HAWKAR

Redefining gastronomical experiences

THANT HTOO AUNG (U2220809L)
SAENG-NIL NATTHAKAN (U2220832B)
CAO JUN MING (U2310254A)
MUHAMMAD ALIFF AMIRUL BIN MOHAMMED ARIFF (U2322581A)
KOW ZI TING (U2310485B)

TABLE OF CONTENTS

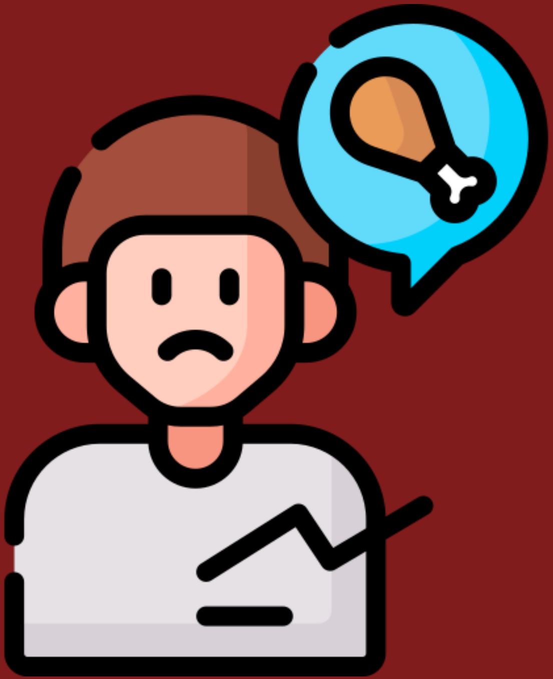
TABLE OF CONTENTS

01

the PROBLEM

more food, less information?

IMAGINE:



You are hungry.

But you have no idea where to eat.

Or what to eat.

IMAGINE:

You want to eat something **affordable**, and **not too far away**. But where?

THE PROBLEM:

That is the problem.

It takes an astronomical amount of time to find our favourite gastronomical delights.

THE PROBLEM:

That's why we have created **Hawkar.**

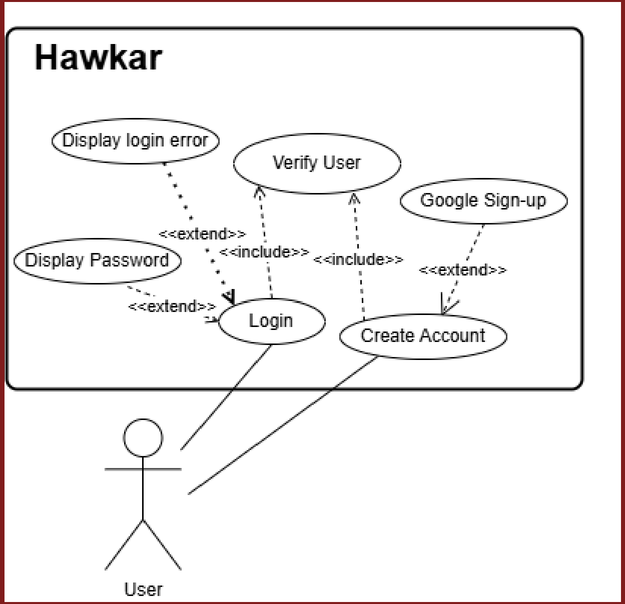
To help consumers find their favourite food, and hawkers reach out to customers.

02

target USERS

Consumers, Hawkers, Admins

USER (base class)



