

School of Computer Science and Engineering

CZ3002 Advanced Software Engineering

**Project Plan**

Project Name: HangOut

****

Group Name: Mac & Cheese

Lab Group: TDDP1

| **Group Members** |
| --- |
| Jethro Phuah An Ping (Team Leader) |
| Lam Zhi Fah (Deputy Team Leader) |
| Alicia Chua Jieying |
| Ernest Ang Cheng Han |
| Ong Jing Heng Shaun |
| Ong Sim Hao |
| Shannon Tan Xinyi |

# Revision History

| **Revision Number** | **Date** | **Primary Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 1.0 | Oct 1st, 2021 | Jethro Phuah, Alicia Chua, Shannon Tan, Lam Zhi Fah, Shaun Ong, Ong Sim Hao, Ernest Ang | First version |
|  |  |  |  |

**Table of Contents**

[**Revision History**](#_heading=h.94w22oti8al4) **1**

[**1 Introduction**](#_heading=h.j6b62kz989k2) **4**

[1.1 Project Overview](#_heading=h.31zrms529c19) 4

[1.2 Project Description and Scope](#_heading=h.726khunwxbb1) 4

[**2 Project Organization**](#_heading=h.irwusy3ku6fp) **6**

[2.1 Team Structure](#_heading=h.3xjq94558y23) 6

[2.2 Roles and Responsibilities](#_heading=h.m1z7zo4rx3tu) 6

[2.3 Team Communication](#_heading=h.y95lhsu5fiaz) 8

[**3 Process Definition**](#_heading=h.yy0a6mdixb6o) **9**

[3.1 Lifecycle Model](#_heading=h.ok8sbuc59rxs) 9

[**4 Schedule**](#_heading=h.c5xfros1x9r2) **10**

[4.1 Activity Dependencies and Schedule](#_heading=h.hxwfbdfa9pn) 10

[4.2 Work Breakdown Structure](#_heading=h.vwskdyall74r) 11

[4.3 Work Packages](#_heading=h.s67ui14coru) 13

[4.4 Activity Dependencies](#_heading=h.i8luuvvwdio5) 13

[4.5 Work Package Details](#_heading=h.7w5ffjuzf6f9) 15

[**5 Project Estimates**](#_heading=h.ejn19uz0w0qi) **19**

[5.1 Code Size Estimation using Function Points](#_heading=h.13iug5qljqwi) 19

[5.1.1 Unadjusted Function Points](#_heading=h.ruw7kqah08oi) 19

[5.1.2 Adjusted Function Points](#_heading=h.pcz7im4e9yay) 21

[5.1.3 Lines of Code](#_heading=h.8l28l8rvv42a) 22

[5.2 Efforts, Duration and Team Size Estimation](#_heading=h.brfwvgn3qkbq) 23

[5.3 Cost Estimates](#_heading=h.3ekvrgklk6h5) 24

[**6 Product Checklist**](#_heading=h.cjcublynngjp) **26**

[**7 Best Practice Checklist**](#_heading=h.f34hmdbxwtdq) **27**

[**8 Risk Management**](#_heading=h.phl6xkfnmjpn) **28**

[**9 Quality Assurance**](#_heading=h.yj86lt88cdos) **30**

[**10 Monitoring & Control**](#_heading=h.gsic0ewokxbk) **31**

# 1 Introduction

# 1.1 Project Overview

HangOut is a mobile application which is open to members of the public in Singapore. It allows users to find places of interest, food hotspots and events based on a preferred location or his/her current location, and provides them with information about the selected options. It also enables users to create or join meetups at a desired location with other users to carry out certain activities.

## 1.2 Project Description and Scope

HangOut is a joint government project initiated by the Singapore Tourism Board (STB) and Ministry of Health (MOH) to promote outside interaction and improve the overall health of Singaporeans.

STB’s client has been given permission to access Singapore roadmaps and facilities and conduct research relating to the project requirements.

Hangout is an intelligent search app that takes into account user input and filters the findings according to the user preferences. Possible user inputs are given as follows:

• The user wishes to visit the west side of Singapore;

• The user intends to eat Japanese cuisine.

• The user intends to arrive at the place of interest at night;

• The user wishes to spend within his budget of $100;

These user inputs serve as a guideline for the search app and do not need to be strictly followed. After deciding on the categories of user input the app will be following, final amendments to these user inputs will be discussed and analysed in a separate document.

