< Application name >	Version: 1.0
Documentation template	Date: <zz/ll/aaaa>

<Travelling Agency>

Project documentation

Student(s):

- S2

Group: 30431

MINISTRY OF EDUCATION

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Contents

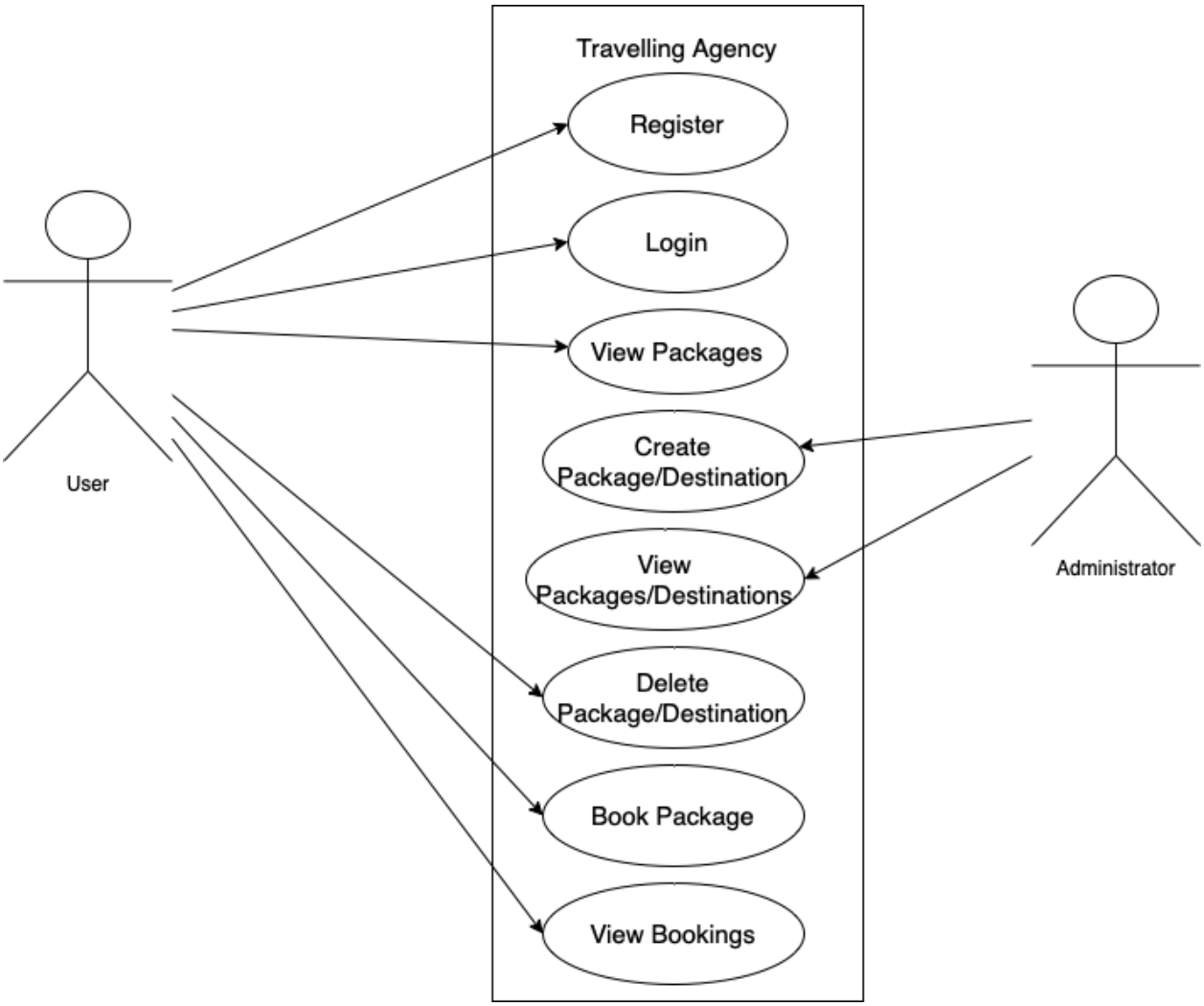
I Project specification

This project involves the development of an application which helps in the management of vacations from a travelling agency. There will be two types of users: the agency, which can create, edit, delete the vacation packages and destinations available, and the regular users, which can register, log in, then are able to view all available packages, and make bookings, as long as the selected package is not fully booked yet. Afterwards, the user can s]check their list of booked packages.

II Use-Case model

2.3 UML Use-Case diagram

Here we can see the Use-Case diagram, which displays all of the application's functionalities, as mentioned before.



