



CP317 Final Project

By Aadhavan Mani, Syed Manzar, Tonny Meque



Contents

Overview

Feasibility Analysis

Software Process
Model Used

SRS Document

UML Diagrams

Use-Case Diagram

Object Diagram

Sequence
Diagram

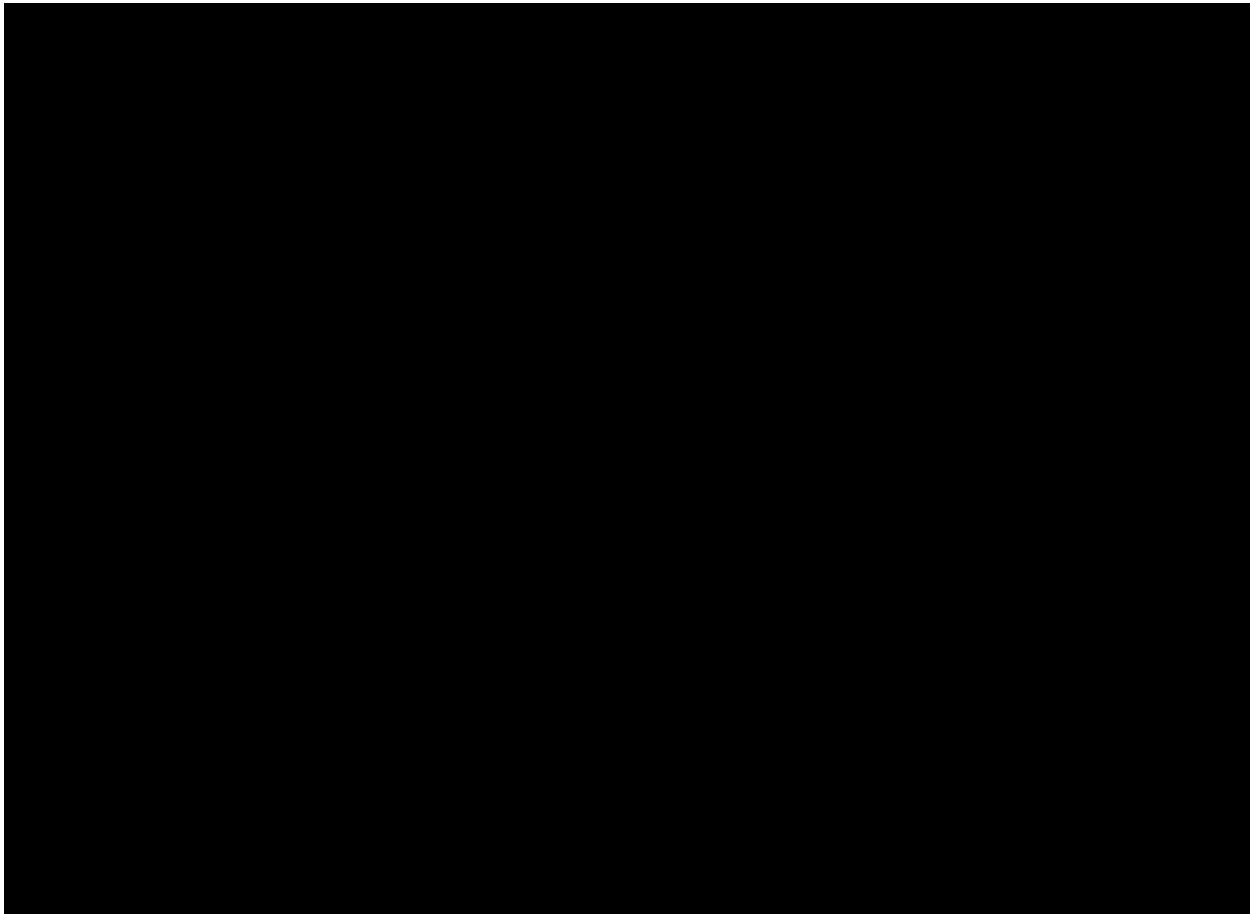
Project
Planning

Gantt Chart

Team

Overview







Feasibility Analysis

1

Technical Feasibility:

Highly feasible.

3

Operational Feasibility:

Highly feasible.

2

Economic Feasibility:

Moderately feasible.

