Petpal

[Project Plan]
Version 1.0

Revision History

Revision Number	Date	Primary Author(s)	Comments
1.0	Sep 28th, 2024	Gambhir Dhruv; Mehta Viral Sujal; Mishra Apurva; Najah Ismail; Nithya Hariharan; Tan Shu Hua, Samantha; Tan Jing Jie	First version

Table of Contents

Revision History....2
Table of Contents....3
1 Introduction....4
1.1 Project Overview....4
1.2 Project Description and Scope....4
2 Project Organization....5
2.1 Team Structure....5
2.2 Roles and Responsibilities....5
2.3 Team Communication....6
3 Process Definition....7
3.1 Lifecycle Model....7
4 Schedule....8
4.1 Activity Dependencies and Schedule....8
4.2 Work Breakdown Structure....9
4.3 Work Packages....10
4.4 Activity Dependencies....10
4.5 Work Package Details....11
5 Project Estimates....15
5.1 Code Size Estimation using Function Points....15
5.1.1 Unadjusted Function Points....15
5.1.2 Adjusted Function Points....19
5.1.3 Lines of Code....19
5.2 Efforts, Duration and Team Size Estimation....20
5.2.1 Distribution of Effort....20
5.3 Cost Estimates....21
6 Product Checklist....22
7 Best Practice Checklist....23
8 Risk Management....24
9 Quality Assurance....25
10 Monitoring & Control....26

1 Introduction

1.1 Project Overview

The PetPal platform offers a unified solution for engaging in various pet-related activities, including pet adoption, pet sitting services, and organizing or joining pet events. Our goal is to enhance the convenience for pet lovers by providing a single platform for all their pet-related needs.

1.2 Project Description and Scope

The PetPal platform integrates features for pet agencies to list pets, for pet sitters to offer their services, and for pet lovers to adopt pets, find sitters, and host or participate in pet-themed events. Key features include:

- User interface showing a dashboard displaying user accounts, registered events, contacted hosts, adoption agencies, and personal profiles.
- Event search functionality to find pet events by location, theme, and duration, along with a page to create and host new events.
- Pet sitter search functionality for locating pet sitters based on proximity, cost, and pet type, along with information on available services.
- Pet adoption search functionality to find pets available for adoption by type, breed, and other criteria.
- Agency interface for pet agencies to list pets available for adoption

2 Project Organization

2.1 Team Structure

Name	Role
Mishra Apurva	Project Manager
Tan Shu Hua, Samantha	QA Manager
Najah Ismail	Release Manager
Nithya Hariharan	Frontend Developer
Gambhir Dhruv	Lead Developer
Tan Jing Jie	QA Engineer
Mehta Viral Sujal	Backend Developer

2.2 Roles and Responsibilities

Project Manager: Mishra Apurva

- Oversee project lifecycle from planning to deployment.
- Coordinate between teams and manage resources.
- Ensure adherence to Agile methodology and manage sprint cycles.
- Track progress using Trello and ensure tasks meet deadlines and requirements.

QA Manager: Tan Shu Hua, Samantha

- Develop and manage QA strategy and test plans.
- Coordinate with QA engineers for comprehensive testing.
- Ensure all features are thoroughly tested in each sprint.
- Track and report bugs in Trello for resolution.

Release Manager: Najah Ismail

- Manage deployment process and automate CI/CD pipelines.
- Collaborate with developers and QA for integration and delivery.

Frontend Developer: Nithya Hariharan

- Develop responsive and SEO-friendly user interface using NextJS.
- Implement SSR and SSG for optimized performance.
- Collaborate with QA for testing and debugging.
- Ensure user experience aligns with project requirements.

Lead Developer: Gambhir Dhruv

- Define technical direction and architecture for the project.
- Mentor development team and ensure code quality.
- Ensure integration between front-end (NextJS) and back-end (Flask).

QA Engineer: Tan Jing Jie

- Perform manual and automated testing on front-end and back-end.
- Identify defects and provide feedback to developers.
- Document and re-test issues after fixes.
- Utilize testing tools and frameworks effectively.

