Personal Car Rental system

Analysis and Design Document

Student:Micheș Mihnea Bogdan

**Group:30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

*The project is called “Personal Car Rental system” and, from now on, will be referred to as PCR.*

*The aim of this project is to provide owners of vehicles with the option of having some extra income by renting their cars to other people. Also, since there are no restrictions on the type of vehicle that can be offered, it gives customers the ability to rent different and probably more interesting vehicles.*

*The main activity related to this project is having a customer meet up with the owner of a vehicle, get the keys, pay and then enjoy said vehicle for the agreed period. In order to achieve this, multiple secondary activities need to take place, such as the vehicle’s owner advertising said vehicle, and the customer searching for vehicle rentals in a certain area.*

*After this activity, both parties have the option of writing a review for the other. For example, maybe the customer finds that the car is not properly maintained. Or maybe the owner will take the car back and find it to be very dirty. There’s the option to complain, or compliment.*

*This application will run in a Web browser. We will also employ administrators to make sure the content of the application is 100% under control.*

# Elaboration – Iteration 1.1

# Domain Model

*The vehicle rental domain does require 2 types of actors in order to function: the owners of the vehicles, and the customers that want to rent them. These are both users of the application, together with the administrator we mentioned above. Our users will be able to write reviews, also mentioned above, which have to me monitored by the administrators. The system also has to keep track of past rentals. For the purpose of this application, we will only implement the car as the only vehicle type, with the possibility for extension.*



# Architectural Design

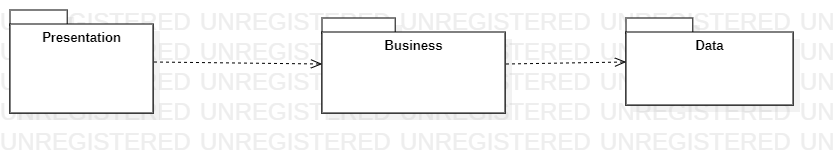
## Conceptual Architecture

*We will use the layered architecture, since this is a Web application. Layered architecture is about organizing code for a good separation of concerns. It consists of four standard layers: presentation, business, persistence and database. The layers of isolation concept also means that each layer is independent of the other layers, thereby having little or no knowledge of the inner workings of the other layers in the architecture. This is the most widely used architectural pattern for E-commerce web applications.*

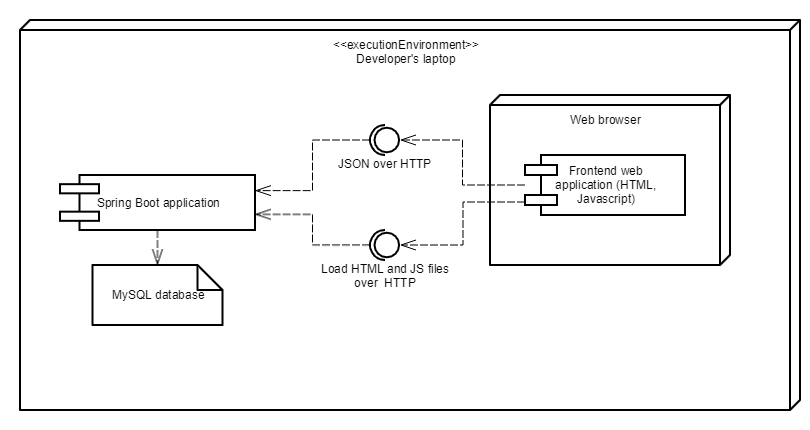
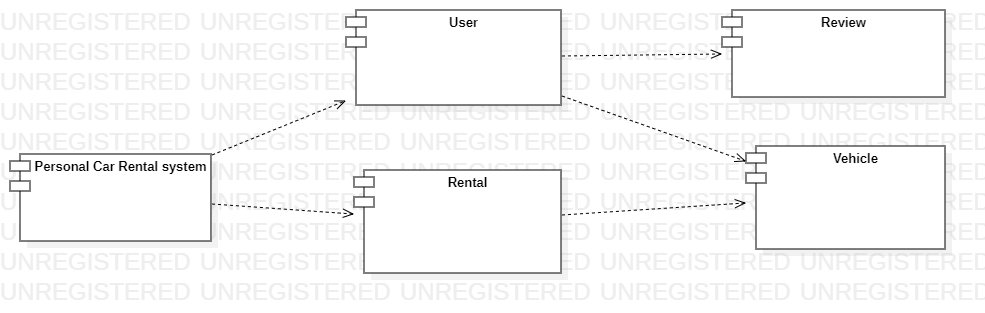
**

