**Mall-E: A Mall Companion App**

**Project Plan**

**By Team Runtime Error:**

Das Atrik (U1823950C)

Padhi Abhinandan (U1823860D)

Chua Chong Wei Melvin (U1921924G)

Chua Chong Hung Kelvin (U1921404D)

Rajuravi Vishal Raj (U1822268B)

Goh Hong Xiang, Bryan (U1920609E)

**Version 1.0**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Number** | **Date** | **Primary Author(s)** | **Comments** |
| 1.0 | Feb 21th, 2021 | Das Atrik, Padhi Abhinandan, Chua Chong Wei Melvin, Chua Chong Hung Kelvin, Rajuravi Vishal Raj, Goh Hong Xiang, Bryan | First version |
|  |  |  |  |

**Table of Contents**

[**1 Introduction**](#_heading=h.8ut1mc887o2x) **4**

[**1.1 Project Overview**](#_heading=h.pr11uhe1r9z9) **4**

[1.2 Project Description and Scope](#_heading=h.270pp1lgrg2l) 4

[**2 Project Organization**](#_heading=h.5fsrioupms6g) **5**

[2.1 Team Structure](#_heading=h.6nu1b9x246j7) 5

[2.2 Roles and Responsibilities](#_heading=h.9v3gl5obwrwo) 5

[2.3 Team Communication](#_heading=h.9v5qp9rmr0ol) 7

[**3 Process Definition**](#_heading=h.rit5vy1ia1oj) **7**

[3.1 Lifecycle Model](#_heading=h.gx05xchmtgu2) 7

[**4 Schedule**](#_heading=h.irxs0kb6li8w) **8**

[4.1 Activity Dependencies and Schedule](#_heading=h.9c7x8z89s5o9) 8

[4.2 Work Breakdown Structure](#_heading=h.rcq2a3wx8dha) 8

[4.3 Work Packages](#_heading=h.6xibavut70ng) 9

[4.4 Activity Dependencies](#_heading=h.9ghqe9w8vo6q) 9

[4.5 Work Packages Details](#_heading=h.9pt42tnogxts) 10

[**5 Project Estimates**](#_heading=h.f05dnxuxjarn) **14**

[5.3 Cost Estimates](#_heading=h.esrc38bg5o5d) 18

[**6 Product Checklist**](#_heading=h.nvvlea4389b) **20**

[**7 Best Practice Checklist**](#_heading=h.3fsyvxmun0ji) **21**

[**8 Risk Management**](#_heading=h.edjydeooaqhs) **22**

[**9 Quality Assurance**](#_heading=h.b7gkmk7qy0h8) **23**

# 1 Introduction

## 1.1 Project Overview

Mall-E is a mobile application that offers Crowd Density Tracking on every mall in Singapore. Our Crowd Density Tracking offers mall-by-mall, level-by-level and restaurant-by-restaurant crowd density.

Side functionalities include waiting time of restaurants in that specific mall as well as the top 5 Covid-19 news articles that occurred in that specific mall.

## 1.2 Project Description and Scope

The purpose of Mall-E main functionality Crowd Density Tracking is to encourage users to avoid crowded areas during this period of pandemic (Covid-19). With cases of Covid-19 not stopping, it is essential for individuals to exercise self responsibilities to avoid crowded areas.

The basis of Mall-E is to use the Best Time API as well as Google Popular Times API for Crowd Density Tracking and waiting times of restaurants respectively. Mall-E will be applying Machine Learning Algorithm to CCTV footages within the malls for level-by-level Crowd Density Tracking.

Favourites function is made available for users that have a frequented mall. Access will be much easier for future usages.

