# Project Plan

**for**

# Cafe Bunny

**PRJPL_PRO_003_V1.4**

**Version 1.4 approved**

**Prepared by Team Code Nation**

**Nanyang Technological University,**

**School of Computer Science & Engineering**

**09/04/2021**

**Submitted to:**

Dr. Shen Zhiqi, Lab Supervisor

# VERSION HISTORY

| Version # | Implemented By | Revision Date | Approved By | Approval Date | Reason |
|---|---|---|---|---|---|
| 1.0 | Ong Yi Shen | 04/03/2021 | Chew Zhi Kang | 05/03/2021 | Project Plan (1st Draft) Sections 4.3-7 |
| 1.1 | Chew Zhi Kang | 10/03/2021 | Ong Yi Shen | 10/03/2021 | Project Plan (2nd Draft) Sections 8-10 |
| 1.2 | Bachhas Nikita | 11/03/2021 | Chew Zhi Kang | 12/03/2021 | Project Plan (3rd Draft) Sections 1 to 4.2 |
| 1.3 | Ong Yi Shen | 26/03/2021 | Chew Zhi Kang | 26/03/2021 | Standardization |
| 1.4 | Ong Yi Shen | 9/04/2021 | Chew Zhi Kang | 09/04/2021 | Updated Estimations |

# TABLE OF CONTENTS

# 1. Introduction

## 1.1 Project Overview

The Cafe Bunny project is aimed specifically towards cafe hoppers - people who love to visit and experience different cafes all over Singapore. The project aims to provide cafe hoppers with a digital platform to record and journal their cafe experience, whilst sharing it with other users. The project is planned to be a mobile application making it convenient for users to keep track of their cafe hopping journey by having the application on their phones at all times.

## 1.2 Project Description and Scope

Since the main audience of this project is cafe hoppers, the application must first allow them to track their cafe hopping journey, second allow them to record their experience and share it with other hoppers, along with their achievements and thirds, allow users to view the cafe's statistics.

Features to achieve Goal 1 - Track User's Cafe Journey:

- With the help of Google Maps API, the application will display all cafes within a certain radius whose markers will be colour coded. The colour in this case will represent whether the user has already visited the cafe or not (e.g green - not visited and blue - visited.
- The user will be presented with an option to "Check In" to record their progress whenever the application detects if the user is visiting a particular cafe.
- The application is also gamified and the users will earn points for every cafe they visit. After each level of achievement is unlocked, users can claim coupons which they can later exchange for discounts or rewards points at these cafes.
- Another aspect is that users can choose and create a bunny avatar for themselves. Different avatars will represent different stages in their cafe hopping journey (e.g. Baby Bunny Avatar represents a newly joined cafe hopper)

Features to achieve Goal 2 - Record and share their experience and achievements:

- Cafe hoppers can post pictures and reviews of the cafes they visit on a forum page. Not only does this help users keep track of their progress, but also allows them to share useful insights with other users.
- Users can also show-off their achievements to other users as well. Higher-ranking cafe hoppers will also have their reviews and photos posted at the top of the forum, motivating users to visit more cafes and gain more achievements.

Features to achieve Goal 3 - View Cafe Statistics:

- Whenever users select a particular cafe, the application will display statistics of the cafe, such as the average rating given by other users and the menu of the cafe.
- They also view the reviews of the cafe by users and see the popularity of the cafe.
- This will also help cafe owners promote their cafes especially if their cafes are located in less popular areas or are not marketed due to marketing budget constraints.

The project consists of creating an application that allows cafe hoppers to record and share their cafe journey. It takes in a number of possible constraints into account. These can be limits such as:

- The user is limited to only cafe hop in Singapore at the present moment
- Inappropriate content can be posted on the forum via the pictures or reviews posted by users. An automated system to automatically flag potentially offensive pictures or reviews will be implemented to ensure that content on the application remains appropriate for users of all ages.

Given these constraints, the application will have to display the locations of all cafes within a certain radius and simultaneously allow users to post appropriated reviews and pictures on the forum page, allowing them to track the progress of their cafe hopping journey.

The application requires Google Maps API to display cafe locations on the Maps Page. All the data of the users will be stored in a realtime database system - Firebase database. System integration will be done through GitHub.

The team will first provide the stakeholders with a prototype consisting of the basic features along with dummy data within the Firebase database, a test report - conveying the different test cases, the inputs, expected outputs and actual outputs -, source code, a change management plan and a design report to shows how the application and the design can be further improved in the future.

