

SeniorConnect Project Project Management Plan

Version 2.0

06/11/2015

GruFamily Project Team
Nanyang Technological University
50 Nanyang Avenue
Singapore 639798
22/09/2015

Revision History

Name	Date	A*MD	Reason For Changes	Version
Wang Siqu	10/09/2015	A, M	Draft project description and team organisation	V1.1
Liu Yikun	29/09/2015	A, M, D	Draft project estimation	V1.2
Liu Yikun	30/09/2015	A, M, D	Draft lifecycle model, resource allocation, data conversion, test strategies and best practice checklists; edited the whole file.	V1.3
Mao Huiqi	04/10/2015	A	Draft project schedule	V1.4
Liu Yikun	05/10/2015	A, M, D	Final modification and proof read	V1.5
Liu Yikun	06/11/2015	M	Final revision	V2

Contents

1	Project Description	1
1.1	Project Overview.....	1
1.2	Scope.....	1
1.3	Project Deliverables	2
1.4	Project Management Plan Updates	3
2	Team Organization	4
2.1	Organizational Structure.....	4
2.2	Organizational Boundaries and Interfaces	5
2.2.1	Boundaries between Project Manager & the Rest of the Team	5
2.2.2	Boundaries between QA Team & Development Team	5
2.2.3	Boundaries between Front-end & Back-end Developer	5
2.2.4	Boundaries between Team & Outside Parties.....	6
2.3	Roles & Responsibilities	7
2.4	Management Process	8
2.4.1	Assumption, Dependencies, and Constrains.....	8
2.4.2	Risk Management.....	8
2.5	Communication Plan	8
3	Resource Allocation	9
3.1	Human Resources	9
3.2	Hardware / Software Resources.....	9
4	Technical Process Modeling	10
4.1	Lifecycle Model	10
4.2	Capability Maturity Model	11
4.3	Project Processes Definition	11
5	Project Estimation	15
5.1	Code Size Estimation Using Function Points (FP)	15
5.1.1	Unadjusted Function Points.....	15
5.1.2	Adjusted Functional Points.....	20
5.1.3	Lines of Code.....	21
5.2	Efforts, Duration and Team Size Estimation.....	22
6	Project Schedule.....	23
6.1	Work Breakdown Structure (WBS)	23
6.2	Activity Dependencies and schedule.....	24
6.2.1	Project Timeline.....	24
6.2.2	Major Activity Dependency and Schedule	25
6.2.3	Detailed Activity Schedule, Dependency and Critical Path.....	26
6.3	Work Packages Overview.....	27
6.4	Work Package Details.....	27
7	Data Conversion and Migration	31
8	Quality Assurance	31
9	Test Strategy	31
9.1	Test Item.....	31
9.2	Approaches	31
9.2.1	Unit Testing	32

9.2.2	Integration Testing	32
9.2.3	System Testing.....	32
9.2.4	Performance Testing	33
9.2.5	Scalability Testing.....	33
9.2.6	User Acceptance Testing.....	33
9.2.7	Regression Testing	33
9.2.8	Data Verification.....	33
9.3	Test Deliverables.....	33
10	Best Practice Checklist	34
10.1	Requirement Checklist	34
10.2	Design Checklist	34
10.3	Implementation Checklist	35
10.4	Testing Checklist	36

1 Project Description

1.1 Project Overview

SeniorConnect (SC) system is a social media application which provides a platform to engage the elderly in social life by allowing chatting, sharing and joining events, and to help the elderly manage their life for daily activities. It will be designed and released as a mobile application so that elderly people can access to it anytime and anywhere. Our design emphasizes on convenient voice channel, simple GUI with enlarged fonts, accessible functionalities and security, to provide a reliable and effective social platform for the elderly. The application is designed to achieve the following objectives.

- a. The application should allow users to make voice calls, video calls and send voice messages to other users as communication channels. The application will not include text messaging because it is not an easy to use function for most elderly.
- b. The application should allow users to join communities categorized by hobbies and other topics. The application should also allow community members to receive notifications of news updates and invitations for events organized by communities.
- c. The application should allow users to share their daily life events and experiences with friends, by providing a platform of information sharing and interactive communication in the form of photo sharing.

The application is designed to be encouraging and motivating for the elderly in achieving those objectives.

1.2 Scope

The scope of the project is to deliver a social media system which assists elderly people to enjoy social life by easy communication and various community events. Detailed requirements are delivered in System Requirement Specifications. The following implementations are included:

1. Overall design and architecture should be settled down.
2. Main functions of the system should be implemented and delivered on various mobile platforms, including but not limited to Android and iOS platform, after necessary testing and reviews.
3. Non-functional requirements should be satisfied.
4. Remaining functions should be delivered in prototype with testing.
5. Other functions which are not able to be delivered within the project duration should be put into future development plan.

1.3 Project Deliverables

Below is a list of artifacts to be delivered to the customer for project certification.

Artifacts	Yes/No
Agency Systems Development Life Cycle Methodology (SDLC)	
Application Technical Architecture	
Budget	
Build vs. Buy Analysis for Software	
Business Case	
Business Functional Requirements	
Change Control Process	
Communication Plan	
Configuration Management Plan	
Critical Success Factors	
Deliverable Acceptance Criteria	
Disaster Recovery / Business Continuity Plan	
Hardware Procurement List	
Issue Log	
Lessons Learned	
Maintenance & Support Plan	
Organizational Breakdown Structure	
Performance / Scalability Test Plan	
Practices & Techniques	
Project Charter	
Project Management Plan	
Project Milestones	
Project Schedule	
Project Success Measurements	
Project Test Plan	
Quality Assurance Plan	
Regression Test Plan	
Risk Assessment / Profile	
Roles & Responsibilities	

Security Architecture	
Service level Agreements	
Software Deployment Plan	
Software Procurement List	
Software User Documentation	
Software Verification & Validation Plan	
SOW (Statement of Work)	
Staffing Plan	
Task Estimates	
Training Plan	
UAT Test Plan	
Unit Test Plan	
Work Breakdown Structure (WBS)	

1.4 Project Management Plan Updates

The Project Management Plan will be created and maintained using standard version control processes. After cover page of this document outlines the revision history. Any future update after the baselined version will be controlled by change management process.

2 Team Organization

2.1 Organizational Structure

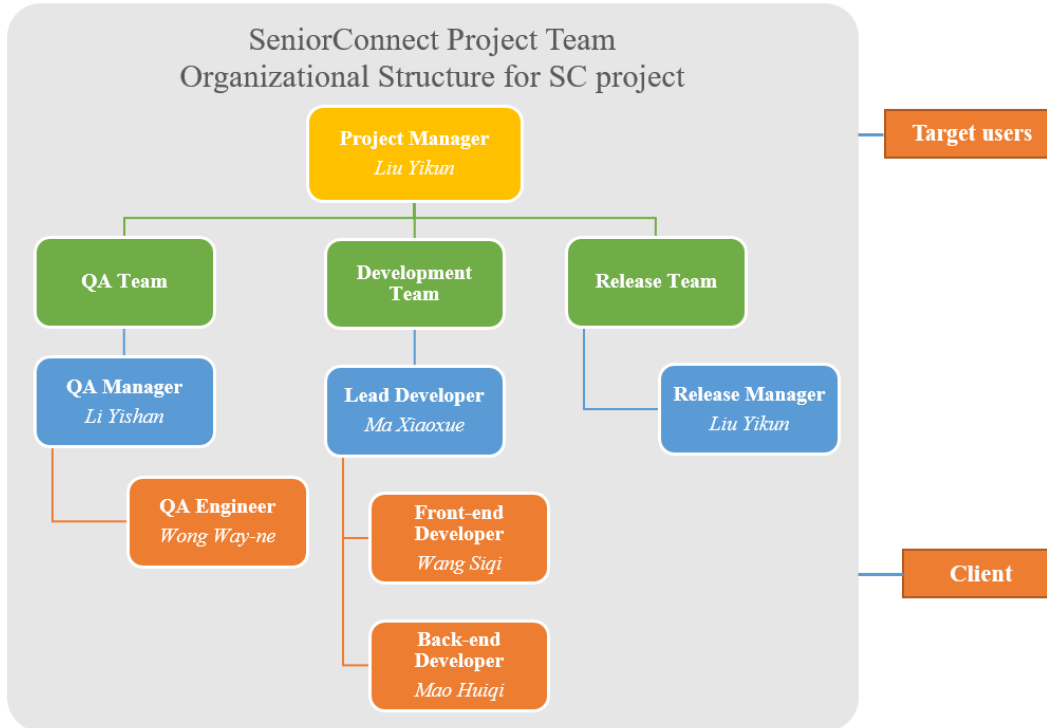


Figure 1: SeniorConnect Organizational Structure

As described in the chart above, the SeniorConnect project team is mainly divided into three sub teams, i.e. QA team, development team and release team. Project Manager is in charge of the overall collaborations and management, positioning at the top.

