
Software Requirements Specification

for

<HawkerSG>

Version 1.0 approved

**Prepared by <Leck Kye-Cin, Lee Jun De, Kavan, Lee Yong Liang, Lee Zi An,
Lim Seow Kiat, Lim Xiao Xuan>**

<SC2006 - HawkerSG>

<15/11/2025>

Table of Contents

Table of Contents.....	2
1. Introduction.....	7
1.1 Purpose.....	7
1.2 Document Conventions.....	7
1.3 Intended Audience and Reading Suggestions.....	7
1.4 Product Scope.....	8
1.5 References.....	8
2. Overall Description.....	9
2.1 Product Perspective.....	9
2.2 Product Functions.....	11
2.2.1 Use Case Diagram.....	11
2.2.2 Major Product Functions.....	11
2.2.3 Class Diagram.....	12
2.3 User Classes and Characteristics.....	12
2.3.1 Consumers.....	12
2.3.2 Business Owners.....	13
2.4 Operating Environment.....	13
2.5 Design and Implementation Constraints.....	13
2.5.1 Frontend Constraints.....	13
2.5.1.1 Framework (Frontend).....	13
2.5.1.2 Code Formatting.....	13
2.5.2 Backend Constraints.....	13
2.5.2.1 Framework (Backend).....	13
2.5.2.2 Database.....	13
2.5.2.3 Code Formatting.....	14
2.5.3 Language Constraints.....	14
2.6 User Documentation.....	14
2.6.1 README Files.....	14
2.6.1.1 Main Repository.....	14
2.6.1.2 Frontend Repository.....	14
2.6.1.3 Backend Repository.....	14
2.6.2 Code Comments.....	14
2.6.3 Backend API Documentation.....	14
2.7 Assumptions and Dependencies.....	15
2.7.1 Assumptions.....	15

2.7.2 Dependencies.....	15
2.7.2.1 Front-end Libraries.....	15
2.7.2.2 Back-end Libraries.....	15
2.7.2.3 External Services.....	15
3. External Interface Requirements.....	16
3.1 User Interfaces.....	16
3.1.1 Style Guides.....	16
3.1.1.1 Theme Colours.....	16
3.1.1.2 Containers and components.....	17
3.1.1.2.1 Cards.....	17
3.1.1.2.2 Buttons.....	17
3.1.1.2.3 Form fields and input characters.....	17
3.1.1.2.4 Image and media containers.....	17
3.1.1.2.5 Background and panels.....	17
3.1.1.3 Typography.....	18
3.1.2 UI Mockup.....	19
3.1.2.1 Home Page.....	19
3.1.2.2 Sign Up.....	20
3.1.2.2.1 Consumer View.....	20
3.1.2.2.2 Business View.....	20
3.1.2.3 Authentication (Login).....	21
3.1.2.3.1 Consumer View.....	21
3.1.2.3.2 Business View.....	21
3.1.2.4 Consumer Function.....	22
3.1.2.4.1 Browse.....	22
3.1.2.4.1.1 Hawker Centres.....	22
3.1.2.4.1.2 Hawker Stalls.....	22
3.1.2.4.2 Search Bar + Recent Searches.....	23
3.1.2.4.3 View Stall Profile.....	23
3.1.2.4.4 User Profile.....	24
3.1.2.4.5 Edit Profile.....	24
3.1.2.4.6 Location Permission.....	25
3.1.2.4.7 Hawkers Nearby.....	25
3.1.2.4.8 Upload Reviews.....	26
3.1.2.5 Business Profile (Functions).....	27
3.1.2.5.2 Edit Stall Profile.....	27
3.1.2.5.3 Edit Menu.....	28

3.1.2.5.4 Edit Operating Hours.....	28
3.1.2.5.5 View Reviews.....	29
3.1.2.5.6 Consumer View Preview.....	29
3.2 Hardware Interfaces.....	30
3.2.1 Device Types.....	30
3.2.2 Data and Control Interactions.....	30
3.2.3 Communication Protocols.....	30
3.3 Software Interfaces.....	30
3.3.1 Operating System.....	31
3.3.2 Web Browsers.....	31
3.3.3 Databases.....	31
3.3.4 Backend Tools.....	31
3.3.5 Frontend Tools.....	31
3.3.6 RESTful APIs.....	32
3.3.7 Data Sharing.....	32
3.4 Communications Interfaces.....	32
4. System Features.....	33
4.1 Authentication (Consumer & Business).....	33
4.1.1 Consumer Signup.....	33
4.1.1.1 Description and Priority.....	33
4.1.1.2 Stimulus/Response Sequences.....	33
4.1.1.3 Functional Requirements.....	34
4.1.2 Business Signup (CorpPass Verification).....	34
4.1.2.1 Description and Priority.....	34
4.1.2.2 Stimulus/Response Sequences.....	35
4.1.2.3 Functional Requirements.....	35
4.1.3 Login (Consumer & Business Owner).....	36
4.1.3.1 Description and Priority.....	36
4.1.3.2 Stimulus/Response Sequences.....	36
4.1.3.3 Functional Requirements.....	36
4.1.4 Password Reset.....	37
4.1.4.1 Description and Priority.....	37
4.1.4.2 Stimulus/Response Sequences.....	37
4.1.4.3 Functional Requirements.....	38
4.2 Consumer Features.....	39
4.2.1 Search (Hawker Centre & Hawker Stall).....	39
4.2.1.1 Description and Priority.....	39

4.2.1.2 Stimulus/Response Sequences.....	39
4.2.1.3 Functional Requirements.....	39
4.2.2 Recent Searches.....	41
4.2.2.1 Description and Priority.....	41
4.2.2.2 Stimulus/Response Sequences.....	41
4.2.2.3 Functional Requirements.....	41
4.2.3 Near Me.....	42
4.2.3.1 Description and Priority.....	42
4.2.3.2 Stimulus/Response Sequences.....	42
4.2.3.3 Functional Requirements.....	42
4.2.4 Favourites.....	43
4.2.4.1 Description and Priority.....	43
4.2.4.2 Stimulus/Response Sequences.....	43
4.2.4.3 Functional Requirements.....	43
4.2.5 Reviews.....	44
4.2.5.1 Description and Priority.....	44
4.2.5.2 Stimulus/Response Sequences.....	44
4.2.5.3 Functional Requirements.....	45
4.2.6 Consumer Profile Management.....	46
4.2.6.1 Description and Priority.....	46
4.2.6.2 Stimulus/Response Sequences.....	46
4.2.6.3 Functional Requirements.....	47
4.2.7 Browse.....	48
4.2.7.1 Description and Priority.....	48
4.2.7.2 Stimulus/Response Sequences.....	48
4.2.7.3 Functional Requirements.....	48
4.3 Business Owner Management Features.....	49
4.3.1 Overview.....	49
4.3.1.1 Description and Priority.....	49
4.3.1.2 Stimulus/Response Sequences.....	49
4.3.1.3 Functional Requirements.....	50
4.3.2 Edit Stall Profile.....	50
4.3.2.1 Description and Priority.....	50
4.3.2.2 Stimulus/Response Sequences.....	50
4.3.2.3 Functional Requirements.....	51
4.3.3 Manage Menu.....	51
4.3.3.1 Description and Priority.....	51

4.3.3.2 Stimulus/Response Sequences.....	51
4.3.3.3 Functional Requirements.....	52
4.3.4 Edit Operating Hours.....	53
4.3.4.1 Description and Priority.....	53
4.3.4.2 Stimulus/Response Sequences.....	53
4.3.4.3 Functional Requirements.....	54
4.3.5 View Reviews.....	54
4.3.5.1 Description and Priority.....	54
4.3.5.2 Stimulus/Response Sequences.....	54
4.3.5.3 Functional Requirements.....	55
4.3.6 Consumer View Preview.....	55
4.3.6.1 Description and Priority.....	55
4.3.6.2 Stimulus/Response Sequences.....	56
4.3.6.3 Functional Requirements.....	56
5. Other Nonfunctional Requirements.....	57
5.1 Performance Requirements.....	57
5.2 Safety Requirements.....	57
5.3 Security Requirements.....	57
5.4 Software Quality Attributes.....	57
5.5 Business Rules.....	58
6. Other Requirements.....	59
6.1 Internationalisation Requirements.....	59
6.2 Legal Requirements.....	59
6.3 Reuse Objectives.....	59
6.4 Accessibility Requirements.....	59
6.5 Database Requirements.....	60
6.6 Maintainability Requirements.....	60
6.7 Reliability Requirements.....	60

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document describes the software requirements for HawkerSG, a web-based platform designed to help consumers discover hawker centres and stalls across Singapore. They can view reviews, share live location for nearby recommendations, and interact with hawker businesses all within HawkerSG.

The purpose of this document is to establish a clear, complete, and verifiable description of the system requirements to be used by developers, testers, project supervisors, and stakeholders.

This SRS focuses exclusively on the software components of HawkerSG and does not cover external hardware, deployment infrastructure, or organisational policies.

1.2 Document Conventions

This document follows the formatting and structural conventions commonly used in IEEE Software Requirements Specifications. Section numbers are used to clearly identify major topics and subtopics. Key system components, modules, and terms are emphasised using **bold text** to improve readability.

Functional requirements presented in later sections are assigned unique identifiers to support traceability throughout development and testing. Any technical terminology or domain-specific concepts referenced in this SRS are defined in the Glossary (Appendix A) to ensure consistent interpretation across all stakeholders.

1.3 Intended Audience and Reading Suggestions

This SRS is intended for stakeholders involved in the development and evaluation of the HawkerSG system, including software developers, testers, designers, project managers, and academic staff overseeing the SC2006 project. Developers will use this document to understand the system's expected behaviour, while testers will rely on the requirements to design and execute test cases. Teaching staff may refer to this SRS when reviewing the team's design and implementation.

New readers should begin with Section 1 for an overview of the system's purpose and scope. Section 2 provides high-level descriptions of the system environment, user classes, and constraints. Readers involved in implementation should focus on Section 4, which contains the detailed functional requirements. Supporting diagrams and analysis models located in the appendices may be consulted for additional clarity.

1.4 Product Scope

HawkerSG is a web-based platform designed to assist consumers in discovering hawker centres and food stalls across Singapore. The system provides key functions such as stall search, browsing by hawker centre, filtering by cuisine or rating, viewing live location recommendations, managing favourite stalls, and creating or retrieving user reviews. Stall owners are able to maintain their stall profiles, including descriptions, operating hours, and menu items.

The primary goal of HawkerSG is to present consolidated, reliable, and easily accessible information about hawker establishments, bridging the gap between consumers and hawkers. By enabling structured search, review, and discovery mechanisms, HawkerSG supports the preservation and promotion of local hawker culture through a modern digital platform. The system also incorporates an LLM-based review filtering mechanism to ensure that submitted reviews are appropriate and aligned with platform standards.

1.5 References

API & Technical Documentation

1. OneMap Singapore. OneMap REST API Documentation.

<https://www.onemap.gov.sg/docs/>

2. GovTech Singapore. CorpPass Authentication Overview.

<https://www.corppass.gov.sg>

3. OpenAI. Content Moderation & Safety API Documentation.

<https://platform.openai.com/docs/guides/moderation>

4. FastAPI. FastAPI Framework Documentation.

<https://fastapi.tiangolo.com/>

5. ReactJS. React Official Documentation.

<https://react.dev/>

6. Twilio. SendGrid Official Documentation

<https://www.twilio.com/docs/sendgrid/api-reference>

Software Engineering & OOP References

7. IEEE. IEEE Recommended Practice for Software Requirements Specifications (IEEE 830-1998).

<https://ieeexplore.ieee.org/document/720574>

8. Khalil Stemmler. Object-Oriented Programming Principles.

<https://khalilstemmler.com/articles/object-oriented/programming/4-principles/>

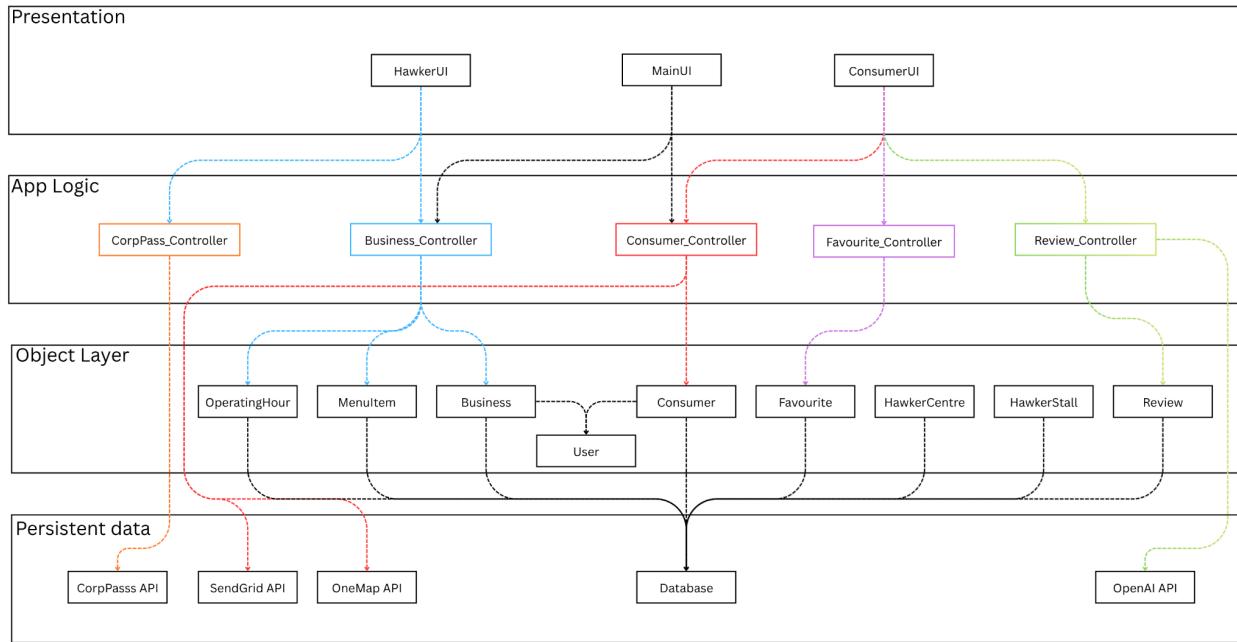
2. Overall Description

2.1 Product Perspective

HawkerSG is a standalone, web-based application designed to unify and streamline the discovery of hawker centres and food stalls in Singapore. It consolidates information traditionally dispersed across multiple sources into a single integrated platform. The system does not extend or replace any existing software; rather, it functions as an independent product that offers search, browsing, review management, favourites tracking, and stall owner updates.

HawkerSG interacts primarily with its backend service and internal database, which together manage user accounts, stall information, reviews, and system logic. It also integrates with external services such as the LLM-based moderation API for review validation.

A system architecture diagram (as shown below) is provided to illustrate the major system components, their interconnections, and any relevant external interfaces.

