Travel agency

Assignment project documentation

Zaharia Tudorita

30432

1. Project specification

The Travelling Agency should be able to:
1. add vacation destination
2. add vacation packages for a specific destination a. should contain information: name, price, period, extra details, number of people that can book the vacation
3. edit an existing vacation package
4. delete an existing vacation package
5. view all its listed vacation packages (with status: BOOKED, NOT_BOOKED, IN_PROGRESS)
6. delete vacation destination

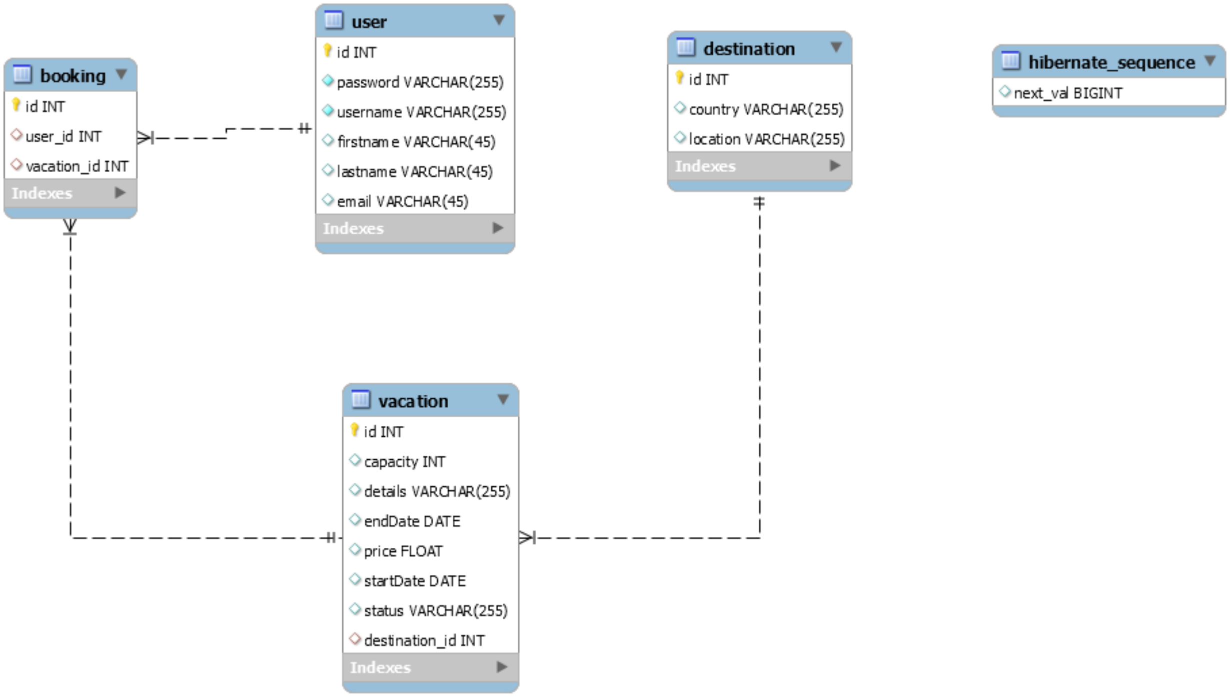
The Regular User should be able to:
1. register on the platform using some credentials (username/email - unique & password)
2. login on the platform
3. view all available vacation packages
4. filter vacation packages by destination/price/period
5. book a vacation
6. view all its booked vacations

Note: Each vacation package has a limit of people that can book the trip. When someone books the trip, you should keep track of the number of people who booked a vacation, so that if the maximum number of people who can book that vacation is reached, another user will not be able to see that vacation anymore in the platform.

Note2: When displaying a vacation to the Travelling Agency, a status will be shown, based on the number of people who booked that vacation, such as:
7. NOT_BOOKED: nobody booked that vacation yet
8. IN_PROGRESS: somebody booked that vacation, but limit not reached
9. BOOKED: limit of people is reached

