ΠΛΗΠΡΟ 2022-2023 Τελικό project.

- **Project ID** : 53
- **Τίτλος** : Time management app
- **Περιγραφή** : Ανάπτυξη εφαρμογής διχείρισης χρόνου δραστηριοτήτων.
- **Φοιτητές** : Παύλος Φουρφουριανάκης, Αντώνης Μαρίνος, Μάριο Αλέξανδρος Τσιμπούκας, Λευτέρης Κλημεντίδης.
- **Σύμβουλος Καθηγητής** : Νίκος Ευθυμιόπουλος.

Επιλογή εργασίας

Μετά απο την ανακοίνωση των θεμάτων και αφού είχαν ήδη οριστκοποιηθεί οι ομάδες, κάναμε μια συνάντηση (πρίν απο την τελική με τον καθηγητή) προκειμένου να αποφασίσουμε ποιό απο τα διαθέσιμα θέματα θα αναλαμβάναμε. Οι 3 επιλογές μας ήταν μεταξύ των 53, 7, 22 και τελικά καταλήξαμε στο 53 καθώς συμφωνήσαμε ότι πέραν των δυσκολιών (είχε κάποια πράγματα τα οποία δεν είχαμε δει στην ύλη), η εφαρμογή είχε αρκετό ενδιαφέρον και μεγάλη χρηστική αξία.

Οργάνωση

Αρχικά πρσπάθήσαμε να οριοθετήσουμε τον χρόνο και τον τρόπο με τον οποίο θα μοιράζαμε τις δουλειές. Η διαδικασία αυτή έγινε υπολογίζοντας πέραν της “εμεπειρίας” και της τρβής κάθε φοιτητή με την python, την διάθεση του γενικότερου χρόνου βάσει προσωπικών και επαγγελματικών υποχρεώσεων, καθώς και λαμβάνοντας υπόψιν και τις υπόλοιπες θεματικές.

Καταλήξαμε σε ενα tsk list το οποίο συζητήσαμε και κάναμε commit βάσει των προαναφερθέντων, αλλά μια πολύ σημαντική παράμετρος που υπήρξε κατά την δάκρεια όλης της υλοποίησης ήταν η συνεννοήση και η συνεργατική δουλειά. Οι βασικοί πυλώνες της υλοποίησης στους οποίους καταλήξαμε ήταν οι εξής :

- Σχεδιασμός → Ανάλυση απαιτήσεων, ΔΡΔ, Βάση δεδομένων.
- Υλοποίηση βάσης → SQLlite, πίνακες, συναρτήσεις κλάσεις, σύνδεση βάσης.
- Υλοποίηση βασικού κώδικά-UI → Σχεδιασμός μενού, πρόσβαση στην βάση, διαχείριση χρηστών, διαχείριση δραστηριοτήτων, διεπαφή χρήστη.
- Γραφικές απεικονίσεις →Γραφήματα.
- Project management → Παρουσιάσεις, διαχείριση αρχείων, documentation.

Αφού καταλήξαμε σε ενα υψηλού επιπέδου καταμερισμό της υλοποίησης σε επι μέρους εργασίες, απφασίσαμε να μοιράσουμε αυτές τις εργασίες μεταξύ μας, πάντα βάζοντας περισότερους απο εναν να δουλέψουν μαζί καθώς, χωρίς να γνωρίζουμε ακριβώς, θεωρήσαμε ότι ο κώδικάς θα ήταν αρκετά σύνθετος, με πολλές γραμμές. Η λογική ήταν αν υπάρχει πάντα κάποιο back up αλλά και συνεχής επικοινωνία ώστε να μπορέσουμε να παρουσιάσουμε ενα ομοιόμορφο κώδικά, του οποίου τα ξεχωριστά modules θα έχουν συνοχή και συνέχεια χωρίς φυσικά να αφαιρεθεί η δημιουργική ελευθερία που θα έπρεπε να έχει κάθε ενας στο κομμάτι της υλοποίηση που του αναλογεί.

