

Εργασία Τεχνολογίας Λογισμικού

Δημήτρης Λάμπρου (Π19086)

Γεώργιος Στριμμένος (Π19164)

Άρμπερ Τζίτζο (Π19243)

Η εργασία έχει υλοποιηθεί σε Rails (MVC αρχιτεκτονική).

source: https://github.com/mpatsia/thesis_management

Στόχοι & Πρόβλημα

Ο βασικός στόχος είναι η ανάπτυξη μιας εφαρμογής που θα λειτουργεί ως διαχειριστικό εργαλείο. Η εφαρμογή θα πρέπει να επιτρέπει την καταχώρηση πληροφοριών σχετικά με τους φοιτητές και τις πτυχιακές εργασίες τους, συμπεριλαμβανομένων των προσωπικών στοιχείων, τίτλου της εργασίας, ημερομηνίας έναρξης, γλωσσών υλοποίησης ή γλώσσας. Η εφαρμογή θα πρέπει να επιτρέπει τη διαχείριση συναντήσεων επίβλεψης των εργασιών από τους καθηγητές (προφανώς remote αλλά όχι μέσω της εφαρμογής). Οι φοιτητές θα πρέπει να μπορούν να ανεβάζουν αρχεία σε μορφή zip. Η εφαρμογή θα πρέπει να επιτρέπει την καταχώρηση του τελικού βαθμού για κάθε πτυχιακή εργασία ενός φοιτητή. Οι καθηγητές θα πρέπει να μπορούν να παρακολουθούν τη συνολική πορεία του κάθε φοιτητή μέσω της εφαρμογής.

Παρουσίαση της RUP

Το **RUP (Rational Unified Process)** είναι ένα μοντέλο ανάπτυξης λογισμικού που προτείνεται για την ανάπτυξη πολύπλοκων και μεγάλης κλίμακας συστημάτων. Αποτελεί ένα πλαίσιο μεθοδολογίας που προωθεί την εφαρμογή βέλτιστων πρακτικών σε όλο τον κύκλο ζωής του λογισμικού. Το RUP βασίζεται σε αρχές όπως η ανάλυση και σχεδιασμός προτού η ανάπτυξη ξεκινήσει, η επανεξέταση και προσαρμογή των διαδικασιών κατά τη διάρκεια του έργου, και η εστίαση στην παραγωγή υψηλής ποιότητας λογισμικού. Το RUP οργανώνεται γύρω από τρεις βασικές φάσεις: τη φάση του ξεκινήματος, τη φάση της κατασκευής και τη φάση της μετάδοσης. Κάθε φάση περιλαμβάνει συγκεκριμένες δραστηριότητες και αρχιτεκτονικές που βοηθούν στην επίτευξη των στόχων του έργου. Το RUP είναι ευέλικτο και προσαρμόσιμο σε διάφορα είδη έργων και ομάδες ανάπτυξης. Το μοντέλο αυτό προωθεί την συνεργασία, την ποιότητα και τη διαχείριση των κινδύνων, κάνοντας το κατάλληλο για προηγμένες ανάπτυξης λογισμικού.

Requirements Analysis (Καταγραφή απαιτήσεων)

Πιο τεχνικό από το Στόχοι & Πρόβλημα.

– Απαιτήσεις (Περιγραφή)

Η εφαρμογή θα παρέχει λειτουργίες για την καταχώρηση προσωπικών στοιχείων φοιτητών, λεπτομερειών πτυχιακών εργασιών, διαχείρισης συναντήσεων, καταχώρησης αρχείων, και σποβολή βαθμολογιών. Θα παρέχει επίσης λειτουργίες παρακολούθησης της πορείας των φοιτητών.

– Λεπτομερείς Απαιτήσεις

α) Η εφαρμογή θα επιτρέπει την καταχώρηση προσωπικών στοιχείων φοιτητών, τίτλου πτυχιακής, ημερομηνίας έναρξης, γλωσσών υλοποίησης, τεχνολογιών, και άλλων σχετικών πληροφοριών. **β)** Οι καθηγητές θα έχουν τη δυνατότητα να καταχωρούν συναντήσεις επίβλεψης για τις πτυχιακές των φοιτητές. **γ)** Θα παρέχει δυνατότητα καταχώρησης αποτελεσμάτων από αυτές τις εργασίες τους. **δ)** Οι φοιτητές θα μπορούν να ανεβάζουν αρχεία στην εφαρμογή, προτιμητέο σε μορφή zip. **ε)** Οι καθηγητές θα μπορούν να παρακολουθούν την συνολική πορεία των φοιτητών, συμπεριλαμβανομένης της κατάστασης των εργασιών.

Διαγράμματα

Φάση 1η:

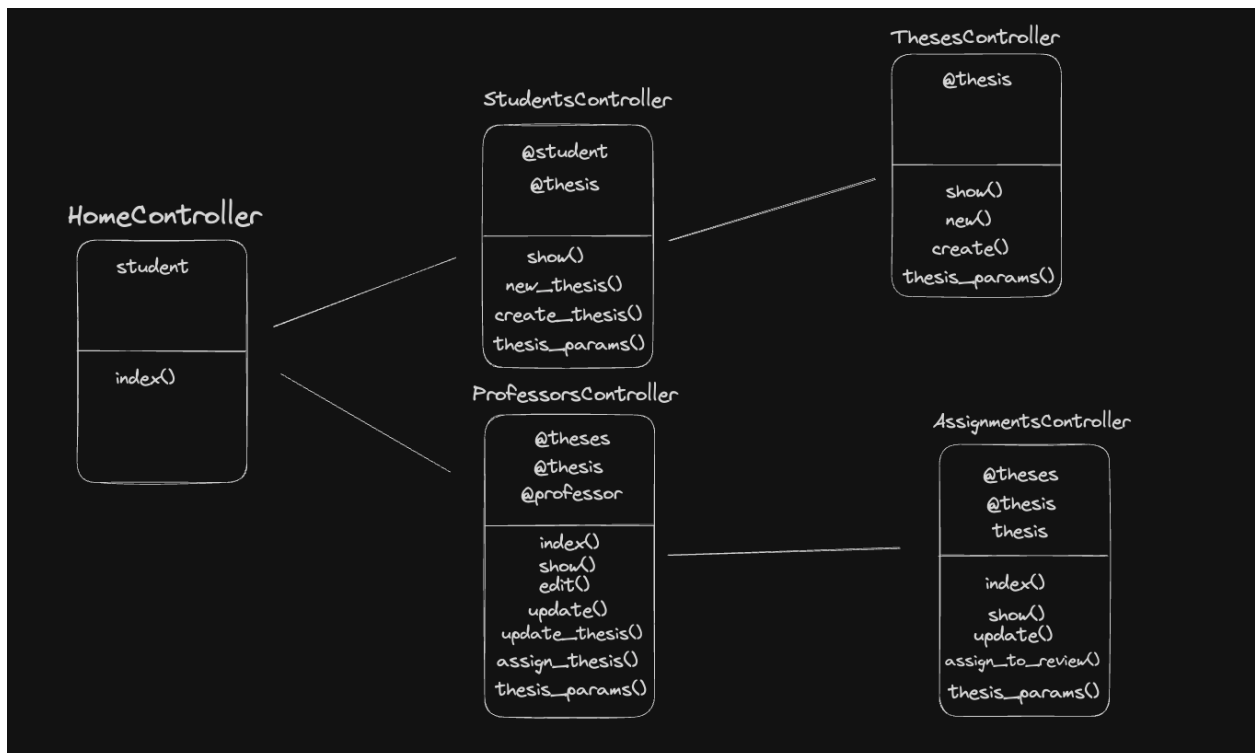
Use Case Diagrams

Τα διαγράμματα περιπτώσεων χρήσης (Use Case Diagrams) αναπαριστούν τις διάφορες ενέργειες και αλληλεπιδράσεις μεταξύ των διαφόρων ρόλων ("roles" ή "actors") και των διαφόρων περιπτώσεων που συμβαίνουν σε ένα πρόγραμμα.



