MYPANTRY APP PROJECT DEMO

by Bit Legion

Adam Tan Jun Xiang	U2120309H
Sandhiya Sukumaran	U2120682B
Thomas Emmanuel Mathew	U2123204K
Khoo Jia Xuan Mavis	U2121376B
Ie Jia Sheng	U2122557E
Nguyen Gia Khanh	U2120680J

Problem Statement

Many people struggle with deciding what to cook with the ingredients they have in their pantry. The limited variety of ingredients options and the lack of inspiration makes it difficult to come up with creative and nutritious meal ideas.

Our Solution?

Created a web app that allows users to create a virtual pantry and generate recipe ideas from selected ingredients in the pantry.

Our target audience?

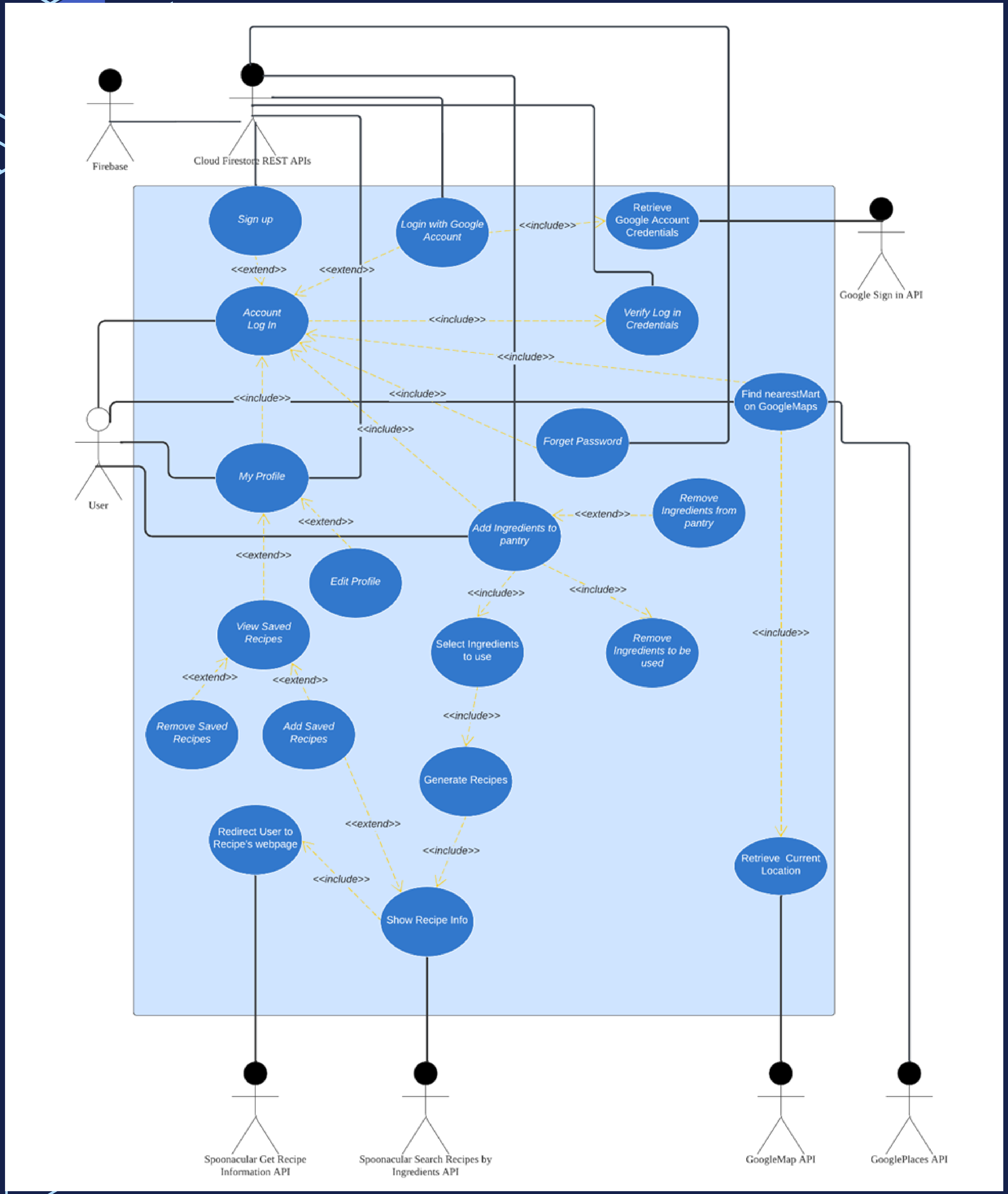
Aspiring home chefs who want to cook delicious meals at home

What does our app have to offer?

Main Functionalities

1. User sign in and registration of account.
2. Add and remove ingredients for "MyPantry".
3. Generate recipes ideas from selected ingredients.
4. Save recipes into a 'Saved Recipes' page.
5. Locate nearby grocery marts.
6. Edit User Profile.

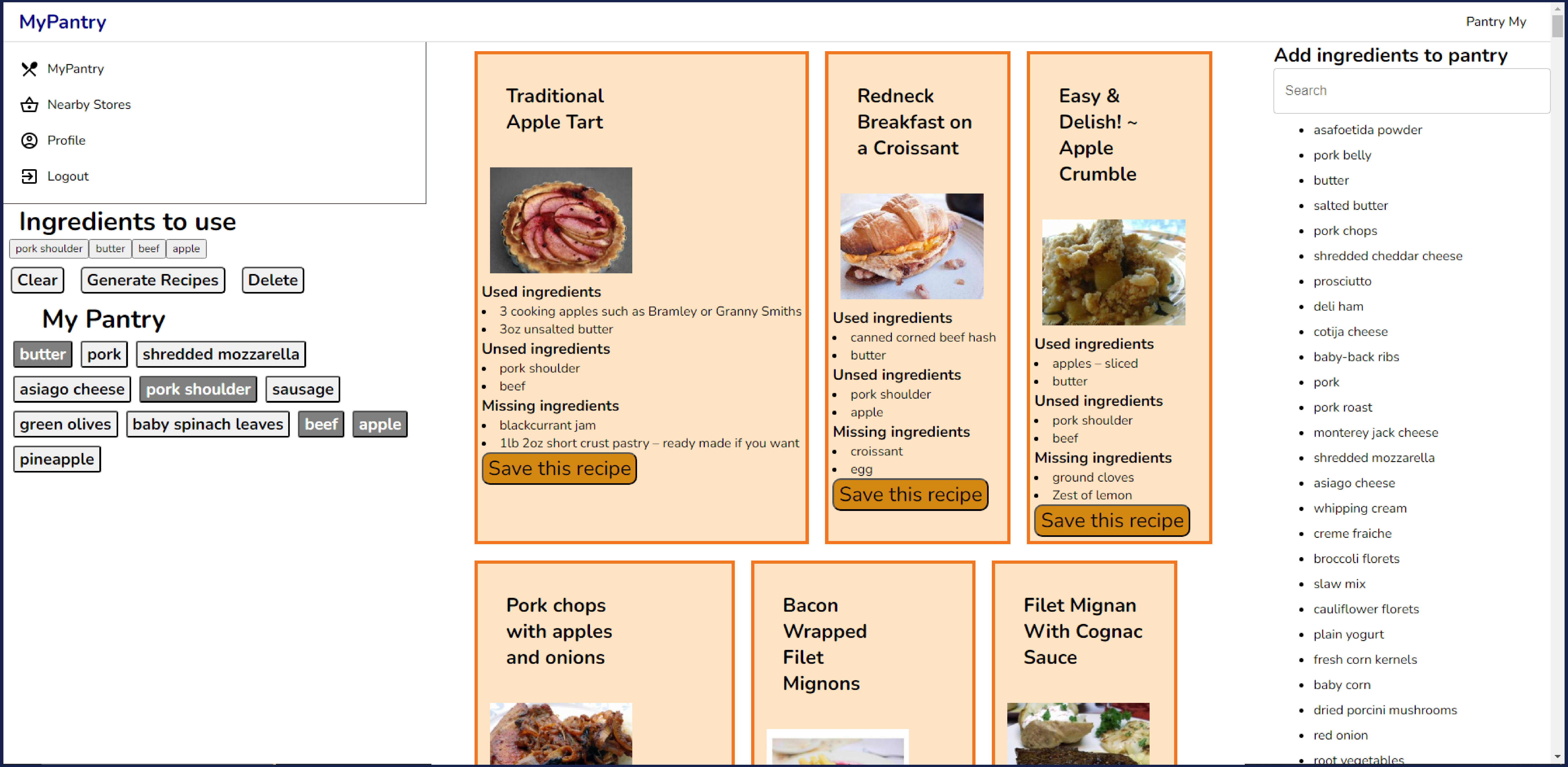
Use case diagram



APIs we used..

