Final Year Project

FYP Report

FYP Team
Mr. Saad Abdur Razzaq
Mr. Muhammad Wasif Anwar
Mr. Zaeem Muhammad Yaseen

Supervised by
Ma’am Tahreem Aslam

Department of Computer Science
National University of Computer and Emerging Science
Chiniot, Pakistan

2023

Anti-Plagiarism Declaration

This is to declare that the above publication was produced under the:
Title: ________________________________
are the sole contribution of the author(s) and no part hereof has been reproduced on **as it is** basis (cut and paste) which can be considered as **Plagiarism?** All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if a violation of this declaration is determined.
Date: ______________

Student 1
Name: Saad Abdur Razzaq

Signature:___________________

Student 2
Name: Muhammad Wasif Anwar

Signature:___________________

Student 3
Name: Zaeem Muhammad Yaseen

Signature:___________________

Supervisor
Name: Ma’am Tahreem Aslam

Signature:___________________

Table of Contents

1. Process Model

1. Our Customer-Centric Approach:

In our food delivery app, we understand that customer preferences and trends in the food industry can change frequently. With Agile, we stay in tune with these changes by continuously gathering customer feedback and adapting our app accordingly.

2. Iterative Development:

Agile's iterative approach is perfect for us to launch our food delivery app. We start with a basic version, allowing users to place orders and provide feedback. With each iteration, we enhance features, add new ones, and improve the overall user experience.

3. Flexibility:

The food industry is subject to external factors such as seasonal variations and changing market trends. Agile's flexibility means we can swiftly adjust our app to accommodate these changes, whether it's updating menus for a new season or responding to customer demands for healthier options.

4. Frequent Releases:

Regular releases are vital for our food delivery app. Agile promotes continuous updates, so we can introduce new features, fix issues, and ensure our app is always running smoothly.

5. Transparency and Collaboration:

Agile methodologies foster open communication among our team. In the context of our food delivery app, this helps us coordinate tasks like order processing, delivery scheduling, and customer support, ensuring efficient operations and quick issue resolution.

6. Reduced Risk:

Launching a basic version of our app early, as encouraged by Agile, allows us to test the market with minimal investment. This reduces the risk of developing a full-featured app that may not meet customer expectations.

7. Prioritization:

Agile's prioritization techniques are valuable when deciding which features to implement. We can focus on functionalities that directly impact the customer experience, such as user-friendly menus, efficient order tracking, and secure payment options.

8. Market Responsiveness:

The food delivery industry is highly competitive, and new trends emerge regularly. Agile enables us to respond swiftly to market changes by incorporating the latest industry trends and customer demands.

9. Quality Assurance:

Continuous testing and quality assurance are critical for our food delivery app to ensure that orders are processed accurately, payments are secure, and our app functions flawlessly.

10. Team Motivation:

Agile empowers our development teams to have a say in what they work on. In the context of our food delivery app, this can lead to the creation of innovative features that streamline the order process and enhance the user experience, keeping the team motivated.

11. Scalability:

As our food delivery app gains popularity, Agile practices help us scale our operations effectively. We can expand the app's capabilities to meet growing demands, add support for more restaurants, and handle increased user traffic.

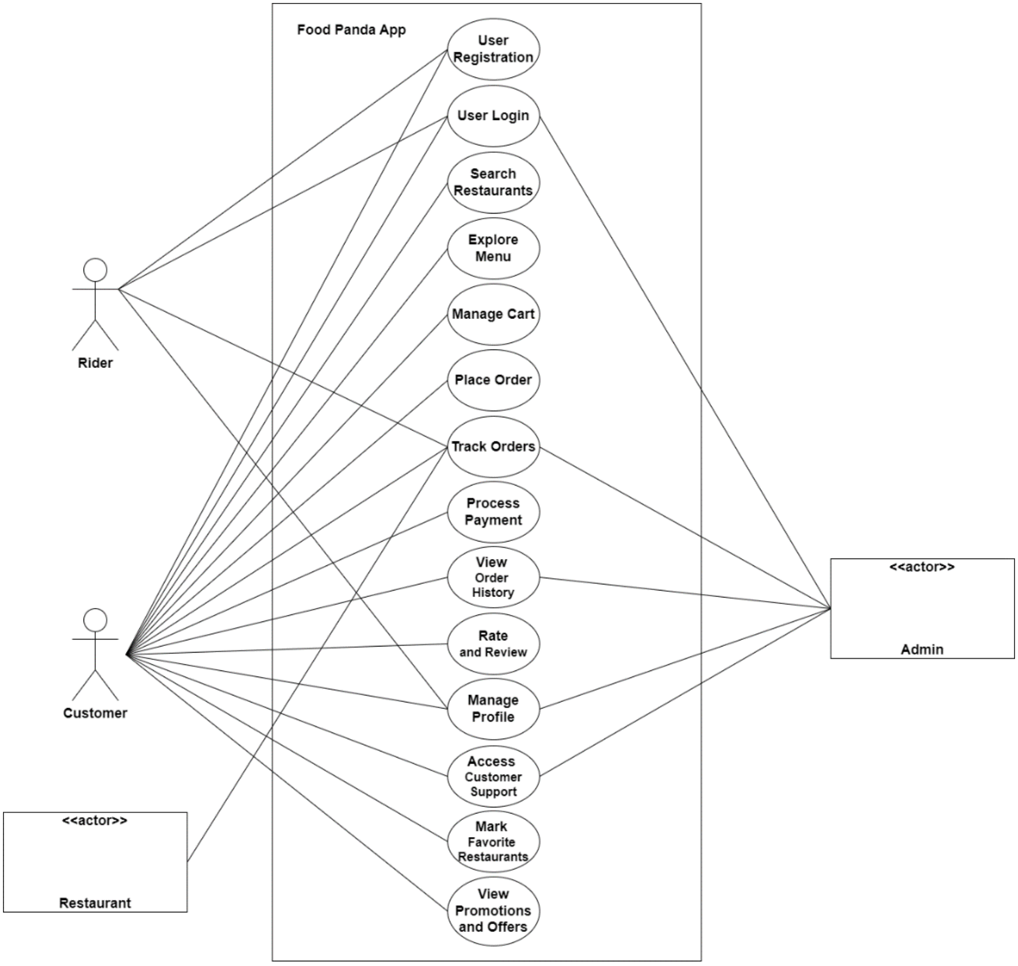
12. Continuous Improvement:

Agile's retrospective meetings allow our teams to reflect on their performance and make improvements. In the food delivery app space, this could involve streamlining delivery routes, enhancing customer support processes, or optimizing app performance to provide a better experience for users.

2. System Modeling

2.1. Krutchen’s 4+1 view model

2.1.1. Use Case Diagram



2.1.2. High Level Use Case

2.1.2.1. User Registration

Use Case Id:	**UC1**
Use case Name:	User Registration
Actor:	Customer, Rider
Type:	Primary
Description:	Users, including Customers and Riders, can register themselves into the system using their email to gain access to the Food Panda application for ordering food and providing delivery services.

2.1.2.2. User Login

Use Case Id:	**UC2**
Use case Name:	User Login
Primary Actor:	Customer, Rider
Secondary Actor:	Admin

Description:	Users, including Customers and Riders, can log in to their accounts in the Food Panda application using their registered email and password, gaining access to personalized features, order management, and food delivery services. Admin can log into his account to manage customers and riders

2.1.2.3. Search Restaurants

Use Case Id:	**UC3**
Use case Name:	Search Restaurants
Actor:	Customer
Type:	Primary
Description:	Customers can search for restaurants in the Food Panda app, browse restaurant options, and view details such as cuisine type, location, ratings, and menu offerings to make informed choices for food orders.

2.1.2.4. Explore Menu

Use Case Id:	**UC4**
Use case Name:	Explore Menu
Actor:	Customer
Type:	Primary
Description:	Customers can explore the menu of a selected restaurant in the Food Panda app, viewing available dishes, their descriptions, prices, and any special offers to make food selections for placing orders.

2.1.2.5. Manage Cart

Use Case Id:	**UC5**
Use case Name:	Manage Cart
Actor:	Customer
Type:	Primary
Description:	Customers can manage their shopping cart in the Food Panda app, allowing them to add, remove, or adjust the quantity of selected food items, review their order, and proceed to the payment process for order placement.

2.1.2.6. Place Order

Use Case Id:	**UC5**
Use case Name:	Manage Cart
Actor:	Customer
Type:	Primary
Description:	Customers can manage their shopping cart in the Food Panda app, allowing them to add, remove, or adjust the quantity of selected food items, review their order,

	and proceed to the payment process for order placement.

2.1.2.7. Track Orders

Use Case Id:	**UC7**
Use case Name:	Track Orders
Primary Actor:	Customer, Rider, Restaurant
Secondary Actor:	Admin
Description:	Users, including Customers and Riders, can track food orders in the Food Panda application, receiving real-time updates on the status of their orders, from confirmation and preparation to delivery. This functionality is essential for both customers awaiting their orders and riders delivering them, ensuring a smooth order tracking process. Restaurants can track orders when being delivered to customer. Admin can track orders to manage any inconvenience at any end.

2.1.2.8. Process Payment

Use Case Id:	**UC7**
Use case Name:	Track Orders
Primary Actor:	Customer, Rider, Restaurant
Secondary Actor:	Admin
Description:	Users, including Customers and Riders, can track food orders in the Food Panda application, receiving real-time updates on the status of their orders, from confirmation and preparation to delivery. This functionality is essential for both customers awaiting their orders and riders delivering them, ensuring a smooth order tracking process. Restaurants can track orders when being delivered to customer. Admin can track orders to manage any inconvenience at any end.