Backend Developer: Mehta Viral Sujal

- Develop server-side logic and APIs using Flask.
- Manage data exchange between front-end and PostgreSQL database.
- Implement security measures and third-party integrations.
- Collaborate with QA for back-end testing and debugging.

2.3 Team Communication

Communication channels include the following:

- Weekly meetings over zoom and in lab sessions.
- Group announcements and updates are sent through a telegram group chat..
- Split up into subgroups as necessary, in order to work more cooperatively on specific problems.
- For feedback on written deliverables is done through comments on Google docs or Word document

3 Process Definition

3.1 Agile

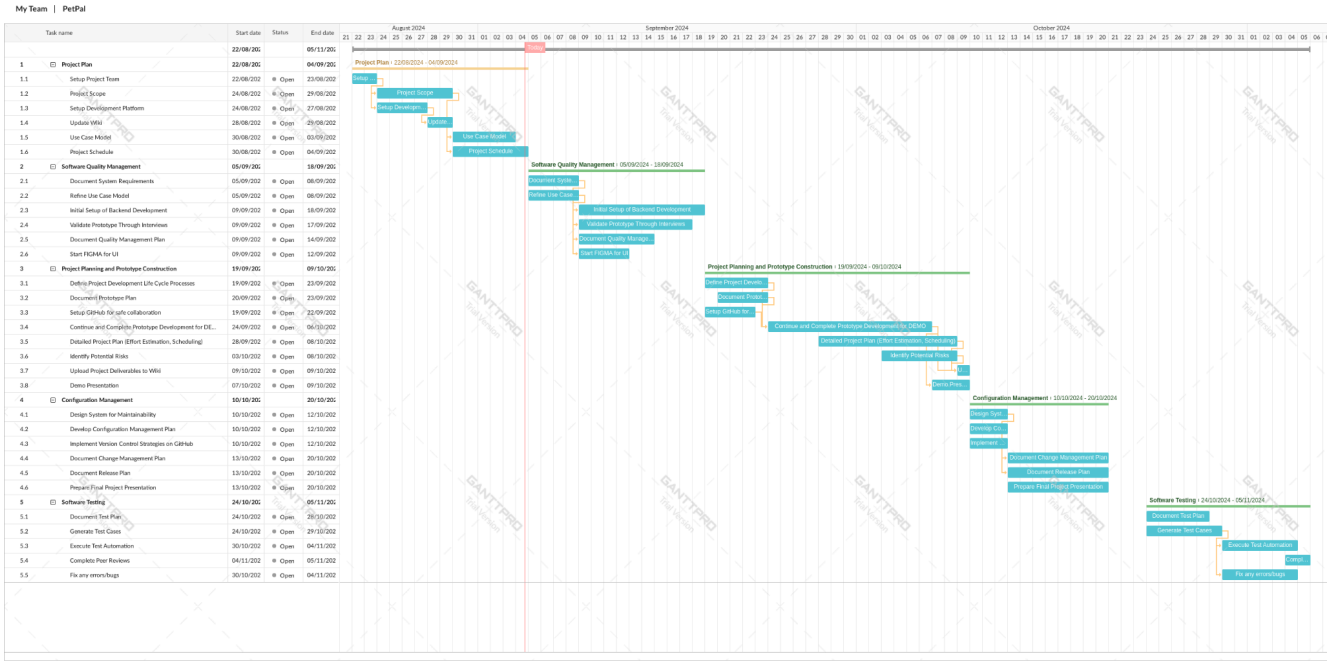
Agile Development Model will be used throughout the PetPal project. This methodology is more flexible than the traditional Waterfall SDLC due to repeated iterations involving design, coding, unit testing, integration, and quality assurance. The Waterfall SDLC is not a viable choice due to the short timeline available for the PetPal project to reach delivery quality.

Methodologies as Spiral are avoided because of concerns over the short timeline.

