

Domain-Model-v0.2



Όνομα Έργου: Train-Up

Κωδικός: Domain-Model

Έκδοση: v0.2

Μέλη Ομάδας

- Γεώργιος Αυγεράκης, AM: 1067441.
- Φωτεινή Παπαζαφειρίου, AM: 1071399.
- Όλγα Μελιάδη, AM: 1080448.
- Μαρία Γκότση, AM: 1064283.
- Κωνσταντίνος Κωνσταντάτος, AM: 1067461.

Κατανομή Ρόλων στο παρόν τεχνικό κείμενο

- Γιώργος Αυγεράκης: Contributor, Peer Reviewer
- Φωτεινή Παπαζαφειρίου: Contributor, Peer Reviewer
- Όλγα Μελιάδη: Contributor, Peer Reviewer
- Μαρία Γκότση: Contributor, Peer Reviewer
- Κωνσταντίνος Κωνσταντάτος: Contributor, Peer Reviewer

Στο παρόν τεχνικό κείμενο όλα τα μέλη της ομάδας συνεισέφεραν εξίσου στην περιγραφή των κλάσεων του έργου.

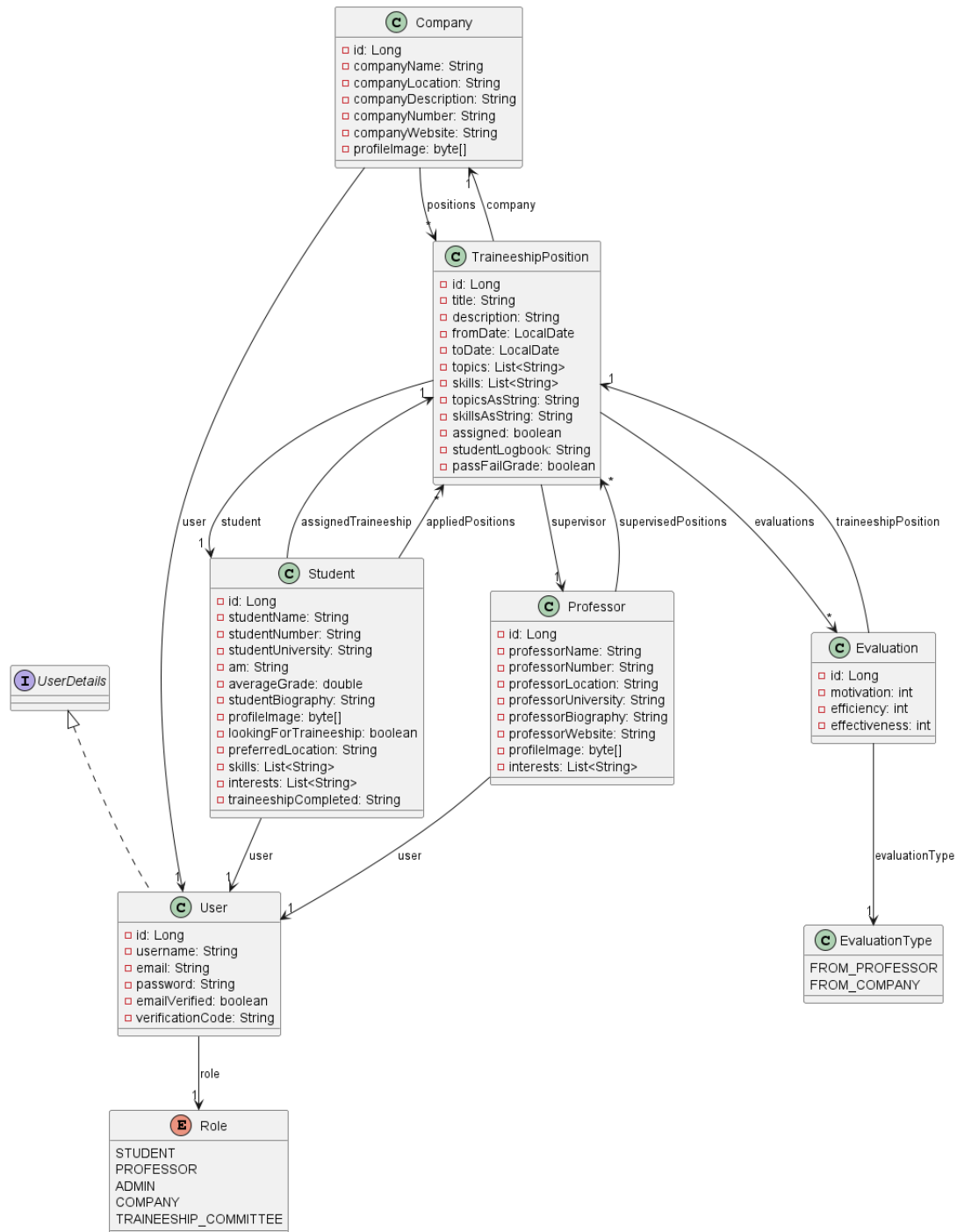
Ο σύνδεσμος για το repository της ομάδας μας στο Github είναι [εδώ](#).