2.1.2.9. View Order History

Use Case Id:	**UC7**
Use case Name:	Track Orders
Primary Actor:	Customer, Rider, Restaurant
Secondary Actor:	Admin
Description:	Users, including Customers and Riders, can track food orders in the Food Panda application, receiving real-time updates on the status of their orders, from confirmation and preparation to delivery. This functionality is essential for both customers awaiting their orders and riders delivering them, ensuring a smooth order tracking process. Restaurants can track orders when being delivered to customer. Admin can track orders to manage any inconvenience at any end.

2.1.2.10. Rate and Reviews

Use Case Id:	**UC10**
Use case Name:	Rate and Review
Actor:	Customer
Type:	Primary
Description:	Customers can rate and review their food ordering experiences in the Food Panda app. After receiving their orders, they can provide ratings and leave detailed reviews for the restaurants they ordered from, contributing to the overall feedback and community ratings.

2.1.2.11. Manage Profile

Use Case Id:	**UC11**
Use case Name:	Manage Profile
Primary Actor:	Customer, Rider
Secondary Actor:	Admin
Description:	Users, including Customers and Riders, can manage their profiles in the Food Panda application, allowing them to update personal information, contact details, delivery addresses, and payment methods. This feature ensures that user profiles are accurate and up-to-date for a seamless experience, both for ordering food and providing delivery services. Admins can manage profiles of particular user to see their activities.

2.1.2.12. Access Customer Support

Use Case Id:	**UC12**
Use case Name:	Access Customer Support
Primary Actor:	Customer
Secondary Actor:	Admin
Description:	Customers can access customer support services in the Food Panda app. In case of inquiries, issues, or assistance requests, customers can contact customer support for help and guidance, ensuring a responsive and supportive service experience. Admins can help in solving queries of customers regarding different issues of food panda app.

2.1.2.13. Mark Favorite Restaurants

Use Case Id:	**UC12**
Use case Name:	Access Customer Support
Primary Actor:	Customer
Secondary Actor:	Admin
Description:	Customers can access customer support services in the Food Panda app. In case of inquiries, issues, or

	assistance requests, customers can contact customer support for help and guidance, ensuring a responsive and supportive service experience. Admins can help in solving queries of customers regarding different issues of food panda app.

2.1.2.14. View Promotions and Offers

Use Case Id:	**UC12**
Use case Name:	Access Customer Support
Primary Actor:	Customer
Secondary Actor:	Admin
Description:	Customers can access customer support services in the Food Panda app. In case of inquiries, issues, or assistance requests, customers can contact customer support for help and guidance, ensuring a responsive and supportive service experience. Admins can help in solving queries of customers regarding different issues of food panda app.

2.1.3. Expanded Use Case

2.1.3.1. User Registration

Use Case ID:	**UC_1**
Use Case Name:	User Registration
Actor:	Customer, Rider
Description:	Users, including Customers and Riders, can register themselves into the system using their email to gain access to the Food Panda application for ordering food and providing delivery services.
Event:	Initiated by the user.
Pre-Condition:	User should enter valid credentials.
Post-Condition:	User should get registered.
Normal Flow:	User enters valid credentials (email, password, contact information).
Alternative Flow:	User registers through Gmail or Facebook.
Exceptions:	Null exceptions during the registration process. Error for wrong credentials.
Includes:	Verification
Special Requirement:	Email, Internet

2.1.3.2. User Login

Use Case ID:	**UC_2**
Use Case Name:	User Login

Actor:	Customer, Rider
Description:	Users, including Customers and Riders, can log in to their accounts in the Food Panda application using their registered email and password, gaining access to personalized features, order management, and food delivery services.
Event:	Initiated by the user.
Pre-Condition:	User should have a registered account.
Post-Condition:	User successfully logs in.
Normal Flow:	User enters registered email and password.
Alternative Flow:	User logins through Gmail or Facebook.
Exceptions:	Error for incorrect credentials.
Includes:	Verification
Special Requirement:	Email, Internet

2.1.3.3. Search Restaurants

Use Case ID:	**UC_3**
Use Case Name:	Search Restaurants
Actor:	Customer, Rider
Description:	Users, including Customers and Riders, can search for restaurants in the Food Panda app, browse restaurant options, and view details such as cuisine type, location, ratings, and menu offerings to make informed choices for food orders.
Event:	Initiated by the user.
Pre-Condition:	User should be logged into the system.
Post-Condition:	User views restaurant details.
Normal Flow:	➢ User enters a search query or views the restaurant list. ➢ User clicks on a restaurant to view details.
Alternative Flow:	Users can choose to use location services to find nearby restaurants.
Exceptions:	Null exceptions if no search results are found.
Includes:	Restaurant Details
Special Requirement:	Availability of Internet

2.1.3.4. Explore Menu

Use Case ID:	**UC_4**
Use Case Name:	Explore Menu

Actor:	Customer
Description:	Customers can explore restaurant menus in the Food Panda app, view available food items, detailed descriptions, prices, and add items to their shopping cart. This feature enables Customers to make informed food choices.
Event:	Initiated by the Customer.
Pre-Condition:	Customer should be logged into the system.
Post-Condition:	Customer views restaurant menu and can add items to the cart.
Normal Flow:	➢ Customer browses the restaurant's menu items. ➢ Customer selects a menu item to view its details. ➢ Customer views the item's name, description, price, and options. ➢ Customer adds the item to their shopping cart.
Alternative Flow:	Customers can customize menu items with options like specifying preferences or adding special instructions.
Exceptions:	Null exceptions if no menu items are found.
Includes:	Shopping Cart
Special Requirement:	Availability of Internet

2.1.3.5. Manage Cart

Use Case ID:	**UC_5**
Use Case Name:	Manage Cart
Actor:	Customer
Description:	Customer s can manage the contents of their shopping cart in the Food Panda app, including adding, removing, and updating items. This allows Customers to review and modify their order before finalizing it.
Event:	Initiated by the Customer.
Pre-Condition:	Customer should be logged into the system and have items in the cart.
Post-Condition:	Customer 's shopping cart is updated as per their actions.
Normal Flow:	➢ Customer views the list of items in the cart. ➢ Customer can increase or decrease the quantity of items in the cart. ➢ Customer can remove items from the cart. ➢ Customer can apply promo codes or discounts if available.

Alternative Flow:	Customer s can customize menu items with options like specifying preferences or adding special instructions.
Exceptions:	Null exceptions if the cart is empty or no items are found for modification.
Includes:	Promo Codes/Discounts
Special Requirement:	Availability of Internet

2.1.3.6. Place Order

Use Case ID:	**UC_6**
Use Case Name:	Place Order
Actor:	Customer
Description:	Customer s can place an order in the Food Panda app, confirming their food choices and proceeding with the payment process. This step initiates the order fulfillment process.
Event:	Initiated by the Customer.
Pre-Condition:	Customer should be logged into the system and have items in the shopping cart.
Post-Condition:	The order is placed, and the Customer receives an order confirmation.
Normal Flow:	➢ Customer reviews the items in the cart. ➢ Customer confirms the order for delivery. ➢ Customer selects the preferred payment method. ➢ Customer enters payment details (e.g., credit card information). ➢ Customer confirms the payment. ➢ The system generates an order confirmation with the estimated delivery time.
Alternative Flow:	Customer s can apply promo codes or discounts if available before confirming the order.
Exceptions:	Null exceptions if the cart is empty or payment details are incorrect.
Includes:	Promo Codes/Discounts
Special Requirement:	Availability of Internet

2.1.3.7. Track Order

Use Case ID:	**UC_7**
Use Case Name:	Track Orders
Actor:	Customer, Rider
Description:	Users can track the status and location of their food orders in real-time within the Food Panda app,

	ensuring transparency and providing estimated delivery times.
Event:	Initiated by the user.
Pre-Condition:	User should be logged into the system and have at least one active order.
Post-Condition:	User views the real-time status and location of the order.
Normal Flow:	➢ User views a list of active orders. ➢ User selects a specific order for tracking. ➢ The system displays the real-time status of the order, from preparation to delivery. ➢ The user receives notifications for key stages in the delivery process, such as order confirmed, food prepared, and out for delivery. ➢ The estimated delivery time is provided to the user.
Alternative Flow:	Users can contact the delivery rider or customer support through the app for inquiries or updates.
Exceptions:	Null exceptions if there are no active orders to track.
Includes:	Notifications
Special Requirement:	Availability of Internet

2.1.3.8. Process Payment

Use Case ID:	**UC_8**
Use Case Name:	Process Payments
Actor:	Customer
Description:	Customer s can process payments for their food orders within the Food Panda app, ensuring secure and convenient transactions.
Event:	Initiated by the Customer.
Pre-Condition:	Customer should be logged into the system, have added items to the shopping cart, and have confirmed the order.
Post-Condition:	Payment is successfully processed, and the order is confirmed.
Normal Flow:	➢ Customer selects the preferred payment method (e.g., credit card, digital wallet, cash on delivery). ➢ Customer enters payment details if required. ➢ Customer confirms the payment. ➢ The system processes the payment and confirms the order. ➢ The user receives an order confirmation with the estimated delivery time.

