



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

**ΣΧΟΛΗ Ψηφιακής Τεχνολογίας
ΤΜΗΜΑ Πληροφορικής και Τηλεματικής**

**Ανάπτυξη εφαρμογής ιχνηλάτησης κρουσμάτων Covid-19 στα
ακαδημαϊκά ιδρύματα
Πτυχιακή εργασία**

Παναγιώτης Κουρπάς

Αθήνα, 2022



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

**ΣΧΟΛΗ Ψηφιακής Τεχνολογίας
ΤΜΗΜΑ Πληροφορικής και Τηλεματικής**

Τριμελής Εξεταστική Επιτροπή

**Κλεοπάτρα Μπαρδάκη (Επιβλέπουσα)
Επίκουρη Καθηγήτρια, Πληροφορικής και τηλεματικής,
Χαροκόπειο Πανεπιστήμιο**

**Τσερπές Κωνσταντίνος
Αναπληρωτής Καθηγητής, Πληροφορικής και τηλεματικής,
Χαροκόπειο Πανεπιστήμιο**

**Σταμάτη Τέτα
Επίκουρη Καθηγήτρια, Πληροφορικής και τηλεματικής,
Χαροκόπειο Πανεπιστήμιο**

Ο Παναγιώτης Κουρπάς

δηλώνω υπεύθυνα ότι:

- 1) Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλλει τα πνευματικά δικαιώματα τρίτων.
- 2) Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω την κυρία Μπαρδάκη για τον χρόνο που αφιέρωσε και τις συμβουλές που μου έδωσε για την συγγραφή και υλοποίηση της πτυχιακή εργασίας. Επιπλέον θα ήθελα να ευχαριστήσω την εξεταστική επιτροπή για την βαθμολόγηση της και την αξιολόγηση της. Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη που μου παρείχε καθόλη την διάρκεια των σπουδών.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Περίληψη στα Ελληνικά	6
Abstract	7
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	8
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ	9
ΕΙΣΑΓΩΓΗ	10
ΚΕΦ.1: Ανάλυση και σχεδίαση	11
1.1 Μοντέλο σχεδίασης καταρράκτη	11
1.2 Use Case Diagram	12
1.3 Class Diagram	13
1.4 Sequence Diagrams	14
1.5 Μη λειτουργικές απαιτήσεις	16
ΚΕΦ.2: Υλοποίηση εφαρμογής	17
2.1. Python	17
2.2. Django	18
2.3. SQLite και PostgreSQL	19
2.4. Django ORM	19
2.5. Deploy Εφαρμογής (Heroku)	19
ΚΕΦ.3: Σενάρια χρήσης	24
3.1. Γενικά	24
3.2 Γραμματεία (Admin)	24
3.3 Φοιτητής (User)	30
3.4 Χρήση ερωτηματολογίου	33
3.5 Απαντήσεις ερωτηματολογίου	36
ΚΕΦ.4: Επίλογος	38
ΒΙΒΛΙΟΓΡΑΦΙΑ	39

Περίληψη στα Ελληνικά

Στην συγκεκριμένη πτυχιακή εργασία αναπτύσσω ένα σύστημα μια **εφαρμογή ιχνηλάτησης κρουσμάτων Covid-19** στα ακαδημαϊκά ιδρύματα. Μέχρι τώρα για να γίνεται η σωστή ιχνηλάτηση των κρουσμάτων στον πανεπιστημιακό χώρο οι καθηγητές μοίραζαν μια **καρτέλα** που περιέχει τις θέσεις του αμφιθέατρου και ο κάθε φοιτητής συμπλήρωνε τον αριθμό μητρώου στην κατάλληλη θέση. Τα **στοιχεία** για την διάλεξη τα συμπλήρωνε ο καθηγητής. Με την συγκεκριμένη εφαρμογή ο φοιτητής απλά κάνει login στην εφαρμογή μέσω ενός browser (τις περισσότερες φορές από το κινητό του) χρησιμοποιώντας το πανεπιστημιακό του λογαριασμό. Μόλις κάνει login ο φοιτητής βλέπει όλες τις σημερινές διαλέξεις, επιλέγει αυτή που παρακολουθεί και συμπληρώνει τον αριθμό της θέσης του. Επιπλέον μπορεί να δει όλες τις θέσεις του της κάθε διάλεξης και μπορεί να δηλώσει το πότε έχει διαγνωστεί με Covid-19 συμπληρώνοντας απλά την ημερομηνία. Η γραμματεία είναι υπεύθυνη να συμπληρώνει τα στοιχεία των διαλέξεων σύμφωνα με την ημερομηνία, το μάθημα, την αίθουσα και όλες τις απαραίτητες πληροφορίες που θα πρέπει να λάβουν οι φοιτητές. Επειτα σε περίπτωση κρούσματος βλέπει τα κρούσματα και όλα τα στοιχεία στα κατάλληλα μενού και καλεί τον κάθε φοιτητή έτσι ώστε να μπει σε καραντίνα.

Με την συγκεκριμένη εφαρμογή βλέπουμε πόσο χρήσιμη μπορεί να γίνει η επιστήμη της πληροφορικής στην καθημερινότητα μας και αυτό γιατί μπορούμε μέσα σε λίγο χρόνο να πάρουμε τα δεδομένα των φοιτητών δηλαδή το σε ποια θέση κάθονται ή το πότε και αν διαγνώστηκαν κρούσματα του κορονοϊού. Με τον μέχρι τώρα τρόπο όλα αυτά γινόντουσαν manual δηλαδή η γραμματεία θα έπρεπε να ψάχνει την κατάλληλη καρτέλα μόλις κάποιος φοιτητής είχε διαγνωστεί με κορονοϊό. Με την ηλεκτρονική εφαρμογή μπορεί να δει τα δεδομένα από κάθε τοποθεσία και κάθε ώρα χωρίς να πρέπει να βρίσκεται κάποιος στο γραφείο αλλά και από το σπίτι του.

Τέλος, αυτό είναι ένα παράδειγμα ενός πληροφοριακού συστήματος. Θα μπορούσαμε να σκεφτούμε και άλλα πράγματα στην καθημερινότητα μας που μια διαδικτυακή εφαρμογή θα μπορούσε να λύσει με ευκολία, όπως π.χ μια λίστα για super market που θα είναι ψηφιακή αλλά και πολλές ιδέες ακόμα. Αυτός είναι ένας από τους λόγους που επέλεξα να κάνω αυτή την πτυχιακή εκτός από το ίδιο το σύστημα.

Abstract

In this dissertation I develop a system a **Covid-19 case tracking application** in academic institutions. Until now, in order to properly track cases on campus, teachers distributed a **card** containing the seats of the amphitheater and each student filled in the registration number in the appropriate place. The **data** for the lecture were completed by the teacher. With this application, the student simply logs in to the application through a browser (most of the time from his mobile phone) using his university account. Once logged in, the student sees all the current lectures, chooses the one he / she attends and fills in his / her position number. In addition he can see all the positions of each lecture and can state when he was diagnosed with Covid-19 by simply filling in the date. The secretariat is responsible for filling in the details of the lectures according to the date, the course, the room and all the necessary information that the students should receive. Then in case of a case he sees the cases and all the elements in the appropriate menus and invites each student to enter quarantine.

With this application we see how useful computer science can become in our daily lives and this is because we can in a short time get the data of students, ie where they sit or when and if coronavirus cases were diagnosed. In the past, all this was done manually, the secretariat should look for the appropriate card as soon as a student was diagnosed with coronavirus. With the electronic application he can see the data from any location and any time without having to be in the office or from home.

Finally, this is an example of an information system. We could think of other things in our daily lives that a web application could easily solve, such as a list for a supermarket that will be digital and many more ideas. That's one of the reasons I chose to do this degree in addition to the system itself.

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικ.1.1 Use Case Diagram	σ.12
Εικ.1.2 Class Diagram.....	σ.13
Εικ.1.2 Sequence Diagram 1.....	σ.14
Εικ.1.2 Sequence Diagram 2.....	σ.15
Εικ.1.2 Sequence Diagram 3.....	σ.15
Εικ.2.1 MVC Diagram.....	σ.18
Εικ.2.2 Heroku Login.....	σ.20
Εικ.2.3 Create New App.....	σ.20
Εικ.2.4 Upload App.....	σ.21
Εικ.2.5 Heroku Procfile.....	σ.21
Εικ.2.6 Create runtime.txt.....	σ.21
Εικ.2.7 Create requirements.txt.....	σ.22
Εικ.2.8 Change settings.py.....	σ.22
Εικ.3.1 Admin login screen.....	σ.24
Εικ.3.2 Admin home page.....	σ.25
Εικ.3.3 Σελίδα προβολής χρηστών.....	σ.25
Εικ.3.4 Σελίδα προβολής αμφιθεάτρων.....	σ.26
Εικ.3.5 Σελίδα δημιουργίας αμφιθεάτρου.....	σ.26
Εικ.3.6 Σελίδα προβολής διαλέξεων.....	σ.27
Εικ.3.7 Σελίδα δημιουργίας διάλεξης.....	σ.27
Εικ.3.8 Σελίδα προβολής ημερομηνίας διάλεξης.....	σ.28
Εικ.3.9 Σελίδα δημιουργίας ημερομηνίας διάλεξης.....	σ.28
Εικ.3.10 Σελίδα προβολής θέσεων φοιτητή.....	σ.29
Εικ.3.11 Σελίδα προβολής θετικών κρουσμάτων.....	σ.29
Εικ.3.12 Student login screen.....	σ.30
Εικ.3.13 Σελίδα προβολής σημερινών διαλέξεων.....	σ.30
Εικ.3.14 Σελίδα εισαγωγής θέσης.....	σ.31
Εικ.3.15 Σελίδα προβολής δεδομένων διάλεξης.....	σ.31
Εικ.3.16 Σελίδα δήλωσης κρούσματος.....	σ.32
Εικ.3.17 Σελίδα δήλωσης κρούσματος (ημερομηνία).....	σ.32
Εικ.3.18 Πρώτη ερώτηση ερωτηματολογίου.....	σ.33
Εικ.3.19 Δεύτερη ερώτηση ερωτηματολογίου.....	σ.34
Εικ.3.20 Τρίτη ερώτηση ερωτηματολογίου.....	σ.34
Εικ.3.21 Τέταρτη ερώτηση ερωτηματολογίου.....	σ.35
Εικ.3.22 Πέμπτη ερώτηση ερωτηματολογίου.....	σ.35
Εικ.3.23 Έκτη ερώτηση ερωτηματολογίου.....	σ.36
Εικ.3.24 Πρώτη απάντηση ερωτηματολογίου.....	σ.36
Εικ.3.25 Δεύτερη απάντηση ερωτηματολογίου.....	σ.36
Εικ.3.26 Τρίτη απάντηση ερωτηματολογίου.....	σ.37
Εικ.3.27 Τέταρτη απάντηση ερωτηματολογίου.....	σ.37
Εικ.3.28 Πέμπτη απάντηση ερωτηματολογίου.....	σ.37

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

UML	Unified Modeling Language
SQL	Structured Query Language
MVC	Model-View-Controller
ORM	Object–relational mapping

ΕΙΣΑΓΩΓΗ

Η εφαρμογή ιχνηλάτησης κρουσμάτων Covid-19 στα ακαδημαϊκά ιδρύματα είναι μια διαδικτυακή εφαρμογή που αναπτύχθηκε με βάση το Django Web Framework. Λειτουργεί μέσω ενός web browser και η ιδέα της εφαρμογής είναι να μπορεί ο κάθε φοιτητής να καταχωρεί την θέση στην οποία κάθεται και σε περίπτωση κρούσματος η γραμματεία να κάνει την κατάλληλη ιχνηλάτηση και να βλέπει τα δεδομένα των φοιτητών. Επιπλέον οι φοιτητές εκτός από το να καταχωρούν την θέση τους, μπορούν να προβάλλουν και τα δεδομένα δηλαδή σε ποια θέση κάθονται και να δηλώσουν την ημερομηνία που διαγνώστηκαν θετικοί στον Covid-19. Τα 2 κύρια πρόσωπα που χρησιμοποιούν την εφαρμογή είναι οι φοιτητές και η γραμματεία.

