

# ΕΦΑΡΜΟΓΗ ΥΠΗΡΕΣΙΩΝ ΦΟΙΤΗΤΗ

PROJECT ΣΤΟ ΜΑΘΗΜΑ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Ομάδα: 22

Κωνσταντίνος Σαμαράς

ΑΜ: 1053598, Email: [up1053598@upnet.gr](mailto:up1053598@upnet.gr)

Ευστράτιος Ζαραδούκας

ΑΜ: 1053630, Email: [up1053630@upnet.gr](mailto:up1053630@upnet.gr)

## 1. ΠΕΡΙΓΡΑΦΗ

Με την τεχνολογική ανάπτυξη και τη χρήση του διαδικτύου να είναι πλέον απαραίτητη σε σχεδόν οτιδήποτε κάνουμε στη ζωή μας, έτσι και τα Πανεπιστήμια πρέπει να ακολουθούν τον ίδιο ρυθμό και να στρέφονται όλο και περισσότερο στη βελτίωση των ηλεκτρονικών υπηρεσιών για την καλύτερη εξυπηρέτηση των φοιτητών και φοιτητριών τους. Στην εργασία που μας ανατέθηκε, έχουμε να δημιουργήσουμε μια εφαρμογή που θα αφορά σε υπηρεσίες φοιτητή ενός νεοσύστατου τμήματος του Πανεπιστημίου μας. Η εφαρμογή έχει σκοπό να διευκολύνει τον τρόπο με τον οποίο αλληλοεπιδρούν οι φοιτητές με την Γραμματεία της σχολής. Για παράδειγμα, η δυνατότητα δήλωσης μαθημάτων μέσω της εφαρμογής είναι πολύ πιο απλή σε σχέση με το να μην υπήρχε ένα σύστημα «Ηλεκτρονικής Γραμματείας».

### 1.1 ΔΥΝΑΤΟΤΗΤΕΣ

Η εφαρμογή που υλοποιήσαμε υποστηρίζει τις παρακάτω διεργασίες:

1. Κατάσταση φοιτητών ανά έτος εισαγωγής.
2. Εγγραφή φοιτητών σε διδασκαλίες/μαθήματα. Ο φοιτητής μπορεί να γραφτεί σε ένα εξάμηνο επιλέγοντας τα μαθήματα που διδάσκονται στο συγκεκριμένο εξάμηνο ή σε προηγούμενα εξάμηνα.
3. Καταχώρηση βαθμολογίας. Ο διαχειριστής-γραμματεία του τμήματος θα έχει πρόσβαση σε μια λίστα με τα ονόματα των φοιτητών που έχουν γραφτεί στο κάθε μάθημα και θα μπορεί να καταχωρεί την βαθμολογία που έχει πάρει ο φοιτητής.
4. Εμφάνιση καταστάσεων φοιτητών εγγεγραμμένων σε κάποιο μάθημα.
5. Εμφάνιση των στοιχείων των καθηγητών.
6. Εμφάνιση των μαθημάτων που προσφέρονται από το Τμήμα μαζί με επιπλέον πληροφορίες για αυτά όπως τα ECTS που έχουν. Ακόμα υπάρχει δυνατότητα εμφάνισης λίστας των μαθημάτων που είτε κορμού είτε τομέα ή έχουν εργαστήριο.
7. Διάφορες βεβαιώσεις φοίτησης για τους φοιτητές.
8. Δήλωση Κατεύθυνσης και εμφάνιση όλων όσων είναι διαθέσιμες στο Τμήμα.

9. Έλεγχος ποιοι φοιτητές πληρούν τα κριτήρια για πτυχίο και εμφάνιση αντίστοιχης λίστας.
10. Εμφάνιση όλων των διπλωματικών που πραγματοποιούνται στο Τμήμα.
11. Ο φοιτητής μπορεί να κάνει νέα αίτηση πιστοποιητικού και να δει το ιστορικό του.
12. Έλεγχος αν υπάρχουν αιτήσεις πιστοποιητικών που δεν έχουν ικανοποιηθεί ακόμα.
13. Προσθήκη νέων φοιτητών, καθηγητών, μαθημάτων, διδασκαλιών από τον διαχειριστή.

Στη βάση δεδομένων μας θέλουμε να αποθηκεύονται τα εξής χαρακτηριστικά:

1. Για κάθε φοιτητή θέλουμε να αποθηκεύουμε το ονοματεπώνυμο, το όνομα πατρός και μητρός, τον αριθμό μητρώου ο οποίος είναι μοναδικός, τη διεύθυνση, το τηλέφωνο, το e-mail του, το έτος εισαγωγής, αν είναι προπτυχιακός ή μεταπτυχιακός, αν είναι ενεργός ή όχι καθώς και η κατεύθυνση που ανήκει εφόσον έχει επιλέξει.
2. Για τα μαθήματα που προσφέρονται από το συγκεκριμένο τμήμα θα αποθηκεύεται η ονομασία κάθε μαθήματος, ο κωδικός που έχει, αν είναι κορμού ή κατεύθυνσης, αν έχει εργαστήριο καθώς επίσης και σε ποιο εξάμηνο αντιστοιχεί στο πρόγραμμα σπουδών και πόσες διδακτικές μονάδες (ECTS) έχει.
3. Για τους καθηγητές του τμήματος το ονοματεπώνυμο, τη διεύθυνση του γραφείου τους, το τηλέφωνο τους, το email και τέλος την βαθμίδα που κατέχουν.
4. Τα μαθήματα που διδάσκει ο κάθε καθηγητής.
5. Τα μαθήματα στα οποία είναι εγγεγραμμένος ο κάθε φοιτητής καθώς επίσης και τον βαθμό που έχει πάρει στα μαθήματα στα προηγούμενα εξάμηνα.
6. Τα πιστοποιητικά και τις αιτήσεις των φοιτητών για αυτά.
7. Οι Κατευθύνσεις με την ονομασία, καθώς και οι φοιτητές από τους οποίους αποτελούνται αυτές.
8. Οι διπλωματικές με την ονομασία, το ΑΜ του φοιτητή και του καθηγητή, ημερομηνία ανάθεσης και ελάχιστης ημερομηνίας ενασχόλησης με αυτήν.

