

Εφαρμογή αναζήτησης ακινήτου (π.χ., κατοικία, επαγγελματικός χώρος, γη) για αγορά/ενοικίαση

ΜΑΘΗΜΑ: ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ (ECE_ΓΚ703)

Ομάδα 43

Ντούνης Κωνσταντίνος

A.M.: 1066525

Στέφας Φίλιππος

A.M. 1066567

1 ΠΕΡΙΛΗΨΗ

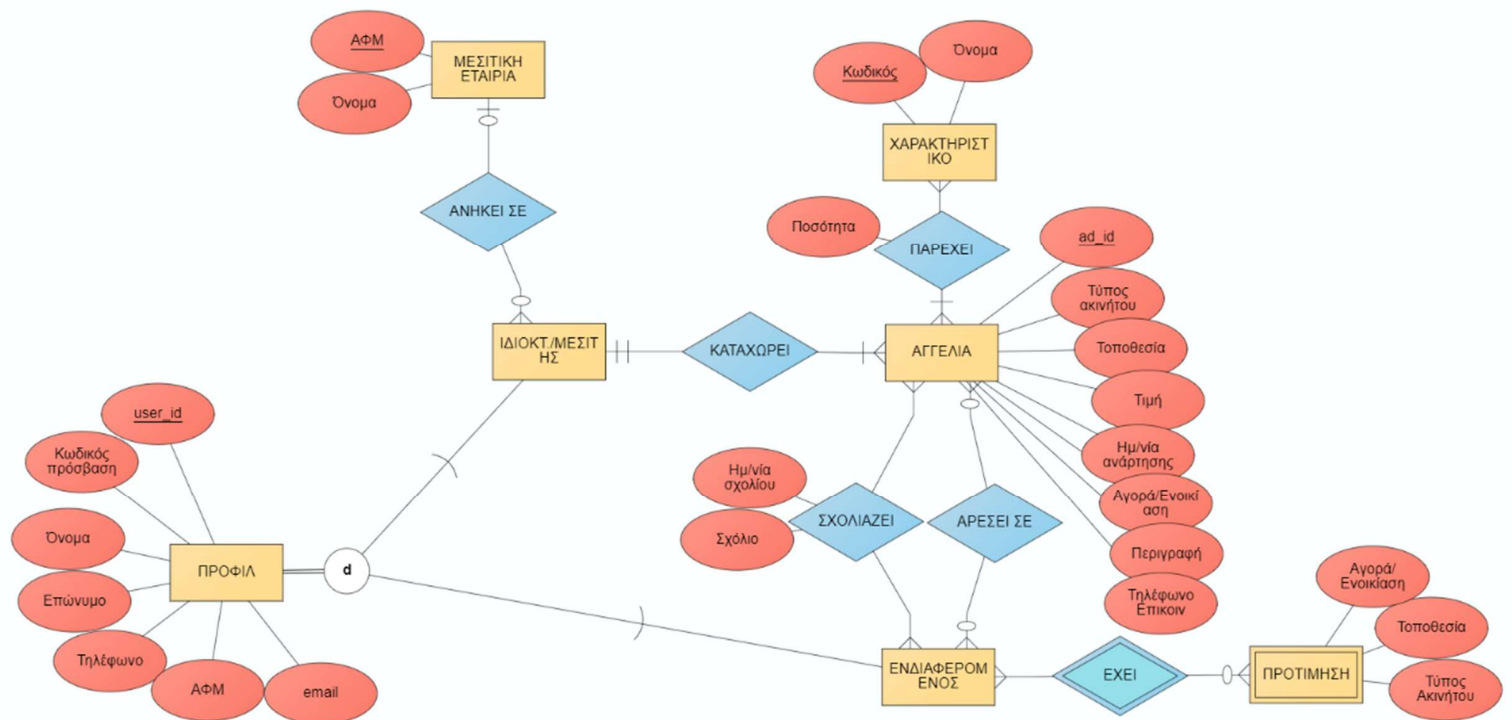
Στα πλαίσια του μαθήματος μας ανατέθηκε να υλοποιήσουμε μια εφαρμογή αναζήτησης ακινήτου. Η βασική ιδέα του μικροκόσμου που δημιουργήσαμε είναι ότι τόσο οι ενδιαφερόμενοι, όσο και οι ιδιοκτήτες/ μεσίτες μπορούν να φτιάξουν ένα προφίλ και να αναζητούν ή να καταχωρούν μία αγγελία στην εφαρμογή. Από την βασική ιδέα φαίνεται ότι στόχος μας ήταν η δημιουργία μίας ρεαλιστικής, αλλά και αμφίδρομης εφαρμογής, που απευθύνεται δηλαδή όχι μόνο σε όσους αναζητούν κατοικία, επαγγελματικό χώρο ή γη, αλλά και σε εκείνους που ενδιαφέρονται να ενοικιάσουν ή να πουλήσουν κάποιο ακίνητο.

