



SOFTWARE QUALITY ASSURANCE (SQA) PLAN

*Version 2.0
13/2/2021*

Team Xeon

Document Number: SL-04

REVISION HISTORY

[illegible]

Signature Page

Prepared by: Teo Boon Shuan **Date:** 10 Feb 2021

Reviewed by 1: Kong Hou Jing **Date:** 11 Feb 2021

Reviewed by 2: Wilson Tai **Date:** 12 Feb 2021

Approved by : Kenny Voo Tze Rung **Date:** 13 Feb 2021

Contents

Purpose and Scope	5
Purpose	5
Scope	5
Reference Documents	6
Management	6
Management Organisation	6
Project Management	6
Assurance Management	6
Tasks	7
Product Assessments	7
Process Assessments	9
Roles and Responsibilities	9
QAM	9
Software Quality Personnel	10
Documents	10
Purpose	10
Minimum Document Requirements	10
Standards, Practices, Conventions and Metrics	10
Purpose	10
Standards	10
Practices	11
Software Quality Programme	11
Standard Metrics	11
Software Reviews	12
Purpose	12
Minimum Software Reviews	12
Test	12
Problem Reporting and Corrective Action	13
Tools, Techniques and Methodologies	14
Software Quality Tools	14
Media Control	14
Record Collection, Maintenance, and Retention	15
Training	15

Risk Management	16
SQA Plan Change Procedure and History	16

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 SmartLib 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, SmartLib 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.

Maintenance of the plan will be carried out throughout the project if needed. QAM will review and update the plan at least once per month, if needed, and it will be reviewed by the project manager.

SmartLib is an intelligent platform that allows users - library users and students - to check the current available capacity of library seats. Its purpose is to locate and assign empty seats to users so that time is not wasted when users roam the library looking for an empty seat. Smartlib also aims to mitigate seat hogging, so that all users have a fair opportunity to exercise their access to the library.

This SQA plan covers the following primary features of the software, namely:

1. **Login/Logout** - This feature allows the user to log into the application to use its features. User identity is authenticated with the user records in the database. A timestamp is saved upon login.
2. **Image Recognition** - This feature captures the presence of each library user when they are occupying their labelled seats.
3. **Live Seat Status Update** - Together with image recognition, this dashboard will constantly update live, on the availability of each seat in the library, by detecting if the seat is occupied by someone present or not.
4. **Seat Reservation**- This feature allows the user to reserve a selected empty seat in the library for up to 15 minutes.
5. **Seat Hogging Warning** - If a seat is occupied with personal belongings for 60 minutes, an alert will be sent to the staff to handle the situation.

2. Reference Documents

- ISO 9001-2015 Quality Management Systems - Requirements
(<https://www.iso.org/standard/62085.html>)
- ISO/IEC 25010 Software and Quality Standard
([ISO 25010 \(iso25000.com\)](https://www.iso.org/standard/55991.html))
- Project Plan
- System Requirement Specifications
- Project Proposal

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 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:

- Project Proposal
- System Requirements Specification (SRS)
- Project Plan
- Risk Management Plan
- Initial Prototype
- Test Plan
- Configuration Management Plan
- Change Management Plan
- Release plan
- Login/Logout
- Booking system
- Detection system
- Alert system

Project proposal – The project proposal should be well-defined and addresses the concerns of the client. The solution proposed by the project team should fulfill all functional and non-functional requirements specified by the client. In addition, all deliverables stated in the “Deliverables Agreement” as well as project management and estimated costs of the project should also be defined clearly in the project proposal.

System Requirement Specification – The SRS should be developed according to the Requirements Management Process defined by the project team. The specification should specify how the system should behave, defined at a high-level what are the main business processes that will be supported, what simplifying assumptions have been made and what key performance parameters will need to be met by the system.

Project Plan – The Project Plan should be developed according to the Project Planning Process defined by the project team. The Project Plan should provide a brief overview of the project, its goals, its objectives, the resources needed, associated budget as well as a work breakdown structure.

Risk Management Plan – The Risk Management Plan should be developed according to the Risk Management Process defined by the project team. The plan should identify and analyse likely risks with both high and low impact, as well as responses to these risks throughout the Software Development Life Cycle (SDLC)

Initial Prototype – The initial prototype should be developed according to the Software Development Process defined by the project team. The prototype should be able to demonstrate functionalities as specified in the “Deliverables Agreement” under the section “Prototype” as agreed upon between the client and the executive committee.