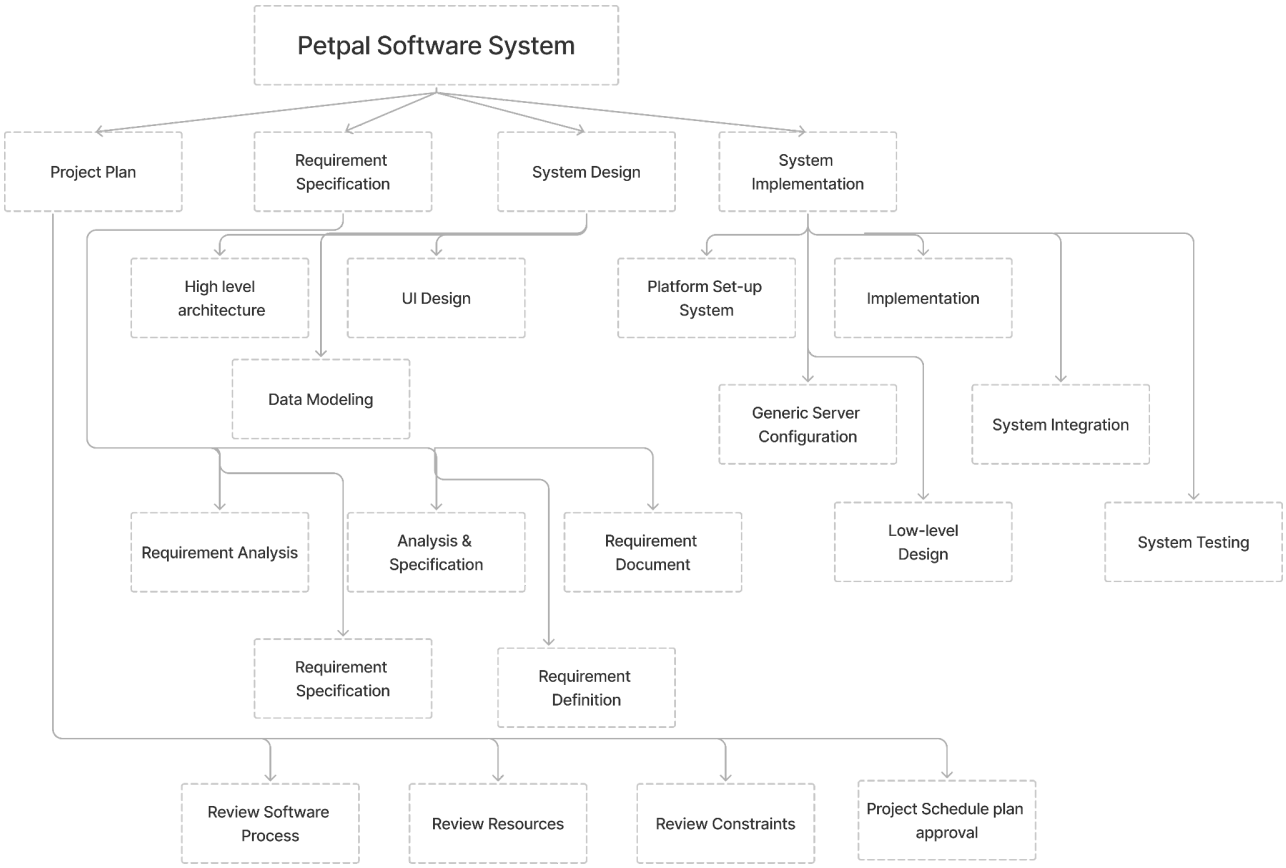
Agile will allow the team to continuously deliver functional components, such as the pet adoption interface, event management features, and pet sitter services, in incremental releases. The team intends to deliver functionality on the System Delivery date indicated in the Estimations section of this document.

