Software Requirements Specification

PRJ 566/666 - Summer 2025

PRJ 566/666 – Team No:

02

Name of Project:

BiteClub

Project Leader:

Cesca Dela Cruz (current)

Last updated:

August 13th, 2025

Team Members:

1. Olha Chovhaniuk 102229150
2. Cesca Dela Cruz 123123150
3. Irish Banga 112435227
4. Jashanpreet Singh 112454228
5. Ka Ying Chan 123231227

TABLE OF CONTENTS

1. Introduction/Overview - Document Information

1.1 Document Authors

Olha Chovhaniuk
Cesca Dela Cruz
Irish Banga
Jashanpreet Singh
Ka Ying, Chan

1.2 Revision History

Week 03	Sections of this document that were completed this week 1. Introduction/Overview (completed) 1.1 Document Authors (completed) 1.2 Revision History (completed) 1.3 Document Conventions (completed) 1.4 Document Purpose (completed) 1.5 Intended Audience (completed) 1.6 Group Agreement (completed) 2. Project Overview 2.1 Project proposal (completed)
Week 04	2.2 Stakeholders and Users (completed) 2.3 Project Scope (completed) 2.4 System Risks (completed) 2.4 Operating Environment (completed)
Week 05	2.6 Functional Requirements (completed) 2.7 Non-Functional Requirements (completed) 3. Process & Data Modeling 3.1 UML Modeling (DFDs only) (completed)
Week 06	2.8 UI/UXD Interface Mockups (completed) 3. Process & Data Modeling 3.1 UML Modeling (ADs only) (completed)
Week 07	3.2 Use case specifications: 3.2.1 Business Rules (completed) 3.2.2 System Use Case Diagrams (completed) 3.2.3 Use Case Descriptions (completed) 3.2.4 Corresponding interface mockups (completed)
Week 08	2.8 UI/UXD Interface Mockups on Figma – Finalize (completed) Video development (completed)
Week 09	4. Domain Class Diagram (completed)
Week 10	5. NoSQL Data Model and Sample JSON Data

	5.1 Data Model (completed) 5.2 Sample JSON Data (completed)
Week 11	6. Work breakdown Structure (WBS) - (completed) 7. Milestones & Acceptance Criteria - (completed)
Final	Implementation Schedule - (completed)
Week 14 – PRJ 666	5.1 Data Model

1.3 Document Conventions

Convention	Explanation	Example
Text in Red (RGB: 255, 0, 0)	Indicates exceptions, errors, or critical issues.	"Error: File not found."
Text in Blue (RGB: 0, 0, 255)	Represents tasks or sections that are in progress.	"In-progress: Section 1.3"
Text in Green (RGB: 0, 176, 80)	Emphasizes important points or key notes.	"New feature: Cuisine challenges."
Text Highlighted in Yellow	Represents recently added content or updates.	"Important: Backup your files regularly."
Italicized Text	Used for definitions, glossary terms, or a shop name.	Definitions: *Sustainability*: the quality of being able to continue over a period of time. Glossary Terms: *Foodie*: A gourmet, or a person who eats food as hobby. Shop name: *Google Maps* is primarily designed to navigate.
~~Strikethrough Text~~	Indicates content that has been deleted or is dated.	~~"We do not need this feature."~~

1.4 Document Purpose

The purpose of the document is to serve as a project guide and reference for the development and implementation of a web application. The document will layout key requirements, deliverables, and

design details for the project. It will serve as a structured reference for all stakeholders and will include functional and non-functional requirements, system design, database structure, and UI/UX specifications. It will ensure consistency in development, define the work structure, and help track progress throughout the project lifecycle.

By documenting these details, the project team can effectively work on the web application and ensure that the final product meets user expectations and previously set objectives. This document will be used to address any conflicting issues and will be the final authority on project requirements, scope, and design decisions.

1.5 Intended Audience

UI/UX Designers: Designers can refer to this document to understand what kind of experience we aim to deliver to the customer. This will allow them to create a consistent and goal-centric design. This document also consists of functional and non-functional requirements of the application which they can refer to while developing the design.

Frontend Developers: Frontend developers can refer to the functional and non-functional requirements and create an appropriate frontend.

Backend Developers: The requirements specified in this document will aid in system design. Backend developers will also contribute Database Artifacts to this document.

Project Manager: The project manager can appropriately delegate work to the resources to meet the success criteria in the allocated time. They can plan out the work based on the requirements specified in the document.

Client: The client can refer to this document to understand the project that is going to be undertaken and ensure that all of the client requirements are met.

1.6 Group Agreement

TEAM AGREEMENT

Team #: 02

Project Title: BiteClub

Project Time Frame: 8 months

Team Members:

- Olha Chovhaniuk
- Cesca Dela Cruz
- Irish Banga
- Jashanpreet Singh
- Ka Ying, Chan

Team Leadership:

Leadership role will rotate among team members every 3 weeks. The leader will be responsible for task distribution, organizing meetings and tracking deliverables.

PRJ 566/PRJ 666

- Olha Chovhaniuk
- Cesca Dela Cruz
- Irish Banga
- Jashanpreet Singh
- Ka Ying, Chan

Team Functions:

- *The team will share information through MS Teams, OneDrive, e-mail and meetings.*
- *Team leader will be responsible for fair division of tasks amongst team members.*
- *Each team member is responsible for completing their tasks and submitting them by the deadline set by the team leader.*
- *Team members will review and provide feedback on each other's work before final submissions.*

Team Meetings:

- Team meetings will be held at least once a week apart from the mandatory class meetings.
- The default meeting time is Wednesdays from 11:00 AM to 12:30 PM. (Subject to change by group consensus.)
- Attending team meetings is mandatory. In case a member is unable to attend, they must inform the team leader in advance.

Team Problems:

- Any conflicts will first be addressed via discussion in team meetings or by notifying the team leader.
- Non-participating members will receive two warnings before being reported to the professor.
- Further violations may result in expulsion from the group which will be decided by a majority vote.
- Any disputes about tasks or the project in general will be resolved through team voting.

Team Commitment

The undersigned members agree to work together on the project until the end of the PRJ666 next Semester. They recognize that as a team and individually they are responsible for the quality of all deliverables.

Name	Date
Olha Chovhaniuk	21 January, 2025
Cesca Dela Cruz	21 January, 2025
Irish Banga	21 January 2025
Jashanpreet Singh	18 January, 2025
Ka Ying, Chan	21 January 2025

2. Project Overview

2.1 Project Proposal

Project Background

Most people have had such an experience: At first, they do not dare to try something new, but they are glad they did once they do. Or, they might be unsure about what dish to choose for the night, so they ask for the chef's recommendation, and they end up loving it.

However, a study conducted by the British restaurant Prezzo in 2024[1] found that 86% of Gen Z adults (aged 18-24 in this study) admitted to suffering from "Menu Anxiety." This refers to the worry of not being able to find a dish that suits their palate on the menu, often leading to regret after ordering.

On the other hand, with the overwhelming variety of food options available, some people experience food decision fatigue. 48% of Gen Z reportedly feel anxious most of the time due to information overload. While some individuals are brave enough to start a food adventure and try new dishes, others prefer to stick with the same food to avoid deciding.

Living in Toronto, we are fortunate to embrace diversity and culture. The city is a melting pot of various ethnicities, offering an extensive range of cuisines to explore. While numerous mobile applications and

websites, such as *Google Maps*, *Yelp*, *Uber Eats*, *TasteToronto*, and *OpenTable*, provide food-related services with features such as navigation, reservations, user reviews, food delivery, and more, there is currently no solution for people who experiencing menu anxiety and food decision fatigue.

A study by Canadian researcher Olivia Bush revealed that the most active age group on social media platforms is 25-34, falling within the Millennials and Gen Z demographic. One of the primary reasons for using social media is to find inspiration for things to do and buy, accounting for 26% of users. Additionally, as of March 2023, another survey carried by Statista.com shown that nearly 50% of millennials worldwide found influencers' recommendations of a brand or product more engaging than traditional advertisements.

We are aware that Millennials and Gen Z are highly receptive to social media challenges and trends as a way to establish a sense of belonging and engagement. Our team aims to take advantage of this phenomenon by carrying this ambience into our platform, *BiteClub*, bringing a positive influence from social media and KOLs, and combining interesting AI gamification features to create tailored food challenges, allowing users to explore unique cuisines or hidden gems they have never tried before with confidence.

Problem Statement

The Problem of:	feeling anxious and/or experiencing decision fatigue when choosing what and where to eat
Affects:	people, specifically Gen Z and Millennials in the Greater Toronto Area (GTA), where options are abundant.
The impact of which is:	stress and dissatisfaction when dining out or ordering in. As a result, they often resort to ordering familiar dishes, missing opportunities to try new foods. This confines their exposure to a wider variety of cuisines and limits their appreciation of the diverse cultures in the city.
A successful solution would:	reduce menu anxiety and food decision fatigue, helping users feel confident with their choices. It would also empower users to try new dishes, allowing them to embrace the melting pot of cultures that the GTA has to offer.

Product Vision

For	Gen Z and Millennials in the Greater Toronto Area (GTA)
Who	experience menu anxiety and/or decision fatigue when choosing what and where to eat.
The Product Name	Our web application, *BiteClub*
That	leverages the power of social media, influencer culture, and AI to improve the dining experiences of users.
Unlike	traditional food review applications, such as *UberEats*, *Yelp*, or *Google Maps*

Our product	streamlines the decision-making process by sharing opinions of influencers on our platform and more importantly, by using AI-assisted gamification. This essentially makes the choice for them through food challenges, but also encourages them to try new dishes, giving them the chance to celebrate and appreciate the multitude of cultures in the city.

2.2 Stakeholders and Users

Stakeholder Name/Identifier	Category
CEO (Chief Executive Officer)	Administration, Sponsor
Product Owner	
Project Manager	Administration
System Administrator	
Security Analyst	
Legal Expert	
Marketing Manager	
Photographer	Administration
UI/UX Designer	Designer
Project Leader	Developer
Frontend Developer	
Backend Developer	
Business Analyst	Analyst, User
Content Moderator	Moderator, User Reviews claims submitted by users regarding inappropriate content and takes necessary action.
Quality Assurance (QA)	Tester
Test Participants for Beta, Usability, and Accessibility Testing and Early Feedback Providers	Test Participants

Restaurant Owners and Managers	External User
Restaurant Staff	
Customer	
General User	

2.3 Project Scope

Project Objectives

This project aims to design and develop a food discovery web application that helps users broaden their palate and appreciate the diverse cuisines in Toronto by reducing anxiety and decision fatigue when choosing where and what to eat. By implementing this web application, we hope to achieve the following:

- Reduce anxiety and fatigue during decision-making
- Decrease the time it takes to decide where and what to eat
- Simplify the discovery process of new restaurants and hidden gems
- Motivate users to try new restaurants and cuisines
- Foster a sense of community among users
- Increase overall satisfaction of dining experiences

The success of these goals will be evaluated during user testing (see *Success Criteria* for more details).

Deliverables

1) *Completed Web Application*
    1. Fully developed and deployed web application
2) *Design and User Interface*
    1. Interface wireframes and mock-ups
    2. System use case diagrams and use case descriptions
    3. Presentation slide deck
    4. Video presentation
3) *System Architecture*
    1. Domain class diagram
    2. NoSQL data models with corresponding documents
4) *Project Management Documents:*
    1. Work Breakdown Structure
    2. Project milestones and acceptance criteria
5) *Agile Artifacts*
    1. Product backlog
    2. Sprint backlog
    3. User stories

6) *Project Submission*
    1. Final presentation slide deck
    2. Final video presentation

Project Boundaries

- While the application will serve two main user types—restaurants and restaurant goers—our designs, documentation, and functionality will primarily focus on the restaurant goer's perspective.
- Design and develop a web application with the following components:
    1. AI-powered restaurant recommendation engine
    2. Search and results filtering options
    3. AI-generated food challenges with real-time geolocation verification
    4. User-generated content and community engagement (restaurant reviews and user blogs)
    5. Integration of reviews from Instagram
    6. User-initiated reporting of inappropriate reviews
    7. Mandatory restaurant and user accounts, along with profiles
    8. User curation of visited restaurants, favourites and wish lists
    9. Real-time restaurant reservations and pre-ordering options
    10. Payment integration for pre-orders
- Implement measures to secure users' sensitive data, particularly password hashing, ensuring privacy is maintained.
- Ensure the application is responsive across various screen sizes and accessible to users with disabilities.

The following items are out of our scope:

1. *Country-wide or international expansion*: The application will only be focusing on restaurants and users in the Greater Toronto Area.
2. *Mobile application development*: The project will focus on developing a web application only.
3. *Social media beyond Instagram*: Integration of other social media platforms like Facebook, Twitter, or Yelp will not be implemented.
4. *User-generated video content*: The ability for users to upload video reviews or content will not be included.
5. *Food pick-up and delivery*: The application will only allow users to pre-order food if they make a reservation for in-person dining.

Project Constraints

- The project will be executed with no allocated financial budget.
- All phases of the project, from requirements gathering to deployment must be completed in less than eight months.
- Communication with restaurants will be limited, if it occurs at all, during the project. The development of the application will primarily rely on publicly available data, such as APIs.

- Some data we require may be unavailable publicly.
- Limited knowledge of certain technologies, particularly AI, may restrict the complexity and functionality of features. Relying on third-party solutions may be necessary.

Project Assumptions

- Restaurant data will be accessed from external APIs.
- Any missing data, such as menu details, will be fabricated as necessary for demonstration purposes.
- Auth0 is a potential method for login and authentication.
- Manual content review by the project team is not included in the project scope. Post-launch, a third-party team will be responsible for reviewing reports of inappropriate content and taking action.

Project Timeline

- Project Start Date: January 6, 2025
- Project End Date: August 15, 2025

Project Risks

- Schedule Risks: Potential loss of team members could impact project timelines, necessitating extra effort from the remaining team to meet deadlines.
- Compliance Risks: Compliance issues with local regulations particularly concerning payment processing.

Resource Requirements

- Personnel:
    - Developers (Frontend, Backend, AI specialist)
    - UI/UX Designer
    - QA Testers
    - Beta Testers
- Technical Resources:
    - Development tools (VS Code, Figma, GitHub)
    - APIs for restaurant data and geolocation features (e.g., Google Maps API)
    - AI: Trained models OR integrating APIs (e.g., OpenAI)
    - Accessibility Testing Tools
    - Hosting platform (e.g. Cloudflare, Vercel)

Quality Standards

- Ensure the application adheres to accessibility guidelines.
- All user data, including passwords, must be securely encrypted.
- Robust and user-friendly interface for easy navigation and responsiveness.
- Perform rigorous testing to ensure minimal downtime during deployment.

Change Control Process

- Proposals for changes will be reviewed and discussed in team meetings.
- Approve changes after evaluation based on their impact on scope, schedule, and resources and team consensus.
- Document all approved changes and update the team and major stakeholders.

Dependencies

- Availability and reliability of third-party APIs (Google Maps API, OpenAI, etc.)
- Availability of participants for beta testing and early feedback, as well as usability and accessibility evaluations.

Success Criteria

- Completion of all deliverables, including deployment and documentation.
- Sign-off from stakeholders on final deliverables.
- Results from the final user surveys and usability tests show the following:
    - 60-75% reduction in decision anxiety and fatigue
    - 50-65% decrease in decision-making time
    - 45-70% decrease in time spent discovering new restaurants and hidden gems
    - 40-60% increase in users' likelihood to try new restaurants and cuisines
    - 89-97% satisfaction score for overall experience with web application

2.4 System Risks

Risk	Response
AI generated challenges could be inaccurate, possibly due to AI hallucinations.	Thorough testing and continuous monitoring to ensure proper behavior.
Potential inconsistencies across different platforms resulting from differences in browsers, operating environments etc.	Test the application on a variety of operating environments with different screen sizes, operating systems, browsers etc.
Security risk of user data due to potential inadequacies in security system.	Follow industry best practices and have security professionals take a second look at the system.
Map integration and other unfamiliar feature implementations may be time consuming and cause delays in project delivery.	Have an appropriate plan to understand the underlying technologies and efficiently implement related features. The plan can be used for predicting how long certain tasks may take.

Necessary restaurant data may be unavailable publicly or have restricted access due to API limitations.	Data will be mocked for the prototyping purposes if we are unable to access actual data.
Low user adoption, insufficient feedback and community engagement could reduce application effectiveness.	We will proactively reach out to have people test run the application and provide feedback.

2.5 Operating Environment

Hardware Requirement

PC

- Processor / CPU: Intel i3 (or equivalent AMD) or better
- RAM: 4GB minimum, 8GB recommended
- Storage: at least 100MB for browser caching and temporary storage, generally sufficient for most web applications with maps integration.
- Graphics: Basic integrated GPU
- Network: Minimum 5 Mbps for basic functionality. 10 Mbps or higher recommended for map rendering performance.

SmartPhone

- Mobile Compatibility: Android 7+ or iOS 14+

Software Requirement

Operating System

The website will be accessible across these OS platforms since they all support modern web browsers. There are no specific OS dependencies for general web access.

- Windows
- MacOS
- Linux

Web Browser

- Google Chrome (89+)
- Mozilla Firefox (88+)
- Microsoft Edge (Chromium based, 89+)
- Safari (14+)

Server-Side Environment

Frameworks

- Frontend:
    - React and Next.js

- Backend:
    - Node.js and Express
- Programming Language:
    - JavaScript (ES6+)

Hosting Providers

- Cloudflare for CDN, security, and caching

Database

- MongoDB (hosted on MongoDB Atlas cloud platform)

Security Handling

- Use Auth0 and JWT for authentication and user session management
- Require HTTPS for secure data transfer

Third-Party Integrations

- Maps API for map rendering
- Social Media API: Instagram
- Related npm dependencies and libraries

2.6 Functional Requirements

1. AI-Generated Food Challenges
    - Users can complete a questionnaire to define their challenge preferences during onboarding, but as they use the app more, challenges are also influenced by their order history and past activity.
    - Users can browse and choose from multiple personalized challenges.
    - Users can accept to participate in three personalized food challenges at most.
    - Users can prompt for different/new personalized food challenges.
    - Users can drop out of their active challenges.
    - Users can manually mark off the progress of their challenges by using geolocation verification.
    - Users can view the total number of challenges they have completed.
    - Users can see where they rank on a leaderboard based on the number of challenges they have completed.
    - Users can earn points from completing challenges.
    - Users can redeem their points for discounts.
    - Users can apply their discounts immediately, should they choose to do so.
    - Users can view the number of points they have earned from completing a challenge.
    - Users can view their current *total* number of points.

2. AI-Personalized Recommendations

- New users receive recommendations based on a questionnaire, but as they use the app more, recommendations are also influenced by their order history and past activity.
- Users can see personalized recommendations upon opening the application.

3. Searching and Filtering Results
    - Users can search for restaurants, users, and blog posts.
    - Users can filter and sort restaurants search based on distance, cuisine, price, rating, dietary restrictions, and open hours.
    - Users can grant permission and provide their current location through GPS to receive location-based results, recommendations and to be able to filter the restaurants search results by distance.

4. Geolocation-Based Visit Confirmation
    - Users can confirm their location by clicking on the “Check-In” button to verify that they have visited the restaurant.

5. Advanced Review System
    - Users can leave detailed reviews for restaurants with an option to add images.
    - Users can also link an Instagram post in the reviews section.
    - Users can edit or delete their reviews from the system.
    - Users can react to other reviews with a like or emojis.
    - Users can report inappropriate or spam reviews.
    - Reviews from users with high engagement scores and credibility will be prioritized under restaurant profiles.

6. Restaurant Accounts and Profiles
    - Restaurant profile will display essential information about the business, such as name, location, contact information, business hours, cuisine, etc.
    - Restaurant users will have limited edit access to their profile.
    - Restaurants can edit their profile to highlight user reviews, add and upload images.
    - Restaurants can also embed Instagram posts and links to blogs to enhance their profile.
    - Reviews left by users will be displayed under restaurant profiles.
    - Restaurants can respond to reviews under their profiles to engage with users.
    - Restaurants can post promotions, events, and announcements.

7. User Accounts and Profiles
    - Users can use email/password or Single Sign-On methods for registration and login authentication.
    - Users can manually log off from the system.
    - Users can view their account details.
    - Users can manage saved lists: favorite restaurant, visited places, wish list spots.

- Users can edit their user information.
- Users can delete accounts permanently.
- Users can view what information is being collected.
- Users must be able to add, edit, and delete their saved Credit Card information.
- Users must be able to set a default payment method.
- Receipts and transaction history must be available to users.
- Users can follow other users.

8. Blog and Community Interactions
    - Users can create a post in the blog section.
    - Users can update their own posts.
    - Users can delete their own posts.
    - Users can react to blog posts with emojis.
    - Users can Embed Multiple Images in Blog Post
    - Users can integrate Instagram Posts
    - Users can comment on blog posts.
    - Users can reply to comments on blog posts.
    - Users can report inappropriate posts or comments.
    - Users can tag a restaurant in their blog post.

9. Reservations
    - Users can reserve a table at a restaurant through the app.
    - Users can select how many people are coming and select a time slot.
    - Users should not be able to book if the restaurant does not have capacity remaining.
    - Users can edit reservations.
    - Users can cancel reservations.

10. Pre-order and In-Person Dining
    - Users can order food in advance, but only if they have a reservation.
    - Restaurant staff receive information regarding user's orders.
    - Users must pay for food orders through the app.

2.7 Non-Functional Requirements

Requirement Type	Requirements
Operational	• The system should support major browsers and operating systems. • The search and filtering system should be scalable, with the ability to add new filter options as necessary without affecting performance.

	• The system's AI should be constantly learning, improving the challenges and recommendations over time. • Discounts awarded to users are based on a points system. • The application's reservation system should be easily integrated with any existing reservation systems the restaurants may have.
Performance	• The system should process user requests and respond within two seconds. • The system must be scalable to handle increasing user traffic (e.g., cloud-based infrastructure). • The leaderboard and users' current number of total points must be reflected in real-time, with no noticeable delay. • Recommendations and search results of restaurants should be up-to-date and based on real-time location or updated settings.
Security	• Users must grant location permission before using the "Check In" feature. • The location detection must be accurate within 50 meters of the restaurant's actual location. • Passwords must be securely hashed. • User authentication must use HTTPS to encrypt data in transit. • The system must comply with PCI DSS (Payment Card Industry Data Security Standard) • User data, such as location, challenge progress, and earned points, must be securely protected from unauthorized access or tampering. • The validity of completed challenges or challenge verification must be protected to prevent cheating and ensure fair play among users.
Integrity	• User consent must be obtained for data collection. • Only authenticated users can leave reviews or comments • Only authenticated users can write a post in the blog section. • Unauthenticated users have read only permission. • User reviews must be monitored for spam and fraud. • Reviews or posts that are reported as spam or inappropriate will be hidden from profiles until review by a moderator. • Registered users will be banned from posting content on the platform if they receive five strikes (for posting inappropriate content) • The system should ensure accurate tracking of points, challenge progress, and leaderboard updates, with no data loss or inconsistencies.