Στα τεχνικά θέματα της εφαρμογής, η εφαρμογή έχει αναπτυχθεί με το Django Web Framework χρησιμοποιώντας την Python για γλώσσα προγραμματισμού. Η βάση δεδομένων όταν τρέχει τοπικά είναι SQLite, όμως εφόσον η συγκεκριμένη εργασία έχει γίνει deploy στο cloud (συγκεκριμένα στο heroku) η βάση είναι Postgres.

Με την συγκεκριμένη πτυχιακή μπορούμε να δούμε και το πόσο σημαντικό είναι το cloud στις μέρες μας, καθώς μπορούμε να ανεβάσουμε οποιαδήποτε εφαρμογή από οποιοδήποτε μέρος του κόσμου στην παραγωγή.

ΚΕΦ.1: Ανάλυση και σχεδίαση

1.1 Μοντέλο σχεδίασης καταρράκτη

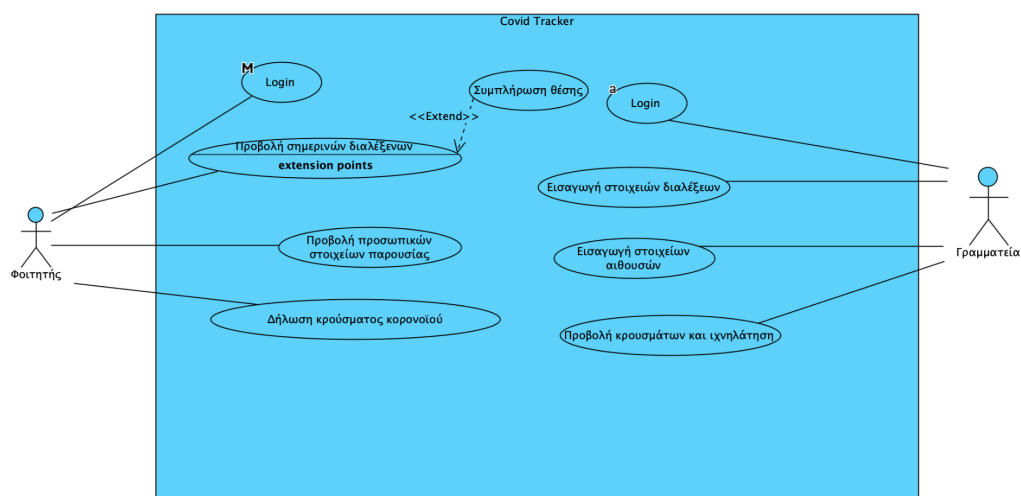
Στην συγκεκριμένη πτυχιακή χρησιμοποιήθηκε το μοντέλο σχεδίασης καταρράκτη (waterfall). Όταν ακολουθείται το συγκεκριμένο μοντέλο ο σχεδιασμός και η υλοποίηση γίνεται σε διαδοχικά βήματα, που το κάθε ένα πρέπει να ολοκληρωθεί πριν ξεκινήσει το επόμενο. Χωρίζεται στις εξής φάσεις:

1. Ανάλυση συστήματος: Η πρώτη φάση αφορά την ανάλυση του συστήματος και των απαιτήσεων του χρήστη δηλαδή τι δεδομένα πρέπει να αποθηκεύονται και τι λειτουργίες ο χρήστης μπορεί να κάνει. Στην συγκεκριμένη πτυχιακή χρησιμοποιήθηκε η γλώσσα UML για αυτό τον σκοπό.
2. Σχεδίαση συστήματος: Η δεύτερη φάση αφορά την σχεδίαση του συστήματος όπως το ποια γλώσσα προγραμματισμού ή τι framework θα χρησιμοποιήσουμε. Στην συγκεκριμένη πτυχιακή χρησιμοποιήθηκε η Python.
3. Υλοποίηση συστήματος: Η τρίτη φάση αφορά την υλοποίηση του συστήματος, δηλαδή την κατάλληλη συγγραφή του κώδικα έτσι ώστε να αναπτύξουμε την εφαρμογή.
4. Testing: Η τέταρτη φάση αφορά την δοκιμή του συστήματος έτσι ώστε να δούμε άμα περιέχει λογικά ή συντακτικά λάθη. Η δοκιμή αυτή γίνεται εξαντλώντας τις λειτουργίες της εφαρμογής και δοκιμάζοντας κάθε πιθανό ενδεχόμενο σαν είσοδο.
5. Διανομή συστήματος: Η πέμπτη και τελευταία φάση αφορά την διανομή του συστήματος δηλαδή την εγκατάσταση, την υποστήριξη και την συντήρηση του συστήματος που έχουμε υλοποιήσει

Τα πλεονεκτήματα της μεθοδολογίας waterfall είναι αρκετά. Αρχικά αποτελεί την παλαιότερη μέθοδο ανάπτυξης λογισμικού. Επιπλέον είναι εύκολο να διαχειριστούμε τα βήματα της εφαρμογής καθώς είναι πολύ συγκεκριμένα και πριν προχωρήσουμε στον επόμενο, έχουμε ολοκληρώσει το προηγούμενο. Τέλος με το συγκεκριμένο μοντέλο υπάρχει καταμερισμός του έργου μεταξύ της ομάδας των προγραμματιστών που θα υλοποιήσουν το πληροφοριακό σύστημα.

Η μεθοδολογία waterfall έχει και μειονεκτήματα. Το κυριότερο είναι ότι σε τυχόν αλλαγές είναι δύσκολο να γυρίσουμε στις προηγούμενες φάσεις του συστήματος. Ωστόσο στην συγκεκριμένη πτυχιακή δεν αποτελεί ιδιαίτερο μειονέκτημα, καθώς η εφαρμογή είχε λεπτομερή περιγραφή και λειτουργία από την αρχή.

1.2 Use Case Diagram



Εικόνα 1.1 - Use Case Diagram

Το Use Case Diagram μας δείχνει τις λειτουργίες που μπορεί να πραγματοποιήσει ο κάθε χρήστης στην εφαρμογή και σε ποια δεδομένα έχει πρόσβαση.

Στην συγκεκριμένη περίπτωση έχουμε 2 κατηγορίες χρηστών την γραμματεία και τους φοιτητές. Η κάθε κατηγορία χρηστών έχει διαφορετικό ρόλο στο σύστημα για αυτό και έχει διαφορετικά Use Cases.

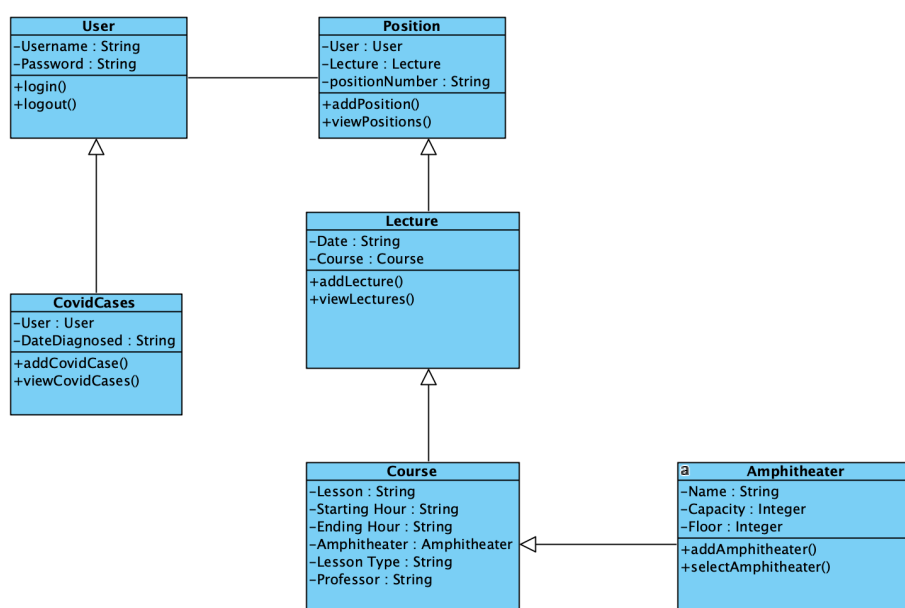
Use Cases φοιτητή:

1. Το πρώτο Use Case που αφορά τον φοιτητή είναι το login στο σύστημα. Ο φοιτητής για να χρησιμοποιήσει το σύστημα θα πρέπει να έχει κάνει login και να ταυτοποιηθούν τα στοιχεία του στο σύστημα.
2. Το δεύτερο Use Case που αφορά τον φοιτητή είναι η προβολή των σημερινών διαλέξεων που εκεί βλέπει την κάθε διάλεξη της ημέρας που έχει κάνει login και τα στοιχεία τους.
3. Στο τρίτο Use Case χρησιμοποιώ το extend γιατί ο φοιτητής θα πρέπει να πατήσει στην κάθε διάλεξη έτσι ώστε να συμπληρώσει την θέση του. Στην περίπτωση που δεν υπάρχει διάλεξη δεν θα μπορεί να εισάγει και την θέση του.
4. Στο τέταρτο Use Case ο φοιτητής μπορεί να προβάλει τα στοιχεία του, δηλαδή την θέση που έχει καθίσει και τα στοιχεία της διάλεξης αυτής.
5. Στο τελευταίο Use Case ο φοιτητής έχει την δυνατότητα να δηλώσει το άμα είναι θετικό κρούσμα κορονοϊού και την ημερομηνία που διαγνώστηκε. Αυτή την πληροφορία θα την δει η γραμματεία έτσι ώστε να εφαρμόσει την κατάλληλη διαδικασία ιχνηλάτησης σύμφωνα με το πρωτόκολλο υγείας.

Use Cases γραμματείας:

1. Το πρώτο Use Case που αφορά την γραμματεία είναι το login καθώς η γραμματεία πρέπει να κάνει login στο σύστημα έτσι ώστε να μπορεί να το χρησιμοποιήσει. Για το login έχει το δικό της λογαριασμό που είναι ο admin του συγκεκριμένου συστήματος.
2. Το δεύτερο Use Case αφορά την εισαγωγή των στοιχείων των διαλέξεων όπως την ημερομηνία.
3. Το τρίτο Use Case αφορά την εισαγωγή των στοιχείων της κάθε αίθουσας δηλαδή το όνομα, την χωρητικότητα και τον όροφο της.
4. Στο τελευταίο Use Case η γραμματεία προβάλλει τα θετικά κρούσματα που έχει δηλώσει ο κάθε φοιτητής και βλέπει σε ποιες διαλέξεις έχει πάει έτσι ώστε να γίνει η ιχνηλάτηση και να βρει όλους τους άλλους φοιτητές που είχαν καθίσει σε κοντινή θέση.

1.3 Class Diagram



Εικόνα 1.2 - Class Diagram

Στο Class Diagram της UML βλέπουμε τις κλάσεις που χρειάζονται για να υλοποιήσουμε το σύστημα έτσι ώστε να είναι λειτουργικό και να μπορούμε να συλλέξουμε τα κατάλληλα δεδομένα από τον κάθε χρήστη.

