Software Quality Assurance (SQA) Plan
By Team ONE

Date: 18 September 2022

Eugene Lim Zhi Jie — Project Manager, Release Manager

Yeo Kai Liang Jasper — Lead Developer

Chang Tze Chuan — Front-end Developer

Isaac Soh Wei Yang — Back-end Developer

Huang Rui Mi — QA Manager

Yap Wee Kiat — QA Engineer

Signature Page

Prepared by: Yap Wee Kiat	**Date: 18th September 2022**
Reviewed by 1: Huang Rui Min	**Date: 18th September 2022**
Reviewed by 2: Yeo Kai Liang Jasper	**Date: 18th September 2022**
Approved by : Eugene Lim Zhi Jie	**Date: 18th September 2022**

Document Change Record

Revision	Description of Change	Approved by	Date
0.1	Initial Template	RuiMin	05/09/22
1.0	Initial Draft	Wee Kiat	05/09/22
1.1	Purpose, Reference Documents and Management	RuiMin	07/09/22
1.2	Task	Wee Kiat	10/09/22
1.3	Documents, Standard, Practices, Convention and Metrics	RuiMin	11/09/22
1.4	Software Review	Wee Kiat	12/09/22
1.5	Test	RuiMin	14/09/22
1.6	Problem Reporting and Corrective Action	Wee Kiat	15/09/22
1.7	Software Quality Tools onwards	RuiMin	17/09/22
1.8	Finalizing	Wee Kiat	18/09/22

Contents

1. Purpose and Scope

1.1. Purpose

The purpose of this Software Quality Assurance (SQA) Plan is to establish the goals, processes, and responsibilities required to implement effective quality assurance functions for the Go Where GaiGai project.

The Software Quality Assurance Plan provides the framework necessary to ensure a consistent approach to software quality assurance throughout the project life cycle. It defines the approach that will be used by the QAM and Software Quality (SQ) personnel to monitor and assess software development processes and products to provide objective insight into the maturity and quality of the software. The systematic monitoring of products, processes, and services will be evaluated to ensure they meet requirements and comply with policies, standards, and procedures, as well as applicable Institute of Electrical and Electronic Engineers (IEEE) and ISO standards.

1.2. Scope

The purpose of SQA is to ensure that the software developed does not deviate from the original intended product. SQA is also concerned with identifying any errors, omissions, inconsistencies, and alternatives, enhancements or improvements that can be made at any stage of development.

Go Where GaiGai is able to return users the results of similar destinations and allow them to add these to a planner, for future usage or to be shared with others. This project uses data extracted from Yelp's public base via python on Jupyter Notebook, the front-end will be developed with Node.js and react on Visual Studio Code.

2. Reference Documents

- IEEE STD 730-2002, IEEE Standard for Software Quality Assurance Plans (http://standards.ieee.org/reading/ieee/std_public/description/se/730-2002_desc.html)
- ISO IEC 90003:2004 Software Standard (http://praxiom.com/iso-90003.htm)
- Node.js documentation (https://nodejs.org/api/index.html)
- ReactJS Documentation (https://reactjs.org/docs/getting-started.html)
- React Leaflet Documentation (https://react-leaflet.js.org/docs/start-introduction/)
- Project Plan
- System Requirement Specifications

3. Management

This section describes the management organizational structure, its roles and responsibilities, and the software quality tasks to be performed.

3.1. Management Organization

The implementation of the quality assurance system is the responsibility of the Quality Assurance Manager (QAM).

3.1.1. Project Management

The Project Manager will be responsible for approving:-

- The system requirement specification document
- The overall time scale for the project
- The choice of system development life cycle
- The choice of software development tools and techniques utilised
- The selection of project teams
- The training of project teams

3.1.2. Assurance Management

The QAM provides Project Management with visibility into the processes being used by the software development teams and the quality of the products being built. The QAM maintains a level of independence from the project and the software developers.

In support of software quality assurance activities, the QAM has assigned and secured Software Quality personnel from the pool of available SQ trainees to coordinate and conduct the SQ activities for the project and report back results and issues.

3.2. Tasks

This section summarizes the tasks (product and process assessments) to be performed during the development of software. These tasks are selected based on the developer's Project Plan and planned deliverables, and identified reviews.

