Petpal

One Stop solution for all your pet needs

Group Petpal

SC3040 ADVANCED SOFTWARE ENGINEERING TEL1

Team Members

Name	Matric Number	Role
Mishra Apurva	U2120474C	Project Manager
Tan Shu Hua, Samantha	U2021180J	QA Manager
Tan Jing Jie	U2221344L	QA Engineer
Najah Ismail	U2120555F	Release Manager
Gambhir Dhruv	U2120075F	Lead Developer
Nithya Hariharan	U2123872G	Frontend Developer
Mehta Viral Sujal	U2123491B	Backend Developer

TABLE OF CONTENTS

01

Product Introduction

Problem Statement

FRAGMENTED ADOPTION PROCESS

Adoption process in Singapore is scattered across multiple platforms, causing inefficiencies and delays for potential adopters.

STRUGGLE TO FIND COMMUNITY AND RESOURCES

Pet owners and adopters face challenges in connecting with reliable resources and fellow pet lovers such as at pet **events**.

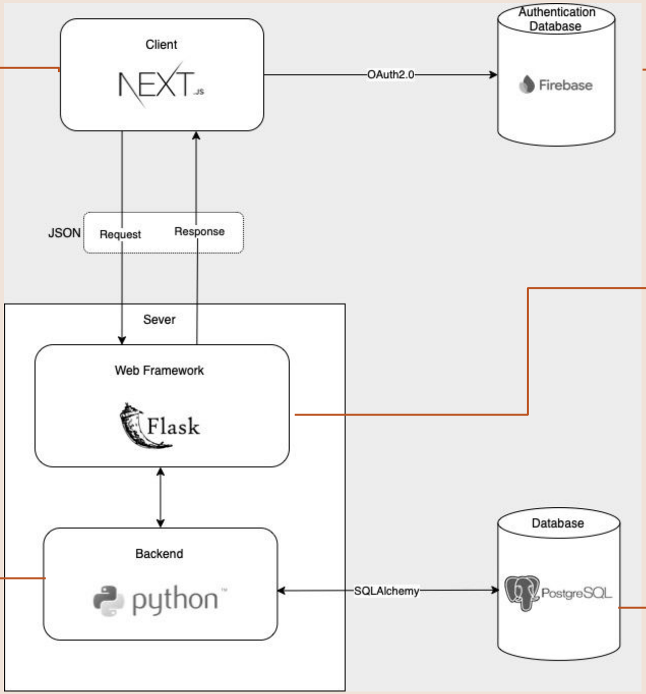
INCREASE IN DEMAND FOR RELIABLE SERVICE

As pet ownership rises, the need for dependable services like **pet sitting** has surged, with many owners struggling to find suitable sitters.

Architecture

NextJS
FrontEnd

Python
BackEnd



Firebase
Authentication

Flask
API

Postgres
Database

Backend

FLASK

Flask acts as the web framework for routing and processing incoming JSON requests from the front-end (Next.js) and sending the corresponding responses.

ORM

SQLAlchemy is used to interact with the PostgreSQL database.

It provides a high-level, Pythonic way of writing database queries and handling database operations without having to write raw SQL.

PostgreSQL

Open-source RDBMS which supports Atomicity, Consistency, Isolation, Durability (ACID)

9 tables including users, pets, adoption, events requests...