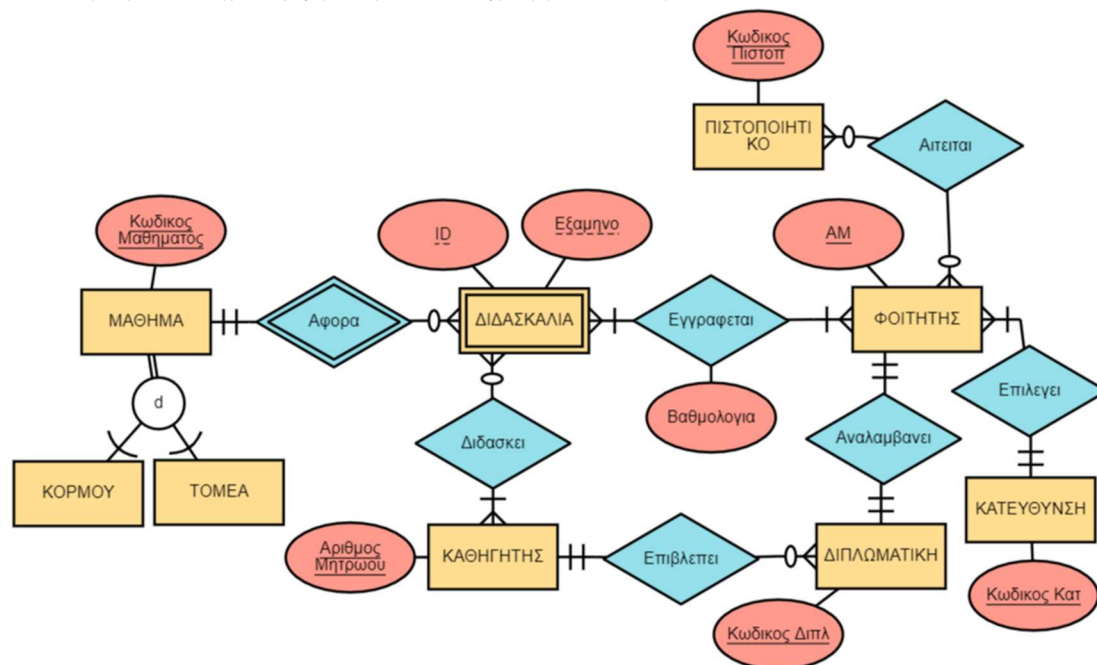
Οι χρήστες της συγκεκριμένης εφαρμογής είναι οι φοιτητές και ο διαχειριστής, που στη συγκεκριμένη περίπτωση θεωρούμε πως είναι το προσωπικό της γραμματείας του τμήματος. Το προσωπικό της γραμματείας θα έχει έναν συγκεκριμένο κωδικό με τον οποίο θα μπορεί να συνδέεται στο σύστημα δηλαδή θα έχει αυξημένες αρμοδιότητες και δυνατότητες συγκριτικά με τους φοιτητές.

## 2. ΜΕΘΟΔΟΛΟΓΙΑ

Για να μπορέσουμε να αποθηκεύσουμε όλες τις πληροφορίες που χρειαζόμαστε για την σχεδίασή της εφαρμογής, χρειαζόμαστε την δημιουργία μιας βάσης δεδομένων. Η λογική δομή μιας βάσης δεδομένων είναι μια απαραίτητη διαδικασία που πρέπει να γίνει πριν από την καταχώρηση και την επεξεργασία των στοιχείων της βάσης δεδομένων. Για αυτό τον λόγο έπρεπε να ακολουθήσουμε μια σειρά από βήματα - στάδια με σκοπό την πλήρη κατανόηση του προβλήματος, την ανάλυση του σε θεωρητικό επίπεδο και έπειτα την υλοποίηση του με την δημιουργία μιας βάσης δεδομένων.

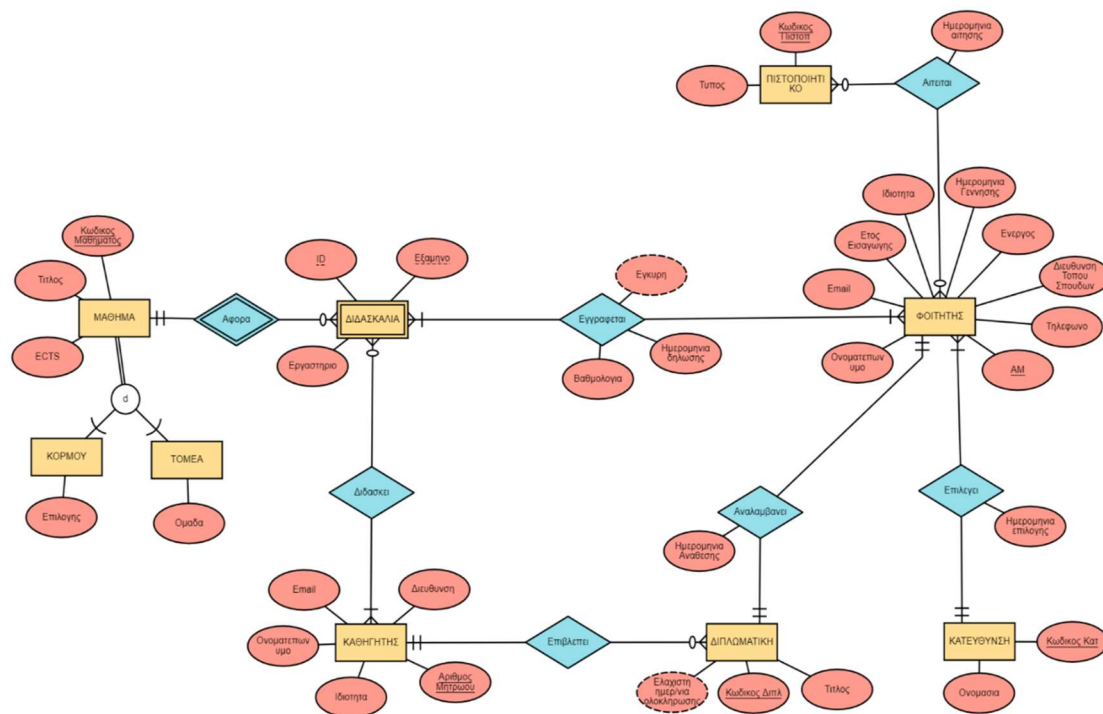
## 2.1 ΔΙΑΓΡΑΜΜΑ ΟΝΤΟΤΗΤΩΝ – ΣΥΣΧΕΤΙΣΕΩΝ (ERD)

