# TRAVELLING AGENCY

Student: Furcovici Teodora-Melania

Group: 30431

Cuprins

[*TRAVELLING AGENCY* 1](#_Toc98148671)

Requirements Analysis

**Assignment Specification**

This is a JAVA Desktop application desired for booking vacations for specific destinations. The application has two types of users: regular user and travelling agency, which must provide a username and a password to use the application. These users work on data that is stored in a database.

**Functional Requirements**

* The **travelling agency (admin)** can perform the following operations:
* add a vacation destination
* delete a destination
* CRUD operations on vacation packages
* view all its listed vacation packages
* register on the platform
* log in on the platform
* The **regular user** can perform the following operations:
* Register on the platform
* Log in on the platform
* View all the available vacation packages
* Book a vacation
* View all its vacations
* Search for a vacation base on a filter

**Non-Functional Requirements**

* **Performance**

This type of application would be useful in any restaurant or bar. It could also be used by catering companies. Operators have at their disposal a real tool through which they can follow everything that happens on their traveling agency platform. Knowing all this data, they have the chance to know their users very well and thus can improve their services and promotions, regarding vacation packages.

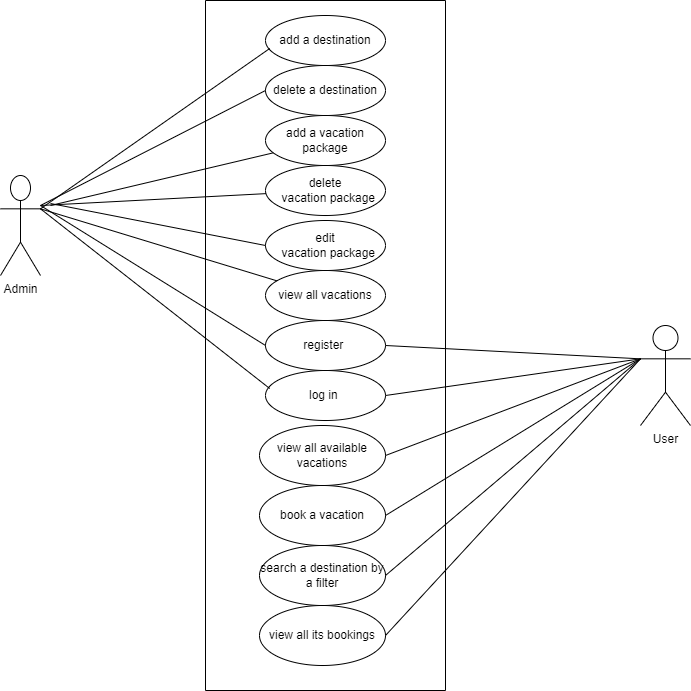
* **Security**

The security of the application will be achieved by requesting a username and a password for login.

* **Usability**

This application can be used in any traveling agency in order to digitize the service management program, for easier communication between clients and the system and to reduce customer service time.

Use case Model



**Use case name:** Add vacation package

**Level: < User-Goal, Subfunction, Summary > .**

**Actor principal/Main actor:** Admin

**Main success scenario:** The admin can perform add operation on vacations packages. If one of the beforementioned doesn`t exist in the application`s list, then it will be successfully added.

**Extension: <Un caz particular al scenariului, fie pozitiv sau negative>** If the admin tries to create a vacation that has been already added in the past, then an error will occur.

**Use case name:** Register

**Level: < User-Goal, Subfunction, Summary > .**

**Actor principal/Main actor:** Admin/user

**Main success scenario:** The admin/user can register on the platform using some credentials. If the account doesn`t exist in the application`s list, then an account will be successfully created.

**Extension: <Un caz particular al scenariului, fie pozitiv sau negative>** If the admin/user tries to register with an account that has been already created in the past, then an error will occur.