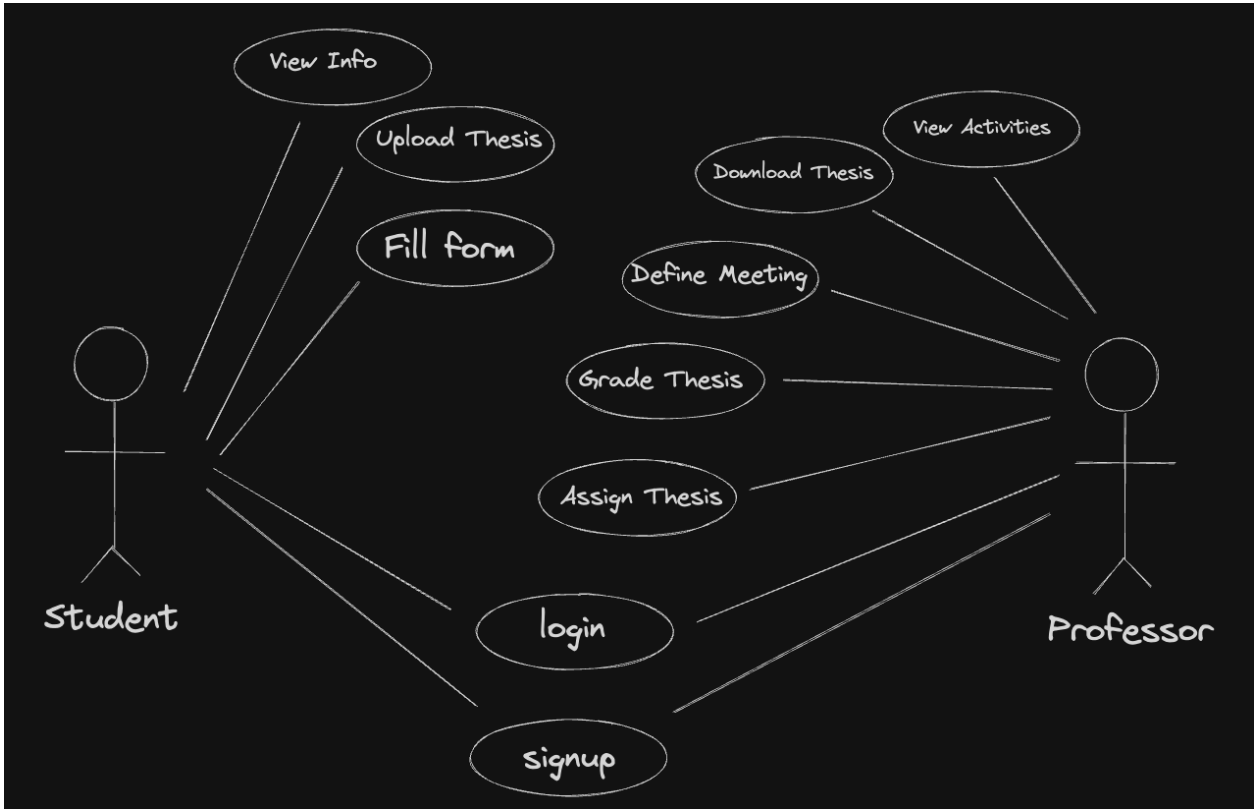
Class Diagrams

Τα διαγράμματα τάξεων (Class Diagrams) αναπαριστούν τις κλάσεις (ή τάξεις) του προγράμματος με τις ιδιότητες και τις μεθόδους τους.

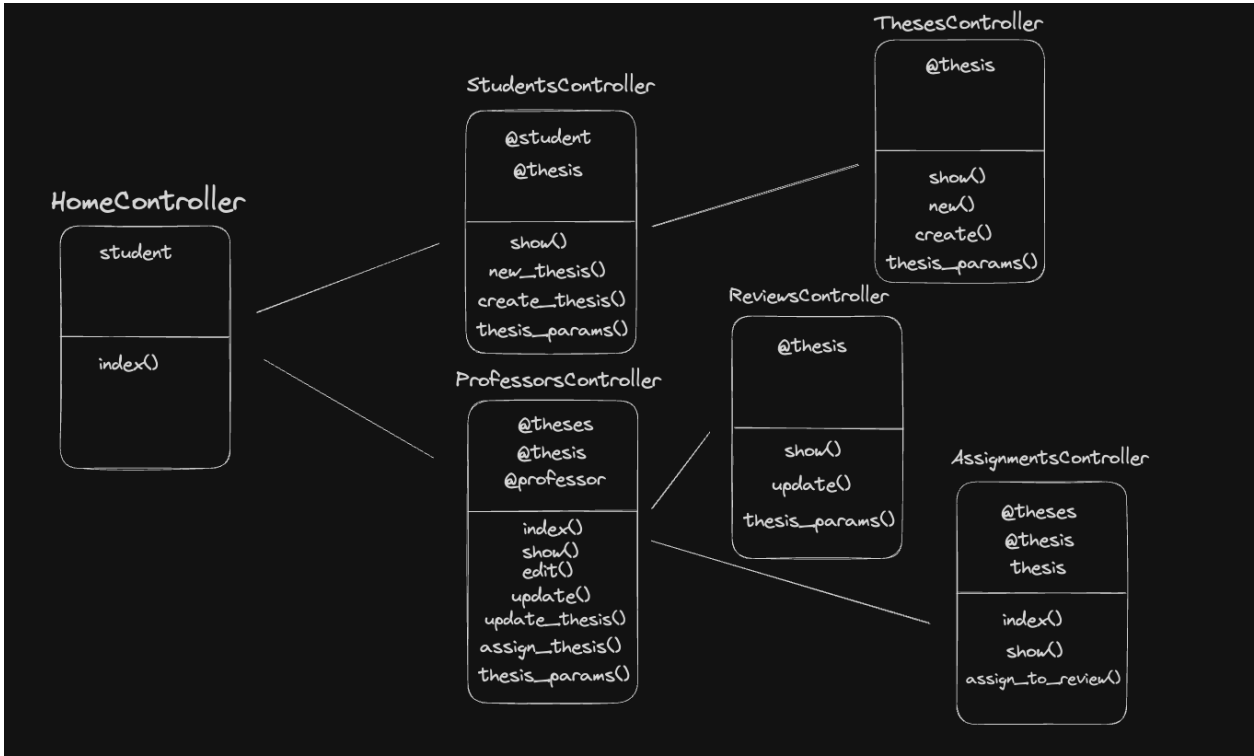


Φάση 2η:

Use Case Diagrams

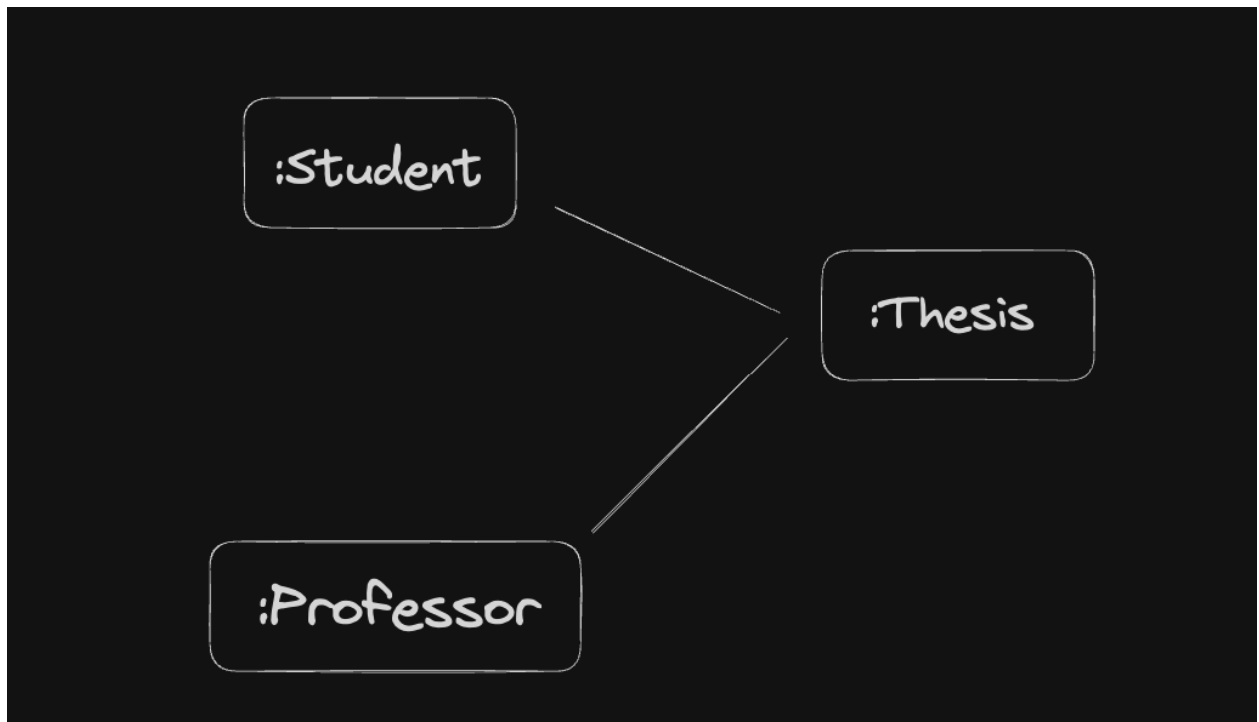


Class Diagrams



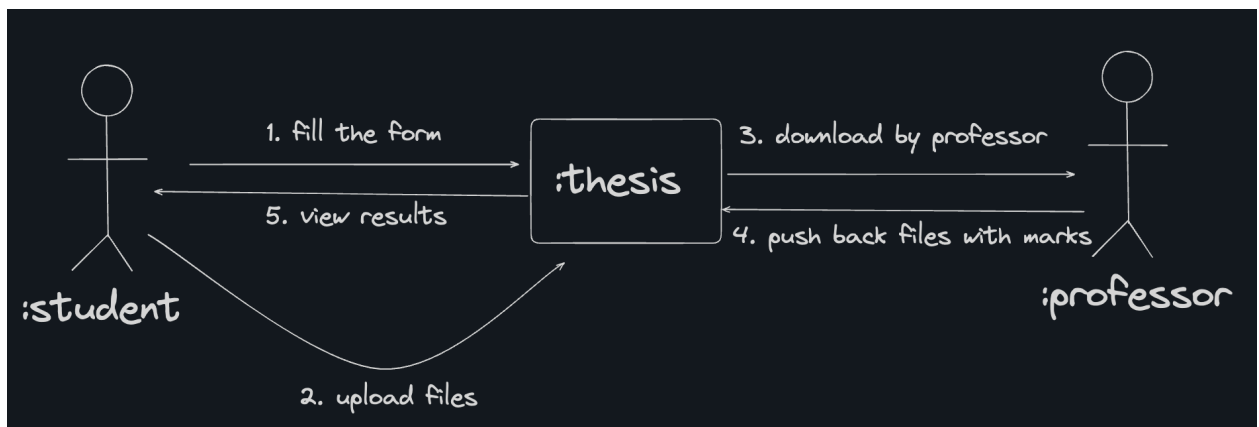
Object Diagram

Το διάγραμμα αντικειμένων (Object Diagram) αναπαριστά αντικείμενα και τις σχέσεις μεταξύ τους.



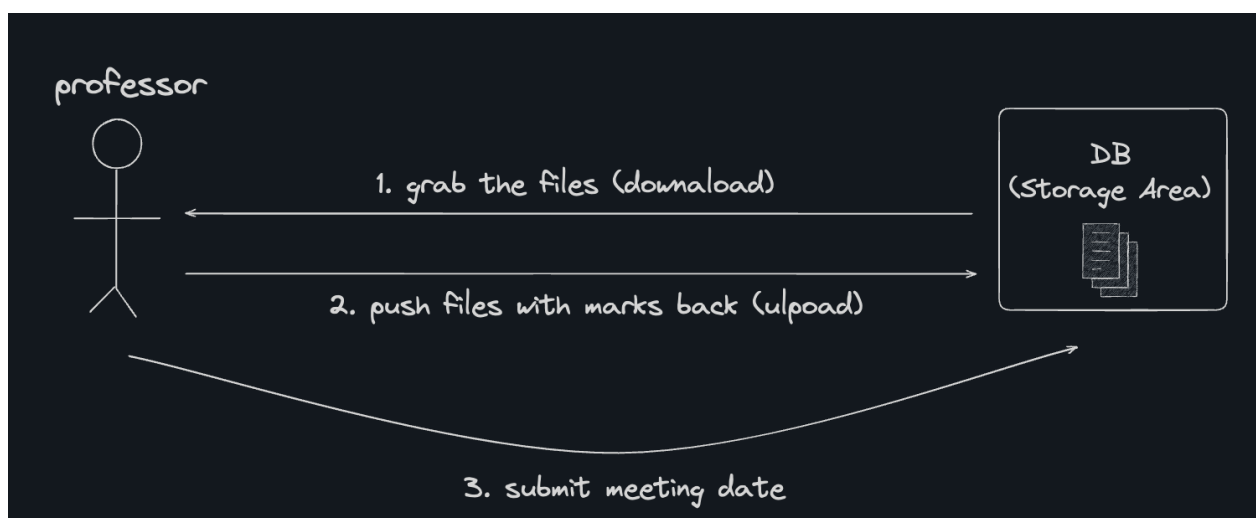
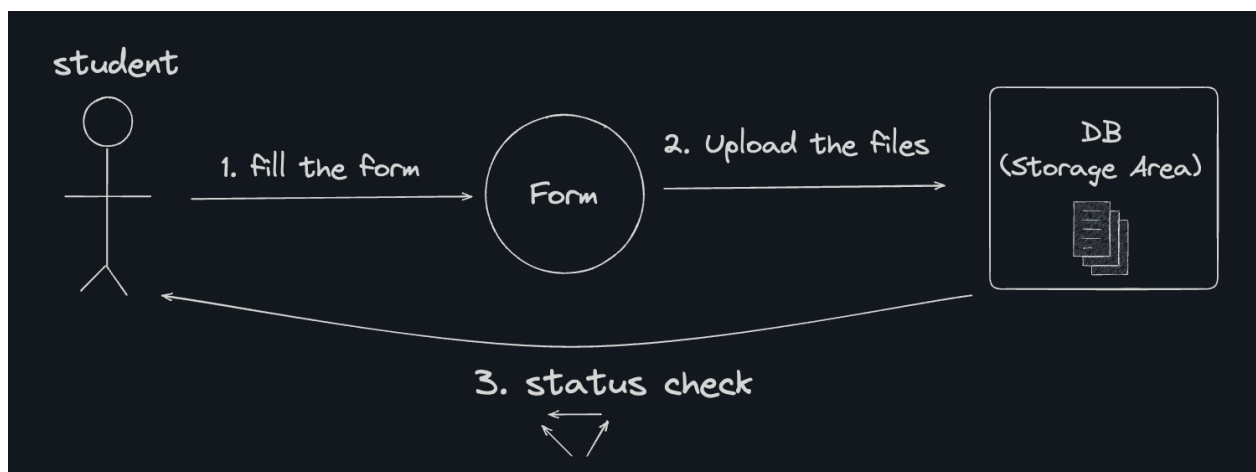
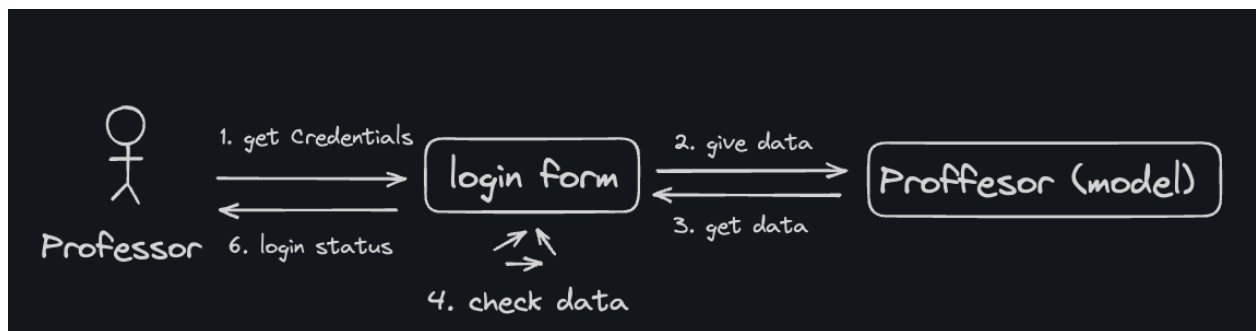
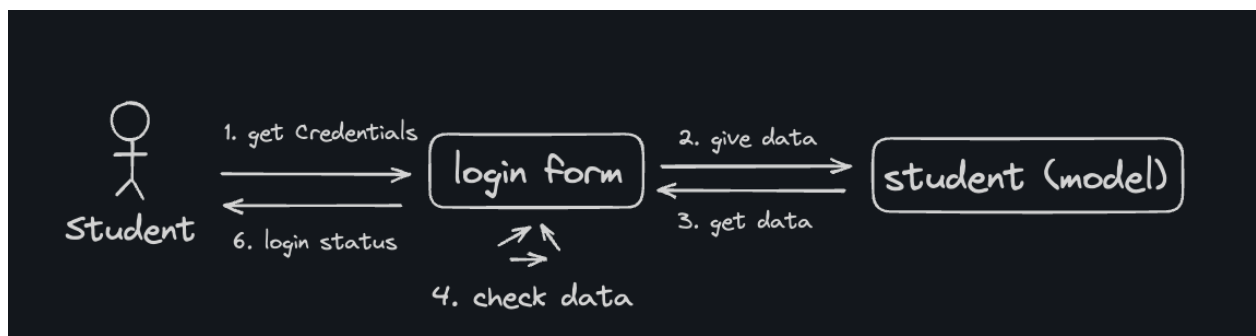
Collaboration Diagram