Επιπλέον, τοποθετήσαμε κάποιες αρχικές ημερομηνίες υλοποίησης προκειμένου να έχουμε έλεγχο της προόδου των εργασιών και φυσικά να είμαστε εντός σχεδιασμού και χρονοδιαγραμμάτων κατάθεσης της εργασίας. Φυσικά πολλά απο αυτά τα πρώτα deadlines επαναπροσδιορίστηκαν κατά την υοποίηση κάτι που ήταν αναμενόμενο καθώς δεν είχε κανείς μας εμπειρία απο τέτοιου μεγέθους υλοποίηση (το μέγεθος δεν είχε καμία χέση με τις υποεργασίες),

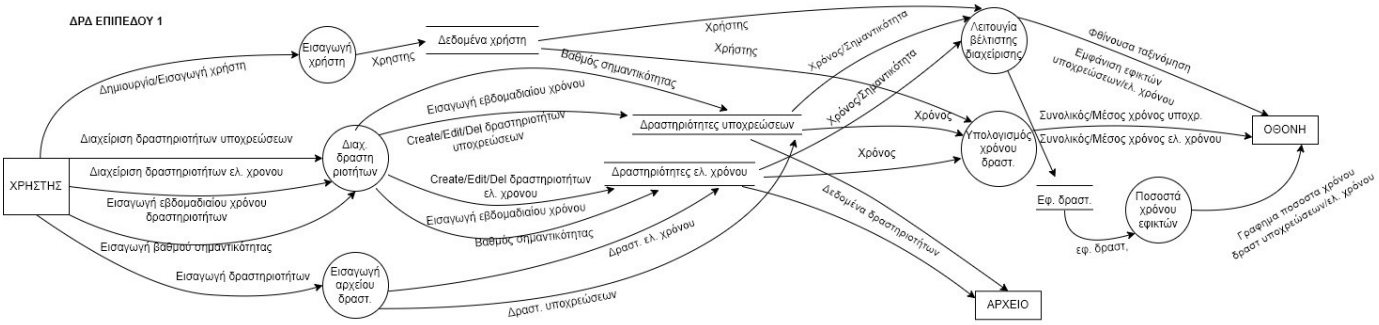
καθώς και το γεγονός ότι αντιμετωπίζαμε πρώτη φορά μια εφαρμογή με απαιτήσεις υψηλής πολυπλοκότητας.

Σχεδιασμός → Ανάλυση απαιτήσεων, ΔΡΔ, Βάση δεδομένων

Ξεκινήσαμε με την ανάγνωσή του κειμένου απαιτήσεων και τον διαχωρισμό των λειτουργικών απαιτήσεων και μη λειτουργικών απαιτήσεων, καθώς και την διατύπωση αυτών. Ήταν πολύ σημαντικό για εμάς να έχουμε μια σωστή και ξεκάθαρη εικόνα των επι μέρους λειτουργιών που θα χρειαστεί η εφαρμογή, καταλήξαμε στις εξής :

- Διαχείριση χρηστών (εισαγωγή, διαγραφή, τροποποίηση).
- Διαχείριση (εισαγωγή, διαγραφή, τροποποίηση) και χαρακτηρισμός δραστηριοτήτων (ελεύθερου χρόνου και υποχρεώσεων).
- Λειτουργία βέλτιστης διαχείρισης (ταξινόμηση δραστηριοτήτων βάσει βαθμού σημαντικότητας και σε σχέση με τον χρόνο).
- Εισαγωγή/εξαγωγή δεδομένων (σε αρχείο).
- Δημιουργία γραφικών παραστάσεων (maplotlib).
- Δημιουργία διεπαφής χρήστη (tkinter).

Έχοντας τα παραπάνω υπόψιν ξεκινήσαμε με την δημιουργία ΔΡΔ επιπέδων 0 και 1, προκειμένου να καταλάβουμε καλύτερα και σχηματικά την ροή των δεδομένων και να αρχίσουμε να σχηματίζουμε κάποιες απο τις βασικές διεργασίες της εφαρμογής. Ένα απο τα διαράμματα μας φαίνεται παρακάτω για να δώσουμε μια ιδέα πως στήσαμε την λογική της εφαρμογής μας.