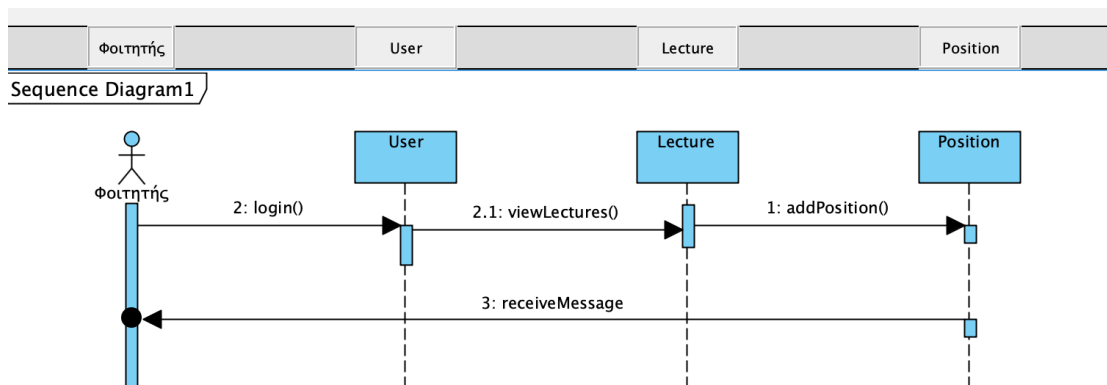
1. **User Class**: Η πρώτη κλάση αφορά τον χρήστη του συστήματος που είναι ο φοιτητής ο οποίος χρησιμοποιεί το σύστημα και έχει username και password σαν στοιχεία.
2. **CovidCases Class**: Η δεύτερη κλάση αφορά τους φοιτητές που έχουν διαγνωστεί με κορονοϊό και έχουν σαν δεδομένα έναν χρήστη δηλαδή έναν φοιτητή και την ημερομηνία διάγνωσης.

3. **Amphitheater Class:** Η τρίτη κλάση έχει τα αμφιθέατρα με τα στοιχεία τους δηλαδή το όνομα, την χωρητικότητα και τον όροφο του κάθε αμφιθεάτρου.
4. **Course Class:** Η τέταρτη κλάση αφορά την διάλεξη που χωρίς την ημερομηνία και έχει το όνομα του μαθήματος, τις ώρες έναρξης και λήξης το αμφιθέατρο που θα γίνει, το αν είναι υποχρεωτικό ή όχι το μάθημα καθώς και το όνομα του καθηγητή.
5. **Lecture Class:** Η πέμπτη κλάση αφορά την ημερομηνία που θα γίνει η κάθε διάλεξη για αυτό και έχει την ημερομηνία σαν στοιχεία και τα στοιχεία ενός μαθήματος που θα γίνει την συγκεκριμένη ημερομηνία.
6. **Position Class:** Η τελευταία κλάση έχει την θέση που έχει συμπληρώσει ο φοιτητής και έχει κοινά κλειδιά με τον User δηλαδή τον φοιτητή και την διάλεξη που έχει παρακολουθήσει ο κάθε φοιτητής. Το μόνο δεδομένο που δεν έχει σαν κοινό κλειδί είναι η θέση που συμπληρώνει ο κάθε φοιτητής.

1.4 Sequence Diagrams

Στην εργασία χρησιμοποιήθηκαν μερικά Sequence Diagrams για κάθε Use Case φοιτητή. Ένα Sequence Diagram δείχνει τις διεργασίες που διατάσσονται ανά χρονικό διάστημα σε ένα πληροφοριακό σύστημα. Απεικονίζει τις διαδικασίες που εμπλέκονται και τη σειρά των μηνυμάτων που ανταλλάσσονται μεταξύ των διαδικασιών που απαιτούνται για την εκτέλεση της λειτουργικότητας.

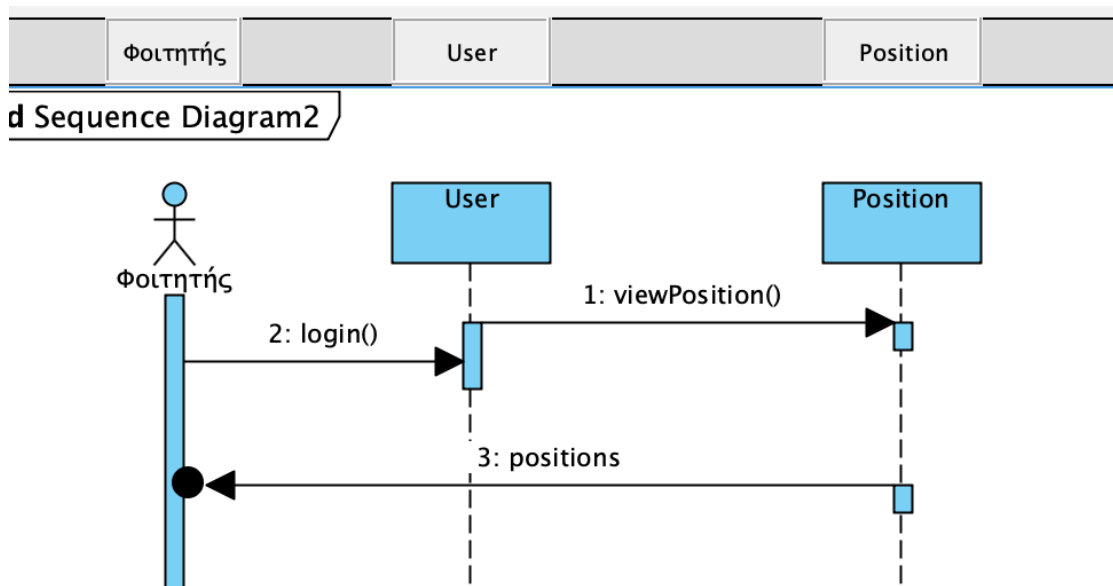
1. Sequence διάγραμμα για προσθήκη θέση του φοιτητή



Εικόνα 1.3 - Sequence Diagram 1

Το συγκεκριμένο διάγραμμα αφορά στο Use Case της προσθήκη θέσης του φοιτητή. Στην αρχή ο φοιτητής κάνει login στο σύστημα όποτε τα στοιχεία του ελέγχονται στην βάση με τους χρήστες. Επειτα βλέπει τις διαλέξεις την σημερινής ημερομηνίας, συμπληρώνει την θέση στην οποία κάθεται και τέλος λαμβάνει το κατάλληλο μήνυμα σε περίπτωση λάθους (π χ αν έχει ξανασυμπληρώσει την ίδια θέση).

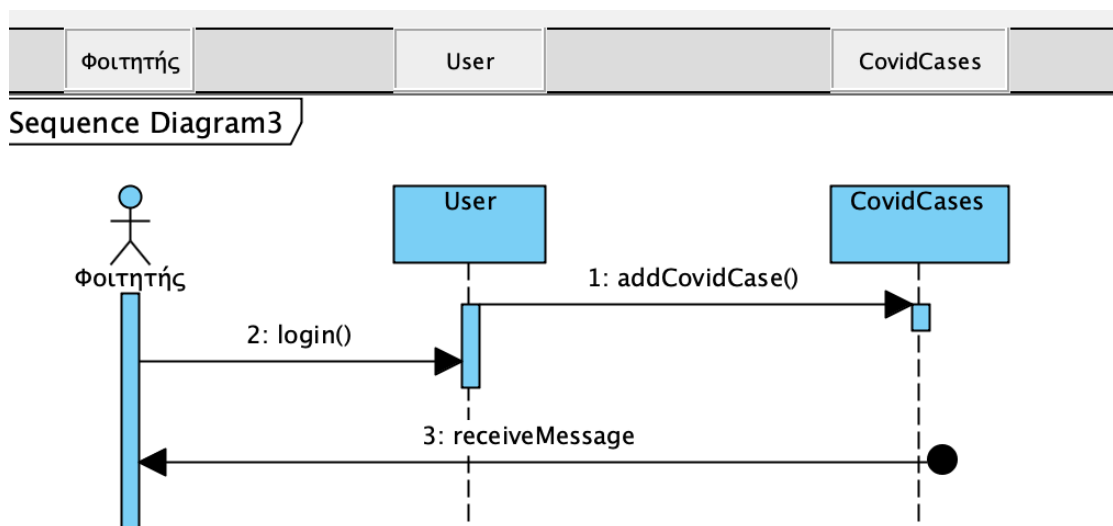
2. Sequence διάγραμμα για την προβολή στοιχείων του φοιτητή



Εικόνα 1.4 - Sequence Diagram 2

Το συγκεκριμένο διάγραμμα αφορά το Use Case του φοιτητή που προβάλλει τις θέσεις στις οποίες κάθεται ανά διάλεξη. Ο φοιτητής κάνει login και επιλέγει την προβολή θέσεων του σε κάθε διάλεξη. Τέλος το σύστημα του επιστρέφει μια σελίδα με τα στοιχεία των θέσεων σε κάθε διάλεξη.

3. Sequence διάγραμμα για δήλωση κρούσματος



Εικόνα 1.5 - Sequence Diagram 3

Το συγκεκριμένο διάγραμμα αφορά το Use Case που δηλώνει πως είναι θετικός στον Covid-19. Ο φοιτητής κάνει login στο σύστημα και μετά πατάει την επιλογή δήλωση θετικού κρούσματος. Στην σελίδα που εμφανίζεται εισάγει την ημερομηνία που διαγνώστηκε θετικός και τέλος το σύστημα του στέλνει το κατάλληλο μήνυμα.

1.5 Μη λειτουργικές απαιτήσεις

Οι μη λειτουργικές απαιτήσεις αφορούν το πως το σύστημα θα στηρίξει τις λειτουργικές απαιτήσεις όπως π.χ. την απόδοση, την χρηστικότητα και την ασφάλεια.

Στις μη λειτουργικές απαιτήσεις έχουμε τις εξής:

1. Ο χρόνο απόκρισης του συστήματος δεν πρέπει να είναι μεγαλύτερος από 2 δευτερόλεπτα. Το σύστημα πρέπει να είναι σχετικά γρήγορο.
2. Το σύστημα πρέπει να λειτουργεί όλη την ώρα ακόμα και τις ώρες που το πανεπιστήμιο δεν λειτουργεί.
3. Μόνο η γραμματεία πρέπει να έχει πρόσβαση στο admin μενού και κανένας εξωτερικός χρήστης.
4. Πρέπει να υπάρχει η ανάλογη προστασία προσωπικών δεδομένων των φοιτητών, έτσι ώστε μόνο η γραμματεία να μπορεί να δει τα δεδομένα της θέσης.
5. Το σύστημα πρέπει να εξυπηρετεί όλους τους φοιτητές του χαροκοπείου πανεπιστημίου δηλαδή πάνω από 2000 χρήστες.
6. Το σύστημα έχει γίνει deploy στο cloud και δεν υπάρχει server στο πανεπιστήμιο που να τρέχει. Επομένως σε περιπτώσεις σφάλματος θα πρέπει να επικοινωνούμε με τον πάροχο των διαδικτυακών υπηρεσιών μας.

ΚΕΦ.2: Υλοποίηση εφαρμογής

Ο κώδικας της εφαρμογής βρίσκεται στο link: <https://github.com/panosit/thesis>

2.1. Python

Η Python είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού υψηλού επιπέδου. Είναι διερμηνευόμενη δηλαδή interpreted που σημαίνει πως η υλοποίηση της αποτελείται από έναν διερμηνέα. Αντιθέτως οι γλώσσες που είναι μεταγλωττισμένες χρησιμοποιούν έναν μεταγλωττιστή (compiler) η οποία διαβάζει κώδικα γραμμένο σε μια γλώσσα προγραμματισμού και τον μεταφράζει σε ισοδύναμο κώδικα σε μια άλλη γλώσσα προγραμματισμού. Επιπλέον η Python υποστηρίζει συλλογή απορριμμάτων έχει δηλαδή δικό της garbage collector που σημαίνει ότι η διαχείριση μνήμης γίνεται από την γλώσσα προγραμματισμού και όχι από εμάς όπως π χ η C. Πλέον χρησιμοποιούμε την έκδοση 3.0 και όχι την 2.0 που υπήρξε παλαιότερα.