Το διάγραμμα συνεργασίας (Collaboration Diagram) αναπαριστά τον τρόπο με τον οποίο οι αντικείμενα δουλεύουν για να εκτελέσουν μια συγκεκριμένη εργασία στο πρόγραμμα. Αυτά τα παραδείγματα δείχνουν πώς οι φοιτητές και οι καθηγητές αλληλεπιδρούν μεταξύ τους σε 2 διαφορετικά σενάρια.



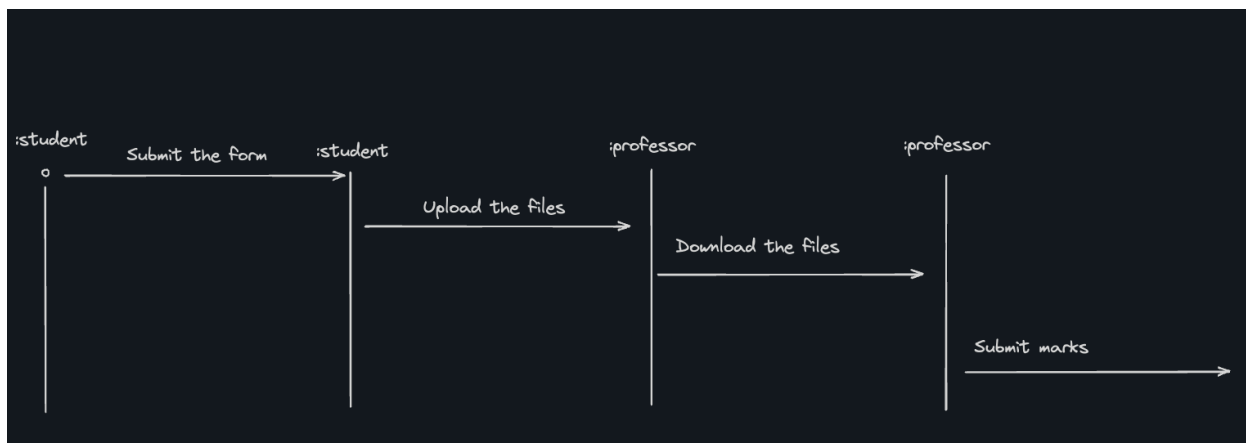
Activity Diagram

Το διάγραμμα δραστηριοτήτων (Activity Diagram) αναπαριστούν τη ροή των διαδικασιών σε ένα σύστημα (πρόγραμμα). Αυτά τα διαγράμματα μας βοηθούν στην κατανόηση του τρόπου με τον οποίο διάφορες ενέργειες γίνονται στο σύστημα.



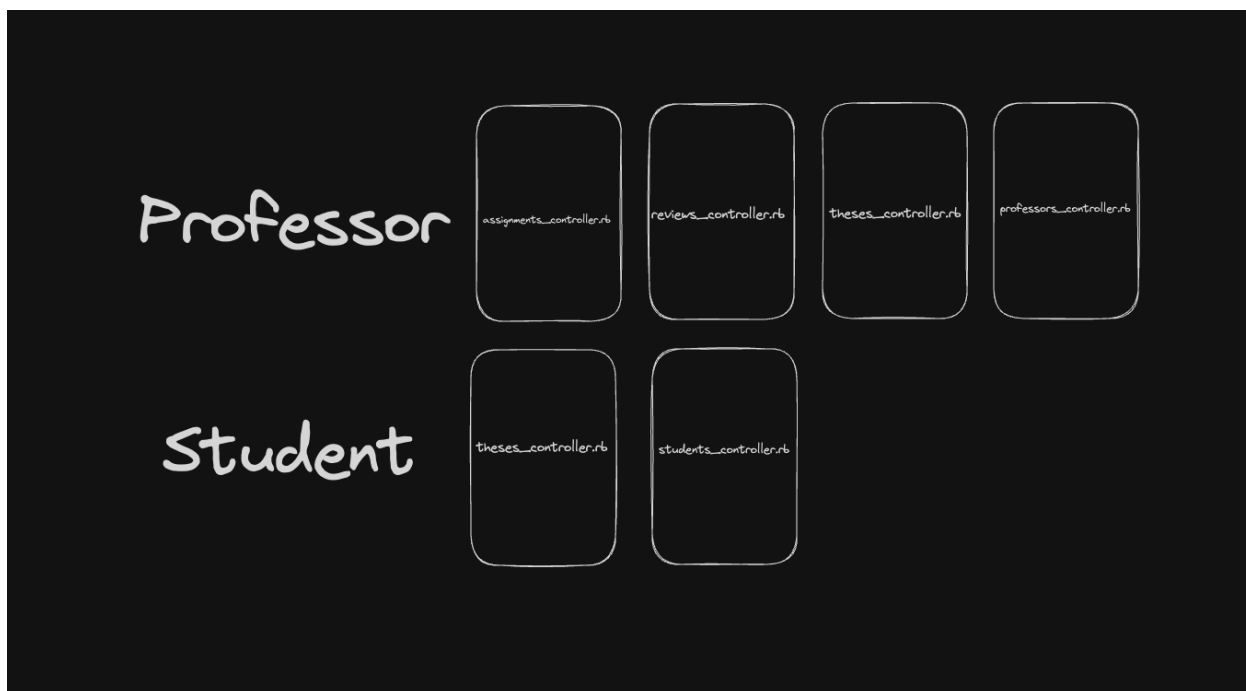
Sequence Diagrams

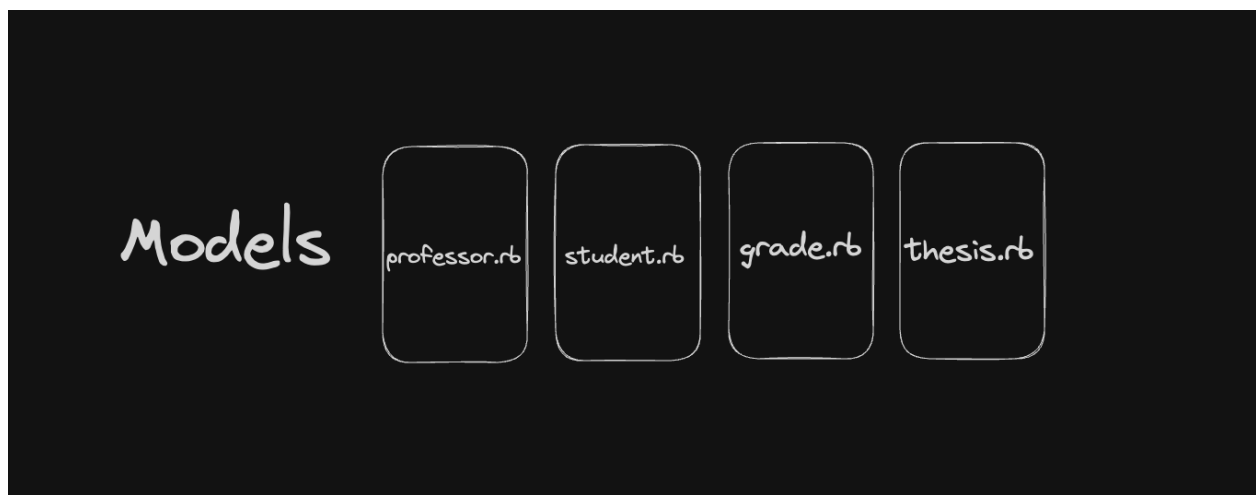
Τα διαγράμματα σειράς (Sequence Diagrams) αναπαριστούν την ροή, την σειρά με την οποία γίνονται η διαδικασίες.