Αρχικά για να ξεκινήσουμε το πρότζεκτ σκεφτήκαμε τι ακριβώς θέλουμε να υλοποιεί, τις απαιτήσεις σχετικά με την βάση δεδομένων και τη δημιουργία κάποιων παραδοχών που κάναμε που αφορούσαν τον μικρόκοσμο που θα αλληλοεπιδρούσε με την εφαρμογή. Έτσι καταλήξαμε να δημιουργήσουμε το διάγραμμα οντοτήτων – συσχετίσεων (ERD).



Σχήμα 1. Μερικό διάγραμμα οντοτήτων σχέσεων ERD.

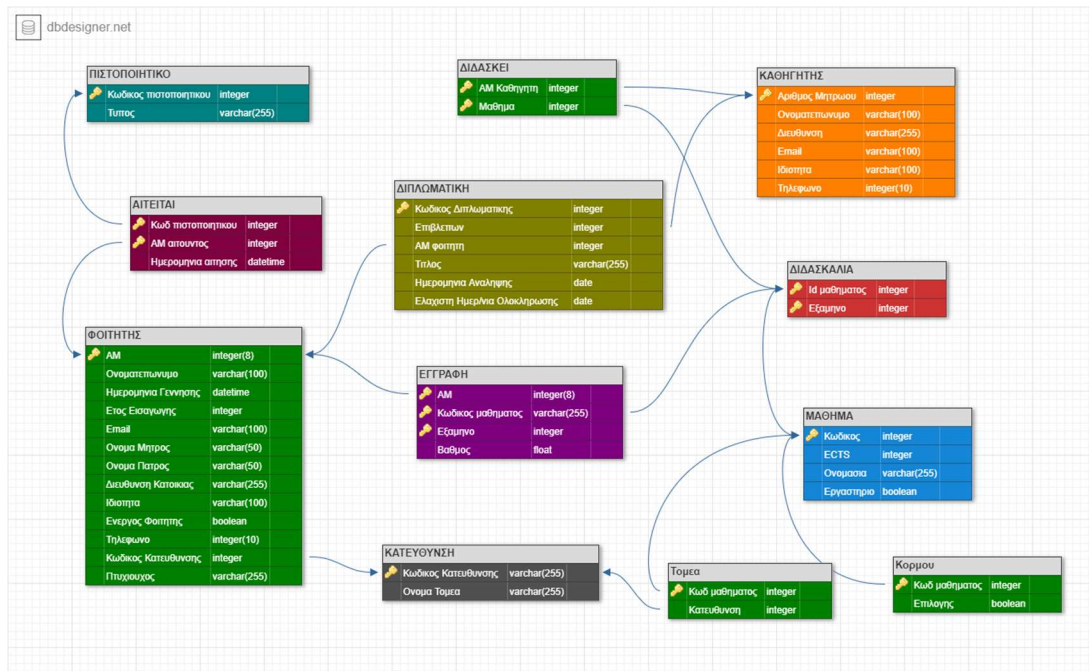
Το Σχήμα 1. απεικονίζει το διάγραμμα οντοτήτων σχέσεων. Αρχικά ο ΦΟΙΤΗΤΗΣ και το ΠΙΣΤΟΠΟΙΗΤΙΚΟ συνδέονται με την σχέση Αιτείται. Στον ΦΟΙΤΗΤΗ περιέχονται όλα τα γνωρίσματα που μας ενδιαφέρουν και το πρωτεύον του κλειδί είναι ο αριθμός μητρώου, AM. Στο ΠΙΣΤΟΠΟΙΗΤΙΚΟ το πρωτεύον κλειδί είναι ο κωδικός του. Η σχέση Επιλέγει συνδέει τον ΦΟΙΤΗΤΗ με την ΚΑΤΕΥΘΥΝΣΗ στην οποία το πρωτεύον κλειδί είναι ο μοναδικός κωδικός της. Η σχέση Εγγράφεται ενώνει την οντότητα ΦΟΙΤΗΤΗΣ με την ΔΙΔΑΣΚΑΛΙΑ η οποία είναι ασθενής οντότητα και έχει σύνθετο κλειδί το ID και το Εξάμηνο. Το ID είναι και ξένο κλειδί που δείχνει στο πρωτεύον κλειδί της οντότητας ΜΑΘΗΜΑ. Το ΜΑΘΗΜΑ έχει ως πρωτεύον κλειδί τον κωδικό μαθήματος και συσχετίζεται με τις οντότητες ΚΟΡΜΟΥ και ΤΟΜΕΑ με ένα περιορισμό μη επικάλυψης (disjoint constraint) με πλήρη συμμετοχή, το οποίο σημαίνει ότι ένα μάθημα μπορεί να κατατάσσεται στην κατηγορία μαθήματα κορμού είτε να κατατάσσεται στην κατηγορία μαθήματα τομέα. Επιστρέφοντας στην οντότητα ΔΙΔΑΣΚΑΛΙΑ, η οποία συνδέεται με την οντότητα ΚΑΘΗΓΗΤΗΣ με την σχέση Διδάσκει με τον ΚΑΘΗΓΗΤΗ να έχει για πρωτεύον κλειδί τον αριθμό μητρώου του. Τέλος, η ΔΙΠΛΩΜΑΤΙΚΗ συνδέεται με τον ΚΑΘΗΓΗΤΗ με την σχέση Επιβλέπει και με τον ΦΟΙΤΗΤΗ με την σχέση Αναλαμβάνει και έχει ως πρωτεύον κλειδί τον κωδικό της διπλωματικής που αναλαμβάνει ο φοιτητής, σε παρακάτω στάδια αυτές οι δυο σχέσεις αντικαθίστανται με γνωρίσματα της οντότητας ΔΙΠΛΩΜΑΤΙΚΗ που είναι ξένα κλειδιά, τα οποία δείχνουν στον αριθμό μητρώου του καθηγητή και τον αριθμό μητρώου του φοιτητή αντίστοιχα.