2 ΜΕΘΟΔΟΛΟΓΙΑ, ΣΥΛΛΟΓΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ

Όταν μας ανατέθηκε το συγκεκριμένο θέμα, αρχικά ανατρέξαμε σε παρόμοιες εφαρμογές real estate στο διαδίκτυο και μελετήσαμε την δομή και την λειτουργία τους. Παράλληλα, ξεκινήσαμε την δημιουργία του μικροκόσμου μας μέσω ενός ERD, κάνοντας τις εξής παραδοχές:

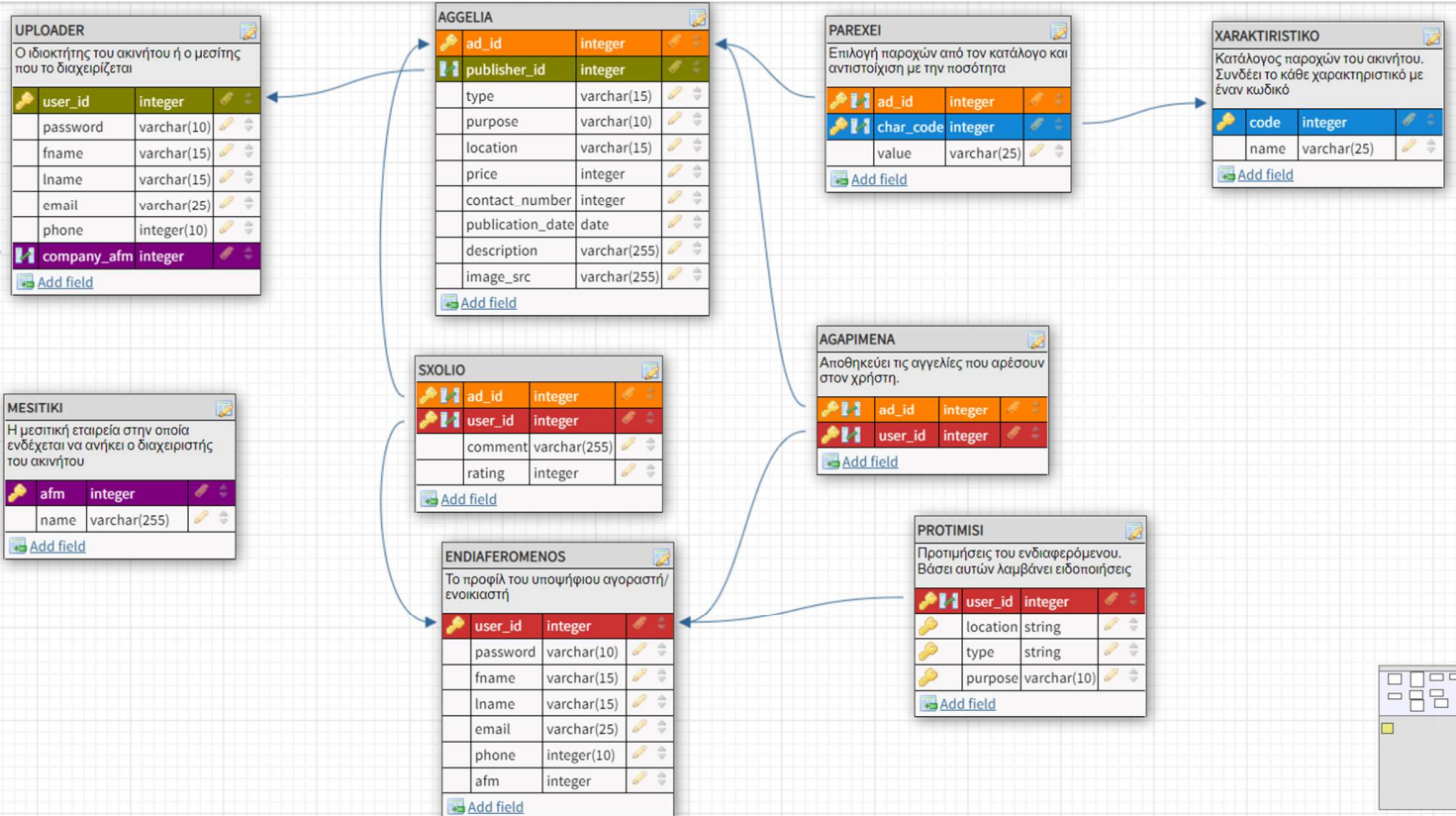
- Οι Αγγελίες και τα Χαρακτηριστικά τους ,οι Χρήστες και τα Σχόλια διαθέτουν αριθμούς που τα χαρακτηρίζουν μοναδικώς.
- Όποιος ενδιαφέρεται να χρησιμοποιήσει την εφαρμογή καλείται να φτιάξει ένα προφίλ.
- Ο κάθε χρήστης μπορεί να είναι είτε ιδιοκτήτης/ μεσίτης και να καταχωρεί μία αγγελία, είτε ενδιαφερόμενος, που φιλτράρει κάποιες αγγελίες βάσει των προτιμήσεών του.
- Η κάθε αγγελία αναφέρει τις παροχές που διαθέτει κάθε ακίνητο.
- Ο κάθε χρήστης μπορεί να αφήνει το σχόλιο του κάτω από τις αναρτημένες αγγελίες.
- Ο μεσίτης μπορεί να ανήκει σε μία μεσιτική εταιρεία.