Given these user preferences, the app filters and presents all the possible options available to the user.

Hangout also helps the user to find interested parties to go out with and explore the suggested places of interest by providing the following functionalities:

• The user can initiate meetups at a scheduled time and place.

• The user can specify the group size and the type of interests they prefer.

• The user can cancel the meetup anytime before the meetup starts.

Lastly, Hangout will periodically suggest the latest social events to users and keep them up to date on the social happenings in Singapore. As such, users will have a reason to venture outside and hangout.

STB’s client shall provide the deliverables outlined in this document, along with database schema, a test data set, and all source code.

The system shall include all necessary user interfaces to satisfy the user’s requirements and preferences.

Should the Geographical Information System (GIS) provided by STB not be sufficient for query usage for the Hangout, STB’s client may be required to purchase a detailed database containing the coordinates of the places of interest around Singapore.

# 2 Project Organization

## 2.1 Team Structure

The following is the list of executive roles, as required by CMM level 3.

• Senior Management: Jethro Phuah

• Software Engineering Project Group: Lam Zhi Fah, Ernest Ang Cheng Han, Ong Jing Heng Shaun, Shannon Tan Xinyi

• Software Quality Assurance Engineer: Ong Sim Hao

• Scrum-in-charge : Alicia Chua Jieying

## 2.2 Roles and Responsibilities

**Project Manager: Jethro Phuah**

• Oversees project progress

• Approves and executes project plan

• Assigns tasks and reports status of project to team members

• Ensures that product requirements are met

• Manages and motivates team members

• Builds relationships among the client and various organization departments

**SCRUM Master: Alicia Chua Jieying**

• Formulates goals

• Analyses and documents core processes and systems

• Ensures alignment of business model and technology

• Develops concepts of the system for designers

• Creates requirement specification document

• Assists Project Manager in customer relations

**UI/UX Designer: Shannon Tan Xinyi**

• Designs User Interface & User Experience

• Creates user manual

**Front-end Developer: Lam Zhi Fah**

• Ensures usability of the system is in accordance with the planned design

• Ensures stability and response time of the system meet the requirements

**Back-end Developer: Ernest Ang Cheng Han, Ong Jing Heng Shaun**

• Implements product based on detailed design document

• Ensures functionality of the system

**Quality Assurance Engineer: Ong Sim Hao**

• Ensures product meet client’s requirement

• Designs testing strategies

• Performs usability testing to detect error and bugs

## 2.3 Team Communication

Hangout communication channels include the following:

• Weekly meetings are held on Wednesdays.

• Group announcements are sent through email: [hangoutteam@gmail.com](mailto:hangoutteam@gmail.com)

• Daily updates and progress can be viewed from Hangout Project Trello Board

• Message or phone calls through Microsoft Teams will be held as necessary

• Project status meetings are held monthly or at every milestone through conference calls

• To work more efficiently on specific problems, the team can be split up into subgroups as necessary

# 3 Process Definition

# 3.1 Lifecycle Model

Our team intends to use the Agile Software Development Life Cycle (SDLC) throughout the Hangout project. The Agile SDLC is the combination of both iterative and incremental process models. It focuses on process adaptability and customer satisfaction by rapid delivery of a working software product. While we do concur that there are other methodologies out there such as the Incremental Development Model or the Waterfall SDLC which might be more simple to follow, have chosen to use the Agile SDLC due to several reasons.

