****

# PROJECT REPORT PRM

# Măm Map- Smart snack food search application

# **I. Software Requirement Specification**

## **1. Overall Description**

### **1.1 Product Overview**

**Măm Map** is a smart snack food search application that helps users discover snack places tailored to their preferences through personalized filters, GPS-based maps, and AI chatbot recommendations. Users can register, log in, update profiles, view detailed merchant information like menus and ratings, leave feedback, and access their review history. The app ranks snack places using a specialized set of assessment standards based on community reviews and ratings.

**Măm Map Merchant** empowers snack shop owners to manage their stores, menus, and customer reviews, while leveraging premium subscriptions for advanced features like AI-generated images/logos and detailed analytics. Merchants can register, log in, update profiles, purchase premium packages, view comprehensive dashboards, manage transaction history, and change passwords to promote their brands and attract customers through authentic community feedback.

The **Administration Console** allows admins to oversee the platform by managing users, merchants, snack food places, food categories, taste preferences, subscription packages, transaction history, and business models. Admins can approve or reject marketplace entries, review detailed vendor information, and access a comprehensive dashboard with system-wide analytics, ensuring a high-quality platform for snack enthusiasts and merchants alike.

### **1.2 Product Background**

In Vietnam, the demand for snack food is surging, particularly among young people and office workers seeking quick, flavorful options. However, existing food discovery apps often provide generic listings without personalized filters, reliable ratings, or community-driven insights tailored to snack preferences. Măm Map addresses this gap by offering a smart, user-friendly platform that enables users to discover, review, and rate snack spots based on authentic experiences and individual preferences, such as taste, budget, and dietary needs. Designed to foster a vibrant snack food community, Măm Map empowers merchants to manage their stores and engage with customers while providing admins with robust tools to oversee the platform and ensure quality.

### **1.3 Existing Systems**

Most current food-related platforms in Vietnam fall short in delivering a tailored snack discovery experience, lacking:

* **Personalized Filtering**: Options to filter by budget, flavor (e.g., savory, sweet), dietary needs (e.g., vegan, gluten-free), or specific snack categories.
* **Reliable Snack Rating System**: Standardized, trustworthy metrics based on community feedback to rank snack places accurately.
* **Merchant Promotion Tools**: Features for snack shop owners to manage their profiles, menus, and reviews or leverage analytics and AI-driven branding (e.g., logo creation).
* **Community Engagement**: Platforms for users to interact, share reviews, and access community-driven content like blogs.

Măm Map integrates these features into a single platform, offering personalized snack discovery, a robust rating system, merchant management tools, and community interaction to deliver a superior, trust-based experience.

### **1.4 Business Opportunity**

The rising popularity of snack culture and demand for personalized food discovery in Vietnam present a significant opportunity for Măm Map to become the leading snack-focused platform. We aim to serve:

* **Users**: Individuals seeking snack spots that match their taste preferences, budget, dietary needs, and location, with smart recommendations via AI and GPS.
* **Merchants**: Snack shop owners looking to increase visibility through verified reviews, manage their stores and menus, and access analytics or AI branding tools via premium subscriptions.
* **Businesses**: Partners interested in data insights into user snack preferences, behavior, and market trends to inform marketing or product strategies.

By bridging these needs, Măm Map aims to create a thriving snack food ecosystem that connects users, merchants, and businesses.

### **1.5 Software Product Vision**

Our vision is to build Vietnam’s premier snack food ecosystem, delivering a seamless and engaging experience through:

* A modern, mobile-friendly **Flutter interface** that ensures a smooth user experience for browsing, filtering, reviewing, and interacting with snack places.
* **Smart filtering options** based on price, flavor, dietary preferences, and food categories, enhanced by GPS mapping and AI chatbot recommendations.
* A **custom snack rating system** driven by community reviews and standardized criteria for accurate, trustworthy rankings.
* **Merchant dashboards** for managing store profiles, menus, reviews, transaction history, and analytics, with premium features like AI-generated images/logos.
* **Backend analytics tools** to provide personalized recommendations for users and actionable insights for merchants and admins, supported by a robust .NET backend.
* An **admin console** for managing users, merchants, snack places, food categories, taste preferences, subscriptions, transactions, and system analytics.

### **1.6 Project Scope & Limitations**

**Scope:**

* A **Flutter mobile app** supporting:
  + User features: Sign-up/login, snack place browsing, personalized filtering, viewing merchant info (menus, ratings), rating/reviewing, viewing review history, GPS-based maps, and AI chatbot interaction.
  + Merchant features: Store and menu management, review responses, analytics, transaction history, AI image/logo support (premium), dashboard views, and password management.
  + Community interaction: User reviews and ratings fostering a snack-loving community.
* A **.NET backend** enabling:
  + Role-based access for users, merchants, and admins.
  + Management of snack places, food categories, taste preferences, subscription packages, transaction history, and business models.
  + Admin oversight for approving/rejecting snack places, reviewing vendor details, and accessing system-wide analytics via a comprehensive dashboard.

**Limitations:**

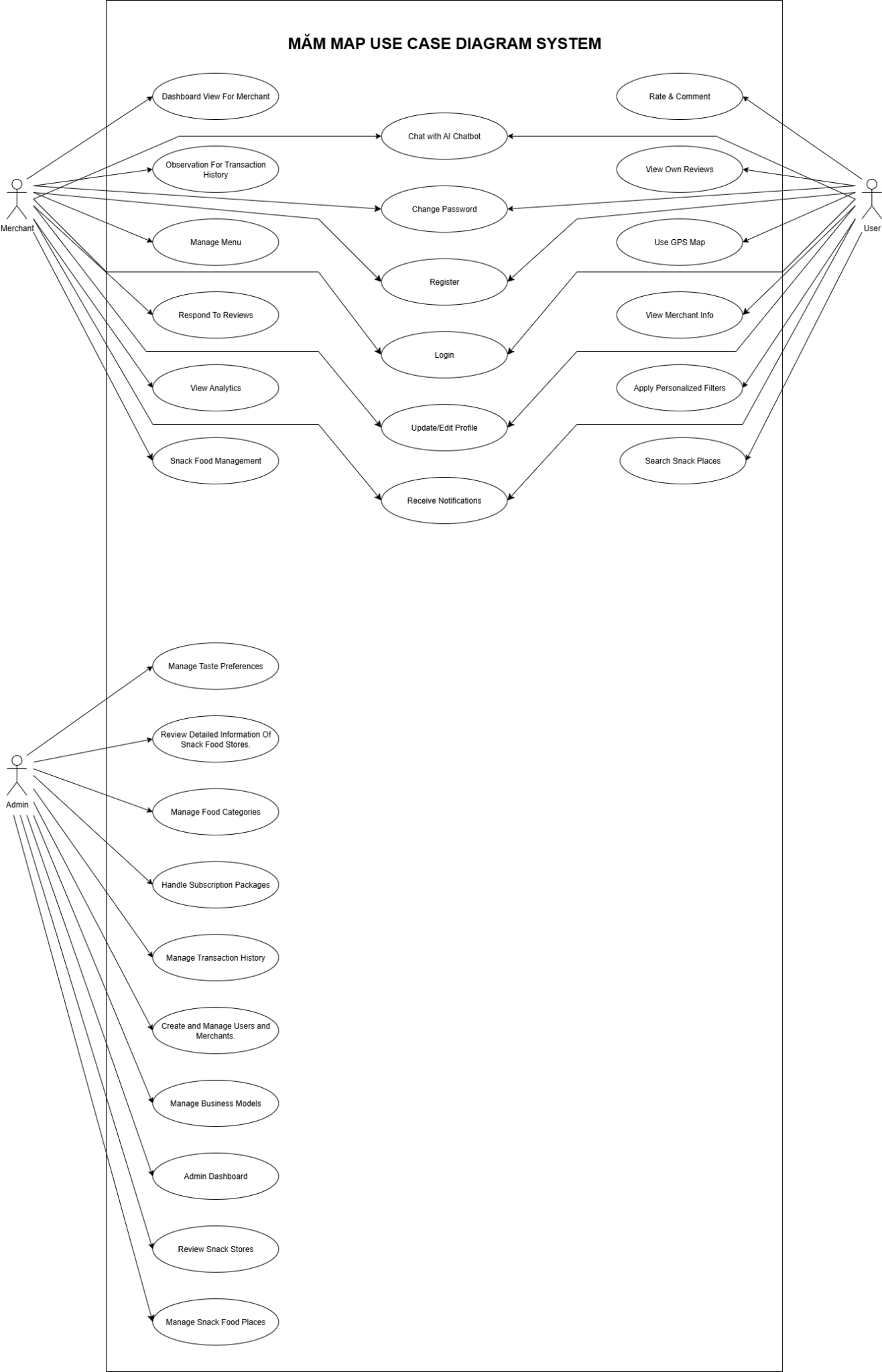
* **No real payment integration**: Transactions (e.g., premium subscriptions) are currently simulated, with full integration planned for future releases.
* **Basic merchant logistics and inventory**: Features like inventory tracking or delivery logistics are limited and slated for future development.
* **Blog/community content**: Currently not implemented, pending future use case or feature definition for blog management.

