



Αναφορά για την Τελική Εργασία του «Τεχνολογία Πολυμέσων»

Δημουλάς Ιωάννης, 03120083

Η εφαρμογή MediaLab.Assistant αναπτύχθηκε με την χρήση του Eclipse IDE for Java Developers – 2024-12. Έγινε χρήση του framework Maven σε Java JDK-21. Ο χρήστης, αφού κατεβάσει και στήσει τις άνωθι προαπαιτήσεις, καλείται να ανοίξει ένα terminal στο root directory της εφαρμογής που βρίσκεται το αρχείο pom.xml και να εκτελέσει την εντολή *mvn clean install*. Τέλος, μέσα από το Eclipse IDE, να εκτελέσει την εφαρμογή επιλέγοντας Run > Run As > App.

Ο διαχωρισμός των ορισμένων κλάσεων έγινε στα εξής πακέτα, που εντοπίζονται στο directory: *Assistant/src/main/java*:

- **MediaLab.Assistant**: Περιλαμβάνει τις κλάσεις των Controllers των διεπαφών, καθώς και της κλάσης *App.java* που εμπεριέχει την *main* μέθοδο.
- **Models**: Περιλαμβάνει τις κλάσεις των βασικών αντικειμένων που χειρίζεται η εφαρμογή, όπως των εργασιών, των κατηγοριών κλπ.
- **Services**: Περιλαμβάνει κλάσεις που χειρίζονται τις λίστες των αντικειμένων κλάσεων του Models, που έχουν οριστεί κατά την διάρκεια της εφαρμογής. Ακολουθούν Singleton Design Pattern προκειμένου να διασφαλιστεί η ορθή προσπέλαση κοινόχρηστων αντικειμένων και η συνέπεια των δεδομένων.
- **Utils**: Περιλαμβάνει miscellaneous κλάσεις, συμπεριλαμβανομένων των Encoder και Decoder που διαχειρίζονται την κωδικοποίηση και αποκωδικοποίηση, αντίστοιχα, των δεδομένων σε Json αρχεία για αποθηκευτικούς λόγους.

Τα συγκεκριμένα Json αρχεία βρίσκονται στον φάκελο medialab στο root directory κι έχουν την κάτωθι αρχιτεκτονική:

categories.json: [{"title": String, "description": String}, {...}]

priorities.json: [{"index": int, "title": String, "description": String}, {...}]

