# CEBU INSTITUTE OF TECHNOLOGY - UNIVERSITY

## COLLEGE OF ENGINEERING AND ARCHITECTURE

## Software Project Management Plan

*for*

*Lucky Paws*
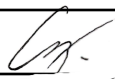
# Signature

| Name | Role | Signature |
|------|------|-----------|
| Amadeo, Carl Marlo M. | App Developer | |
| Camandona, Tristan Ace M. | App Developer | |
| Dela Cerna, Lyndon Kirk R. | App Developer | |
| Flores, Michael B. | App Developer | |
| Mobe, Mita S. | App Developer | |

*Table 1. Signature*

# Change History

| Version | Date | Reason For Change |
|---------|------|-------------------|
| 1 | July 12, 2022 | Making the SPMP |
| | | |
| | | |
| | | |
| | | |

*Table 2. Change History*

# Preface

This document includes the following proposed plan and timetable of tasks that would serve as a roadmap for the project's proponents as the developers decide on their course of action. Additionally, it would outline the process model that would be applied to properly complete this project.

# Table of Contents

# List of Figures

# List of Tables

# 1. Overview

## 1.1. Project Summary

### 1.1.1. Purpose, scope and objectives

This project's goal is to design and create a system for managing pet adoptions. The system will make it possible for animal shelters and rescue organizations to oversee the adoption procedure and monitor the development of each pet. Additionally, it will offer a way to keep track of records and report on adoptions. Adopting a pet will be simple, quick, practical, and effective through this application. This program is accessible to everyone, but it can only be used on desktop computers with internet connectivity. This is convenient for those who love pets or are looking for pets that are available for adoption.

This project's objectives are:

- It aims to provide updates on pets that are available for adoption by providing animal shelters and rescues with an application.
- Users can adopt different kinds of pets in such a way that they can choose via a click the pet that suits their standards of pet.
- They can choose the quality of a pet they want, like in terms of breed, type, life range, and many more.
- to help animal shelters and rescue organizations find owners for the animals they have saved.
- To evaluate the system in order to determine the user's ability, capability, and dependability in pet adoption.
- to make the process of adopting pets simple, quick, and convenient.

### 1.1.2. Assumptions and constraints

Internet connectivity is necessary to run this app, to receive notifications or updates from the app about the pets to be adopted. And also, having a laptop or a desktop computer is also needed to download the software application.The app's users are the ones who post the animals that are available for adoption, thus their access is important. The pet feed would not include any material if there were no users to submit available pets for adoption, hence it is essential. People with bad intentions, such as adopting the animals to eat them, adopting to try to make a profit by selling the animals, etc., might be able to get into the app, but we will ensure safety of the animals, therefore will

require users to provide important information to prove their identity and track them if necessary. We will be able to identify a user who abuses the animals they adopt since they will be required to provide personal information up front and be asked for a legitimate ID, which allows us to identify them. With this, the developers will be able to report and trace them with the help of the National Bureau of Investigation.

### 1.1.3. Project deliverables

All of the items in this section will be sent to the client before the project is finished.

- Project Documentation
  - ➢ Software Requirement Specifications(SRS)
  - ➢ Software Project Management Plan(SPMP)
  - ➢ Software Design Description(SDD)
  - ➢ Software Text Document(STD)
- A prototype version of the Lucky Paws will be presented through the implementation and testing of increments.

### 1.1.4. Schedule and budget summary

| Task Name | Duration | Start | Finish | Cost |
|---|---|---|---|---|
| ● Lucky Paws | 36 days | Fri 01/ 07/ 22 | Fri 05/ 08/ 22 | ₱0.00 |
| ➢ Proposal Stage | 4 days | Fri 01/ 07/ 22 | Mon 04/ 07/ 22 | ₱0.00 |
| ➢ Requirements Phase | 6 days | Mon 04/ 07/ 22 | Sat 09/ 07/ 22 | ₱0.00 |
| ➢ Specification Phase | 8 days | Mon 11/ 07/ 22 | Sun 17/ 07/ 22 | ₱0.00 |
| ➢ Design Phase | 11 days | Mon 18/ 07/ 22 | Thu 28/ 07/ 22 | ₱0.00 |
| ➢ Implementation and Integration Phase | 24 days | Mon 11/ 07/ 22 | Wed 03/ 08/ 22 | ₱0.00 |
| ➢ Final Integration and Revision | 2 days | Wed 03/ 08/ 22 | Thu 04/ 08/ 22 | ₱0.00 |
| Final Submission and Defend | 1 day | Fri 05/ 08/ 22 | Fri 05/ 08/ 22 | ₱0.00 |

*Table 1.1.4. Schedule and Budget Summary*

## 1.2.   *Evolution of plan*

| Version | Primary Author(s) | Description of Version | Date Expected |
|---|---|---|---|
| First Draft | Team | An initial draft was created for the Software Project Management Plan documentation and distributed to the whole team for review and comments. | July 12, 2022 |
| Final Draft | Team | The first completed draft is to be passed to the instructor/adviser for final review comments and corrections. | July 14, 2022 |
|  |  |  |  |
|  |  |  |  |