Alternative Flow:	Customers can apply promo codes, discounts, or loyalty points if available to reduce the order total before confirming payment.
Exceptions:	➢ Null exceptions if payment details are incorrect or cannot be processed. ➢ If the payment method selected is "cash on delivery," the system should mark the order for cash payment upon delivery.
Includes:	Promo Codes/Discounts, Loyalty Points
Special Requirement:	Availability of Internet

2.1.3.9. View Order History

Use Case ID:	**UC_9**
Use Case Name:	View Order History
Actor:	Customer
Description:	Customers can view their order history in the Food Panda app, providing access to details of past orders for reference and reordering.
Event:	Initiated by the Customer.
Pre-Condition:	Customer should be logged into the system.
Post-Condition:	Customer views the order history.
Normal Flow:	➢ Customer views a list of past orders, including order numbers, restaurants, order dates, and order status. ➢ Customer selects a specific order to view its details. ➢ The system displays the order details, including items ordered, total cost, delivery address, and order status.
Alternative Flow:	Customers have the option to reorder items from their order history with a single click.
Exceptions:	Null exceptions if there is no order history available.
Includes:	Reorder Functionality
Special Requirement:	Availability of Internet

2.1.3.10. Rate and Reviews

Use Case ID:	**UC_9**
Use Case Name:	View Order History
Actor:	Customer

Description:	Customers can view their order history in the Food Panda app, providing access to details of past orders for reference and reordering.
Event:	Initiated by the Customer.
Pre-Condition:	Customer should be logged into the system.
Post-Condition:	Customer views the order history.
Normal Flow:	➢ Customer views a list of past orders, including order numbers, restaurants, order dates, and order status. ➢ Customer selects a specific order to view its details. ➢ The system displays the order details, including items ordered, total cost, delivery address, and order status.
Alternative Flow:	Customers have the option to reorder items from their order history with a single click.
Exceptions:	Null exceptions if there is no order history available.
Includes:	Reorder Functionality
Special Requirement:	Availability of Internet

2.1.3.11. Manage Profiles

Use Case ID:	**UC_11**
Use Case Name:	Manage Profile
Actor:	Customer, Rider
Description:	Users can manage their profiles within the Food Panda app, allowing them to update personal information, change account settings, and maintain accurate and up-to-date profiles.
Event:	Initiated by the user.
Pre-Condition:	User should be logged into the system.
Post-Condition:	User successfully updates their profile.
Normal Flow:	➢ User selects the specific profile information they want to update (e.g., name, email, phone number). ➢ User makes the desired changes. ➢ User saves the updated profile information.
Alternative Flow:	Users can change account settings such as notification preferences, password, or profile picture.
Exceptions:	Null exceptions if there are issues with updating the profile information.
Includes:	Update Profile Information

Special Requirement:	Availability of Internet

2.1.3.12. Access Customer Support

Use Case ID:	**UC_12**
Use Case Name:	Access Customer Support
Actor:	Customer
Description:	Customer s can access customer support within the Food Panda app to seek assistance, report issues, or get help with any problems they encounter while using the application or during the order process.
Event:	Initiated by the Customer.
Pre-Condition:	Customer should be logged into the system.
Post-Condition:	User receives assistance or submits a support request.
Normal Flow:	➢ Customer selects the type of assistance they need (e.g., report an issue, get order help, or general inquiries). ➢ Customer describes the problem or request in detail. ➢ Customer submits the request for assistance.
Alternative Flow:	Customers may choose to contact customer support via other channels, such as phone or email, if these options are available.
Exceptions:	Null exceptions if there are issues with submitting the support request.
Includes:	Customer Support Assistance
Special Requirement:	Availability of Internet

2.1.3.13. Mark Favorite Restaurants

Use Case ID:	**UC_13**
Use Case Name:	Mark Favorite Restaurants
Actor:	Customer
Description:	Customers can mark their favorite restaurants within the Food Panda app, allowing them to easily access and reorder from their preferred dining establishments.
Event:	Initiated by the customer..
Pre-Condition:	Customer should be logged into the system.
Post-Condition:	Customer successfully marks a restaurant as a favorite.

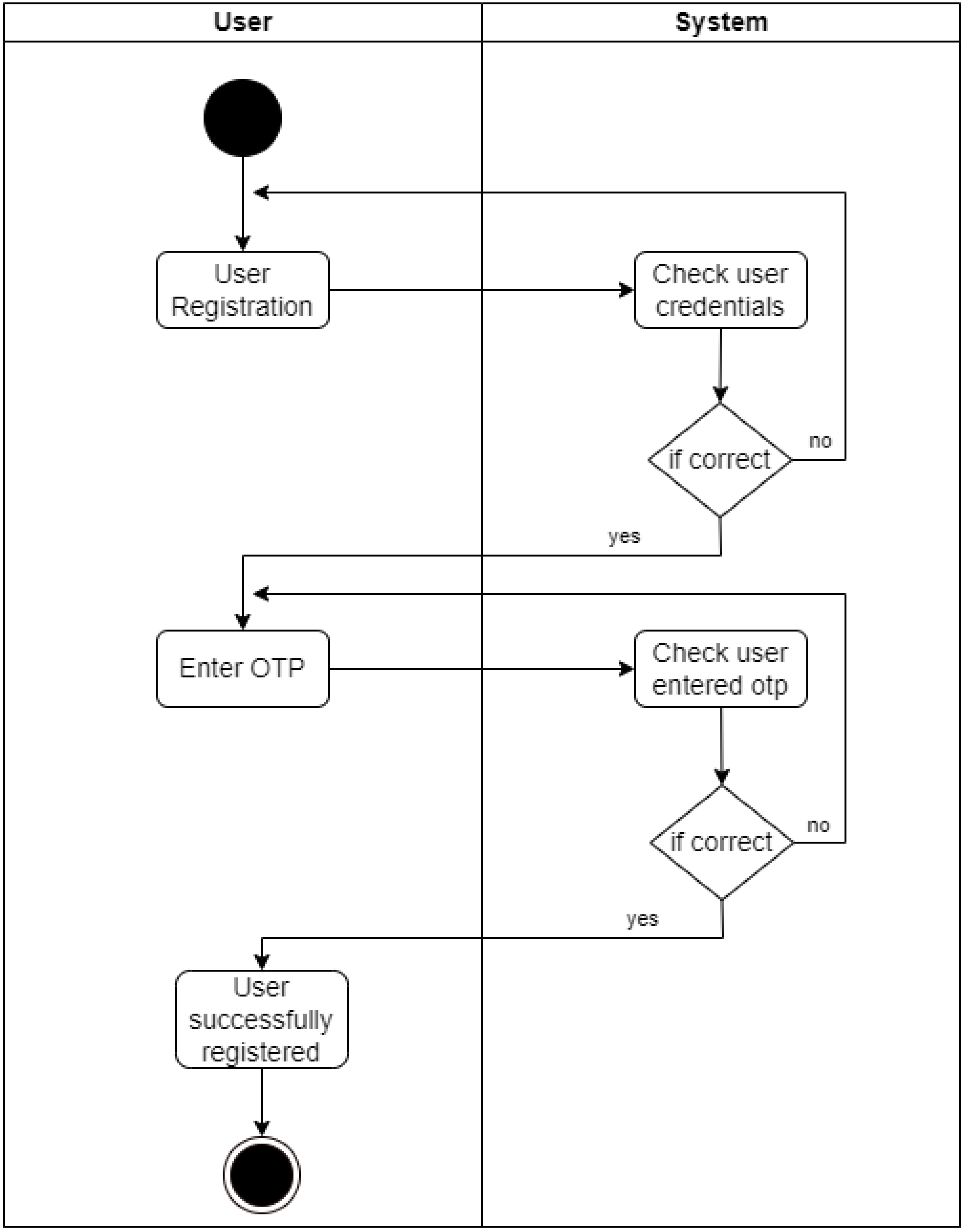
Normal Flow:	➢ Customer browses through the list of available restaurants. ➢ Customer selects a restaurant they want to mark as a favorite.
Alternative Flow:	Customers can unmark a restaurant as a favorite if they no longer wish to keep it in their favorites list.
Exceptions:	Null exceptions if there are issues with marking or unmarking a restaurant as a favorite.
Includes:	Manage Favorite Restaurants
Special Requirement:	Availability of Internet

2.1.3.14. View Promotions and Offers

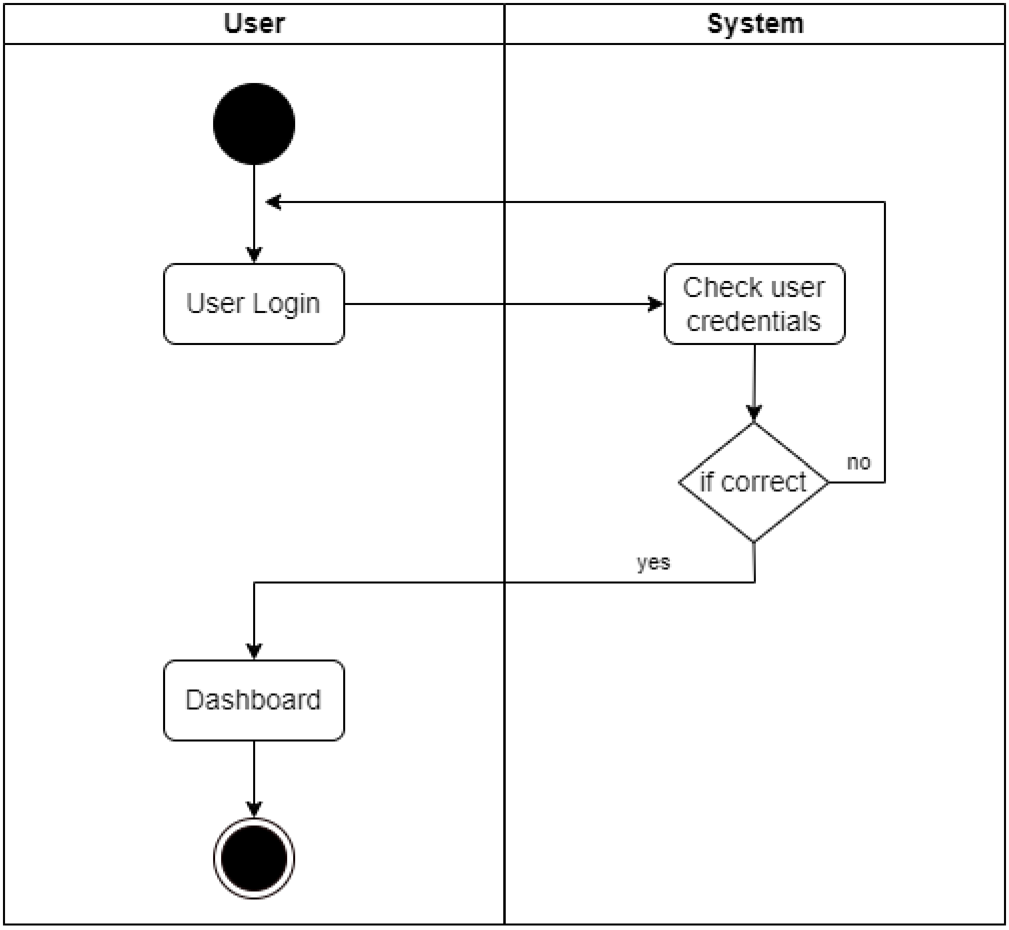
Use Case ID:	**UC_14**
Use Case Name:	View Promotions and Offers
Actor:	Customer
Description:	Customers can view the latest promotions, discounts, and special offers within the Food Panda app, enabling them to take advantage of cost-saving opportunities when placing orders.
Event:	Initiated by the customer..
Pre-Condition:	Customer should be logged into the system.
Post-Condition:	Customer successfully views available promotions and offers.
Normal Flow:	➢ Customer views a list of ongoing promotions and special offers from various restaurants. ➢ Customer selects a specific promotion or offer to learn more. ➢ Customer can choose to apply the promotion or offer to their order, if applicable.
Alternative Flow:	Some promotions or offers may require a promo code for application. In such cases, the customer can enter the code during the order process.
Exceptions:	Null exceptions if there are issues with viewing or applying promotions and offers.
Includes:	Apply Promotions and Offers
Special Requirement:	Availability of Internet