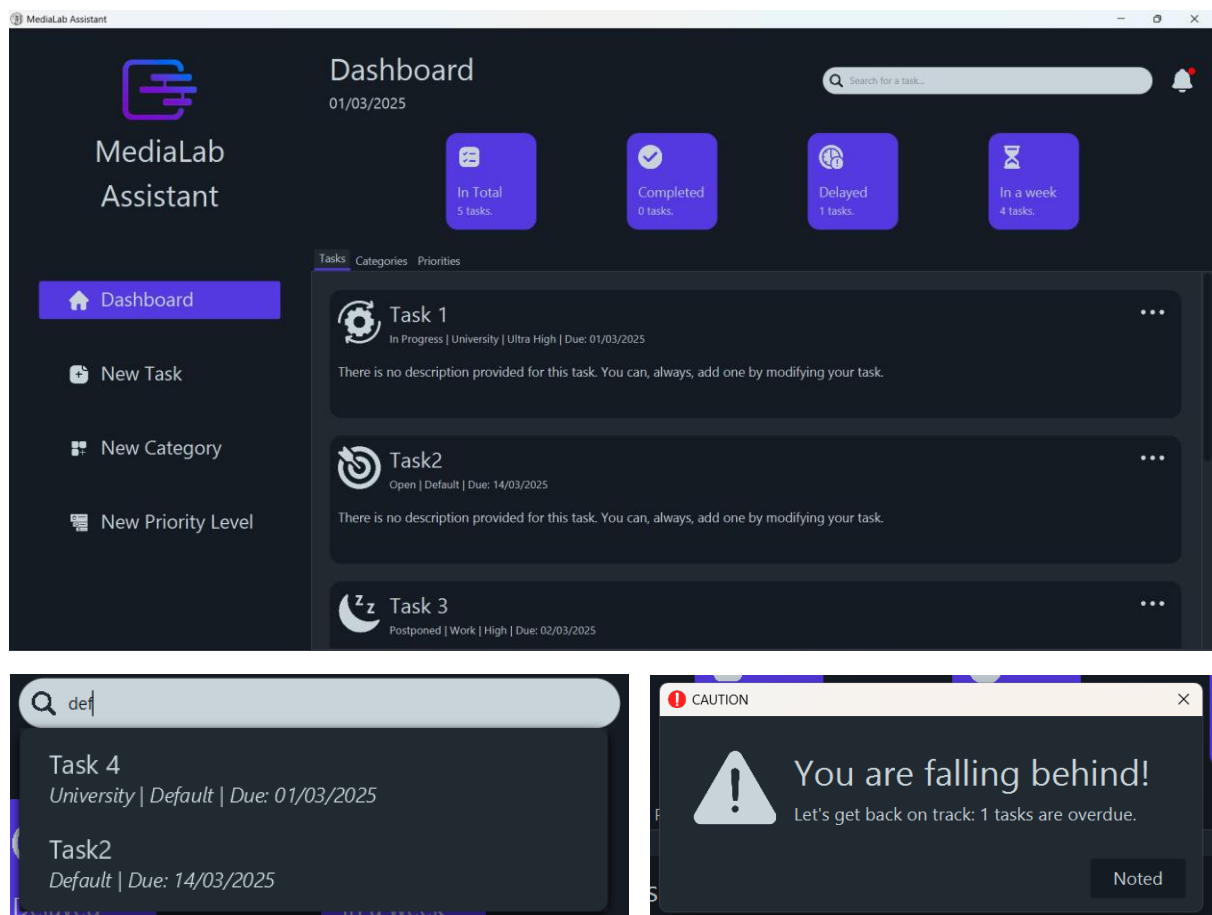
tasks.json: [{"title": String, "description": String, "deadline": String, "category": String, "priority": String, "Status": String, "notifications": [{"type": String, "notificationDate": String}, {...]}], {...}]

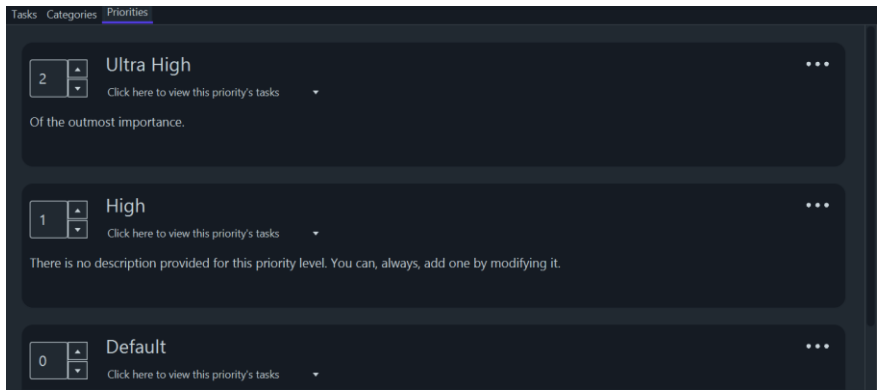
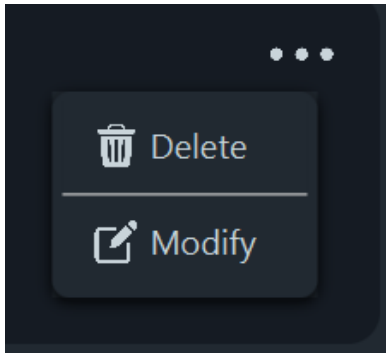
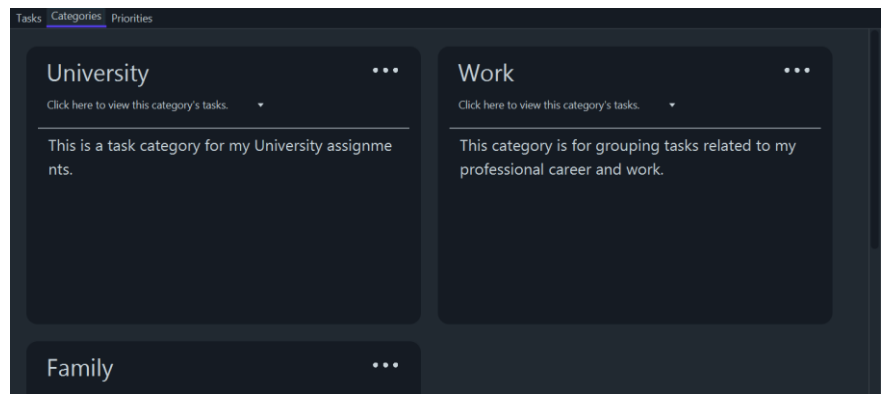
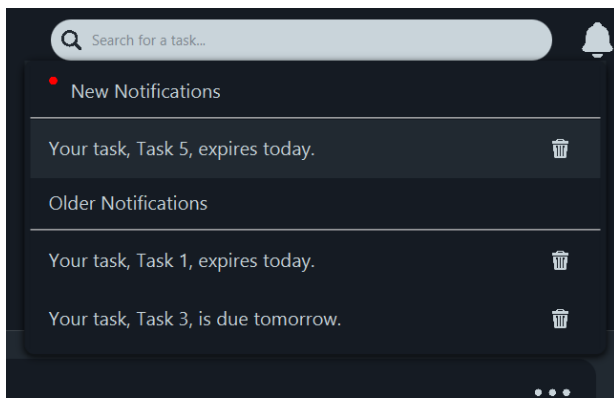
notifications.json: [{"task": String, "type": String, "notificationDate": String}, {...}]