Σχήμα 2. Πλήρες διάγραμμα οντοτήτων σχέσεων ERD.

## 2.2 ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ

Στη συνέχεια, αφού καταλήξαμε σε ένα ERD, το οποίο μας κάλυπτε εννοιολογικά, έπρεπε να προχωρήσουμε στο επόμενο στάδιο της ανάπτυξης της βάσεως δεδομένων μας, το σχεσιακό μοντέλο (Relational Model). Έτσι, ακολουθώντας τους κανόνες μετατροπής του εννοιολογικού σχήματος στο σχεσιακό μοντέλο και χρησιμοποιώντας την ιστοσελίδα [dbdesigner.net](http://dbdesigner.net) καταλήξαμε στο παρακάτω σχήμα. Άξιο αναφοράς είναι το γεγονός ότι η σχέση ΔΙΠΛΩΜΑΤΙΚΗ διαθέτει δυο πεδία τα ΑΜ φοιτητή και Επιβλέπων, τα οποία είναι ξένα κλειδιά και παίρνουν τιμές από τα πρωτεύοντα κλειδιά των σχέσεων ΦΟΙΤΗΤΗΣ και ΚΑΘΗΓΗΤΗΣ αντίστοιχα. Η παραπάνω αντιμετώπιση μας επιτρέπει να μειώσουμε τον αριθμό των σχέσεων στο μοντέλο μας, αφού παραλείπουμε τις συσχετίσεις Επιβλέπει και Αναλαμβάνει από το εννοιολογικό σχήμα μας.



Σχήμα3: Σχεσιακό Μοντέλο της εφαρμογής υποστήριξης φοιτητών

## 2.3 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ SQL

Στην επόμενη φάση του πρότζεκτ χρησιμοποιήσαμε πάλι το dbdesigner.net και το σχεσιακό μοντέλο που αναπτύξαμε προηγουμένως για να δημιουργήσουμε τον κώδικα της βάσης δεδομένων μας. Αυτό το επιτύχουμε χρησιμοποιώντας την λειτουργία Export του dbdesigner.net, στην οποία επιλέξαμε την MySQL και μας έδωσε τον βασικό κορμό της βάσης μας. Προφανώς, χρειάστηκε να αρκετές αλλαγές στον κώδικα χρησιμοποιώντας εντολές από την data definition language της SQL, όπως DROP, ADD\_CONSTRAINT, CREATE TABLE.

```
CREATE TABLE `DIPLOMATIKI` (
  `dipl id` INT PRIMARY KEY,
  `AM student` INT NOT NULL UNIQUE,
  `epiblepon` INT NOT NULL,
  `title` VARCHAR(50),
  `assignment date` DATE NOT NULL,
  `min completion date` DATE NOT NULL,
  FOREIGN KEY (`epiblepon`) REFERENCES `PROFESSOR` (`AM`) ON UPDATE CASCADE ON DELETE CASCADE,
  FOREIGN KEY (`AM student`) REFERENCES `STUDENT` (`AM`) ON UPDATE CASCADE ON DELETE CASCADE
) ENGINE = InnoDB DEFAULT CHARSET = utf8;
```

Όμως για να μπορέσουμε να ξεκινήσουμε την ανάπτυξη και τον πειραματισμό με την βάση μας σε προγραμματιστικό επίπεδο έπρεπε να προσθέσουμε κάποια αρχικά δεδομένα για να μπορούμε να βλέπουμε τις αλλαγές που κάνουμε και τι αποτελέσματα έχουν αυτές στα δεδομένα μας.



```

INSERT INTO
`LESSON`
VALUES
(1, 5, 'Βάσεις Δεδομένων'),(2, 5, 'Λειτουργικά Συστήματα'),
(3, 5, 'Αρχές Προγραμματισμού'),(4, 5, 'Λογιστικά Φύλλα'),(5, 3, 'Ψηφιακή Λογική'),
(6, 3, 'Φυσική'),(7, 5, 'Μαθηματικά'),(8, 5, 'ΣΗΕ'),(9, 5, 'ΣΑΕ'),
(10, 5, 'Μηχανές'),(11, 6, 'ΗΜ ΠΕΔΙΑ 1'),(12, 8, 'ΗΜ ΠΕΔΙΑ 2'),
(13, 8, 'Ηλεκτρικά Κυκλώματα'),(14, 5, 'Ασφάλεια Υπολογιστών'),(15, 5, 'Ασυρματη Διαδοση'),
(16, 5, 'Επικοινωνίες'),(17, 5, 'Αρχιτεκτονική Υπολογιστών'),(18, 3, 'Word Pro'),
(19, 5, 'SQL Server'),(20, 5, 'Visual C'),(21, 5, 'Μαθηματικά 2'),
(22, 2, 'SQL Windows'),(23, 3, 'Ψηφιακά Κυκλώματα'),(24, 2, 'Power Point');

```

Σημειωτέων πως για την ανάπτυξη της βάσης μας σε αυτό το στάδιο και στο επόμενο χρησιμοποιήσαμε το MySQLWorkbench αλλά και την ιστοσελίδα phpMyAdmin, την οποία μας παρείχε το μάθημα για να αποθέσουμε τη βάση μας στο διαδίκτυο.

