**Analysis and Design Document**

**Student: Dan Fulga**

**Group: 30432**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <19/04/2018> | <1.1> | <Revision 1.1> | <Dan Fulga> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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**

Car Sales and car inventory store project

This is an online car and car parts store that has listings of various cars along with their features. It also consists of car parts and accessories. The project allows users to buy cars online. It allows users to check various car stats including car engine, milage, tank capacity and other factors. Credit card payment facility available for car parts.

The project features:

* Visitor Registration/ Login module.
* User may check various car listing with features.
* User may check the car features and inventory parts.
* User may select and add products to shopping cart.
* Credit card payment option for car parts shopping.
* Test drive booking registration available.
* Car loan and other car booking facilities available in car buying section.
* *[***Elaboration – Iteration 1.1**
* **Domain Model**



There will be two different types of users - the administrator and the regular user (the client). Both of them will have associated profiles. The administrator has absolute control over the application : he can add new cars for sale, new inventory parts, set their stock (the amount of cars/parts they have available in the inventory) and so on.

The regular user can navigate through all these car listings and car parts and select the desired brand, desired options and so on and then he can proceed to buy the car or the desired inventory part.

The tables present in the diagram are the main entities of the program.

*[***Architectural Design**

* **Conceptual Architecture**

The main conceptual architectural pattern used in this application will be the Model-View-Controller pattern (MVC).This pattern is used to separate application's concerns.

* Model - Model represents an object or JAVA POJO carrying data. It can also have logic to update controller if its data changes
* View - View represents the visualization of the data that model contains.
* Controller - Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.



* **Package Design**



* **Component and Deployment Diagrams**
* *[***Deployment Diagram**



* **Component diagram**



* **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.]*

* **Future improvements**

*[Present future improvements for the system]*

* **Bibliography**