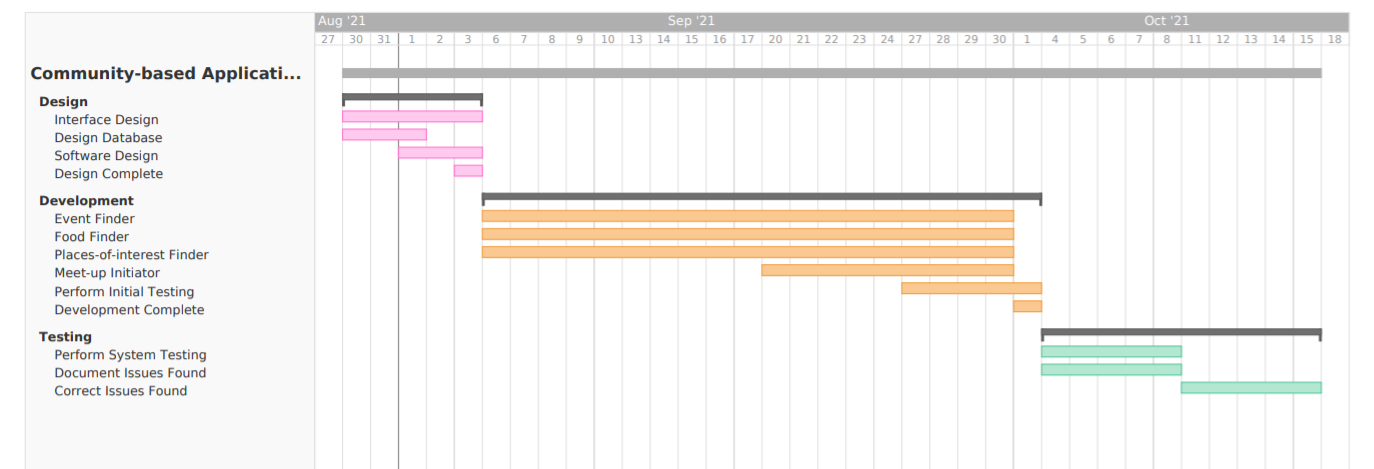
First of all, the Agile SDLC is much more flexible compared to the traditional Waterfall SDLC due to repeated iterations involving design, coding, unit testing, integration, and quality assurance. The Waterfall SDLC is not a viable choice due to the short timeline available for Hangout to reach delivery quality and the fact that it needs to be able to accommodate changing requirements.

Another reason why the Agile SDLC is chosen is due to the nature of our app. Since our Hangout app is built for the public masses, users’ feedback and testing are necessary to further update and improve the app such that the algorithms behind it behave in an accurate and logical manner to provide a more effective experience for users. As such, the Agile SDLC would be perfect for such a scenario.

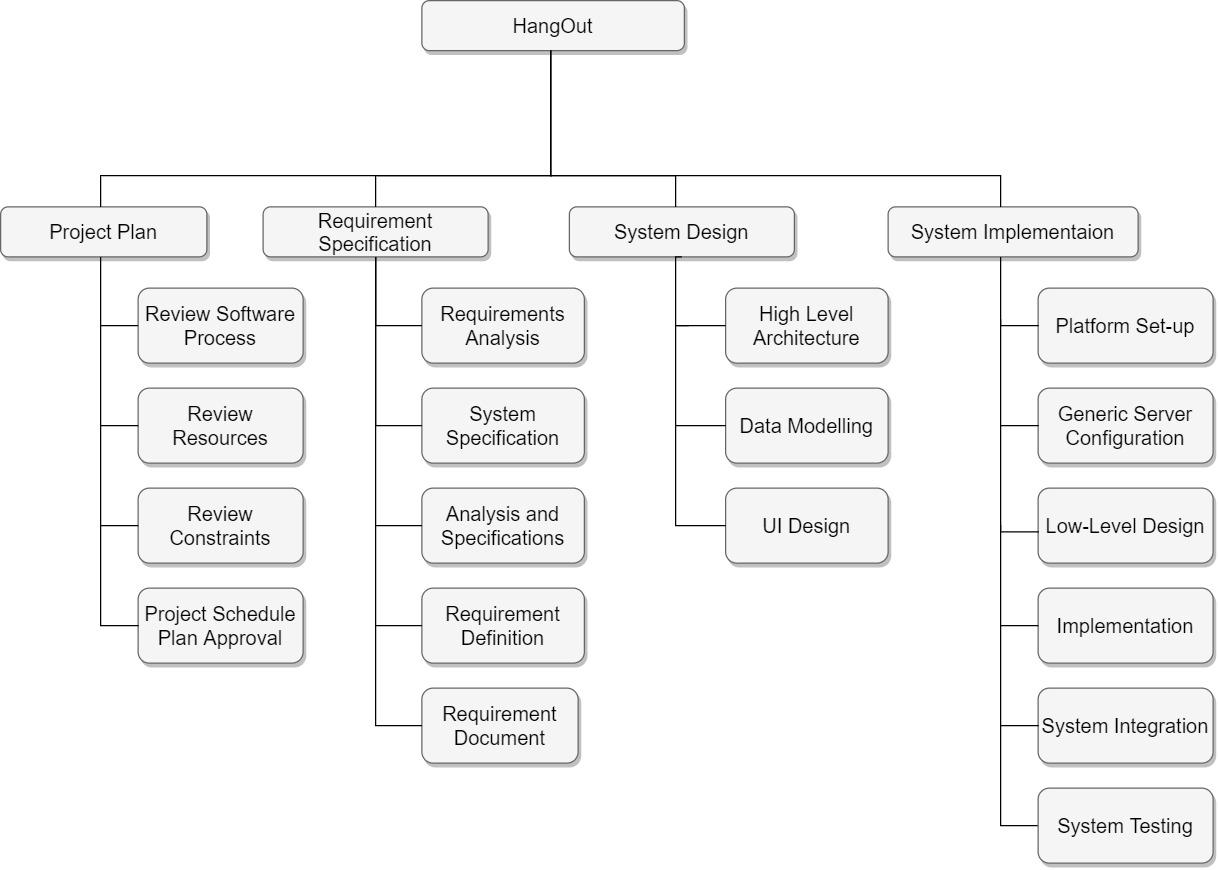
Last but not the least, the Agile SDLC breaks down the product into smaller incremental builds. This fits well with our Hangout App as we do have several use cases that can be completed on its own namely the Find Food, Events and Places of Interest vs the Join Meetups. Under the Agile SDLC, we are able to build the app by implementing the use cases periodically before sending it for client testing.