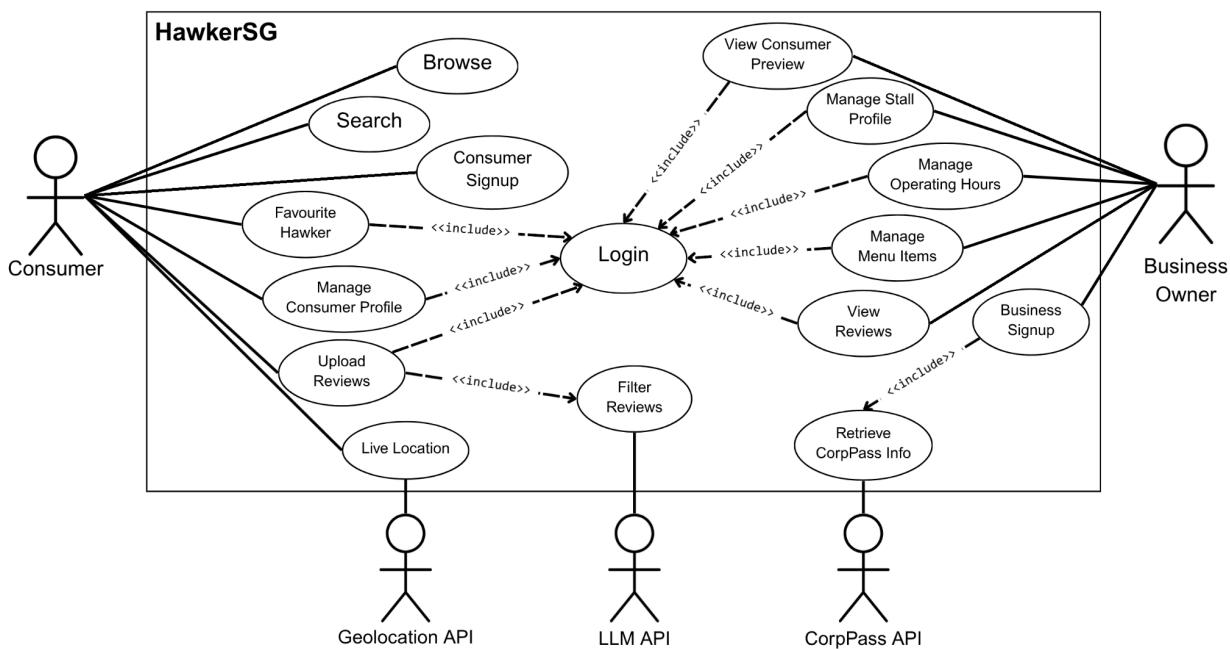


2.2 Product Functions

2.2.1 Use Case Diagram

The use case diagram provides a high-level overview of how the different user groups interact with HawkerSG. It illustrates the core system functions available to consumers and business owners, including search, browsing, reviews, favourites management, and stall profile maintenance.

The diagram (shown below) summarizes the system's primary services and their relationships to each actor to support a clear understanding of overall system behaviour.



2.2.2 Major Product Functions

HawkerSG offers several major functional capabilities to support consumer discovery and stall owner management. These include:

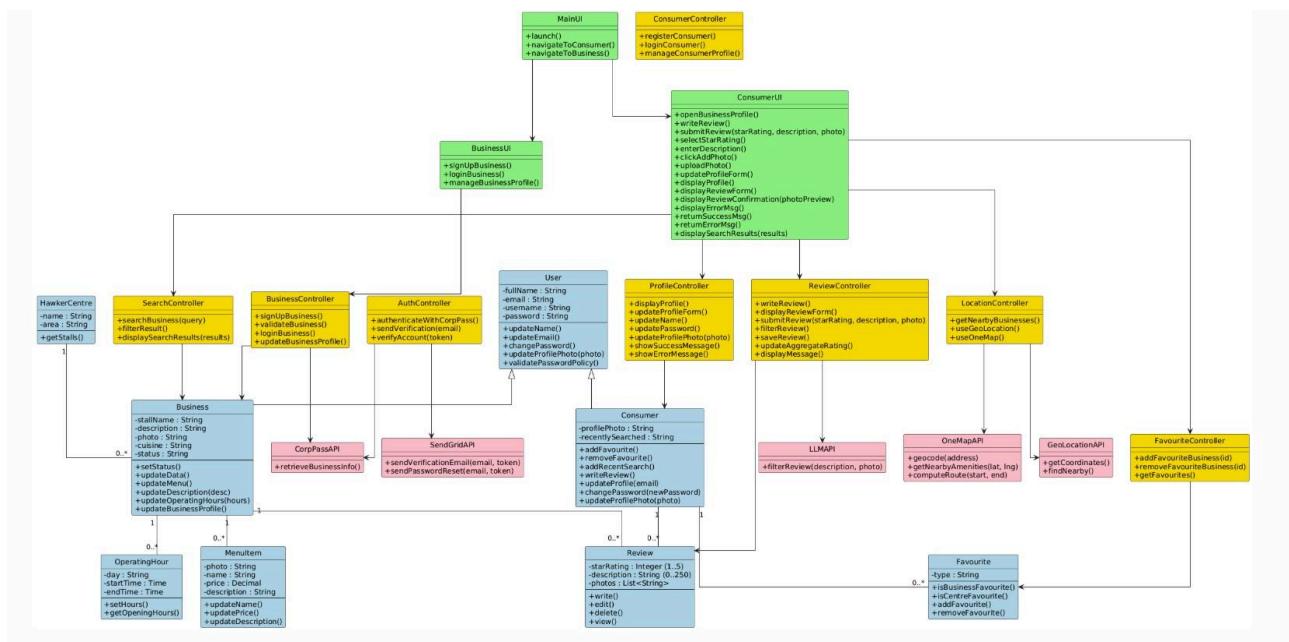
- Searching for hawker stalls and hawker centres.
- Browsing hawker centres and viewing stalls under each centre.
- Filtering stalls by cuisine, rating, and status.
- Viewing recommendations based on the user's live location.
- Submitting, retrieving, and deleting reviews.
- Managing a list of favourite stalls.
- Allowing business owners to update stall descriptions, menus, and operating hours.
- Moderating reviews using an external LLM-based service.
- Supporting authentication for both consumer and business accounts.

More detailed, feature-specific requirements are documented in Section 4 of this SRS.

2.2.3 Class Diagram

The class diagram presents the structural design of HawkerSG by illustrating the main system entities, their attributes, and the relationships between them. It reflects the underlying data model used by both the backend and frontend layers.

This diagram helps developers and reviewers understand how information is organised within the system and how objects interact during runtime.



2.3 User Classes and Characteristics

HawkerSG is designed for two main user classes. Each class has distinct characteristics, goals, and access privileges.

2.3.1 Consumers

Consumers represent general users who browse, search, and interact with hawker stall information. These users typically possess basic digital literacy and expect a simple, intuitive user interface. Consumers interact with the system frequently and rely heavily on features such as search, browsing, reviews, favourites, and location-based recommendations.

2.3.2 Business Owners

Business owners are stall operators who maintain and update their stall profiles. They generally access the system less frequently than consumers but require precise and reliable tools for updating stall information, menus, and operating hours. They are expected to have basic proficiency in using web forms and system dashboards.

2.4 Operating Environment

HawkerSG operates within a standard web environment. Users access the system through modern web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari. The backend services run on a server environment supporting Python-based FastAPI and a SQLite database during development. The system relies on stable network connectivity for all major operations, including search, authentication, and review submission.

The LLM-based moderation service requires internet access to communicate with the external API.

2.5 Design and Implementation Constraints

2.5.1 Frontend Constraints

2.5.1.1 Framework (Frontend)

The frontend must be developed using React and Vite, as required by the project specifications.

2.5.1.2 Code Formatting

All frontend code must follow consistent style guidelines, including clear component structure, proper indentation, and meaningful naming conventions.

2.5.2 Backend Constraints

2.5.2.1 Framework (Backend)

The backend must be implemented using FastAPI in Python, following the architectural expectations outlined in the project instructions.

2.5.2.2 Database

The system is required to use SQLite during development to store users, stalls, reviews, favourites, and related data. Queries must adhere to SQLite-compatible syntax.

2.5.2.3 Code Formatting

Backend code must follow Pythonic conventions, including PEP8 styling, readable function/method names, and documented endpoints.

2.5.3 Language Constraints

Both the frontend and backend components must be implemented in JavaScript/TypeScript (React) and Python (FastAPI), respectively.

No other programming languages may be used for core system functionality.

2.6 User Documentation

2.6.1 README Files

2.6.1.1 Main Repository

The main project repository must contain a README file describing overall system structure, setup instructions, and project overview.

2.6.1.2 Frontend Repository

The frontend folder must include installation steps, script usage, environment variable configuration, and instructions for running the development server.

2.6.1.3 Backend Repository

The backend folder must include details on API endpoints, virtual environment setup, database structure, and moderation API configuration.

2.6.2 Code Comments

Both frontend and backend code must include clear and concise comments explaining important logic, component behaviour, and API interactions. Comments must be updated to reflect the current implementation.

2.6.3 Backend API Documentation

API routes must be documented through automatically generated FastAPI documentation (Swagger UI). This documentation must be accessible to developers for testing and integration.

2.7 Assumptions and Dependencies

2.7.1 Assumptions

The system assumes that users access HawkerSG through modern browsers with stable internet connections.

It is also assumed that external services, such as the LLM moderation API and OneMap geolocation API, remain reliable and available.

2.7.2 Dependencies

2.7.2.1 Front-end Libraries

The frontend depends on React, Vite, and associated UI or utility libraries used throughout development.

2.7.2.2 Back-end Libraries

The backend depends on FastAPI, SQLite, and additional Python libraries for routing, authentication, and data processing.

2.7.2.3 External Services

- HawkerSG relies on external APIs including:
- LLM moderation service for review validation
- OneMap for geolocation
- SendGrid or similar services for email notifications

3. External Interface Requirements

3.1 User Interfaces

3.1.1 Style Guides

3.1.1.1 Theme Colours

HawkerSG adopts a warm and vibrant colour palette that reflects its mission of preserving Singapore's hawker culture. The theme evokes the familiar spirit of the Little Red Dot, creating a sense of warmth and homeliness for users. *Rojo*, a bright red, is used for primary icons and buttons, while *Cornell Red* provides a darker shade for hover states, giving clear visual feedback for user interactions. The hero banner features a smooth red-to-yellow gradient that creates a welcoming and visually appealing introduction to the website. Secondary buttons use *Jonquil* yellow, with *Amber* yellow applied on hover to maintain design consistency.

B91C1C	DC2626	F97316	FBBF24	FACC15	FCD34D
Cornell red	Rojo	Pumpkin	Amber	Jonquil	Mustard

TailwindCSS default neutral colors were used to create a cohesive and readable palette that conveys hierarchical information. Rich black (gray 900) was applied for large headings to provide emphasis and denote top-level content. Other shades of gray including gray 300, gray 400, gray 500, gray 600, and gray 700 were used for borders, smaller text, captions, minor headings, and labels, helping users distinguish different levels of information. White and seasalt (gray 50) were the primary background colors, providing a neutral and easy-on-the-eye canvas that does not distract from the main content.

111827	374151	4B5563	6B7280	9CA3AF	D1D5DB	F9FAFB	FFFFFF
Rich black	Charcoal	Charcoal	Slate gray	Cadet gray	French gray	Seasalt	White

3.1.1.2 Containers and components

3.1.1.2.1 Cards

Cards are used throughout the application to display stall information, menu items, reviews and other grouped content. They have fully rounded corners to create a friendly and approachable feel, consistent with the overall warm and vibrant theme. Shadows are applied to cards to give a subtle sense of depth, and overflow is hidden to maintain a clean and contained layout. On hover, shadows intensify slightly, signaling interactivity.

3.1.1.2.2 Buttons

All buttons have fully rounded corners, matching the card aesthetic for a consistent look across interactive elements. Hover effects provide visual feedback, and icons like hearts for favorites or stars for ratings are included within buttons for added clarity.

3.1.1.2.3 Form fields and input characters

Input containers, including search bars, signup forms, and login fields, follow the same rounded aesthetic as cards and buttons. Borders are neutral gray, providing contrast without overwhelming the user. Focus states are indicated with darker or colored borders to signal user interaction.

3.1.1.2.4 Image and media containers

Images are contained within cards or dedicated media containers with rounded corners and hidden overflow to maintain a polished layout. Thumbnails and hero images adapt responsively to different screen sizes, ensuring consistency across devices.

3.1.1.2.5 Background and panels

Containers that hold multiple components, such as grids or stacked sections, use neutral backgrounds like white or light gray to separate content from the page background. Padding and spacing create visual hierarchy and improve readability, guiding the user naturally through information.

3.1.1.3 Typography

The application uses Tailwind's default sans-serif font family, specified as font-sans. The font stack includes ui-sans-serif, system-ui, sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji". On Windows, text will typically render using Segoe UI, while on MacOS it will appear as SF Pro, so the appearance may differ slightly between operating systems. For the diagram shown, the environment is Windows, so Segoe UI is used.

Text throughout the application is sized using Tailwind's scale, including xs, sm, medium, lg, xl, 2xl, 3xl and 4xl. Headings and prominent text generally appear larger, while body content uses medium and small sizes. Captions, footnotes, and less prominent text use the smaller sizes. This approach provides a consistent typographic style across the interface, ensures readability on different devices and operating systems, and allows design flexibility while maintaining a clear visual hierarchy.

Heading	48px / 4XL	ui-sans-serif	system-ui
SubHeading	36px / 3XL	ui-sans-serif	system-ui
Text (2XL)	30px / 2XL	ui-sans-serif	system-ui
Text (XL)	24px / XL	ui-sans-serif	system-ui
Text (LG)	18px / LG	ui-sans-serif	system-ui
Text (MD / Medium)	16px / MD	ui-sans-serif	system-ui
Text (SM)	14px / SM	ui-sans-serif	system-ui
Text (XS)	12px / XS	ui-sans-serif	system-ui

3.1.2 UI Mockup

3.1.2.1 Home Page

The image shows the home page of the HawkerSG website. At the top, there is a navigation bar with the HawkerSG logo, a search bar, and links for 'Browse', 'Near Me', 'Login', and 'Sign Up'. The main banner features a red background with the text 'Discover Singapore's Hawker Heritage' and a subtitle 'Find your favorite stalls, explore authentic local cuisine, and discover hidden gems in Singapore's iconic hawker centers'. A yellow button labeled 'Explore Hawker Centers' is centered below the subtitle. Below the banner, there are five food icons: a crab, a plate of rice with chicken, a bowl of noodle soup, a bowl of ramen, and a dumpling. The section titled 'Featured Hawker Centers' displays three cards with information about Maxwell Food Centre, Lau Pa Sat, and Newton Food Centre. Each card includes a photo, the center's name, address, a brief description, the number of stalls, and an 'Open daily' status. Red arrows on the left and right sides of the cards indicate they can be swiped. At the bottom, there are two buttons: 'View All Hawker Centers' and 'Hawkers near me'. The footer section is red with the text 'Own a Hawker Stall?' and a call-to-action 'Register Your Stall'.

HawkerSG

Search hawker centers or stalls...

Browse Near Me Login Sign Up

Discover Singapore's Hawker Heritage

Find your favorite stalls, explore authentic local cuisine, and discover hidden gems in Singapore's iconic hawker centers

Explore Hawker Centers

Featured Hawker Centers

Discover the most popular destinations for authentic local cuisine

Maxwell Food Centre
1 Kadayanallur St, Singapore 069184
Famous hawker center in Chinatown known for Tian Tian Hainanese Chicken Rice
85 stalls Open daily

Lau Pa Sat
18 Raffles Quay, Singapore 048582
Historic Victorian market building offering diverse local cuisine
56 stalls Open daily

Newton Food Centre
500 Clemenceau Ave N, Singapore 229495
Popular 24-hour food court famous for seafood and satay
92 stalls Open daily

< >

View All Hawker Centers **Hawkers near me**

Own a Hawker Stall?

Join our platform to showcase your stall, manage your menu, and connect with customers

Register Your Stall

3.1.2.2 Sign Up

3.1.2.2.1 Consumer View

Create your account
Join Singapore's hawker community

Account Type

Consumer Business Owner

Full Name

Email address

Password

Confirm Password

By signing up, you agree to our [Terms and Conditions](#) and [Privacy Policy](#).

Create Account

Already have an account? [Sign in here](#)

3.1.2.2.2 Business View

Create your account
Join Singapore's hawker community

Account Type

Consumer Business Owner

Sign Up With corppass

By signing up, you agree to our [Terms and Conditions](#) and [Privacy Policy](#).

Already have an account? [Sign in here](#)

Create your account
Join Singapore's hawker community

Account Type

Consumer Business Owner

Full Name

Email Address

Unique Entity Number(UEN)

Password

Confirm Password

By signing up, you agree to our [Terms and Conditions](#) and [Privacy Policy](#).