3.2.1. Product Assessments

The following product assessments will be conducted by SQ personnel:

- **User Interface (UI)**

  The UI of Go Where GaiGai will be assessed to ensure it is simple and consistent. The interface's design should follow a similar pattern that already exists, making usability clear and self-explanatory. Users should get familiar with the application's interface and navigate around without much effort.
- **Info Window**

  Go Where GaiGai must be able to show the information of places based on users' searches. The assessment with regards to this functionality must ensure that the information that is being displayed is up-to-date and so that users will not experience any troubles due incorrect details.
- **Add destination**

Go Where GaiGai aims to let users add destinations based on searches into a planner. Assessment on this functionality is to ensure that the destination is correctly added into the planner, so that when users choose to share or go with their choices they will not end up in somewhere foreign to what they remembered adding.

- **Import/Export planner**

  Assessment on this feature is to make sure that planner that users have done is correctly exported as a file (.txt) so that they can view it offline and also share their planner with others. It also wants to ensure that the planner can be properly imported to our application for editing of the planner.

- **Category and Location Search component**

  To improve the scalability of Go Where GaiGai, the location of category and location search should work properly. This is because as ratings and shops may change over time, to keep users up-to-date and satisfied it is necessary to ensure they can get the correct information from the search. Assessment on this feature will be on the efficiency and latency of search.

3.2.2. Process Assessments

The following process assessments will be conducted by SQ personnel:

- **Requirement management process**

  It is the process of eliciting, documenting, organizing, and controlling changes to the requirements. This enables the development team to identify, control, and track requirements and changes that occur as the software development process progresses.

- **Change management process**

  Changes in the process are unavoidable and should be made to improve the productivity and quality of the process. The SQ team follows a set-up for implementing software process improvement which is targeted towards improving the quality, increasing productivity on the development of the product.

- **Maintainability management process**

  Go Where GaiGai must be maintainable, meaning that its faults can be easily corrected, improve performance and change accordingly to the requirements.

- **Risk management process**

  The SQ team has risk management measures and protocol to mitigate risks and minimize the impact. This process is to avoid disasters and heavy losses. Some of the common risks are running over budget, behind schedule and lack of resources for implementation of some features.

3.3. Roles and Responsibilities

This section describes the roles and responsibilities for each assurance person assigned to the Project.

3.3.1. QAM

Responsibilities include, but are not limited to:

- Secure and manage SQ personnel resource levels
- Ensure that SQ personnel have office space and the appropriate tools to conduct SQ activities
- Provide general guidance and direction to the SQ personnel responsible for conducting software quality activities and assessments
- Assist SQ personnel in the resolution of any issues/concerns and/or risks identified as a result of software quality activities
- Escalate any issues/concerns/risks to project management

3.3.2. Software Quality Personnel

Responsibilities include, but are not limited to:

- Develop and maintain the project software quality assurance plan
- Generate and maintain a schedule of software quality assurance activities
- Conduct process and product assessments, as described within this plan
- Identify/report findings, observations, and risks from all software assurance related activities to the QAM

4. Documents

4.1. Purpose

This section identifies the minimum documentation governing the requirements, development, verification, validation, and maintenance of software that falls within the scope of this software quality plan. Each document below shall be assessed (reviewed) by SQ personnel.

4.2. Minimum Document Requirements

- System Requirement Specifications
- Use Case Description
- Use Case Diagram
- System Architecture
- Project Proposal Gantt Chart
- Project Proposal Initial UI Mockup
- Project Plan
- Risk Management Plan
- Configuration Plan
- Change Management Plan
- Design Report on Software Maintainability
- Release Plan
- Test Plan

5. Standards, Practices, Conventions and Metrics

5.1. Purpose

This section highlights the standards, practices, quality requirements, and metrics to be applied to ensure a successful software quality program.

5.2. Software Quality Programme

These practices and conventions are tools used to ensure a consistent approach to software quality for all programs/projects.

The four most important qualities are functionality, reliability, usability and maintainability.

Functionality is important to ensure that the project is able to deliver the functions as stated to the user, if the project is unable to meet the requirements, users can easily find other applications that are providing the same functions.

Reliability is needed to ensure that the project is able to maintain its level of performance under stated conditions for a stated period of time, if the project is not even reliable, users will be deterred from even using the application.

