

Activity Center

Documentatie

Student: Grumazescu Silviu

Grupa: 30231

Cuprins

1. Introducere
 - 1.1 Abstract
 - 1.2 Functionalitati
2. Descrierea proiectului
 - 2.1 Hello world
 - 2.2 Descriere detaliata proiect
3. Detalii de implementare
 - 3.1 Securitate
 - 3.2 Baza de date
 - 3.3 Design pattern utilizat
 - 3.4 Harta locatiilor
4. Referinte

1. Introducere

1.1 Abstract

În acest proiect s-a dezvoltat o aplicație client-server folosind tehnicile de programare învățate la curs, partea de server fiind implementată în limbajul Java iar cea de interfață cu utilizatorul în JavaScript și framework-ul Vue. Proiectul numit “Activity Center” reprezintă o pagină web ce arată utilizatorului ce puncte de interes se află în jurul său. Aplicația este o unealtă foarte utilă pentru persoanele care vizitează un oraș nou și doresc să afle ce activități se desfășoară în jur, de la restaurante până la cluburi și hoteluri.

1.2 Funcționalități

Proiectul are implementat un sistem de înregistrare și logare. Utilizatorul va fi capabil să vizualizeze toate punctele de interes din jurul său, să le filtreze după anumiți parametri, să le vizualizeze locația pe o hartă și să își programeze în calendar o dată și o oră când vrea să participe la respectivele activități.