2.1.4. Activity Diagram

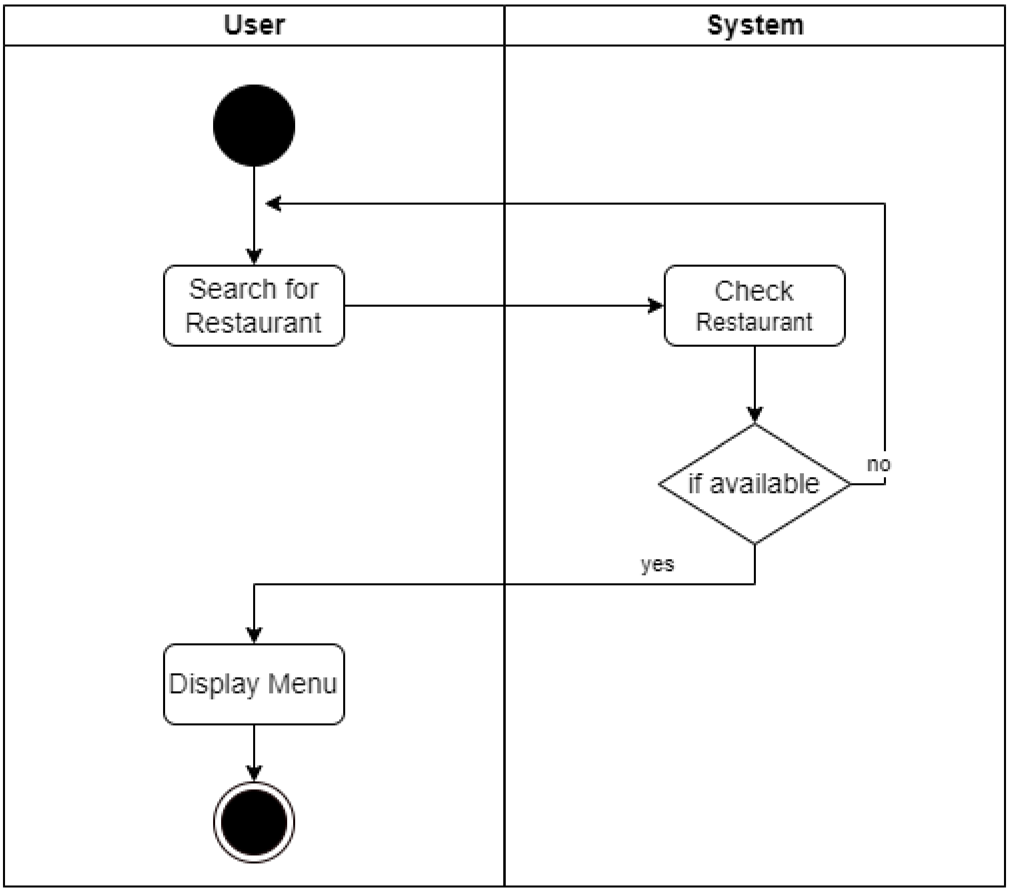
2.1.4.1. User Registration



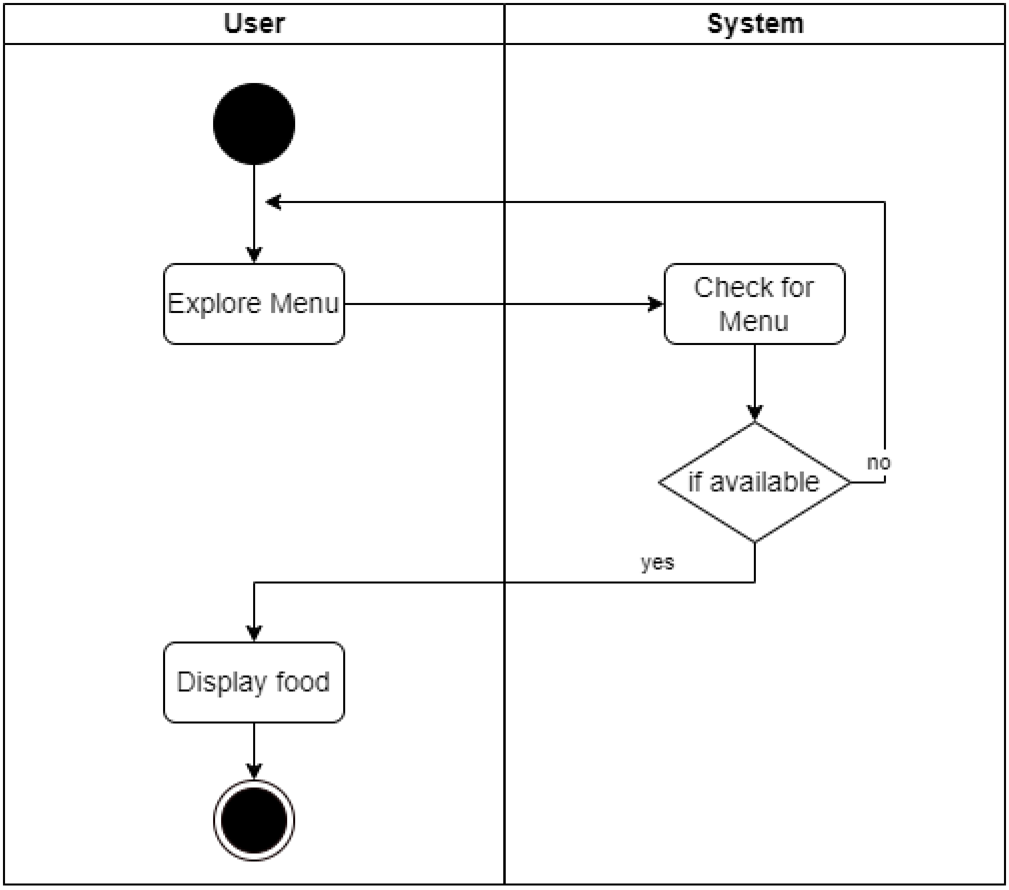
2.1.4.2. User Login



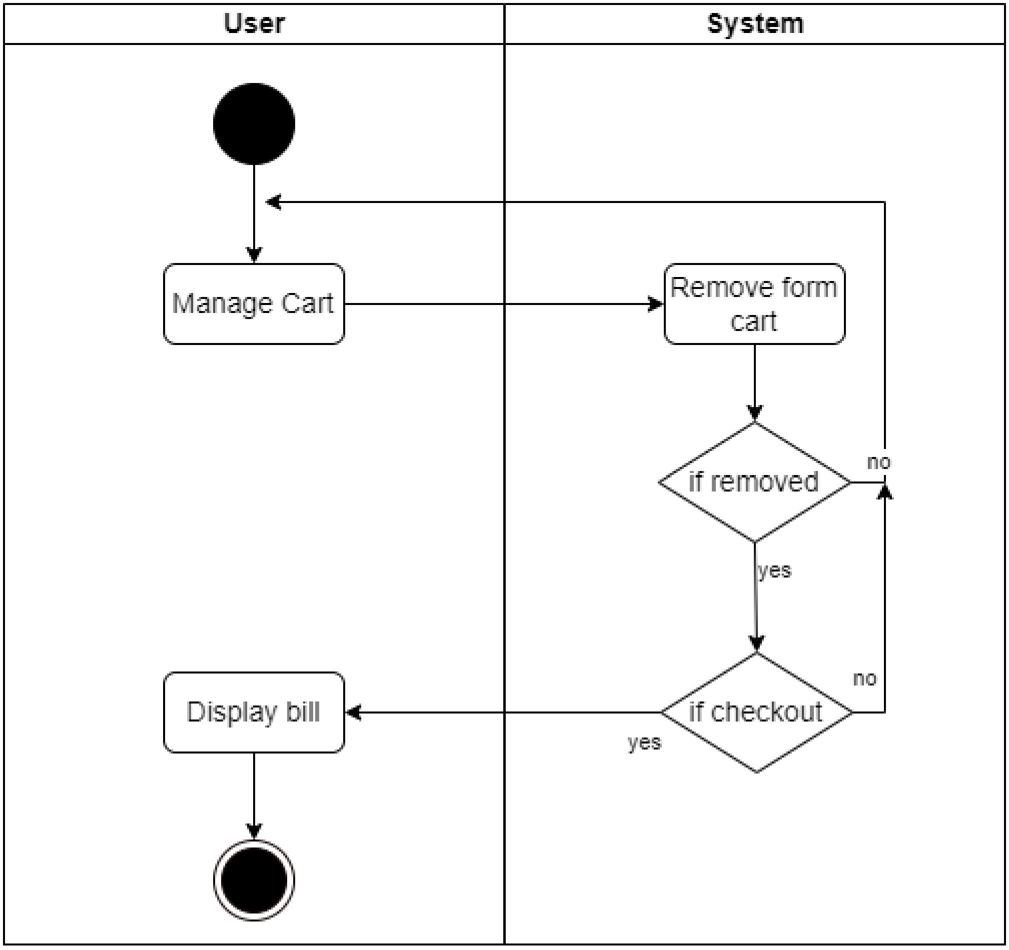
2.1.4.3. Search Restaurants