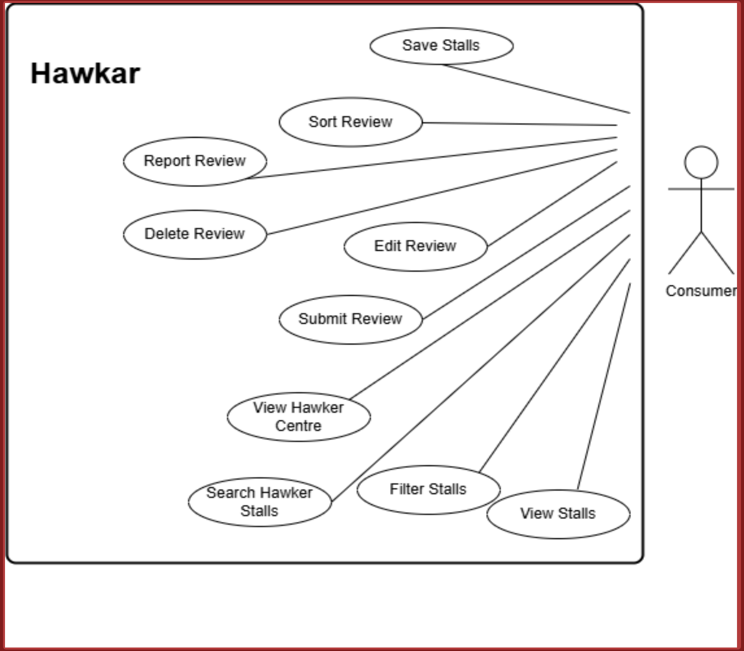
create account

Through Google Sign-up or otherwise

login

display password

CONSUMERS



view
Hawker Centres, Stalls

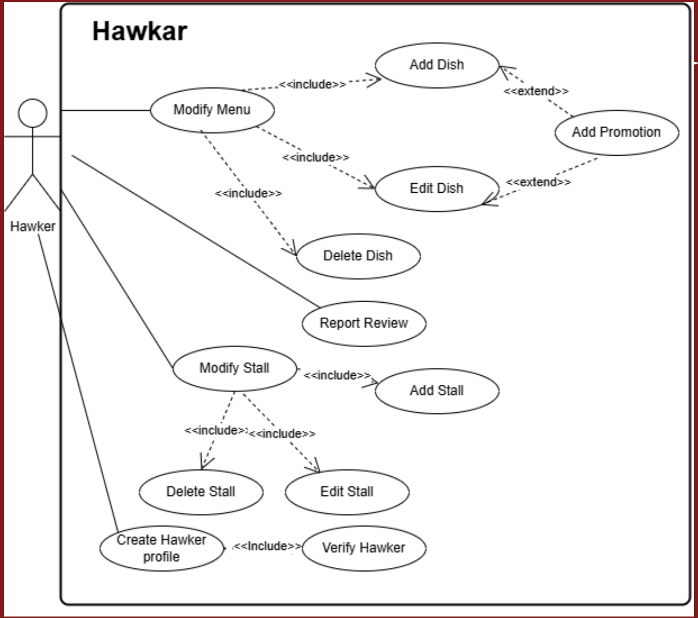
save
Stalls

filter
Stalls by Operating Hours/Price Range/Location/Hygiene Ratings

add, edit, delete, report
Stall Reviews

sort (WIP)
Reviews by Recency/Rating Values

HAWKERS



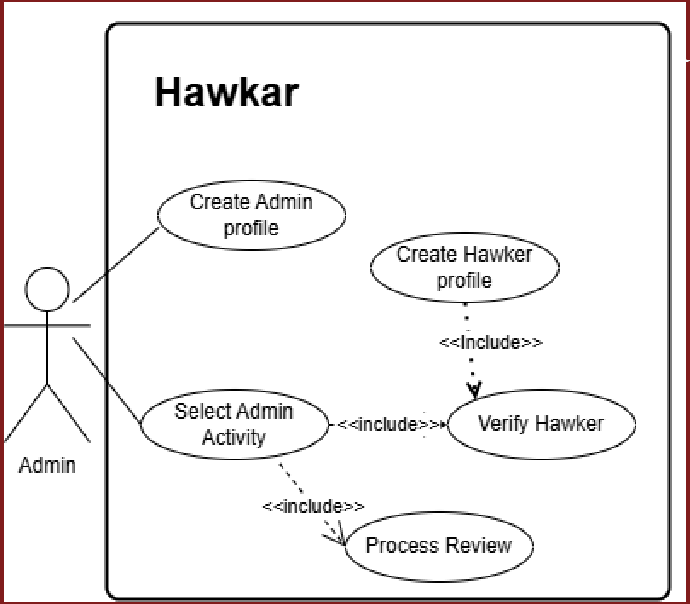
add, edit, delete

Dishes, Stalls, Promotions

report

Reviews

ADMINS



verify
Hawker Accounts

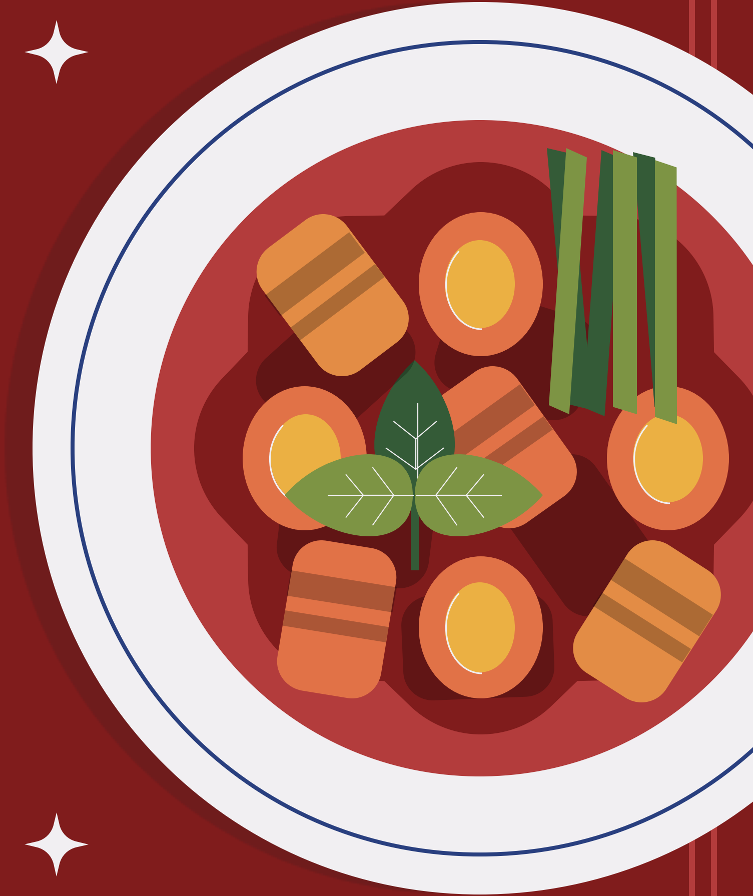
process
Reviews

03

DEMO

04

SE Practices



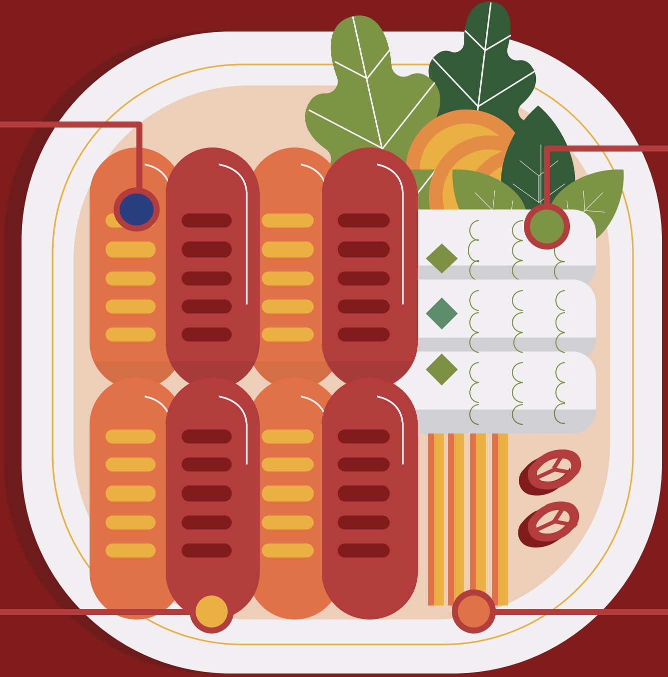
SEPARATION OF CONCERNS

enhances
MODULARITY

enhances
SCALABILITY

enhances
MAINTAINABILITY

decreases
