

TASK MANAGEMENT SYSTEM

1. Γενική Περιγραφή

Στο πλαίσιο αυτής της εργασίας υλοποιήθηκε ένα σύστημα διαχείρισης εργασιών, δηλαδή μια εφαρμογή που επιτρέπει στον χρήστη να οργανώνει, να παρακολουθεί και να διαχειρίζεται τις εργασίες του.

Ειδικότερα, η εφαρμογή *MediaLab Assistant* παρέχει στον χρήστη τις εξής δυνατότητες:

- Δημιουργία, προβολή, τροποποίηση και διαγραφή εργασιών
- Δημιουργία, προβολή, τροποποίηση και διαγραφή κατηγοριών
- Δημιουργία, προβολή, τροποποίηση και διαγραφή προτεραιοτήτων
- Δημιουργία, προβολή, τροποποίηση και διαγραφή υπενθυμίσεων
- Υπολογισμός στατιστικών για τις εργασίες του
- Αναζήτηση εργασιών

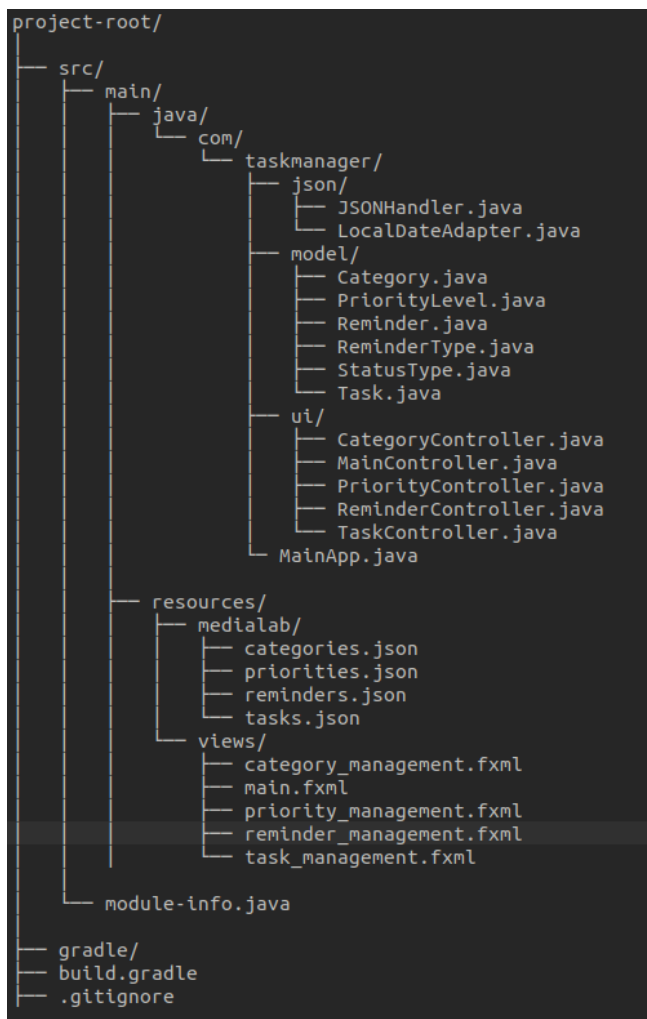
2. Σχεδιασμός και Υλοποίηση

Η εφαρμογή βασίζεται στον αντικειμενοστραφή προγραμματισμό, όπου οι κύριες οντότητες (εργασίες, κατηγορίες, προτεραιότητες και υπενθυμίσεις) εκπροσωπούνται από αντίστοιχες κλάσεις. Η εφαρμογή υλοποιεί σχέσεις μεταξύ αντικειμένων με βάση την αρχή has-a. Η σχέση has-a δείχνει ότι ένα αντικείμενο περιέχει άλλα αντικείμενα ως μέλη.

Η διάταξη του κώδικα της εφαρμογής, για την καλύτερη κατανόηση των περιεχομένων της εργασίας και πως αυτά επικοινωνούν, φαίνεται στην *εικόνα 1*.

Η λογική της εφαρμογής είναι διαχωρισμένη σε controllers, οι οποίοι συνδέουν το backend με το GUI. Για τους περισσότερους περιέχεται και τεκμηρίωση σύμφωνα με τις προδιαγραφές του εργαλείου javadoc.

Συγκεκριμένα ο MainController.java αποτελεί την καρδιά της εφαρμογής, λειτουργώντας ως ο βασικός διαχειριστής όλων των δεδομένων και των αλληλεπιδράσεων που εμφανίζονται στο κεντρικό παράθυρο. Αρχικοποιείται μόλις η εφαρμογή ξεκινήσει, καθώς η κλάση MainApp αναλαμβάνει να του περάσει όλα τα απαιτούμενα δεδομένα μέσω των μεθόδων setTasks, setCategories, setPriorities, και setReminders. Με την εισαγωγή αυτών των δεδομένων, ο MainController αναλαμβάνει να δημιουργήσει και να εμφανίσει τις κατάλληλες λίστες (ListViews) στο main.fxml και να προετοιμάσει τη αλληλεπίδραση με τον χρήστη. Ορίζει τους listeners για όλα τα στοιχεία του main.fxml. Αυτοί επιτρέπουν το double-click για την τροποποίηση στοιχείων όπως εργασίες, κατηγορίες, προτεραιότητες, ή υπενθυμίσεις με χρήση και των δευτερευόντων controllers.