Αλλαγές στην νέα Έκδοση

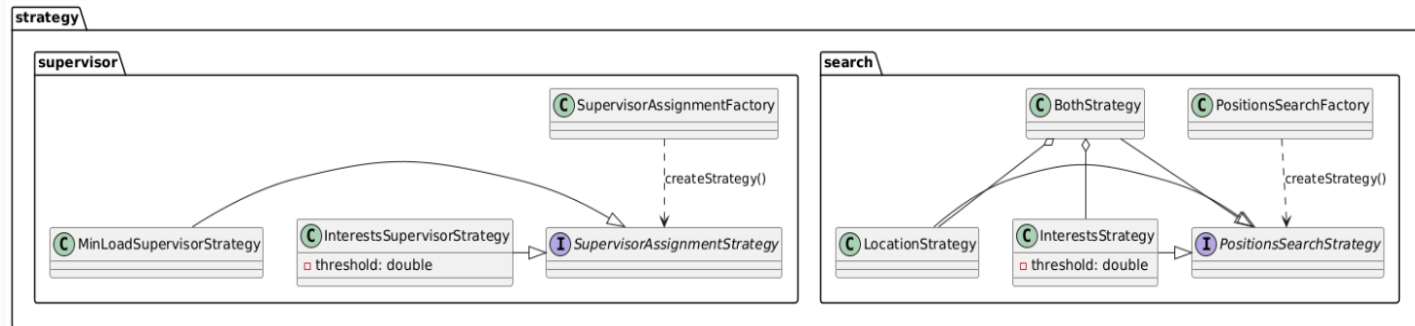
Έχουμε προσθέσει στο domain model πεδία στις κλάσεις και την στρατηγική αναζήτησης την αναλύουμε ξεχωριστά καθώς όλα αυτά τα διαγράμματα θα «ενωθούν» αργότερα σε μια τελική μορφή συσχέτισης στο project. [Οι αλλαγές δεν επισημαίνονται με μπλε χρώμα παρακάτω καθώς έχουν γίνει σε όλα τα σημεία της αναφοράς με πιο αναλυτικό τρόπο](#)

Διάγραμμα Domain Model

Στο σχήμα της εικόνας 1 παρουσιάζουμε το διάγραμμα κλάσεων του Domain Model. Στην παρούσα έκδοση του συγκεκριμένου τεχνικού κειμένου και στην εικόνα 2 την λογική της στρατηγικής αναζήτησης καθηγητή και θέσεων πρακτικής άσκησης.



Εικόνα 1 Domain Model Diagram



Εικόνα 2 Διάγραμμα Στρατηγικής

Περιγραφή κλάσεων

Στην παρακάτω περιγραφή των κλάσεων στοχεύουμε να φανούν οι βασικές υποψήφιες κλάσεις. Καθώς θα προχωρήσει η ανάλυση και η υλοποίηση, ενδέχεται να προστεθούν ή να αφαιρεθούν κλάσεις ή να γίνουν πιο εξειδικευμένες οι σχέσεις.