- Google sign in API
- Spoonacular Recipe Information by ID API
- Spoonacular search recipe by ingredients API
- Google Map API
- Google Places API
- Firestore API methods
  - onSnapshot
  - getDocs
  - getFirestore
  - deleteDoc

Project demonstration



Ingredients to use

pork shoulder | butter | beef | apple

My Pantry

butter | pork | shredded mozzarella | asiago cheese | pork shoulder | sausage | green olives | baby spinach leaves | beef | apple | pineapple

Traditional Apple Tart

Used ingredients

- 3 cooking apples such as Bramley or Granny Smiths
- 3oz unsalted butter

Unsed ingredients

- pork shoulder
- beef

Missing ingredients

- blackcurrant jam
- 1lb 2oz short crust pastry – ready made if you want

Redneck Breakfast on a Croissant

Used ingredients

- canned corned beef hash
- butter

Unsed ingredients

- pork shoulder
- apple

Missing ingredients

- croissant
- egg

Easy & Delish! ~ Apple Crumble

Used ingredients

- apples – sliced
- butter

Unsed ingredients

- pork shoulder
- beef

Missing ingredients

- ground cloves
- Zest of lemon

Pork chops with apples and onions

Bacon Wrapped Filet Mignons

Filet Mignan With Cognac Sauce

Add ingredients to pantry

- asafoetida powder
- pork belly
- butter
- salted butter
- pork chops
- shredded cheddar cheese
- prosciutto
- deli ham
- cotija cheese
- baby-back ribs
- pork
- pork roast
- monterey jack cheese
- shredded mozzarella
- asiago cheese
- whipping cream
- creme fraiche
- broccoli florets
- slaw mix
- cauliflower florets
- plain yogurt
- fresh corn kernels
- baby corn
- dried porcini mushrooms
- red onion
- root vegetables

Technologies we used....

React JS

Firebase

Google Cloud

Software engineering practices

1

KANBAN (AGILE)

Kanban is an agile method that allows us to visualize workflow and track project progress.

2

FREQUENT REFACTORING

Continuously modifying and reorganizing our code to improve its readability. The variables are also named to make their functionality clearer.

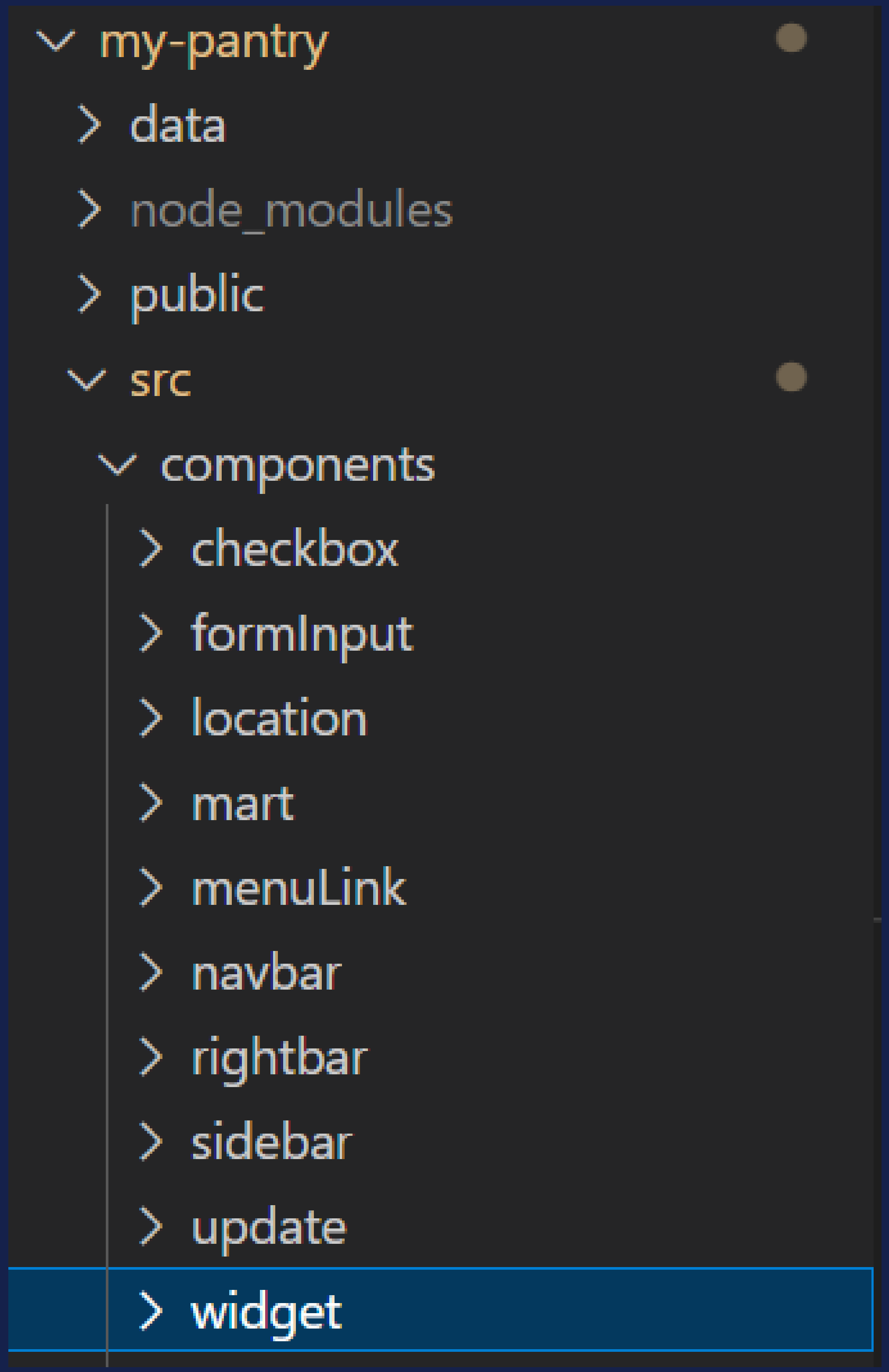
Software Principles applied...