Component Diagrams

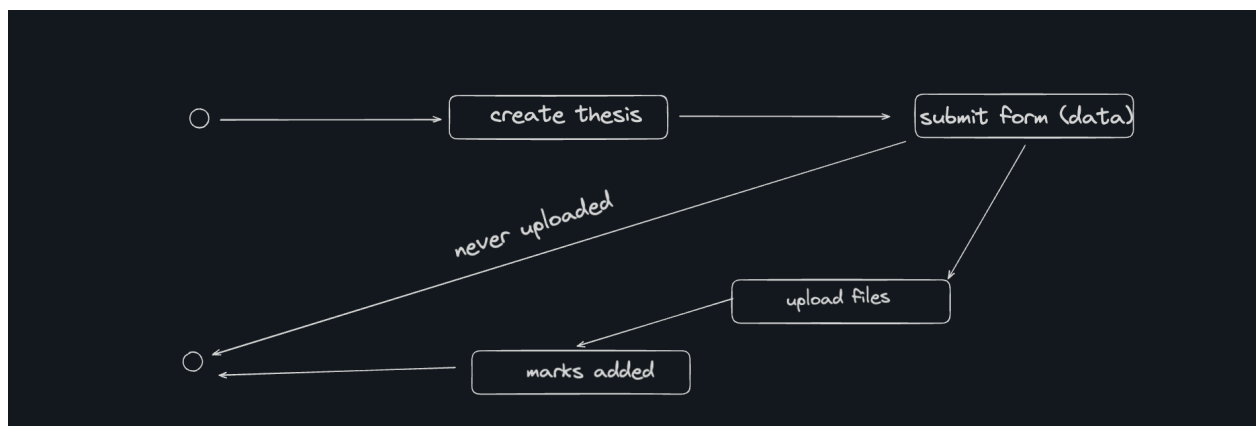
Τα διαγράμματα εξαρτημάτων (Component Diagrams) αναπαριστούν τη δομή του προγράμματος, τα διάφορα κομμάτια του δηλαδή.





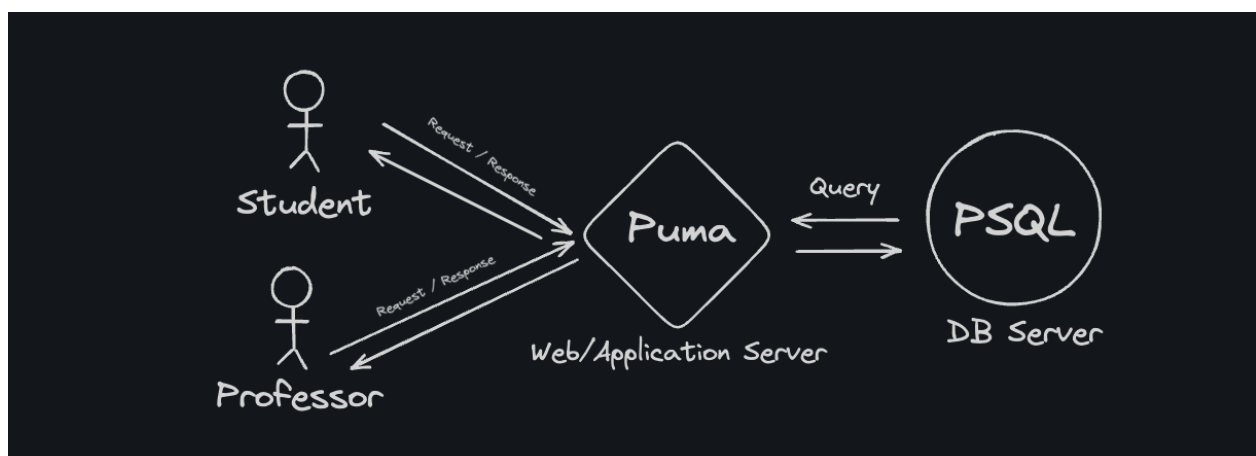
State Diagrams

Τα διαγράμματα καταστάσεων (State Diagrams) αναπαριστούν τις διάφορες καταστάσεις (states) ενός αντικειμένου ή των αντικειμένων στο σύστημα.



Deployment Diagrams

Τα διαγράμματα διανομής (Deployment Diagrams) αναπαριστούν τον τρόπο με τον οποίο το πρόγραμμα θα μεταφερθεί και θα στηθεί για προς χρήση.

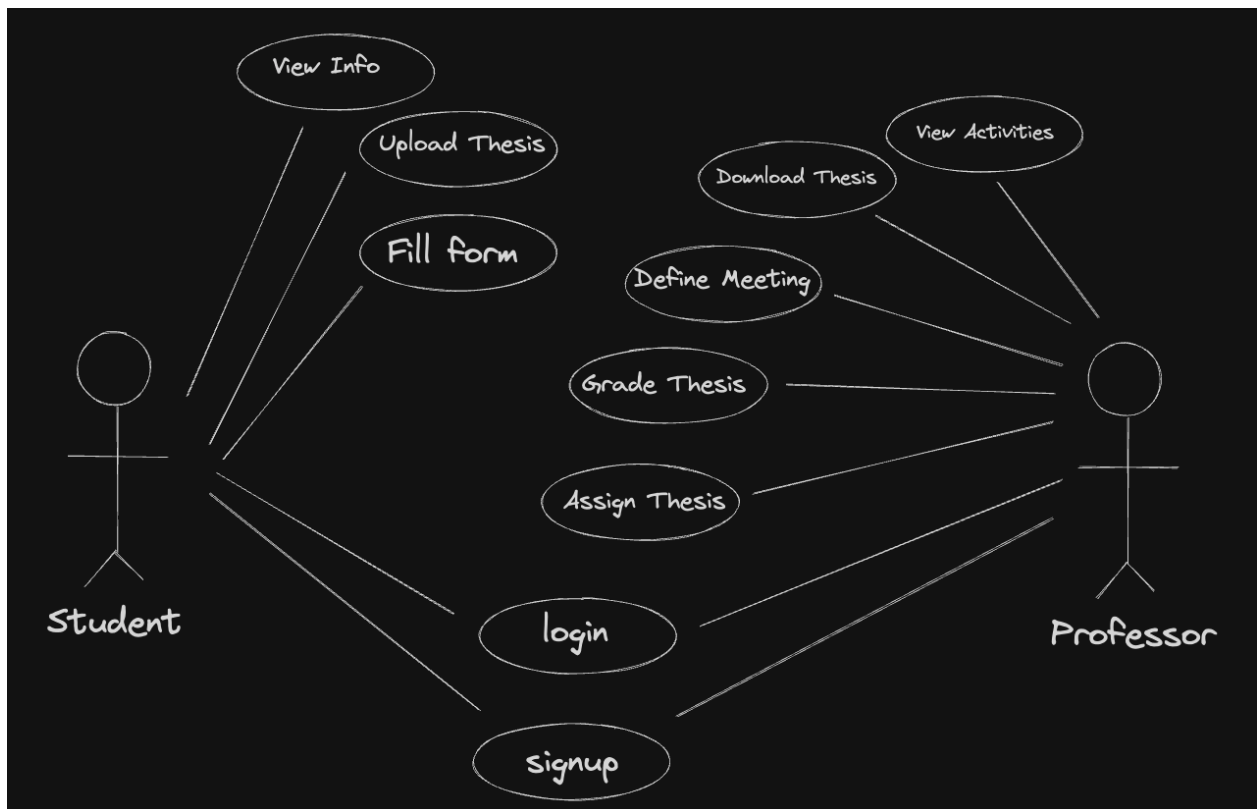


Έλεγχος:

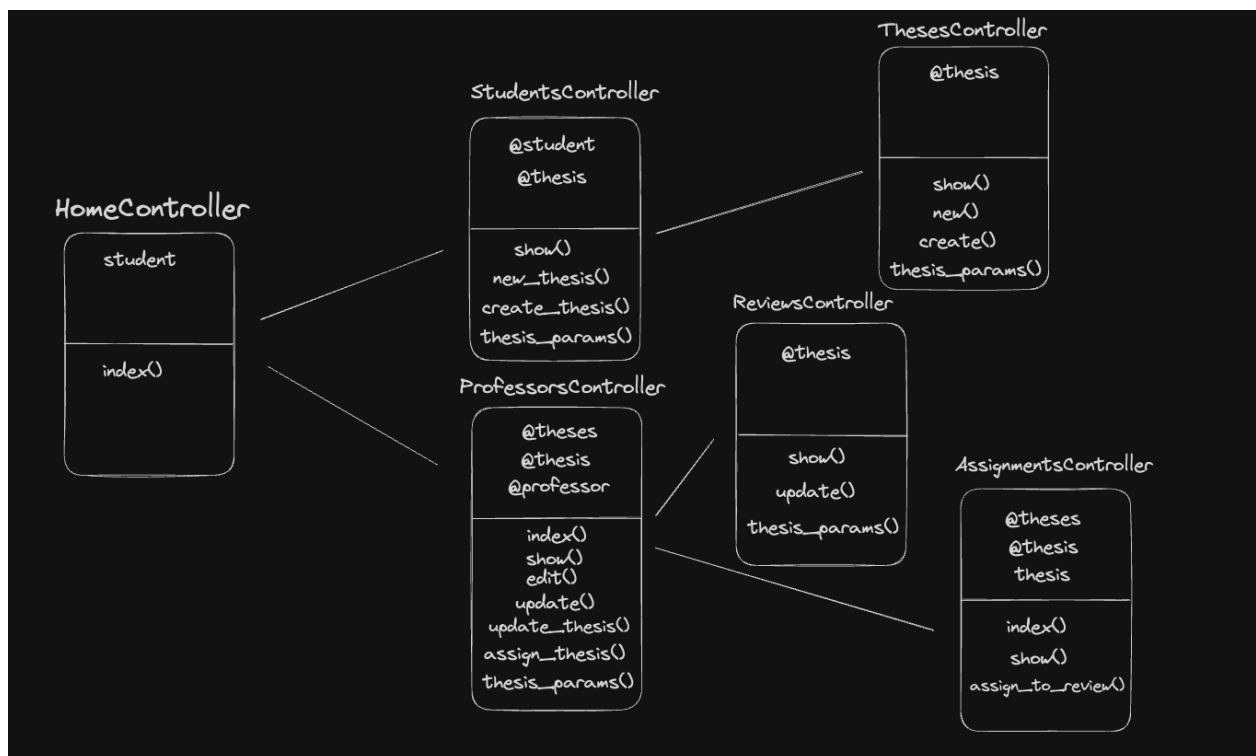
Ορίσαμε τις δομές, τους χαρακτήρες και τις βασικές λειτουργίες τους. Οι σχέσεις είναι απλές ενώ ταυτόχρονα η λειτουργικότητα περιγράφεται κατάλληλα από τα διαγράμματα.

Φάση 3η:

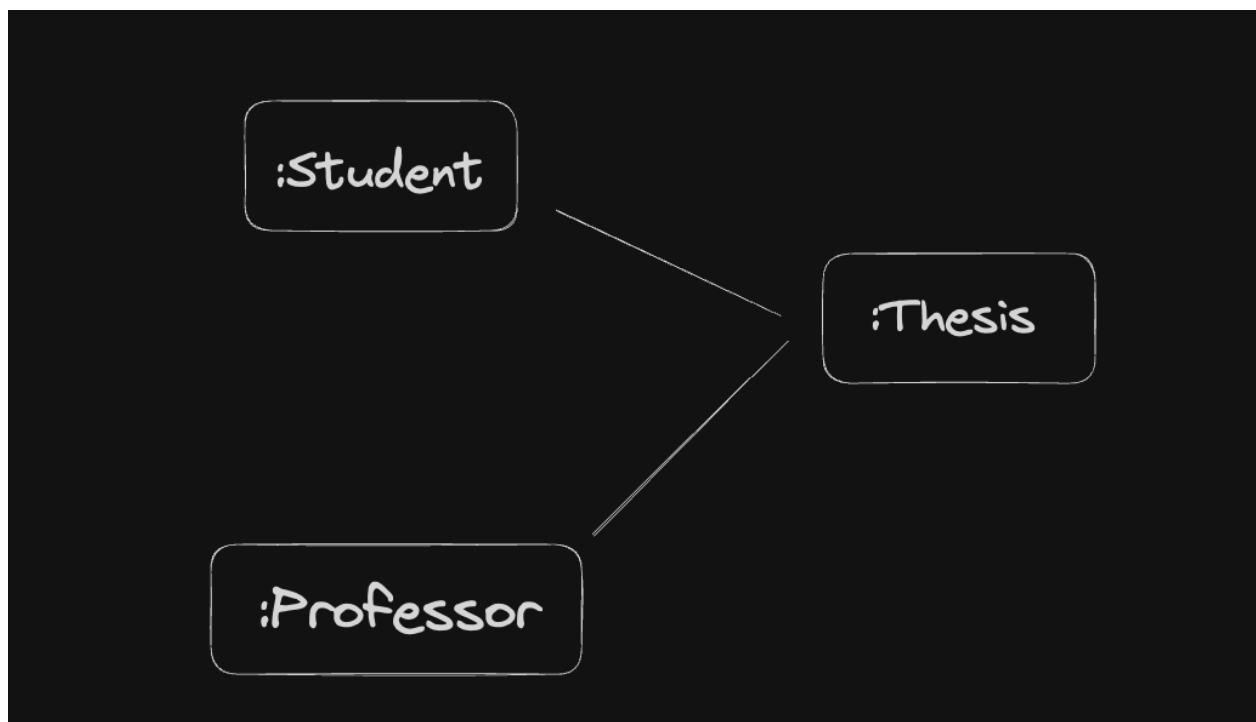
Use Case Diagrams



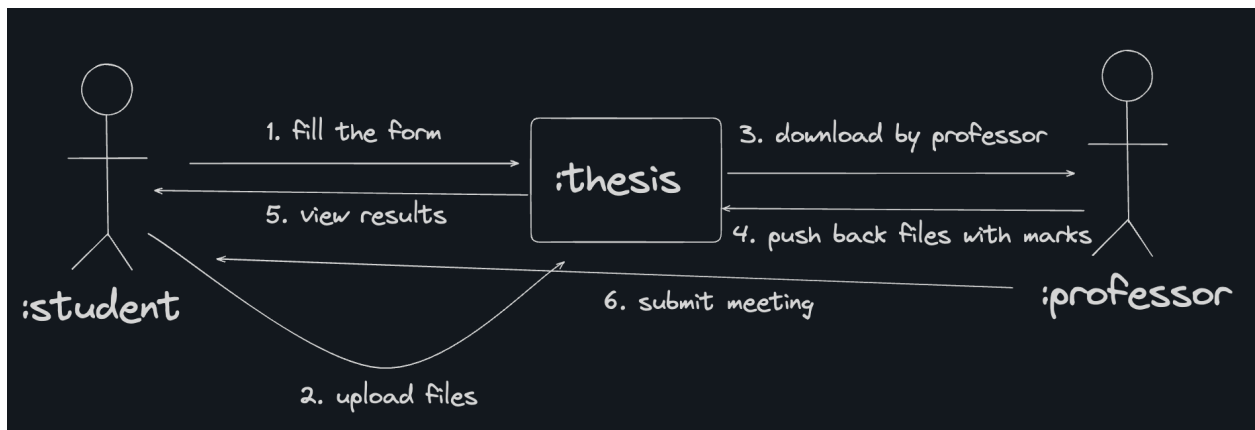
Class Diagrams



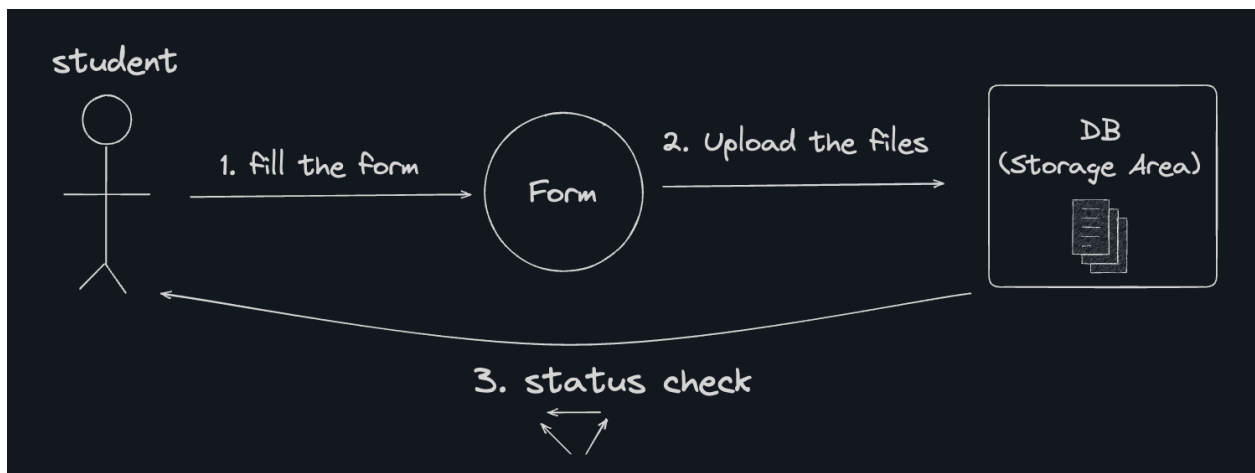
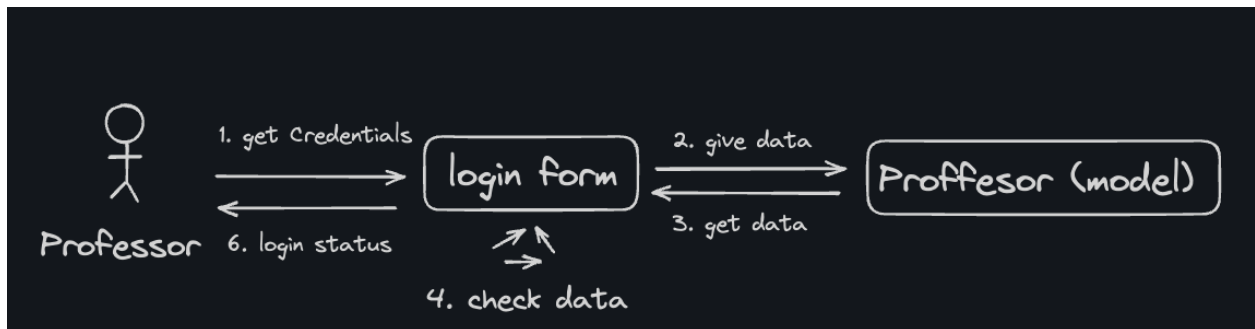
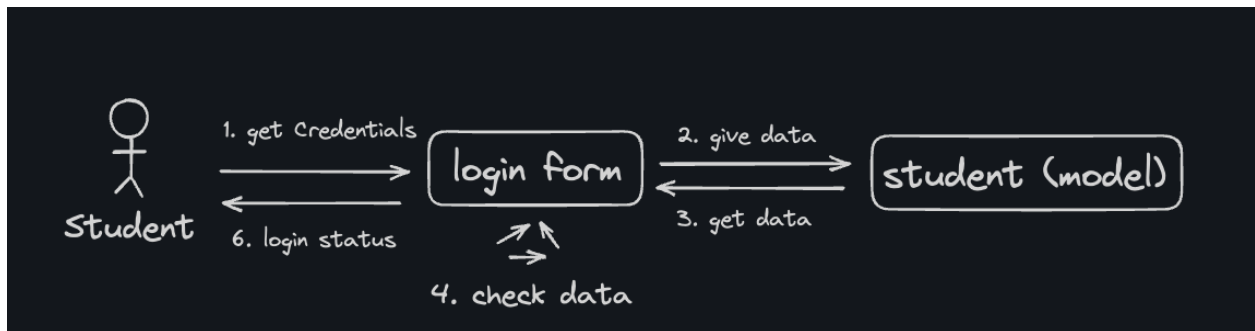
Object Diagram

