Technical University of Cluj-Napoca

Traveling Agency

Assignment 1



Student: Domokos Robert-Stefan

Group: 30431

Content

1. Objective
2. Analysis
3. Design and implementation
4. Objectives

The main objective of this assignment is to implement and design an application for a travelling agency which lets a customer or a user to book a package for his next vacation. This package has a name, destination, status, number of days/period, price, and extra details.

This application also lets the administrator from the traveling agency to manage the destinations and the packages present in the database.

Some secondary objectives are:

* To implement and manage the database using hibernate.
* To implement a Log in graphical interface
* To use layered architectures in the design of the application

1. Analysis

As already mentioned, the main scenarios and use case are represented by what the administrator can do to the database and what action the user is allowed to do.

Use case diagram:

Diagram

Description automatically generated

1. Design and implementation

* Database Diagram:

Diagram

Description automatically generated

The database was made mainly using hibernate and all the packages and destination were added from the administrator GUI.

The classes were implemented and created by using a layered architecture:

* Database layer: represented by the database itself and the model of it

Graphical user interface, text, application

Description automatically generated

* Business layer: represented by the service package

Text

Description automatically generated

* Presentation layer: represented by the controller package and presentation package (GUI’s of the program)

Text

Description automatically generated Text

Description automatically generated

* Persistence layer: represented by the repositories which are mainly composed by the main methods and querries that manage the data from the model

Text

Description automatically generated

Class design:

A screenshot of a computer

Description automatically generated with medium confidence

Model classes + fields:

Graphical user interface

Description automatically generated

Repository classes + fields and methods:

Graphical user interface, text

Description automatically generated

Service classes + fields and methods:

Text

Description automatically generated

Controller classes +fields and methods:

Graphical user interface, text

Description automatically generated