The Separation of Concerns (SOC) Principle

Code is organized into small reusable components in ReactJs to promote modularity

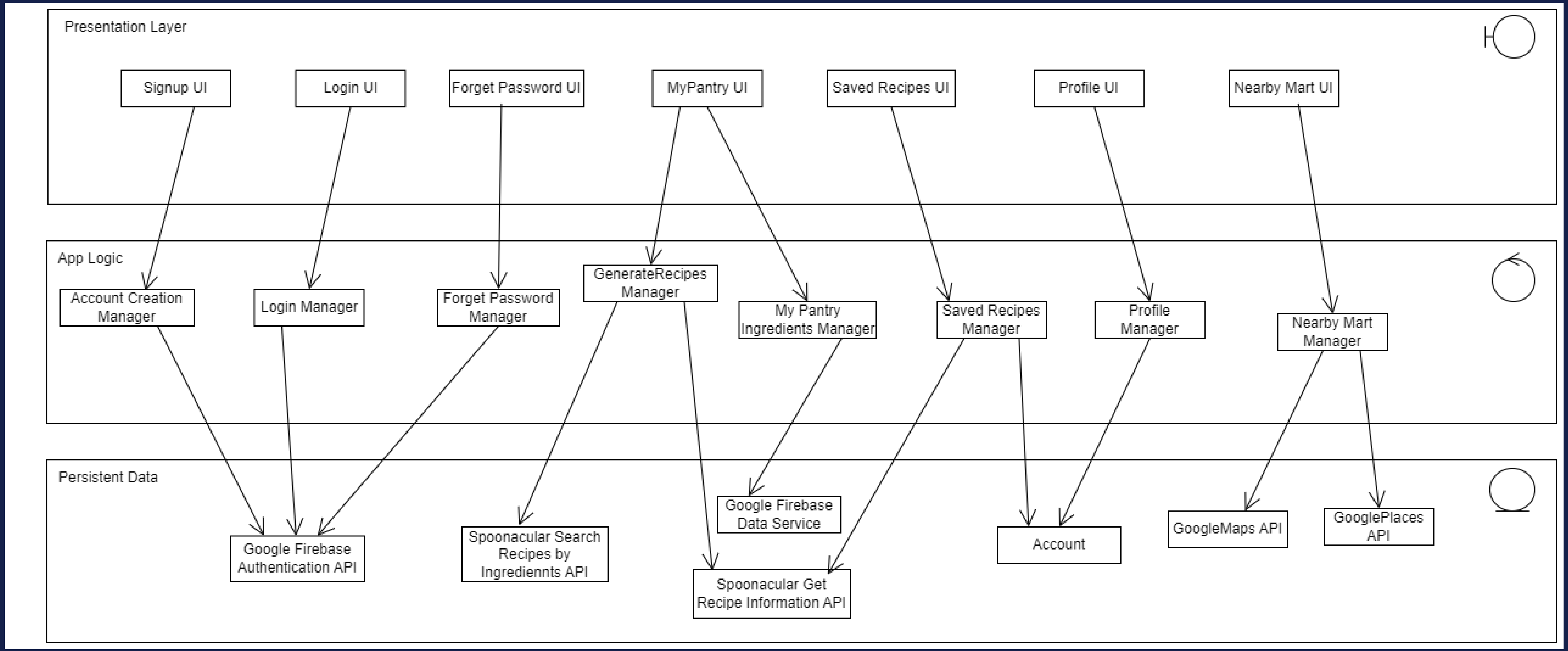
Don't Repeat Yourself (DRY) Principle

Creating components that are reuseable throughout the application. This reduces the amount of code to be written.



```
my-pantry
  data
  node_modules
  public
  src
    components
      checkbox
      formInput
      location
      mart
      menuLink
      navbar
      rightbar
      sidebar
      update
      widget
```

SYSTEM ARCHITECTURE



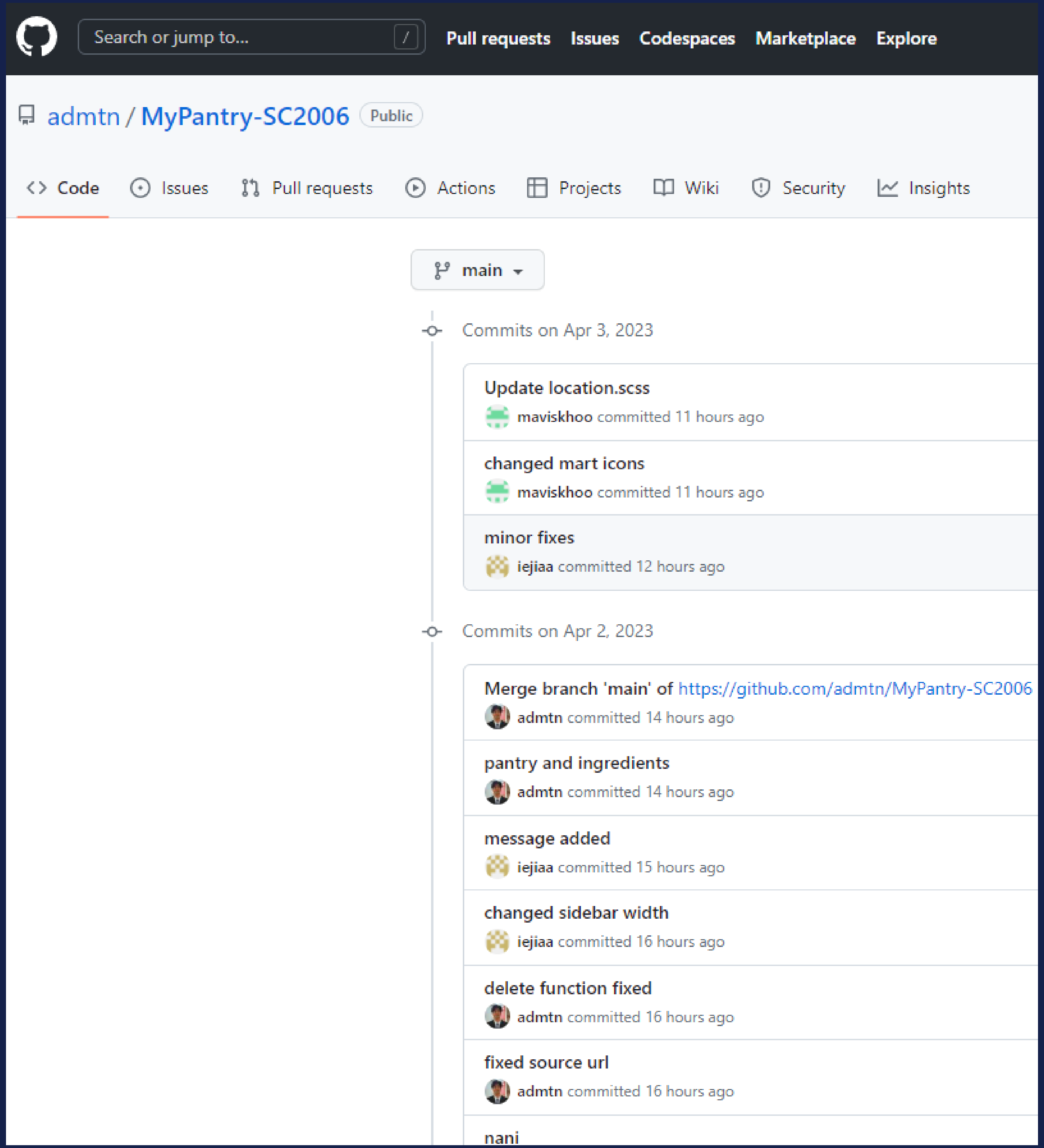
The Three-Tier Architecture

A software application architecture that organizes applications into three logical and physical computing tiers: the presentation tier, the application tier, and the data tier