Υλοποίηση βάσης → SQLlite, πίνακες, συναρτήσεις κλάσεις

Αναφορικά με την βάση δεδομένων, είχαμε την επιλογή ναδημιουργήσουμε δομές δεδομένων μέσα στην python παρόλα, αυτά αποφασίσαμε να προχωρήσουμε με την χρήση SQL (συγκεκριμένα SQLlite, την οποία και μάθαμε στην ΠΛΗ11), θεωρώντας ότι θα κάνει το μοντέλο μας πιο κλιμακούμενο και θα δούλευε πιο αποδοτικά σε πολλά δεδομένα. Αυτή η απόφαση θεωρούμε ότι σε μεγάλο βαθμό έκρινε και την πολυπλοκότητα της υλοποίησης μας, αλλά δυστυχώς έκανε και πολύπιο δύσκολή την αλλαγή/διόρθωση τυχόν σφαλματων.

Σε κάθε περίπτωση, χτίσαμε την βάση στην λογική ενος απλού σχήματος βασικών πινάκων με τον διαχωρισμό σε “Χρήστες”, “Δραστηριότητες υποχρεώσεων” και “Δραστηριότητες ελεύθερου χρόνου” και ενώσαμε σχεσιακά τους πίνακες με το id του χρήστη. Παράλληλα κάναμε μια ακόμα επιλογή (η οποία προέκυψε απο παρανόηση στην ανάγνωση του κειμένου) και αντί να τοποθετεί ο χρήστης συνολικό ελεύθερο χρόνο, επιλέξαμε να βάζουμε χρόνο ανα δραστηριότητα (το οποίο

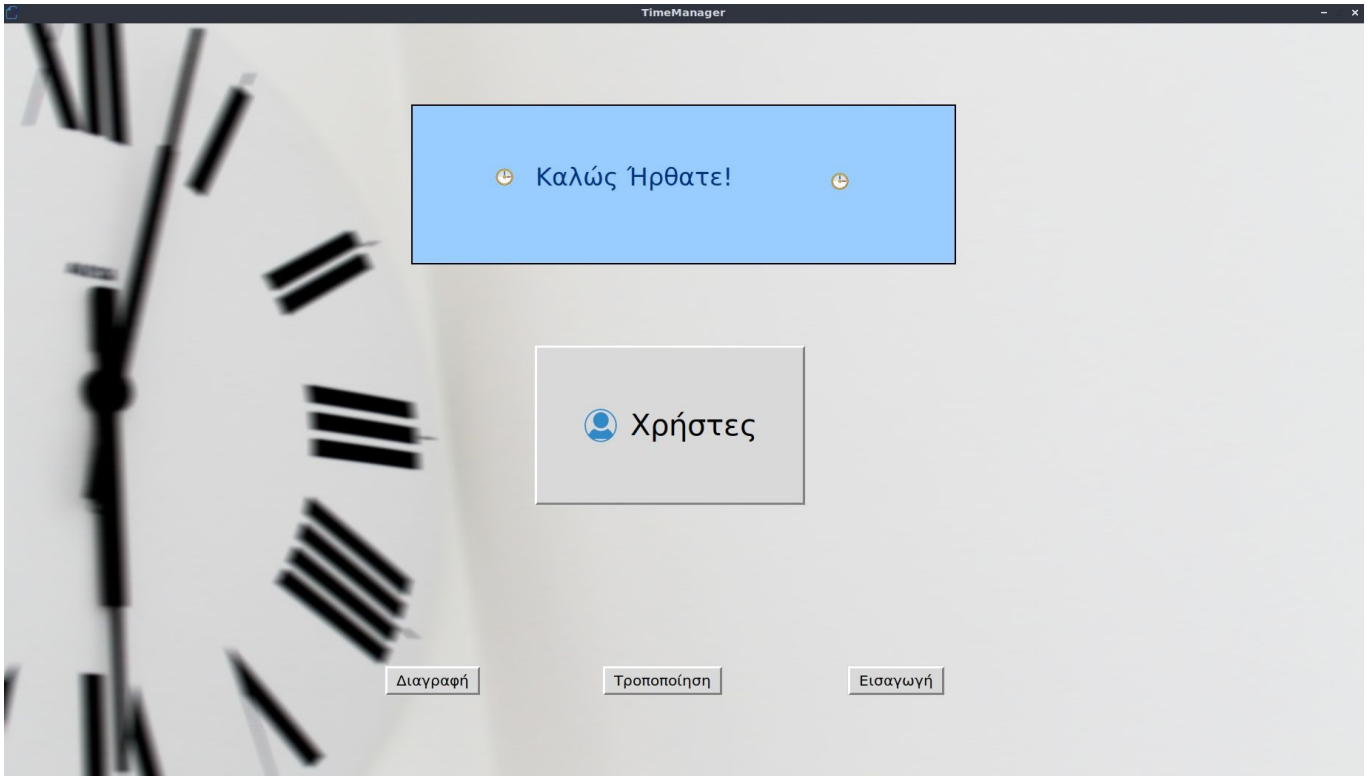
φυσικά ήταν πεδίο στην βάση). Αυτό ατην συνέχεια και καθώς είχμε συνδέσει όλα τα κομμάτια κώδικα, ήταν πάρα πολύδύσκολο να το αλλάξουμε και κατ' επέκταση το κρατήσαμε λόγω χρόνου κυρίως.