➤ User

Η βασική οντότητα που αναπαριστά κάθε εγγεγραμμένο χρήστη της πλατφόρμας.

- **Πεδία:** id, username, email, password, emailVerified, verificationCode.
- **Σχέσεις:**
 - 1–1 με Role (για τον ρόλο: STUDENT, PROFESSOR, COMPANY, ADMIN, TRAINEESHIP_COMMITTEE).
 - 1–1 με Student ή Professor ή Company μέσω των associations “user” (καθώς κάθε ειδικός τύπος χρήστη έχει και εγγραφή σε User).
- **Ρόλος στα Use Cases:** Κεντρικός κόμβος για σύνδεση/εγγραφή, έλεγχο δικαιωμάτων και πλοήγηση ανάλογα με τον ρόλο.

➤ **Role**

Enum που καθορίζει τον τύπο/εξουσιοδότηση ενός User.

- **Τιμές:**
 - STUDENT
 - PROFESSOR
 - COMPANY
 - ADMIN
 - TRAINEESHIP_COMMITTEE
- **Χρήση:** Ελέχει ποια μενού και ποιες λειτουργίες είναι προσβάσιμες σε κάθε χρήστη.

➤ **Student**

Εξειδίκευση του User για φοιτητές.

- **Πεδία:** studentName, studentNumber, studentUniversity, am, averageGrade, studentBiography, profileImage, lookingForTraineeship, preferredLocation, skills, interests, traineeshipCompleted, κ.ά.
- **Σχέσεις:**
 - 1–N με TraineeshipPosition (μέσω appliedPositions) — οι θέσεις που υπέβαλε αίτηση.
 - 1–1 με TraineeshipPosition (μέσω assignedTraineeship) — η θέση που του ανατέθηκε.
 - 1–N με Evaluation (μέσω evaluations) — οι αξιολογήσεις που έλαβε.
- **Ρόλος στα Use Cases:** Υποβολή αιτήσεων, ενημέρωση logbook, προβολή αποτελεσμάτων και προφίλ.

➤ **Professor**

Εξειδίκευση του User για καθηγητές-επόπτες.

- **Πεδία:** professorName, professorNumber, professorLocation, professorUniversity, professorBiography, profileImage, interests.
- **Σχέσεις:**
 - 1–N με TraineeshipPosition (μέσω supervisedPositions) — οι θέσεις που επιβλέπει.
 - 1–N με Evaluation (μέσω evaluations) — οι αξιολογήσεις που έχει κάνει.
- **Ρόλος στα Use Cases:** Αξιολόγηση φοιτητών, προβολή θέσεων υπό επίβλεψη και ενημέρωση προφίλ.

➤ **Company**

Εξειδίκευση του User για εταιρείες.

- **Πεδία:** companyName, companyLocation, companyDescription, companyNumber, companyWebsite, profileImage.
- **Σχέσεις:**
 - 1–N με TraineeshipPosition (μέσω positions) — οι θέσεις πρακτικής που έχει αναρτήσει.
- **Ρόλος στα Use Cases:** Ανάρτηση, τροποποίηση και διαγραφή θέσεων πρακτικής, προβολή προφίλ εταιρείας.

➤ **TraineeshipPosition**

Αντικείμενο που αναπαριστά μια διαθέσιμη ή ανατεθειμένη θέση πρακτικής.

- **Πεδία:** title, description, fromDate, toDate, topics, skills, assigned (flag), studentLogbook, passFailGrade, κ.ά.
- **Σχέσεις:**
 - N–1 με Company (η εταιρεία που την προσφέρει).