4 Schedule

4.1 Activity Dependencies and Schedule



4.2 Work Breakdown Structure



4.3 Work Packages

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

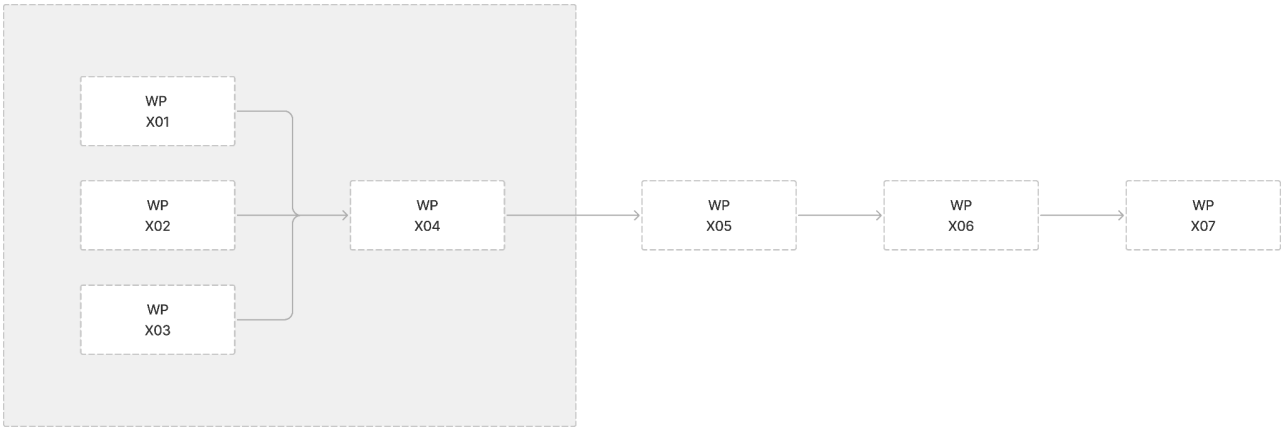
1. Project Plan
2. Requirement Specification
3. User Interface
4. Technical Architecture
5. Data Modeling
6. Coding & Unit Testing
7. Integration & Quality Assurance

4.4 Activity Dependencies

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

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

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



.

4.5 Work Package Details

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

Project	PetPal Software System
Work Package	X01— Project Plan (1 of 7)
Assigned To	**Mishra Apurva** & Tan Shu Hua, Samantha & Najah Ismail & Nithya Hariharan & Gambhir Dhruv & Tan Jing Jie & Mehta, Viral Sujal
Effort	7PD
Start Date	Thursday, 10/10/24
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	PetPal Software System
Work Package	X02— Requirement Specification (2 of 7)
Assigned To	**Nithya Hariharan** & Tan Shu Hua, Samantha & Najah Ismail & Mishra Apurva & Gambhir Dhruv & Tan Jing Jie & Mehta, Viral Sujal
Effort	7PD
Start Date	Thursday, 05/09/24
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
Activities	Identify "the customer", interview customers, write and inspect customer requirements and build requirements.
Outputs	A written document of the requirement specification.

Project	PetPal Software System
Work Package	X03— User Interface (3 of 7)
Assigned To	**Najah Ismail** & Tan Shu Hua, Samantha & Mishra Apurva & Nithya Hariharan & Gambhir Dhruv & Tan Jing Jie & Mehta, Viral Sujal
Effort	7PD
Start Date	Wednesday, 18/09/24
Purpose	To build the user interface between the system and the customer, to make it easy use, and friendly to the customer
Inputs	User information
Activities	To get the user information, user request, display the dialog between system and user, display the result of request
Outputs	User Interface

Project	PetPal Software System
Work Package	X04— Technical Architecture (4 of 7)
Assigned To	**Gambhir Dhruv**& Tan Shu Hua, Samantha & Najah Ismail & Nithya Hariharan & Mishra Apurva & Tan Jing Jie & Mehta, Viral Sujal
Effort	14PD
Start Date	Thursday, 12/09/24
Purpose	To do the high level architecture design
Inputs	Project Plan Work Packages (X01 to X03 inclusive).
Activities	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.
Outputs	High Level Design and Architectural Specification.