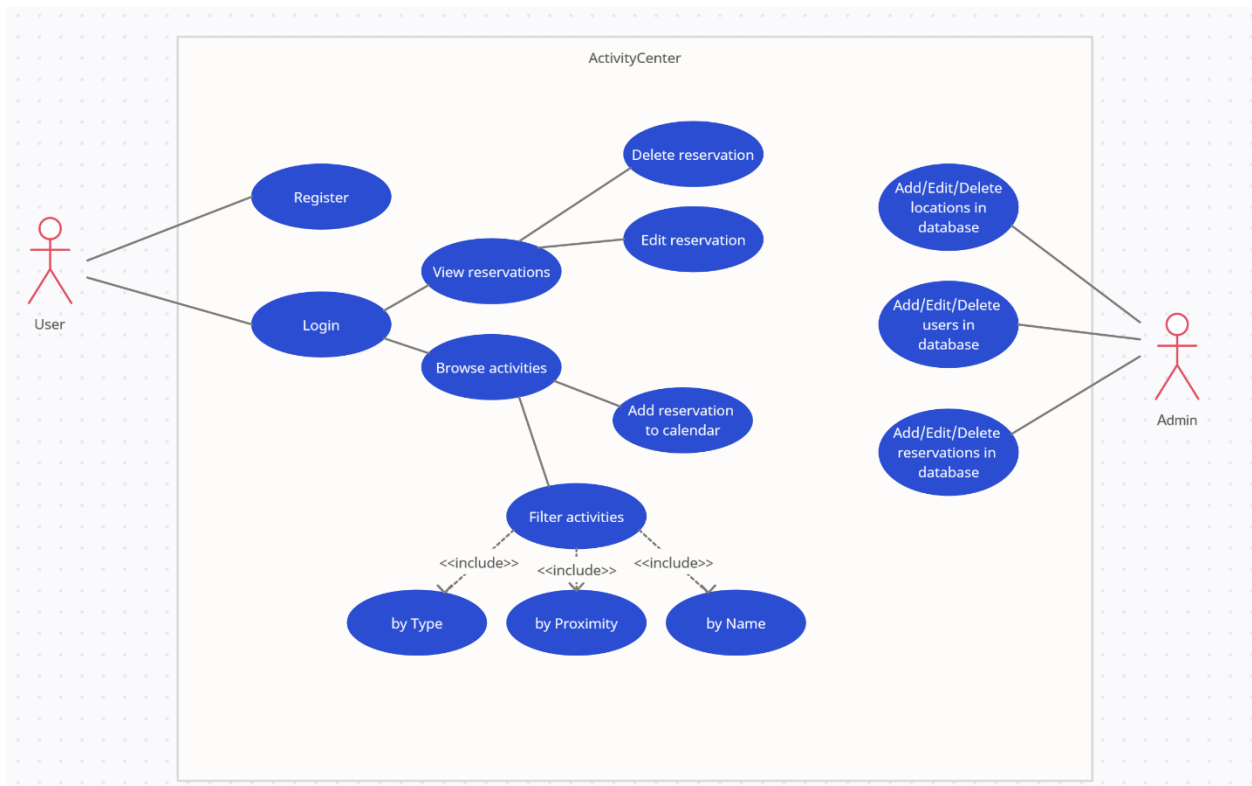
2. Descrierea proiectului

2.1 Hello world

Structura de bază a proiectului, respectiv proiectul Hello World este compus din două părți:

- Partea de server a fost dezvoltată în limbajul Java folosindu-ne de framework-ul Spring pentru server. Framework-ul Spring ne permite să cream endpoint-uri pentru a primi request-uri HTTP de la un browser. Pentru aplicația Hello World am creat un endpoint simplu la URL-ul “/getHelloWorld” care returnează în body un String cu textul “Hello World”
- Partea de interfață este realizată în JavaScript și framework-ul Vue. Framework-ul Vue poate injecta dinamic într-o pagină HTML date programate de noi, în cazul nostru dorim să afișăm textul primit de la server. Pentru aceasta vom trimite un request HTTP de GET la server folosind AXIOS, și vom afișa mesajul într-un <div>.

2.2 Descriere detaliata proiect



In imaginea de mai sus se poate observa diagrama de use-case.

Utilizatorul este intampinat de o pagina Home, iar folosind butoanele din partea superioara a paginii, acesta se poate indrepta fie spre pagina de login, fie spre cea de inregistrare.

Field-urile din form-ul de inregistrare sunt verificate in cazul introducerii de date invalide, cum ar fi verificarea formatului mail-ului sau daca sunt campuri goale.

Odata logat, utilizatorul are acces la pagina de Explore. Pagina ii permite sa vizualizeze toate punctele de interes din jur, si sa le filtreze in 3 moduri:

- Dupa nume: se introduce in caseta text un keyword dupa care se filtreaza numele activitatilor
- Dupa tip: se alege unul dintre tipurile: Food, Hotel, Entertainment
- Dupa proximitate: se introduce in caseta text distanta pana la care se vor filtra activitatile in km.