## Package Design

*Since we chose the Layered Architecture, the package diagram is made accordingly:*

**

## Component and Deployment Diagrams

**

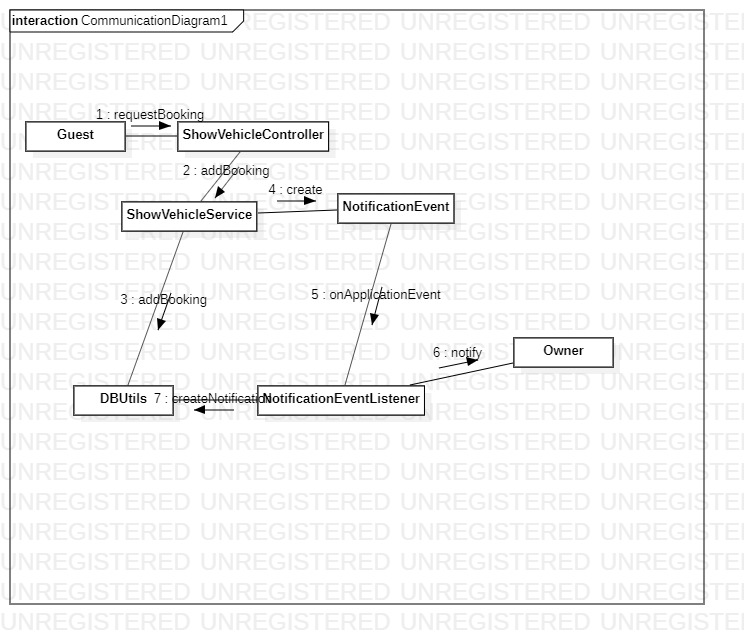
# Elaboration – Iteration 1.2

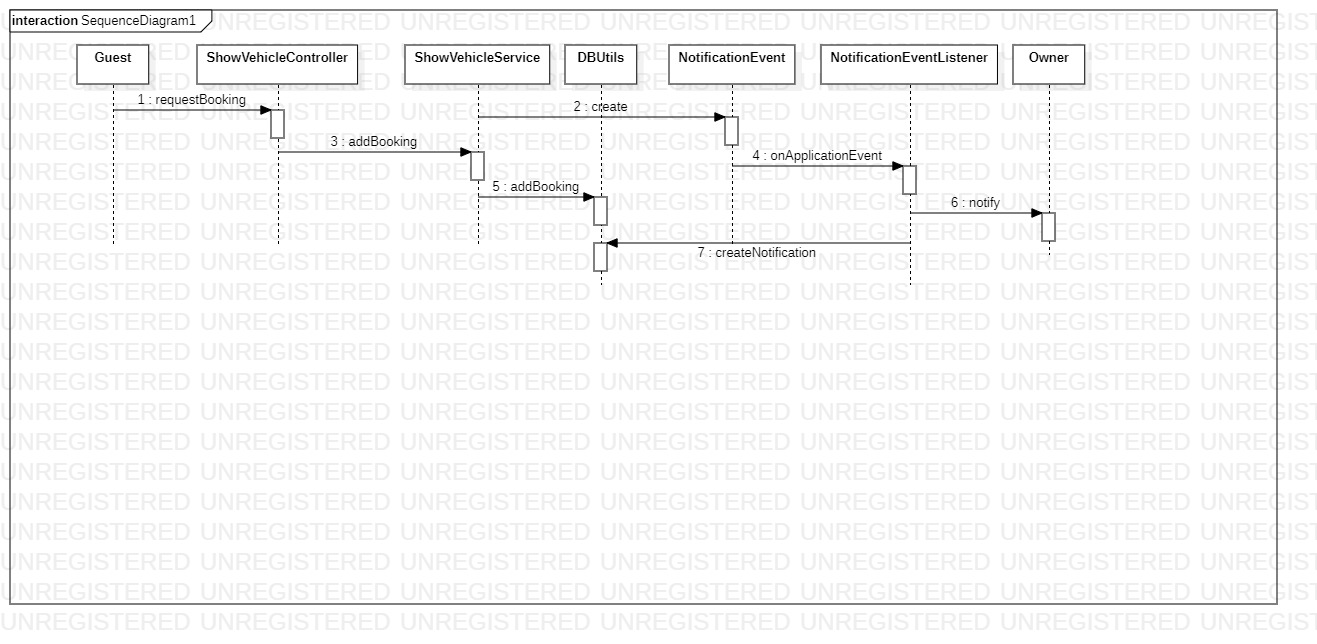
