Report #2

For

“Better Automobile Inventory Management”

CSCI441\_VA

Software Engineering

Fall 2019

<https://github.com/gculver/SoftwareEngineering_FinalProject>

Team:

Paul Whitely

James Cox

Grant Culver

September 9, 2019

Revision History:

|  |  |
| --- | --- |
| Version No. | Date of Revision |
| v.1 | 9/30/2019 |
|  |  |
|  |  |

**Contents**

Contents

1 Interactive Diagrams

1.1 Use Case 1: InvSettings ..…………………………………………………………. 4

1.2 Use Case 3: FetchData .…………………………………………………………... 4

1.1 Use Case 4: CalcSupply..…………………………………………………………. 5

2 Project Management

2 Project Management……………………..…………………………………………... 7

3 References

3 References………….……………………..…………………………………………... 8

4 Class Diagram and Design Patterns

2.1 Class Diagram…………………………..…………………………………………... 9

2.2 Design Patterns……………………………...……………………………………… 10

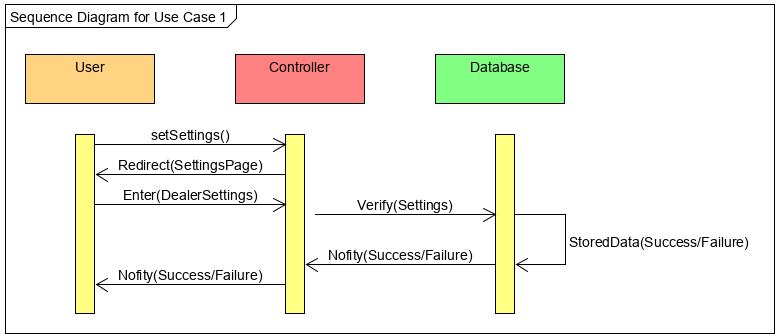
3 System Architecture

3.1 System Architecture…..…………………..………………………………………... 10

1. **Interaction Diagrams**

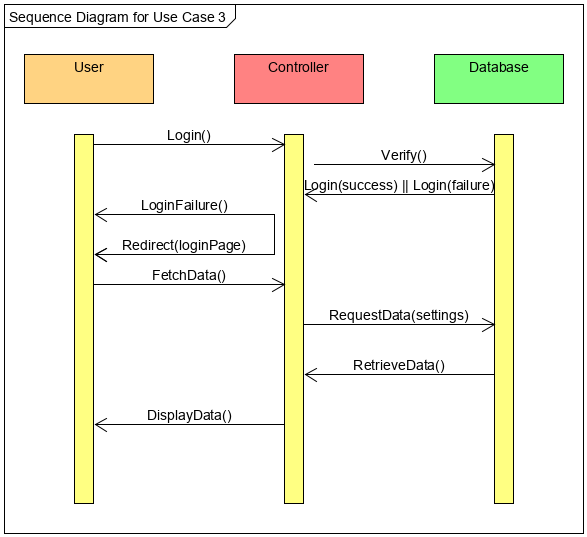
**1.1 Use Case 1: InvSettings**

We decided to assign the responsibility to set initial settings for the system to the controller, as mentioned by the **Expert Doer Principle** mentioned in textbook. This will allow the system to have a short communication chain between the related objects. The controller is the principle object and the secondary object in this situation would be the database. The database would be responsible to verify and store the settings that are received. In this instance, we believe it is necessary to use the publisher-subscriber design pattern to improve and implement this use case scenario. As related to this case, the user would be the subscriber and the system would be the publisher. Once the user subscribes valid input information, the publisher releases information of concern to the subscriber. I.E. A “success” message that notifies the user that the settings are set and acceptable. If the subscriber inputs invalid metrics, the publisher shows an error message showing invalid inputs.



