SeniorConnect Project
Software Test Plan

Version 2

01/11/2015

SeniorConnect Project Team
Nanyang Technological University
50 Nanyang Avenue
Singapore 639798

Revision History

Name	Date	AMD*	Reason For Changes	Version
Wang Siqi	01/11/2015	A, M	Draft test plan	V1.1
Ma Xiaoxue	04/11/2015	A,M	Finalize test plan	V2

*A - Added M - Modified D – Deleted

SeniorConnect System

Software Test Plan

14/10/2015

Software Requirement Specification Approvals:

LI YISHAN 01/11/2015

SQA Manager **Date**

刘 [illegible] 01/11/2015

Project Manager **Date**

Contents

1. Purpose

1.1. Objectives

The Software Test Plan (STP) is composed to define the scope, approaches and resources of all testing activities. This test plan describes the overall approach about the testing of SeniorConnect (SC) project. This STP also describes detailed tests approaches. Besides, this document defines the general test process together with the pass/fail and suspension criteria. The break-down responsibilities for tests are specified and the risks involved in the testing process are identified and analyzed.

1.2. Scope

Testing will be executed several times at specified points during the entire software development life cycle. Test plan should cover all levels of testing activities. Testing is also dependent on other activities like function implementation. As a result of this, test plan should be continuously re-visited during the life cycle. Revision should be made according to the latest progress of the project.

1.3. Reference Material

The following documents have been referenced by this STP:

Document Name	**Document Control Version**	**Insurance Date**
PPRO_FUN.2	2.0	08/09/2015
SRS_FUN.3	3.0	22/09/2015
SQAP_FUN.3	3.0	30/09/2015
RMP_FUN.2	2.0	30/09/2015
PP_FUN.2	2.0	22/09/2015
IEEE 829-2008 Standard for Software and System Test Documentation	NIL	NIL

2. Test Item

Table 1 shows the items to be tested and their version. The Document Version should be up-to-date.

Document Name	**Document Version**
SRS_FUN.3	3.0
UCD_FUN.5	5.0
TSTC_BAT.2	2.0

Table 1 Test Items

3. Features to be Tested

Theoretically, at the minimal test level, all the functional and non-functional requirements stated in the Software Requirement Specification should be tested. All the test cases should be developed and the coverage of tests should be monitored and tracked. The other features to be tested during regression testing, system testing and user acceptance testing shall be described in this document.

4. Features Not to be Tested

Due to the time constrain of implementation and testing period, the implementation of the tests depends on the actual status of project progress. The component level tests will focus on the components implemented. All components carrying major functionalities will be tested. Those components haven't been implemented will not be tested. However, SeniorConnect SQA team shall still generate the test cases for selected unimplemented components for future development.

5. Approaches

The hierarchical order of the different test levels is listed in the figure below. The test processes at lower level must be completed before proceeding to next higher level. Test starts with major technical reviews, including software requirements, software design and code reviews. This level lays the foundation for higher level test processes. Faults and failures in requirements, design, code and documentation can be more effectively discovered and avoided during reviews. Before entering the test phase, Software Requirement Review ensure that all the functional and non-functional requirements are stated in atomic and testable manner. This is the basis for functional and performance testing. Software Design Review and Code Review (not major technical review) pre-cleans the code for a more efficient and effective test process. Besides major technical reviews, software test strategy and test cases should also be defined and reviewed by the SeniorConnect project team before real testing commences.