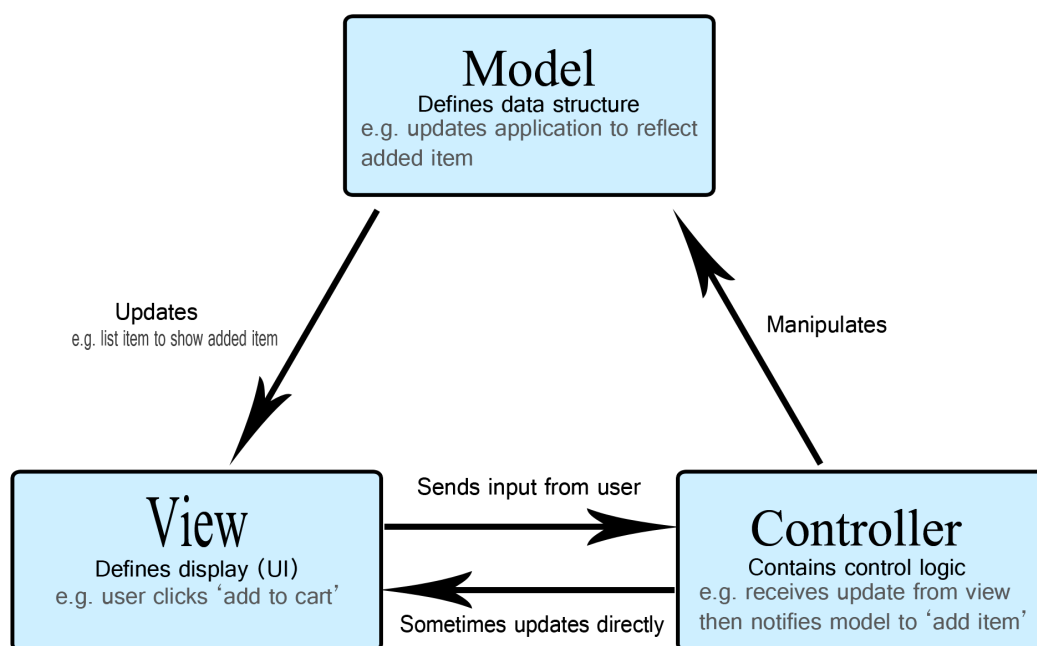
Στα ιστορικά στοιχεία της γλώσσας, η Python δημιουργήθηκε από τον Ολλανδό Γκουίντο βαν Ρόσσουμ το 1989 και κυκλοφόρησε για πρώτη φορά το 1991. Ο λόγος που δημιουργήθηκε ήταν για να υπάρχει μια πιο εύκολη γλώσσα προγραμματισμού σε σχέση με τις άλλες εκείνη την εποχή όπως η C++. Έχει αρκετές βιβλιοθήκες και frameworks όπως το Django και αυτός είναι ένας από τους κύριους λόγους που έγινε δημοφιλής.

Τα αρνητικά της Python είναι ότι είναι πιο αργή σε σχέση με άλλες γλώσσες προγραμματισμού για αυτό και δεν χρησιμοποιείται για low level προγραμματισμό όπως π χ τα λειτουργικά συστήματα. Είναι open source και διατίθεται από την συγκεκριμένη ιστοσελίδα (<https://www.python.org/>). Έχει αρκετές χρήσεις από την τεχνητή νοημοσύνη μέχρι τον προγραμματισμό web εφαρμογών που έχει χρησιμοποιηθεί για αυτή την πτυχιακή.

Τέλος αποτελεί μια από τις πιο δημοφιλέστερες γλώσσες προγραμματισμού γιατί είναι αρκετά εύκολη στην εκμάθηση της.

2.2. Django

Το Django είναι ένα Python web framework υψηλού επιπέδου. Χρησιμοποιεί την αρχιτεκτονική του MVC (Model-View-Controller) δηλαδή το πρόγραμμα “σπάει” σε τρεις λειτουργίες: 1. Model: Η συγκεκριμένη λειτουργία διαχειρίζεται τα δεδομένα της εφαρμογής. 2. View: Η συγκεκριμένη λειτουργία εμφανίζει στον χρήστη τις πληροφορίες σε οποιαδήποτε μορφή όπως π.χ ένα διάγραμμα ή ένα πίνακα 3. Controller: Η συγκεκριμένη λειτουργία δέχεται το input και ανάλογα με τα commands εμφανίζει το ανάλογο αποτέλεσμα.



Εικόνα 2.1 - MVC Diagram

Το MVC το χρησιμοποιούν πολλές διαδικτυακές εφαρμογές διότι μας βοηθάει να κατασκευάσουμε εύκολα διαδικτυακές (web) εφαρμογές.

Το Django είναι δωρεάν και είναι open source. Μπορούμε να το κατεβάσουμε στην ιστοσελίδα (<https://www.djangoproject.com/download/>). Έρχεται με πολλές λειτουργίες όπως το δικό ORM που θα εξηγήσω αργότερα, admin panel έτσι ώστε ο διαχειριστής να μπορεί να προσθέτει δεδομένα στην βάση με σχετική ευκολία και με δικά του settings που μας βοηθάνε στο πως θα ανεβάσουμε την εφαρμογή σε κάποιο server ή το που θα αποθηκεύσουμε τα στατικά αρχεία του project.

Είναι αρκετά ασφαλές και μας προστατεύει από επιθέσεις τύπου SQL Injection. Διάσημες εφαρμογές που χρησιμοποιούν το συγκεκριμένο framework είναι το Instagram η ο Mozilla.

2.3. SQLite και PostgreSQL

Στην συγκεκριμένη πτυχιακή όταν η εφαρμογή “τρέχει” τοπικά στον υπολογιστή τότε χρησιμοποιεί SQLite ενώ όταν γίνεται deploy σε κάποιο cloud ή σε κάποιον server τότε χρησιμοποιεί PostgreSQL. Επιπλέον και 2 χρησιμοποιούν την SQL για την διαχείριση των δεδομένων.

Η SQL είναι μια γλώσσα για την διαχείριση σχεσιακών βάσεων δεδομένων, περιλαμβάνει δυνατότητες προβολής δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα. Χρησιμοποιείται για τα CRUD Operations δηλαδή Create, Read, Update και Delete.

Η SQLite είναι μια σχεσιακή βάση δεδομένων που τρέχει τοπικά στον υπολογιστή μας η το κινητό μας. Χρησιμεύει για να αποθηκεύσουμε δεδομένα τοπικά χωρίς να βρίσκονται κάπου στο διαδίκτυο. Αυτός είναι και ο λόγος που την χρησιμοποιώ σε αυτή την πτυχιακή όταν το project τρέχει localhost και δεν είναι στο διαδίκτυο.

Η PostgreSQL είναι μια σχεσιακή βάση δεδομένων open source. Την χρησιμοποιώ στην συγκεκριμένη πτυχιακή έτσι ώστε να αποθηκεύει τα δεδομένα ενώ το project βρίσκεται στο διαδίκτυο και λειτουργεί κανονικά.

2.4. Django ORM

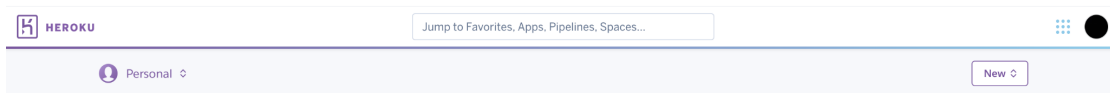
Το ORM είναι ο τρόπος που χρησιμοποιούμε αντικειμενοστραφή προγραμματισμό μαζί με μια βάση δεδομένων. Στην ουσία κάθε table στην βάση αποτελεί και ένα αντικείμενο της Python ή κάποιας αντικειμενοστραφής γλώσσας προγραμματισμού και έτσι μας είναι πιο εύκολο να επεξεργαστούμε πληροφορίες χωρίς να γράψουμε SQL εντολές. Αυτό γίνεται διότι το Framework του Django μας παρέχει ORM και χρησιμοποιούμε python για την επεξεργασία δεδομένων.

2.5. Deploy Εφαρμογής (Heroku)

Για να κάνω deploy την εφαρμογή χρησιμοποίησα μια cloud υπηρεσία που παρέχει η Salesforce (<https://www.salesforce.com/eu/>) το heroku (<https://www.heroku.com/home>). Η ιστοσελίδα που είναι ανεβασμένο το συγκεκριμένο project είναι η <https://thesishua.herokuapp.com/>. Το heroku μας βοηθάει να ανεβάσουμε μια εφαρμογή στον παγκόσμιο ιστό προσφέροντας μας ένα περιβάλλον έτσι ώστε να κάνουμε upload τον κώδικα μας και να “τρέξει” σε

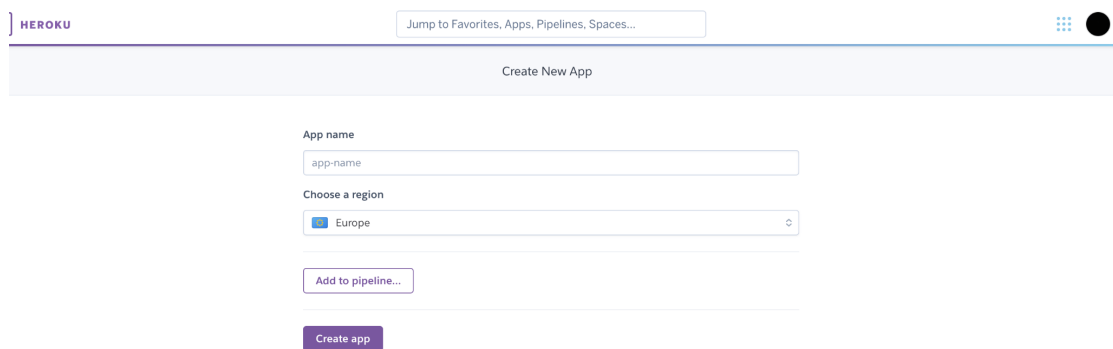
έναν απομακρυσμένο server. Ο τρόπος που έκανα deploy την συγκεκριμένη πτυχιακή είναι ο εξής:

1. Για αρχή πρέπει να φτιάξουμε λογαριασμό στην πλατφόρμα μέσω της συγκεκριμένης ιστοσελίδας (<https://signup.heroku.com/login>).
2. Μετά κάνουμε login και επιλέγουμε το new έτσι ώστε να φτιάξουμε το project.