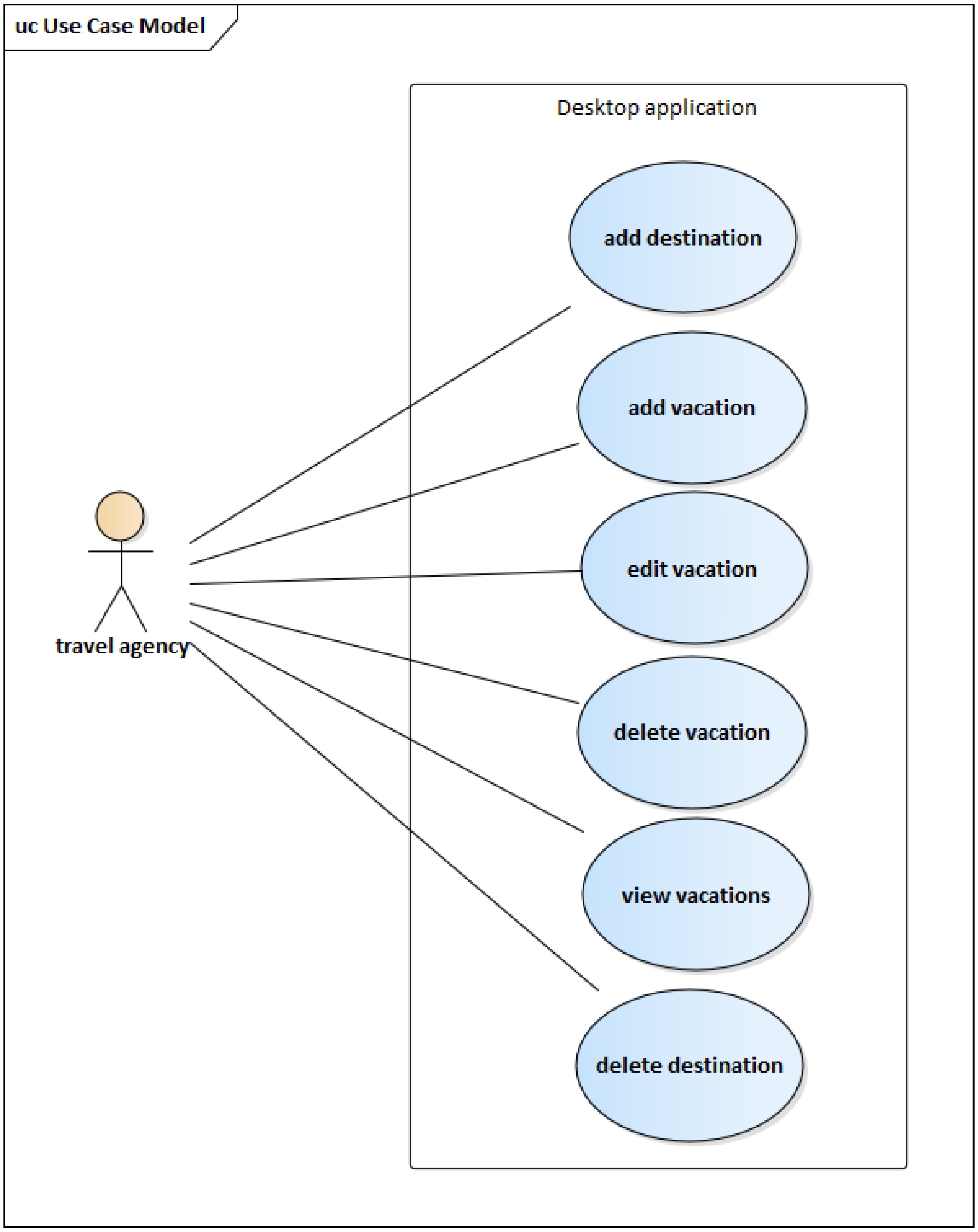
2. Database Diagram

User is responsible of creating bookings, consequently it must contain a list of bookings made by himself. Furthermore, each vacation is dependent on destination, with other words every time we create a vacation, beforehand we must choose a destination available elsewhere the row column won’t be persisted.