Create Account

Already have an account? [Sign in here](#)

3.1.2.3 Authentication (Login)

3.1.2.3.1 Consumer View

The screenshot shows the HawkerSG consumer login interface. At the top, there is a header with the HawkerSG logo, a search bar containing "Search hawker centers or stalls...", and navigation links for "Browse", "Near Me", "Login", and a red "Sign Up" button. Below the header, the main content area features the HawkerSG logo and the text "Welcome back". A sub-instruction "Sign in to your account" is displayed. An "Account Type" section contains two buttons: "Consumer" (highlighted in pink) and "Business Owner". Below this are fields for "Email address" and "Password", with a "Forgot password?" link and an eye icon next to the password field. A large red "Sign In" button is centered at the bottom of the form. A small note at the very bottom says "Don't have an account? [Sign up here](#)".

3.1.2.3.2 Business View

The screenshot shows the HawkerSG business login interface, which is nearly identical to the consumer version. It features the HawkerSG logo, a search bar, and navigation links at the top. The main content area includes the "Welcome back" message and "Sign in to your account" instruction. The "Account Type" section has "Consumer" and "Business Owner" buttons, with "Business Owner" being highlighted in pink. Below are fields for "Email address" and "Password", with a "Forgot password?" link and an eye icon. A large red "Sign In" button is at the bottom. A note at the bottom states "Don't have an account? [Sign up here](#)".

3.1.2.4 Consumer Function

3.1.2.4.1 Browse

3.1.2.4.1.1 Hawker Centres

The screenshot shows the HawkerSG app interface for browsing hawker centers. At the top, there is a search bar with placeholder text "Search hawker centers or stalls...". Below the search bar are two buttons: "Hawker Centers (123)" and "Stalls (3001)". The main content area displays a grid of five hawker center cards:

- Maxwell Food Centre**: Located at 1 Kadayanallur St, Singapore 069184. It has 85 stalls and a rating of 4.2 stars. Status: Open daily.
- Lau Pa Sat**: Located at 18 Raffles Quay, Singapore 048582. It has 56 stalls and a rating of 4.5 stars. Status: Open daily.
- Newton Food Centre**: Located at 500 Clemenceau Ave N, Singapore 229495. It has 92 stalls and a rating of 4.7 stars. Status: Open daily.
- Tiong Bahru Market**: Located at 51 Old Airport Road. It has 47 stalls and a rating of 4.7 stars. Status: Open daily.
- Bedok Food Centre**: Located at 51 Old Airport Road. It has 45 stalls and a rating of 4.5 stars. Status: Open daily.

3.1.2.4.1.2 Hawker Stalls

The screenshot shows the HawkerSG app interface for browsing hawker stalls. At the top, there is a search bar with placeholder text "Search hawker centers or stalls...". Below the search bar are two buttons: "Hawker Centers (123)" and "Stalls (3001)". The main content area displays a grid of two hawker stall cards:

- Tian Tian Hainanese Chicken Rice**: Chinese cuisine. Rating: 4.9 (1247). Operating hours: 10:00 - 20:00. Description: Famous for tender chicken rice that has been featured in international media.
- Zhen Zhen Porridge**: Chinese cuisine. Rating: 4.8 (432). Operating hours: 10:00 - 20:00. Description: Traditional Teochew porridge with various accompaniments.

3.1.2.4.2 Search Bar + Recent Searches

The screenshot shows the HawkerSG website's search interface. At the top, there is a search bar with a magnifying glass icon and a dropdown menu. Below the search bar, there is a "Recent Searches" section with three items: "satay", "market", and "chicken rice". The main content area displays search results for "Hawker Centers (2)". Two cards are shown: "Maxwell Food Centre" and "Tiong Bahru Market". Each card includes a thumbnail image, the stall name, address, a brief description, and operational status.

Hawker Center	Address	Description	Operational Status
Maxwell Food Centre	1 Kadayanallur St, Singapore 069184	Famous hawker center in Chinatown known for Tian Tian Hainanese Chicken Rice	85 stalls, Open daily
Tiong Bahru Market	30 Seng Poh Rd, Singapore 168898	Lively food court with Singaporean hawker stalls & a "wet market" selling produce, meat & fish.	77 stalls, Open daily

3.1.2.4.3 View Stall Profile

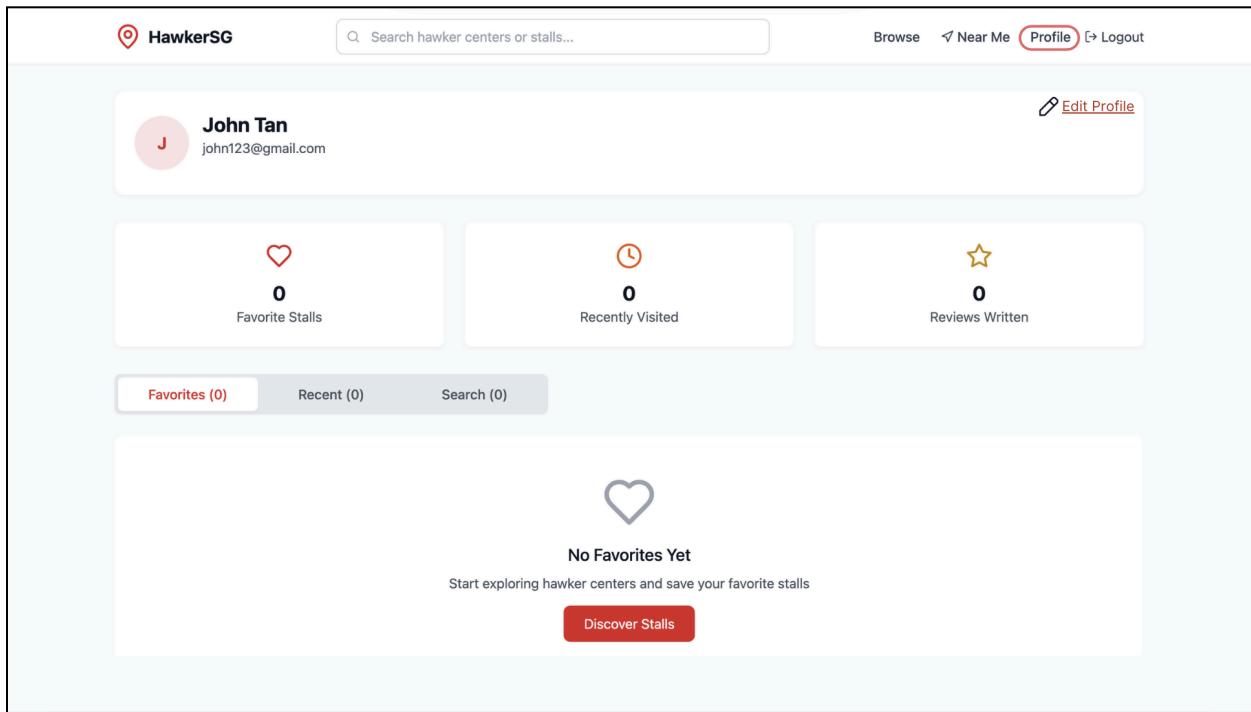
The screenshot shows the profile page for "Phawo Thai Food" at Yishun Park Hawker Centre. The page includes a photo of the stall, the stall's name and cuisine type, star ratings, operating hours, address, a menu section with three items, a reviews section with one review, and a description of the stall's offerings.

Item	Description	Price
Green Curry	Traditional green curry with your choice of meat (Pork, Chicken, Beef)	S\$6.00
Basil Pork Rice	Authentic basil pork rice with a side of our family-style Thai herbs to give a kick of flavour!	S\$5.50
Spring Rolls (5pcs)	Thai Spring Rolls with fillings of (Mushroom, Glass Noodles, Carrot, White Pepper, Minced Chicken, Garlic, Bean Sprouts & Spring Onions)	S\$5.00

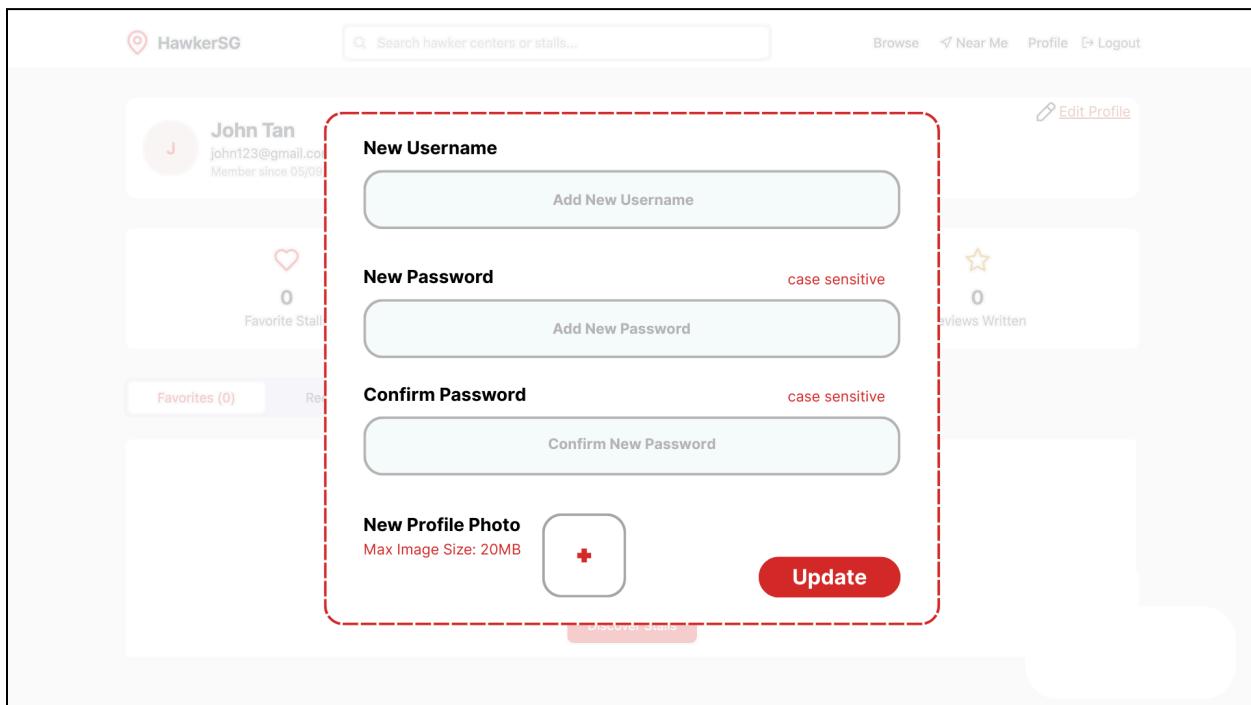
Reviews (1)

User	Date	Rating
john tan	13 November 2025	★★★☆☆

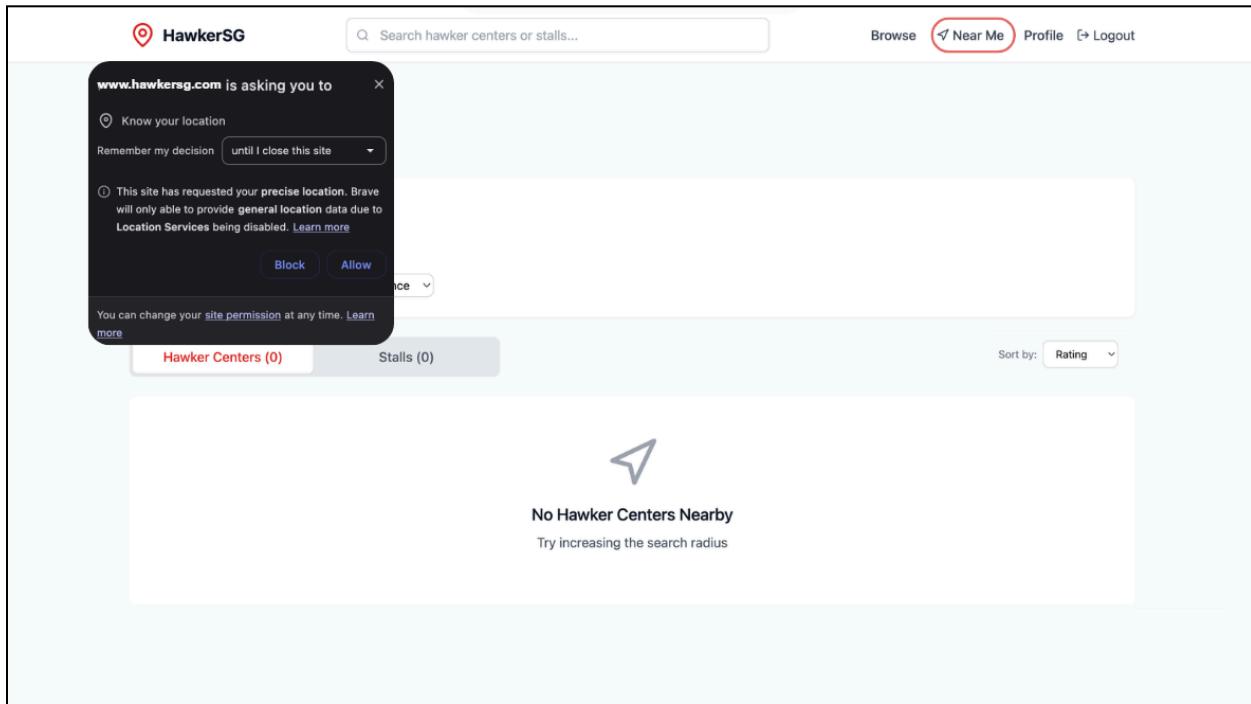
3.1.2.4.4 User Profile



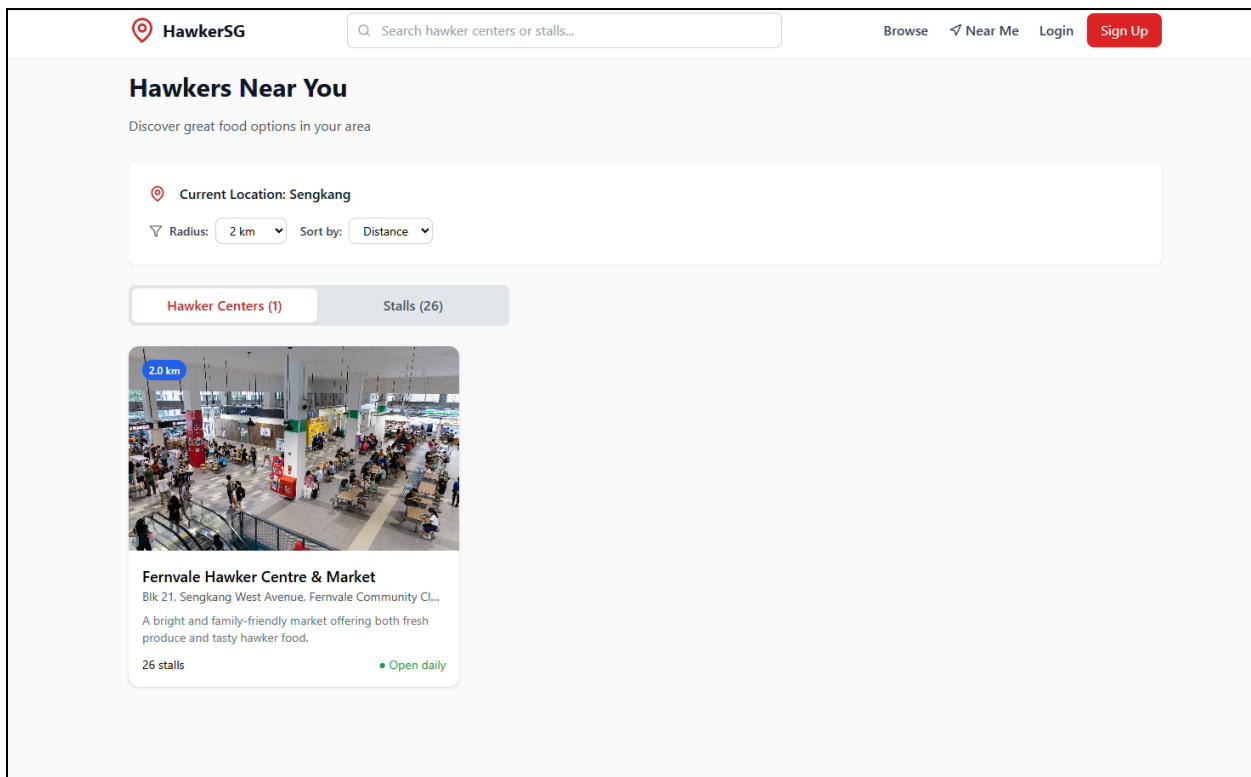
3.1.2.4.5 Edit Profile



3.1.2.4.6 Location Permission



3.1.2.4.7 Hawkers Nearby



3.1.2.4.8 Upload Reviews

Write a Review

Phawo Thai Food

Share your experience with other food lovers

Rating

☆ ☆ ☆ ☆ ☆

Your Review

Tell others about your experience...