Βάσει των παραπάνω, χρησιμοποιήσαμε το ERDMaker για να φτιάξουμε το εκτεταμένο διάγραμμα οντοτήτων-συσχετίσεων. Μετά από σταδιακές διορθώσεις και συζήτηση με τους καθηγητές κατά την ενδιάμεση παρουσίαση (8/11) καταλήξαμε στο παρακάτω μοντέλο:



Εικόνα 1: Τελικό διάγραμμα οντοτήτων-συσχετίσεων

Στην συνέχεια, ασχοληθήκαμε με την δημιουργία του αντίστοιχου σχεσιακού μοντέλου μέσω του DB Designer. Παρατηρούμε στην παρακάτω εικόνα ότι εκτός από τις οντότητες δημιουργήσαμε και τον πίνακα «ΠΑΡΕΧΕΙ» για την αντίστοιχη σχέση. Φτιάξαμε, επίσης, για τις συσχετίσεις «ΣΧΟΛΙΑΖΕΙ» και «ΑΡΕΣΕΙ ΣΕ» τους πίνακες «ΣΧΟΛΙΟ» και «ΑΓΑΠΗΜΕΝΑ» αντίστοιχα, επειδή είναι τύπου M:N.



Εικόνα 2: αντίστοιχο λογικό σχεσιακό μοντέλο

Μέχρι το τέλος του Νοεμβρίου είχαμε ολοκληρώσει την Φάση Α της εργασίας που αφορά την σχεδίαση του ERD και του αντίστοιχου relational schema και καταλήξαμε στο πώς θα διαχειριστούμε το υπόλοιπο της εργασίας.

Μέσα στον Δεκέμβριο, δημιουργήσαμε την database, εκτελώντας τις SQLite εντολές που βρίσκονται στο αρχείο create_tables.sql που συντάξαμε. Με την χρήση της βιβλιοθήκης sqlite3 της python επιτυγχάνεται η επικοινωνία της εφαρμογής με την βάση δεδομένων. Επομένως με την εκτέλεση του αρχείου create_tables.py δημιουργείται η βάση δεδομένων που είναι ακόμη κενή.

