



PROJECT EMBER

The platform for elderly to connect freely.

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

TEAM CHAMPIONS:

Arkar Min	(U1721052K)
Lee Wan Sin	(U1620682G)
Daniel Benjamin Delago	(N1804923C)
Jeslynn Teow Zhini	(U1622004L)
Ting Jie Yi Rachel	(U1621976F)
Georgi Krasimirov Markov	(N1800364H)
Sean Tan Jun Yu	(U1520094B)

Partnered and Sponsored by:



**NANYANG
TECHNOLOGICAL
UNIVERSITY**



Table of Contents

1 Introduction	3
1.1 Project Overview.....	3
1.2 Project Description and Scope	3
2 Project Organisation	3
2.1 Team Structure	3
2.2 Roles and Responsibilities.....	4
2.3 Team Communication.....	5
3 Process Definition	6
3.1 Lifecycle Model	6
3.2 Advantages of Model.....	6
4 Schedule.....	7
4.1 Activity Dependencies and Schedule.....	7
4.1.2 Timeline (with milestones)	7
4.2 Work Breakdown Structure	9
4.3 Work Packages.....	10
4.4 Activity Dependencies	10
4.5 Work Package Details	11
5 Project Estimates	15
5.1 Code Size Estimation using Function Points	15
5.1.1 Unadjusted Function Points.....	15
5.1.2 Adjusted Function Points.....	18
5.1.3 Lines of Code.....	19
5.2 Efforts, Duration and Team Size Estimation	20
5.2.1 Distribution of Effort.....	20
5.3 Cost Estimates.....	21
6 Risk Management	22
7 Quality Assurance Plan	24
8 Monitoring & Control.....	25

1 Introduction

1.1 Project Overview

Our project “Ember” is a mobile application which provides a platform for the elderly of our society to expand their social circle, where they can interact and mingle with other elderlies. Our aim would be to reduce the number of elderlies who are lonely, preventing from slipping into depression. With information on their interests and hobbies, we would be making use of a matchmaking algorithm in order to match them with people who are similar to themselves, and hopefully with good chemistry.

1.2 Project Description and Scope

“Ember” is a project that was taken in collaboration with Lions Befrienders Singapore, a social service organisation whose mission is to provide friendship and care for seniors to age in a community with community participation, enabling them to enjoy meaningful and enriching lives.

The basis of “Ember” would be to make use of user’s interests, hobbies, locations and other details in order to match them with another user of similar interests. The idea behind it is to ensure that they have a common interest, which would lead to a topic to talk about which they can further expand from there.

The system will return a list of “matches” of other users with similar interests, in which user can choose which user they would like. In the event where both users have indicated interest in each other, system will send out a notification to inform them, and open a chatroom for them where they can chat with each other.

The user interface will include all necessary user interfaces to quickly and determine the user’s requirements and preferences. However, the system will have no method of validating whether information is valid or invalid and validity of details would remain up to the user.

2 Project Organisation

2.1 Team Structure

The following is the list of roles of the project team:

Roles	Member
Project Manager	Arkar Min
Lead Developer	Daniel
Front-End Developer	Sean
Back-End Developer	George
Quality Manager	Bernice
Quality Engineer	Jeslyнна
Release Engineer	Rachel

2.2 Roles and Responsibilities

Project Manager: Arkar Min

- Monitor's team's progresses and deadlines
- Plan Project Scope
- Plan System Architecture and Design
- Develop Use Cases
- Documentation
- Involved in all activities

Lead Developer: Daniel

- Plan System Architecture and Design
- Develop Use Cases
- Oversees the work done by other developers
- Choose software development environment and tools
- Documentation

Front-End Developer: Sean

- Plan System Architecture and Design
- Develop Use Cases
- Develop the front-end interface
- Choose software development environment and tools
- Documentation

Back-End Developer: George

- Plan System Architecture and Design
- Develop Use Cases
- Develop the back-end system
- Choose software development environment and tools
- Documentation

Quality Manager: Bernice

- Monitor the overall quality of the system
- Documentation
- Evaluate the System Architecture and Design
- Evaluate Use Cases
- Ensure the final product meets the requirements
- Develop quality management plan

Quality Engineer: Jeslynnna

- Evaluate the System Architecture and Design
- Documentation
- Ensure the final product meets the requirements
- Develop quality management plan
- Prepare Test Cases

