Sidorencu Oana

Grupa 236/2

**LABORATOR 5-8**

**GIT: https://github.com/SidorencuOanaAlexandra/Mobile-TravelApp**

**1. Daca vorbim de o aplicație gen cel de la licența: alegerea si motivarea limbajelor de programare, framework-urilor alese cu descrierea pe scurt inclusiv a alternativelor existente dar nealese.**

Pentru implementarea aplicatiei de mobile m-am folosit de Android Studio care este un mediu de dezvoltare integrat (IDE) pentru sisteme de operare Android de la Google, construit pe software-ul IntelliJ IDEA de la JetBrains si conceput special pentru dezvoltarea Android. Android Studio este un inlocuitor pentru Eclipse Android Development Tools (E-ADT) . Android Studio suporta limbaje de programare precum Java,C++, Go sau Kotlin.

Pentru aplicatia mea m-am folosit de kotlin ca si limbaj de programare. Am ales kotlin deoarece este un limbaj asemanator cu Java si a devenit limbajul preferat al celor de la Google in materie de aplicatii pentru mobile.

Exista mai multe framework-uri pentru dezvoltarea de aplicatii mobile precum: Unity folosit pentru dezvoltarea de jocuri, React Native, Flutter care permite construire de apliatii mobile atat pe android cat si pe iOS.

**2.Daca vorbim de o aplicație gen cel de la licența: compararea aplicației (cum se preconizează ca ar arata el in forma finala) cu aplicații similare existente deja. Descrierea acestor aplicații atât din punct de vedere a funcționalităților oferite cat si a tehnologiilor folosite in implementarea lor. Un plan de deployment – cum se distribuie aplicatia la clienti (cum se instaleaza, cum punem pe un server sa fie accesibil in internet)**

Dupa logare, o sa se deschida pagina principal in care apare o lista random cu locatii turistice.

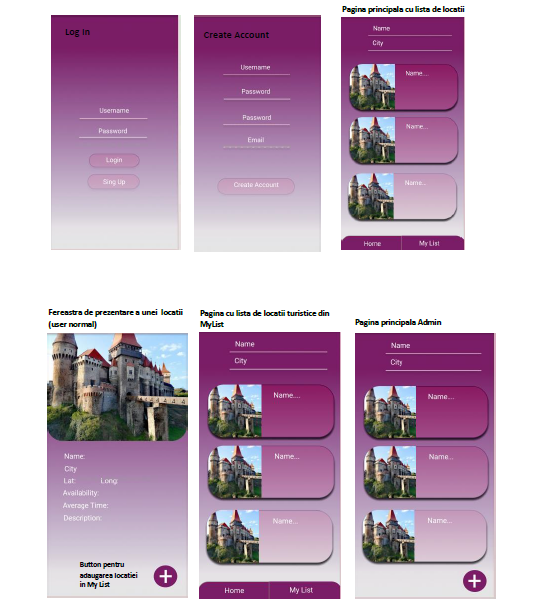
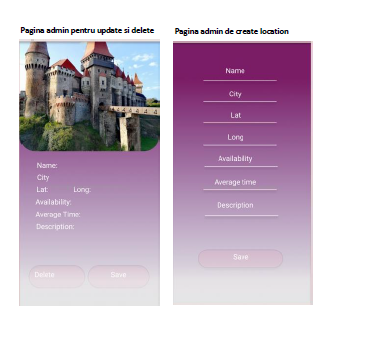
Utilizatorul poate introduce un oras astfel lista o sa se modifice cu locatiile turistice prezente in orasul selectat sau poate sa isi selecteze locatia curenta astfel lista o sa se modifice cu locatiile turistice aflate in apropiere de locatia curenta. La apasarea pe una dintre locatiile turistice din lista prezentata pe pagina principala, o sa se deschida o noua fereastra in care o sa apara detaliile locatiei respectice precum : nume, oras, tara, programul de inchidere si deschidere, descriere si timpul mediu de vizitare a locatiei respective. Utilizatorul isi poate adauga acea locatie intr-o lista de preferate prin apasarea unui buton. Pe pagina principala o sa fie un alt buton care o sa permita utlizatorului sa isi vizualizeze locatiile adaugate la favorite.