*Table 1.2.1 Evolution  Plan*

# 2. References

*https://en.wikipedia.org/wiki/Database*
*https://en.wikipedia.org/wiki/Verification_and_validation#cite_note-1*
*https://en.wikipedia.org/wiki/Windows_Forms#:~:text=A%20Windows%20Forms%20application%20is,box%20or%20click%20a%20button.*
*https://www.merriam-webster.com/dictionary/subordinate%20clause#:~:text=grammar,ill%22%20is%20a%20subordinate%20clause.*
*https://www.computerhope.com/jargon/w/windows.htm#:~:text=Microsoft%20Windows%20(also%20referred%20to,and%20connect%20to%20the%20Internet.*
*https://www.computerhope.com/jargon/o/os.htm*
*https://www.techtarget.com/searchapparchitecture/definition/software*

# Definitions

| Terms | Definitions |
|---|---|
| DB | (Database) In computing, a **database** is an organized collection of data stored and accessed electronically. Small databases can be stored on a file system, while large databases are hosted on computer clusters or cloud storage. The design of databases spans formal techniques and practical considerations including data modeling, efficient data representation and storage, query languages, security and privacy of sensitive data, and distributed computing issues including supporting concurrent access and fault tolerance. |
| V&V | (Verification and validation) **Verification and validation** (also abbreviated as **V&V**) are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it fulfills its intended purpose. |
| WFA | A ***Windows Forms application*** is an event-driven application supported by Microsoft's .NET Framework. Unlike a batch program, it spends most of its time simply waiting for the user to do something, such as fill in a text box or click a button. The code for the application can be written in a .NET programming language such as C# or Visual Basic. |
| Sub Clause | **Subordinate clause,** a clause that does not form a simple sentence by itself and that is connected to the main clause of a sentence. |
| Milestones | An action or event marking a significant change or stage in development. |
| OS | **Operating System** is software installed on a computer's hard drive that enables the computer hardware to communicate and operate with the computer software. Without a computer operating system, a computer and software programs would be useless. |
| Software | is a set of instructions, data or programs used to operate computers and execute specific tasks. |
| Windows | Microsoft Windows (also referred to as Windows or Win) is a graphical operating system developed and published by Microsoft. It provides a way to store files, run software, play games, watch videos, and connect to the Internet. |

# Project organization

## 4.1. External structure

None. The developers do not have a specific or definite client, this sub-clause is irrelevant.