# 2 Project Organization

## 2.1 Team Structure

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

|  |  |
| --- | --- |
| **Role(s)** | **Name** |
| Project Manager / Lead Developer | Das Atrik |
| Quality Assurance Manager / Engineer | Rajuravi Vishal Raj |
| Release Manager / Engineer | Kelvin Chua Chong Hung |
| Front-End Developer / System Architect | Melvin Chua Chong Wei |
| Front-End Developer / Usability Analyst | Goh Hong Xiang, Bryan |
| Front-End / Back-End Developer | Padhi Abhinandan |

## 2.2 Roles and Responsibilities

**Project Manager: Das Atrik**

• Oversees project progress

• Approves and executes project plan

• Assigns tasks and reports status of the project to team members

• Manages and motivates team members

• Represents the team to the outside world

• Oversees the progress of the software development (Lead Developer)

• Creates requirement specification document (Lead Developer)

**Quality Assurance Manager / Engineer: Rajuravi Vishal Raj**

• Ensures acceptable software qualities

• Crafts the project quality plan

• Ensures acceptable software qualities

• Designs and implements test plan/procedures

• Verify software requirements

**Release Manager / Engineer: Kelvin Chua Chong Hung**

• Crafts the project release plan

• Creates release builds

• Oversees and approve prototype releases

• Integrate changes for delivery

**System Architect: Melvin Chua Chong Wei:**

• Designs logical system based on requirements

• Translates logical design into detailed design

• Creates detailed design document

• In-charge of user interface development (Front-End Developer)

• Ensure user interface follows guidelines [Shneiderman's 8 golden rules] (Front-End Developer)

**Usability Analyst: Goh Hong Xiang, Bryan**

• Designs User Interface Mock-Ups

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

• Creates user manual

• In-charge of user interface development (Front-End Developer)

• Ensure user interface follows guidelines [Shneiderman's 8 golden rules] (Front-End Developer)

**Front-End / Back-End Developer: Padhi Abhinandan**

• Integration of coded modules into a functioning system

• Implements product based on the detailed design document

• Develops frontend-backend communication API

• Oversees backend architecture

• In-charge of user interface development

• Ensure user interface follows guidelines [Shneiderman's 8 golden rules]

## 2.3 Team Communication

The project team communication channels include the following:

• Weekly meetings are held on Sundays.

• Shared-Drive set up for storage of work documents

• Git-Hub used for collaboration to code

• Telegram Group for constructability

• Split up into subgroups as necessary, in order to work more cooperatively on specific problems.

# 3 Process Definition

## 3.1 Lifecycle Model

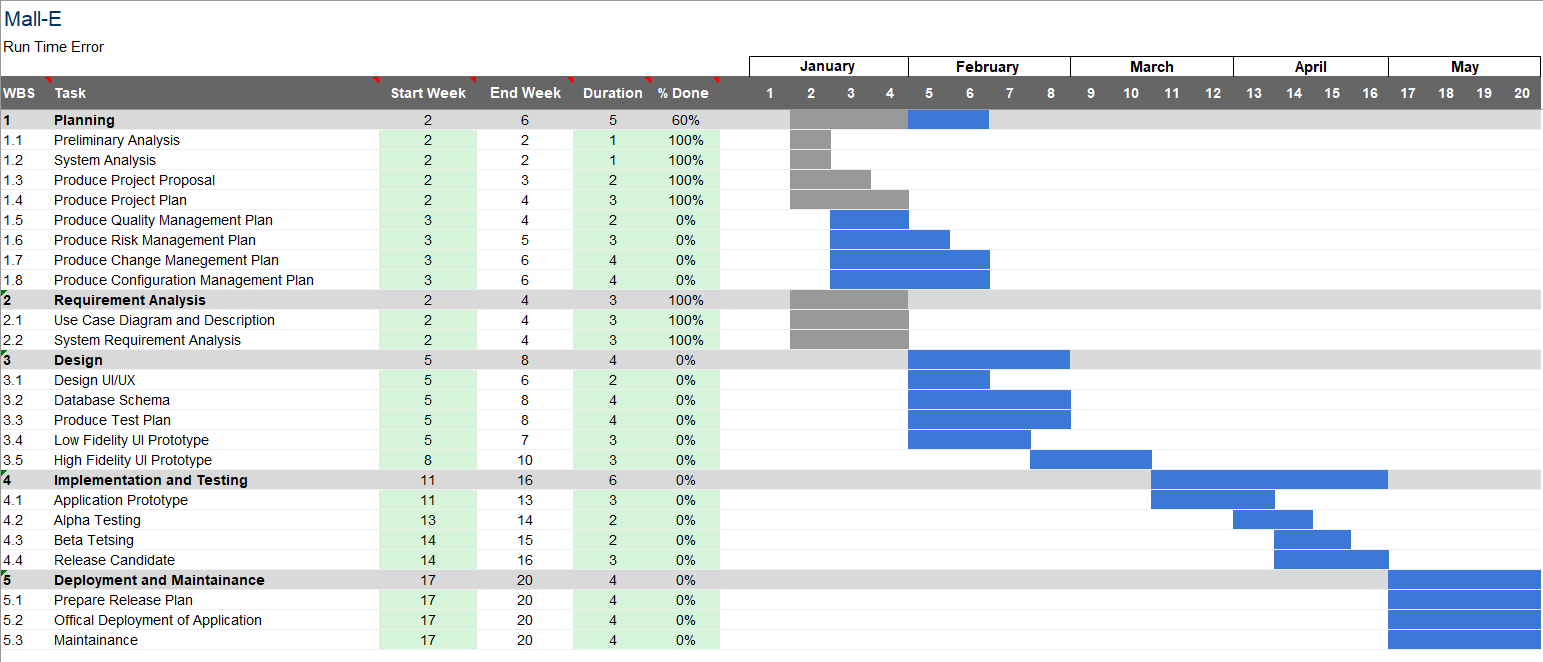
The most suitable and preferred lifecycle model for the Mall-E project is the Incremental Development Model which is a very versatile model. The main advantage of this model is the repeated number of iterations involved in design, development, coding, integration, unit testing and quality assurance. In order to meet the project requirements within the scheduled deadline and reach the assured quality, Incremental Development Model is definitely a more viable choice than the traditional Waterfall SDLC. Although the overall cost is higher in this model, the initial delivery cost is lower. [http://tryqa.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/]