## 2.4 ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΔΙΕΠΑΦΗ (PYTHON)

Το τελευταίο και πιο σημαντικό κομμάτι για την δημιουργία της βάσης μας και για να γίνει χρήσιμη για τους χρήστες που στοχεύουμε είναι η διασύνδεση της με κάποια γλώσσα προγραμματισμού και την ανάπτυξη μιας απλής διεσπάρης για να μπορεί ο χρήστης να διαχειριστεί και να αλληλοεπιδράσει με την βάση μας. Επιλέξαμε να αναπτύξουμε την εφαρμογή μας σε Python λόγω της πολύ εύκολης διασύνδεσης με την MySQL και την δυνατότητα εύκολης υλοποίησης μια απλής διεσπάρης για να χειρίζεται ο χρήστης την βάση μας. Το βασικό module που χρησιμοποιήσαμε ονομάζεται mysql-connector για να δημιουργήσουμε την σύνδεση του προγράμματος μας με την βάση.

```

import mysql.connector
import random
from tkinter import *

# Σύνδεση με mysql

def connect():
    try:
        conn = mysql.connector.connect(
            host="150.140.186.221", user="db20_up1053630", passwd="up1053630", database="project_db20_up1053630")
    except mysql.connector.Error as e:
        print(e)
        return False
    return conn

```

Έπειτα, χρησιμοποιώντας το module tkinter της Python φτιάξαμε ένα μικρό παράθυρο με κουμπιά, τα οποία ο χρήστης μπορεί να τα πατήσει για να εκτελέσει διάφορες ερωτήσεις προς την βάση δεδομένων.

```

print("User mode")
window = Tk()
window.title("Υπηρεσίες Φοιτητή: Λειτουργία Φοιτητή")
w = 1000
h = 400
window.geometry('%dx%d+%d+%d' % (w, h, 0, 0))

# READ
read = Label(window, text="ΠΛΗΡΟΦΟΡΙΕΣ").grid(column=0, row=0)
# Εμφανίστε τα στοιχεία όλων των καθηγητων.
profBtn = Button(window, text="Εμφάνιση πληροφοριων των καθηγητών",
                  command=lambda: professor(conn, mycurs)).grid(column=0, row=1)
# Εμφανίστε τα μαθηματα στη βάση δεδομένων κατά αλφαβητική σειρά.
lessonsBtn = Button(window, text="Εμφάνιση όλων των μαθημάτων", command=lambda: lesson(
    conn, mycurs)).grid(column=0, row=2)
# Εμφανίστε τα μαθηματα που αντιστοιχούν σε συγκεκριμένο φοιτητή

```

Κάθε κουμπί είναι εκτελεί μια συνάρτηση στην οποία έχουμε προσθέσει το κατάλληλο ερώτημα σε γλώσσα SQL(Data Manipulation Language) εμφυτευμένη μέσα σε κώδικα Python:

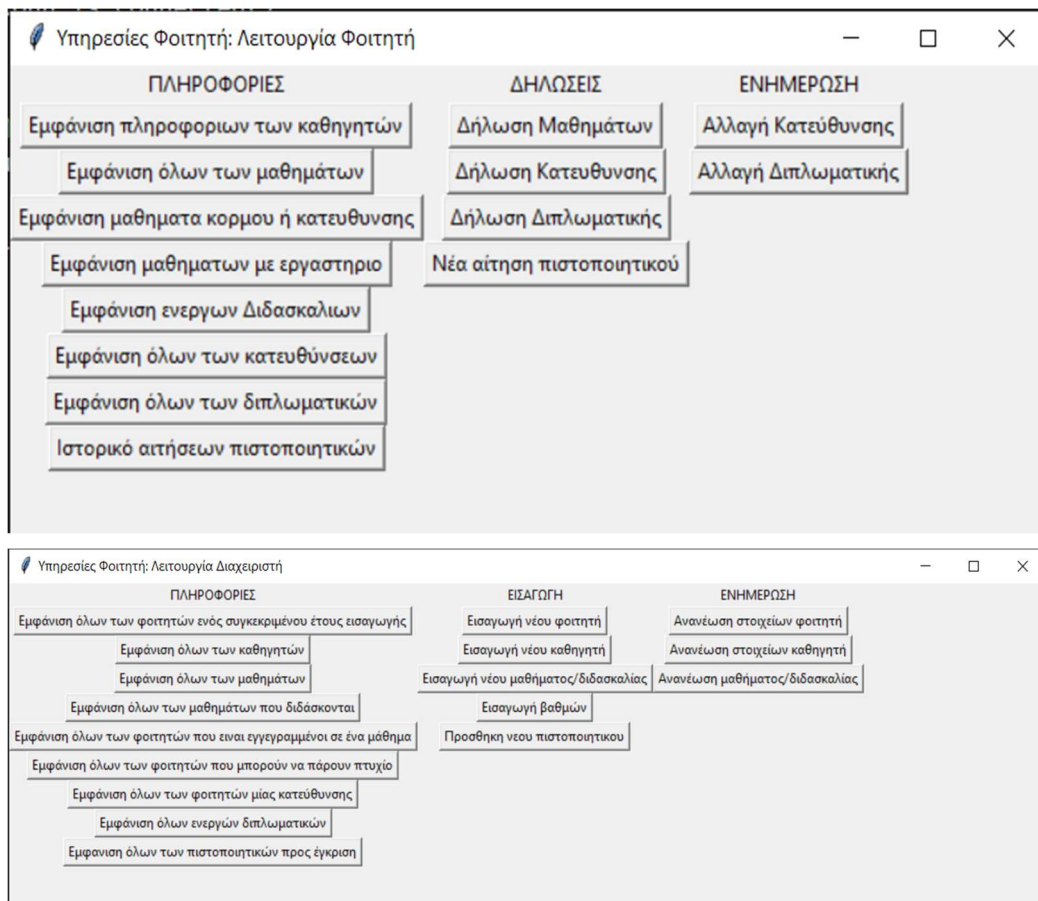
```
# Εμφανίστε τα στοιχεία όλων των καθηγητων.
def professor(conn, mycurs):
    try:
        mycurs.execute("SELECT * FROM `PROFESSOR`")
        print("ΚΑΘΗΓΗΤΕΣ: \n")
        for x in mycurs:
            print(x)

    except mysql.connector.Error as e:
        print(e)
```