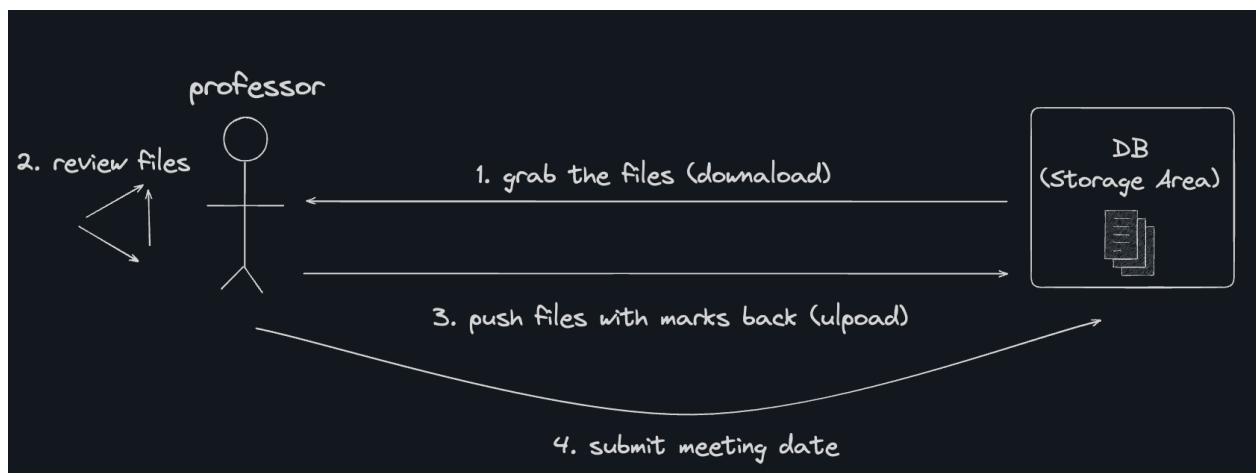


Collaboration Diagram

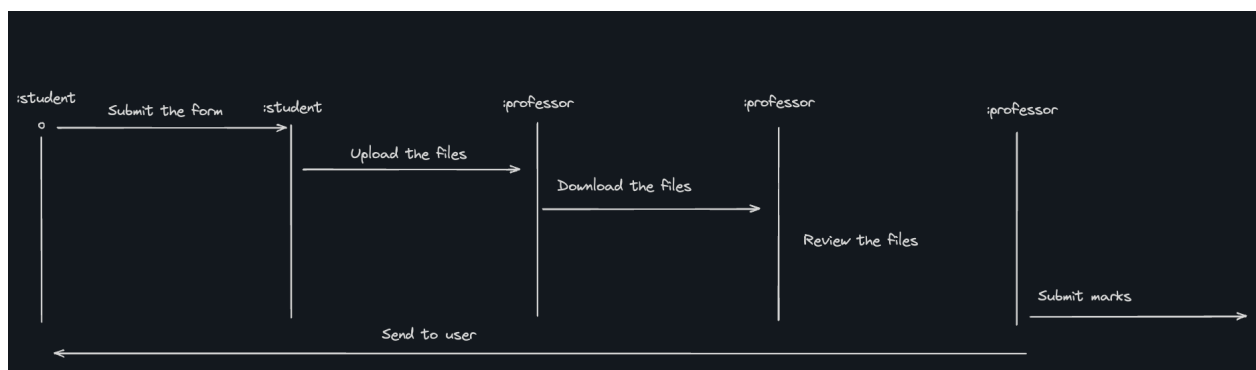


Activity Diagram

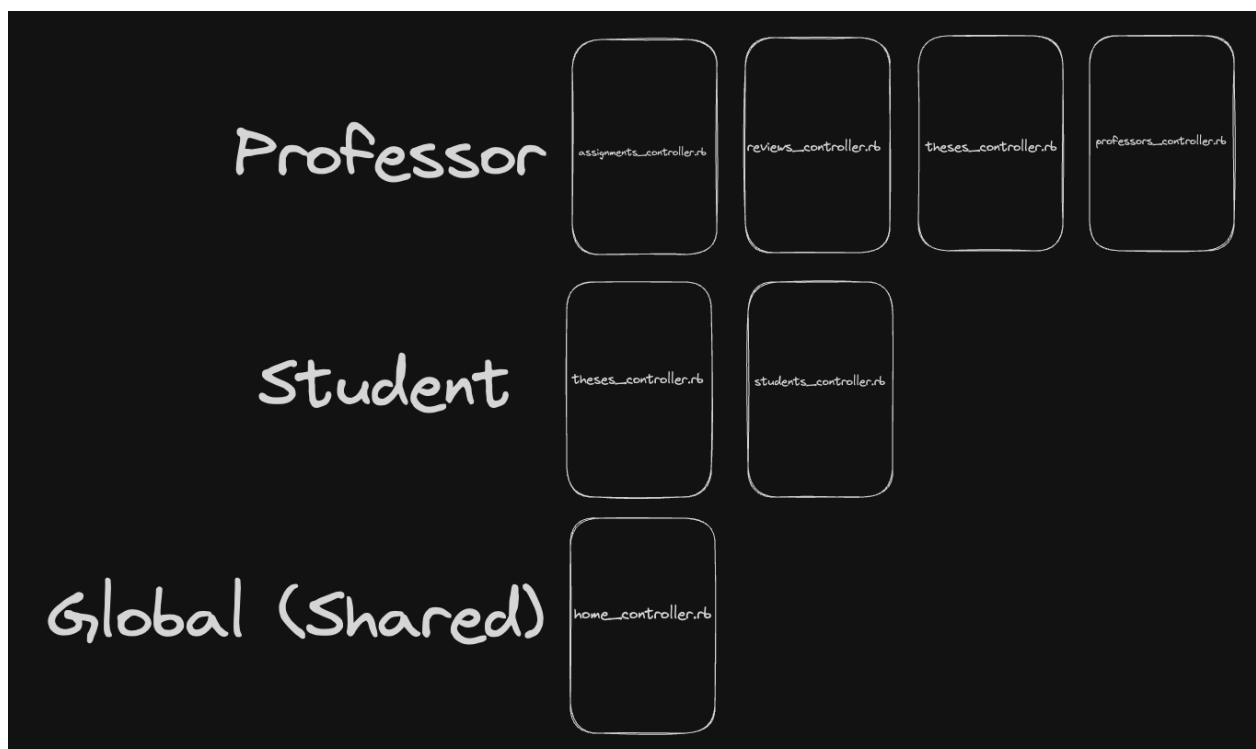


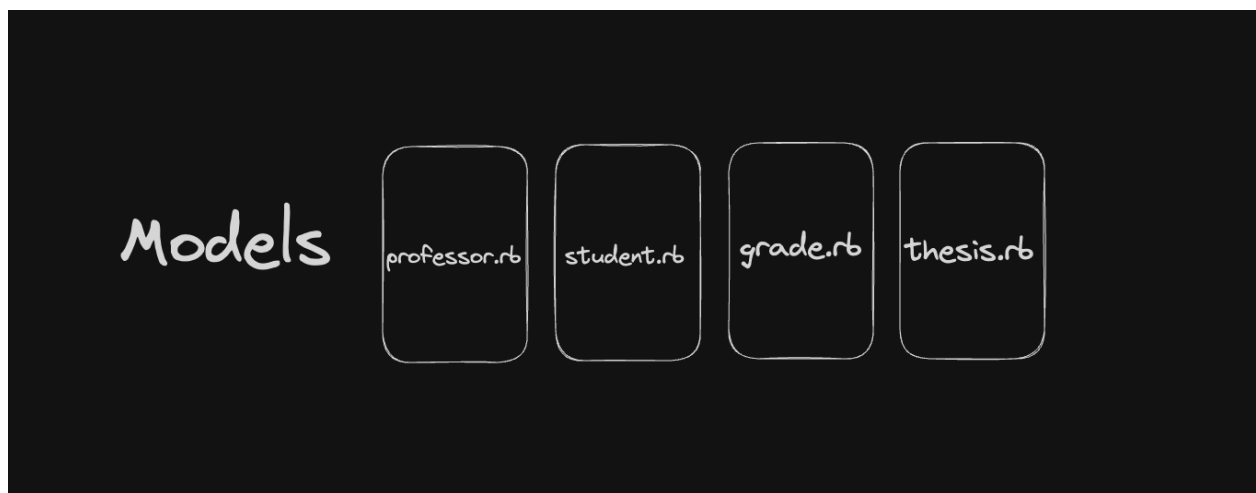


Sequence Diagrams

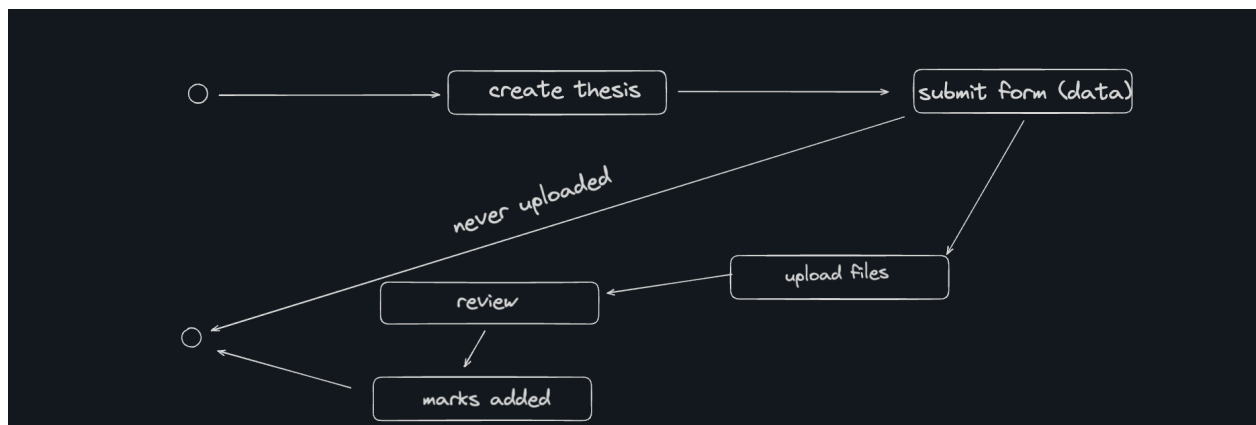


Component Diagrams

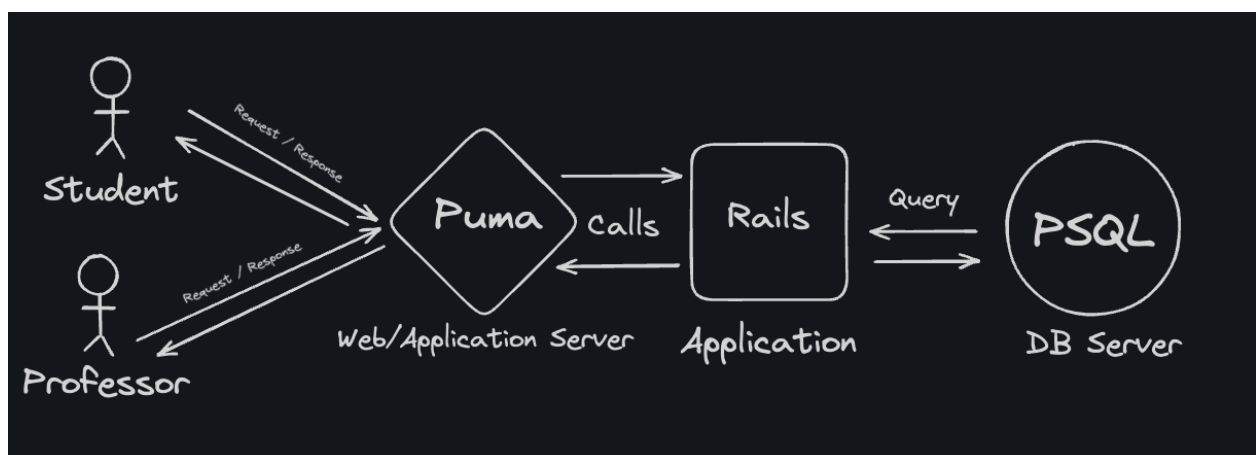




State Diagrams



Deployment Diagrams



Έλεγχος 2:

Ορίσαμε τις δομές, τους χαρακτήρες και τις βασικές λειτουργίες τους. Οι σχέσεις είναι απλές ενώ ταυτόχρονα η λειτουργικότητα περιγράφεται κατάλληλα από τα διαγράμματα. Έγιναν κάποιες αλλαγές στις διαδικασίες κυρίως του professor.