- N-1 με Student (μέσω assignedTraineeship) — ο φοιτητής που την έχει αναλάβει.
- N-1 με Professor (μέσω supervisor) — ο επιβλέπων καθηγητής.
- 1-N με Evaluation (μέσω evaluations) — οι αξιολογήσεις εταιρείας και καθηγητή.
- **Ρόλος στα Use Cases:** Κέντρο όλων των αλληλεπιδράσεων: αναζήτηση-φιλτράρισμα (strategies), υποβολή αιτήσεων, ανάθεση, ενημέρωση logbook, αξιολογήσεις.

➤ Evaluation

Καταγράφει το αποτέλεσμα μιας αξιολόγησης φοιτητή σε θέση πρακτικής.

- **Πεδία:** motivation, efficiency, effectiveness.
- **Σχέσεις:**
 - N-1 με TraineeshipPosition (σε ποια θέση αναφέρεται).
 - N-1 με EvaluationType (προέλευση: FROM_PROFESSOR ή FROM_COMPANY).
- **Ρόλος στα Use Cases:** Αποθήκευση και προβολή βαθμολογιών από καθηγητή ή εταιρεία, απαραίτητο για την τελική “finalize traineeship”.

➤ EvaluationType

Enum με τις δύο πηγές αξιολόγησης:

- FROM_PROFESSOR
- FROM_COMPANY
Χρησιμοποιείται για να ξεχωρίζουν οι αξιολογήσεις κατά προβολή και λογική ολοκλήρωσης.

Οι παρακάτω κλάσεις υπάρχουν από την προηγούμενη έκδοση και θα αναλυθούν με διάγραμμα πιο αναλυτικά σε μεταγενέστερη έκδοση του μοντέλου για να υπάρχει μια καλύτερη ανάλυση περι των service. (Σαν ιδέα όμως παραμένει ίδια για το μοντέλο μας)

- **UserService:** Κλάση που παρέχει λειτουργίες για εγγραφή (signup), σύνδεση (login) και έξοδο (logout) των οντοτήτων **User**.
- **CompanyService:** Κλάση που είναι υπεύθυνη για τη διαχείριση του προφίλ των εταιρειών, τη δημιουργία νέων θέσεων πρακτικής, την προβολή των διαθέσιμων και των ανατεθειμένων θέσεων, καθώς και την αξιολόγηση φοιτητών που εργάζονται σε αυτές τις θέσεις.
- **StudentService:** Κλάση που παρέχει λειτουργίες για τη δημιουργία και ενημέρωση του προφίλ ενός φοιτητή, την αναζήτηση/εμφάνιση διαθέσιμων θέσεων, την υποβολή αίτησης (apply) σε θέση πρακτικής, καθώς και την ενημέρωση ενός logbook (ημερολογίου πρακτικής).
- **ProfessorService:** Κλάση που είναι υπεύθυνη για τη δημιουργία/ενημέρωση του προφίλ καθηγητή και την αξιολόγηση των φοιτητών σε θέσεις που επιβλέπει.
- **CommitteeService:** Κλάση που αφορά την επιτροπή πρακτικής. Παρέχει λειτουργίες για την ανάθεση θέσεων πρακτικής σε φοιτητές (βάσει διαφορετικών κριτηρίων), την ανάθεση καθηγητών ως επιβλεπόντων (επίσης με εναλλακτικές στρατηγικές) και, τέλος, την ολοκλήρωση ή ακύρωση μιας θέσης πρακτικής (pass/fail).