# 2. Project Organization

## 2.1 Team Structure

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

- Project Manager: Ong Yi Shen
- Lead Developer: Mohamad Asyraaf Bin Abdul Rahman
- Front-End Developer: Jordan Tan Rei Yao
- Back-End Developer: Bachhas Nikita
- QA Manager and Engineer: Ng Qin Wei
- Release Engineer/Manager: Chew Zhi Kang

## 2.2 Roles and Responsibilities

Project Manager: Ong Yi Shen

- Monitors project process
- Control project direction
- Approves and executes project plan
- Assigns tasks and reports status of project to team members
- Manages and motivates the team
- Represents the team to the outside world

Lead Developer: Mohamad Asyraaf Bin Abdul Rahman

- Mentors other developers in the team
- Directing development team in design, development, coding, testing and debugging of the applications
- Develops the basic foundation of the project
- Aids in planning the product, process and schedule for the project
- Provides feedback marketing and sales team
- Decides the technology needed for the project

Front-End Developer: Jordan Tan Rei Yao

- Responsible for implementing visual elements that users see and interact with in an application
- Translate UI/UX design wireframes to actual code that will produce the visual elements of the application
- Working alongside with the UI/UX design and the graphical/design team
- Optimise application for maximum speed, stability and scalability

Back-End Developer: Bachhas Nikita

- Responsible for server-side application logic
- Design and implement data storage solutions
- Responsible for integrating user-facing elements developed by front-end developers with server side logic
- Responsible for implementation of security and data protection

QA Manager and Engineer: Ng Qin Wei

- Ensures acceptable software quality
- Designs testing strategies
- Creates and manages test plans
- Verify software requirements
- Executes test procedures

Release Engineer/Manager: Chew Zhi Kang

- Ensures the application is configured and coded properly for successfully integration
- Responsible for assessing quality and risks of the products
- Builds test environment and troubleshoot any issues pertaining to the product's performance
- Document all processes for future references
- Assembles data on release metrics, oversee source code management system and document steps during the release process
- Organise release schedule for the product

## 2.3 Team Communication

The team's communication channels include the following:

- Weekly or biweekly Zoom meeting sessions (depending on the necessity of the meetup)
- Biweekly meetings during CZ3002 lab session on odd weeks
- Group announcements and updates are carried out through the team's Whatsapp platform
- Minor discussions and doubts are cleared via Whatsapp either through the group channel or through private messages between team members
- Telephone discussions are held as necessary
- The team is split into two subgroups: team members Ng Qin Wei, Mohamad Asyraaf Bin Abdul Rahman and Jordan Tan Rei Yao are in charge of coding and the development stage of the project. Whereas team members Ong Yi Shen, Bachhas Nikita and Chew Zhi Kang are in charge of thoroughly documenting all necessary information.

# 3. Process Definition

## 3.1 Lifecycle Model

We aim to use the Waterfall Software Development Lifecycle (SDLC) Model throughout the Cafe Bunny Project. We have chosen a formal document driven model over an incremental and iterative project as supporting documents written alongside the project will aid in the planning, designing and developmental stages of the product.

Keeping in mind that this is a long-term project, that we would like to expand and update further in the future, having documents would really represent our initial idea and design well. If we had chosen an incremental and iterative method, the focus would have been more on the design, code and testing stages of the process, which could lead to problems such as feature scope, delayed schedule, budget constraints and several more due to the lack of proper planning. Not only that, we also do not have frequent collaboration with the customers to repeatedly ask for their feedback on the product.

Since we had the requirements well understood, milestones were easy to formulate and the project planning was also easy to manage, with each phase having a specific set of requirements. As the phases are completed one at a time, tasks are also easy to split between the team according to their experience and skills. At the end, all processes and results should be well documented.

By the end of week 11 of the NTU's AY 2020/2021 Semester 2, the cafe bunny prototype should be ready to be presented to the stakeholders. The project will continue on for the rest of the year according to the duration estimated, implementing new features, and refining the current one as well.

# 4. Project Estimates

## 4.1 Code Size using Function Points

### 4.1.1 Unadjusted Function Points

Code Bunny supports the following proposed features:

- User login
- Displaying a map with waypoints of nearby cafes
- Tracking of user's visited cafes
- Leveling and achievement system
- Posting reviews and pictures of cafes
- Display cafe information
- Display cafe reviews
- Generating discount codes

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 evaluation of the complexity of each component is as follows:

**Rating Inputs:**

- Posting reviews and pictures of cafes - Medium
- Tracking of user's visited cafes - High

**Rating Outputs:**

- Displaying a map with waypoints of nearby cafes - High
- Display login page - Low
- Generating discount codes - High

**Rating Inquiries:**

- User login - Low
- Retrieve user information that is displayed in the achievements page - Medium
- Retrieve cafe information that is displayed in the cafe page such as pictures and reviews - High

**Rating Logical Files:**

- Cafe information - High
- User information - Medium
- Tracking of user's visited cafes - Medium

