιώντας Jaccard similarity metric με τα topics της θέσης.
 2. *Strategy 2 (Location):* Το σύστημα φιλτράρει τις διαθέσιμες θέσεις βάσει της προτιμώμενης τοποθεσίας του φοιτητή και της τοποθεσίας της εταιρείας.
 3. *Strategy 3 (Combined):* Το σύστημα συνδυάζει τα παραπάνω κριτήρια.
 4. Σε όλες τις περιπτώσεις, οι θέσεις πρέπει να έχουν required skills που ταιριάζουν με τις δεξιότητες του φοιτητή.
- **Post conditions:** Παρουσιάζεται φιλτραρισμένη λίστα διαθέσιμων θέσεων πρακτικής.

3.15 UC15 - Strategy Selection (For Position and Supervisor Assignment)

- **Use case ID:** UC15 (Εσωτερικό του συστήματος, καλείται από UC10 & UC11, σχετίζεται με NF1.2, NF1.3)
- **Actors:** Committee (μέσω του συστήματος)
- **Preconditions:** Καλείται η επιλογή στρατηγικής (είτε για αναζήτηση θέσης είτε για ανάθεση επιβλέποντα).
- **Main flow of events:**
 1. Η επιτροπή (μέσω UI) επιλέγει το κριτήριο για την τρέχουσα λειτουργία (π.χ., για ανάθεση θέσης: "ενδιαφέροντα", "τοποθεσία", "συνδυασμός"· για ανάθεση καθηγητή: "ενδιαφέροντα καθηγητή", "φόρτος καθηγητή").
 2. Το σύστημα (CommitteeService) χρησιμοποιεί ένα Factory (π.χ., PositionsSearchFactory, SupervisorAssignmentFactory) για να δημιουργήσει το αντικείμενο της κατάλληλης στρατηγικής.
 3. Η επιλεγμένη στρατηγική εκτελείται.
- **Post conditions:** Η κατάλληλη στρατηγική έχει επιλεγεί και εκτελεστεί δυναμικά.

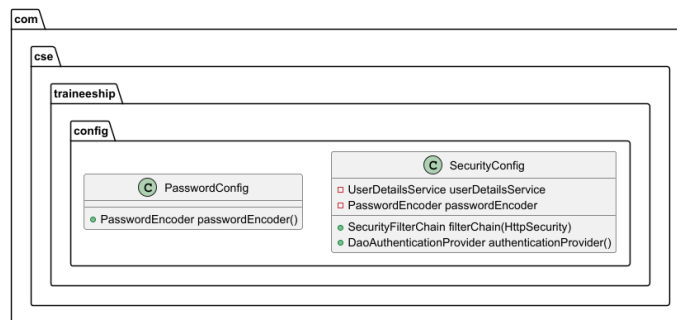
4 Design

4.1 Architecture

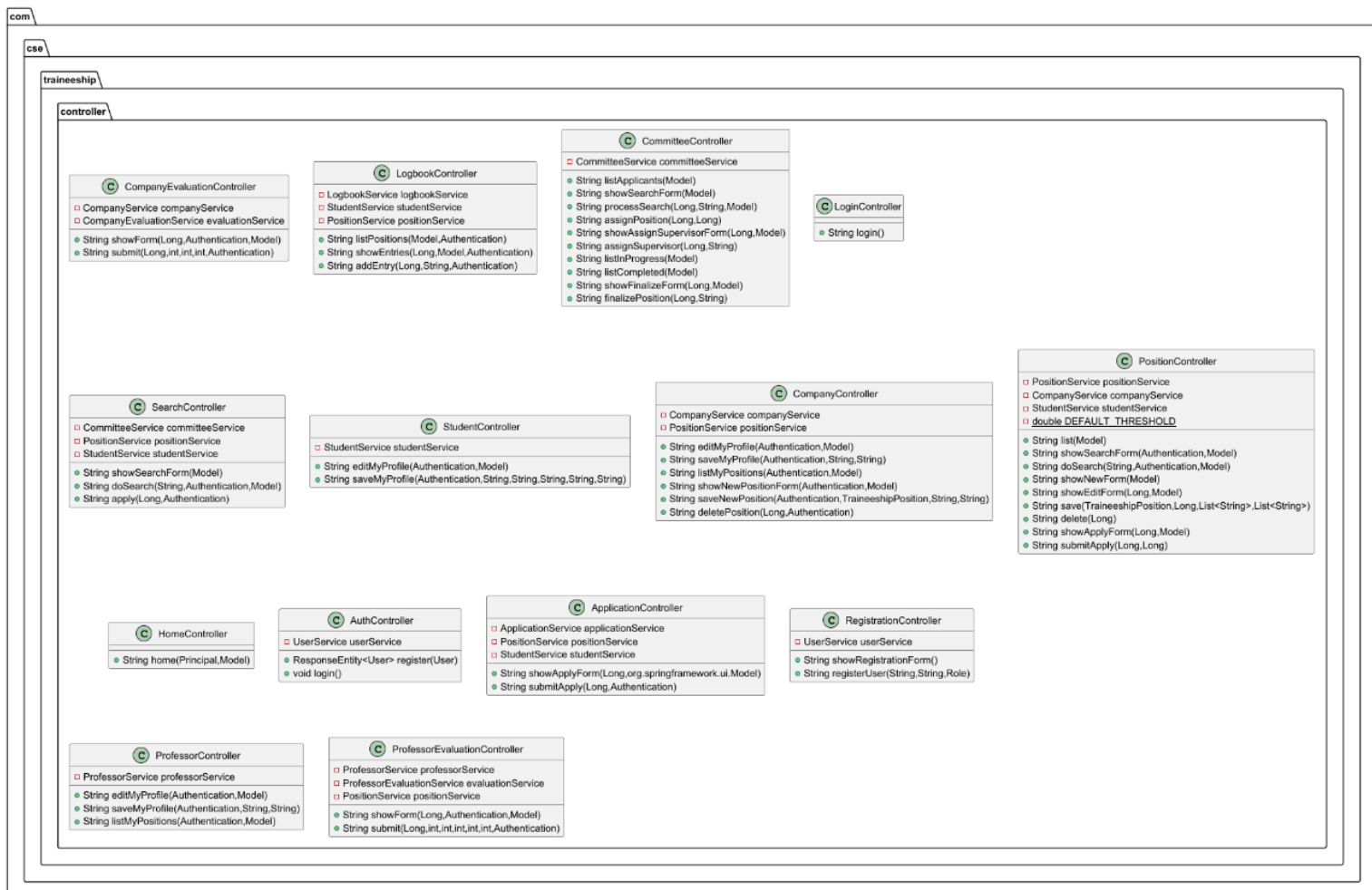
Η αρχιτεκτονική της εφαρμογής ακολουθεί το μοτίβο **Model-View-Controller (MVC)**, αξιοποιώντας το πλαίσιο **Spring Boot**. Αυτή η προσέγγιση προάγει τον σαφή διαχωρισμό αρμοδιοτήτων (separation of concerns), χαμηλή σύζευξη (low coupling) και υψηλή συνοχή (high cohesion), όπως απαιτείται από το NF1.1. Η αρχιτεκτονική βασίζεται επίσης σε μοτίβα Enterprise Application Architecture (EAA) του Martin Fowler, όπως το **Service Layer** για την αποσύνδεση της επιχειρησιακής λογικής από το front-end, και το **Data Mapper** για την αντιστοίχιση των αντικειμένων του domain model με την βάση δεδομένων.