Τα αποτελέσματα που επιστρέφονται εμφανίζονται στο terminal:

```
ΚΑΘΗΓΗΤΕΣ:
(1, 'Giorgos Papadopoulos', None, None, 'Επίκουρος Καθηγητής', 666443217)
(2, 'Giannis Papadopoulos', None, None, 'Καθηγητής', 123443217)
(3, 'Ioannis Pavlou', None, None, 'Καθηγητής', 669993217)
(4, 'ΑΡΙΣΤΟΤΕΛΗΣ ΣΚΟΠΕΛΙΤΗΣ', None, None, 'Αναπληρωτής Καθηγητής', None)
(5, 'ΕΛΕΥΘΕΡΙΟΣ ΠΑΠΑΧΡΗΣΤΟΣ', None, None, 'Επίκουρος Καθηγητής', None)
(6, 'ΑΘΑΝΑΣΙΟΣ ΜΠΑΤΣΙΩΤΗΣ', None, None, 'Καθηγητής', None)
```

Επίσης, σκεφτήκαμε πως οι άνθρωποι που μπορεί να χρησιμοποιήσουν την εφαρμογή μας μπορούν να κατηγοριοποιηθούν είτε ως χρήστες (Φοιτητές) είτε ως διαχειριστές (Γραμματεία, Τεχνικό Προσωπικό). Έτσι, φτιάξαμε για τον καθένα διαφορετική διεπαφή και την εμπλουτίσαμε με διάφορες δυνατότητες για τις οποίες δίνονται παραδείγματα παρακάτω, διότι για άλλες λειτουργίες ενδιαφέρεται ο φοιτητής και για άλλες η γραμματεία. Τέλος, αυτός ο διαχωρισμός είναι απαραίτητος για λόγους ασφάλειας, διότι δεν γίνεται ένας φοιτητής να έχει πρόσβαση σε προσωπικές πληροφορίες άλλων φοιτητών.



### 3. ΑΞΙΟΛΟΓΗΣΗ

Μιας και η εφαρμογή αφορά υπηρεσίες για ένα φοιτητή είναι κατανοητό ότι πρέπει να σχετίζεται και με το Πανεπιστήμιο. Έτσι καταλήξαμε στη δημιουργία μιας βάσης δεδομένων στην οποία θα αποθηκεύονται όλες οι προσωπικές πληροφορίες του φοιτητή, οι διδασκαλίες στις οποίες έχει εγγραφεί καθώς και οι βαθμολογίες του σε αυτές. Ακόμα ικανοποιήσαμε την ανάγκη του να κάνει αιτήσεις για πιστοποιητικά ευκολά και να βλέπει το ιστορικό του ενώ επιπλέον μπορεί να δηλώσει την Κατεύθυνση του. Μετά ασχοληθήκαμε με τα μαθήματα που προσφέρονται και θεωρήσαμε ότι ένας φοιτητής χρειάζεται να γνωρίζει εάν υπάρχει εργαστηριακό σκέλος ή αν είναι μάθημα κορμού ή ανήκουν σε κάποια κατεύθυνση, καθώς και ποιος καθηγητής το διδάσκει. Έτσι προέκυψε στη βάση να βρίσκονται πληροφορίες για τους καθηγητές, οι οποίες μπορούν να βρεθούν από τους φοιτητές σε περίπτωση που θελήσουν να επικοινωνήσουν με τους καθηγητές τους, ενώ και όταν ο φοιτητής αναλαμβάνει μια διπλωματική εμφανίζεται και ο καθηγητής που την επιβλέπει. Συνοψίζοντας κριτήρια επιτυχίας για εμάς είναι ένα σωστό εννοιολογικό μοντέλο του μικρόκοσμου, αυτοματοποίηση της εισαγωγής, επεξεργασίας και διαγραφής δεδομένων μέσω αλγορίθμων που αναπτύξαμε στην ργthon και τέλος την εμφάνιση των σωστών δεδομένων στην έξοδο, για κάθε εντολή που δίνει ο χρήστης.



#### 4. ΔΕΔΟΜΕΝΑ

Για να καταλήξουμε σε αυτό το μοντέλο αξιολόγησης χρησιμοποιήσαμε κυρίως δεδομένα από το Τμήμα ΗΜΤΥ στο οποίο φοιτούμε και στη δική του ηλεκτρονική γραμματεία. Μαθήματα της βάσης μας μπορεί να είναι όμοια με αυτά του προγράμματος σπουδών του Τμήματος, ενώ για ονόματα φοιτητών και καθηγητών χρησιμοποιήσαμε μερικά που βρήκαμε σε εργαστηριακές ασκήσεις του μαθήματος Βάσεις Δεδομένων ενώ άλλα είναι φανταστικά. Η ενσωμάτωση των δεδομένων στη βάση έγινε για μερικά μέσω των εντολών της MySQL INSERT INTO ενώ για άλλα μέσω python με εντολές for προσέχοντας πάντα μην εισαχθεί κάτι λάθος στη βάση μας. Όπως φαίνεται στην παρακάτω εικόνα, γεμίζουμε τον πίνακα ΔΙΔΑΣΚΕΙ, αναθέτοντας τυχαία έναν καθηγητή για κάθε μάθημα που υπάρχει αποθηκευμένο στη βάση μας.

```
def all_lessons(conn, mycurs): # Επιστρέφει το συνολικό αριθμο μαθηματων στη βαση
    try:
        mycurs.execute('SELECT COUNT(*) FROM `LESSON`;')
        rows = mycurs.fetchall()
        number = rows[0]
        return int(number[0])

    except mysql.connector.Error as e:
        print(e)

def didaskei(conn, mycurs): # Γεμίζουμε τον πίνακα ΔΙΔΑΣΚΕΙ
    num_of_lessons = all_lessons(conn, mycurs)
    mycurs.execute('SELECT `am` FROM `PROFESSOR`;')
    kath = []
    for x in mycurs:
        kath.append(x[0])
    len1 = num_of_lessons + 1
    for i in range(1, len1):
        random_didaskei = kath[random.randint(0, len(kath)-1)]
        try:
            mycurs.execute(
                'INSERT INTO `DIDASKEI` VALUES ({},{})'.format(random_didaskei, i))
            conn.commit()

        except mysql.connector.Error as e:
            print(e)

    didaskei(conn, mycurs)
```