**Rating Interfaces:**

- Google Maps API - High

Calculation of Unadjusted Function Points:

| **Characteristic** | **Low** | | **Medium** | | **High** | |
|---|---|---|---|---|---|---|
| Inputs | 0 | × 3 | 1 | × 4 | 1 | × 6 |
| Outputs | 1 | × 4 | 0 | × 5 | 2 | × 7 |
| Inquiries | 1 | × 3 | 1 | × 4 | 1 | × 6 |
| Logical Files | 0 | × 7 | 2 | × 10 | 1 | × 15 |
| Interfaces | 0 | × 5 | 0 | × 7 | 1 | × 10 |
| **Unadjusted FP** | 7 | | 28 | | 51 | |
| **Total=L+M+H** | 86 | | | | | |

## 4.1.2 Adjusted Function Points

| **Influence Factors** | **Score** | **Detail** |
|---|---|---|
| Data Communications | 3 | Communications between client phones and server for saving and retrieving user information and cafe information |
| Distributed Functions | 0 | Entirely client side processing |
| Performance | 3 | The app must be able to run on modern smartphones that have been released from 2015 onwards |

| | | |
|---|---|---|
| Heavily used | 4 | Heavy usage will be expected when the app is first launched |
| Transaction rate | 3 | Transaction rate expected to be high during peak hours |
| On-line data entry | 3 | Uploading of pictures and reviews are required |
| End-user efficiency | 4 | User interface must be appealing to the users |
| On-line data update | 3 | App must be able to updated in real time with other user's reviews and pictures |
| Complex processing | 1 | All of the data is simply fetched and displayed from the servers. Only minor processing is needed for calculating the level of the player and their achievements. |
| Reusability | 0 | Requirements do not state that the app will be repurposed for any other uses |
| Installation Ease | 3 | The app must be easy to install. Can be facilitated through the use of the app store |
| Operational Ease | 4 | The app must be intuitive and easy to use |
| Multiple sites | 0 | Does not apply |
| Facilitate change | 2 | Must be able to support patches to correct any bugs or issues that may arise when the app is launched |
| Total score | 33 | |
| **Influence Multiplier**<br>= Total score × 0.01 + 0.65 = 33 × 0.01 + 0.65 = 0.98 | | |
| **Adjusted FP**<br>= Unadjusted FP × Influence Multiplier = 84.28 | | |

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

### 4.1.3 Lines of Code

According to Casper Jones statistics, each Function Point requires 26 lines of code if the application is implemented using Dart.
Therefore, we have: Lines of Code = 84.28 FP × 26 LOC/FP = 2192 LOC

## 4.2 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.

- 5 working days per week
- Effort = Size / Production Rate = (2192 LOC) / (39 LOC/PD) = 57 PD
- Duration = 3 × (Effort) ^ 1/3 = 3 × (57) ^ 1/3 = 11.55 Days
- Initial schedule = 11.55 Days / 5 days a week = 2.31 Weeks
- Team size = 57 PD / 11.55 D = 4.94 P = 5 People
- Working hours include 9 hours in a working day.
- Total person-hours (PH) = 57 PD × 9 hours = 513 PH

### 4.2.1 Distribution of Effort

| Work Package | Distribution | Estimates |
|---|---|---|
| Project Plan | 20% | 102.6 |
| Requirement Specification | 16% | 82.08 |
| System Design | 18% | 92.34 |
| System Implementation | 15% | 76.95 |
| Testing | 15% | 76.95 |
| Deployment and maintenance | 16% | 82.08 |
| **Extrapolated total effort** | | 513 |
| 5% for project management | | 25.65 |
| 8% for contingency | | 41.04 |
| **Total effort** | | 579.69 |

