EX-SELL

Software Quality Assurance (SQA) Plan

Version 1.6 approved

Prepared by Zenith

Yiu Hong Sum	U1621435G
Lim Kian Hock Bryan	U1620949L
Cheng Yang Zhen	U1521618K
Lim Yan Jun	U1622311B
Soong Jie Ming	U1521535B
Chen Zhenni	U1622603D
Chang Keat Lueng Aaron	U1622360F

26th February 2018

Signature Page

Prepared by:	**Yiu Hong Sum**	**Date: 26 February 2018**
	Lim Kian Hock Bryan	**Date: 26 February 2018**
	Cheng Yang Zhen	**Date: 26 February 2018**
	Lim Yan Jun	**Date: 26 February 2018**
	Soong Jie Ming	**Date: 26 February 2018**
	Chen Zhenni	**Date: 26 February 2018**
	Chang Keat Lueng Aaron	**Date: 26 February 2018**

Reviewed by 1: Dr Shen Zhiqi **Date:_____**

Reviewed by 2: _____ **Date:_____**
<(Reviewer Name)>

Approved by : _____ **Date:_____**
(Name)

Document Change Record

Revision	Description of Change	Approved by	Date
0.10	**Initial Template**	**Yan Zhen**	**12/02/2018**
0.20	**Purpose, Scope, Reference Document and Management**	**Yang Zhen**	**12/02/2018**
0.30	**Task, Product and Process Assessment**	**Zhen Ni**	**20/02/2018**
0.40	**Updated Process Assessment**	**Jie Ming**	**20/02/2018**
0.50	**Roles and Responsibility, and Document Requirement**	**Hong Sum**	**20/02/2018**
0.60	**Standards, Practices, Conventions and Metrics**	**Aaron**	**20/02/2018**
0.70	**Software Reviews**	**Zhen Ni**	**20/02/2018**
0.80	**Test**	**Zhen Ni**	**20/02/2018**
0.90	**Problem Reporting and Corrective Actions**	**Jie Ming**	**20/02/2018**
1.00	**Tool, Techniques and Methodologies**	**Bryan**	**20/02/2018**
1.10	**Media Control**	**Hong Sum**	**22/02/2018**
1.20	**Record Collection, Maintenance and Retention**	**Yan Jun**	**22/02/2018**
1.30	**Training**	**Yan Jun**	**22/02/2018**
1.40	**Risk Management**	**Zhen Ni**	**22/02/2018**
1.50	**SQA Plan Change Procedure and History**	**Zhen Ni**	**22/02/2018**
1.60	**Revised Quality Plan**	**Hong Sum**	**25/02/2018**

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 Ex-Sell E-Commerce 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 scope of this plan outlines the procedures, techniques and tools which will be adhered and used to ensure quality in the development of the Ex-Sell E-Commerce, and that it does not deviate from the original intended product. As such, the plan will identify the responsibilities of the project manager, quality manager and the lead developers in the software quality assurance. Quality assuring activities and methodologies will be reviewed and audited by the Quality Assurance team. This ensure that we will be able to identify any errors, omissions, inconsistencies in the development, and alternatives, enhancements or improvements can be made at any stage of development.

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 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. *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:

Product Assessments 001	**Requirements Of System**
Entry Criteria	After SPRINT meeting
Description	The core functionalities of the ex-sell website and special features.
Exit Criteria	Functional and non-functional requirements documentation
Procedure	QAM will review the documentation.
Notes	Nil

Product Assessments 002	**Initial Prototype**
Entry Criteria	After SPRINT meeting
Description	The initial prototype will contain the core functionalities of the ex-sell website.
Exit Criteria	Review of functionalities checks Bug Reports
Procedure	QAM will review product requirements that were stipulated in previous meetings. QAM will then test the prototype of bugs QAM will update bug report
Notes	Nil

Product Assessments 003	**Subsequent Prototype**

Entry Criteria	After initial prototype
Description	The prototype will contain the improvements from the previous prototype.
Exit Criteria	Review of functionalities checks Review of previous bugs Bug Reports
Procedure	QAM will review product requirements that were stipulated in previous meetings. QAM will review the previous bugs. QAM states down resolved bugs and updates bug report.
Notes	Nil