Project	PetPal Software System
Work Package	X05— Data Modeling (5 of7)
Assigned To	**Mehta, Viral Sujal** & Tan Shu Hua, Samantha & Najah Ismail & Nithya Hariharan & Gambhir Dhruv & Tan Jing Jie & Mishra Apurva
Effort	7PD
Start Date	Thursday, 19/09/24
Purpose	To build the project's database
Inputs	Project Plan Work Packages (X01 to X05 inclusive).
Activities	Analyze the data flow relationships, entity relationships
Outputs	A written document of the data modeling

Project	PetPal Software System
Work Package	X06— Coding & Unit testing (6 of 7)
Assigned To	**Tan Shu Hua, Samantha** & Mishra Apurva & Najah Ismail & Nithya Hariharan & Gambhir Dhruv & Tan Jing Jie & Mehta, Viral Sujal
Effort	14PD
Start Date	Friday, 18/10/24
Purpose	To implement the system as per the requirements specification and other associated documents. This work package includes such additional activities as preliminary unit testing.
Inputs	Project Plan Work Package X06.
Activities	Programmers will implement the modules according to the design specifications noted in the Specification document.
Outputs	Source code and header files

Project	PetPal Software System
Work Package	X07— Integration & System Testing (7 of 7)
Assigned To	**Tan Jing Jie** & Tan Shu Hua, Samantha & Najah Ismail & Nithya Hariharan & Gambhir Dhruv & Mishra Apurva & Mehta, Viral Sujal
Effort	14PD
Start Date	Sunday 20/10/24
Purpose	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	Project Plan Work Package X07.
Activities	The Integration testing team may try to simulate how a user might interact with the system. Similar to Unit Testing, Integration Testing may require the development of stubs and drivers as well, but here this is more geared towards the higher (overall system) level. Testers may also examine issues such as system performance and integrity. Heuristics assessment plays an important role in this work package, as intelligence components will define eventual system success.
Outputs	A test report.

5 Project Estimates

5.1 Code Size Estimation using Function Points

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

5.1.1 Unadjusted Function Points

Petpal supports the following proposed functions:

Users:

- View and edit personal profile
- Register and login
- Find Sitters/Events/Adoption Listings
- Register interest for Sitters/Events/Adoption Listings
- Post a listing for Sitter/Event
- View interests for Sitter/Event listing

Agency:

- View and edit personal profile
- Register and login
- Post Adoption Listings
- Post Events

The measure of unadjusted function points is based on five primary component elements: External Inputs (EI), External Outputs (EO), External Inquiries (EQ), Internal Logical Files (ILF), and External Interface Files (EIF). Each element ranges from Low Complexity, Medium Complexity to High Complexity. The detailed evaluation is as follows:

External Inputs:

- Registration (User and Agency)

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)

- Login

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)

- Edit profile

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)

- Create sitting request

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)

- Register interest in sitting requests, adoption listings, and events

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)

- Add pets

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)

- Add event listing

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)

Total EIs:
Low Complexity: 3
Average Complexity: 4

External Outputs:

- View profile (user and agency)

Files Type Referenced (FTR)	**Data Elements**		
	1-4	5-15	Greater than 15
Less than 2	Low (4)	Low (4)	Average (5)
2	Low (4)	Average (5)	High (7)
Greater than 2	Average (5)	High (7)	High (7)

- View sitting requests

Files Type Referenced (FTR)	**Data Elements**		
	1-4	5-15	Greater than 15
Less than 2	Low (4)	Low (4)	Average (5)
2	Low (4)	Average (5)	High (7)
Greater than 2	Average (5)	High (7)	High (7)

- View adoption and event listings

Files Type Referenced (FTR)	**Data Elements**		
	1-4	5-15	Greater than 15
Less than 2	Low (4)	Low (4)	Average (5)
2	Low (4)	Average (5)	High (7)
Greater than 2	Average (5)	High (7)	High (7)

Total EOs:
Average Complexity: 3

External Enquiries:

- Search and Filter Functions for Sitting Requests, Adoption Listings, and Events

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)

Total EQs:
Average Complexity: 1

Internal Logical Files:

- User Profiles, Agency Profiles, Pets, Sitting Requests, Adoption Listings, Events, Interest Registrations

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)

Total ILFs:
Low Complexity: 1

External Logical Files:

- Map data

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

Total ELFs:
Low Complexity: 1

Calculation of Unadjusted Function Points:

Characteristic	**Low**		**Medium**		**High**	
External Inputs	3	× 3	4	× 4	0	× 6
External Outputs	0	× 4	3	× 5	0	× 7
External Inquiries	0	× 3	1	× 4	0	× 6
Internal Logical Files	1	× 7	0	× 10	0	× 15
External Interface Files	1	× 5	0	× 7	0	× 10
Unadjusted FP	21		25		0	
Total=L+M+H	46					

5.1.2 Adjusted Function Points

Influence Factors	Score	Detail
Data Communications	3	Standard internet communication.
Distributed Functions	2	Centralized processing with standard data transfer.
Performance	2	Performance is moderately important.
Heavily used	1	Anticipated negligible peak usage periods.
Transaction rate	2	Daily peak transaction period is anticipated.
On-line data entry	4	More than 30% of transactions are interactive data entry
End-user efficiency	3	User-friendly interfaces are important.
On-line data update	3	Frequent updates to internal files.
Complex processing	2	Moderate.
Reusability	1	Not important.
Installation Ease	1	Standard.
Operational Ease	1	Standard.
Multiple sites	2	User requirements do not require considering the needs of more than one user/installation site.
Facilitate change	2	Moderate.
Total score	32	
Influence Multiplier = Total score × 0.01 + 0.65 = 32 × 0.01 + 0.65 = 0.97		
Adjusted FP = Unadjusted FP × Influence Multiplier = 46 × 0.97 = 44.62		

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

5.1.3 Lines of Code

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

Therefore, we have: **Lines of Code** = 44.62 FP × 29 LOC/FP = **1293.98 LOC**

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

5.2 Efforts, Duration and Team Size Estimation

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

- Working days include 1 day (s) in a week.
- Effort = Size / Production Rate = (**1293.98** LOC) / (39 LOC/PD)[1] = 33.18 PD
- Duration = 3 × (Effort) $^{1/3}$ = 3 × (33.18) $^{1/3}$ = 9.87 Days
- Initial schedule = 9.87 Days / 1 day a week = 9.87 Weeks
- Team size = 33.18 PD / 9.87 D = 5.49 P = 3 Persons
- Working hours include 8 hours in a working day.
- Total person-hours (PH) = 33.18 PD × 8 hours = 265.44 PH

*Expectation for calculation is that 3 persons are working on the project for approximately 8 hours per week. We account that a person may be able to work 4 hours per week and hence accounting for the 6 member team. This can be shown in the team size calculator by adjusting the Lines of code per Person Day. In other words, it is 3 full time workers or 6 part time workers a week.

Effort = Size / Production Rate = (**1293.98** LOC) / (**39/2** LOC/PD)[1] = 66.36PD

Duration = 12.50

Team Size = 5.3

5.2.1 Distribution of Effort

1990's Industry Data	**Work Package**	**Distribution**	**Estimates**
Preliminary Design 18 %	Project Plan	9%	23.89
	Requirement Specification	9%	23.89
Detailed Design 31 %	User Interface	10%	26.54
	Technical Architecture	11%	29.20
	Data Modeling	10%	26.54
Code & Unit Testing 32 %	Code & Unit testing	25%	66.36
	Online Documentation	7%	18.58
Integration & Test 18 %	Integration & Quality Assurance	18%	47.78
	Extrapolated total effort		265.44 PH
	2% for project management		5.30
	3% for contingency		7.96
	Total effort		278.7

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

5.3 Cost Estimates

We have provided a detailed breakdown of the estimated monthly costs for our project in the table below. The table includes major expenses, such as employee salaries, infrastructure costs (including equipment and office rental).

For our cloud hosting platform, we are using Azure's free tier. We estimate that the free subscription would be sufficient for our initial traffic, hence no cost is estimated for cloud hosting in this current phase.

We are also using Google Firebase for authentication, which is completely free of charge on all plans, and hence no cost is estimated.