4

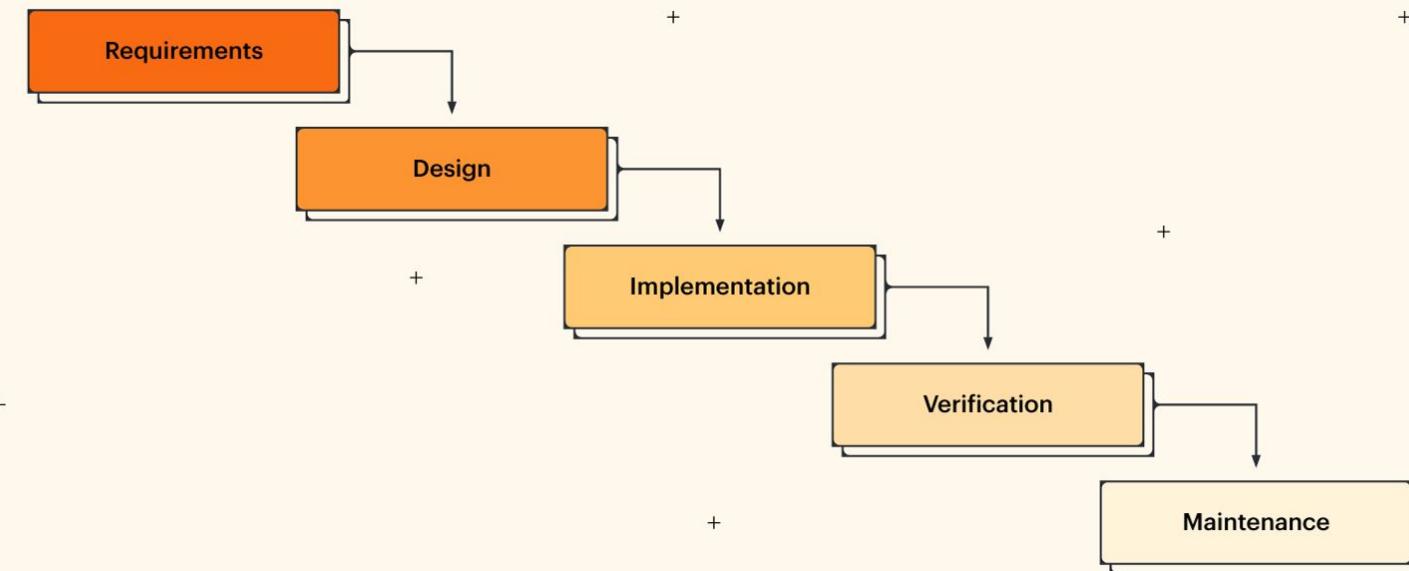
Conclusion:

High technical feasibility and operational feasibility. However, its economic feasibility is moderately feasible.





Software Process Model Used: Waterfall Method



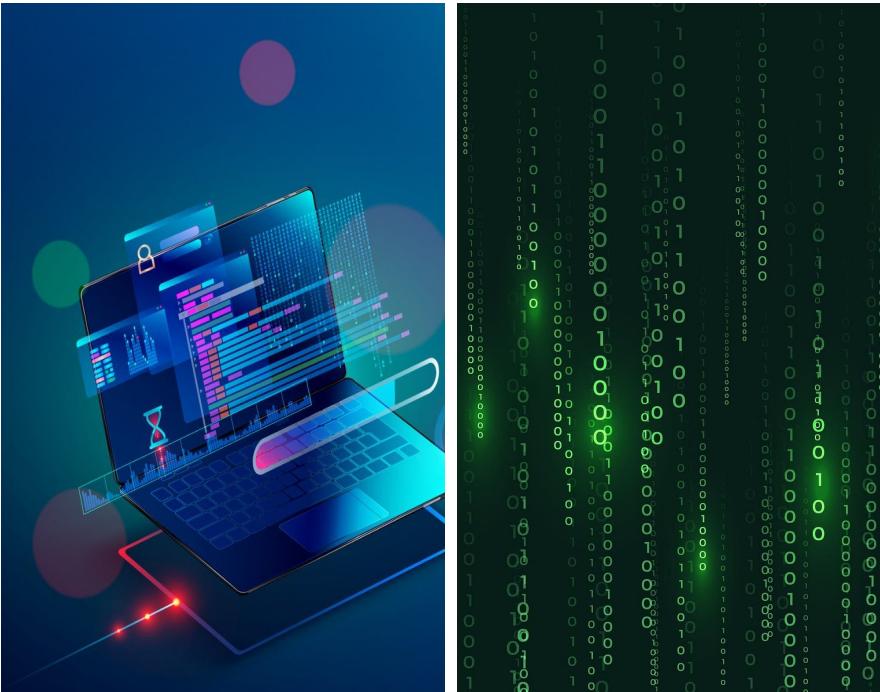


Software Requirements Specification (SRS) Document



What is an SRS Document?

- Very important document in Software development.
- Helps the developer to design the software by describing the functions, functional and non-functional requirements, constraints, scope, and dependencies.
- Helps reduce time, cost, and effort.
- Reduces the chances of needing to redesign the code.