Photos (Optional)

Add Photo

Add up to 5 photos to help others know what to expect

[Cancel](#) [Post Review](#)

3.1.2.5 Business Profile (Functions)

3.1.2.5.1 Overview

Business Profile
Manage your stall presence on HawkerSG

[Consumer View Preview](#)

[Overview](#) [Stall Profile](#) [Menu](#) [Operating Hours](#) [View Reviews](#)

Stall Overview

N/A Average Rating	0 Total Reviews	0 Menu Items	Open Current Status
-----------------------	--------------------	-----------------	------------------------

Your Stall: Phawo Thai Food

Cuisine: Thai
Location: 51 YISHUN AVENUE 11 #01-31, Yishun park hawker centre, Singapore 768867, Yishun Park Hawker Centre
Description:

HawkerSG

3.1.2.5.2 Edit Stall Profile

Business Profile
Manage your stall presence on HawkerSG

[Consumer View Preview](#)

[Overview](#) [Stall Profile](#) [Menu](#) [Operating Hours](#) [View Reviews](#)

Stall Profile

Stall Name *

Cuisine Type *

Stall Location *

Description

Stall Photo

HawkerSG

Add a high-quality photo of your stall (max 1).

Update Stall Profile

3.1.2.5.3 Edit Menu

Business Profile
Manage your stall presence on HawkerSG

Logout Consumer View Preview

Overview Stall Profile **Menu** Operating Hours View Reviews

Menu Management

+ Add Menu Item

Menu Item 1

Item Name * Enter item name

Description Describe this item...

Price (\$) * 0

Photo (Optional) Add Photo

Update Menu

3.1.2.5.4 Edit Operating Hours

Business Profile
Manage your stall presence on HawkerSG

Logout Consumer View Preview

Overview Stall Profile **Operating Hours** View Reviews

Operating Hours

Close Stall:

Monday	<input checked="" type="checkbox"/> Open	From: 10:00 am	To: 08:00 pm	Copy to all
Tuesday	<input checked="" type="checkbox"/> Open	From: 10:00 am	To: 08:00 pm	Copy to all
Wednesday	<input checked="" type="checkbox"/> Open	From: 10:00 am	To: 08:00 pm	Copy to all
Thursday	<input checked="" type="checkbox"/> Open	From: 10:00 am	To: 08:00 pm	Copy to all
Friday	<input checked="" type="checkbox"/> Open	From: 10:00 am	To: 08:00 pm	Copy to all
Saturday	<input checked="" type="checkbox"/> Open	From: 10:00 am	To: 08:00 pm	Copy to all
Sunday	<input checked="" type="checkbox"/> Open	From: 10:00 am	To: 08:00 pm	Copy to all

Quick Actions

[Set All: 10:00 - 20:00](#) [Close Sundays](#) [Weekend Extended](#)

Save Hours

3.1.2.5.5 View Reviews

The screenshot shows the HawkerSG Business Profile interface. At the top, there's a header with a location pin icon and the text "HawkerSG". On the right is a "Logout" link. Below the header, a "Business Profile" section says "Manage your stall presence on HawkerSG". To the right of this is a "Consumer View Preview" link. A navigation bar below includes "Overview", "Stall Profile", "Menu", "Operating Hours", and "View Reviews" (which is underlined). Under "View Reviews", it shows "0.0 0 reviews" with a 5-star rating icon. To the right are filter options: "Filter: All" (selected), "5★", "4★", "3★", "2★", "1★", and a "Sort: Most recent" dropdown. Below this are star rating breakdowns: 5★ (0), 4★ (0), 3★ (0), 2★ (0), and 1★ (0). A "Reviews" section below shows a message: "No reviews found for the selected filters." and "Showing 0 items".

3.1.2.5.6 Consumer View Preview

The screenshot shows the HawkerSG Consumer View Preview for a stall named "Phawo Thai Food". The top left shows the location pin icon and "HawkerSG". The top center shows the current location: "Yishun Park Hawker Centre / Phawo Thai Food". The main content area has a white background. It starts with a "Menu" section which says "Menu information not available". Below that is a "Reviews (0)" section with a speech bubble icon and the message "No Reviews Yet. Be the first to share your experience!". At the bottom left is a "Description" section.

3.2 Hardware Interfaces

HawkerSG is designed to operate primarily as a web-based application, minimizing the need for direct hardware interfaces. However, the following outlines the logical and physical characteristics of the interfaces between the software and hardware:

3.2.1 Device Types

The software is designed to be compatible with various devices, including desktop computers, laptops, tablets, and smartphones.

3.2.2 Data and Control Interactions

The user's device interacts with the software through standard input/output hardware such as keyboards, touchscreens, and mice for input; displays for output; and network interface cards for internet connectivity.

3.2.3 Communication Protocols

The platform uses HTTPS for secure web traffic. The APIs, websocket connection, and services will communicate over standard web ports (80 for HTTP and 443 for HTTPS).

3.3 Software Interfaces

HawkerSG interfaces with several software components:

- **Frontend:** is built using React 18 as the core UI framework, leveraging hooks for state management and lifecycle operations. Custom hooks including useAuth() and useData() encapsulate shared business logic and data fetching operations, promoting code reuse across components. Local Storage enables client-side persistence for user-specific data such as search history and favorite stalls, maintaining user preferences between sessions without requiring server-side storage.
- **Backend:** The backend system is constructed using FastAPI with automatic OpenAPI documentation generation. SQLAlchemy provides database abstraction and relationship management between Python objects and database tables. Pydantic handles data validation and serialization through defined schemas. Authentication and security are implemented using JWT (JSON Web Tokens) for stateless authentication with configurable expiration timing. The system employs OAuth2 Password Flow as a standardized authentication protocol, allowing secure credential exchange.
- **Database:** SQLAlchemy ORM serves as the primary database abstraction layer, managing relationships and providing interface for data operations. Data models and relationships follow polymorphic patterns.

- **External APIs & Services:** The application integrates with several external services to enhance functionality. The OneMap API serves as the primary geolocation service, providing Singapore-specific address validation. For email communications, the system integrates with SendGrid as the transactional email service provider. Review moderation and utilizes OpenAI's GPT models through their API interface. This external AI service processes user-submitted reviews to detect policy violations and inappropriate content through a structured classification system.

3.3.1 Operating System

Any OS capable of running a modern web browser (Windows, macOS, Linux, iOS, Android).

3.3.2 Web Browsers

Latest versions of Chrome, Firefox, Safari, Edge, and others capable of supporting HTML5, CSS3, and JavaScript.

3.3.3 Databases

- **SQLAlchemy (ORM - Object Relational Mapper)**
 - **Version:** Modern SQLAlchemy with async support
 - **Purpose:** Database abstraction and object mapping
 - **Communications:** Synchronous database queries using session management
- **Database System:** PostgreSQL/SQLite (based on SQLAlchemy models)
- **Data Sharing:** Database session dependency injection via `get_db()` function

3.3.4 Backend Tools

- 1) FastAPI (version 0.121.1) for server-side logic
- 2) Python (version 3.13.2) for scripting

3.3.5 Frontend Tools

- 1) React.js (version 18.3.1)
- 2) TypeScript (version 5.6.3)
- 3) TailwindCSS (version 3.4.17)
- 4) Prettier and ESLint for code formatting

3.3.6 RESTful APIs

The application implements a RESTful API architecture using FastAPI, separating between public and protected endpoints. Authentication endpoints (/consumer/signup, /consumer/login, /business/signup, /business/login) handle user registration and JWT token generation, while protected endpoints require Bearer token authentication for operations like profile updates, review creation, and favorite management.

3.3.7 Data Sharing

Data sharing across software components follows a structured pattern using Pydantic schemas for request/response validation and SQLAlchemy models for database interaction. The frontend and backend communicate through defined JSON payloads for all data exchanges and data persistence is maintained through the SQL database with defined data models.

3.4 Communications Interfaces

For HawkerSG, various communication interfaces are integral to ensuring a seamless and responsive user experience:

- **HTTP/HTTPS:** The primary communication protocol for HawkerSG is HTTP/S. The frontend web browser communicates with the backend server over HTTPS, securing the communication by encrypting all data in transit. The backend is built with FastAPI, a modern Python web framework, which handles HTTP requests and responses efficiently.
- **API Communication:** The frontend interacts with the backend through a RESTful API. This API defines a set of endpoints for creating, retrieving, updating, and deleting data. All data is exchanged in JSON format.
- **Electronic Forms:** User-submitted data from all electronic forms is transmitted to the server via secure HTTPS POST requests, with comprehensive client-side and server-side validation ensuring data integrity through structured JSON formatting.
- **Email:** The system sends password reset confirmations using the SendGrid API. Messages are formatted in HTML or plain text (MIME standard) and delivered via HTTPS requests to SendGrid's servers.
- **Network Protocols:** The system utilizes standard TCP/IP protocols with HTTP/HTTPS for all client-server communication, ensuring reliable and secure data transmission that adheres to established internet standards across all application features.
- **Security and Encryption:** Token-based authentication is implemented using JWT that contains essential account information. This token structure enables authorization checks that validate user permissions for accessing specific resources and performing authorized actions. Password encryption is done using the Argon2 hashing algorithm that is resistant to brute-force attacks for storing and verifying user credentials.

4. System Features

Do refer to the ‘Use Case Diagram’, ‘Use Case Descriptions’ and ‘Sequence Diagrams’ to have a visual on the use case diagram, detailed use case descriptions and flow of events using sequence diagrams.

4.1 Authentication (Consumer & Business)

4.1.1 Consumer Signup

4.1.1.1 Description and Priority

Description:

This feature allows a Consumer to create an Account in the System by providing Personal Information, including username, email, and password. The System validates all inputs, checks email uniqueness, and enforces password rules before creating the Account.

Priority: High

4.1.1.2 Stimulus/Response Sequences

1. The Consumer selects the “Sign Up” option.
 - a. The System displays the form requesting the required Personal Information (Username, Email, and Password).
2. The Consumer submits the form.
 - a. The System validates all Personal Information.
 - i. If the email is invalid or already linked to an existing Account, the System displays an error.
 - ii. If the password violates the password policy, the System displays a password policy violation error.
 - b. If all inputs are valid, the System creates the Account.
 - i. The System activates the Account and redirects the Consumer to the login page.

4.1.1.3 Functional Requirements

REQ-1: The System shall allow a Consumer to initiate Account creation.

REQ-2: The System shall display a form requesting Personal Information.

REQ-3: The System shall collect the Consumer's username.

REQ-4: The System shall collect the Consumer's email and verify that it is not associated with an existing Account.

REQ-5: The System shall collect and validate the Consumer's password against the System's password policy.

REQ-6: The System shall prevent Account creation when validation fails and shall display appropriate error messages.

REQ-7: The System shall create and activate the Consumer Account after the sign-up process is completed.

REQ-8: The System shall redirect the Consumer to the Login page upon successful Account creation.

4.1.2 Business Signup (CorpPass Verification)

4.1.2.1 Description and Priority

Description:

This feature allows a Business Owner to register for an Account within the System. Business Owners must verify their identity using a simulated CorpPass process. The System retrieves and validates Stall Information using the Business Owner's UEN.

Priority: High

4.1.2.2 Stimulus/Response Sequences

1. The Business Owner selects the “Business Owner” option.
 - a. The System displays the Business Owner registration form requesting Personal Information.
 - b. The Business Owner selects “Sign up with CorpPass.”
2. The System redirects the Business Owner to the simulated CorpPass interface.
 - a. The Business Owner signs in via CorpPass.
 - i. The System retrieves the Business Owner’s UEN and corresponding Stall Information.
 - ii. The System validates the UEN.
 - iii. If invalid or not associated with a Hawker Stall registered in the System, an error is displayed.
 - iv. If valid, the System pre-fills Stall Information into the registration form.
 - b. The Business Owner fills in the password field.
 - i. If the password violates the password policy, the System displays a password policy violation error.
3. The Business Owner confirms registration.
 - a. The System creates the Business Owner Account.
4. The System displays a success message and directs the Business Owner to the login page.

4.1.2.3 Functional Requirements

REQ-1: The System shall allow a Business Owner to initiate Account creation.

REQ-2: The System shall display a registration form requesting required Personal Information.

REQ-3: The System shall perform CorpPass verification before an Account is created.

REQ-4: The System shall retrieve UEN-linked Stall Information from the CorpPass API.

REQ-5: The System shall validate that the retrieved Stall Information matches a Hawker Stall in the System.

REQ-6: The System shall pre-fill Stall Information after successful CorpPass verification.

REQ-7: The System shall validate the Business Owner's password according to the System's password policy.

REQ-8: The System shall prevent account creation if password validation fails and shall display an appropriate password policy violation error message.

REQ-9: The System shall create the Business Owner Account after all required fields have been successfully validated.

REQ-10: The System shall display a success message and redirect the Business Owner to the login page after account creation.

4.1.3 Login (Consumer & Business Owner)

4.1.3.1 Description and Priority

Description:

This feature allows both Consumers and Business Owners to log in to the System using their registered email and password. The System verifies the submitted credentials and grants access to the appropriate interface based on the user type.

Priority: High

4.1.3.2 Stimulus/Response Sequences

1. The user selects "Login."
 - a. The System displays the Login form with fields for account type, email, and password.
2. The user enters their credentials and submits the form.
 - a. The System validates the submitted data.
 - b. If credentials are invalid, the System displays an error message and does not log the user in.
3. If credentials are valid, the System redirects to the Home page (for Consumers) or redirects to the Business Profile (for Business Owners).

4.1.3.3 Functional Requirements

REQ-1: The System shall display login fields for account type, email, and password.

REQ-2: The System shall validate that the submitted email corresponds to an existing account.

REQ-3: The System shall validate that the submitted password matches the stored password for the account.

REQ-4: The System shall display an error message if the credentials are invalid.

REQ-5: The System shall authenticate valid accounts and route Users to the appropriate interface (HomePage for consumers or Business Profile for business owners).

REQ-6: The System shall prevent login if required fields (email or password) are left empty.

4.1.4 Password Reset

4.1.4.1 Description and Priority

Description:

This feature allows any Account holder (Consumer or Business Owner) to reset their password if forgotten. The System verifies the user's identity by sending a secure password-reset link to the registered email address. The user can then create a new password that complies with the System's password policy.

Priority: High

4.1.4.2 Stimulus/Response Sequences

1. The user selects "Forgot Password."
 - a. The System requests the email associated with the Account.
2. The user enters their email and submits the request.
3. The System validates the email:
 - a. If the email is not registered, the System displays an error message.
 - b. If the email is valid, the System sends a password reset link to the email.
4. The user clicks the password reset link received in their email.
 - a. The System displays the password reset form.
5. The user enters a new password and submits the form.
6. The System validates the password according to the password policy.

- a. If invalid, the System displays an error message and prevents the update.
 - b. If valid, the System updates the account's stored password.
7. The System confirms the successful password reset and displays a confirmation message.

4.1.4.3 Functional Requirements

REQ-1: The System shall allow users to initiate the password reset process by selecting “Forgot Password”.

REQ-2: The System shall request the email associated with the account.

REQ-3: The System shall validate that the submitted email belongs to a registered account.

REQ-4: The System shall display an error message if the email is not valid.

REQ-5: The System shall send a secure password reset link to the registered email after successful email validation.

REQ-6: The System shall provide access to a password reset form when the user clicks the reset link.

REQ-7: The System shall allow the user to enter a new password through the password reset form.