Release Engineer: Rachel

- Prepare Release Plan
- Prepare Test Cases
- Ensure the final product meets the requirements
- Report Bug Issues to developers
- Documentation

2.3 Team Communication

“Ember” team’s communication are as follows:

- Weekly meetings are held on Tuesday
- Group Announcements, major updates and enquiries are broadcasted on the WhatsApp Group Chat
- Split-Up into sub-groups at appropriate parts of the project
- Documents disseminated on shared Google Drive folder

3 Process Definition

3.1 Lifecycle Model

Project Team would be making use of the Agile Development Model which is also a type of incremental model. Our software would be developed in incremental, rapid cycles – i.e. small incremental releases with each release building on the previous functionality.

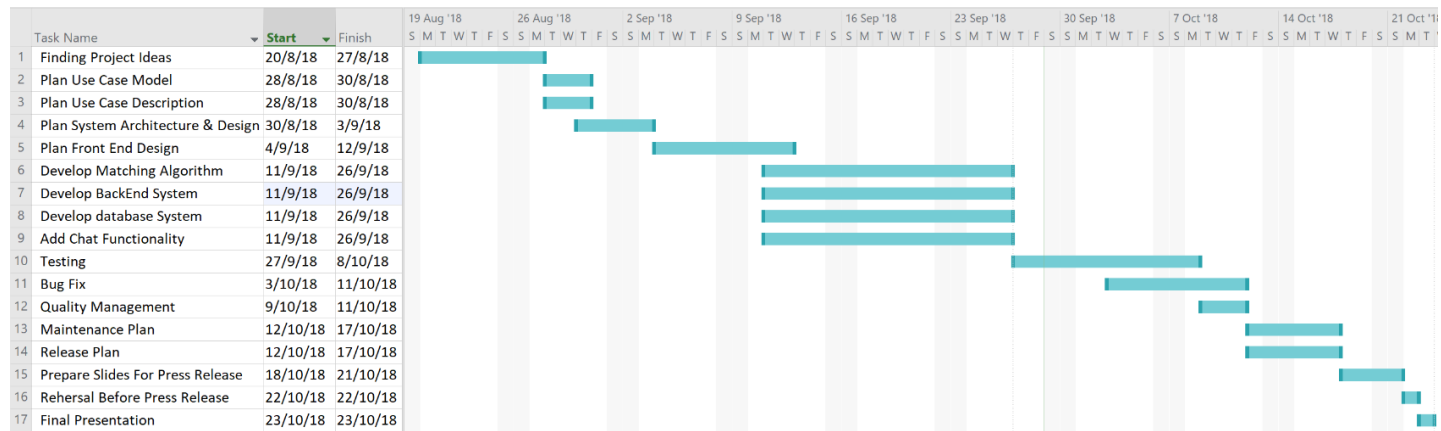
Agile is based on adaptive software development methods, as compared to the traditional SDLC models such as Waterfall model, as it leaves more flexibility to incorporate changes. No detailed planning would be required and there is only clarity on future tasks in respect of features to be developed.

3.2 Advantages of Model

Advantages are as follows:

- Interaction between client and project team is emphasized where there is a need for them to constantly interact with each other.
- Working software is delivered regularly due to constant updates based on the iterative model.
- There is a regular adaptation to changing circumstances, where late changes in requirements can also be allowed.
- Flexibility given to developers where they are allowed to make changes in each update as compared to trying to make updates after final system has been established.
- With minimal rules, documentation can easily be employed.