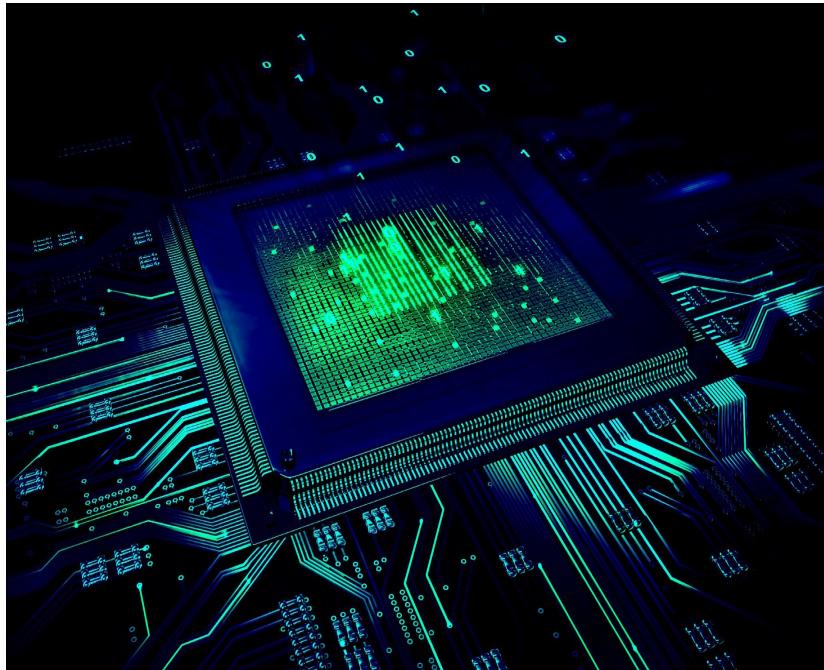




The SRS Document for Our Food Recipe App

Section 1: Introduction

- Purpose
- Project Scope
- Definitions, Acronyms and Abbreviations
- References

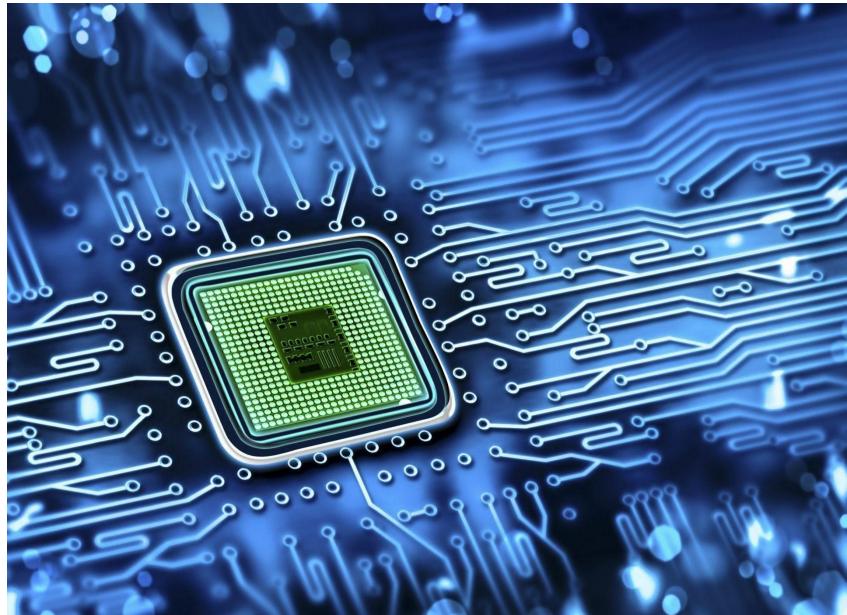




The SRS Document for Our Food Recipe App

Section 2: Overall Description

- Product Perspective
- Product Functions
- User Characteristics
- Constraints
- Assumptions and Dependencies

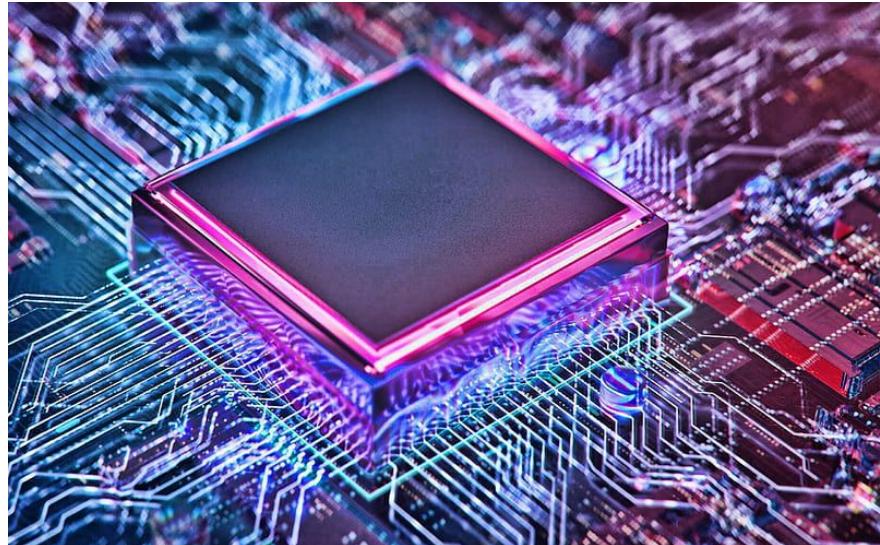




The SRS Document for Our Food Recipe App

Section 3: Specific Requirements

- External Interfaces
- Functional Requirements
- Performance Requirements
- Quality Attributes