Usability is the capability of the project to be understood, learned, used and attractive to the user, when used under specified conditions. The simple user interface will allow users to easily understand how to use the application, resulting in a lower learning curve. The ease of learning in using our application will be attractive to users and it is hassle-free which does not require signing up, which is especially attractive to users who do not wish to sign up.

Maintainability is the capability of the project to be modified. Modifications may include corrections, improvements or adaptations of the software to changes in the environment and in the requirements and functional specifications. This ensures that the application can be updated accordingly whenever the shop at the particular location closes, which constantly keeps our application up to date with the latest destination information.

5.2.1. Standard Metrics

The following standard metrics are the minimum planned metrics that will be collected, reported, and maintained in the area of software quality assurance:

- Fan in/Fan out
- Length of code
- Cyclomatic complexity
- Depth of conditional nesting
- Maintenance Quality
- Velocity

6. Software Reviews

6.1. Purpose

This section identifies the number and type of system/subsystem reviews and engineering peer reviews that will be supported by the SQ Personnel. The project milestone chart, and the SQ Personnel resource levels determine the reviews that are supported.

The objective is to have productivity improved in the development team, making testing process and time cost effective. And hope to achieve minimal defects in the end product.

6.2. Minimum Software Reviews

For each review, SQ will assess the review products to assure that review packages are being developed according to the specified criteria, the review content is complete, accurate, and of sufficient detail, and Requests for Action are captured, reviewed, and tracked to closure. In addition, SQ will assess the processes used to conduct the reviews to determine if appropriate personnel are in attendance, correct information is presented, entry and exit criteria are met, and appropriate documents are identified for update.

The following software reviews will be assessed by SQ:

- **Project Plan Review**
    - Software Specification Review
        - Ensuring the SRS performance requirements are practical, complete and consistent with the higher level specification requirements
        - Ensure all derived requirements are identified and documented
        - Ensure requirements stated in SRS are testable and measurable
        - Ensure there are complete verifiable requirements for all performance requirements
        - Evaluate agreements on interfaces and boundaries
        - Evaluate results of functional analyses
        - Evaluate identified software risks and proposed mitigation method
        - Evaluate applicable design constraints and human factor considerations
- **Requirements Analysis Review**
    - Review study and refining system, hardware or software requirements
    - Determining whether the above stated requirements are met
- **Software Design Review**
    - Preliminary Design Review

        - Ensure that the software requirements are reflected in the software architecture
        - Specify whether effective modularity is achieved
        - Define interfaces for modules and external system elements
        - Ensure maintainability has been considered
        - To access the quality factor
    - Critical Design Review
        - Assure that there are no defects in the technical and conceptual designs
        - Verify that the design being reviewed satisfies the design requirements established in the architectural design specifications
        - Assess the functionality and maturity of the design critically
        - Justify the design to the outsiders so that the technical design is more clear, effective and easy to understand
    - Program Design Review
        - Assure that the feasibility of the detailed design
        - Assure that the interface is consistent with the architectural design
        - Specify whether the design is compatible to implementation language
        - Ensure that structured programming constructs are used throughout
        - Ensure that the implementation team is able to understand the proposed design
- **Test Plan Review**
- **Acceptance Review**
- **Project Closing Review**
    - Ensure all requirements from the project scope document have been met

- External review after final delivery
- Gather all documents, saving them as data to use for future references

7. Test

SQ personnel will assure that the test management processes and products are being implemented per Test Plan. This includes all types of testing of software system components as described in the test plan, specifically during integration testing (verification) and acceptance testing (validation). SQ personnel will monitor testing efforts to assure that test schedules are adhered to and maintained to reflect an accurate progression of the testing activities. SQ will assure that tests are conducted using approved test procedures and appropriate test tools, and that test anomalies are identified, documented, addressed, and tracked to closure. In addition, SQ will assure that assumptions, constraints, and test results are accurately recorded to substantiate the requirements verification/validation status. SQ personnel will review post-test execution related artifacts including test reports, test results, problem reports, updated requirements verification matrices, etc.

- **Unit Testing**

  Code will be done on an individual unit to test that it is functioning correctly. White box testing is used, this technique is easy to find loopholes in the design of an application. This testing helps ensure proper operation of a module because tests are generated with knowledge of the internal workings of the module.