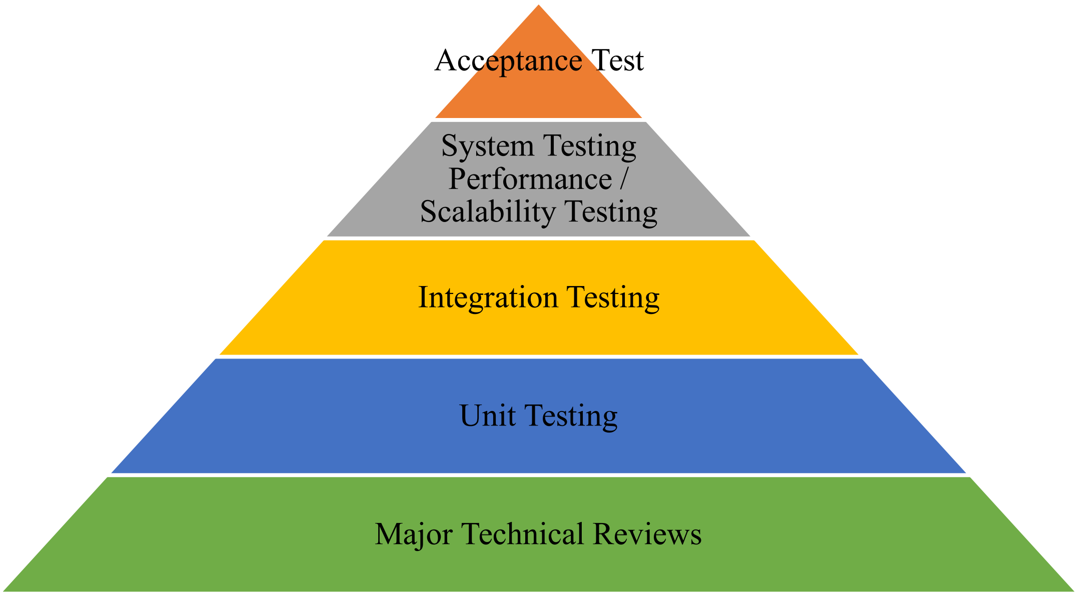

Figure 1 Evaluation Levels

There are some general entry criteria for testing phases.

1. All test items should be completed before testing.
2. All test tools and test infrastructure should be available for use during testing.
3. The correct versions of code should be migrated to the correct testing environment before testing.

Testing process starts from the unit level, elevates to integrated software level and then system level, and finally concludes with user acceptance level. Different testing techniques and tools will be applied for different level of testing.

The SeniorConnect QA team will act as testing team which is independent from the development team. The testing process and debugging process should be seperated.

After each testing phase conducted by QA team, a trouble report will be generated and passed to the development team for debugging.

In section 8.2, different test approaches will be described in greater details, including their objective, test phase entry/exit criteria and tools utilized.

5.1. Unit Testing

- Objective

  This test aims to verify all software elements function properly in isolation.

- Test Phase Entry Criteria

  1. Software Requirement Review, Software Design Review, Code Review have been conducted and the review processes have been audited.
  2. Test strategy and STP have been reviewed and approved.
  3. Test cases for Unit Test are developed and reviewed.

- Implementation

  1. Program Reviews

     Program reviews should be conducted within the development team mainly in form of code walk-through. Developers should review codes developed by each other in this phase.

  2. Black-box Testing

     SC QA team shall conduct black-box testing for each independent Unit. The test cases shall be generated in accordance with the requirements specified in the Software Requirement Specification. The percentage of test case coverage should be monitored and tracked. The coverage for functional and non-functional requirements are specified as following:

     - Functional requirements coverage:

       1. Each functional requirement should be covered by at least one test case.
       2. Several requirements can be covered by the same test case. In this case, the expected outputs for each requirement should be stated separately.

     - Non-functional requirements coverage

       1. Each atomic non-functional requirement should be covered by at least one test case.

  3. White-box Testing

     The SC QA team also need to carry out white-box testing for Unit Test. During the implementation, correctness of most user inputs shall be ensured by using input methods like dropdown list, check boxes. The web APIs shall be treated as individual units for Unit Test. The following aspects will be tested:

- Basic path for each request shall be tested. Basic paths are defined by the most common pattern of usage.
- Boundary conditions for each requests are tested.
- All error handling paths shall be tested.
- Information flow for each interface shall be tested.
- Integrity of the local data shall be tested.

SeniorConnect QA team uses cascading test cases. For the functions that depend on the data collected by other requests, this precedence should be followed during testing. For example, Log-In related functions shall be the first one to be tested. After successful log-in, other functions dependent on it can be tested.

- Test Phase Exit Criteria

1. All Unit Testing activities defined in Software Test Cases have been carried out as specified in this STP and test strategy.
2. Trouble Report has been generated.
3. All major bugs are fixed and tracked to closure.

- Tools

1. PyUnit: Python unit test framework
2. Flask-Testing: flask testing extention

5.2. Integration Testing

- Objective

This test will be carried out when any software elements passed unit tests are integrated together. Integration test shall be realized in an incremental and iterative way until the whole system integration is completed. The tests shall focus on the integration of several software components which work together to achieve certain functionalities accurately and efficiently.