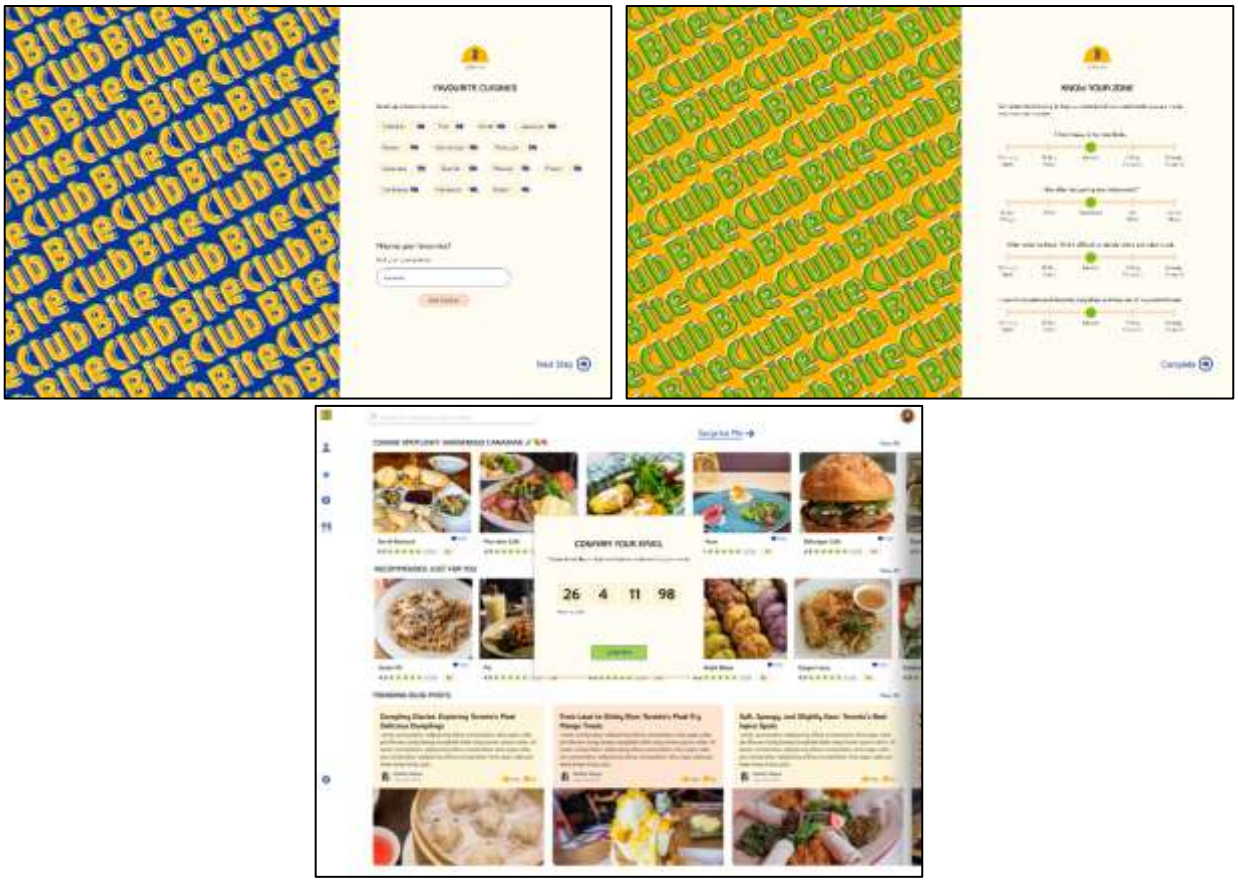
	• The system should generate challenges that adapt to users' unique preferences and limitations, while also ensuring challenges increase in difficulty as the user progresses.

2.8 UI/UXD Interface Mock-ups

To view the introduction video for the application mock-up, visit this link.

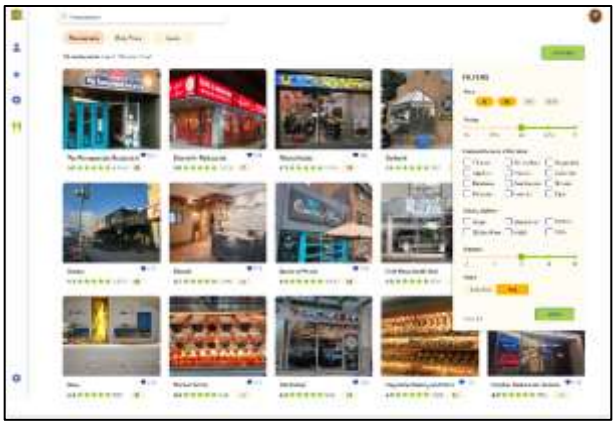
To view the full range of mock-ups, visit our Figma project.

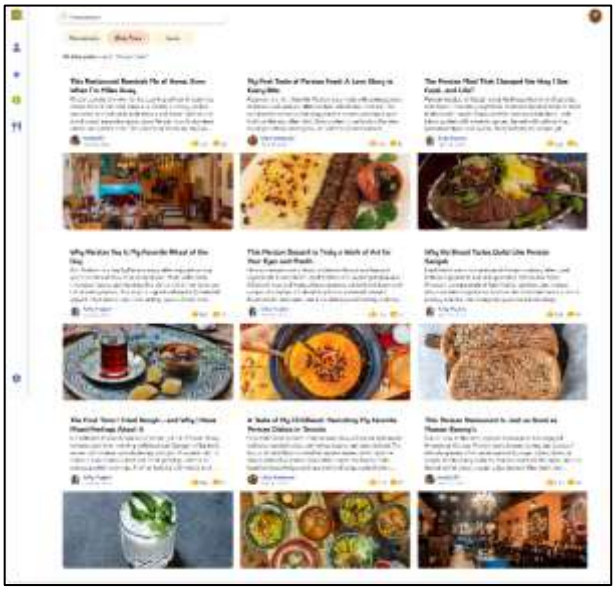
1. Scenario – Onboarding and AI-Personalized Recommendations
   Our target user is "Sarah," a 21 year-old woman living in Toronto. She wants to expand her palate and explore the various cuisines in the city, but doesn't know where to start. She registers for our desktop application "*BiteClub*" to reach her goals. She completes the questionnaire during onboarding so she can receive personalized restaurant recommendations. After onboarding, she views her recommendations.



2. Scenario – Restaurant and Blog Post Search

   Sarah wants to see more restaurants serving her favourite cuisine: Persian. She searches for *Persian*, and sees multiple results of related restaurants and blog posts. She filters the restaurant results based on her price and distance preferences.



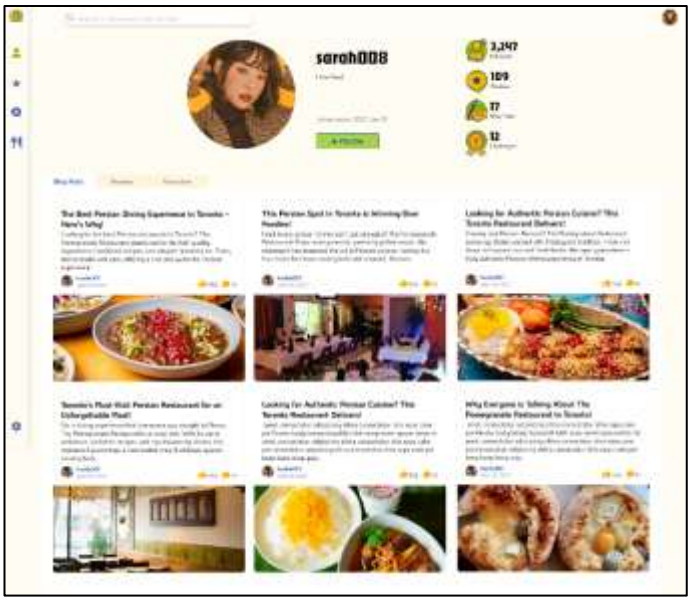


3. Scenario – Restaurant Profiles

   Sarah selects one of the restaurant results, *The Pomegranate*. She wants to determine if it's a good choice, so she looks at other users' reviews, blog posts where the restaurant is mentioned, and photos of the restaurant. She “likes” the most informative reviews. From the reviews, it seems the restaurant is worth trying, so she views its location and businesses hours, and saves the restaurant to her profile.

4. Scenario – User Profiles

   After using *BiteClub* for a few weeks, Sarah discovered many new restaurants, and even built a small following through the many reviews and blog posts she's written. She loves sharing her experiences with the community and uses her profile to manage her content.



5. Scenario – AI Generated Challenges and Geolocation-Verified Check-In

   Despite the many restaurants she discovered through recommendations and through her own searches, Sarah wants to push herself even further to explore foods outside her comfort zone. So, she participates in food challenges. Not only do the challenges encourage her to try new foods, but to her delight, she gets rewarded with points that can be used for discounts on future orders and restaurant visits.

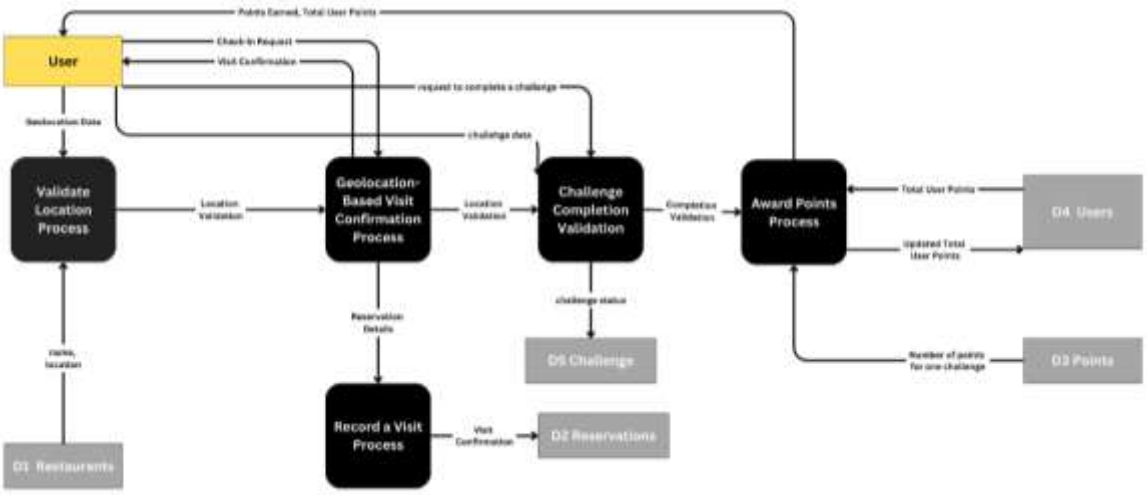
3. Process and Data Modeling

3.1 UML/DFD Modeling and Data Modeling

Data Flow Diagrams

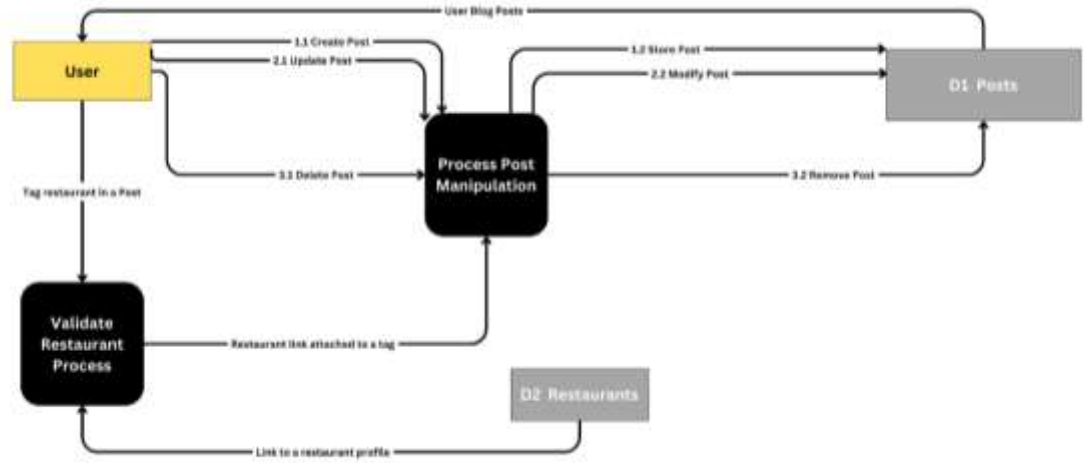
Geolocation-Based Visit Confirmation

View on Canva.



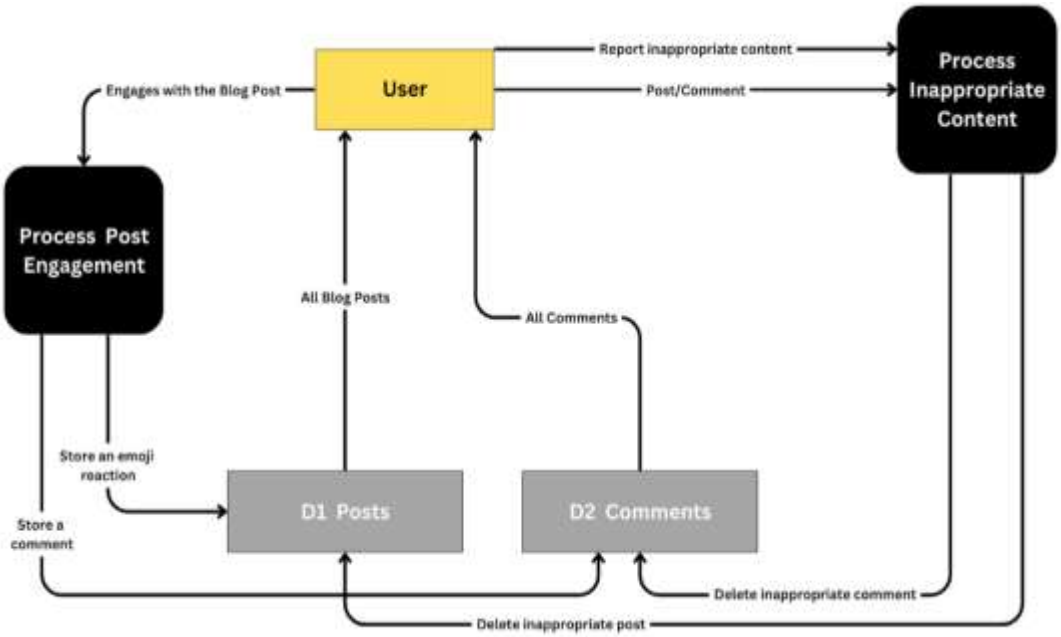
Blog Post Management

View on Canva.



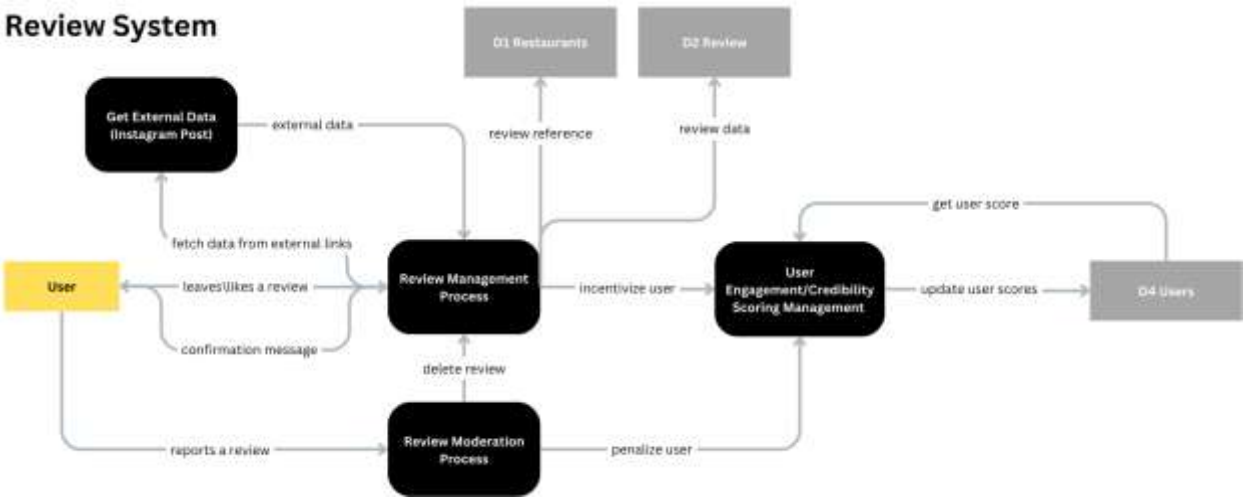
Blog Post Interactions

View on Canva.



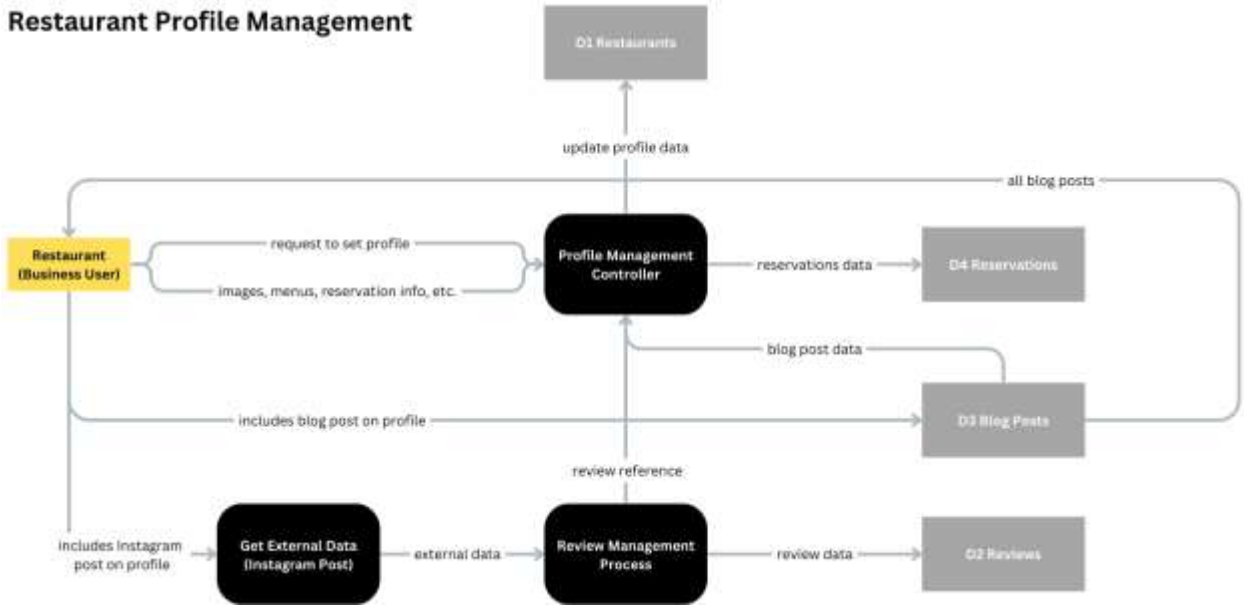
Review System

View on Canva.



Restaurant Profile Management

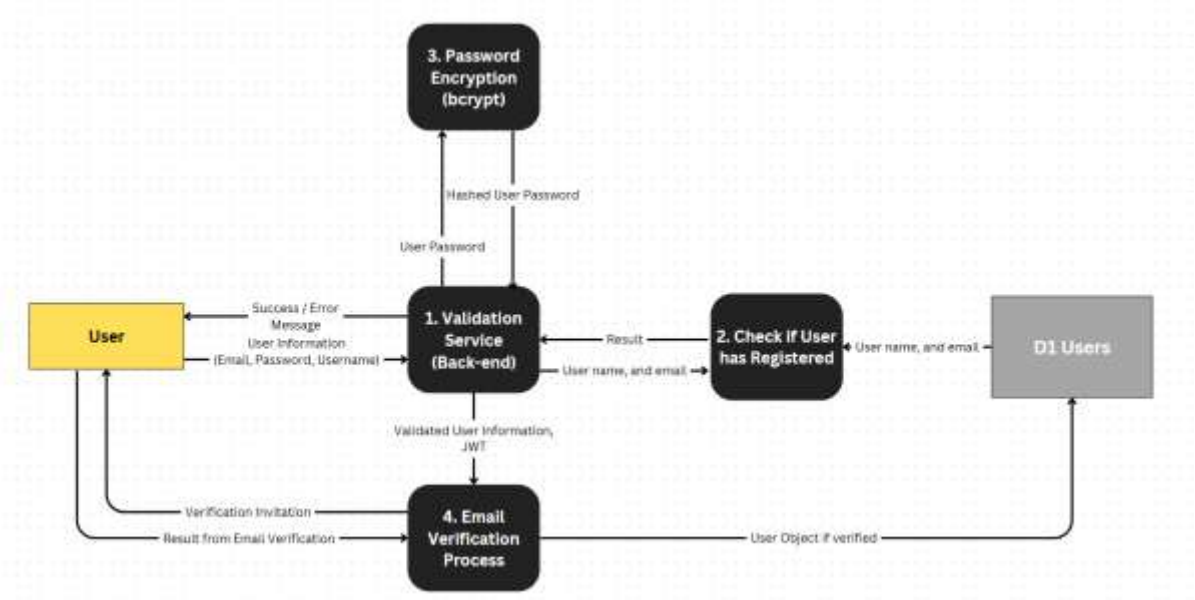
View on Canva.



