Vacation Booking System

Raita Anamaria

Gr 30431

Assignment 1

Diagrama bazei de date reprezinta relatiile pe care le-am creeat intre obiecte, si anume:

-intre user si vacation package avem o relatie de many to many, deoarece un user poate rezerva mai multe vacante si o vacanta poate fi rezervata de mai multi useri. Astfel, am generat tabelul intermediar user\_vac care tine indecsi pentru a reprezenta aceasta relatie.

-intre vacation destination si vacation package este o relatie de one to many, deoarece o destinatie poate continue mai multe pachete, insa un pachet apartine unei singure destinatii. De asemenea, aici am impus restrictia “On delete/update Cascade”, pentru ca atunci cand se sterge o destinatie vrem sa stergem si pachetele acelei destinatii.

Diagram

Description automatically generated

Diagrama claselor reprezinta incercarea de a le aseza conform arhitecturii Layer. Pachetul de entities continue clasele User, VacationDestination si VacationPackage, corespondente tabelelor din baza de date,si respective Layerului de Database.

Query-urile aferente au fost create in layerul de Persistence, unde sunt prezente clase a caror denumire contine Repository. In principal sunt query-uri de select all, select all where, update, delete sau simpla operatie de persistence.

Folosirea functiilor din clasele de repository se petrece in layerul de Business, adica in clasele a caror nume continue Service. Acolo exista si diferite validari ale datelor, pentru campul unique, pentru field-uri goale, pentru situatia in care query-ul returneaza o lista goala din baza de date.

Iar in clasele de controller sunt apelate diferite clase de service, in functie de operatiile pe care doresc sa le fac in fiecare dintre ferestre. De asemenea, aici exista si anumite prelucrari de date, de exemplu dintr-o data de tip string o transform in tipul date si o trimit catre service deja gata transformata.

Pentru filtrari, am folosit functia de select all pentru a retrage toate entry-urile din baza de date si m-am folosit de lambda streams pentru a filtra deoarece era mai usor de implementat.

De asemenea, o dificultate a fost la relatia de many to many, deoarece trebuia sa adaug atat userul cat si vacation package in lista aferenta a celuilalt si apoi sa fac merge.

De asemenea, a fost mai greu de gestionat lucrurl cu campurile Date deoarece validarea si parsarea stringului de input a fost mai dificila.

A picture containing graphical user interface

Description automatically generated

Diagrama de Use Case reprezinta Requirementurile Functionale pe care le-am primit pentru proiect condensate intr-o schema usor de inteles. Datorita modului in care am implementat, am adaugat functionalitatea de View All Destinations, deoarece atunci cand introducem un Vacation Package este nevoie de un Id.

Diagram

Description automatically generated