# Design Model

## Dynamic Behavior

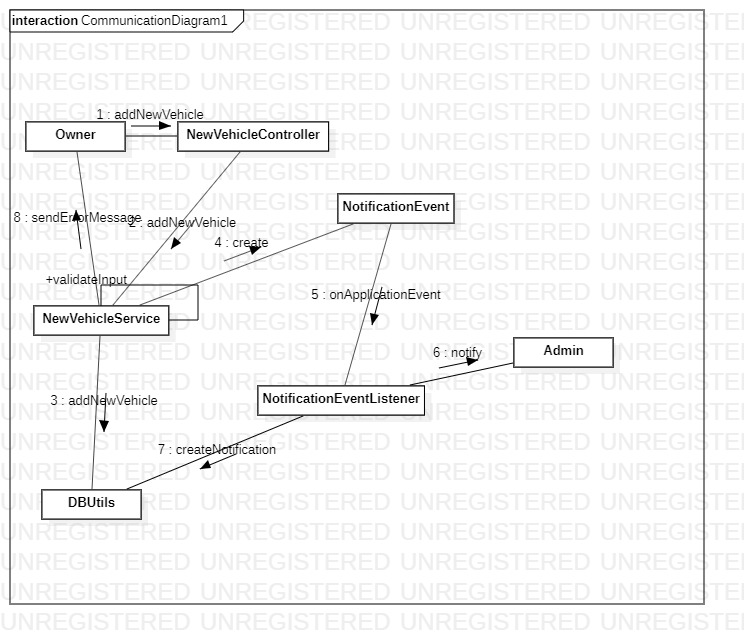
*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

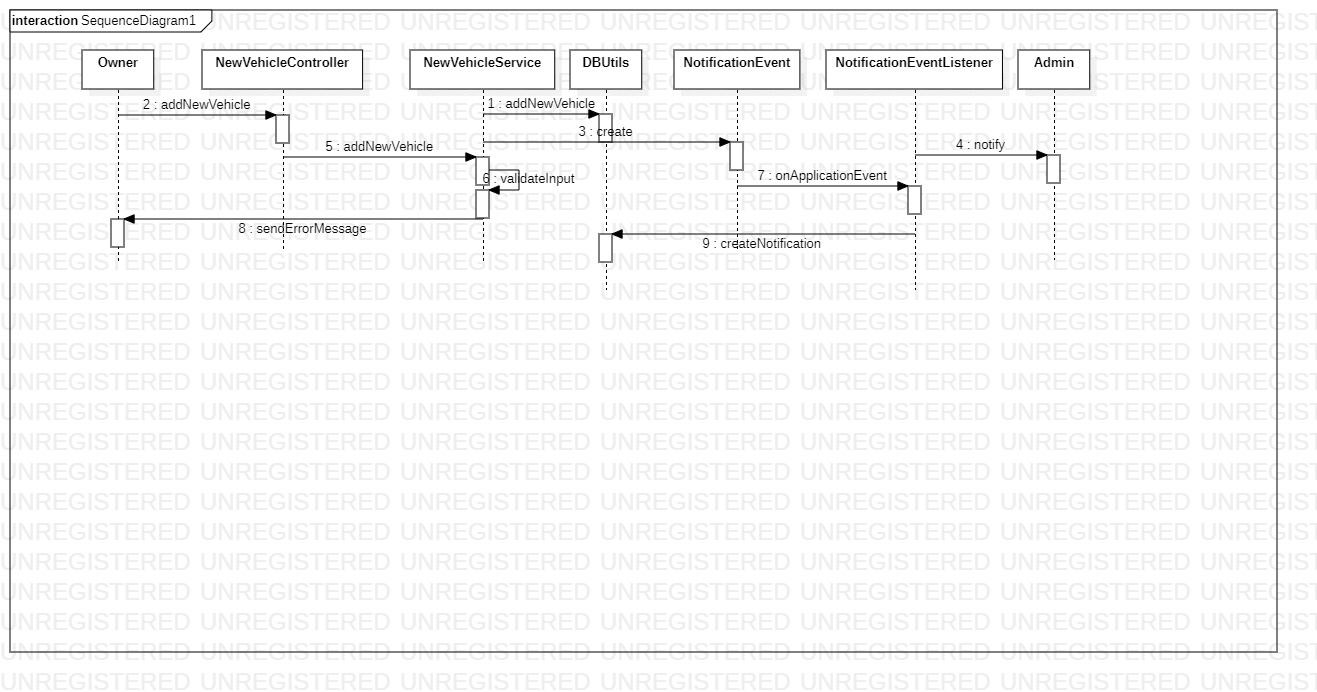
Scenario 1 : Guest creates a Booking for a Vehicle.