# 4 Schedule

## 4.1 Activity Dependencies and Schedule



## 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. 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 |
| --- | --- | --- | --- |
| 01 | Project Plan | 4 days | -- |
| 02 | Requirement Specification | 6 days | -- |
| 03 | User Interface | 5 days | -- |
| 04 | Technical Architecture | 20 days | 01, 02, 03 |
| 05 | Data Modeling | 5 days | 04 |
| 06 | Coding & Unit Testing | 17 days | 05 |
| 07 | Integration & System Testing | 7 days | 06 |

## 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** | HangOut |
| --- | --- |
| **Work Package** | 01. Project Plan |
| **Assigned To** | Jethro Phuah An Ping, Alicia Chua Jieying, Shannon Tan Xinyi, Ernest Ang Cheng Han, **Ong Sim Hao**, Shaun Ong Jing Heng, Lam Zhi Fah |
| **Effort** | 4 PD |
| **Start Date** | 23 September 2021 |
| **Purpose** | To determine the overview of the project, to be refined in later work packages. |
| **Inputs** | None |
| **Activities** | This work package includes providing an overview of the project, its objectives and a set of proposed project deliverables throughout the development of the software cycle. The members responsible for this work package will also translate ideas developed in group discussions into a formal report. |
| **Outputs** | A written document of the Project Plan |

| **Project** | HangOut |
| --- | --- |
| **Work Package** | 02. Requirement Specification (2 of 7) |
| **Assigned To** | Jethro Phuah An Ping, Alicia Chua Jieying, **Shannon Tan Xinyi**, Ernest Ang Cheng Han, Ong Sim Hao, Shaun Ong Jing Heng, Lam Zhi Fah |
| **Effort** | 6 PD |
| **Start Date** | 27 September 2021 |
| **Purpose** | To establish a common understanding between the customer and the software project team of the customer’s requirements and needs that are addressed by the project. |
| **Inputs** | Customers’ requirements |
| **Activities** | This work package includes identifying target audiences who are deemed to be “customers”, carrying out individual interviews or focus groups with the customers, writing and analysing customer requirements, and building requirements in the project. |
| **Outputs** | A written document of the Requirement Specification |

| **Project** | HangOut |
| --- | --- |
| **Work Package** | 03. User Interface (3 of 7) |
| **Assigned To** | Jethro Phuah An Ping, **Alicia Chua Jieying**, Shannon Tan Xinyi, Ernest Ang Cheng Han, Ong Sim Hao, Shaun Ong Jing Heng, Lam Zhi Fah |
| **Effort** | 5 PD |
| **Start Date** | 3 October 2021 |
| **Purpose** | To develop an effective communication medium between the system and the customer such that it is easy to use and friendly to the customer. |
| **Inputs** | User information, requirements specification |
| **Activities** | This work package includes identifying interface options and creating screen layouts that forms the basis of an user interface prototype. The interface is able to obtain the user information and request so that it will display the dialog between system and user and the results of request. |
| **Outputs** | User interface and a written document of the User Interface report |