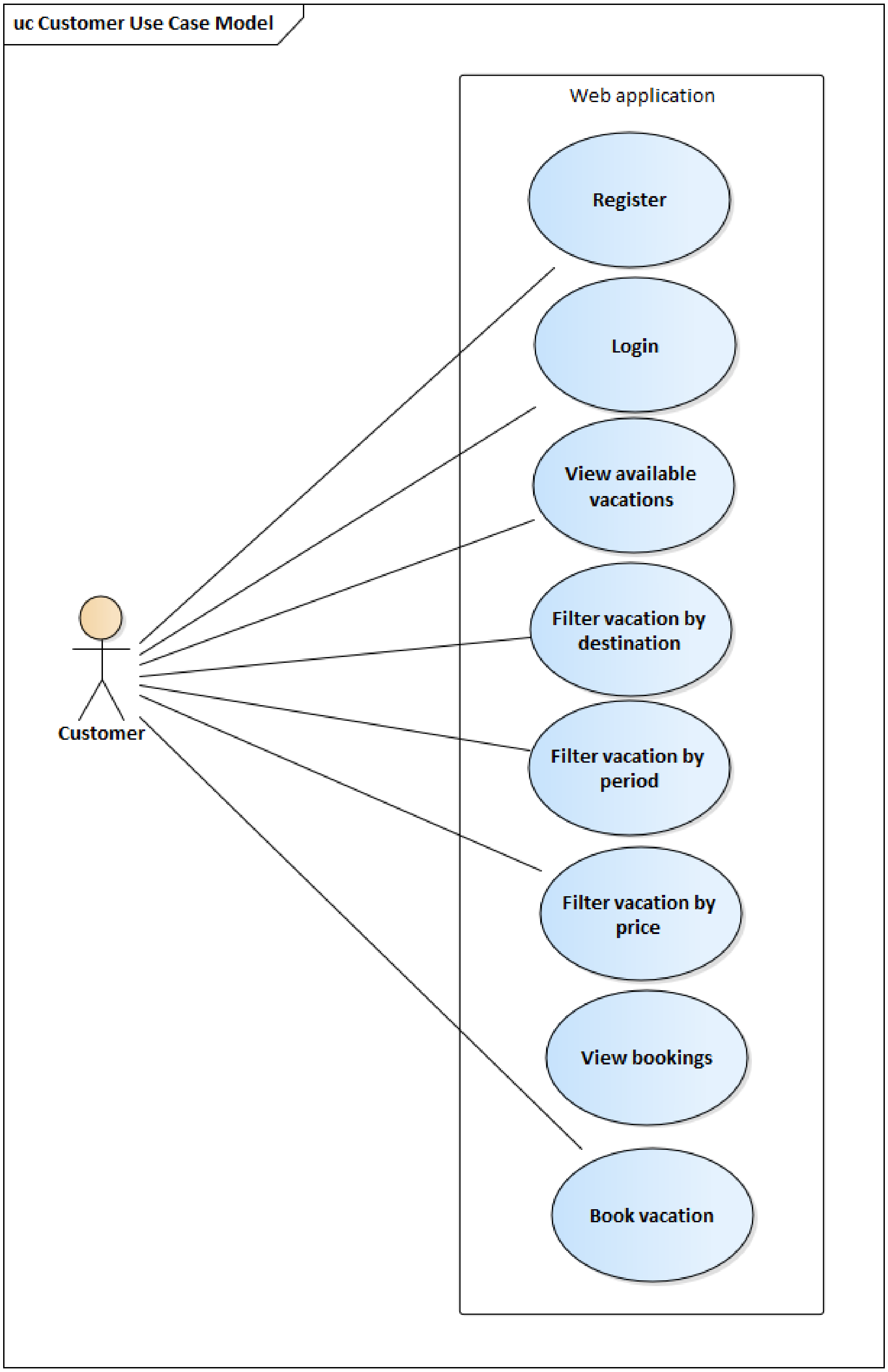


3. Use Case Diagram

3.1 Agency Use Case Diagram



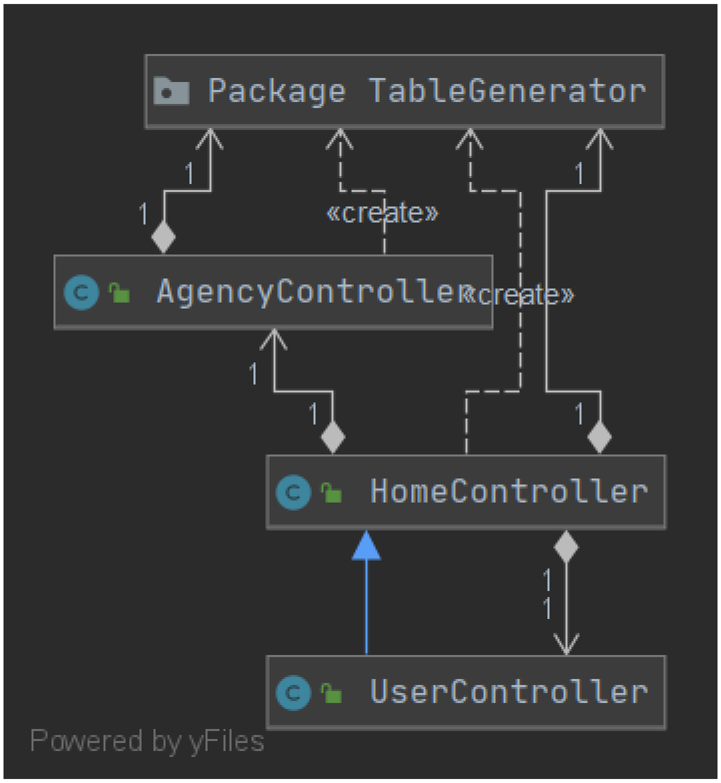
3.2 Customer Use Case Diagram

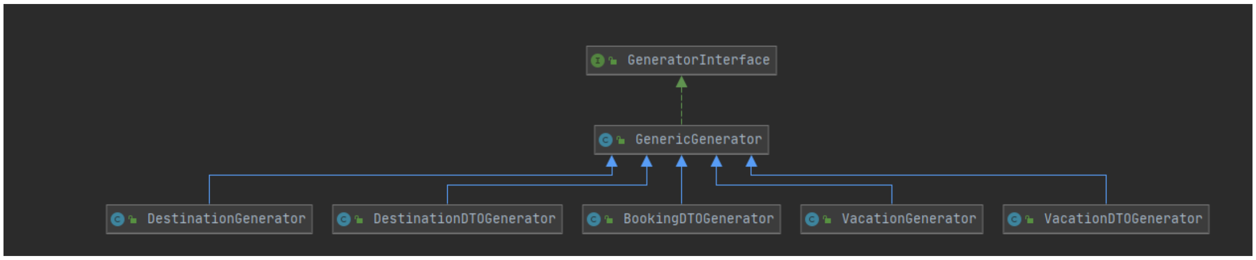


4. Class Diagram

4.1 Controller package