COUPLING



LAYERED ARCHITECTURE

Presentation

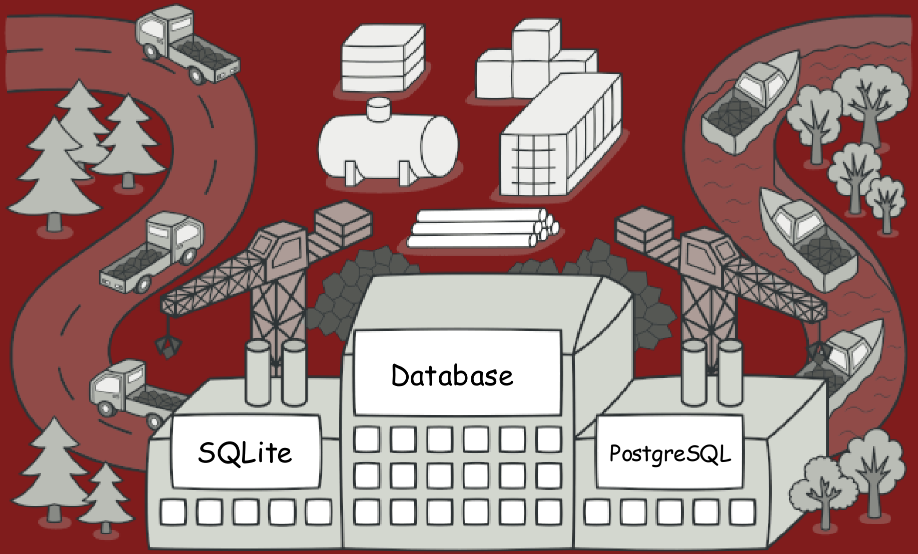
Application

Object

Persistence

Image made by Stockio from www.flaticon.com

FACTORY PATTERN

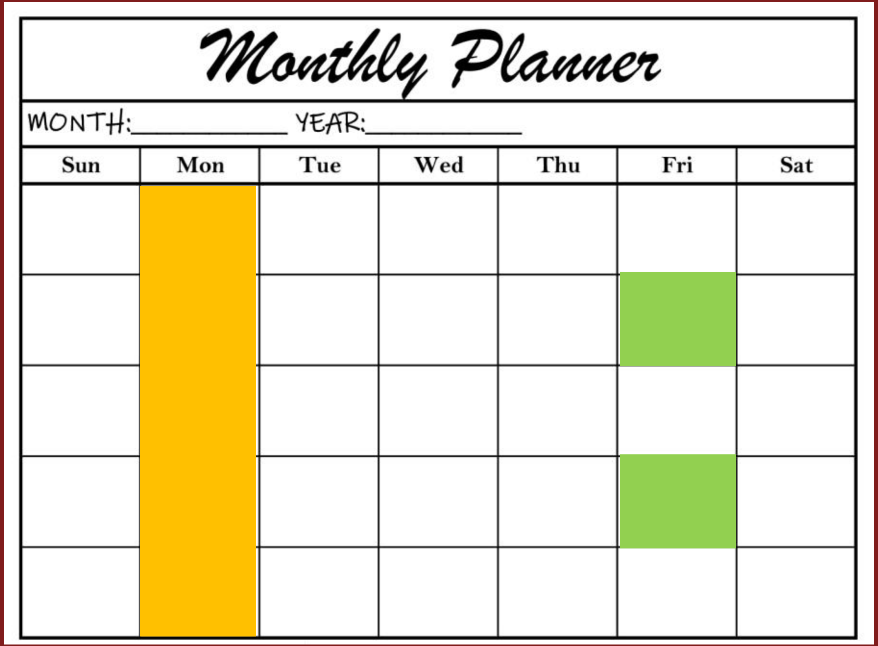


SQLite
database

PostgreSQL
database

Image is taken from https://refactoring.guru/design-patterns/factory-method

SCRUM



Monthly Planner

MONTH:__________ YEAR:__________

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	🟨					
	🟨				🟩	
	🟨					
	🟨				🟩	
	🟨					

Start
Sprint Planning

🟨 **Every Monday**
Standup

🟩 **2 weeks**
Sprint Review

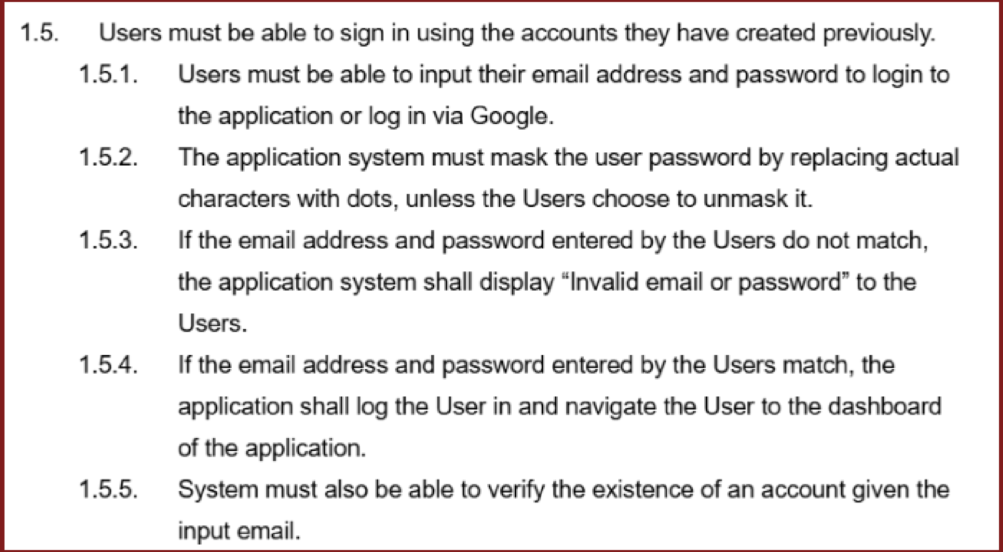
05

Traceability

Lifeline of the login function

LOGIN

Functional requirements

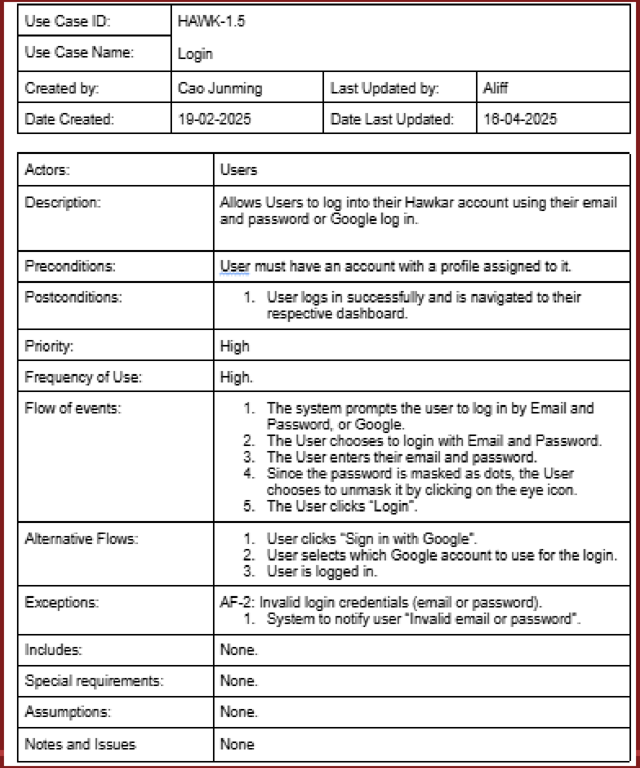


1.5. Users must be able to sign in using the accounts they have created previously.

- 1.5.1. Users must be able to input their email address and password to login to the application or log in via Google.
- 1.5.2. The application system must mask the user password by replacing actual characters with dots, unless the Users choose to unmask it.
- 1.5.3. If the email address and password entered by the Users do not match, the application system shall display “Invalid email or password” to the Users.
- 1.5.4. If the email address and password entered by the Users match, the application shall log the User in and navigate the User to the dashboard of the application.
- 1.5.5. System must also be able to verify the existence of an account given the input email.