| **Project** | HangOut |
| --- | --- |
| **Work Package** | 04. Technical Architecture (4 of 7) |
| **Assigned To** | **Jethro Phuah An Ping**, Alicia Chua Jieying, Shannon Tan Xinyi, Ernest Ang Cheng Han, Ong Sim Hao, Shaun Ong Jing Heng, Lam Zhi Fah |
| **Effort** | 20 PD |
| **Start Date** | 8 October 2021 |
| **Purpose** | To implement the high level architecture design of the project. |
| **Inputs** | Project Plan, Requirement Specification, User Interface |
| **Activities** | This work package, which mainly involves high level design, includes defining the architecture of the software system and identifying the various components and how they are inter-related to and communicate with each other. It is also important to note that designers need to decide on the software and hardware infrastructures, such as what operating system on which the software is built, the language used to implement the software, and addressing certain design topics including maintainability, portability, and reusability. |
| **Outputs** | A written document on the High Level Design and Architectural Specification |

| **Project** | HangOut |
| --- | --- |
| **Work Package** | 05. Data Modelling (5 of 7) |
| **Assigned To** | Jethro Phuah An Ping, Alicia Chua Jieying, Shannon Tan Xinyi, **Ernest Ang Cheng Han**, Ong Sim Hao, Shaun Ong Jing Heng, Lam Zhi Fah |
| **Effort** | 5 PD |
| **Start Date** | 28 October 2021 |
| **Purpose** | To build the project’s database. |
| **Inputs** | Project Plan, Requirement Specification, User Interface, Technical Architecture |
| **Activities** | This work package includes analysing the data flow relationships and entity relationships. |
| **Outputs** | A written document of the Data Modelling and project database |

| **Project** | HangOut |
| --- | --- |
| **Work Package** | 06. Coding and Unit Testing (6 of 7) |
| **Assigned To** | Jethro Phuah An Ping, Alicia Chua Jieying, Shannon Tan Xinyi, Ernest Ang Cheng Han, Ong Sim Hao, **Shaun Ong Jing Heng**, Lam Zhi Fah |
| **Effort** | 17 PD |
| **Start Date** | 2 November 2021 |
| **Purpose** | To implement the system as per the requirements specification and other associated documents. This work package includes such additional activities as preliminary unit testing. |
| **Inputs** | Project Plan, Requirement Specification, User Interface, Technical Architecture, Data Modelling |
| **Activities** | This work pacakge includes implementing the modules according to the design specifications noted in the Specification document, preparing unit test plans and examining different paths through the modules. |
| **Outputs** | A written document of the Unit Test report, source code and header files |

| **Project** | HangOut |
| --- | --- |
| **Work Package** | 07. Integration and System Testing (7 of 7) |
| **Assigned To** | Jethro Phuah An Ping, Alicia Chua Jieying, Shannon Tan Xinyi, Ernest Ang Cheng Han, Ong Sim Hao, Shaun Ong Jing Heng, **Lam Zhi Fah** |
| **Effort** | 7 PD |
| **Start Date** | 19 November 2021 |
| **Purpose** | To identify and fix logical and syntactical errors faced during the implementation of the system, and set up drivers and stubs to observe how the module responds to various inputs. Black box testing as well as white box testing are conducted to check for logical errors. All the testing procedures will be documented in the Test Plan report. If problems are found, they will be noted and fixed at the earliest possible time. The team will also evaluate the project’s overall performance to provide confidence that the project satisfies the relevent quality standards. |
| **Inputs** | Project Plan, Requirement Specification, User Interface, Technical Architecture, Data Modelling, Coding and Unit Testing |
| **Activities** | This work pacakge includes examing issues such as system performance and integrity and use metrics to develop strategies for improving software processes. The Integration testing team may try to simulate how a user might interact with the system. Similar to Unit Testing, Integration Testing may require the development of stubs and drivers as well, but here this is more geared towards the higher (overall system) level. Heuristics assessment plays an important role in this work package, as intelligence components will define eventual system success. Thus, combining all the above factors will help to improve the quality of the final product. |
| **Outputs** | A written document of the Test Report and Quality Assurance Report |

# 5 Project Estimates

## 5.1 Code Size Estimation using Function Points

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

### 5.1.1 Unadjusted Function Points

Hangout App supports the following proposed functions:

Public User:

* Register for his/her account
* Login to his/her account
* Edit/View his/her profile
* Find food options
* Find places of interest
* Find events happening around them
* Create meetups and join meetups