## **II. User Requirements**

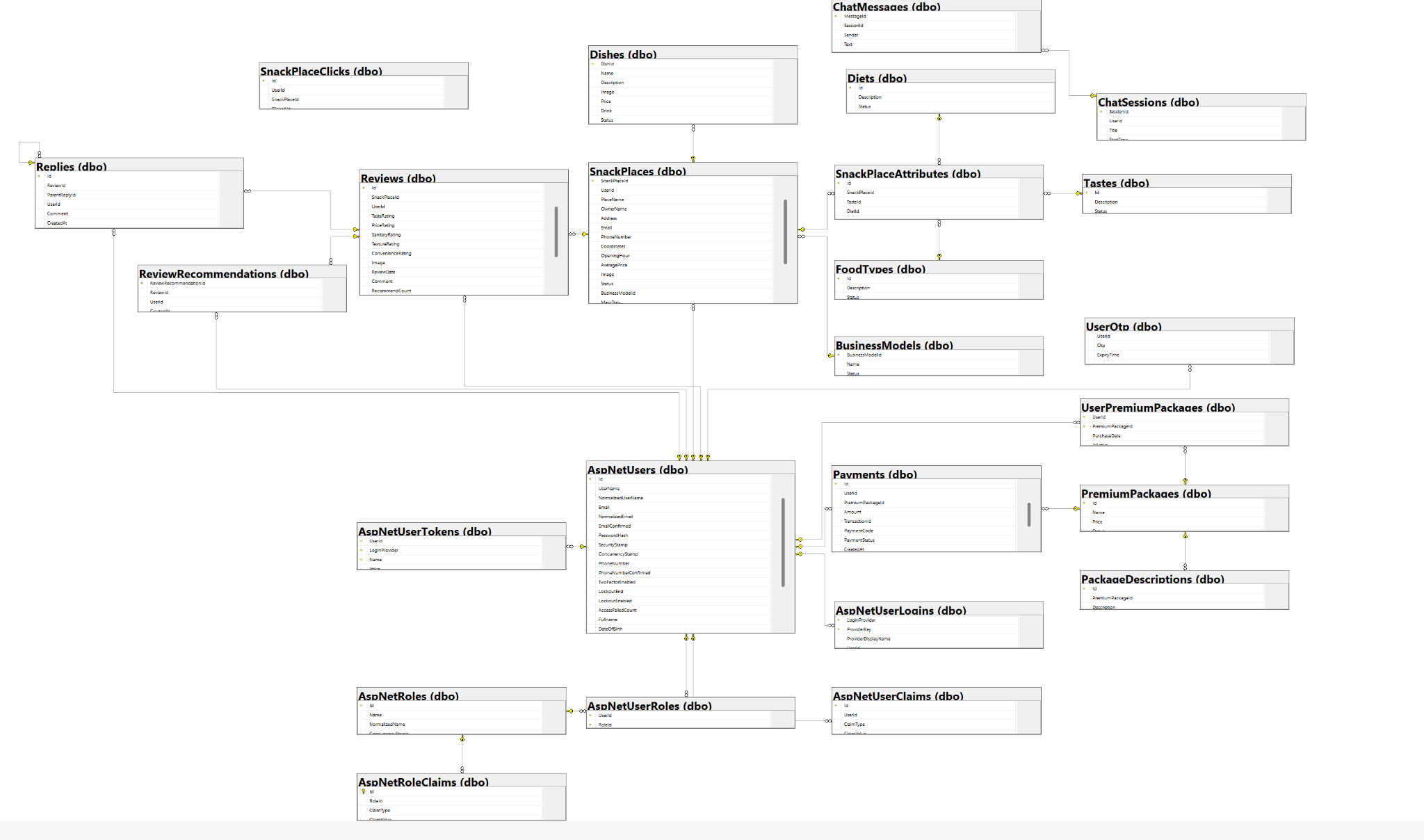
### **2.1 System Actors**

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | **User** | A snack lover using the **Măm Map User App** to explore snack spots, apply personalized filters, view merchant info, rate and review dishes, read blog content, interact with the community, and chat with the built-in AI chatbot trained on curated food data. |
| 2 | **Merchant** | A snack vendor using the **Măm Map Merchant App** to manage shop profiles, upload and update menu items, view customer reviews, respond to feedback, and gain exposure through community engagement. |
| 3 | **Admin** | A system manager accessing the **Admin Web Portal** to oversee user and merchant accounts, manage blog content, moderate reviews, configure reward systems, monitor app usage analytics, and manage the AI chatbot’s training dataset. |

### **2.2 Use Cases Diagram**



### **2.3 ERD Diagram:**



|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use Case** | **Actors** | **Use Case Description** |
| UC\_01 | Register | User, Merchant | Register a new account. |
| UC\_02 | Login | User, Merchant | Log into the system. |
| UC\_03 | Update/Edit Profile | User, Merchant | Edit profile details. |
| UC\_04 | Apply Personalized Filters | User | Filter search results by preference. |
| UC\_05 | Search Snack Places | User | Find snack places by keyword or category. |
| UC\_06 | View Merchant Info | User | View menu, ratings, and shop overview. |
| UC\_07 | Rate and Comment | User | Leave feedback and rate shops. |
| UC\_08 | View Own Reviews | User | Access personal review history. |
| UC\_09 | Use GPS Map | User | View snack places on map. |
| UC\_10 | Chat with AI Chatbot | User, Merchant | Interact with AI chatbot for various support purposes, such as suggestions, image or logo creation, and more. |
| UC\_11 | Purchase Premium Package | Merchant | Buy subscription for premium features. |
| UC\_12 | Manage Menu | Merchant | Add/edit/delete menu items. |
| UC\_13 | Respond to Reviews | Merchant | Reply to user comments. |
| UC\_14 | View Analytics | Merchant | See shop performance metrics. |
| UC\_15 | Snack food management | Merchant | Manage snack food stores, view store names and activity details. |
| UC\_16 | AI images and Logo Support | Merchant | With AI support packages, merchant are able to use AI to generate images or logo |
| UC\_17 | Dashboard view for Merchant | Merchant | View all information and detailed in their own dashboard |
| UC\_18 | Change Password | Merchant/User | Manage their privacy with changing their password in setting based on their old password |
| UC\_19 | Observation for transaction history | Merchant | View list of past transactions/orders. |
| UC\_20 | Manage taste preferences | Admin | CRUD and manage all information allowing users to make selections. |
| UC\_21 | Manage food categories. | Admin | Handle CRUD operations for food types.Administer food classifications. |
| UC\_22 | Handle Subscription Packages | Admin | Create and manage user subscription plans. |
| UC\_23 | Manage transaction history. | Admin | View and manage user transaction history. |
| UC\_24 | Manage business models. | Admin | Define and configure business models. |
| UC\_25 | Dashboard with overall management and complete statistics. | Admin | Comprehensive admin dashboard for managing and tracking all data. |
| UC\_26 | Review detailed information of snack food stores. | Admin | View detailed info, performance metrics, and community reputation scores of snack vendors (ratings, reports, order volume). |
| UC\_27 | Manage Snack Food Places | Admin | Monitor user reports, handle policy violations, temporarily hide/lock listings, or adjust categories/metadata. *Does not approve shop creation.* |

### **2.4 Use Case Specification**

#### ***2.4.1 Member***

##### **2.4.1.1: Register**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_01 – Register | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Member | **Secondary Actors:** | **N/A** |
| **Description:** | The system allows a Member to create a new account. | | |
| **Pre-conditions:** | PRE-1: The user does not have an existing account. | | |
| **Post–conditions:** | POST-1: A new account is created, and the user can log in. | | |
| **Normal Flow:** | 1. The Customer accesses the registration page. 2. The Customer enters information: email, password, full name, phone number. 3. The System validates the input (e.g., email not already registered, password strength). 4. The System sends a verification code (if applicable). 5. The Customer confirms the code (if applicable). 6. The System creates a new account and notifies of successful registration. | | |
| **Alternative Flows:** | 3a. Email already exists: The System displays an error and prompts for a different email.  3b. Invalid data: The System displays an error and prompts for re-entry.  4a. Incorrect/Expired verification code: The System displays an error and allows resending the code. | | |
| **Exceptions:** | Email/SMS delivery failure or database error → display generic error, ask to retry. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | Low | | |
| **Business Rules:** | **BR-1: Password must be at least 8 characters, include uppercase, lowercase, and numeric.** | | |
| **Other Information:** | Email/SMS verification can be added. | | |
| **Assumptions:** | A-1: Users have an internet connection to access the registration page.  A-2: The system has the capability to send email/SMS verification codes. | | |

