Software Quality Assurance (SQA) Plan

By Kim Seo Jin

Team Wanderers

Date: 13 February 2025

Signature Page

Prepared by: Kim Seo Jin	**Date: 12 February 2025**
Reviewed by 1: Isaac Chun Jun Heng	**Date: 18 February 2025**
Reviewed by 2: Yu Zi Hao Albert	**Date: 18 February 2025**
Approved by: Isaac Chun Jun Heng	**Date: 18 February 2025**

Document Change Record

Revision	Description of Change	Approved by	Date
0.10	**Initial Template**	**Kim Seo Jin**	**12/02/25**
0.20	**Formatted heading indents and refined all sections**	**Isaac Chun Jun Heng**	**18/02/25**
0.30	**Modified software tools in scope and SQ tools**	**Yu Zi Hao Albert**	**19/02/25**

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 **Wanderers** 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 to identify any errors, omissions, inconsistencies, and alternatives, enhancements or improvements that can be made at any stage of development.

Wanderers is a user-friendly collaborative travel planning platform to streamline itinerary planning, group collaboration and budgeting processes. The software items included in this plan encompass all components of the application.

The table below describes suppliers of the software, services and frameworks we would use and their intended use:

Item	**Intended Usage**
Next.js	Front-end user interface for responsive and interactive user experience
Express.js	Backend framework to ensure efficient API handling and seamless data management.
Supabase	Database used for synchronization and secure data storage
Nodemailer	Sending of emails during the instances of account management and collaboration between users

Google Maps/Places API	Acquire details and labels of a particular location, while giving auto complete queries to typed locations by user.
Git	Version control and release management of Wanderers
Vercel	Deployment of front-end for preview per commit push to staging and production environments

The intended use of this application is to offer users a convenient, simplified travel planning platform with features such as collaborative editing and transparent budget splitting.

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)
- Project Plan
- System Requirement Specifications
- Express.js (https://devdocs.io/express/)
- Next.js (https://nextjs.org/docs)
- Supabase (https://supabase.com/docs)
- Nodemailer (https://nodemailer.com)
- Git branching strategies (https://medium.com/@selvamraju007/git-branching-strategies-a6eafe4541ae)
- Google Maps API Docs (https://developers.google.com/maps/documentation)
- Vercel (https://vercel.com/docs/deployments/overview)

3. Management

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

3.1. Management Organisation

The implementation of 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. Purpose

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 the application must be user-friendly and responsive, ensuring ease of use and optimal performance. Each section of the application should provide a clear layout with a minimal learning curve. The navigation system should be intuitive, allowing the user to easily switch between different pages. Elements such as buttons, input fields and interactive components must be well-placed and visually accessible for the users, following the requirements specified in the requirements section of our documents.
- **Itinerary Planning Page**
  - The itinerary planner must be intuitive, dynamic and allow collaboration between the users. Users should be able to add and edit the trip details effortlessly. Upon such updates, these changes should be reflected to other users with minimal delay to ensure a smooth and collaborative process.
- **Budget Management Page**
  - The budget management page must provide a clear cost breakdown for individual and shared expenses, ensuring

transparency. It should do the calculations automatically and split among the users according to user preferences.

- **Discussion Page**
  - The discussion page serves as a centralized discussion hub for users to coordinate plans without switching to external messaging apps. Users should be able to send messages, share links and attach images to facilitate planning within the application effortlessly without any other setup done by the user. It should also support multiple languages in chat.

3.2.2 Process Assessments

The following process assessments will be conducted by SQ personnel:

- Requirement Management Process
  - The SQ personnel are responsible for defining, documenting and validating both functional and non-functional requirements of Wanderers. Frequent team meetings and reviews will ensure the requirements to be aligned to the user needs. Any changes in the requirement must be tracked and updated in JIRA to maintain clarity and consistency.
- Change Management Process
  - Any modification to team composition, technology stack or development procedures must be evaluated for feasibility and potential impact. SQ personnel are responsible for preventing unexpected issues and aligning unexpected issues and aligning with project goals. Changes and updates must be documented and communicated to all stakeholders to ensure a smooth transition. The changelog in the repository should be updated and an architectural design record must be made to reflect this change if the change regards a major change in the process (i.e we are upgrading to Next.js version XX)
- Maintainability Management Process
  - Wanderers have to be maintainable with good code quality. The codebase should be clean, modular, and well-documented to ensure readability and ease the debugging process in the future. The SQ personnel will be in charge of setting the baseline standards for maintainability, ensuring the best practices such as automated testing and version control. In addition, static code analysis and code linting will be added in the continuous integration check to ensure that all existing and new code meets current industry standards.