Authentication

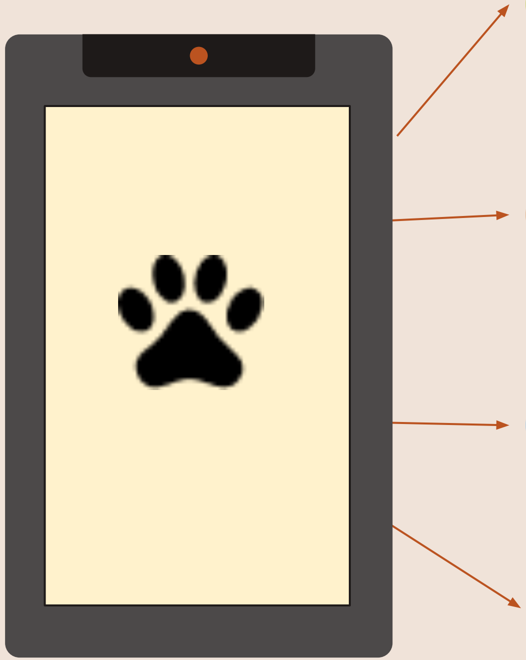
Firebase

Firebase for authentication by leveraging OAuth 2.0 to securely manage user authentication.

Efficiently handles token generation, validation, and user session management, simplifying the integration of authentication.

Additionally, it supports advanced features such as multi-factor authentication and real-time user account management.

Main Functions



Events

The system stores and hosts pet events, allows users to register, and provides a search feature based on event theme and location.

Adoption

The system manages pets for adoption, lists them by criteria, tracks user interest, and lets agencies respond to requests or remove pets anytime.

Sitters

The system manages pet sitting services, allows sitters to register with wage and location preferences, enables user messaging, and lets sitters remove their services anytime.

Interests

Consolidated dashboard to view user activity across all functions.