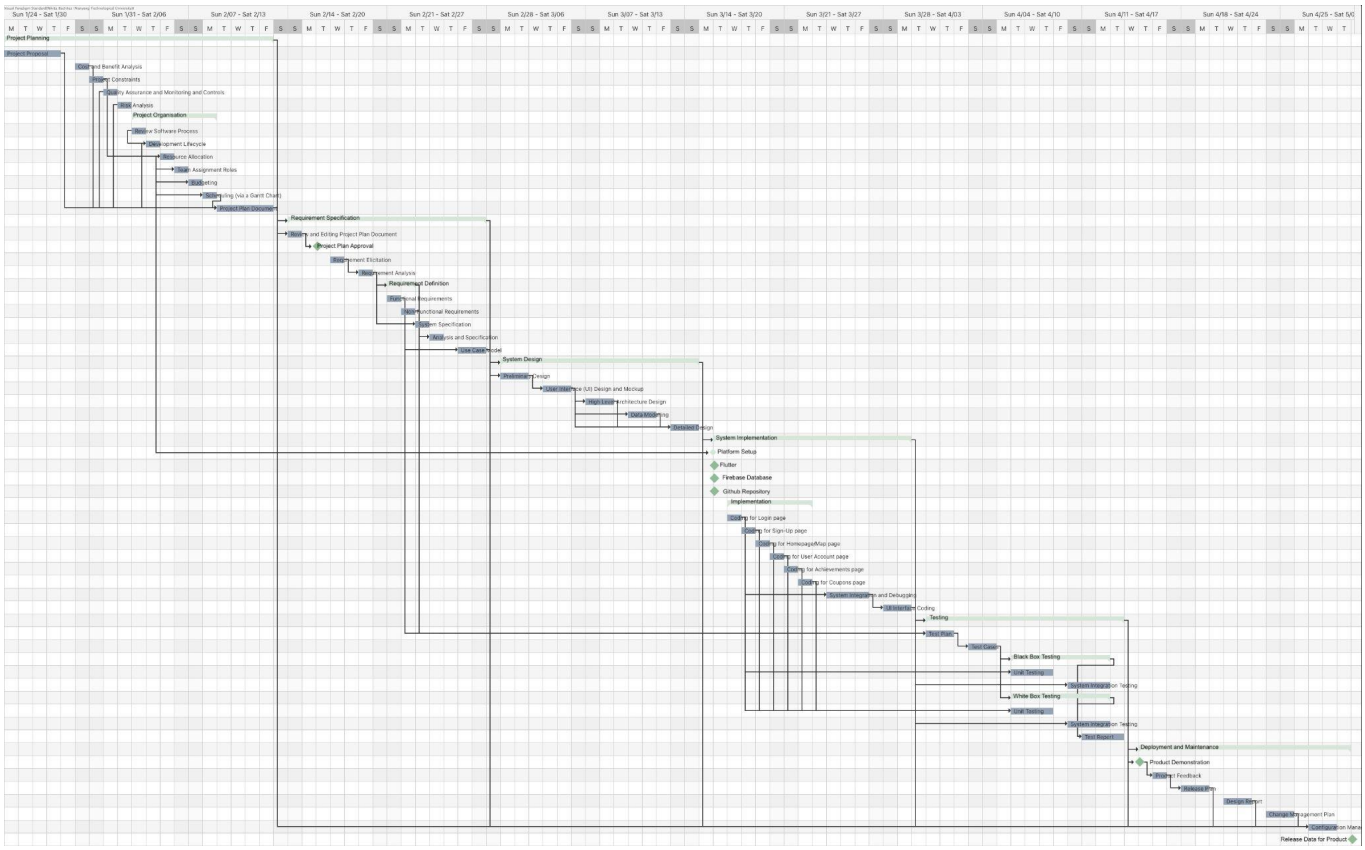
## 4.3 Cost Estimates

| Type | Name | Quantity | Cost | Duration | Total Cost |
|---|---|---|---|---|---|
| Staff | Project Manager | 1 | $10000/Month | 4 Months | $40000 |
| | Lead Developer | 1 | $8000/Month | 4 Months | $32000 |
| | Front End Developer | 1 | $7000/Month | 4 Months | $28000 |
| | Back End Developer | 1 | $7000/Month | 4 Months | $28000 |
| | QA Manager & Engineer | 1 | $7000/Month | 4 Months | $28000 |
| | Release Manager & Engineer | 1 | $7000/Month | 4 Months | $28000 |
| | Software Developer | 5 | $4000/Month | 4 Months | $80000 |
| | Feasibility Analyst | 2 | $4000/Month | 2 Months | $16000 |
| Hardware | Desktop PCs | 11 | $2000 | - | $22000 |
| | Backup Desktop PCs | 5 | $2000 | - | $10000 |
| | Servers | 4 | $3000/Month | 12 Months | $144000 |
| | Backup Servers | 2 | $1500/Month | 12 Months | $36000 |
| | Printer | 2 | $400 | - | $800 |
| | Stationeries | 1 | $100 | - | $100 |
| Software | Microsoft Office | 15 | $80 | - | $1200 |