REQ-8: The System shall validate the new password according to the System's password policy.

REQ-9: The System shall display an error message if the password fails validation and prevent updating the password.

REQ-10: The System shall update the account's stored password upon successful validation.

REQ-11: The System shall display a confirmation message after the password has been successfully reset.

4.2 Consumer Features

4.2.1 Search (Hawker Centre & Hawker Stall)

4.2.1.1 Description and Priority

Description:

This feature allows a Consumer to search for Hawker Centres or Hawker Stalls by entering keywords into the Search bar. The System retrieves matching Stall Information or Hawker Centre data and returns them as Search Results. The System also validates search input, trims unnecessary spacing, and enforces a maximum character limit of 100.

Priority: High

4.2.1.2 Stimulus/Response Sequences

1. The Consumer clicks the Search bar.
 - a. The System activates the input field and allows input up to 100 characters.
2. The Consumer enters a keyword and submits the search.
 - a. The System trims spacing and validates the keyword length.
 - i. If the input exceeds 100 characters, the System stops accepting input and displays a popup message informing the Consumer that the input has exceeded the limit.
 - ii. If the input is valid, the System performs a search for Hawker Centres and Hawker Stalls that match the keyword.
3. The System displays Search Results.
 - a. If no results are found, the System displays a “No Results Found” message.
 - b. If results exist, the System displays a list of matching Hawker Centres or Hawker Stalls.
4. The Consumer selects a result from the list.
 - a. The System displays the Stall Information page for stalls or the Hawker Centre details page for the hawker centre.

4.2.1.3 Functional Requirements

REQ-1: The System shall provide a search bar for Consumers to enter keywords.

REQ-2: The System shall allow up to a maximum of 100 characters input.

REQ-3: The System shall display a popup message and stop accepting input when the Consumer enters more than 100 characters.

REQ-4: The System shall trim whitespace and validate keyword length before performing a search.

REQ-5: The System shall search and return matching Hawker Centres or Hawker Stalls based on the keyword provided.

REQ-6: The System shall display a “No Results Found” message when no results match the keyword.

REQ-7: The System shall display the correct details page (Stall Information page for stalls or Hawker Centre details page for hawker centres) when a search result is selected.

4.2.2 Recent Searches

4.2.2.1 Description and Priority

Description:

This feature stores a history of a Consumer's previous Search inputs to allow easy re-searching and to improve convenience.

Priority: Medium

4.2.2.2 Stimulus/Response Sequences

1. The Consumer performs a Search.
 - a. The System records the Search term and saves it to the search history.
2. The Consumer opens the Search page.
 - a. The System displays recent Search terms.
3. The Consumer selects a past Search term.
 - a. The System re-executes the Search.

4.2.2.3 Functional Requirements

REQ-1: The System shall store the Consumer's recent Search terms.

REQ-2: The System shall display the Consumer's recent Search terms when the Search page is accessed.

REQ-3: The System shall allow Consumers to re-search using stored Search terms.

REQ-4: The System shall allow Consumers to clear their Search history.

4.2.3 Near Me

4.2.3.1 Description and Priority

Description:

This feature allows the Consumer to discover nearby Hawker Stalls and Hawker Centres using their current location, obtained through the OneMap API.

The System calculates distance and returns results sorted by proximity.

Priority: High

4.2.3.2 Stimulus/Response Sequences

1. The Consumer selects “Near Me” in the navigation bar
 - a. The System requests location access permission.
2. If the Consumer allows access:
 - a. The System retrieves coordinates via the Geolocation API.
 - b. The System performs reverse geocoding of the user’s coordinates and translates it into an area name, such as Yishun, via the OneMap API.
3. The System calculates distances between the Consumer and Hawker Stalls.
4. The System displays a list of nearby Hawker Centres or Hawker Stalls sorted by distance.
5. If the user denies location access:
 - a. The System displays an alternative Search option.

4.2.3.3 Functional Requirements

REQ-1: The System shall request permission from the Consumer to access their current location.

REQ-2: The System shall obtain the Consumer’s latitude and longitude using the device’s location services.

REQ-3: The System shall retrieve stall coordinates from the database and/or OneMap API.

REQ-4: The System shall calculate distances between the Consumer's location and nearby hawker stalls.

REQ-5: The System shall display results sorted by proximity.

REQ-6: The System shall display fallback search options if the Consumer denies location access.

4.2.4 Favourites

4.2.4.1 Description and Priority

Description:

This feature allows a Consumer to add or remove Hawker Stalls from their list of Favourites.

Favourites allow quicker access to preferred stalls.

Priority: High

4.2.4.2 Stimulus/Response Sequences

1. The Consumer presses the heart icon
 - a. If the stall is not yet a favourite, the system will mark it
 - b. If the stall is already a favourite, the system will unmark it

4.2.4.3 Functional Requirements

REQ-1: The System shall allow Consumers to add a Hawker Stall to Favourites.

REQ-2: The System shall prevent duplicate favourite entries.

REQ-3: The System shall allow Consumers to remove a Stall from Favourites.

REQ-4: The System shall display all favourite Stalls in the Favourites tab in the consumer profile page.

4.2.5 Reviews

4.2.5.1 Description and Priority

Description:

This feature allows a Consumer to write, edit, and view reviews for Hawker Stalls.

Each Consumer can only have one Review per Stall, and submitting a new one replaces the old one.

Reviews include star ratings, descriptions, and optional photos.

All reviews are filtered using the LLM API.

Priority: High

4.2.5.2 Stimulus/Response Sequences

A. Writing a Review

1. The Consumer selects a Hawker Stall.
2. The Consumer selects Write Review.
 - a. The System displays inputs for rating, description, and photos.
3. The Consumer submits the review.
 - a. The System validates:
 - i. star rating (1–5)
 - ii. description length (≤ 250 chars)
- b. The System performs moderation of the review using the LLM API.
 - i. If flagged, the System displays a moderation error.
 - ii. If approved, the System saves the review.
4. The System displays “Review Added”.

B. Editing a Review

1. The Consumer selects a Hawker Stall with an existing review from the Consumer.
2. The Consumer selects Write Review.

- a. The System displays inputs for rating, description, and photos.
3. The Consumer submits the revised Review.
 - a. The System validates:
 - i. star rating (1–5)
 - ii. description length (≤ 250 chars)
 - b. The System performs moderation of the review using the LLM API.
 - i. If flagged, the System displays a moderation error.
 - ii. If approved, the System replaces the old review with the updated version.
4. The System displays "Review Updated".

C. Viewing Reviews

1. The Consumer opens a Stall Information page.
2. The System displays all reviews associated with the stall.

4.2.5.3 Functional Requirements

REQ-1: The System shall allow a Consumer to write a Review for any Hawker Stall.

REQ-2: The System shall enforce a limit of one Review per Consumer per Stall.

REQ-3: The System shall replace the old Review if the Consumer submits a new one.

REQ-4: The System shall validate the star rating (1–5).

REQ-5: The System shall validate review description length (≤ 250 characters).

REQ-6: The System shall send the Review for moderation using the LLM API.

REQ-7: The System shall save and display the Review if approved.

REQ-8: The System shall inform the Consumer if moderation fails.

REQ-9: The System shall display all reviews for a Hawker Stall.

4.2.6 Consumer Profile Management

4.2.6.1 Description and Priority

Description:

This feature allows a Consumer to view their **Personal Information**, including:

- Username
- Email
- Password
- Profile photo
- Favourites
- Recently visited
- Reviews

Consumers can edit their name, password, and profile photo.

Priority: Medium

4.2.6.2 Stimulus/Response Sequences

1. The Consumer selects “**Profile**”.
 - a. The System displays the Consumer Profile.
2. The Consumer selects “**Edit Profile**.
 - a. The System displays editable fields.
 - i. New Username
 - ii. New Password
 - iii. Confirm New Password
 - iv. Profile Picture
 - b. The Consumer fills in the relevant fields.
 - c. The System validates the fields
 - i. If no fields are modified, it will display no changes detected

- ii. If the password is one of the modified fields, it will check if the password policy is met or if the password and the confirm password match.
- iii. If the profile picture is one of the modified fields, it will check if it is a valid image file type and also ensure that it is within the acceptable file size limit
- d. The System saves the updated Personal Information.
- e. The System confirms the update.
 - i. If the password has been modified, the consumer will be logged out and redirected to the login page

4.2.6.3 Functional Requirements

REQ-1: The System shall allow Consumers to view their existing profile information.

REQ-2: The System shall allow Consumers to edit their Personal Information.

REQ-3: The System shall validate all updated Personal Information fields.

REQ-4: The System shall enforce password policy requirements for password updates.

REQ-5: The System shall validate profile picture uploads to ensure correct file format and file size.

REQ-6: The System shall save the updated Personal Information after validation.

REQ-7: The System shall display a confirmation message after saving changes.

4.2.7 Browse

4.2.7.1 Description and Priority

Description:

Displays comprehensive lists of the individual Hawker Centres and Hawker Stalls.

Consumers can filter the Hawker Stalls by status, cuisine, and star rating.

Priority: Medium

4.2.7.2 Stimulus/Response Sequences

1. The Consumer selects “Browse” on the navigation bar
2. A list of Hawker Centres and Stalls will be displayed
3. The following options will be displayed for filtering:
 - a. A checkbox to toggle for stalls that are open/closed
 - b. A dropdown to select a rating ranging from 2 to 5
 - c. A dropdown to select cuisine such as Chinese, Thai, and so on.
4. The system displays the filtered results
 - a. If there are no results, return the message “No results to display.”

4.2.7.3 Functional Requirements

REQ-1: The System shall allow Consumers to view the list of hawker centres and stalls

REQ-2: The System shall allow filtering by stall status, star rating, and cuisine.

REQ-3: The System shall display a message if no results match filters.

4.3 Business Owner Management Features

4.3.1 Overview

4.3.1.1 Description and Priority

Description:

The Overview page provides a summary of the Business Owner stall, including key metrics such as rating, total reviews, number of menu items, and current stall status. It also displays basic stall information and provides navigation to stall management features.

Priority: Medium

4.3.1.2 Stimulus/Response Sequences

1. Business Owner logs in.
 - a. The System shows the Business Profile with the Overview tab selected.
2. The System displays the following information:
 - a. Average Rating
 - b. Total Reviews
 - c. Menu Items
 - d. Current Status
3. The System displays the stall main information:
 - a. Stall Name
 - b. Cuisine Type
 - c. Location
 - d. Stall Image
 - e. Description
4. Business Owner may select any of the management tabs:
 - a. Stall Profile
 - b. Menu
 - c. Operating Hours
 - d. View Reviews
5. Business Owner may select “Consumer View Preview”

- a. The System displays a Consumer version of the stall profile.

4.3.1.3 Functional Requirements

REQ-1: The System shall display the Business Profile Overview as the default landing page after Business login.

REQ-2: The System shall display non-editable stall summary metrics (Average Rating, Total Reviews, Menu Items, Current Status).

REQ-3: The System shall display non-editable stall information (name, cuisine, location, image, description).

REQ-4: The System shall provide navigation tabs for Stall Profile, Menu, Operating Hours, and View Reviews.

REQ-5: The System shall provide a “Consumer View Preview” button.

4.3.2 Edit Stall Profile

4.3.2.1 Description and Priority

Description:

This feature allows Business Owner to edit Stall Profile with fields such as stall name, stall photo, cuisine type, stall description.

Priority: High

4.3.2.2 Stimulus/Response Sequences

1. The Business Owner navigates to the Stall Profile tab.
 - a. The System displays editable Stall Profile details.
2. The Business Owner updates stall details:
 - a. Stall Name

- b. Cuisine Type
 - c. Description
 - d. Stall Image
3. Business Owner selects “Update Stall Profile”
- a. The System validates all fields
 - b. If invalid, the System displays an appropriate error message.
 - c. If valid, the System saves updated stall profile details.
4. The System displays a confirmation message.

4.3.2.3 Functional Requirements

REQ-1: The System shall allow Business Owners to edit Stall Profile details.

REQ-2: The System shall validate updated Stall Profile fields before saving.

REQ-3: The System shall update and store modified Stall Profile after validation.

REQ-4: The System shall confirm successful updates after saving.

4.3.3 Manage Menu

4.3.3.1 Description and Priority

Description:

This feature allows Business Owners to manage Menu items by adding, editing, or deleting individual dishes. Each menu item may include a name, description, image, and price.

Priority: High

4.3.3.2 Stimulus/Response Sequences

1. Business Owners selects the “Menu” tab.
- a. The System displays the Menu Management page.
 - b. If no menu items exist, the System shows a “No Menu Items” placeholder with a button “Add First Item”.

2. Business Owner may add a Menu Item by selecting “Add Menu Item” or “Add First Item”.
 - a. The System inserts a new Menu Item card with empty fields for item name, price, description, upload photo and a delete icon.
3. Business Owner may edit any Menu Item by entering or modifying its fields directly inside the card.
 - a. The System updates the form but does not save them yet.
4. Business Owner may remove a Menu Item by selecting the delete icon.
 - a. The System will remove the Menu Item card immediately
5. Business Owner selects “Update Menu”.
 - a. The System validates all Menu Item cards by the required fields, valid numeric price and valid image formats.
 - b. If any fields are invalid, the System highlights errors and prevents saving.
 - c. If validation is successful, the System saves all Menu Items at once and displays a confirmation message.

4.3.3.3 Functional Requirements

REQ-1: The System shall allow Business Owners to manage Menu items.

REQ-2: The System shall provide input fields for name, description, images, and price.

REQ-3: The System shall validate all Menu item inputs when saving.

REQ-4: The System shall update all Menu items when the Business Owner selects “Update Menu”.

REQ-5: The System shall allow deleting Menu items using a delete icon.

4.3.4 Edit Operating Hours

4.3.4.1 Description and Priority

Description:

This feature allows the **Business Owner** to update their stall's **Operating Hours**, including opening time, closing time, and days of operation.

The System displays the current Operating Hours and allows modification for each day of the week.

Priority: High

4.3.4.2 Stimulus/Response Sequences

1. **The Business Owner navigates to the Operating Hours tab.**
 - a. The System displays the current Operating Hours structured by day.
 - b. The System also shows a toggle for **Close Stall** (emergency closure).
2. **The Business Owner may click “Close Stall.”**
 - a. The System marks the stall as closed permanently until reopened by the Business Owner.
3. **The Business Owner may edit one or more fields:**
 - a. Opening time
 - b. Closing time
 - c. Marking specific days as **Closed**
4. **The Business Owner may select any of the Quick Action buttons:**
 - a. “**Set All: 10AM – 8PM**”. System fills all seven days with Opening 10:00 AM and Closing 8:00 PM.
 - b. “**Close Sundays**”. System unchecks the “Open” checkbox for Sunday.
 - c. “**Weekend Extended**”. System sets Friday and Saturday to Opening 8:00 AM and Closing 10:00 PM.
5. **The Business Owner selects “Save Operating Hours.”**
 - a. The System validates all inputs.
 - b. If valid, the System saves the updated Operating Hours.
6. **The System displays a confirmation message.**

4.3.4.3 Functional Requirements

REQ-1: The System shall allow Business Owners to edit Operating Hours for each day of the week.

REQ-2: The System shall allow Business Owners to mark the stall as closed using the **Close Stall** toggle.

REQ-3: The System shall provide **Quick Action** options to auto-fill or modify Operating Hours (e.g., Set All, Close Sundays, Weekend Extended).

REQ-4: The System shall validate all edited Operating Hours before saving.

REQ-5: The System shall save updated Operating Hours after successful validation.

REQ-6: The System shall display a confirmation message after saving.

4.3.5 View Reviews

4.3.5.1 Description and Priority

Description:

This feature allows Business Owners to view all Consumer reviews submitted for their Hawker Stall. Business owners can filter the view by number of stars or sort it through various categories, such as Most Recent, Highest Rating, and Lowest Rating.