LOGIN

Use Case Description

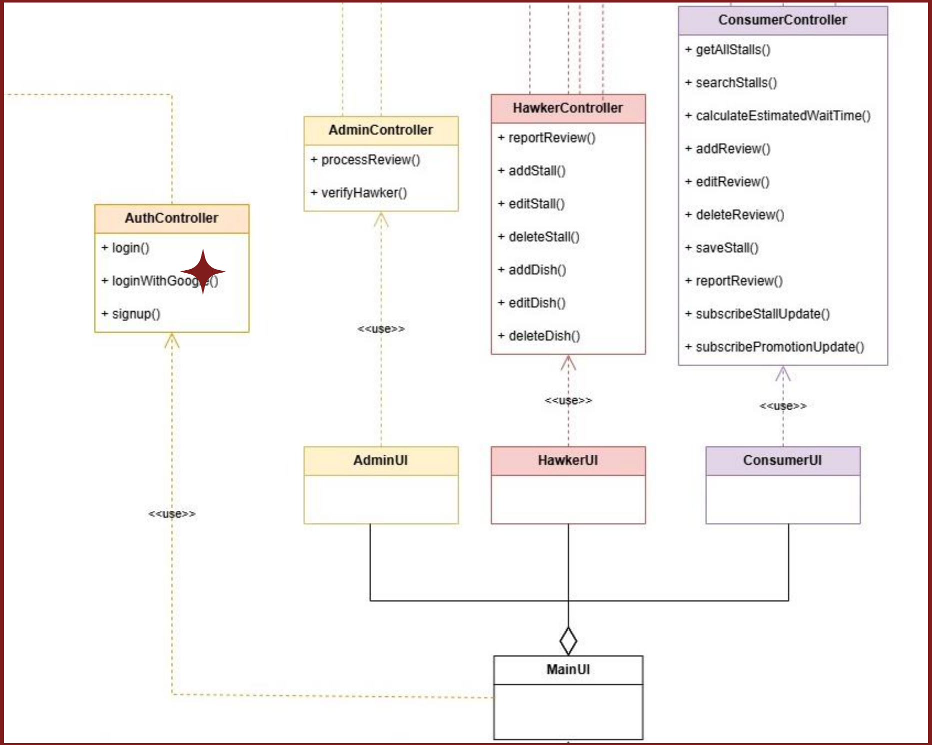


Use Case ID:	HAWK-1.5		
Use Case Name:	Login		
Created by:	Cao Junming	Last Updated by:	Aliff
Date Created:	19-02-2025	Date Last Updated:	18-04-2025

Actors:	Users
Description:	Allows Users to log into their Hawkar account using their email and password or Google log in.
Preconditions:	User must have an account with a profile assigned to it.
Postconditions:	1. User logs in successfully and is navigated to their respective dashboard.
Priority:	High
Frequency of Use:	High.
Flow of events:	1. The system prompts the user to log in by Email and Password, or Google. 2. The User chooses to login with Email and Password. 3. The User enters their email and password. 4. Since the password is masked as dots, the User chooses to unmask it by clicking on the eye icon. 5. The User clicks "Login".
Alternative Flows:	1. User clicks "Sign in with Google". 2. User selects which Google account to use for the login. 3. User is logged in.
Exceptions:	AF-2: Invalid login credentials (email or password). 1. System to notify user "Invalid email or password".
Includes:	None.
Special requirements:	None.
Assumptions:	None.
Notes and Issues	None