For project Mall-E, we have decided to avoid other SDLC methodologies such as Spiral due to the short timeline. This model is very flexible for development and less costly to change the scope. This model makes it easier for the team to test and debug issues during smaller iterations. It also enables the team to manage risk easily because the risky pieces are handled during each iteration.

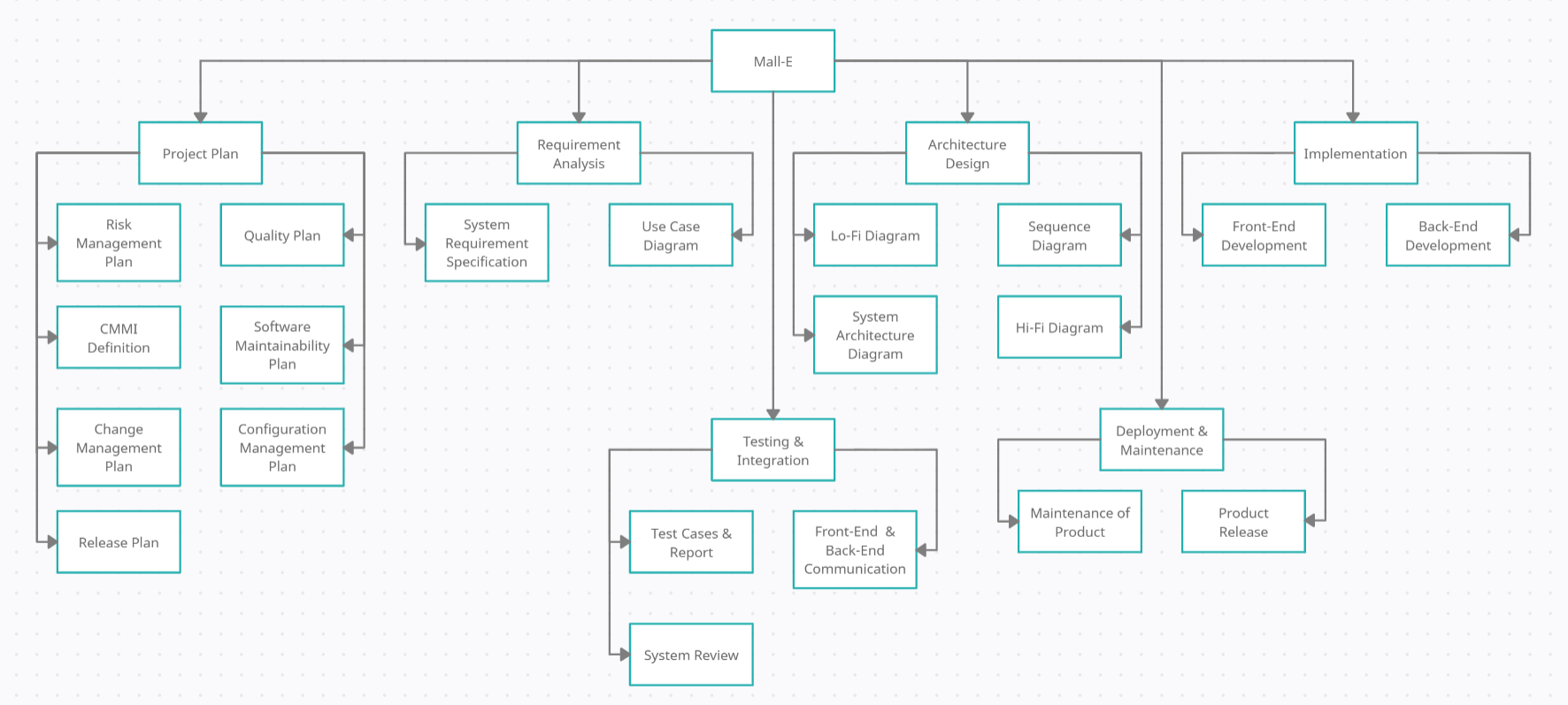
The first iteration of the functionality is scheduled to me delivered as indicated in the scheduled project timeline. After receiving feedback from the stakeholders, further iterations will be planned and developed as per the requirements. Thereafter, stakeholders can respond to each build.

