DOCUMENTAȚIA PROIECTULUI

To Do List App

STUDENȚI

Thira-Scheitzner Eduard-Paul

**Cuprins**

½ - 1 pagina despre prezentarea proiectului, ce isi propune, ce probleme rezolva, solutii

½ - 1 pagina despre tehnologiile folosite

½ - 1 pagina despre partea de “backend” - request-uri folosite

1-3 pagini despre arhitectura (componentele principale, navigare, baza de date etc.)

1-2 pagini despre cum poate fi utilizata aplicatia: tipuri de utilizatori, ce vede fiecare, autentificare etc

½ - 1 pagina concluzii

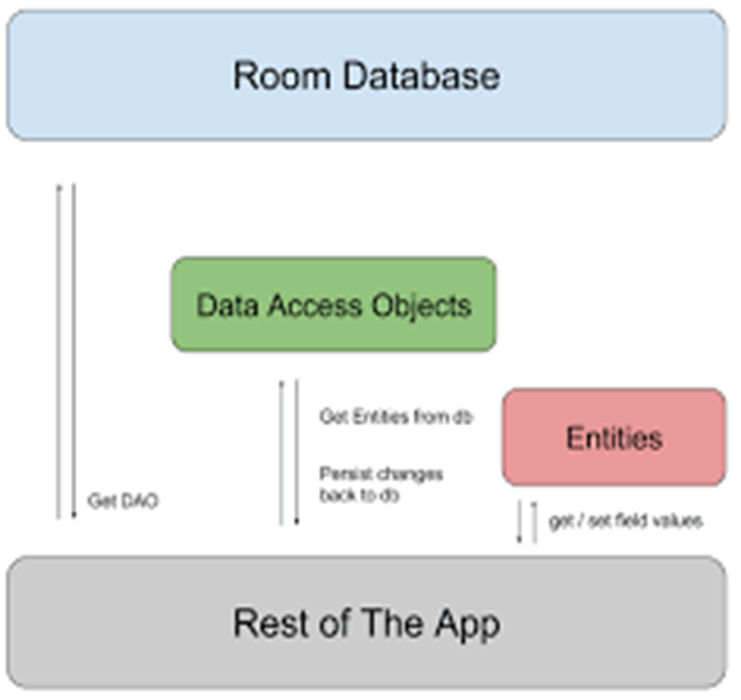
Prezentare Proiect

Proiectul acesta este o aplicatie de organizare si manageriere a obiectivelor si a activitatilor zilnice din viata cotidiana a fiecarui om.

Acesta ofera o modalitate inovativa de sectionare si organizare a task-urilor si obiceiurilor folosindu-se de tehnologi precum room database pentru stocarea datelor utilizatorului si http request-uri pentru crearea mai eficienta a diversor elemente. Scopul pricipalal proiectului este de a oferi o interfata colaborativa pentru organizarea de firme si proiecte in care fiecare utilizator contribuie la scopul final de finalizare al proiectului. De aceea in baza de date fiecare utilizator poate adauga task-uri in mod anonim si sa modifice task-uri pe care le pot vedea toti utilizatori la deschiderea aplicatiei.

Tehnologii folosite

Pentru construirea acestui proiect s-a folosit Room Database pentru inserarea si salvarea datelor utilizatorilor si pentru procesul de Inregistrare si logare a utilizatorilor pentru a evita utilizatori cu acelasi nume si adresa de email.



Pentru managerierea si crearea request-urilor de catre utilizator s a folosit platforma volley in combinatie cu formatul json pentru a eficientiza cererile mai mari de creere a categoriilor si a task-urilor.

Limbajul in care aceasta aplicatie a fost scris este in preponderenta Java pentru evitarea sintaxei greoaie a limbajelor low level precum C++ si a ne putea de bibleoteci predefinite din limbaj precum bibleoteca Stream care a ajutat la filtrarea efcienta a datelor fara a fi nevoie de creare de versiuni noi a bazei de date pe parcursul dezvoltari acestui proect. De asemenea Java a fost extrem de folositor in crearea de structuri precum recycler view, deoarece exista interfete care eficientizeaza codarea acestei structuri prin implementarea de functii automata.

Json-urile au fost utile deoarece au permis sablonizarea unor date cee ace a falicitat salvarea si iterarea lor pentru diverse sarcini precum requesturile descries anterior sau pentru modificarea de date fara a modifica datele salvate in baza de date.