Monthly Estimated Costs of Project

Item	Supplier	Quantity	Unit Price (SGD)	Total
Project Manager	-	1	$8,000.00	**$8,000.00**
Software Developers	-	3	$5,000.00	**$15,000.00**
QA/Release Engineers	-	3	$4,000,00	**$12,000.00**
Computers	Dell	5	$1,000.00	**$5,000.00**
Computers	Apple	2	$1,400.00	**$2,800.00**
Office rental	NTU	1	$5,000.00	**$5,000.00**
Cloud Hosting	Azure	1	$0.00	**$0.00**
Firebase Authentication	Google	1	$0.00	**$0.00**
			TOTAL	**$47,800.00**

6 Product Checklist

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

Project Deliverable	**Estimated Deadline**
Meeting Minutes	Every meeting
Trello Backlog	Updated after every lab
Team Information	5th September 2024 (Lab 2)
Project Proposal	
Use Case Model & Description	
System Requirement Specification	19th September 2024 (Lab 3)
Quality Plan	
Project Plan	10th October 2024 (Lab 4)
Risk Management Plan	
Prototype	
Design Report on Software Maintainability	24th October 2024 (Lab 5)
Configuration Management Plan	
Change Management Plan	
Release Plan	
Presentation Slides	7th November 2024 (Final Submission)
Module/System Test Plan	
System Release (Demo)	

7 Best Practice Checklist

Practice	9
Document Everything All documentation must be in a standardized format to ensure consistency and ease of understanding. Source code should include detailed comments explaining functionality and logic. Commit messages must be meaningful, clearly describing the changes made.	
Pay Attention to Requirements Requirements must be checked for ambiguity, completeness, accuracy, and consistency. The requirement documentation must contain a complete functional specification to ensure a shared understanding of what needs to be built.	
Keep It Simple Complexity management is one of the major challenges. Strive to minimize interfaces between modules, procedures, and data, and efficiently use abstraction in functions and modules. Avoid fancy product functions; design should meet customer requirements without unnecessary features.	
Require Visibility It is essential to see what is being built to measure progress and take management action. Good communication among team members and management is crucial. Developers are required to make their code available for review, and designs should be reviewed for appropriateness.	
Plan for Continuous Change All manuals, designs, tests, and source code should have revision numbers and dates, with revision history comments and change marks to indicate changes. New revisions should be approved before being made and checked for quality and compliance after being made. Use a configuration management system and establish processes for handling changes systematically. Maintenance requirements must be acknowledged and planned for accordingly.	
Do Not Underestimate Time or Effort Needed for a Feature Care must be taken to obtain accurate estimates for time, effort, overhead, meeting time, and especially effort on integration, testing, documentation, and maintenance. Accurate estimation helps avoid unexpected delays and ensures that the project stays on track.	
Code Reviews and Testing Code reviews are a much more efficient method to find software defects. Plan and manage code reviews between team members; all code must be reviewed before being committed to the main branch.	
Software testing Both black box and white box testing, involving unit, functional, integration, and acceptance testing will be used. Follow a test-first approach where tests are written before or along with all the source code, not at the end. All code must be merged on the feature branch and tested before a pull request to the main branch.	

8 Risk Management

Besides the general risk management, the following risks have been identified for the PetPal project:

Data breaches or security vulnerabilities
Impact Severity: High
Probability: 20%
Impacts: If security vulnerabilities or data breaches occur, sensitive user data could be exposed, leading to loss of trust, legal consequences, and potential financial losses. The reliability of third-party services like Firebase or Azure could also be affected, leading to significant outages.
Risk Reduction: Continuously monitor the third-party services and implement regular security audits to identify and fix vulnerabilities before they are exploited.

Inability to meet project deadlines
Impact Severity: High
Probability: 25%
Impacts: Missing deadlines could result in project delays, increased costs, and inability to meet client expectations, which could harm the project's overall success.
Risk Reduction: Implement a robust project schedule with clear milestones and track progress rigorously to ensure timely delivery.

Unavailability of key project members
Impact Severity: Medium
Probability: 10%
Impacts: The departure of key team members could lead to a loss of critical knowledge and expertise, causing disruptions and delays in project progress.
Risk Reduction: Maintain comprehensive documentation and ensure knowledge sharing among the team to mitigate the impact of any unexpected departures.