##### **2.4.1.2: Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_02 – Login | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Member | **Secondary Actors:** | **N/A** |
| **Description:** | The system allows a Member to authenticate and access personalized features. | | |
| **Pre-conditions:** | PRE-1: Member has a valid account.  PRE-2: Member has an active internet connection. | | |
| **Post–conditions:** | POST-1: The customer will be logged in.  POST-2: Member lands on the Dashboard. | | |
| **Normal Flow:** | 1. The Customer accesses the login page. 2. The Customer enters email and password. 3. The System authenticates the provided information. 4. If authentication is successful, the Customer is redirected to the home page. | | |
| **Alternative Flows:** | 3a. Incorrect credentials: The System displays an error and allows re-entry.  3b. Account locked/unverified: The System displays an error and provides instructions for resolution. | | |
| **Exceptions:** | 1-EF: At any time, the app cannot communicate with the server/core system (due to network malfunction/technical issues), the system displays an error message.  2-EF: The data in the registration form is in the wrong format. | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High** | | |
| **Business Rules:** | **BR-2: After 5 failed attempts, lock account for 30 minutes.** | | |
| **Other Information:** | “Forgot Password” link available. | | |
| **Assumptions:** | A-3: Users remember their passwords or a password recovery mechanism is in place.  A-4: The system has a mechanism to handle consecutive failed login attempts (e.g., temporary lockout). | | |