## 4.2. Internal structure
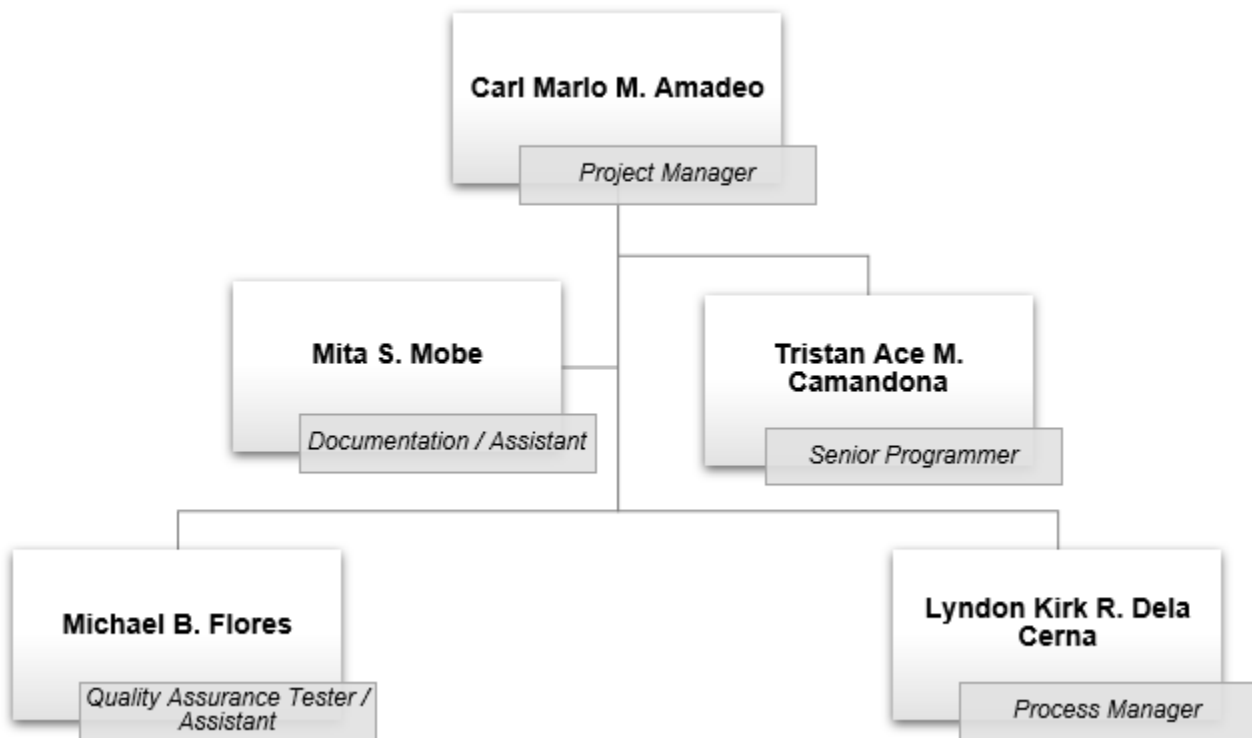


*Figure 4.2.1. Organization Diagram*

## 4.3. Roles and responsibilities

| Role | Responsibilities |
|---|---|
| **Project Manager** | Software project planning and monitoring. |
| | Milestone and schedule planning and monitoring. |
| | Set and communicate the team meeting agendas. |
| | Risk Management. |

| Role | Responsibilities |
|---|---|
| | Establish severity of risks and define target dates. |
| **Documentation** | Software requirement planning and acquisition<br><br>Ensure the SRS has completely cover all user's requirements<br><br>Responsible to track user's requirements through entire project<br><br>Ensure compliance with the processes outlined according documentation.<br><br>Conduct process reviews to ensure compliance. |
| **Quality Assurance Tester** | Deliverable planning.<br><br>Manage deliverable expectations to ensure that they are in sync with the team's delivery potential.<br><br>Risk and error management. |
| **Process Manager** | Process definition.<br><br>Set and implement development standards.<br><br>Leads the team in defining the processes, practices, and procedures.<br><br>Conduct process reviews to ensure compliance.<br><br>Ensures that the team's recording all required data. |
| **Senior Programmer** | Leads the team in producing, assessing, updating, and reporting on the plan.<br><br>Provides the planning assumptions.<br><br>Consolidates the individual plans into the team plan.<br><br>Programs the application. |
| **Assistant** | Assist in planning, programming, documenting, and testing the application. |

*Table 4.3.1 Role and Responsibilities*

| Role | Persons' responsible |
|------|----------------------|
| **Project Manager** | Carl Marlo M. Amadeo |
| **Documentation** | All members |
| **Senior Programmer** | Tristan Ace M. Camandona |
| **Quality Assurance Tester** | Michael B. Flores |
| **Process Manager** | Lyndon Kirk R. Dela Cerna |
| **Assistant** | Mita S. Mobe & Michael B. Flores |

*[Table 4.3.2] Role Allocation*

# Managerial process plans

## 5.1. Start-up plan

### 5.1.1. Estimation plan

The software project, Lucky Paws, must be completed by the summer term of academic year 2021–2022. Our Software Engineering course requires this project. By the end of this summer term, it ought to be finished. The schedule we adhere to is based on the projected completion date our instructor provided. We must adhere to the deadlines set forth for completing each phase if we are to effectively manage the project's flow and structure. Even though we try hard to meet deadlines, we might sometimes miss them because some of us are still learning how to use Windows Forms apps.

### 5.1.2. Staffing plan

The main contributors to the project are listed in the table.

| Name | Affiliation to project | A.Y. 2022 |
|---|---|---|
| Carl Marlo M. Amadeo | Student / Project Manager | Full Time |
| Tristan Ace M. Camandona | Student / Senior Programmer | Full Time |
| Lyndon Kirk R. Dela Cerna | Student / Process Manager | Full Time |
| Michael B. Flores | Student / Quality Assurance Tester, Assistant | Full Time |
| Mita S. Mobe | Student / Documentation, Assistant | Full Time |

*Table 5.11.2 Staff Plan*

### 5.1.3. Resource acquisition plan

The project's development team learns about the resources they require by consulting available software, textbooks, computer resources, and the internet. The project manager helps the group figure out what resources and help they need to get the things they need.

### 5.1.4. Project staff training plan

The development team for this project is made up entirely of students. Each person will be assigned to the section or module where they are most qualified to do so. This project can be finished

with the aid of the foundational knowledge acquired in school. However, careful and in-depth research must be done in order to be able to meet the project's standards.

## 5.2. Work plan

### 5.2.1. Work activities

For the Work Activities:

| Task Name | Duration | Start | Finish |
|---|---|---|---|
| **Lucky Paws** | **36 days** | **Fri 01/ 07/ 22** | **Fri 05/ 08/ 22** |
| **● Proposal Stage** | **4 days** | **Fri 01/ 07/ 22** | **Mon 04/ 07/ 22** |
| ➤ Brainstorming of Idea | 3 days | Fri 01/ 07/ 22 | Sun 03/07/22 |
| ➤ Proposal of Project | 1 day | Mon 04/ 07/ 22 | Mon 04/ 07/ 22 |
| ➤ Approval of Project | 1 day | Mon 04/ 07/ 22 | Mon 04/ 07/ 22 |
| **● Requirements Phase** | **6 days** | **Mon 04/ 07/ 22** | **Sat 09/ 07/ 22** |
| **➤ Creating SRS** | **2 days** | **Mon 04/ 07/ 22** | **Tue 05/ 07/ 22** |
| - Drafting SRS | 1 day | Mon 04/ 07/ 22 | Mon 04/ 07/ 22 |
| - Drafting for User Interface | 1 day | Tue 05/ 07/ 22 | Tue 05/ 07/ 22 |
| ➤ Evaluation of SRS | 1 day | Wed 06/ 07/ 22 | Wed 06/ 07/ 22 |
| ➤ Revision of SRS | 3 days | Wed 06/ 07/ 22 | Sat 09/ 07/ 22 |
| **● Specification Phase** | **8 days** | **Mon 11/ 07/ 22** | **Sun 17/ 07/ 22** |
| ➤ Created draft of SPMP | 3 days | Mon 11/ 07/ 22 | Wed 13/ 07/ 22 |
| ➤ Evaluation of SPMP | 1 day | Thu 14/ 07/ 22 | Thu 14/ 07/ 22 |
| ➤ Revision of SPMP | 4 days | Thu 14/ 07/ 22 | Sun 17/ 07/ 22 |
| **● Design Phase** | **11 days** | **Mon 18/ 07/ 22** | **Thu 28/ 07/ 22** |
| **➤ Created Draft of SDD** | **4 days** | **Mon 18/ 07/ 22** | **Thu 21/ 07/ 22** |
| - Gathering Data | 4 days | Mon 18/ 07/ 22 | Thu 21/ 07/ 22 |
| - Research about Pet Adoption | 2 days | Mon 19/ 07/ 22 | Thu 21/ 07/ 22 |
| ➤ Evaluation of SDD | 1 day | Fri 22/ 07/ 22 | Fri 22/ 07/ 22 |
| ➤ Revision of SDD | 3 days | Fri 22/ 07/ 22 | Sun 24/ 07/ 22 |
| ➤ Create Draft of STD | 2 days | Mon 25/ 07/ 22 | Tue 26/ 07/ 22 |
| ➤ Evaluation of STD | 1 day | Wed 27/ 07/ 22 | Wed 27/ 07/ 22 |
| ➤ Revision of STD | 1 day | Wed 27/ 07/ 22 | Thu 28/ 07/ 22 |
| **● Implementation and Integration Phase** | **24 days** | **Mon 11/ 07/ 22** | **Wed 03/ 08/ 22** |
| **➤ Increment 1** | **4 days** | **Mon 11/ 07/ 22** | **Thu 14/ 07/ 22** |
| **➤ Creation of UI** | **3 days** | **Mon 11/ 07/ 22** | **Wed 13/ 07/ 22** |
| - Get Started | 1 day | Mon 11/ 07/ 22 | Mon 11/ 07/ 22 |
| - Log In | 1 day | Tue 12/ 07/ 22 | Tue 12/ 07/ 22 |

