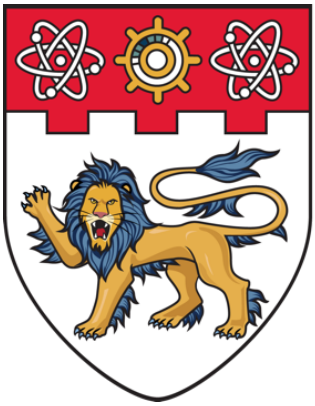

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Software Quality Assurance Plan (SQA) for TimeWise

Team 0
Lab group : SSP7

Date: February, 2020

Group members	**Roles**
Mantri Raghav	Project Manager
Dwivedee Lakshyajeet	Development Lead
Harding James	Back-End Developer
Alex Bernini	Front-End Developer
Mittal Madhav	Release Manager
Xue Xueting	QA Engineer
Koh Hui Ling	QA Engineer
Lek Jie Ling	QA Manager

Signature Page

	Names	Date
Prepared By	James Harding, Lek Jie Ling, Alex Bernini	16/02/2020
Reviewed By 1	Lakshyajeet Dwivedee	17/02/2020
Reviewed By 2	Madhav Mittal	17/02/2020
Approved By	Raghav Mantri	18/02/2020

Document Change Record

Revision	Description of Change	Approved by	Date
1.0	Change formatting	Dwivedee Lakshyajeet	13/2/2020
1.1	Purpose to Management	James Harding	14/2/2020
1.2	Documents to Standards	Lek Jie Ling	15/2/2020
1.3	Software Reviews onwards	Alex Bernini	16/2/2020
1.4	Fix small errors	Raghav Mantri	18/2/2020

Table of Contents

Purpose and Scope

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 TimeWise 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.

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.

The TimeWise application will automatically scrape the student's course schedule from NTU's STARS system and display it in a 7-day to-do list format. The user will also be allowed to add their own tasks. During the addition of a new task, the app will recommend 5 empty 1-hour slots to the user during which they can do the task. Notifications will be sent to the user 30 minutes before their scheduled task.

The team has 8 members each with assigned roles and some with multiple roles. We have 3 people on backend development 1 of whom is dedicated to this specific task. There are also 3 people on front end development, 1 of which is dedicated to this task. Then we have 2 quality assurance engineers. Finally, we have the respective managers and leaders being quality assurance manager, release manager, project manager and lead developer. The team will be responsible for developing and documenting the project, as well as ensuring the product meets sufficient standards.

Tests will be conducted on the relevant subsystems of the project, those being, the UI pages, the database, the internal logic (such as when data is added to the database after being retrieved via the web scraper) and the web scraper.

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 (https://docs.google.com/document/d/1INIxE1mOqsPpd9vlP6F7dyleiIWRtQ9yCmaalFz3hec/edit?usp=sharing)
- SRS(https://docs.google.com/document/d/1nNQvd9Vt3bUoiSarviE7sclgevVap5TTrTvlUom0RsI/edit?usp=sharing)
- React documentation (https://reactjs.org/docs/getting-started.html)
- Node.js documentation (https://nodejs.org/api/index.html)
- MongoDB documentation (https://docs.mongodb.com/)
- Github documentation (https://help.github.com/en/github)
- 1008-1987-IEEE Standard for Software Unit Testing (https://standards.ieee.org/standard/1008-1987.html)
- 829-2008 - IEEE Standard for Software and System Test Documentation (https://ieeexplore.ieee.org/document/4578383)
- Learning node.js by Alex Zanfir (https://www.linkedin.com/learning/learning-node-js-2/)
- Kanban board (https://www.atlassian.com/agile/kanban/boards)

Management

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

Management Organisation

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

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

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.

Tasks

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

1. Product Assessments

The following product assessments will be conducted by SQ personnel:

- Web scraper
- UI pages
- Database system
- Recommendation component
- Task adding/editing/deleting component
- Notification component

2. Process Assessments

The following process assessments will be conducted by SQ personnel:

- Requirement Management Process
- Change Management Process
- Maintainability Management Process
- Risk Management Process

Roles and Responsibilities

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

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

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

Documents

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.