4.1 Activity Dependencies and Schedule



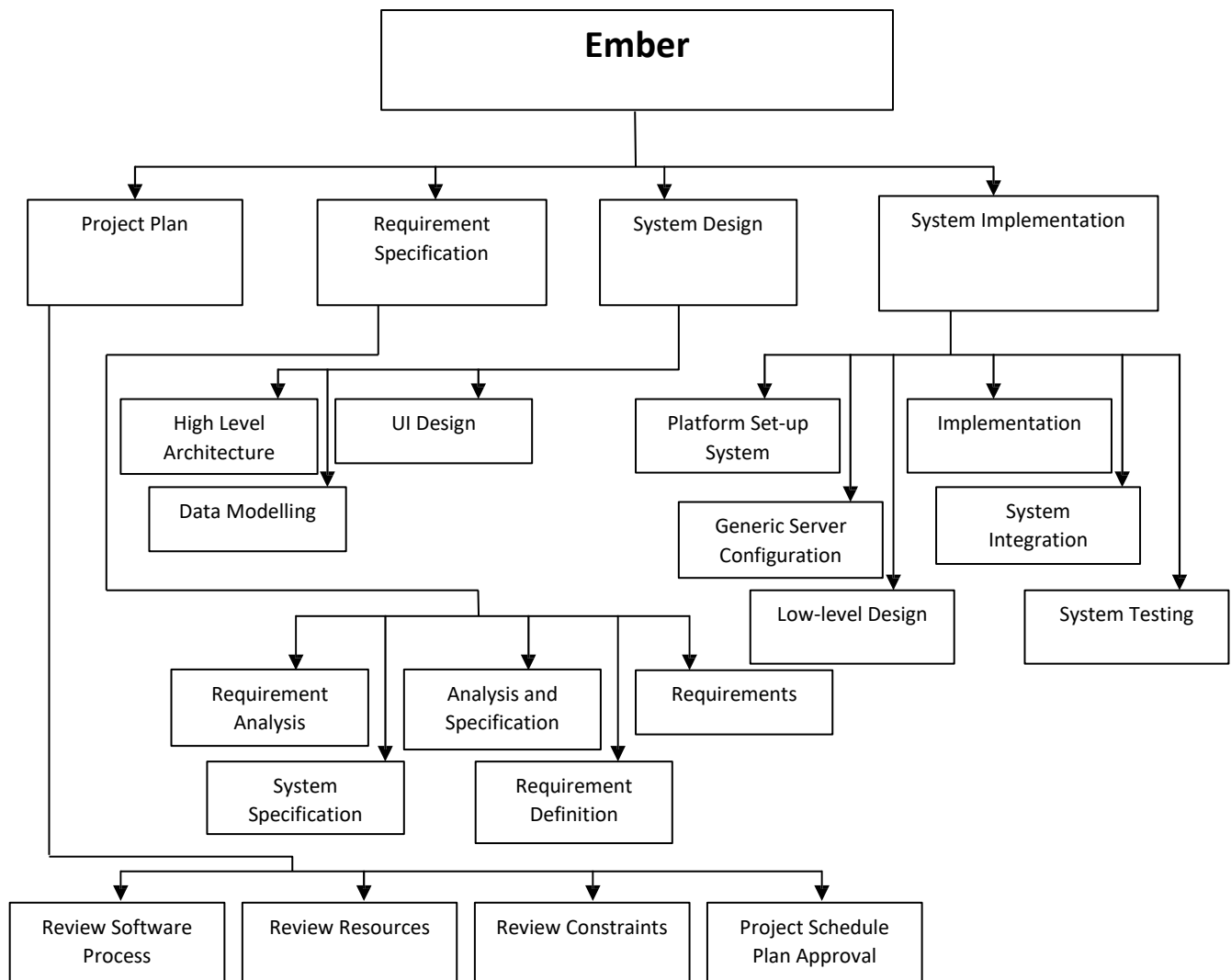
4.1.2 Timeline (with milestones)

Total Duration - 10 weeks

[illegible]

Task	Week 10 (22 Oct to 26 Oct)				
	22	23	24	25	26
Rehearsal before Press Release					
Final Presentation					

4.2 Work Breakdown Structure



4.3 Work Packages

The entire project work is broken down by the important phases of the software development life cycle. They include the following:

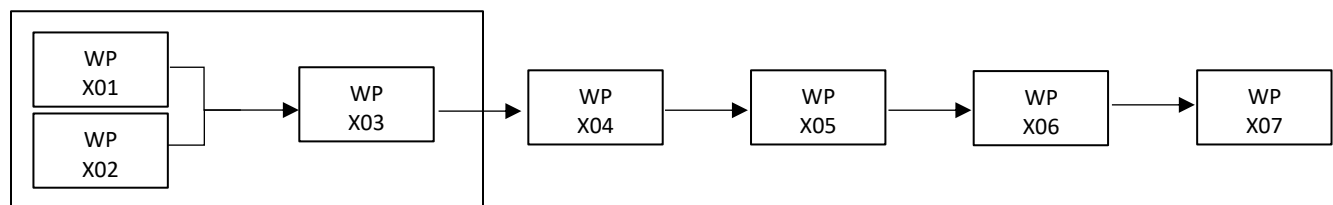
1. Project Plan
2. System Requirement Specification
3. User Interface
4. Technical Architecture
5. Data Modeling
6. Coding & Unit Testing
7. Integration & Quality Assurance

