



Go Where GaiGai

Design Report on Software Maintainability Version 1.0

By: Eugene Lim Zhi Jie, Yeo Kai Liang Jasper, Chang Tze Chuan, Isaac Soh Wei Yang,
Huang Ruimin, Yap Wee Kiat

Document Change Record

Revision	Description of change	Approved by	Date
0.10	Initial Template	-	-
0.20	Add Designed Strategies, Architectural Design Patterns	Huang Ruimin	15/10/2022
0.30	Revision of report presentation format	Yeo Kai Liang, Jasper	18/10/2022
1.00	Revised Edition	Eugene Lim Zhi Jie	

1. Design Strategies	3
1.1 Planning Phase Before Development	3
1.2 Process of Developing	3
1.3 Correction by Nature	3
1.3.1 Corrective Maintainability	3
1.3.2. Preventive Maintainability	3
1.4. Enhancement by Nature	3
1.4.1. Adaptive Maintainability	3
1.4.2. Perfective Maintainability	4
2. Architectural Design Patterns	4
3. Software Configuration Management Tools	5
3.1. MediaWiki	5
3.2. GitHub	5
3.3. Google Drive	5

1. Design Strategies

1.1 Planning Phase Before Development

Before starting the project, there were plans of developing additional functions after the initial launch of the web application. Furthermore, as more users hopped onto the web application, the capacity to handle higher user traffic is considered as well. Hence, the web application's scalability is the main focus for future improvements.

Go Where GaiGai web application will be built fundamentally on ReactJS, many new React components and states will be created and maintained eventually as more functionalities are added. Thus, it was decided that the Redux framework will be incorporated into the design pattern. This pattern is used to simplify the need for stores for React states. Redux combines all the stores into a single store for easier management of different React states and faster debugging.

1.2 Process of Developing

Development was carried out in small bursts of test driven cycles. Peer coding sessions were conducted in pairs within the development team to iron out possible bugs and development roadblocks. Non-development team members conducted testing after every development cycle. Feedbacks on bugs, possible improvements on design and user experience were provided back to the development team for the subsequent development cycle.

1.3 Correction by Nature

Corrections of the web application were made after every test iteration. Below are the criterias used:

1.3.1 Corrective Maintainability

Error identification 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

Enhancements of the web application were made after every test iteration. Below are the criterias used:

1.4.1. Adaptive Maintainability

Can be easily adapted to a new operational environment.

1.4.2. Perfective Maintainability

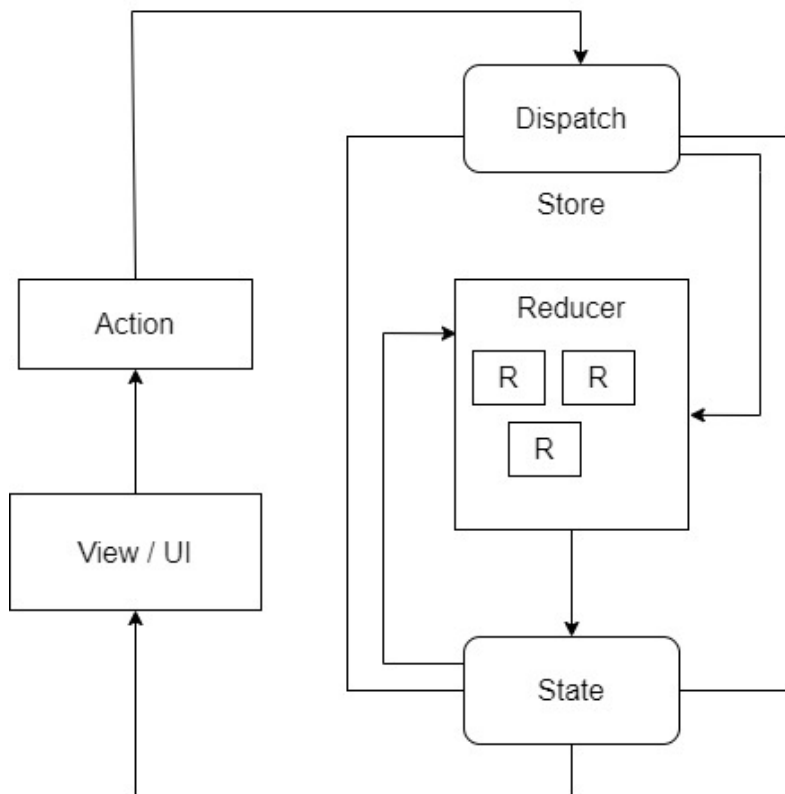
After product delivery, quickly detect an error and correct it, reducing maintenance costs and time required.

2. Architectural Design Patterns

As mentioned in the previous section, Go Where GaiGai is built on the Redux framework which works well with ReactJS. The framework includes the following components: view/ui, action, store, dispatch and reducer with the explanation below:

- **View/UI:** the main look of the web application where users will interact with.
- **Action:** a JavaScript object that only contains information for the store. An example would be the individual location information that users want to add into their planner.
- **Store:** a component that contains all the states every component uses which will be used to pass around other components. It allows easier maintenance of states by removing the need to maintain states inside parent components.
- **Dispatch:** a function in store that calls for an action, triggering a change of state.
- **Reducer:** a pure function feature that holds the logic and calculation needed to perform on state.

Following is a diagram view of architecture design pattern for Go Where GaiGai:



3. Software Configuration Management Tools

This section will include the software configuration and management tools used for the project.

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 is a hypertext publication collaboratively edited and managed by its own audience, using a web browser. It also keeps previous versions of the wikis, allowing for easy reversion to an older version of the page, preventing any loss of information

3.2. GitHub

Github is a source code hosting service for software development and version control using Git. It also supports issue tracking and notifies users of merging conflicts when merging codes of different versions. A Github repository was created to host and control the project's source code version. Individual branches were used for individual development and will be merged into the main branch using pull requests. The use of Github is to ensure no changes during development will overwrite any working program's functionality. It was also used for easier tracking of what was being changed in the source code.

3.3. Google Drive

Google Drive is a cloud file storage and synchronisation service developed by Google. It allows users to create and store different types of documents such as words, powerpoints, excel and raw data files which can be accessed by different users within a group. Furthermore, it allows cross editing and has version control which allows users to revert back to previous versions of documents in case of overwriting through edits. Google Drive is used in the project as a shared file storage for sharing raw data needed for development, writing of reports and presentation slides. It is also used to backup documents needed for the project.