GOOD TRACEABILITY

1. Version Control

Used GitHub to track history and changes



Commits on Apr 3, 2023

Update location.scss
maviskhoo committed 11 hours ago

changed mart icons
maviskhoo committed 11 hours ago

minor fixes
iejiaa committed 12 hours ago

Commits on Apr 2, 2023

Merge branch 'main' of https://github.com/admtn/MyPantry-SC2006
admtn committed 14 hours ago

pantry and ingredients
admtn committed 14 hours ago

message added
iejiaa committed 15 hours ago

changed sidebar width
iejiaa committed 16 hours ago

delete function fixed
admtn committed 16 hours ago

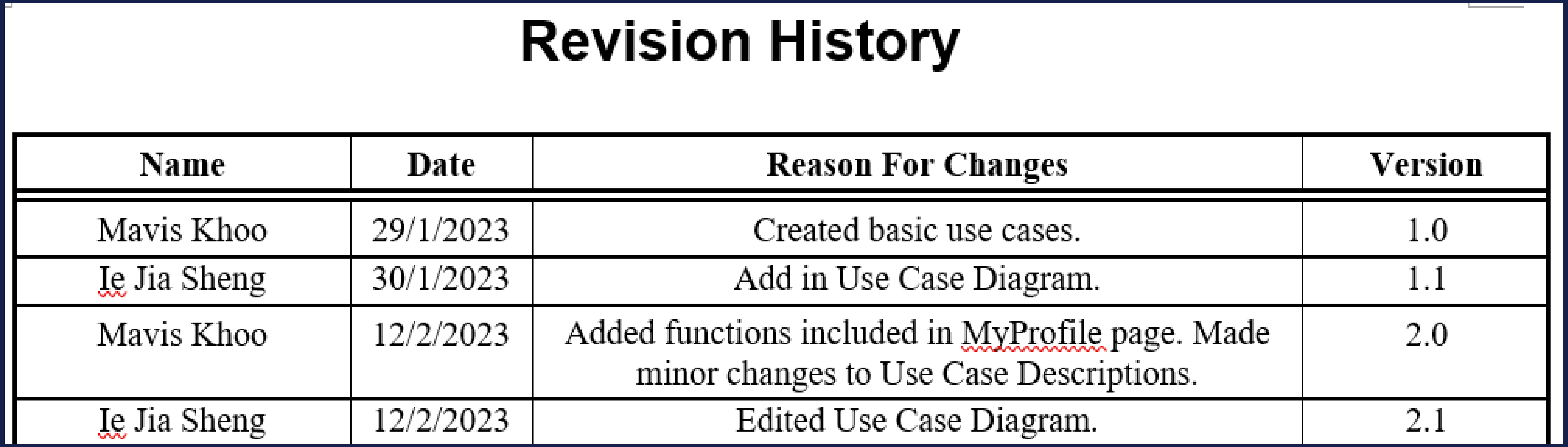
fixed source url
admtn committed 16 hours ago

nani

GOOD TRACEABILITY

2. Documentation

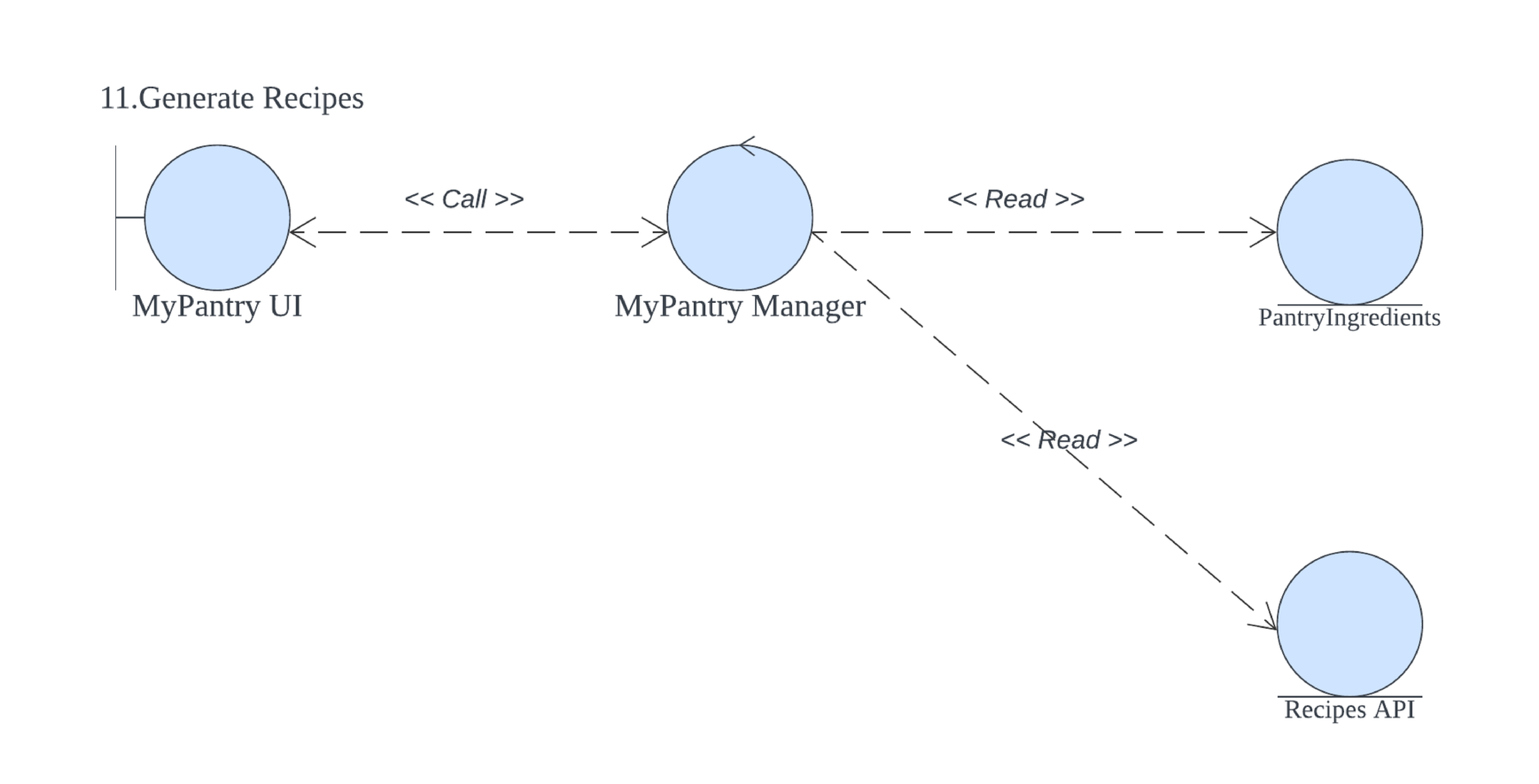
Ensures that the requirements are well defined and shown throughout the software development lifecycle