1. QA team

The QA manager leads the QA engineer to do work regarding on quality assurance, and directly communicates with team manager regarding on quality assurance issues.

2. Development team

The lead developer is in charge of the development and implementation. Lead Developer directs the front-end and back-end developer and directly communicates with team manager.

3. Release team

The release manager creates baselines and builds and integrates changes for delivery. Release manager communicates with project manager directly.

The whole team communicates with outside parties. In SC project, the main outside stakeholders are the client and the targeted users, i.e. School of Computer Engineering of Nanyang Technological University, and the elderly in Singapore respectively.

2.2 Organizational Boundaries and Interfaces

2.2.1 *Boundaries between Project Manager & the Rest of the Team*

Project manager plays an important role during the development process. He/she has to work closely with three sub teams to manage their work and relationships, ensuring a clear job division as well as necessary collaborations, and is responsible for the approval of essential documents produced by the sub teams. The communications with three sub teams are mainly implemented via regular meetings and reviews. Besides, project manager takes high responsibility in communication with clients and outside parties.

2.2.2 *Boundaries between QA Team & Development Team*

Within the SW team, development team is in charge of software design and implementation, while QA team is in charge of the quality assurance regarding on the software project. Main job responsibilities of the teams are listed as below.

QA team responsibilities:

- Software Quality Assurance Plan (SQAP) Process
- Capability Maturity Model Integration (CMMI) Process
- All other issues regarding on testing and audit

Development Team:

- Software design
- User Interface Prototype
- Implementation and coding

Each team works independently. However, collaborations and commutations between the two teams are important during development. The QA team review and test the output from development team, and all QA processes need the effort of development team to fulfill requirements. The overall design and architecture also needs the discussion and reviews from the whole team.

2.2.3 *Boundaries between Front-end & Back-end Developer*

The overall software development responsibilities are divided into two parts, i.e. front-end and back-end implementation. The main responsibilities of the two groups are listed below.

Front-end Developer:

- Software architecture design
- User interface design and prototyping
- System front-end programming

Back-end Team

- Software architecture design
- Server and database implementation

Both groups take part in the software architecture design process. During implementation, two groups take different responsibilities. Front-end group is in charge of the implementation of client-side functions and the user interface. Back-end group is in charge of server, back-end application and database implementation. The separation

requires good communication and cooperation.

2.2.4 Boundaries between Team & Outside Parties

There is boundary between the project team and outside parties. Team members, including project manager and both QA team and development team, are in charge of the development of the software project. The main outside parties related to the project is the client, i.e. School of Computer Engineering of Nanyang Technological University, who will review the project, and the targeted users of the final product. Outside parties will not be directly involved in the development process, but they play an important role for the team since this project requires frequent feedback from targeted users. Communications with outside parties is necessary for the team. The communication with outside parties are mainly between project manager and SCE representative. Besides, the SC team also builds a communication interface with targeted users by conducting survey, doing user interviews in the requirement analysis phase and prototype consulting in implementation phase.

2.3 Roles & Responsibilities

Table 1: Details of Roles and Responsibilities among Project Team

Role	Name	Responsibilities
Project Manager	Liu Yikun	<ul style="list-style-type: none">Responsible for overall delivery of product
QA Manager	Li Yishan	<ul style="list-style-type: none">Responsible for overall product and process quality, implementation of QA processesReport to Project ManagerMonitor and coordinate the work of QA Engineer
QA Engineer	Wong Way-ne	<ul style="list-style-type: none">Devise test plans, conduct testsReport to QA Manager
Lead Developer	Ma Xiaoxue	<ul style="list-style-type: none">Responsible for overall technical lead Responsible for technical aspects of product releaseReport to Project ManagerMonitor and coordinate the work of front-end and back-end developer
Front-end Developer	Wang Siqi	<ul style="list-style-type: none">Android front-end programmingParticipate in the entire SDLC, generating work products including documentation, source code, unit and integration testsReport to Lead Developer
Back-end Developer	Mao Huiqi	<ul style="list-style-type: none">Server, application and database programmingParticipate in the entire SDLC, generating work products including documentation, source code, unit and integration testsReport to Lead Developer
Release Engineer/Manager	Liu Yikun	<ul style="list-style-type: none">Create baselines and build and integrate changes for delivery

2.4 Management Process

2.4.1 Assumption, Dependencies, and Constrains

Assumptions

- There are no hardware or software conflicts among development.
- The project team will be able to access to needed systems during development.
- All required resources, such as servers and test devices, will be available when the team requires them.

Dependencies

- Implementation will only be carried on after the requirement and design documents are reviewed and approved by the QA team.
- The product will firstly be implemented and delivered on Android platform. After the initial launch, iOS implementation will be carried on.

Constraints

- **Technology Constraints:** The technology skills of team members limit the product design. The implementation of the whole design will be within the team's capability to ensure the final product will be properly delivered at the end of the project duration.
- **Time Constraints:** The development of the project is limited by the time. In order to finish the project in time, the product will only be partially implemented with the main functions for initial release. Functions which cannot be implemented will be put into continuous further development plan.
- **Manpower Constraints:** The product is fully designed and implemented by a team of six people. The team is in charge of the whole development process, including QA, design and coding. Due to the limited manpower, design has to be manageable and the product will only be implemented stick to the team's capability.

2.4.2 Risk Management

In order to deliver the final product within budget, on time and in good quality, risk management is necessary for the project planning. Detailed risk management plan is documented in a separate Risk Management Plan.

2.5 Communication Plan

The major communication channel used to communicate within the SC team is WhatsApp. All files and documents are shared on Dropbox. Meetings are organized on regular basis on Monday or Tuesday for discussion based on team members' availability. Meeting minutes are recorded for further reference and backlogs are used to track the main tasks during the entire development process.

The SC team communicate with SCE representative bi-weekly. Email updates will be sent to SCE representative.

3 Resource Allocation

This section describes the resources required to complete the project on time and on budget.

3.1 Human Resources

The SC development team are organized into 3 sub teams, i.e. the development team, the QA team and the release team, under a project manager.

3 experienced engineers are allocated to the development team, which remain independent and report directly to the project manager. The development team members are capable of using various frameworks, HTML, CSS, Javascript and MySQL, which are essential in the development of SC system.

2 personnel are allocated to the QA team, which is responsible for the quality assurance processes. The QA team possess sufficient knowledge in review, audit and other techniques required to ensure the quality of the system.

1 personnel is allocated to the release team, which is responsible for the release of the system. The release engineer have sufficient knowledge in preparation and execution of the release plan.

A detailed allocation of human resources can be referred in Section 2 Team Organization.

3.2 Hardware / Software Resources

The technologies to be used for the implementation of SC system are the server running on the client's organization, i.e. SCE NTU. However, the product prototype required by the client will be developed using the server running on team member's personal computer.

At least 6 mobile phones are required for continuous testing of the implementation. They are provided by the client's organization and is allocated to the development team. Out of the 6 mobile phones, 3 of them have Android system and the rest have iOS system. However, for the product prototype required by the client, only 3 Android mobile phones will be used due to time constraint in configuring 2 platforms.