*Table 5.2.1. Work Activities*

| | | | |
|---|---|---|---|
| - Home | 1 day | Tue 12/ 07/ 22 | Tue 12/ 07/ 22 |
| - Pet Feed | 1 day | Wed 13/ 07/ 22 | Wed 13/ 07/ 22 |
| ➢ Integration | 1 day | Wed 13/ 07/ 22 | Wed 13/ 07/ 22 |
| ➢ Revise STD | 1 day | Wed 27/ 07/ 22 | Thu 28/ 07/ 22 |
| ➢ Submit and Evaluate Increment | 1 day | Thu 14/ 07/ 22 | Thu 14/ 07/ 22 |
| ➢ **Increment 2** | **8 days** | **Fri 15/ 07/ 22** | **Fri 22/ 07/ 22** |
| ➢ **Features** | **6 days** | **Fri 15/ 07/ 22** | **Wed 20/ 07/ 22** |
| - Profile | 3 days | Fri 15/ 07/ 22 | Sun 17/ 07/ 22 |
| - Pick Me | 1 day | Mon 18/ 07/ 22 | Mon 18/ 07/ 22 |
| - Look For Shelter | 2 days | Tue 19/ 07/ 22 | Wed 20/ 07/ 22 |
| - Log out | 1 day | Wed 20/ 07/ 22 | Wed 20/ 07/ 22 |
| ➢ Updating UI | 2 days | Wed 20/ 07/ 22 | Thu 21/ 07/ 22 |
| ➢ Integration | 1 day | Thu 21/ 07/ 22 | Thu 21/ 07/ 22 |
| ➢ Revise STD | 1 day | Wed 27/ 07/ 22 | Wed 27/ 07/ 22 |
| ➢ Submit and Evaluate Increment | 1 day | Fri 22/ 07/ 22 | Fri 22/ 07/ 22 |
| ➢ **Increment 3** | **7 days** | **Sat 23/ 07/ 22** | **Fri 29/ 07/ 22** |
| ➢ Integrating Database for Account | 3 days | Sat 23/ 07/ 22 | Mon 25/ 07/ 22 |
| ➢ **Features** | **5 days** | **Sat 23/ 07/ 22** | **Wed 27/ 07/ 22** |
| - Create account | 3 days | Sat 23/ 07/ 22 | Mon 25/ 07/ 22 |
| - Post a pet | 2 days | Mon 25/ 07/ 22 | Wed 27/ 07/ 22 |
| ➢ **Updating UI** | **2 days** | **Wed 27/ 07/ 22** | **Thu 28/ 07/ 22** |
| - Profile | 2 days | Wed 27/ 07/ 22 | Thu 28/ 07/ 22 |
| - Pet feed | 2 days | Wed 27/ 07/ 22 | Thu 28/ 07/ 22 |
| ➢ Integration | 1 day | Thu 28/ 07/ 22 | Thu 28/ 07/ 22 |
| ➢ Revise STD | 1 day | Thu 28/ 07/ 22 | Thu 28/ 07/ 22 |
| ➢ Submit and Evaluate Increment | 1 day | Fri 29/ 07/ 22 | Fri 29/ 07/ 22 |
| ➢ **Increment 4** | **4 days** | **Sat 30/ 07/ 22** | **Wed 03/ 08/ 22** |
| ➢ Integrating Database for Pets Posted | 3 days | Sat 30/ 07/ 22 | Mon 01/ 07/ 22 |
| ➢ **Features** | **2 days** | **Mon 01/ 07/ 22** | **Tue 02/ 07/ 22** |
| - Pet feed will generate | 2 days | Mon 01/ 07/ 22 | Tue 02/ 07/ 22 |
| photos posted by user. | | | |
| ➢ Updating UI | 2 days | Mon 01/ 07/ 22 | Tue 02/ 07/ 22 |
| ➢ Finalization | 1 day | Tue 02/ 08/ 22 | Tue 02/ 08/ 22 |
| ➢ Verification | 1 day | Tue 02/ 08/ 22 | Tue 02/ 08/ 22 |
| ➢ Validation | 1 day | Tue 02/ 08/ 22 | Tue 02/ 08/ 22 |
| ➢ Revise STD | 1 day | Tue 02/ 08/ 22 | Tue 02/ 08/ 22 |
| ➢ Submit and Evaluate Increment | 1 day | Wed 03/ 08/ 22 | Wed 03/ 08/ 22 |
| ● **Final Integration and Revision** | **2 days** | **Wed 03/ 08/ 22** | **Thu 04/ 08/ 22** |
| ➢ **Documents** | **2 days** | **Wed 03/ 08/ 22** | **Thu 04/ 08/ 22** |
| ➢ Software Requirement Specification | 2 days | Wed 03/ 08/ 22 | Thu 04/ 08/ 22 |
| ➢ **Software Project Management Plan** | **2 days** | **Wed 03/ 08/ 22** | **Thu 04/ 08/ 22** |
| Revise Workplan | 2 days | Wed 03/ 08/ 22 | Thu 04/ 08/ 22 |
| ➢ **Software Design Description** | **2 days** | **Wed 03/ 08/ 22** | **Thu 04/ 08/ 22** |
| Updating User Interface | 2 days | Wed 03/ 08/ 22 | Thu 04/ 08/ 22 |
| ➢ **Software Test Document** | **2 days** | **Wed 03/ 08/ 22** | **Thu 04/ 08/ 22** |
| Making Test Cases | 2 days | Wed 03/ 08/ 22 | Thu 04/ 08/ 22 |
| ➢ Application Testing | 1 day | Thu 04/ 08/ 22 | Thu 04/ 08/ 22 |
| ➢ Submit and Evaluate STD | 1 day | Thu 04/ 08/ 22 | Thu 04/ 08/ 22 |
| Final Submission and Defend | 1 day | Fri 05/ 08/ 22 | Fri 05/ 08/ 22 |

### *5.2.2. Schedule allocation*

This sub clause of the SPMP shall assign and schedule available tasks in the most effective and economical way possible among co-developers. The Project Manager shall allocate and assign the developer's tasks without over (or under) allocating their availability. With this, developers are ought to execute and finish their given tasks in the given schedule. Any constraints nor changes caused by factors external to the project must be indicated in the work activity schedule. Including frequent updates about the assigned tasks using objective indicators to assess the scope and quality of work products. Schedule allocation may use the following techniques for depicting work schedules; activity list, activity updates, & Gantt chart.

The project duration is constrained to one summer term, with final deliverables due at the end of the summer term, in the first week of August 2022. We will control our schedule with these main artifacts.

### *5.2.3. Resource allocation*

In this sub clause of the SPMP, each work activity listed in the project's work breakdown structure shall have its resources listed in detail. The distribution of resources may, if necessary, take into account aspects like employees by skill level, computing resources, software tools, specialized testing and simulation facilities, and administrative support. The resources shall include the caliber work of the developer on the given tasks and the numbers associated by the resources. Each sort of resource for each type of labor activity shall have its own line item. From the work packages of the WBS, a summary of the resource needs for the various work activities shall be gathered and provided in tabular form.