Minimum Document Requirements

- System Requirement Specifications
- Project proposal
- Quality plan
- Project plan
- Risk management
- Test plan
- Test cases and test coverage report
- Release plan
- Change management plan
- Configuration management plan
- Design report on software maintainability

Standards, Practices, Conventions and Metrics

Purpose

This section highlights the standards, practices, quality requirements, and metrics to be applied to ensure a successful software quality program. The purpose of following a comprehensive set of software standard testing execution is to ensure that there would be minimal errors experienced by the user and to ensure smooth deployment when various parts of the code is changed and updated.

Software Quality Programme

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

1. **Functionality** - Functionality is necessary for the web application. All processes must function as expected by the user and the admins without errors and the data processed and stored must be accurate.
2. **Simplicity** - Simplicity is an important quality for the web application. A cluttered interface would be distracting and might directly lead the users to the incorrect modules. The application should be easy to use without prompting the user. This will greatly improve user satisfaction.
3. **Reliability** - The application should be up, running 24/7 with minimal downtimes especially during peak hours and heavy traffic.
4. **Portability** - The application shall be carried to both platforms, iOS and android.

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 to failure
- Defect density
- Problems per User-Month during UAT
- Defect removal effectiveness
- Fix backlog and backlog management index
- Percent delinquent fixes
- Fix quality

Software Reviews

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.

Software reviews are essential for assuring code submitted by members within the team meet a coding standard - for example have appropriate commenting, low coupling, high cohesion etc.

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
- Code Review (GitHub Pull Request Reviews)

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.

On top of this, unit testing will be used, primarily for the backend of the application. Test Driven Development (TDD) will be followed - building a test suite, and then implementing code to pass the tests. SQ personnel will be responsible for running tests and tracking what is passing, failing, and yet to be implemented.

Problem Reporting and Corrective Action

SQ team collects and analyzes the QA activity metrics as well as assigns, tracks and verifies corrective actions resulting from audits, evaluation and monitoring in a centralized Reporting and Corrective Action System implemented on a kanban board system using GitHub Projects Boards. The lists set up are

1. Problems/Issues
    a. Issues unassigned to any SQ personnel
    b. Detailed descriptions of issue with a tracking label or tag
    c. Priority of the issue
2. In progress
    a. Current tasks by different SQ personnels are displayed so the others can monitor and be aware of the progress in various parts of the project
3. Review
    a. Issues ready to be reviewed by other SQ personnels to check on quality
    b. If issue is resolved, it would be moved to “Completed” list, if it needs more work, it would be moved to “Progress” list
4. Completed

a. Resolved issues

Tools, Techniques and Methodologies

SQ personnel will require access to the following:

Software Quality Tools

- GitHub Project Boards
- React Native Testing Library - front-end testing
- Chai/Mocha Testing Suite - backend-testing
- Google Docs

Techniques and Methodologies

The primary methodology to be followed when developing the application is Test Driven Development (TDD). This involves writing tests of what we want a certain module of the software to do, then implementing the code to make the tests pass.

Media Control

SQ deliverables will be documented in one of the following Google software applications: Google Docs and Google Sheets. Deliverables will be in soft copy, with the exception of completed checklists from process and product assessments. Google Docs track the document history and version changes. Software Quality personnel will also request space on the project's MediaWiki for SQ records. This MediaWiki is password protected and backed up nightly.

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	6 Months
SQA Checklists	SQ Personnel	6 months
Deliverable Defects	SQ Personnel	6 months

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

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.

To mitigate and prevent risks, we adopt the Scrum model, suggesting that the project progresses via a series of sprints. Each sprint is a sub-task of the entire project. During the sprint, the team will plan, build, test, and review a set of features. Each sprint will be 2 weeks long. At the end of each sprint, a sprint review will be conducted, to demonstrate the new

functionality to the product owner. The product owner will then provide feedback on what could be done in the following sprint, to improve the product.

This will help in reducing the risk of:

- Producing a product that does not match the users' needs
- The project going over the time/budget constraints

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.