4.4 Activity Dependencies

The following table describes the dependencies of the deliverable work packages:

Work Package #	Work Package Description	Duration	Dependencies
X01	Project Plan	7 days	-
X02	System Requirement Specification	7 days	-
X03	Technical Architecture	14 days	X01, X02
X04	User Interface	7 days	X03
X05	Data Modeling	7 days	X04
X06	Coding & Unit Testing	14 days	X05
X07	Integration & System Testing	14 days	X06

The following Activity Network Diagram describes the above in more graphical detail:



Note that work package X04 is dependent on all work packages encapsulated by the larger boxes linked to its left. For instance, WP X05 may not start until WP X01- X04 has been finished.

4.5 Work Package Details

Work packages are listed below. A team member, indicated in bold, has been assigned as primarily responsible for each work package and will coordinate that package.

Project	Ember
Work Package	X01 - Project Plan (1 of 7)
Assigned To	Arkar Min , Lee Wan Xin, Jeslynn Teow Zhini, Ting Jie Yi Rachel, Daniel Benjamin Delago, Georgi Krasimirov Markov, Sean Tan Jun Yu
Effort	7 PD
Start Date	Tuesday, 18/09/2018
Purpose	To outline the requirements and aims of the project, as well as to ensure proper planning and implementation throughout the project.
Inputs	None
Activities	The Project Plan provides a summary of the project, including the objectives of developing Ember Application, and the proposed set of deliverables to be completed throughout the development of the software cycle. The tentative tasks allocation and estimated workload will be included and cost of the project planned out accordingly.
Output	A written document of the Project Plan Proposal.

Project	Ember
Work Package	X02 - System Requirement Specification (2 of 7)
Assigned To	Arkar Min , Lee Wan Xin, Jeslynn Teow Zhini, Ting Jie Yi Rachel, Daniel Benjamin Delago, Georgi Krasimirov Markov, Sean Tan Jun Yu
Effort	7 PD
Start Date	Tuesday, 04/09/2018
Purpose	To ensure that the system requirements of the project specified by the customer corresponds to the understanding of the software development team.
Inputs	Customer's requirements
Activities	Identify "the customer", interview customer, get the requirements of the customers and ensure that the requirements are targeted.
Outputs	A written document of the requirement specification.

Project	Ember
Work Package	X03 - Technical Architecture (3 of 7)
Assigned To	Sean Tan Jun Yu , Daniel Benjamin Delago, Georgi Krasimirov Markov, Arkar Min, Lee Wan Xin, Jeslynn Teow Zhini, Ting Jie Yi Rachel
Effort	14 PD
Start Date	Thursday, 30/08/2018
Purpose	To design and implement high-level architectural design
Inputs	Project Plan Work Packages (X01 to X03 inclusive).
Activities	To implement a high-level architectural design, the architecture of the software system to be built must first be defined. The various components required in the system have to be identified, and developers will have to decide on the relation between the varying components, and how they interact with each other in the software system. Designers will also decide on the required software and hardware infrastructures required by the application, including but not limited to the operating system on which the software is built, the language used to implement the software and the devices required to run the application. Design topics including maintainability, portability, and reusability will be addressed here as well.
Outputs	High Level Design and Architectural Specification.

Project	Ember
Work Package	X04 - User Interface (4 of 7)
Assigned To	Sean Tan Jun Yu , Daniel Benjamin Delago, Georgi Krasimirov Markov, Arkar Min, Lee Wan Xin, Jeslynn Teow Zhini, Ting Jie Yi Rachel
Effort	7 PD
Start Date	Tuesday, 04/09/2018
Purpose	To build a user interface between the system and the customer, ensuring that the user interface is user-friendly (for the elderly), and that the layout and format is clear and visible.
Inputs	User information
Activities	To get the user information and user request as required, and then display the dialog between the system and user, as well as the results of the request.
Outputs	User Interface

Project	Ember
Work Package	X05 - Data Modeling (5 of7)
Assigned To	Georgi Krasimirov Markov , Sean Tan Jun Yu, Daniel Benjamin Delago, Arkar Min, Lee Wan Xin, Jeslynn Teow Zhini, Ting Jie Yi Rachel
Effort	7 PD
Start Date	Tuesday, 11/09/2018
Purpose	To build the project's database.
Inputs	Project Plan Work Packages (X01 to X05 inclusive).
Activities	Analyse the data flow and entity relationships.
Outputs	A written document of the use case descriptions and database schema, Development of backend system.