The team provided the majority of the resources, and each of us is providing whatever is required to complete the project. The team worked on every single document or activity that was necessary. Five developers must put in three hours every day for Increment 1. (from July 11, 2022 to July 14, 2021). 5 developers must put in 2 hours every day for Increment 2. (from July 15, 2022 to July 22, 2022). Five developers must put in three hours every day for Increment 3. (from July 23, 2022 to July 29, 2021). For Increment 4, five developers must put in four hours a every day (from July 30, 2022, to August 03, 2022).

### *5.2.4. Budget allocation*

In this sub clause of the SPMP, all necessary resource budgets used on the project shall be listed in the project's work breakdown structure in accurate details. The activity budget shall include the estimated cost for all the expenses used in the project such as software tools, transportation

expenses, project resources, computer resources, special testing and stimulation effects, and administrative support. Each sort of resource for each type of the project's expenses shall have its own line item. The activity budgets shall use spreadsheets and shall be gathered and provided in tabular form.

The essential paperwork and software are created without incurring any costs on the part of the team of developers.

## 5.3. Control plan

### 5.3.1. Requirements control plan

The SRS document is in charge of managing the Lucky Paws requirement specification. The most recent and updated version of the aforementioned SRS document will serve as the team's primary source of information when developing the application to make sure that all necessary conditions are satisfied. Changes to the project requirements will reflect the SRS document for this project.

### 5.3.2. Schedule control plan

Group meetings are used to distribute duties to the members. The manager of the development team will be the one to assign specific responsibilities to each team member. The team manager will be in charge of monitoring the members' advancement. To guarantee that the task is not inactive, regular updates are made. The project manager will schedule additional meetings to keep the process moving if any tasks go behind schedule.

### 5.3.3. Budget control plan

None. The developers don't spend anything for the app to be developed.

### 5.3.4. Quality control plan

The project manager will schedule periodic meetings at the end of each phase to monitor and control the project's progress. Members are required to report on how the duties they perform are changing. A highly useful strategy for ensuring that one is still adhering to the specifications being worked on is to continually assess the work that has been completed. This method ensures that any error that has to be fixed will be fixed as soon as possible. The project will be done easily if all members are tracked on a regular basis.

### 5.3.5. Reporting plan

Team members will have a time to time update to their project manager and will inform their accomplishments and problems/ difficulties they've encountered. All accomplished tasks will be submitted to the documentation officer for the summarization. Then the reporting or discussion of the assigned tasks will follow. After the discussion, the project manager will discuss further about the details and information of the given tasks. Then the project manager will sum up everything for clarification.

### 5.3.6. Metrics collection plan

Each member will report to their project manager with regards to the status or progress of their tasks, and the problems encountered while doing the task. Then the project manager will evaluate each member's output  and contributions in the given tasks.

### 5.3.7  Risk management plan

During each meeting, the project manager will initiate risk identification. The developers will come up with their ideas and share the problems in their required task. The risks will be determined by the difficulties encountered by each team member while completing the assigned task. The project manager will create a list of priority risks based on the problems reported by each member. The team will then decide on the best course of action to address the prioritized risks. The members will do their assigned task and update the team manager.

### 5.3.8  Project closeout plan

The group must formally terminate the project. All of the documentations must be submitted to the instructor for any clarifications or suggestions of the project. The final and complete deliverable project must be submitted on or before the given deadline.

# 6.  Technical process plans

## 6.1  Process Model

This project will be carried out following the incremental model. This process model will be used to outline the project's various implementation phases. There will be regular updates and reviews for each step to ensure that the project is still in compliance with the requirements and specifications.
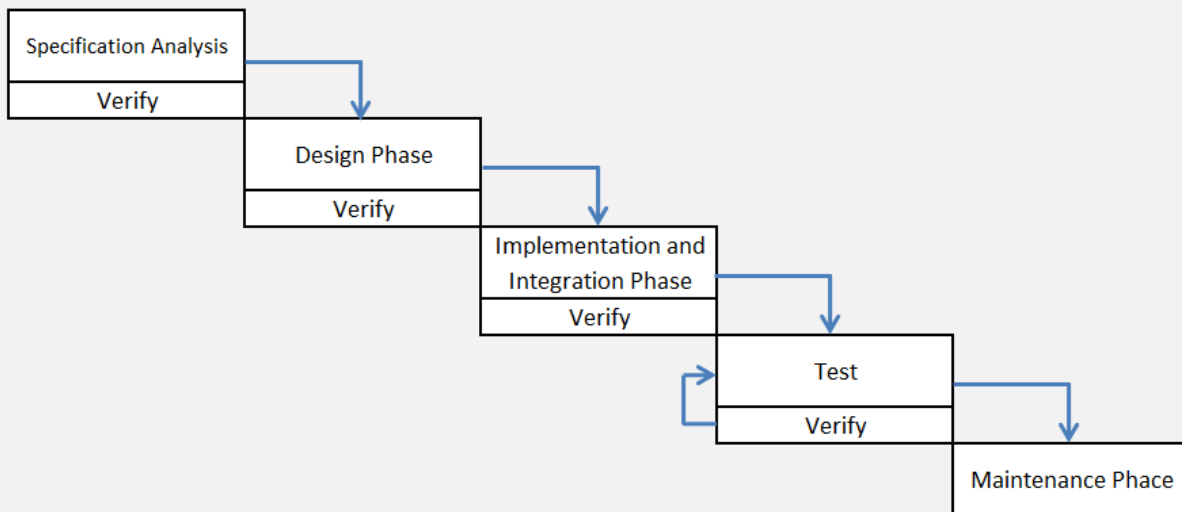


