YOUR PERFECT CAR

Analysis and Design Document

Student: Popovici Marius

**Group: 30238**

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

I would like to build an application for car renting. A client should have the possibility of creating an account and logging into it. Then, a list of disponible cars will be presented to clients and they will choose the one they consider worth renting. They will have the opportunity of selecting the period of renting and the price will be inversely proportional to the period (ex. 25 euro/day for 1-3 days of renting; 20 euro/day for 4-7 days). The client will also be able to sort the disponible offer by price or manufacturing years of cars. A car can be rented for picking (the client should go to the sedium and pick it) or delivering (the car will be delivered to the specified location). Also, a client can even request a driver, but only in the following interval: 08:00 – 20:00.

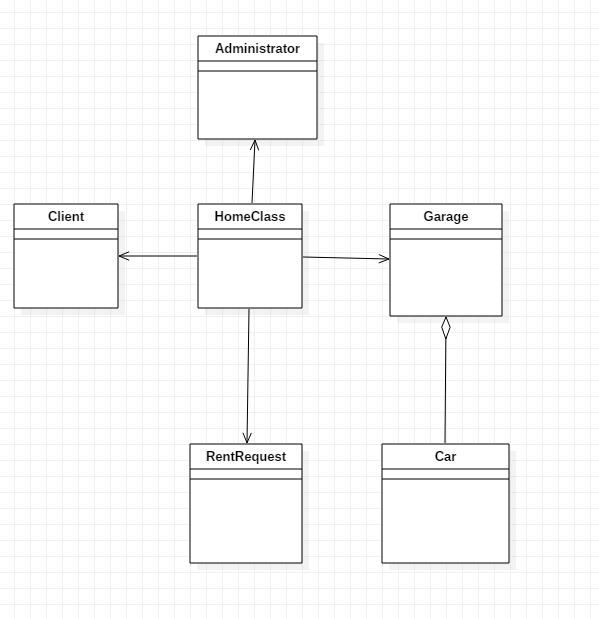
The app will have a menu from which a client could select to go to some pages, like: “Our cars”, “Terms and conditions”, “Contact” and “About us”.

# Elaboration – Iteration 1.1

# Domain Model

In software engineering, a domain model is a conceptual model of the domain that incorporates both behaviour and data. In Unified Model Language, a domain model is represented using a class diagram.

The domain model for this application contains model clases like: Client, Admin, Car, Garage, RentRequest.



# Architectural Design

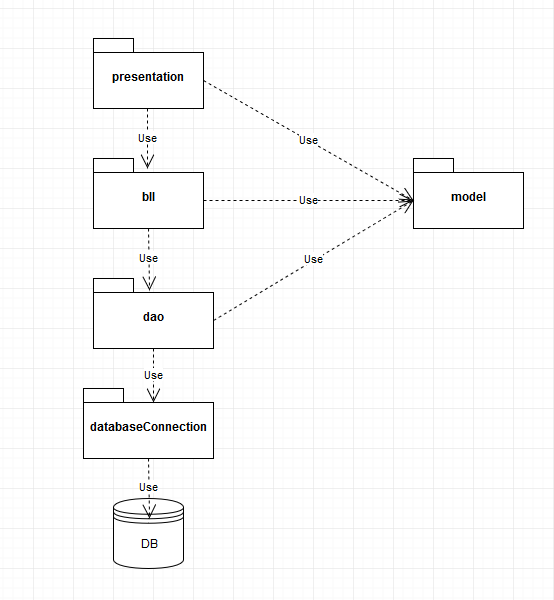
## Conceptual Architecture

*[Define the system’s conceptual architecture; use an architectural style and pattern - highlight its use and motivate your choice.]*

An **architectural pattern** is a general, reusable solution to a commonly occurring problem in software architecture within a given context. Architectural patterns are similar to software design pattern but have a broader scope. I will use the layered pattern for a good structure of the application. This way, the app will be divided on five layers: database connection, model, dao (bridge between model classes and their table equivalent in the database), bll (bussiness logic) and presentation package (responsabile with the interaction between users and application).

## Package Design

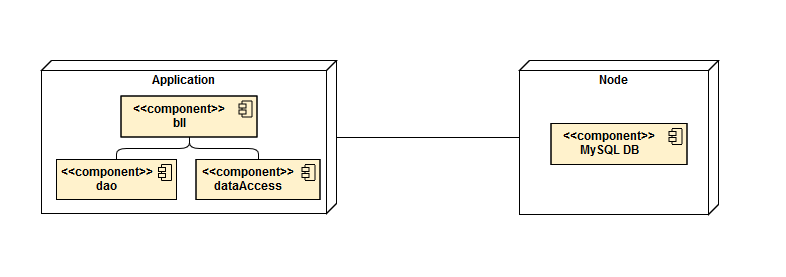
*[Create a package diagram]*

**

## Component and Deployment Diagrams

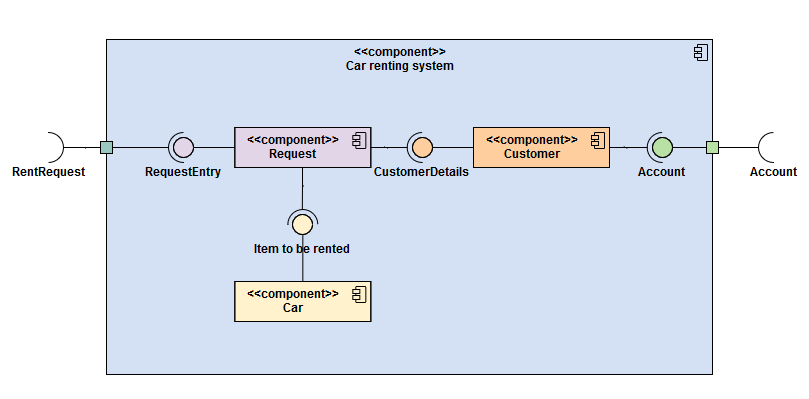
*[Create the component and deployment diagrams.]*

Deployment diagram:



Component diagram:

For renting operation:



# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

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

## Class Design

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

# 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.]*

This part will contain test case scenarions which have the role of verifying the functionality of the system.

# Future improvements

*[Present future improvements for the system]*

A future improvement could be:

- creating a chat, for real time communication between clients and staff

- a map of locations where a car can be reserved for picking.

# Bibliography