Administrator:

* Login to system via registered email address
* Add places of interest
* Add food options
* Add events

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 user’s information during registration: (user’s email address, first name, last name, address)
* Preferences (i.e. food cuisine, price) and Constraints (i.e. sorting the results by cheapest price, event location)
* Creating a meetup: (meetup time, address, short description and name)

| **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 user’s search criteria (food option, places of interest, event and available meetups)
* Displaying a list of users who have requested to join the meetup
* Receipt email confirmation when user successfully joined a meetup
* Displaying user’s account information

| **File Types Referenced (FTR)** | **Data Elements** | | |
| --- | --- | --- | --- |
|  | 1-5 | 6-19 | Greater than 19 |
| less than 2 | Low (4) | Low (4) | Average (5) |
| 2 or 3 | Low (4) | Average (5) | High (7) |
| Greater than 3 | Average (5) | High (7) | High (7) |

**Rating Inquiries:**

* Selecting food options according to the user’s criteria
* Selecting the events according to the user’s criteria
* Selecting the places of interest according to the user’s criteria
* Selecting a meetup to join

| **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:**

* Database on Food, Events and Places of Interest
* User’s created meetups and joined meetups
* User’s Account

| **Record Element Types (RET)** | **Data Elements** | | |
| --- | --- | --- | --- |
|  | 1 to 19 | 20 - 50 | 51 or More |
| 1 RET | Low (7) | Low(7) | Average (10) |
| 2 to 5 RET | Low (7) | Average (10) | High (15) |
| 6 or More RET | Average (10) | High (15) | High (15) |

**Rating Interfaces:\**

* 2 External Files Referenced (Singapore Districts, Singapore Map)

| **Record Element Types (RET)** | **Data Elements** | | |
| --- | --- | --- | --- |
|  | 1 to 19 | 20 - 50 | 51 or More |
| 1 RET | Low (7) | Low(7) | Average (10) |
| 2 to 5 RET | Low (7) | Average (10) | High (15) |
| 6 or More RET | Average (10) | High (15) | High (15) |

Summary of above analysis:

| **Element** | **Complexity** | **Detail** |
| --- | --- | --- |
| Inputs | Low | Gathering user’s information during registration |
| Low | Preference and Constraints |
| Medium | Creating a meetup |
| Logical Files | High | User’s created meetups and joined meetups |
| Medium | User’s Account |
| Medium | Database on Food, Event and Places of Interest |
| Outputs | High | Displaying a list of the results matching the user’s search criteria |
| Low | Displaying a list of users who have requested to join the meetup |
| Medium | Receipt email confirmation when user successfully joined a meetup |
| Low | Displaying user’s account information |
| Inquiries | Medium | Selecting food options according to the user’s criteria |
| Medium | Selecting the events according to the user’s criteria |
| Medium | Selecting the places of interest according to the user’s criteria |
| Medium | Selecting a meetup to join |
| Interfaces | Medium | Singapore Districts, Singapore Map |

Calculation of Unadjusted Function Points:

| **Characteristic** | **Low** | | **Medium** | | **High** | |
| --- | --- | --- | --- | --- | --- | --- |
| Inputs | 2 | × 3 | 1 | × 4 | 0 | × 6 |
| Outputs | 2 | × 4 | 1 | × 5 | 1 | × 7 |
| Inquiries | 0 | × 3 | 4 | × 4 | 0 | × 6 |
| Logical Files | 0 | × 7 | 2 | × 10 | 1 | × 15 |
| Interfaces | 0 | × 5 | 2 | × 7 | 0 | × 10 |
| **Unadjusted FP** | 14 |  | 59 |  | 22 |  |
| **Total=L+M+H** | 95 | | | | | |

### 

### 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 | 3 | 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 | 2 | Some security or timing considerations are included. |
| Transaction rate | 3 | Daily peak transaction period is anticipated. |
| On-line data entry | 5 | More than 50% of transactions are interactive data entry |
| End-user efficiency | 2 | Four to five of the efficiency designs are included |
| On-line data update | 3 | Online update of major internal logical files is included. |
| Complex processing | 1 | Any one of the complex components |
| Reusability | 4 | 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 *but* special  setup is required for installation. |
| Operational Ease | 1 | 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, and/or logic combinations on one or more internal logical files (count as three items). |
| Total score | 35 | |
| **Influence Multiplier**  = Total score × 0.01 + 0.65 = 35 × 0.01 + 0.65 = 1.00 | | |
| **Adjusted FP**  = Unadjusted FP × Influence Multiplier = 95 × 1.00 = 95 | | |

| **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** = 95 FP × 29 LOC/FP = **2755 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 = (2755 LOC) / (39 LOC/PD)1 = 70.6 PD

