Car Service Appointments Application

Analysis and Design Document

Student: Trif Marina

**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

*[Present the project specification]*

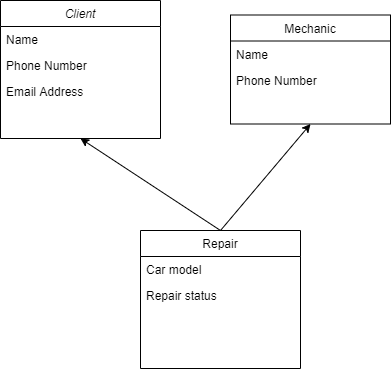
The project consists a web application that deals with car service appointments. It provides a system for clients sending requests to book an appointment for a car repair to the car service and to receive approval/disapproval. It also allows users to view or modify information about the status of the repair process and provide a transparent environment for all actions in this process.

# Elaboration – Iteration 1.1

# Domain Model

*[Define the domain model and create the conceptual class diagrams]*

The domain revolves around the car repair which contains information about the car, the status of the repair, the problem identified, the parts needed, the changes in the repair process etc. This also depends on both of the users involved in the repair: the client and the engineer. The user domain contains a separate class for each type of users, containing contact information or other essential details.



# Architectural Design

## Conceptual Architecture

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

The architectural style preferred is an evolutionary one in order to easier integrate change during the development process. Since the application deals with requests, approval and status an event driven approach could be taken into consideration.

## Package Design

*[Create a package diagram]*

## Component and Deployment Diagrams

*[Create the 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]*

## 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.]*

Test cases will first cover all the use cases scenarios. Unit testing will be done in every module such that functionality can be ensured.

# Future improvements

*[Present future improvements for the system]*

Future improvements can be: receiving notifications or confirmation emails about the status of the repair, being able to pay through the application, including options for car delivery etc.

# Bibliography