Priority: Medium

4.3.5.2 Stimulus/Response Sequences

1. The Business Owner selects the “View Reviews” tab.
 - a. The System loads the Business Owner stall review statistic and review list.
2. The System displays a summary section, including average rating, total number of reviews, and star distribution bars.
3. Business Owner may filter reviews by rating using the respective buttons.
 - a. The System updates the list to show only reviews matching the selected rating.
4. Business Owner may sort reviews using the Sort dropdown.

- a. The System rearranges the review list accordingly.
5. The System displays the filtered and sorted list of reviews.
- a. If no reviews match the selected filter or sort, the System displays a “No reviews found for the selected filters.”

4.3.5.3 Functional Requirements

REQ-1: The System shall display all reviews submitted for the Business Owner’s stall.

REQ-2: The System shall show a star-rating summary, including average rating and distribution.

REQ-3: The System shall allow filtering reviews by rating.

REQ-4: The System shall allow sorting reviews.

REQ-5: The System shall update the review list based on the selected filter or sort.

REQ-6: The System shall display review details, including consumer username, rating, text, images, and date.

REQ-7: The System shall display a “No reviews found” placeholder when applicable.

REQ-8: The System shall restrict Business Owners from modifying or deleting reviews.

4.3.6 Consumer View Preview

4.3.6.1 Description and Priority

Description:

This feature allows the Business Owner to preview how their stall profile will appear to Consumers.

The preview mirrors the actual Consumer-facing layout, including stall name, description, menu items, reviews, operating hours, and any uploaded images.

The preview is read-only and exists solely to help Business Owners verify that their updates are displayed correctly.

Priority: Medium

4.3.6.2 Stimulus/Response Sequences

1. The Business Owner navigates to the **Consumer View Preview** tab.
 - a. The System retrieves the stall's current profile details, including:
 - Stall name
 - Description
 - Operating hours
 - Menu items
 - Reviews
 - Images
 - Status (open/closed)
2. The System renders the stall page in the same format as a Consumer would see it.
3. The Business Owner reviews the displayed information.
4. If any details appear incorrect, the Business Owner may return to the corresponding edit tabs (Profile, Menu, Operating Hours) to make adjustments.
5. The System does not allow editing while in Preview mode.

4.3.6.3 Functional Requirements

REQ-1: The System shall display the Business Owner's stall profile using the same layout as the Consumer stall view.

REQ-2: The System shall retrieve all relevant stall information for display, including menu, operating hours, images, and reviews.

REQ-3: The System shall render the preview as read-only with no editable fields.

REQ-4: The System shall update the preview after the Business Owner makes changes in other sections.

REQ-5: The System shall show a placeholder message if any required stall information is missing.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. The Near Me geolocation feature shall return results within 3 seconds, even with 500 concurrent location-based queries.
2. The system must be able to support a minimum of 5,000 concurrent active users without degradation in performance.
3. Every single page should load in under 2 seconds, with the exception of the CorpPass login.
4. Users must be able to upload a file of up to 20 MB at a minimum rate of 1Mb/s.
5. Rendering of images such as menu items and stall photos shall utilize a Content Delivery Network (CDN) to ensure images load within 0.5 seconds for users globally.

5.2 Safety Requirements

1. The system should implement a process for performing daily, full backups of the database.
2. All images uploaded either by Consumer or Business shall be processed by an automated content moderation tool to detect and flag high-risk content, including: Graphic Violence, Adult Content (Pornography/Nudity), and Illegal Acts.
3. The system's sign-up page and the review submission page must include conspicuous links to the platform's Terms of Service and Privacy Policy to ensure users are aware of how their data and content are handled.

5.3 Security Requirements

1. The system shall anonymize Consumer location data after the immediate service request (e.g., finding nearby stalls) is fulfilled, and shall not store precise historical location data linked to the Consumer's.
2. The system must adhere to all local Singapore data protection laws, including the Personal Data Protection Act (PDPA), specifically regarding the collection and use of Consumers' email and location data.
3. Data compromises will not be tolerated, and the database should be secured by ensuring access to key personnel only, such as developers working on the codebase.

5.4 Software Quality Attributes

1. The code must always include useful comments regarding its purpose to ensure interpretability, code reusability, and maintainability.

2. The API endpoints shall include documentation regarding the endpoint URL, required request body, and the expected response body.

5.5 Business Rules

1. A customer can only post one single review per stall.
2. Only verified business accounts can edit their stall profile.

6. Other Requirements

6.1 Internationalisation Requirements

1. HawkerSG will initially support only the English language for both consumer and business users.
Future iterations may consider expanding support to common local languages or dialects to enhance accessibility for Singapore's broader, diverse population.
2. All date, time, and location representations should follow the Singapore standard (GMT+8) to ensure clarity and consistency across all user interactions.

6.2 Legal Requirements

1. The system must comply with Singapore's Personal Data Protection Act (PDPA) when collecting, storing, or processing user information such as email addresses, reviews, and business data.
2. All submitted reviews and business information must adhere to local digital content regulations, including restrictions on abusive language, inappropriate content, or defamatory statements.
These rules are enforced through the LLM-based moderation system.

6.3 Reuse Objectives

1. Core components such as authentication, review handling, and stall information management should be designed to be modular and reusable for future extensions of HawkerSG or similar community-based platforms.
2. All backend APIs should follow REST standards, ensuring they can be integrated easily with mobile applications or extended third-party services in future phases.

6.4 Accessibility Requirements

1. The system should meet fundamental web accessibility guidelines such as clear text contrast, keyboard navigability, and descriptive alt-text for images, ensuring that

HawkerSG remains usable by individuals with visual or motor impairments.

2. Interactive elements such as search filters and dropdowns should remain usable on both desktop and mobile screens without requiring precision gestures.

6.5 Database Requirements

1. The database must support efficient retrieval of stall information, reviews, favourites, and business profiles with minimal latency.
2. The system should maintain referential integrity between entities such as Users, Stalls, Reviews, and Hawker Centres.
3. All sensitive user information stored in the database must be hashed or encrypted where appropriate.

6.6 Maintainability Requirements

1. The codebase should follow consistent naming conventions, folder structures, and documentation practices to support ease of debugging and onboarding new developers.
2. The backend project must include type hints and docstrings to improve long-term maintainability.

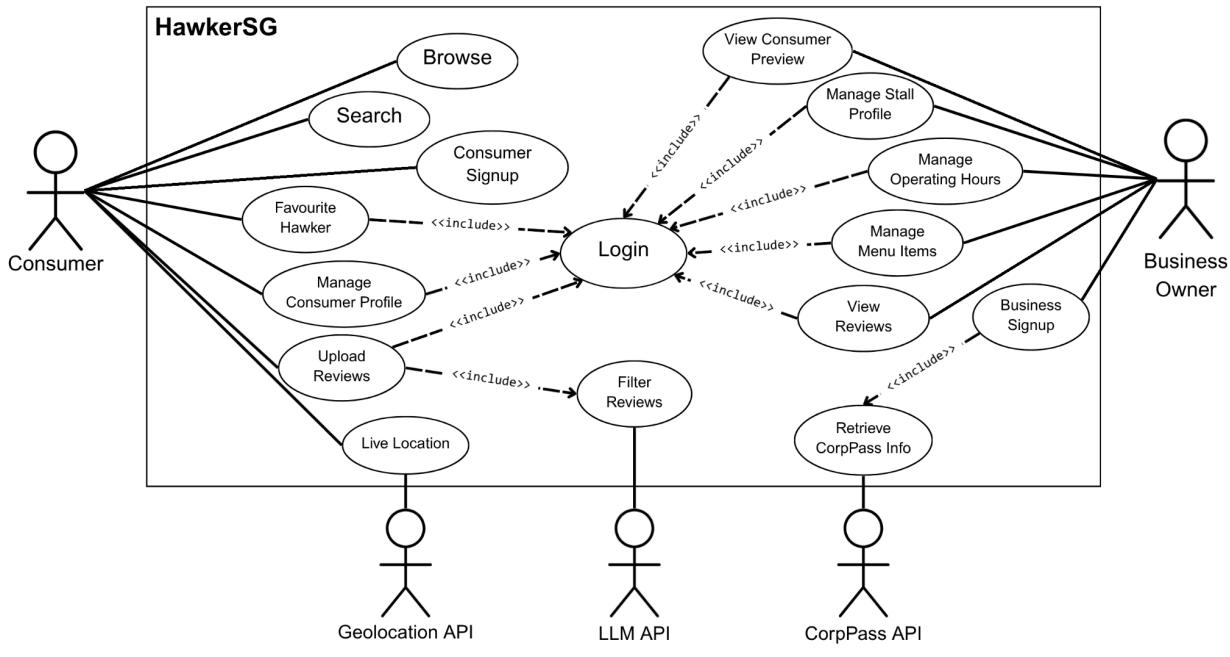
6.7 Reliability Requirements

1. The system should remain operational and accessible at least 95% of the time, excluding planned maintenance.
2. Critical features such as login, review retrieval, and stall browsing should fail gracefully with proper error messages in case of service interruptions.

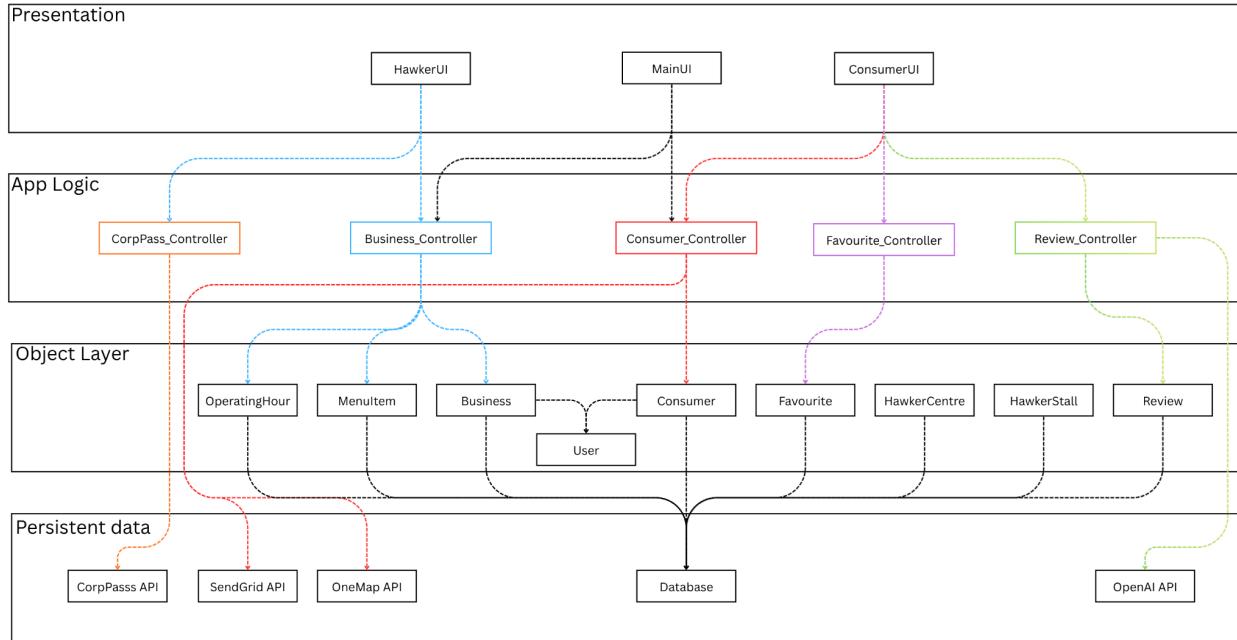
Appendix A: Glossary

Acronym / Abbreviation	Meaning
API	<p>Application Programming Interface</p> <ul style="list-style-type: none"> - A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.
Front-end	Front-end is what the user sees and interacts with in the application
Back-end	Back-end is the server-side components of an application that run behind the scenes to power what the user sees and interacts with on the front end
PDPA	<p>The Personal Data Protection Act (PDPA)</p> <ul style="list-style-type: none"> - PDPA provides a baseline standard of protection for personal data in Singapore.
SRS	<p>Software Requirements Specification (SRS)</p> <ul style="list-style-type: none"> - A detailed document outlining all the functional and non-functional requirements for a software project.