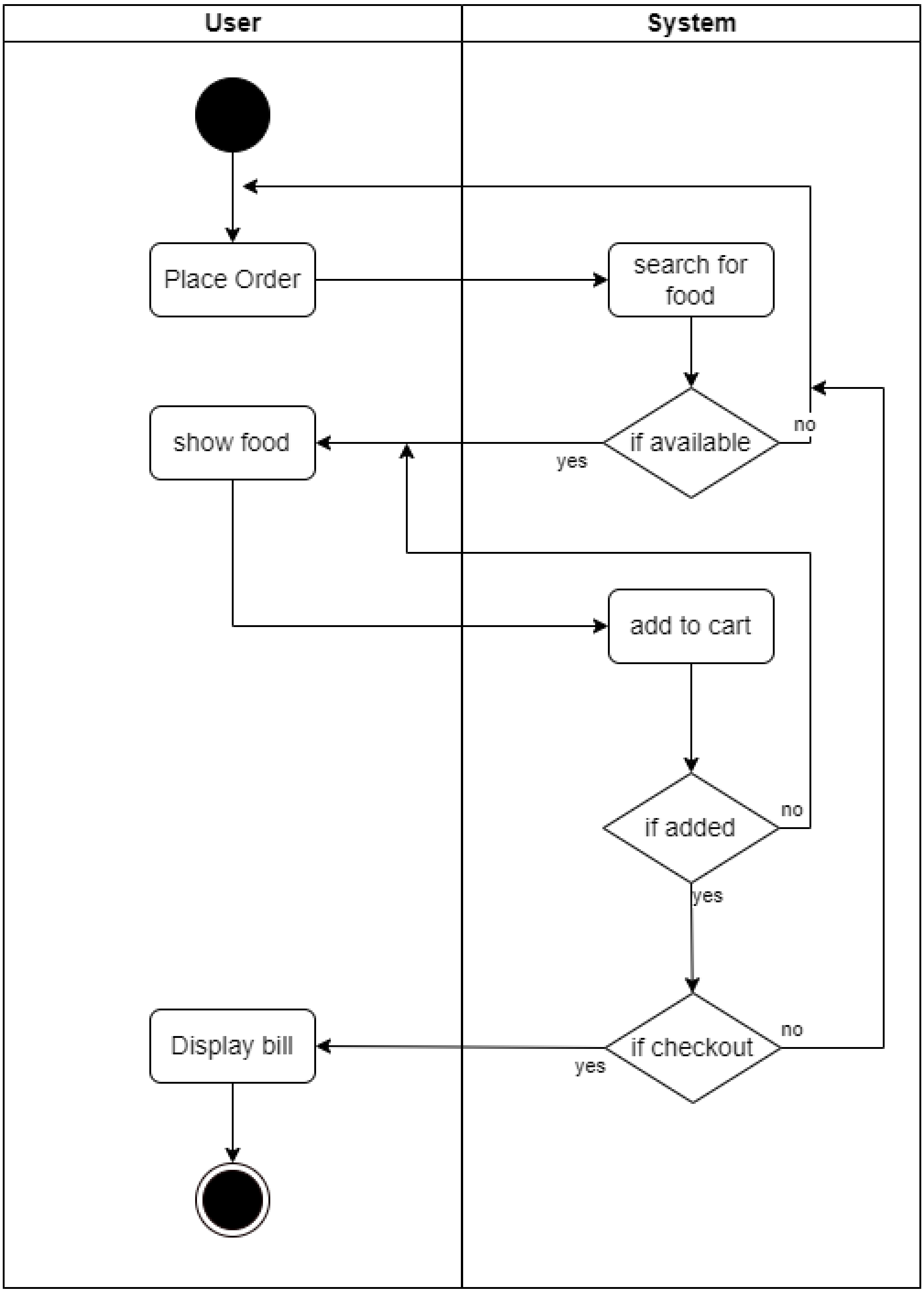
2.1.4.4. Explore Menu



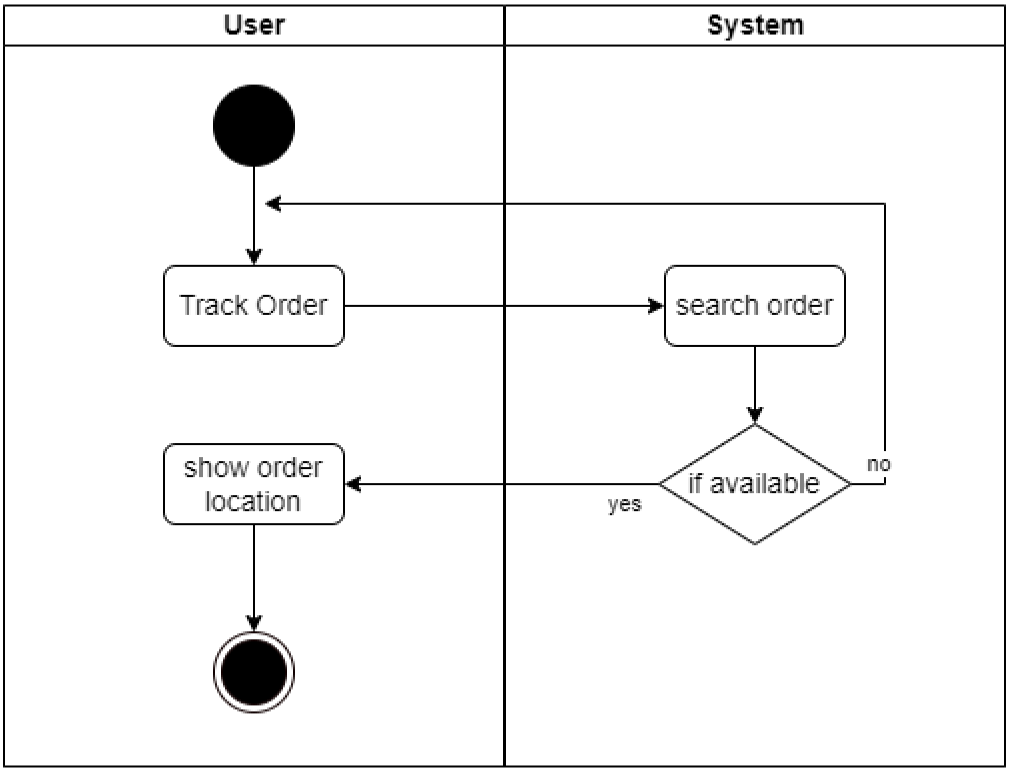
2.1.4.5. Manage Cart



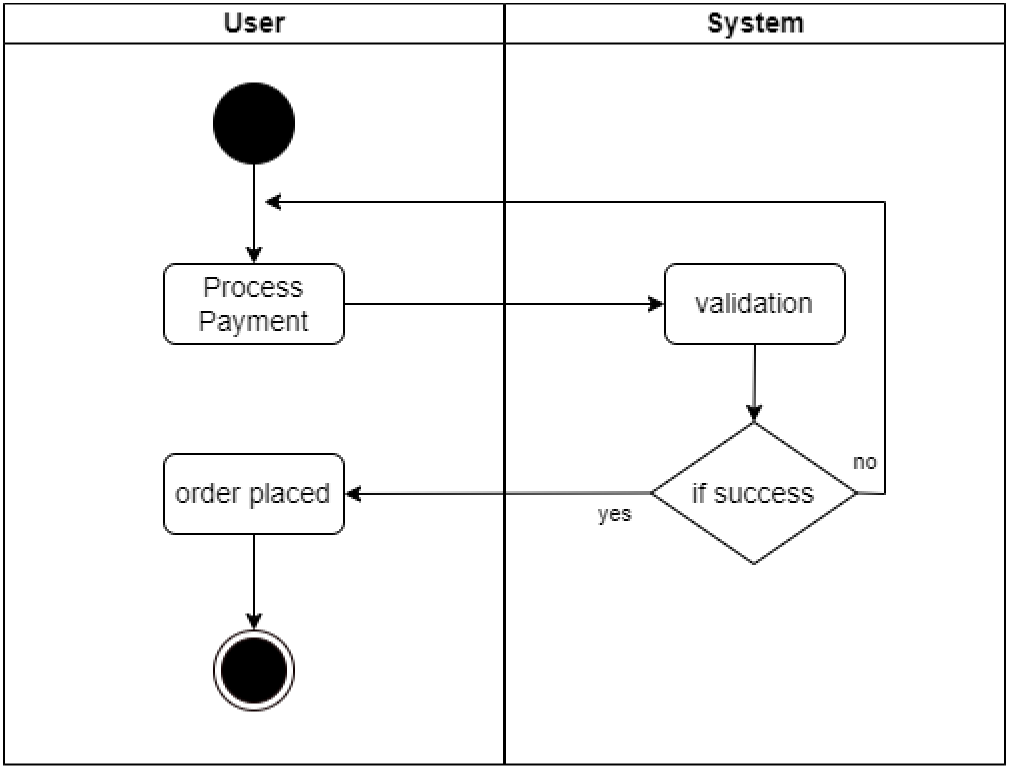
2.1.4.6. Place Order



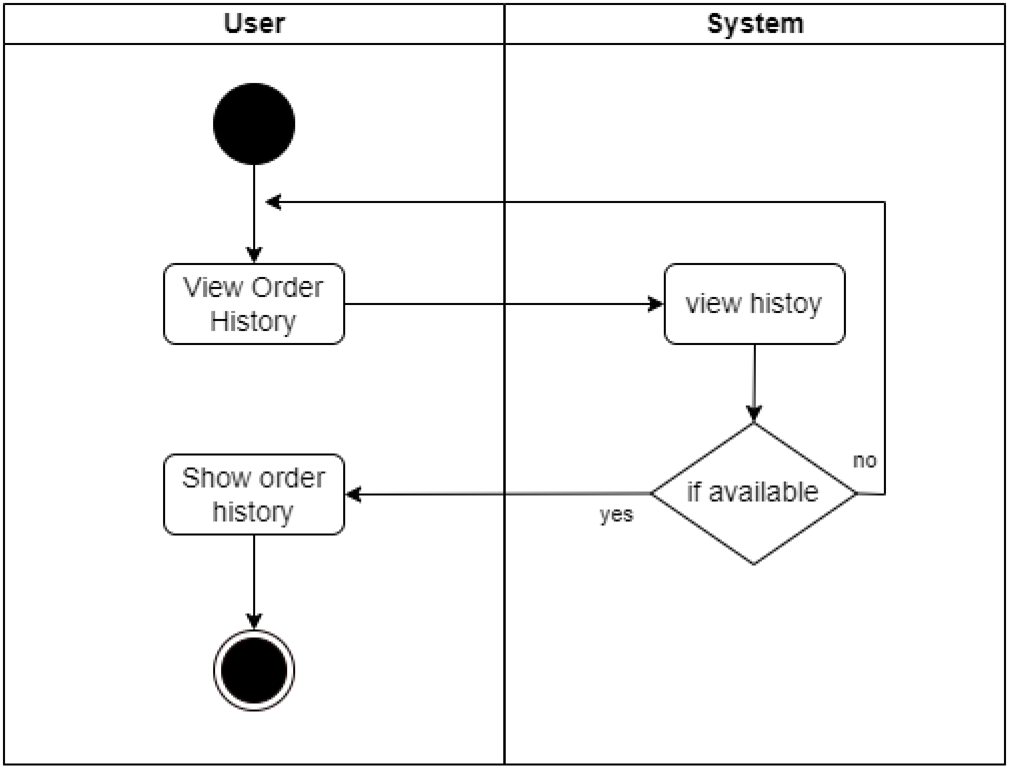