Most of the work will be done on PC workstations in client's organization. These workstations provide sufficient hardware support and necessary software required for this project. The software packages used for this project include Microsoft Word for documentation, Microsoft Project for scheduling, Visual Paradigm for UML, Github and SVN for collaboration.

Some work will also be done on team member's computer. The software packages in these computers for this project include Integrated Development Environments such as IntelliJ and PyCharm, and Dropbox, Github and SVN for collaboration.

4 Technical Process Modeling

4.1 Lifecycle Model

Agile development Lifecycle is to be adopted in the development of SeniorConnect system. The development is planned to use iterative and incremental (evolutionary) approach performed in a highly collaborative form to produce high quality product in a cost effective and timely manner which meets the changing needs of the stakeholders of SC system.

There are 3 factors driving Agile Development Lifecycle to be chosen.

Technical Issues

- It is not necessary and possible for the SC project gather all user requirements before moving to implementation since it does not have stringent requirements with a social media nature and the requirements keep changing in this new market.
- Fast launch of the product is important to the team since there is currently no similar product focusing on the same target users and entering the market before any other competitor means an easy capture of the market.
- The incremental delivery strategy, where SC is delivered to client and target users and rapid feedback is gotten from them, is realistic. This is because the SC team build communications with stakeholders on a frequent basis, and the team, target users and client are in a relative close location which facilitate easy meeting.
- The SC system will be developed by a team of 6 people who reside relatively close and can communicate informally. This small co-located team and informal communication will make Agile Development the most effective.

Human issues

- The designers and programmers in the SC development team have relatively high skill levels which is sufficient in adopting Agile Development Lifecycle.

Organizational Issues

- The SC development team consists of 6 young engineers who prefer fast delivery of product and reduce project overhead like documentation. Thus the team is willing to try Agile Development without a traditional mind.

In order to achieve a fast launch of product and ensure the quality at the same time, some Waterfall features are adopted in the development as well. The initial launch of this product is planned within 1 iteration, which has requirement analysis, design and implementation phases. However, within the implementation phase, 3 iterations are planned for the team to incrementally implement functions with the highest priority till the lowest. Within each iteration, more detailed requirements regarding the respective functions are gathered and risks are analyzed in order to be mitigated. Then the implementation will be done fast and working product is delivered to the target users and client for feedback.

After the initial release of product, more iterations for future development will be planned accordingly, which is not in the scope within this project.

A more detailed schedule is to be referred in Section 6 Project Schedule.

4.2 Capability Maturity Model

In reference to Capability Maturity Model (CMM), the SC team aims to achieve CMM level two, where processes are well defined and institutionalized. The following key process are going to be implemented.

- Software configuration management
- Software quality assurance
- Software project tracking & oversight
- Software project planning
- Requirement management
- Peer reviews
- Intergroup coordination
- Software product engineering
- Organization process definition
- Organization process focus

4.3 Project Processes Definition

Process 01		Software Configuration Management
Entry Criteria		Configuration objectives
Process Description		This process is setting up the configuration management standard for the project to follow. Old documents will be kept labeled with versions.
Exit Criteria		Configuration management standard guideline Configuration management tools
Process Procedures	Begin	Establish standard for configuration management. Identity configuration management goals. Choose appropriate tool for configuration items. Set release control standard. Specify the methodologies to suggest change requests. Set standard of maintain historical data.
	End	Document the configuration management standard guideline.

Process 02		Software Quality Assurance
Entry Criteria		The QA team receives a deliverable to be controlled.
Process Description		This process takes control of the software quality over every releases of a deliverable from development team to QA team. The QA team should ensure all of the deliverables reach certain pre-defined quality standard.
Exit Criteria		1. Approved of deliverables from the QA team 2. Acknowledgement of revising from the QA team.
Process Procedures	Begin	The QA team understand the pre-defined quality standards of the deliverable. QA group select testing or comparison methodologies to compare the

	End	<p>pre-defined quality standards and deliverable quality.</p> <ol style="list-style-type: none"> 1. If quality gets approved, sign approval document from QA group. 2. If quality gets rejected, change requests or bugs reports will be sent to development team. Process loops back to waiting for the revised deliverable.
--	-----	---

Process 03		Software Project Tracking & Oversight
Entry Criteria		Any point at lifecycle.
Process Description		This process tracks resources and performance of the project for management reference.
Exit Criteria		End of project. Documentation of the assessment of project.
Process Procedures	Begin	(Every Monday) Review all the deliverables of the week Re-review changes of last review Log on Review Form and Minutes of Review
	End	Manage revision history of documents

Process 04		Software Project Planning
Entry Criteria		Proposal approved
Process Description		The whole project team have to determine the basic direction of the project, including durations, costs, number of team members, team members' roles, and risk analysis. In addition, best practice suggestions should be included for the team to follow.
Exit Criteria		Approved project plan
Process Procedures	Begin	Design process structure. Establish project schedule. Establish project commitments. Design work breakdown structure. Assign team member roles. Estimate project. Plan risk management methods. Plan project quality management. Plan change control methods.
	End	Establish testing plan.

Process 05		Requirement Management
Entry Criteria		Proposal approved
Process Description		Identify customers' needs and specify major functions of product.
Exit Criteria		User Requirement Specification Software Requirement Specification
Process	Begin	Conduct survey and interview users to collect their needs and wants.

Procedures		Produce draft copy of user requirement specification. Review draft copy of user requirement specification with users. Revise user requirement specification. Review software requirement specification with software designers and developers and document SRS. Review SRS with user and software designers and developers till all parties come to agreement.
	End	

Process 06		Peer Reviews
Entry Criteria		At the end of any stages at lifecycle.
Process Description		Upon finish some deliverable, team members will review the deliverable and provide feedback for further improvements.
Exit Criteria		Minutes of Review Review Form
Process Procedures	Begin	Plan peer review. Identify the review standard, settle down the method to be used. Establish completion criteria. Document all the unfinished tasks on Minutes of Review. Distribute the review materials.
	End	Log on Review Form whenever a change is to be made. Document the review result.

Process 07		Intergroup Coordination
Entry Criteria		Any point at lifecycle.
Process Description		This process establishes means of communication within the entire team among different sub teams.
Exit Criteria		End of project.
Process Procedures	Begin	The development team and the QA team agree on means of delivery of deliverables. The front-end and back-end development group agree on means of delivery of slides of designs. All teams agree on mechanism of Instant Communication.
	End	Project Manager settle down codebase sharing and project management platform.

Process 08		Software Product Engineering
Entry Criteria		Any point at lifecycle.
Process Description		This process ensures that the team can produce correct, consistent software products effectively and efficiently by performing consistently a well-defined engineering process that integrates all the software engineering activities.
Exit Criteria		Quality acknowledged by the QA team. Reports on improvements. End of project.

Process Procedures	Begin	Present current problems to team members. Brainstorm solutions. Determine alternatives of solution. Discuss tradeoff of each alternative. Choose the most appropriate one and make a plan. Assign tasks to team members. The QA team works on the deliverable.
	End	Complete documentation

Process 09		Organization Process Definition
Entry Criteria		Any point at lifecycle.
Process Description		This process establishes the organizational responsibility for software process activities in order to guarantee the organization's overall software process capability.
Exit Criteria		End of project.
Process Procedures	Begin	List all the processes throughout the project. Identify support or collaborations needed from outside parties and within the organization, i.e. SCE.
	End	Complete documentation.

Process 10		Organization Process Focus
Entry Criteria		Any point at lifecycle.
Process Description		This process develops and maintains a usable set of software process assets that improve process performance across the projects and provide a basis for defining meaningful data for quantitative process management.
Exit Criteria		End of project.
Process Procedures	Begin	Identify processes that involve organization's collaboration. Identify supports needed from organization for each process. Identify focuses and concerns. Identify potential conflicts among different teams. Suggest solutions. Complete documentation.
	End	Keep tracking and raising concerns throughout the project.