- Risk Management Process
    - SQ Personnel will oversee risk identification, assessment, and mitigation strategies to ensure system reliability. The key risks to be identified include failures in functionality, security vulnerabilities, or performance bottlenecks that affect user experience. Through proactive measures and contingency plans, the SQ team will aim to minimize risks and maintain reliability and performance of the application. Other than these major risks, risks to project timelines have to be identified early so that proactive action can be taken – either by crashing the timeline or quick re-priorisation of tasks.

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

- Project Proposal
- Project Plan
- System Requirement Specifications
- Risk Management Plan
- Configurations Management Plan
- Release Plan
- Test Plan
- Test Cases, and Test Coverage Report
- Change Management Plan
- Design Report on Software Maintainability

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.

- **Functionality**
  a. Functionality is the key as the application aims to provide all-in-one collaborative travel platform that requires fast and efficient updates, live chatrooms, and automated cost sharing systems. This aspect ensures all aspects of the trip planning are streamlined and efficient.
- **Usability**
  a. Usability is crucial as the platform's main focus is on the ease of use, ensuring that users can intuitively plan trips, modify itineraries, and track expenses without requiring technical expertise
- **Maintainability**
  a. Maintainability is essential for Wanderers as the application be a dynamic platform with potential updates and new features in the future, hence this would ensure code is clean, modular and well-documented such that the developers can easily modify and fix bugs in the future.
- **Reliability**
  a. Reliability is important for Wanderers, as this ensures the platform to remain stable and function as expected under various conditions. The application must allow users to access, update, and retrieve their travel plans without crashes, errors, or data inconsistencies.

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:

- Mean Time Between Failures
- Error Rate
- User Satisfaction Rating
- Bug Resolution Time
- Test Case Pass Rate
- Release Frequency

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.

6.1. 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
- Requirements Analysis Review
- Software Design Review
- Test Plan Review
- Acceptance Review

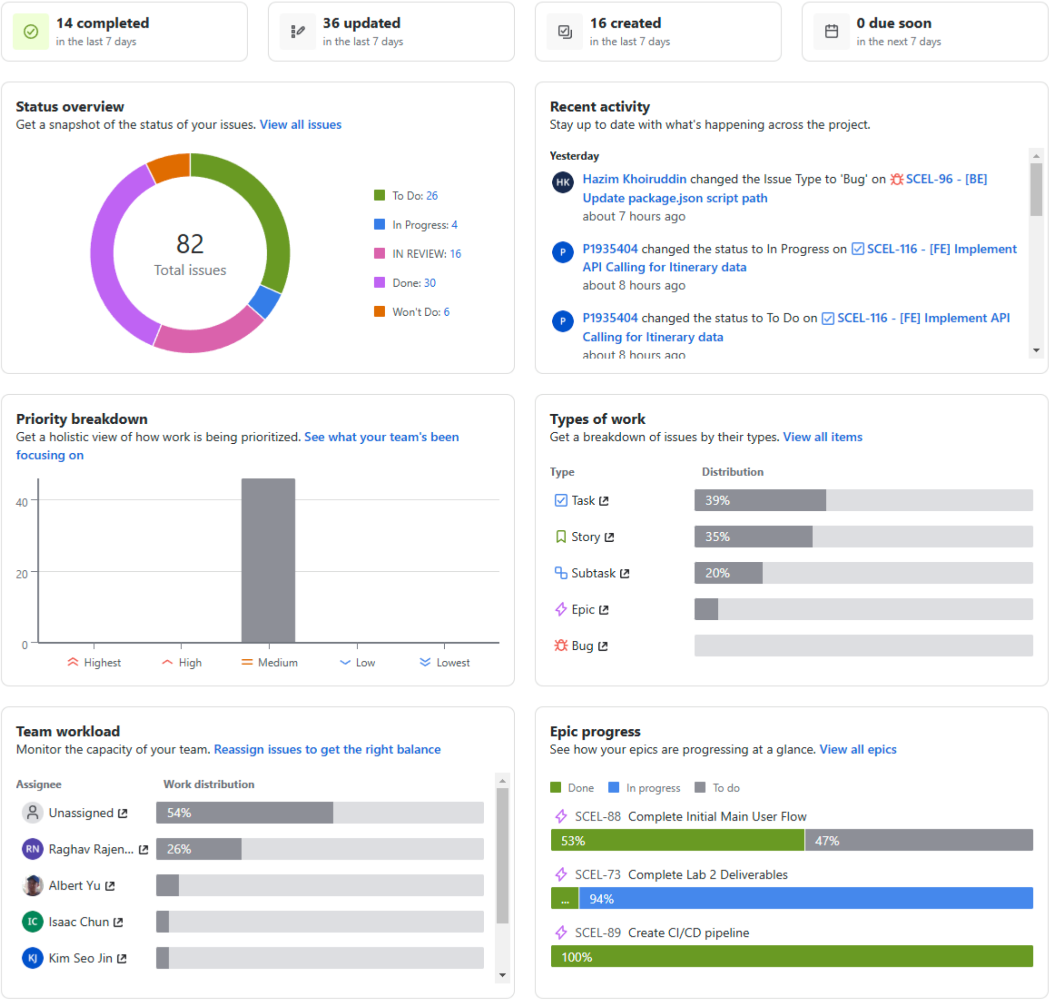
7. Test

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.