Test Plan – The Test Plan should be developed according to the Test Management Process as defined by the Quality Assurance team. The plan should describe the scope, approach, resources and schedule of intended test activities. It identifies amongst other test items, the features to be tested, the testing tasks, party responsible for each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning.

Configuration Management Plan - The Configuration Management Plan should be developed according to the Configuration Management Process defined by the project team. The plan should document and inform project stakeholders about Configuration Management within a project, what Configuration Management tools will be used, and how they will be applied by the project. The Configuration Management Plan details the methodology that the Project Manager (PM) and lead developer will use to control program documentations and the program baseline (Technical, Functional and Allocated).

Change Management Plan – The Change Management Plan should be developed according to the Change Management process as defined by the project team. The change management plan should define activities and roles to manage and control change during the execute and control stage of the project. Change should be measured against the project baseline.

Release Plan – The Release Plan should be developed according to the Release Management Process as defined by the Release Manager. The Release plan should clearly define the release requirements, release criteria, release goals as well as a release schedule overview.

Login/Logout - The login logout component should allow the system to authenticate the user and allow the user to have certain permission either student or admin privileges and should provide all functionalities for its functional requirements specified in the System Requirement Specification.

Booking System - The booking system should provide all functionalities for its functional requirements specified in the System Requirement Specification.

Detection System - The detection system uses a camera to identify humans and objects and should provide all functionalities for its functional requirements specified in the System Requirement Specification.

Alert System - The alert system should be able to alert the liberia if a seat is hog and should provide all functionalities for its functional requirements specified in the System Requirement Specification.

3.2.2. Process Assessments

The following process assessments will be conducted by SQ personnel:

- Requirement Management
- Project Management
- Project Monitoring and Control
- Risk Management
- Test Management
- Software Configuration Management
- Release Management
- Software Development

The following processes listed above will be assessed by SQ personnel during scheduled internal audits made known beforehand to parties involved as well as software reviews attended by the SQ personnel. The processes will be assessed using a standardized rubric which is the ISO 9001-2015 Process Audit Checklist, which can be found in MediaWiki under “Lab 2 Deliverables”. The assessments will then be reviewed by the QA Manager and QA team.

An issue will be logged if there is a problem without a visible plan for resolution. Once a list of issues has been compiled, it will be reviewed with the project manager to see if any new or additional information might mitigate or eliminate any of them. Issues identified as well as the recommended corrective actions to be taken will be handled as specified in section 8 of this document titled “Problem Reporting and Corrective Action”.

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

- Software Requirement Specification
- Quality Plan
- Project Plan
- User Documentation
- Source Code Documentation
- Risk Management Plan
- Design report on software maintainability
- Configuration Management Plan
- Change Management Plan
- Release Plan
- Test Cases and Requirements Test Coverage Report

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. Standards

We will be adopting the ISO 25010 and ISO 12207 in this project. The ISO 225010 is an international standard for the evaluation of software quality based on the 8 quality, Functionality, Efficiency, Compatibility, Usability, Reliability,

Security , Maintainability and Portability. The ISO 12207 is an international standard for software life cycle processes.

5.3. Practices

The team will be adopting Waterfall Model for the software development project. The project will be done sequentially in phases from requirement specifications, design, implementation, testing to maintenance.

The Architecture model the team will adopt for this project is the client and server architecture with the database being at the server side. The database have CRUD function as we grant the users the ability to create, read, update and delete the cases that they have access to.

5.4. Conventions

The team will be following the standard naming convention when naming the classes, methods and variables of the software (eg. camelCase, PascalCase or snake_case). The software will be written in standardized format and indentation and all the class methods will be commented/documented properly by the team.

5.5. Software Quality Programme

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

With reference to the ISO/IEC 25010 Quality Model, the four most important qualities for SmartLib are **Functionality**, **Reliability**, **Usability** and **Portability**.

Firstly, SmartLib must be able to perform all the predefined features and functionality completely, correctly and appropriately. The functionalities must meet implied needs as stated in the SRS under specified conditions.

Secondly, SmartLib has to be able to carry out the defined task reliably under normal conditions during the working hours. The application must be able to work seamlessly without any lagging or long period of downtime. In the event of interruption or failure, the system must be able to recover from it within a specified period of time.

Thirdly, SmartLib has to be able to meet the needs of our intended users, students and librarians. The interface of SmartLib has to be user friendly, so that users are able to learn and use all the functionalities within a specified period of time.

Lastly, SmartLib has to be able to work perfectly in different libraries in NTU. The system must be able to adapt to different hardware such as cameras. The SmartLib must be able to work effectively and efficiently in different operating systems and browsers.