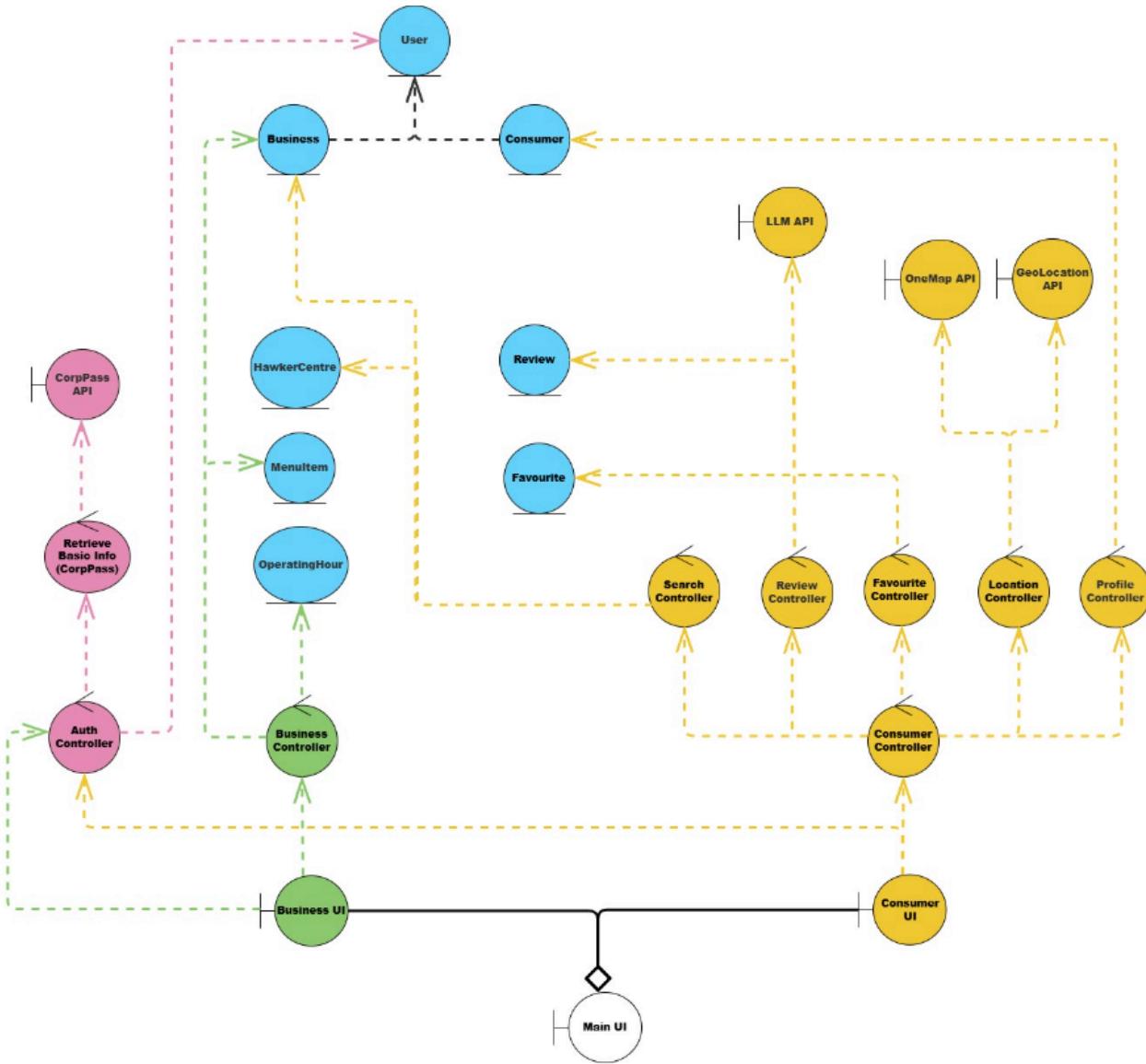
Appendix B: Analysis Models



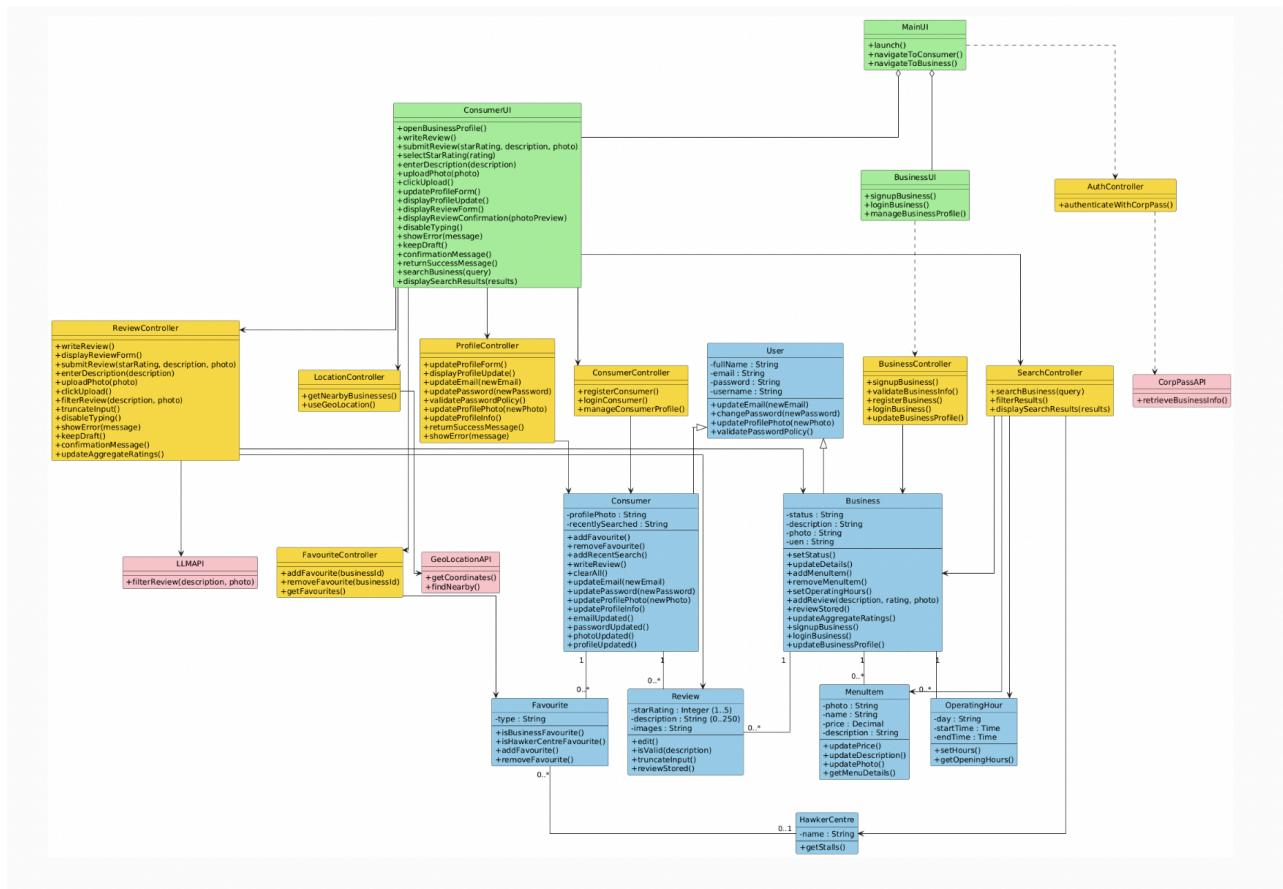
Use Case Diagram



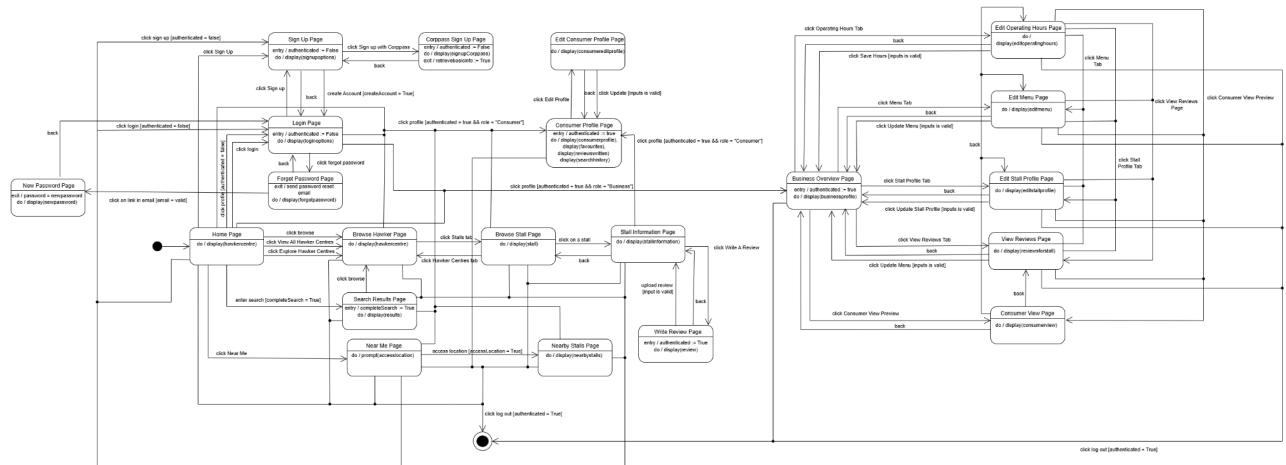
Architecture Diagram



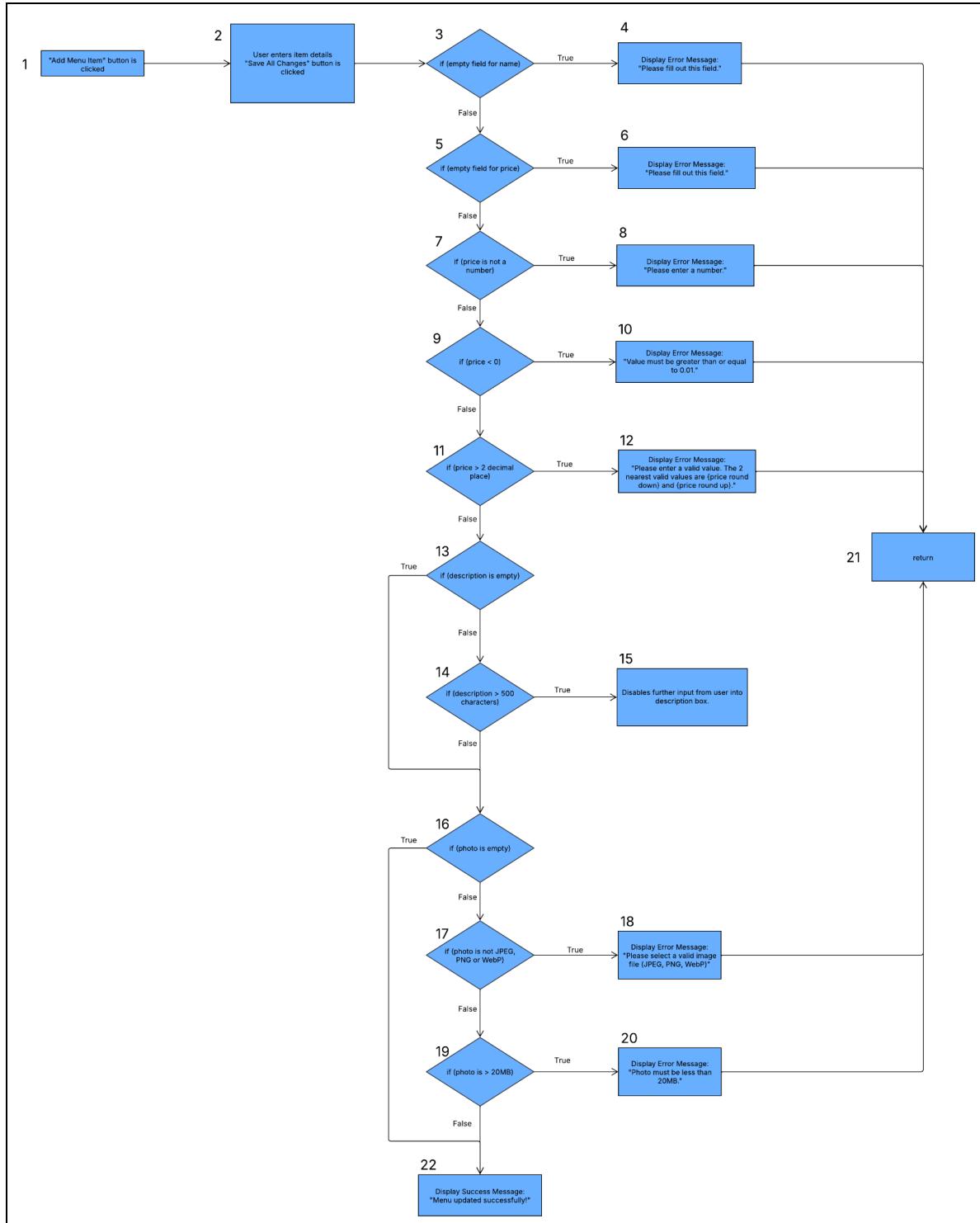
Key Boundary Classes and Control Classes