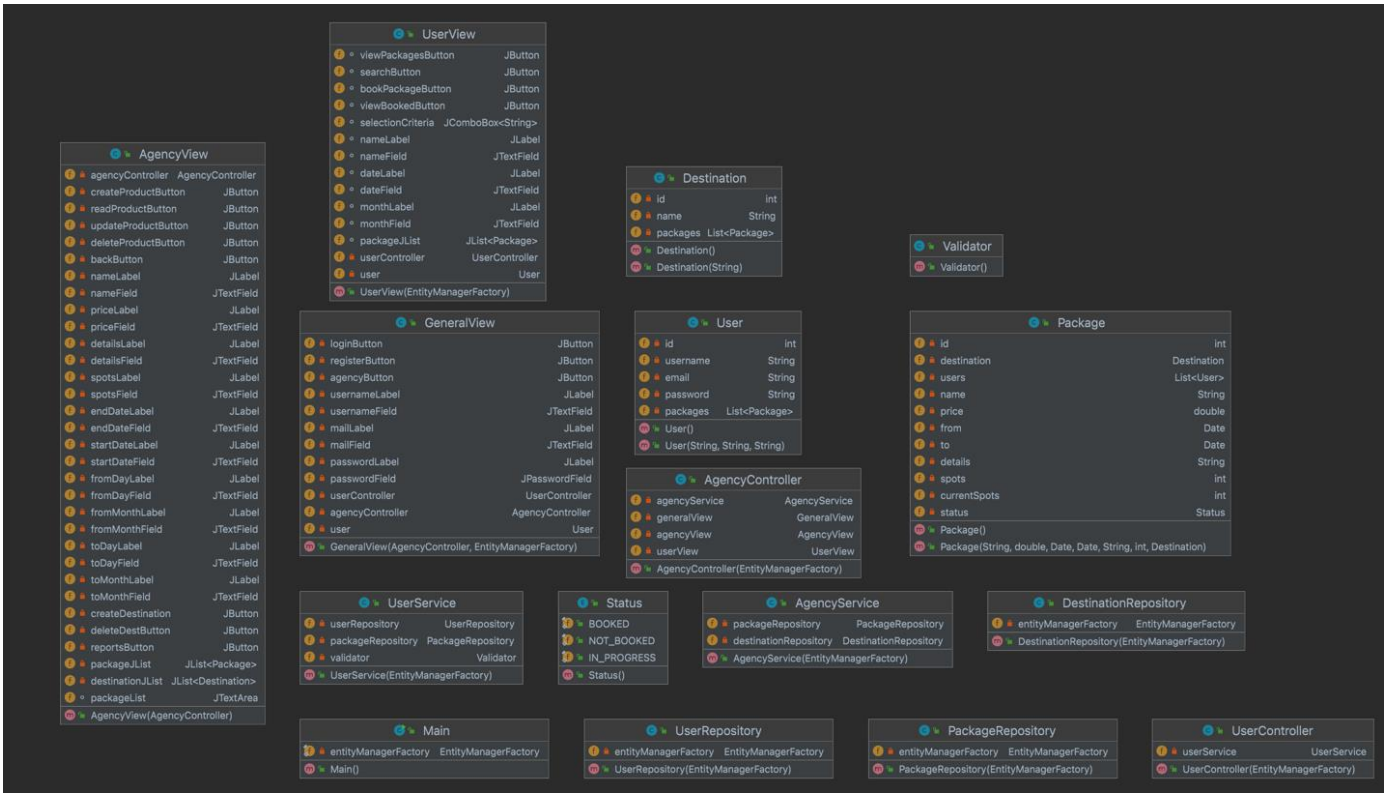
III Architectural design

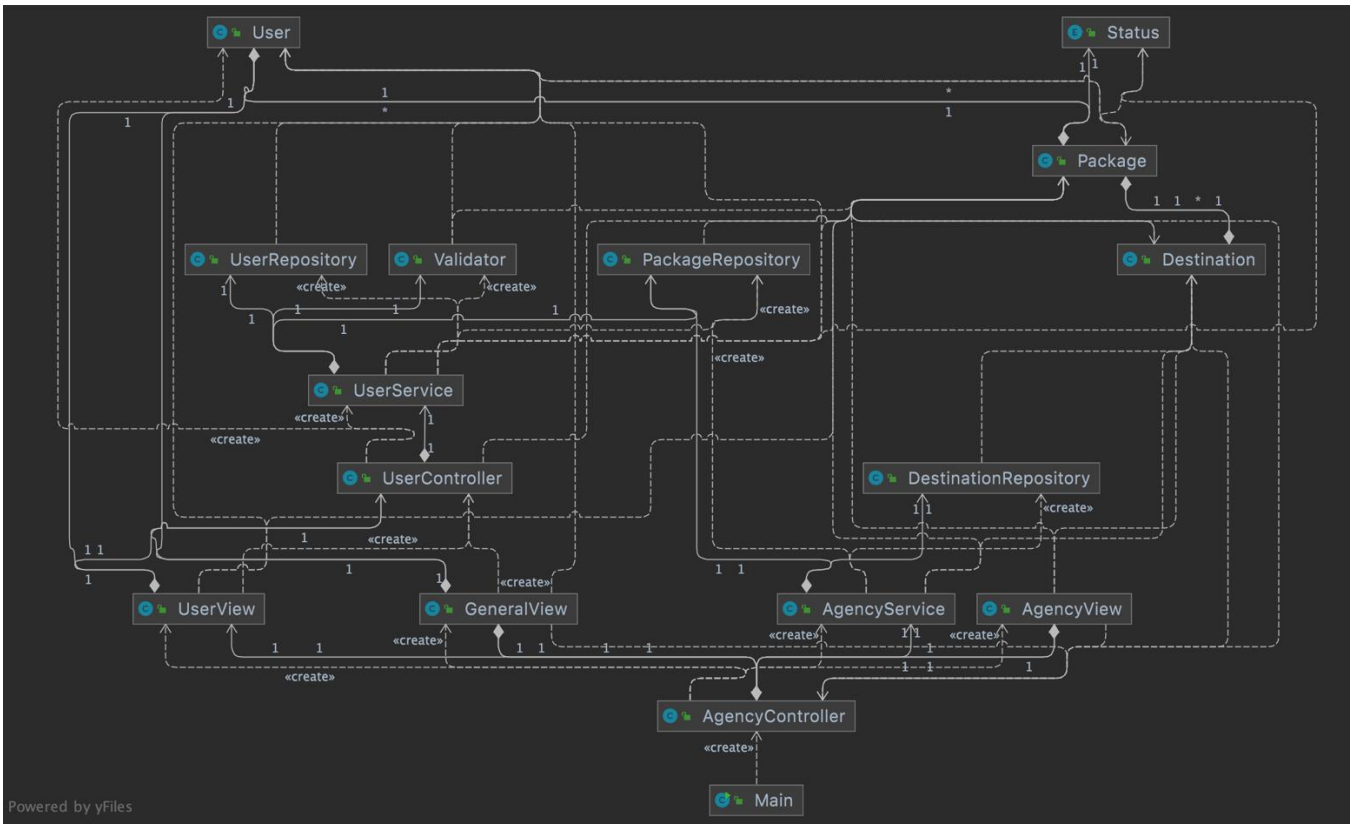
This project has been created according to the Layered architecture, which we will notice from the following diagrams, which display how the classes interact, as well as their belonging to the various layers of the architecture.