Product Assessments 004	**Finalised Product**
Entry Criteria	When prototype consists of all functionalities
Description	The final product will contain all functionalities of the ex-sell website in working condition.
Exit Criteria	Review of functionalities checks No bugs founds
Procedures	QAM will review product requirements that were stipulated in previous meetings. QAM will check across bug reports to make sure all bugs are resolved QAM will check for any new bugs QAM certifies that the product is free of bugs
Notes	Nil

3.2.2. Process Assessments

The process assessments ensure that the project's process lifecycle adheres to the process requirements as stated by the initial project proposal. The following process assessments will be conducted by SQ personnel:

Process Assessment 001	**Ensuring Conformity between Process life cycle and Requirements**
Entry Criteria	Project Plan
Description	This stage is ensure that Process life cycle does not deviate from the original schedule and also deals with alternate plans such as change management procedure.
Exit Criteria	Quality assurance policy
Procedure	Review list of requirements given by contract and project plan Set schedule for conducting reviews and creating audits of ex-sell prototypes Standardise and document policies to manage process changes Standardise and document policies for making bug report, release control Every prototype introduced will be checked for bugs and errors. This will be compared with the estimated number of bugs for a given number of lines of code. Compile findings into a report to be sent to Project Manager, QAM and lead Developer to go through.
Notes	Nil

Process Assessment 002	**Evaluation of Software Environments**
Entry Criteria	Project Plan
Description	This stage ensures that the libraries or technology used to develop

	ex-sell are in conformance with the contract
Exit Criteria	Non-conformance report (if any.)
Procedure	Before every milestone as noted in the project schedule, conduct a meeting with developer lead to level the type of libraries and packages that may be used for development. Requirements will be explicitly stated and updated should and changes occur. During the development phase, every prototype produced will be scrutinized for any discrepancies in conformance. If there exists and discrepancies in conformance, SQ personnel will highlight to QAM and together with the lead developer, will hold an immediate meeting to discuss and highlight issues with the abovementioned.
Notes	Nil

Process Assessment 003	**Evaluation of Process Measurements for Conformance**
Entry Criteria	Project Plan
Description	This stage ensures that a quantitative way to measure the progress of the development of the project is introduced, as well as reviewed regularly to ensure that project is heading forth as planned.
Exit Criteria	Non-conformance report (if any.)

Procedure	Every milestone, QAM will liaise with lead developer to estimate the lines of code needed to achieve that milestone, as well as different components of the milestone. Lead Developer will disseminate information to developers on the team At the end of every month, developers will be required to submit a simple report on the component that they are working or have worked on, as well as the lines of code produced (This can be done through version control commits) Should any fail to meet the estimated lines of code or progress, measures will be discussed between lead developer and QAM to improve on current status. These measures can include a change in story tickets, etc.
Notes	Peer Reviews can be another form of measurement to determine who is contributing the most and the least to the project

Process Assessment 004	**Evaluation of Staff Skills and Knowledge for Conformance**
Entry Criteria	Any stage of the project lifecycle
Description	This stage ensures that a everyone on the project team is skilled enough to handle the technical requirements of the project. This ranges from management to developers and quality engineers as well.
Exit Criteria	Non-conformance report (if any.)
Procedure	Based on project plan, QAM will evaluate if personnel have the necessary qualifications develop the required components for the project. If there are a mismatch of skill sets, changes to the duties will be discussed between the Project Manager, QAM and the lead developer. If there are any further gaps in knowledge, personnel will be sent for 2 weeks of training in the specific domain(s) to level up their skillsets. Steps 1 - 3 will be monitored every month by QAM to ensure a

	clean track record.
Notes	

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 Management
- Software Model Prototype
- Risk Management
- Design report on software maintainability
- Configuration Management Plan
- Change Management Plan
- Release Plan
- Test Plan and Documentation

5. Standards, Practices, Conventions and Metrics

5.1. *Purpose*

In every software development processes, developers tend to maintain a certain bias that might affect the creation of the final products, which may not be well received by the customers at the end of the project. Therefore, by following international standards and practices, this issue could be avoided as it ensures that all biases are avoided, and key issues are accurately addressed.