# 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 Analysis Elicitation

3. System Architecture

4. Implementation / Coding

5. Unit Testing

6. Maintenance

## 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 | 28 days | -- |
| X02 | Requirement Analysis Elicitation | 14 days | -- |
| X03 | System Architecture | 35 days | X01,X02 |
| X04 | Implementation / Coding | 35 days | X03 |
| X05 | Unit Testing | 15 days | X04 |
| X06 | Deployment and Maintenance | 15 days | X05 |

## 4.5 Work Packages 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.

|  |  |
| --- | --- |
| **Work Package** | X01 - Project Plan |
| **Assigned To** | **Das Atrik**, Chua Chong Wei Melvin, Chua Chong Hung Kelvin, Rajuravi Vishal Raj, Goh Hong Xiang Bryan, Padhi Abhinandan |
| **Effort** | 28 Days |
| **Start Date** | 19/01/2021 |
| **Purpose** | To supply clarity on the project on execution phases such as the project's goals, aims, scope of work, team formation, milestones, work breakdown, and risks etc. This communicates the estimated progress of the whole project to external parties, as well as forces the team to think through / familiarise themselves with every aspect of the project. |
| **Inputs** | - |
| **Activities** | Proposals are crafted before the actual development of the application   * Project Plan * Risk Management Plan * CMMI Definition * Software Maintainability Plan * Release Plan * Change Management Plan * Configuration Management Plan |
| **Outputs** | X01 |

|  |  |
| --- | --- |
| **Work Package** | X02 - Requirement Analysis Elicitation |
| **Assigned To** | **Rajuravi Vishal Raj**, **Das Atrik**, **Chua Chong Wei Melvin**, **Goh Hong Xiang Bryan**, Chua Chong Hung Kelvin, Padhi Abhinandan |
| **Effort** | 14 Days |
| **Start Date** | 02/02/2021 |
| **Purpose** | This communicates the functionality of the application to external parties, as well as forces the team to think through / familiarise themselves with every function of the application.   * It gives a clear guideline to the developers working on the project. * To build the user interface between the system and the customer, to make it easy use, and friendly to the customer |
| **Inputs** | - |
| **Activities** | System Elicitation are to be crafted before actual development of the project:   * System Requirement Specification * Quality Plan |
| **Outputs** | X02 |