Project Architecture: UML Diagrams



Use-case Diagram

Object Diagram



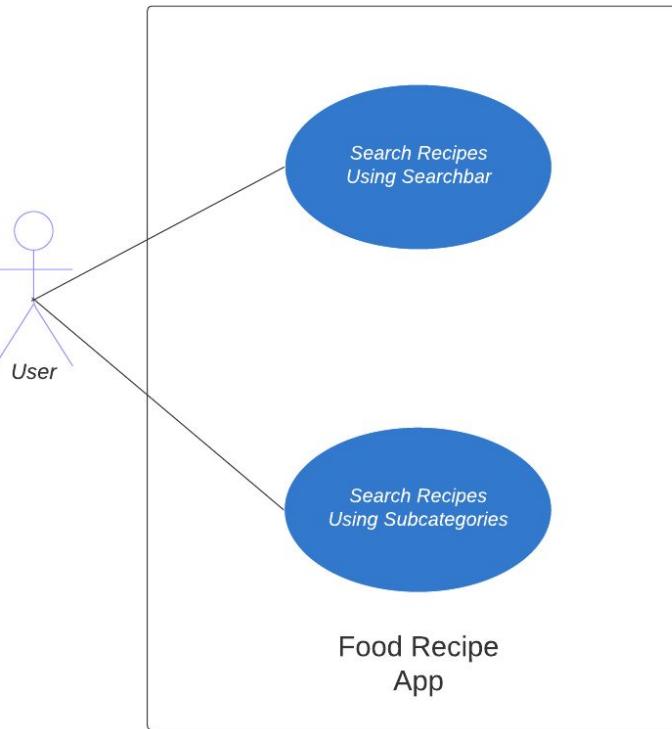
Sequence Diagram



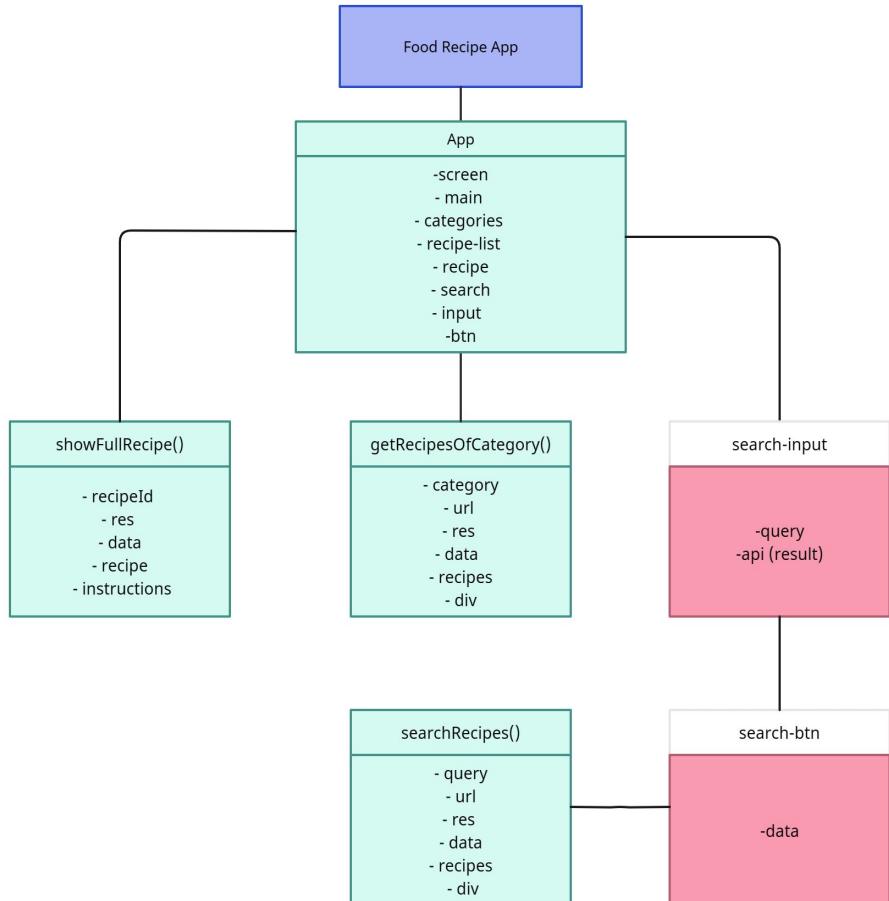
Use-Case Diagram

- Attributed to Swedish Engineer Ivar Jacobson.
- Helps define the interactions between the actor, and the use cases.
- Here the actor is the app user
- 2 Use Cases
 - Searching using the search bar
 - Searching using the “Subcategories”