Registration

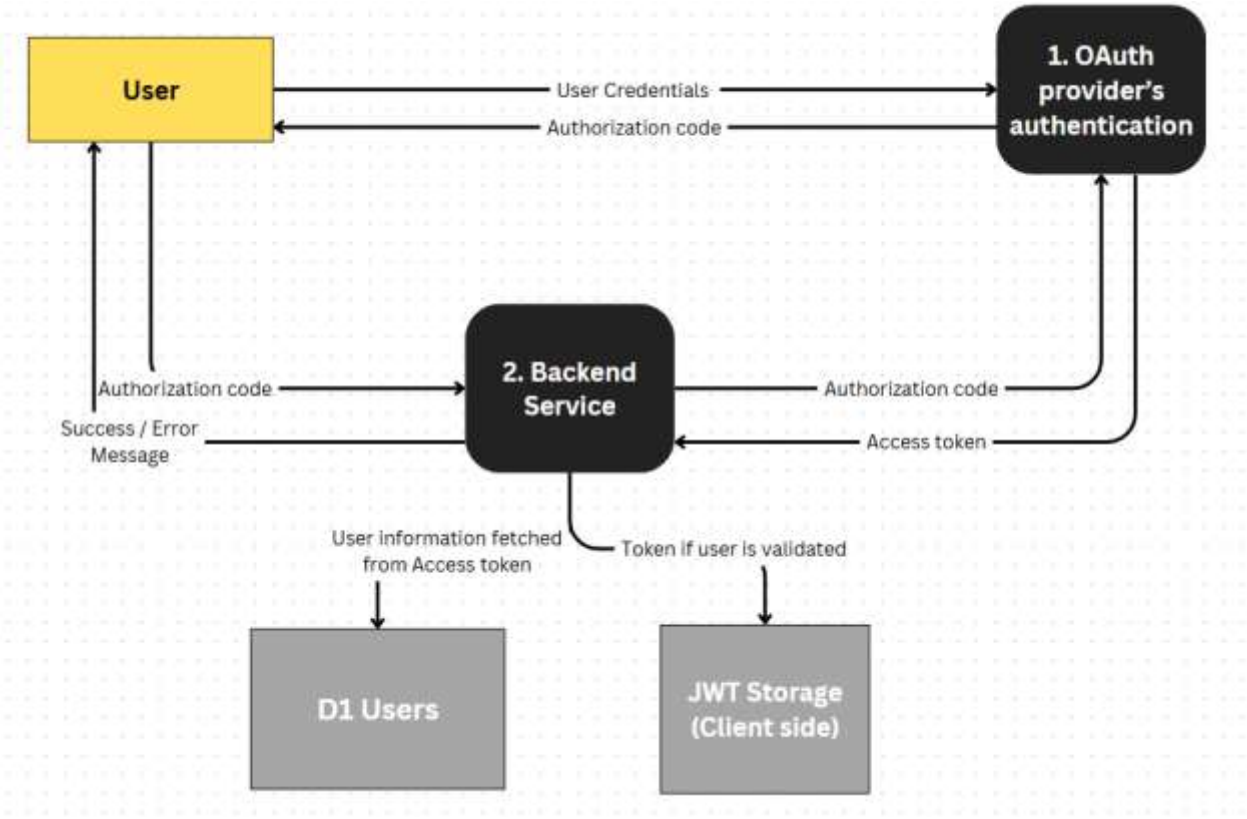
Method 1: User Register via Email

[View on Canva.]



Method 2: User Register via SSO

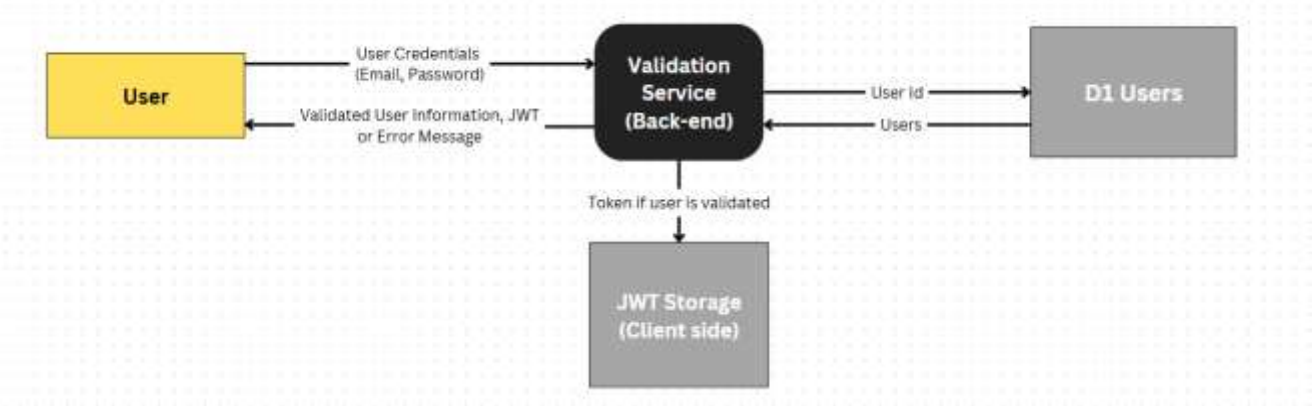
[View on Canva.]



Login

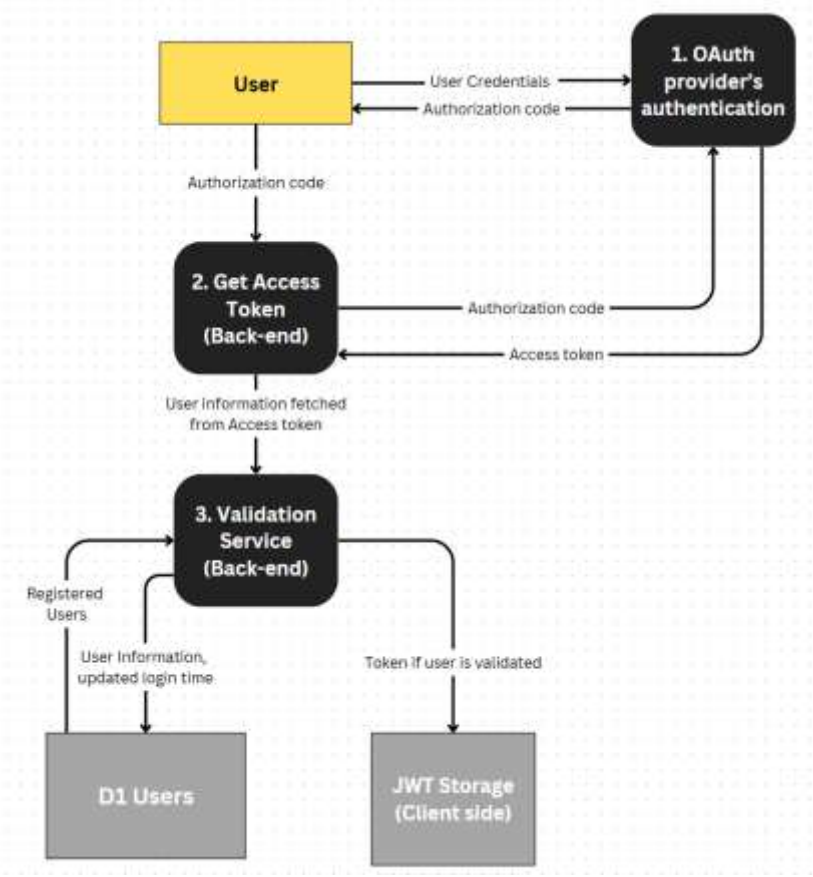
Method 1: User login via Email

View on Canva.



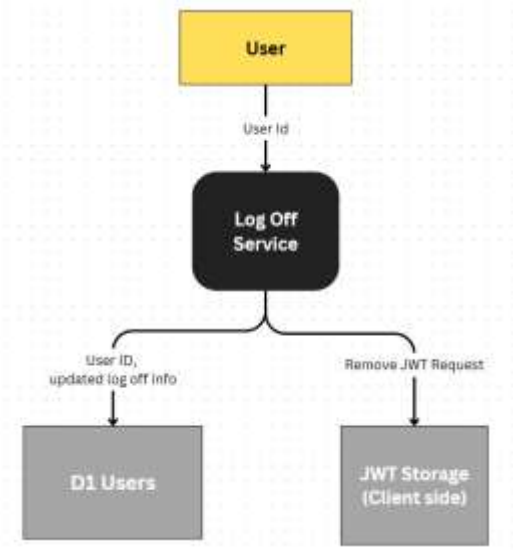
Method 2: User login via SSO

View on Canva.



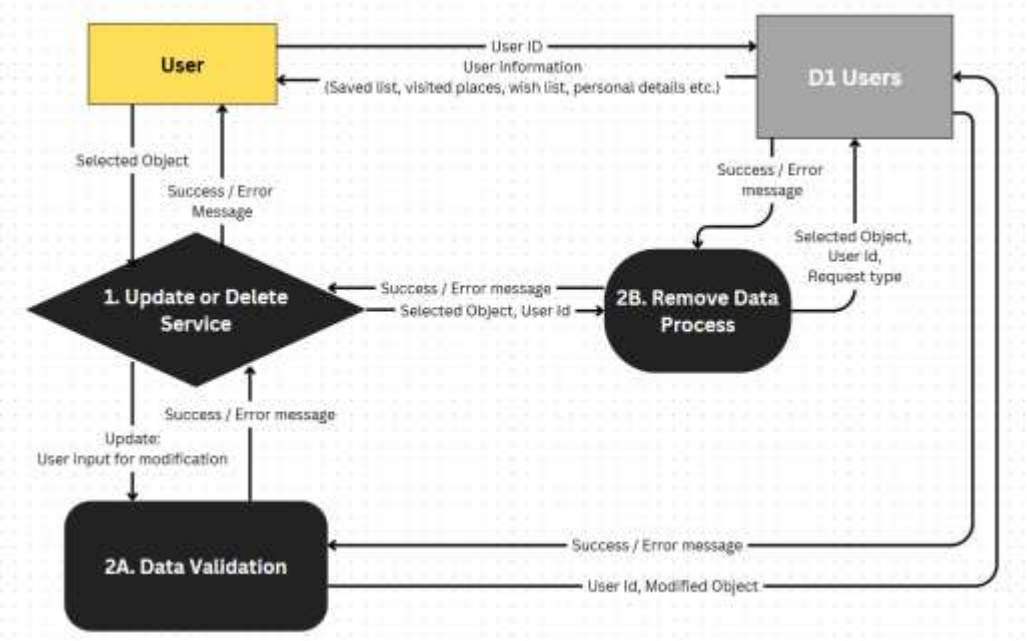
Log off

View on Canva.



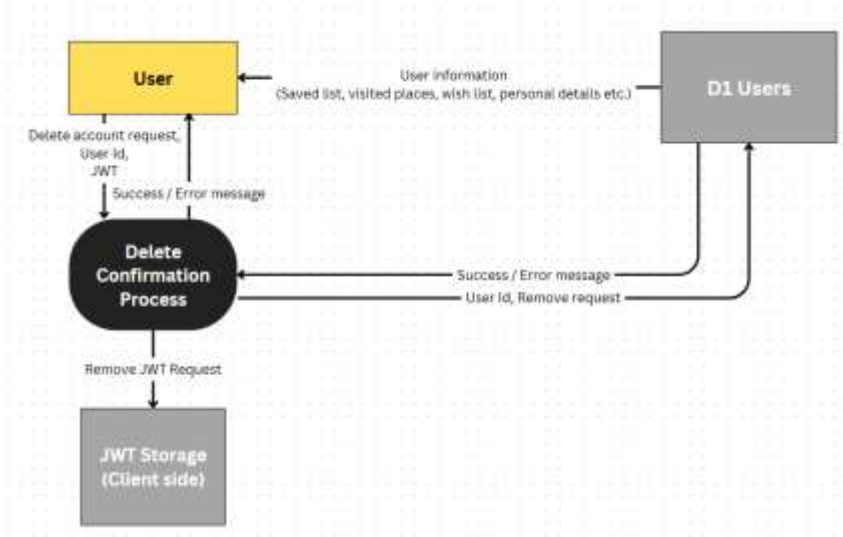
Modify Account Information

View on Canva.



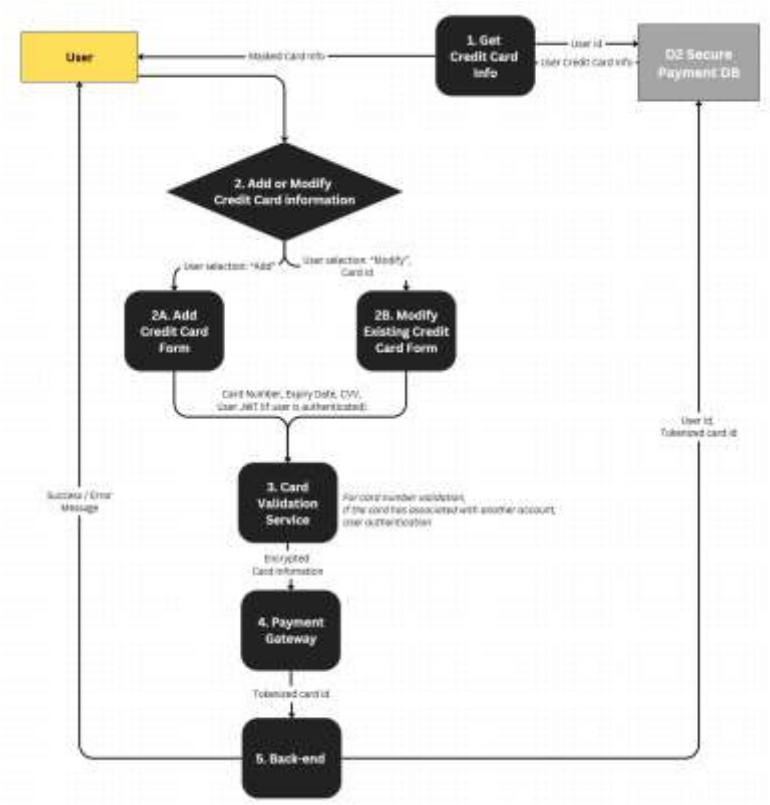
Delete Account

View on Canva.



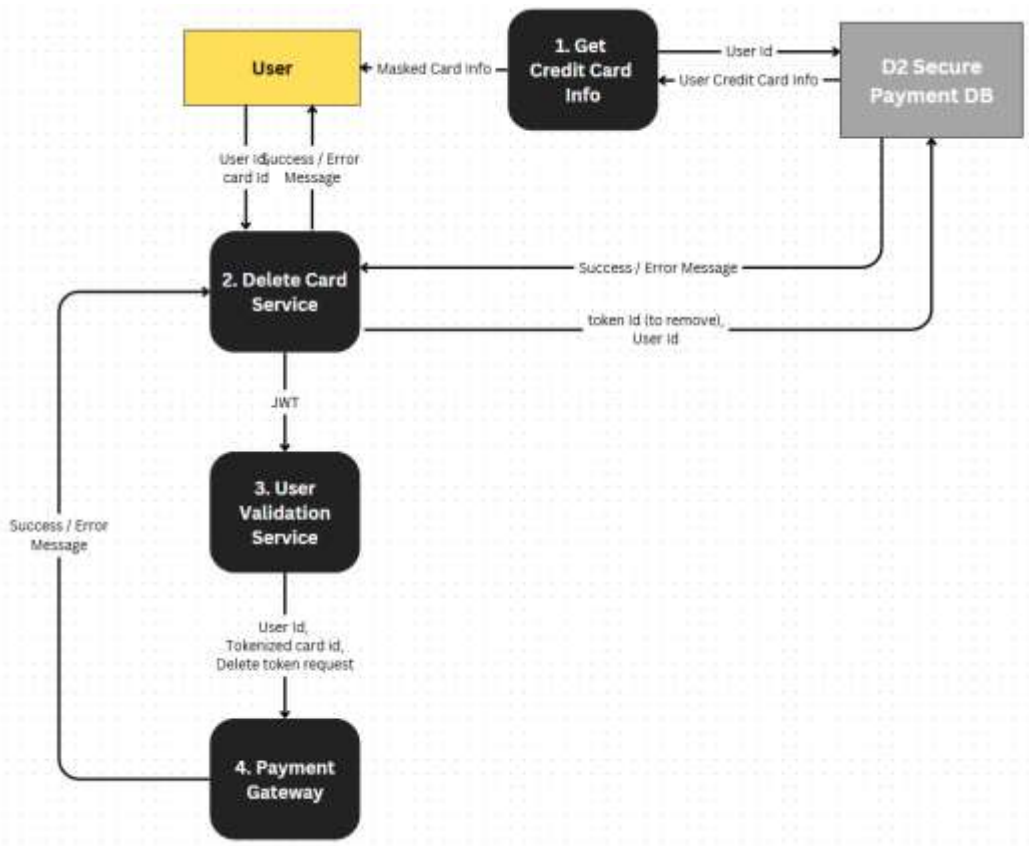
Add or Modify Credit Card

View on Canva.



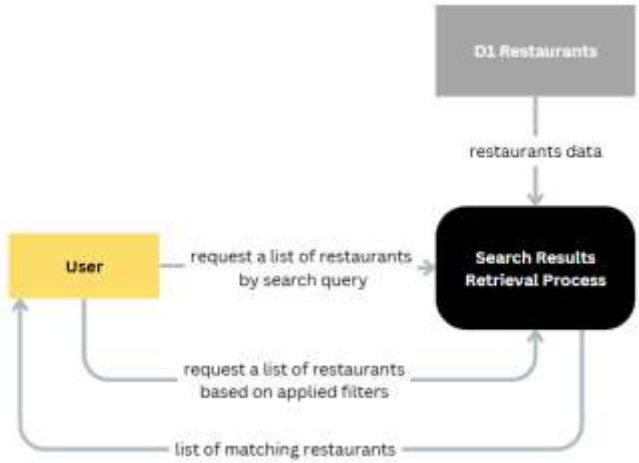
Delete Credit Card

View on Canva.



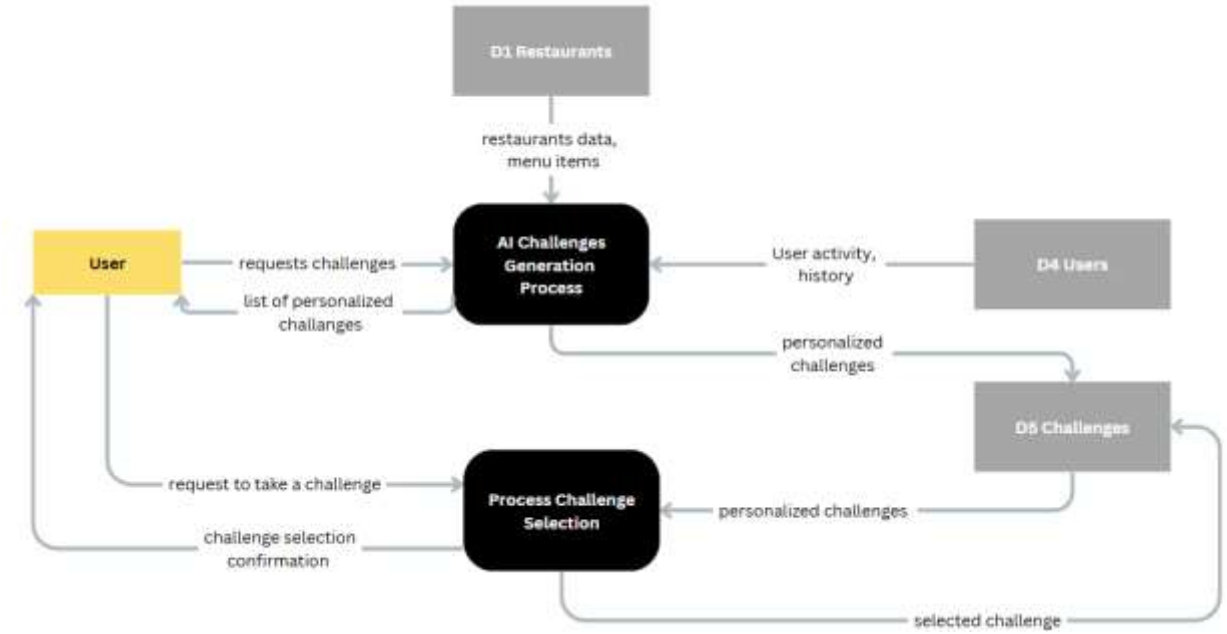
Search and Filter Restaurants

View on Canva.



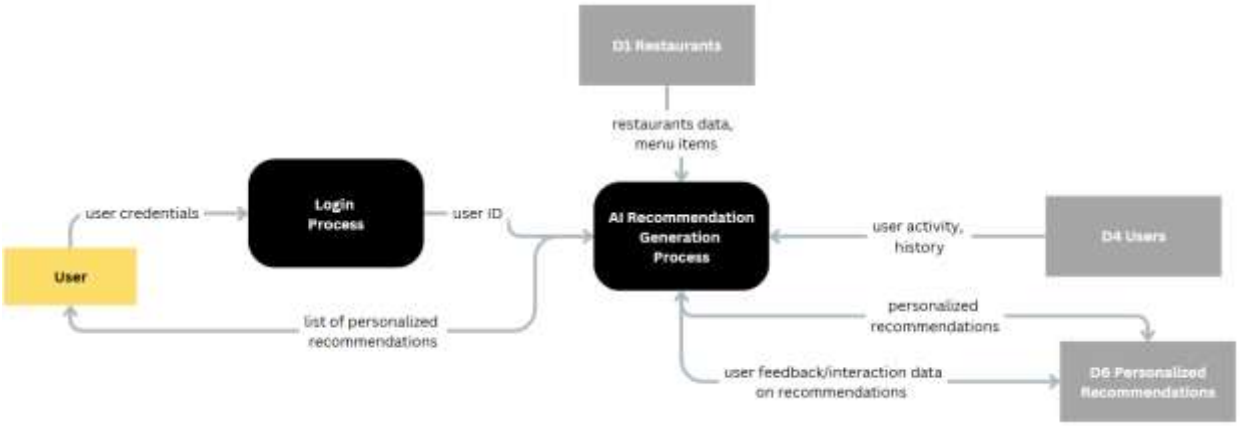
Select AI-Generated Challenge

View on Canva.



AI-Personalized Recommendations

View on Canva.