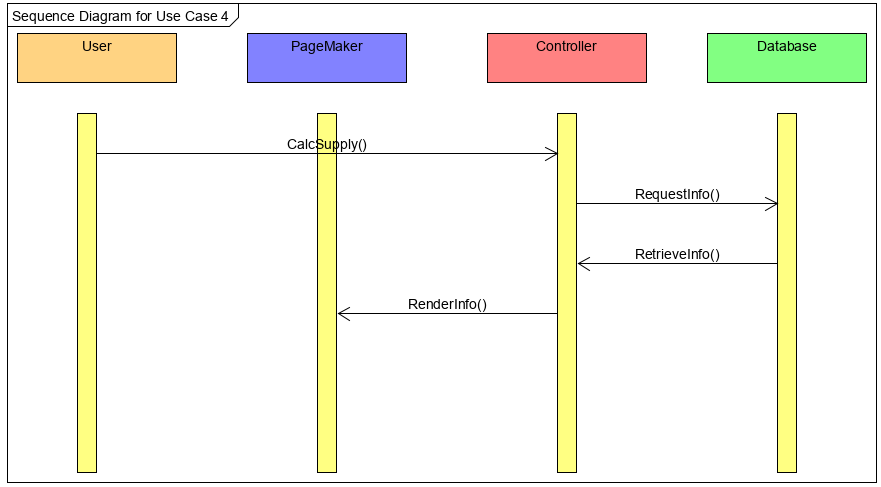
**1.2 Use Case 3: FetchData**

When a user attempts to fetch the data to calculate and display information from the user, the system attempts to verify the user authorization by querying the database for the correct account information. In this case, the databases main responsibility it to keep track of and report back to the user the relevant data. This data is retrieved by the system, which in-turn is responsible for passing information from the customer to the database and then relaying relevant information from the database to the customer based on the requests. For this interaction, we use the Publisher-Subscriber design pattern to improve this uses case’s design and functionality. When the user selects the desired data to be retrieved, they receive current inventory information that is able to correctly calculate the needed month’s supply and the inventory position. In this test case, the publisher gives the subscriber the pertinent information that the subscriber needs to make intelligent decisions.



**1.3 Use Case 4: CalcSupply**

Favoring the High Cohesion design principle the controller will be responsible for receiving information from the Database to calculate and display the inventory position and then pass the rendered page to the user.



1. **Project management**

The team members will share equally in design and development of the new car inventory management application. This is a project intended not only to serve a practical purpose, but to further the learning of the team members. The final product will allow users to better plan for automobile inventory levels at a dealership.

A detailed outline of the work needing to be done can be seen in the Gantt chart below. We will work together as a group to equally divide up the responsibilities in each of these tasks and come together as a team to create a finished project. We plan on using the tools available with Wrike.com to manage our project moving forward so that we can stay on task and complete our inventory management application fully functioning and on time to meet our customer’s needs.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sep 22-28 | Sep 29-Oct 5 | Oct 6-12 | Oct 13-19 | Oct 20-26 | Oct 27-Nov2 | Nov 3-9 | Nov 10-16 | Nov 17-23 |
| Setup project management and schedule on Wrike.com | |  |  |  |  |  |  |  |
| Setup Github access and test with team. | |  |  |  |  |  |  |  |
| Prepare LAMP stack and development environment. | |  |  |  |  |  |  |  |
|  | Setup basic site with all components activated | |  |  |  |  |  |  |
|  | Connect to data-gathering service and develop environment. | |  |  |  |  |  |  |
|  |  | Development of Core Features | | | |  |  |  |
|  |  |  |  |  |  | Review application performance, bugs and enhancements. | |  |
|  |  |  |  |  |  |  | Finalize application and documentation. | |

1. **References**

In addition to Linux, PHP and mySQL applications, the application will be using the Subrion CMS to speed development of the site layout and provide an attractive user interface, login components and other features.

<https://subrion.org/templates/>

The application will be using the phpGrid free component to allow content to be displayed in a grid format without hand-coding. phpGrid also has a phpChart component that will be used for displaying information in chart form.

<https://phpgrid.com/>

We will be using wrike.com as a project management tool to allow us to keep on track and complete the project on time.

[https://www.wrike.com/](https://www.wrike.com/workspace.htm?acc=3108024#path=folder&id=-1&c=timeline3&so=10&bso=10&sd=0)

We will be using github as a version control software so that we can work together as a team in the same repositories and have a version control to professionally manage our development.

<https://github.com/gculver/SoftwareEngineering_FinalProject>

1. **Class Diagram and Design Patterns**

**2.1 Class Diagram**



**2.2 Design Patterns**

The primary design pattern for this project will be the Model-View-Controller. The data model will be stored in a mySQL database and constraints and any necessary interface will be developed here. The Controller portion of the system will be written using PHP to interface with the database and external APIs. The system will present views of the information using HTML, Javascript and PHP.

1. **System Architecture**

The automotive inventory system will use a component-based design. Each non-trivial piece of the system will be based upon components that each have a well-defined purpose. These components will come from frameworks (e.g. PHP), external APIs or be custom-coded to support the application.

The system will be based on a LAMP stack. The operating system will be Linux supported by and Apache web server. Database storage will be MySQL and the primary programming language will be PHP. Javascript libraries will be utilized primarily to support the user interface and external libraries might be written in other languages (such as C++) that will be called from PHP code.

The system will be developed on local machines with updates published to GitHub. Testing and the final versions will be hosted on a cloud service to verify proper operation and to support potential usage by commercial customers.