Use Case Diagram

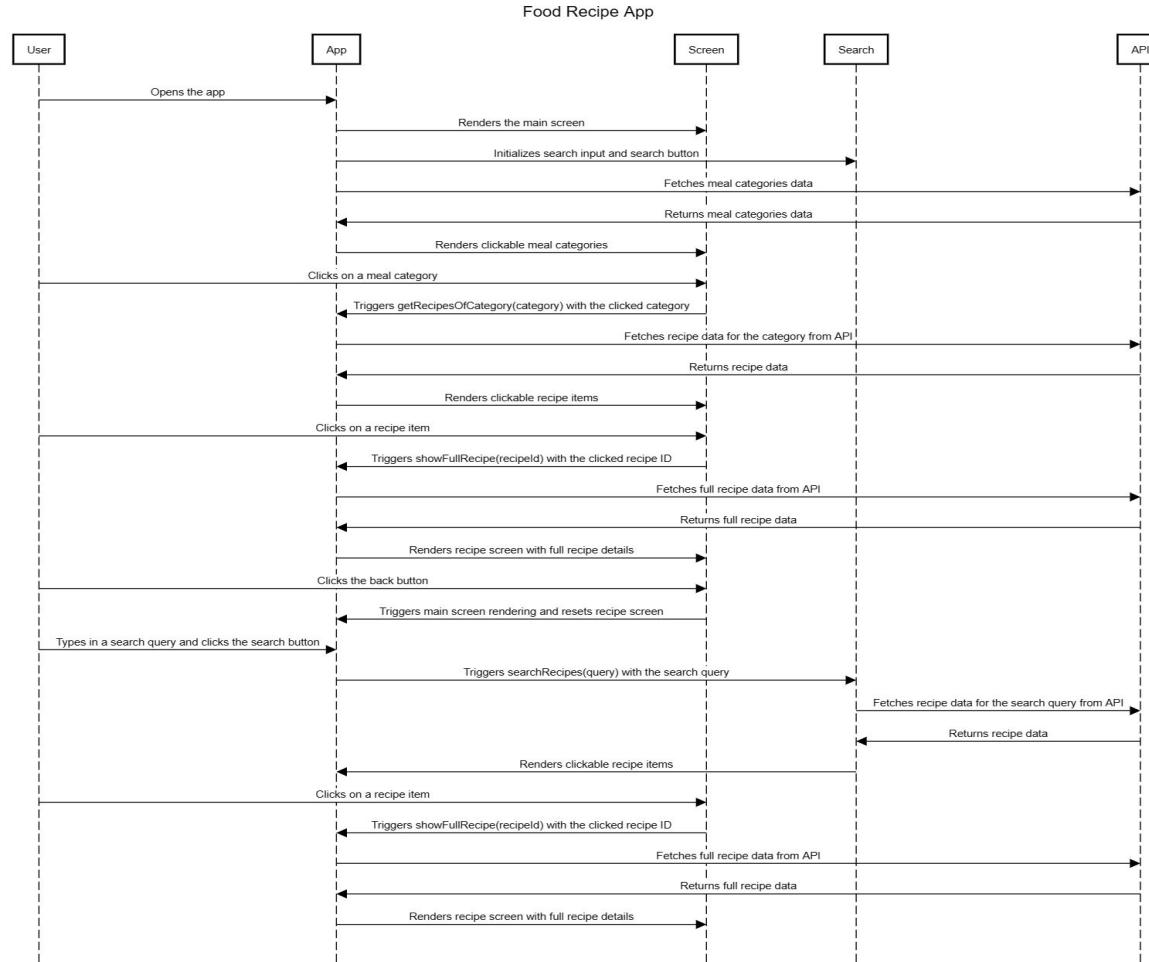


Object Diagram





Sequence Diagram





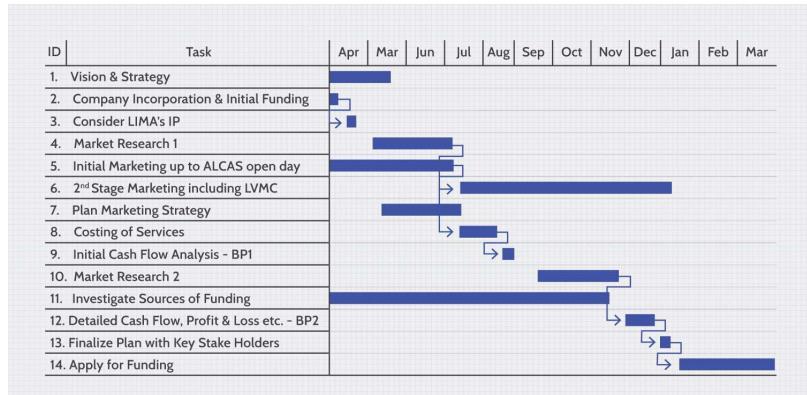
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name = "viewport" content="width=device-width, initial-scale=1.0">
    <title>Food Pantry App</title>
    <!--The <link> element links to an external stylesheet-->
    <link rel="stylesheet" type="text/css" href="style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Hind&display=swap" href="https://fonts.googleapis.com/css2?family=Hind&display=swap" type="stylesheet">
</head>
<body>
    <!--the main container for the app-->
    <div class="app">
        <div class="screen main-screen">
            <!--displays the app logo-->
            <div class="logo">
                Food Pantry App
            </div>
            <!--Displays the search bar-->
            <div class="search-container">
                <input type="text" placeholder="Search categories..." class="search-input">
                <button class="search-btn">Search</button>
            </div>
            <!--Displays the recipe categories-->
            <div class="titles">
                <h1>Ingredients</h1>
            </div>
            <div class="categories"></div>
            <div class="titles">
                <h1>Recipes</h1>
            </div>
            <div class="recipe-list"></div>
        </div>
        <div class="screen recipe-screen hidden">
            <div class="back-btn">&lt;</div>
            <div class="thumbnail">
                <img>
            </div>
            <!--Displays the recipe details-->
            <div class="details">
                <h2></h2>
                <div>
                    <h4>Ingredients</h4>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