##### **2.4.1.3: Update Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_03 – Manage Profile | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Member | **Secondary Actors:** | **N/A** |
| **Description:** | Members can view and update personal information, settings, and goals. | | |
| **Pre-conditions:** | PRE-1: Member is logged in. | | |
| **Post–conditions:** | POST-1: Updated profile is persisted. | | |
| **Normal Flow:** | 1. Member navigates to Profile. 2. The system displays current profile data. 3. Members edit fields (name, avatar, target quit date, etc.). 4. Member clicks Save. 5. System validates and saves changes, displays “Update successful.” | | |
| **Alternative Flows:** | 3a.Invalid input (e.g., unsupported image size, date format) → display error, highlight field. | | |
| **Exceptions:** | Database error → display “Unable to save profile, please try again.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-3: Avatar file size must not exceed 2 MB.** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.4: Apply personalized filter**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_04 – Manage Profile | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Member | **Secondary Actors:** | **N/A** |
| **Description:** | Member can filter based on their own favorites | | |
| **Pre-conditions:** | PRE-1: Member is logged in. | | |
| **Post–conditions:** | POST-1: Updated profile is persisted. | | |
| **Normal Flow:** | 1. Member navigates to homepage. 2. The system displays numerous snack food categories 3. Member click on filter icon 4. Member choose what they like. 5. System validates and show available foods | | |
| **Alternative Flows:** | 3a.Invalid input (e.g., unsupported image size, date format) → display error, highlight field. | | |
| **Exceptions:** | Database error → display “Unable to get favorites food” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-3: Avatar file size must not exceed 2 MB.** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.5: Search Snack Places**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_05 – Search Snack Places | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | User | **Secondary Actors:** | **N/A** |
| **Description:** | The member can search for snack places by keyword or category and apply personalized filters based on their saved preferences (e.g., favorite snack types, dietary restrictions, or location proximity) to view relevant snack place results. | | |
| **Pre-conditions:** | PRE-1: Member is logged in. PRE-2: Member has saved preferences or favorites in their profile (optional for basic search).  PRE-3: The system has access to a database of snack places with associated categories and keywords. | | |
| **Post–conditions:** | POST-1: The system displays a list of snack places matching the search criteria and applied filters.  POST-2: If personalized filters are applied, the results reflect the member’s preferences (e.g., favorite snack categories). | | |
| **Normal Flow:** | 1. The member navigates to the homepage or search page of the snack place application. 2. The system displays a search bar and available snack food categories (e.g., savory, sweet, healthy, gluten-free). 3. The member enters a keyword (e.g., “pizza,” “ice cream”) or selects a category (e.g., “desserts,” “street food”). 4. The member clicks the filter icon to apply personalized filters. 5. The system retrieves the member’s saved preferences (e.g., favorite snack types, dietary restrictions) from their profile. 6. The member selects additional filters (e.g., distance, rating, price range) or relies on saved favorites. 7. The system validates the input and filters, then queries the database for snack places matching the keyword/category and filters. 8. The system displays a list of available snack places with details (e.g., name, location, rating, category). 9. The member can interact with the results (e.g., view details, save to favorites, or get directions). | | |
| **Alternative Flows:** | 3a. If the member enters an invalid keyword (e.g., special characters not supported or no matching results):   * The system displays an error message: “No snack places found for your search. Please try another keyword or category.” * The member is prompted to revise the search input or select a category.   6a. If the member has no saved preferences:   * The system prompts the member to use default filters (e.g., all categories, no dietary restrictions) or save preferences in their profile. * The flow continues from step 7. | | |
| **Exceptions:** | Database error during search or filter application → display “Unable to retrieve snack places. Please try again later.” | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High** | | |
| **Business Rules:** | **BR-1: Search keywords must be alphanumeric and support common snack-related terms.**  **BR-2: Filters must include at least one of the following: category, location, dietary preferences, or user favorites.**  **BR-3: Results must be sorted by relevance (e.g., keyword match, proximity, or user preferences) by default.** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.6: View Merchant Info**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_06 – View Merchant Info | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Member | **Secondary Actors:** | **N/A** |
| **Description:** | The member can view detailed information about a selected snack place (merchant), including the menu, customer ratings, and shop overview (e.g., address, hours, contact details), to make informed decisions about visiting or ordering. | | |
| **Pre-conditions:** | PRE-1: Member is logged in.  PRE-2: Member has performed a search (via UC\_05) or navigated to a merchant’s profile from a list, favorites, or direct link.  PRE-3: The system has access to a database containing merchant information, including menus, ratings, and shop details. | | |
| **Post–conditions:** | POST-1: The system displays the merchant’s menu, ratings, and shop overview.  POST-2: The member can interact with the displayed information (e.g., browse the menu, view ratings, save the merchant to favorites).  POST-3: The member’s view history is optionally saved for personalized recommendations (if applicable). | | |
| **Normal Flow:** | 1. The member selects a snack place (merchant) from search results, a favorites list, or a direct link on the application. 2. The system displays the merchant’s profile page with the menu, ratings, and shop overview in an organized format 3. The member browses the menu, views ratings (e.g., star rating, customer reviews), and checks the shop overview. | | |
| **Alternative Flows:** | 2a. If the merchant’s information is incomplete (e.g., missing menu or ratings):   * The system displays available information and a message: “Some details are unavailable. Please check back later.” * The member can still interact with available data (e.g., view shop overview).   2b. If the merchant is no longer available (e.g., closed or removed from the database):   * The system displays an error: “This merchant is no longer available.” * The member is redirected to the search page or homepage. | | |
| **Exceptions:** | Database error during retrieval of merchant info → display “Unable to load merchant information. Please try again later.” | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High** | | |
| **Business Rules:** | **BR-1: Merchant information must include at least one of the following: menu, ratings, or shop overview.**  **BR-2: Ratings must be displayed as an average (e.g., 4.5/5 stars) based on customer reviews, if available.**  **BR-3: Menu items must include names and prices; descriptions and images are optional.** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.7: Rate and Comment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_04 – Manage Profile | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | User | **Secondary Actors:** | **N/A** |
| **Description:** | Users can leave ratings (e.g., stars) and write comments on snack places they have visited or ordered from. | | |
| **Pre-conditions:** | PRE-1: User is logged in. PRE-2: User has previously interacted with the shop (e.g., ordered or visited). | | |
| **Post–conditions:** | POST-1: Review is saved and visible to other users.  POST-2: Shop rating is recalculated. | | |
| **Normal Flow:** | 1. User opens a shop’s detail page. 2. User selects a star rating and writes a comment. 3. User submits the feedback. 4. System validates and saves the review. 5. Review becomes publicly visible. | | |
| **Alternative Flows:** | 2a. User leaves empty comment or rating → system prevents submission.  4a. Duplicate review detected → show message: “You’ve already reviewed this shop.” | | |
| **Exceptions:** | Database error → display “Unable to save review.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-24: Rating must be from 1 to 5 stars.**  **BR-25: Comments must be under 500 characters.** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.8: View Own Reviews**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_04 – Manage Profile | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | User | **Secondary Actors:** | **N/A** |
| **Description:** | Users can view a history of all reviews they have submitted to various snack places. | | |
| **Pre-conditions:** | PRE-1: User is logged in. | | |
| **Post–conditions:** | POST-1: User sees a list of all reviews they have submitted. | | |
| **Normal Flow:** | 1. User opens their profile. 2. User navigates to the “My Reviews” tab. 3. System displays all user’s reviews with shop name, date, and content. 4. User can edit or delete reviews if permitted. | | |
| **Alternative Flows:** |  | | |
| **Exceptions:** | Server error → show message: “Unable to load your reviews.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-26: Only the author of a review can edit or delete it. BR-27: Deleted reviews are not shown to the public.** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.9: Use GPS Map**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_04 – Manage Profile | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | User | **Secondary Actors:** | **N/A** |
| **Description:** | User can view snack places on a map interface with real-time GPS location. | | |
| **Pre-conditions:** | PRE-1: User has allowed GPS access.  PRE-2: Device supports geolocation. | | |
| **Post–conditions:** | POST-1: Nearby snack shops are displayed relative to user location. | | |
| **Normal Flow:** | 1. User opens map view. 2. System requests location access. 3. Once granted, user sees snack shop pins on map. 4. User taps on a shop to view more details or navigate. | | |
| **Alternative Flows:** | 2a. User denies location access → show fallback “Enter location manually.”  3a. No shops nearby → display “No snack places found in your area.” | | |
| **Exceptions:** | GPS failure or timeout → show “Unable to get your location.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-28: Location must auto-refresh every 5 minutes or on user request.** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.10: Chat with AI Chatbot**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_04 – Manage Profile | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Merchant | **Secondary Actors:** | **N/A** |
| **Description:** | Merchant interacts with an AI chatbot to receive snack recommendations or support generating image and logo. | | |
| **Pre-conditions:** | PRE-1: Merchant is logged in (if using personal preferences).  PRE-2: Chatbot module is online. | | |
| **Post–conditions:** | POST-1: User receives snack suggestions or answers from chatbot.  POST-2: Merchant buy subscription packages | | |
| **Normal Flow:** | 1. Merchant opens the chatbot interface. 2. Merchant types a message 3. Chatbot processes input and responds with image suggestion 4. The merchant may follow up or end conversation. | | |
| **Alternative Flows:** | 3a. Chatbot doesn’t understand → respond with “Sorry, I didn’t get that. Try rephrasing.”  4a. Merchant asks for unavailable feature → show fallback/help message. | | |
| **Exceptions:** | Chatbot service unavailable → display “AI assistant is currently offline.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-29: AI chatbot must not provide offensive or unsafe recommendations.**  **BR-30: All conversations are logged for feedback/training purposes (if permitted).** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.11: Purchase Premium Package**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_11 – Purchase Premium Package | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Merchant | **Secondary Actors:** | **N/A** |
| **Description:** | The merchant can purchase a premium subscription package to access enhanced features, such as promoted listings, advanced analytics, or additional customization options for their snack place profile. | | |
| **Pre-conditions:** | PRE-1: Merchant is logged in to their account. PRE-2: The system has a configured list of premium subscription packages with defined features and pricing. PRE-3: The merchant has a valid payment method (e.g., credit card, digital wallet) supported by the payment processor. | | |
| **Post–conditions:** | POST-1: The merchant’s subscription is upgraded to the selected premium package. POST-2: The payment is successfully processed and recorded. POST-3: The merchant gains access to premium features immediately. POST-4: The merchant receives a confirmation of the purchase (e.g., via email or in-app notification). | | |
| **Normal Flow:** | 1. The merchant navigates to the subscription or account management section of the application. 2. The system displays available premium subscription packages, including features, pricing, and duration (e.g., monthly, yearly). 3. The merchant selects a premium package (e.g., “Premium Plus” for $19.99/month). 4. The system prompts the merchant to enter or confirm payment details (e.g., credit card, PayPal). 5. The merchant submits the payment information. 6. The system validates the payment details and sends a request to the payment processor. 7. The payment processor confirms the transaction is successful. | | |
| **Alternative Flows:** | 3a. If the merchant already has an active premium subscription:   * The system displays the current subscription details and options to upgrade, downgrade, or renew. * The merchant selects an option, and the flow continues from step 4.   5a. If the payment details are invalid (e.g., incorrect card number, insufficient funds):   * The system displays an error message: “Invalid payment details. Please check your information and try again.” * The merchant is prompted to revise the payment details and resubmit. * The flow returns to step 5. | | |
| **Exceptions:** | Database error during subscription update → display “Unable to update subscription. Please try again later. | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Low to Medium** | | |