|  |  |
| --- | --- |
| **Work Package** | X03 - System Architecture |
| **Assigned To** | **Chua Chong Wei Melvin**, **Goh Hong Xiang Bryan**, Padhi Abhinandan, Chua Chong Hung Kelvin, Rajuravi Vishal Raj, Das Atrik |
| **Effort** | 35 Days |
| **Start Date** | 02/02/2021 |
| **Purpose** | This communicates the functionality of the application to external parties, as well as forces the team to think through / familiarise themselves with every function of the application.   * It gives a clear guideline to the developers working on the project * High level design entails defining the architecture of the software system and identifying the various components and how they are inter-related to and interactive with each other. Designers also 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 so on. Design topics including maintainability, portability, and reusability will be addressed here as well. |
| **Inputs** | X01, X02 |
| **Activities** | System Architecture are to be crafted before actual development of the project:   * Lo-Fidelity / Hi-Fidelity Diagram * Sequence Diagram * Use Case Diagram * System Architecture Diagram |
| **Outputs** | X03 |

|  |  |
| --- | --- |
| **Work Package** | X04 - Implementation / Coding |
| **Assigned To** | **Das Atrik**, **Padhi Abhinandan**,Chua Chong Wei Melvin, Goh Hong Xiang Bryan, Chua Chong Hung Kelvin, Rajuravi Vishal Raj |
| **Effort** | 35 Days |
| **Start Date** | 16/02/2021 |
| **Purpose** | To implement the system as per the requirements specification and other associated documents. |
| **Inputs** | X03 |
| **Activities** | Front-End and Back-End Developers will start to work towards delivering a final working product |
| **Outputs** | X04 |

|  |  |
| --- | --- |
| **Work Package** | X05 - Unit Testing |
| **Assigned To** | **Rajuravi Vishal Raj**,Das Atrik, Padhi Abhinandan, Chua Chong Wei Melvin, Goh Hong Xiang Bryan, Chua Chong Hung Kelvin |
| **Effort** | 35 Days |
| **Start Date** | 23/03/21 |
| **Purpose** | * To implement the system as per the requirements specification and other associated documents. * To identify and fix logical and syntactical errors produced during the implementation of the System, and setting up drivers and stubs to see how the module responds to various inputs. Black box testing as well as white box testing might be 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. |
| **Inputs** | X04 |
| **Activities** | Quality Assurance Manager will take charge and with accordance to the testing plan, implement the test on the application. |
| **Outputs** | X05 |

|  |  |
| --- | --- |
| **Work Package** | X06 - Deployment and Maintenance |
| **Assigned To** | **Chua Chong Hung Kelvin**,Rajuravi Vishal Raj,Das Atrik, Padhi Abhinandan, Chua Chong Wei Melvin, Goh Hong Xiang Bryan |
| **Effort** | 15 Days |
| **Start Date** | 23/03/21 |
| **Purpose** | * To release the product into the market * Maintenance of the product as there might be new improvements / updates after the official release date also new bugs not found during the testing phase might appear as there are tentatively more users now. |
| **Inputs** | X05 |
| **Activities** | The Release Manager will take charge and with accordance to the release plan, implement the release of the product. Maintenance will also be led by the Release Manager as feedback of the application will be available after a few weeks of the release. The Release Manager to take note of future updates / improvements to the product with accordance to the feedback available. |
| **Outputs** | X06 |

# 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**

Mall-E supports the following proposed functions:

* The system must allow the user to register an account
* The system must allow a registered user to login
* The system must display the various malls on a map with their corresponding colored markers based on the crowd density
* The system must allow the user to click on a marker to view detailed information about the mall
* The system must allow the user to click the “More Information” button which redirects the user to a page which contains mall overview and a list of restaurants within it
* The system must display the restaurant details as a list
* The system must allow the user to bookmark certain malls in the favorites tab
* The system must display the top 5 news articles related to COVID notices in malls

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 information: (name, email, password)
* Favorites tab (i.e. names of malls bookmarked by user)

|  |  |  |  |
| --- | --- | --- | --- |
| **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 map containing markers representing each mall
* Displaying mall details
* Displaying restaurant details
* Displaying news

|  |  |  |  |
| --- | --- | --- | --- |
| **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 which mall details page to display based on user’s selection
* Selecting mall from Favorites tab

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**

* Login
* Register

|  |  |  |  |
| --- | --- | --- | --- |
| **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:

* 1 External Files Referenced (Notifications about urgent COVID notices)

|  |  |  |  |
| --- | --- | --- | --- |
| **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 information |
| Medium | Favorites tab |
| Logical Files | High | Login |
| High | Register |
| Outputs | High | Displaying a map containing markers representing each mall |
| Medium | Displaying mall/restaurant details |
| Medium | Displaying news |
| Inquiries | Medium | Selecting which mall details page to display based on user’s  selection |
| Medium | Selecting mall from Favorites tab |
| Interfaces | High | Notifications |

Calculation of Unadjusted Function Points:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Characteristic** | **Low** | | **Medium** | | **High** | |
| Inputs | 1 | × 3 | 1 | × 4 | 0 | × 6 |
| Outputs | 0 | × 4 | 2 | × 5 | 1 | × 7 |
| Inquiries | 0 | × 3 | 2 | × 4 | 0 | × 6 |
| Logical Files | 0 | × 7 | 0 | × 10 | 2 | × 15 |
| Interfaces | 0 | × 5 | 0 | × 7 | 1 | × 10 |
| **Unadjusted FP** | 3 |  | 22 |  | 47 |  |
| **Total=L+M+H** | 72 | | | | | |

**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 | 2 | 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 | 4 | Daily peak transaction period is anticipated. |
| On-line data entry | 2 | 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 | 4 | Online update of major internal logical files is included. |
| Complex processing | 2 | Any one of the complex components |
| Reusability | 3 | 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 | 3 | No special considerations were stated by the user *but* special  setup is required for installation. |
| Operational Ease | 3 | 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 | 2 | 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 | 40 | |
| **Influence Multiplier**  = Total score × 0.01 + 0.65 = 40 × 0.01 + 0.65 = 1.05 | | |
| **Adjusted FP**  = Unadjusted FP × Influence Multiplier = 72 × 1.05 = 75.6 | | |

|  |
| --- |
| **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 53 lines of code if the application is implemented using JavaScript.

Therefore, we have: **Lines of Code** = 75.6 FP × 53 LOC/FP = **4007 LOC**

**5.2 Efforts, Duration and Team Size Estimation**

To estimate the effort and duration required for Mall-E, we have used Function Points as the basis to calculate Effort, Duration, and Team Size, as well as to determine the appropriate schedule. The estimates have been expanded to account for project management and additional 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.

• **Effort** = Size / Production Rate = (4007 LOC) / (62 LOC/PD)\*= 64.63 PD

• **Duration** = 3 × (Effort) 1/3 = 3 × (64.63) 1/3 = 12.04 Days

• **Initial schedule** = 12.04 Days / 5 days a week = 2.41 Weeks

• **Team size** = 64.63 PD / 12.04 D = 5.37 P = 6 Persons (round up)

• There are 8 hours in a working day, and 5 working days in a week.

• **Total person-hours (PH)** = 64.63 PD × 8 hours = 517.04 PH

\* Note: Lines of code per Person Day statistics based on Industrial Benchmarks, 1997: 62 LOC/PD for Canada.

**5.2.1 Distribution of Effort**

|  |  |  |  |
| --- | --- | --- | --- |
| **1990’s Industry Data** | **Work Package** | **Distribution** | **Estimates** |
| Preliminary Design:  18% | Project Plan | 9% | 46.53 |
| Requirement Specification | 9% | 46.53 |
| Detailed Design: 25% | User Interface | 7% | 36.19 |
| Technical Architecture | 11% | 56.87 |
| Data Modeling | 7% | 36.19 |
| Code & Unit Testing:  26% | Code & Unit testing | 21% | 108.58 |
| Online Documentation | 5% | 25.85 |
| Integration & Test: 31% | Integration & Quality Assurance | 31% | 160.28 |
|  | **Extrapolated total effort** |  | 517.04 |
|  | 2% for project management |  | 10.34 |
|  | 3% for contingency |  | 15.51 |
|  | **Total effort** |  | 542.89 |



Note: The above duration estimates are based on the assumption that each team member works an equal amount on any given task.