Demo

02

Design for Maintainability

Maintainability Design

Design Pattern

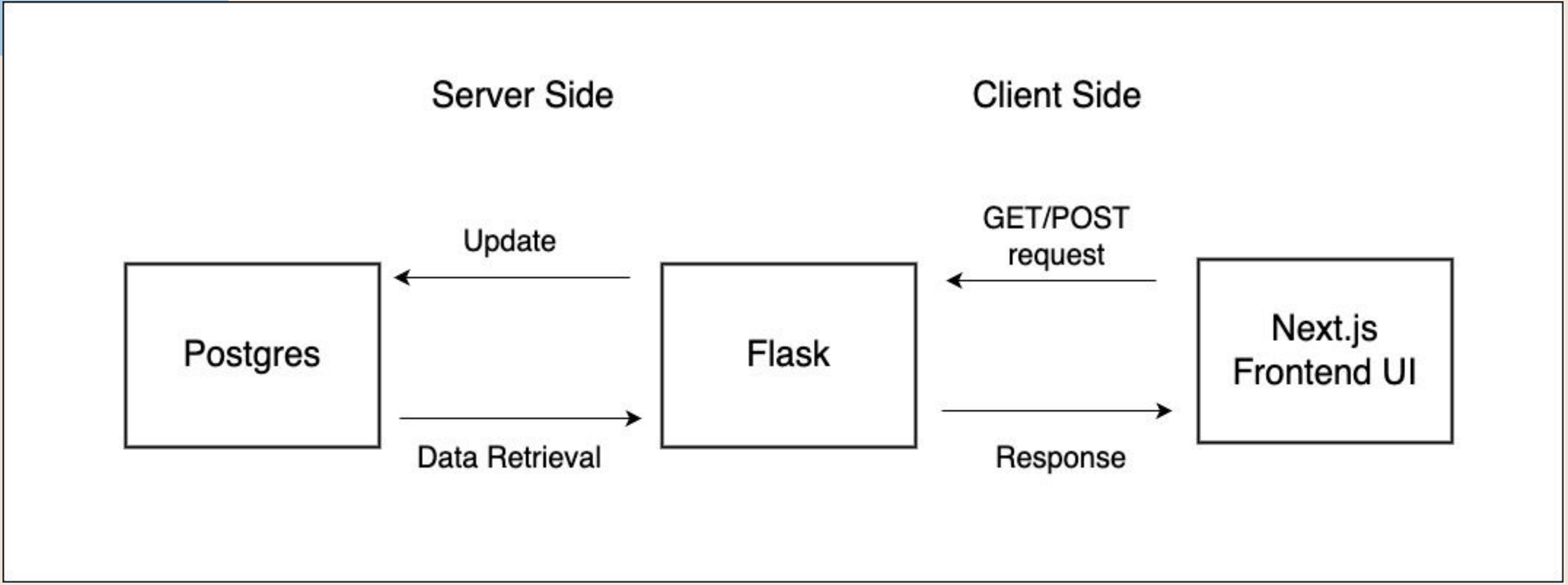
Design Ideas

Configuration Management

Standardization

Maintainability Design

Design Pattern: Client-server architecture



Maintainability Design

Design Ideas

- Design principles
  - low coupling, high cohesion
  - Well-defined user interface

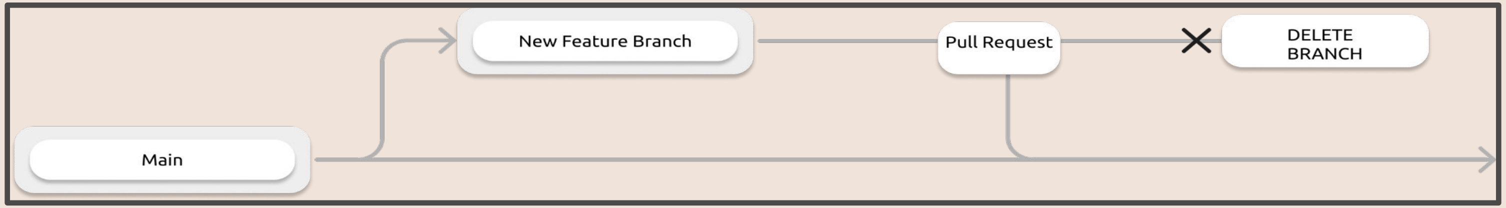
- Flask web framework: abstraction saves time

Maintainability Design

Configuration Management

Code Version Control

- **Feature branches** are used for development.
- Each feature branch is **descriptively** named (e.g., "addDatabase") to track changes associated with specific development tasks.
- After **merging** the developed feature with main branch, the feature branch is **deleted**



Documentation

- OneDrive
- MediaWiki

Maintainability Design

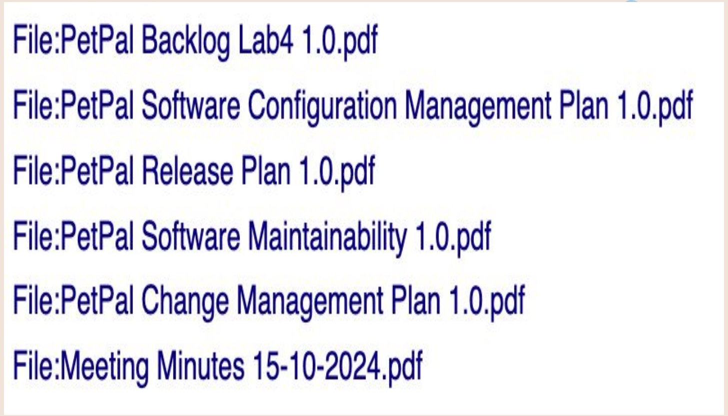
Standardization

Document Identification:
All project documents follow naming convention <document type>_<document number>.<revision index> ensuring clear version control and traceability.

Branches Named after feature

Adapt conventional commits: “feat” and “fix” tags

Version Numbering System:
Petpal uses a standardized versioning format: XXX_Vm.n, where major and minor version numbers clearly distinguish between significant releases and minor updates.


File:PetPal Backlog Lab4 1.0.pdf
File:PetPal Software Configuration Management Plan 1.0.pdf
File:PetPal Release Plan 1.0.pdf
File:PetPal Software Maintainability 1.0.pdf
File:PetPal Change Management Plan 1.0.pdf
File:Meeting Minutes 15-10-2024.pdf

03

Software Quality Assurance

Software Quality Assurance

Technical Review

Objective:
Ensure product functionality aligns with requirements and design.