3.2 Package diagram

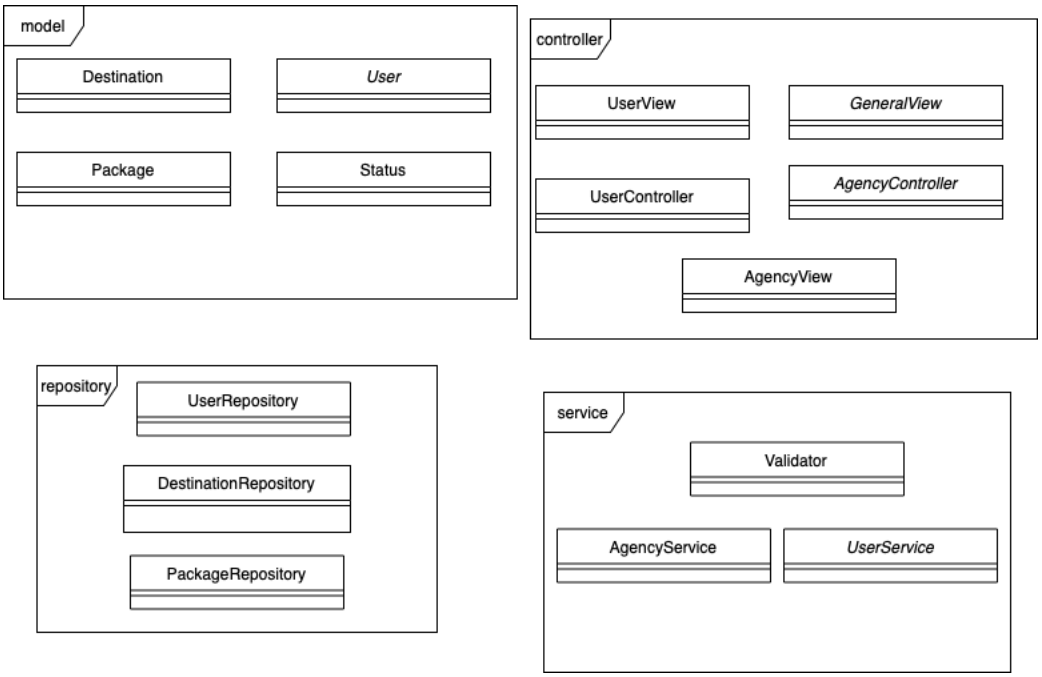
3.3 Class diagram

Here we can observe the class diagram of this project:





The last diagram displays how the classes are places within the packages:



3.4 Database (E-R/Data model) diagram

The database diagram is the following:

IV Supplementary specifications

4.1 Non-functional requirements

< Specificatiile non-functionale ale aplicatiei. Se va discuta la laborator./>

4.2 Design constraints

V Testing

5.1 Testing methods/frameworks

All functionalities have been tested successfully, including some intentionally wrong actions, in order to assess the safety of the application. As an example, the agency is unable to create a package starting or ending on a date before the current date. Likewise, the user is unable to log into

5.2 Future improvements

Future improvements include creating a better looking, more user-friendly interface, as the current one is functional, but not as much esthetically pleasing. Besides, various more validation test could be done on the inputs provided by the user.