| **Business Rules:** | **BR-1: Only verified merchants with a registered snack place can purchase a premium package. BR-2: Payment must be processed through a secure, supported payment processor. BR-3: Premium features are activated immediately upon successful payment.** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.12: Manage Menu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_12 – Manage Menu | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Merchant | **Secondary Actors:** | **N/A** |
| **Description:** | The merchant can add, edit, or delete menu items for their snack place, including details such as item name, price, description, and optional attributes like dietary tags or images, to keep the menu up-to-date and appealing to customers. | | |
| **Pre-conditions:** | PRE-1: Merchant is logged in to their account. PRE-2: Merchant’s account is verified and associated with a registered snack place. PRE-3: The system has access to a database containing the merchant’s snack place and menu items. | | |
| **Post–conditions:** | POST-1: The merchant’s menu is updated in the database with the added, edited, or deleted items. POST-2: Changes are reflected in the snack place’s profile, visible to users (e.g., via UC\_06 – View Merchant Info). POST-3: The merchant receives confirmation of successful menu updates via in-app notification or UI feedback. | | |
| **Normal Flow:** | 1. The merchant navigates to the menu management section of the application (e.g., via a “Manage Menu” option in their account or snack place dashboard). 2. The system displays the current menu, including a list of existing items (e.g., name, price, description, dietary tags, images) and options to add, edit, or delete items. 3. **Add Item:**    1. The merchant selects the “Add Item” option.    2. The system presents a form to enter item details (e.g., name, price, description, dietary tags, optional image upload).    3. The merchant fills out the form and submits it.    4. The system validates the input (e.g., name is non-empty, price is a valid number, image size ≤ 2 MB).    5. The system saves the new item to the database and updates the menu display. 4. **Edit Item:**    1. The merchant selects an existing item and chooses the “Edit” option.    2. The system displays a pre-filled form with the item’s current details.    3. The merchant updates the details (e.g., changes price, adds an image) and submits the form.    4. The system validates the input and updates the item in the database. 5. **Delete Item:**    1. The merchant selects an existing item and chooses the “Delete” option.    2. The system prompts for confirmation (e.g., “Are you sure you want to delete this item?”).    3. The merchant confirms the deletion.    4. The system removes the item from the database and updates the menu display. 6. The system confirms the action (add, edit, or delete) with a success message (e.g., “Menu updated successfully”). 7. The updated menu is reflected in the snack place’s profile for users to view. | | |
| **Alternative Flows:** | 5a. If the merchant cancels the deletion after the confirmation prompt:   * The system returns the merchant to the menu management page without making changes.   6a. If the database fails to save changes (e.g., due to a server error):   * The system displays an error message: “Unable to update menu. Please try again later.” * The merchant is returned to the menu management page to retry. | | |
| **Exceptions:** | Database error during add/edit/delete operations → display “Unable to update menu. Please try again later.” | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **Medium to High** | | |
| **Business Rules:** | **BR-1: Only verified merchants with a registered snack place can manage menu items. BR-2: Menu items must include at least a name and price; description, dietary tags, and images are optional. BR-3: Image uploads must not exceed 2 MB and must be in supported formats (e.g., JPEG, PNG).** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.13: Respond to Reviews**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_13 – Respond to Reviews | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Merchant | **Secondary Actors:** | **N/A** |
| **Description:** | The merchant can view customer reviews for their snack place and submit responses to address feedback, thank customers, or resolve concerns, enhancing customer engagement and improving the snack place’s reputation. | | |
| **Pre-conditions:** | PRE-1: Merchant is logged in to their account. PRE-2: Merchant’s account is verified and associated with a registered snack place. PRE-3: The system has access to a database containing reviews for the merchant’s snack place (e.g., Reviews table with fields like reviewId, userId, snackPlaceId, rating, comment). PRE-4: At least one customer review exists for the merchant’s snack place. | | |
| **Post–conditions:** | POST-1: The merchant’s response is saved in the database and linked to the corresponding review. POST-2: The response is visible in the snack place’s profile (e.g., via UC\_06 – View Merchant Info) for users to see. POST-3: The merchant receives confirmation of the successful response submission via in-app notification or UI feedback. | | |
| **Normal Flow:** | 1. The merchant navigates to the review management section of the application (e.g., via a “Reviews” tab in their account or snack place dashboard). 2. The system displays a list of customer reviews for the merchant’s snack place, including details like rating, comment, username, and date. 3. The merchant selects a review to respond to. 4. The system presents a response form or text input field for the merchant to enter their reply. 5. The merchant enters a response (e.g., “Thank you for your feedback! We’re glad you enjoyed our snacks.”) and submits it. | | |
| **Alternative Flows:** | 6a. If the database fails to save the response (e.g., due to a server error):   * The system displays an error message: “Unable to submit response. Please try again later.” * The merchant is returned to the review management page to retry.   10a. If user notifications are not configured or the user’s contact details are unavailable:   * The system skips the notification step and logs the issue for administrative review. * The flow completes with the response saved and displayed. | | |
| **Exceptions:** | Database error during response submission → display “Unable to submit response. Please try again later.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-1: Only verified merchants with a registered snack place can respond to reviews for their business. BR-2: Responses must be non-empty, within a 500-character limit, and comply with content guidelines (e.g., no offensive language). BR-3: Each review can have only one merchant response, unless the system allows editing or deleting responses.** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.14: View Analytics**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_14 – View Analytics | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Merchant | **Secondary Actors:** | **N/A** |
| **Description:** | The merchant can view performance metrics for their snack place, such as page views, customer ratings, review counts, popular menu items, or sales data (if applicable), to assess business performance and make informed decisions. Access to advanced analytics may require a premium subscription. | | |
| **Pre-conditions:** | PRE-1: Merchant is logged in to their account. PRE-2: Merchant’s account is verified and associated with a registered snack place. PRE-3: The system has access to a database containing performance data for the merchant’s | | |
| **Post–conditions:** | POST-1: The merchant views a dashboard or report displaying relevant performance metrics for their snack place. POST-2: The system logs the merchant’s interaction with the analytics dashboard for audit purposes (if applicable). POST-3: The merchant can export or share basic metrics (e.g., as a PDF report), if supported by the system. | | |
| **Normal Flow:** | 1. The merchant navigates to the analytics section of the application (e.g., via an “Analytics” or “Performance” tab in their account or snack place dashboard). 2. The system verifies the merchant’s subscription status to determine access to basic or advanced analytics features. 3. The system retrieves performance metrics from the database, including:    1. Page views (e.g., number of visits to the snack place’s profile).    2. Customer ratings (e.g., average rating from Reviews table).    3. Review counts (e.g., total number of reviews).    4. Popular menu items (e.g., most viewed or ordered items from UC\_12 – Manage Menu).    5. Sales data (e.g., total orders or revenue, if applicable). | | |
| **Alternative Flows:** | 2a. If the merchant does not have a premium subscription and attempts to access advanced analytics:   * The system displays a message: “Advanced analytics require a premium subscription. Upgrade now?” * The merchant is prompted to purchase a premium package (via UC\_11) or view basic metrics only. * The flow continues with basic metrics displayed in step 4.   4a. If no performance data is available (e.g., new snack place with no views or reviews):   * The system displays a message: “No analytics data available yet. Promote your snack place to start collecting metrics.” * The merchant is returned to the dashboard with options to promote their profile (if available). | | |
| **Exceptions:** | * Database error during metrics retrieval → display “Unable to load analytics. Please try again later.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-1: Only verified merchants with a registered snack place can access analytics for their business. BR-2: Basic analytics (e.g., page views, average rating) are available to all merchants; advanced analytics (e.g., sales trends, customer demographics) require a premium subscription. BR-3: Metrics must be accurate and based on data from the past 90 days (unless filtered otherwise).** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.15: Snack Food place management**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_15 – Manage Snack Food Store | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Merchant (Store Owner) | **Secondary Actors:** | **Admin (for approval only)** |
| **Description:** | This use case describes how a merchant can create, update, and manage their snack food store. Once submitted, the store must be approved by the admin before becoming visible or active on the platform. | | |
| **Pre-conditions:** | PRE-1: Member is logged in. | | |
| **Post–conditions:** | PRE-2: Merchant profile is valid and allowed to register a store. | | |
| **Normal Flow:** | 1. Merchant navigates to the “Register Store” or “My Store” section. 2. System displays a form for store details (name, image, location, description, etc. 3. Merchant fills in and submits the form. 4. System stores the data and marks the store as **Pending Approval**. 5. Admin receives notification and reviews the submission separately. | | |
| **Alternative Flows:** | **3a. Invalid input (e.g., missing name, image too large):**  → Display error messages, highlight invalid fields. **5a. Store already exists:**  → Notify merchant: “You have already registered a store.” | | |
| **Exceptions:** | → Display message: “Unable to save store information. Please try again later.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-3: Avatar file size must not exceed 2 MB.** | | |
| **Other Information:** | N/A | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.16: AI Support for Image and Logo Creation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_16 – AI Support for Image and Logo Creation | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Merchant | **Secondary Actors:** | **AI Service Provider (System Component)** |
| **Description:** | This use case describes how a merchant with an active package subscription can use AI tools to generate images and logos for their snack food store or profile. | | |
| **Pre-conditions:** | PRE-1: Member is logged in.  PRE-2: Merchant has an **active subscription package** that includes AI support features. | | |
| **Post–conditions:** | POST-1: Generated image or logo is saved to the merchant’s profile or store settings.  POST-2: Merchant can download or apply the image/logo directly. | | |
| **Normal Flow:** | 1. Merchant navigates to the **AI Tools** section. 2. System checks merchant’s subscription package. 3. System displays the image/logo creation interface. 4. Merchant enters prompt or selects design preferences (e.g., style, color, concept). 5. System uses AI to generate one or more image/logo suggestions. 6. Merchant previews the results. 7. Merchant selects one to save or download. | | |
| **Alternative Flows:** | **2a. Subscription does not include AI tools:**  → System displays message: “AI tools are only available for premium subscribers.”  **5a. AI fails to generate result (timeout or invalid prompt):**  → System displays error and suggests prompt revision. | | |
| **Exceptions:** | → Display message: “Unable to generate image/logo at this time. Please try again later.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Low to Medium (depends on marketing cycles or store setup phase)** | | |
| **Business Rules:** | **BR-10: Only merchants with an active package that includes AI support can access this feature.**  **BR-11: AI-generated images must be under 5MB and in .png or .jpg format.**  **BR-12: Max 3 AI generations per day (configurable based on package level).** | | |
| **Other Information:** | AI support is handled via third-party APIs and may require tokens or usage credits. | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.17: Dashboard View for Merchant**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_17 – Dashboard View for Merchant | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Merchant | **Secondary Actors:** | **System (for analytics and data aggregation)** |
| **Description:** | This use case allows a merchant to view a personalized dashboard displaying store performance metrics, including rating, number of reviews, access statistics, suggestion rates, and daily click data. | | |
| **Pre-conditions:** | PRE-1: Merchant is logged in. PRE-2: Merchant has at least one active store registered. | | |
| **Post–conditions:** | POST-1: Dashboard is successfully loaded and displays updated data metrics. POST-2: Merchant can navigate to specific metrics for deeper analysis (if available). | | |
| **Normal Flow:** | 1. Merchant opens the app and navigates to the **Dashboard (Số liệu)** tab. 2. System fetches real-time metrics including: 3. Star rating (e.g., 4.6/5) 4. Total reviews (e.g., 2) 5. Suggestion rate (e.g., 56%) 6. Access count (e.g., 4 visits) 7. Clicks per day (bar chart) 8. Dashboard displays the overview in visual cards and simple graphs. 9. Merchant reviews the performance metrics and may adjust their store strategy accordingly. | | |