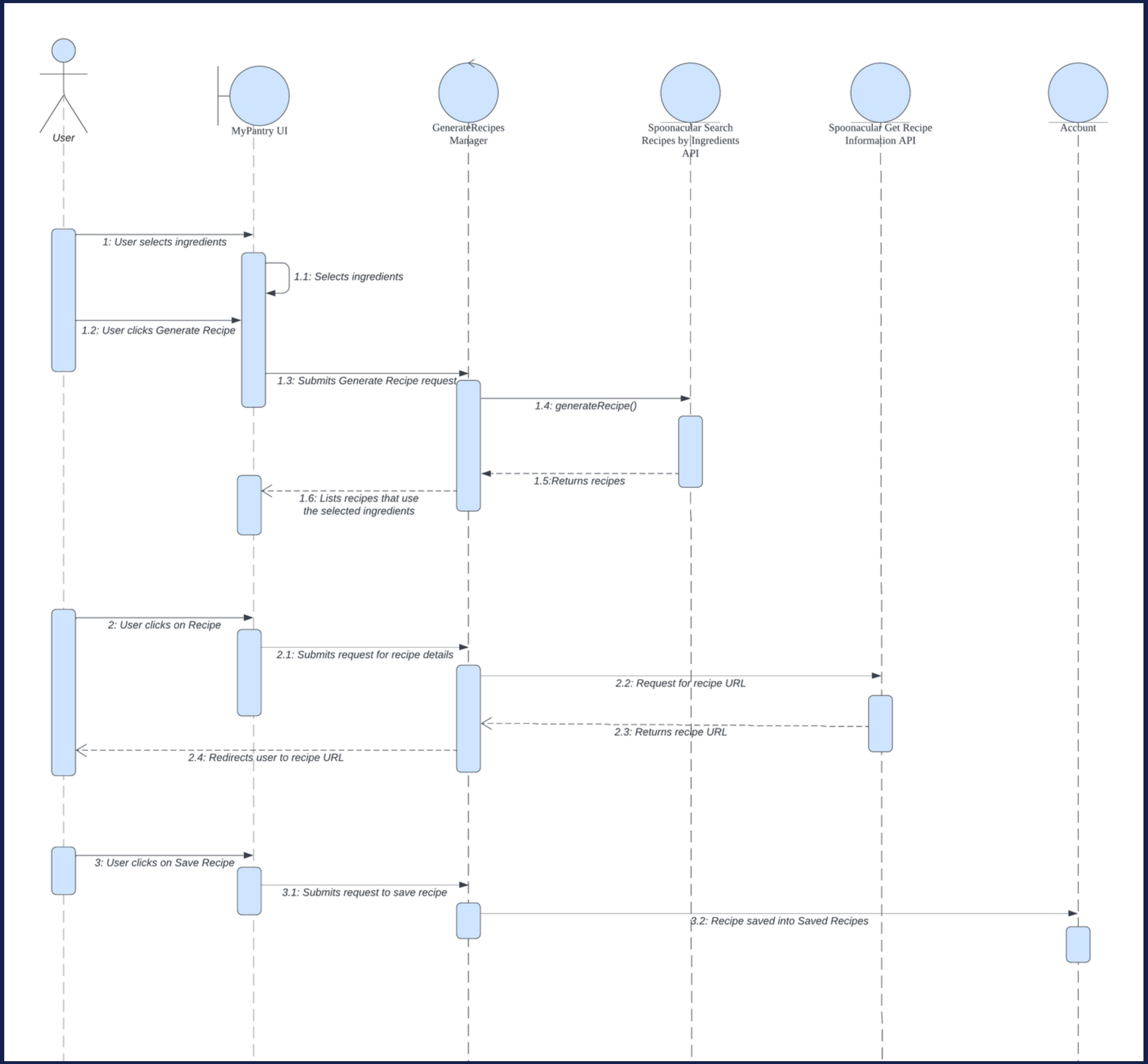
Revision History

Name	Date	Reason For Changes	Version
Mavis Khoo	29/1/2023	Created basic use cases.	1.0
Ie Jia Sheng	30/1/2023	Add in Use Case Diagram.	1.1
Mavis Khoo	12/2/2023	Added functions included in MyProfile page. Made minor changes to Use Case Descriptions.	2.0
Ie Jia Sheng	12/2/2023	Edited Use Case Diagram.	2.1

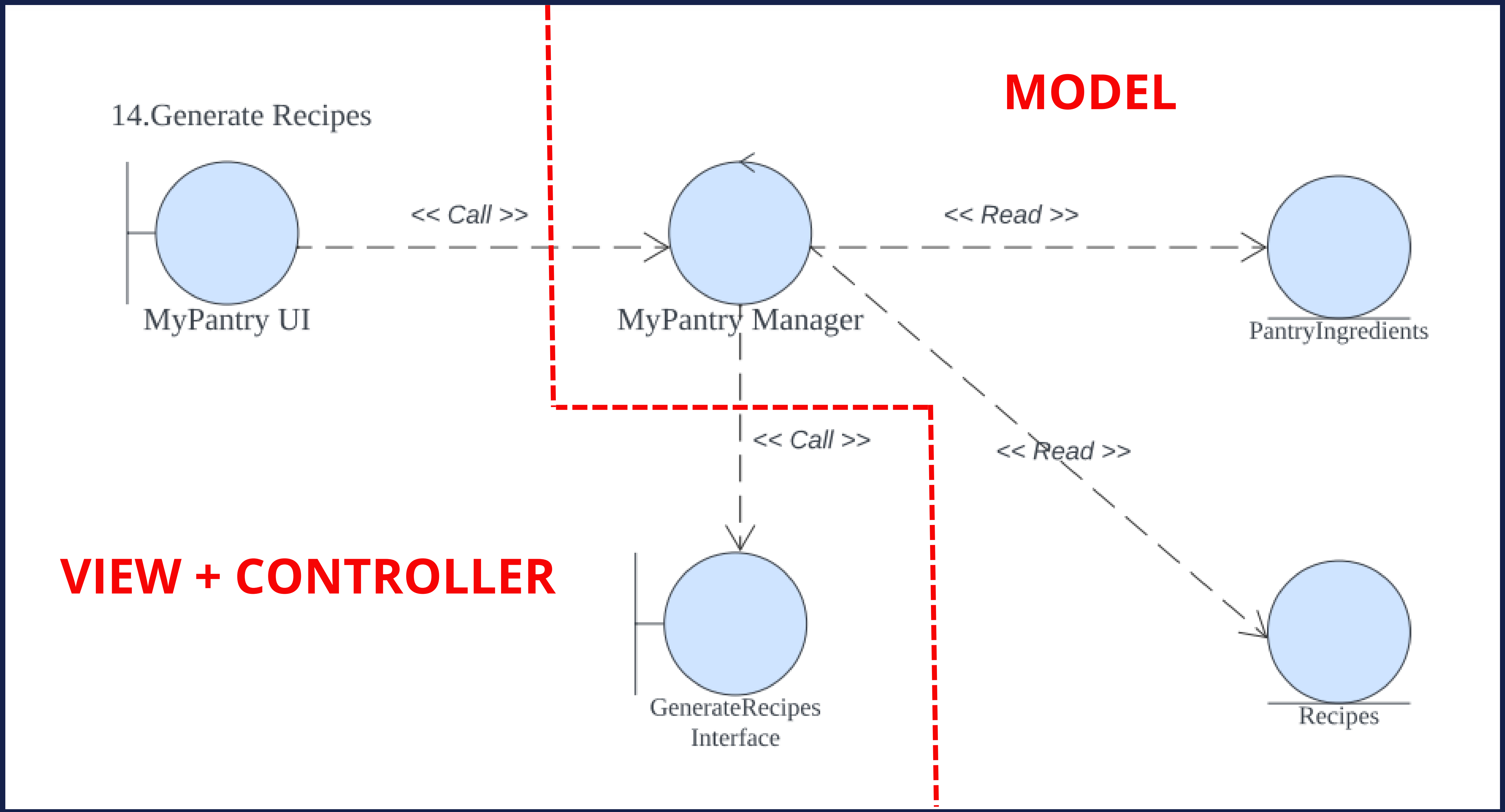
GENERATE RECIPE: CLASS DIAGRAM



GENERATE RECIPE: SEQUENCE DIAGRAM


User
MyPantry UI
GenerateRecipes Manager
Spoonacular Search Recipes by Ingredients API
Spoonacular Get Recipe Information API
Account
1: User selects ingredients
1.1: Selects ingredients
1.2: User clicks Generate Recipe
1.3: Submits Generate Recipe request
1.4: generateRecipe()
1.5:Returns recipes
1.6: Lists recipes that use the selected ingredients
2: User clicks on Recipe
2.1: Submits request for recipe details
2.2: Request for recipe URL
2.3: Returns recipe URL
2.4: Redirects user to recipe URL
3: User clicks on Save Recipe
3.1: Submits request to save recipe
3.2: Recipe saved into Saved Recipes

DESIGN PATTERN



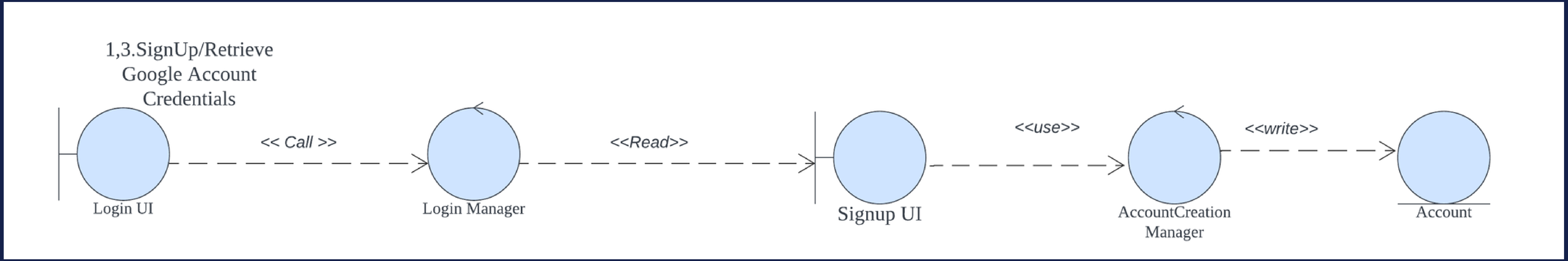
MVC Design Pattern

We separated the application into three parts: the model, the view and the controller.