Τα βασικά πακέτα της εφαρμογής είναι:

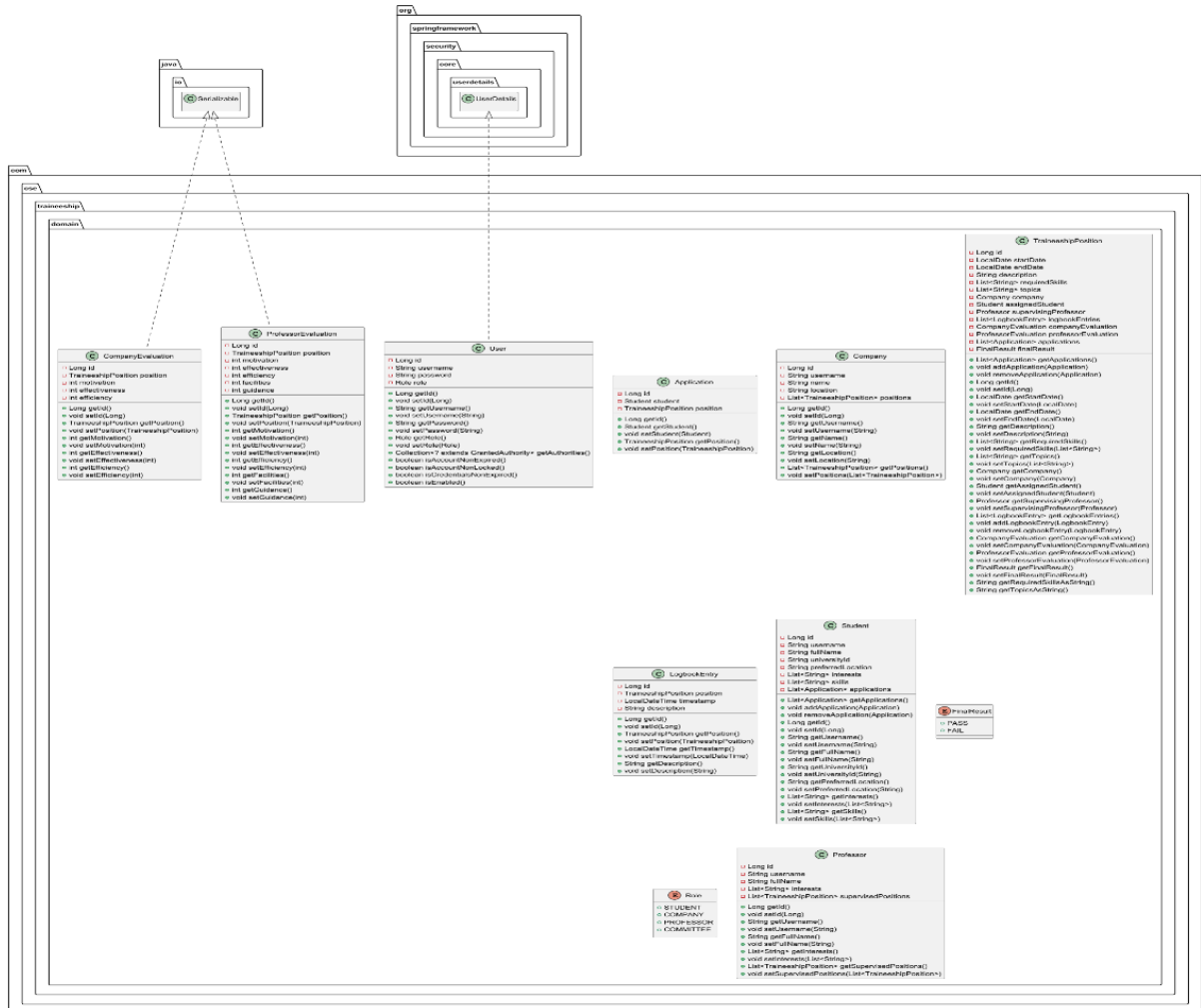
- **com.cse.traineeship.config**: Περιέχει κλάσεις διαμόρφωσης της εφαρμογής, όπως η διαμόρφωση ασφαλείας (Spring Security) και η διαμόρφωση για την κρυπτογράφηση των κωδικών.