Εικόνα : Γέμισμα πίνακα ΔΙΔΑΣΚΕΙ.

#### 5. ΚΑΤΑΜΕΡΙΣΜΟΣ ΕΡΓΑΣΙΩΝ

Για τον καταμερισμό εργασιών υπήρχε από την αρχή συνεννόηση να γίνουν όλες οι εργασίες από κοινού. Είχαμε συζητήσει τι θέλουμε να υλοποιήσουμε στο σύνολο και βήμα βήμα ολοκληρώνοντας ο καθένας ένα οποιοδήποτε κομμάτι το έστελνε στον άλλον προς αξιολόγηση. Έτσι δεν ανέλαβε κανείς κάτι εξ ολοκλήρου μόνος του.

## 6. ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ

Ξεκινήσαμε με το ERD περίπου την επόμενη εβδομάδα από τη στιγμή που μας ανατέθηκε το project ώστε να είναι έτοιμο για την παρουσίαση. Μετά από τη παρουσίαση και τη καθοδήγηση των διδασκόντων κάναμε μερικές αλλαγές και το φέραμε στη τελική του μορφή μία εβδομάδα περίπου πριν τις διακοπές των Χριστουγέννων. Με το υπόλοιπο project ασχοληθήκαμε από τις 27-28 Δεκέμβρη και το ολοκληρώσαμε λίγες μέρες πριν την τελική ημερομηνία υποβολής της εργασίας. Χοντρικά, ο συνολικός αριθμός ημερών που ασχοληθήκαμε με το συγκεκριμένο project είναι 15-18 μέρες.

## 7. ΠΑΡΑΡΤΗΜΑ

Παρακάτω υπάρχουν οι σύνδεσμοι με τα προγράμματα και εργαλεία που χρησιμοποιήσαμε:

Python: <https://www.python.org/>

ERDMaker: <https://www.erdmaker.com/>

DBDesigner: <https://www.dbdesigner.net/>

MySQL: <https://www.mysql.com/>

Phpmyadmin: <http://150.140.186.221:8081/phpmyadmin/index.php>

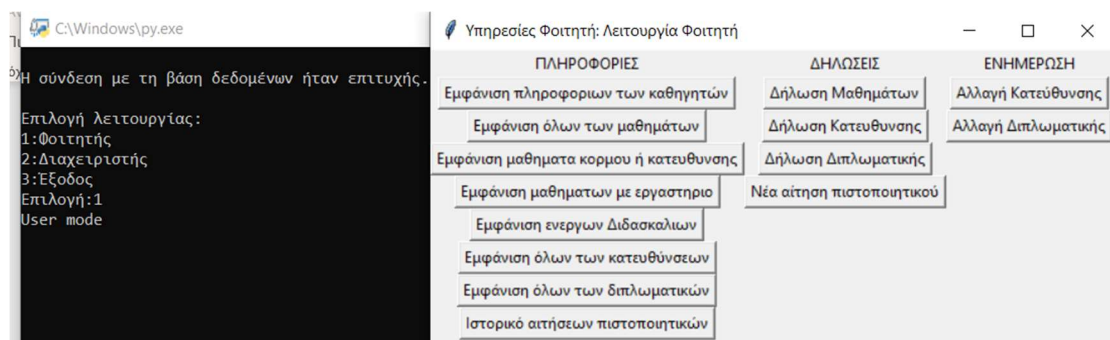
Username phpmyadmin: db20\_up1053630

Password phpmyadmin: up1053630

Password για admin στην εφαρμογή: 123

Παραδείγματα χρήσης της εφαρμογής:

Αρχικά, εισέρχονται στην εφαρμογή έχουμε να επιλέξουμε αν είμαστε φοιτητής, διαχειριστής ή να βγούμε από την εφαρμογή. Στο περιβάλλον του φοιτητή έχουμε τις εξής επιλογές σε άλλο αναδυόμενο παράθυρο, όπως φαίνονται στην παρακάτω εικόνα.