5.1.1. Standards

The standard that we will be adopting is the ISO 9001 and ISO 25010. The ISO 9001 Gives your company competitive advantages as it improves the marketability of company products and services. It helps improve the business process such as Project Management, Software Development Life Cycle, and Quality Assurance and Testing, Configuration Management. The ISO 25010 is an international standard for software quality assurance to address potential human biases that might adversely affect the delivery and the development process of software development project.

5.1.2. Practices

Our project will be adopting the Agile software development methodology. Agile will aid us in focusing on getting out an initial prototype for us to test and obtain feedback from. It also provides us the capability to include new functions or refine old ones. This way, we can ensure a more continuous delivery of our software product as well as ensuring quality work that targets the user requirements.

5.1.3. Conventions

We will be using letter case-separated words with our naming conventions. This is help with ease of typing and readability and to avoid confusion when an outside developer looks at the code and perform some modifications per the client request.

5.2. *Software Quality Programme*

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

Having a fast response for a transaction or search of an item will enable the user to have a better experience. Having an efficient system will reduce any frustration the user might experience.

After a user purchases a product, no other user should be allowed to purchase it. The server needs to be reliable such that there are no overlapping purchases which will cause confusion and customer dissatisfaction.

Simplicity will make the product easy to navigate along with having a visual appeal. User experience will also be enhanced, and this will prevent the user from facing any hassle while manoeuvring around the app.

Different pages will have similar if not the same design, this allows the users to have a sense of familiarity when manoeuvring between pages which will enhance their experience and reduce time to learn and search for items.

5.2.1. Standard Metrics

The following standard metrics are the minimum planned metrics that will be collected, reported, and maintained in software quality assurance:

- Fan in/Fan-out
- Length of code
- Cyclomatic complexity
- Depth of conditional nesting
- Weighted methods per class
- Cyclomatic complexity

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
- Software Specifications Review
- Architecture Design Review
- Test Plan Review
- Acceptance Review
- Peer Reviews
    - Code Review
    - Walkthrough
- Post-Implementation 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.

To ensure that our software fulfils the requirements defined in the System Requirement Specifications and to assure to the customer that the software system will be stable for consumer usage upon releasing it to the market, Software Testing will be conducted as part of our Software Development Process. Several levels of tests will be conducted, as follows:

Unit Testing

Unit testing, also known as component testing is meant to ensure that each component or function will operate accurately as it is intended to. Individual classes will be tested to ensure reliability and functionality within a unit-level. Testing module will be created before the tested code to ensure that the code is testable. Example of these would be that user are able to publish a listing successfully or, only users that have verified their emails will be allowed to login.

These tests will be done initially by the software development team and developers themselves to ensure that most implementation errors are removed. The QA team will then further verify through another round of testing.

Integration Testing

Integration is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. Integration testing will be done after unit testing, where the individual working components will be combined to ensure that interactions between the different components are working properly.

System Testing

System testing comprises of a series of different tests where the purpose is to fully exercise the software system. Although each test may have a different purpose, but they will all try to expose the system limitations.

Testing will be conducted by the QA team and any bugs or outcome that is found to be different from the expected result will be reported back to the development team.

Validation Test

The purpose of validation is to prove that the software system performs as specified in the requirements documents listed. Examples of such test will be validation of emails from a domain and testing at the boundary values.