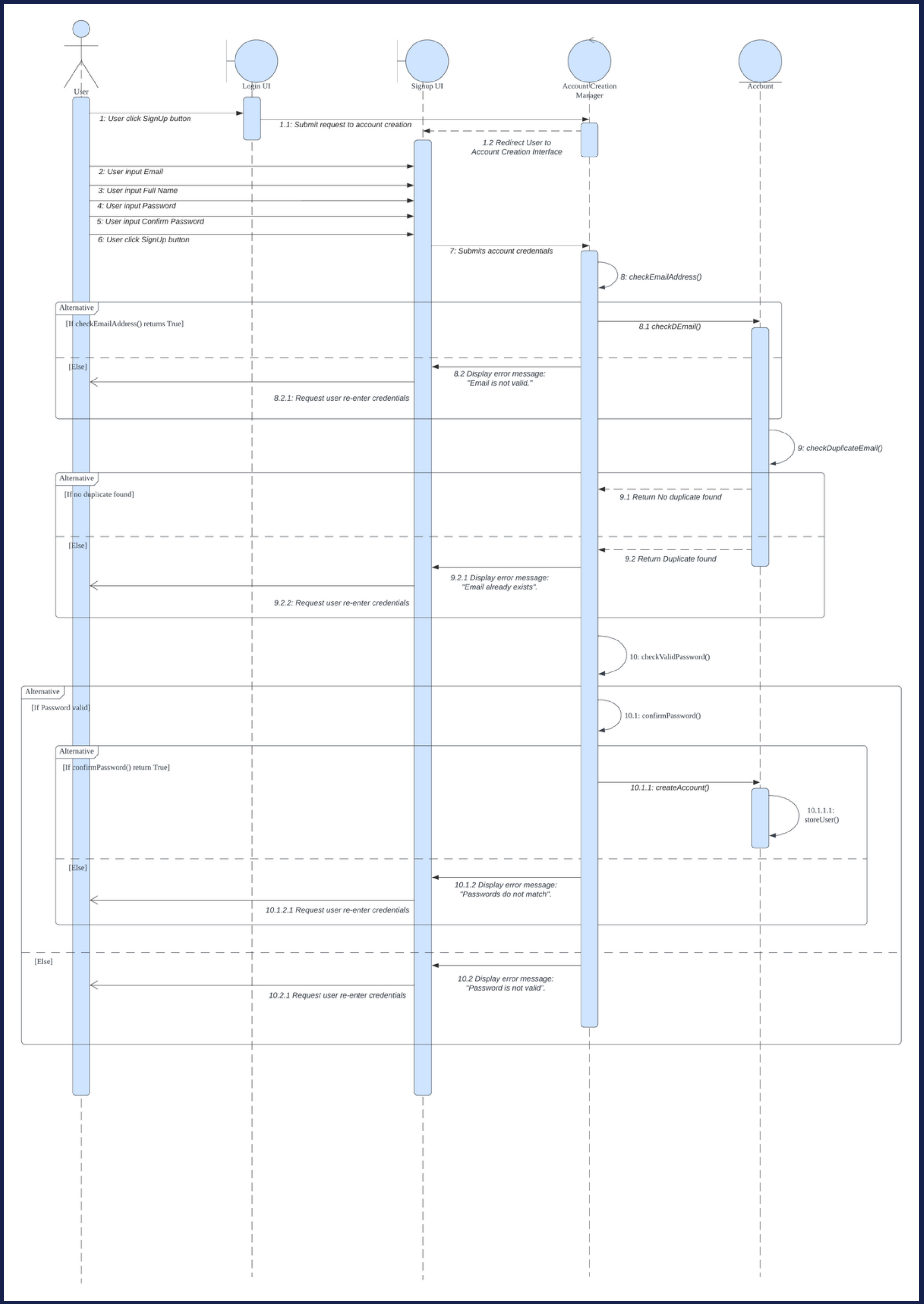
React supports the MVC pattern.