Έστω ότι διαλέγουμε την επιλογή Εμφάνιση ενεργών διδασκαλιών θα λάβουμε ως απάντηση το:

```
C:\Windows\py.exe
1:Φοιτητής
2:Διαχειριστής
3:Έξοδος
Επιλογή:1
User mode
ΛΙΣΤΑΣΚΑΛΙΑ:

title: Λογιστικά Φύλλα, semester: 1, ECTS: 5, professor: Giorgos Papadopoulos
title: Μαθηματικά, semester: 1, ECTS: 5, professor: Giannis Papadopoulos
title: Power Point, semester: 1, ECTS: 2, professor: ΕΛΕΥΘΕΡΙΟΣ ΠΑΠΑΧΡΗΣΤΟΣ
title: Αρχές Προγραμματισμού, semester: 2, ECTS: 5, professor: Giorgos Papadopoulos
title: Φυσική, semester: 2, ECTS: 3, professor: Giannis Papadopoulos
title: Ψηφιακή Λογική, semester: 3, ECTS: 3, professor: Giannis Papadopoulos
title: Ηλεκτρικά Κυκλώματα, semester: 3, ECTS: 8, professor: Ioannis Pavlou
title: Μαθηματικά 2, semester: 3, ECTS: 5, professor: ΑΘΑΝΑΣΙΟΣ ΜΠΛΑΤΣΙΩΤΗΣ
title: Ψηφιακά Κυκλώματα, semester: 4, ECTS: 3, professor: ΑΘΑΝΑΣΙΟΣ ΜΠΛΑΤΣΙΩΤΗΣ
title: ΣΗΕ, semester: 5, ECTS: 5, professor: Giannis Papadopoulos
title: ΗΜ ΠΕΔΙΑ 1, semester: 5, ECTS: 6, professor: Ioannis Pavlou
title: ΗΜ ΠΕΔΙΑ 2, semester: 6, ECTS: 8, professor: Ioannis Pavlou
title: Λειτουργικά Συστήματα, semester: 7, ECTS: 5, professor: Giorgos Papadopoulos
title: Visual C, semester: 7, ECTS: 5, professor: ΑΘΑΝΑΣΙΟΣ ΜΠΛΑΤΣΙΩΤΗΣ
title: Ασυρματη Διαδοση, semester: 7, ECTS: 5, professor: ΑΡΙΣΤΟΤΕΛΗΣ ΣΚΟΠΕΛΙΤΗΣ
title: Βάσεις Δεδομένων, semester: 7, ECTS: 5, professor: Giorgos Papadopoulos
title: Αρχιτεκτονική Υπολογιστών, semester: 8, ECTS: 5, professor: ΑΡΙΣΤΟΤΕΛΗΣ ΣΚΟΠΕΛΙΤΗΣ
title: SQL Server, semester: 8, ECTS: 5, professor: ΕΛΕΥΘΕΡΙΟΣ ΠΑΠΑΧΡΗΣΤΟΣ
title: Επικοινωνίες, semester: 8, ECTS: 5, professor: ΑΡΙΣΤΟΤΕΛΗΣ ΣΚΟΠΕΛΙΤΗΣ
title: Word Pro, semester: 9, ECTS: 3, professor: ΕΛΕΥΘΕΡΙΟΣ ΠΑΠΑΧΡΗΣΤΟΣ
title: Ασφάλεια Υπολογιστών, semester: 9, ECTS: 5, professor: Ioannis Pavlou
title: SQL Windows, semester: 9, ECTS: 2, professor: ΑΘΑΝΑΣΙΟΣ ΜΠΛΑΤΣΙΩΤΗΣ
```

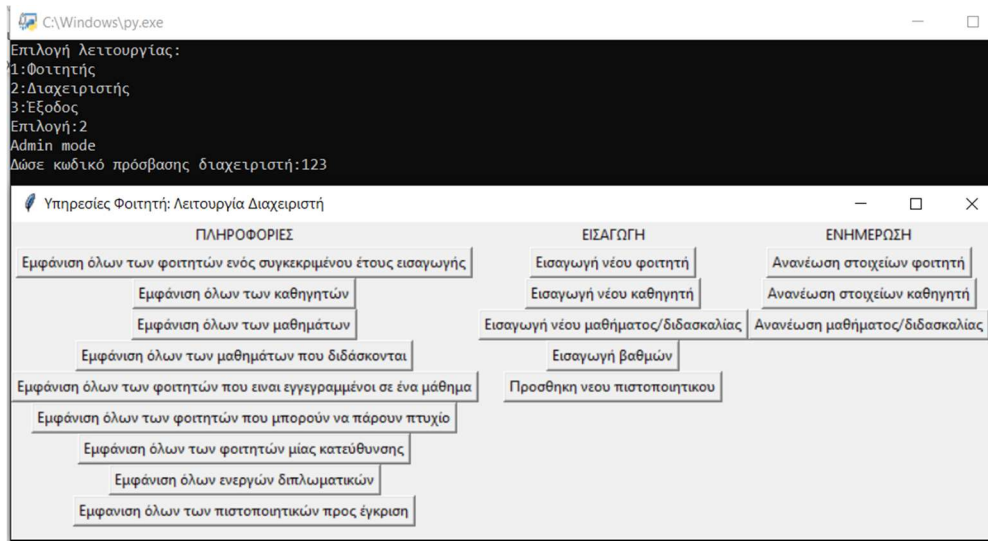
Αν επιλέξουμε την Εμφάνιση όλων των Κατευθύνσεων παίρνουμε ως απάντηση

```
C:\Windows\py.exe
Η σύνδεση με τη βάση δεδομένων ήταν επιτυχής.

Επιλογή λειτουργίας:
1:Φοιτητής
2:Διαχειριστής
3:Έξοδος
Επιλογή:1
User mode

Υπαρχουν οι εξης κατευθυνσεις:
ID: 1, ΟΝΟΜΑ: 'ΥΠΟΛΟΓΙΣΤΕΣ'
ID: 2, ΟΝΟΜΑ: 'ΕΠΙΚΟΙΝΩΝΙΕΣ'
ID: 3, ΟΝΟΜΑ: 'ΤΕΧΝΟΛΟΓΙΑ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ'
ID: 4, ΟΝΟΜΑ: 'ΗΛΕΚΤΡΟΝΙΚΗ'
ID: 5, ΟΝΟΜΑ: 'ΣΑΕ'
ID: 6, ΟΝΟΜΑ: 'ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ'
```

Στην περίπτωση που είμαστε ο διαχειριστής μόλις πατήσουμε αυτή την επιλογή θα μας ζητήσει να πληκτρολογήσουμε τον κωδικό του διαχειριστή και μόλις το κάνουμε θα εμφανιστεί το παράθυρο που μπορούμε να κάνουμε τις λειτουργίες μας.



Αν χρησιμοποιήσουμε το κουμπί Εισαγωγή νέου φοιτητή έχουμε:

