CZ3002 Advanced Software Engineering

FOODIFY

Contribute towards zero waste and sustainability by consuming your food articles before expiry

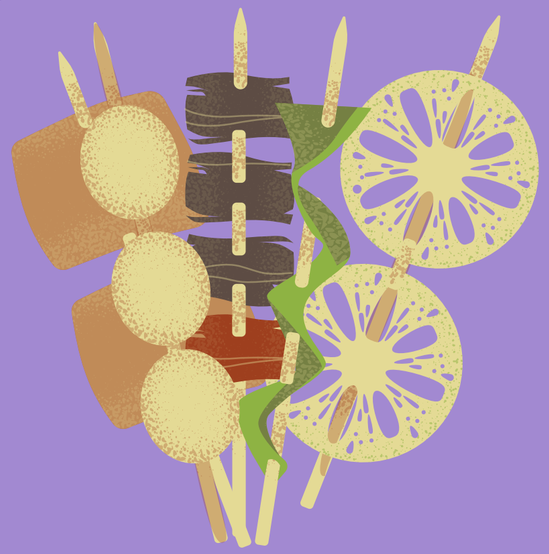
Our Team

Duong Ngoc Yen	U2020271G
Kolady Anamika Martin	U2023784E
Jim Sean	U2021239A
Verma Nandini	U2023494A
Anandarajan Sindini	U2021988A

FOODIFY

PRESENTATION OUTLINE

- Product Introduction
- Design for Maintainability
- Software Quality Assurance
- Project Management
- Risk Management



PRODUCT INTRODUCTION

FOODIFY

HEY, THIS IS CHLOE !

There are some problems she is facing:

- Last week she bought a box of 8 apples. She is unable to finish 3 of them.
- She forgot when they expire.
- Struggles to find someone to whom she can sell these apples.
- Looking for some exciting recipes to prepare using the extra apples.

WHAT IS FOODIFY

Web Application to Reduce Food Wastage

Sell Excess Food at Discounted Rates

Explore Exciting Recipes

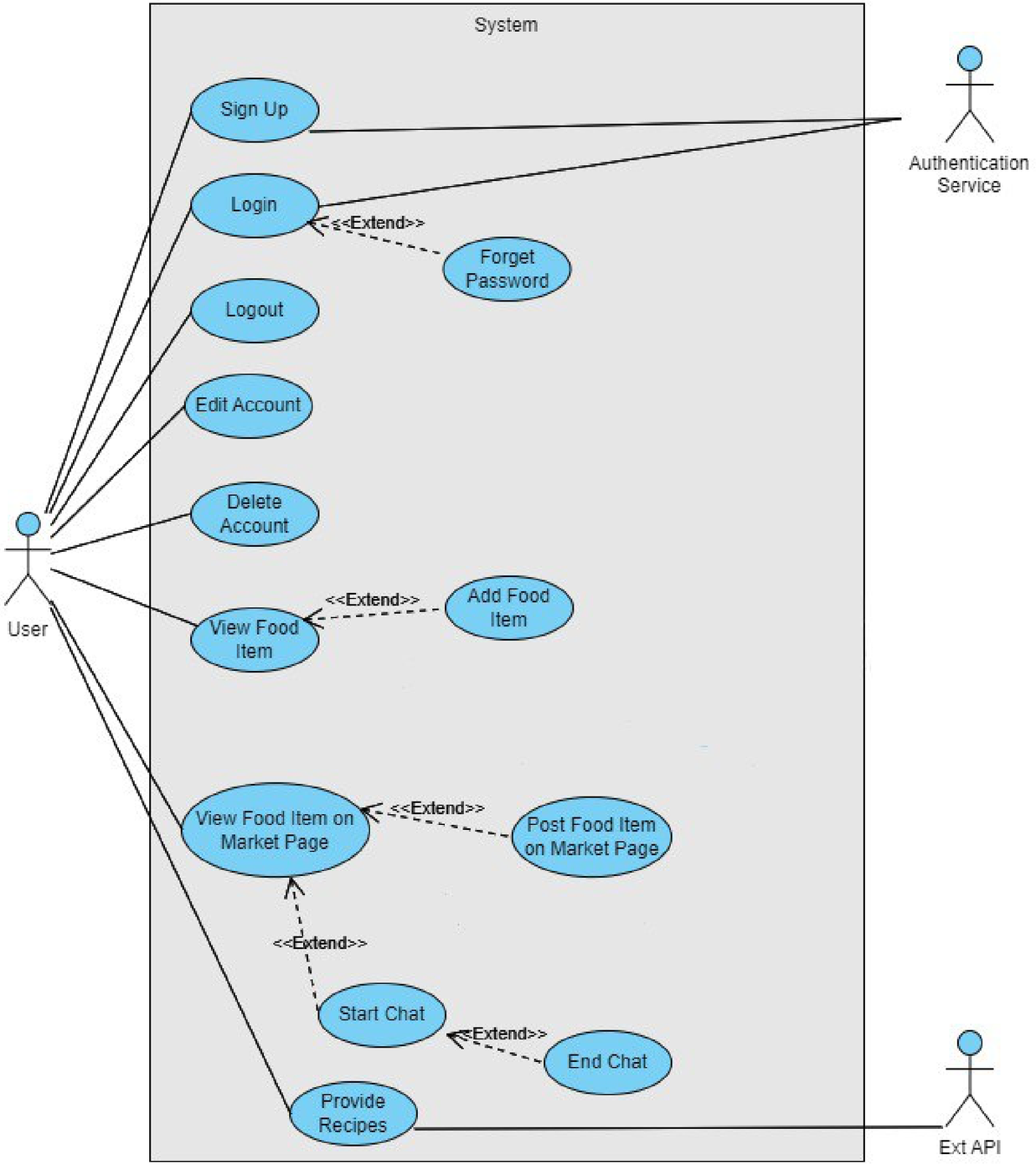
Target Users in Singapore

Individuals who are **not able to keep track of the expiry** of their foods.

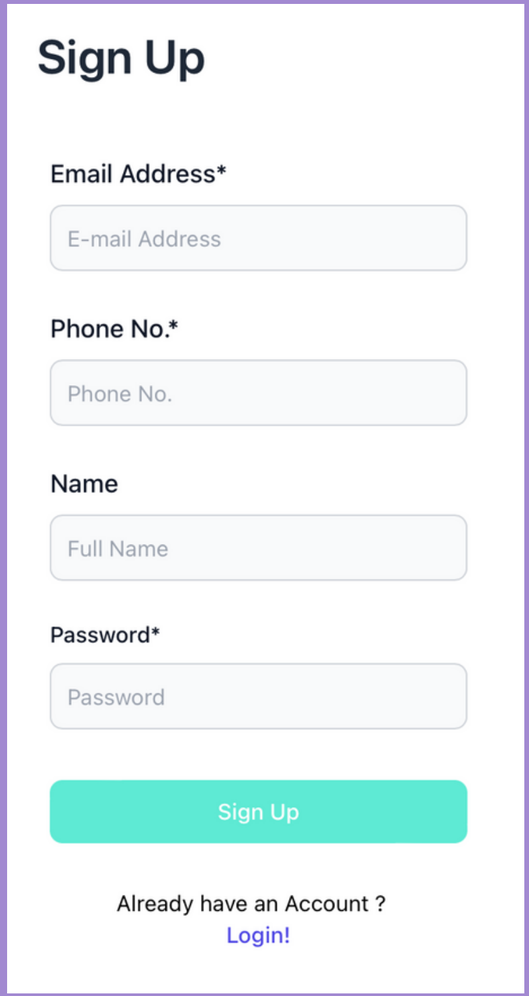
Individuals who want to **sell excess food to reduce wastage**

