



ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΜΑΘΗΜΑ: ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΓΙΑ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ
ΕΞΑΜΗΝΟ ΜΑΘΗΜΑΤΟΣ: 6ο**

**ΔΙΔΑΣΚΟΝΤΕΣ: ΑΜΠΑΤΖΟΓΛΟΥ ΑΠΟΣΤΟΛΟΣ
ΚΑΡΑΚΑΣΙΔΗΣ ΑΛΕΞΑΝΔΡΟΣ**

ΦΟΙΤΗΤΕΣ:

- 1. ΜΟΣΧΟΥ ΑΘΑΝΑΣΙΟΣ (ics22020)**
- 2. ΧΑΤΖΗΑΝΤΩΝΗ ΑΝΔΡΕΑΣ (ics21138)**



Recycling Reward App

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων.....	2
1. Εισαγωγή.....	3
1.1 Σκοπός.....	3
1.2 Γενικές Λειτουργίες.....	3
2. Γενική Περιγραφή.....	4
2.1 Προοπτική της εφαρμογής.....	4
2.2 Περιορισμοί.....	4
2.3 Παραδοχές.....	5
3. Σχέδιο Δεδομένων.....	5
3.1 Βάσεις Δεδομένων.....	5
3.2 Αρχεία.....	5
4. Σχέδιο Μονάδων.....	6
4.1 Παρουσίαση κλάσεων.....	6
4.1.1 Frontend.....	6
4.1.2 Backend.....	8

Recycling Reward App - Links
Github: <https://github.com/CodeBrosLab/RCL-App>
YouTube: <https://youtu.be/KICXNI6E5c4>

1. Εισαγωγή

1.1 Σκοπός

Το παρόν έγγραφο έχει ως σκοπό τον σαφή προσδιορισμό του τρόπου λειτουργίας της εφαρμογής, η οποία υλοποιήθηκε στα πλαίσια της εργασίας αυτού του μαθήματος, με στόχο να εφαρμόσει μια από τις προτεραιότητες της Ευρωπαϊκής Ένωσης που είναι η βιωσιμότητα και η εξοικονόμηση πόρων. Επιτρέπει στους χρήστες να απορρίπτουν είδη προς ανακύκλωση και να λαμβάνουν ανταμοιβές μέσω συστήματος πόντων.

1.2 Γενικές Λειτουργίες

Οι Λειτουργίες της εφαρμογής είναι οι εξείς.

Απλός Χρήστης

R1: Μηχανισμός για login / εγγραφή.

R2: Προφίλ χρήστη με στατιστικά / progress bar μέχρι το επόμενο reward.

R3: Φόρμα για καταγραφή είδους υλικών που έχουν απορριφθεί στον κάδο.

Χρήστης Διαχειριστής:

R1: Έγκριση καταγραφής και αυτόματη απόδοση πόντων για reward.

R2: Προβολή κορυφαίων στατιστικών.

R(2++): Φόρμα παραμετροποίησης rewards βάσει υλικού / ποσότητας ανακύκλωσης.

2. Γενική Περιγραφή

2.1 Προοπτική της εφαρμογής

Το σύστημα που έχει αναπτυχθεί η συγκεκριμένη εφαρμογή δεν είναι αυτόνομο αλλά σχετίζεται με άλλα συστήματα λογισμικού. Πέρα από το υπό ανάπτυξη προϊόν που περιγράφηκε στην ενότητα 1.2, περιλαμβάνει,

- Την εφαρμογή για κινητές συσκευές: Κάθε χρήστης έχει στην κινητή του συσκευή κατεβασμένη την εφαρμογή η οποία του επιτρέπει να κάνει τις προαναφερθέντες ενέργειες για να ανακυκλώσει τα προϊόντα.
- Τον εξυπηρετητή (backend) μεταξύ εφαρμογής και βάσης δεδομένων υλοποιημένος με Spring Boot για χρήση API Calls.
- Την βάση δεδομένων με κάποιες προκαθορισμένες τιμές για σκοπούς επίδειξης.

2.2 Περιορισμοί

- Το Σύστημα προϋποθέτει την έκδοση Java 17 και άνω καθώς επίσης και την έκδοση του **Android Studio Flamingo | 2022.2.1** ή νεότερη.
- Το Σύστημα προϋποθέτει την χρήση Android Gradle Version 8.0 καθώς και Minimum SDK Version 24 (API 24: Android 7.0 Nougat).
- Το Σύστημα προϋποθέτει την χρήση ενός συστήματος σαν το XAMPP για την λειτουργία ενός Apache Server και μιας βάσης δεδομένων.
- Η δικτυακή διεπαφή της εφαρμογής στον διακομιστή γίνεται τοπικά, με την χρήση της διεύθυνσης IPv4. Το αρχείο όπου μπορεί να καθοριστεί για τον κάθε υπολογιστή βρίσκεται στο Frontend με όνομα Strings.xml και όνομα ιδιότητας IPv4.