**Use case name:** Book a vacation

**Level: < User-Goal, Subfunction, Summary > .**

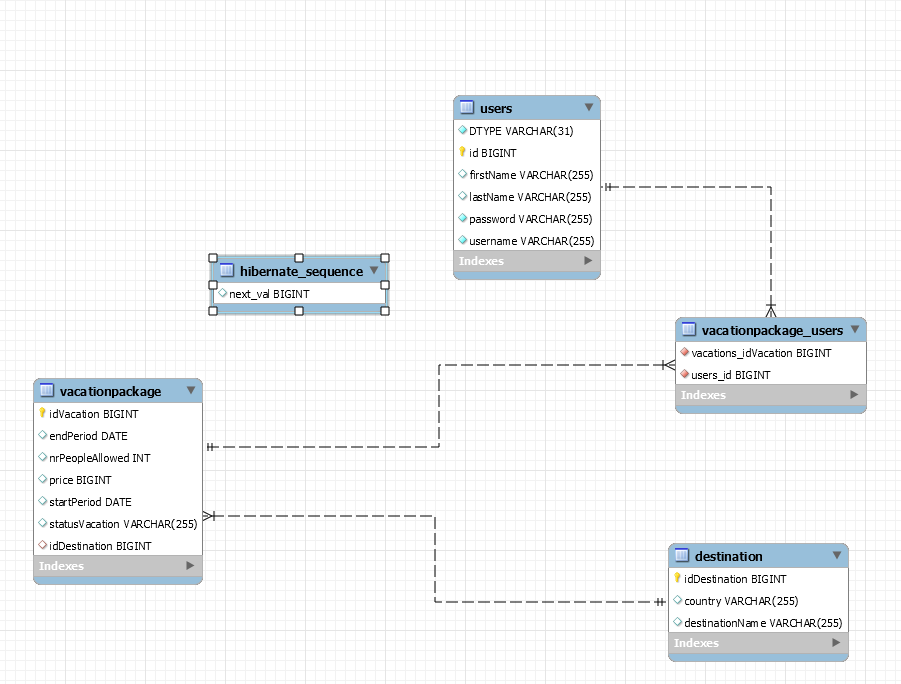
**Actor principal/Main actor:** User

**Main success scenario:** The user can book a vacation using its account. If on that vacation package there are available seats, then the vacation will be booked for that user and add to its vacation list.

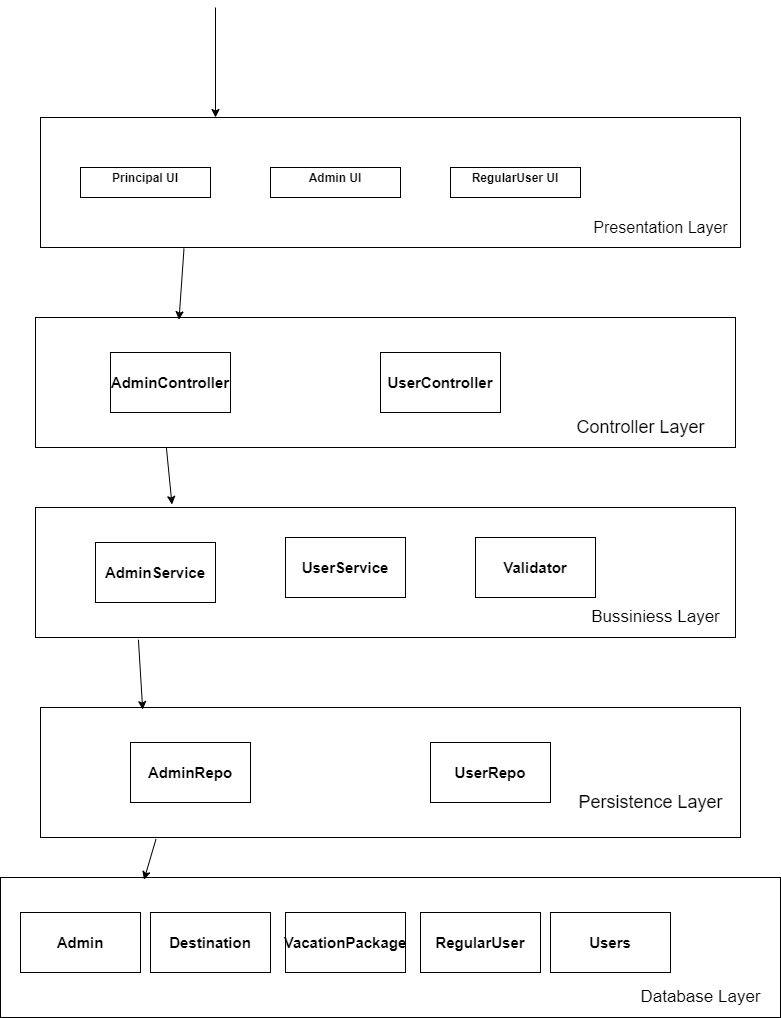
**Extension: <Un caz particular al scenariului, fie pozitiv sau negative>**

Diagrams

**Database Diagram**



**Class Diagram**



System Testing

To verify and test the correctness of the application implementation, I created a very explicit graphical interface that provides informational messages or error messages for each action taken. for situations such as exhausting the seats for a certain vacation package, editing a vacation with incomplete data, trying to log in without an account, introducing the wrong password/username etc. error messages will be displayed with messages to help solve these inconveniences.