Individuals who want to **buy food at discounted rates**

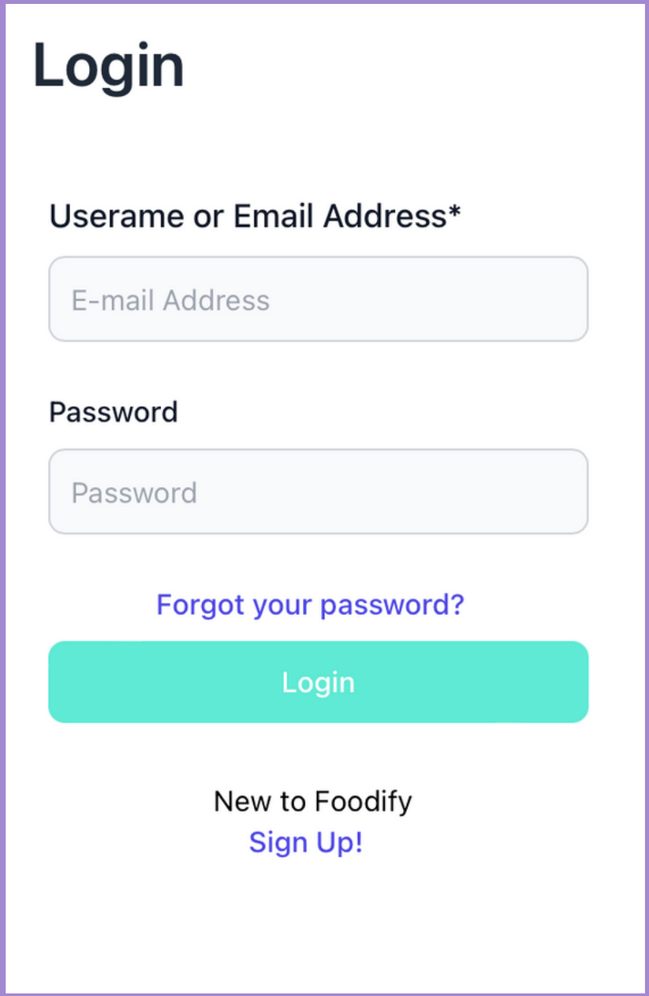
Use Case Diagram



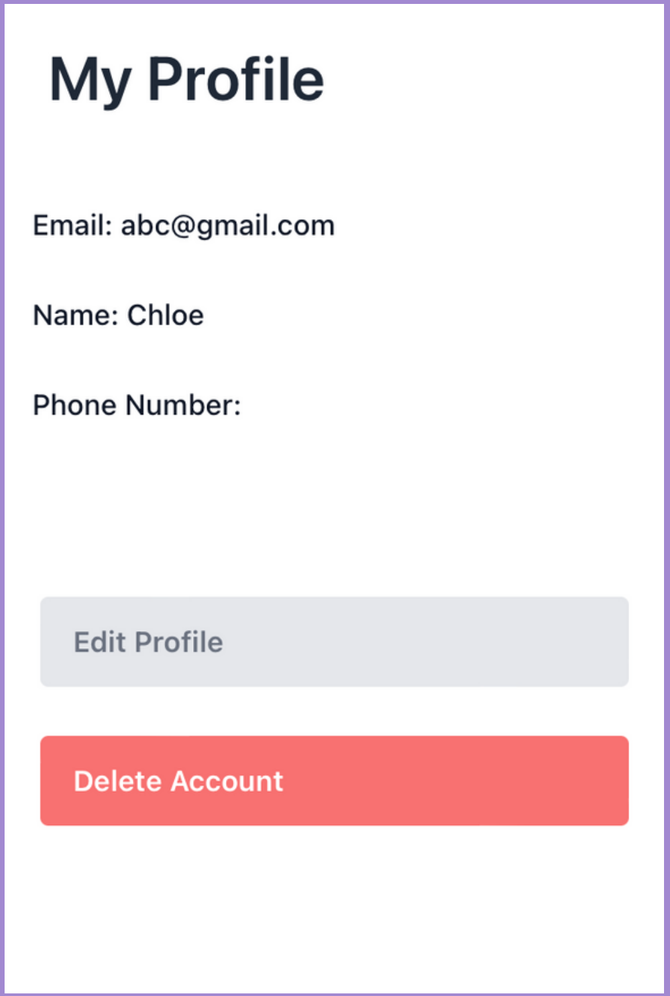
User Registration, Login and Profile



Sign Up

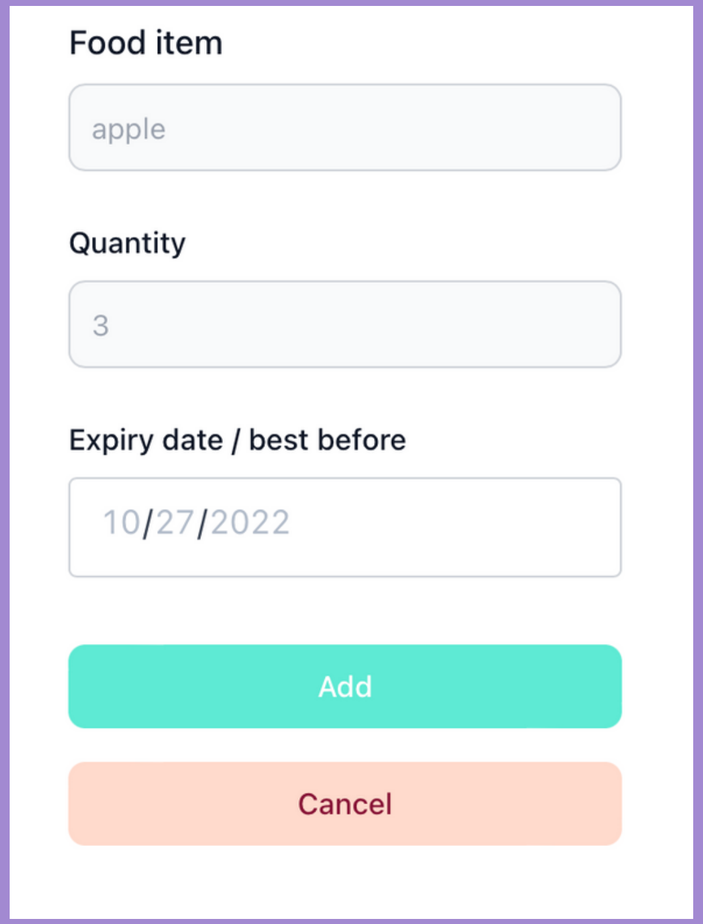


Login

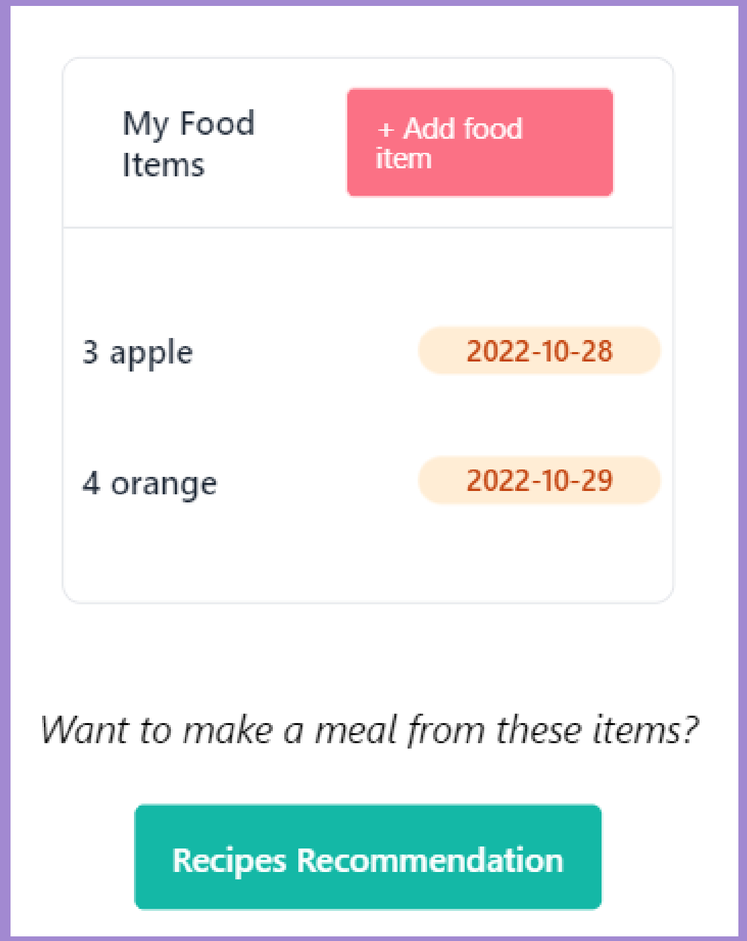


Profile

Add and View Food Items

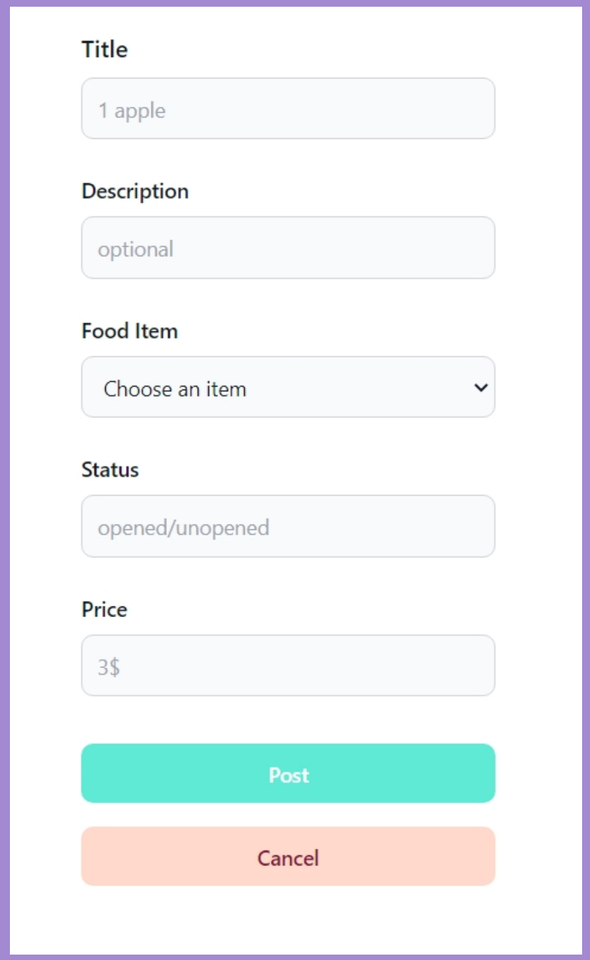


Add Food Item

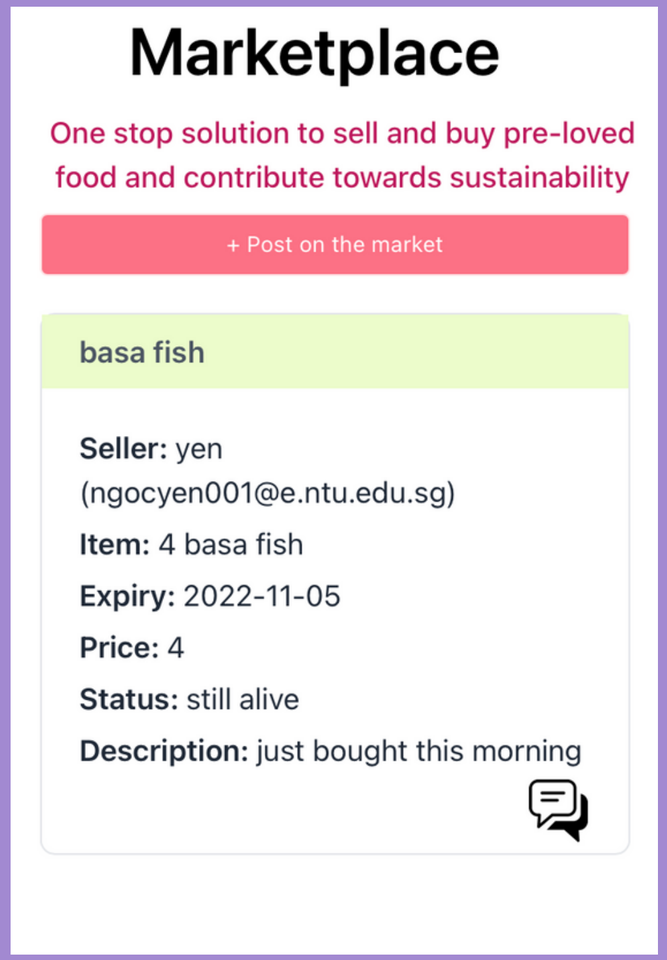


View Food Items

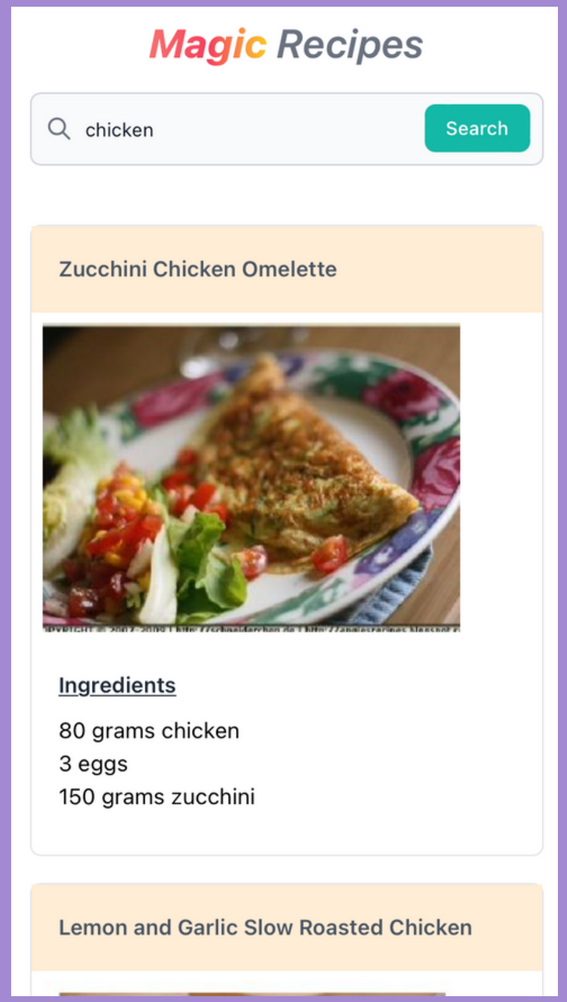
Sell Food Items, Marketplace, Recipes



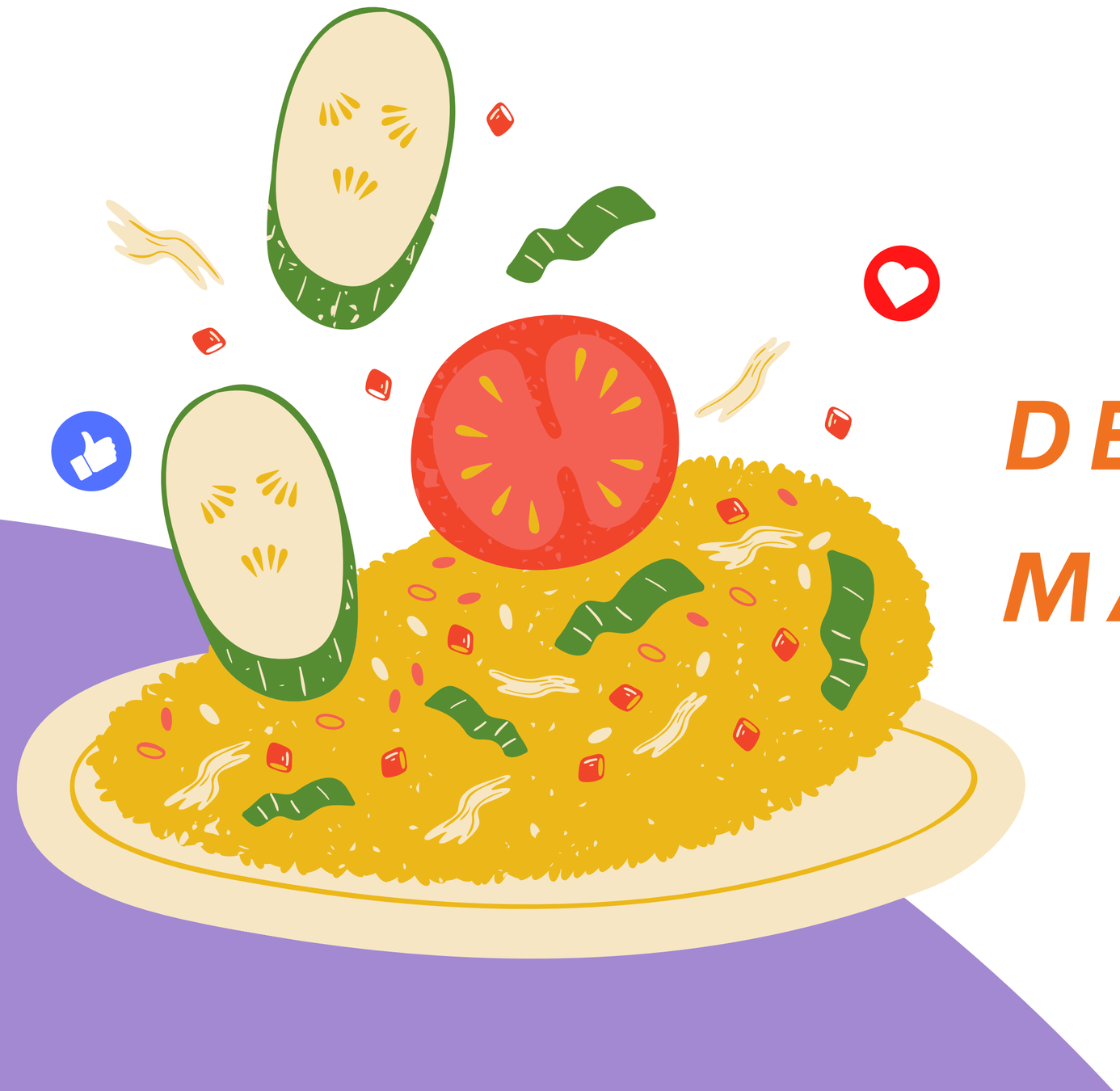
Post Food Item



Marketplace



Explore Recipes



DESIGN FOR MAINTAINABILITY

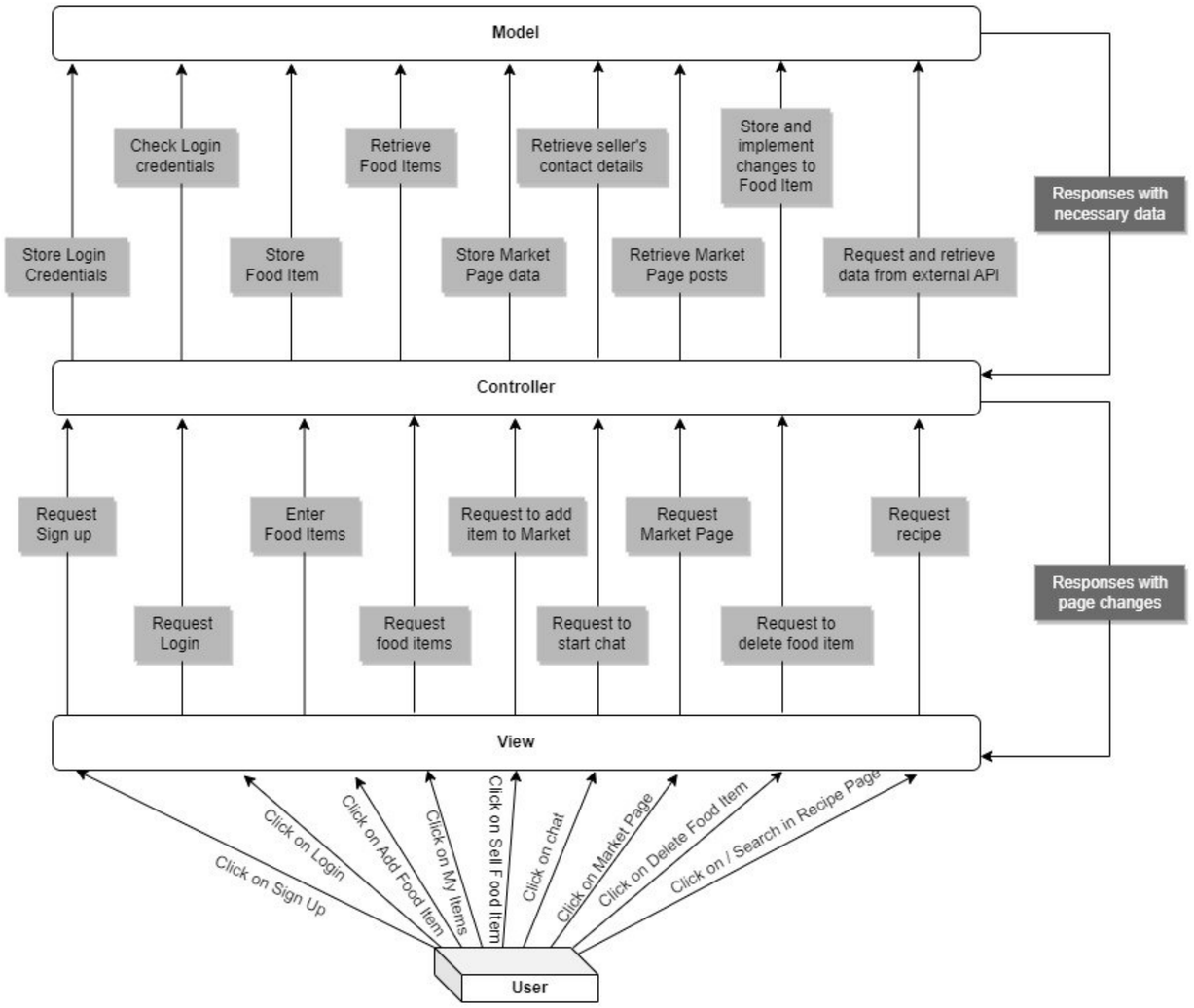
System Design Considerations

- Easy maintainability of the application
- Able to support new features or changes made in existing functionalities
- In the future, needs to support a large number of users

Model-View-Controller

Separates application into three components:

- **Model**
- **View**
- **Controller**



Why did we choose this architecture?

1 Easily Modifiable

- Adding/updating any features in a certain section does not affect the entire architecture.
- Increases flexibility and scalability

Why did we choose this architecture?

2 Easy planning and maintainability

- Helpful during the initial planning phase as it gives developers an outline of how to arrange their ideas into actual code

- Different components of the application can be independently maintained - High Modularity

Database

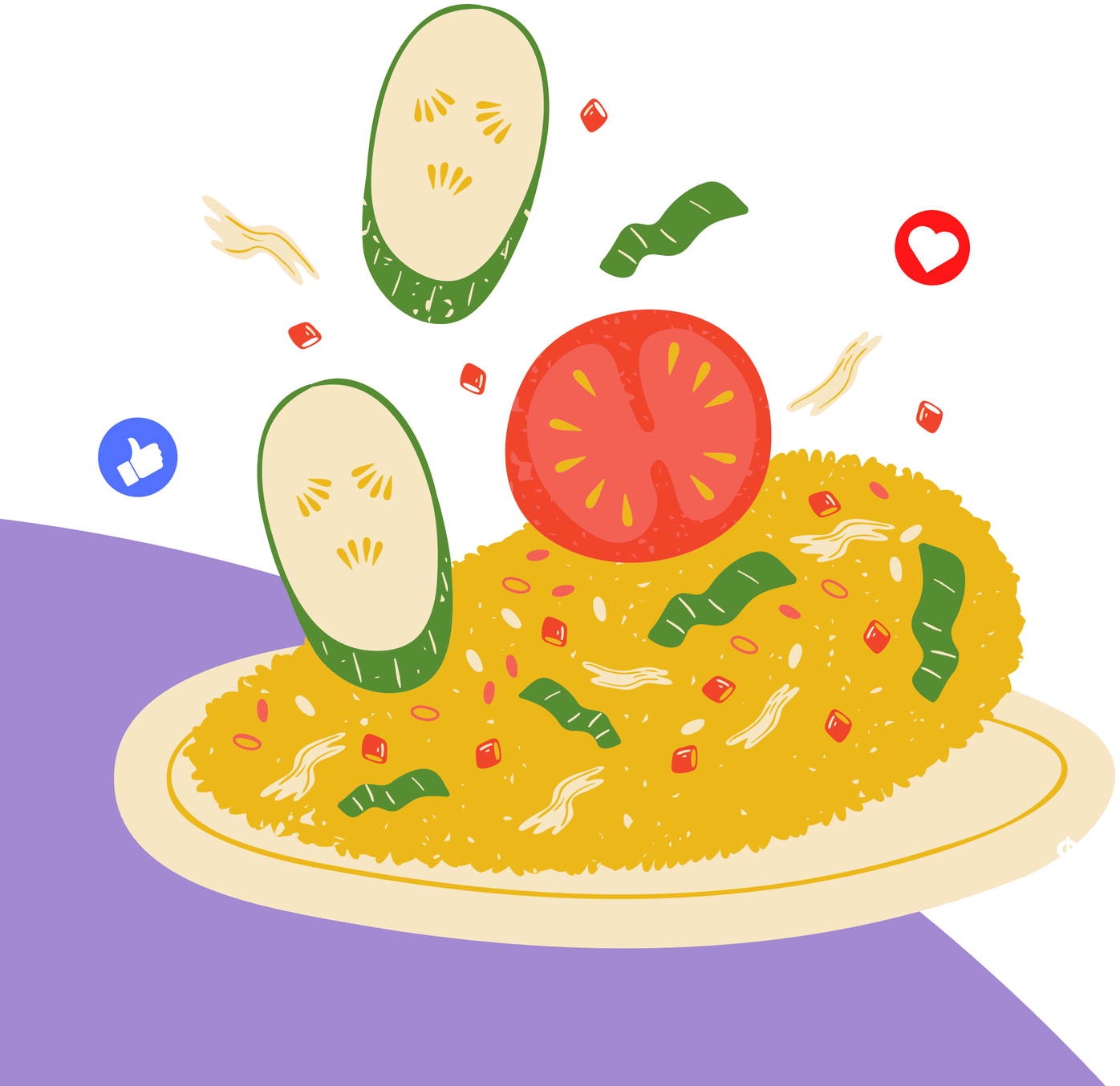
- **Google Firebase** is used as our database.
- Holds user's data - name, phone number, email, passwords, food items, etc.
- Firebase authentication used for login.

Configuration Management

MediaWiki

GitHub

Google Drive



SOFTWARE QUALITY ASSURANCE

FOODIFY

Technical Review

Planning	Choose team, material, dates
Orientation	Present product, progress, goals
Preparation	Check product, note issues
Review Meeting	Consolidate issues
Rework	Correct defects
Verification	Verify product, assess quality

FOODIFY

Technical Review

WALKTHROUGHS

- Informal walkthrough
- Developers not directly involved to perform review and raise issues found
- Iterative rectification of issues identified

FOODIFY

Management Review

- By Program Manager, Quality Manager and Lead Developer
- Every 2 weeks
- Include:
  - Status of schedules
  - Confirm requirements
  - Verify effectiveness
- Help meet time and budget constraints

FOODIFY

Audits

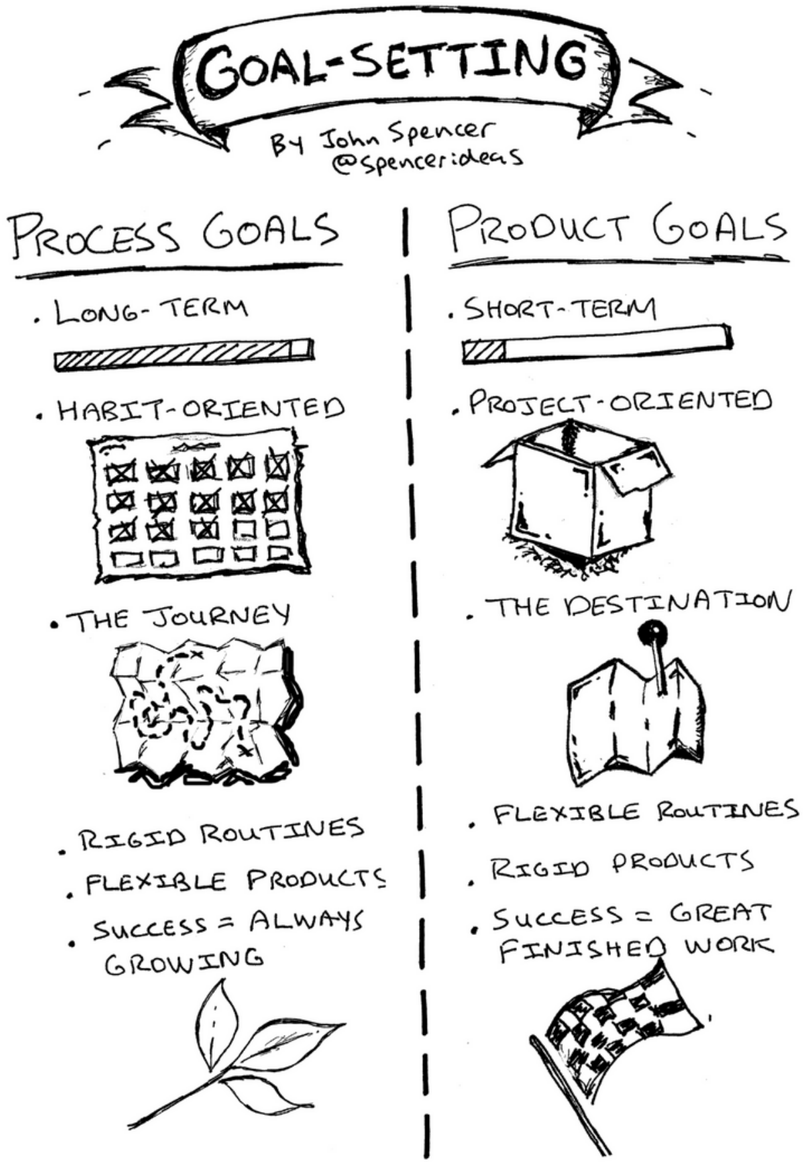
1. Consult people external to organization to conduct independent evaluation

2. Audit to be conducted less often than other reviews

3. Mainly conducted to ensure Team RunTime Terror conforms to regulations, standard and plans such as the Project Plan

FOODIFY

Process Goals



- Identify key areas of quality improvement
- Establish a Quality Culture in the team
- Traceability in Product Lifecyle, technical soundness, conformance to requirements etc. for Foodify

FOODIFY

Commitment to Perform

- **Briefings** on project progress and organization policies
- **QAM** to gauge commitment from team and conformance to proper standards

Ability to Perform

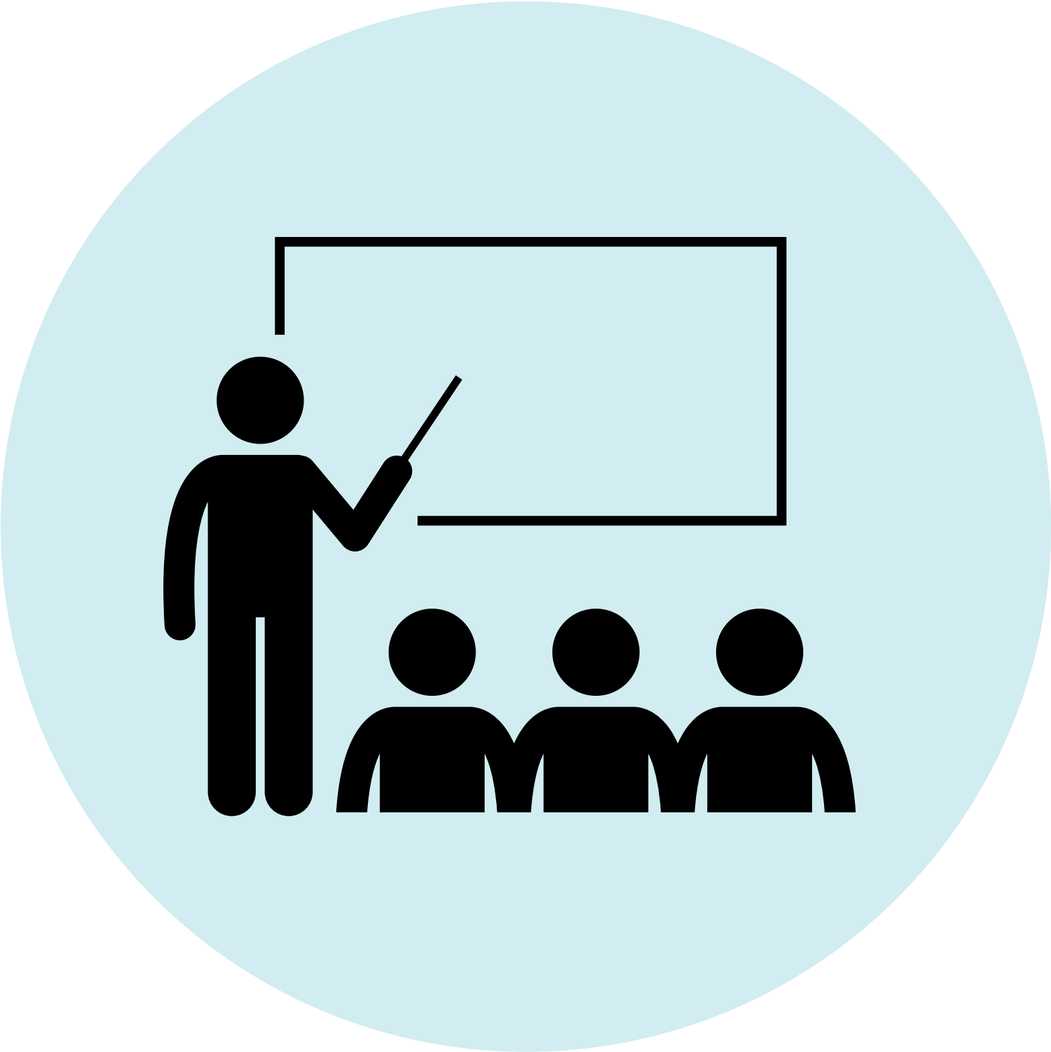
Resources

Access to all requisite Hardware & Software resources:

- PCs
- Hosting services (Firebase)

FOODIFY

Ability to Perform

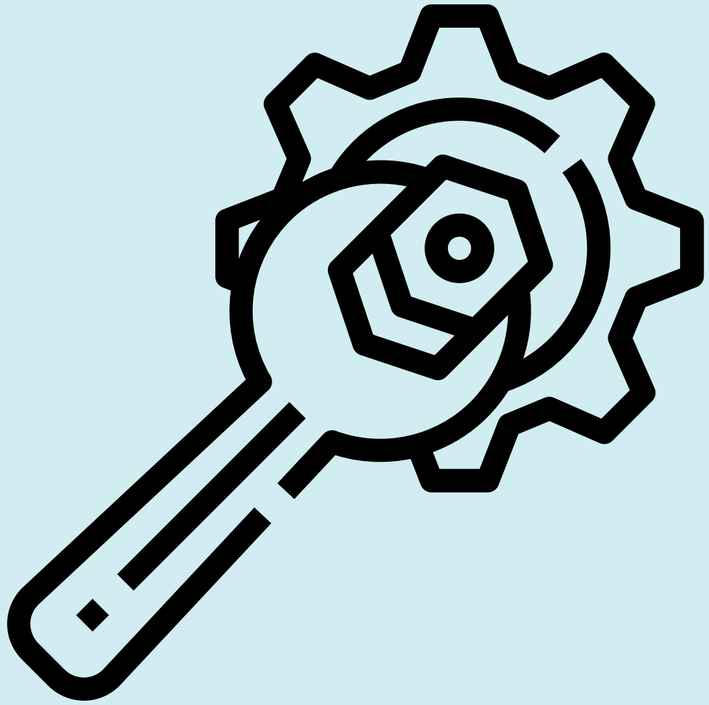


Training

The trainings are periodically revised according to the latest industry standards (IEEE) and trends

FOODIFY

Ability to Perform



Tools

- Project Management using Telegram, Zoom, when2meet
- Version Control with Git
- VSCode Application Development Suite of tools

FOODIFY

Activities Performed

Plans and Procedures

- System Requirement Specification
- Quality Plan
- Risk Management Plan
- Project Plan
- Configuration Management Plan
- Design report on software maintainability
- Change Management Plan
- Release Plan