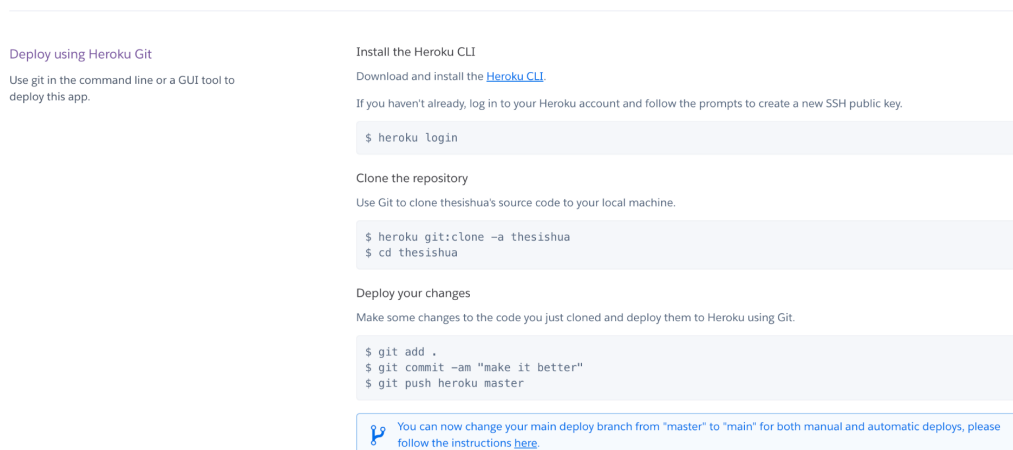
Εικόνα 2.2 - Heroku Login

3. Επιλέγουμε το κατάλληλο region και το όνομα της εφαρμογής μας.



Εικόνα 2.3 - Create New App

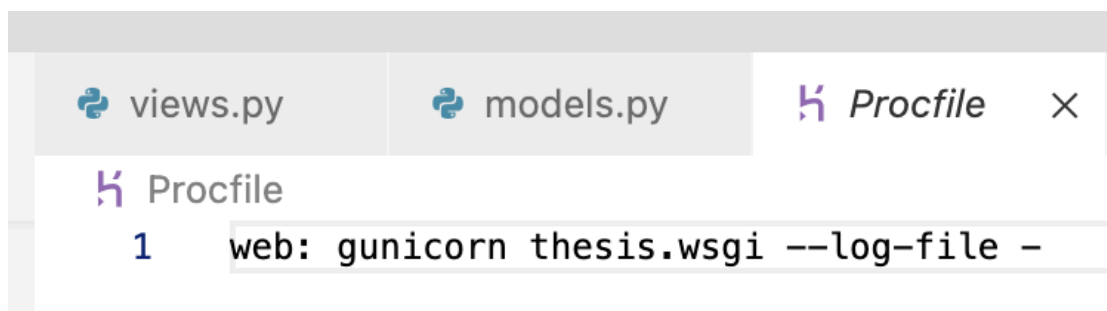
4. Προϋπόθεση για να κάνουμε upload την εφαρμογή είναι να έχουμε κατεβάσει το heroku cli (<https://devcenter.heroku.com/articles/heroku-cli>).
5. Τέλος κάνουμε login μέσω του command **heroku login**. Εκεί θα ανοίξει ένας web browser που θα πρέπει να βάλουμε τα στοιχεία του λογαριασμού μας. Μετά χρησιμοποιούμε τις κατάλληλες εντολές του **git** που μας δείχνει η φωτογραφία και το project γίνεται deploy.



Εικόνα 2.4 - Upload App

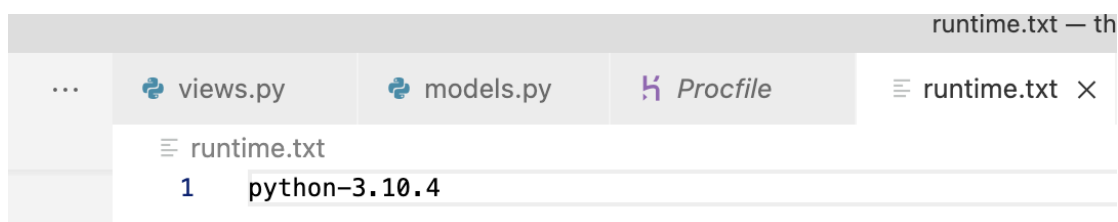
Επειδή πρόκειται για Django (Python) project πρέπει να κάνουμε και κάποιες αλλαγές στον κώδικα μας όπως

1. Την δημιουργία **Procfile** που θα περιέχει το όνομα της εφαρμογής.



Εικόνα 2.5 - Heroku Procfile

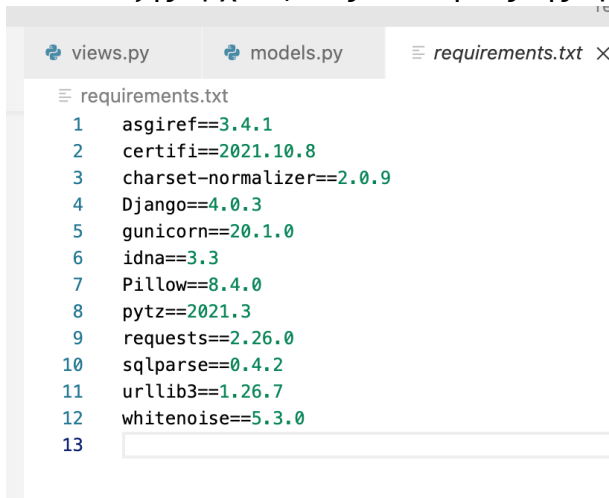
2. Την δημιουργία ενός **runtime.txt** αρχείου που θα περιέχει την έκδοση της Python με την οποία τρέχει.



Εικόνα 2.6 - Create runtime.txt

3. Να κάνουμε install 2 βιβλιοθήκες της python την gunicorn (<https://gunicorn.org/>) και το whitenoise middleware (<http://whitenoise.evans.io/en/stable/>).

4. Μόλις κάνουμε install τις 2 συγκεκριμένες βιβλιοθήκες εκτελούμε την εντολή **pip freeze > requirements.txt** . Η συγκεκριμένη εντολή μας παράγει το εξής αρχείο με τις απαιτήσεις της εφαρμογής.



```
views.py  models.py  requirements.txt x
requirements.txt
1  asgiref==3.4.1
2  certifi==2021.10.8
3  charset-normalizer==2.0.9
4  Django==4.0.3
5  gunicorn==20.1.0
6  idna==3.3
7  Pillow==8.4.0
8  pytz==2021.3
9  requests==2.26.0
10 sqlparse==0.4.2
11 urllib3==1.26.7
12 whitenoise==5.3.0
13
```

Εικόνα 2.7 - Create requirements.txt

5. Μόλις ολοκληρώσουμε τα συγκεκριμένα βήματα μένουν κάποιες αλλαγές στο αρχείο **settings.py** και μπορούμε να κάνουμε deploy στο heroku. Οι συγκεκριμένες αλλαγές φαίνονται στην επόμενη φωτογραφία.

```
1  ∨ from pathlib import Path
2  import os
3
4  BASE_DIR=Path(__file__).resolve().parent.parent
5  SECRET_KEY='django-insecure-vg9=05b!l4g-dbnym4s@y9v&14c0+@7&c(o5u)eok^tp)3c$!8'
6  DEBUG=True
7  ALLOWED_HOSTS=['*']
8  ∨ INSTALLED_APPS=[
9      'covidtracker.apps.CovidtrackerConfig',
10     'django.contrib.admin',
11     'django.contrib.auth',
12     'django.contrib.contenttypes',
13     'django.contrib.sessions',
14     'django.contrib.messages',
15     'django.contrib.staticfiles',
16 ]
17 ∨ MIDDLEWARE=[
18     'whitenoise.middleware.WhiteNoiseMiddleware',
19     'django.middleware.security.SecurityMiddleware',
20     'django.contrib.sessions.middleware.SessionMiddleware',
21     'django.middleware.common.CommonMiddleware',
22     'django.middleware.csrf.CsrfViewMiddleware',
23     'django.contrib.auth.middleware.AuthenticationMiddleware',
24     'django.contrib.messages.middleware.MessageMiddleware',
25     'django.middleware.clickjacking.XFrameOptionsMiddleware',
26 ]
```

Εικόνα 2.8 - Change settings.py

Η πρώτη αλλαγή είναι στο allowed hosts να βάλω το * που σημαίνει ότι η εφαρμογή μπορεί να τρέξει παντού και όχι μόνο localhost. Η δεύτερη αλλαγή

είναι στα installed apps να βάλω το όνομα της εφαρμογής του django που έχω φτιάξει. Τέλος στο middleware έπρεπε να προσθέσω το whitenoise έτσι ώστε να λειτουργήσει η εφαρμογή.

Με την συγκεκριμένη πτυχιακή μπορούμε να καταλάβουμε και την σημασία του cloud computing. Το cloud computing έχει πάρα πολλά πλεονεκτήματα όπως το κόστος διότι άμα επέλεγα να ανεβάσω την εφαρμογή με κανονικό τρόπο θα έπρεπε να είχα αγοράσει κάποιον server και να είχα κάποια γραφεία ή κάποιον χώρο για να τον αποθηκεύσω. Με το cloud έχουμε κάποιο κόστος σε λεφτά αλλά είμαστε εντελώς ελεύθεροι στο από που και πως θα διαχειριστούμε το project. Επιπλέον δεν έχουμε περιορισμούς στο χώρο και στα δεδομένα καθώς ανα πάσα στιγμή μπορούμε να επεκτείνουμε το πρόγραμμά μας με τον cloud πάροχο μας.

ΚΕΦ.3: Σενάρια χρήσης

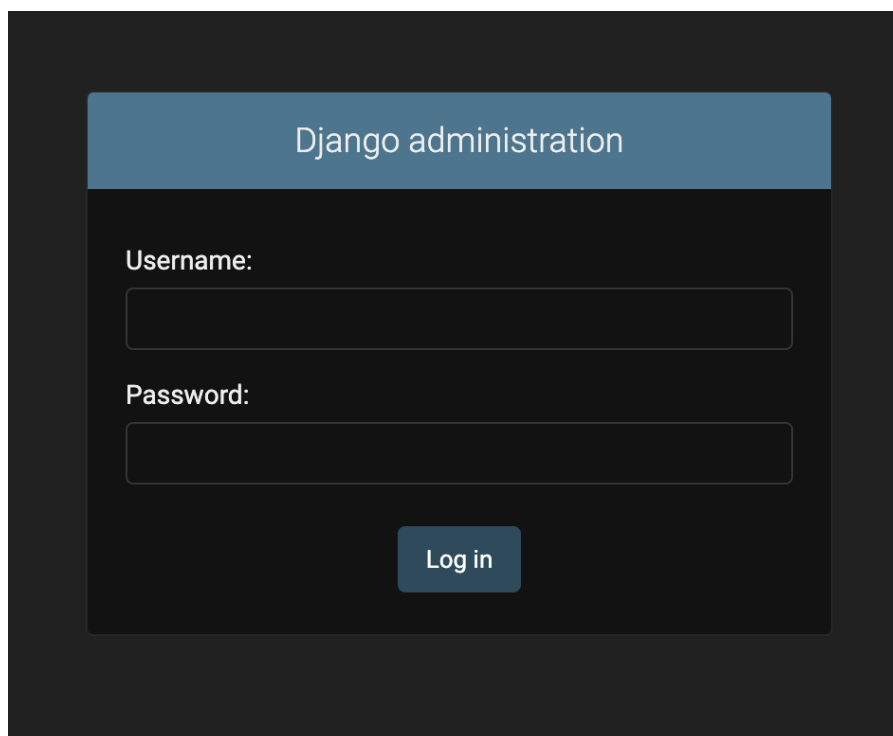
3.1. Γενικά

Στο συγκεκριμένο σύστημα οι φοιτητές ή η γραμματεία κάνουν login με το πανεπιστημιακό τους λογαριασμό. Δεν υπάρχει η δυνατότητα register καθώς χρησιμοποιούνται οι πανεπιστημιακοί λογαριασμοί. Επίσης η γραμματεία έχει διαφορετικό μενού από τον φοιτητή και έχει διαφορετικά operations σε σχέση με τον φοιτητή. Η γραμματεία χρησιμοποιεί το **/admin** url της εφαρμογής ενώ οι φοιτητές χρησιμοποιούν την απλή εφαρμογή.

