**Software Quality Assurance (SQA) Plan**

**By Team APT200**

**Date: 15 September 22**

**Signature Page**

**Prepared by: \_\_\_\_Lim Kai Sheng\_\_\_\_\_\_\_\_\_\_\_\_ Date: 08/09/22**

**(Name)**

**Reviewed by 1: \_\_\_Lee Wen Bin Andre\_\_\_\_\_\_\_\_ Date: 10/09/22  
 <(Reviewer Name)>**

**Reviewed by 2: \_\_\_\_\_Andrew Ng\_\_\_\_\_\_\_\_\_\_\_\_ Date: 15/09/22  
 <(Reviewer Name)>**

**Approved by : \_\_\_\_\_\_\_\_Ling Yin\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 (Name)**

**Document Change Record**

| **Revision** | **Description of Change** | **Approved**  **by** | **Date** |
| --- | --- | --- | --- |
| **1.0** | **Preliminary Version** | **Lim Kai Sheng** | **08/09/22** |
| **2.0** | **Content & Documentation Review** | **Andre Lee** | **10/09/22** |
| **3.0** | **Documentation Review** | **Ling Yin** | **21/09/22** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

[**1. Purpose and Scope**](#_heading=h.po8sum150fss) **5**

[1.1 Purpose](#_heading=h.sj5w7mkwjkma) 5

[1.2 Scope](#_heading=h.mn8jbnf4gpzv) 5

[**2. Reference Documents**](#_heading=h.mwzxcmuv2ux6) **6**

[**3. Management**](#_heading=h.grjpmfmz9uct) **7**

[3.1 Management Organization](#_heading=h.165b109xoo8n) 7

[3.1.1 Project Management](#_heading=h.9yygg9f497ox) 7

[3.1.2 Assurance Management](#_heading=h.5iwh7ccqq21) 8

[3.2 Tasks](#_heading=h.ldedqb4oqy7z) 8

[3.2.1 Product Assessments](#_heading=h.w9o6tutgixj4) 8

[3.2.2 Process Assessments](#_heading=h.sxsawdu7o4n4) 10

[3.3 Roles and Responsibilities](#_heading=h.wjjpyax61o5d) 10

[3.3.1 QAM](#_heading=h.b1p5zftuvtbe) 10

[3.3.2 Software Quality Personnel](#_heading=h.1pltudkkz2tt) 10

[4. Documents](#_heading=h.44sinio) **11**

[4.1 Purpose](#_heading=h.rikwbjhz74i5) 11

[4.2 Minimum Document Requirements](#_heading=h.i5o5j8kg3tu6) 11

[**5. Standards, Practices, Conventions and Metrics**](#_heading=h.k3mqn5571lk3) **12**

[5.1 Purpose](#_heading=h.16d788aapci7) 12

[5.2 Software Quality Programme](#_heading=h.bhigghcp5e70) 12

[5.2.1 Standard Metrics](#_heading=h.yo7goh1jlk3a) 12

[**6. Software Reviews**](#_heading=h.18ohmkdhwvka) **13**

[6.1 Purpose](#_heading=h.e78lzpr88nw6) 13

[6.2 Minimum Software Reviews](#_heading=h.oaj83wevhs2n) 13

[**7. Test**](#_heading=h.ufdgbilx1x82) **13**

[**8. Problem Reporting and Corrective Action**](#_heading=h.8lhfp6qas2te) **14**

[**9. Tools, Techniques and Methodologies**](#_heading=h.kywjbq1jxhfi) **15**

[Software Quality Tools](#_heading=h.cjrf2fexotid) 15

[**10, Media Control**](#_heading=h.vp1afoxdhxr4) **15**

[**11. Supplier Control**](#_heading=h.giqd3bad2w) **16**

[**12. Record Collection, Maintenance, and Retention**](#_heading=h.riwt8wucrhij) **16**

[13. Training](#_heading=h.3o7alnk) **16**

[**14. Risk Management**](#_heading=h.1l19li97ohrn) **17**

[**15. SQA Plan Change Procedure and History**](#_heading=h.wmnm1vrexgjw) **17**

# 

# 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 project MerchantDice.

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, MerchantDice, 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.

MerchantDice seeks to be a one stop platform where buyers and sellers can come together and exchange goods conveniently via an online space. This reduces the need to travel to a physical thrift store in order to be able to thrift shop. Thereby allowing our users to enjoy the benefits of thrifting while still being able to enjoy the convenience of eCommerce.

The software items covered by the SQA and their intended use are as follows:

1. Login/Logout: This feature allows the user to log into the application to gain access to its functionalities. The user is authenticated by checking the records in the user database.
2. Manage Products: This feature allows users who are sellers, to manipulate the product database such as create, read, update and delete products.
3. Explore Products: This feature allows users to view the product listing, as well as additional functionalities like search, sort and favorite products.
4. View Store: This feature allows the user to engage with the sellers regarding the selected product such as display reviews, as well as functionalities like write review and purchase product.
5. Contact Seller: This feature allows the buyer to chat with the seller through a chat screen. This is clear any clarifications the buyer has about the product and also to prevent any miscommunication from occurring.

