Travelling Agency

Assignment documentation

Student(s):

* Nadu Laura

**Group: 30431**

**Contents**

[I Project specification 3](#_Toc98890158)

[1.1 Package Diagram 3](#_Toc98890159)

[II Use-Case model 4](#_Toc98890160)

[2.1 Users and stakeholders 4](#_Toc98890161)

[2.2 Use-Case identification 4](#_Toc98890162)

[2.3 UML Use-Case diagram 6](#_Toc98890163)

[III Architectural design 6](#_Toc98890164)

[3.1 Package diagram 7](#_Toc98890165)

[3.2 Class diagram 8](#_Toc98890166)

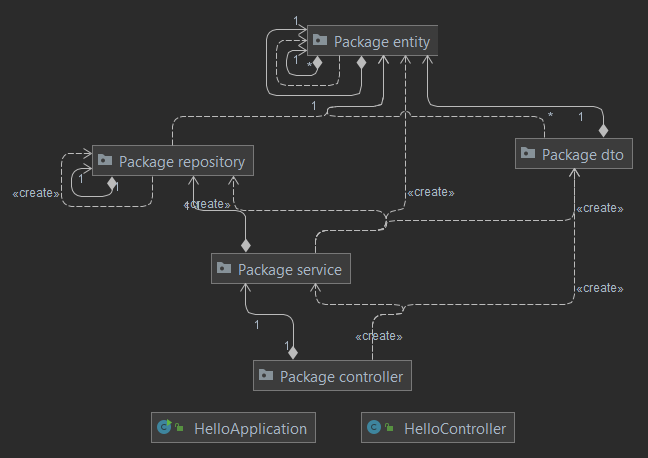
[3.4 Database (E-R/Data model) diagram 9](#_Toc98890167)

# I Project specification

This project aims to simulate an application for a travelling agency, which can have 2 users : REGULAR USER, TRAVELLING AGENCY. The application will present a login interface in which the users can login and, the regular users could also register new accounts.

The regular user could see available vacation packages and make reservations, while the travelling agency could manage these kind of packages.

## 1.1 Package Diagram



# II Use-Case model

The Regular User will be able to view all vacation packages and then book one from the list shown. He will also be able to view his booked vacation packages. On top of that, the Customer will have the option to filter the vacations by available destinations, start date and price above a certain threshold.

The Travelling Agency Admin will be able to perform CRUD operations on Vacation Packages and Destinations

## 2.1 Users and stakeholders

Users

* Travelling Agency
* Regular User

Stakeholders

* Development team
* Quality Assurance
* Product Owner
* Project Manager
* Booking

## 2.2 Use-Case identification

**Name Use case** – Reserve Vacation

**Level** : User Goal

**Main actor** : Regular User

**Main success scenario** : A user chooses a vacation package from the Available Packages tab and then he presses the “Reserve Vacation Package” button. If the vacation package is not fully booked, then the user will see the booked vacation in the tab named “Reserved vacation packages”.

**Extension** : The fully booked packages will not be available for booking.

**Name Use case** – Filter Vacation Packages

**Level** : User Goal

**Main actor** : Regular User

**Main success scenario** : The User can pick a filter type ( destination / price / period ) and then based on the chosen filter, the User will be able to choose the wanted destination, or a price and the filter will show the vacation packages with higher price than the filter selected.

**Extension** : If no vacation packages match the filters, an empty table will be shown. The filters work both on the reserved and available vacation packages tabs.

**Name Use case** – Filter Vacation Packages by Status

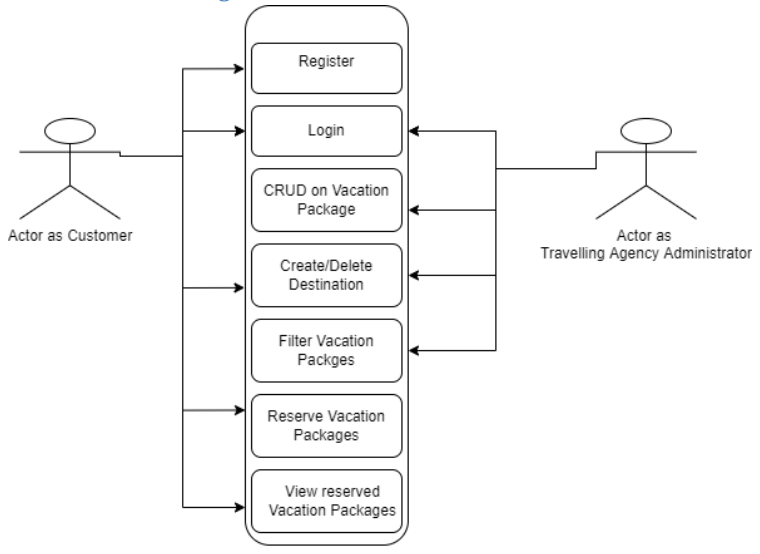
**Level** : User Goal

**Main actor** : Travelling Agency

**Main success scenario** : The User will be able to filter by the status of the vacation package which can be BOOKED, NOT\_BOOKED or IN\_PROGRESS. No other types ca be chosen.