## 5.3 Cost Estimates

**Hardware:**

**Developer Workstations**

|  |  |
| --- | --- |
| 6 - XPS 8640 Desktop | Total  $1,400.00 x 6  = $8,400.00 |
| Intel Core i9-10900K processor |
| 12 GB RAM |
| 500 GB SSD drive |

**Software:**

|  |  |  |
| --- | --- | --- |
| Hosting Server (AWS) | $30,000.00 | Total  $54,800.00 |
| Database (MongoDB) | $20,000.00 |
| BestTime API Subscription | $4,800.00 |

**Staff:**

|  |  |  |
| --- | --- | --- |
| 1 - Project Manager | $90,000.00 | Total  $270,000.00 |
| 5 - Project team members | $36,000.00 x 5 = $180,000 |

**Other Costs:**

|  |  |  |
| --- | --- | --- |
| NTU Office space | $4,000.00 | Total  $12,500.00 |
| Public Transport | $500.00 |
| Market Research | $8,000.00 |

**Total: $345,700.00**

The customer will supply the required hardware and software necessary to run the application on their smartphones. RunTime Error is not responsible in any way for supplying said systems. Mall-E’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. RunTime Error will also demonstrate the completed product.

# 6 Product Checklist

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

|  |  |  |
| --- | --- | --- |
| **S/N** | **Project Deliverable** | **Estimated Deadline** |
| 1 | Project Proposal | 02 Feb, 2021 |
| 2 | Use Case Descriptions |  |
| 3 | System Requirements Specification | 16 Feb, 2021 |
| 4 | Quality Plan |  |
| 5 | Project Plan | 23 March, 2021 |
| 6 | Risk Management Plan |  |
| 7 | Prototype Demo |  |
| 8 | Design Report on Software Maintainability | 06 April, 2021 |
| 9 | Configuration Management Plan |  |
| 10 | Change Management Plan |  |
| 11 | Release Plan |  |
| 12 | Demo Presentation Slides | 20 April, 2021 |
| 13 | Module/System Test Plan |  |
| 14 | Test Cases and Requirements |  |
| 15 | Test Coverage Report |  |
| 16 | CMMI Level 2 Definition |  |
| 17 | **System Release (Demo)** |  |

# 7 Best Practice Checklist

|  |  |
| --- | --- |
| **S/N** | **Practice** |
| 1 | Document everything that is done; all documentation must be in a standardized format, following these principles:   * Line spacing/paragraphing style must look clean and readable * Font and font color must be consistent throughout the documents * Technically sound language must be used |
| 2 | Pay attention to requirements specifications, especially the following standards:   * Unambiguous language must be used * Accurate, consistent, and achievable requirements must be listed * Requirements must be discussed and agreed by all stakeholders * Both functional and non-functional requirements must be stated * All requirements must be atomic and non-repeating |
| 3 | Complexity management is one of the major challenges. Best to keep it simple. Strive to:   * Minimize interfaces between procedures, modules, and data. * Avoid complicated product functions, design and functionality must meet the customer requirements * Minimize interfaces between people, which might otherwise lead to exponential communication costs |
| 4 | Require Visibility. There must be a tangible measure of the progress during the development phase and must take management action accordingly. The manager must have good rapport with their employees and must require developers to make code available for review |
| 5 | Plan for continuous change. We must:   * Use a configuration management system and make processes * New revisions should be approved by the Release Manager before being made and checked for quality and compliance after being made * Required maintenance every 3 months * All manuals designs, test, source code should have revision numbers and dates revision history comments, change marks to indicate the changes |
| 6 | Don’t underestimate. We must be careful to obtain accurate estimates for: time, effort, overhead, meeting time, and especially effort on integration, testing, documentation and maintenance. |
| 7 | When conducting testing and/or code reviews, keep in mind the following practices:   * Code reviews are a much more efficient method to find software defects. Plan and manage code reviews between team members * Both black box and white box testing will be conducted * Testing will involve unit, functional, integrating and acceptance testing * Stress and load test cases must be added * Test cases must check for boundary conditions * There must be a test case to test each requirement * Test cases must check for incorrect inputs |

# 8 Risk Management

Besides the general risk management, the following risks have been identified for the Mall-E project:

**More changes to requirements than anticipated**

Impact Severity: High

Probability: 40%

Impacts: The requirement specifications might change depending on which stage of development we are at. The stakeholders may push for a complete redesign if they change their mind.

Risk Reduction Strategy: Make stakeholders aware of potential repercussions of requirement changes such as delays and budget increase. Have ample discussions while eliciting requirements.

**System size underestimated** Impact Severity: Moderate Probability: 20%

Impacts: More work will be needed on design and coding which could negatively impact schedule.