2.1.4.7. Track Order



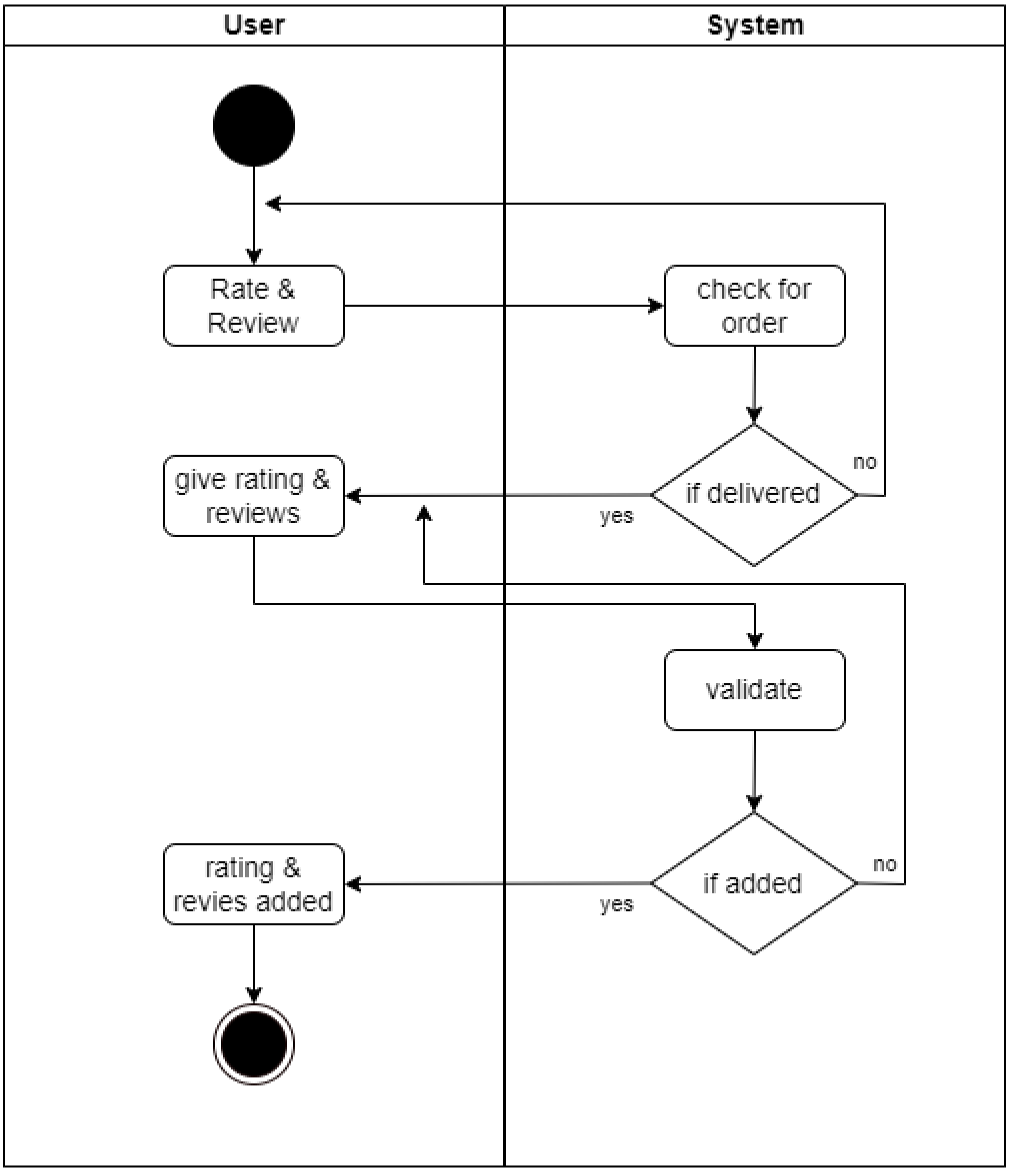
2.1.4.8. Process Payment



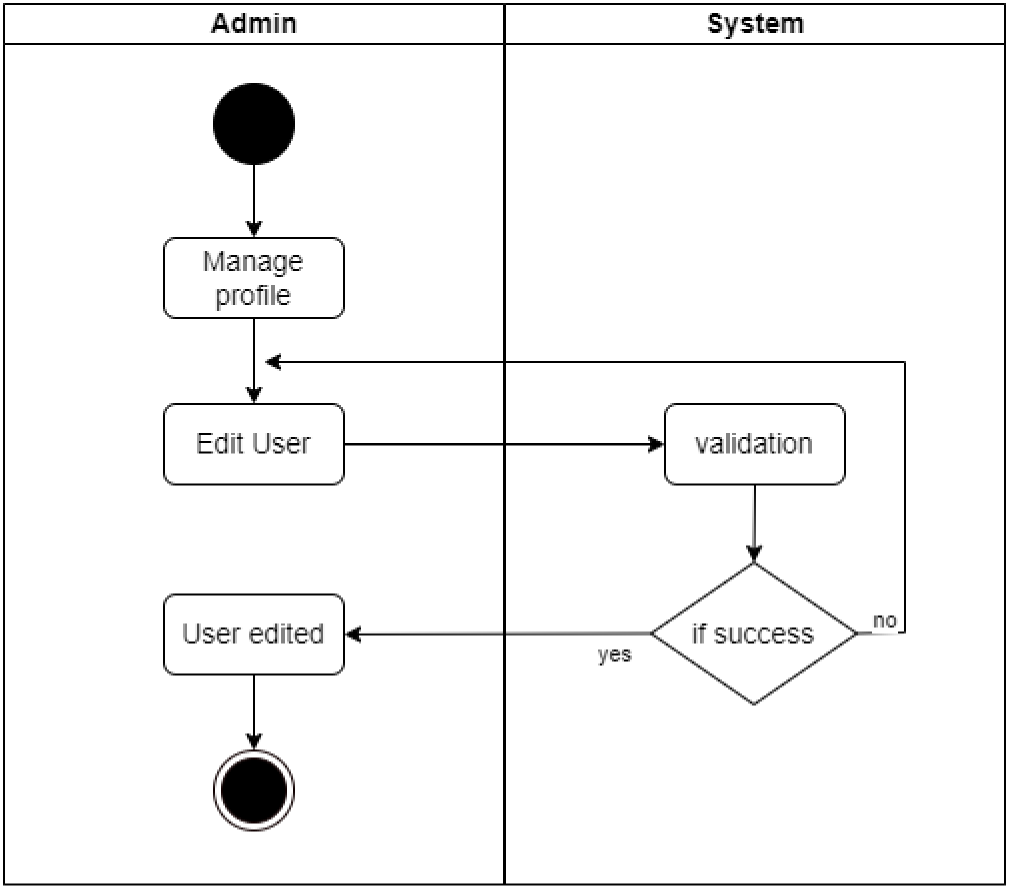
2.1.4.9. View Order History



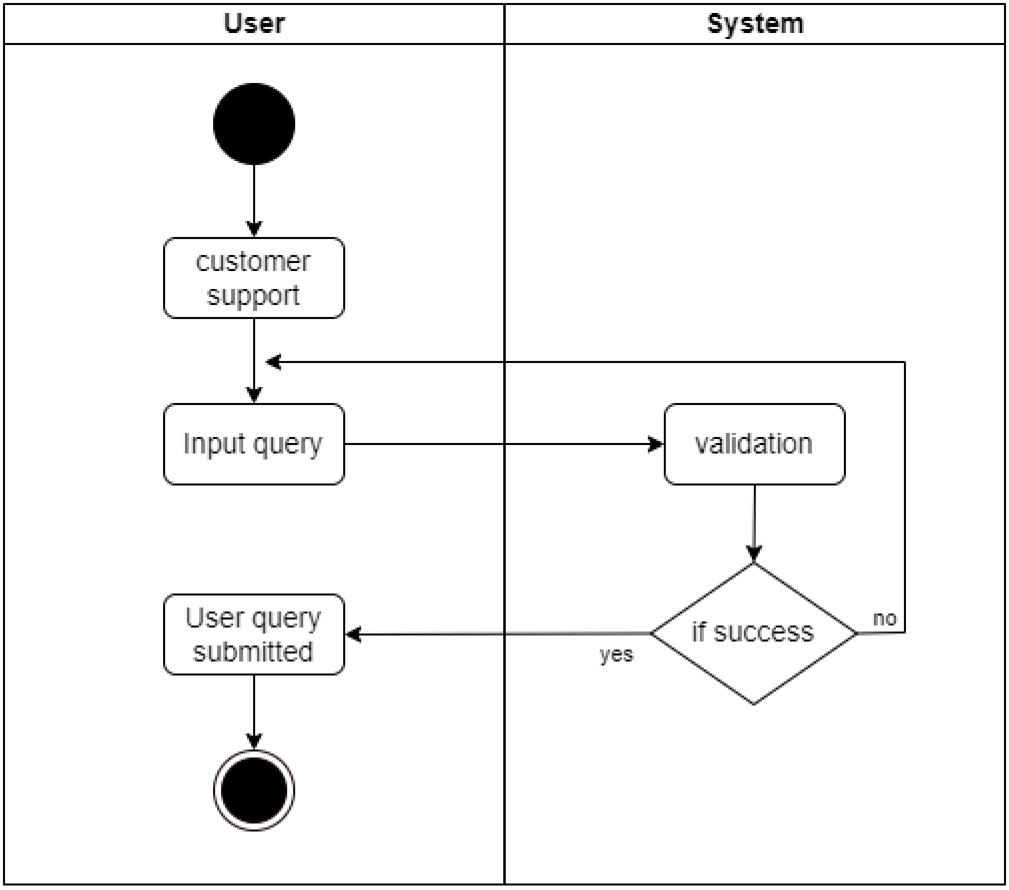
2.1.4.10. Rate and Reviews