Η εφαρμογή είναι προσβάσιμη μέσω ενός web browser. Οι φοιτητές συμπληρώνουν τα κατάλληλα στοιχεία που θα δείξω μετά. Η γραμματεία κάνει το ίδιο αλλά είναι υπεύθυνη σε περίπτωση κρούσματος να δει τα στοιχεία στην βάση δεδομένων και να στείλει κάποιο email ή να καλέσει τους φοιτητές να παραμείνουν σε καραντίνα.

3.2 Γραμματεία (Admin)

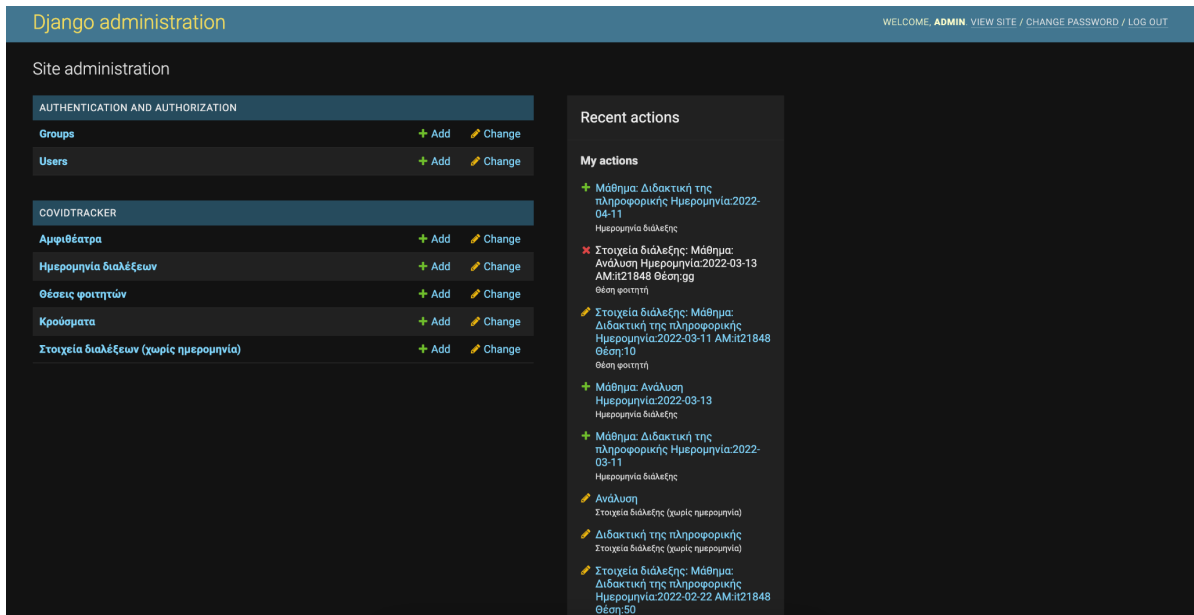
Η γραμματεία έχει δικό της login page, του admin.



Εικόνα 3.1 - Admin login screen

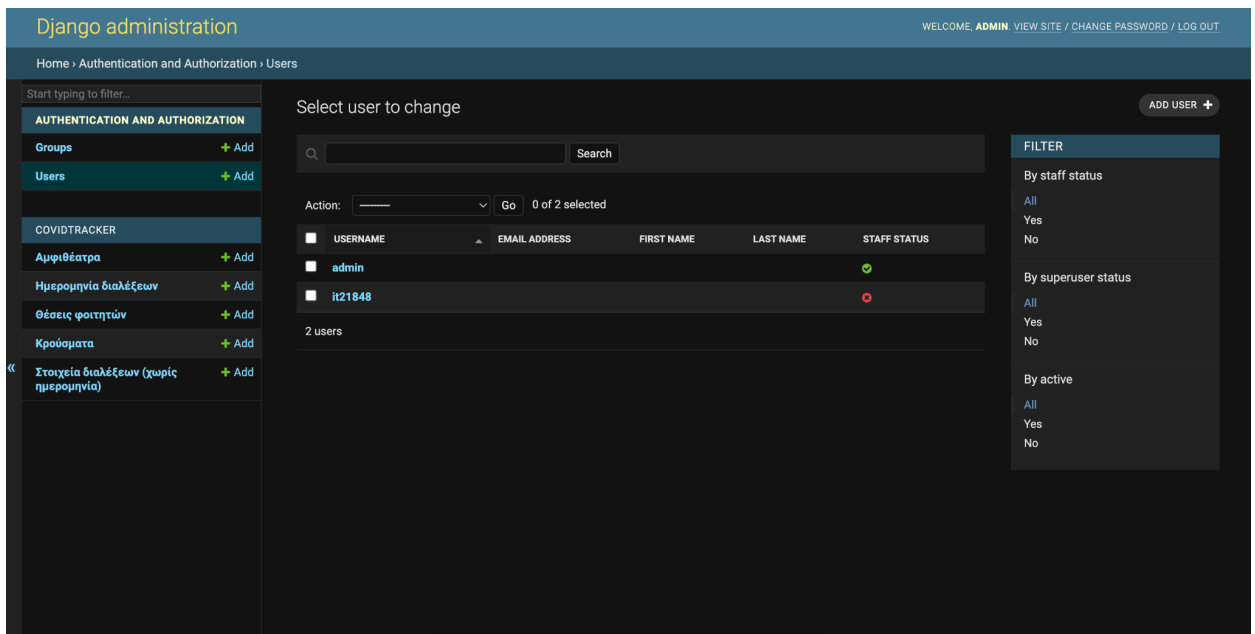
Στην παραπάνω φωτογραφία φαίνεται το admin page όπου εισέρχεται με τα στοιχεία του ακαδημαϊκού λογαριασμού της. Οι φοιτητές δεν έχουν πρόσβαση στην συγκεκριμένη σελίδα για λόγους ασφαλείας.

Μόλις κάνει login η γραμματεία βλέπει το admin μενού το οποίο περιέχει όλα τα δεδομένα της εφαρμογής.



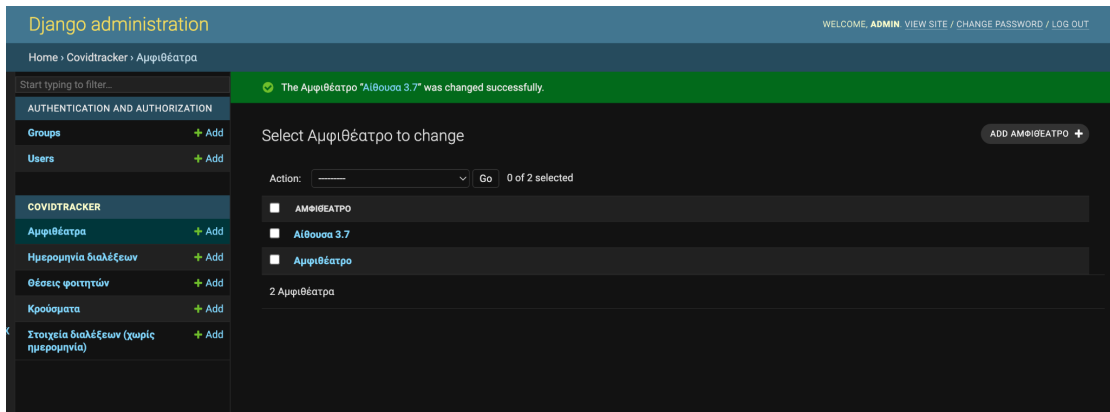
Εικόνα 3.2 - Admin home page

Στην παραπάνω φωτογραφία φαίνονται όλα τα δεδομένα της εφαρμογής όπως τα αμφιθέατρα που έχει το πανεπιστήμιο και τους χρήστες. Η γραμματεία επιλέγει την κατάλληλη επιλογή έτσι ώστε να επεξεργαστεί ή να δει τα δεδομένα στο κάθε table της βάσης δεδομένων.



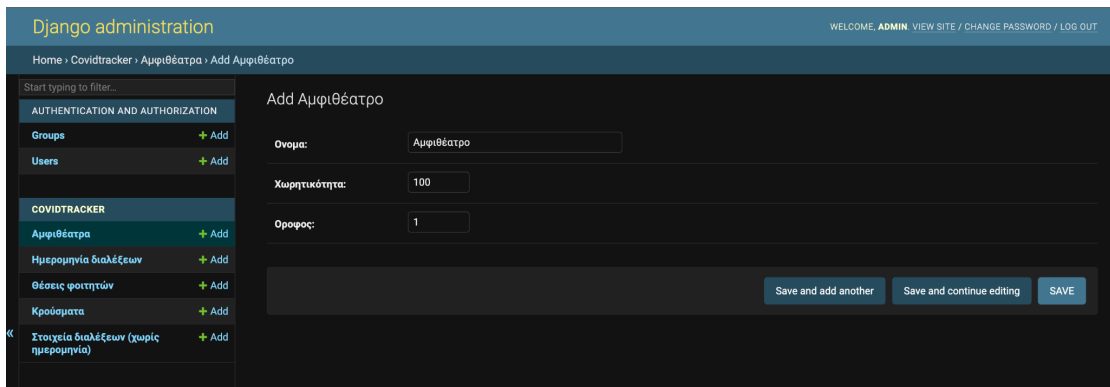
Εικόνα 3.3 - Σελίδα προβολής χρηστών

Η πρώτη επιλογή μας δείχνει τους χρήστες της εφαρμογής και τους αριθμούς μητρώων τους. Στο συγκεκριμένο παράδειγμα έχει μόνο έναν φοιτητή καθώς η εφαρμογή είναι σε δοκιμαστικό στάδιο.



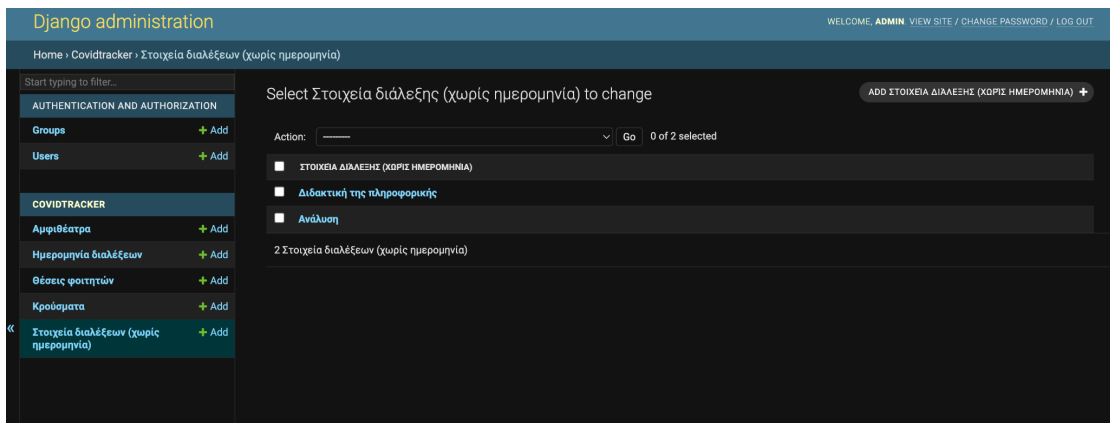
Εικόνα 3.4 - Σελίδα προβολής αμφιθεάτρων

Η δεύτερη επιλογή έχει τα αμφιθέατρα. Εκεί η γραμματοεργασία μπορεί να προσθέσει ένα αμφιθέατρο ή να δει τα στοιχεία του.



Εικόνα 3.5 - Σελίδα δημιουργίας αμφιθεάτρου

Στη συγκεκριμένη σελίδα η γραμματοεργασία προσθέτει τα στοιχεία του αμφιθεάτρου, το όνομα του, την χωρητικότητά του και τέλος τον όροφο που βρίσκεται.