Minor bugs
Impact Severity: Medium
Probability: 40%
Impacts: Minor bugs can lead to usability issues and customer dissatisfaction if left unresolved, but they are unlikely to significantly impact the overall project schedule or cost.
Risk Reduction: Regular code reviews and testing should be conducted to identify and fix bugs early in the development process

Feature Creep
Impact Severity: High
Probability: 60%
Impacts: Will delay the project and may not be able to meet the set deadlines.
Risk Reduction: Maintain traceability between documentation and code to not go beyond the specified requirements.

Inconsistent code design
Impact Severity: Medium
Probability: 40%
Impacts: Modules cannot be separated
Risk Reduction: Design a system architecture and class diagram and ensure that it is followed when coding the application.

Insufficient Testing
Impact Severity: Medium
Probability: 50%
Impacts: Could lead to undiscovered bugs and failures, resulting in poor product quality and increased post-deployment issues.
Risk Reduction: Establish a robust testing strategy with automated and manual tests, ensuring comprehensive coverage of key features.

Increased Server Load from Higher User Base or Malicious Attack
Impact Severity: Medium
Probability: 50%
Impacts: Could lead to degraded performance, downtime, or unavailability of services, causing user dissatisfaction and potential revenue loss.
Risk Reduction: Implement scalable infrastructure, load balancing, and security measures to manage and mitigate unexpected spikes in server traffic.

Limited Scalability Under High Traffic Conditions
Impact Severity: Low
Probability: 30%
Impacts: System performance may degrade under high traffic, leading to slower response times, bottlenecks, and potential system crashes, impacting user experience and growth potential.
Risk Reduction: Design a scalable architecture with horizontal scaling, use cloud-based services, and optimize resource allocation to handle increased traffic efficiently.

9 Quality Assurance

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

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

In addition, the PetPal shall make use of the following testing methodologies:

- **Unit Testing**: Each individual component or function of the system will be tested in isolation to verify its correct operation. Unit tests will be executed during the development phase to ensure that the smallest testable parts of the application meet specified requirements and function as intended.
- **Integration Testing**: After individual components are verified, integration testing will be conducted to assess the interaction between different modules. This testing will ensure that components work together seamlessly, identifying and resolving issues related to the interfaces or data flow between units.
- **System Testing**: This involves testing the entire system as a whole to ensure all integrated components function together as expected. The goal is to validate the system against its requirements and ensure that it meets functional and non-functional specifications, such as performance and security.
- **User Acceptance Testing (UAT)**: This phase ensures the software meets the needs and expectations of the end users (pet agencies, adopters, pet owners, and sitters). It involves real-world scenarios and tasks to verify that the system behaves correctly in a production-like environment before the final release.
- **Performance Testing**: This involves assessing the system's performance under various conditions, such as heavy loads or extended usage periods. The goal is to identify any bottlenecks or issues that could affect user experience, particularly during high-traffic periods on the platform.

Furthermore, these methodologies will be used to test two important aspects of the PetPal:

- **System Function** will be tested to ensure that software flaws are eliminated, and
- **Algorithmic Function** will be tested to ensure that heuristic aspects of the project (such as user preference rankings) perform realistically to provide value to the users.

PetPal's methodology makes broad use of realistic test cases. Detailed test data is an important part of the final project delivery. The PetPal's development team will validate code and heuristic result ranking technology using realistic scenarios. In addition, extreme cases will be used to ensure that the system behaves correctly in degenerate cases.

The Quality Assurance Manager (QAM) and Software Quality personnel will also be responsible for assessing and maintaining software quality, ensuring that products and processes meet IEEE and ISO standards.

10 Monitoring & Control

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

Regular reviews of project progress:
The PetPal team meets bi-weekly to review the progress of all project tasks, including management, planning, analysis, development, and testing.

Identification of major project risks:
Through regular meetings and thorough discussions, possible major risks and issues are identified and classified based on their criticalness. Preventive measures are put into place to alleviate them. An assessment of risk can be found in the Risk management document.

Timeline Planning and task decomposition:
To create an accurate timeline, tasks are broken down into smaller, manageable divisions, and each part is assessed for its requirements. This breakdown helps with assigning tasks and balancing the workload. During the implementation phase, these smaller parts also make it easier to track progress in detail. The timeline and task breakdown can be found in the Estimates and Work Breakdown Structure sections of this document.