| **Alternative Flows:** | **2a. No data available (e.g., new merchant):**  → System displays placeholders or a message like “No data available yet.” | | |
| **Exceptions:** | → Display message: “Unable to load dashboard data. Please check your connection.” | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High – expected daily or weekly.** | | |
| **Business Rules:** | **BR-20: Data must be updated in real-time or near-real-time.**  **BR-21: Only metrics related to the merchant’s store(s) are shown.**  **BR-22: Click data should be grouped by day for accurate trend analysis.** | | |
| **Other Information:** | Analytics data may be powered by a third-party service or internal event tracking system. | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.18: Change Password**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_18 – Dashboard View for Merchant | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Merchant / User | **Secondary Actors:** | **System (Authentication Module)** |
| **Description:** | This use case describes how a merchant or user can manage their account privacy by changing their password in the settings section, using their current password for verification. | | |
| **Pre-conditions:** | PRE-1: User is logged in.  PRE-2: User has access to their current password. | | |
| **Post–conditions:** | POST-1: User’s password is successfully updated.  POST-2: System logs the user out and requests re-login with the new password (optional, if enforced by policy). | | |
| **Normal Flow:** | User navigates to **Settings > Change Password** section.  System displays form with three fields:   * Current password * New password * Confirm new password   User fills all fields and submits.  System verifies that the current password is correct.  System checks new password validity (e.g., minimum length, complexity).  System updates the password in the database.  System shows success message and (optionally) logs the user out. | | |
| **Alternative Flows:** | **4a. Current password incorrect:  → System displays error: “Incorrect current password.”**  **5a. New password too weak or doesn’t match confirmation:  → System shows validation error: “Password must be at least 8 characters and match confirmation.”** | | |
| **Exceptions:** | → Show message: “Unable to update password. Please try again later.” | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **Low to Medium** | | |
| **Business Rules:** | **BR-30: New password must be at least 8 characters, include letters and numbers.**  **BR-31: New password must not match the previous password. BR-32: Password change should be confirmed via success message.** | | |
| **Other Information:** | For additional security, password changes may trigger an email or SMS confirmation. | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.19: Observation for Transaction History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_19 – Observation for Transaction History | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Merchant | **Secondary Actors:** | **System (Database / Transaction Log Service)** |
| **Description:** | This use case allows a merchant to view and search through their transaction history using filters such as transaction ID or other criteria (e.g., date, amount). | | |
| **Pre-conditions:** | PRE-1: Merchant is logged in. PRE-2: Merchant has at least one completed transaction recorded in the system. | | |
| **Post–conditions:** | POST-1: Transaction history is displayed with correct information.  POST-2: Merchant can search or filter specific transactions as needed. | | |
| **Normal Flow:** | 1. Merchant navigates to **Transaction History** section. 2. System displays a list of recent transactions (paginated or grouped by date). 3. Merchant enters a transaction ID or filter criteria in the search bar. 4. System fetches and displays matching transaction(s) with details such as: 5. Transaction ID 6. Amount 7. Date & Time 8. Payment method 9. Status (e.g., Completed, Failed) 10. Merchant reviews or exports the data if needed. | | |
| **Alternative Flows:** | **3a. No match found for entered ID:  → Display: “No transactions found with the given ID.” 2a. No transactions yet:  → Display placeholder message: “No transaction history available.”** | | |
| **Exceptions:** | → Show message: “Unable to retrieve transaction data. Please try again later.” | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **Low to Medium** | | |
| **Business Rules:** | **BR-40: Only transactions linked to the logged-in merchant should be displayed.**  **BR-41: Search by ID must be exact match.**  **BR-42: Results should be sorted by most recent by default.** | | |
| **Other Information:** | Advanced filters (by date range, payment type) may be available in future versions. | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.20: Manage Taste Preferences**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_20 – Manage Taste Preferences | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Admin | **Secondary Actors:** | **System (Database / Transaction Log Service)** |
| **Description:** | The admin can perform CRUD (Create, Read, Update, Delete) operations on taste preference information (e.g., savory, sweet, spicy, sour) to maintain a list of selectable options for users to filter snack places or personalize their profiles. This ensures users can select taste preferences when searching for snack places or setting up their profiles. | | |
| **Pre-conditions:** | PRE-1: Admin is logged in to their account with administrative privileges. PRE-2: The system has access to a database containing taste preference data (e.g., FoodTypes or Diets table with fields like tasteId, name, description). PRE-3: The admin has access to the taste preference management interface (e.g., via an admin dashboard). | | |
| **Post–conditions:** | POST-1: The taste preference data is updated in the database with any created, updated, or deleted entries. POST-2: Changes to taste preferences are reflected in user-facing features (e.g., UC\_05 – Search Snack Places, UC\_04 – Manage Profile). POST-3: The admin receives confirmation of successful CRUD operations via in-app notification or UI feedback. | | |
| **Normal Flow:** | The admin navigates to the taste preference management section of the application (e.g., via a “Manage Taste Preferences” option in the admin dashboard).   1. The system displays a list of existing taste preferences (e.g., name, description, associated snack places) with options to create, update, or delete entries. 2. **Create Taste Preference:**    * The admin selects the “Add Taste Preference” option.    * The system presents a form to enter details (e.g., name, description, optional image).    * The admin fills out the form (e.g., Name: “Spicy,” Description: “Foods with bold, fiery flavors”) and submits it.    * The system validates the input (e.g., name is non-empty, unique, image size ≤ 2 MB).    * The system saves the new taste preference to the database and updates the list. | | |
| **Alternative Flows:** | **6b. If the admin cancels the deletion after the confirmation prompt:**   * **The system returns the admin to the taste preference management page without making changes.**   **7a. If the database fails to save changes or retrieve data (e.g., due to a server error):**   * **The system displays an error message: “Unable to update or load taste preferences. Please try again later.”** * **The admin is returned to the taste preference management page to retry.** | | |
| **Exceptions:** | Database error during CRUD operations → display “Unable to update or load taste preferences. Please try again later.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Low to Medium** | | |
| **Business Rules:** | **BR-1: Only users with admin privileges can perform CRUD operations on taste preferences. BR-2: Taste preferences must include a unique name; descriptions and images are optional. BR-3: Image uploads must not exceed 2 MB and must be in supported formats (e.g., JPEG, PNG).** | | |
| **Other Information:** | Admin has appropriate privileges.  System is online and functioning normally. | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.21: Manage Food Categories**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_21 – Manage Food Categories | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Admin | **Secondary Actors:** | **System (Database / Transaction Log Service)** |
| **Description:** | The admin can perform CRUD (Create, Read, Update, Delete) operations on food categories (e.g., dessert, savory, vegan, street food) to maintain a list of classifications used for organizing snack places or menu items. This ensures users can filter snack places by category and merchants can assign categories to their offerings. | | |
| **Pre-conditions:** | PRE-1: Admin is logged in to their account with administrative privileges. PRE-2: The system has access to a database containing food category data (e.g., FoodTypes table with fields like categoryId, name, description). PRE-3: The admin has access to the food category management interface (e.g., via an admin dashboard). | | |
| **Post–conditions:** | POST-1: The food category data is updated in the database with any created, updated, or deleted entries. POST-2: Changes to food categories are reflected in user-facing features (e.g., UC\_05 – Search Snack Places, UC\_12 – Manage Menu) and merchant-facing features (e.g., menu item categorization). POST-3: The admin receives confirmation of successful CRUD operations via in-app notification or UI feedback. | | |
| **Normal Flow:** | The admin navigates to the food category management section of the application (e.g., via a “Manage Food Categories” option in the admin dashboard).   1. The system displays a list of existing food categories (e.g., name, description, associated snack places or menu items) with options to create, update, or delete entries. 2. **Create Food Category:**    * The admin selects the “Add Food Category” option.    * The system presents a form to enter details (e.g., name, description, optional image).    * The admin fills out the form (e.g., Name: “Vegan,” Description: “Plant-based snacks”) and submits it.    * The system validates the input (e.g., name is non-empty, unique, image size ≤ 2 MB).    * The system saves the new food category to the database and updates the list. | | |
| **Alternative Flows:** | **6b. If the admin cancels the deletion after the confirmation prompt:**   * **The system returns the admin to the food category management page without making changes.**   **7a. If the database fails to save changes or retrieve data (e.g., due to a server error):**   * **The system displays an error message: “Unable to update or load food categories. Please try again later.”** * **The admin is returned to the food category management page to retry.** | | |
| **Exceptions:** | * Database error during CRUD operations → display “Unable to update or load food categories. Please try again later.” * Network connectivity issue → display “Network error. Please check your connection and try again.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Low to Medium** | | |
| **Business Rules:** | **BR-1: Only users with admin privileges can perform CRUD operations on food categories. BR-2: Food categories must include a unique name; descriptions and images are optional. BR-3: Image uploads must not exceed 2 MB and must be in supported formats (e.g., JPEG, PNG).** | | |
| **Other Information:** | The system may support bulk operations (e.g., uploading multiple food categories via CSV) if configured. | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.22: Handle Subscription Packages**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_22 – Handle Subscription Packages | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Admin | **Secondary Actors:** | **System (Database / Transaction Log Service)** |
| **Description:** | Admin can create, update, and manage subscription plans available to users and merchants. This includes setting pricing, duration, and benefits of each package. | | |
| **Pre-conditions:** | PRE-1: Admin is authenticated and logged in. | | |
| **Post–conditions:** | POST-1: Updated subscription plans are saved and available in the system.  POST-2: Users and merchants can see the updated package offerings. | | |
| **Normal Flow:** | 1. Admin logs in to the system. Admin navigates to the subscription management section. 2. Admin selects to create a new package or edit an existing one. 3. Admin fills in or modifies the package details (name, price, duration, features). Admin confirms and submits the form. 4. System validates the input and saves the data. 5. System displays confirmation message. | | |
| **Alternative Flows:** | **5a. Input validation fails (e.g., missing required fields, invalid pricing) → system shows error and highlights the problematic fields.** | | |
| **Exceptions:** | → system shows "Unable to save package. Please try again later." | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-10: Each subscription plan must have a unique name. BR-11: Duration must be a positive integer (in days).**  **BR-12: Price must be ≥ 0** | | |
| **Other Information:** | Admin has appropriate privileges.  System is online and functioning normally. | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.23: Manage Transaction History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_23 – Handle Subscription Packages | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Admin | **Secondary Actors:** | **System (Database / Transaction Log Service)** |
| **Description:** | Admin can view, filter, and manage all users’ transaction history, including purchases, subscriptions, and refunds. | | |
| **Pre-conditions:** | PRE-1: Admin is authenticated and logged in. | | |
| **Post–conditions:** | POST-1: Admin can see a filtered list of transactions.  POST-2: Admin may flag or delete specific entries if needed (depending on system policy). | | |
| **Normal Flow:** | 1. Admin accesses the transaction management panel. 2. System displays a searchable and filterable list of user transactions. 3. Admin enters filters (e.g., date range, user ID, transaction type). System updates the list based on filters. Admin selects a specific transaction to view details. 4. Admin may take permitted actions (e.g., flag, mark as resolved, export data). | | |
| **Alternative Flows:** | **4a. No transactions match the filter → system displays “No results found.” 5a. Transaction not found or corrupted → system displays “Transaction details unavailable.”** | | |
| **Exceptions:** | Database or server error → system displays “Unable to retrieve transactions. Please try again later.” | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-13: Admin cannot delete financial records unless permitted by audit policy.**  **BR-14: Admin actions on transactions must be logged for traceability.** | | |
| **Other Information:** | Admin has appropriate privileges. System is online and functioning normally. | | |
| **Assumptions:** | N/A | | |