- Test Phase Entry Criteria

1. Unit Test has been completed.

4. Test cases for Integration Test are developed and reviewed.
5. The drivers for testing have been developed.

- Implementation

The QA team shall carry out both decomposition-based and usage-based testing. However, decomposition-based testing will be the primary testing method. Usage-based testing shall only be carried out for critical interfaces and functions.

1. Decomposition-based Integration Testing

   The QA team choose bottom-up approach because it allows higher level of parallelism, easier test specification and easier product control. However, it takes longer time for bottom-up approach to get a working program. The general process is listed as following:

   ➢ Low level units are combined into clusters that perform a specific software function.

   ➢ The cluster is tested with the test driver which coordinates test case inputs and outputs.

   ➢ Drivers are removed and tested clusters are further combined and start a new round of test.

2. Usage-based Integration Testing

   Test cases shall be structured in a mock-up of typical operational environment. The following steps shall be performed:

   ➢ The most commonly used basic paths shall be tested thoroughly and completely to ensure correctness.

   ➢ The test cases shall make assumptions to the most common user inputs and user reactions.

❖ Test Phase Exit Criteria

1. Software integration is completed.
2. All planned Unit Testing activities have been carried out as specified in this STP and test strategy.
3. Trouble Report has been generated.
4. All major bugs are fixed and tracked to closure.

❖ Tools

1. Wifi adapter: to send/receive requests
2. Manually run the test scripts

5.3. System Testing

❖ Objective

The system testing focuses on the interaction between functions within entire integrate system. This testing level aims to verify that the whole system meets specified requirements, both functional and non-functional.

- Test Phase Entry Criteria
  1. Integration Test has been completed.
  2. Test scripts for System Test have been developed and reviewed.
- Implementation
  1. Functional Testing

     The functional testing shall be conducted through an organized testing flow which is described by the test script. The test flow is designed to match and address the items stated in functional requirements. The test script consists of a series of input cases with test flows. The test cases used in Integration Test can be reused to construct test scripts where applicable.

     The test script shall tempt to achieve full coverage by structuring the tests in both positive and negative way, simulating the usage pattern of an unfamiliar user. Positive tests shall verify the software functions as specified in the Project Plan and Requirement Specification. Negative tests are used to explore the software behaviors when it encounters unexpected inputs. The following criteria must be met for functional testing:

     - 100% functional requirements coverage
     - 100% basic paths coverage
     - 100% common usage scenarios coverage
     - 80% of the common invalid input coverage
  2. Pilot Testing

     The integrated software shall be tested among a selected group of end users (around 5~7). The QA team shall monitor Pilot Test process and observe the users' behaviors for general usage patterns. The QA team should also record the bugs found in the Trouble Report.
- Test phase exit criteria
  1. All planned System Testing activities have been carried out as specified in this STP and test strategy.
  2. Trouble Report has been generated.
  3. All major bugs are fixed and tracked to closure.

5.4. Performance / Scalability Testing

- Objective

  This testing level aims to verify the integrated system meets the non-functional requirements specified in the Requirement Specification, especially for availability, security and timing requirements.
- Test Phase Entry Criteria

1. System Test has been completed.
2. Test scripts for Performance / Scalability Test have been developed and reviewed.

- Implementation

1. Load Test

   This test aims to verify the system can handle concurrent accesses as specified in the requirements. The QA team should conduct Load Test by simulating multi-user or multi-threaded access. The performance of the user-end application and database shall be monitored. The detailed test process is:

   - Check whether there are databases reside on a single computer.
   - Check whether there are multiple users access the same database through the user-end applications.
   - Check whether the load on different application servers are balanced.
   - Check whether the users can follow the test script developed for system testing.

2. Performance Test

   The QA team shall conduct Performance Test by running the test cases and record down the execution time required and response time for each type of requests.

3. Security Test

   The QA team shall conduct Security Test to ensure that all the security requirements are satisfied as described in the System Requirement Specification.

- Test phase exit criteria

1. All planned Performance / Scalability Testing activities have been carried out as specified in this STP and test strategy.
2. System performance has been recorded.
3. Improvements for needed performance aspects have been made.