- **Integration Testing**

  This is done after unit testing is completed. Integration testing is to take all tested individual modules, integrate them, test them again, and develop the software that is according to design specification. Units validated during unit testing are combined to form a subsystem, then integration testing is done to ensure all modules work properly as per the user requirements.

- **System testing**

  The system is tested against non-functional requirements such as accuracy, reliability and speed. The main reason is to check that functional design specifications are met and how integrated modules work together. Association between objects, control and infrastructure, and the compatibility of the earlier released software versions with new versions are tested.

8. Problem Reporting and Corrective Action

SQ personnel generate, track, and trend assessment findings and observations in a centralized Reporting and Corrective Action System, has been implemented on a Kanban Board hosted on Github Project Board. Any problem identified during QC reviews or QA audits will be posted on the board in full view of the QA and Development team. Tasks on this board will be assigned the appropriate team/team member to resolve the issue.

The team follows a set-up for efficient problem reporting and corrective action

1. **Identifying the problem**

   a. Problem is defined clearing and has a detailed description of it
   b. Problem is not assigned to and SQ personnel
   c. Priority of problem should be mentioned at the beginning
2. **In progress**
   a. Assign problem to a SQ personnel
   b. SQ personnel should constantly update the team whenever progress has been made, this whole process should be transparent and known to everyone in the team.
3. **Reviewing solution**
   a. When a solution is ready, inform the team.
   b. After review, if the problem is solved, flag the problem as completed. Else the problem continues as in progress.
4. **Completed**
   a. Contains a list of problems that is resolved.

9. Tools, Techniques and Methodologies

SQ personnel will require access to the following:

9.1. Software Quality Tools

- Microsoft Office tools (i.e., Word, Excel, and PowerPoint)
- MediaWiki
- GitHub
- Google Docs
- Google Slides
- Draw.io
- React Testing - FrontEnd testing

10. Media Control

SQ deliverables will be documented in one of the following Google software applications: Google Docs, Google sheets and Google Drive. Deliverables will be in soft copy, with the exception of completed checklists from process and product assessments. See Section 12 for additional details on the collection and retention of key records. Software Quality personnel will request space on the project's secured server for SQ records. This server is password protected and backed up nightly.

1. Google Software Application

   SQ deliverables will be documented in the following Google Docs (Google Drive). Deliverables will be in softcopy except for completed checklists from process and product assessments. Google Docs tracks the version history and all changes to the documents

2. MediaWiki

   SQ personnel will request space on the project's MediaWiki for SQ records. This MediaWiki is password protected and backed up every night to ensure disaster management and discrepancies in the documents

11. Supplier Control

Supplier Control not applicable for this project

12. Record Collection, Maintenance, and Retention

SQ personnel will maintain records that document assessments performed on the project. Maintaining these records will provide objective evidence and traceability of assessments performed throughout the project's life cycle. There are two types of records that will be maintained: Hardcopy and Electronic. SQ personnel will maintain electronic or hard copies of all assessment reports and findings. SQ Project folders will contain hardcopies of the assessment work products such as completed checklists, supporting objective evidence, and notes.

The table below identifies the record types that will be collected, as well as the Record Custodian and Retention period

Record Title	**Record Custodian**	**Record Retention**
SQA Assessments	SQ Personnel	One Year
SQA Checklists	SQ Personnel	One Year
Deliverable Defects	SQ Personnel	One Year

13. Training

SQ personnel have fundamental knowledge in the following areas through prior experience, training, or certification in methodologies, processes, and standards:

- Audits and Reviews (Assessments)
- Risk Management
- Software Assurance
- Configuration Management
- Software Engineering
- ISO 9001, ISO 9000-3
- CMMI
- Verification and Validation

14. Risk Management

SQ personnel will assess the project's risk management process and participate in bi-weekly risk management meetings and report any software risks to the QAM and the project manager.

- Identifying risk
- Qualitative and quantitative risk analysis
- Risk response

15. SQA Plan Change Procedure and History

SQ personnel are responsible for the maintenance of this plan. It is expected that this plan will be updated throughout the life cycle to reflect any changes in support levels and SQ activities. Proposed changes shall be submitted to the Quality Assurance Manager (QAM), along with supportive material justifying the proposed change.