**Total Cost: $494100**
**Contingency (10% of total cost): $49410**
**Total Cost with Contingency: $543510**

# 5. Schedule

## 5.1 Activity Dependencies and Schedule

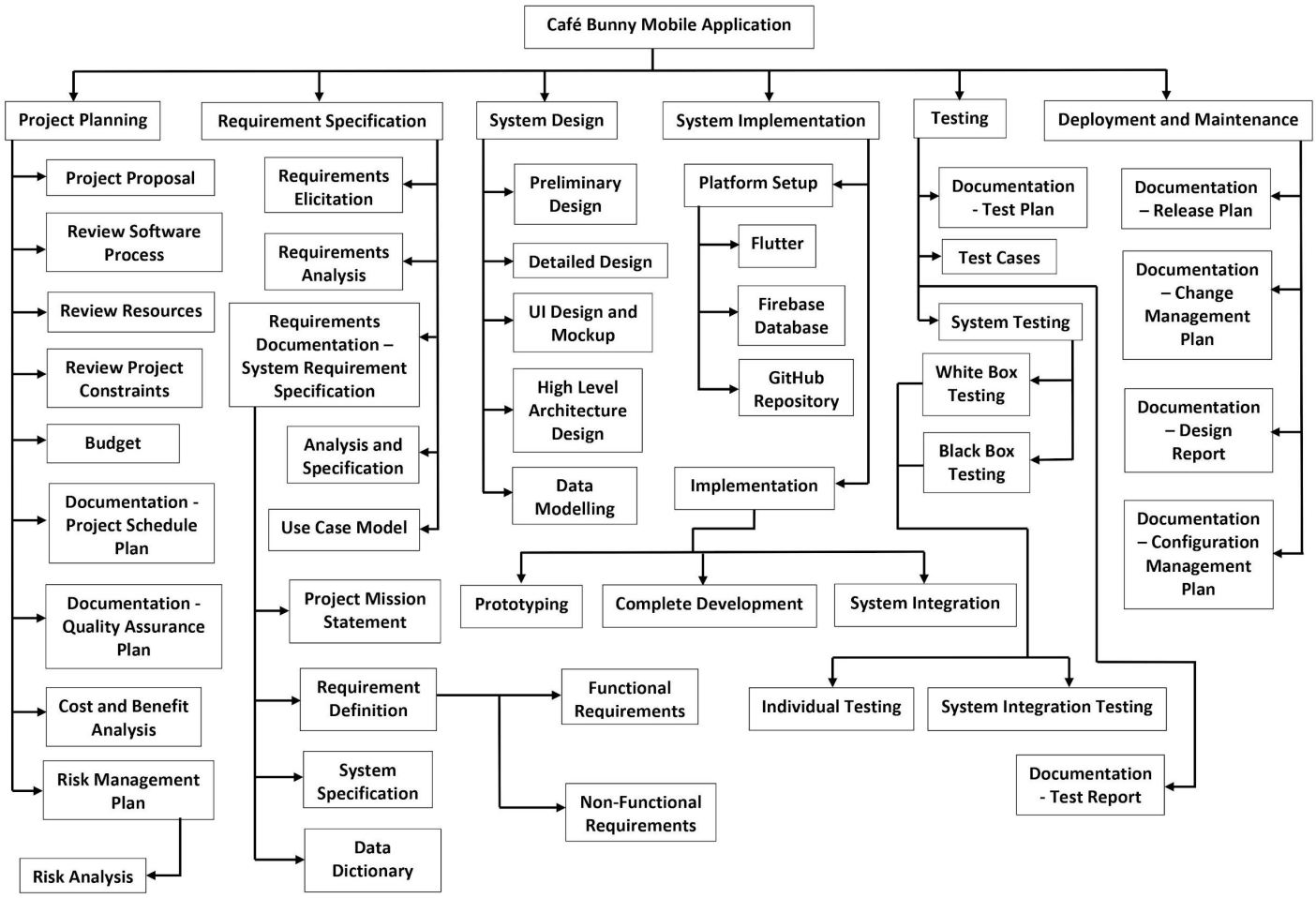


| | | |
|---|---|---|
| ◢ Project Planning | 01/25/2021 | 02/13/2021 |
| Project Proposal | 01/25/2021 | 01/29/2021 |
| Cost and Benefit A... | 01/30/2021 | 01/31/2021 |
| Project Constraints | 01/31/2021 | 02/01/2021 |
| Quality Assurance ... | 02/01/2021 | 02/02/2021 |
| Risk Analysis | 02/02/2021 | 02/03/2021 |
| ◢ Project Organisation | 02/03/2021 | 02/09/2021 |
| Review Software... | 02/03/2021 | 02/04/2021 |
| Development Lif... | 02/04/2021 | 02/05/2021 |
| Resource Alloca... | 02/05/2021 | 02/06/2021 |
| Team Assignme... | 02/06/2021 | 02/07/2021 |
| Budgeting | 02/07/2021 | 02/08/2021 |
| Scheduling (via ... | 02/08/2021 | 02/09/2021 |
| Project Plan Docu... | 02/09/2021 | 02/13/2021 |
| ◢ Requirement Specific... | 02/14/2021 | 02/28/2021 |
| Review and Editing... | 02/14/2021 | 02/15/2021 |
| Project Plan Appro... | 02/16/2021 | 02/16/2021 |
| Requirement Elicit... | 02/17/2021 | 02/18/2021 |
| Requirement Analy... | 02/19/2021 | 02/20/2021 |
| ◢ Requirement Defini... | 02/21/2021 | 02/23/2021 |
| Functional Requi... | 02/21/2021 | 02/22/2021 |
| Non-Functional ... | 02/22/2021 | 02/23/2021 |
| System Specificati... | 02/23/2021 | 02/24/2021 |
| Analysis and Speci... | 02/24/2021 | 02/25/2021 |
| Use Case Model | 02/26/2021 | 02/28/2021 |

