

**SHELVD**

# **Design Report on Software Maintainability**

Version 1.0

By: Arun Ezekiel S/O Richard, Cheong Yu Qing, Crystal, Kenneth Goh Jing Wei,  
Muhammad Sufyan Bin Mohd Jais, Nyan Maw Htun, Ryu Hyunsun, Zhu Yuhao

Document Change Record

Revision	Description of Change	Approved by	Date
0.10	Initial Template	Nyan Maw Htun	25/03/2024
1.00	Revised Edition	Nyan Maw Htun	27/03/2024

## **Contents**

1.	Design Strategies	3
1.1.	The Planning Phase Before Development	3
1.2.	The Process of Developing	3
1.3.	Correction by Nature	3
1.4.	Enhancement by Nature	3
2.	Architectural Design Patterns	4
3.	Software Configuration Management Tools	4
3.1.	MediaWiki	4
3.2.	GitHub	4
3.3.	Google Drive	4

## 1. Design Strategies

### 1.1. The Planning Phase Before Development

From the start, we would analyse and predict what kind of improvements we would be implementing in the future after the release of this application.

Once the widespread usage of this application becomes commonplace, it will have to be scaled up to handle even higher traffic rates. Hence, we targeted Scalability as one of the factors that we would have to look into in the future.

Since *shelvd* is heavily dependent on the user experience, we decided to adapt the Model-View-Control (MVC) model as our architectural mode. This system infrastructure framework will be beneficial for Graphic User Interface design. It allows easy to change and add functions without breaking and affecting the other functions.

### 1.2. The Process of Developing

We are testing out in a small, test driven development. Team members, non-developers and developers alike, perform the role of testers and provide continuous feedback on the design and usability of the application.

### 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

**shelvd** is using the Model-View-Controller (MVC) architectural design pattern. The Model layer is where it holds the user data and records for the usage from the user interface layer.

The View layer is where it store/shows the components that display data from the Model layer to the user.

Lastly, 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.

## 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.