Scenario 2: Owner wants to create a new Vehicle

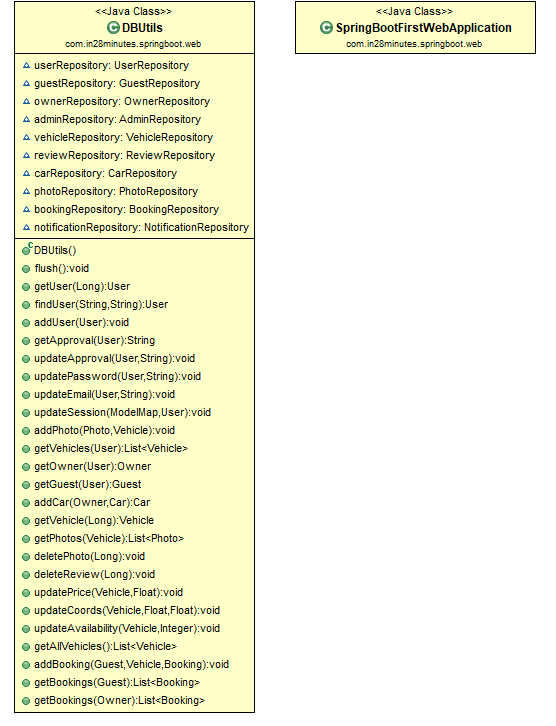




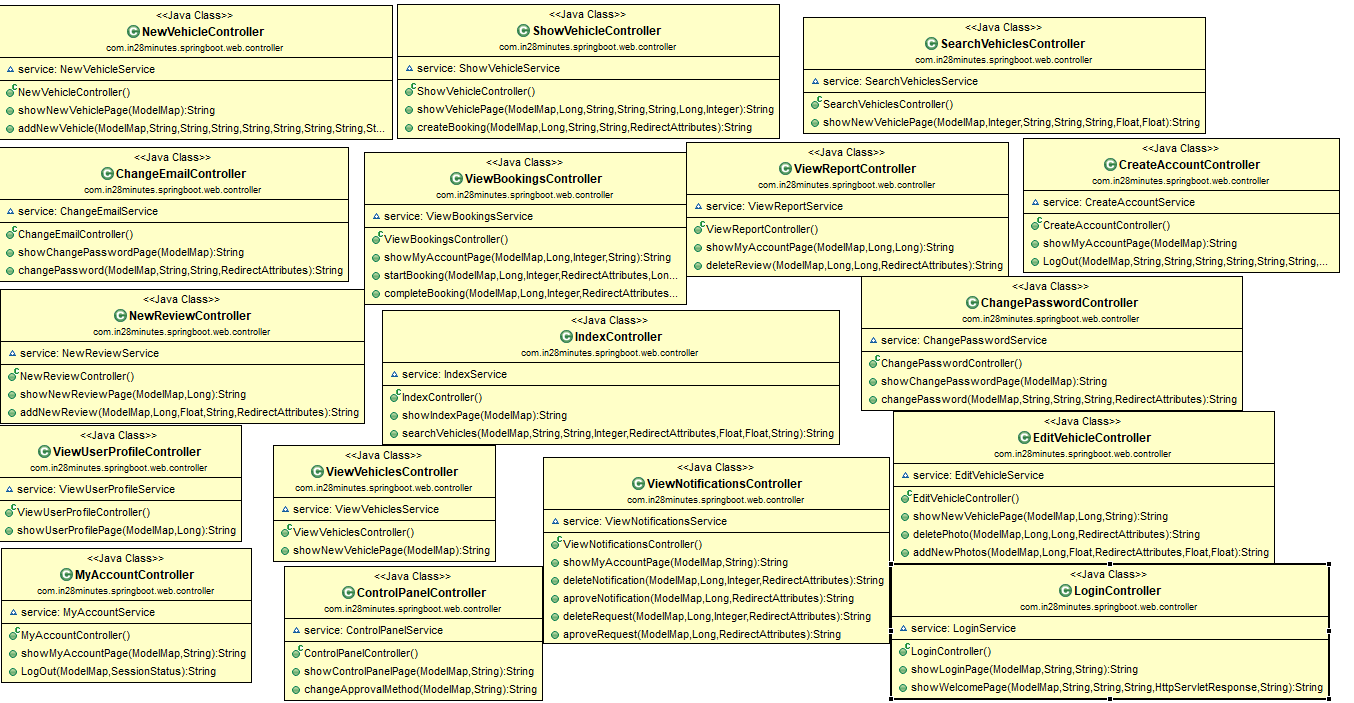
## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