Additionally, the constant updating and classifying of tests based on their types (smoke, regression, integration) based on capabilities (front-end, back-end, activity, itinerary) should be documented so that it is easier to identify which component/capability causes the most errors, is the most stable.

8. Problem Reporting and Corrective Action

SQ personnel generate, track, and trend assessment findings and observations in JIRA, our centralized Reporting and Corrective Action System. JIRA ensures real-time visibility into project progress and issue resolution. Each identified issue is logged as a ticket in JIRA, including detailed descriptions, severity levels, assignees, due dates, and comments for effective tracking. The workflow follows structured columns such as "To Do," "In Progress," and "Done," providing a visual representation of the issue lifecycle. Automated workflows within JIRA streamline issue tracking and resolution, ensuring that bugs, technical debts, and process inefficiencies are promptly assigned and escalated. Notifications and alerts are triggered based on predefined conditions, ensuring stakeholders remain informed about updates. To maintain transparency and accountability, corrective action status and assessment data are communicated to the Quality Assurance Manager (QAM) and Project Manager through various channels. These channels include Bi-weekly Sprint, where issue trends, blockers, and resolutions are discussed. JIRA also provides a dashboard and report on the amount of bugs/issues that were raised, with other alternatives such as burndown charts for determining how fast the team handles bug or urgent tickets that require immediate action.



14 completed in the last 7 days

36 updated in the last 7 days

16 created in the last 7 days

0 due soon in the next 7 days

Status overview

Get a snapshot of the status of your issues. View all issues

Recent activity

Stay up to date with what's happening across the project.

Yesterday

Hazim Khoiruddin changed the Issue Type to 'Bug' on SCEL-96 - [BE] Update package.json script path
about 7 hours ago

P1935404 changed the status to In Progress on ☑ SCEL-116 - [FE] Implement API Calling for Itinerary data
about 8 hours ago

P1935404 changed the status to To Do on ☑ SCEL-116 - [FE] Implement API Calling for Itinerary data
about 8 hours ago

Priority breakdown

Get a holistic view of how work is being prioritized. See what your team's been focusing on

Types of work

Get a breakdown of issues by their types. View all items

Type	Distribution
Task	39%
Story	35%
Subtask	20%
Epic	
Bug	

Team workload

Monitor the capacity of your team. Reassign issues to get the right balance

Assignee	Work distribution
Unassigned	54%
Raghav Rajen...	26%
Albert Yu	
Isaac Chun	
Kim Seo Jin	

Epic progress

See how your epics are progressing at a glance. View all epics

SCEL-88 Complete Initial Main User Flow
53% 47%

SCEL-73 Complete Lab 2 Deliverables
... 94%

SCEL-89 Create CI/CD pipeline
100%

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)
- Version Control Systems (i.e., Git)
- React Developer Tools (https://chromewebstore.google.com/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi?hl=en) for checking console logs and ensuring correct interactions from end to end.
- Jira (Create bug ticket for developers to act on)

10. Media Control

SQ deliverables will be documented in one of the following Microsoft software applications: Word, Excel, or PowerPoint. All documentation and deliverables will be in soft copy/digital format. Completed checklists from process and product assessments would be stored securely in NTU OneDrive and Confluence Wiki linked to our JIRA board as a form of backup, which can then be tagged to a particular ticket. See Section 12 for additional details on the collection and retention of key records. Software Quality personnel will ensure that access to our shared folder in OneDrive is limited to only individuals that has an NTU verified email and is a member of the team. Regular backups of data from OneDrive will be transferred onto Confluence Wiki would also be done weekly to ensure data integrity and availability in the event of outage of service.

Configuration management would be tied towards our release version of the software, (i.e Wanderers has an upcoming release v0.0.4, then our documentation and deliverables for software quality should follow the format **SQA_Checklist_v0.0.4.x**, where x is the number of times we run a software quality check on a particular version). This way, it is easy to tie particular checks to a release and identify the number of times we have done a check on a release.

11. 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

12. 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

13. Risk Management

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

14. 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.