5 Project Estimation

5.1 Code Size Estimation Using Function Points (FP)

In this section the unadjusted function points provided by SC system is calculated using the Function Points Model, and then adjusted function points using influence factors is calculated.

5.1.1 *Unadjusted Function Points*

According to System Requirement Specifications, the system will provide following functions.

- Registration
- Login/logout
- Send voice messages
- Listen to voice messages
- Make voice calls
- Make video calls
- Create a group chat
- Add friends to a group chat
- Quit a group chat
- Share photos in moments
- View moments
- Like moments
- Unlike moments
- View community list
- Join new community
- Quit a community
- Receive community event notification
- View community event list
- Join an event
- Drop an event
- View joined event
- View list of friends
- Send a friend request
- Accept a friend request
- Decline a friend request
- View personal profile
- View own photos

The measure of unadjusted function points is based on five primary component elements of these functions: Inputs, Outputs, Inquiries, Logical Files, and Interfaces. Each element ranges from Low Complexity, Medium Complexity to High Complexity. Below are the matrixes of evaluation.

5.1.1.1 Rating of Inputs

Table 2: Summary of Inputs

Input	Data Elements	DETs	FTRs	Rating
Registration	<ul style="list-style-type: none"> • New user • Phone number • Password • Send verification code • Verification code • Register • Error message 	7	1	Low (3)
Login	<ul style="list-style-type: none"> • Phone number • Password • Login • Error message 	4	1	Low (3)
Logout	<ul style="list-style-type: none"> • Logout 	1	1	Low (3)
Send voice messages	<ul style="list-style-type: none"> • Record • Stop record • Voice message • Send • Error message • Successful message (display voice message on current chat screen) 	6	2	Med. (4)
Make voice calls	<ul style="list-style-type: none"> • Friend • Voice call • Error message 	3	1	Low (3)
Accept voice calls	<ul style="list-style-type: none"> • Accept • Result message 	2	1	Low (3)
Decline voice calls	<ul style="list-style-type: none"> • Decline • Result message 	2	1	Low (3)
Stop voice calls	<ul style="list-style-type: none"> • Stop • Result message 	2	1	Low (3)
Make video calls	<ul style="list-style-type: none"> • Friend • Video call • Error message 	3	1	Low (3)
Accept video calls	<ul style="list-style-type: none"> • Accept • Result message 	2	1	Low (3)
Decline video calls	<ul style="list-style-type: none"> • Decline • Result message 	2	1	Low (3)
Stop video calls	<ul style="list-style-type: none"> • Stop • Result message 	2	1	Low (3)
Create a group chat	<ul style="list-style-type: none"> • Group name 	3	1	Low (3)

	<ul style="list-style-type: none"> • Friends • Create 			
Add friends to a group chat	<ul style="list-style-type: none"> • Add • Friends • Confirm 	3	1	Low (3)
Quit a group chat	<ul style="list-style-type: none"> • Quit 	1	2	Low (3)
Share photos in moments	<ul style="list-style-type: none"> • Share • Photo • Confirm 	3	2	Low (3)
Like moments	<ul style="list-style-type: none"> • Like 	1	1	Low (3)
Unlike moments	<ul style="list-style-type: none"> • Unlike 	1	1	Low (3)
Join new community	<ul style="list-style-type: none"> • Join 	1	1	Low (3)
Quit a community	<ul style="list-style-type: none"> • Quit 	1	1	Low (3)
Join an event	<ul style="list-style-type: none"> • Click to join • Error Message 	2	1	Low (3)
Drop an event	<ul style="list-style-type: none"> • Click to drop 	1	1	Low (3)
Send a friend request	<ul style="list-style-type: none"> • Add new friend • Friend • Send 	3	1	Low (3)
Accept a friend request	<ul style="list-style-type: none"> • Accept 	1	1	Low (3)
Decline a friend request	<ul style="list-style-type: none"> • Decline 	1	1	Low (3)

Table 3: Rating of Inputs

Files Referenced (FTRs)	Data Elements (DETs)		
	1-4	5-15	>=16
0-1	Low (3)*21	Low (3)*1	Med. (4)
2	Low (3)*2	Med. (4)*1	High (6)
>=3	Med. (4)	High (6)	High (6)

5.1.1.2 Rating of Outputs

Table 4: Summary of Output

Output	Data Elements	DETs	FTRs	Rating
Receive video call notification	<ul style="list-style-type: none"> • Friend • Accept • Decline 	3	1	Low (4)
Receive voice call notification	<ul style="list-style-type: none"> • Friend • Accept • Decline 	3	1	Low (4)
Receive voice messages	<ul style="list-style-type: none"> • Friend/Group • Voice message 	2	3	Low (4)
Receive community event notification	<ul style="list-style-type: none"> • Notification 	1	2	Low (4)
Statistic diagram for active hours	<ul style="list-style-type: none"> • User name • Date • Active hours 	3	1	Low (4)

Table 5: Rating of Outputs

Files Referenced (FTRs)	Data Elements (DETs)		
	1-5	6-19	>=20
0-1	Low (4)*3	Low (4)	Med. (5)
2-3	Low (4)*2	Med. (5)	High (7)
>=4	Med. (5)	High (7)	High (7)

5.1.1.3 Rating of Inquiries

Table 6: Summary of Inquiries

Inquiries	Data Elements	DETs	FTRs	Rating
View chat history	<ul style="list-style-type: none"> • Chat list • Chat • History 	3	2	Low (3)
View community event list	<ul style="list-style-type: none"> • Community list • Event • Event name • Event time 	4	2	Low (3)
View community list	<ul style="list-style-type: none"> • View community • Community name • Joined or not 	3	1	Low (3)
View joined event	<ul style="list-style-type: none"> • Calendar • Year • Month • Week 	6	2	Med. (4)

	<ul style="list-style-type: none"> • Day • Event name 			
View list of friends	<ul style="list-style-type: none"> • View friends • Friends name • Friends avatar 	3	1	Low (3)
View moments	<ul style="list-style-type: none"> • View moments • Moments photo • Like 	3	2	Low (3)
View personal profile	<ul style="list-style-type: none"> • Profile • User profile image • User name • User phone number • User active hour analysis 	5	1	Low (3)
View personal photos	<ul style="list-style-type: none"> • View photos • Photos 	2	1	Low (3)

Table 7: Rating of Inquiries

Files Referenced (FTRs)	Data Elements (DETs)		
	1-5	6-19	>=20
0-1	Low (3)*4	Low (3)	Med. (4)
2-3	Low (3)*3	Med. (4)*1	High (6)
>=4	Med. (4)	High (6)	High (6)

5.1.1.4 Rating of Logical Files

- Chat history
- User accounts
- User-group
- Group
- Community
- Events
- User-event
- User-community
- Moments

Table 8: Rating of Logical Files

Record Element Types (RETs)	Data Elements (DETs)		
	1-19	20-50	>=51
1	Low (7)*6	Low (7)	Med. (10)
2-5	Low (7)*3	Med. (10)	High (15)
>=6	Med. (10)	High (15)	High (15)

5.1.1.5 Rating of Interfaces

- Access local Contact List to send friend request
- Access local Photo Album to share photo on moments

Table 9: Rating of Interfaces

Record Element Types (RETs)	Data Elements (DETs)		
	1-19	20-50	>=51
1	Low (5)	Low (5)	Med. (7)
2-5	Low (5)	Med. (7)*2	High (10)
>=6	Med. (7)	High (10)	High (10)

5.1.1.6 Calculation of Unadjusted Function Points

Table 10: Unadjusted FP

Characteristics	Low Complexity		Med. Complexity		High Complexity	
Inputs	24	x3	1	x4	0	x6
Outputs	5	x4	0	x5	0	x7
Inquiries	7	x3	1	x4	0	x6
Logical Files	9	x7	0	x10	0	x15
Interfaces	0	x5	2	x7	0	x10
Unadjusted FP	176		22		0	
Total = L + M + H	198					