Στην συνέχεια χτίσαμε τις ανάλογες δομές δεδομένων στην python για να μπορούν να τροφοδοτούν την βάση με input και αντιτοιχα να τραβάνε πληροφορία απο την βάση. Κάναμε χρήση του sqllite3 module στην python για να συνδέσουμε τις βάσεις. Παράλληλα χτίσαμε την κλάση database και τις εντολές για δημιουργία των αντίστοιχων πινάκων (εφόσον δεν υπάρχουν). Μετά δημιουργήσαμε συναρτήσεις προκειμενου να μπορεί ο χρήστης της εφαρμογής να κάνει αλλαγές στην βάση (διαγραφή γραμμή, διαγραφή όλων των χρηστών κτλ.). Επιπλέον συναρτήσεις για την διαχείριση δεδομένων δραστηριοτήτων (προθήκη δραστηριότητας, τροποποίηση τύπου δραστηριότητας, ορισμός κλειδιών, διαγραφή δραστηριοτήτων κτλ.).

Υλοποίηση βασικού κώδικά → Σχεδιασμός μενού, πρόσβαση στην βάση, διαχείριση χρηστών, διαχείριση δραστηριοτήτων, διεπαφή χρήστη .

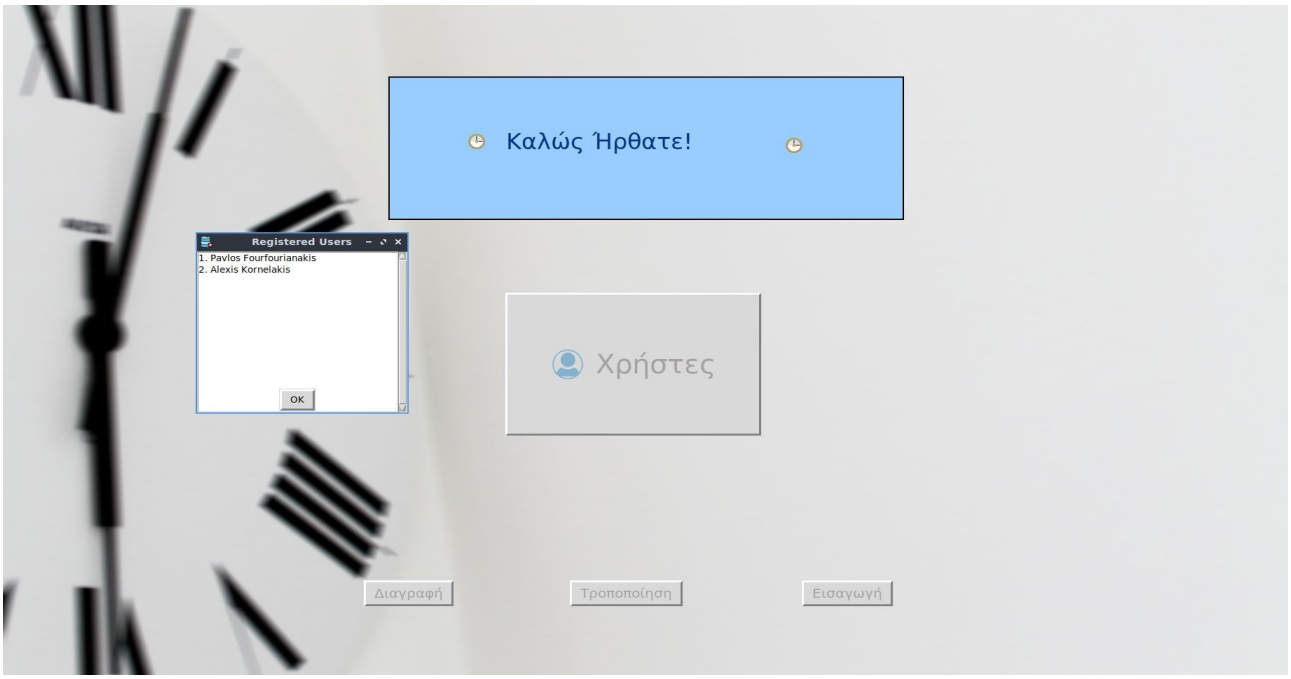