FOODIFY

Activities Performed

Traceability

- All documents are tracked and stored using Media WIKI and SVN Repository
- Necessary corrective action through a change request workflow

FOODIFY

Measurement, Analysis and Verification

- Periodic assessment and reporting of the project's progress throughout the SDLC
- The performance metrics are made quantifiable and verifiable
- The results are analyzed by PM , and changes suggested are based on the current state
- The amendments are verified using audits by management and SQA

PROJECT MANAGEMENT

Project Organisation

Anandarajan Sindini
Project Manager

Verma Nandini
Lead Developer
Front-end Developer

Kolady Anamika Martin
QA manager
Release Engineer

Chua Jim Sean
Back-end Developer

Duong Ngoc Yen
QA Engineer

PROJECT ESTIMATES

Functional Points (1/2)

Elements	Complexity	Detail
Inputs	Low	User Account
	Low	Food Item
	Average	Food Items on Market Page
	Low	Recipe
Logical Files	High	Food Items on Market Page
	Medium	Recipe Recommendations
	Medium	User Account

Functional Points (2/2)

Elements	Complexity	Detail
Outputs	Low	Delete Food Items
	Average	Delete Food Items on Market Page
	Low	Display recipes
Inquiries	Low	Selecting Food Items
	Low	Recipe Recommendations Ingredients
Interfaces	Average	Food Items, Recipe Recommendations
	Low	User Account

Unadjusted Functional Points

Chracteristics	Low Complexity		Medium Complexity		High Complexity	
Inputs	3	x3	1	x4	0	x6
Outputs	2	x4	1	x5	0	x7
Inquiries	2	x3	0	x4	0	x6
Int. Logic Files	0	x7	2	x10	1	x15
Ext. Logic Files	1	x5	3	x7	0	x10
Unadjusted FP	28		50		15	
Total	**93**					

Adjusted Functional Points

Total Influence Score = 37

Influence Multiplier

= 0.65 + (0.01 x 37) = 1.02

Adjusted Function Points

= 93 x 1.02 = 94.86

Distribution of Effort (1/2)

1990's Industry Data	Work Package	Distribution	Estimates
Preliminary Design (18%)	Project Plan	9%	51.12
	Requirement Specification	9%	51.12
Detailed Design (25%)	User Interace	7%	39.76
	Technical Architecture	11%	62.48
	Data Modeling	7%	39.76
Coding & Unit testing (26%)	Coding & Unit testing	21%	119.28
	Online Documentation	5%	28.4

Distribution of Effort (2/2)

1990's Industry Data	Work Package	Distribution	Estimates
Integration & Test (31%)	Integration & Quality Assurance	31%	176.08
Extrapolated Total Effort			568
2% for Project Management			11.36
3% for Contingency			17.04
Total Effort			**597.4**

Cost Estimates: Hardware

Developer workstations	Cost
7-Dell Precision Workstation 330	$0
Pentium IV 1.4GHz single processor	$0
256 MB RAM	$0
20.8GB IDE drive	$0
Total	**$0**

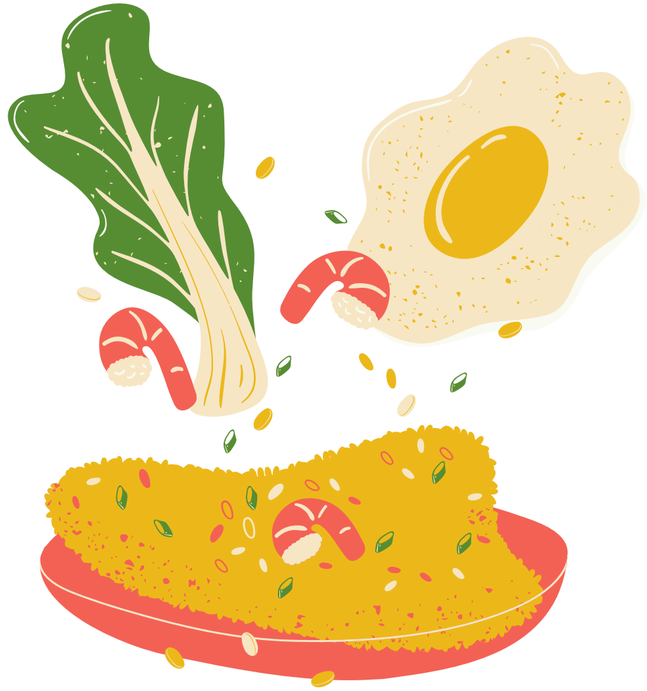
Cost Estimates: Software

GNU, Apache, or Other Free License-based Software	Cost
Apache Web Server	$0
Perl	$0
Software License Provided by Third Party	**Cost**
Microsoft Office 2000	$0
Microsoft Project 2000	$0
Products ESTIMATE Professional	$0
Total	**$0**

Cost Estimates: Other resources

Staff	Cost
5 Employees with 597.4 working Hours with $18.00/hour	$10,753.20

Stationary	Cost
Paper, Photocopying, and other miscellaneous cost	$50
Total	**$10,803.20**

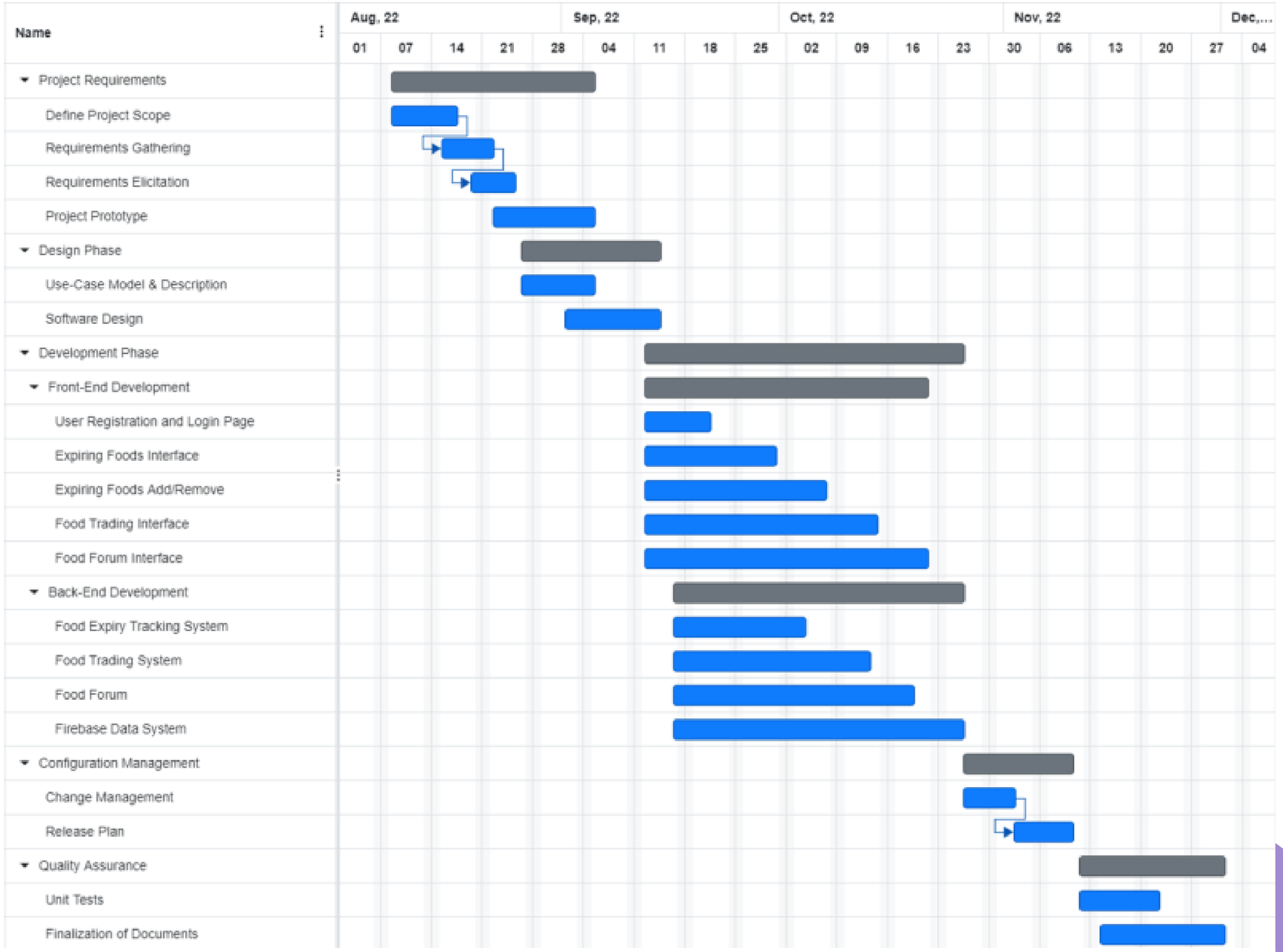


S$ 10,803.20

Activity Dependencies

Work Package #	Description	Duration	Dependencies
X01	Project Plan	8 days	NA
X02	Requirement Specification	10 days	NA
X03	User Interace	7 days	NA
X04	Technical Architecture	14 days	X01, X02, X03
X05	Data Moedling	7 days	X04
X06	Implementation & Unit Testing	30 days	X05
X07	System Testing & Quality Assurance	30 days	X06