Project	Ember
Work Package	X06 - Coding & Unit testing (6 of 7)
Assigned To	Georgi Krasimirov Markov , Sean Tan Jun Yu, Daniel Benjamin Delago, Arkar Min, Lee Wan Xin, Jeslynn Teow Zhini, Ting Jie Yi Rachel
Effort	14 PD
Start Date	Tuesday, 11/09/2018
Purpose	To implement the system as per the system requirements specification and other associated documents. This work package includes such additional activities as preliminary unit testing.
Inputs	Project Plan Work Package X06.
Activities	Programmers will implement the modules according to the design specifications noted in the Specification document. Individual units of source code will be tested to ensure correct functionality and usage.
Outputs	Source code and header files.

Project	Ember
Work Package	X07 - Integration & System Testing (7 of 7)
Assigned To	Lee Wan Xin , Arkar Min, Jeslynn Teow Zhini, Ting Jie Yi Rachel, Daniel Benjamin Delago, Georgi Krasimirov Markov, Sean Tan Jun Yu
Effort	14 PD
Start Date	Thursday, 27/09/2018
Purpose	The logical and syntactical errors that surfaces during the implementation of the system will be identified and corrections will be made to rectify the errors. The functionality of the software as well as the included algorithms will be tested to ensure efficiency and correctness. Black box testing, as well as white box testing will be conducted to check for logical errors within the system. Different inputs will also be tested to check for the various possible data flow of the system. The detailed testing plan will be documented in the Test Plan Document.
Inputs	Project Plan Work Package X07.
Activities	The Integration and Quality Assurance team will attempt to stimulate real world users' interaction with the system. Integration testing tests the overall performance of the system rather than individual functionalities and ensure that the system performance and integrity is maintained throughout the entire process. Heuristics assessment should also be focused on in this work package as the intelligence components such as the matching algorithm will define the eventual system success.
Outputs	Quality plan.

5 Project Estimates

5.1 Code Size Estimation using Function Points

We calculated unadjusted function point based on the complexity of functions provided by this system. Code size is then estimated by adjusted function point.

5.1.1 Unadjusted Function Points

EMBER supports the following proposed functions:

Customer:

- Gather general information, preferences and interests from customer
- Collect and analyze customer preferences
- Provide sorted matches based on customer's entered preferences
- Chat with other matched customers

Administrator:

- List customer preferences
- List customer matches
- List Chats

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. The detailed evaluation of the complexity is as follows:

Rating Inputs:

- Gathering customer's information: (name, age, address, picture)
- Preferences (i.e. hobbies, gender preferences) and Constraints (i.e. sorting the results by most compatible, age group, gender)

Files Type Referenced (FTR)	Data Elements		
	1-4	5-15	Greater than 15
Less than 2	Low (3)	Low (3)	Average (4)
2	Low (3)	Average (4)	High (6)
Greater than 2	Average (4)	High (6)	High (6)

Rating Outputs:

- Displaying a list of the results matching the customer's desired preferences (hobbies, gender preferences, location, language)
- Displaying Chat
- Displaying customer's account information

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (3)	Low (3)	Average (3)
2 or 3	Low (3)	Average (4)	High (6)
Greater than 3	Average (3)	High (6)	High (6)

Rating Inquiries:

- Matching according the customer's preferences
- Selecting compatible matches
- Selecting matched customers' information
- Getting Chat logs

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (3)	Low (3)	Average (4)
2 or 3	Low (3)	Average (4)	High (6)
Greater than 3	Average (4)	High (6)	High (6)

Rating Logical Files:

- Matching via preferences
- Customer Account

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (6)	Low(6)	Average (10)
2 to 5 RET	Low (6)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

Rating Interfaces:

- 1 External Files Referenced (Location)

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (6)	Low(6)	Average (10)
2 to 5 RET	Low (6)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

Summary of above analysis:

Element	Complexity	Detail
Inputs	Low	Gathering customer's information
	Low	Preference and Constraints
Logical Files	High	Matching via preferences
	Medium	Customer Account
Outputs	High	Display Matching Results
	Low	Display Chat
	Low	Display Customer Information
Inquiries	High	Matching based on Customer's preferences
	Low	Selecting Chat logs
	Low	Selecting customer account information
Interfaces	Low	Location