5.1.2 Adjusted Functional Points

5.1.2.1 Calculation of Influence Factors

Table 11: Influence Factors

Influence Factors	Score (0-5)	Descriptions
Data Communications	5	SeniorConnect is a highly interactive application that involves communication between mobile end and server end.
Distributed Functions	0	All the data and components is stored and maintained on server.
Performance	5	The server response time and mobile end response time is essential to this social application.
Heavily Used	5	The number of users is anticipated to be large.
Transaction Rate	5	Volume of transactions is significant since there is a large user base.
On-line Data Entry	5	Almost all the transactions are interactive.
End-user Efficiency	4	Mobile end's efficiency affects user satisfaction thus it is essential.
On-line Update	4	Online update is critical to ensure data

		consistency.
Complex Processing	1	There is not much mathematic or logical processing since most functions only require data flow.
Reusability	3	Code may be reused for future improvement but it is not an emphasis.
Installation Ease	1	Mobile application is naturally easy in installation.
Operational Ease	3	The start-up, back-up, and recovery procedures need to be designed effective and automated.
Multiple Sites	1	Mobile application is to be published online for installation on personal mobile.
Facilitate Change	4	The application should be specifically designed, developed, and supported to facilitate change in some functions since future customisation and change is highly probable with new requirements.
Total Score	46	
Influence Multiplier = Total score x 0.01 + 0.65 = 46 x 0.01 + 0.65 = 1.11		
Adjusted Function Points = Unadjusted FP x Influence Multiplier = 198 x 1.11 = 220		

5.1.3 Lines of Code

System will be mainly developed in HTML using Ionic Framework. According to Capers Jones' study, the average number of source lines per FP is 34.

Therefore,

$$LOC = 220FP \times 34 \frac{LOC}{FP} = 7480 LOC$$

5.2 Efforts, Duration and Team Size Estimation

The *COCOMO 81: Intermediate Model* is used to estimate effort, schedule and cost. We consider our team a "small" team with "good" experience working with "less than rigid" requirements, which indicates "Organic". So the parameters we use:

$$a = 3.2; b = 1.05; c = 2.5; d = 0.38$$

EAF consideration calculated from 15 Cost Drivers:

Table 12: Cost Drivers

Cost Drivers	Ratings
Product Attributes	
Required software reliability	1.00
Size of application database	1.16
Complexity of the product	1.00
Hardware Attributes	
Run-time performance constraints	1.00
Memory constraints	1.00
Volatility of the virtual machine environment	0.87
Required turnabout time	0.87
Personnel Attributes	
Analyst capability	1.19
Applications experience	1.13
Software engineer capability	1.17
Virtual machine experience	1.10
Programming language experience	1.00
Project Attributes	
Application of software engineering methods	0.82
Use of software tools	0.83
Required development schedule	1.08
Total EAF	1.12

$$\text{Effort} = a \times (KDSI)^b \times EAF = 3.2 \times 7.48^{1.05} \times 1.12 = 29.65 \text{ PM}$$

$$\text{Duration} = c \times \text{Effort}^d = 2.5 \times 29.65^{0.38} = 9.06 \text{ M}$$

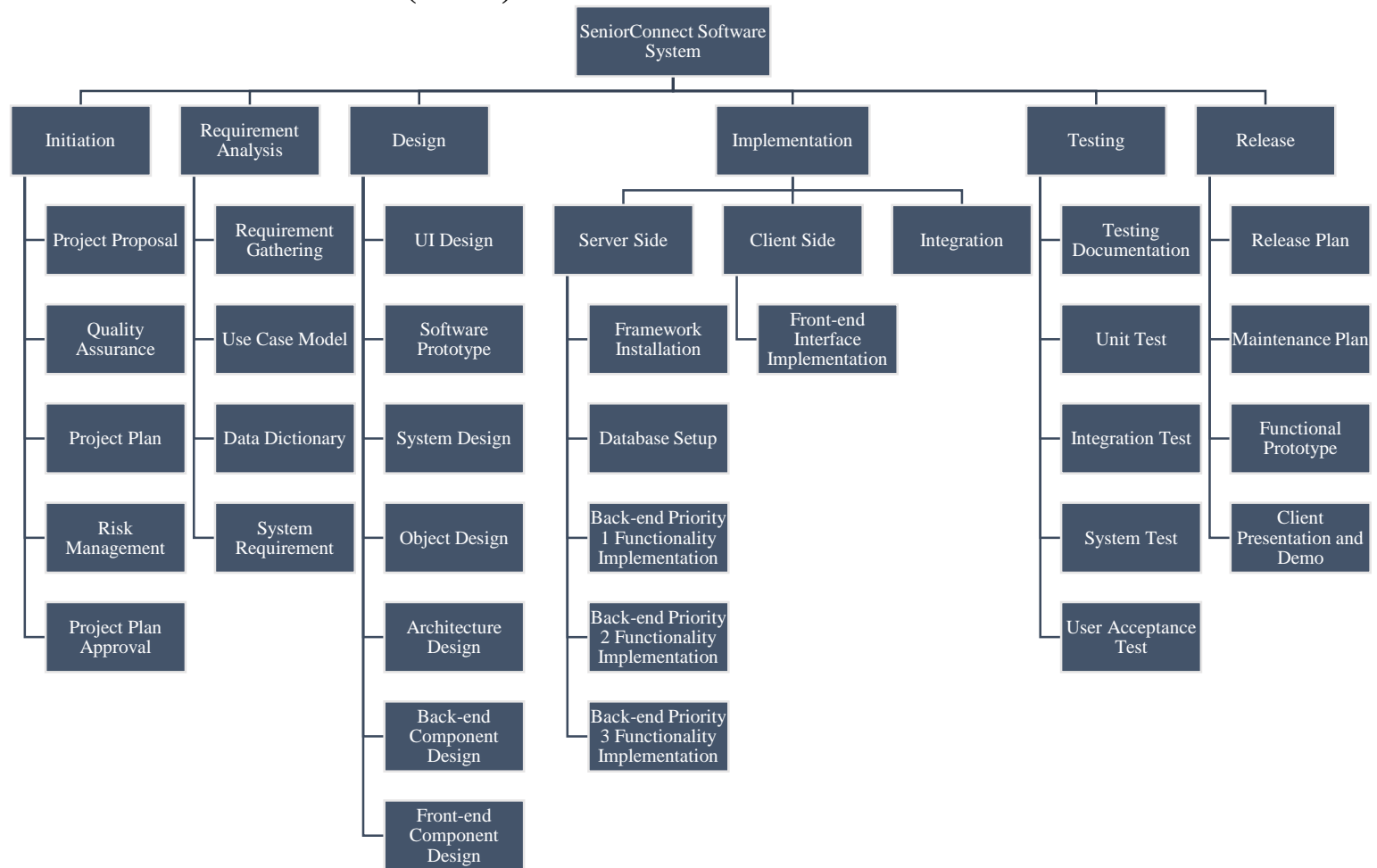
$$\text{Team Size} = \text{Effort} / \text{Duration} = 29.65 \text{ PM} / 9.06 \text{ PM} = 4 \text{ People}$$

In summary, according to the estimation from *COCOMO 81: Intermediate Model*, SeniorConnect system requires a team of 4 members and the estimated duration is 9.06 months.

Given the fact that SeniorConnect consists of 6 team members and the project is scheduled to be completed within 10 months, the project size is under control.

