Design Report on Software Maintainability

Version 1.0

By: Ling Yin, Lim Kaisheng, Andre Lim, Andre Lee, Andrew Ng, Trevor Lim, Yang Yang

Document Change Record

Revision	Description of Change	Approved By	Date
1.0	Initial Draft	Ling Yin	27 Oct 2022
1.1	Insert Architecture diagram	Ling Yin	27 Oct 2022

Contents

1. Design Strategies

1.1 Planning Phase

In order to bring our solution toward our goal, we need to ensure that the designs are not only effective in achieving the goals but they can also be maintained after which. Hence we adopt the Model View Control (MVC) model as our system architecture. It allows for change easily as it does not affect other functions easily.

1.2 Development Phase

We use Agile Methodolgies to develop our application. We also do test driven development such that after each sprint, the features are tested to ensure the features are working. Besides that, we will also work closely with our clients to ensure the usability of our product

1.3. Correction by Nature

We will correct our application while testing the application. And this is what we will look out for:

1.3.1. Corrective Maintainability

Fault detection done through testing.

1.3.2. Preventive Maintainability

Features implemented in atomic manner, each feature, tested independently, error detected easily.

1.4. Enhancement by Nature

We will enhance our application while testing the application. And this is what we will look out for:

1.4.1. Adaptive Maintainability

Can be easily adapted to a new operational environment.

1.4.2. Perfective Maintainability

After product delivery, quickly detect an error and correct it, reducing maintenance costs and time required.

1.5. Maintainability Practices

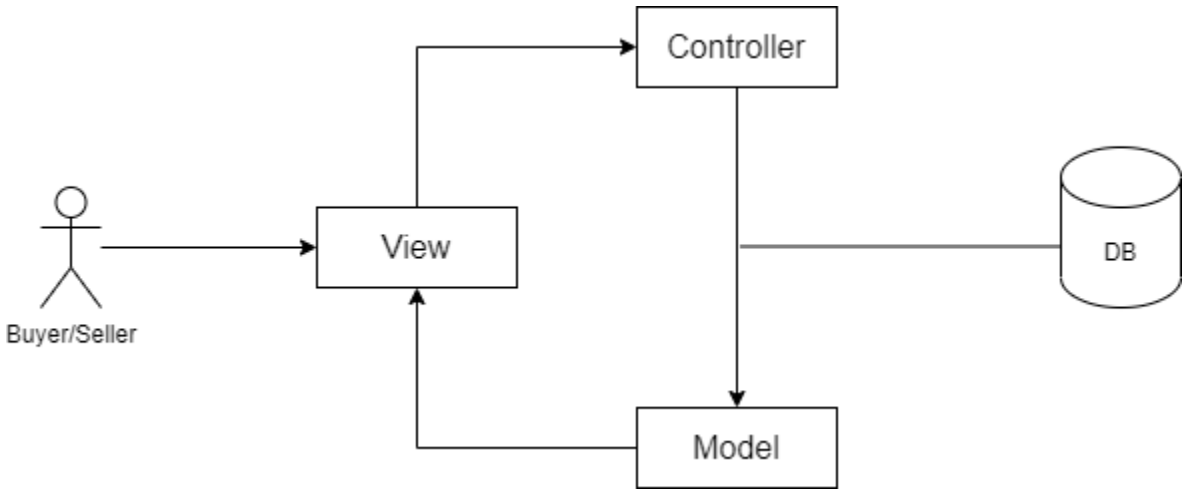
To uphold quality in both process and product, we have implemented the following maintainability practices over the course of our project:

- Readable Code
- Version Control
- Standardized
- Documentation
- Modularity

2. Architectural Design Patterns

MerchantDice is using the Model-View-Controller (MVC) architectural design pattern.

- The View layer is where it shows the components that display data from the Model layer to the user. The user interacts with this layer.
- The Controller layer is where the components of the application that receive and carry out commands from the users to alter the View or Model layer.
- The Model layer is where it holds the user data and records for the usage from the user interface layer.



3. Software Configuration Management Tools

This is where we will discuss on version control management, and tracking on who made what changes and when.

3.1. MediaWiki

MediaWiki is a free and open-source application. This service is used as it is easy for beginners to pick up. There are many FAQs provided which can teach users the functions required by the users. There is a wide range of functions which allows users to create their information in different styles. It also allows users to concurrently edit the page at the same time. Hence, editing of the page will not result in a loss of information.

3.2 GitHub

GitHub is a source code hosting platform using the distributed version control and source code management Git. GitHub is chosen for its familiarity and support provided by various IDE applications. GitHub also supports issue tracking similar to a ticketing system. Whether it's a software bug, code enhancement or documentation, users can open an issue, label them appropriately and assign them for other team members to resolve. All users involved will receive timely updates on the progress of the issue.

3.3. Google Drive

Google Drive service is used as a file storage and for the backup of documents initially created. This service allows users to share and store files within the group easily. This service allows users to edit documents concurrently and supports version control.