Pe pagina in care apar locatiile adaugate la favorite, utilizatorul poate selecta cateva dintre ele iar dupa apasarea unui buton o sa apara un fereastra in care acesta trebuie sa introduca un interval de timp ex: (14:00-18:00),ziua si locul din care vrea sa plece iar la apasarea butonului ok o sa se deschida o noua fereastra. In acesta fereastra o sa apara o harta cu un traseul care trebuie parcurs pentru a putea vizita locatiile selectate. Acest traseu este generat pe baza unui algoritm inteligent astfel incat distanta parcursa intre locatiile turistice sa fie minima si sa se tina cont de orele de inchidere si deschidere a fiecarei locatii turistice astfel incat utilizatorul sa le poata vizita pe toate. Acest traseu generat poate si el sa fie adaugat intr-o lista de trasee prin apasarea unui buton si introducerea unui nume pentru acel traseu. Daca locatiile turistice selectate de utilizator nu pot fi vizitate in perioada de timp aleasa, acesta o sa primeasca un mesaj de eroare cu motivul pentru care nu poate vizita anumite locuri turistice in perioada selectata.

Pe piata exista deja aplicatii destinate oamenilor care calatoresc precum Tripadvisor sau Polarsteps dar niciuna nu ofera o functionalitatea care genereaza un traseu eficient pentru a vizita mai multe locatii turistice.

Tripadvisor este una dintre cele mai cunoscute aplicatii de genul, aceasta ofera sugestii de locatii turistice, hoteluri sau locuri unde poti manca, pe baza orasului selectat sau pe baza locatiei curente iar Polarsteps permite doar generarea de trasee intre diferite locatii, dar aceste trasee generate nu tin cont de timpul de inchidere si deschidere sau durate medie de vizitare a locatiilor selectate.

Nu este forma finala!



**3.Daca vorbim de o aplicație gen cel de la licența: implementarea de funcționalități din aplicație (un subset de funcționalități), cod pentru testarea automata a funcționalităților (unittest, integration test), creare/folosire de source code repository (ex. git), de instrumente de continous integrasion (ex. jenkins), de instrumente gen Docker pentru mediu de development reproductibil.**

Momentan pentru aplicatie am implementat doar functionalitatile de crud (add,update,delete) pentru o locatie turistica si read.

Nu am implementat cod pentru testate!

**CRUD. Prezentati cate un caz de utilizare pentru fiecare operatie: - create. - read. - update. - delete.**

CREATE new location – dupa logarea unui utilizator cu drepturi de admin, acesta poate apasa pe butonul “(+)” care o sa il redirectioneze catre o pagina unde poate sa introduca datele pentru o noua locatie(name,city,lat,long,description,availability,average time). Dupa introducerea datelor apasa pe butonul “Save” si astfel se creeaza un nou LOCATION.

READ all locations – se afiseaza o lista cu toate LOCATION-urile pe pagina principala.

UPDATE location- adminul poate sa selecteze o locatie din lista de locatii afisate pe pagina principala. Acesta o sa fie redirectionat catre o alta pagina unde apar toate detaliile despre acea locatie. Fieldurile cu detaliile despre locatia selectata, se pot modifica de catre admin iar dupa apasarea butonului de “Save” aceste modificari o sa fie persistate in baza de date.

DELETE location – adminul poate sa selecteze o locatie din lista de locatii afisate pe pagina principala. Acesta o sa fie redirectionat catre o alta pagina unde apar toate detaliile despre acea locatie + 2 butoane de “Save” si “Delete”. Daca acesta apasa pe butonul de delete acea locatie o sa fie stearsa din baza de date.

**Prezentati detaliile legate de modul in care se vor persista datele, care din operatii se vor efectua atat local cat si pe server.**

Toate datele(locatiile si userii) o sa fie retinute pe server.

Operatiile care o sa se execute atat local cat si pe server o sa fie adaugarea si stergeare din lista de favorite a utilizatorilor si din lista de trasee a utilizatorului.

Operatiile care se executa pe server dar nu local o sa fie operatiile de crud (add,remove,update) realizate de utilizatorii cu drepturi de admin intrucat aceste modificari trebuiesc vazute imegiat de catre ceilalti utilizatori. De asemenea operatia de create account este tot o operatie care se executa pe server.

Operatia care se executa local este cea de log in.

**Ce se intampla cand dispozitilul nu are acces la server.**

Daca utilizatorii normali nu au acces la server acestia se pot loga si isi pot vizualiza doar lista cu locatiile turistice adaugate la favorite sau lista de trasee salvate. De asemenea acestia pot sterge locatiile din lista de favorite sau traseele din lista de trasee ulterior dupa conectarea la server o sa faca o verificare, daca exista diferente intre baza de date locala si cea de pe server atunci o sa se faca update la baza de date a serverului.

Operatiea de create account nici ea nu este disponibila daca userul nu este conectat la server.

Utilizatorii cu drepturi de admin daca nu se pot conecta la server acestia nu pot face nimic deoarece operatiile de crud nu sunt salvate local in baza de date ci direct in baza de date de pe server.