5.5. Regression Testing

- Objective

This test ensures no new bugs are introduced due to new implementation and changes.

- Test Phase Entry Criteria

1. Required changes have been made to implement new features and fix bugs.

2. System testing has been successfully conducted and it is ready to enter UAT phase.

❖ Implementation

SQA team should conduct the following test to ensure no new defects are introduced:

➢ Run all previously created test cases and ensure all test cases are passed, unless the function for the test cases has been decommissioned.

➢ Create new test cases to test changed part of the code base.

➢ Create new test cases to test other functions that are dependent on the changed part

❖ Test phase exit criteria

1. All planned Regression Test activities have been carried out as specified in this STP and test strategy.
2. All test cases are passed.

❖ Tools

Tools used in all previous phases

5.6. User Acceptance Testing

❖ Objective

This stage of test aims to verify the software works appropriately for targeted user group and whether the users would like to accept this system. This testing level shall be carried out after the completion of System Testing and Performance/Scalability Test.

❖ Test Phase Entry Criteria

1. System Test and Performance / Scalability Test has been completed.
2. Test flow for Acceptance Test have been developed and reviewed.

❖ Implementation

1. Alpha Test

   The QA team shall conduct Alpha Test on selected users. The application shall be installed on users' mobile devices before testing. The test shall be conducted at developer's site and monitored by the QA team. The users are free to use any functions that they are interested in within the system. The test would like to simulate real usage scenarios to the greatest extent.

2. Beta Test

   The QA team shall conduct Beta Test on selected users. The application shall be installed on users' mobile devices before testing. The test shall be conducted at customer's site without any supervision. The users are free to use any functions that they are interested in within the system. The QA team should gather feedbacks on the system after the test.

❖ Test phase exit criteria

1. All planned Acceptance Testing activities have been carried out as specified in this STP and test strategy.
2. The results and user feedback have been recorded.
3. Improvements for usability and other aspects have been made.

6. Pass/Fail Criteria

The pass/fail status of a test stage purely depends on the severity of the issues found in test cases. It does not depend on number of test cases passed or failed.

Severity	**Description**	**Allowed?**
S1	Major and minor bug with important functionality	No
S2	Major bug with other functionalities	No
S3	Minor bug with other functionalities and recoverable minor bugs	For one version unless otherwise agreed

Table 2 Issue Severity

7. Suspension Criteria & Resumption Requirements

Suspension happens when severe issues are discovered in the tests. During suspension, the test stage cannot be completed until the suspension is lifted (resumption).

The following scenarios will lead to suspension:

1. S1 or S2 issue
2. Unresolved S3 issue from previous version, and the issue has not been moved to the next release for resolution.

Resumption will only happen when both of the scenarios do not hold.

8. Testing Process

8.1. Test Deliverables

The following deliverables will be produced during the testing phase:

1. Software Test Plan: document that defines the scope, approach, requirements of all testing activities
2. Test Breakdown and Schedule: the high-level schedule and task break-down of the test activities can be found in the Software Project Plan
3. Test Scripts, Test Cases, Test Drivers: the test cases will be recorded to track and monitor the coverage of the tests conducted
4. Trouble Report: report that records and tracks the bugs discovered during testing. It should be passed to the development team for bug fixing.

8.2. Responsibility Matrix

The responsibilities of SC project team members during test phase are tabulated in the following responsibility matrix.

Activity	Project Manager	QA Manager	QA Engineer	Lead Developer	Front-end /Back-end Developer	Release Manager
Test Planning and Estimation	×	×		×		×
Prepare Test Strategy & STP		×	×			×
Review Test Strategy & STP	×	×		×	×	
Approve Test Strategy & STP	×	×				
Provide Unit Test Items				×	×	
Compose Test Script		×	×			
Create Test Cases		×	×			
Review Test Scripts and Test Cases		×		×		×
Test Set-up and Execution		×	×			

Suspend and Resume Tests		✕	✕			
Generate Trouble Report		✕	✕			
Bug fixes and re-test				✕	✕	
Retests and Regression Test		✕	✕			
Hold Bug Review Meeting	✕	✕				
Conduct UAT	✕	✕	✕			
Close Test Phase	✕	✕				

8.3. Test Schedule and Resources

Test Driven Development methodology should be followed for testing. The testing plan phase will start one months before the implementation phase, by when the items should be ready for unit test before release of version 1.0. The test cases should be developed based on the requirements and independent from the implementation code. The software components will be tested as they are being developed. The individual components will be sent for Unit Test immediately after they are coded. They will need to pass Unit Test individually. After the whole system implementation finishes and the components passed Unit Test and components interact with each other passed integration test, the system tests shall then start.