6 Project Schedule

6.1 Work Breakdown Structure (WBS)



6.2 Activity Dependencies and schedule

6.2.1 Project Timeline

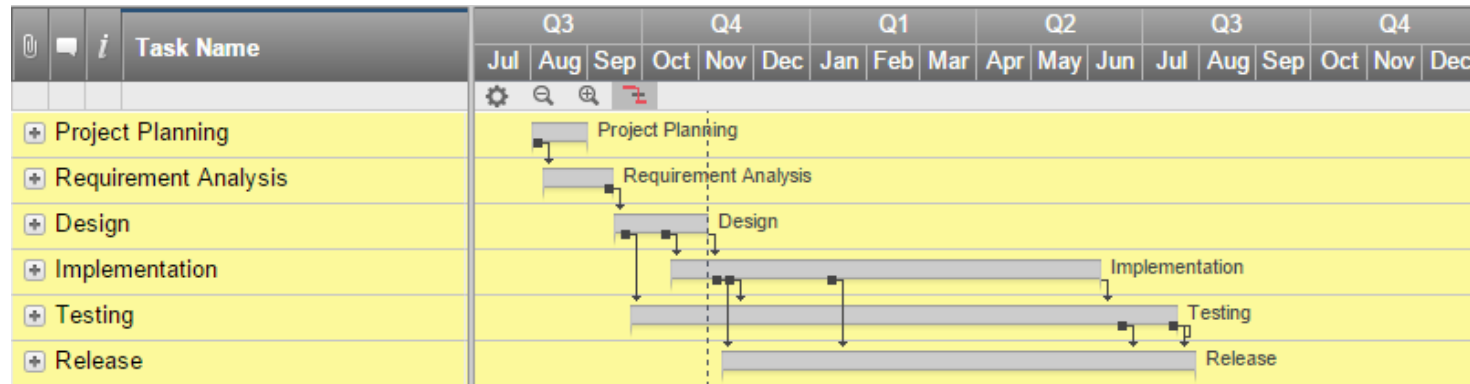


Figure 2: Project Timeline

Project SeniorConnect is divided by six main tasks:

- Project Planning
- Requirement Analysis
- Design
- Implementation
- Testing
- Release

Implementation and *Testing* are major phases and take the longest time. *Release* phase can be divided into documentation phase and release product phase, and there are multiple releases for SC project due to the iterations in implementation phase. The documentation phase finishes before *Implementation* and *Testing*.

6.2.2 Major Activity Dependency and Schedule

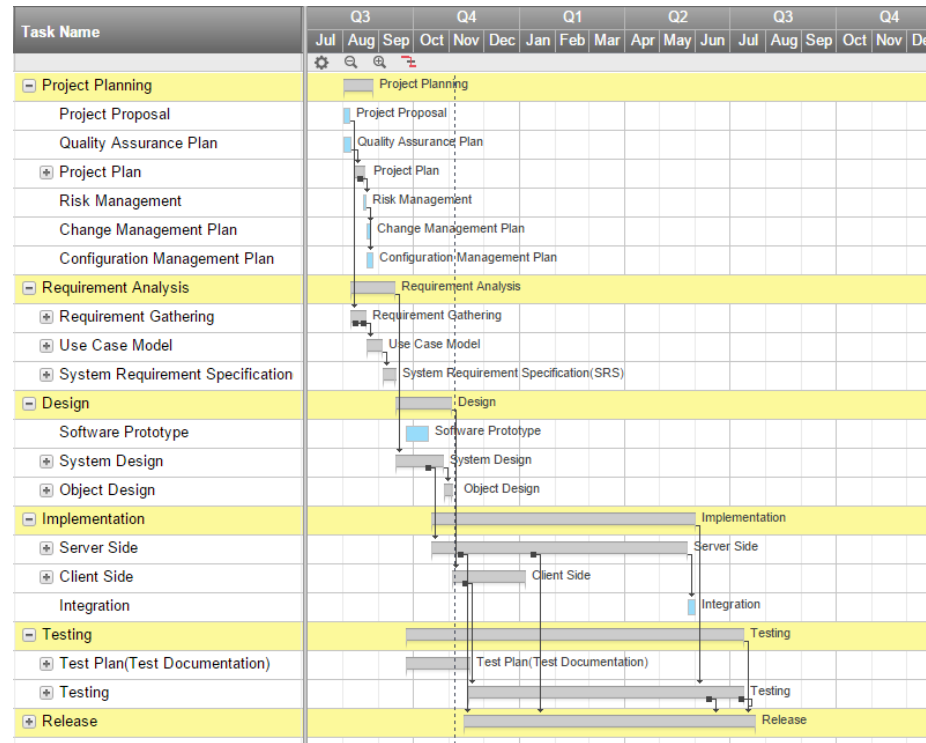


Figure 3: Major Activity Dependency and Schedule

For clearance and simplicity purpose, major activity dependency and schedule are shown here. For example,

- Project plan will not get started after project proposal is approved;
- Design documentation will depend on requirement documentation;
- Actual testing activities will start after testing documentations are done;
- Testing documentation will depend on requirement documentation and prototype etc.

Based on the estimation from last section, we need ten months for coding and implementation, plus two month of requirement elicitation.

6.2.3 Detailed Activity Schedule, Dependency and Critical Path

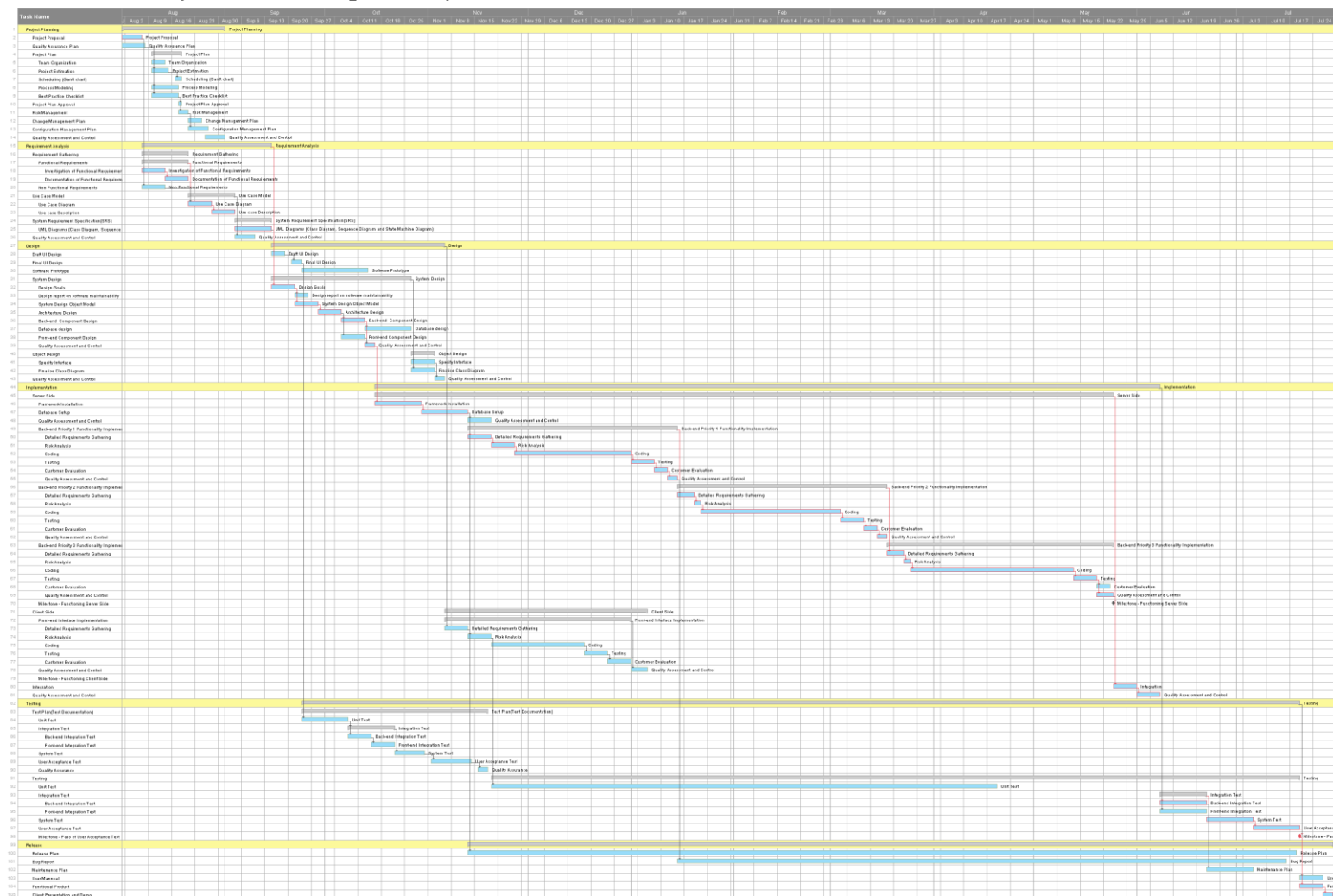


Figure 4: Detailed Activity Schedule, Dependency and Critical Path

6.3 Work Packages Overview

The following table summarizes the deliverable work packages and their dependency. Code SC stands for SeniorConnect.