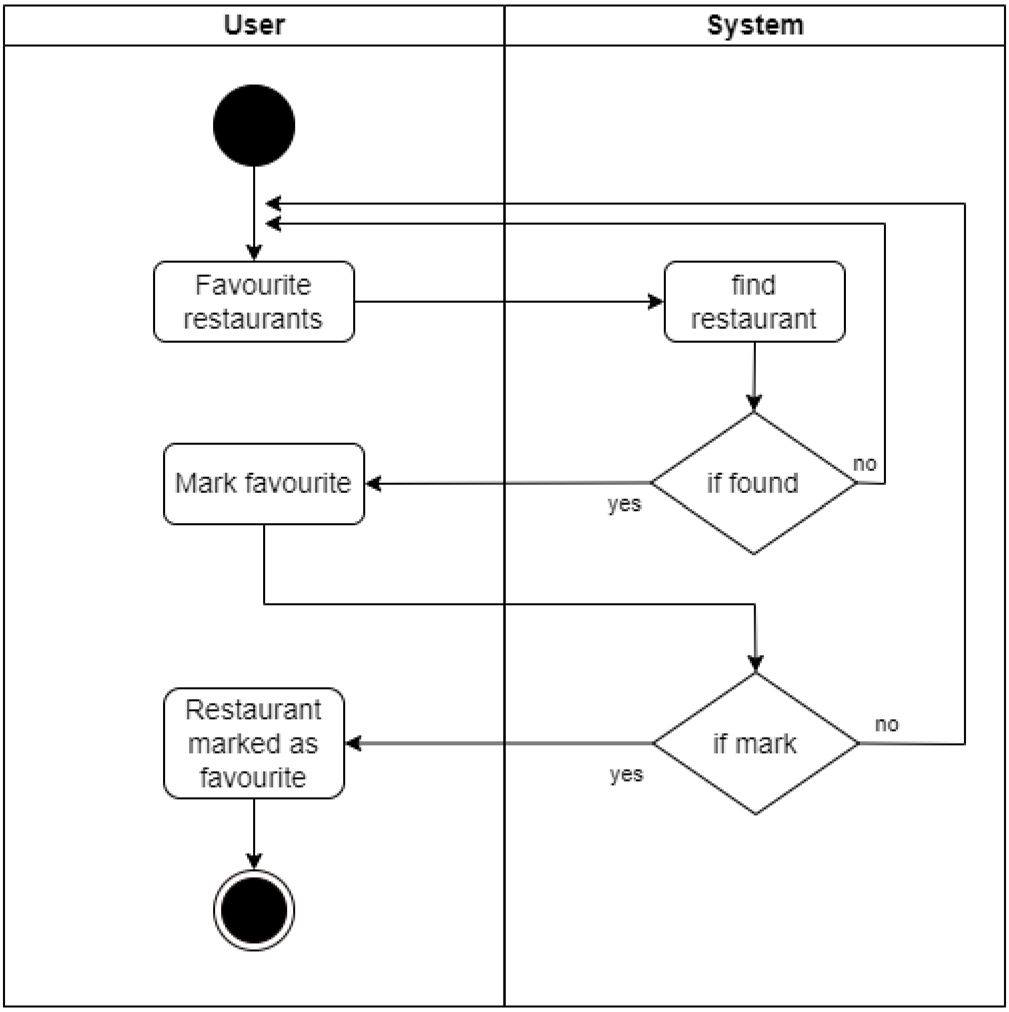
2.1.4.11. Manage Profiles



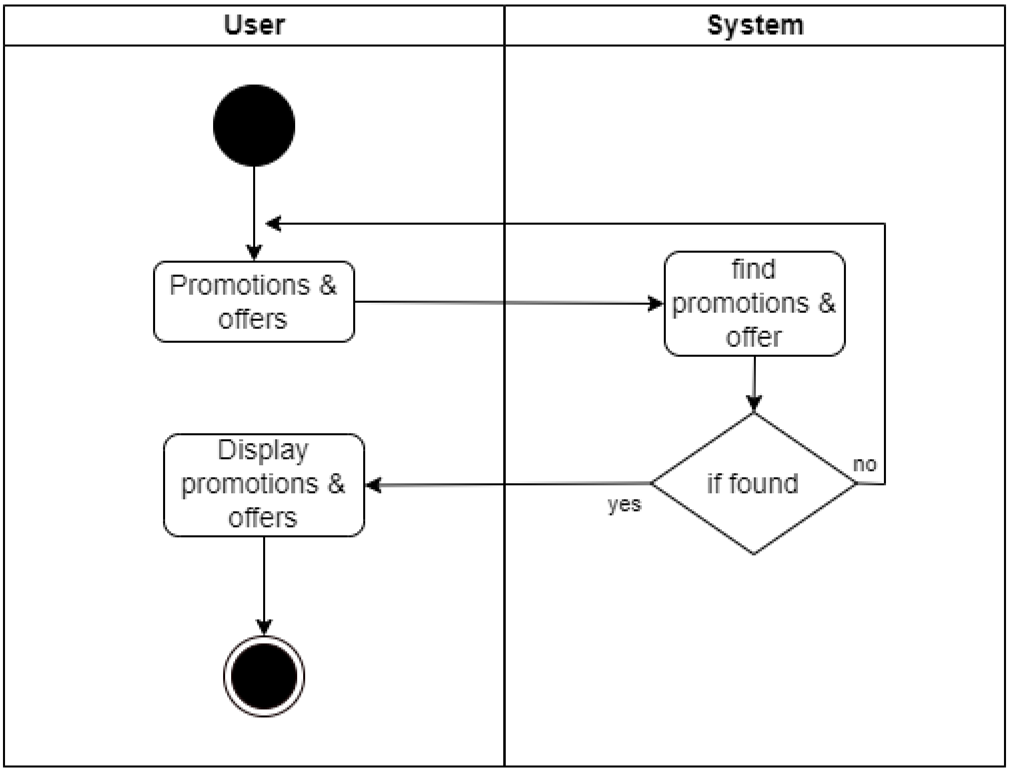
2.1.4.12. Access Customer Support



2.1.4.13. Mark Favorite Restaurants

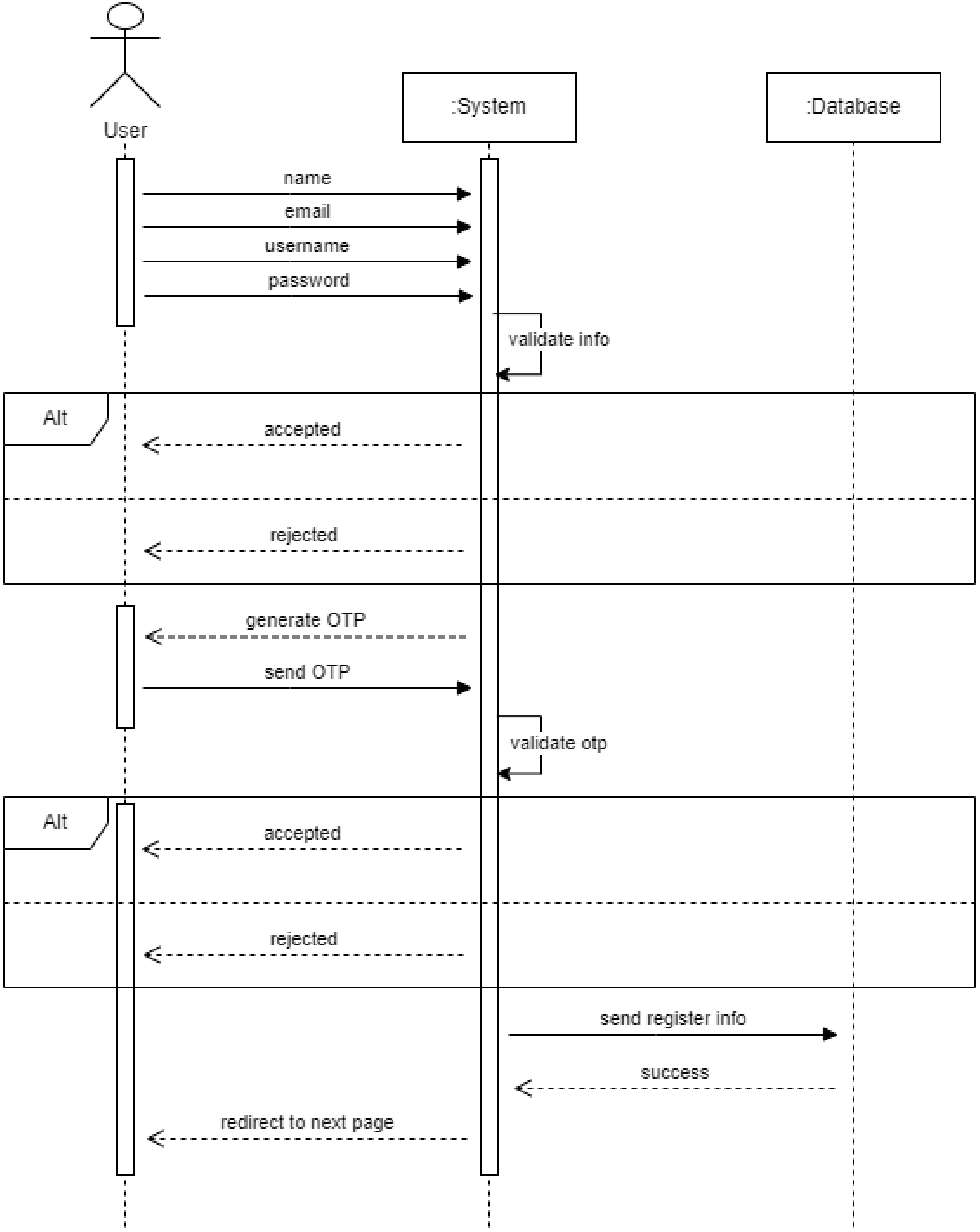