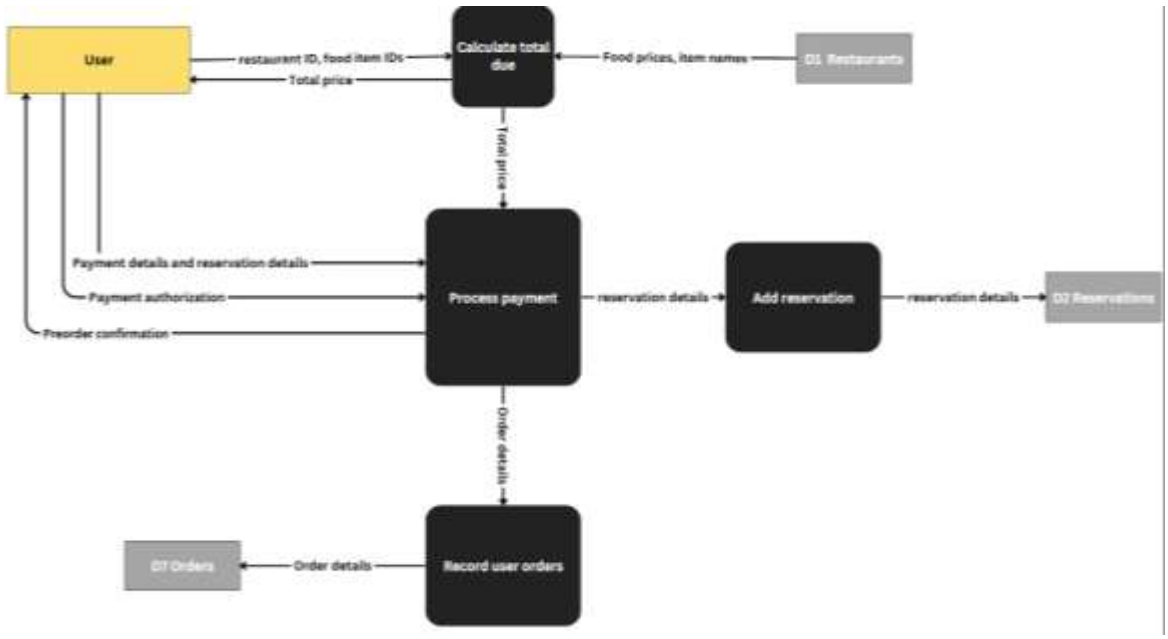


Add a Reservation

View on Canva.

Pre-Ordering

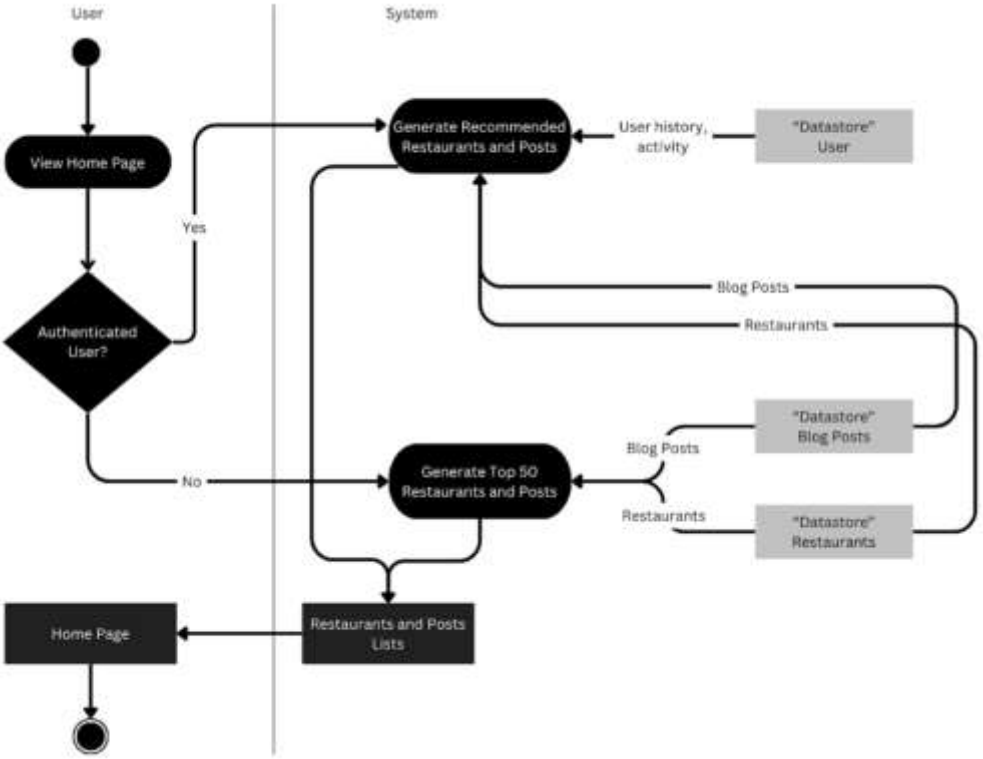
View on Canva.



Activity Diagrams

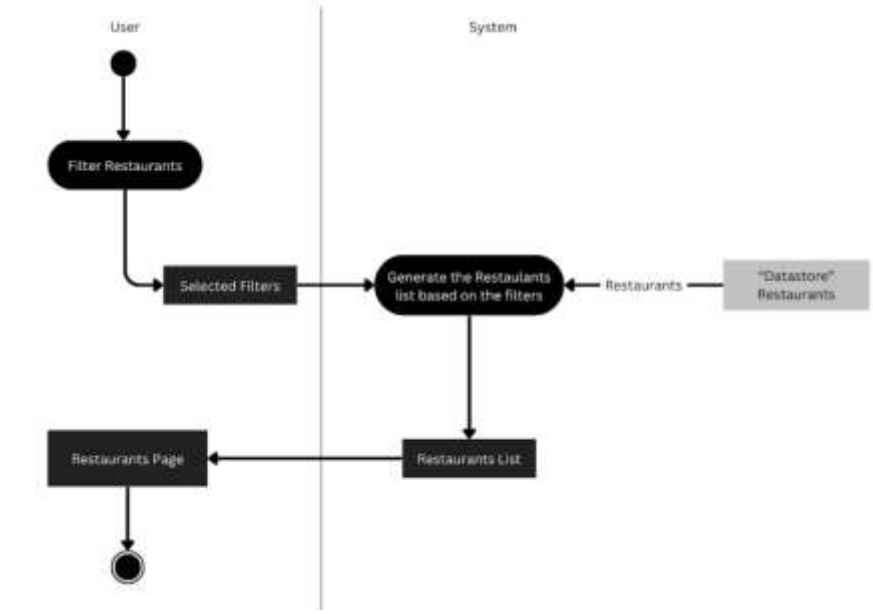
Home Page: AI Personalized Recommendations

View on Canva.



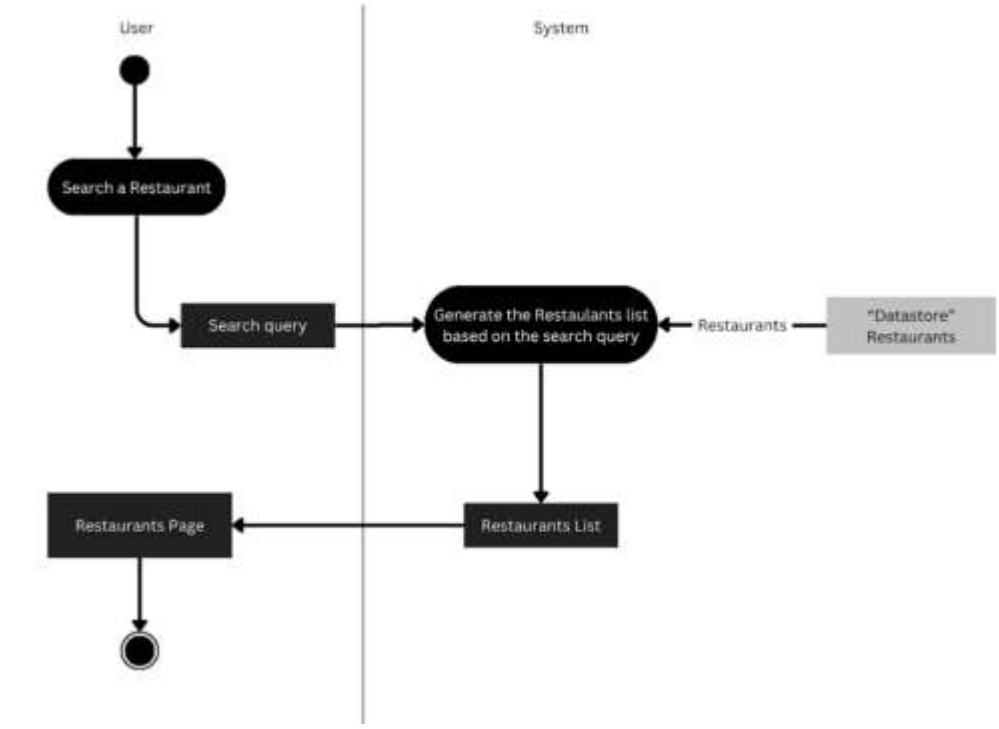
Restaurants Page: Filter Restaurants

View on Canva.



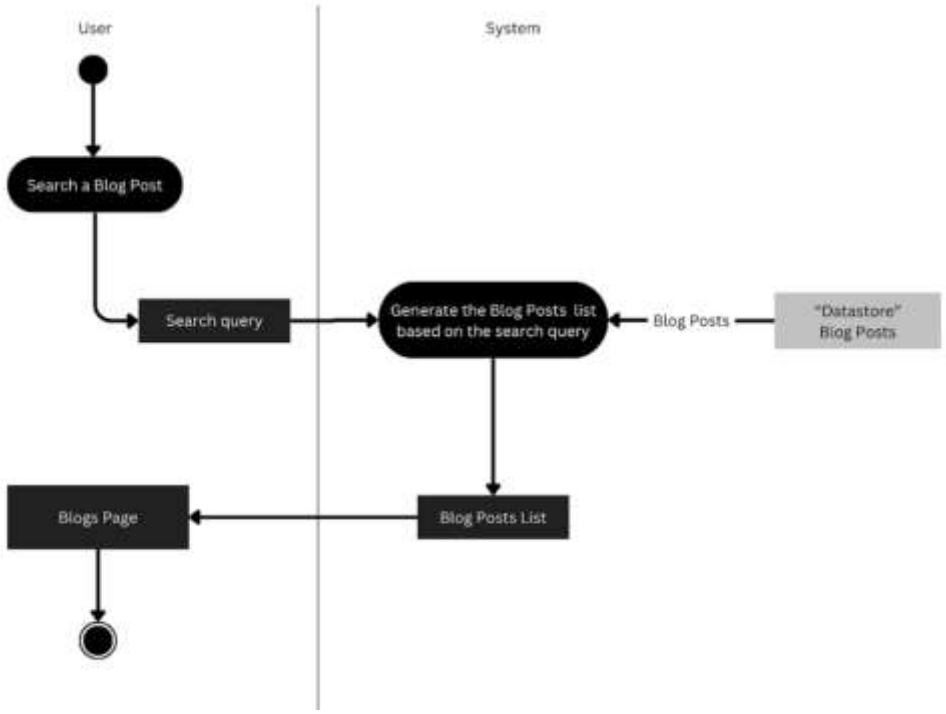
Restaurants Page: Search for a Restaurant

View on Canva.



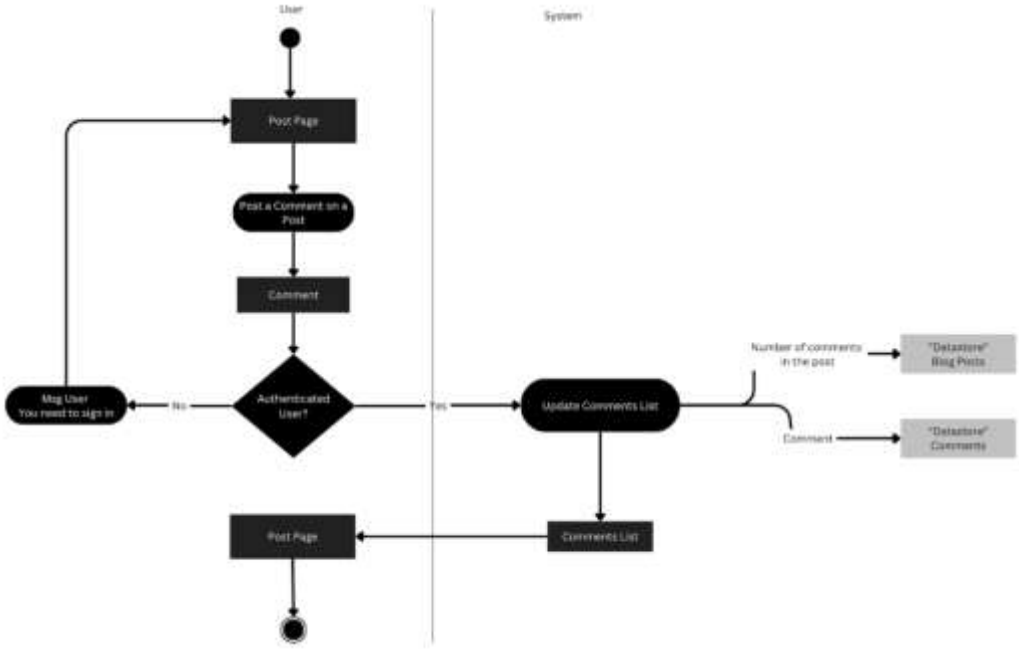
Blogs Page: Search for a Blog Post

View on Canva.



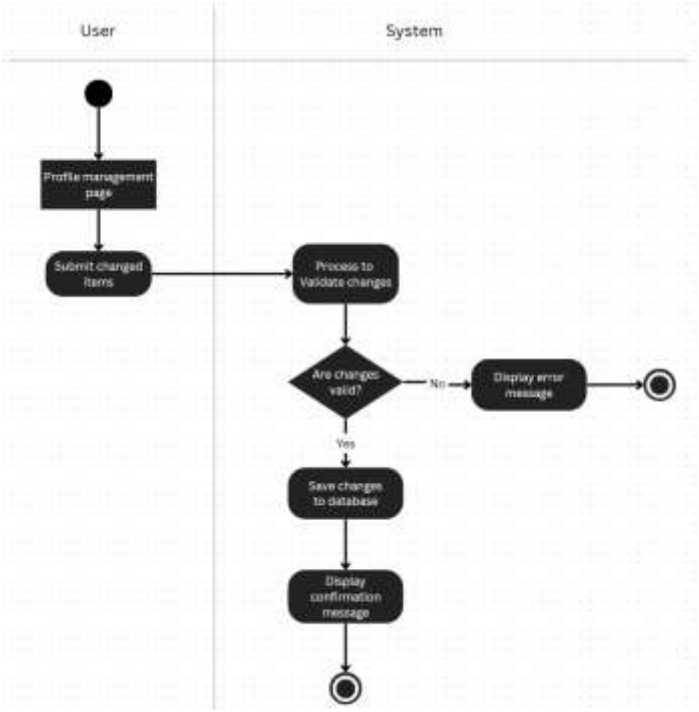
Expanded Blog Post Page: Interaction

View on Canva.



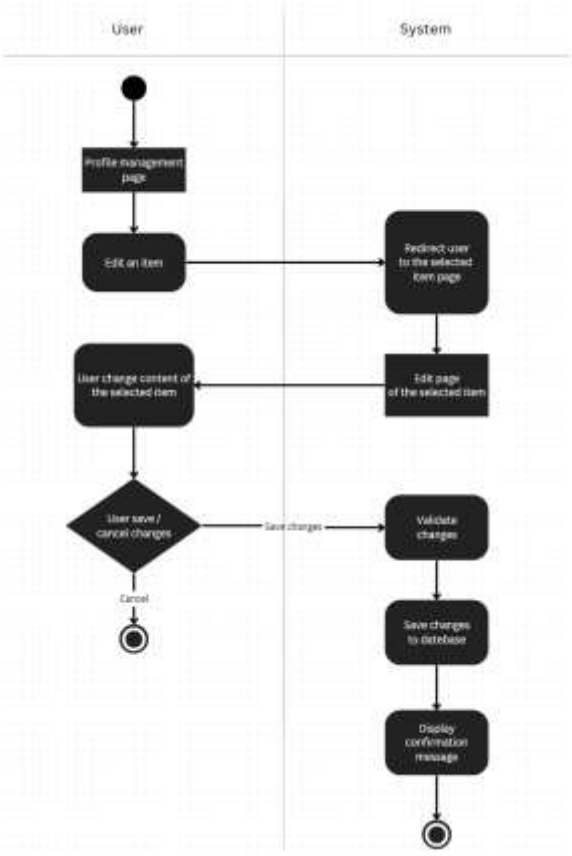
User Profile: Managing Profile Picture, Username, Password, Bio and Display Preferences

View on Canva.



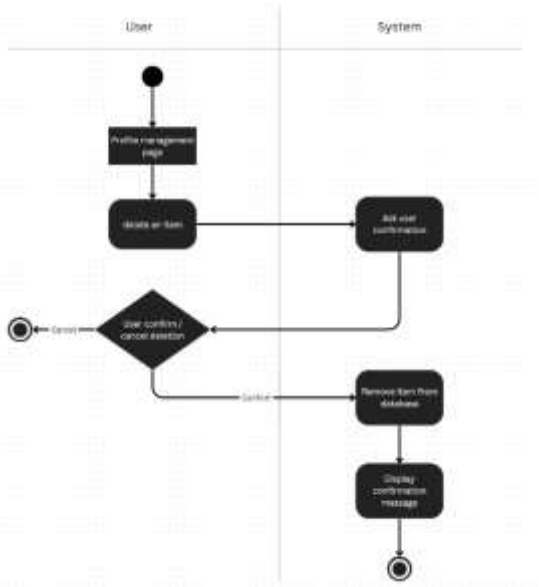
User Profile: Modifying User Blog Posts, Reviews, and Saved Items

View on Canva.



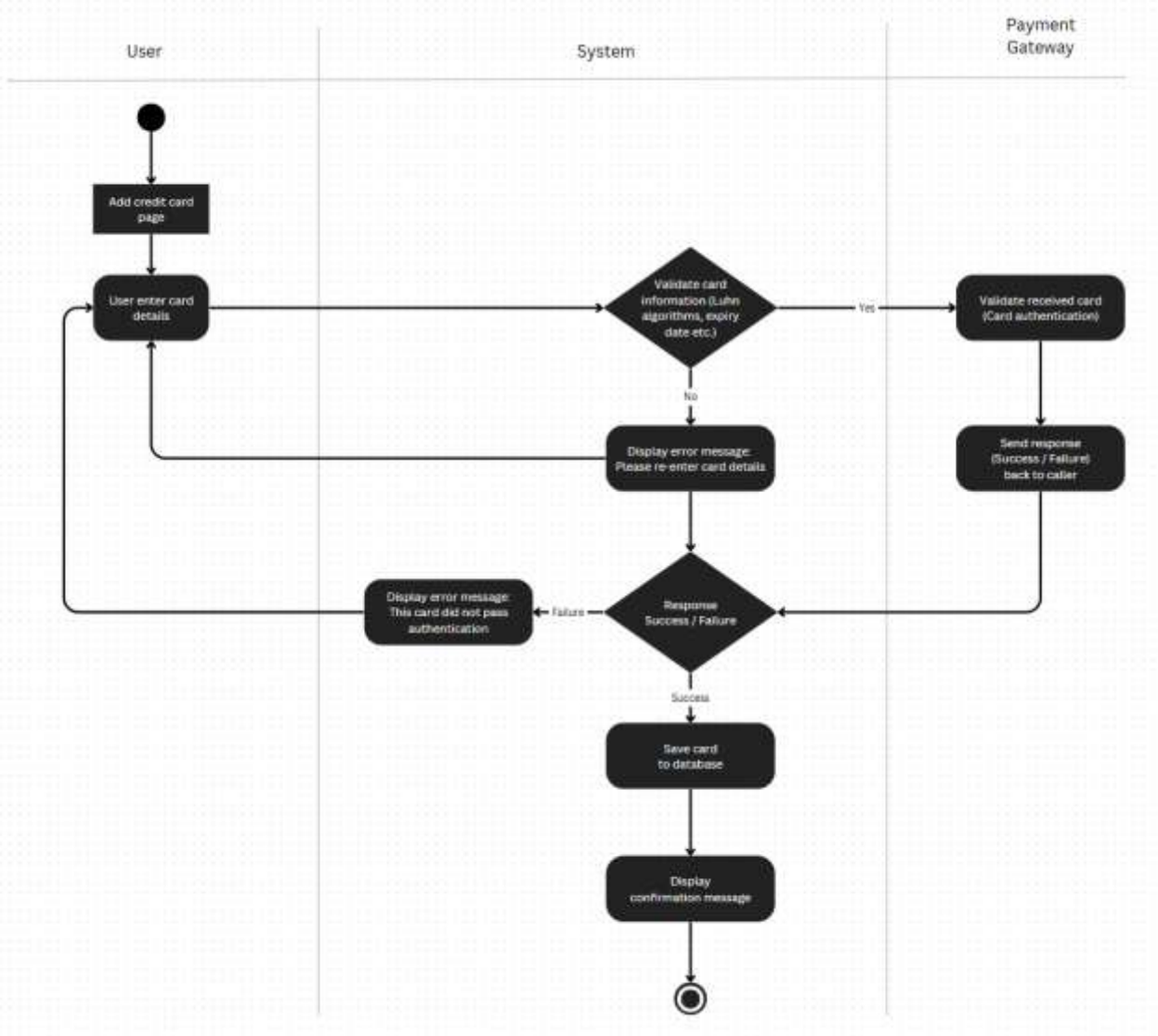
User Profile: Deleting User Blog Posts, Reviews, and Saved Items

View on Canva.