Gantt Chart

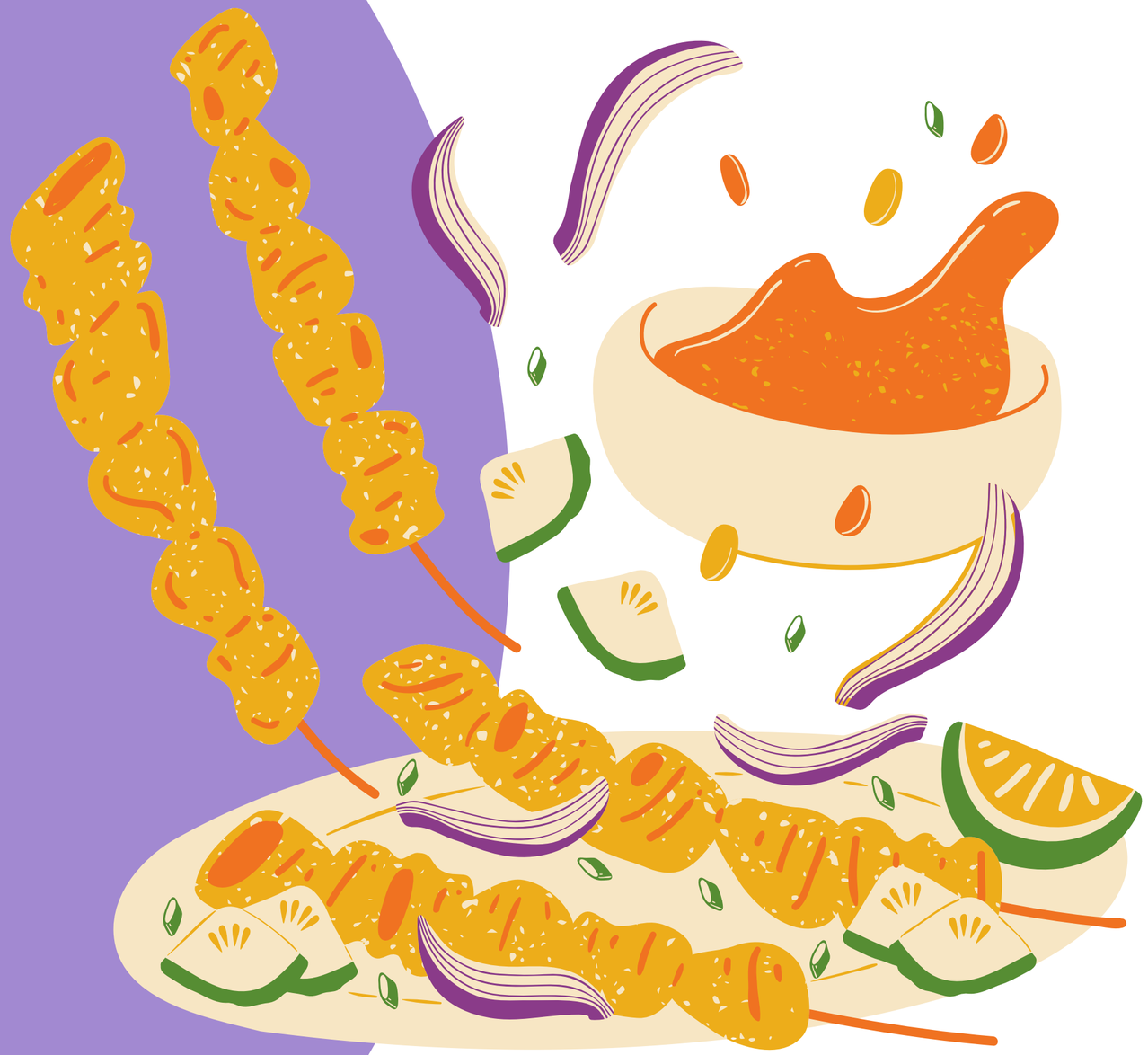

Name
Aug, 22
Sep, 22
Oct, 22
Nov, 22
Dec,...
01 07 14 21 28 04 11 18 25 02 09 16 23 30 06 13 20 27 04
Project Requirements
Define Project Scope
Requirements Gathering
Requirements Elicitation
Project Prototype
Design Phase
Use-Case Model & Description
Software Design
Development Phase
Front-End Development
User Registration and Login Page
Expiring Foods Interface
Expiring Foods Add/Remove
Food Trading Interface
Food Forum Interface
Back-End Development
Food Expiry Tracking System
Food Trading System
Food Forum
Firebase Data System
Configuration Management
Change Management
Release Plan
Quality Assurance
Unit Tests
Finalization of Documents

RISK MANAGEMENT

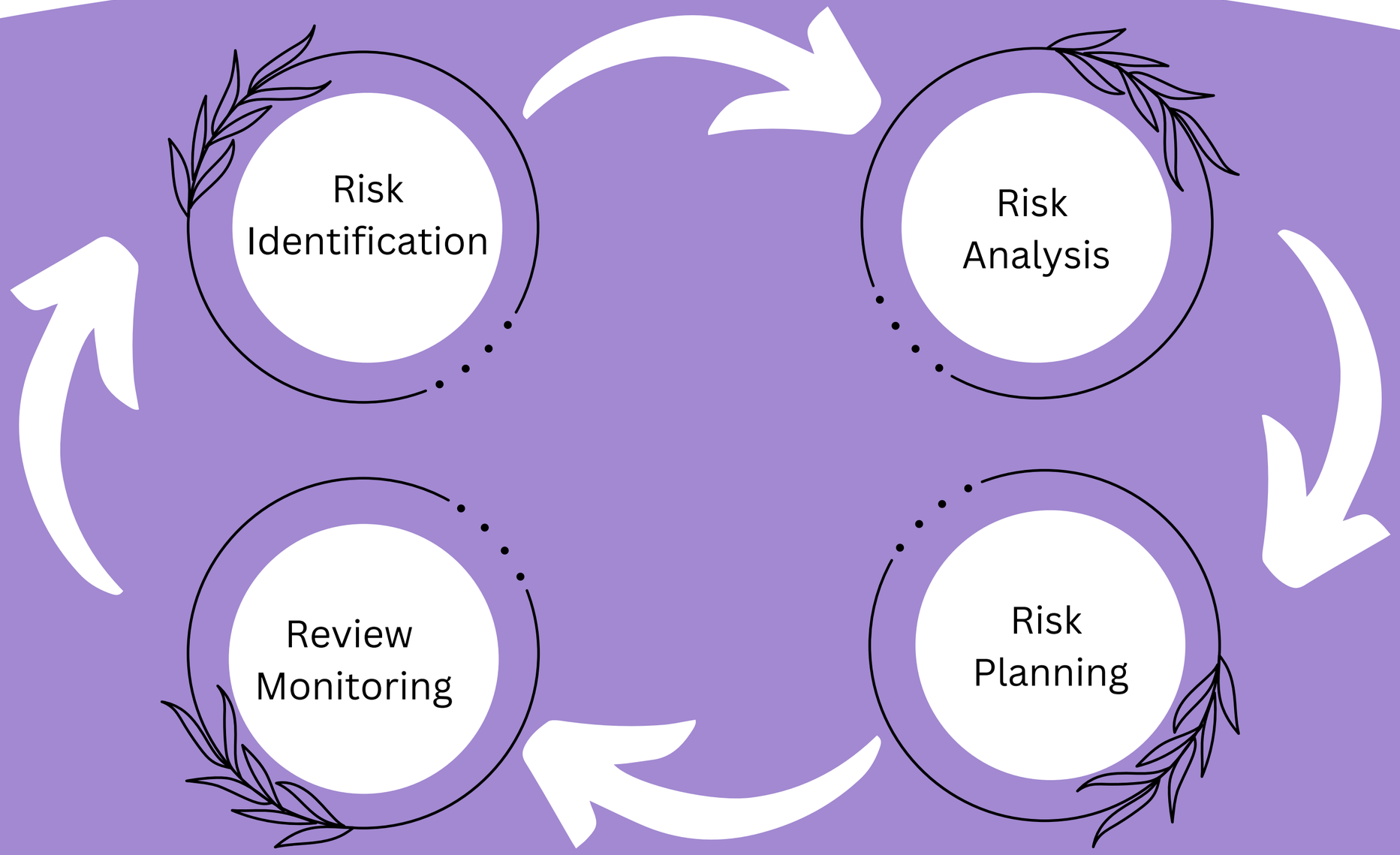
Types of Risk

1. Project Risks
2. Technical Risks
3. Business Risks
4. People Risks
5. Estimation Risks
6. Tools Risks

and many more...