• Duration = 3 × (Effort) 1/3 = 3 × (70.6) 1/3 = 12.4 Days

• Initial schedule = 12.4 Days / 5 days a week = 2.48 Weeks

• Team size = 70.6 PD / 12.4 D = 5.69 P = 6 Persons

• Working hours include 8 hours in a working day.

• Total person-hours (PH) = 70.6 PD × 8 hours = 564.8 PH

**5.2.1 Distribution of Effort**

| **1990’s Industry Data** | **Work Package** | **Distribution** | **Estimates** |
| --- | --- | --- | --- |
| Preliminary Design (18%) | Project Plan | 9% | 50.83 |
| Requirement Specification | 9% | 50.83 |
| Detailed Design (25%) | User Interface | 7% | 39.54 |
| Technical Architecture | 11% | 62.13 |
| Data Modeling | 7% | 39.4 |
| Code & Unit  Testing (26%) | Code & Unit testing | 21% | 118.61 |
| Online Documentation | 5% | 28.24 |
| Integration & Test (31%) | Integration & Quality Assurance | 31% | 175.09 |
|  | **Extrapolated total effort** |  | 564.8 |
|  | 2% for project management |  | 11.30 |
|  | 3% for contingency |  | 16.94 |
|  | **Total effort** |  | 593.04 |

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

1 Lines of code per Person Day statistics based on Industrial Benchmarks, 1997: 31 LOC/PD for United States; 62 LOC/PD for Canada

## 

## 5.3 Cost Estimates

**Hardware:**

**Developer workstations:**

| **7** - ASUS VivoBook | **Total: $0.00** |
| --- | --- |
| Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz |
| 16GB RAM |
| 1TB SSD |

**Software:**

**Firebase, or Other Free License-Based Software**

| Firebase Web Server | $0.00 |
| --- | --- |

**Software License Provided by Third Party:**

| Microsoft Office 2019 | $0.00 |
| --- | --- |
| Microsoft Project 2019 | $0.00 |

**Other Resources:**

**Staff:**

| 7 Employees with 593.04 hrs with 20.00/hr | $11860.80 |
| --- | --- |

The customer will supply the required hardware and software necessary to run the Firebase server. Mac and Cheese is not responsible in anyway for supplying said systems. Mac and Cheese ’s hardware and software responsibilities relate only to our own development needs to accomplish the project we have been asked to complete, and which has been described in the introduction section of this document. Mac and Cheese will also demonstrate the completed product

# 6 Product Checklist

The plan is that the items listed below will be delivered on the stated deadlines.

| **Project Deliverable** | **Estimated Deadline** |
| --- | --- |
| Project Plan | Oct 6th, 2021 |
| Requirements Specification | Oct 20th , 2021 |
| Design Document | Oct 27th, 2021 |
| Module/System Test Plan | Nov 10th , 2021 |
| System Release (Demo) | Nov 17th, 2021 |

# 7 Best Practice Checklist

| **Practice** | 9 |
| --- | --- |
| Every step must be properly documented in a standardized format. |  |
| The Software Requirements Specifications document must contain complete functional specifications. Pay attention to all requirements such as ambiguity, completeness, accuracy, and consistency. |  |
| As complexity management is difficult, try to keep the product simple by:  • Minimizing interfaces between modules, procedures and data.  • Minimizing interfaces between people to reduce exponential communication cost  • Avoiding product functions that are over the top and unnecessary as long as the core functionality meets the customer requirements |  |
| High visibility and transparency must be maintained at all times so progress can be measured. Should deadlines fail to be met, management action has to be taken promptly. In order for high visibility to be possible, the manager must have good rapport with his or her employees. Code done by the developers should be reviewed regularly for appropriateness and design. |  |
| Account for changes regularly.  • All manuals designs, test, source code should have revision numbers and dates revision history comments, change marks to indicate the changes  • New revisions should be approved before being made and checked for quality and compliance after being made  • Use a configuration management system and make processes  • Maintenance is mandatory and should be conducted regularly. |  |
| Ensure to do accurate time estimation for: time, effort, overhead, meeting time. In case of underestimation, swift management action needs to be taken. |  |
| Regular code reviews are a much more efficient method to find software defects. Plan and manage code reviews between team members |  |
| Software testing will use both black box and white box testing to detect software defects. It will involve unit, functional, integrating and acceptance testing. |  |

# 8 Risk Management

The following risks have been identified for the Hangout Project. We have chosen to classify impact severity on a scale of low, medium and high.