Key Focus
- Code Reviews (functionality, maintainability)
- Design Reviews (architecture, module structure)
- Test Plan Reviews (coverage, test cases)

Outcome: Identify discrepancies, improve code quality, and validate design.

Management Review

Objective:
Evaluate project progress, resource allocation, and adherence to timelines.

Key Focus:
- Schedule adherence
- Budget tracking
- Risk assessment and mitigation

Outcome: Ensure project alignment with business objectives and resolve management-level concerns.

Audit

Objective: Independent assessment of compliance with standards and regulation (based on IEEE, ISO).

Audit Areas:
- Documentation accuracy (specifications, reports)
- Process adherence (change management, testing)
- Product integrity

Outcome: Ensure quality processes are followed

Software Quality Assurance

Tools for tracking

- Git and GitHub for version control
- Trello for task and assignee tracking
- Zoom for reviews

Best Practices

- Testing
  - Unit testing for each feature
  - Integration testing
  - System testing
- Continuous integration
- Assess product at each stage against the SRS
- Updates to design
  - Documentation update
  - Inform whole team
- All team members take responsibility for quality assurance

QA: Risk Management

Management & Review

- **Bi-monthly reviews** of identified risks, with corrective actions tracked.

- **Regular feedback loops** during meetings including technical reviews to refine processes and improve software quality over time.

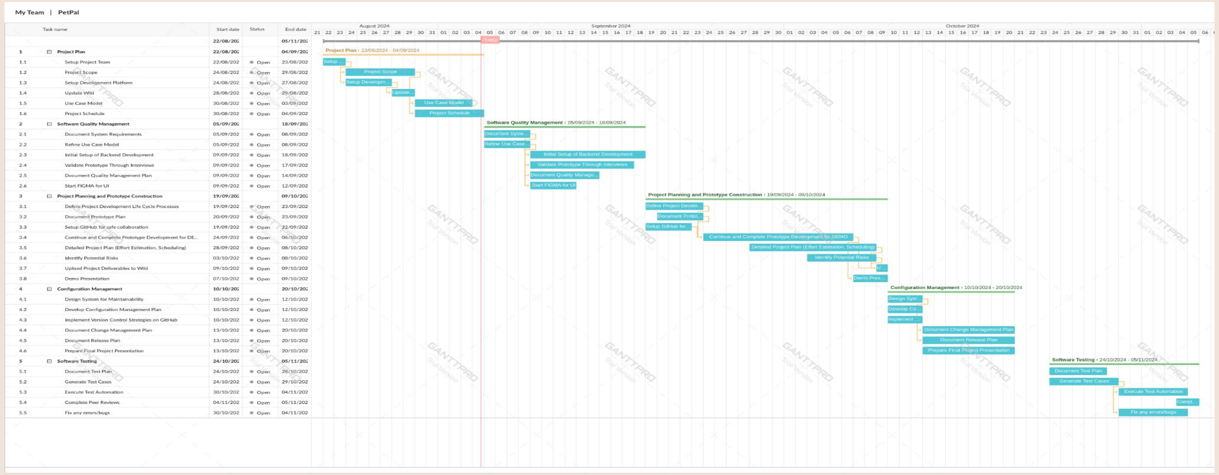
04

Project Management

Product Management

Process and Schedule

- **Agile** Development Model
- Increased **flexibility** with 2 week sprints.
- **Continuous delivery** of functional components



August 2024
September 2024
October 2024
Today
Project Plan › 22/08/2024 - 04/09/2024
Software Quality Management › 05/09/2024 - 18/09/2024
Project Planning and Prototype Construction › 19/09/2024 - 09/10/2024
Configuration Management › 10/10/2024 - 20/10/2024
Software Testing › 24/10/2024 - 05/11/2024
GANTTPRO Trial Version