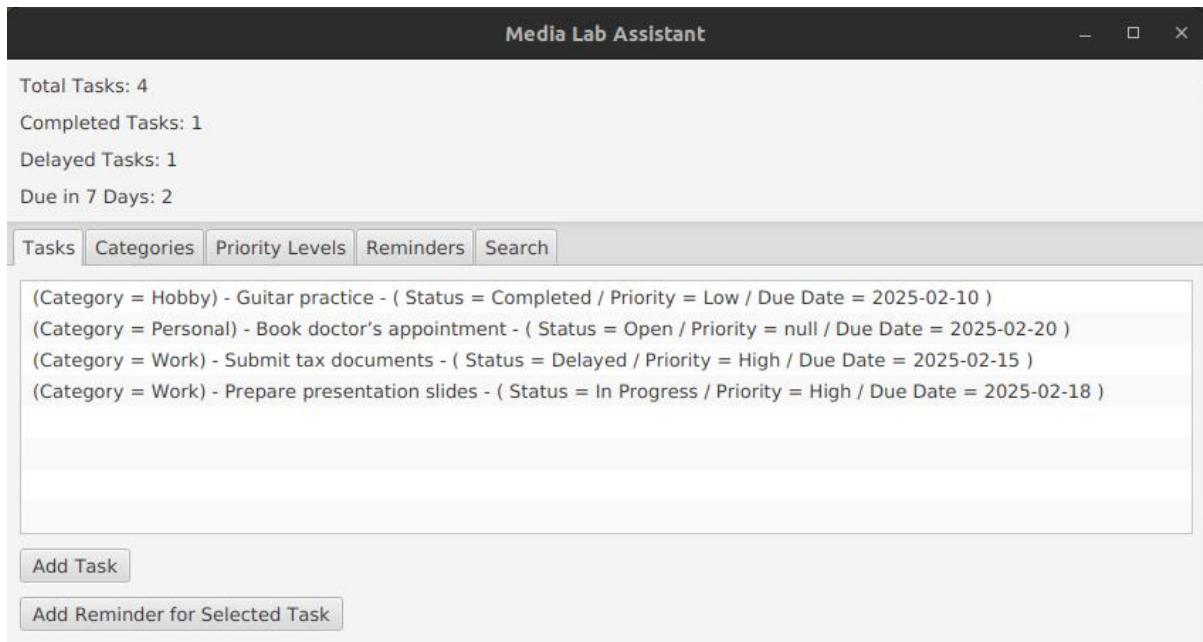
(Εικόνα1)

Ταυτόχρονα υπάρχουν controllers που είναι υπεύθυνοι για την αποθήκευση και την ανάκτηση των δεδομένων απο JSON αρχεία. Συγκεκριμένα χρησιμοποιήθηκαν οι βιβλιοθήκες FileReader και google.gson για την επικοινωνία του συστήματος με τα json αρχεία που υπάρχουν στον φάκελο medialab. Ο φάκελος αυτός περιέχει 1 json αρχείο για κάθε σημαντική κλάση (task.json, categories.json, priorities.json και reminders.json) Μαζί με τον κώδικα της εργασίας παραδίδεται και ένα test.txt το οποίο περιέχει dummy data για αυτά τα json files (στο link του google dirve που δίνεται πιο κατω)

Για την παρουσίαση αυτών των δεδομένων στη γραφική διεπαφή (UI), δημιουργούνται γραφικά στοιχεία ListViews, τα οποία χρησιμοποιούνται απο τα fxml files. Συγκεκριμένα, παρουσιάζονται οι εργασίες ταξινομημένες ανα κατηγορία αλλά χωρίς εμφανή διαχωρισμο καθώς θα έπρεπε να χρησιμοποιηθεί διαφορετική περιπλοκή δομή (οπου δεν χρειάζεται σύμφωνα με την σημείωση της ενοτητας A3). Ακόμα έχουμε κάνει την εξής παραδοχή:

- Για την δημιουργία μιας ειδοποίησης θα πρέπει να έχει επιλεγθεί το task στο οποίο θα αντιστοιχεί αυτή η ειδοποίηση

Επιπλέον, η εφαρμογή υποστηρίζει την εμφάνιση παραθύρων διαλόγου για τη δημιουργία ή επεξεργασία δεδομένων καθώς και Alert Boxes για πιθανά σφάλματα κατα την εισαγωγή των δεδομένων



3. Κώδικας - Εκτέλεση Εφαρμογής

Η εφαρμογή υλοποιήθηκε με:

- openjdk 17.0.13 2024-10-15
- JavaFX 17.0.13 (modules: javafx.controls,javafx.fxml)
- IntelliJ IDEA

Ο κώδικας υπάρχει στο ακόλουθο github repository (θα γίνει public μετά την διορία)

https://github.com/ntua-el20827/NTUA_Multimedia_Technology

Ταυτόχρονα, το test.txt με τα dummy data και αναλυτική περιγραφή των λειτουργιών των controller μπορούν να βρεθούν σε αυτό το link

Για την εκτέλεση της εφαρμογής:

- Βεβαιωθείτε ότι έχετε κατεβάσει την JavaFX (<https://gluonhq.com/products/javafx/>) καθώς χρειαζόμαστε τα modules javafx.controls,javafx.fxml
- Κατεβάστε τον κώδικα υλοποίησης
- Στο φακελο TaskManagerFX/ τρέξτε τις ακόλουθες εντολές:
 - o ./gradlew clean build
 - o ./gradlew run --args="--module-path <path to javafx sdk/lib> --add-modules javafx.controls,javafx.fxml"