| | | |
|---|---|---|
| ◢ System Design | 03/01/2021 | 03/15/2021 |
| Preliminary Design | 03/01/2021 | 03/03/2021 |
| User Interface (UI) ... | 03/04/2021 | 03/06/2021 |
| High Level Archite... | 03/07/2021 | 03/09/2021 |
| Data Modelling | 03/10/2021 | 03/12/2021 |
| Detailed Design | 03/13/2021 | 03/15/2021 |
| ◢ System Implementati... | 03/16/2021 | 03/30/2021 |
| ◢ Platform Setup | 03/16/2021 | 03/16/2021 |
| Flutter | 03/16/2021 | 03/16/2021 |
| Firebase Databa... | 03/16/2021 | 03/16/2021 |
| Github Repository | 03/16/2021 | 03/16/2021 |
| ◢ Implementation | 03/17/2021 | 03/23/2021 |
| Coding for Login... | 03/17/2021 | 03/18/2021 |
| Coding for Sign-... | 03/18/2021 | 03/19/2021 |
| Coding for Hom... | 03/19/2021 | 03/20/2021 |
| Coding for User ... | 03/20/2021 | 03/21/2021 |
| Coding for Achie... | 03/21/2021 | 03/22/2021 |
| Coding for Coup... | 03/22/2021 | 03/23/2021 |
| System Integration... | 03/24/2021 | 03/27/2021 |
| UI Interface Coding | 03/28/2021 | 03/30/2021 |
| ◢ Testing | 03/31/2021 | 04/14/2021 |
| Test Plan | 03/31/2021 | 04/02/2021 |
| Test Cases | 04/03/2021 | 04/05/2021 |
| ◢ Black Box Testing | 04/06/2021 | 04/13/2021 |
| Unit Testing | 04/06/2021 | 04/09/2021 |
| System Integrati... | 04/10/2021 | 04/13/2021 |
| ◢ White Box Testing | 04/06/2021 | 04/13/2021 |
| Unit Testing | 04/06/2021 | 04/09/2021 |
| System Integrati... | 04/10/2021 | 04/13/2021 |
| Test Report | 04/11/2021 | 04/14/2021 |

| | | |
|---|---|---|
| ◢ Deployment and Mai... | 04/15/2021 | 04/30/2021 |
| Product Demonstr... | 04/15/2021 | 04/15/2021 |
| Product Feedback | 04/16/2021 | 04/17/2021 |
| Release Plan | 04/18/2021 | 04/20/2021 |
| Design Report | 04/21/2021 | 04/23/2021 |
| Change Managem... | 04/24/2021 | 04/26/2021 |
| Configuration Man... | 04/27/2021 | 04/29/2021 |
| Released Date for ... | 04/30/2021 | 04/30/2021 |

# 5.2 Work Breakdown Structure



## 5.3 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 Planning
2. Requirements Specification
3. System Design
4. System Implementation
5. Testing
6. Deployment and Maintenance

## 5.4 Activity Dependencies

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

| Work Package # | Work Package Description | Duration | Dependencies |
|---|---|---|---|
| X01 | Project Planning | 3 weeks | - |
| X02 | Requirement Specification | 2 weeks and 1 day | X01 |
| X03 | System Design | 2 weeks and 1 day | X02 |
| X04 | System Implementation | 2 weeks and 1 day | X03* |
| X05 | Testing | 2 weeks and 1 day | X04 |
| X06 | Deployment and Maintenance | 2 weeks and 1 day | X05 |

## 5.5 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** | Code Bunny |
| **Work Package** | X01— Project Planning (1 of 6) |
| **Assigned To** | Ong Yi Shen |
| **Effort** | 102.6 PH |
| **Start Date** | 25th January 2021, Monday |
| **Purpose** | To determine an introductory overview of the project, to be refined in later work packages. |
| **Inputs** | None |
| **Activities** | This work package includes providing a brief overview of the project, its objectives, and a set of proposed project deliverables throughout the development of the software cycle. The people responsible for this work package will also be transcribing ideas brought up in the group meeting discussion into a formal report. |
| **Outputs** | A written document of the Project Plan Introduction. |