Σε εισαγωγικά είναι η ονομασία του key ακολουθούμενο από τον τύπο του value που αναμένεται. Το {...} συμβολίζει την ενδεχόμενη ύπαρξη παρόμοιων αντικειμένων μέσα στην λίστα.

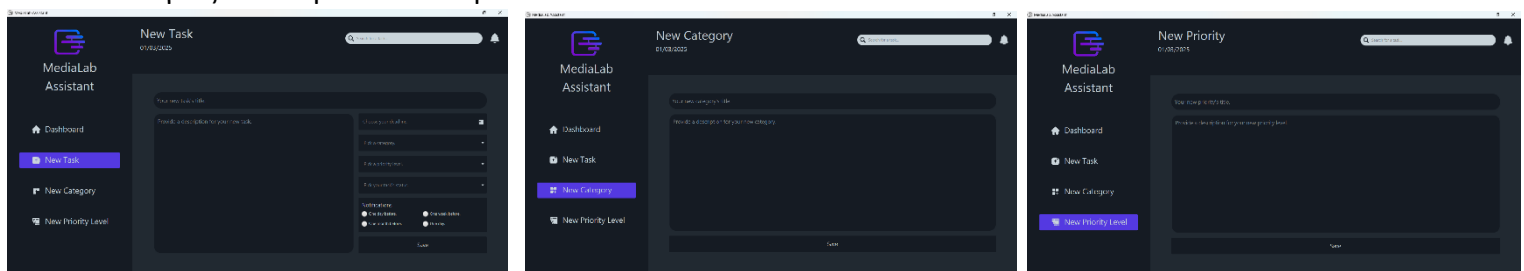
Κατά την διαδικασία της αποκρυπτογράφησης, πρώτα δημιουργούνται τα αντικείμενα των κατηγοριών και προτεραιοτήτων. Στην συνέχεια, ορίζονται οι εργασίες. Σε αυτό το σημείο, πρέπει να αναφέρω ότι έχω πάρει την παραδοχή, ο τίτλος μιας εργασίας, μιας κατηγορίας ή μιας προτεραιότητας λειτουργεί ως το id της. Δηλαδή δεν υφίστανται δύο αντικείμενα αυτών των κλάσεων με τον ίδιο τίτλο. Έτσι, στα πεδία “category” και “priority” του *tasks.json* καταγράφονται οι αντίστοιχοι τίτλοι, οπότε και γίνεται μια αναζήτηση στην λίστα των ήδη ορισμένων αντικειμένων για την εύρεση του κατάλληλου. Έτσι, εγγυάται η συνέπεια των αντικειμένων. Όσον αφορά το *notifications.json*, η ύπαρξη αυτού του αρχείου δικαιολογείται λόγω της απόφασης που πήρα να διατηρώ ιστορικό, το οποίο μάλιστα θα περιλαμβάνει διαβασμένες και αδιάβαστες ειδοποιήσεις. Συνεπώς, στο *tasks.json* περιέχονται οι ειδοποιήσεις που έχουν οριστεί για την εκάστοτε εργασία, στο *notifications.json* περιέχονται ειδοποιήσεις των οποίων η ημερομηνία εμφάνισης έχει παρέλθει και έχουν ιδωθεί από τον χρήστη και, στο τέλος της διαδικασίας αποκρυπτογράφησης, γίνεται μια σάρωση των επιλεγμένων ειδοποιήσεων όλων των εργασιών για τυχόν ειδοποιήσεις των οποίων η ημερομηνία εμφάνισης έχει παρέλθει αλλά που δεν έχουν ακόμη ιδωθεί από τον χρήστη.