Table 13: Work Packages

Work Package ID	Name	Duration	Dependency
SC01	Project Proposal	5 Days	-
SC02	Quality Plan	6 Days	-
SC03	Project Plan	7 Days	SC01, SC02
SC04	SRS	7 Days	-
SC05	Software Prototype	14 Days	SC04
SC06	Risk Management	3 Days	SC02, SC04
SC07	Implementation	163 Days	SC04, SC05, SC06
SC08	Testing	209 Days	SC07, SC06
SC09	Release	181 Days	SC08, SC06

6.4 Work Package Details

Project	SeniorConnect Software System
Work Package	SC01 – Project Proposal (1 of 9)
Assigned To	Mao Huiqi, Ma Xiaoxue, Wang Siqi
Effort	5 PD
Start Date	01/08/2015
Purpose	Clarify the project purpose and justification and to gain approval to proceed towards developing the software system
Inputs	Market research
Activities	This work package includes providing a brief overview of the project, its objectives, technical approach in generating the concept and a set of proposed project deliverables throughout the development of the software cycle.
Outputs	Project Proposal

Project	SeniorConnect Software System
Work Package	SC02 – Quality Plan (2 of 9)
Assigned To	Liu Yikun, Li Yishan, Wong way-ne
Effort	6 PD
Start Date	01/08/2015
Purpose	To identify software quality standard and evaluate overall project performance on a regular basis to provide confidence that the project will satisfy the relevant quality standards
Inputs	<ul style="list-style-type: none"> • CMMI Process Definition

	<ul style="list-style-type: none"> • IEEE Standard 730-1998 • ISO 9126-1
Activities	<ul style="list-style-type: none"> • Define ‘software quality’ explicitly • Plan a set of activities that will ensure every software engineering work product later will have high quality • Use metrics to develop strategies for improving the software process to improve the quality of the end product
Outputs	Software Quality Assurance Plan

Project	SeniorConnect Software System
Work Package	SC03 – Project Plan (3 of 9)
Assigned To	Liu Yikun, Mao Huiqi, Wang Siqu, Wong Way-ne
Effort	6 PD
Start Date	10/08/2015
Purpose	To determine an introductory overview of the project including team organization, estimation, schedule and so on, to be refined in later work packages
Inputs	<ul style="list-style-type: none"> • Project Proposal • Quality Plan • Standard template
Activities	This work package includes providing a brief overview of the project, its objective, team organization, project estimation, schedule, process model and best practice check. The people responsible for this work package will also be transcribing ideas brought up in the group meeting discussion into a formal report
Outputs	Project Plan

Project	SeniorConnect Software System
Work Package	SC04 – Software Requirement Specification (SRS) (4 of 9)
Assigned To	Ma Xiaoxue, Mao Huiqi, Wang Siqu
Effort	7 PD
Start Date	04/09/2015
Purpose	To establish a common understanding between the customer and the software project team of the customers’ requirements to be addressed by the project
Inputs	<ul style="list-style-type: none"> • Project Proposal • Customer’s requirements
Activities	Identify who is the customer, interview customers, collect customers’ requirements and organize them
Outputs	SRS

Project	SeniorConnect Software System
Work Package	SC05 – Software Prototype (5 of 9)
Assigned To	Liu Yikun, Li Yishan, Ma Xiaoxue, Mao Huiqi, Wang Siqu, Wong Way-ne
Effort	14 x 6PD
Start Date	24/09/2015
Purpose	To allow users of the software to evaluate developers' proposals for the design of the eventual product by actually trying them out, rather than having to interpret and evaluate the design based on descriptions. Prototyping can also be used by end users to describe and prove requirements that have not been considered
Inputs	SRS
Activities	This work package includes prototyping the software system into a roughly workable prototype with basic flows according to the user requirements
Outputs	Software Prototype

Project	SeniorConnect Software System
Work Package	SC06 – Risk Management (6 of 9)
Assigned To	Ma Xiaoxue, Li Yishan
Effort	3 PD
Start Date	18/08/2015
Purpose	This work package is developed to document the project's approach to risk management, risks, and decisions made about what should be done with each risk. Risks and risk actions are then tracked to closure
Inputs	<ul style="list-style-type: none"> • Project Proposal and SRS • Past experiences and statistics
Activities	<ul style="list-style-type: none"> • Risk identification • Mitigate a risk event • Deflecting the risk items to another entity • Consider risk event's probability • Consider the effect's potential severity
Outputs	Risk Management Plan

Project	SeniorConnect Software System
Work Package	SC07 – Implementation (7 of 9)
Assigned To	Liu Yikun, Ma Xiaoxue, Mao Huiqi, Wang Siqu, Li Yishan, Wong Way-ne
Effort	163 x 6PD
Start Date	15/10/2015
Purpose	To implement the system as per the requirements specification and other associated documents

Inputs	<ul style="list-style-type: none"> • SRS • Software Prototype
Activities	<ul style="list-style-type: none"> • Android App • IOS App • Server Implementation
Outputs	Source Code

Project	SeniorConnect Software System
Work Package	SC08 – Testing (8 of 9)
Assigned To	Ma Xiaoxue, Li Yishan
Effort	209 PD
Start Date	23/09/2015
Purpose	This work package is developed to document the project's testing approach and plan
Inputs	<ul style="list-style-type: none"> • SRS • Software Source Code
Activities	<ul style="list-style-type: none"> • Design test cases • Unit test • Integration test • System test (load test, penetration test) • Acceptance test • Defect tracking
Outputs	Test Plan, Test Cases and Requirements Test Coverage report

Project	SeniorConnect Software System
Work Package	SC09 – Release (9 of 9)
Assigned To	Liu Yikun, Mao Huiqi, Wang Siqu
Effort	181 PD
Start Date	13/11/2015
Purpose	This work package is developed to define the contents of a release or a specific shippable product increment
Inputs	<ul style="list-style-type: none"> • Test Documentation
Activities	<ul style="list-style-type: none"> • Release Plan • Document how to maintain the system • Functional Software • Client Demo and Presentation
Outputs	Release Plan, Maintenance Plan

7 Data Conversion and Migration

Data migration and data conversion is the transition of data from the legacy system to the newly developed system. In the context of SC, there is no data conversion from the legacy system. However, SC needs to interface with user's local contact list to add friends and access user's local album and photo functions to post photos.

8 Quality Assurance

The Software Quality Assurance Plan aims to serve as a comprehensive rulebook, guideline, reference and specification for the SQA processes throughout the whole SC project, which aids the achievement of more disciplined, well-planned, well-documented, thoroughly implemented and soundly validated SQA processes.

A detailed plan can be referred in the separate Software Quality Assurance Plan for SC system.

9 Test Strategy

This section shall describe the overall test approach that will be followed by the SC project team. In the following sub-sections, the types of test that shall be carried out by SC project team will be described. More details on tests strategy, test planning and any deviation from this test strategy section shall be documented in the Test Plan. In the Test Plan, the general test process that shall be followed by each of the tests is defined; the pass/fail and suspension criteria are also defined; the break-down responsibilities for tests are specified.

9.1 Test Item

Table 14: Test Items

Document Name	Document Version
System Requirement Specifications (Functional)	V3
System Requirement Specifications (Non-Functional)	V3
User Case Model	V5

9.2 Approaches

Figure 10 illustrates the hierarchical order of the different test levels. Before going up to the next higher level, the test processes at current level need to be completed first.