**Extension** : If no vacation packages match the filters, an empty table will be shown.

## 2.3 UML Use-Case diagram



# III Architectural design

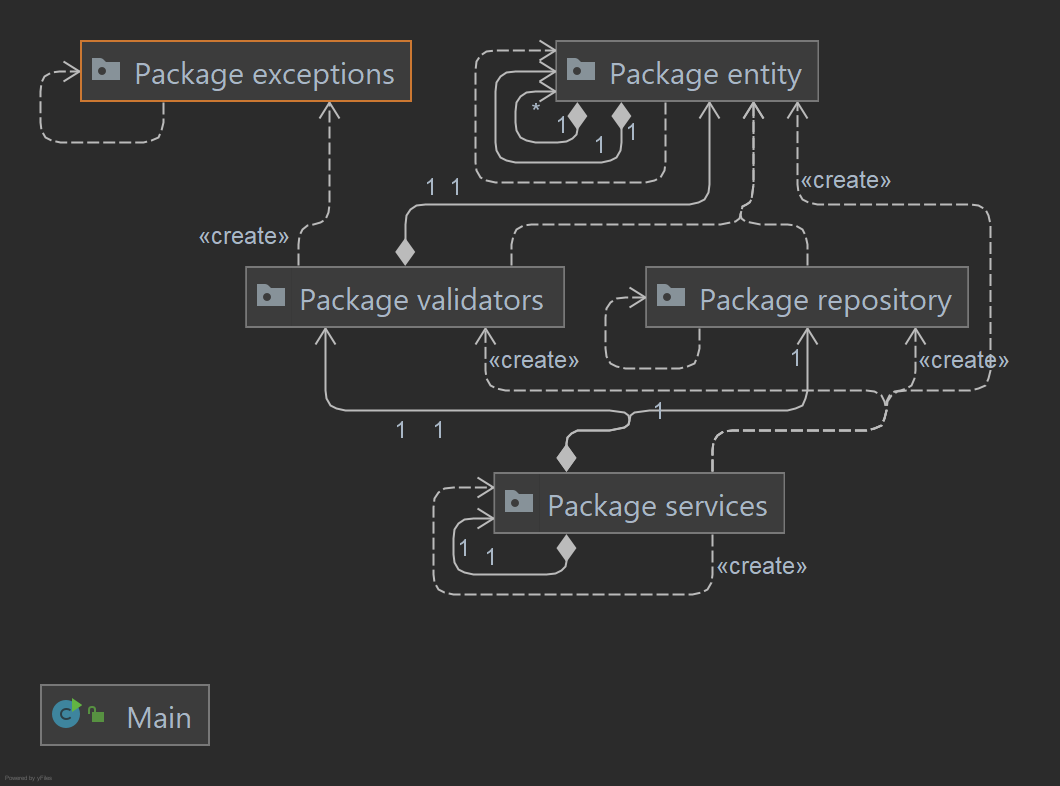
The architecture used is the **Layered Architecture**. It is based on 3 layers, which are:

→ **Persistence Layer** – which contains the ORMs and the connection to the database, such as Repositories. This Layer is represented by the repository and entity packages

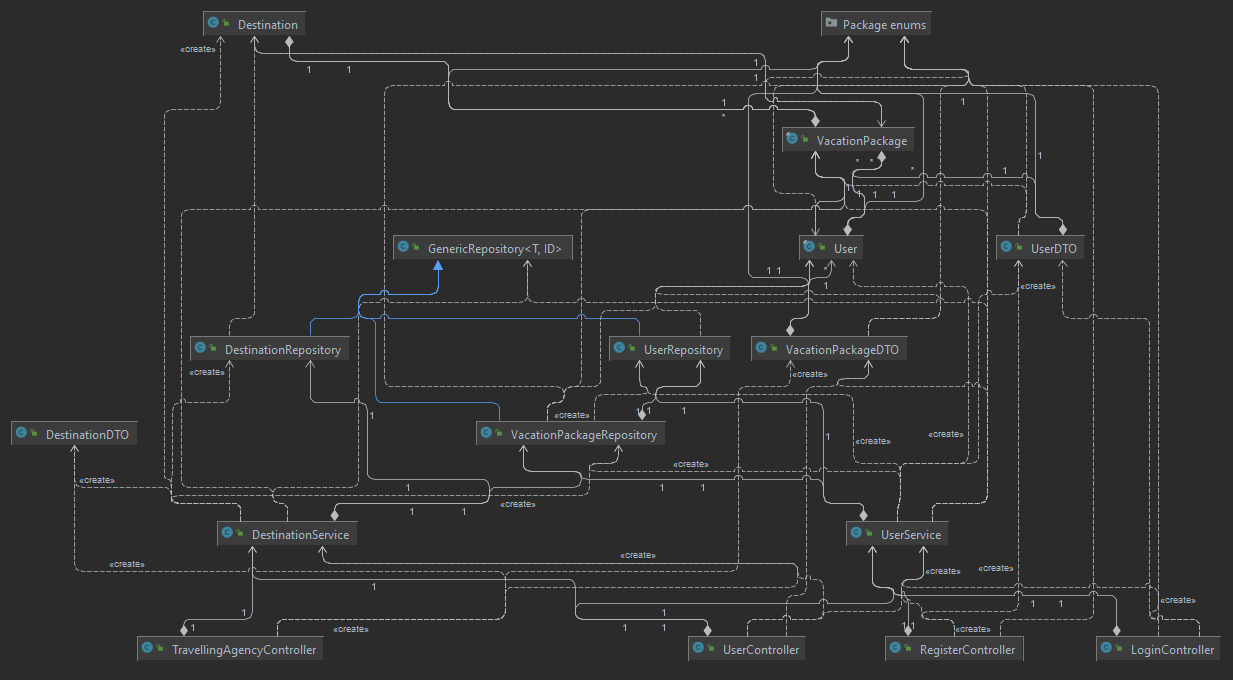
→ **Bussiness Layer** – which contains the services and all the logic of the application, resided in the service and dto packages.

→ **Presentation Layer** – which contains the controllers and the GUI actions, resided in the controller package and the external .fxml resources.

## 3.1 Package diagram

**

## 3.2 Class diagram



## 3.4 Database (E-R/Data model) diagram