Στρατηγική Ανάλυση

Η ανάθεση των θέσεων στους μαθητές πραγματοποιείται μέσω εναλλακτικών στρατηγικών που λαμβάνουν υπόψη τις ενδιαφέροντα των μαθητών, την προτιμώμενη θέση ή και τα δύο. Για να γίνουν οι εναλλακτικές στρατηγικές εναλλάξιμες και να διευκολύνουμε την επέκταση της εφαρμογής με περισσότερες στρατηγικές, βασιζόμαστε στο πρότυπο στρατηγικής GoF, δηλ. οι στρατηγικές ανάθεσης θέσης υλοποιούνται σε διαφορετικές κλάσεις που παρέχουν την ίδια διεπαφή. Η υπηρεσία επιτροπής δημιουργεί την κατάλληλη στρατηγική χρησιμοποιώντας ένα παραμετροποιημένο "factory". Η ανάθεση καθηγητών σε θέσεις πρακτικής άσκησης υλοποιείται επίσης μέσω εναλλακτικών στρατηγικών που λαμβάνουν υπόψη τα ενδιαφέροντα ή τον φόρτο εργασίας των καθηγητών (Εικόνα 2). Όπως και προηγουμένως, για να γίνουν οι εναλλακτικές στρατηγικές εναλλάξιμες και να διευκολύνουμε την επέκταση της εφαρμογής με περισσότερες στρατηγικές, βασιζόμαστε στο GoF μοτίβο στρατηγικών, δηλαδή οι στρατηγικές ανάθεσης καθηκόντων επόπτη υλοποιούνται σε διαφορετικές κλάσεις που παρέχουν την ίδια διεπαφή.

Συγκεκριμένα:

1. Στρατηγική αναζήτησης πρακτικής θέσης από φοιτητή:

- **PositionsSearchStrategy** (interface): Ορίζει τη μέθοδο `searchPositions(...)` για φιλτράρισμα λίστας διαθέσιμων θέσεων ανά φοιτητή.
- **InterestsStrategy**: Υλοποιεί `PositionsSearchStrategy` φιλτράροντας θέσεις με βάση το “similarity” ανάμεσα στα topics της θέσης και τα interests του φοιτητή. Ο παράγων `threshold` καθορίζει το ελάχιστο όριο ομοιότητας.
- **LocationStrategy**: Υλοποιεί `PositionsSearchStrategy` φιλτράροντας θέσεις με βάση τη γεωγραφική τοποθεσία της εταιρείας σε σχέση με την `preferredLocation` του φοιτητή.
- **BothStrategy**: Υλοποιεί `PositionsSearchStrategy` συνδυάζοντας δύο άλλες στρατηγικές (`Interests` + `Location`), επιστρέφοντας μόνο τις θέσεις που πληρούν και τα δύο κριτήρια (τομή αποτελεσμάτων).
- **PositionsSearchFactory**: Στατικό “εργοστάσιο”: αναλαμβάνει, βάσει ενός αλφαριθμητικού `strategyType`, να επιστρέψει την κατάλληλη υλοποίηση `PositionsSearchStrategy` (`INTERESTS`, `LOCATION` ή `BOTH`).

2. Στρατηγική αναζήτησης επιβλέποντα καθηγητή σε κάποια θέση από τον «committee»

- **SupervisorAssignmentStrategy(interface)**: Ορίζει τη μέθοδο `assign(...)` που επιλέγει καθηγητή για μια θέση πρακτικής.
- **InterestsSupervisorStrategy**: Υλοποιεί `SupervisorAssignmentStrategy` επιλέγοντας τον καθηγητή με τη μεγαλύτερη `similarity` ανάμεσα στα interests του καθηγητή και στα topics της θέσης, πάνω από ένα όριο `threshold`.
- **MinLoadSupervisorStrategy**: Υλοποιεί `SupervisorAssignmentStrategy` αναθέτοντας τον καθηγητή με τον ελάχιστο τρέχοντα φόρτο (μικρότερος αριθμός ήδη επιβλεπόμενων θέσεων).
- **SupervisorAssignmentFactory**: Στατικό “εργοστάσιο”: δημιουργεί την κατάλληλη `SupervisorAssignmentStrategy` (π.χ. `INTERESTS` ή `MIN_LOAD`) με βάση παράμετρο `strategyType`, κρύβοντας την λογική επιλογής από τον καλούντα.

Εργαλεία που χρησιμοποιήθηκαν

Η συγγραφή του παρόντος τεχνικού κειμένου έγινε με την χρήση του Microsoft Word. Το “Domain Model Diagram” υλοποιήθηκε με τη χρήση του εργαλείου plant-uml.