Model: eg setState() function
View: JSX file type
Controller: Event Handlers

ACCOUNT CREATION: CLASS DIAGRAM



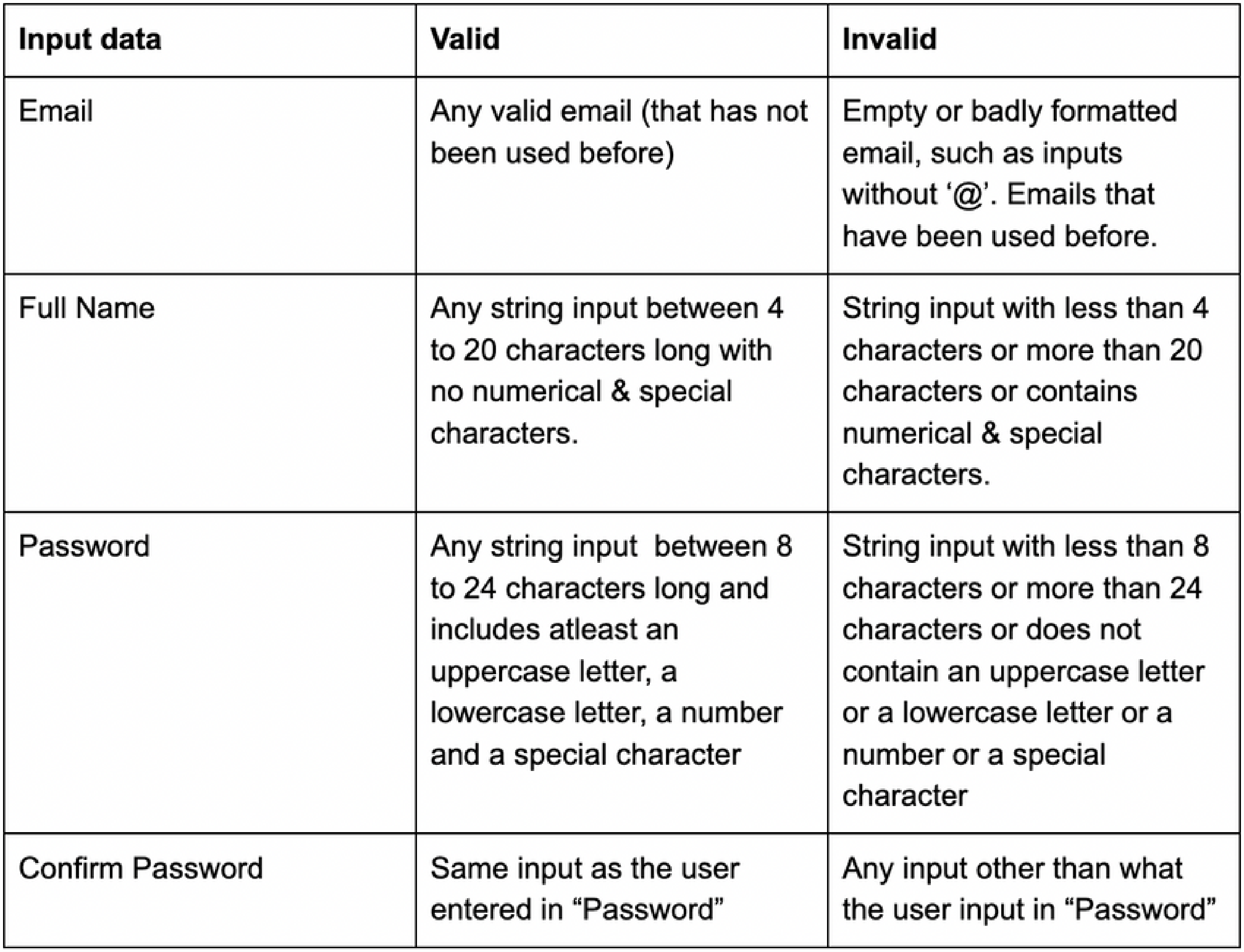
ACCOUNT CREATION: SEQUENCE DIAGRAM


User
Login UI
Signup UI
Account Creation Manager
Account
1: User click SignUp button
1.1: Submit request to account creation
1.2 Redirect User to Account Creation Interface
2: User input Email
3: User input Full Name
4: User input Password
5: User input Confirm Password
6: User click SignUp button
7: Submits account credentials
8: checkEmailAddress()
Alternative
[If checkEmailAddress() returns True]
8.1 checkDEmail()
[Else]
8.2 Display error message: "Email is not valid."
8.2.1: Request user re-enter credentials
9: checkDuplicateEmail()
Alternative
[If no duplicate found]
9.1 Return No duplicate found
[Else]
9.2 Return Duplicate found
9.2.1 Display error message: "Email already exists".
9.2.2: Request user re-enter credentials
10: checkValidPassword()
Alternative
[If Password valid]
10.1: confirmPassword()
Alternative
[If confirmPassword() return True]
10.1.1: createAccount()
10.1.1.1: storeUser()
[Else]
10.1.2 Display error message: "Passwords do not match".
10.1.2.1 Request user re-enter credentials
[Else]
10.2 Display error message: "Password is not valid".
10.2.1 Request user re-enter credentials

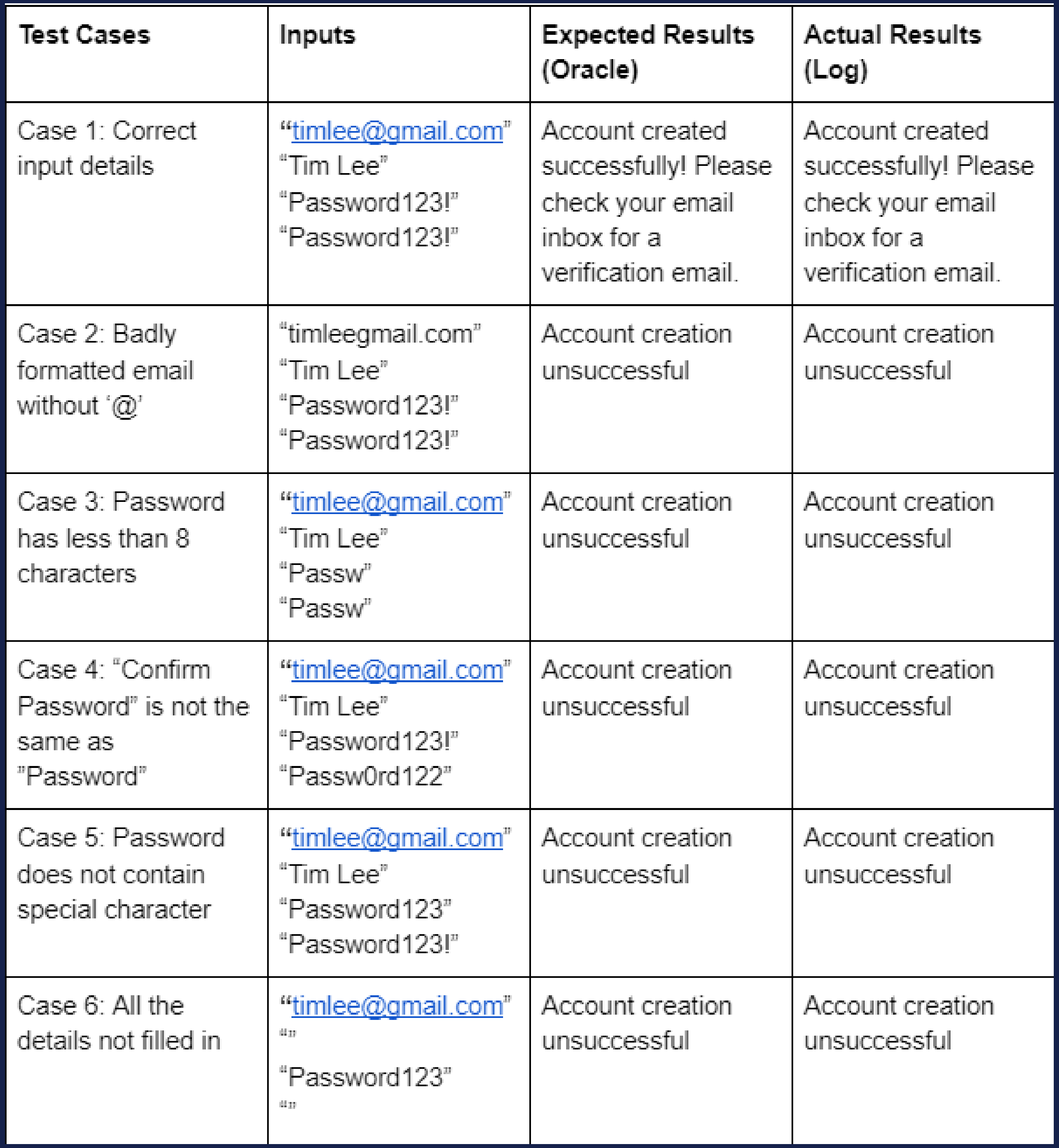
TESTING

Black Box

Test Case: Creating an Account



Input data	Valid	Invalid
Email	Any valid email (that has not been used before)	Empty or badly formatted email, such as inputs without ‘@’. Emails that have been used before.
Full Name	Any string input between 4 to 20 characters long with no numerical & special characters.	String input with less than 4 characters or more than 20 characters or contains numerical & special characters.
Password	Any string input between 8 to 24 characters long and includes atleast an uppercase letter, a lowercase letter, a number and a special character	String input with less than 8 characters or more than 24 characters or does not contain an uppercase letter or a lowercase letter or a number or a special character
Confirm Password	Same input as the user entered in “Password”	Any input other than what the user input in “Password”