The high level test schedule can be found in the Project Plan. The SQA team will be assigned to the testing tasks. Release team will join them once the release environment is setup for a smooth release. Considering the limited human resource for testing, the development team may also help with Unit Testing when necessary. However, the test cases, scripts, and drives should be developed only by SQA team to ensure that they are independent from the implementation code.

9. Environmental Requirements

9.1. Hardware

Testing Environment:

1. A server.
2. A Relational database.
3. One iOS device.
4. Two mainstream Android devices
5. A personal computer

9.2. Software

Test Case Management Tool: Git

Automation Tool:

1. Nginx
2. JetBrains PyCharm
3. XCode
4. Git

Bug Tracking Tool: Excel sheet

9.3. Infrastructure

Network connections are available on all Test Systems as required

9.4. Test Repository

Git repository setup by SQA team of SeniorConnect project.

10. Risk and Assumptions

The risks involved in the testing phase are identified and documented in the table below.

	Risk	Response Strategy	Probability	Severity
1	Some unimplemented features are untestable.	Identify the features to be tested and will not be tested clearly. Keep a record of the features that have not been tested.	High	Low
2	Delays in the implementation phase result in the late delivery of the Test Items.	Identify the critical activities during upstream planning. Control and monitor the slack in critical implementation activities. If necessary, adjust release scope to leave enough time for testing.	High	High May cause delays in test schedule. Not enough testing may affect quality of released application.
3	Insufficient time for test phase.	Re-schedule the testing tasks. Involve the PM and other team members.	Medium	High May affect the product delivery time and quality.
4	Underestimated time for fixing bugs, especially critical ones.	Establish standard procedures for bug tacking and fixing. Estimation for test schedule should take the fixing time into consideration and add sufficient buffers. Automate regression testing to the maximum extent.	Medium	High The subsequent testing activities will be delayed. The test schedule and release quality may be affected.
5	Illness of testing personnel.	Re-schedule the testing tasks. Involve the PM and other team members.	Low	High May affect the final product delivery time and quality.
6	Break-down of the testing hardware or software tools.	Back up the test cases and test results frequently. The bug report should be kept in multiple repositories and back up frequently. Source help from professional on hardware/software fixing.	Low	High May affect the final product delivery time and quality.
7	Impossible to cover all Android models in the market.	Test representative models that acquired high market shares and are expected to be popular for some time.	High	Medium The application may be incompatible with some Android models.

8	Designed test steps are not followed.	QA Manager should brief the QA team regarding the importance of the testing tasks. QA Manager should conduct random checks to see whether the test steps are being followed.	Medium	Medium Some of the functions not tested thoroughly may contain undiscovered bugs.

11. Change Management

When the Test Items are ready, the development team should inform the SQA team for testing. The development team should specify the code and its version for testing and their location in Git repository. Once the testing begins, no changes or modifications should be made to the implemented units, component or system under testing. If any changes are made during the bug fixing, the changes should be communicated to SQA team. The SQA team will re-test the changed components and conduct regression testing on the fixed code.

After the test cases or the test items are baselined, changes can only be made to them by going through the formal change request procedures. Every member of SeniorConnect project team can require a change on the documents. The change procedures and the Change Request Form can be found in Software Configuration Management Plan_v2.

12. Glossary

Abbreviation	*Description*
SC	SeniorConnect project/system
PPRO	Project Proposal
SQAP	Software Quality Assurance Plan
SRS	System Requirements Specification
UCM	Use Case Model
UCD	Use Case Description
PP	Project Plan
DMA	Design and Maintainability Analysis
CHMP	Change Management Plan
CMP	Configuration Management Plan
RMP	Risk Management Plan
TSTP	Software Test Plan
TSTC	Software Test Case
RP	Release Plan
CPD	Capability Management Maturity Level 2 Definition