Στη συνέχεια, φτιάξαμε τα create_ads.py, create_users_csv.py, create_comm_fav_prot.py, που δημιουργούν για κάθε πίνακα αρχεία .csv με δοκιμαστικά δεδομένα που θα φορτωθούν στην βάση μέσω του insert_data.py. Στο αρχείο create_ads.py εισάγουμε τις αγγελίες και τις παροχές τους με την βοήθεια της βιβλιοθήκης requests. Πιο συγκεκριμένα, κάνουμε requests στο site της Χρυσής Ευκαιρίας λαμβάνουμε json αρχεία με τα χαρακτηριστικά της κάθε αγγελίας. Αυτά τα φιλτράρουμε κάνοντας extract τις πληροφορίες που θέλουμε και δημιουργούμε τα .csv των αγγελιών και των παροχών τους. Στα άλλα 2 create αρχεία δημιουργούμε τυχαίες πληροφορίες. Στο αρχείο create_users_csv.py λαμβάνουμε τυχαία ονόματα που παράγονται από την βιβλιοθήκη names της python και τους αντιστοιχίζουμε τα υπόλοιπα δεδομένα με τυχαίο τρόπο, όπως για παράδειγμα τον αριθμό τηλεφώνου και το αν ο εν λόγω χρήστης ανήκει σε κάποια μεσιτική εταιρεία.

Ομοίως με τυχαίο τρόπο αντιστοιχήσαμε τόσο τις προτιμήσεις και τα αγαπημένα στον κάθε ενδιαφερόμενο, όσο και τα σχόλια σε κάθε αγγελία.

Τελικά, κατά την διάρκεια της διακοπής του Ακαδημαϊκού εξαμήνου δοκιμάστηκαν για αρχή (στο SQLite DB Browser) και ύστερα υλοποιήθηκαν στο περιβάλλον της διεπαφής του χρήστη (χρησιμοποιώντας τις παραμέτρους που επέλεγε εκείνος) οι λειτουργίες προσθήκης, αλλαγής, διαγραφής αλλά και αναζητήσεις εγγραφών των κύριων πινάκων:

ΑΝΑΖΗΤΗΣΗ

1. ΑΝΑΖΗΤΗΣΗ ΑΓΓΕΛΙΑΣ ΜΕ ΠΕΡΙΓΡΑΦΗ

πχ. SELECT * FROM AGGELIA WHERE title LIKE '%key%' OR description LIKE '%key%';

2. ΦΙΛΤΡΑ ΤΟΠΟΘΕΣΙΑΣ - ΣΚΟΠΟΥ - ΕΙΔΟΥΣ - ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ (ΠΧ ΤΕΤΡΑΓΩΝΙΚΑ, ΔΩΜΑΤΙΑ)

πχ. SELECT AGGELIA.ad_id,AGGELIA.price FROM AGGELIA,PAREXEI AS P1,PAREXEI AS P2
WHERE location='Πάτρα' AND type='residence' AND purpose='rent'
AND (price BETWEEN {min_price} AND {max_price})
AND AGGELIA.ad_id=P1.ad_id AND P2.ad_id=AGGELIA.ad_id
AND P2.char_code=2 AND P2.value='1' AND P1.char_code=1;

3. ΕΥΡΕΣΗ ΑΓΓΕΛΙΩΝ ΠΟΥ ΤΑΙΡΙΑΖΟΥΝ ΣΤΟΝ ΧΡΗΣΤΗ ΜΕ ΒΑΣΗ ΤΟΝ ΑΡΙΘΜΟ ΤΟΥ

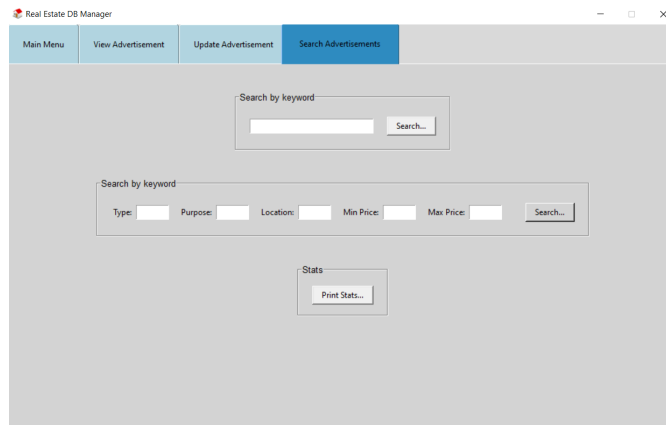
πχ. SELECT ad_id FROM AGGELIA NATURAL JOIN PROTIMISI WHERE user_id=123;