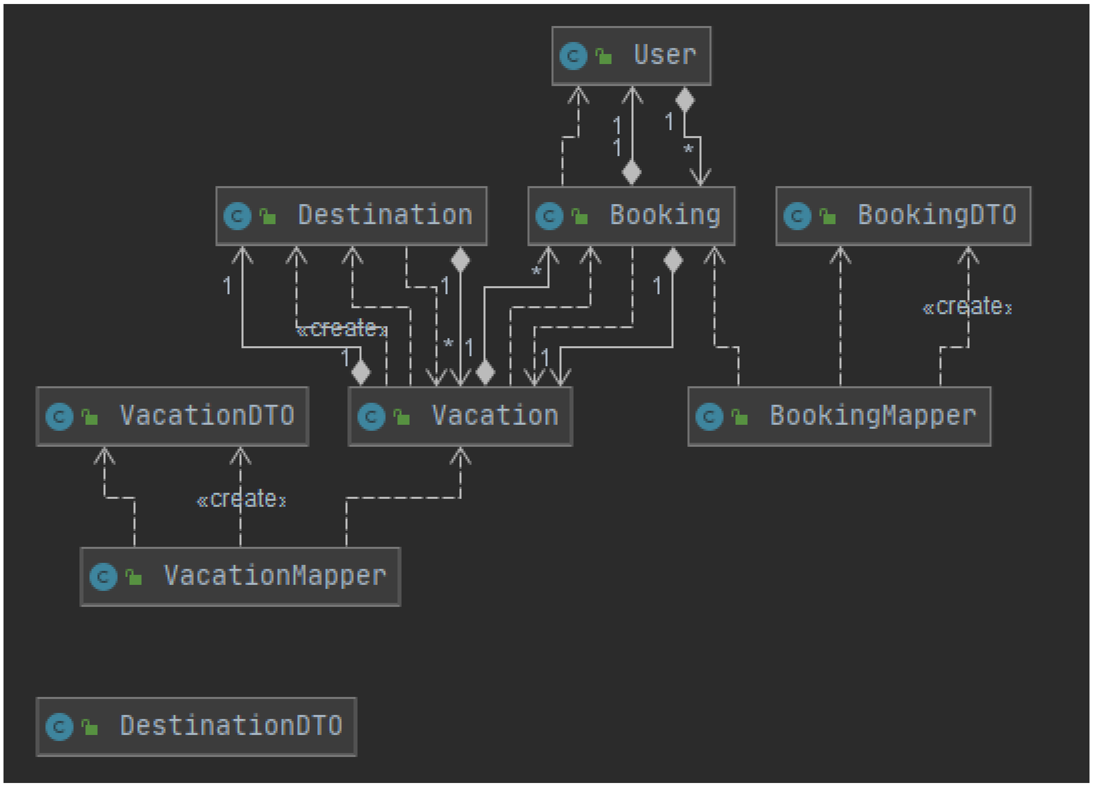
Here we can visualize the package Controller with its three main controllers for home page, agency, and user. The image below shows the package Table Generator responsible for creating a swing table for each object in Model.