##### 

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_24 – Create and Manage Users and Merchants | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Admin | **Secondary Actors:** | **System (Database / Transaction Log Service)** |
| **Description:** | Admin can create, update, suspend, and delete user and merchant accounts. This includes managing roles, viewing profile data, verifying merchant registration, and handling user-related issues. | | |
| **Pre-conditions:** | PRE-1: Admin is authenticated and has appropriate privileges.  PRE-2: System is operational with access to user and merchant databases. | | |
| **Post–conditions:** | POST-1: New users or merchants are registered in the system.  POST-2: Updated user or merchant information is saved.  POST-3: Suspended or removed accounts are updated in real-time. | | |
| **Normal Flow:** | 1. Admin logs into the admin dashboard. 2. Admin navigates to the user/merchant management section. 3. Admin views the list of all users and merchants. Admin selects to add, edit, verify, suspend, or delete an account. 4. Admin submits the action. 5. System processes the request and updates the database. System confirms the successful action with a message. | | |
| **Alternative Flows:** | **4a. Admin attempts to modify a protected system account → system shows permission error.**  **5a. Submitted form contains invalid or incomplete data → system highlights fields and prompts for correction.** | | |
| **Exceptions:** | → system displays: “Unable to complete user management action. Try again later.” | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **High** | | |
| **Business Rules:** | **BR-15: Admin cannot delete root or super-admin accounts.**  **BR-16: Merchant accounts must be verified before gaining full access. BR-17: All actions must be logged for audit purposes.** | | |
| **Other Information:** | Verification emails may be triggered for new merchant registrations. | | |
| **Assumptions:** | Admin has access to full account management tools in the dashboard. | | |

##### **2.4.1.24: Manage Business Model**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_25 – Manage Business Models | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Admin | **Secondary Actors:** | **System (Database / Transaction Log Service)** |
| **Description:** | The admin can perform CRUD (Create, Read, Update, Delete) operations to define and configure business models (e.g., subscription plans, commission rates, promotional packages) that govern merchant participation and platform revenue. This ensures the application supports flexible monetization strategies and merchant offerings. | | |
| **Pre-conditions:** | PRE-1: Admin is logged in to their account with administrative privileges. PRE-2: The system has access to a database containing business model data (e.g., BusinessModels table with fields like modelId, name, type, pricing, features). PRE-3: The admin has access to the business model management interface (e.g., via an admin dashboard). | | |
| **Post–conditions:** | POST-1: The business model data is updated in the database with any created, updated, or deleted entries. POST-2: Changes to business models are reflected in merchant-facing features (e.g., UC\_11 – Purchase Premium Package, UC\_14 – View Analytics) and platform operations. POST-3: The admin receives confirmation of successful CRUD operations via in-app notification or UI feedback. | | |
| **Normal Flow:** | 1. The admin navigates to the business model management section of the application (e.g., via a “Manage Business Models” option in the admin dashboard). 2. The system displays a list of existing business models (e.g., name, type, pricing, features, associated merchants) with options to create, update, or delete entries. 3. **Create Business Model:**    1. The admin selects the “Add Business Model” option.    2. The system presents a form to enter details (e.g., name, type [subscription, commission, promotional], pricing, features, duration).    3. The admin fills out the form (e.g., Name: “Premium Subscription,” Type: “Subscription,” Price: “$19.99/month,” Features: “Advanced analytics, promoted listings”) and submits it.    4. The system validates the input (e.g., name is non-empty, unique, pricing is valid, features are supported).    5. The system saves the new business model to the database and updates the list. | | |
| **Alternative Flows:** | **3a. If the admin submits invalid input for creating/updating a business model (e.g., empty name, duplicate name, invalid pricing):**   * **The system displays an error message: “Invalid input. Please check the highlighted fields and try again.”** * **The system highlights the invalid fields (e.g., empty name, negative price).** * **The admin corrects the input and resubmits.** * **The flow returns to step 3 (create) or 5 (update).**   **6a. If the business model is associated with active merchants and cannot be deleted without reassignment:**   * **The system displays a message: “This business model is in use by active merchants. Please reassign or migrate merchants before deleting.”** * **The admin resolves the dependencies (e.g., reassigns to another model) and retries.** * **The flow returns to step 6.** | | |
| **Exceptions:** | Database error during CRUD operations → display “Unable to update or load business models. Please try again later.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Low** | | |
| **Business Rules:** | **BR-1: Only users with admin privileges can perform CRUD operations on business models. BR-2: Business models must include a unique name and type (e.g., subscription, commission, promotional); pricing and features are required for certain types. BR-3: Deleting a business model requires resolving dependencies (e.g., reassigning associated merchants).** | | |
| **Other Information:** | * The system may support previewing business model changes before saving, if configured. * Business models may include subscription plans (e.g., UC\_11), commission rates per order, or promotional packages for merchants. | | |
| **Assumptions:** | N/A | | |

##### **2.4.1.25: Admin Dashboard Overview**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_26 – Create and Manage Users and Merchants | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Admin | **Secondary Actors:** | **System (Database / Transaction Log Service)** |
| **Description:** | Admin accesses a centralized dashboard that displays system-wide statistics and management tools, including user growth, merchant activity, sales analytics, subscription reports, and platform performance. | | |
| **Pre-conditions:** | PRE-1: Admin is logged in with full privileges.  PRE-2: System has collected enough data to display meaningful statistics. | | |
| **Post–conditions:** | POST-1: Admin can view summarized and real-time metrics. POST-2: Admin may navigate to related management modules (e.g., users, merchants, transactions). | | |
| **Normal Flow:** | 1. Admin logs into the system. 2. Admin opens the dashboard module from the main menu. 3. System loads and displays key metrics and charts: 4. Total users and merchants 5. Subscription statistics 6. Transaction volumes 7. User engagement data 8. Admin reviews the dashboard and clicks any section for more detailed data. 9. System redirects to the respective detailed management area if needed. | | |
| **Alternative Flows:** | **3a. No data available for a certain chart → show “No data available” message.**  **4a. User clicks on disabled or restricted modules → show access denied.** | | |
| **Exceptions:** | → system shows fallback message: “Unable to load analytics. Please try again later” | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **Very High** | | |
| **Business Rules:** | **BR-18: Dashboard data must be updated in near real-time or at scheduled intervals.**  **BR-19: All charts must have tooltips for clarity. BR-20: Sensitive information must be hidden or obfuscated depending on admin level.** | | |
| **Other Information:** | May include export options (PDF, CSV) for reporting purposes. | | |
| **Assumptions:** |  | | |