Calculation of Unadjusted Function Points:

Characteristic	Low		Medium		High	
Inputs	1	× 3	0	× 4	0	× 6
Outputs	1	× 3	0	× 5	0	× 7
Inquiries	1	× 3	0	× 4	0	× 6
Logical Files	1	× 6	0	× 10	1	× 15
Interfaces	1	× 6	0	× 7	0	× 10
Unadjusted FP	21		0		15	
Total=L+M+H	36					

5.1.2 Adjusted Function Points

Influence Factors	Score	Detail
Data Communications	5	Application is more than a front-end, and supports more than one type of teleprocessing communications protocol.
Distributed Functions	4	Distributed processing and data transfer are online and in both directions.
Performance	4	Response time or throughput is critical during all business hours. No special design for CPU utilization was required. Processing deadline requirements with interfacing systems are constraining.
Heavily used	1	Some security or timing considerations are included.
Transaction rate	3	Daily peak transaction period is anticipated.
On-line data entry	5	More than 30% of transactions are interactive data entry
End-user efficiency	5	Four to five of the efficiency designs are included
On-line data update	5	Online update of major internal logical files is included.
Complex processing	2	Any one of the complex components
Reusability	2	The application was specifically packaged and/or documented to ease re-use, and the application is customized by the user at source code level.
Installation Ease	1	No special considerations were stated by the user <i>but</i> special setup is required for installation.
Operational Ease	5	Effective start-up, back-up, and recovery processes were provided, but no operator intervention is required (count as two items).
Multiple sites	0	User requirements do not require considering the needs of more than one user/installation site.
Facilitate change	1	Flexible query and report facility is provided that can handle complex requests, for example, <i>and/or</i> logic combinations on one or more internal logical files (count as three items).
Total score	43	
Influence Multiplier		
= Total score \times 0.01 + 0.65 = $43 \times 0.01 + 0.65 = 1.08$		
Adjusted FP		
= Unadjusted FP \times Influence Multiplier = $36 \times 1.08 = 38.88$		

Scoring (0 – 5)
0 = No influence
1 = Insignificant influence
2 = Moderate influence
3 = Average influence
4 = Significant influence
5 = Strong influence

5.1.3 Lines of Code

According to Capers Jones statistics, each Function Point requires 29 lines of code if the application is implemented using C++.

Therefore, we have: **Lines of Code** = $38.88 \text{ FP} \times 29 \text{ LOC/FP} = \mathbf{1127.52 \text{ LOC}}$

5.2 Efforts, Duration and Team Size Estimation

To estimate the effort and duration required for the project, we use function points as the basis to calculate Effort, Duration, Team size and finally the schedule. The estimates are expanded to account for project management and extra contingency time to obtain the total average effort estimates. From these averages, the duration of each work package in working days is estimated based on the following calculations.

- Working days include 5 days in a week.
- Effort = Size / Production Rate = $(1127.52 \text{ LOC}) / (43 \text{ LOC/PD})^1 = 26 \text{ PD}$
- Duration = $3 \times (\text{Effort})^{1/3} = 3 \times (26)^{1/3} = 8.88 \text{ Days}$
- Initial schedule = $8.88 \text{ Days} / 5 \text{ days a week} = 1.77 \text{ Weeks}$
- Team size = $26 \text{ PD} / 8.88 \text{ D} = 2.92 \text{ P} = 3 \text{ Persons}$
- Working hours include 2.5 hours in a working day.
- Total person-hours (PH) = $26 \text{ PD} \times 2.5 \text{ hours} = 65 \text{ PH}$

5.2.1 Distribution of Effort

1990's Industry Data	Work Package	Distribution	Estimates
Preliminary Design 15 %	Project Plan	7.5%	11.25
	Requirement Specification	7.5%	11.25
Detailed Design 25 %	User Interface	7%	10.5
	Technical Architecture	11%	16.5
	Data Modeling	7%	10.5
Code & Unit Testing 30 %	Code & Unit testing	25%	37.5
	Online Documentation	5%	7.5
Integration & Test 30 %	Integration & Quality Assurance	30%	45
	Extrapolated total effort		150
	3% for project management		4.5
	4% for contingency		6
	Total effort		154.5

These duration estimates are based on the assumption that each team member works an equal amount on any given work package.