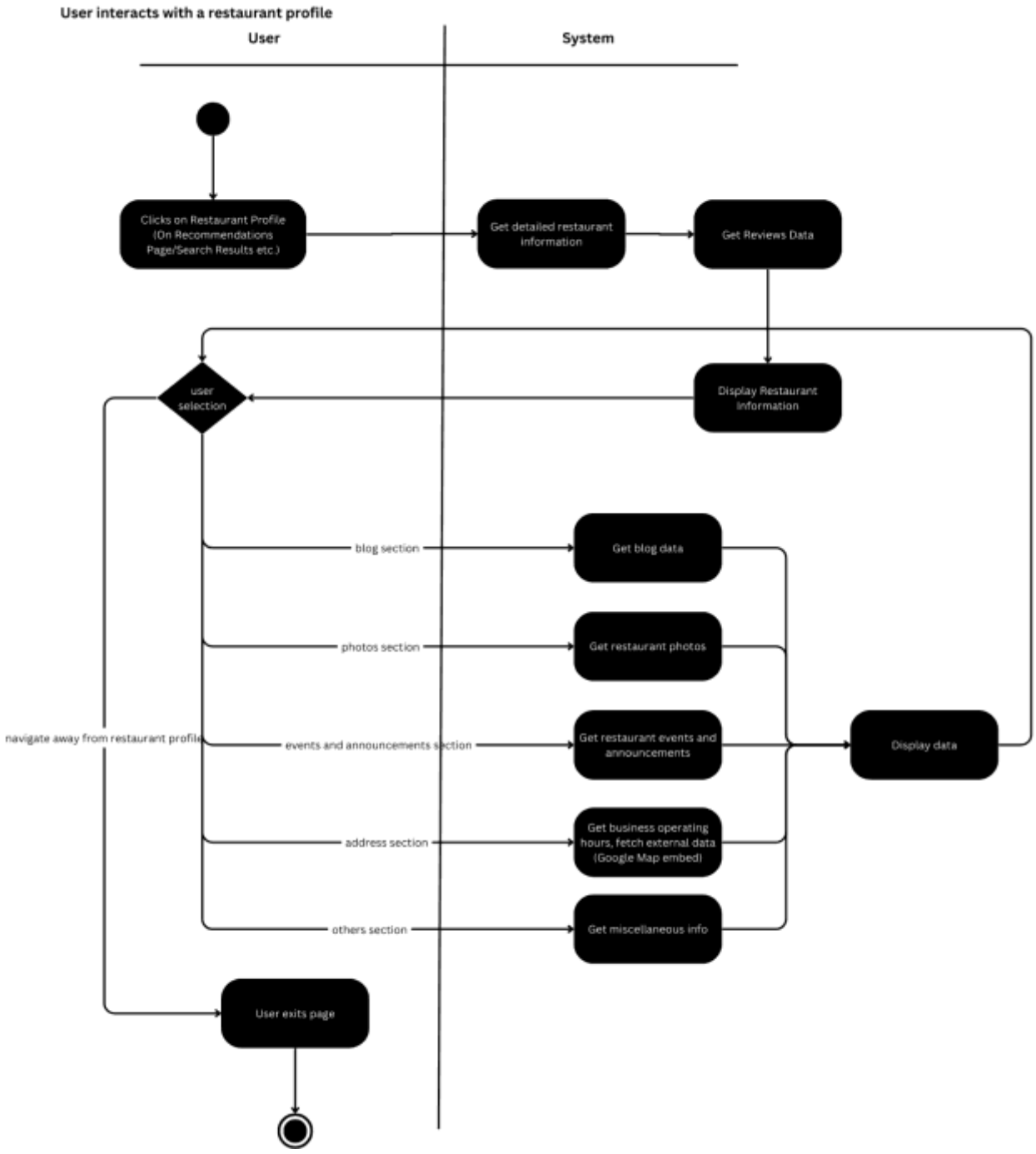
Credit Card Page: Add Credit Card

View on Canva.



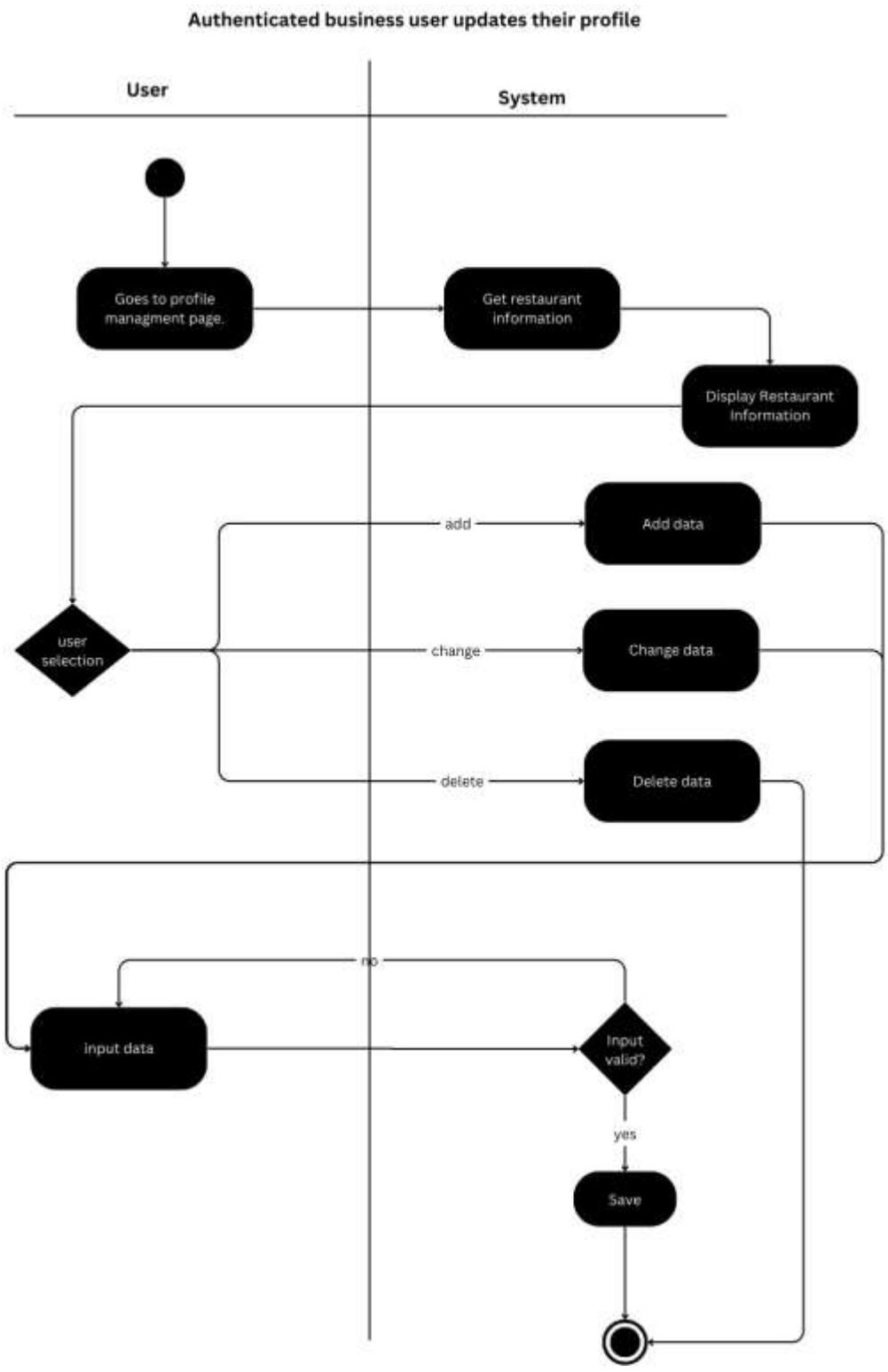
Restaurant Profile: User Interaction

View on Canva



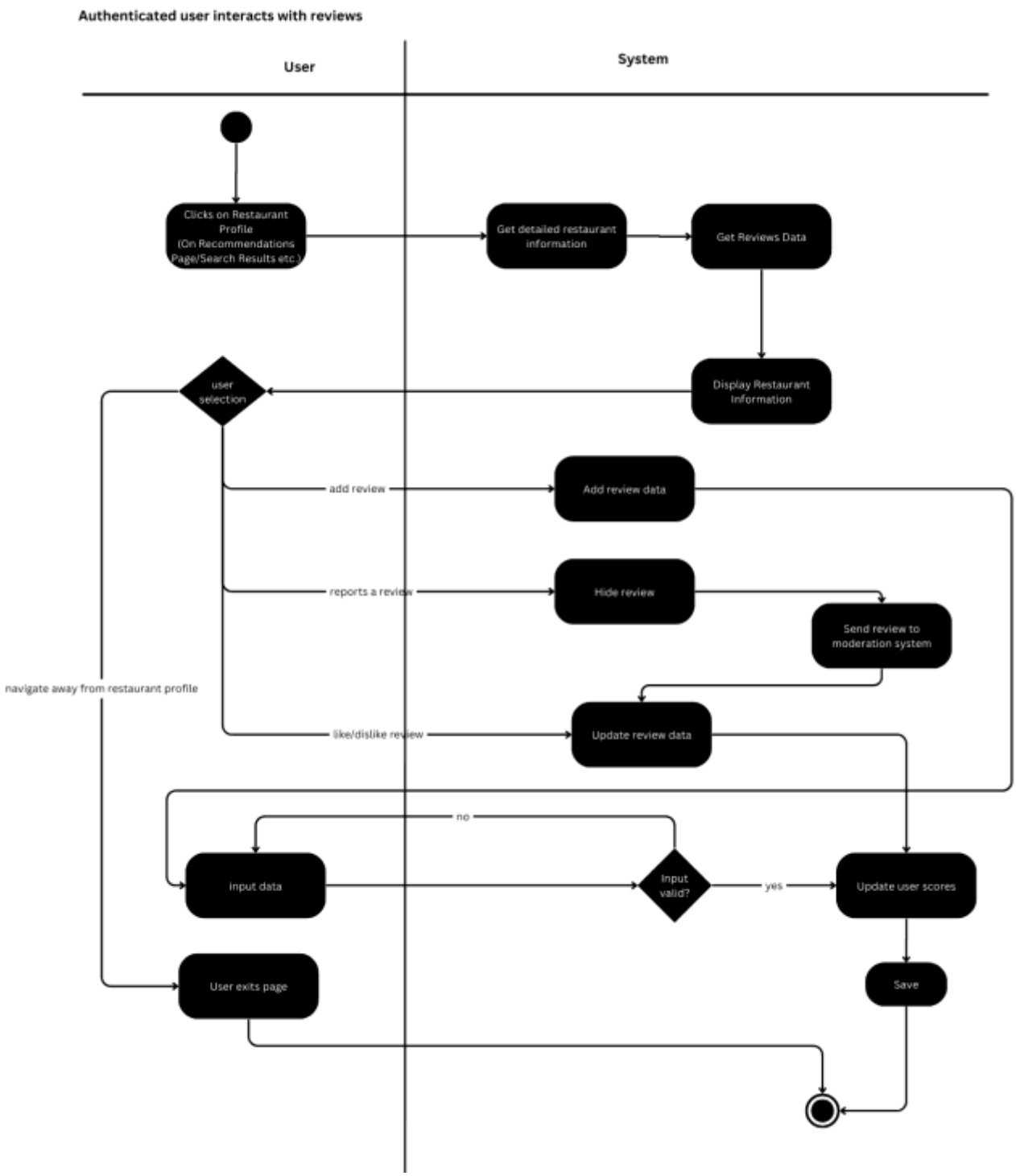
Restaurant Profile: Business User Updates their Profile

View on Canva



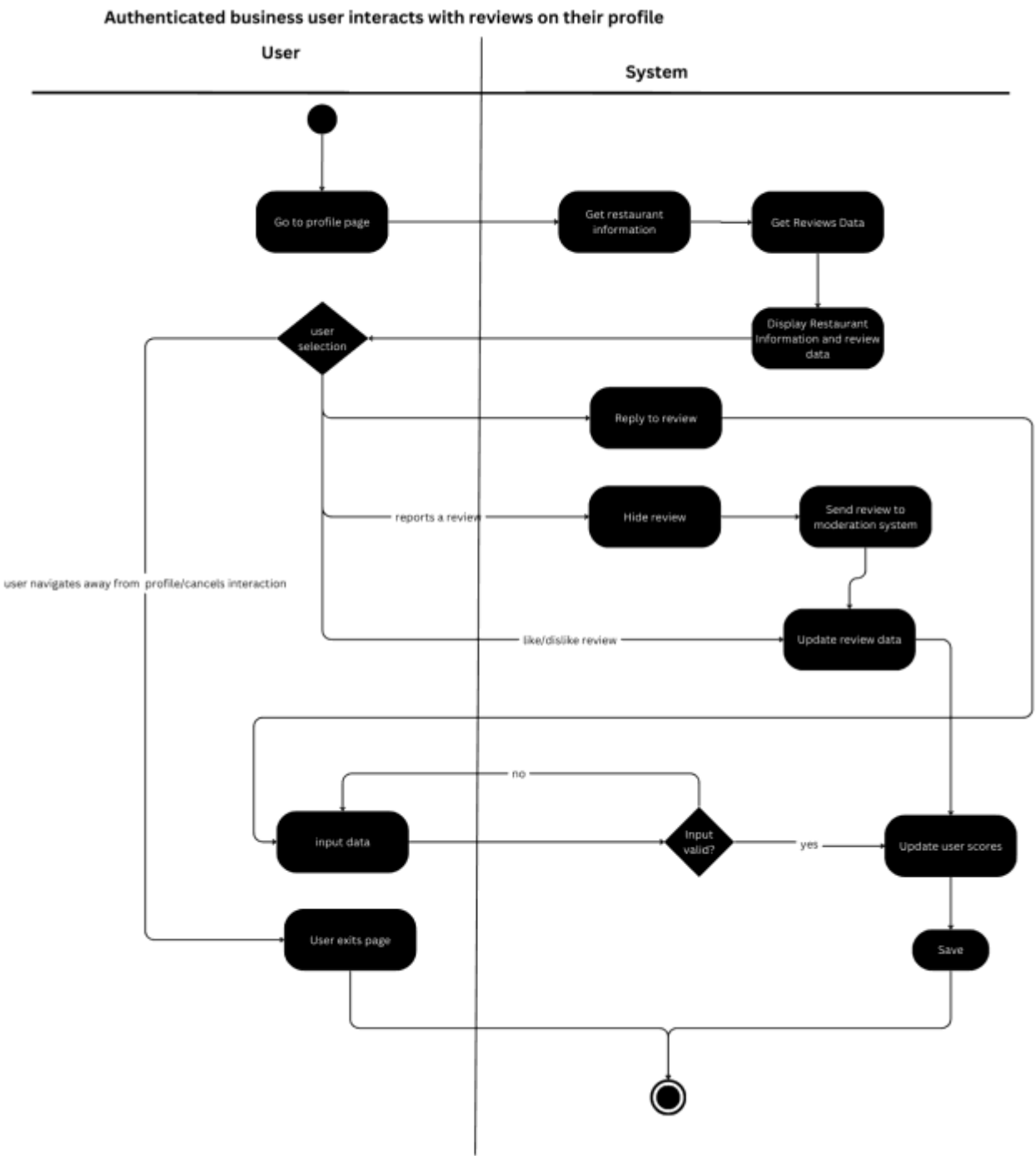
Restaurant Profile: User Interacts with Reviews

View on Canva



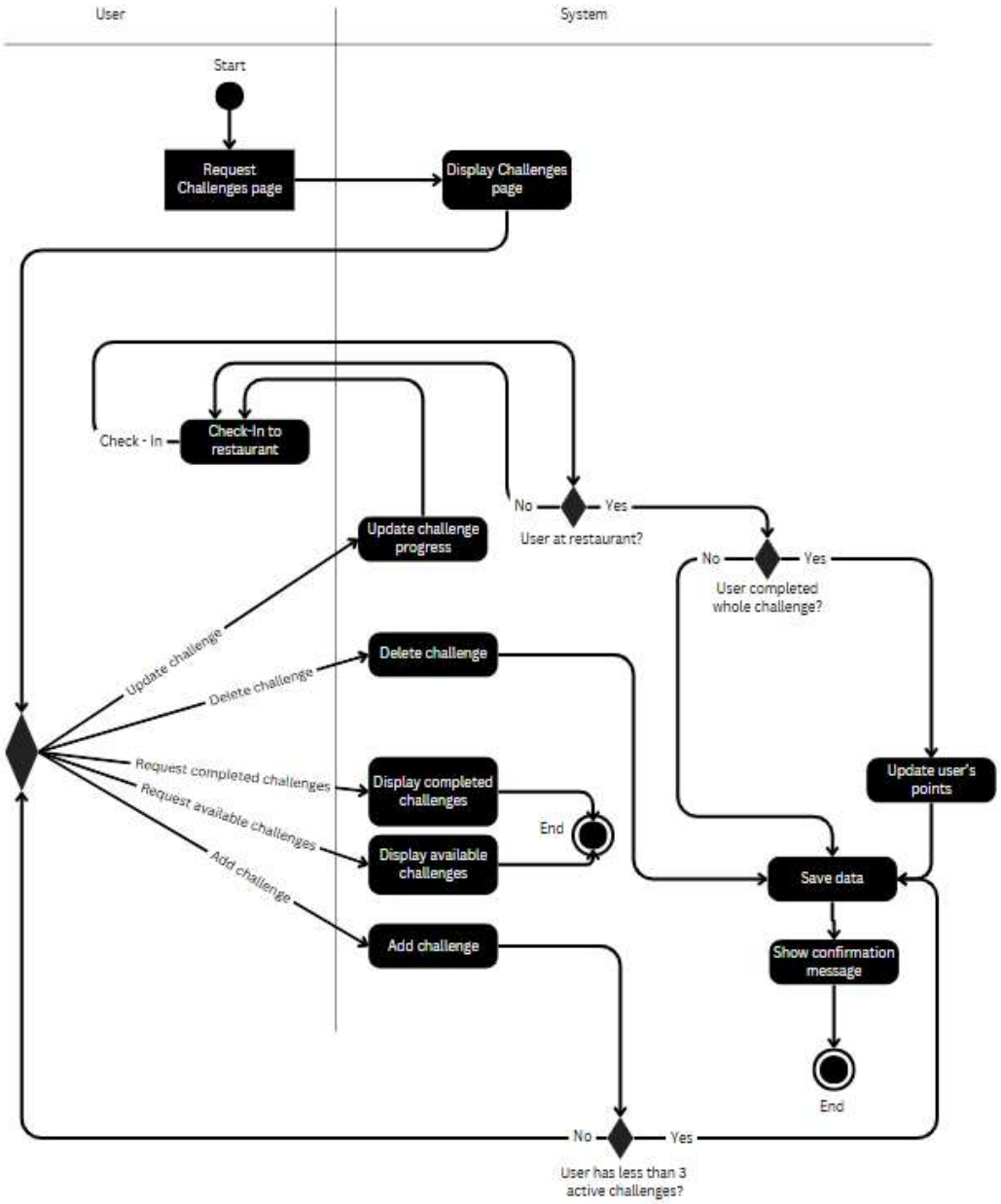
Restaurant Profile: Business User Interacts with Reviews on their Profile

View on Canva

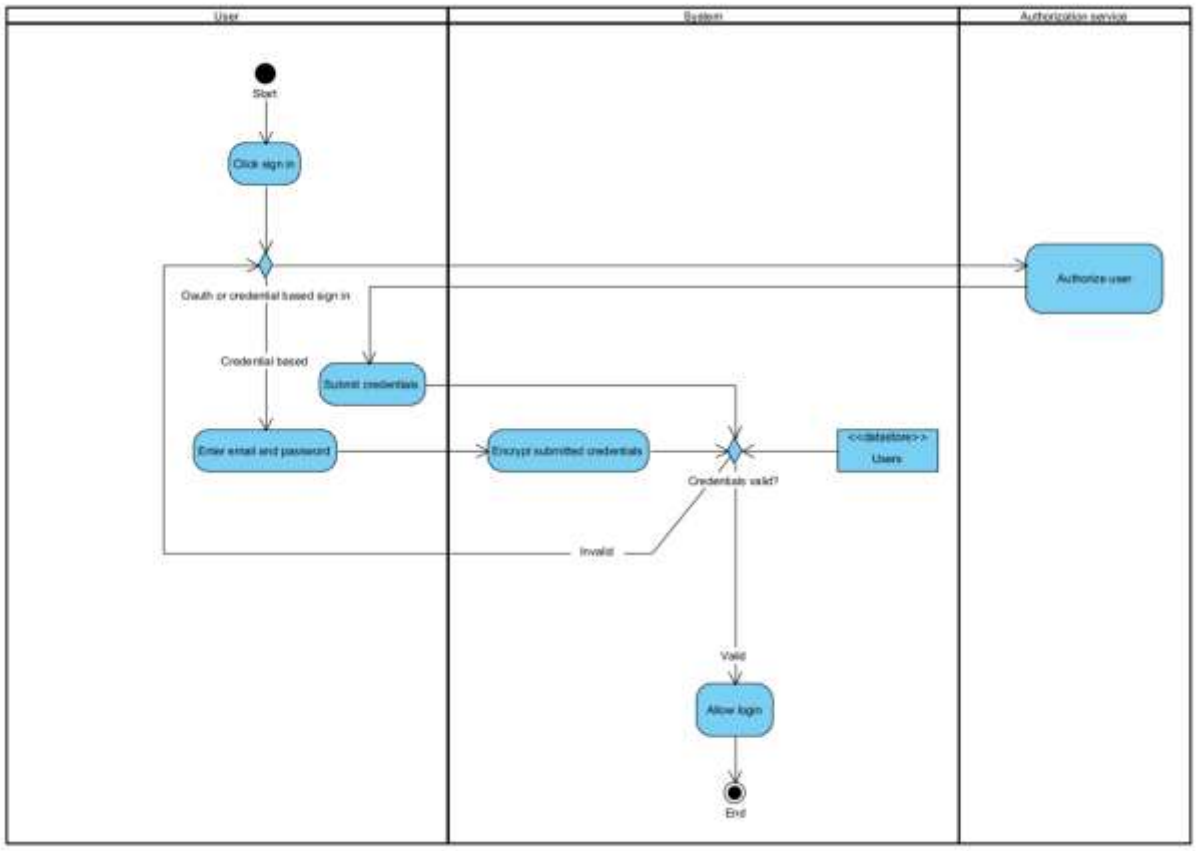


Challenges Page: User Interacts with Challenges

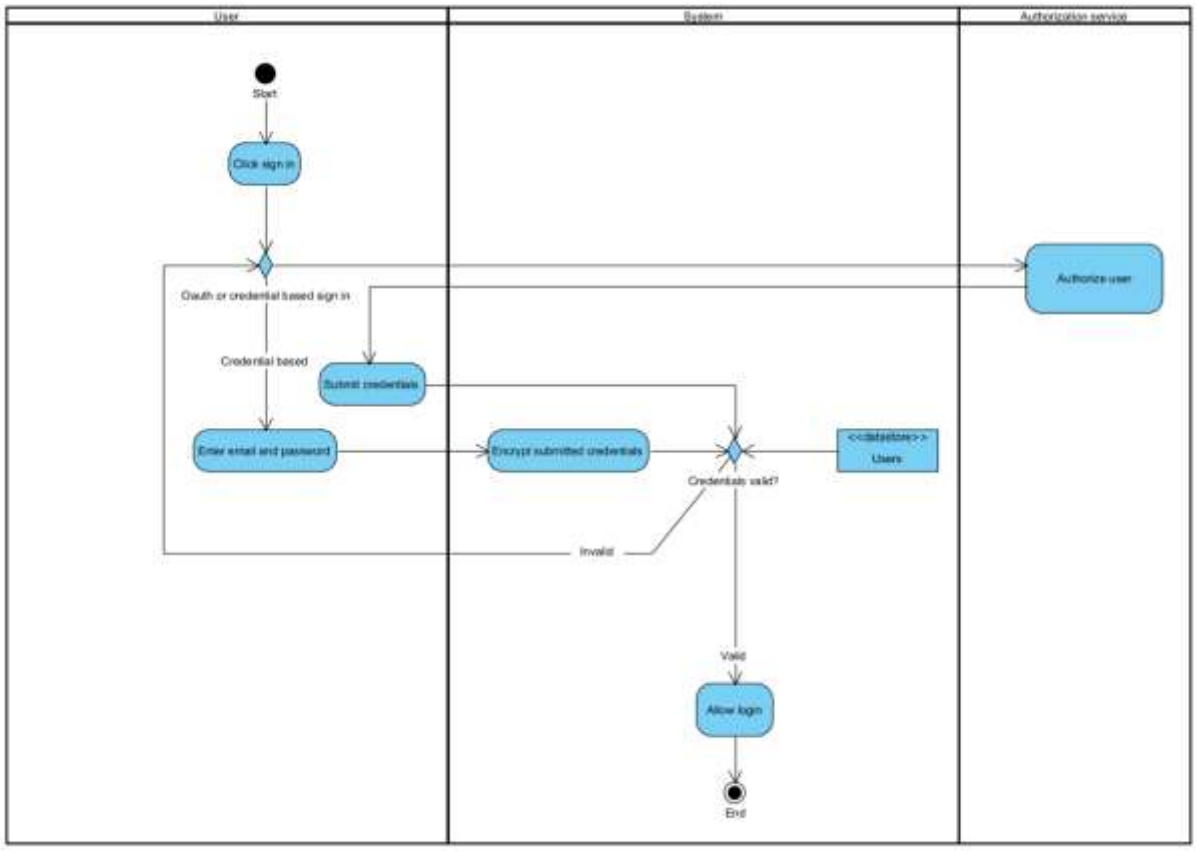
View on Canva


User
System
Start
Request Challenges page
Display Challenges page
Check - In
Check-In to restaurant
No
Yes
User at restaurant?
Update challenge progress
No
Yes
User completed whole challenge?
Update challenge
Delete challenge
Delete challenge
Request completed challenges
Request available challenges
Add challenge
Display completed challenges
Display available challenges
End
Update user's points
Save data
Show confirmation message
End
Add challenge
No
Yes
User has less than 3 active challenges?