5.5.1. 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:

- No. of Error Messages - Usability, Reliability

- Program Size (SLOC) - Reliability
- Effectiveness of Exception Handling - Reliability
- Functionality Check based on Specified Use Case - Functionality
- Number of Machine Dependent Calls - Portability
- Fan-in/Fan-out - Complexity of Overall App

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.

6.2.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
- Requirements Analysis Review
- Software Design Review
- Test Plan Review
- Acceptance Review
- Peer Reviews
- Code Reviews

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).

The tests will be conducted as follows:

1) Unit Testing

Tests will be conducted on individual units or components of the system. The purpose is to validate that each unit of the software code performs as expected.

2) Integration Testing

Tests will be conducted on a group of software modules that are integrated logically. The purpose is to expose defects in the interaction between these software modules when they are integrated

3) System Testing

Tests will be conducted on the complete and fully integrated software product. The purpose is to evaluate the end-to-end system specification, to ensure it meets the quality requirement.

4) User Acceptance Test (UAT)

Users will test the actual software in real-world scenarios. The purpose is to give users the chance to interact with the software and find out if the functionality of the system works as expected and not overlooked or miscommunicated.

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.

8. Problem Reporting and Corrective Action

There are some foreseeable problems which may arise in the course of the software development:

1. Documentation
 - 1.1. Incompleteness/ Lacking essential content of documentation
 - 1.2. Incorrect information in documentation
 - 1.3. Error in documentation
 - 1.4. Outdated documentation
 - 1.5. Formatting issues
2. Development
 - 2.1. Incorrect functionality
 - 2.2. Lack of functionality
 - 2.3. Inconsistency with planned design

SQ personnel generate, track, and trend assessment findings and observations in a centralized Reporting and Corrective Action System. In general, these approaches can be taken for corrective action:

1. SQ personnel will conduct regular and scheduled checks and reviews to discover and identify potential problems.
2. Upon discovering a problem with the project, SQ personnel or members will be required to inform the Project Manager and QA Manager.

3. Project Manager and QA Manager will review the problem and determine the severity of the problem
4. Depending on the severity of the problem, relevant members in the project group will be notified of the problem, and meetings will be held if necessary.
5. Corrective action will be determined through discussion.
6. The problem and corrective action to be taken will be documented down in the Problem Reporting and Corrective Action Document Excel Sheet located in the shared Google Drive project folder.
7. Team members will be appointed to implement the corrective action by a given deadline.
8. Once corrective actions have been implemented, SQ personnel/ SQ Manager will review the corrective actions. If it is not implemented correctly or not up to standard, the team will revert back to Step 3 for review again.
9. If corrective actions are satisfactory, the problem case will be closed and logged down in the Excel Sheet in the shared Google Drive project folder.

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)
- Google Drive (For collaborative work)
- Modern Web Browser (i.e. Google Chrome, Microsoft Edge, Safari)
- Appropriate IDE (i.e Visual Studio Code)

10. Media Control

SQ deliverables will be documented in one of the following Microsoft software applications: Word, Excel, or PowerPoint. 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.

The following services are used for this project for media control:

1. MediaWiki
2. Git and GitHub
3. Google Drive

MediaWiki is a popular free server-based software, which is powerful, scalable and feature-rich. It is easy to pick up and use even for beginners as it uses MediaWiki's

wikitext format, thus users can edit them easily without any knowledge of HTML or CSS. MediaWiki keeps track of all edits submitted, and it is also reversible to any previous versions. It also can manage images and multimedia files uploaded to the MediaWiki server.

Git is an open-sources version control system first developed in 2005. It is installed on your local system and also allows for branching. GitHub is a cloud-based Git repository hosting service, which allows sharing of codes, thus allowing for collaboration and making revisions and edits.

Google Drive is used for collaboration for all the documentations, as it has a built-in office suite (Google Docs, Google Sheets, etc), which also allows for real-time collaboration and version control. It is also used for file storage and file sharing.

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.

With regards to this Software Quality Plan, the SQ team has identified an additional risk that has not been accounted for - Time Mismanagement. As such, a risk assessment has been done to mitigate it.

Risk	Risk Category	Risk Probability	Project Impact	Trigger of Risk	Risk Strategy	Risk Response	Risk Zone	Risk Owner
Time Mismanagement	People	High	High	1. Underestimation of the complexity of the project 2. Overconfidence of capabilities	Mitigate	1. Conduct regular meetings to ensure timeliness 2. Project team may need to work overtime	High	Quality 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.