Εικόνα 3.6 - Σελίδα προβολής διαλέξεων

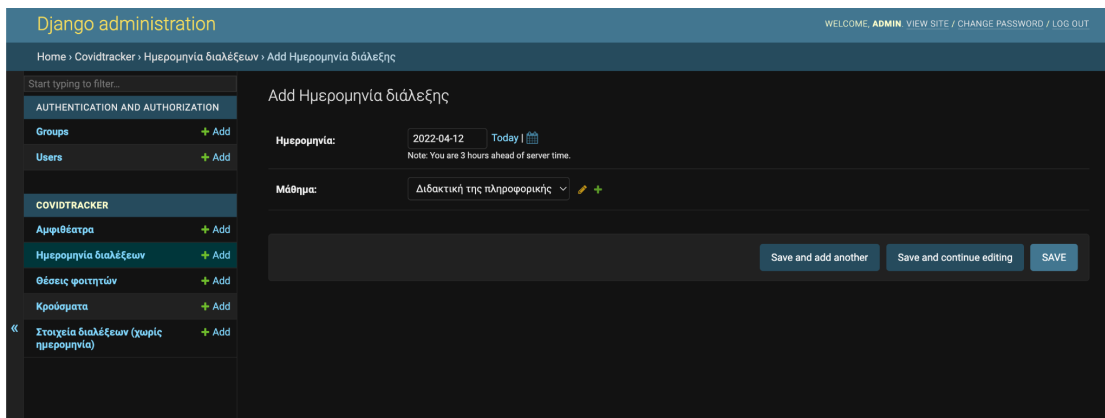
Η τρίτη επιλογή έχει την προβολή των στοιχείων κάθε διάλεξης χωρίς ημερομηνία. Εκεί η γραμματεία μπορεί να προσθέσει μια διάλεξη ή να δει τα στοιχεία της.

Εικόνα 3.7 - Σελίδα δημιουργίας διάλεξης

Στο συγκεκριμένο παράδειγμα η γραμματεία προσθέτει την ώρα έναρξης, την ώρα λήξης, τον καθηγητή, το μάθημα, τον τύπο μαθήματος και τέλος το αμφιθέατρο για κάθε μάθημα. Η γραμματεία δεν σημειώνει την ημερομηνία του κάθε μαθήματος διότι υπάρχει ειδική επιλογή για αυτό έτσι ώστε να μην επαναλαμβάνονται τα στοιχεία των διαλέξεων.

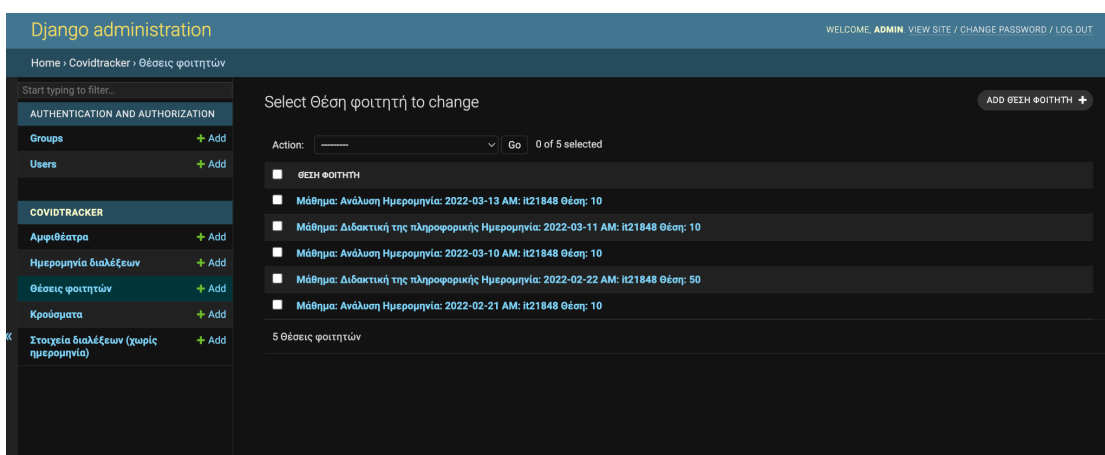
Εικόνα 3.8 - Σελίδα προβολής ημερομηνίας διάλεξης

Η τέταρτη επιλογή έχει την προβολή της ημερομηνίας διαλέξεων και του μαθήματος που έχει γίνει. Εκεί η γραμματεία επιλέγει το πότε θα πραγματοποιηθεί η κάθε διάλεξη.



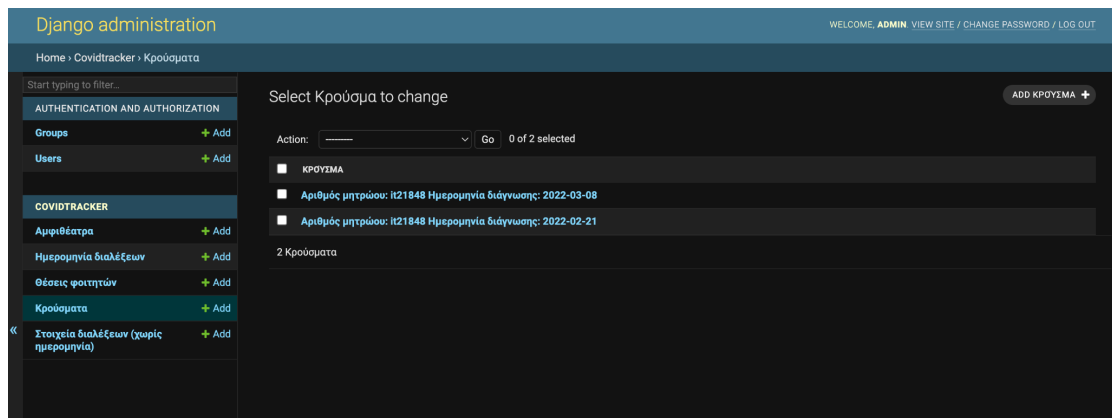
Εικόνα 3.9 - Σελίδα δημιουργίας ημερομηνίας διάλεξης

Στο συγκεκριμένο μενού η γραμματεία επιλέγει την ημερομηνία που θα πραγματοποιηθεί το κατάλληλο μάθημα και το όνομα του μαθήματος. Το μάθημα έχει δημιουργηθεί από το παραπάνω μενού στα στοιχεία διαλέξεων. Στον φοιτητή εμφανίζονται τα μαθήματα μόνο της σημερινής ημέρας.



Εικόνα 3.10 - Σελίδα προβολής θέσεων φοιτητή

Η πέμπτη επιλογή μας δείχνει την θέση του κάθε φοιτητή ανά ημερομηνία διάλεξης στο κάθε μάθημα. Αυτή η επιλογή χρησιμοποιείται για να γίνει η ιχνηλάτηση κρουσμάτων σε περίπτωση κρούσματος έτσι ώστε να βρεθούν όλα τα κρούσματα. Η γραμματεία βλέπει σε ποια θέση κάθεται ο κάθε φοιτητής έτσι ώστε να γίνει η ιχνηλάτηση.



Εικόνα 3.11 - Σελίδα προβολής θετικών κρουσμάτων

Στην τελευταία επιλογή η γραμματεία βλέπει τα κρούσματα και την ημερομηνία διάγνωσης του κάθε κρούσματος. Στην συγκεκριμένη εφαρμογή ο κάθε φοιτητής έχει την δυνατότητα να δηλώσει πότε θα είναι θετικό κρούσμα ενώ μέχρι τώρα έπαιρνε τηλέφωνο την γραμματεία για να την ειδοποιήσει.

3.3 Φοιτητής (User)

Στην πλοήγηση του home screen της εφαρμογής ο χρήστης βλέπει το login page της εφαρμογής.

Hua Covid-19 Tracker

Username (**without** @hua.gr)

Password

Sign in

Εικόνα 3.12 - Student login screen

Εκεί εισάγει τα στοιχεία του ακαδημαϊκού λογαριασμού του χωρίς το @hua.gr όπως διευκρινίζει η φωτογραφία. Δεν υπάρχει η δυνατότητα register στην εφαρμογή καθώς ο χρήστης συνδέεται με τον ακαδημαϊκό του λογαριασμό που τα στοιχεία του είναι ήδη καταγεγραμμένα στο σύστημα.

HUA COVID-19 TRACKER

it21848

Δήλωση Κρούσματος Covid-19

Show All Data

Log out

Σημερινές Διαλέξεις

Μάθημα	Καθηγητής	Τύπος μαθήματος	Ημερομηνία	Ωρα έναρξης	Ωρα λήξης	Αίθουσα	Χώρητικότητα Αίθουσας
Διδακτική της πληροφορικής	Κλεοπάτρα Μπαρδάκη	Επιλογής	April 13, 2022	1 p.m.	4 p.m.	Αίθουσα 3.7	50

Εικόνα 3.13 - Σελίδα προβολής σημερινών διαλέξεων

Μόλις ο χρήστης κάνει login στο σύστημα βλέπει τις σημερινές διαλέξεις και τα στοιχεία τους όπως φαίνεται στην παραπάνω φωτογραφία. Σε αυτό με το μενού επιλέγει το link που δείχνει το μάθημα το οποίο παρακολουθεί ο φοιτητής.

HUA COVID-19 TRACKER							
it21848		Δήλωση Κρούσματος Covid-19		Show All Data		Log out	
Στοιχεία Διάλεξης							
Μάθημα	Καθηγητής	Τύπος μαθήματος	Ημερομηνία	Ωρα έναρξης	Ωρα λήξης	Αίθουσα	Χώρητικότητα Αίθουσας
Διδακτική της πληροφορικής	Κλεοπάτρα Μπαρδάκη	Επιλογής	April 13, 2022	1 p.m.	4 p.m.	Αίθουσα 3.7	50

Εισάγετε τον αριθμό της θέσης σας

Αριθμός θέσης

Εισαγωγή

Εικόνα 3.14 - Σελίδα εισαγωγής θέσης

Μόλις επιλέξει το link ανοίγει η παραπάνω σελίδα που δείχνει τα στοιχεία της διάλεξης, εκεί ο φοιτητής εισάγει το αριθμό η τα γράμματα της θέσης του. Σε περίπτωση που ο φοιτητής εισάγει την θέση του ενώ έχει εισάγει πιο πριν στην

ίδια διάλεξη την θέση του εμφανίζεται το κατάλληλο μήνυμα λάθους. Το ίδιο συμβαίνει και στην περίπτωση που κάποιος άλλος έχει εισάγει τον ίδιο αριθμό θέσης.

HUA COVID-19 TRACKER							
		it21848	Δήλωση Κρούσματος Covid-19	Show All Data	Log out		
Θέση	Μάθημα	Καθηγητής	Τύπος μαθήματος	Ημερομηνία	Ωρα Έναρξης	Ωρα λήξης	Αίθουσα
50	Διδακτική της πληροφορικής	Κλεοπάτρα Μπαρδάκη	Επιλογής	Feb. 22, 2022	1 p.m.	4 p.m.	Αίθουσα 3.7
10	Ανάλυση	Κλεοπάτρα Μπαρδάκη	Υποχρεωτικό	March 10, 2022	10 a.m.	1 p.m.	Αμφιθέατρο
10	Διδακτική της πληροφορικής	Κλεοπάτρα Μπαρδάκη	Επιλογής	April 13, 2022	1 p.m.	4 p.m.	Αίθουσα 3.7

Εικόνα 3.15 - Σελίδα προβολής δεδομένων διάλεξης

Στην επιλογή **Show All Data** ο φοιτητής βλέπει όλα τα δεδομένα που δείχνει η παραπάνω φωτογραφία. Στα δικά του δεδομένα έχει πρόσβαση μόνο αυτός και η γραμματεία. Δεν μπορεί να πειράξει τα συγκεκριμένα δεδομένα ο φοιτητής μόνο να τα προβάλει.