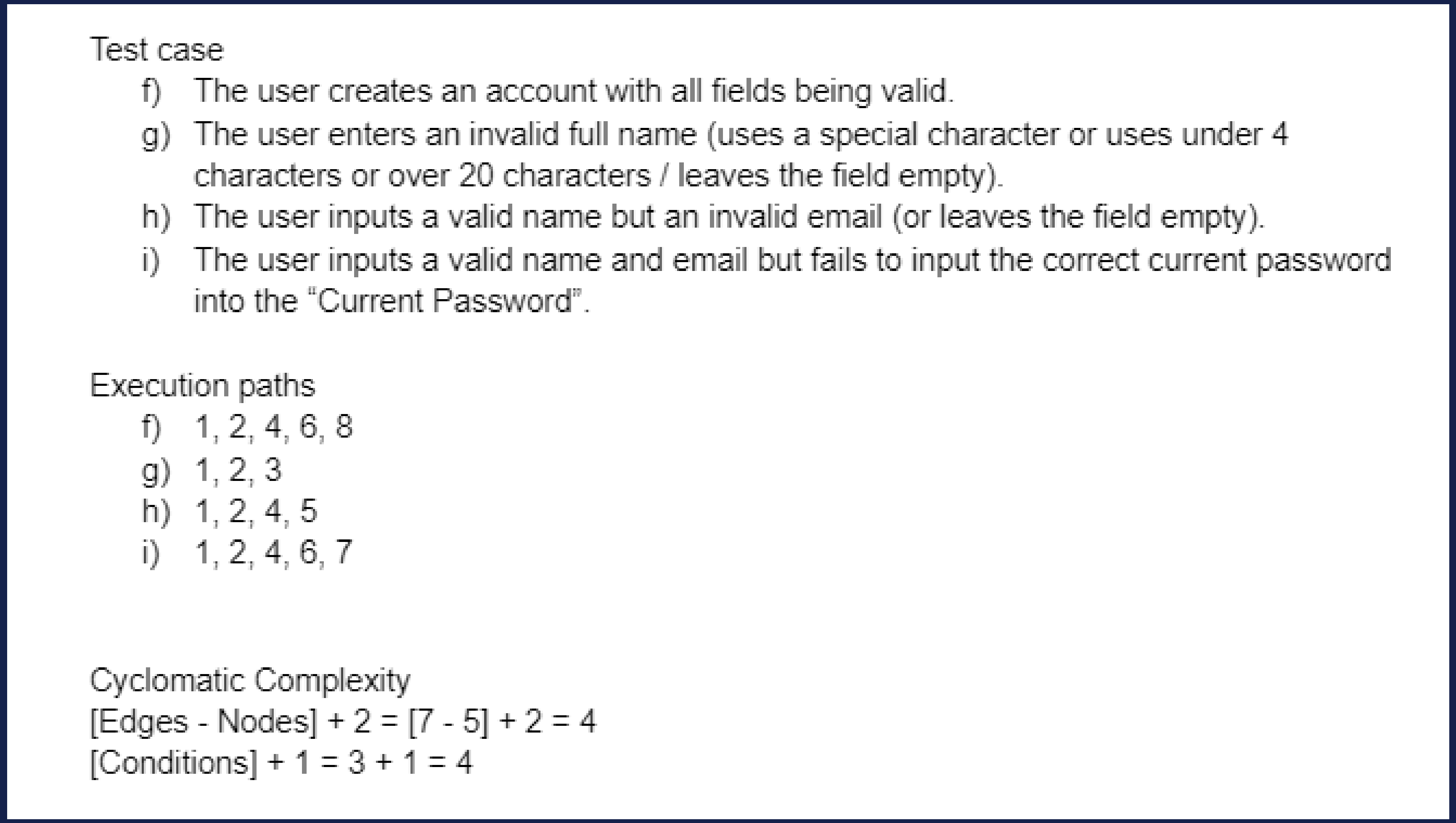


Test Cases	Inputs	Expected Results (Oracle)	Actual Results (Log)
Case 1: Correct input details	“timlee@gmail.com” “Tim Lee” “Password123!” “Password123!”	Account created successfully! Please check your email inbox for a verification email.	Account created successfully! Please check your email inbox for a verification email.
Case 2: Badly formatted email without ‘@’	“timleegmail.com” “Tim Lee” “Password123!” “Password123!”	Account creation unsuccessful	Account creation unsuccessful
Case 3: Password has less than 8 characters	“timlee@gmail.com” “Tim Lee” “Passw” “Passw”	Account creation unsuccessful	Account creation unsuccessful
Case 4: “Confirm Password” is not the same as ”Password”	“timlee@gmail.com” “Tim Lee” “Password123!” “Passw0rd122”	Account creation unsuccessful	Account creation unsuccessful
Case 5: Password does not contain special character	“timlee@gmail.com” “Tim Lee” “Password123” “Password123!”	Account creation unsuccessful	Account creation unsuccessful
Case 6: All the details not filled in	“timlee@gmail.com” “” “Password123” “”	Account creation unsuccessful	Account creation unsuccessful

TESTING

White Box

Test Case: Updating Profile



Test case

f) The user creates an account with all fields being valid.

g) The user enters an invalid full name (uses a special character or uses under 4 characters or over 20 characters / leaves the field empty).

h) The user inputs a valid name but an invalid email (or leaves the field empty).

i) The user inputs a valid name and email but fails to input the correct current password into the "Current Password".

Execution paths

f) 1, 2, 4, 6, 8

g) 1, 2, 3

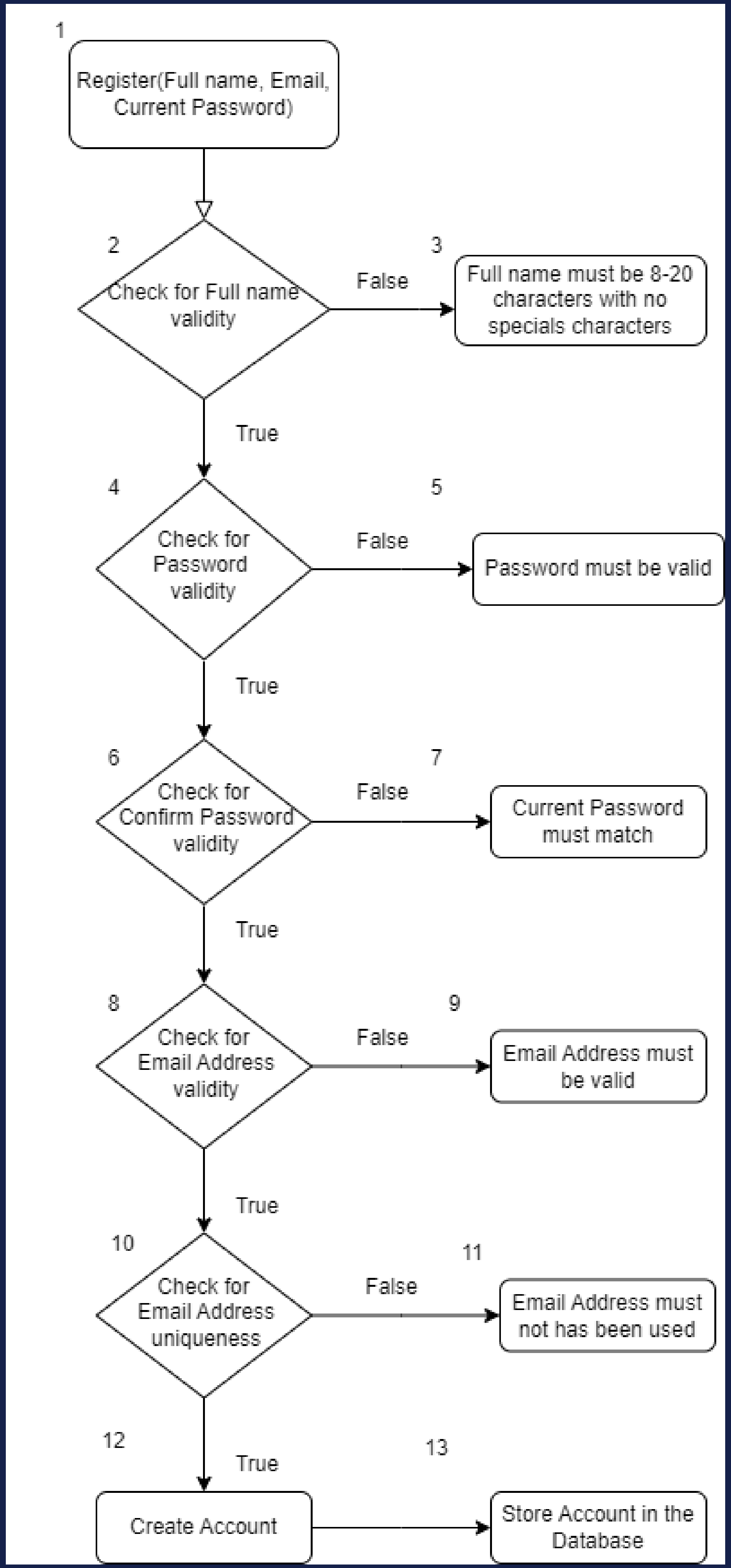
h) 1, 2, 4, 5

i) 1, 2, 4, 6, 7

Cyclomatic Complexity

[Edges - Nodes] + 2 = [7 - 5] + 2 = 4

[Conditions] + 1 = 3 + 1 = 4


1
Register(Full name, Email, Current Password)
2
Check for Full name validity
False
3
Full name must be 8-20 characters with no specials characters
True
4
Check for Password validity
False
5
Password must be valid
True
6
Check for Confirm Password validity
False
7
Current Password must match
True
8
Check for Email Address validity
False
9
Email Address must be valid
True
10
Check for Email Address uniqueness
False
11
Email Address must not has been used
12
True
13
Create Account
Store Account in the Database

THANK YOU