4. ΣΤΑΤΙΣΤΙΚΑ ΣΤΟΙΧΕΙΑ ΓΙΑ ΚΑΘΕ ΚΑΤΗΓΟΡΙΑ ΚΑΙ ΔΙΑΘΕΣΗ ΑΚΙΝΗΤΟΥ



The screenshot shows a web application titled "Real Estate DB Manager". It has a navigation bar with "Main Menu", "My Advertisements", and "Profile Details". The "My Advertisements" section is active, displaying a table of advertisements. The table has columns: Ad_id, Title, Location, Purpose, Price, and Contact Number. Below the table are "Action Buttons": a green "+" button, "View Advertisement", "Delete Advertisement", and "Modify Advertisement".

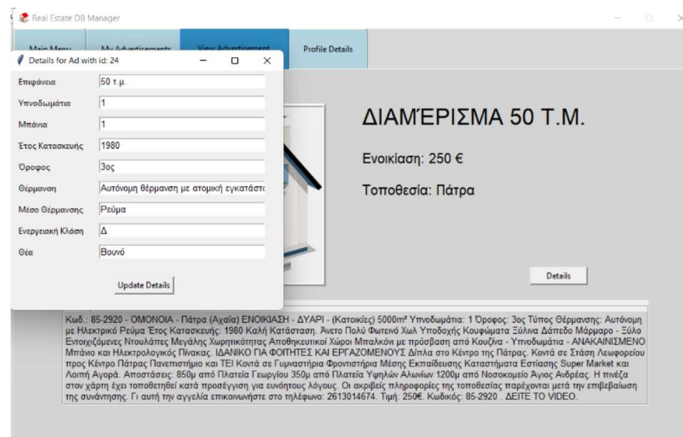
| Ad_id | Title | Location | Purpose | Price | Contact Number |
|-------|---------------------|----------|---------|--------|----------------|
| 1 | Διαμέρισμα 78 τ.μ. | Πάτρα | rent | 4000 € | 6900000000 |
| 2 | Διαμέρισμα 58 τ.μ. | Πάτρα | rent | 580 € | 6900000000 |
| 4 | Διαμέρισμα 80 τ.μ. | Πάτρα | rent | 750 € | 6900000000 |
| 5 | Διαμέρισμα 100 τ.μ. | Πάτρα | rent | 900 € | 6900000000 |
| 7 | Διαμέρισμα 32 τ.μ. | Πάτρα | rent | 400 € | 6900000000 |
| 10 | Διαμέρισμα 52 τ.μ. | Πάτρα | rent | 450 € | 6900000000 |
| 12 | Διαμέρισμα 28 τ.μ. | Πάτρα | rent | 280 € | 6900000000 |
| 14 | Διαμέρισμα 39 τ.μ. | Πάτρα | rent | 240 € | 6900000000 |
| 15 | Διαμέρισμα 35 τ.μ. | Πάτρα | rent | 250 € | 6900000000 |
| 20 | Διαμέρισμα 97 τ.μ. | Πάτρα | rent | 400 € | 6900000000 |
| 21 | Διαμέρισμα 60 τ.μ. | Πάτρα | rent | 230 € | 6900000000 |
| 24 | Διαμέρισμα 50 τ.μ. | Πάτρα | rent | 250 € | 6900000000 |
| 25 | Διαμέρισμα 115 τ.μ. | Πάτρα | rent | 500 € | 6900000000 |
| 27 | Διαμέρισμα 45 τ.μ. | Πάτρα | rent | 350 € | 6900000000 |
| 28 | Διαμέρισμα 60 τ.μ. | Πάτρα | rent | 280 € | 6900000000 |