My Team | PetPal

		Task name	Start date	Status	End date
			22/08/202		05/11/202
1	⊟	**Project Plan**	22/08/202		04/09/202
1.1		Setup Project Team	22/08/202	● Open	23/08/202
1.2		Project Scope	24/08/202	● Open	29/08/202
1.3		Setup Development Platform	24/08/202	● Open	27/08/202
1.4		Update Wiki	28/08/202	● Open	29/08/202
1.5		Use Case Model	30/08/202	● Open	03/09/202
1.6		Project Schedule	30/08/202	● Open	04/09/202
2	⊟	**Software Quality Management**	05/09/202		18/09/202
2.1		Document System Requirements	05/09/202	● Open	08/09/202
2.2		Refine Use Case Model	05/09/202	● Open	08/09/202
2.3		Initial Setup of Backend Development	09/09/202	● Open	18/09/202
2.4		Validate Prototype Through Interviews	09/09/202	● Open	17/09/202
2.5		Document Quality Management Plan	09/09/202	● Open	14/09/202
2.6		Start FIGMA for UI	09/09/202	● Open	12/09/202
3	⊟	**Project Planning and Prototype Construction**	19/09/202		09/10/202
3.1		Define Project Development Life Cycle Processes	19/09/202	● Open	23/09/202
3.2		Document Prototype Plan	20/09/202	● Open	23/09/202
3.3		Setup GitHub for safe collaboration	19/09/202	● Open	22/09/202
3.4		Continue and Complete Prototype Development for DE...	24/09/202	● Open	06/10/202
3.5		Detailed Project Plan (Effort Estimation, Scheduling)	28/09/202	● Open	08/10/202
3.6		Identify Potential Risks	03/10/202	● Open	08/10/202
3.7		Upload Project Deliverables to Wiki	09/10/202	● Open	09/10/202
3.8		Demo Presentation	07/10/202	● Open	09/10/202
4	⊟	**Configuration Management**	10/10/202		20/10/202
4.1		Design System for Maintainability	10/10/202	● Open	12/10/202
4.2		Develop Configuration Management Plan	10/10/202	● Open	12/10/202
4.3		Implement Version Control Strategies on GitHub	10/10/202	● Open	12/10/202
4.4		Document Change Management Plan	13/10/202	● Open	20/10/202
4.5		Document Release Plan	13/10/202	● Open	20/10/202
4.6		Prepare Final Project Presentation	13/10/202	● Open	20/10/202
5	⊟	**Software Testing**	24/10/202		05/11/202
5.1		Document Test Plan	24/10/202	● Open	28/10/202
5.2		Generate Test Cases	24/10/202	● Open	29/10/202
5.3		Execute Test Automation	30/10/202	● Open	04/11/202
5.4		Complete Peer Reviews	04/11/202	● Open	05/11/202
5.5		Fix any errors/bugs	30/10/202	● Open	04/11/202

Product Management

Effort Estimation

- **Lines of Code** based on function points
  - = 44.62 FP × 29 LOC/FP = 1293.98 LOC