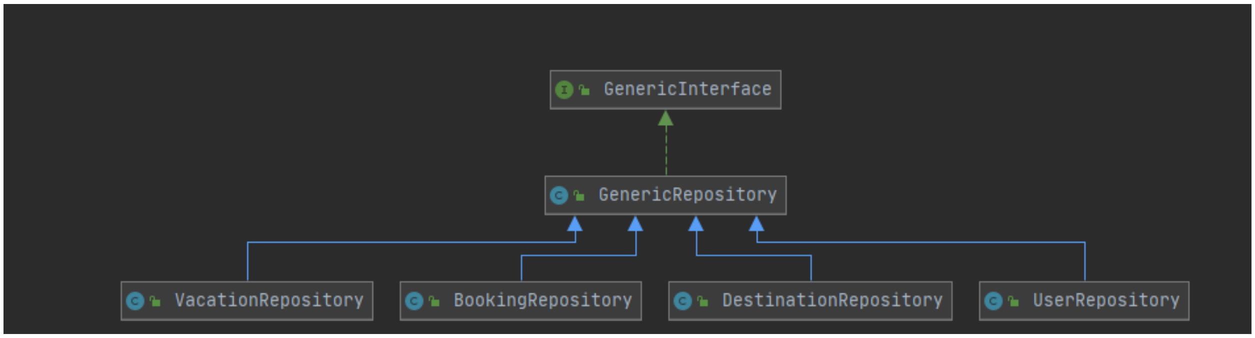
4.2 Model package

In this package we can observe the dependencies between objects such as vacation bounded to destination. Furthermore, there are two more packages inside, Mapper and DTO. DTOs are the derivate objects going to be shown on the UI interface, on the other hand Mapper classes have the task to migrate from Simple object to DTO and opposite.



