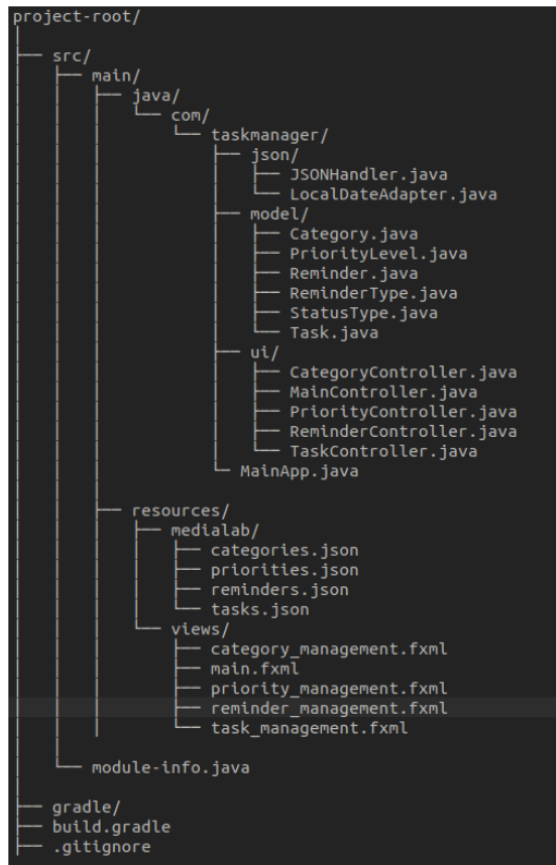


## Σχεδιασμός και Υλοποίηση

Η εφαρμογή βασίζεται στον αντικειμενοστραφή προγραμματισμό, όπου οι κύριες οντότητες (εργασίες, κατηγορίες, προτεραιότητες και υπενθυμίσεις) εκπροσωπούνται από αντίστοιχες κλάσεις. Η λογική της εφαρμογής είναι διαχωρισμένη σε controllers, οι οποίοι συνδέουν το backend με το GUI. Ταυτόχρονα υπάρχουν controllers που είναι υπεύθυνοι για την αποθήκευση των δεδομένων σε JSON αρχεία.



### Αντικείμενα:

Στην εφαρμογή, τα δεδομένα οργανώνονται μέσω αντικειμένων που αντιστοιχούν στις κλάσεις του συστήματος:

- Task (Εργασία): Κεντρική οντότητα της εφαρμογής
- Category (Κατηγορία): Αντιπροσωπεύει μια ομάδα εργασιών.
- PriorityLevel (Επίπεδο Προτεραιότητας): Περιγράφει τη σημαντικότητα μιας εργασίας.
- Reminder (Υπενθύμιση): Συνδέεται με μία εργασία και ενημερώνει τον χρήστη για σημαντικά γεγονότα σχετικά με την προθεσμία.

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. Συγκεκριμένα, ένα Task περιέχει ένα Category, ένα PriorityLevel και μια λίστα από Reminders

Κάθε αντικείμενο περιέχει τα χαρακτηριστικά και τις μεθόδους που είναι απαραίτητα για την λειτουργικότητά του, ενώ οι λίστες δεδομένων (π.χ., λίστα εργασιών, κατηγοριών) διαχειρίζονται συλλογές από αυτά τα αντικείμενα. Για την παρουσίαση αυτών των δεδομένων στη γραφική διεπαφή (UI), δημιουργούνται γραφικά στοιχεία ListViews, τα

οποία χρησιμοποιούνται απο τα fxml files.

### **Κλάση MainApp:**

Είναι υπεύθυνη για την εκκίνηση της εφαρμογής. Δουλειά της είναι

- Να φορτώσει τα δεδομένα απο τα JSON αρχεία μέσω της κλάσης JSONHandler, δημιουργώντας και τα απαραίτητα αντικείμενα και να αρχικοποιήσει τις λίστες που εμφανίζονται στο UI
- Να εμφανίσει το αρχικό view του GUI
- Να περάσει στον MainController όλα τα αντικείμενα
- Κατα το κλείσιμο της εφαρμογής να σώσει τις αλλαγές που έχει κάνει ο χρήστης στα JSON αρχεία και πάλι χρησιμοποιώντας την βοηθητική κλάση JSONHandler

### **Controllers**

Οι controllers στην εφαρμογή λειτουργούν ως ο συνδετικός κρίκος μεταξύ της λογικής του συστήματος (backend) και της γραφικής διεπαφής. Δουλειά τους είναι η διασφάλιση της επικοινωνίας και της αλληλεπίδρασης του χρήστη με τα δεδομένα

**MainController:** 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. Για παράδειγμα, όταν ο χρήστης διπλοπατήσει σε μια εργασία στη λίστα, ο Controller εκτελεί τη μέθοδο editTask, η οποία ανοίγει έναν διάλογο μέσω του κατάλληλου αρχείου FXML (task\_management.fxml) και διασυνδέεται με τον TaskController για να διαχειριστεί τις λεπτομέρειες της επεξεργασίας. Μόλις ολοκληρωθεί η διαδικασία, ο MainController λαμβάνει τα αποτελέσματα – είτε το νέο αντικείμενο της εργασίας, είτε αλλαγές στο υπάρχον – και ενημερώνει τις λίστες του UI. Παράλληλα, διασφαλίζει τον συγχρονισμό του UI με τις αλλαγές στα δεδομένα και την τήρηση της συνέπειας των αυτών (για παράδειγμα, αφαιρώντας υπενθυμίσεις για διαγραμμένες ή ολοκληρωμένες εργασίες). Με την ίδια λογική υλοποιείται και η προσθήκη νέων εργασιών, κατηγοριών, προτεραιοτήτων και υπενθυμίσεων. Ταυτόχρονα υλοποιεί και υποστηρικτικές λειτουργίες όπως η προβολή στατιστικών δεδομένων και η ανίχνευση εκπρόθεσμων εργασιών

### **Δευτερεύοντες Controllers (TaskController, CategoryController etc):**

Οι δευτερεύοντες controllers, εστιάζουν περισσότερο στη διαχείριση των μεμονωμένων αντικειμένων της εφαρμογής και εξυπηρετούν τη δημιουργία ή την τροποποίηση τους μέσω παραθύρων διαλόγου. Ο ρόλος τους είναι να λαμβάνουν δεδομένα από τον χρήστη, να επεξεργάζονται τις αλλαγές και να επιστρέφουν τα αποτελέσματα στον MainController

### **Γραφική Διεπαφή**

Η γραφική διεπαφή της εφαρμογής υλοποιήθηκε χρησιμοποιώντας το πλαίσιο JavaFX και τα αρχεία FXML, τα οποία σχεδιάστηκαν και επεξεργάστηκαν μέσω του εργαλείου Scene

Builder.

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