- **Effort:**

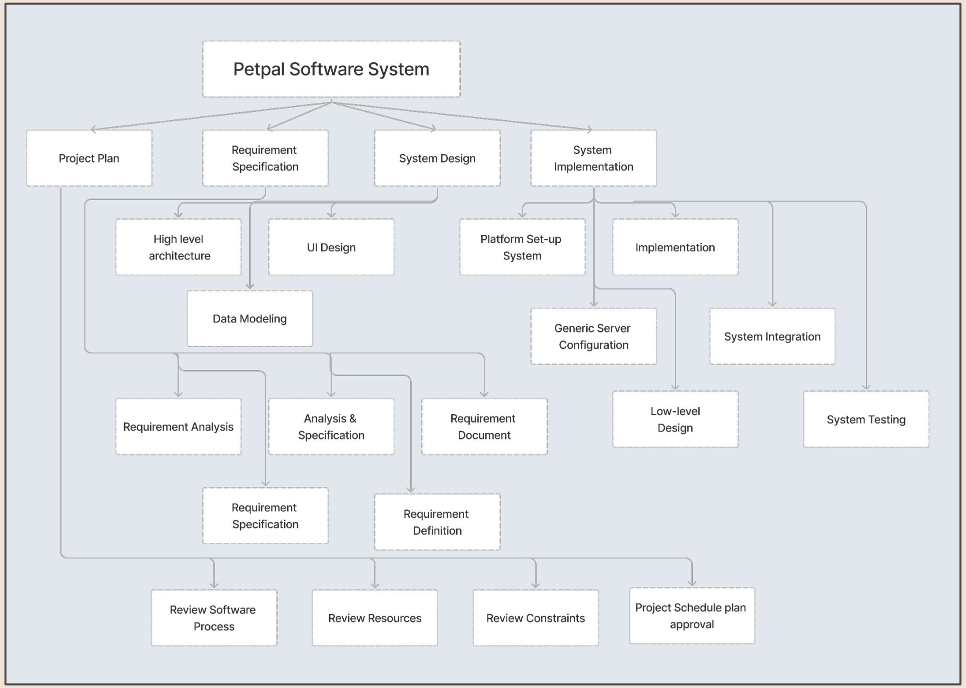
We account that a person may be able to work **4 hours per week** and hence accounting for the 7 member team.

Duration = 8.50

Team Size = 7.3

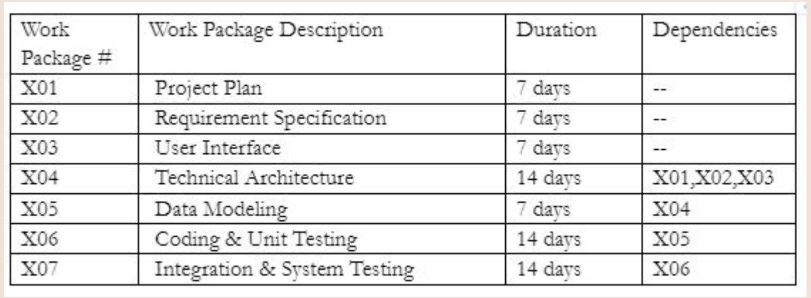
Product Management

Project Organisation

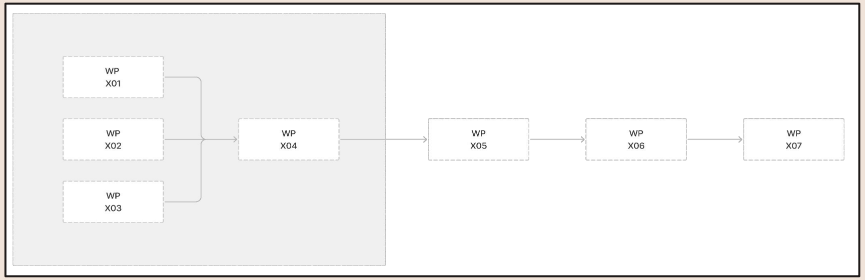


Product Management

Dependency Resolution



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

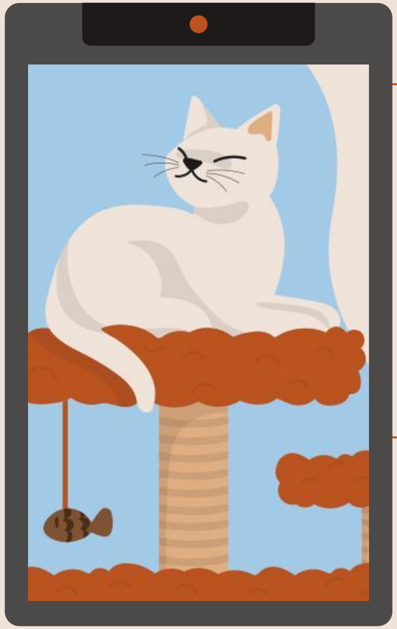


- **Resource & Capacity Planning to avoid dependency conflicts.**

05

Risk Management

Risk Management



Purpose and Scope

•Defined process for identifying, assessing, responding to, and monitoring risks.
•Applies to all project phases, including planning, development, and deployment.
•Focuses on critical risks in areas like **project schedule, resource allocation, technical functionality, and security**.