Στο main πρόγραμμα μας αρχικά κάνουμε ειαγωγή των απαραίτητων βιβλιοθηκών καθώς και του database προγράμματος μας το οποίο θα χρησιμοποιηθεί για την διαχείριση της βάσης απο την python. Ξεκινάμε με την εισαγωγή των tkinter για την δημιουργία της διεπαφής χρήστη, τα image components απο την PIL για την διαχείριιση (αλλαγή μεγέθους), προβολή, φόρτωση και αποθήκευση εικόνων.

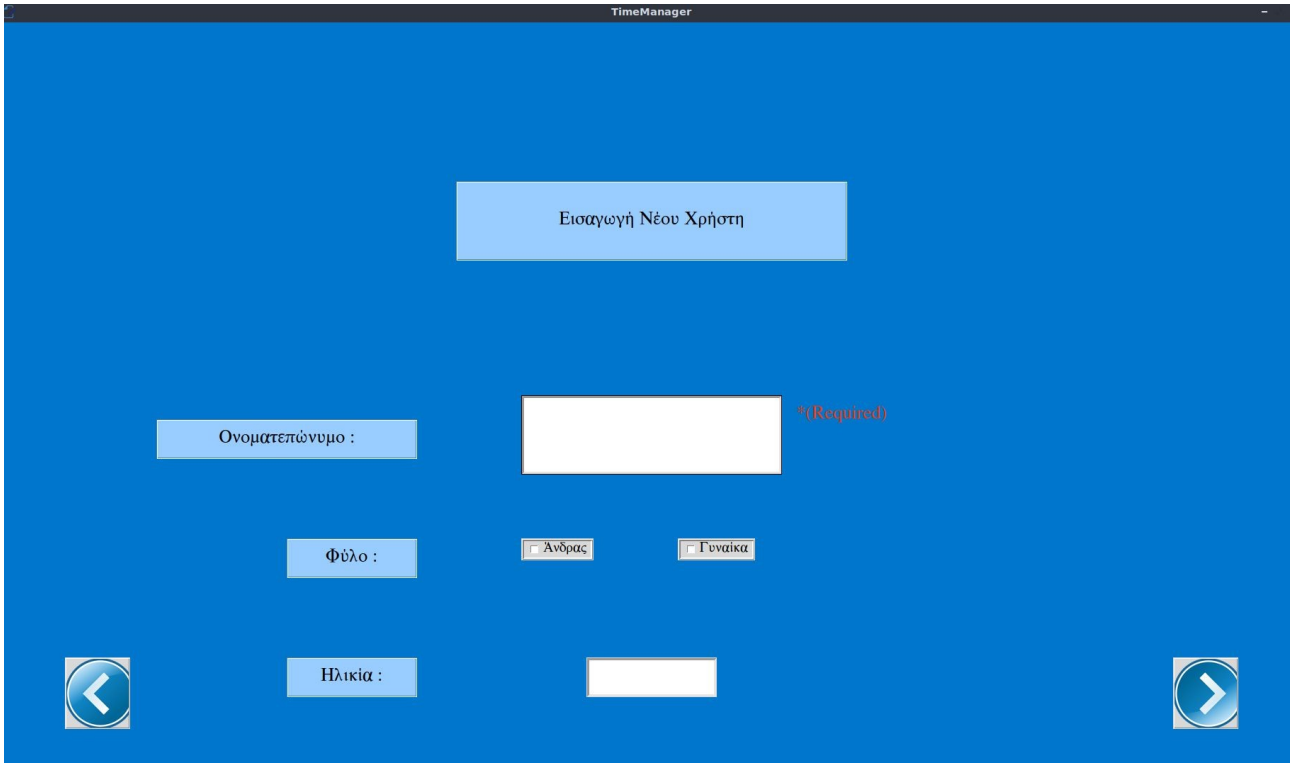
Μετά απο την φόρτωση της βάσης ξεκινάμε με το αρχικό μενού που βάζουμε μηνύματα και κουμπιά για το βασικό navigation του χρήστη και στην συνέχεια φορτώνουμε τις εικόνες, ενώ έχουμε ήδη επιλέξει τα χρώματα καθώς και το μέγεθος με τον τρόπο που θα τρέξει η εφαρμογή (popout window). Όλες οι παραπάνω ενέργειες περιγράφονται στις συναρτήσεις mainMenu & __init__ ενώ η επιλογή χρώματος για το background γίνεται αμέσως μετά απο την εισαγωγή των βιλβιθηκών. Θεωρούμε ότι το UI στο βασικό μενού είναι αποδεκτό, σίγουρα θα μπορούσε να βελτιωθεί περαιτέρω. Ενα παράδειγμα φαίνεται παρακατω με ενα screen shot απο το κεντρικό μενού.