Backend si requesturi folosite

In acest proiect s au folosit 2 request-uri de get all pentru a insera automat de date mai mari de catre utilizator fara a apasa butonl de add in mod repetat.

Primul request a fost folosit in partea TDL( To Do List ) a proiectului pentru a adauga 5 noi butoane in recycler view. Numele acestor 5 butoane este luat prin intermediul unei cereri de tip get care obtine json-ul il deserializeaz si itereaza prin vector pentru a obtine numele butoanelor.

Al doilea request a fost folosit in partea HabitTrackers a proiectului intr-o maniera similara descrisa mai sus, insa spre deosebire de cel mai sus vectorul din JSON este de doua ori mai lung, iar butoanele create pot accesa un HabitTracker

Mai sus se afla o poza cu json-ul care a treuit sa fie deserializat si inserat in recycler view

A picture containing text, screenshot

Description automatically generated

Arhitectura Aplicatiei

In figuara de mai jos se poate observa modul in care un posibil utilizator poate naviga prin aceasta aplicatie si vom descrie fiecare ferreastra posibila pe care o poate accesa

A screenshot of a computer

Description automatically generated

First Fragment ii ofera utilizatorului 2 optiuni din care sa aleaga LogIn si SignUp. Login il duce pe utilizator spre fereastra de logare, iar cel SignUp duce catre fereastrade inregistrare.

Fereastra LogIn permite utilizatorului sa acceseze functionalitatile aplicatiei de pe un cont vechi, iar cautatrea sa face in functie de numele utilizatorului si parola acestuia, daca perechea nume parola exista in baza de date atunci utilizatorul poate accesa aplicatia si e trimis catre main menu altfel utilizatorul ramane blocat in fereastra de LogIn.

Fereastra SignUp permite inregistrarea utilizatorului cu un cont nou. Acest lucru se face prin cautarea perechii nume adresa de email in baza de date in tabelul utilizator, daca se gaseste perechea nume email atunci utilizatorul ramane in aceasi fereastra, altfel utilizatorul este redirectionat catre fereastra MainMenu.

Fereastra MainMenu reprezinta ofera utilizatorului 3 butoane, 2 dintre care trimit utilizatorul catre o noua ferastra, iar clalat buton daca este apasat isi va schimba culoarea.

Daca utilizatorul va apasa pe butonul Habits acesta va fi redirectionat catre fereastra TDLs care continue un recyclerview si un TextView prin care se pot adauga noi butoane in baza de date, butoanele adaugate reprezint a TDL-uri goale .

Daca utilizatorul apasa pe butonul TDL acesta va fi redirectionat catre fereastra HabitTrackers care permite inserarea de HabitTrackers vide in baza de date.

Daca utilizatorul apasa pe butonul dates acesta isi va schimba culoarea atata timp cat este apasat.

Daca utilizatorul apasa pe oricare din butoanele din recycler view din fereastra HabitTrackers acesta va fi redirectionat catre un todolist in care utilizatorul poate sa insereze Task-uri.

Utilizarea aplicatiei

In aceasta aplicatie exista 2 tipuri de utilizatori utilizatori inregistrati si utilizatori neinregistrati. Utilzatori inregistrati trebuie sa se logheze in aplicatie pentru accesa functionalitatile comune. Logarea se face prin cautarea perechii nume parola in baza de date si in functie de asta se aleg alte actiuni.

Partea de SignUp se face prin cautarea perechii nume adresa de email si in functie de rezultatul obtinut se fac alte actiuni aferente care au fost descrise in arihitectura aplicatiei.

Utililizatori neinregistrati nu vor avea access la main menu si la celelalte ferstre care se pot accesa din main menu.

Utilizatori inregistrati au access la toate ferstrele si pot insera in baza de date TDL-uri HabitTrackers si Taskuri.

A screenshot of a computer

Description automatically generated

In figura de mai sus se poate observa procesul de autentificare descris anterior.

Concluzii

Aceasta aplicatie, desi are o idee buna la baza va avea nevoie de niste modificari majore pentru a putea fi uploadata pe Google Play. Principalele lacune ale aplicatiei acesteia este aprtea de UI care ar trebui remodelata de la 0 si implementarea toatala a bazei de date astfel incat fiecare utilizator sa isi poata adauga sterge si modifica task-urile si obiceiurile. De asemenea ar fi bine de implementat o functionalitate de stergere de cont si adaugarea de mai multe http requesturi pentru a eficientiza aplicatia