Project Planning

To help provide an idea of the **timeline** of our project and the progress of the different aspects of it, we made a **Gantt Chart**

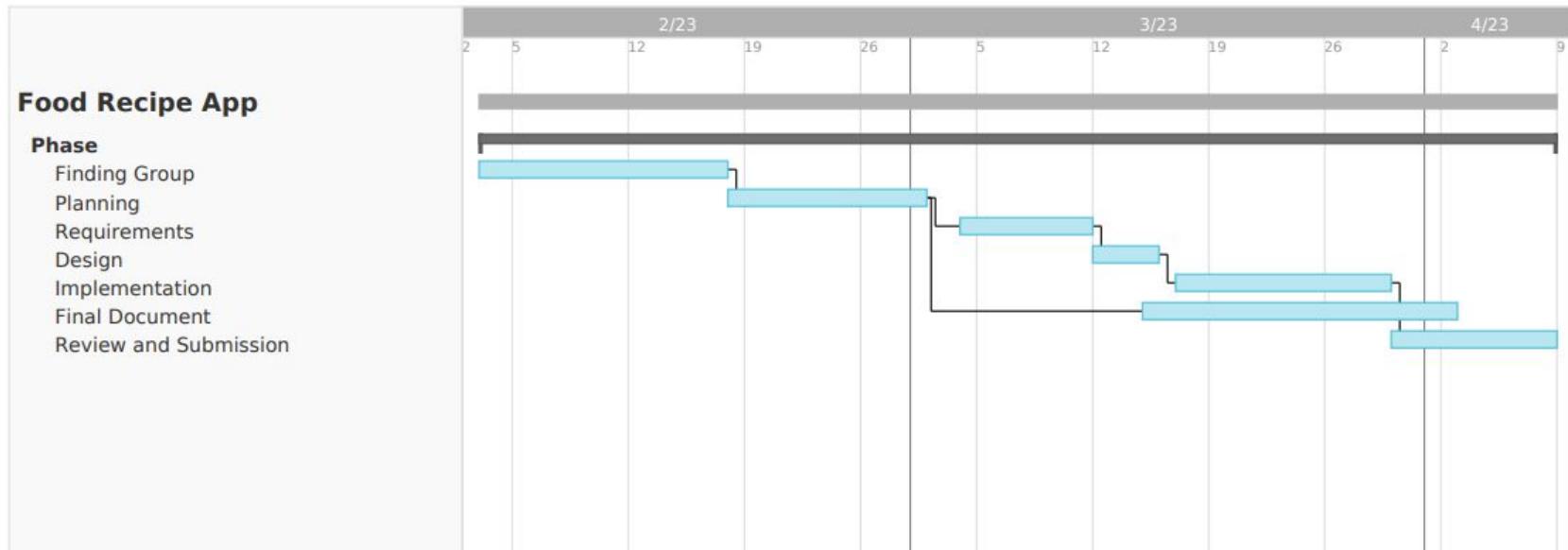
About Gantt Charts

- Invented in the early 19th Century by American engineer Henry Gantt
- Why is it useful?
 - Scheduling
 - Planning
 - Keeping track of the progress
 - Helps with visualising
 - Changes and modification