Risk Management Process



Risk Identification

Project team + Stakeholders

Technical

People

Tools

Technical Risk

1. Reuirement specifications might be changed too much that it changes to something much different than initial requirements

People Risk

2. Team memebers falling sick or leaving project

4. Undermined motivation

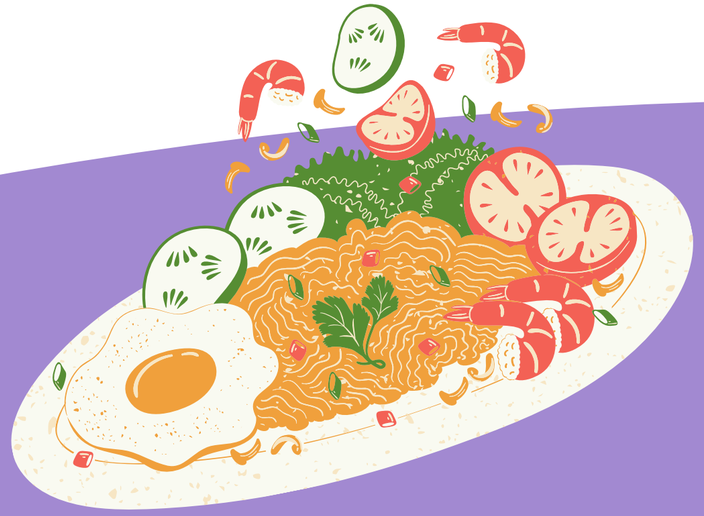
1. Conflicts between developers and project manager

3. Developers are part of multiple projects and might be unavailable to attend meetings

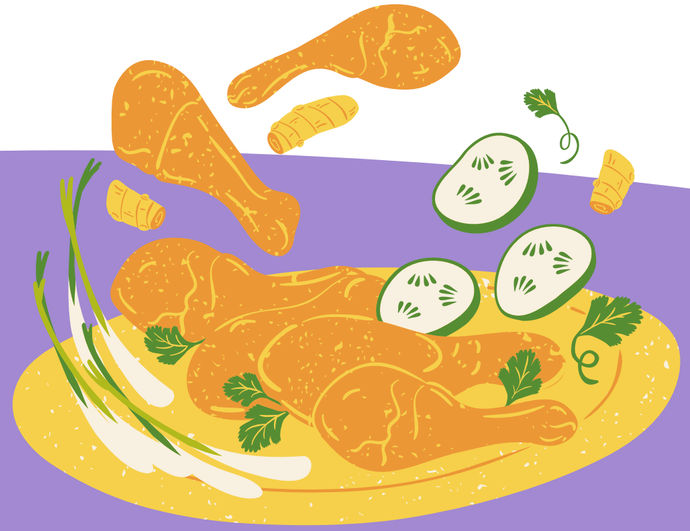
5. Inadequate domain knowledge

Tools Risk

1. Google Authenticator fails



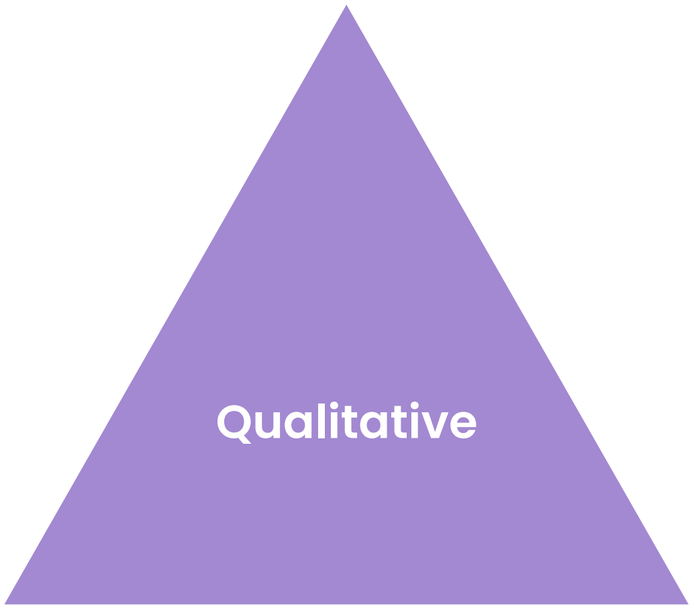
2. Server crashes and essential data is lost

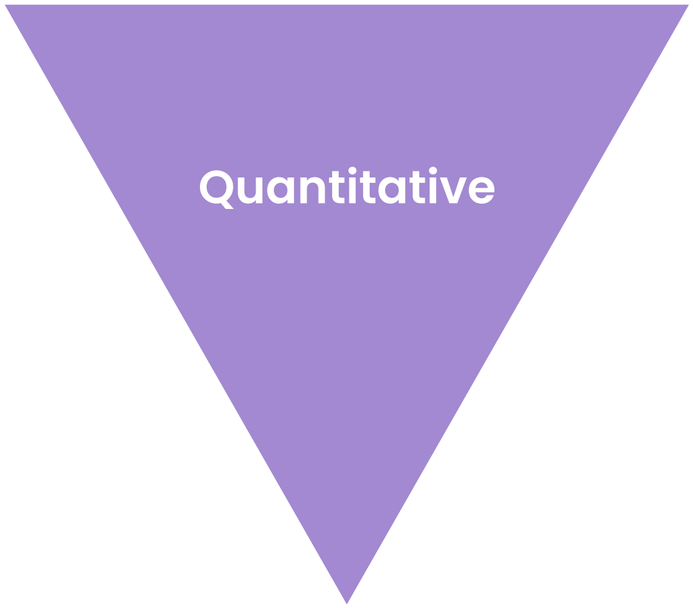


3. Reused software components might have bugs and defects which may break the code

Risk Analysis

Identified risks need to be assessed





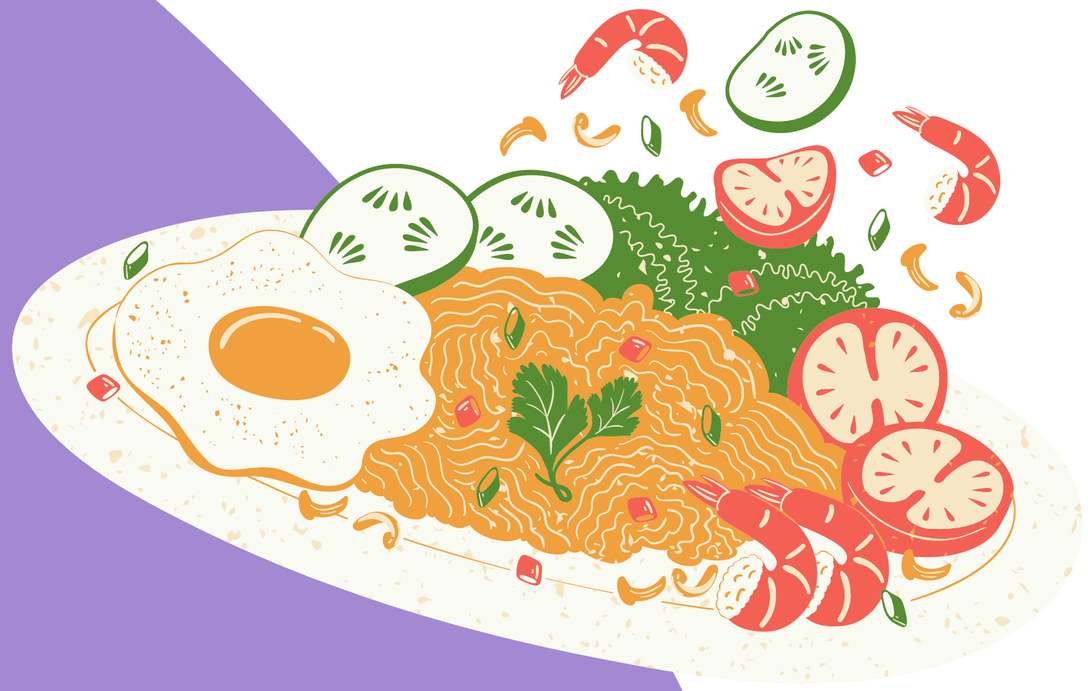
Qualitative Risk Analysis

Probability \ Impact	Low	Medium	High
High	Low	Medium	High
Medium	Low	Medium	Medium
Low	Low	Low	Low

Quantitative Risk Analysis

Area of Risk	Severity	Likelihood	Level of Control	Significance
Technical	3	2	2	7
People	3	3	3	9
Organisational	2	1	2	5
Tools	2	2	2	6
Requirement	4	3	1	8
Estimation	3	3	4	10

Risk Response Planning



AVOID

MITIGATE

ACCEPT

TRANSFER

Avoid

Tools
Investment in high quality technology Do not use Highly redundant databases Highly scalable servers

Mitigate

People	Tools	Estimation
More overlap between work of team members to ensure a substitute is available A team member to ensure team wellness, cohesion and monitoring of KPIs	Thorough analysis of the tools to be used Remove defective components with reliable bought in tools	Investigate use of a program to generate accurate estimation of timelines and project milestones

Risk Monitoring

Level of risk will be tracked, monitored and reported throughout the project

Done through:

Weekly Scrum Calls

Constant testing

Thank you