Registration Page: Registration and Onboarding



Sign-In Page: User Sign-In



3.2 Use Case Specification

3.2.1 Business Rules

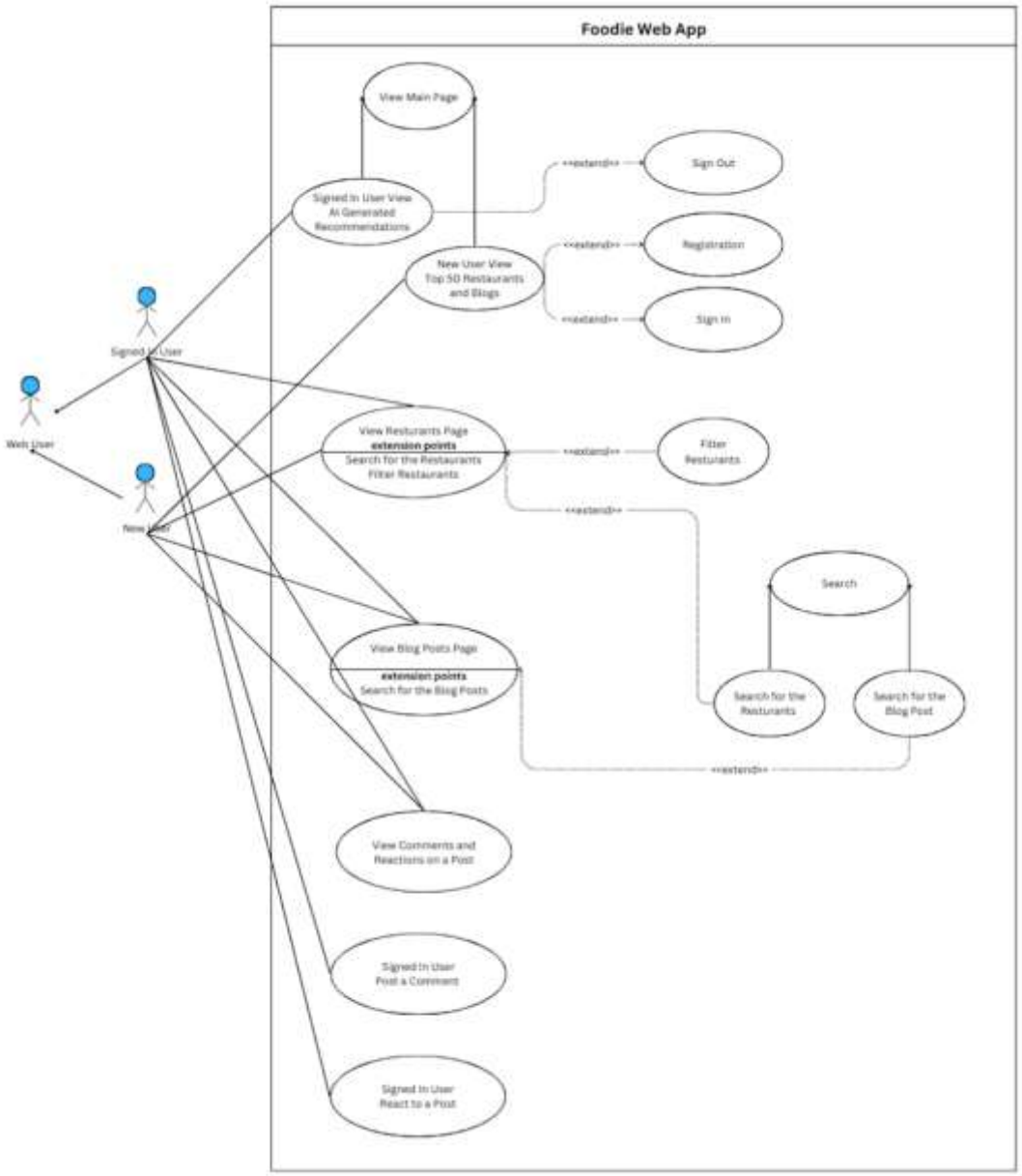
Business Rule Number	Business Rule Description	Related UC
BR01	Only authenticated (signed-in) users can add a comment or reaction to a blog post.	UC01
BR02	Authenticated users must be allowed to modify their personal information, including their username, password, and account settings. Additionally, they must have the ability to remove their blog posts and delete their account.	UC02
BR03	User must provide the email and password or use OAuth to sign up.	UC03, UC01
BR04	Users must answer a questionnaire before completing registration.	UC03

BR05	Users must provide the email and password or use OAuth to sign in.	UC03, UC01
BR06	Only authenticated users can participate in food challenges.	UC04
BR07	Users must enable location permissions to verify their geolocation when checking-in.	UC04
BR08	Only authenticated users can write, edit, or delete reviews.	UC05
BR09	Reviews must contain at least one of the following: text, an image, or an external review (i.e., an Instagram post link).	UC05
BR10	Reviews will be hidden if reported by other users and deleted when found inappropriate by a moderator.	UC05, UC06
BR11	Business users can only reply to reviews on their own restaurant profiles.	UC06
BR12	Business users can attach Instagram posts as external reviews to enhance their profile.	UC06
BR13	Users can like or report any review, once per review.	UC05, UC06
BR14	Users' engagement and credibility scores will decrease if their reviews are reported and found inappropriate by a moderator.	UC05
BR15	Reviews are prioritized based on the reviewer's platform engagement and credibility score.	UC05

3.2.2 System Use Case Diagrams

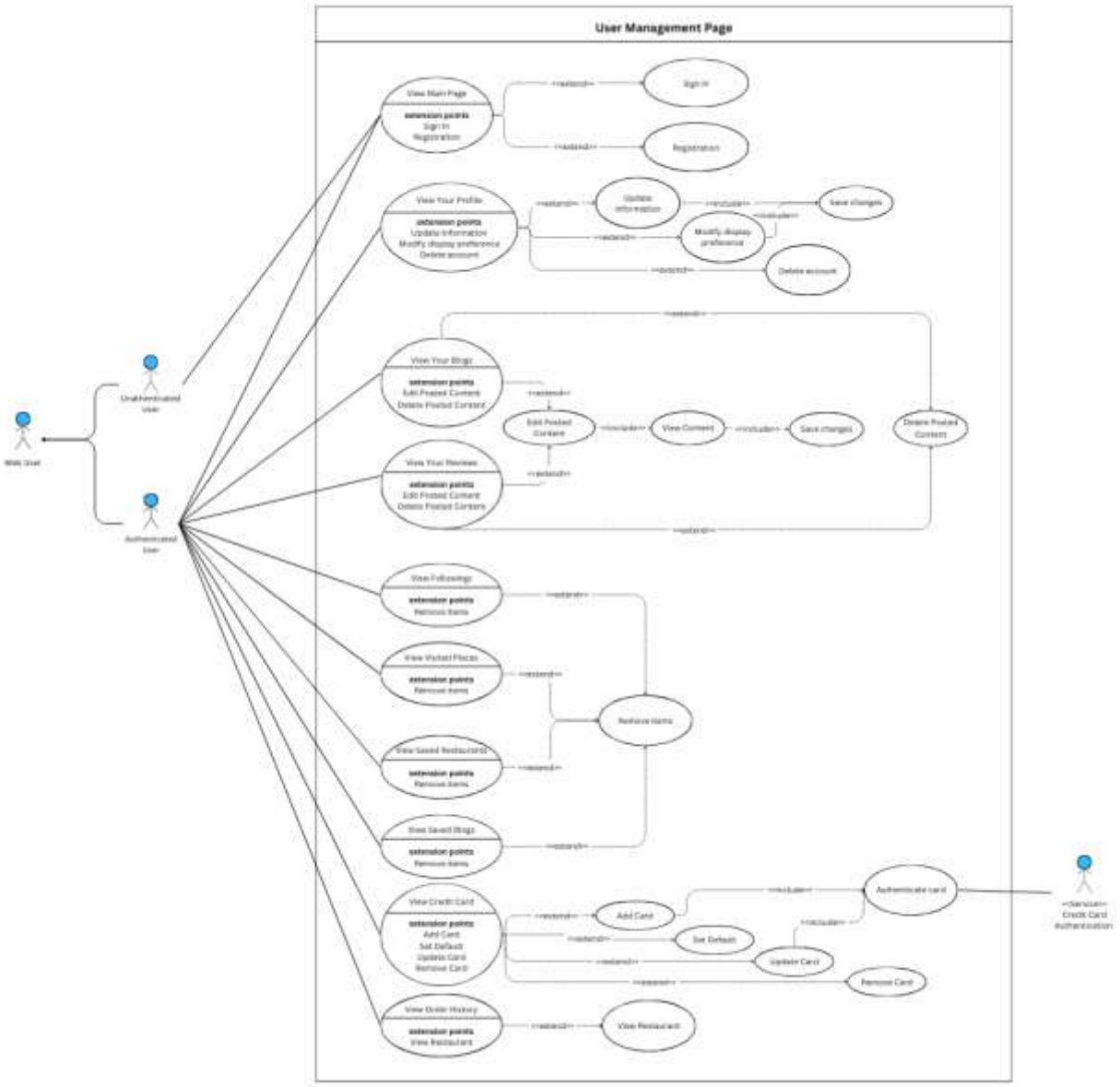
UC01

View on Canva

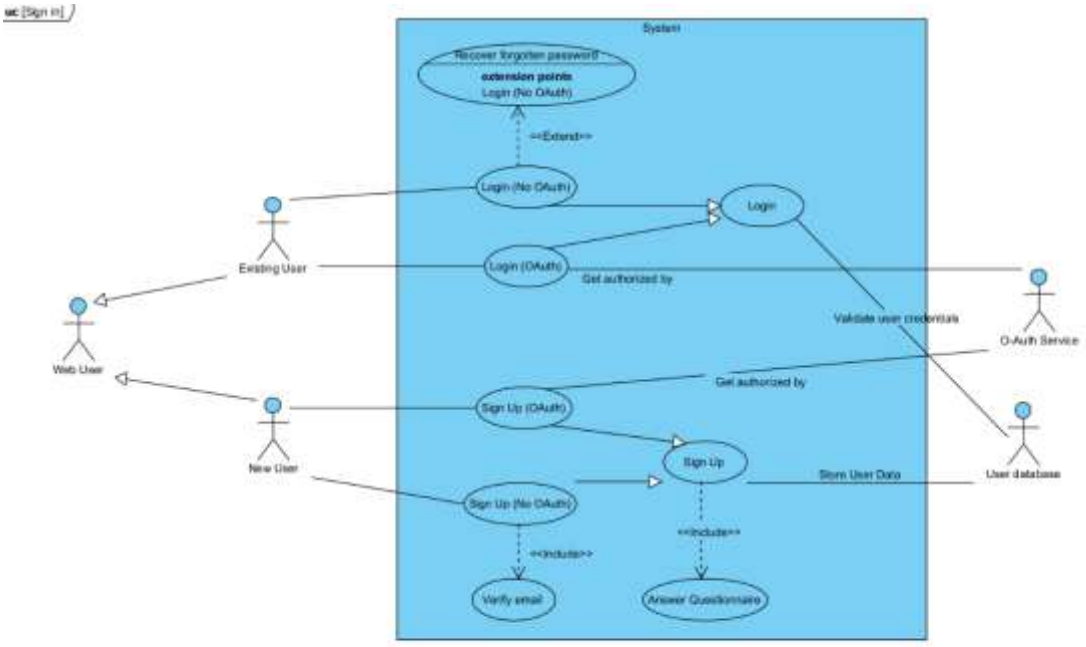


UC02

View on Canva

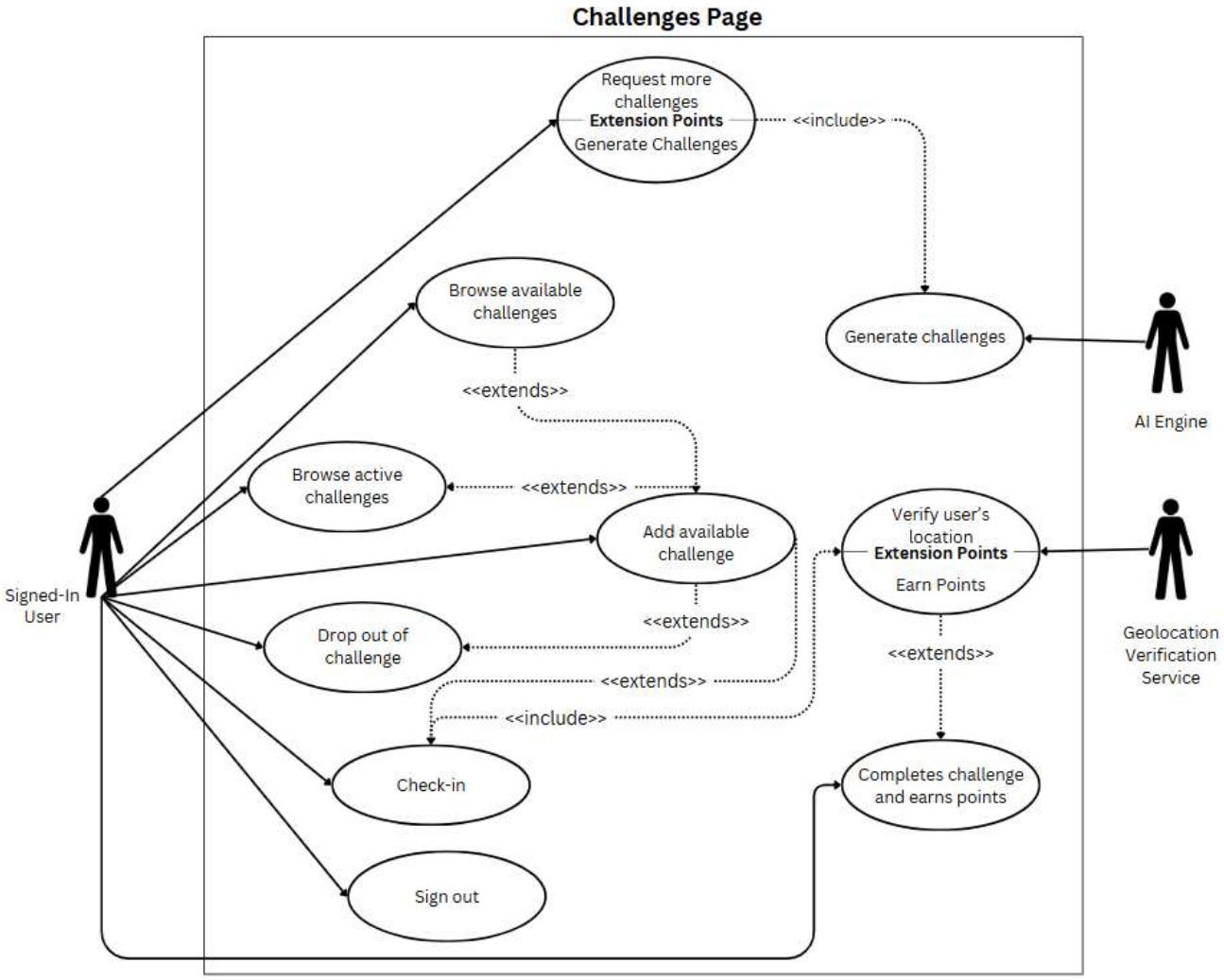


UC03


uc [Sign in]
System
Recover forgotten password
extension points
Login (No OAuth)
<<Extend>>
Login (No OAuth)
Login
Existing User
Login (OAuth)
Get authorized by
Validate user credentials
O-Auth Service
Web User
Get authorized by
Sign Up (OAuth)
New User
Sign Up
Store User Data
User database
Sign Up (No OAuth)
<<Include>>
<<Include>>
Verify email
Answer Questionnaire

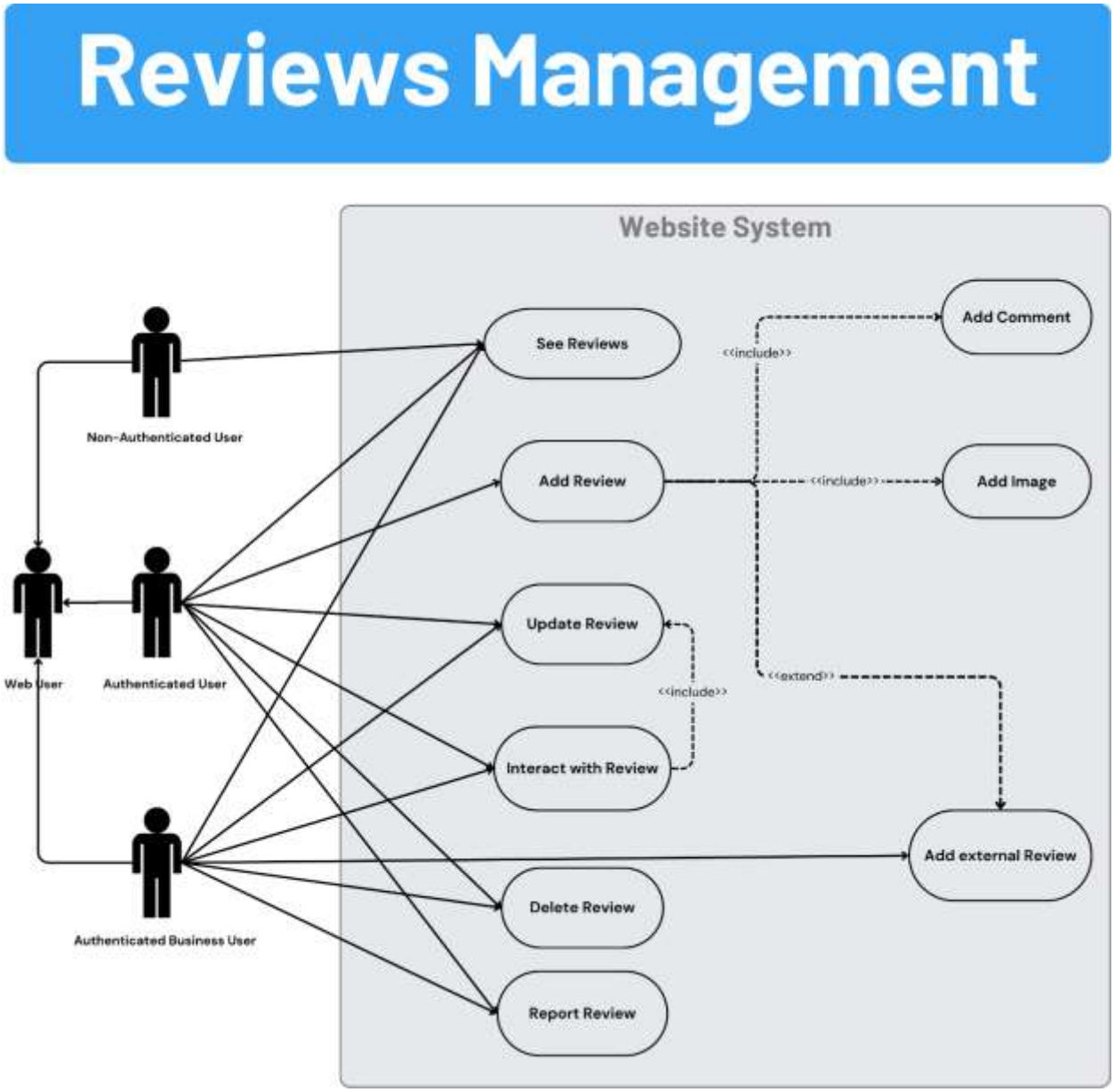
UC04

View on Canva



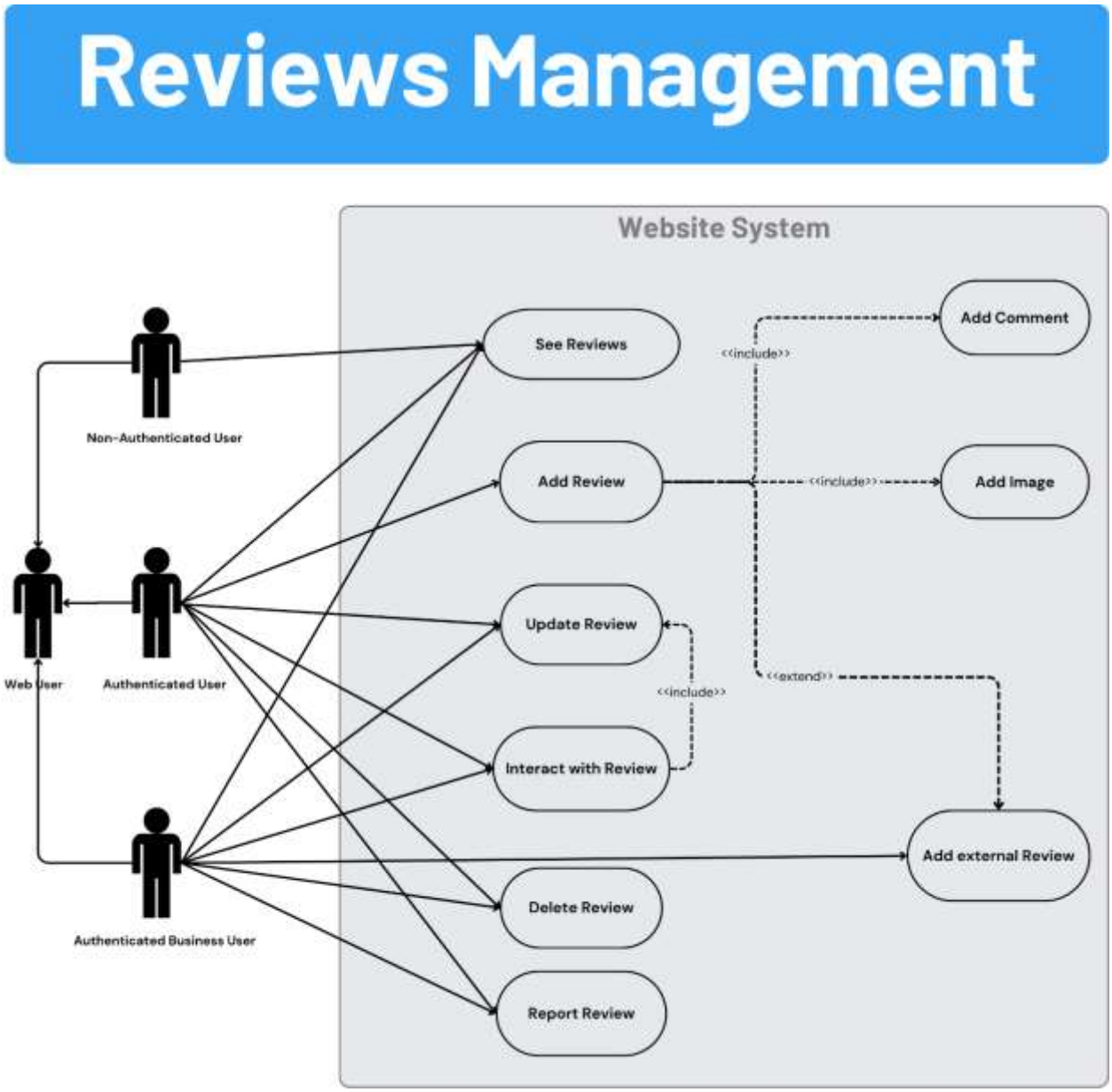
UC05 – Review System (User)

View on Canva



UC06 – Review System (Business User)

View on Canva



3.2.3 Use Case Descriptions Tables

UC01

User Story: As a food enthusiast, I want to discover more restaurants and read related experiences of other people so that I can broaden my palate and explore all my dining options.