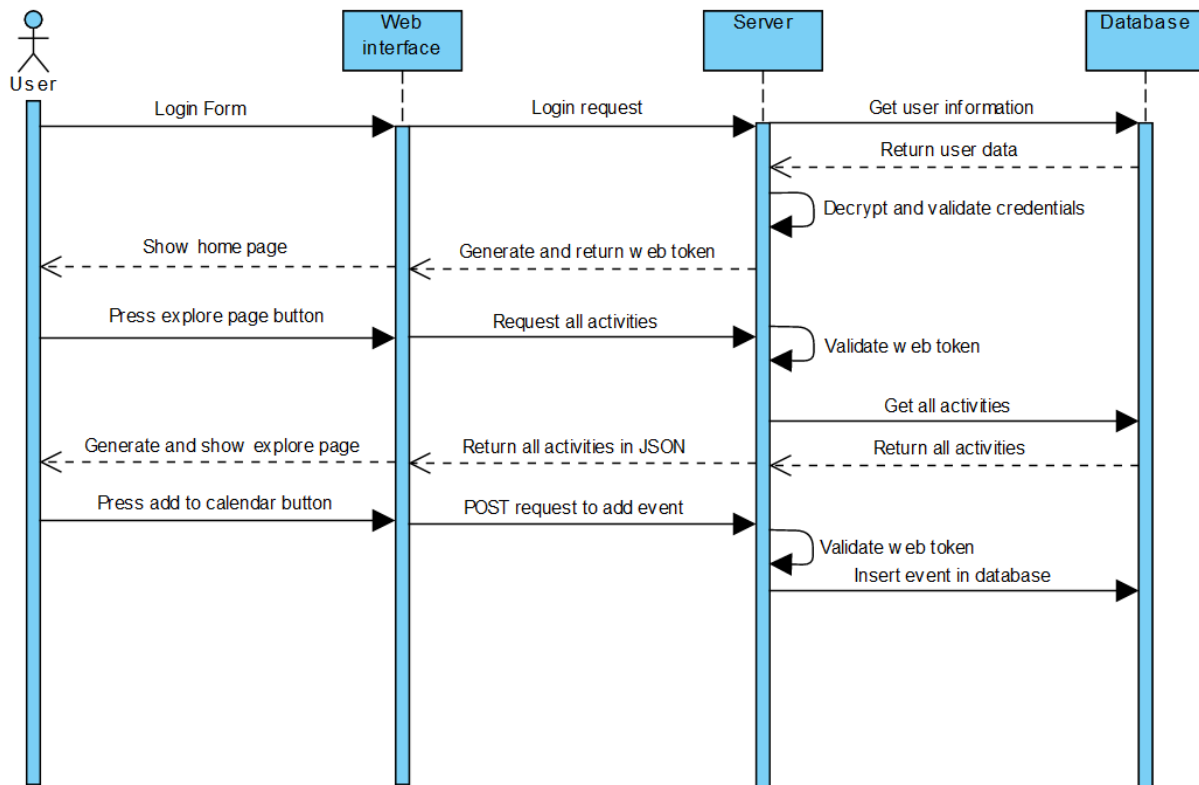
Pagina Explore dispune de o harta in care vor fi automat adaugate semne la locatia fiecărei activitati afisate.

Butonul „Show on map” din dreptul activitatii ne va focaliza harta la punctul respectiv de interes.

Prin apasarea butonului „Add to calendar” din dreptul unei activitati, utilizatorul introduce in baza de date o activitate pe care o poate vedea salvata pe pagina My Events.

In ultimul rand butonul Gallery ne va deschide o galerie de poze a activitatii selectate.

O diagrama de secventa pentru operatia de inserare a unui eveniment in calendar se poate vizualiza mai jos:



3. Detalii de implementare

3.1 Securitate

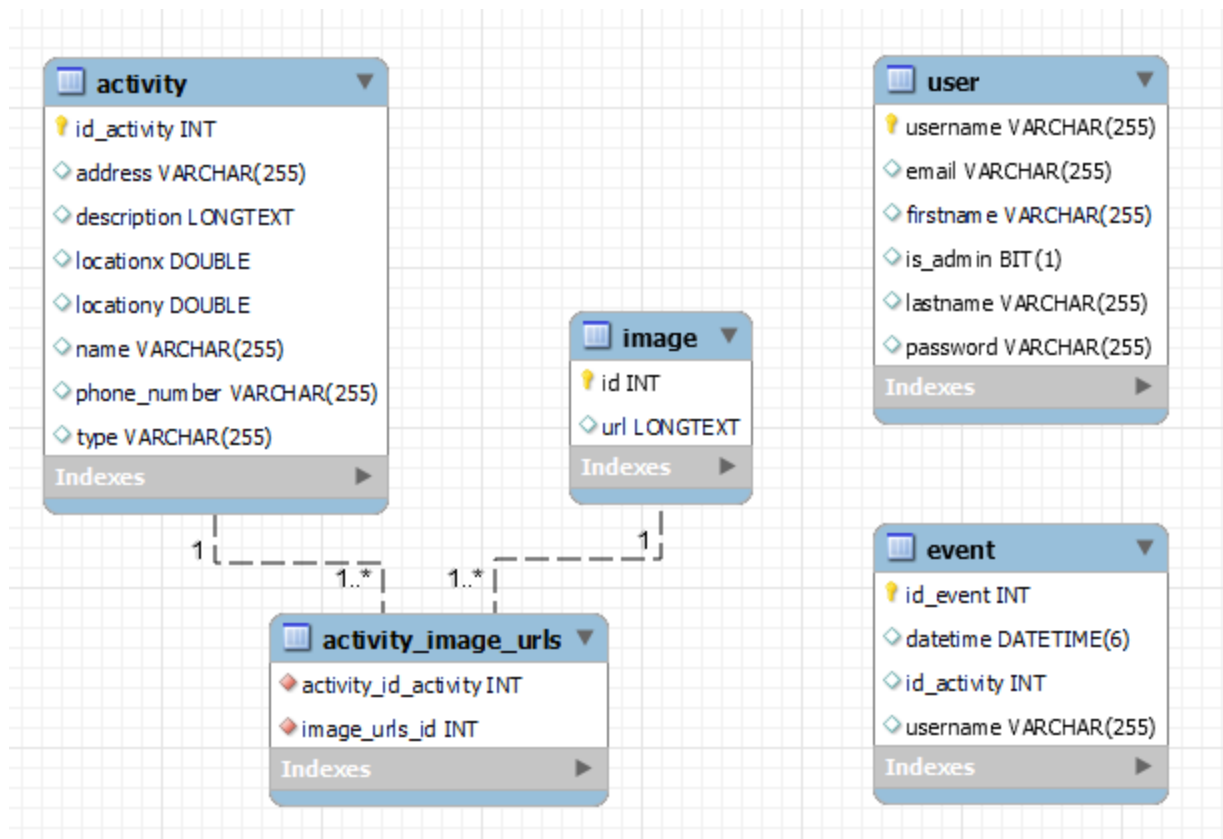
Sistemul de logare se foloseste de uneltele oferite de Spring pentru securitate. Astfel, in momentul logarii, server-ul va returna un webtoken ce va fi folosit la fiecare request, iar in absenta acestuia, un utilizator nelogat nu va putea accesa informatii din baza de date.

In aditie, parola utilizatorului este criptata in baza de date folosindu-se de un algoritm de hash-urare.

3.2 Baza de date

Baza de date este creata utilizand MySQL si operatiile CRUD sunt realizate prin intermediul framework-ului Spring respectiv interfetelor JpaRepository. Spring ne genereaza automat baza de date in functie de clasele definite de noi cu anotatia @Entity.

Schema bazei de date se poate vizualiza mai jos:



3.3 Design pattern utilizat

Design pattern-ul utilizat cel mai predominant in acest proiect este Singleton. Acesta consta in crearea unei clase ce are o singura instanta in proiect.

In cazul proiectului nostru, acesta este utilizat in cadrul comunicarii cu baza de date. Framework-ul Spring ne permite prin intermediul anotatiei `@Autowired` sa creem un astfel de obiect singleton fara al instanta in constructor, clasele Controller din aplicatia server avand cate un obiect Singleton pentru fiecare tabela din baza de date.

3.4 Harta locatiilor

Harta locatiilor este implementata folosind API-ul de la Google. Acesta ne permite sa afisam pe pagina noastra web o harta reala prin care se poate naviga, dar si sa inseram marker-e la coordonate specifice.

Pentru a putea filtra elementele dupa proximitate, stocam in baza de date coordonatele punctului respectiv de interes, si aplicam un algoritm de calcul al distantei fata de pozitia curenta.

4. Referinte

- [1] Singleton pattern - https://www.tutorialspoint.com/design_pattern/singleton_pattern.htm
- [2] Documentatie Vue - <https://vuejs.org/guide/introduction.html>
- [3] Documentatie Spring - <https://docs.spring.io/spring-framework/docs/current/reference/html/>
- [4] Documentatie bootstrap - <https://getbootstrap.com/docs/4.1/getting-started/introduction/>