User Acceptance Test

User Acceptance Test is being conducted at the last phase of the development where the actual intended users test the software system to ensure that it is able to handle all the required use cases of their job scopes and the functions are according to the requirement specifications

8. Problem Reporting and Corrective Action

8.1. *Possible Problems which may arise*

Document Problems:

- Non-compliance with other project documents
- Non-compliance with the house style (SCMP)
- Incompleteness
- Errors

Code Problems:

- Lack of functionality
- Wrong functionality
- Non-Compliance with coding or commentary standards

Requirement Change by the Customer

8.2. *Problem Reporting Procedure*

When the documentation requires rectification due to an error or being incomplete, the member which has discover the problem will be required to perform the following procedures:

- Report the situation to the Project Manager and QA Manager
- The Project Manager notifies and provides suggestions to the corresponding team member to complete or change accordingly. The member appointed will then be responsible to solve the problem.
- The team member will have to notify the Project Manager and QA Manager when the problem is solved. They will have to check that the problem is solved.
- When the member cannot solve the problem, they have to notify the Project Manager and QA Manager. They will then have to set up a meeting with the group and a decision must be made to deal with the problem.

In the event when the system or code contains a function which might be incomplete, does not comply with the requirements or is missing totally, the member who discovers the problem will be required to perform the following procedures:

- Report the situation to the Project Manager and Lead Developer.
- The Project Manager notifies and provides suggestion to the corresponding team member to complete or change accordingly. The member appointed will then be responsible to solve the problem.
- The team member will have to notify the Project Manager and Lead Developer when the problem is solved. They will have to check that the problem is solved.
- when the member cannot solve the problem, they have to notify the Project Manager and Lead Developer. They have to set up a meeting with the group and a decision must be made to deal with the problem.

In the event that there is a change in the software requirements by the customer, the following procedures should be taken by the Project Manager:

- The Project Manager must set up a meeting and notify all the members regarding the changes.

- The Project Manager will have to appoint team members to edit the changes accordingly.
- Appointed team members will have to change the application and documentations accordingly.
- Once changes are complete, to notify the Project Manager and the necessary quality checks will be done to ensure that changes comply to the change requested by the customer.

All changes and revisions are to be updated and notified during our weekly meetings. If there is a need to hold an urgent meeting, the issues will first be brought up via phone calls or video conferences to determine the best course of action.

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)
- Microsoft Visual Studio 2017 Enterprise
- MySQL Workbench
- GitHub

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, except for completed checklists from process and product assessments. See Section 11 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.

As the project needs to be stored in a secured server, the following services are used:

1. MediaWiki
2. SVN
3. Google Drive
4. GitHub

MediaWiki is a free and open-source application. This service is used as it is easy for beginners to pick up. There are many FAQs provided which can teach users the functions required by the users. There is a wide range of functions which allows users to create their information in different styles. It also allows users to concurrently edit the page at the same time. Hence, editing of the page will not result in a loss of information.

SVN is used for versioning and for revision control. This service is used as it helps us to keep track of the different versions of source code files and documents being uploaded and edited. It allows users to rename, move, remove and copy the file while keeping the full revision history. Tortoise can be used to integrate windows explorer and hence, allowing us to view and edit the files easily.

Google Drive service is used as a file storage and for the backup of documents initially created. This service allows users to share and store files within the group easily. Also, this service allows users to edit documents concurrently. Version control is also available in this service.

GitHub is used for versioning and for revision control across all team members. This service helps to update and revise different version of source code files and documents being uploaded upon pulling request.

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.

The table below identifies some of the risk that can arise, probability, seriousness of each risk as well as a strategy to manage the risk.

Risk Type	Risk	Probability	Effects	Strategy
Resource	Staff unavailability	Moderate	High May push back selective stages of the project, and ultimately delay the final deployment date.	Get at least 2 members working on the same item so that the other can cover or know what is going on for easier handover
Financial	Cost Overrun	Moderate	Moderate The budget may prove to be more than sufficient	Remove unnecessary expenses
Hardware	Hardware not up to standard	High	High It may lead to schedule overrun as hardware cannot be used	Get a replacement from the vendor in the fastest available timeframe
Hardware	Unable to receive the hardware on time	Low	Moderate It will delay the integration of the hardware with the system.	Outsource to other suppliers

Communication	System not meeting the requirements	Moderate	High It will lead to schedule overrun if it is found out at a later stage (E.g. User Testing)	Re-clarify the requirements

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.