Use Case Name: Interact with the Main, Restaurants, and Blog Posts Pages	**ID**: UC01	**Importance Level**: Medium
Primary Actor: Web User (Signed In / Not Signed In)	**Use Case Type**: Essential	
Stakeholders and Interests: User: Wants to explore the *BiteClub* website by browsing content, searching for restaurants or blog posts, and engaging with the content (via comments/reactions) if signed in. *BiteClub* System: Provides access to the main page, restaurant listings, blog posts, and the ability to search and interact with content based on the user's sign-in status.		
Brief Description This use case describes how the user interacts with the website by navigating through the main, restaurant, and blog posts pages. They can search for content (restaurants or blog posts) and, if signed in, post comments and reactions on blog posts.		
Trigger: The user accesses the website through the main page URL or by navigating directly to the restaurant or blog posts pages. **Type**: Internal		
Relationships: **Associations**: Web User, *BiteClub* System **Include**: Sign Out, Sign In, Registration **Extend**: Filter Restaurants, Search for the Restaurants, Search for the Blog **Generalization**: • View Page: Signed In User: Can access personalized recommendations and post comments or reactions.		

Not Signed In User: Can view top 50 restaurants and posts but cannot post comments or reactions.

- Search for content:

Search for Restaurants: User can search for restaurant listings.

Search for Blog Posts: User can search for blog posts.

Normal Flow of events:

1. The user navigates to the main page, restaurants page, or blog posts page. If the user is signed in, they see personalized recommendations; if not, they see top 50 restaurants and posts list.
2. The user can search for a restaurant or for a blog post.
3. The user can filter the restaurants.
4. If signed in, user can post a comment or reaction

Sub flow:

User signed in flow:

1. User clicks on a blog post.
2. If the user is signed in, they see options to post a comment or reaction.
3. User submits a comment/reaction.

User not signed in flow:

1. User clicks on a blog post.
2. If the user is not signed in, they don't see options to post a comment and cannot react to a post.

Alternate/Exceptional Flow:

1. If the search returns no results:
    a. If the user searches for content and there are no results, the system shows a message indicating no results found and prompts the user to try again.

UC02

User Story: As a food influencer, I want to modify my posted content and control the information I share with the public so that I can keep my blog up to date and correct any typos or mistakes. Additionally, I need a function to update my personal information, allowing me to refresh my public image as needed.

Use Case Name: User Portfolio Management	**ID**: UC02	**Importance Level**: High
Primary Actor: Authenticated Users	**Use Case Type**: Essential	

Stakeholders and Interests:

Authenticated Users

- Want to manage their personal information, blogs, reviews, and saved content.
- Need the ability to edit or delete their posted content.
- Expect secure handling of credit card information for transactions.

Unauthenticated Users

- Want access to the system but must register or log in to manage content.

System Administrator

- Ensures secure access and data integrity for user management.

Credit Card Authentication Service Provider

- Verifies credit card details for authentication and transactions.

Brief Description

This use case describes what authenticated users can view and modify on their profile management page compared to unauthenticated users.

Trigger: The user navigates to the user management page.

Type: Internal

Preconditions: The user must be logged in to access the user management page.

Postconditions: The user's profile changes are saved successfully.

Relationships:

Associations:

Web User (authenticated users and unauthenticated users), Credit card authentication service provider

Include:

View Content, save changes, authenticate card

Extend:

Sign In, registration, update information, modify display preference, delete account, edit posted content, delete posted content, remove items, add card, set default, update card, remove card, view restaurant.

Generalization:

View Page:

- Authenticated User: Can access the user management page to update and modify content.
- Unauthenticated User: Does not have the URL to access the user page. They can sign in or register.

Sort by:

- Sort by Date: User can sort posted blogs and reviews by date.
- Sort by Name: User can sort posted blogs and reviews by title.

Normal Flow of events:

1. The user navigates to the main page. If authenticated, they can access the user management page; otherwise, they can remain on the main page and click on 'Sign In' or 'Register' to access the user management page.
2. Authenticated users first land on the 'Your Profile' page.
3. They can navigate to other sections using buttons, including 'Your Blogs', 'Your Reviews', 'Followings', 'Visited Places', 'Saved Restaurants', 'Saved Blogs', 'Credit Card', 'Order History'.
4. Authenticated users can modify items under certain sections.
5. If users modified an item, they must save the changes in order to update it to the database.

Sub flow:

"Your Blogs" / "Your Reviews":

1. Only the 'Your Blogs' and 'Your Reviews' sections have an edit mode, as the content is created by the user. 2. Users can view all items in the section (i.e., their blogs or reviews). 3. If a user selects 'View' or 'Edit', they will be redirected to the expanded version for editing. a. After editing, users will need to save the modification. 4. If users choose to remove an item, no redirection is needed. 'Credit Card': 1. Adding or editing a card requires authentication by the credit card authentication service provider. 'Followings', 'Visited Places', 'Saved Restaurants', 'Saved Blogs': 1. Users can view all items in these sections. 2. Users are allowed to remove items.
Alternate/Exceptional Flow: - N/A

UC03

User stories: As a user, I want to be able to register and login so that I can access the app features requiring an account.

Use Case Name: Login (NO OAuth)	**ID**: UC03	**Importance Level**: High
Primary Actor: Web User (Not logged In)	**Use Case Type**: Essential	
Stakeholders and Interests: User: Wants to explore the *BiteClub* website by browsing content, searching for restaurants or blog posts, and engaging with the content.		
Brief Description This use case described the process of logging in. In this method, the user can log in by using their email and password. The OAuth login is covered in another use case.		

Trigger: The user opens the sign in page through clicking the sign in button or directly visiting the website.

Type: Internal

Preconditions: User is on the login page

Postconditions: User is logged in

Relationships:

Associations: Web User, User Database

Include:

Extend: Recover password

Generalization: Log in

Normal Flow of events:

1. User uses the OAuth service or email and password to sign in
2. User information gets sent to the server.
3. Server checks the database to see if the user is registered
4. If user is found and has provided correct details, sign in user.

Sub flow:

User uses OAuth

1. User clicks the OAuth service.
2. User gets authorized by the service.
3. User information is sent to the server.
4. The server validates that the user has previously registered.
5. Allows the user to log in if validation is successful.

User not signed in flow:

1. User enters email and password.
2. The encrypted information is sent to the server.
3. The server validates the information with the information found in database.
4. Allows the user to log in if successful.

Alternate/Exceptional Flow:

User forgets password: 1. User click forgot password 2. System sends a verification code to email 3. User can optionally resend another verification email. 4. User enters the verification code. If correct, the system allows the user to enter and save new password.

UC04

User Story: As a picky eater, I want to participate in food challenges that encourage me to explore foods outside my comfort zone, so that I can expand and diversify my palate while earning points for discounts on future orders and restaurant visits.

Use Case Name: Interact with Challenges	**ID**: UC04	**Importance Level**: High
Primary Actor: Signed-In User	**Use Case Type**: Essential	
Stakeholders and Interests: User: Participates in food challenges to earn points and get discounts on future orders. AI Engine: Generates challenges based on user preferences and activity. Geolocation Verification Service: Verifies the user's location when they check in. Application System: Provides access to the challenges page, manages the user's interactions with challenges, and coordinates between the user and the supporting services (i.e. AI Engine and Geolocation Verification Service).		
Brief Description This use case describes how registered users interact with the *Challenges* page. Once users sign into their accounts, they can browse challenges, request for more challenges to browse, add challenges so they can participate in them, and drop out of challenges. They progress through challenges by checking-in when they are at the given restaurant. Once they complete a challenge (i.e. visited each restaurant), they earn points.		
Trigger: User navigates to the *Challenges* page after signing in. **Type**: External		
Relationships 1. The Signed-In User interacts with the system to browse, participate in, progress through, and complete challenges. 2. The AI Engine generates challenges based on user preferences and activity.		

3. The Geolocation Verification Service verifies the user's location when they check in.

Associations:

User:

1. Browse available challenges
2. Request more challenges
3. Browse active challenges
4. Add available challenge
5. Drop out of challenge
6. Check-in
7. Earn points
8. Sign out

AI Engine:

1. Generate challenges

Geolocation Verification Service:

1. Verify user's location

Include:

1. User check-in includes verification of user's location.
2. User request for more challenges includes AI generation of challenges.

Extend:

1. Browsing available challenges extends to adding a challenge.
2. Adding a challenge extends to browsing active challenges, check-in, and dropping out of a challenge.
3. Verification of user location extends to user earning points.

Generalization:

Not applicable to use case.

Normal Flow of events:

1. User navigates to the *Challenges* page.
2. User browses the available challenges.
3. User adds a challenge.
4. User checks-in.
5. User's location is verified through Geolocation Verification Service.
6. If user checks-in at the last remaining restaurant, the challenge is complete and user is rewarded points.

Sub flow: • After adding a challenge, User may drop out of challenge. • After browsing available challenges, User may request more challenges to browse. • After navigating to the Challenges page, User may browse challenges they are currently participating in (active challenges).
Alternate/Exceptional Flow: 1. If User's location cannot be verified after check-in: a. The Application System will inform User that they are not at the expected location. b. User may attempt to check-in again.

UC05

User Story: As a signed-in user, I want to write a review on a restaurant's profile so that I can share my experience and help others make informed decisions. I should be able to add text, attach images, or submit an external review using an Instagram post link. Additionally, I can edit or delete my reviews and interact with others by liking or reporting them. Reviews should be ranked based on engagement and credibility scores to ensure useful feedback is highlighted. If I navigate away while writing a review, I should receive a prompt to save or discard my progress.

Use Case Name: User writes a Review	**ID**: UC05	**Importance Level**: High
Primary Actor: Signed-In User	**Use Case Type**: Essential	
Stakeholders and Interests: Signed-In User: Leaves reviews on restaurant profiles earning credibility and engagement score based on activity on their reviews. Application System: Provides access to the reviews on restaurant profiles, manages the user's interactions with reviews and allows addition, deletion and updating of reviews.		
Brief Description This case describes how registered users interact with the reviews on a *Restaurant* page. Once users sign into their accounts, they can navigate to a restaurant's profile page, leave, like or report a review. They can essentially add, edit or delete their review. Leaving reviews or interacting with other reviews boosts a user's credibility and engagement score and the reviews on a profile are prioritized by these scores, giving other users reliable information		
Trigger: User navigates to a Restaurant's profile page after signing in.		

Type: External

Relationships

Associations:

Signed-In User:

1. Browse existing reviews on a Restaurant profile
2. Add a review
3. Add external review
4. Update a review
5. Delete a review
6. Like a review
7. Report a review

Include:

1. Add an image as part of a review
2. Add a text comment as part of a review
3. Update the data associated with review when a user interacts with it

Extend:

1. Add an external review (Instagram post link)

Generalization:

Not applicable to use case.

Normal Flow of events:

1. User navigates a *Restaurant's* profile page.
2. User clicks on add a review.
3. User writes a (text + image) review or adds external review (Instagram post link).
4. User posts the review.

Sub flow:

- Users can navigate and see all the reviews on a restaurant's profile.
- Users can modify their existing reviews on a restaurant's profile.
- Users can delete their existing reviews on a restaurant's profile.
- Users can like any review.
- Users can report any review.

Alternate/Exceptional Flow:

1. If user navigates away from profile page while writing a review. a) User will be prompted to complete the review and given the option to exit without saving.

UC06

User Story: As a signed-in business user, I want to reply to reviews on my restaurant's profile so that I can engage with customers and address their feedback. I should be able to browse all reviews on my profile, post a text reply to any review, and edit or delete my replies if needed. Additionally, I can attach Instagram posts as external reviews to enhance my profile. I should also have the ability to like or report reviews to ensure a well-moderated review section. If I navigate away while composing a reply, I should receive a prompt to save or discard my response.

Use Case Name: Business user replies to a review	**ID**: UC06	**Importance Level**: Medium
Primary Actor: Signed-In Business User	**Use Case Type**: Essential	
Stakeholders and Interests: Signed-In Business User: Navigates through the reviews on their profile, engaging with them and boosting connection with their customers. Application System: Provides access to the reviews on a business user's, allows a user to reply to a review and manages the user's interactions with reviews and allows addition, deletion and updating of external reviews.		
Brief Description This case describes how registered business users interact with the reviews on their profile. Once business users sign into their accounts, they can navigate their profile and see all reviews. They can add external reviews (Instagram post links) to boost their profile and reply to reviews to engage with customers. They can also like or report reviews.		
Trigger: User navigates to their Restaurant profile page after signing in. **Type**: External		
Relationships **Associations**: Signed-In User: 1. Browse existing reviews on a Restaurant profile 2. Add a review		

3. Add external review
4. Update a review
5. Delete a review
6. Like a review
7. Report a review

Include:

N/A

Extend:

N/A

Generalization:

Not applicable to use case.

Normal Flow of events:

1. User navigates to their profile
2. User navigates to a review on their profile
3. User clicks on reply button on a review
4. User writes a text-only reply to the review
5. User posts their reply

Sub flow:

- Users can navigate and see all the reviews on their profile.
- Users can modify their existing replies to reviews on their profile.
- Users can delete their existing replies to reviews on a restaurant's profile.
- Users can like any review.
- Users can report any review.

Alternate/Exceptional Flow:

1) If user navigates away from profile page while writing a reply to a review.
   a) User will be prompted to complete the review and given the option to exit without saving.

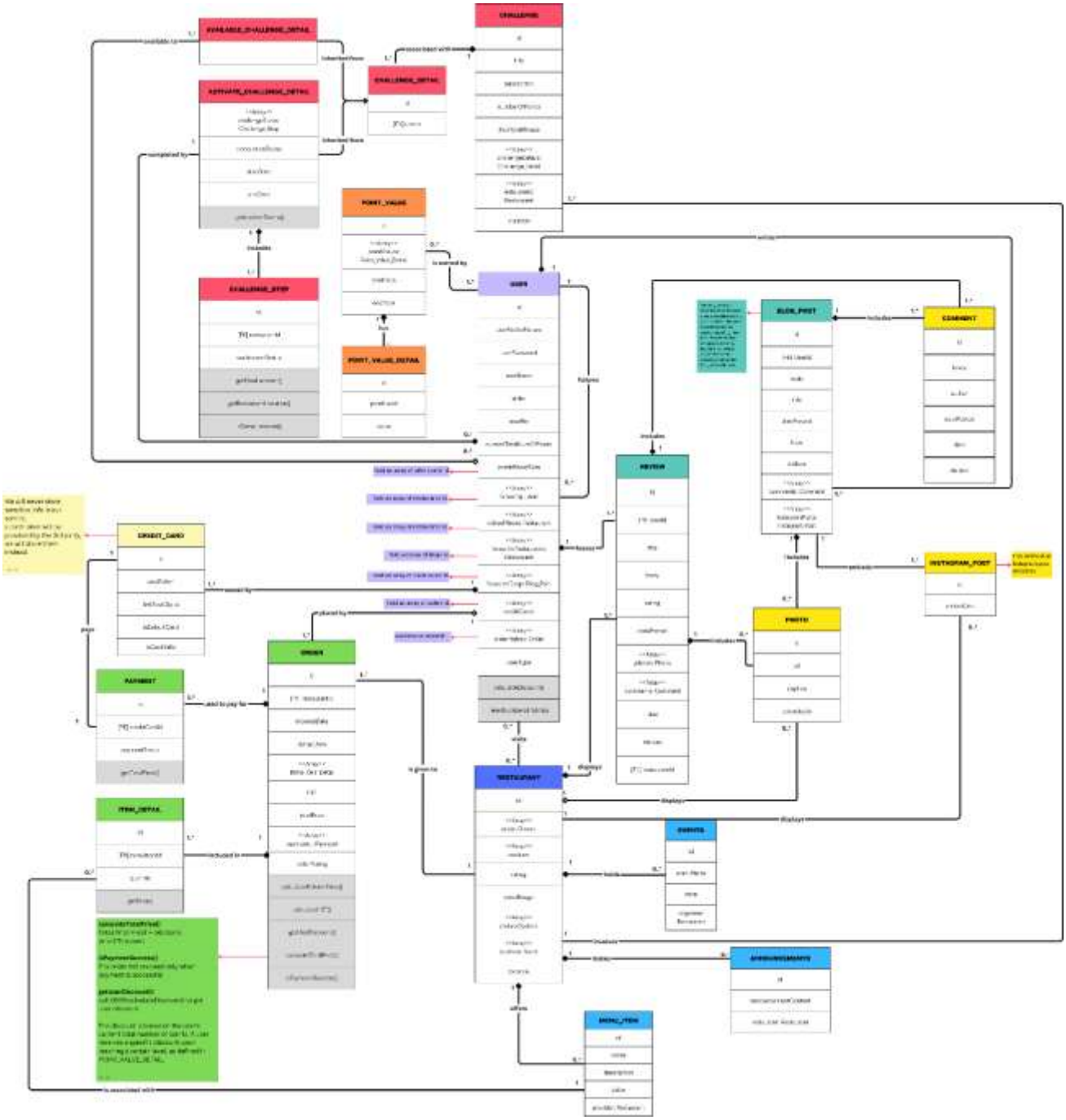
3.2.4 Corresponding Mockups

Use Case	Corresponding Mockup
UC01	Section 2.8 - Scenario 2
UC02	Section 2.8 - Scenario 4

UC03	Section 2.8 - Scenario 1
UC04	Section 2.8 - Scenario 5
UC05	Section 2.8 - Scenario 3
UC06	Section 2.8 - Scenario 3

4. Domain Class Diagram (updated)

View on Canva



5. Database

Data Model (Collections)

Refer to the database models report at: Documents

USER COLLECTION	
id	Integer
username	String
userBio	String
userProfilePicture	url: String caption: String updated_at: Date
strike	Integer
numOfPoints	Integer
pointsResetDate	Date
following	FOLLOWING ARRAY
visitedPlaces	VISITED_PLACES ARRAY
favouriteRestaurants	FAVOURITE_RESTAURANTS ARRAY
favouriteBlogs	FAVOURITE_BLOGS ARRAY
creditCards	CREDIT_CARDS ARRAY
orderHistory	ORDER_HISTORY ARRAY
userType	String
location	type: String coordinates: Double Array [longitude, latitude]

FOLLOWING ARRAY	
user_id	Integer

VISITED_PLACES ARRAY	
restaurant_id	Integer

FAVOURITE_RESTAURANTS ARRAY	
restaurant_id	Integer

FAVOURITE_BLOGS ARRAY	
blog_post_id	Integer

CREDIT_CARDS ARRAY	
credit_card_id	Integer

ORDER_HISTORY ARRAY	
order_id	Integer

BLOG_POST COLLECTION	
id	Integer
body	String
title	String
date_posted	Date
likes	count: Integer users: Integer Array (Array of user id's) [1, 2, 3]
dislikes	count: Integer users: Integer Array (Array of user id's) [1, 2, 3]
comments	COMMENTS ARRAY
Instagram_posts	INSTAGRAM_POSTS ARRAY
photos	PHOTOS ARRAY
user_id	Integer

COMMENTS ARRAY	
id	Integer
body	String
author	String
date_posted	Date
likes	count: Integer users: Integer Array (Array of user id's) [1, 2, 3]
dislikes	count: Integer users: Integer Array (Array of user id's) [1, 2, 3]

INSTAGRAM_POSTS ARRAY	
id	Integer
embedLink	String

PHOTOS ARRAY	
id	Integer
url	String
caption	String
updated_at	Date

REVIEW COLLECTION	
id	Integer
body	String
title	String
rating	Integer
date_posted	Date
comments	COMMENTS ARRAY
photos	PHOTOS ARRAY
likes	count: Integer users: Integer Array (Array of user id's) [1, 2, 3]
dislikes	count: Integer users: Integer Array (Array of user id's) [1, 2, 3]
user_id	Integer
restaurant_id	Integer

CREDIT_CARD	
id	Integer
cardToken	String
lastFourDigits	Integer
isDefaultCard	Boolean
isCardValid	Boolean

PAYMENT	
id	Integer
creditCardId	Integer

paymentStatus	String

ITEM_DETAIL	
id	Integer
menuItemId	Integer
quantity	Integer

ORDER	
id	Integer
restaurantId	Integer
orderedDate	Date
visitedDate	Date
items	ITEM_DETAIL Array
HST	Double
totalPrice	Double
payments	PAYMENT Array
orderRating	Double

CHALLENGE	
id	Integer
title	String
description	String
numberOfPoints	Integer
thumbnailImage	String
challengeDetails	CHALLENGE_DETAIL Array
restaurants	RESTAURANT Array
duration	Integer

CHALLENGE_DETAIL (ABSTRACT)	
id	Integer
userId	Integer

AVAILABLE_CHALLENGE_DETAIL	
id (inherited from CHALLENGE_DETAIL)	Integer
userId (inherited from CHALLENGE_DETAIL)	Integer

ACTIVATE_CHALLENGE_DETAIL	
id (inherited from CHALLENGE_DETAIL)	Integer
userId (inherited from CHALLENGE_DETAIL)	Integer
challengeSteps	CHALLENGE_STEP Array
completionStatus	String
startDate	Date
endDate	Date

CHALLENGE_STEP	
id	Integer
restaurantId	Integer
verificationStatus	Boolean

POINT COLLECTION	
id	Integer
details	POINT_VALUE_DETAIL ARRAY
start_date	Date
end_date	Date

POINT_VALUE_DETAIL ARRAY	
detail_id	Integer
pointLevel	Integer
value	Double

INSTAGRAM_POST	
postId	Integer
embedLink	String

EVENT	
eventId	Integer
eventName	String
eventDate	Date
organizingRestaurant	Integer

ANNOUNCEMNT	
announcementId	Integer
announcementDetails	String
announcingRestaurant	Integer

MENU_ITEM	
id	Integer
name	String
description	String
price	Double

RESTAURANT	
id	Integer
orders	ORDER Array
cuisines	MENU_ITEM array
rating	Double
priceRange	min: Double max: Double
dietaryOptions	String Array
BusinessHours	BUSINESS_HOURS Array
location	String

BUSINESS_HOURS	
day	String
opening	String
closing	String

Data Sample (JSON Documents)

These JSON documents represent entries in each of the main collections, showing how data is organized within the NoSQL database.