Συνεχίσαμε με την εισαγωγή σε διαφορετικά submenu ανάλογα με τιςεπιλογές του χρήστη, χτίζοντας τις κατάλληλες συναρτήσεις και φυσικά το αντίστοιχο UI, προκειμένου να γίνονται οι κατάλληλες ενέργειες ανάλογα με τα κουμπιά τα οποία πατάει ο χρήστης. Εδω γενικά φαίνεται πόση δουλειά έχει γίνει στον κώδικά με την tkinter για να γίνονται τα sitch frame actions και την σωστή αντιστοίχηση των κουμπιών με functionalities. Αυτό στο σύνολο του είχε αρκετή δυσκολία καθώς δεν είχαμε στο παρελθόν ασχοληθεί με έργο τέτοιας κλίμακας και την χρήση εικόνων και γραφικών, οπότεείχε αρκετό ψάξιμο και trial and error προσέγγιση.

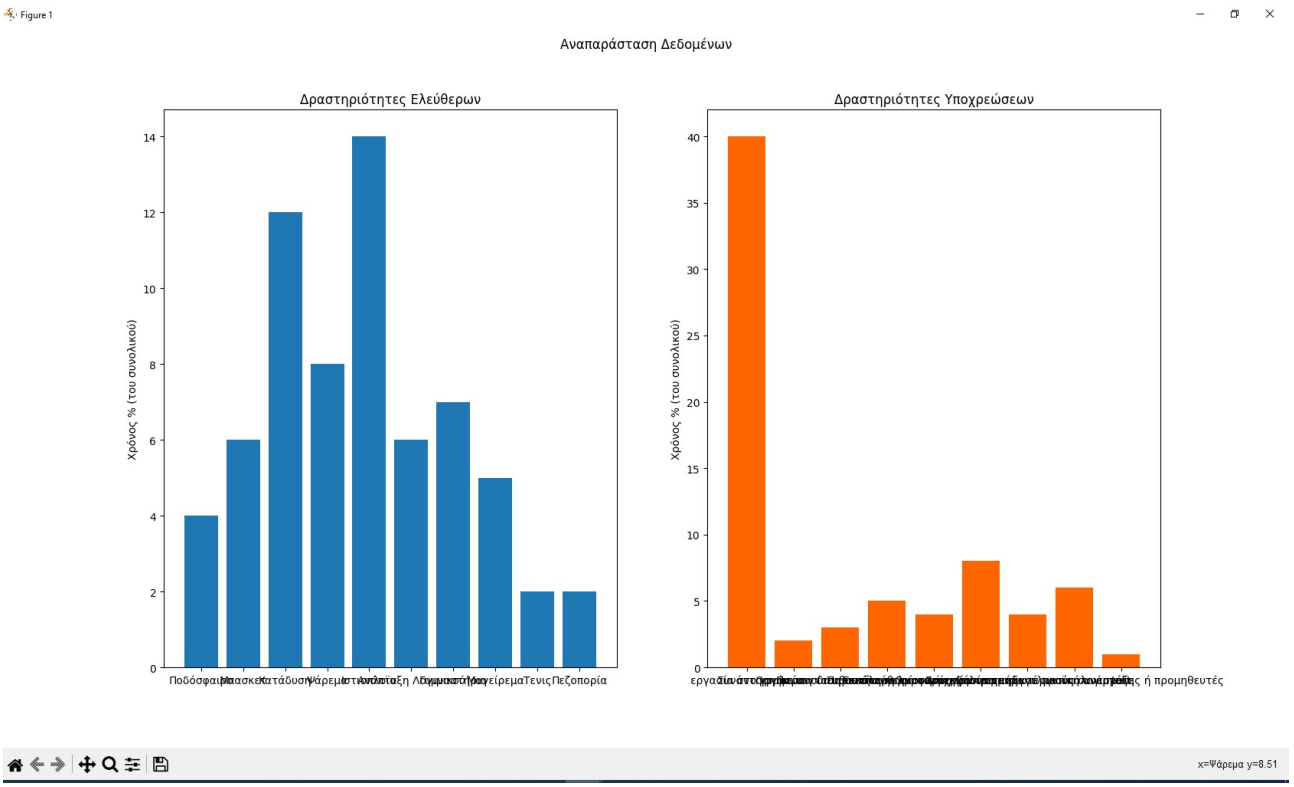
Επιλέον έχουν φτιαχτεί πολλές συναρτήσεις σε αυτό το κομμάτι με διαφορετικά σενάρια (if, elif, else, καθώς και χρήση try, except) για την διαχείριση των βασικών entities της βάσης, εισαγωγή και αποθήκευση δεδομένων, διαγραφή, updates και αλλαγές. Επίσης έχει υλοποιηθεί όπου χρειάζεται αμυντικόςπρογραμματισμός προκειμένου να βεβαιωθούμε ότι δεν θα επτρέψουμε στν χρήστη να ειάγει στην βάση μη αποδεκτές τιμές. Παρακάτω ορισμένα παραδείγματα απο τα διαφορετικά μενού της εφαρμογής.