HUA COVID-19 TRACKER										
		it21848	Δήλωση Κρούσματος Covid-19	Show All Data	Log out					
Δήλωση Κρούσματος										
Ημερομηνία διάγνωσης										
dd/mm/yyyy										
Βεβαίωση Κρούσματος										

Εικόνα 3.16 - Σελίδα δήλωσης κρούσματος

Στην σελίδα της εφαρμογής **Δήλωση Κρούσματος Covid-19** ο φοιτητής έχει την δυνατότητα να δηλώσει ότι βρέθηκε θετικό στον κορονοϊό και να εισάγει την ημερομηνία που διαγνώστηκε όπως δείχνει η ακόλουθη φωτογραφία.

Δήλωση Κρούσματος

Ημερομηνία διάγνωσης

April 2022

↑

↓

M	T	W	T	F	S	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Clear

Today

Βεβαίωση Κρούσματος

Εικόνα 3.17 - Σελίδα δήλωσης κρούσματος (ημερομηνία)

Το μενού του φοιτητή αποτελείται από τις 5 παραπάνω σελίδες και καλύπτει πλήρως την συλλογή δεδομένων από τους φοιτητές που μέχρι τώρα γινόταν με μια καρτέλα. Επιπλέον δίνει την δυνατότητα στους φοιτητές να προβάλουν τα δεδομένα τους, αλλά και την ηλεκτρονική δήλωση θετικού κρούσματος Covid-19.

3.4 Χρήση ερωτηματολογίου

Για την αξιολόγηση της εφαρμογής χρησιμοποιήθηκε ερωτηματολόγιο σε πιθανούς χρήστες της (φοιτητές). Οι ερωτήσεις ήταν οι εξής:

Πως σας φαίνεται το Login Page της εφαρμογής? Υπάρχει κάποια αλλαγή που θα κάνατε?

Hua Covid-19 Tracker

Username (without @hua.gr)

Password

Sign in

Κείμενο μακροσκελούς απάντησης

Εικόνα 3.18 - Πρώτη ερώτηση ερωτηματολογίου

Πως σας φαίνεται η σελίδα επιλογής των διαλέξεων της εφαρμογής? Υπάρχει κάποια αλλαγή που θα κάνατε?

HUA COVID-19 TRACKER

it21848

Δήλωση Κρούσματος Covid-19

Show All Data

Log out

Σημερινές Διαλέξεις

Μάθημα	Καθηγητής	Τύπος μαθήματος	Ημερομηνία	Ωρα έναρξης	Ωρα λήξης	Αίθουσα	Χώρητικότητα Αίθουσας	Οροφος
Ανάλυση	Κλεοπάτρα Μπαρδάκη	Υποχρεωτικό	March 16, 2022	10 a.m.	1 p.m.	Αμφιθέατρο	100	1

Κείμενο μακροσκελούς απάντησης

Εικόνα 3.19 - Δεύτερη ερώτηση ερωτηματολογίου

Πως σας φαίνεται η σελίδα εισαγωγής θέσης φοιτητή της εφαρμογής (ανάλογα με την διάλεξη που έχουμε επιλέξει)? Υπάρχει κάποια αλλαγή που θα κάνατε?

HUA COVID-19 TRACKER

it21848

Δήλωση Κρούσματος Covid-19

Show All Data

Log out

Στοιχεία Διάλεξης

Μάθημα	Καθηγητής	Τύπος μαθήματος	Ημερομηνία	Ωρα έναρξης	Ωρα λήξης	Αίθουσα	Χώρητικότητα Αίθουσας	Οροφι
Ανάλυση	Κλεοπάτρα Μπαρδάκη	Υποχρεωτικό	March 16, 2022	10 a.m.	1 p.m.	Αμφιθέατρο	100	1

Εισάγετε τον αριθμό της θέσης σας

Αριθμός θέσης

Εισαγωγή

Κείμενο μακροσκελούς απάντησης

Εικόνα 3.20 - Τρίτη ερώτηση ερωτηματολογίου

Πως σας φαίνεται η σελίδα προβολής στοιχείων της εφαρμογής? Υπάρχει κάποια αλλαγή που θα κάνατε?

HUA COVID-19 TRACKER							
it21848							
Δήλωση Κρούσματος Covid-19							
Show All Data							
Log out							
Θέση	Μάθημα	Καθηγητής	Τύπος μαθήματος	Ημερομηνία	Ωρα έναρξης	Ωρα λήξης	Αίθουσα
10	Ανάλυση	Κλεοπάτρα Μπαρδάκη	Υποχρεωτικό	Feb. 21, 2022	10 a.m.	1 p.m.	Αμφιθέατρο
50	Διδακτική της πληροφορικής	Κλεοπάτρα Μπαρδάκη	Επιλογής	Feb. 22, 2022	1 p.m.	4 p.m.	Αίθουσα 3.7
10	Ανάλυση	Κλεοπάτρα Μπαρδάκη	Υποχρεωτικό	March 10, 2022	10 a.m.	1 p.m.	Αμφιθέατρο
10	Διδακτική της πληροφορικής	Κλεοπάτρα Μπαρδάκη	Επιλογής	March 11, 2022	1 p.m.	4 p.m.	Αίθουσα 3.7
10	Ανάλυση	Κλεοπάτρα Μπαρδάκη	Υποχρεωτικό	March 13, 2022	10 a.m.	1 p.m.	Αμφιθέατρο

Κείμενο μακροσκελούς απάντησης

Εικόνα 3.21 - Τέταρτη ερώτηση ερωτηματολογίου

Πως σας φαίνεται η σελίδα δήλωσης κρούσματος με βάση την ημερομηνία διάγνωσης? Υπάρχει κάποια αλλαγή που θα κάνατε?

HUA COVID-19 TRACKER

it21848

Δήλωση Κρούσματος Covid-19

Show All Data

Log out

Δήλωση Κρούσματος

Ημερομηνία διάγνωσης

dd/mm/yyyy

Βεβαίωση Κρούσματος

Κείμενο μακροσκελούς απάντησης

Εικόνα 3.22 - Πέμπτη ερώτηση ερωτηματολογίου

Κατά την γνώμη σας χρειάζεται κάποια αλλαγή στην εφαρμογή? Λείπει κάποιο στοιχείο?

Κείμενο μακροσκελούς απάντησης

Εικόνα 3.23 - Έκτη ερώτηση ερωτηματολογίου

Το ερωτηματολόγιο χρησιμοποιήθηκε έτσι ώστε να δω άμα είναι λειτουργική η εφαρμογή και οι φοιτητές μπορούν να την χρησιμοποιήσουν με ευκολία.

3.5 Απαντήσεις ερωτηματολογίου

Για τις απαντήσεις του ερωτηματολογίου ελήφθησαν δείγματα από 3 φοιτητές.

Πως σας φαίνεται το Login Page της εφαρμογής? Υπάρχει κάποια αλλαγή που θα κάνατε?

3 απαντήσεις

Ικανοποιητικό

Ίσως με περισσότερα χρώματα

Κατατοπιστικό

Εικόνα 3.24 - Πρώτη απάντηση ερωτηματολογίου

Πως σας φαίνεται η σελίδα επιλογής των διαλέξεων της εφαρμογής? Υπάρχει κάποια αλλαγή που θα κάνατε?

3 απαντήσεις

Θα ήθελα πιο έντονο το περίγραμμα στον πίνακα και το μάθημα να γίνεται με click in button

Οχι

Κατατοπιστικό αλλά έχει πολλές στήλες

Εικόνα 3.25 - Δεύτερη απάντηση ερωτηματολογίου

Πως σας φαίνεται η σελίδα εισαγωγής θέσης φοιτητή της εφαρμογής (ανάλογα με την διάλεξη που έχουμε επιλέξει)? Υπάρχει κάποια αλλαγή που θα κάνατε?

3 απαντήσεις

Θα ήθελα να ήταν προκαθρισμένες οι επιλογές θέσης εξηγώντας σειρά και αριθμός καθισματος

Ακριβής και με όσες λεπτομέρειες χρειάζεται

Θα προτιμούσα να είναι θαμπό το UI του προηγούμενου ερωτήματος ή να έχει το καινούριο UI κάποιο περίγραμμα

Εικόνα 3.26 - Τρίτη απάντηση ερωτηματολογίου

Πως σας φαίνεται η σελίδα προβολής στοιχείων της εφαρμογής? Υπάρχει κάποια αλλαγή που θα κάνατε?

3 απαντήσεις

Απλή και κατανοητή

Όχι

Κατατοπιστικό αλλά έχει πολλές στήλες

Εικόνα 3.27 - Τέταρτη απάντηση ερωτηματολογίου

Πως σας φαίνεται η σελίδα δήλωσης κρούσματος με βάση την ημερομηνία διάγνωσης? Υπάρχει κάποια αλλαγή που θα κάνατε?

3 απαντήσεις

Μια χαρα

Έχει όσα χρειάζεται

Κατατοπιστικό

Εικόνα 3.28 - Πέμπτη απάντηση ερωτηματολογίου

Οι αλλαγές που έγιναν στην εφαρμογή μετά τις απαντήσεις του ερωτηματολογίου ήταν οι εξής:

1. Στην login σελίδα το sign in button από μαύρο έγινε μπλε για να είναι πιο φιλική στο χρήστη η εφαρμογή.
2. Κατάργηση της στήλης ορόφου αίθουσας από τα μενού της εφαρμογής καθώς οι φοιτητές γνωρίζουν τον όροφο της κάθε αίθουσας και θεωρείται περιττή πληροφορία.

ΚΕΦ.4: Επίλογος

Στην συγκεκριμένη πτυχιακή αναλύω, υλοποιώ και βγάζω στην παραγωγή μια διαδικτυακή εφαρμογή που έχει να κάνει με την ιχνηλάτηση κρουσμάτων. Εχω χρησιμοποιήσει διάφορα εργαλεία για την ανάπτυξη λογισμικού που ανέλυσα και έκανα deploy στο cloud.

Η συγκεκριμένη πτυχιακή με βοήθησε στο να σχεδιάζω και να υλοποιώ πολύπλοκα συστήματα καθώς ήταν πρώτη φορά που υλοποίησα μια τόσο

μεγάλη εφαρμογή. Επιπλέον απέκτησα αρκετές δεξιότητες σχετικά με διάφορα εργαλεία πληροφορικής, αλλά και το θεωρητικό μου υπόβαθρο σχετικά με τις βάσεις δεδομένων, την αρχιτεκτονική του λογισμικού και την μεθοδολογία υλοποίησης συστημάτων βελτιώθηκε σε πολύ μεγάλο βαθμό.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Mozilla MVC Controller:
<https://developer.mozilla.org/en-US/docs/Glossary/MVC>

[2] UML Language:
<https://www.uml.org/>

[3] Python Docs:
<https://docs.python.org/3/>

[4] Django:
<https://docs.djangoproject.com/en/4.0/>

[5] SQLite:
<https://www.sqlite.org/docs.html>

[6] PostgreSQL:

<https://www.postgresql.org/docs/>

[7] Heroku Cloud:

<https://devcenter.heroku.com/categories/reference>

[8] Salesforce Cloud:

<https://developer.salesforce.com/docs>

[9] Whitenoise:

<http://whitenoise.evans.io/en/stable/>

[10] Gunicorn:

<https://gunicorn.org/>