Ανοίγοντας, τώρα, την διεπαφή, η εφαρμογή οδηγεί στον χρήστη στην κύρια οθόνη, το Dashboard, που περιλαμβάνει μεταξύ άλλων ένα search bar για την αναζήτηση εργασιών σύμφωνα με τις προδιαγραφές της εκφώνησης, ένα εικονίδιο καμπάνας για το άνοιγμα του ιστορικού ειδοποιήσεων, τέσσερα πλαίσια με πληροφορίες κατά τις προδιαγραφές της εκφώνησης, ένα μενού αριστερά για την περιήγηση σε διαφορετικές σελίδες/λειτουργίες και ένα tab pane για την προβολή των ορισμένων εργασιών, κατηγοριών και προτεραιοτήτων. Επίσης, σε περίπτωση ύπαρξης καθυστερημένων εργασιών εμφανίζεται το κατάλληλο popup screen κατά την έναρξη της εφαρμογής.





Όπως διακρίνεται και στις εικόνες ο χρήστης μπορεί να διαγράψει και να τροποποιήσει ειδοποιήσεις, εργασίες, κατηγορίες και προτεραιότητες. Για τις κατηγορίες πρόσθεσα και description που δεν χρειαζόταν, το ίδιο και στις προτεραιότητες -στις οποίες επιπρόσθετα πρόσθεσα και βαθμό προτεραιότητας. Τα μπλε πλαίσια είναι interactable, δηλαδή ο χρήστης μπορεί να τα πατήσει και στο tab των tasks να του εμφανιστούν μόνο οι καρτέλες των κατάλληλων εργασιών. Τέλος, αν ο χρήστης πατήσει σε κάποια καρτέλα ή, επιπλέον, στις ειδοποιήσεις και στις επιλογές του search για τις εργασίες, τότε οδηγείται στην κατάλληλη σελίδα τροποποίησης του εκάστοτε αντικειμένου. Παρακάτω, διακρίνονται οι διεπαφές δημιουργίας καινούριων αντικειμένων.



Πριν την δημιουργία ή την τροποποίηση κάποιου αντικειμένου γίνεται ο κατάλληλος έλεγχος σύμφωνα με τις παραδοχές μας και στις κατάλληλες περιπτώσεις εμφανίζεται το κατάλληλο popup μήνυμα σφάλματος.

Σε γενικές γραμμές υλοποιήθηκαν όλες οι προδιαγραφές της εκφώνησης και πάρθηκαν οι παραδοχές που αναφέρθηκαν προηγουμένως. Ο σχεδιασμός των διεπαφών έγινε με την βοήθεια του Scene Builder και για το styling έγινε χρήση css αρχείων. Επίσης, προσπάθησα να κάνω bind τα διάφορα στοιχεία του UI στο μέγεθος της οθόνης αλλά σε κάθε περίπτωση, μεγέθη οθονών διαφορετικών από την δικιά μου δεν είναι βέλτιστα.

Η κλάση που επέλεξα να τεκμηριώσω κατά τις προδιαγραφές του εργαλείου Javadoc είναι το Category.java στο πακέτο Models.