Risk Reduction Strategy: Update estimates such as function points and other key measures of system size in section 5.1 often as the project progresses.

**Problems coordinating within group**

Impact Severity: Moderate

Probability: 35%

Impacts: Miscommunication between developers and managers. Managers may not be able to measure progress. Cultural differences leading to disputes. Portions of projects not completed.

Risk Reduction Strategy: Follow communication plans as documented in section 2.3. Have team-bonding activities to foster team spirit

**Staff leaving before project complete**

Impact Severity: Extreme

Probability: 10%

Impacts: More work would be left per member of the remaining team, and any specialized skills or knowledge would be lost.

Risk Reduction Strategy: Hire members according to a contract which would lead to severe penalties if broken before the expiry date

**No cooperation from third party stakeholders like government, API vendors, restaurant managers**

Impact Severity: Extreme

Probability: 25%

Impacts: The data represented on our app would be inaccurate and untimely.

Risk Reduction Strategy: Get written proof of their agreement to cooperate with us by providing us resources before starting development. Find alternative sources of data otherwise

# 9 Quality Assurance

Quality Assurance has been covered in detail in a separate document titled “Quality Plan”, which is available in our Team Wiki. Detailed guidelines have been provided in the Quality Plan document to ensure quality assurance has been achieved.

Several points have been discussed in the Quality Plan, including management, documentation, implementation, and testing. In addition, Runtime Error shall maintain the project (Mall-E) by following the conventions and ensuring high standards. Thus, we guarantee good practice in specifications and implementation throughout the duration of the project.

While the Quality Plan details quality assurance, and details about specific test procedures shall be provided in the System Test Plan, note that Runtime Error will also make use of the following two testing methodologies:

* **Unit Testing:** thisinvolves testing system components individually. For Mall-E, this involves testing of each of the following: Login/Registration component, Mall Crowd Density component, Restaurant Waiting Time component, and News component.
* **In-Place Testing:** thisinvolves testing of the whole system as a unit. Mall-E will be tested via software emulators and real Android/iOS devices.

**10 Monitoring & Control**

Many measures are required in order to properly monitor the progress of Mall-E. Some of the measures used for monitoring and control are as follows:

**Quantitative measurement of resource consumption:** it is crucial to monitor resource consumption for a software project like Mall-E, as improper resource consumption may potentially lead to problems. Thus, monitoring the daily consumption of each individual resource is important to obtain a quantitative measurement of the project’s resource consumption over the duration of the project. Using such quantitative measurements would allow the team members to act in a timely manner in order to keep the project on track, while respecting the project schedule and budget.

**Identification of major project risks:** It is a good practice to identify the major risks that may occur throughout the duration of the project as early as possible. Early risk identification helps the project team avoid those risks and to correctly handle any problems. Examples of major risks and preventive measures associated with Mall-E are discussed in the Risk Management section of this document.

**Regular reviews of project progress:** Throughout the duration of the Mall-E project, the members of Team Runtime Error shall participate in weekly meetings to review the progress of the project and all associated tasks, including management, planning, analysis, development, and testing. Project progress is also continuously monitored via our main communication channel of Telegram (smartphone messaging application), which is also used to discuss any problems encountered during our project. For version control, i.e., to keep track of the project’s code and all changes and developments in the code, a private GitHub repository is being used. Google Drive is also being used for cloud storage, which stores our project documents, and allows for editing of the documents. Lastly, the Team Wiki is being used to formally manage all of the documentation for the project. Final versions of the code developed shall also be uploaded to the Apache Subversion (SVN) repository (in addition to GitHub).

**Timeline Planning and task decomposition:** Timeline planning for this project has been done according to the pre-set deadlines for each deliverable and project milestone. For each deliverable and project milestone, the required tasks have been broken down into sub-tasks to facilitate timeline planning and task decomposition. The detailed timeline/schedule is available in the Gantt Chart provided in this document’s Schedule section, specifically section 4.1. Although the actual product development for Mall-E may not perfectly meet the ideal deadlines and milestones, the team shall try to minimize the difference between the pre-set deadlines and completion date for each task. The detailed task decomposition is available in the Work Breakdown Structure (WBS) in section 4.2.