Package web:

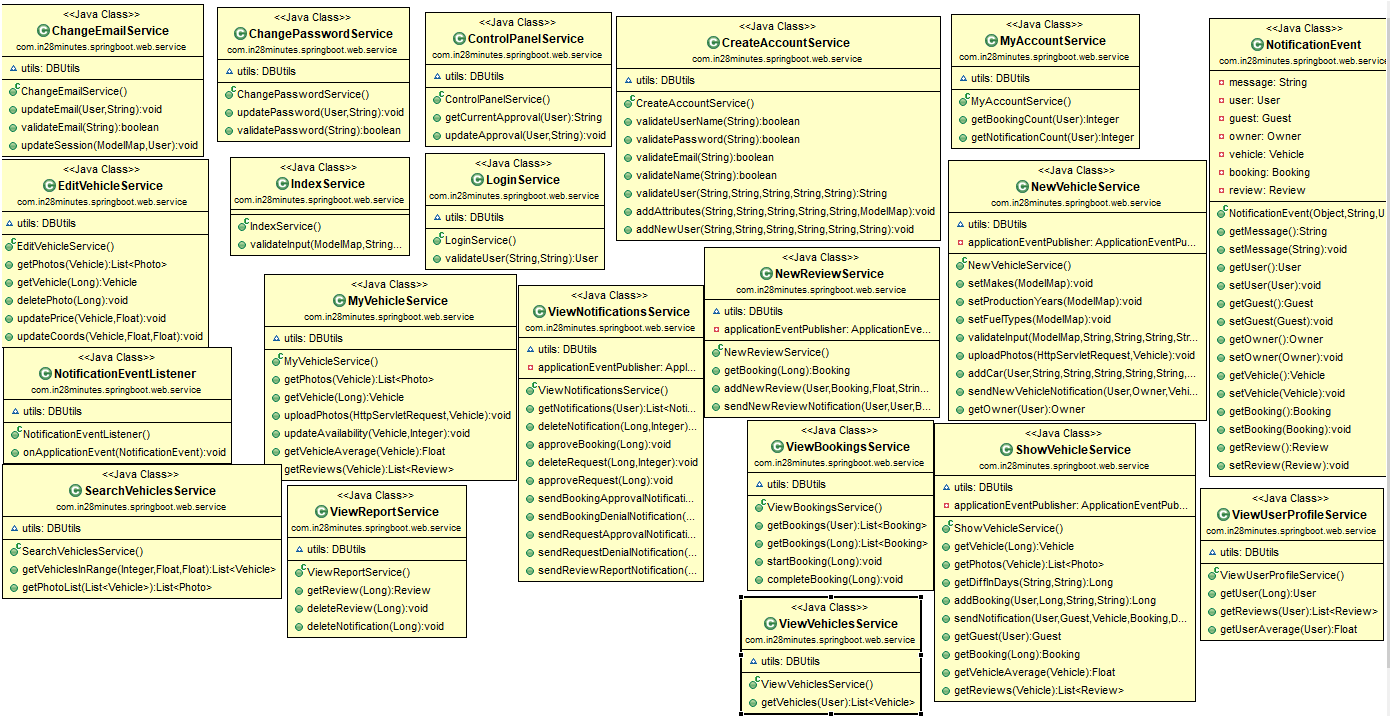


Package controller:

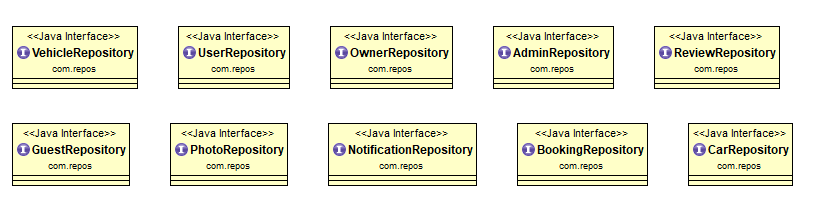


# 

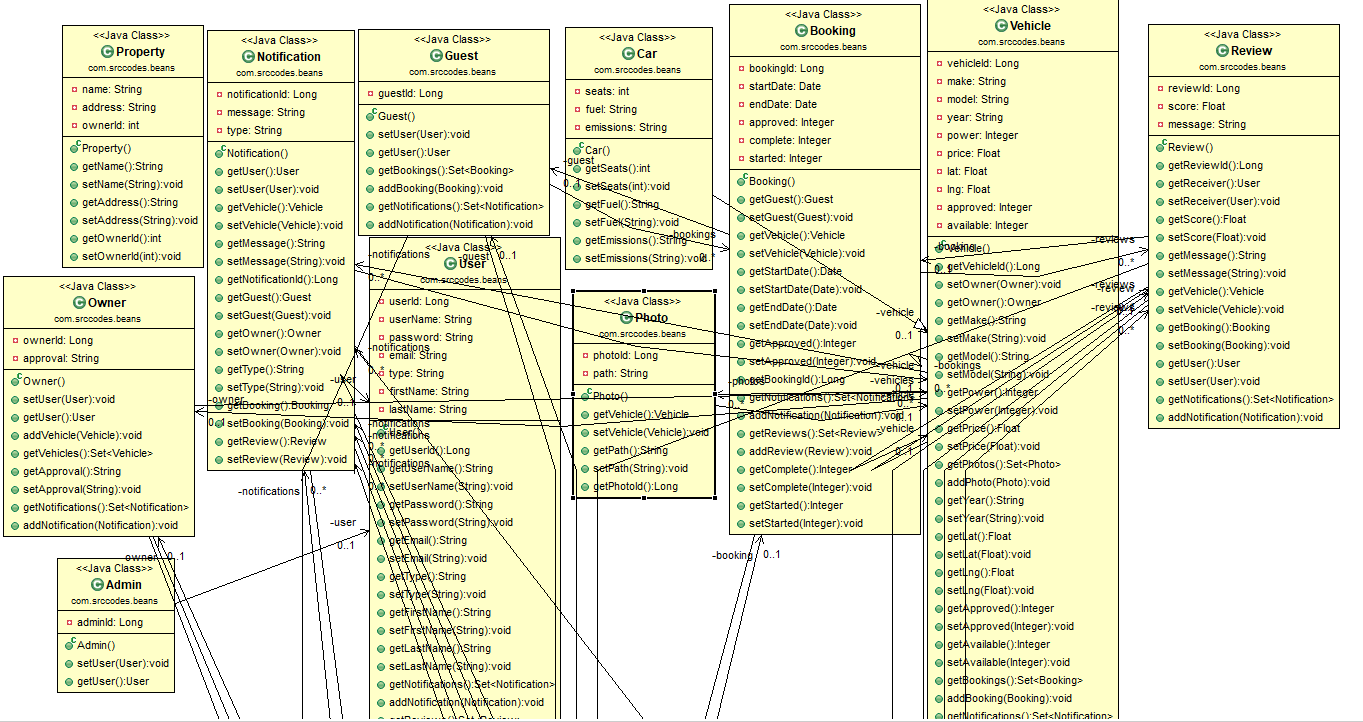
Package service:



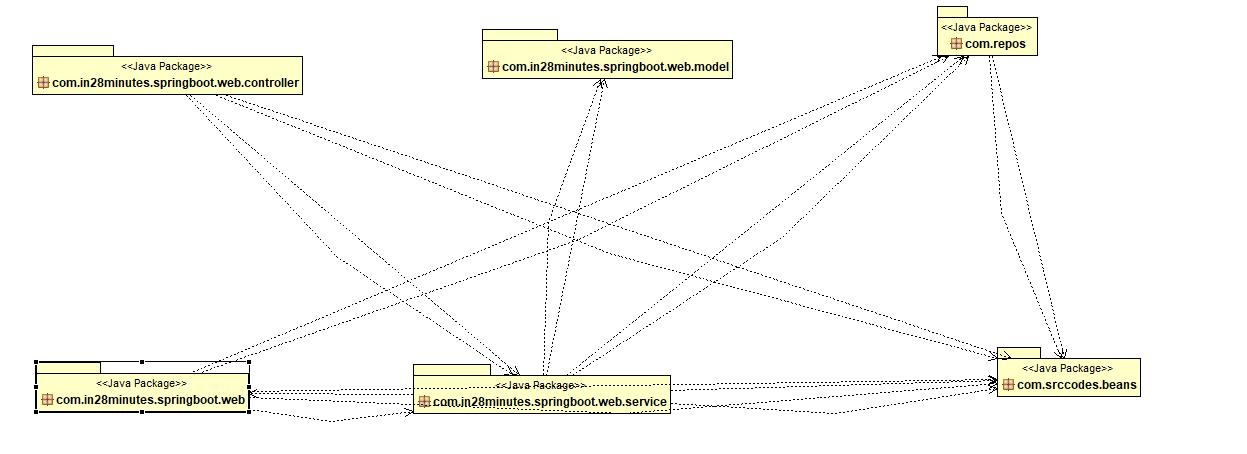
Package repos:



Package beans:

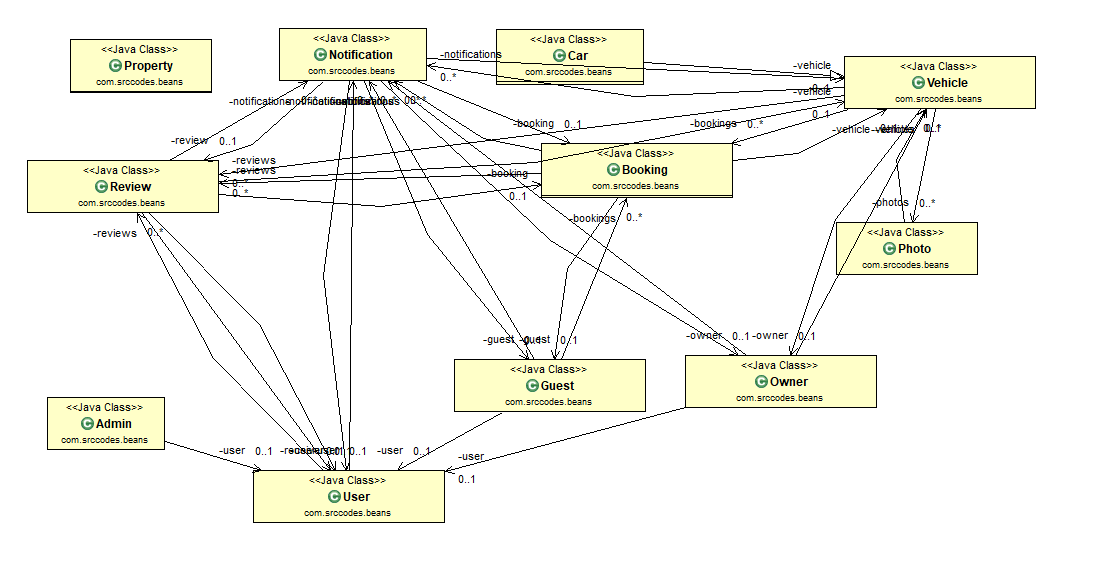


Package diagram:



# Data Model

*[Create the data model for the system.]*

**

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*

# Bibliography