*Figure 6.1.1. Process Model*

### Requirements and Specifications Phase

Primary system specifications and requirements have been gathered in this phase through observation and research.

The team will use methods of observation and study to gather the fundamental system requirements. In order to get a visual perspective of the program as a whole, a straightforward prototype in the form of mocked up UIs will also be produced. The SRS, SPMP, and the project proposal will be the deliverables.

### Design Phase

The architectural and intricate designs of the project are taken into account throughout the

design phase. The SDD will be the deliverable in this case.

### *Implementation Phase*

Plans and designs are converted into codes during the execution phase. Additionally, this phase has the following subphases:

- Detail design
- Design review and inspection
- Code
- Code review
- Compile
- Code inspection
- Unit test

### *Integration and Testing Phase*

The separate modules will be integrated with one another and tested to ensure that they are functioning properly during this phase of the project. This process will be repeated until all modules are integrated and prepared for the system test.

## 6.2  Methods, tools, and techniques

This subclause of the SPMP shall specify the development methodologies, programming languages and other notations, and the tools and techniques to be used to specify, design, build, test, integrate, document, deliver, modify and maintain the project deliverable and non-deliverable work products. In addition, the technical standards, policies, and procedures governing development and/or modification of the work products shall be specified.

The methods and techniques listed in this table will be evaluated and applied in specific areas of the project as appropriate:

| Category | Methods and Techniques |
|----------|------------------------|

| Requirements Elicitation | Meetings<br>Brainstorming<br>Research |
|---|---|
| Formal Specification and Analysis | To model the structural parts of the requirements and design, an incremental model is used.<br>Use cases are used to define requirements. |
| Prototype | Two UI prototypes to validate a technical or design decision. One is for the desktop computer of the user and the other one is for the developer.<br>Queueing system in iterations to validate a technical or design decision. |

*Table 6.2.1 Methods and Techniques*

| Category | Tools |
|---|---|
| Operating System | Windows 7 to Windows 11 |
| Development languages and databases | Databases: Microsoft SQL Server<br>Language: C# |
| Document | All documents will be written using Google docs. |
| Project Planning and Tracking | Microsoft Excel |

*Table 6.3.1 Methods and Techniques*

## 6.3  Infrastructure Plan

The team of developers will be working on their assigned tasks in their respective homes. They will make use of digital advancements such as computers, laptops, and the internet to complete the task. Furthermore, the developers will continue to keep in touch to check the progress of their tasks. Afterwards, they will be meeting personally to gather everything, have a thorough checking and finalize the given task.

## 6.4  Product Acceptance Plan

As of today, these aims are for the CIT-U faculty and students only because this is only for the CIT-U based project. The clients will affirm the needed documents (i.e. agreements) and receive every

time over time of the project formally. Before the application meets the acceptable requirements, the user must be able to create an account, log in, browse through the pet feed, choose a pet to adopt, and submit a pet for adoption. The user should be able to input their personal information and input the necessary details to proceed with creating an account and be able to log in. Log in. The user should be able to input their username and password and be redirected to the homepage once logged in. Browse through the pet feed, where the user should be able to see various posts from various users. Users should be able to pick a pet they admire through a "pick me" button, and then the user should be directed to fill out a form for the pet to be adopted. Users should be able to list pets for adoption by going to a form where they can describe the animal and upload a picture.

## 7.1. Configuration management plan

The management process and planning are discussed in the Managerial Process Plans portion of this document, along with the SRS, and they will be carefully maintained during the course of the project.

## 7.2. Verification and validation plan

Several tasks collectively make up continuing activities that go across the different life cycle phases. Their general activities are traceability analysis, evaluation, interface analysis, and testing. These activities are horizontal threads that ties together the subsequent phase activities and allow verification to be more effectively conducted.

### Traceability analysis

The traceability is the ability to identify the relationships between originating requirements and their resulting system features. It permits tracking forward or backward through the network of interrelationships that are created as requirements are decomposed and refined through a system's life cycle. Traceability allows verification of the properties set forth in the concept and that requirement specifications have been carried forward to the design specification, implemented in the code, included in the test plan and cases, and provided to the customer and user in the resulting system.