2.3 Παραδοχές

- Κάθε χρήστης ταυτοποιείται μέσω ενός username και password.
- Για σκοπούς απλότητας δεν χρησιμοποιήθηκαν πολιτικές ασφαλείας σχετικά με την ταυτοποίηση του χρήστη.
- Ο Διαχειριστής της εφαρμογής είναι προκαθορισμένος από πριν και δεν μπορούν να δημιουργηθούν παραπάνω από έναν χωρίς προγραμματιστική παρέμβαση.

3. Σχέδιο Δεδομένων

3.1 Βάσεις Δεδομένων

Γίνεται χρήση της MySQL για την αποθήκευση των δεδομένων της εφαρμογής σε 4 πίνακες: **users**, **recycle_items**, **recycle_requests** και **request_list_items**. Η αλληλεπίδραση του Server με την βάση γίνεται μέσω **Jakarta Persistence API** (JPA), καθώς αυτή η τεχνολογία χρησιμοποιεί αντικείμενα κλάσεων για την επικοινωνία με την βάση δεδομένων. Πιο συγκεκριμένα, κάνει αντιστοίχιση κλάσεων και σχεσιακών πινάκων. Κάθε αντικείμενο μιας κλάσης ισοδυναμεί με μια γραμμή στον αντίστοιχο σχεσιακό πίνακα και κάθε ιδιότητα ενός αντικειμένου ισοδυναμεί με μια στήλη στον αντίστοιχο σχεσιακό πίνακα.

3.2 Αρχεία

Τα αρχεία που χρησιμοποιούνται και δημιουργούνται είναι τα εξής:

- Φάκελος **Backend** με τον υλοποιημένο κώδικα σε **Spring Boot**, τα αρχεία **schema.sql** και **data.sql** για την περιγραφή του σχήματος της βάσης δεδομένων και για την αρχικοποίηση της βάσης δεδομένων αντίστοιχα, ένα αρχείο **initialScriptForDB.sql** για την δημιουργία της βάσης στο XAMPP και ένα αρχείο **application.properties** που παρέχει τα στοιχεία για την σύνδεση στην βάση.
- Φάκελος **Frontend** με τον υλοποιημένο κώδικα σε Java για το Android Studio μαζί με όλα τα απαραίτητα αρχεία καθώς και εικόνες, Fonts κλπ.

4. Σχέδιο Μονάδων

4.1 Παρουσίαση κλάσεων

Οι κλάσεις του Frontend χωρίζονται σε 4 packages: **Activities**, **Adapters**, **HttpRequests** και **Model**. Οι κλάσεις του Backend χωρίζονται επίσης σε 4 packages: **Controllers**, **Model**, **Repositories** και **Services**.

4.1.1 Frontend

Φάκελος - *Activities*

- *LoginActivity*: Αρχική Οθόνη κατά την εκκίνηση της εφαρμογής, υπεύθυνη για την εισαγωγή των στοιχείων του χρήστη ή για την μετάβαση στην Οθόνη Εγγραφής σε περίπτωση που ο χρήστης είναι νέος χρήστης.
- *RegistrationActivity*: Οθόνη Εγγραφής υπεύθυνη για την εισαγωγή των στοιχείων του χρήστη.
- *AdminOpenRequestsActivity*: Οθόνη όπου ο Διαχειριστής βλέπει τα ανοιχτά αιτήματα και ανάλογα τα Εγκρίνει ή τα Απορρίπτει.
- *RecycleActivity*: Οθόνη υπεύθυνη για την επιλογή των ανακυκλώσιμων υλικών, της ποσότητας τους και αποστολής τους.
- *StatisticsActivity*: Οθόνη η οποία δείχνει τους 3 χρήστες με την ψηλότερη καταγραφή πόντων στο σύστημα.
- *UpdateRecycleItemPointsActivity*: Οθόνη στην οποία ο διαχειριστής μπορεί να τροποποιήσει τους πόντους επιβράβευσης για κάθε ανακυκλώσιμο αντικείμενο.
- *YourRewardsActivity*: Οθόνη Υπεύθυνη για την παρουσίαση των πόντων που έχει ο κάθε χρήστης αλλά και μιας μπάρας με την πρόοδο του χρήστη και πόσο απέχει από την επόμενη βαθμίδα.

Φάκελος - *HttpRequests*

Περιέχει όλες τις κλάσεις οι οποίες χειρίζονται τα δεδομένα από κάθε activity και ανάλογα τις ανάγκες καλούν τον διακομιστή με **GET** ή **POST** API Calls.

Φάκελος - *Model*

- *OpenRequestDetails* -
- *RecycleItem*
- *RecycleListItem*:

Οι κλάσεις αυτές έχουν σχεδιαστεί με βάση συγκεκριμένα σχεδιαστικά πρότυπα για την ενδοεπικοινωνία μεταξύ των άλλων κλάσεων, επιτρέποντας την ανταλλαγή δεδομένων και την αποθήκευση πληροφοριών για τα διάφορα αντικείμενα.

Φάκελος - *Adapters*

- *OpenRequestBodyListAdapter*
- *OpenRequestExpandableListViewAdapter*
- *RecycleItemsAdapter*
- *RequestListItemAdapter*:

Οι κλάσεις αυτές σχεδιάστηκαν για να γεφυρώσουν το χάσμα μεταξύ μιας Οθόνης/Activity και της κλάσης της και αφορούν στοιχεία του περιβάλλοντος εργασίας που εμφανίζει τα δεδομένα αυτά.