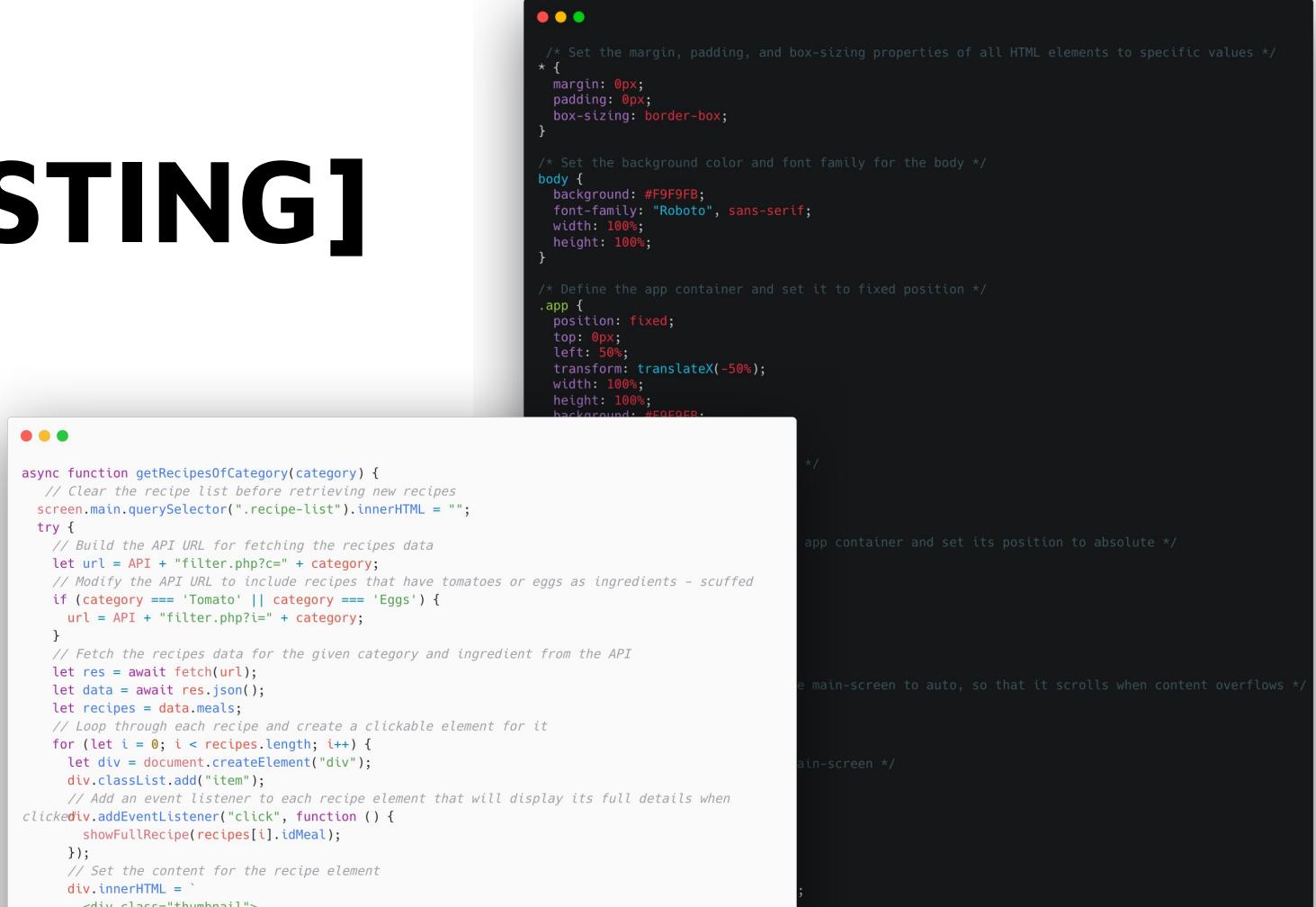




The Gantt Chart for Our Food Recipe App



[TESTING]



```
/* Set the margin, padding, and box-sizing properties of all HTML elements to specific values */
* {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
}

/* Set the background color and font family for the body */
body {
  background: #F9F9FB;
  font-family: "Roboto", sans-serif;
  width: 100%;
  height: 100%;
}

/* Define the app container and set it to fixed position */
.app {
  position: fixed;
  top: 0px;
  left: 50%;
  transform: translateX(-50%);
  width: 100%;
  height: 100%;
  background: #E0E0E0;
}

/* Main function to get recipes by category */
async function getRecipesOfCategory(category) {
  // Clear the recipe list before retrieving new recipes
  screen.main.querySelector(".recipe-list").innerHTML = "";
  try {
    // Build the API URL for fetching the recipes data
    let url = API + "filter.php?c=" + category;
    // Modify the API URL to include recipes that have tomatoes or eggs as ingredients - scuffed
    if (category === 'Tomato' || category === 'Eggs') {
      url = API + "filter.php?i=" + category;
    }
    // Fetch the recipes data for the given category and ingredient from the API
    let res = await fetch(url);
    let data = await res.json();
    let recipes = data.meals;
    // Loop through each recipe and create a clickable element for it
    for (let i = 0; i < recipes.length; i++) {
      let div = document.createElement("div");
      div.classList.add("item");
      // Add an event listener to each recipe element that will display its full details when clicked
      div.addEventListener("click", function () {
        showFullRecipe(recipes[i].idMeal);
      });
      // Set the content for the recipe element
      div.innerHTML = `
        <div class="thumbnail">
          <img alt="Thumbnail image for the recipe" />
        </div>
        <div class="title">
          ${recipes[i].strMeal}
        </div>
        <div class="description">
          ${recipes[i].strDescription}
        </div>
        <div class="tags">
          ${recipes[i].strTags}
        </div>
        <div class="info">
          ${recipes[i].strSource}
        </div>
        <div class="info">
          ${recipes[i].strInstructions}
        </div>
        <div class="info">
          ${recipes[i].strIngredient1} ${recipes[i].strIngredient2} ${recipes[i].strIngredient3}
        </div>
        <div class="info">
          ${recipes[i].strMeasure1} ${recipes[i].strMeasure2} ${recipes[i].strMeasure3}
        </div>
      `;
      screen.main.appendChild(div);
    }
  } catch (error) {
    console.error(error);
  }
}

// Function to show full recipe details
function showFullRecipe(id) {
  // Fetch the full recipe details from the API
  let url = API + "meal/" + id;
  fetch(url)
    .then(res => res.json())
    .then(data => {
      let div = document.createElement("div");
      div.classList.add("full-recipe");
      div.innerHTML = `
        <h2>${data.meals[0].strMeal}</h2>
        <p>${data.meals[0].strDescription}</p>
        <img alt="Full recipe image" />
        <ul>
          <li>${data.meals[0].strIngredient1} ${data.meals[0].strMeasure1}</li>
          <li>${data.meals[0].strIngredient2} ${data.meals[0].strMeasure2}</li>
          <li>${data.meals[0].strIngredient3} ${data.meals[0].strMeasure3}</li>
        </ul>
      `;
      screen.main.appendChild(div);
    })
    .catch(error => console.error(error));
}
```