ΤΡΟΠΟΠΟΙΗΣΗ

1. ΕΠΙΛΟΓΗ ΚΑΙ ΤΡΟΠΟΠΟΙΗΣΗ ΣΤΟΙΧΕΙΩΝ ΚΑΙ ΠΑΡΟΧΩΝ ΑΓΓΕΛΙΑΣ

πχ. UPDATE AGGELIA SET title='ΝΕΟΣ ΤΙΤΛΟΣ' WHERE ad_id=123;



ΛΕΙΤΟΥΡΓΙΕΣ

1. ΕΙΔΟΠΟΙΗΣΗ ΠΤΩΣΗΣ ΤΙΜΗΣ ΣΕ ΚΑΘΕ ΠΤΩΣΗ ΤΙΜΗΣ ΓΙΑ ΑΓΑΠΗΜΕΝΕΣ ΑΓΓΕΛΙΕΣ
2. ΕΜΦΑΝΙΣΗ ΑΓΓΕΛΙΑΣ ΠΟΥ ΕΠΙΛΕΓΕΤΑΙ ΜΑΖΙ ΜΕ ΣΧΟΛΙΑ ΚΑΙ ΜΕΣΟ RATING

3 ΑΞΙΟΛΟΓΗΣΗ

Μετά την παρουσίαση του τρόπου χρήσης του προγράμματος βλέπουμε ότι είναι λειτουργικό και αρκετά ρεαλιστικό. Παρέχονται όλες οι λεπτομέρειες που χρειάζεται ένας χρήστης κατά την αναζήτηση ακινήτου. Δεν υπάρχουν σφάλματα και δεν παρουσιάζονται προβλήματα στον χρήστη κατά την εκτέλεσή του.

Ακόμη, ο μέσος χρήστης θα δυσκολευόταν να χειριστεί το πρόγραμμα από το command line της python. Επομένως, με την βοήθεια της βιβλιοθήκης tkinter δημιουργήσαμε την γραφική διεπαφή της εφαρμογής μας, καθώς θεωρήσαμε ότι με αυτόν τον τρόπο θα γινόταν ακόμα πιο εύχρηστη.

4 ΚΥΡΙΕΣ ΕΝΕΡΓΕΙΕΣ

Οι επιμέρους εργασίες που δημιούργησαν το παρόν αποτέλεσμα μοιράστηκαν ως εξής:

- Μετά από μελέτη των ζητημάτων και άλλων διαθέσιμων υλοποιήσεων και συζήτηση, οργανώθηκαν τα διαγράμματα ERD και Relational schema, από κοινού.
- Επίσης συνεργατική ήταν η δημιουργία των SQLite queries που εφαρμόστηκαν στην υλοποίησή μας και η παραγωγή δεδομένων που εισάχθηκαν.
- Διαφοροποίηση υπήρξε στην συνέχεια του project, ο φοιτητής Ντούνης Κωνσταντίνος ασχολήθηκε με την δημιουργία της γραφικής διεπαφής, ενώ ο Στέφας Φίλιππος με την συγγραφή της αναφοράς.

5 ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ

- i. Εγκατάσταση της βιβλιοθήκης pandas και pillow (pip install ...)
- ii. Εκτέλεση αρχείου create_tables.py (Τρέχει το SQL script για δημιουργία πινάκων)
- iii. Εκτέλεση αρχείου insert_data.py
- iv. Εκτέλεση αρχείου gui.py
- v. Στο Login screen χρησιμοποιούμε όνομα χρήστη admin – κωδικό admin για να χρησιμοποιήσουμε όλες τις διαθέσιμες λειτουργίες του διαχειριστή.

Στην python χρησιμοποιήθηκαν οι εξής βασικές βιβλιοθήκες:

- Sqlite3: για την επικοινωνία με την βάση και την εκτέλεση των queries
- Pandas: για την εισαγωγή των δεδομένων
- Tkinter: για τη δημιουργία του γραφικού περιβάλλοντος
- Requests: για το αίτημα των αγγελιών
- Json: για επεξεργασία των αιτημάτων

Ο σύνδεσμος για το GitHub: <https://github.com/KonstantinosN00/DatabasesProject>