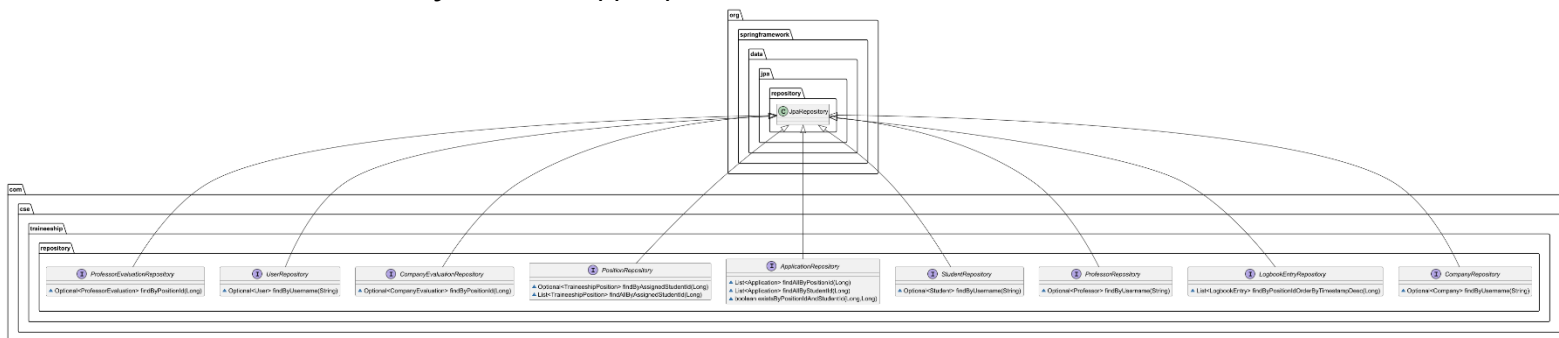
- **com.cse.traineeship.controller**: Περιέχει τις κλάσεις Controller που διαχειρίζονται τα αιτήματα HTTP από τον χρήστη, αλληλεπιδρούν με τις κλάσεις Service για την εκτέλεση της επιχειρησιακής λογικής και επιστρέφουν τις κατάλληλες Views (σελίδες Thymeleaf) στον χρήστη.



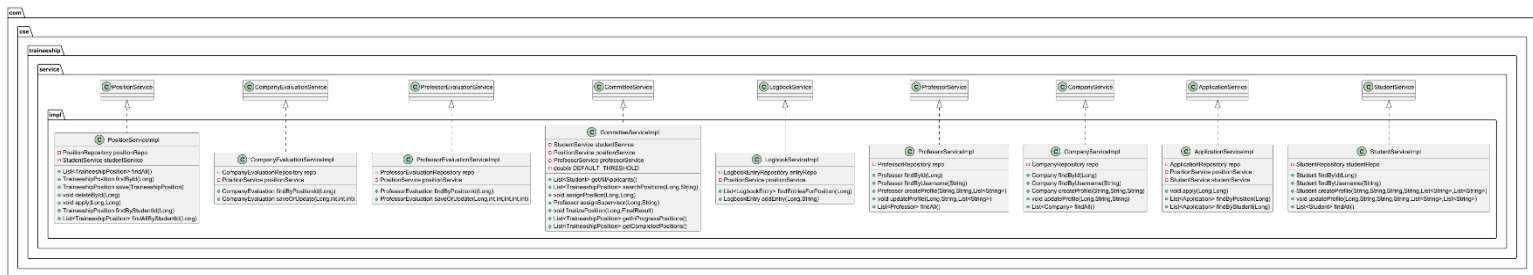
- **com.cse.traineeship.domain:** Περιλαμβάνει τις βασικές οντότητες (Entities) του μοντέλου δεδομένων της εφαρμογής, όπως User, Student, Company, Professor, TraineeshipPosition, Evaluation, Role. Αυτές οι κλάσεις αντιστοιχίζονται σε πίνακες της βάσης δεδομένων μέσω JPA annotations.



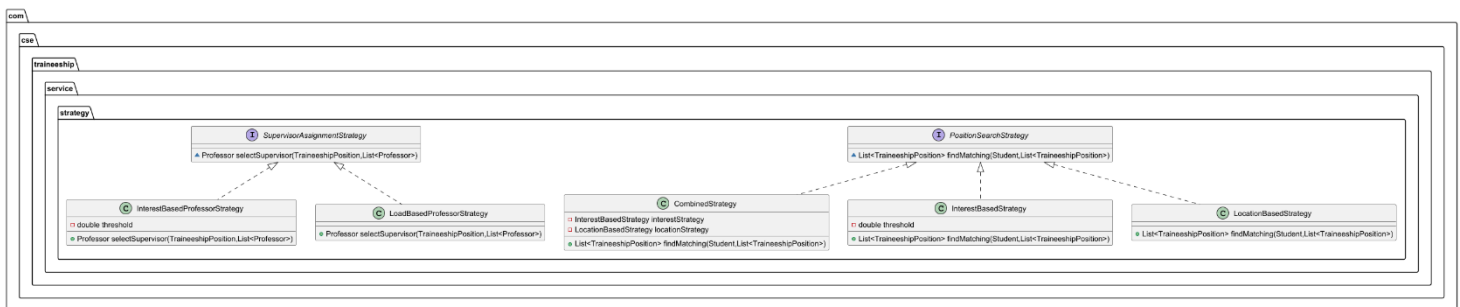
- **com.cse.traineeship.repository:** Περιέχει τα interfaces που επεκτείνουν το JpaRepository του Spring Data JPA. Αυτά τα interfaces παρέχουν τις βασικές CRUD (Create, Read, Update, Delete) λειτουργίες για τις οντότητες του domain model, υλοποιώντας το Data Mapper pattern.



- **com.cse.traineeship.service**: Περιέχει τα interfaces των services και τις υλοποιήσεις τους (στο υποπακέτο impl). Οι κλάσεις Service ενσωματώνουν την επιχειρησιακή λογική της εφαρμογής, αποσυνδέοντάς την από τους Controllers και τα Repositories (Service Layer pattern).



- **com.cse.traineeship.service.strategy**: Περιέχει τις υλοποιήσεις των διαφορετικών στρατηγικών για την αναζήτηση θέσεων πρακτικής και την ανάθεση επιβλεπόντων καθηγητών, υλοποιώντας το **Strategy design pattern** (GoF). Αυτό ικανοποιεί τις απαιτήσεις NF1.2 και NF1.3 για επεκτασιμότητα και παραμετροποίηση των στρατηγικών αναζήτησης/ανάθεσης χωρίς αλλαγή στον κώδικα των services. Οι factories για τη δημιουργία των στρατηγικών βρίσκονται συνήθως μέσα στις κλάσεις service που τις χρησιμοποιούν (π.χ. CommitteeServiceImpl).



4.2 Design

CRC Cards

Class Name: User

- **Responsibilities:**
 - Διατηρεί βασικά στοιχεία χρήστη (username, password).
 - Διαχειρίζεται τον ρόλο του χρήστη (student, company, professor, committee).
- **Collaborations:**
 - Spring Security (UserDetails)
 - Role (Enum)

- (Specific user types like Student, Professor, etc., can inherit or be associated)

Class Name: Student

- **Responsibilities:**

- Διατηρεί πληροφορίες φοιτητή (όνομα, ΑΜ, δεξιότητες, ενδιαφέροντα, προτιμώμενη τοποθεσία).
- Υποβάλλει αίτηση για πρακτική.
- Συμπληρώνει logbook.
- Διαχειρίζεται την κατάσταση αναζήτησης πρακτικής (lookingForTraineeship).

- **Collaborations:**

- TraineeshipPosition (assignedTraineeship)
- CommitteeService (για υποβολή αίτησης)
- User (association/inheritance)

Class Name: Company

- **Responsibilities:**

- Διατηρεί πληροφορίες εταιρείας (company name, location).
- Δημιουργεί και διαχειρίζεται αγγελίες θέσεων πρακτικής.
- Αξιολογεί φοιτητές σε πρακτική.

- **Collaborations:**

- TraineeshipPosition (List of positions offered)
- Evaluation (για αξιολογήσεις φοιτητών)
- User (association/inheritance)

Class Name: Professor

- **Responsibilities:**

- Διατηρεί πληροφορίες καθηγητή (full name, interests).
- Επιβλέπει πρακτικές ασκήσεις.
- Αξιολογεί φοιτητές και εταιρείες.

- **Collaborations:**

- TraineeshipPosition (List of supervisedPositions)
- Evaluation (για αξιολογήσεις φοιτητών/εταιρειών)
- User (association/inheritance)

Class Name: AuthController - LoginController / RegistrationController

- **Responsibilities:**

- Διαχειρίζεται HTTP αιτήματα για την εγγραφή νέων χρηστών.

- Διαχειρίζεται HTTP αιτήματα για τη σύνδεση (login) και αποσύνδεση (logout) χρηστών.
- Εμφανίζει τις φόρμες εγγραφής και σύνδεσης.
- Επικυρώνει τα δεδομένα εισόδου για εγγραφή/σύνδεση.
- Καλεί μεθόδους του UserService για τη δημιουργία νέων χρηστών.
- **Collaborations:**
 - UserService (για τη δημιουργία και έλεγχο χρηστών).
 - User (ως ModelAttribute για τη φόρμα εγγραφής).
 - Validator (Spring Validator, για την επικύρωση δεδομένων φόρμας).
 - SecurityContextHolder (Spring Security, για logout).
 - Model (Spring MVC, για προσθήκη attributes στο view).

Class Name: TraineeshipPosition

- **Responsibilities:**
 - Εκπροσωπεί μια θέση πρακτικής.
 - Διατηρεί πληροφορίες θέσης (description, dates, topics, skills).
 - Διατηρεί την κατάσταση ανάθεσης (isAssigned).
 - Διατηρεί το logbook του φοιτητή και τον τελικό βαθμό (passFailGrade).
- **Collaborations:**
 - Company (who offers the position)
 - Student (who is assigned, if any)
 - Professor (who supervises, if any)
 - Evaluation (List of evaluations related to this position)

Class Name: Evaluation (Company - Professor)

- **Responsibilities:**
 - Αξιολογεί μια πτυχή της πρακτικής (φοιτητή από εταιρεία/καθηγητή, εταιρεία από καθηγητή).
 - Διατηρεί κριτήρια αξιολόγησης (motivation, efficiency, effectiveness, etc.).
 - Καθορίζει τον τύπο της αξιολόγησης.
- **Collaborations:**
 - TraineeshipPosition (the traineeship being evaluated)

Class Name: ApplicationService (Implementation ApplicationServiceImpl)

- **Responsibilities:**
 - Διαχειρίζεται τις αιτήσεις (Application) των φοιτητών για θέσεις πρακτικής.

- Δημιουργεί μια νέα αίτηση όταν ένας φοιτητής υποβάλλει ενδιαφέρον για μια θέση.
- Ανακτά τις αιτήσεις (π.χ., όλες οι ενεργές, αιτήσεις ενός φοιτητή, αιτήσεις για μια θέση).
- Ενημερώνει την κατάσταση μιας αίτησης.
- **Collaborations:**
 - ApplicationRepository, StudentRepository, TraineeshipPositionRepository (για τη σύνδεση της αίτησης με τη θέση).
 - Application (domain object).
 - Student (domain object).
 - TraineeshipPosition (domain object).

Class Name: CommitteeService (Implementation: CommitteeServiceImpl)

- **Responsibilities:**
 - Διαχειρίζεται τις αιτήσεις φοιτητών.
 - Αναθέτει θέσεις πρακτικής σε φοιτητές χρησιμοποιώντας στρατηγικές.
 - Αναθέτει επιβλέποντες καθηγητές σε πρακτικές χρησιμοποιώντας στρατηγικές.
 - Παρακολουθεί την πρόοδο και ολοκληρώνει τις πρακτικές.
- **Collaborations:**
 - StudentRepository, TraineeshipPositionRepository, ProfessorRepository
 - PositionsSearchFactory, SupervisorAssignmentFactory
 - IPositionsSearchStrategy, ISupervisorAssignmentStrategy (and their concrete implementations)
 - Student, Professor, TraineeshipPosition (domain objects)

Class Name: IPositionsSearchStrategy (Interface)

- **Responsibilities:**
 - Ορίζει τη διεπαφή για την αναζήτηση θέσεων πρακτικής για έναν αιτούντα.
- **Collaborations:**
 - TraineeshipPositionRepository, StudentRepository, CompanyRepository (μέσω των υλοποιήσεων)
- **Responsibilities:**
 - Υλοποιούν μια συγκεκριμένη λογική αναζήτησης θέσεων.
- **Collaborations:**

- TraineeshipPositionRepository, StudentRepository, CompanyRepository

Class Name: ISupervisorAssignmentStrategy (Interface)

- **Responsibilities:**
 - Ορίζει τη διεπαφή για την ανάθεση επιβλέποντα καθηγητή σε μια θέση πρακτικής.
- **Collaborations:**
 - TraineeshipPositionRepository, ProfessorRepository (μέσω των υλοποιήσεων)
- **Responsibilities:**
 - Υλοποιούν μια συγκεκριμένη λογική ανάθεσης επιβλέποντα.
- **Collaborations:**
 - TraineeshipPositionRepository, ProfessorRepository

5 Tests

Κατά τη διάρκεια των Sprints, αναπτύχθηκαν και εκτελέστηκαν εκτενείς δοκιμές που καλύπτουν όλους τους βασικούς τομείς της εφαρμογής, ακολουθώντας τις βέλτιστες πρακτικές και αξιοποιώντας τις δυνατότητες του Spring Boot Testing Framework, JUnit 5 και Mockito.

- **Controller Tests:** Έγινε χρήση του Spring Boot MockMvc για την επαλήθευση της σωστής λειτουργίας των endpoints των controllers. Ελέγχθηκε η σωστή δρομολόγηση των αιτημάτων, η διαχείριση των παραμέτρων, η κλήση των κατάλληλων service methods και η επιστροφή των αναμενόμενων views ή HTTP status codes. Παραδείγματα κλάσεων: AuthControllerTest, CompanyControllerTest, StudentControllerTest, ProfessorControllerTest, CommitteeControllerTest.
 - **AuthControllerTest.java**
 - **Περιγραφή:** Ελέγχει σωστή απόδοση του login και redirect στο home.
 - **User Story:** US-01 (Είσοδος χρήστη στο σύστημα)
 - **Test Cases:**
 - Επιστρέφει τη σελίδα login.
 - Επιστρέφει τη σελίδα home μετά την επιτυχή σύνδεση.
 - **LoginControllerTest.java**
 - **Περιγραφή:** Παρόμοια λογική με το AuthController – πιθανόν να έχει συγχωνευθεί.
 - **User Story:** US-01 (Είσοδος)

- **RegistrationControllerTest.java**
 - **Περιγραφή:** Ελέγχει εμφάνιση register page για κάθε ρόλο.
 - **User Story:** US-02 (Εγγραφή νέου χρήστη)
 - **Test Cases:**
 - showRegisterForm() για STUDENT / COMPANY / PROFESSOR

- **CommitteeControllerTest.java**
 - **Περιγραφή:** Ελέγχει τις ενέργειες του ρόλου COMMITTEE.
 - **User Stories:**
 - US-10: Αναζήτηση και Αντιστοίχιση Φοιτητών
 - US-11: Ανάθεση Επιβλέποντα
 - US-12: Τελική Απόφαση
 - **Test Cases:**
 - listApplicants()
 - processSearch()
 - assignPosition()

- **CompanyControllerTest.java**
 - **Περιγραφή:** Ελέγχει την προβολή/επεξεργασία προφίλ εταιρείας και θέσεων.
 - **User Story:** US-06 (Διαχείριση Θέσεων Πρακτικής από Εταιρεία)
 - **Test Cases:**
 - viewProfile()
 - updateProfile()
 - viewOwnPositions()

- **LogbookControllerTest.java**
 - **Περιγραφή:** Ελέγχει καταχωρήσεις στο logbook του φοιτητή.
 - **User Story:** US-09 (Καταγραφή δραστηριοτήτων)
 - **Test Cases:**
 - listPositions() – επιστρέφει τις θέσεις του φοιτητή
 - showEntries() – δείχνει καταχωρήσεις
 - addEntry() – προσθέτει νέα εγγραφή

- **PositionControllerTest.java**

- **Περιγραφή:** Διαχειρίζεται θέσεις πρακτικής – δημιουργία, επεξεργασία, ανάθεση.
 - **User Stories:**
 - US-06 (Διαχείριση Θέσεων από Εταιρεία)
 - US-10 (Αναζήτηση Θέσεων από Φοιτητή)
 - **Test Cases:**
 - list(), showNewForm(), save(), apply()
- **PositionControllerTestSec.java**
 - **Περιγραφή:** Επιπλέον tests για security access (π.χ., roles).
 - **User Story:** US-15 (Ασφάλεια Πρόσβασης ανά Ρόλο)
 - **Test Cases:**
 - Πιθανό redirect αν κάποιος unauthorized προσπαθήσει πρόσβαση.
 - **ProfessorControllerTest.java**
 - **Περιγραφή:** Ελέγχει την προβολή/αξιολόγηση από τον καθηγητή.
 - **User Story:** US-13 (Επιβλέπων καθηγητής – αξιολόγηση)
 - **Test Cases:**
 - viewProfile(), updateProfile(), viewAssignedPositions()
 - **ProfessorEvaluationControllerTest.java**
 - **Περιγραφή:** Ελέγχει την καταχώρηση αξιολόγησης από καθηγητή.
 - **User Story:** US-13 (Αξιολόγηση από καθηγητή)
 - **Test Cases:**
 - showForm(), submitEvaluation()
- **Repository Tests:** Ελέγχθηκαν οι βασικές CRUD (Create, Read, Update, Delete) λειτουργίες για κάθε JPA Repository interface. Χρησιμοποιήθηκε η annotation `@DataJpaTest` για τη φόρτωση μόνο του απαραίτητου context για την αλληλεπίδραση με την ενσωματωμένη βάση δεδομένων H2. Ελέγχθηκε η σωστή αποθήκευση, ανάκτηση και διαγραφή οντοτήτων. Αυτά τα τεστ λόγω χρήσης SQL στον τελικό κώδικα αφαιρέθηκαν για να μην έχουμε conflicts αφήνοντας μόνο ένα βασικό “testSaveApplicationFailsWithoutStudentAndPosition”. Παραδείγματα κλάσεων: `UserRepositoryTest`, `StudentRepositoryTest`, `TraineeshipPositionRepositoryTest`.

- **ApplicationRepositoryTest**

- **User Story:** US-15: Ως Επιτροπή, θέλω να μπορώ να αναθέτω φοιτητές σε θέσεις ώστε να προκύπτουν ορισμένες θέσεις πρακτικής.
- **Τύπος Τεστ:** Repository Test με @DataJpaTest
- **Περιγραφή:**
 - Ελέγχει ότι αποτυγχάνει η αποθήκευση Application όταν λείπουν υποχρεωτικές συσχετίσεις (Student και Position).
 - Επαληθεύει βασική ακεραιότητα οντοτήτων και σχέσεων.
 - Επιβεβαιώνει τη σωστή συμπεριφορά του ApplicationRepository.

Τα υπόλοιπα repository tests (CompanyEvaluationRepositoryTest, LogbookEntryRepositoryTest, ProfessorEvaluationRepositoryTest) **παραμένουν αλλά είναι κενά**, για λόγους πληρότητας αρχείων.

- **Service Tests:** Έγινε εκτεταμένη χρήση mocking με το Mockito για να διασφαλιστεί η ορθότητα της επιχειρησιακής λογικής στις κλάσεις Service, χωρίς άμεση εξάρτηση από την πραγματική βάση δεδομένων ή άλλα εξωτερικά συστήματα. Τα Repositories και άλλα εξαρτώμενα services έγιναν mock. Ελέγχθηκε η σωστή αλληλεπίδραση με τα mocks και η επιστροφή των αναμενόμενων αποτελεσμάτων. Παραδείγματα κλάσεων: UserServiceImplTest, CompanyServiceImplTest, StudentServiceImplTest, CommitteeServiceImplTest.

- **CompanyEvaluationServiceImplTest.java**

- **Περιγραφή:** Ελέγχει τη λειτουργία αποθήκευσης και ανάκτησης αξιολογήσεων από την εταιρεία.
- **User Story:**
 - US-18: Αξιολόγηση φοιτητή από την εταιρεία
- **Test Cases:**
 - testSaveEvaluation()
 - testFindByPositionId()

- **CompanyServiceImplTest.java**

- **Περιγραφή:** Ελέγχει την εύρεση και αποθήκευση στοιχείων εταιρείας.
- **User Stories:**
 - US-2: Εγγραφή/Είσοδος εταιρείας
 - US-3: Προβολή / Επεξεργασία προφίλ
- **Test Cases:**
 - testFindByUsername()
 - testFindAll()
 - testSaveCompany()

- **LogbookServiceImplTest.java**
 - **Περιγραφή:** Ελέγχει την προσθήκη και εμφάνιση καταχωρήσεων logbook.
 - **User Story:**
 - US-14: Εισαγωγή Καταχώρησης
 - **Test Cases:**
 - testAddEntry()
 - testFindEntriesForPosition()
- **4. PositionServiceImplTest.java**
 - **Περιγραφή:** Ελέγχει τη διαχείριση θέσεων πρακτικής: δημιουργία, ανάθεση, ανάκτηση.
 - **User Stories:**
 - US-6: Δημιουργία / Επεξεργασία Θέσης
 - US-7: Προβολή Διαθέσιμων Θέσεων
 - US-8: Διαγραφή Θέσης
 - **Test Cases:**
 - testFindAll()
 - testFindByld()
 - testSave()
 - testDeleteByld()
 - testApply()
 - testFindAllByStudentld()
- **5. ProfessorEvaluationServiceImplTest.java**
 - **Περιγραφή:** Ελέγχει τη λειτουργία αποθήκευσης και ανάκτησης αξιολογήσεων από τον καθηγητή.
 - **User Story:**
 - US-19: Αξιολόγηση εταιρείας από επιβλέποντα
 - **Test Cases:**
 - testSaveEvaluation()
 - testFindByPositionld()
- **ProfessorServiceImplTest.java**
 - **Περιγραφή:** Ελέγχει την ανάκτηση στοιχείων καθηγητή από βάση δεδομένων.
 - **User Story:**

- US-4: Προβολή / Επεξεργασία Προφίλ
- **Test Cases:**
- testFindByUsername()
- testFindAll()
- testSave()
- **StudentServiceImplTest.java**
 - **Περιγραφή:** Ελέγχει την ανάκτηση και αποθήκευση στοιχείων φοιτητή.
 - **User Stories:**
 - US-1: Εγγραφή/Είσοδος Φοιτητή
 - US-13: Επεξεργασία Προφίλ
 - **Test Cases:**
 - testFindByUsername()
 - testFindAll()
 - testUpdateProfile()
- **UserServiceImplTest.java**
 - **Περιγραφή:** Ελέγχει τη διαχείριση χρηστών και τον έλεγχο username.
 - **User Story:**
 - US-0: Εγγραφή χρήστη
 - **Test Cases:**
 - testExistsByUsername()
 - testCreateUser()
 - testCreateUserThrowsExceptionWhenExists()
 - testFindByUsername()
- **ApplicationServiceImplTest.java**
 - **Περιγραφή:** Ελέγχει τη λειτουργία αποθήκευσης και εύρεσης αιτήσεων πρακτικής.
 - **User Story:**
 - US-9: Υποβολή Αίτησης από Φοιτητή
 - **Test Cases:**
 - testSave()
 - testFindAll()
 - testSaveApplicationFailsWithoutStudentAndPosition()

- **CommitteeServiceImplTest.java**
 - **Περιγραφή:** Ελέγχει τις ενέργειες της επιτροπής για αναθέσεις, τελικές αποφάσεις και αναζητήσεις.
 - **User Stories:**
 - US-10: Αναζήτηση και Αντιστοίχιση Φοιτητών
 - US-11: Ανάθεση Επιβλέποντα
 - US-12: Τελική Απόφαση
 - **Test Cases:**
 - testGetAllApplicants()
 - testSearchPositions()
 - testAssignPosition()
 - testAssignSupervisor()
 - testFinalizePosition()
 - testGetInProgressPositions()
 - testGetCompletedPositions()
- **Strategy Tests:** Υλοποιήθηκαν αυτόνομα tests για κάθε συγκεκριμένη υλοποίηση των στρατηγικών (τόσο για την αναζήτηση θέσεων όσο και για την ανάθεση επιβλεπόντων). Αυτά τα tests εξασφάλισαν ότι κάθε στρατηγική λειτουργεί σωστά με βάση τα κριτήριά της και ότι η εναλλαξιμότητά τους (interchangeability) διατηρείται. Χρησιμοποιήθηκε mocking για τις εξαρτήσεις από repositories.
Παραδείγματα: InterestBasedStrategyTest, LocationBasedStrategyTest, LoadBasedProfessorStrategyTest.
 - **InterestBasedStrategyTest**
 - Τι κάνει: Ελέγχει ότι η στρατηγική συμφωνίας ενδιαφερόντων επιστρέφει τις σωστές θέσεις για έναν φοιτητή.
 - Mocked: Θέσεις και φοιτητής με συγκεκριμένα interests.
 - Έλεγχος: Επιστρέφονται μόνο θέσεις με κοινά interests άνω του ορίου.
 - Κάλυψη: Interest-based στρατηγική στο CommitteeServiceImpl.
 - **LocationBasedStrategyTest**
 - Τι κάνει: Επαληθεύει ότι η στρατηγική βασισμένη στην τοποθεσία επιστρέφει θέσεις στην ίδια πόλη με τον φοιτητή.
 - Mocked: Θέσεις και φοιτητής με preferredLocation.
 - Έλεγχος: Φιλτράρει σωστά μόνο τις αντίστοιχες τοποθεσίες.
 - Κάλυψη: Χρήση στο searchPositions της επιτροπής.

- **CombinedStrategyTest**
 - Τι κάνει: Συνδυάζει το location και τα interests και ελέγχει αν τα ζυγίζει σωστά για να φέρει τις πιο κατάλληλες θέσεις.
 - Mocked: Θέσεις με διαφορές σε location & interests.
 - Έλεγχος: Η βαθμολογία αξιολόγησης (combined score) υπολογίζεται σωστά και οι θέσεις ταξινομούνται αντίστοιχα.
 - Κάλυψη: Χρήση στην επιλογή στρατηγικής από την επιτροπή.
- **LoadBasedProfessorStrategyTest**
 - Τι κάνει: Ελέγχει ότι η ανάθεση επιβλέποντα γίνεται στον καθηγητή με το μικρότερο φόρτο (πλήθος εποπτευόμενων).
 - Mocked: Καθηγητές με διαφορετικά πλήθη εποπτευόμενων.
 - Έλεγχος: Επιλέγεται πάντα ο με το λιγότερα supervised positions.
 - Κάλυψη: assignSupervisor του CommitteeServiceImpl.
- **Integration Tests:** Καλύφθηκαν βασικά end-to-end σενάρια της εφαρμογής, από το αίτημα HTTP μέχρι την αλληλεπίδραση με τη βάση δεδομένων (χρησιμοποιώντας @SpringBootTest με WebEnvironment.RANDOM_PORT ή MOCK). Αυτά τα tests επαληθεύουν τη σωστή συνεργασία μεταξύ των διαφόρων επιπέδων της εφαρμογής (Controllers, Services, Repositories). Μια κλάση BasicEndpointsTest μπορεί να χρησιμοποιηθεί για να ελέγξει την προσβασιμότητα βασικών σελίδων.
 - **Integration Test**
 - **Κλάση:** BasicEndpointsTest
 - @SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
 - **Σκοπός:** Ελέγχει ότι βασικές σελίδες της εφαρμογής (login, register, home) είναι προσβάσιμες με HTTP requests.
 - **User Story:** US-STUDENT-1: Εγγραφή φοιτητή US-COMPANY-1: Προσπέλαση dashboard, US-GENERIC-0: Login/Logout
 - **Έλεγχος:** Παρουσίαση HTML περιεχομένου Status code 200 για /, /login, /register, κ.ά.
- **Application Boot Test:** Η κλάση TraineeshipManagementApplicationTests (που δημιουργείται αυτόματα από το Spring Initializr) επιβεβαιώνει ότι το Spring context της εφαρμογής φορτώνει σωστά χωρίς σφάλματα κατά την εκκίνηση.
 - **Application Boot Test**
 - **Κλάση:** TraineeshipManagementApplicationTests
 - @SpringBootTest

- **Σκοπός:** Επιβεβαιώνει ότι το Spring context φορτώνεται σωστά και η εφαρμογή μπορεί να ξεκινήσει χωρίς exceptions.
- **User Story:** Όλα τα σενάρια λειτουργίας εξαρτώνται από επιτυχή εκκίνηση (γενική προϋπόθεση του συστήματος).
- **Έλεγχος:** Δεν υπάρχουν σφάλματα φόρτωσης beans ή conflicts στο context.

Όλα τα tests γράφτηκαν με **JUnit 5** και **Mockito**. Για τους Controller tests, έγινε χρήση της annotation `@WebMvcTest` και για τα Repository tests της `@DataJpaTest`, ώστε να φορτώνεται μόνο το απαραίτητο slice του application context, κάνοντας τα tests ταχύτερα και πιο απομονωμένα.