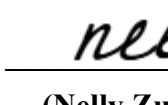
Software Quality Assurance (SQA) Plan
By Pawfection

Date: 22 September 2025

Signature Page

Prepared by:  **Date:_____19/9/25_____**

(Nelly Zulkiflee)

Reviewed by 1:  **Date:_____19/9/25___**

(Athena Choo)

Reviewed by 2:  **Date:_____23/9/25_____**

(Chloie Tan Yue Yun)

Approved by :  **Date:_____19/9/25_____**

(Chan Zi Jian)

Document Change Record

Revision	Description of Change	Approved by	Date
0.1	**Initial draft created**	**Nelly**	**10/09/25**
0.2	**Edited Management Section**	**Zi Jian**	**12/09/25**
0.3	**Refined Product & Process Assessments**	**Zi Jian**	**14/09/25**
0.4	**Updated Documents, Standard & Metrics**	**Zi Jian**	**15/09/25**
0.5	**Revised Reviews, Test & Corrective Action**	**Zi Jian**	**16/09/25**
0.6	**Polished Tools, Media Control & Records**	**Zi Jian**	**17/09/25**
0.7	**Finalized Training, Risk Management**	**Zi Jian**	**19/09/25**
1.0	**Internal Review feedback incorporated**	**Zi Jian**	**19/09/25**

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 *PawfectMatch* 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.

The *PawfectMatch* project involves the development of a responsive, web-based application that connects pet owners with service providers in Singapore. The system will allow providers to list their services, and pet owners to search, filter, compare, and book with confidence. Core features include user authentication, service listings, booking management, reviews and ratings, and user/provider profiles.

This quality plan applies to all software deliverables produced by the *PawfectMatch project team*, including source code (frontend, backend, and database), documentation (requirements, design, user guides, and technical manuals), test cases and results, as well as prototype and production-ready systems.

The application will be developed using modern web technologies — ** for the frontend, ** for the backend, and ** for the database — and will be accessible on both desktop and mobile browsers. The intended outcome is a secure, user-friendly, and scalable platform that improves convenience for pet owners, enhances visibility for providers, and contributes to the growth of Singapore's pet service ecosystem.

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
- 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), with oversight by the Project Manager (PM).

3.1.1. Project Management

The Project Manager (PM) 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:

- Register & Login Modules
  - Assess correctness, reliability and security of account creation and authentication functions including error handling and password complexity enforcement
- User and Pet Profile Management
  - Review Create/Edit User Profile and Pet Profile use cases for data completeness, validation accuracy and ease of modification
- Search Function
  - Verify efficiency, accuracy and response time of search results based on criteria such as experience, availability, rating and location
- Display Profile
  - Assess usability and clarify of profile details shown to Pet Owners and Caretakers

3.2.2. Process Assessments

The following process assessments will be conducted by SQ personnel:

- Requirement Management Process
  - Evaluate how system requirements (from SRS and use cases) are documented, traced and updated.
  - Ensure consistency between requirements, design and implementation
- Change Management Process
  - Assess the effectiveness of handling changes to requirements, design or code.
  - Verify that all changes are reviewed, approved and documented with minimal disruption to ongoing work
- Configuration Management Process
  - Review version control practices (Git/GitHub) to ensure that code, documents and artifacts are properly baselined, versioned and traceable.
- Agile Development Process
  - Assess sprint planning, backlog prioritization and JIRA tracking to confirm that tasks are completed on time, reviewed and integrated with quality checkpoints

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
- Project Proposal
- Use Case Model

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.

1. Efficiency
    - The software must make optimal use of resources (CPU, memory, response time) to deliver fast and reliable performance.
2. Maintainability
    - The software should be easy to modify, extend and fix with clear structure and documentation.
3. Reliability

    - The software must operate without failure under stated conditions, ensuring stability and correctness
4. Usability
    - The software should be intuitive, easy to learn and simple for end-users to operate effectively.

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:

- Cyclomatic Complexity
    - Measures control flow complexity (number of independent paths)
    - Higher values increases testing effort and reduce reliability
- Depth of Conditional Nesting
    - Captures the max depth of nested if or decision structures
    - Deep nesting increases error-proneness and reduces readability
- Length of Code (LOC)
    - Total lines of code in a component
    - Excessive LOC suggests inefficiency and maintainability challenges
- Fan-in / Fan-out
    - Measures module coupling by counting calls into/out a function
    - High coupling reduces reliability and complicates maintenance
- Mean Time to Failure (MTTF)
    - Average operational time before the software encounters a failure
    - Indicates stability and robustness of the system
- Fog Index (Documentation Readability)
    - Readability score of documentation based on sentence/word complexity
    - Ensures user manuals and guides are understandable by the intended audience

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

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.

8. Problem Reporting and Corrective Action

SQ personnel will generate, track, and trend assessment findings and observations in an online Reporting and Corrective Action System maintained in a shared Excel spreadsheet. Assessment data and corrective action status will be updated regularly in the spreadsheet to ensure visibility.

Assessment results and corrective action updates will be:

- Communicated to the QAM through weekly QA review meeting and direct access to the shared spreadsheet
- Reported to the PM in the form of a consolidated summary (e.g. weekly or bi-weekly status reports) highlighting open issues, corrective actions in progress and closure status.

This ensures that both the QAM and PM have timely visibility into quality issues and their resolution.

9. Tools, Techniques and Methodologies

SQ personnel will require access to the following:

9.1. Software Quality Tools

- Microsoft Office Suite (i.e., Word, Excel, and PowerPoint) – for documentation, reporting and corrective action tracking
- VSCode – for code review and verification activities
- GitHub – for version control, code reviews and configuration management
- JIRA – for task tracking, backlog management and sprint reviews
- Draw.io / Figma – for reviewing design diagrams and UI prototypes as part of quality assessments

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 maintain SQ records in the project's GitHub repository, which provides secured access, nightly backups through GitHub infrastructure and full version control. Configuration management of SQ deliverable (e.g. SQ Assessment Reports, Test Reports, Corrective Action Logs) will follow standard version naming conventions (e.g. QA_Report_v1.0, QA_Report_v1.1) to ensure traceability and integrity

11. 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. Electronic records will be stored in the project's GitHub repository to ensure secure access and version control.

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
- Verification and Validation

14. 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.

In addition, SQ personnel will:

- Monitor risks identified in the Risk Management document and ensure that mitigation actions are tracked to closure.
- Verify that risks related to software quality (e.g. missed requirements, inadequate test coverage, configuration control issues) are logged in JIRA and reviewed during sprint retrospectives.

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.