Link to the .JSON file

User Collection Document


```
{
"id": 1,
"username": "sarah008",
"userBio": "I love food.",
"userProfilePicture": {
"url": "https://example.com/photos/my_pic.jpg",
"caption": "",
"updated_at": "2025-02-11T18:45:00Z"
},
"strike": 0,
"numOfPoints": 2200,
"pointsResetDate": "2026-02-01T00:00:00Z",
"following": [
{
"user_id": 2
},
{
"user_id": 3
},
{
"user_id": 4
},
{
"user_id": 7
},
{
"user_id": 10
}
],
"visitedPlaces": [
{
"restaurant_id": 1
},
{
"restaurant_id": 8
},
{
"restaurant_id": 235
},
{
"restaurant_id": 2
```

```
}
],
"favouriteRestaurants": [
{
"restaurant_id": 8
},
{
"restaurant_id": 2
}
],
"favouriteBlogs": [
{
"blog_post_id": 3
},
{
"blog_post_id": 24
}
],
"creditCards": [
{
"credit_card_id": 1
},
{
"credit_card_id": 2
}
],
"orderHistory": [
{
"order_id": 1
},
{
"order_id": 23
},
{
"order_id": 158
},
{
"order_id": 232
}
],
"userType": "General",
"location": {
 "type": "Point",
 "coordinates": [-79.3832, 43.6532]
```

```
}

}
```

Blog Post Collection Document


```
{
"id": 1,
"body": "This is the body of Post 1",
"title": "Title 1",
"date_posted": "2025-02-12T00:00:00Z",
"likes": {
"count": 3,
"users": [101, 102, 103]
},
"dislikes": {
"count": 2,
"users": [201, 202]
},
"comments": [
{
"id": 1,
"body": "This is the body of Comment 1",
"author": "Jane MacDonald",
"date_posted": "2025-02-15T00:00:00Z",
"likes": {
"count": 3,
"users": [101, 102, 103]
},
"dislikes": {
"count": 2,
"users": [201, 202]
},
},
{
"id": 2,
"body": "This is the body of Comment 2",
"author": "Larry Smith",
"date_posted": "2025-02-17T00:00:00Z",
"likes": {
"count": 3,
"users": [101, 102, 103]
},
"dislikes": {
"count": 2,
```

```
"users": [201, 202]
}
},
{
"id": 3,
"body": "This is the body of Comment 3",
"author": "Kate Shook",
"date_posted": "2025-02-17T00:00:00Z",
"likes": {
"count": 3,
"users": [101, 102, 103]
},
"dislikes": {
"count": 2,
"users": [201, 202]
}
}
],
"instagram_posts": [
{
"id": 1,
"embedLink": "https://www.instagram.com/p/BJFGFIhyTA/"
},
{
"id": 2,
"embedLink": "https://www.instagram.com/p/CRYzH8JjHJL/"
}
],
"photos": [
{
"id": 1,
"url": "https://example.com/photos/mountain_view.jpg",
"caption": "A breathtaking view of the mountains",
"updated_at": "2025-02-10T14:23:00Z"
},
{
"id": 2,
"url": "https://example.com/photos/city_lights.jpg",
"caption": "City lights glowing at night",
"updated_at": "2025-02-11T18:45:00Z"
}
],
"user_id": 1
}
```

Review Collection Document


```
{
"id": 1,
"body": "This is the body of Review 1",
"title": "Title 1",
"rating": 4,
"date_posted": "2025-02-15T19:49:00Z",
"comments": [
{
"id": 1,
"body": "This is the body of Comment 1",
"author": "Jane MacDonald",
"date_posted": "2025-02-15T00:00:00Z",
"likes": {
"count": 3,
"users": [101, 102, 103]
},
"dislikes": {
"count": 2,
"users": [201, 202]
}
},
{
"id": 2,
"body": "This is the body of Comment 2",
"author": "Larry Smith",
"date_posted": "2025-02-17T00:00:00Z",
"likes": {
"count": 3,
"users": [101, 102, 103]
},
"dislikes": {
"count": 2,
"users": [201, 202]
}
},
{
"id": 3,
"body": "This is the body of Comment 3",
"author": "Kate Shook",
"date_posted": "2025-02-17T00:00:00Z",
"likes": {
"count": 3,
"users": [101, 102, 103]
```

```
},
"dislikes": {
"count": 2,
"users": [201, 202]
}
}
],
"photos": [
{
"id": 1,
"url": "https://example.com/photos/pizza.jpg",
"caption": "Pizza with peperoni",
"updated_at": "2025-02-10T14:23:00Z"
},
{
"id": 2,
"url": "https://example.com/photos/pasta.jpg",
"caption": "Pasta with tuna",
"updated_at": "2025-02-11T18:45:00Z"
}
],
"likes": {
"count": 3,
"users": [101, 102, 103]
},
"dislikes": {
"count": 2,
"users": [201, 202]
},
"user_id": 1,
"restaurant_id": 1
}
```


Order Collection Document


```
{
"id": 5001,
"restaurantId": 301,
"orderedDate": "2025-03-18T12:30:00Z",
"visitedDate": "2025-03-18T13:00:00Z",
"items": [
{
"id": 2001,
```

```
"menuItemId": 601,
"quantity": 2
},
{
"id": 2002,
"menuItemId": 602,
"quantity": 1
}
],
"HST": 1.50,
"totalPrice": 20.50,
"payments": [
{
"id": 101,
"creditCardId": 1,
"paymentStatus": true,
}
],
"orderRating": 4.5
},
```


Item Detail Collection Document


```
[
{
"id": 2001,
"menuItemId": 601,
"quantity": 2
},
{
"id": 2002,
"menuItemId": 602,
"quantity": 1
}
]
```


Payment Collection Document


```
{
"id": 101,
"creditCardId": 1,
"paymentStatus": true,
```

```
}
```

Credit Card Collection Document

```
{
"id": 1,
"cardToken": "abcd1234efgh5678",
"lastFourDigits": "5678",
"isDefaultCard": true,
"isCardValid": true
}
```

Challenge Collection Document

The Challenge collection includes sample entries for the following classes:

1) Challenge
2) Available_Challenge_Detail
3) Activiate_Challenge_Detail
4) Challenge_Step

Because Challenge_Detail is an *abstract* parent class to Available_Challenge_Detail and Activiate_Challenge_Detail, sample entries are not provided for it explicitly.

```
{
"id": 1,
"title": "Around the World in 60 Days",
"description": "See the world in Toronto! Try a cuisine from 5 different
continents. ",
"numberOfPoints": 400,
"thumbnailImage": "https://databaseexample.com/images/image.jpg",
"challengeDetails": [
{
"id": 1,
"userId": 1
},
{
"id": 2,
"userId": 2
},
{
"id": 3,
"userId": 3,
"challengeSteps": [
{
```

```
"id": 1,
"restaurantId": 1,
"verificationStatus": true
},
{
"id": 2,
"restaurantId": 2,
"verificationStatus": true
},
{
"id": 3,
"restaurantId": 3,
"verificationStatus": false
},
{
"id": 4,
"restaurantId": 4,
"verificationStatus": true
},
{
"id": 5,
"restaurantId": 5,
"verificationStatus": true
}
],
"completionStatus": "progressing",
"startDate": "2025-02-01T00:00:00Z",
"endDate": "2025-03-11T00:00:00Z"
},
{
"id": 4,
"userId": 4,
"challengeSteps": [
{
"id": 1,
"restaurantId": 1,
"verificationStatus": true
},
{
"id": 2,
"restaurantId": 2,
"verificationStatus": false
},
{
```

```
"id": 3,
"restaurantId": 3,
"verificationStatus": false
},
{
"id": 4,
"restaurantId": 4,
"verificationStatus": false
},
{
"id": 5,
"restaurantId": 5,
"verificationStatus": false
}
],
"completionStatus": "dropped",
"startDate": "2025-01-01T00:00:00Z",
"endDate": "2025-01-07T00:00:00Z"
}
],
"restaurants": [
{ "restaurantId": 1 },
{ "restaurantId": 2 },
{ "restaurantId": 3 },
{ "restaurantId": 4 },
{ "restaurantId": 5 }
],
"duration": 60
}
```


Point Collection Document


```
{
"id": 1,
"details": [
{
"detail_id": 1,
"pointLevel": 1000,
"value": 5
},
{
"detail_id": 2,
```

```
"pointLevel": 1500,
"value": 10
},
{
"detail_id": 3,
"pointLevel": 2000,
"value": 15
},
{
"detail_id": 4,
"pointLevel": 2500,
"value": 20
},
{
"detail_id": 5,
"pointLevel": 2750,
"value": 25
}
],
"start_date": "2025-01-01T00:00:00Z",
"end_date": "2027-01-01T00:00:00Z"
}
```


Restaurant Document


```
{
"id": 1,
"details": [
{
"detail_id": 1,
"pointLevel": 1000,
"value": 5
},
{
"detail_id": 2,
"pointLevel": 1500,
"value": 10
},
{
"detail_id": 3,
"pointLevel": 2000,
```

```
"value": 15
},
{
"detail_id": 4,
"pointLevel": 2500,
"value": 20
},
{
"detail_id": 5,
"pointLevel": 2750,
"value": 25
}
],
"start_date": "2025-01-01T00:00:00Z",
"end_date": "2027-01-01T00:00:00Z"
}
```


Menu-Item


```
{
"id": 1,
"name": "Fried Rice",
"description": "Delicious fried white rice",
"price": 13.50
}
```


Comment


```
{
"id": 1,
"body": "I love coming here! Their fried rice is my favourite.",
"author": "19923",
"datePosted": "20-02-2025",
"likes": {
"count": 2,
"users": ["24440", "50001"]
},
"dislikes" {
"count": 2,
"users": ["12342", "92214"]
}
}
```

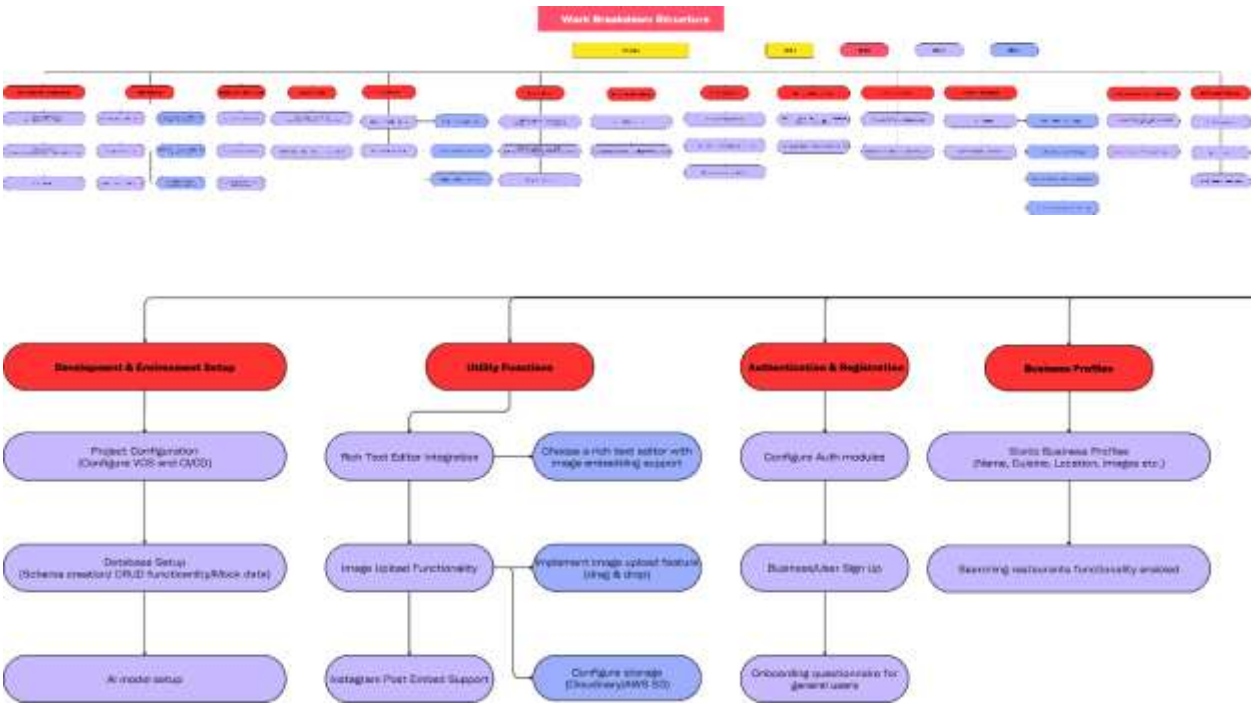

Photo

```
 {
"id": 1,
"url": "www.friedricepicture/photo.png",
"caption": "Tasty Fried Rice!",
"uploadedAt": "2020-02-2025"
}
```

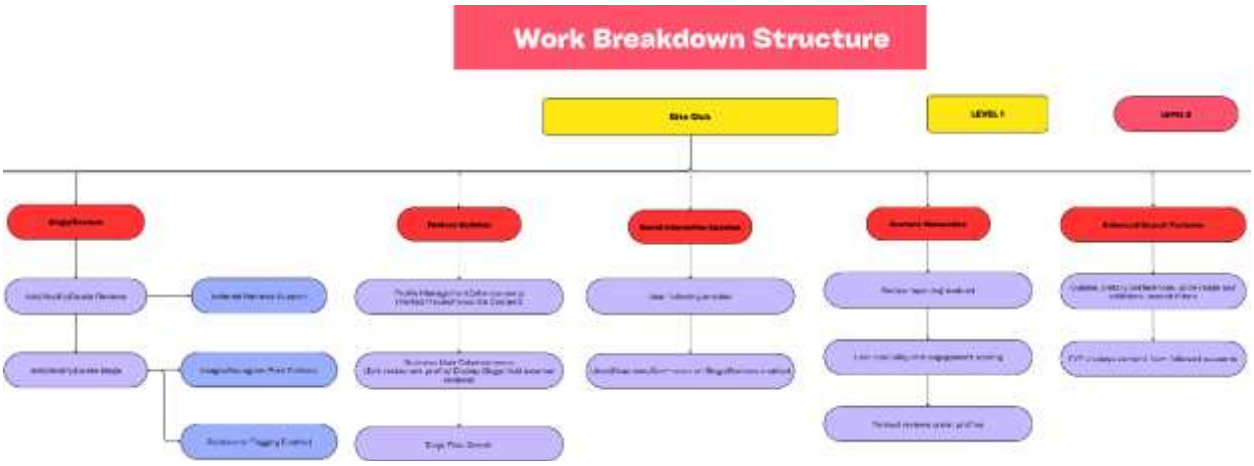
6. Work Breakdown Structure (WBS)

6.1 Work Breakdown Structure

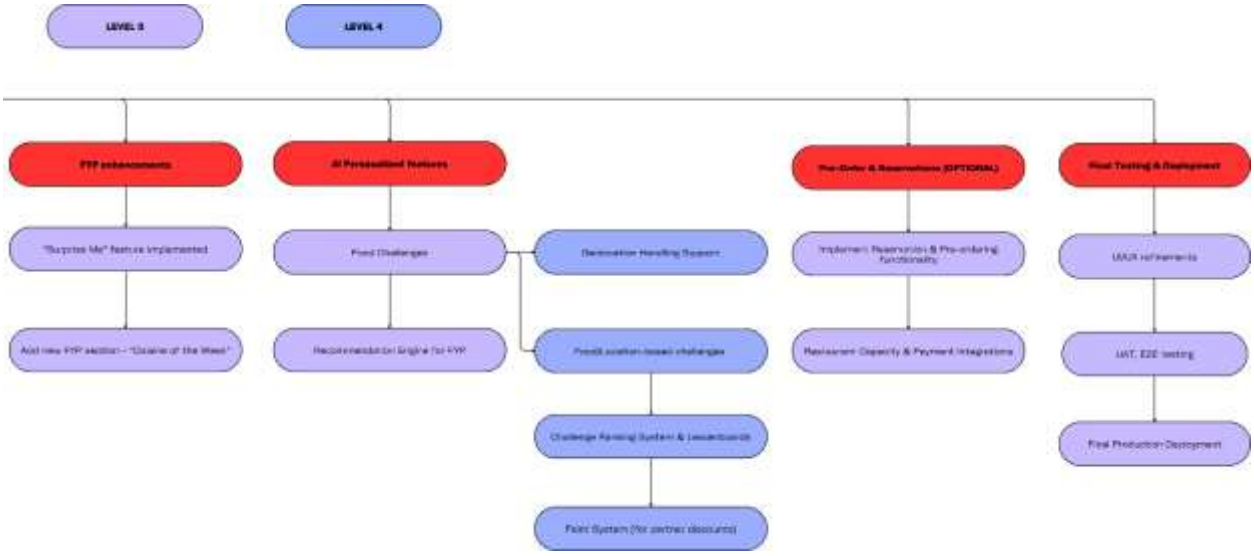
View on Canva



Work Items (Level 2) 1 – 4



Work Items (Level 2) 5 - 9



Work Items (Level 2) 10 – 13

7. Milestones and Acceptance Criteria

7.1 Milestone One - Prototype Development

Users can register as either business or general users via Google or email/password. During registration, general users are prompted with an onboarding questionnaire. After logging in, users can view and update their personal information.

Acceptance Criteria:

- General users are prompted with an onboarding questionnaire during registration.
- Authenticated users can access the user dashboard after logging in.

- Authenticated users can update their personal information (e.g. username, password, profile picture).
- Authenticated business users can update their business information (e.g. basic details, photos, announcements).
- Authenticated users can search for specific restaurant profiles.
- Authenticated users can navigate to a restaurant's profile from the main page.
- Unauthenticated users will have restricted access to functionalities.

7.2 Milestone Two - User Engagement & Content Creation

Users can share their dining experiences through blog posts and reviews, tag restaurants, and enhance their profiles with visited places and favorite content.
Business users can manage their profiles, showcase blogs, and engage with customer feedback through internal and external reviews.

Acceptance Criteria:

- Authenticated users can create, edit, and delete a post and review.
- Authenticated users can embed images, Instagram posts, and tag a restaurant in the post.
- Any user can search for a blog post and get a list of relevant posts based on the search.
- Authenticated users can add restaurants or blog posts to their list of favorite items.
- Authenticated users can manage visited places and favorite content in their profiles.
- Authenticated business users can update restaurant profiles, display relevant posts, and add external reviews.

7.3 Milestone Three - Social Features & Content Moderation

Users will be able to engage with a dynamic food community, and businesses can interact with customers and manage their reputation

Acceptance Criteria:

- Authenticated users can follow other users and see their activity (e.g., blog posts).
- Authenticated users can like, react to, and comment on blog posts and like any helpful reviews.
- Business owners can reply to comments and reviews on their restaurant profiles.
- Authenticated users can report inappropriate content, ensuring a safe and respectful community.
- Registered users will be banned from posting content on the platform if they receive five strikes (for posting inappropriate content)
- The search functionality will offer more filtering options, such as cuisine and dietary preferences for all users.

7.4 Milestone Four – AI – powered challenges & recommendations

Users get smart, tailored recommendations and fun food challenges, making discovery more engaging. Businesses gain traffic from personalized promotions.

Acceptance Criteria:

- Authenticated users can browse, add and drop AI-generated challenges.
- Authenticated users can request to view more AI-generated challenges by scrolling.
- Authenticated users will be rewarded with points after completion of a challenge.
- Authenticated users are ranked on the challenge leaderboard based on the number of challenges they completed.
- An authenticated user can progress through a challenge by confirming their visit to the restaurant and clicking the "Check-In" button.
- The system checks whether the user's geolocation is within 100m of the restaurant's geolocation. Based on the results, the user will receive either a confirmation or an error message.
- On the main page, an authenticated user can view a new set of 10 personalized restaurant recommendations every week that they have not visited or ordered from before.

7.5 Milestone Five – Final Enhancements & Testing

Users enjoy a fully refined app with smooth navigation, while businesses benefit from additional features like reservations.

Acceptance Criteria:

- Users & Business owners experience a polished UI/UX with accessibility improvements.
- All features are rigorously tested with E2E (End-To-End) testing and bug fixes.
- User Acceptance Testing (UAT) is complete, with assured responsiveness across all app pages.
- The application meets all functional and non-functional requirements.
- Successful deployment of application with no issues.

8. Implementation Schedule

Agile Backlog on GitHub Issues: https://github.com/ochovhaniuk98/PRJ566NBB-02/issues

9. Client / Faculty Sign-off

Date: _____, 2025

X_____.

Name of Client/Rep/Professor