**Unanticipated changes to product requirements:**

Impact Severity: High

Probability: 25%

Impact on Project: Changes to product requirements may range from small edits in existing product requirements, to the need to completely redesign or implement a new product requirement. The former can be easily implemented with minimal impacts, while the latter may have catastrophic consequences on the project, such as delays in project timeline or budgetary repercussions.

Risk Reduction Methods: The risks of unanticipated product requirement changes can be mitigated by implementing regular meetings and testing to ensure that existing product requirements meet the needs of the end users.

**Poor communication between project members:**

Impact Severity: High

Probability: 30%

Impact on Project: Large-scale software projects depend on work modularisation in its early stages to speed up the development process, before all the individual parts are put together to create the final product. Poor communication and miscommunication between group members may lead to a non-cohesive final product, delaying the project timeline or leading to budgetary repercussions.

Risk Reduction Methods: The risks of poor communication/miscommunication between group members can be mitigated by the scheduling of regular weekly meetings to discuss project updates and clarify bottlenecks/doubts. Strict adherence to project checklists can also mitigate this risk.

**Resource Underestimation/Changes:**

Impact Severity: High

Probability: 10%

Impact on Project: Resource underestimation can come in a variety of forms, such as cost underestimation. Underestimating costs may lead to delayed project schedules, or hasty changes to product requirements to stay within project budget. Resource changes may come in the form of human resource (Group members leaving before completion of a project). This may lead to the loss of specialised skills or knowledge required to complete the project, leading to delays in project schedule in re-acquiring manpower to fill the role of the member that has left.

Risk Reduction Methods: Regular meetings must be scheduled to establish rapport with group members and recognise any potential conflicts which may lead to their leaving. Costs can be monitored with stringent documentation and regular meetings to ensure the project stays within budget.

# 9 Quality Assurance

Quality assurance can be achieved by strict adherence to the Quality plan. Hangout will adopt the following Quality Assurance methodologies for both functional and non-functional testing.

Functional Testing:

The functional testing methodologies to be adopted are as follows:

• **Unit Testing:**

Each individual unit or function is tested.

• **Integration Testing:**

Each individual software module is combined to be tested in bigger groups.

• **System Testing:**

System level testing is conducted on a complete and fully integrated software product in line with quality assurance.

• **Acceptance Testing:**

Acceptance testing is conducted in the shoes of an end user to verify the software system.

Non-Functional Testing:

The non-functional testing methodologies to be adopted are as follows:

• **Performance Testing:**

Load testing will be conducted to ensure that the software program operates well under the stress of multiple users.

• **Usability Testing:**

Usability testing will be conducted to ensure that the program is intuitive to use.

# 10 Monitoring & Control

The monitoring and controlling of a software project are integral to its success. Here are some of the most important procedures taken to successfully monitor the software project:

**Identification and management of substantial project risks:**

The late identification of substantial project risks can throw a wrench in the works of a successful project, as they leave project managers with insufficient time to take preventative measures to rectify these issues. Therefore, major risks and possible preventive measures have been identified under ‘Risk Management’ to avoid this issue.

**Monitoring of project progress and controlling quality:**

Throughout the entirety of the software project, meetings shall be scheduled on a weekly basis to monitor the project’s progress, identify bottlenecks and resolve potential conflicts in the development process. During these meetings, the quality of the project deliverables shall be assessed to ensure that they meet the standards defined in the project management plan. More details on quality control can be found under ‘Quality Assurance’.

**Controlling Changes in project plan:**

Changes to the schedule, costs, or scope of the project must be closely monitored, deliberated and approved by the relevant parties before they are pushed out to avoid unrealistic/unachievable task objectives.

**Project Breakdown and Timeline Management:**

The Hangout Project Plan details integral components that serve to detail a timeline for the project, including the project Schedule, project estimates and various checklists to be met. Strict adherence to these sections will allow for successful management of the project through clear work decomposition.