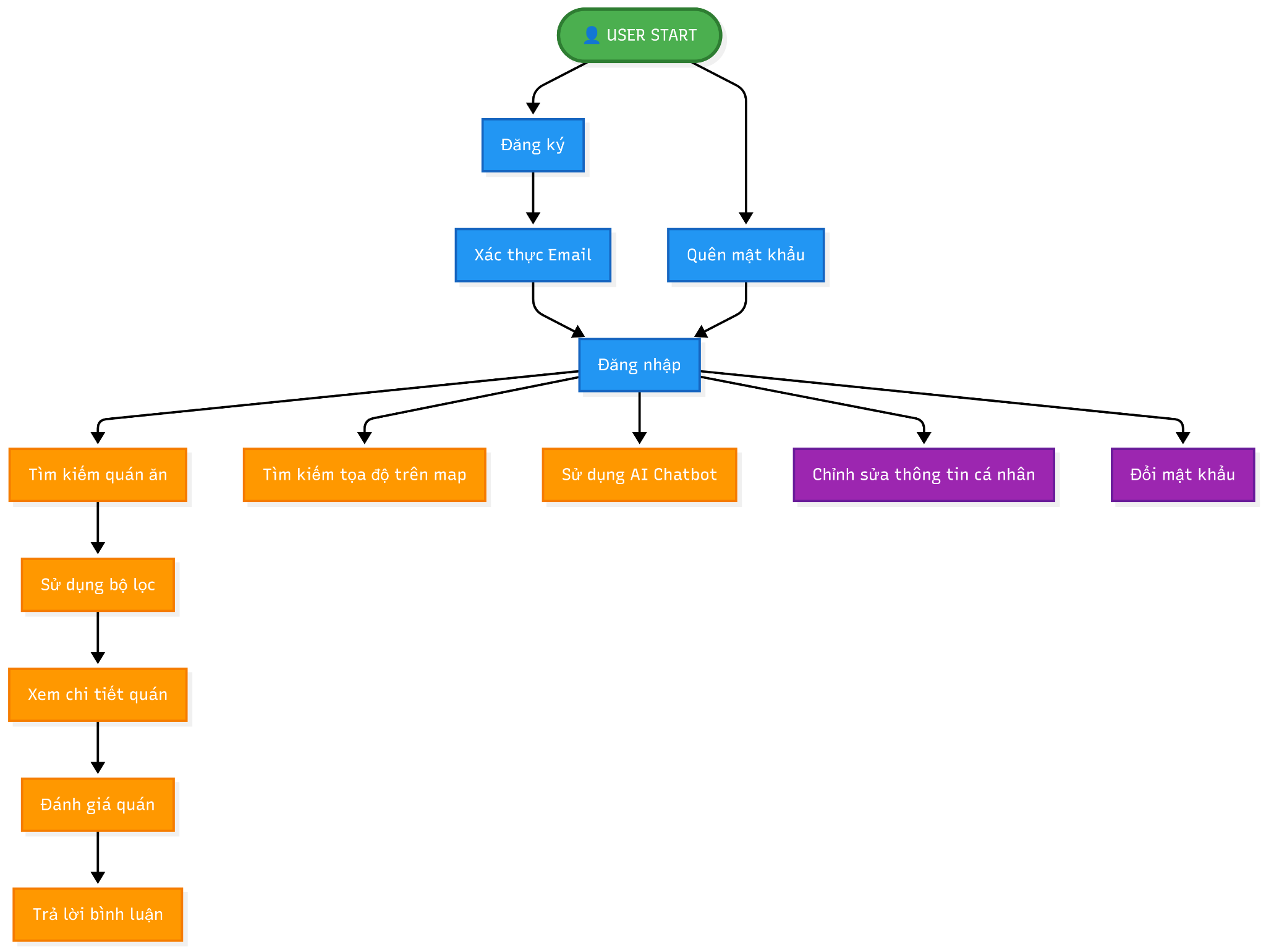
##### **2.4.1.26:Review Detailed Information of Snack Food Stores**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_27 – Review Detailed Information of Snack Food Stores | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Admin | **Secondary Actors:** | **System (Database / Transaction Log Service)** |
| **Description:** | The admin can view detailed information about snack food stores (vendors), including store profiles, menus, customer reviews, ratings, activity metrics (e.g., page view), and subscription status, to monitor vendor performance, ensure compliance, or support moderation tasks. | | |
| **Pre-conditions:** | PRE-1: Admin is logged in to their account with administrative privileges. PRE-2: The system has access to a database containing snack food store data (e.g., SnackPlaces, Menus, Reviews, and BusinessModels tables). PRE-3: At least one snack food store is registered in the system. PRE-4: The admin has access to the store review interface (e.g., via an admin dashboard). | | |
| **Post–conditions:** | POST-1: The admin views a comprehensive report or dashboard with detailed information about the selected snack food store. POST-2: The system logs the admin’s interaction with the store data for audit purposes (if applicable). POST-3: The admin can take follow-up actions (e.g., contact merchant, flag store for review) if supported by the system. | | |
| **Normal Flow:** | 1. The admin navigates to the store review section of the application (e.g., via a “Review Snack Food Stores” option in the admin dashboard). 2. The system displays a list of registered snack food stores with basic details (e.g., store name, ID, status, location). 3. The admin selects a specific store to view its detailed information. | | |
| **Alternative Flows:** | **3a. If no stores are registered in the system:**   * **The system displays a message: “No snack food stores available for review.”** * **The admin is returned to the admin dashboard.**   **4a. If certain data is unavailable for the selected store (e.g., no reviews, no orders):**   * **The system displays available information and a message: “Some details are unavailable for this store.”** * **The admin can continue browsing available data.**   **6a. If the admin selects an invalid filter or sort option (e.g., future date range):**   * **The system displays an error message: “Invalid filter or sort selection. Please choose a valid option.”** * **The admin corrects the selection and resubmits.** * **The flow returns to step 6.** | | |
| **Exceptions:** | Database error during data retrieval → display “Unable to load store details. Please try again later.” | | |
| **Priority:** | **Medium** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-1: Only users with admin privileges can access detailed snack food store information. BR-2: Store details must include at least the store name and status; other data (e.g., menu, reviews) is optional based on availability. BR-3: Activity metrics must be accurate and based on data from the past 90 days (unless filtered otherwise).** | | |
| **Other Information:** | * The system may support exporting store details as a report (e.g., PDF) for documentation. * The admin may filter or search stores by criteria like name, location, or subscription status. * The system may highlight stores with compliance issues (e.g., low ratings, unverified status). | | |
| **Assumptions:** | * The admin has a device with internet access to use the application. * The database includes tables like SnackPlaces (for store profiles), Menus (for menu items), Reviews (for ratings and comments), and BusinessModels (for subscription status). | | |

##### **2.4.1.27:Manage Snack Food Places**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Specification** | | | |
| **Use Case ID and Name:** | UC\_28 – Manage Snack Food Places | | |
| **Created By:** |  | **Date Created:** |  |
| **Primary Actor:** | Admin | **Secondary Actors:** | **System (Database / Transaction Log Service)** |
| **Description:** | The admin can review pending snack food place registrations, approve or reject them, and provide a reason for rejection if necessary. This ensures only compliant and verified snack food places are active in the marketplace, maintaining platform quality. | | |
| **Pre-conditions:** | PRE-1: Admin is logged in to their account with administrative privileges. PRE-2: The system has access to a database containing snack food place data (e.g., SnackPlaces table with fields like snackPlaceId, name, status, and submitted details). PRE-3: At least one snack food place is in a “pending” status awaiting approval. PRE-4: The admin has access to the snack food place management interface (e.g., via an admin dashboard). | | |
| **Post–conditions:** | POST-1: The snack food place’s status is updated in the database (e.g., “approved,” “rejected,” or remains “pending”). POST-2: If approved, the snack food place is activated in the marketplace and visible in user-facing features (e.g., UC\_05 – Search Snack Places, UC\_06 – View Merchant Info). POST-3: If rejected, the merchant is notified with a reason for rejection (e.g., via email or in-app notification). POST-4: The admin receives confirmation of successful approval or rejection actions via in-app notification or UI feedback. | | |
| **Normal Flow:** | 1. The admin navigates to the snack food place management section of the application (e.g., via a “Manage Snack Food Places” option in the admin dashboard). 2. The system displays a list of snack food places with a “pending” status, including basic details (e.g., store name, merchant ID, submission date). 3. The admin selects a pending snack food place to review its details. 4. The system retrieves and displays comprehensive details from the database (e.g., store name, address, contact info, menu, images, verification documents). 5. The admin reviews the details and chooses an action:    1. **Approve**: The admin selects the “Approve” option.    2. **Reject**: The admin selects the “Reject” option and provides a reason (e.g., “Incomplete verification documents” or “Non-compliant menu”). | | |
| **Alternative Flows:** | **2a. If no snack food places are in a “pending” status:**   * **The system displays a message: “No pending snack food places available for review.”** * **The admin is returned to the admin dashboard.**   **4a. If certain data is unavailable for the selected snack food place (e.g., missing verification documents):**   * **The system displays available information and a message: “Some details are unavailable for this snack food place.”** * **The admin can proceed with approval or rejection based on available data.**   **6a. If the admin attempts to reject without providing a reason:**   * **The system displays an error message: “Please provide a reason for rejection.”** * **The admin enters a reason and resubmits.** * **The flow returns to step 6.** | | |
| **Exceptions:** | Database error during status update → display “Unable to update snack food place status. Please try again later.” | | |
| **Priority:** | **High** | | |
| **Frequency of Use:** | **Medium** | | |
| **Business Rules:** | **BR-1: Only users with admin privileges can approve or reject snack food places. BR-2: Snack food places must have a “pending” status to be eligible for approval or rejection. BR-3: A rejection must include a reason, which is stored in the database and sent to the merchant. BR-4: Approved snack food places are immediately visible in user-facing features (e.g., UC\_05, UC\_06). BR-5: All approval/rejection actions must be logged for auditing purposes.** | | |
| **Other Information:** | * The system may allow admins to filter pending snack food places by criteria like submission date or merchant name. * The system may support re-registration for rejected snack food places (e.g., similar to July 15, 2025, doctor re-registration process). | | |
| **Assumptions:** | * The admin has a device with internet access to use the application. * The SnackPlaces table includes fields like snackPlaceId, name, status (e.g., pending, approved, rejected), and verification details. | | |

# **III. User Workflow Diagram:**

****

# **IV. Merchant Workflow Diagram:**

# **V. Admin Workflow Diagram:**

# **VI. Conclusion**

Măm Map is poised to revolutionize snack food discovery in Vietnam by delivering a comprehensive, user-centric platform that addresses the growing demand for personalized, trustworthy, and engaging snack experiences. By empowering users to find snack places tailored to their tastes, budgets, and dietary needs through smart filters, GPS maps, and AI-driven recommendations, Măm Map creates a seamless and delightful user experience. For merchants, the platform offers robust tools to manage stores, menus, and reviews, while premium features like AI-generated branding and analytics enable them to boost visibility and attract customers. The admin console ensures platform quality through effective management of users, merchants, snack places, and business models, supported by detailed analytics and oversight capabilities. Despite current limitations, such as simulated payments and basic logistics, Măm Map’s Flutter-based mobile app and .NET backend provide a strong foundation for future enhancements, including real payment integration and advanced merchant features. With its focus on community-driven feedback and a vibrant snack food ecosystem, Măm Map is well-positioned to become Vietnam’s leading snack discovery platform, connecting snack enthusiasts, merchants, and businesses in a dynamic and innovative way.