| | |
|---|---|
| **Project** | Code Bunny |
| **Work Package** | X02— Requirements Specification (2 of 6) |
| **Assigned To** | Ong Yi Shen |
| **Effort** | 82.08 PH |
| **Start Date** | 14th February 2021, Sunday |
| **Purpose** | To establish a common understanding between the customer and the software project team of the customers' requirements to be addressed by the project |
| **Inputs** | Customer's requirements, Project Plan Work Package |
| **Activities** | This work package includes the steps that will be taken to identify the stakeholders requirements. Their requirements will be documented and any uncertainties will need to be clarified with the client. |
| **Outputs** | System requirements specification<br>Use cases<br>Transcripts of the meetings with the customers |

| | |
|---|---|
| **Project** | Code Bunny |
| **Work Package** | X03— System Design (3 of 6) |
| **Assigned To** | Mohamad Asyraaf Bin Abdul Rahman<br>Jordan Tan Rei Yao<br>Bachhas Nikita |
| **Effort** | 92.34 PH |
| **Start Date** | 1st March 2021, Monday |
| **Purpose** | Flesh out the design and architecture of the application |
| **Inputs** | Requirements Specification (X01 - X02 Inclusive) |
| **Activities** | This work package includes the steps taken to Identify the different systems and components that are necessary in the app. The systems and components will be formally represented in documents using UML diagrams |
| **Outputs** | Class diagram<br>Sequence diagram<br>Architecture diagram |

| | |
|---|---|
| **Project** | Code Bunny |
| **Work Package** | X04— System Implementation (4 of 6) |
| **Assigned To** | Mohamad Asyraaf Bin Abdul Rahman<br>Jordan Tan Rei Yao<br>Bachhas Nikita |
| **Effort** | 76.95 PH |
| **Start Date** | 16th March 2021, Tuesday |
| **Purpose** | Create a prototype to present to stakeholders and gain feedback<br>Start development on the Code Bunny app |
| **Inputs** | System Design Work Package (X01-X03 Inclusive)<br>Some of Work Package X04 has no dependencies and can begin earlier |
| **Activities** | This work package includes the steps to create a working prototype to demonstrate to the stakeholders. Feedback will be gained on the prototype and they will be implemented into the final product. Work on creating the release version of the application will also be started. |
| **Outputs** | Prototype of the Code Bunny app<br>Fully realised and functional Code Bunny app |

| | |
|---|---|
| **Project** | Code Bunny |
| **Work Package** | X05— Testing (5 of 6) |
| **Assigned To** | Ng Qin Wei |
| **Effort** | 76.95 PH |
| **Start Date** | 31st March 2021, Wednesday |
| **Purpose** | To determine if the app fits the requirements and performs as expected. Bugs must be identified and patched |
| **Inputs** | System Implementation Work Package (X01 - X04 Inclusive) |
| **Activities** | This work package includes the steps taken to test if the app satisfies the requirements set by the stakeholders. Test cases will be created to rigorously test the application to meet the quality standards set by the company. Any bugs found will need to be documented and sorted by severity. The bugs will then be fixed depending on their severity. |
| **Outputs** | Test cases<br>Reports on the bugs that have been discovered and the steps taken to fix them |

| | |
|---|---|
| **Project** | Code Bunny |
| **Work Package** | X06— Deployment and Maintenance (6 of 6) |
| **Assigned To** | Chew Zhi Kang |
| **Effort** | 82.08 PH |
| **Start Date** | 15th April 2021, Thursday |
| **Purpose** | Release the application to stakeholders. Provide post launch support and maintenance. |
| **Inputs** | Testing Work Package (X01-X05 Inclusive) |
| **Activities** | This work package includes the steps taken to deploy the application to stakeholders. Servers may need to be scaled to meet demand. Issues such as bugs will need to be quickly responded to. Feedback from the players will be obtained and patches will be released. |
| **Outputs** | Production ready Code Bunny app |

# 6. Product Checklist

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

| Project Deliverable | Estimated Deadline |
|---|---|
| Project Plan | Feb 13th, 2021 |
| Requirements Specification | Feb 28th , 2021 |
| Design Document | Mar 15th, 2021 |
| Module/System Test Plan | Apr 14th , 2021 |
| Product Demonstration | Apr 15th , 2021 |
| Full Release | Apr 30th , 2021 |