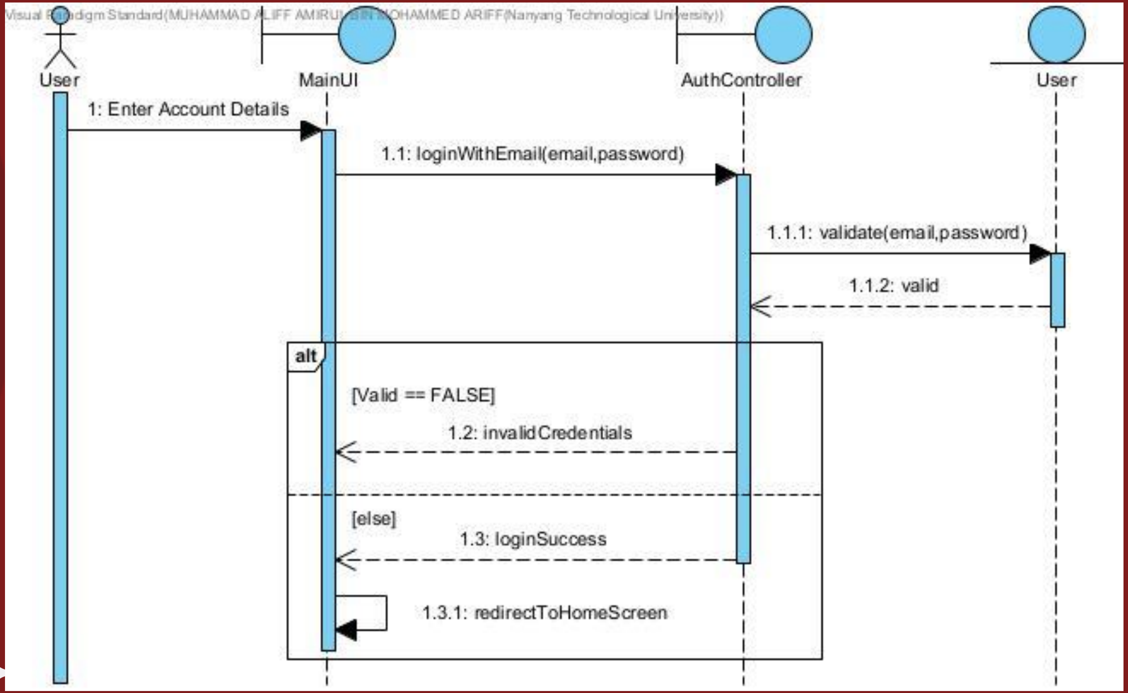
LOGIN

Class Diagram



LOGIN

Sequence Diagram


Visual Paradigm Standard(MUHAMMAD ALIFF AMIRUL BIN MOHAMMED ARIFF(Nanyang Technological University))
User
MainUI
AuthController
User
1: Enter Account Details
1.1: loginWithEmail(email,password)
1.1.1: validate(email,password)
1.1.2: valid
alt
[Valid == FALSE]
1.2: invalidCredentials
[else]
1.3: loginSuccess
1.3.1: redirectToHomeScreen