# 

# 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

# 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 utilized
* 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
* Technical Implementation:
  + Frontend Components
  + Frontend Web Pages
  + Middleware Authentication
  + Backend API calls
  + Backend Database Management

**Project proposal** - The project proposal should be well-defined and address the client's concerns. 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, should 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, define at a high level 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, objectives, resources needed, associated budget, and 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 analyze 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 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, the party responsible for each task, the 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 the project will apply them. The Configuration Management Plan details the methodology that the Project Manager (PM) and the lead developer will use to control program documentation 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 execution 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 defined by the Release Manager. The Release plan should clearly define the release requirements, release criteria, and release goals, as well as a release schedule overview.

**Frontend Components** - The frontend components should be developed according to Figma Mockup’s Components. Tests should be evaluated on its integration of third-party services to a web application.

**Frontend Web Pages** - The frontend web pages should be developed according to the Figma Mockup’s Pages. The webpages should be evaluated from a user point of view. Tests and feedback should be documented with a clear pinpoint of client-side’s problem

**Middleware Authentication** - The middleware authentication should be developed according to the Json Web Tokens (JWT) industry standard. The QA validation aim is to verify the integrated and packaged middleware. Such that, user specific functions are always authenticated correctly, and token are correctly expired to prevent identity theft.

**Backend API calls** - The backend system should provide all functionalities for its functional requirements specified in the System Requirement Specification. All API are documented and verified under unit testing, and integrated testing. Testing are assessed in both correct and incorrect call.

**Backend Database Management** - The database system should be able to maintain data for users, whether they be buyers or sellers, products and their images, chat messages and transaction histories as specified in the schema??? It must also provide query results when searched for by the front end. This will be implemented using mongoose.

### 3.2.2 Process Assessments

The following process 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
* Manage Products
* Explore Products
* View Store
* Contact Seller

## 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
* Quality Plan
* Software Requirement Specifications
* Risk Management Plan
* Software Maintainability Report
* 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 *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 9126 Quality Model, the four most important qualities for MerchantDice are Functionality, Efficiency, Usability and Reliability.

The application needs to fulfill all the functional requirements specified in the Software Specification Requirements (SRS) completely and correctly. It must also be efficient in the sense that the application, MerchantDice, must be able to deliver the appropriate performance without consuming too much resources. In addition, MerchantDice also has to be simple and easy to use so as to cater to a wide range of intended users. Lastly, the concept of reliability is important as MerchantDice must be able to maintain its level of performance under stated conditions in a stated period of time such that business is not impacted.

### 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:

* Functionality Checklist in accordance with Use Cases
* Number of Error Messages
* Program size in Lines of Code
* Fan in/ Fan-out
* Cyclomatic Complexity
* Depth of conditional nesting

# 

# 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
* Code 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).

The following tests will be conducted:

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.

1. 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.

1. 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.

1. 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 potential problems which may arise in the course of the software development.

1. Documentation
   1. Incorrect information in documentation
   2. Incomplete documentation
   3. Formatting issues
2. Development
   1. Incorrect functionality
   2. Incomplete functionality
   3. Inconsistency with planned design

SQ personnel generate, track, and trend assessment findings and observations in a centralized Reporting and Corrective Action System. The following depicts the detailed steps to be adopted:

1. SQ personnel will conduct regular and scheduled checks and reviews to discover and identify any potential problems.
2. Upon discovering a problem with the project, SQ personnel or members will be required to notify the Project Manager and QA manager.
3. The Project Manager and QA manager will review the problem and determine its severity.
4. Depending on the severity of the problem, relevant members in the project will be called upon for discussions.
5. In the discussion, corrective actions to be taken are determined.
6. The problem and associated corrective actions 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 delegated the required implementations for the corrective actions by a stipulated deadline.
8. Once the corrective actions have been implemented, SQ personnel/ SQ Manager will review the corrective actions. If the corrections made are not implemented correctly or fall short of standards, the team will then need to revert back to step 3.
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:

## *Software Quality Tools*

* Microsoft Office tools (i.e., Word, Excel, and PowerPoint)
* Google Drive (For Collaboration)
* IDE (i.e., Visual Studio Code)
* Web Browser (i.e., Google Chrome)
* CI/CD (i.e., GitHub Actions)

# 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 medial control:

1. Github’s readme.md
2. MediaWiki

Github is an open-source version control system. It is a cloud based repository hosting service, which our team use heavily in collaboration. A readme.md under the “test” folder will provide the team, and other end users with a clear picture of what is being tested, and how it is being tested.

Furthermore, in the developmental phase, github action will be used to do continuous integration with unit and integration testing with each push and pull request. Thus, documentation in github - readme.md will provide a good jist of the present work.

MediaWiki is used because it has rich features in documentation. Users can edit them easily without any knowledge of HTML or CSS. It keeps track of all edits submitted, and it is also reversible to any previous versions. It can also manage images and multimedia files uploaded to MediaWiki Server. By leveraging MediaWiki, it will provide our team with good documentation for the QA plan.

# 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.

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 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 |

# 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.