# 7. Best Practice Checklist

| Practice |
| --- |
| Document what we do; all documentation must be in a standardized format. The headers of documentation must have a font size of 20 while the rest of the document must have a font size of 12. The font used must be Times New Roman. Diagrams must be created using lucidchart and conform to UML. |
| Pay attention to requirements, check for ambiguity, completeness, accuracy, and consistency. The requirement documentation must contain a complete functional specification. |
| Keep it simple. Complexity management is one of the major challenges. Strive to:<br>• Minimize interfaces between modules, procedures and data.<br>• Minimize interfaces between people, otherwise exponential communication cost<br>• Avoid fancy product functions, design as long as the functionality meets the customer requirements |
| Communication is essential. Employees must not hesitate to offer up their suggestions or sound out any discrepancies to the management. Managers will also need to work closely with their employees. Developers are required to make their code available for review. |
| Plan for continuous change. We must:<br>• All manuals designs, test, source code should have revision numbers and dates revision history comments, change marks to indicate the changes<br>• New revisions should be approved before being made and checked for quality and compliance after being made<br>• Use a configuration management system and make processes<br>• Required maintenance |
| A good work life balance is highly recommended. Employees must not be forced to work overtime for long periods of time and appropriate breaks should be given to employees. |

| |
|---|
| 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. |
| Pair programming is recommended to reduce bugs and encourage discussion on the current code they are working on. |
| 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. It will involve unit, functional, integrating and acceptance testing. |

# 8. Risk Management

On top of general risk management, our team has discovered the following potential risks for the Cafe Bunny Project.

| | |
|---|---|
| **Risk** | Insufficient server capacity |
| **Impact Severity** | High |
| **Probability** | Medium |
| **Impacts** | Causes downtime for users of the app, reducing user satisfaction |
| **Risk Reduction** | Prepare additional servers during estimated peak periods of user activity |

| | |
|---|---|
| **Risk** | Underestimate time required to develop software |
| **Impact Severity** | High |
| **Probability** | High |
| **Impacts** | Delays in the project can cause a snowball effect delaying the project even further |
| **Risk Reduction** | Do estimations on the size of the project and plan the schedule based on estimations with a 10% increase buffer time |

| | |
|---|---|
| **Risk** | Proposed changes to requirements that require major design rework |
| **Impact Severity** | Medium |
| **Probability** | Medium |
| **Impacts** | Cause delays in the project which can cause a snowball effect delaying the project even further |
| **Risk Reduction** | Response is to avoid by doing proper requirement specification and having proper channels with stakeholders to communicate any changes during the requirements phase of the project |

| | |
|---|---|
| **Risk** | Key developers are ill and unavailable at critical times |
| **Impact Severity** | Medium |
| **Probability** | Low |
| **Impacts** | Cause uncertainties and delay in the project |
| **Risk Reduction** | Accept it and try to proceed with the project based on plans |

| | |
|---|---|
| **Risk** | Server data loss |
| **Impact Severity** | High |
| **Probability** | Low |
| **Impacts** | Reduced user satisfaction and trust in the app |
| **Risk Reduction** | Keeping backup stores of data on backup servers |

| | |
|---|---|
| **Risk** | Critical bugs that impact development times |
| **Impact Severity** | High |
| **Probability** | Medium |
| **Impacts** | Can take a long time to fix and cause project delays |
| **Risk Reduction** | Allocating extra time to the project for contingency purposes |

| | |
|---|---|
| **Risk** | Testing tools in its infancy |
| **Impact Severity** | Medium |
| **Probability** | Low |
| **Impacts** | Might not be well documented and unstable which could mean the time saved from the additional features might not be as much as expected |
| **Risk Reduction** | Avoid using new tools and go for the more established ones |

# 9. Quality Assurance

The team records procedures and details in the Quality Plan, with regards to the team's quality criteria, assessment process and processes for review of quality plan.

The team records test procedures and details in the Test Plan, with regards to the outcome of integration testing and cross-reference with the original SRS document.

In addition, Cafe Bunny shall make use of two testing methodologies:

- **Unit Testing** involves testing system components individually.
- **In-Place Testing** involves testing of the complete system as a unit.

Furthermore, the team will use these methodologies to evaluate two important aspects of the Cafe Bunny system:

- **System Function** ensures that the team reduces software flaws, and
- **Algorithmic Function** ensures that heuristic aspects of the project perform realistically to supply value to the users.

# 10. Monitoring & Control

The team requires procedures to keep track of the progress of the software project successfully.

**Regular reviews of project progress:** Throughout the duration of the Cafe Bunny project, the team shall meet in accordance with the team communication outlines set out in section 4.3 to review the progress of all project tasks, including management, planning, analysis, development, and testing.

**Timeline Planning and task decomposition:** This document outlines an estimated timeline for the project. The team has constructed the timeline by hierarchically decomposing tasks into measurable subcomponents and estimating requirements for each. At the same time, this decomposition aids in task assignment and balancing. Throughout the implementation phase, these subcomponents allow for fine-grained measurement of progress. The team has included project subcomponents and timeline estimates in the Estimates and Work Breakdown Structure sections of this document.

**Identification of major project risks:** Early identification of major risks to the project allows for placement of preventative measures before problems can develop. The team has identified major risks in the Risk Management section of this document, as well as more detailed breakdowns in the Risk Management Plan.