LOGIN

Design Considerations

Functional

Ensure singular login state throughout server

Isolate user privileges according to different roles

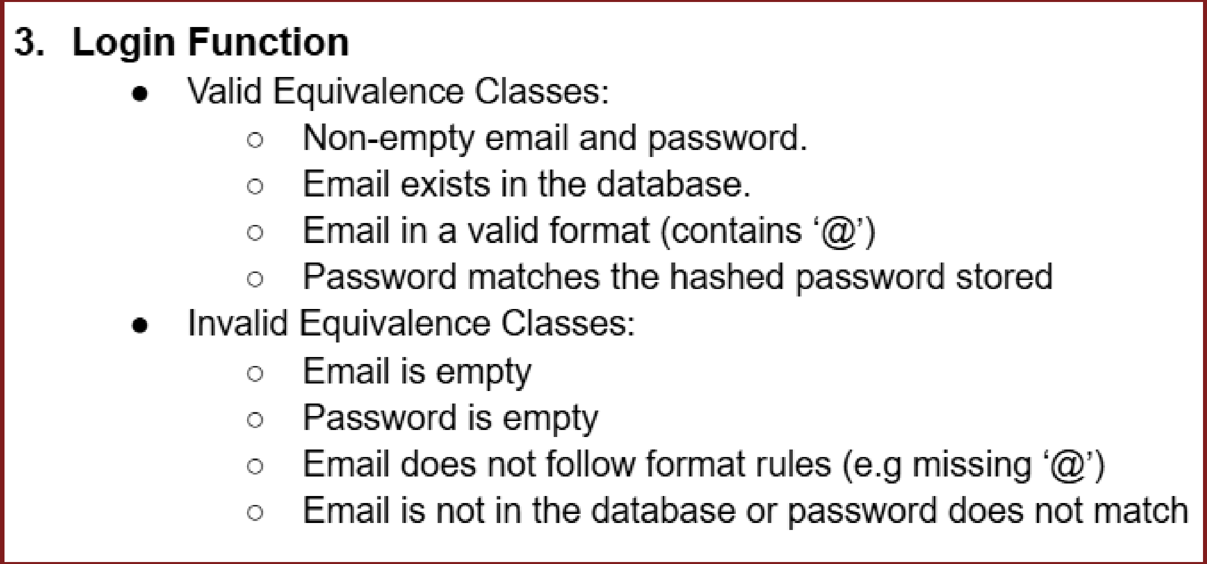
Non-Functional

Decrease the need for repeated logins by users

Image made by Kemalmoe from www.flaticon.com

LOGIN

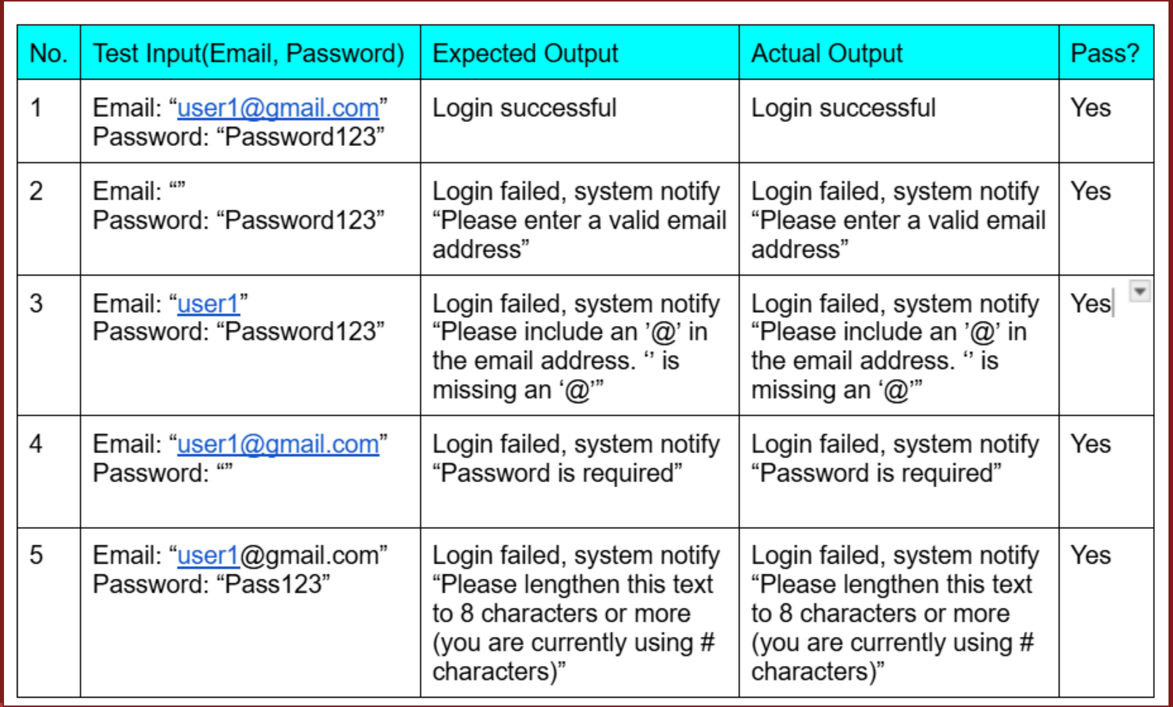
Testing Procedures – Equivalence Class Testing



3. **Login Function**
   - Valid Equivalence Classes:
     - Non-empty email and password.
     - Email exists in the database.
     - Email in a valid format (contains '@')
     - Password matches the hashed password stored
   - Invalid Equivalence Classes:
     - Email is empty
     - Password is empty
     - Email does not follow format rules (e.g missing '@')
     - Email is not in the database or password does not match

LOGIN

Testing Procedures – Boundary Value Testing



No.	Test Input(Email, Password)	Expected Output	Actual Output	Pass?
1	Email: "user1@gmail.com" Password: "Password123"	Login successful	Login successful	Yes
2	Email: "" Password: "Password123"	Login failed, system notify "Please enter a valid email address"	Login failed, system notify "Please enter a valid email address"	Yes
3	Email: "user1" Password: "Password123"	Login failed, system notify "Please include an '@' in the email address. '' is missing an '@'"	Login failed, system notify "Please include an '@' in the email address. '' is missing an '@'"	Yes
4	Email: "user1@gmail.com" Password: ""	Login failed, system notify "Password is required"	Login failed, system notify "Password is required"	Yes
5	Email: "user1@gmail.com" Password: "Pass123"	Login failed, system notify "Please lengthen this text to 8 characters or more (you are currently using # characters)"	Login failed, system notify "Please lengthen this text to 8 characters or more (you are currently using # characters)"	Yes