4.1.2 Backend

Φάκελος - Controllers

- *RequestsController*: Η κλάση η οποία είναι υπεύθυνη για τον χειρισμό όλων των requests προς τον server. Καλεί κάθε φορά το αντίστοιχο service για την εξυπηρέτηση του εκάστοτε αιτήματος.

Φάκελος - Model

- *LoginRequest*: Τα στοιχεία εισόδου (username, password) μετατρέπονται σε αντικείμενο αυτής της κλάσης προκειμένου να γίνει έλεγχος αν ο αντίστοιχος χρήστης υπάρχει στην βάση.
- *RecycleItem*: Αντιπροσωπεύει τα αντικείμενα τα οποία μπορούν να ανακυκλωθούν και ισοδυναμεί με τον σχεσιακό πίνακα recycle_items.
- *RecycleRequest*: Το αίτημα για απόδοση πόντων επιβράβευσης από τον χρήστη προς τον εξυπηρετητή, μετατρέπεται σε αντικείμενο αυτής της κλάσης προκειμένου να αποθηκευτεί στην βάση δεδομένων μέσω του αντίστοιχου repository. Ισοδυναμεί με τον πίνακα recycle_requests.
- *RecycleRequestDTO*: Είναι η αντίστοιχη Data Transfer Object κλάση της *RecycleRequest*. Χρησιμοποιείται για την εμφάνιση δεδομένων στο frontend, όταν ο διαχειριστής επιθυμεί να δει τα requests για τα οποία δεν έχουν αποδοθεί ακόμα πόντοι.
- *RecycleRequestListItem*: Αντιπροσωπεύει τα αντικείμενα που περιλαμβάνει στο αίτημα του ο εκάστοτε χρήστης, περιέχει ιδιότητες ονόματος προϊόντος προς ανακύκλωση και ανακυκλωθείσας ποσότητας και αποτελεί μέρος των αντικειμένων της κλάσης *RecycleRequest*. Ισοδυναμεί με τον σχεσιακό πίνακα request_list_items.
- *RecycleRequestListItemDTO*: Η αντίστοιχη Data Transfer Object κλάση για τα *RecycleRequestListItem* αντικείμενα. Είναι αρκετά παρόμοια με την προαναφερθείσα κλάση, ωστόσο παραλείπονται κάποια πεδία τα οποία δεν είναι αναγκαίο να εμφανίζονται στο frontend.
- *User*: Αντιστοιχεί στον πίνακα users της βάσης δεδομένων.
- *UserDTO*: Η αντίστοιχη Data Transfer Object κλάση για την *User*. Παραλείπονται εξίσου πεδία τα οποία δεν πρέπει να επιστρέφονται στο frontend προς εμφάνιση.

Φάκελος - *Repositories*

- *RecycleItemsRepository*: Χρησιμοποιείται απο το JPA προκειμένου να αποθηκεύει στην βάση δεδομένων αντικείμενα τύπου *RecycleItems*.
- *RecycleRequestListItemsRepository*: Το αντίστοιχο repository για τα *RecycleRequestListItem* αντικείμενα.
- *RecycleRequestsRepository*: Το αντίστοιχο repository για τα αντικείμενα *RecycleRequest*.
- *UsersRepository*: Το αντίστοιχο repository για τα αντικείμενα τύπου *User*.

Στο σημείο αυτό αξίζει να αναφερθεί ότι ο υποφάκελος *Repositories* δεν περιλαμβάνει κλάσεις αλλά interfaces. Τα interfaces αυτά επεκτείνουν το *JpaRepository* και δεν περιέχουν υπογραφές μεθόδων προς υλοποίηση. Επίσης το JPA υλοποιεί τις μεθόδους που απαιτούνται αυτόματα, συνεπώς δεν χρειάζεται να υλοποιηθούν οι μέθοδοι που κληρονομούνται μέσω του *extends*.

Φάκελος - *Services*

- *RecycleItemsService*: Περιέχει ένα *RecycleItemsRepository* αντικείμενο ώστε να επικοινωνεί με την βάση δεδομένων. Παρέχει μεθόδους προσβάσιμες από τον *RequestsController*, για τον χειρισμό των *RecycleItem* αντικειμένων και για την αποθήκευση/ανάγνωσή τους από την βάση δεδομένων.
- *RecycleRequestListItemsService*: Περιέχει ένα *RecycleRequestListItemsRepository* αντικείμενο και ο *RequestsController* καλεί μεθόδους του συγκεκριμένου service για τον χειρισμό των *RecycleRequestListItem* αντικειμένων.
- *RecycleRequestsService*: Το αντίστοιχο service για τον χειρισμό *RecycleRequest* αντικειμένων.
- *UserService*: Το αντίστοιχο service για τον χειρισμό των αντικειμένων τύπου *User*.

Τέλος να σημειωθεί ότι υπάρχει και μια κλάση ***BackendRclApplication***, η οποία αποτελεί το σημείο εκκίνησης του Server.