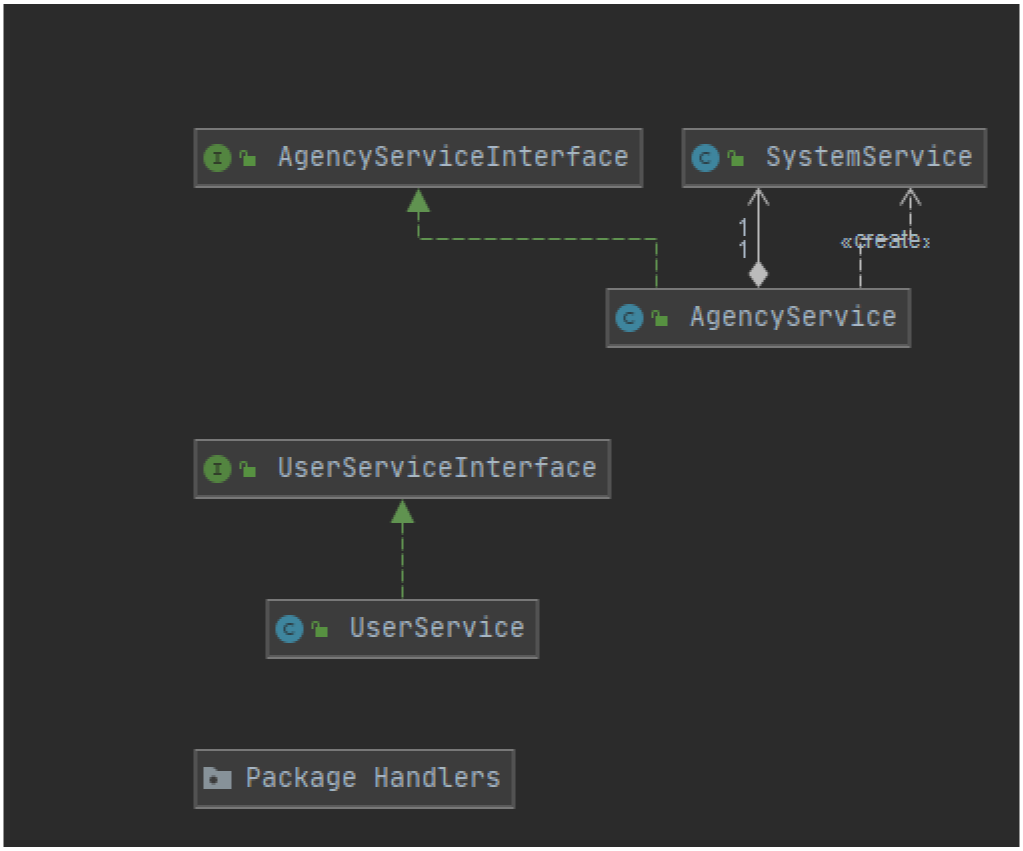
4.3 Repository package

Repository classes are centered around the generic interface containing the method signatures for CRUD and other operations on the database. Generic Repository is going to implement the interface and thereafter be specified for each type of object in the database.



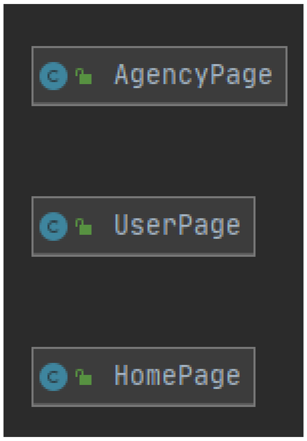
4.4. Service package

This package contains the business classes implementing the processes in functional requirements. Furthermore contains a package with handlers for each type of error encountered.

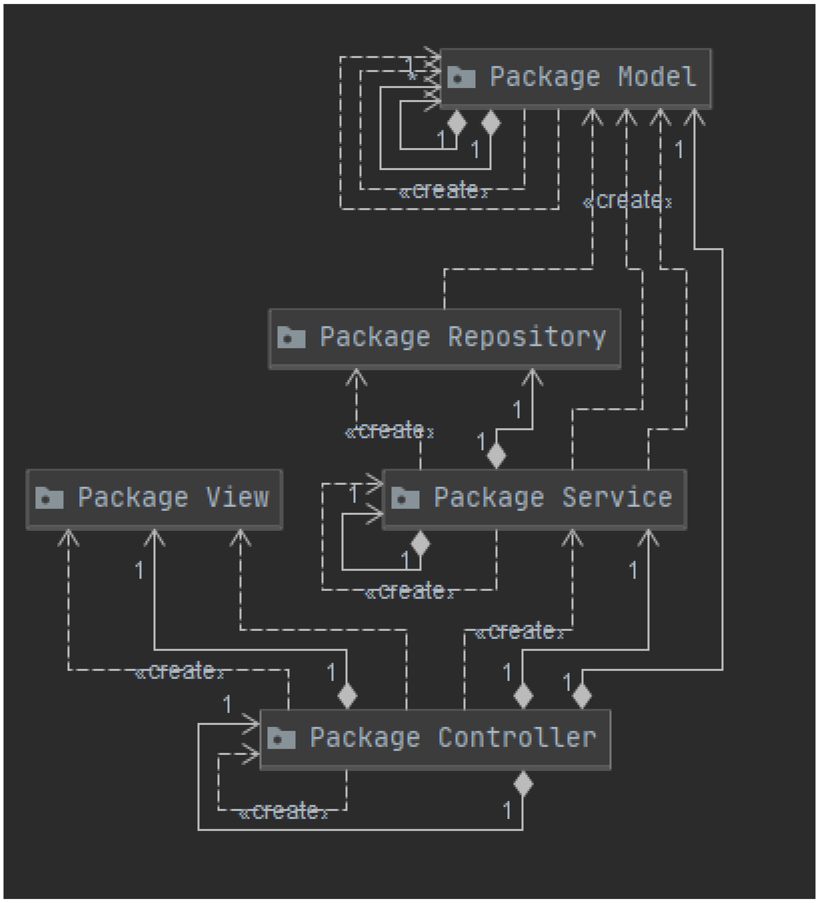


4.5 View package

Contains the UI pages which are going to be shown on display.



4.4. Demo package


Package Model
Package Repository
Package View
Package Service
Package Controller
«create»
«create»
«create»
«create»
«create»
«create»
«create»
1
1
1
1
1
1
1
1
1
1
1
1
1
1