Tasks

•QA Manager leads the risk management process.
•Risk identification includes **an evaluation of environmental factors, project scope, deliverables, assumptions**, and **constraints**. A **Risk Log** is created and updated in the project library.

Risk Analysis

A risk matrix is used to define and categorize risks.

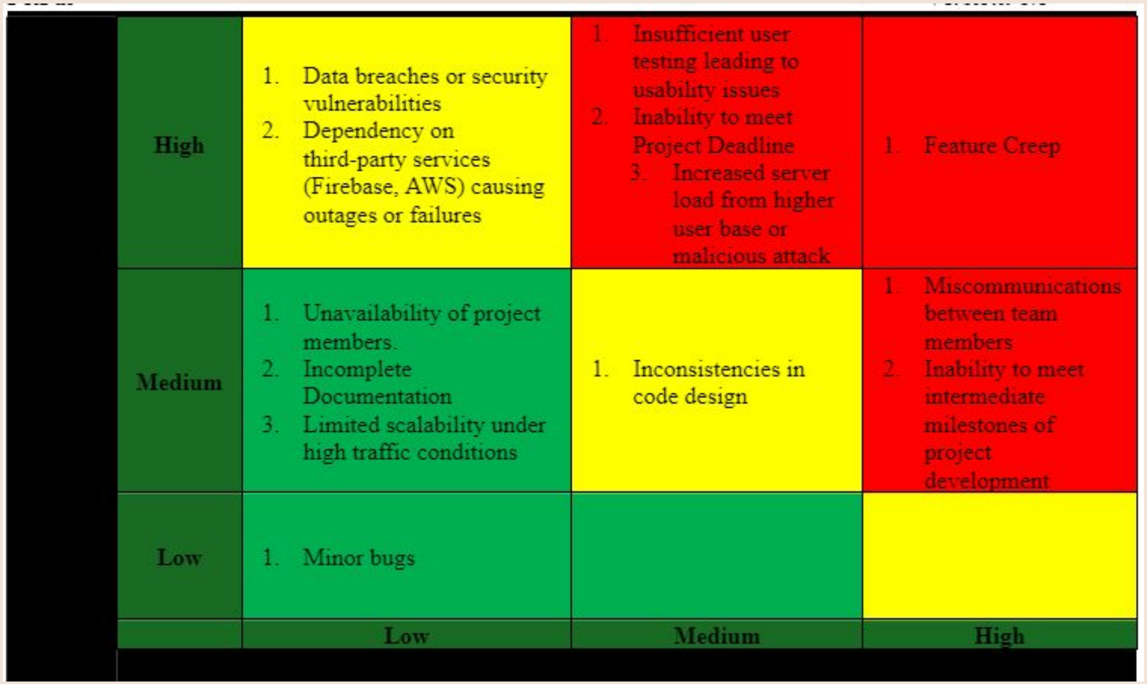
Qualitative Analysis

- Categorized by **probability** (High > 70%, Medium 30-70%, Low < 30%) and **impact** (High, Medium, Low) on project cost, schedule, or performance.
- Risks in the **Red and Yellow zones** of the Impact-Probability Matrix will require **mitigation or contingency planning**.

Quantitative Analysis

- Prioritized risks from the qualitative analysis are further analyzed to assign **numerical** ratings.
- These ratings help estimate the potential impact of each risk on project timelines and resource allocation.
- Quantitative results are documented and used to refine risk responses.