5.3 Cost Estimates

Hardware:

Developer Workstations:

Acer Veriton X2640G SFF	Total: \$0.00
Intel Core i7-8700k	
16GB RAM	
40GB IDE Drive	

Software:

GNU, Apache, or Other Free License-based Software:

Apache Web Server	\$0.00
-------------------	--------

Software License Provided by Third Party:

Microsoft Office 365 ProPlus	\$0.00
Google Drive	\$0.00

Other Resources:

Staff:

7 Employees with 154.5 working hours with \$10.00/hour	\$10, 815.00
---	--------------

Stationery:

Paper, photocopying, and other miscellaneous cost	\$68.00
--	---------

Total: \$10, 883.00

6 Risk Management

Besides the general risk management, the following risks have been identified for the Project Ember:

A) Users demanded to add more features to the product

Impact Severity: Very Low

Probability: 40%

Impacts: Additional effort from the developers will be required to make a plan to develop additional features and integrated them into the product.

Risk Reduction: Arrange bonus for developers to keep them motivated

B) Key Staffs are ill during critical time

Impact Severity: High

Probability: 10%

Impacts: Negatively impact the schedule

Risk Reduction: Organize the team in a way that people understand each other's jobs

C) Server is not able to fulfill the user's demands

Impact Severity: High

Probability: 15%

Impacts: The application will not work as expected

Risk Reduction: Investigate other alternative high performance server

D) System size underestimated

Impact Severity: Moderate

Probability: 35%

Impacts: More work will be needed to be spent on design and coding, could negatively impact the schedule.

Risk Reduction: Update estimates as the project progresses

E) Misinterpretation of requirements

Impact Severity: High

Probability: 15%

Impacts: Developer could wrongly interpret the requirements and implement wrong features

Risk Reduction: Make a good quality plan and Lead developer must ensure his team members clearly understand the user's requirements

F) Time required to implement Chat function underestimated

Impact Severity: High

Probability: 10%

Impacts: This could impact the software delivery time

Risk Reduction: Try to finish other features as fast as possible to get more time to implement chat functions.

G) Time required for testing and debugging is underestimated

Impact Severity: High

Probability: 15%

Impacts: The application will not work as expected and it could impact the software delivery time

Risk Reduction: Assign enough man powers for testing and debugging processes

7 Quality Assurance Plan

The project will achieve the quality assurance by following the standard set by the company. The specific procedure and details shall be provided in the Quality Plan.

Specific test procedures and details shall be provided in the Module/System Test Plan.

In addition, project EMBER will make use of three testing methodologies:

- Unit Testing: involve testing system components individually
- In-Place Testing: involve testing of the whole system as a unit
- Acceptance Testing: involve testing of each feature by user to ensure it meets the user requirement

Furthermore, these methodologies will be used to test two important aspects for the project Ember:

- System function will be tested to ensure that software flaws are eliminated
- Algorithmic function will be tested to ensure that matching algorithm perform well to give value to the users

Project Ember documents user requirements and project requirements carefully and will make sure every team member has same understanding about them all the time. Since one of the goals of the project to achieve robustness, testing will be done considering all extreme cases (such as no matching people around the user's living area).

8 Monitoring & Control

Many procedures are required in order to be able to successfully monitor the progress of a software project. Some of the most important are:

Identification of major project risks: Early identification of major risks to the project allows for placement of preventative measures before problems can develop. Major risks have been identified in the Risk Management section of this document, along with the measures being taken to avoid them.

Regular reviews of project progress: Project Ember team members will meet weekly to review the progress of all project tasks, including management, planning, analysis, development and testing. At those meeting, project members will ensure that every task meets the quality requirements.

Quantitative measurement of resource consumption: Estimates of the Project Ember's resource requirements, primarily in terms of human resources, can provide a quantitative measurement of project progress when compared to progress in terms of project milestones. The percentage estimates of each milestone's resource requirements provided in this document allow for easy progress tracking.

Timeline Planning and task decomposition: This document outlines an estimated timeline for the project. A accurate timeline can be assembled by hierarchically decomposing tasks into measurable subcomponents and estimating requirements for each. At the same time, this decomposition can assist in task assignment and balancing. Throughout the implementation phase, these subcomponents can allow for fine-grained measurement of progress. Project subcomponents and timeline estimates are included in the Estimates and Work Breakdown Structure sections of this document.