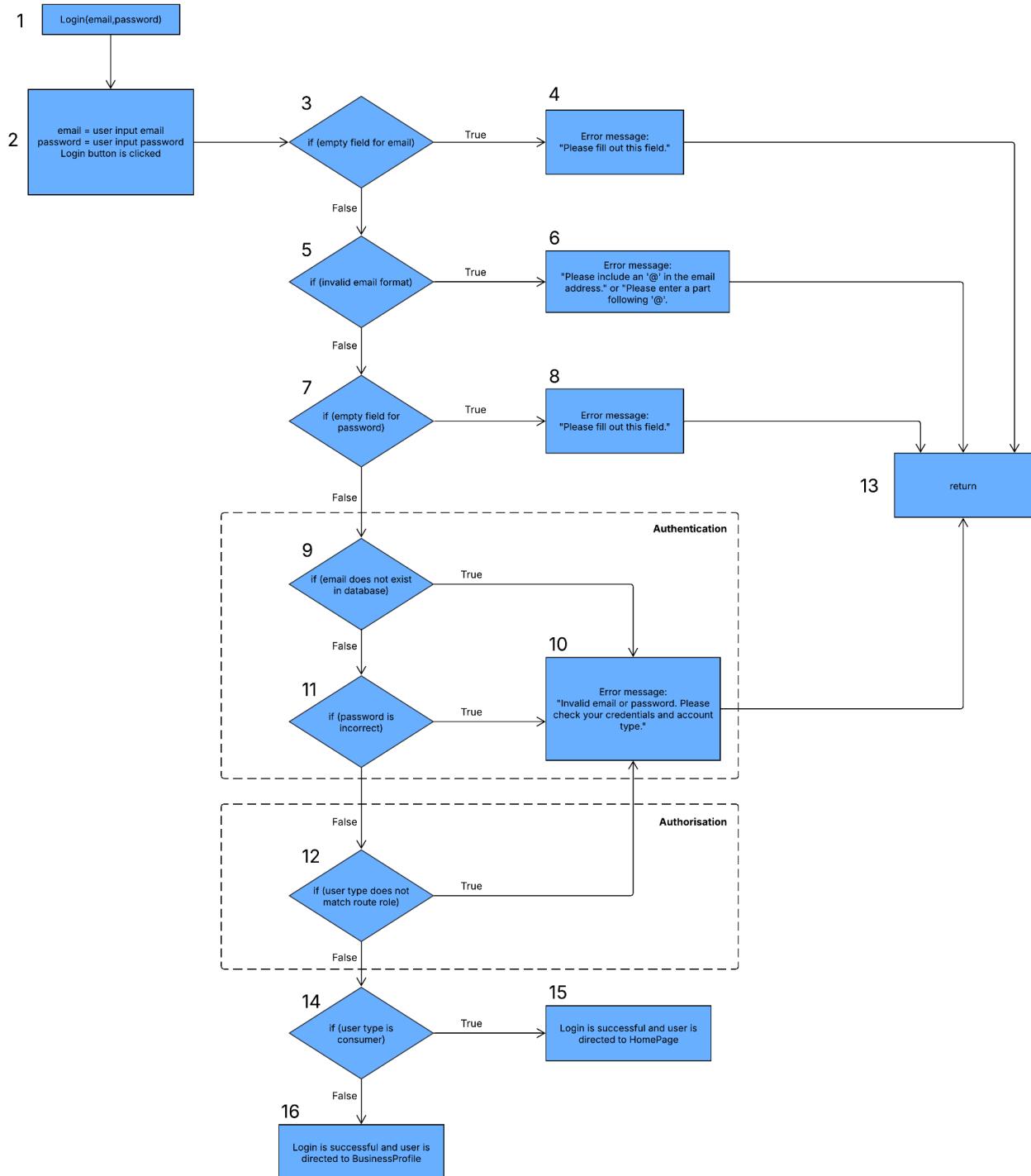
Class Diagram of Entity Classes



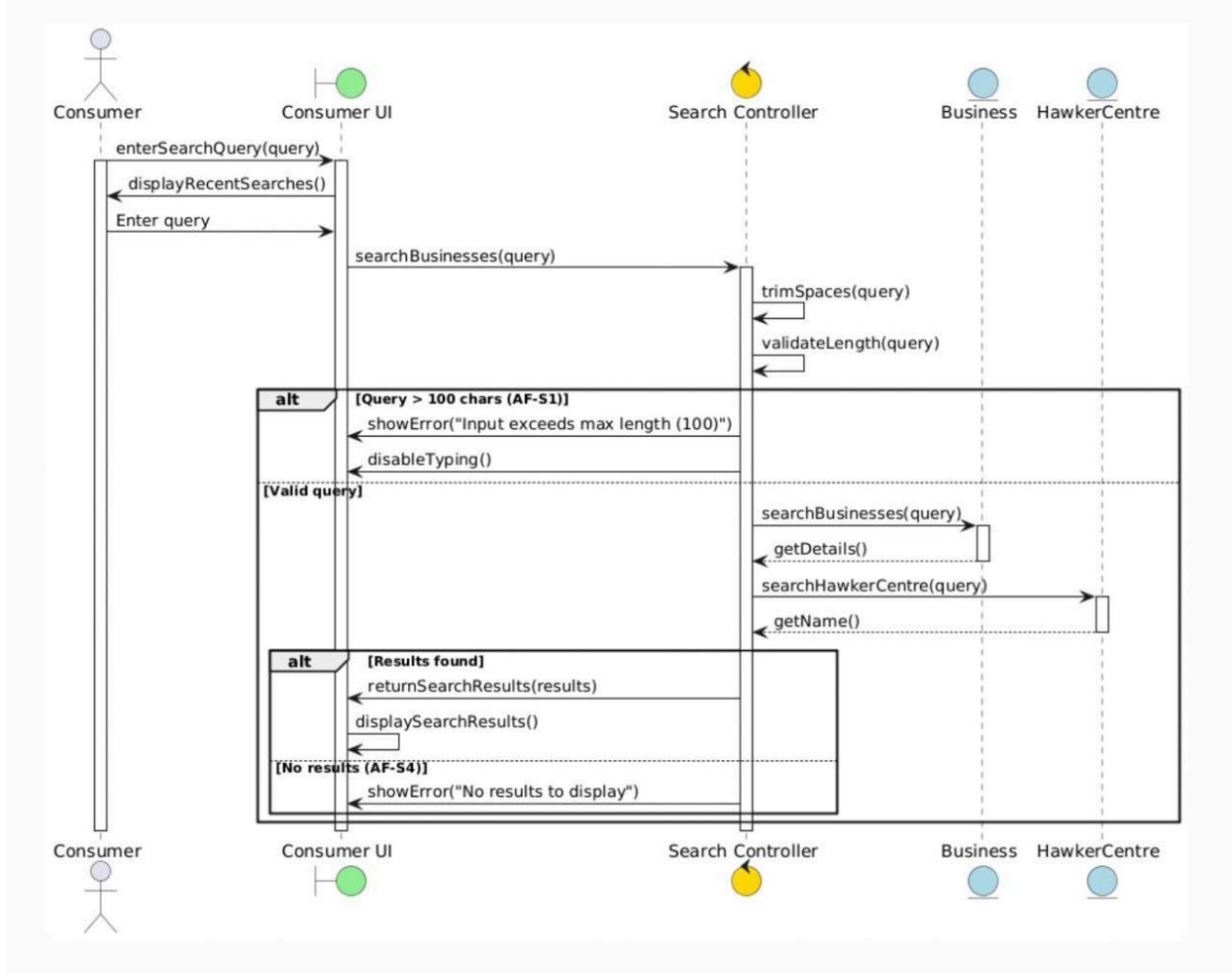
Dialog Map



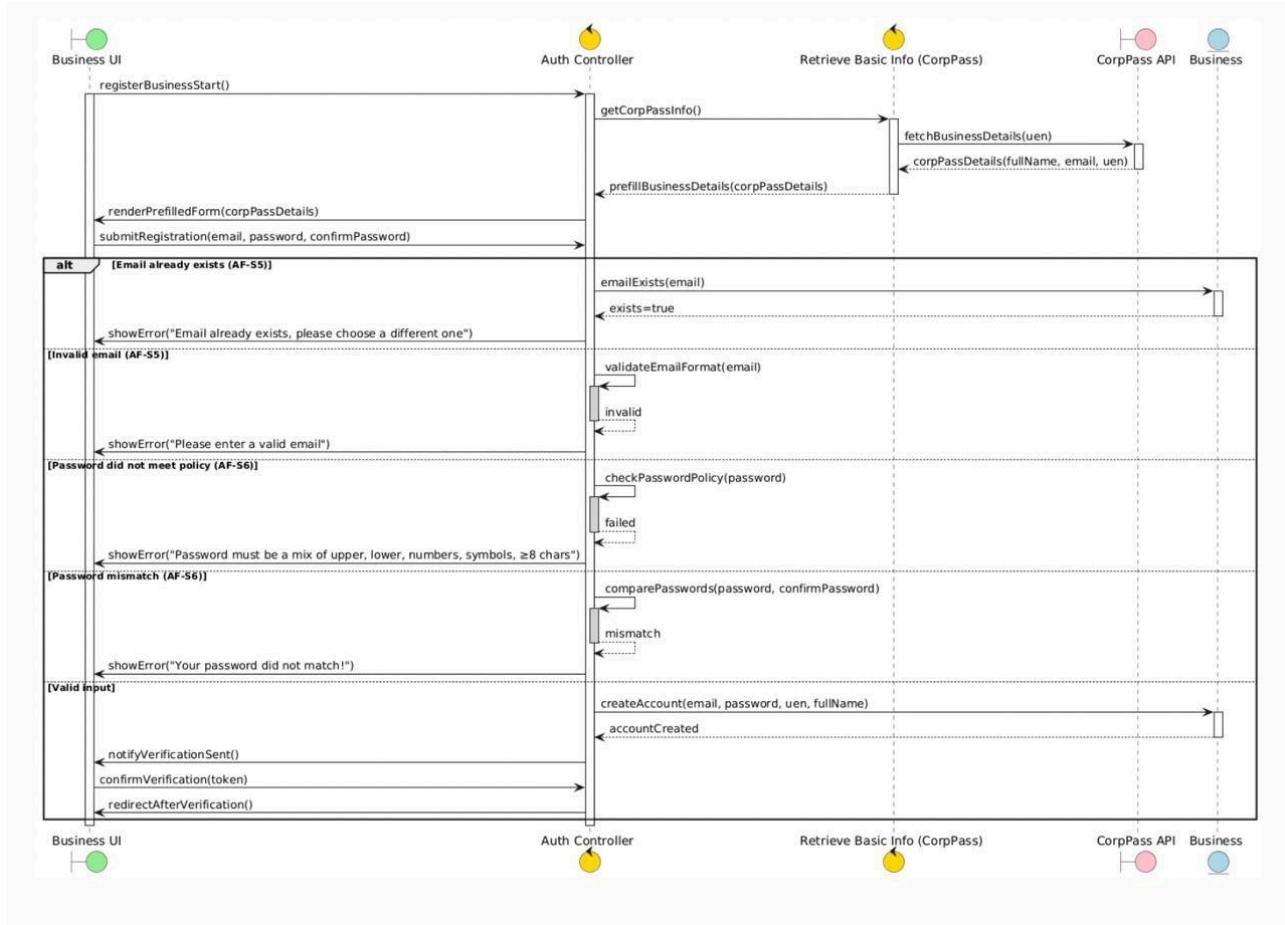
Control Flow Graph - AddMenuItem



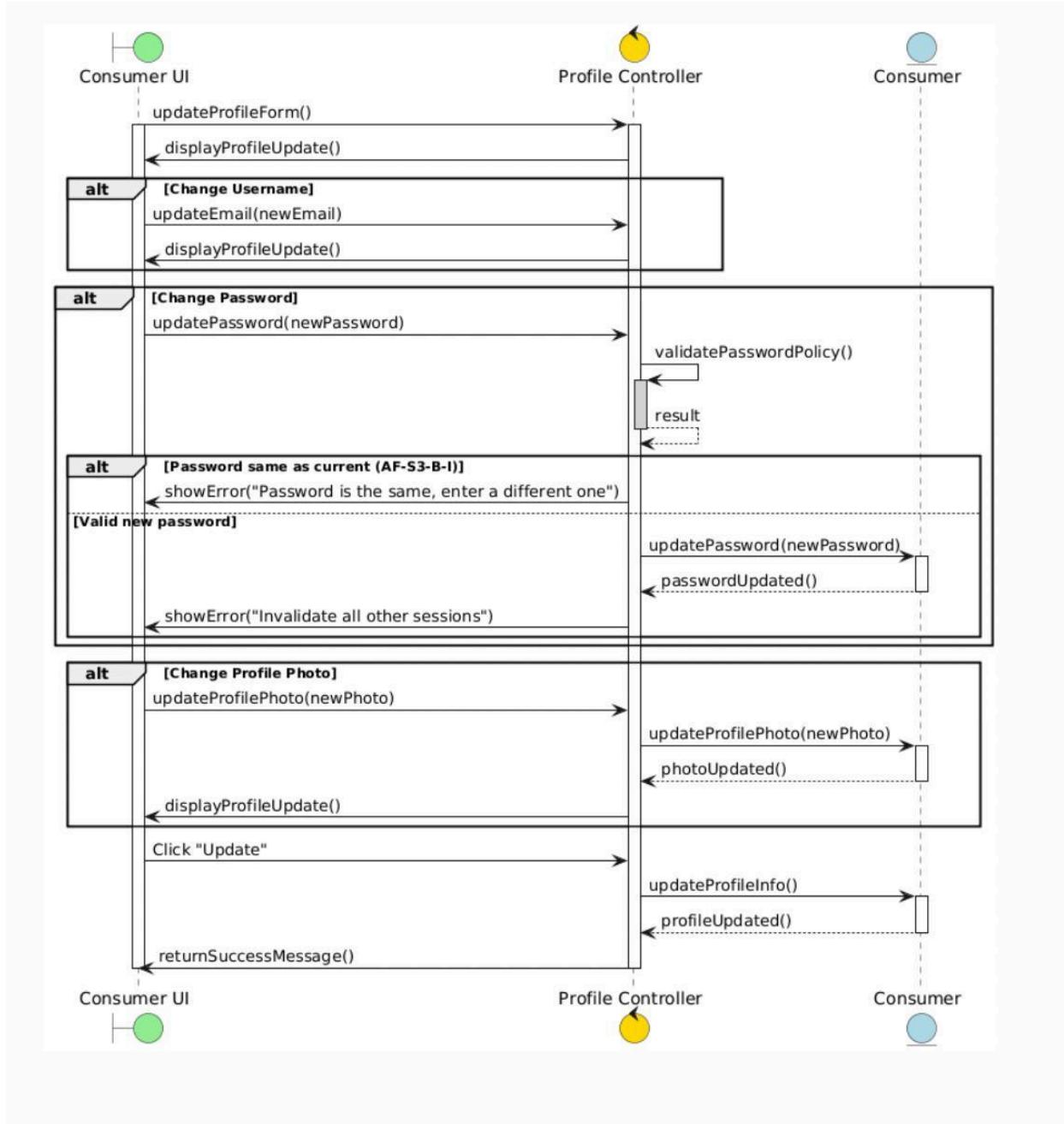
Control Flow Graph - Login



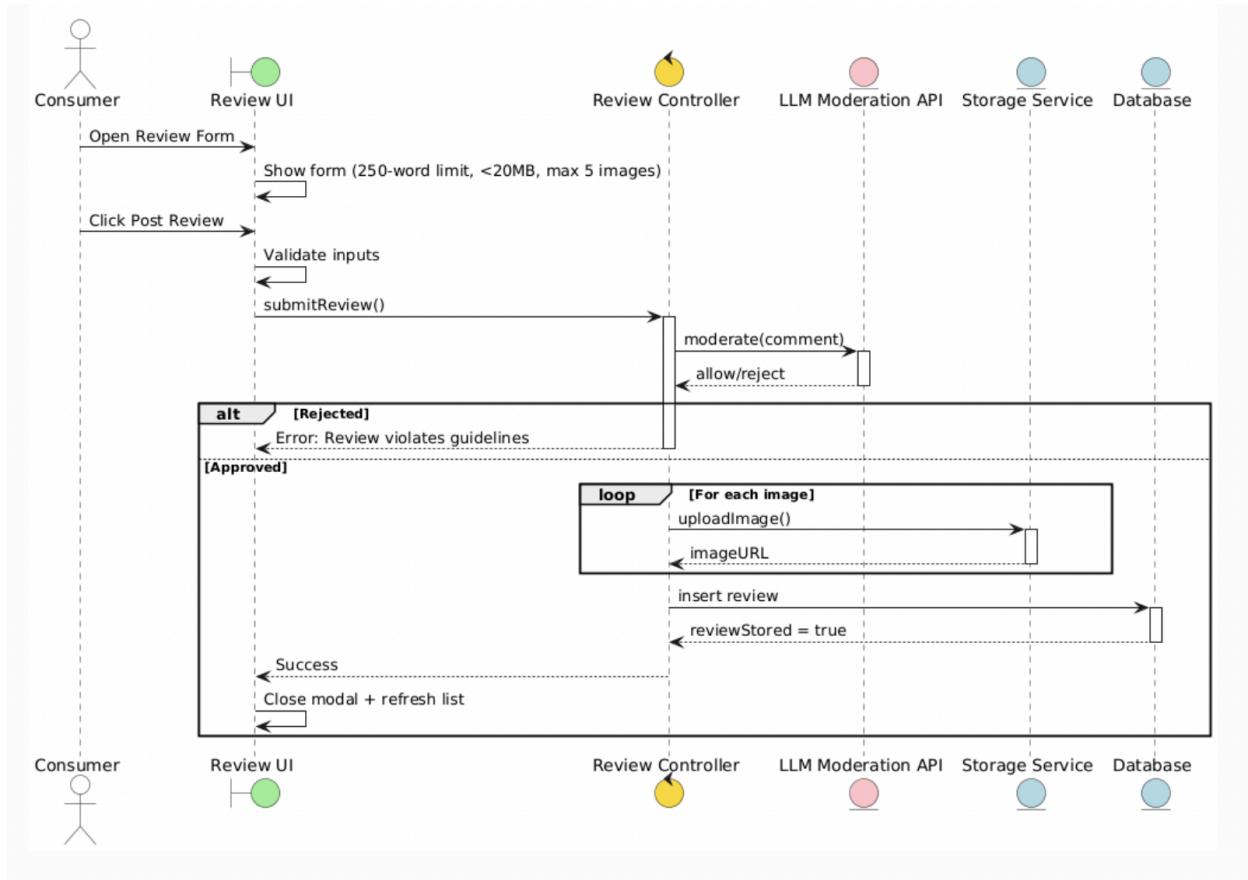
Sequence Diagram - Search Use Case



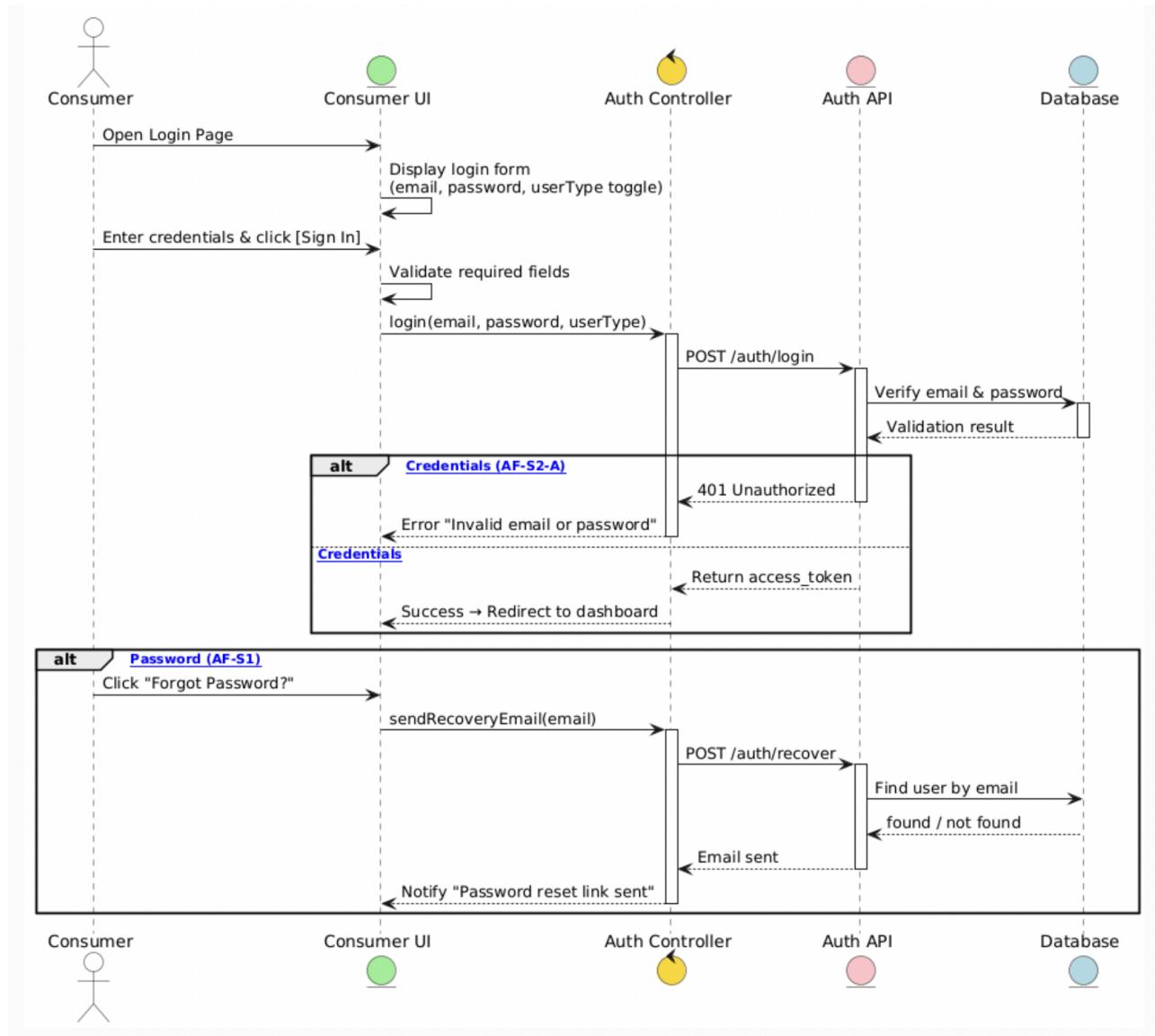
Sequence Diagram - Business Sign Up



Sequence Diagram - Manage Consumer Profile



Sequence Diagram - Upload Reviews



Sequence Diagram - Log in

Appendix C: To Be Determined List

There are no TBD items for this current version of SRS.

Source: http://www.frontiernet.net/~kwiegers/process_assets/srs_template.doc