### Evaluation

Evaluation ascertains the value or worth of an item and help to assure that a system meets its specifications. Evaluations are performed by many persons across all life cycle phases, on both interim and final software products, and may be either a comprehensive or selective assessment of a

system. Evaluations are used through all phases and for all type of software products, including user documents, manuals, and other project documents. These may be of many forms, such as text or graphic representations, and in various media, such as paper, magnetic tape, diskette, and computer files. This range of product types and forms requires a large variety of techniques for performing and managing software evaluations.

### *Interface analysis*

When information is passed across a boundary, there is always the possibility of losing some information or alerting the information content. The task of interface analysis serves to ensure the completeness, accuracy, and consistency of these interfaces. Interface requirements at the design and implementation phases should be identified and analyzed at the functional, physical, and data interface level. The goal of interface analysis is to evaluate the specific software deliverables (e.g., requirements, design, code) for correct, consistent, complete, and accurate interpretation of the interface requirements.

### *Testing*

In the context of software verification and validation, testing can be defined as the testing that is performed in support of the V&V objectives. These objectives may differ from those of the developer. Testing is performed at several points in the life cycle, starting from the requirement phase up to the test phase. The various test activities are listed below:

### Component Testing

Testing conducted to verify the implementation of the design for one software element or a collection of software elements.

### Integrating Testing

An orderly progression of testing in which software elements, hardware elements, or both are combined and tested until the entire system has been integrated.

### System Testing

The process of testing an integrated hardware and software system to verify that the system meets its specified requirements.

### Acceptance Test

Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system.

This section explains out V&V plan for each phase of software development.

| Milestone | V&V Input | V&V Tasks | V&V Output |
|---|---|---|---|
| Project Proposal Completion and Submission | - Project Proposal Document | - Proposal Review and Verification<br>- Proposal Tracing and Validating | Project Proposal Revised document |
| Software Requirements Specification Completion and Submission | - Software Requirements Specification (SRS) Document | - SRS review and verification<br>- SRS Tracing and Validating | SRS Revised Document |
| Software Project Management Plan Completion and Submission | - Software Project Management Plan (SPMP) Document | - SPMP review and verification<br>- SPMP Tracing and Validating | SPMP Revised Document |
| Software Design Document Completion and Submission | - Software Design Document (SDD) | - SDD review and verification<br>- SDD Tracing and Validating | SDD Revised document |
| Software Test Document Completion and Submission | - Software Test Document<br>- Source Code<br>- Software Prototype | - Test Procedure Review and Verification<br>- Test Error Tracing and Validating<br>- Module Test | Completed Project |
| Implementation | - Source Code | - Source Code Review and Verification<br>- Source Code Tracing, Debugging, and Validating<br>- Integration Test<br>- System Test | Implemented Project |

*Table 7.2.2 V&V Plan*

## 7.3. Documentation plan

Documents specified here will be produced during the project development phase. All members will be responsible for the documentation of the progress of the project. Here are the list of the documents that will be created and maintained under version control:

- Project Proposal
- Software Requirement Specification (SRS) - an in-depth description of a software product to be developed.
- Software Project Management Plan (SPMP) - summarizes the project management models or plan.
- Software Design Document (SDD)
- Software Test Documentation (STD)

## 7.4. Quality assurance plan

The QAT manager will ensure that the overall quality of the application developed exceeds the expectations of the user. The QAT manager will be responsible for the management and testing of the codes produced that will result in the achievement of the target purpose of the application project. Along with the QAT manager, the project manager will also be present and will aid in the testing of the codes to review and keep track of the possible errors that may occur and get rid of them.

## 7.5. Reviews and audits

The QAT manager, programmers and project manager are the ones responsible for the project reviews that will be performed on a weekly basis to ensure that the program functions properly and according to its purpose. On the other hand, the audits will only be performed on request and will be observed only by the project manager.

## 7.6. Problem resolution plan

The problems encountered by the team within the project will always be reported to the the project manager and be noted by the documentation officer. The project manager will be responsible for providing the instructions to the members assigned in their respective fields in solving the problems encountered covered within their fields. Each member will also be providing suggestions to the project manager for solutions to the problems, but the final plan of solution will depend on the project manager.

## 7.7. Subcontractor management plan

Not applicable for the Lucky Paws app.

## 7.8. Process improvement plan

The creation of the  process improvement plan will be in the "improvement plans" section during the final phases of the project. As of now, it is not feasible to be made.

## 8.0  Additional Plans

This clause of the SPMP shall contain additional plans required to satisfy product requirements and contractual terms. Additional plans for a particular project may include plans for assuring that safety, privacy, and security requirements for the product are met, special facilities or equipment, product installation plans, user training plans, integration plans, data conversion plans, system transition plans, product maintenance plans, or product support plans.

# 9. Plan Annexes

Annexes may be included, either directly or by reference to other documents, to provide supporting details that could detract from the SPMP if included in the body of the SPMP.

# 10. Index