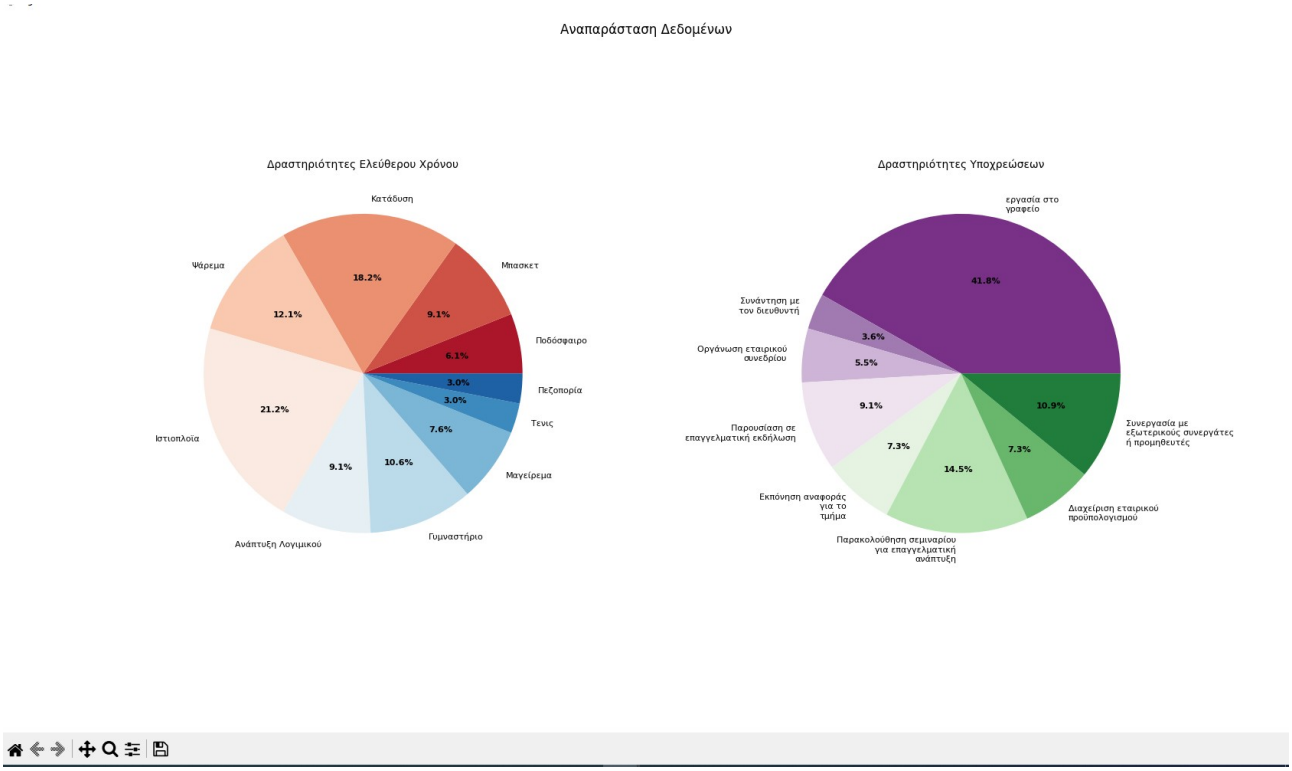




Γραφικές απεικονίσεις → Γραφήματα

Με την χρήη της matplotlib όπως προτάθηκε και στην εκφώνηση της άσκησης έγινε η σχετική διαχείριση των γραφικών παραστάσεων της εφαρμογής. Αρχικά η προσπάθεια έγινε με την προβολή γραφημάτων τύπου bar (bar charts), αλλά καθώς το αποτέλεσμα ασθητικά αλλά και χρηστικά δεν ήταν το αναμενόμενο, υοποιήθηκε προβολή γραφήματος τύπου “πίτας” (pie chart). Παρδειγματα των προαναφερθέντων μπορείτε να δείτε παρακάτω.





Project management → Παρουσιάσεις, διαχείριση αρχείων, documentation

Για την διαχείριση του project χρησιμοποιήσαμε το gihub (project documentation, version control σε repository) καθώς και το discord μαζί με κάποια features του google drive.

Σαν τελικό γενικό σχόλιο, κάναμε κάποιες επιλογές που δυσκόλεψαν πολύ τις διαδικασίες και το σύνολο του κώδικά που έπρεπε να γράψουμε. Επιπλέον δυσκολευτήκαμε να διορθσουμε λάθη καθώς η διασύνδεση των διαφορετικών συστημάτων και κομματιων κώδικά το έκανε κάποιες φορές πρακτικά αδύνατο (πρόβλημα συνολιικού χρόνου). Σίγουρα όμως μάθαμε πολλά μέσα απο όλη την διαδικασία που κληθήκαμε να διαχειριστούμε και είχαμε την ευκαιρία να εφαρμόσουμε γνώσεις απο άλλα κομμάτια της πληροφορικής κάθως και να διασυνδέσουμε διαφορετικά συστήματα, πράγμα που είχε πολύ μεγάλο ενδιαφέρον.