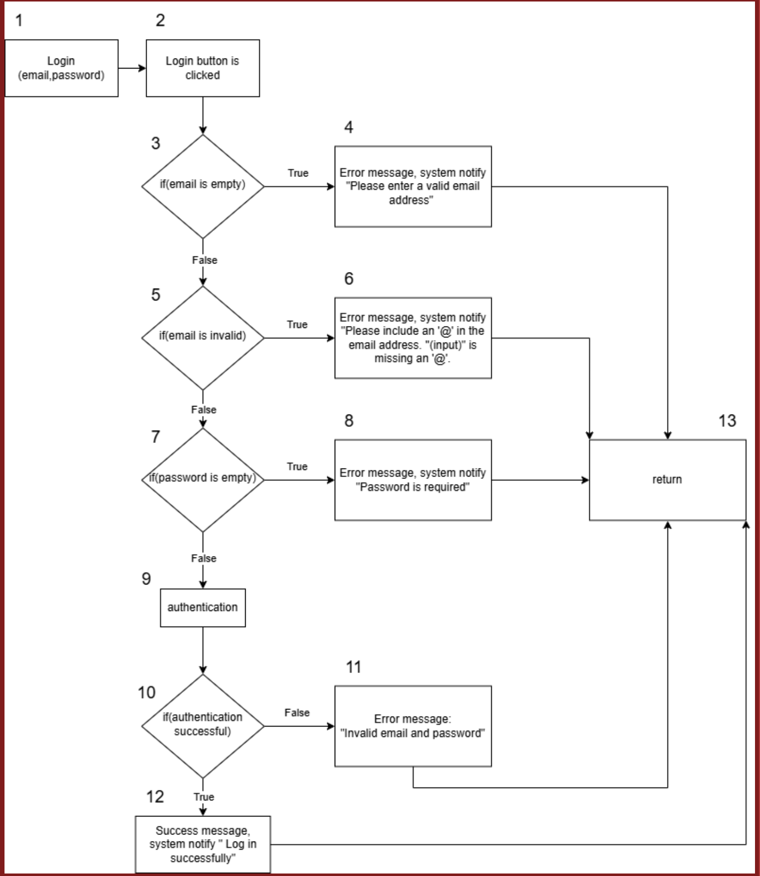
LOGIN

Testing Procedures – White Box Testing

Cyclomatic Complexity: 6

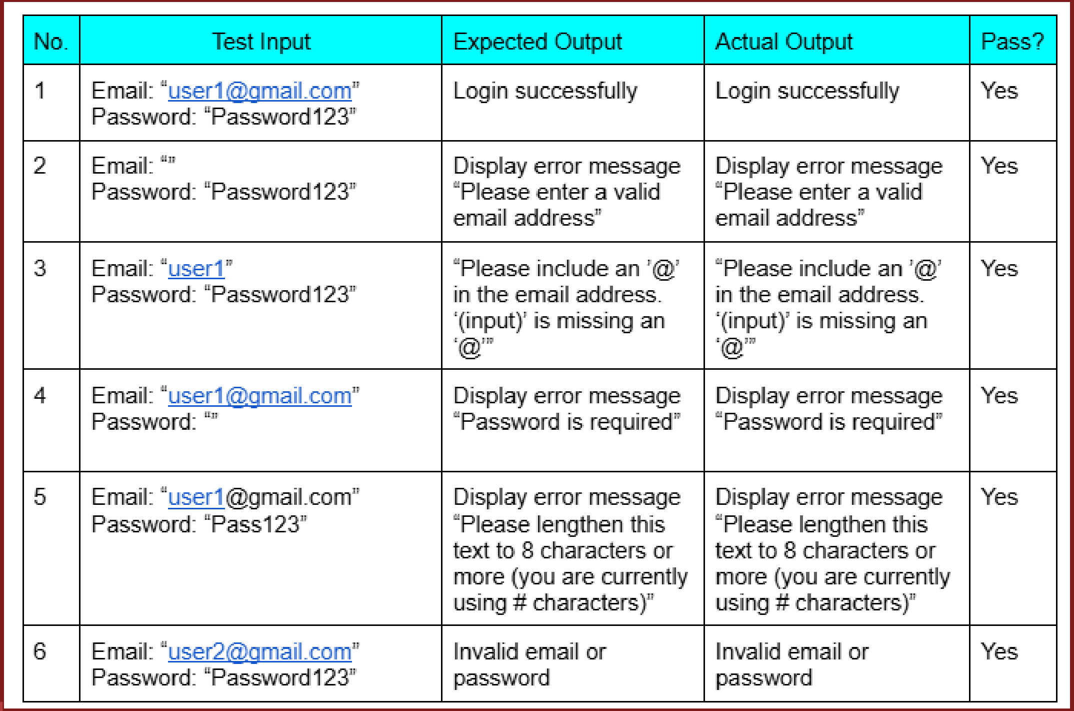
Basis Path:

Baseline path: 1, 2, 3, 5, 7, 9, 11, 12, 14 15
Basis path 2: 1, 2, 3, 4, 15
Basis path 3: 1, 2, 3, 5, 6, 15
Basis path 4: 1, 2, 3, 5, 7, 8, 15
Basis path 5: 1, 2, 3, 5, 7, 9, 10, 15
Basis path 6: 1, 2, 3, 5, 7, 9, 11 , 12, 13, 15


1
Login (email,password)
2
Login button is clicked
3
if(email is empty)
True
4
Error message, system notify "Please enter a valid email address"
False
5
if(email is invalid)
True
6
Error message, system notify "Please include an '@' in the email address. "(input)" is missing an '@'.
False
7
if(password is empty)
True
8
Error message, system notify "Password is required"
13
return
False
9
authentication
10
if(authentication successful)
False
11
Error message: "Invalid email and password"
12
True
Success message, system notify " Log in successfully"

LOGIN

Testing Procedures – White Box Testing



No.	Test Input	Expected Output	Actual Output	Pass?
1	Email: “user1@gmail.com” Password: “Password123”	Login successfully	Login successfully	Yes
2	Email: “” Password: “Password123”	Display error message “Please enter a valid email address”	Display error message “Please enter a valid email address”	Yes
3	Email: “user1” Password: “Password123”	“Please include an ’@’ in the email address. ‘(input)’ is missing an ‘@’”	“Please include an ’@’ in the email address. ‘(input)’ is missing an ‘@’”	Yes
4	Email: “user1@gmail.com” Password: “”	Display error message “Password is required”	Display error message “Password is required”	Yes
5	Email: “user1@gmail.com” Password: “Pass123”	Display error message “Please lengthen this text to 8 characters or more (you are currently using # characters)”	Display error message “Please lengthen this text to 8 characters or more (you are currently using # characters)”	Yes
6	Email: “user2@gmail.com” Password: “Password123”	Invalid email or password	Invalid email or password	Yes

Thank You

Have a great day!

References

Factory method. (2023, January 1). Refactoring and Design Patterns.

https://refactoring.guru/design-patterns/factory-method

(n.d.). Flaticon. https://www.flaticon.com/