Risk Matrix



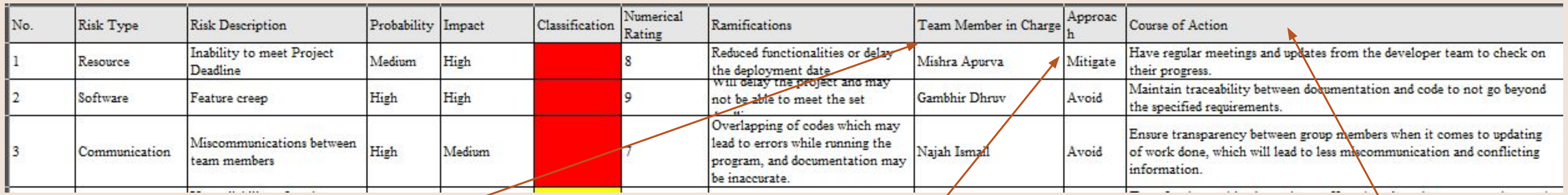
High	1. Data breaches or security vulnerabilities 2. Dependency on third-party services (Firebase, AWS) causing outages or failures	1. Insufficient user testing leading to usability issues 2. Inability to meet Project Deadline 3. Increased server load from higher user base or malicious attack	1. Feature Creep
Medium	1. Unavailability of project members. 2. Incomplete Documentation 3. Limited scalability under high traffic conditions	1. Inconsistencies in code design	1. Miscommunications between team members 2. Inability to meet intermediate milestones of project development
Low	1. Minor bugs		
	Low	Medium	High

- **High-impact risks** include
  - data breaches,
  - third-party dependencies,
  - feature creep,
  - missed deadlines,

- **lower-impact risks** may involve
  - minor bugs
  - documentation gaps.

Risk Planning

Risk Management Log is used to document Risk Planning



No.	Risk Type	Risk Description	Probability	Impact	Classification	Numerical Rating	Ramifications	Team Member in Charge	Approac h	Course of Action
1	Resource	Inability to meet Project Deadline	Medium	High	🟥	8	Reduced functionalities or delay the deployment date	Mishra Apurva	Mitigate	Have regular meetings and updates from the developer team to check on their progress.
2	Software	Feature creep	High	High	🟥	9	Will delay the project and may not be able to meet the set	Gambhir Dhruv	Avoid	Maintain traceability between documentation and code to not go beyond the specified requirements.
3	Communication	Miscommunications between team members	High	Medium	🟥	7	Overlapping of codes which may lead to errors while running the program, and documentation may be inaccurate.	Najah Ismail	Avoid	Ensure transparency between group members when it comes to updating of work done, which will lead to less miscommunication and conflicting information.

Member in Charge

Each major risk is assigned to a project team member for monitoring to ensure accountability.

Response Approach

Response strategies include: **Avoid** (eliminate the cause), **Mitigate** (reduce probability or impact), **Accept** (acknowledge risk without action), or **Transfer** (delegate responsibility, e.g., insurance).

Course of Action

A **detailed course of action** is created in case the risk materializes.

Risk Management

Monitoring & Reporting

- Risks are **tracked** and **updated** throughout the project lifecycle.
- A **"Top 10 Risk List"** is maintained and regularly reviewed during team meetings as part of the project status reporting.
- Shared with the team and management through status reports and updates.

•

Tools & Logging

- A **Risk Log,** maintained by the QA Manager, is a central tool for documenting identified risks and their status.
- The log is **reviewed** as part of **project team meetings** to ensure continuous monitoring.

CONCLUSION

PetPal

Implemented 1-stop platform:

1. Adoption
2. Events
3. Pet Sitting
4. User interests

Team Learning

- Tech stack
  - Next.js
  - Flask
  - Docker
  - Postgres
- Software best practices
  - Development
  - Documentation

THANK YOU