At the basement level, the major technical reviews, including software requirements, software design and code reviews, lays a foundation for higher level test processes. Faults in requirements, design, code and documentation can be more effectively uncovered during reviews. Before entering the test phase, System Requirement Review ensure that the functional and non-functional requirements, which are the basis for functional and performance testing, are stated in an atomic and testable way. Software Design Review and Code Review (not major technical review) pre-cleans the code for testing, ensures that the test efforts are concentrated on more critical aspects.

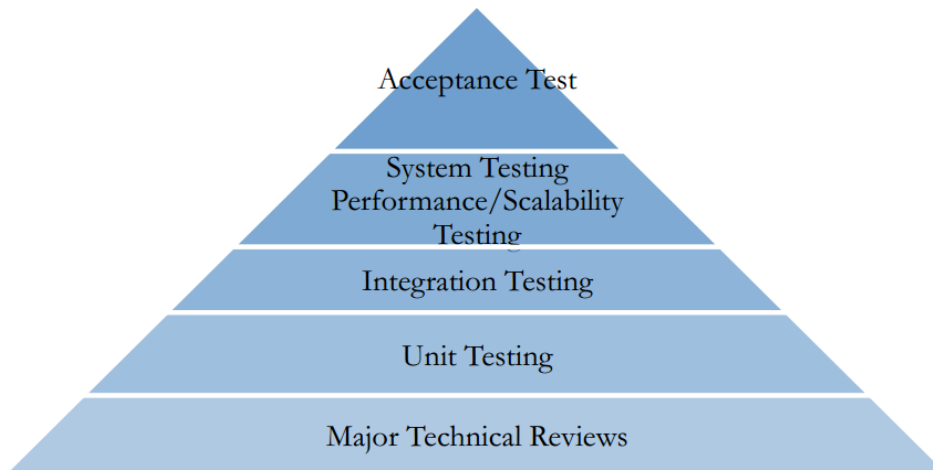


Figure 5: Evaluation Levels

Besides these formal technical reviews, the test strategy and test cases should also be reviewed by the whole SC project team before then can be used in the testing.

There are also other general entry criteria for testing phase. Firstly, all test items should be completed before testing. Secondly, all test tools intended to be utilized by the resting team should be available and test infrastructure are available for use during testing. Thirdly, the correct versions of code should have been migrated to the correct testing environment before testing.

Testing begins at the unit level, and then starts to elevate to integrated software level and then the system level, finally the acceptance level. Different testing techniques and tools are appropriate for different level of testing.

The SC SQA team will also act as testing team that is independent from the development team. SC project team intends to separate the process of testing and debugging. After SC SQA team conducted testing, a trouble report shall be generated and passed to the development team for debugging.

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

9.2.1 Unit Testing

This is the testing of each individual module in isolation to verify that the modules created by the technical team correctly outputs expected results with a given input. This basic test that will allow us to ensure that bugs are detected early and rectified before the module becomes embedded into the system. JUnit will be used for Unit Test.

9.2.2 Integration Testing

This occurs when we test several modules working together to verify that their combined output is expected given a set of inputs. The main aim of this test is to verify that the modules created by different engineers can communicate with each other correctly and thus function as a whole. This testing will transit to system testing when all modules are completed.

9.2.3 System Testing

Test the interaction between functions within entire integrate system with end to end flow. It is also to verify that the whole system meets specified requirements, both functional and nonfunctional.

9.2.4 Performance Testing

Test to verify that the integrated system meets the non-functional requirements specified in the Requirement Specification, especially for availability, security and timing requirements. Due to the mobile nature of our system we will also be measuring several other metrics as performance will be affected by these factors:

- Network data usage which is when the internet connection is poor
- Memory usage especially when the device memory becomes severely constrained.
- CPU usage as it directly affects how much energy is used on end user's devices.

9.2.5 Scalability Testing

The system will be based on a client server model as such we will measure the resources needed by the server to service a small group of test clients. This allows us to ensure that the server we pick to host the system will be adequate to serve peak loads while still remaining responsive to clients within the performance requirements set out in the System Requirement Specification.

9.2.6 User Acceptance Testing

This occurs when we submit the first version to our customer to allow them to validate that the system is up to their expectations and thus accept our system as completed.

9.2.7 Regression Testing

Any changes to the completed system will undergo regression testing to verify that the changes have no adverse effects on the existing modules that are already in use. Changes may be caused by bug fixes, improvements to the existing features or completely new features feeding data to existing modules.

9.2.8 Data Verification

Test to verify that data has been migrated without any inaccuracy and inconsistency during conversion. In the SC project, the main test is to verify the system can access local contact list correctly.

9.3 Test Deliverables

Upon completion of the tests listed in the previous section, the following documents will be produced as evidence of quality of the SeniorConnect system during the various stages of testing.

- Test plan
- Test breakdown and schedules
- Test scripts, Test cases
- Defects report

10 Best Practice Checklist

10.1 Requirement Checklist

No.	Item	Fulfilment
1	Are requirements complete in terms of construction?	
2	Is each requirement uniquely identifiable?	
3	Are all requirements consistent within document?	
4	Does each requirement properly reflect the need?	
5	Is there any ambiguity in any requirement?	
6	Is each requirement traceable and testable?	
7	Is each requirement necessary in the scope?	
8	Are all requirements properly reviewed?	
9	Are requirements consistent across all documents?	

10.2 Design Checklist

No.	Item	Fulfilment
1	Does the design support both product and project goals?	
2	Is the design complete?	
3	Is the design able to achieve targeted requirements?	
4	Is the design clear and concise?	
5	Is the design feasible for implementation and maintenance?	
6	Does the design has minimized dependency among components?	
7	Is the design able to cater to possible risks?	
8	Is the design open to extension and modification?	
9	Can the design be reused for future similar projects if possible?	
10	Is the design traceable from requirements?	
11	Have enough details been incorporated into design documents to properly and sufficiently define the basic architecture of implemented product?	
12	Are the terminologies and naming conventions consistent across all design documents and requirement documents?	
13	Are all design documents consistent with each other in terms of functionality and design principles?	
14	Is the design compatible and consistent with implementation environment?	
15	Does the design meet required performance level in terms of efficiency and resource consumption?	
16	Are there any duplicated components in design documents?	
17	Can this design model be further simplified for easy understanding and implementation?	

10.3 Implementation Checklist

No.	Item	Fulfilment
1	Does the design provide enough detail for implementation?	
2	Do the implementation team have enough knowledge in the context of programming language interfaces and system environments?	
3	Is the implementation traceable from design and requirement level?	
4	Does the code follow the same naming convention and coding style?	
5	Is the code very much dependent on the language capability?	
6	Does the implementation address all requirements?	
7	Is the code able to be complied?	
8	Are the implementation details hidden as much as possible in terms of encapsulation?	
9	Is the code easy to understand?	
10	Does the implementation pass tests defined according to requirements and design?	
11	Is the code properly documented and complementarily explained by necessary documents?	
12	Is the implementation able to achieve targeted performance level in terms of efficiency and resources?	
13	Are the errors handled properly in the implementation?	
14	Is the implementation easy to change if the any modification is required for functionality or design?	
15	Is the implementation having enough modularity and least data dependency?	
16	Is the implementation able to achieve self-documentation to the best extend?	
17	Are the comments in code consistent with code itself?	

10.4 Testing Checklist

No.	Item	Fulfilment
1	Are the test cases built according to the requirement?	
2	Is each requirement covered by one or more test cases?	
3	Is each test case properly and completely defined?	
4	Does the test case cover normal case?	
5	Does the test case cover boundary case?	
6	Does the test case cover error case?	
7	Does the test case cover incorrect data?	
8	Do the test cases consist of at least one stress testing?	
9	Is regression test included if appropriate?	
10	Is each test case uniquely identified and without duplication?	
11	Are test cases properly reviewed before proceeding to testing process?	