Test Descriptions

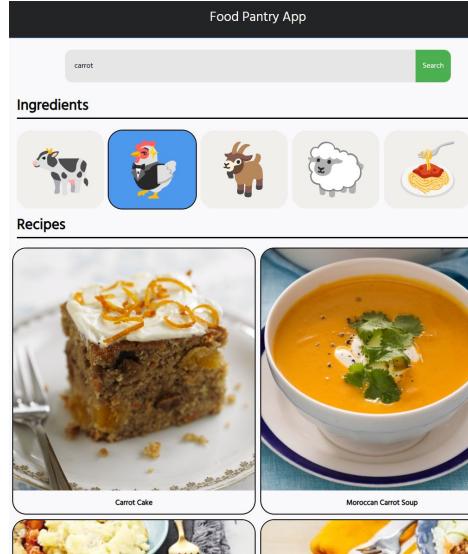
Food Pantry App

carrot

Ingredients

Recipes

Carrot Cake Moroccan Carrot Soup



1. This test case is designed to verify that the recipe search system returns multiple recipes containing "carrot" as an ingredient or in the name when "carrot" is entered as input.
2. Test the usability of the category buttons that return recipes with that category.

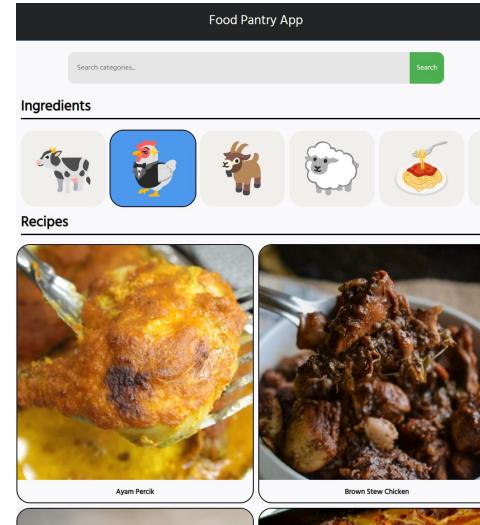
Food Pantry App

Search categories...

Ingredients

Recipes

Ayam Perik Brown Stew Chicken





Test Screenshots with Output

Food Pantry App

cake

Ingredients

Recipes

Pancakes

Rock Cakes

Beef and Mustard Pie

Ingredients

- Beef - 1kg
- Plain Flour - 2 lbs
- Rapeseed Oil - 2 lbs
- Red Wine - 200ml
- Beef Stock - 400ml
- Onion - 1 finely sliced
- Carrots - 2 chopped
- Thyme - 3 songs
- Mustard - 2 lbs
- Egg Yolks - 2 free range
- Puff Pastry - 400g
- Green Beans - 300g
- Butter - 25g
- Salt - pinch
- Pepper - pinch

Instructions

1. Preheat the oven to 150C/300F/Gas 2.
2. Toss the beef and flour together in a bowl with some salt and black pepper.
3. Heat a large casserole until hot, add half of the rapeseed oil and enough of the beef to just cover the bottom of the casserole.
4. Fry until browned on each side, then remove and set aside. Repeat with the remaining oil and beef.
5. Return the beef to the pan, add the wine and cook until the volume of liquid has reduced by half, then add the stock, onion, carrots, thyme and mustard, and season well with salt and pepper.
6. Cover with a lid and place in the oven for two hours.
7. Remove from the oven, check the seasoning and set aside to cool. Remove the thyme.

Food Pantry App

Ingredients

Recipes

Kapsalon

Keleya Zaara



Thank you.