2.1.4.14. View Promotions and Offers

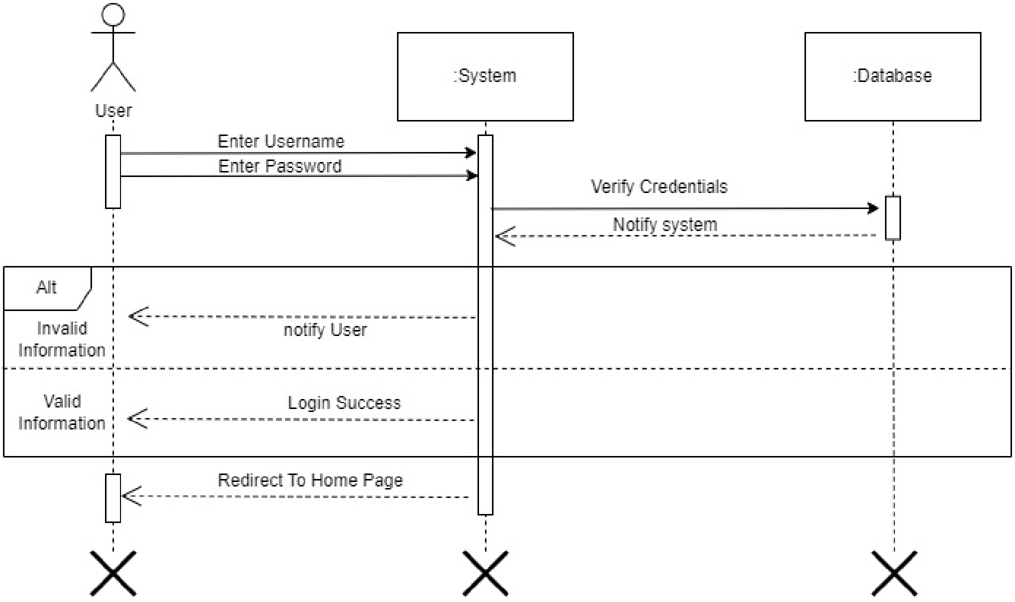


2.1.5. Sequence Diagram

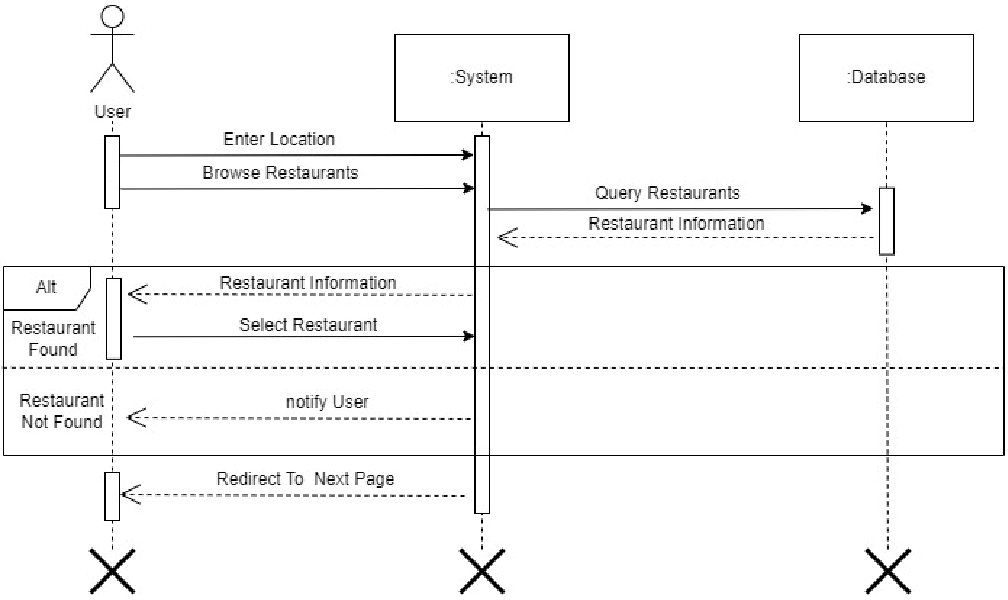
2.1.5.1. User Registration



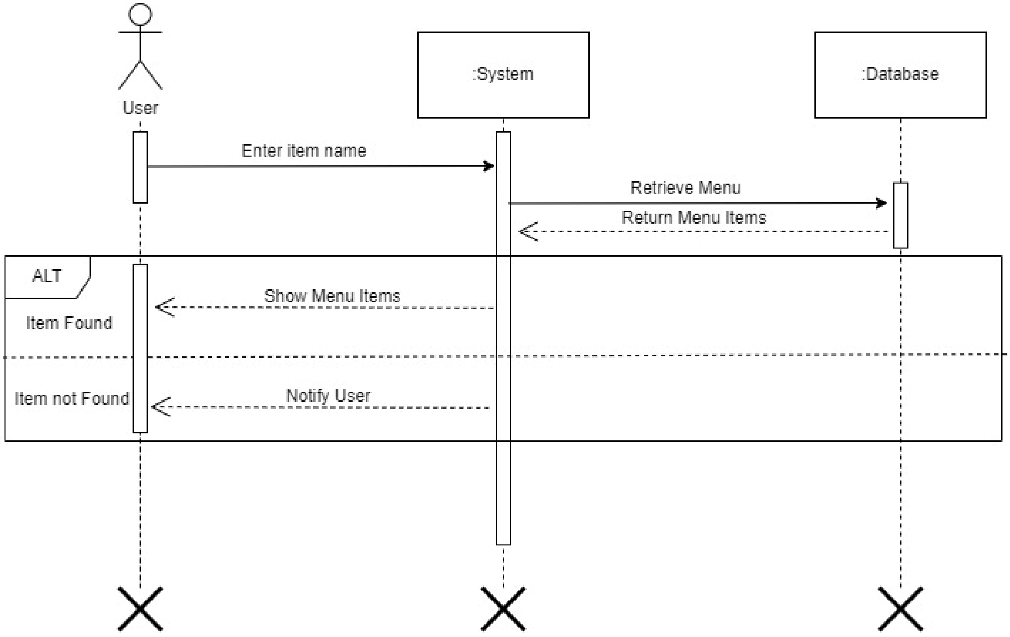
2.1.5.2. User Login



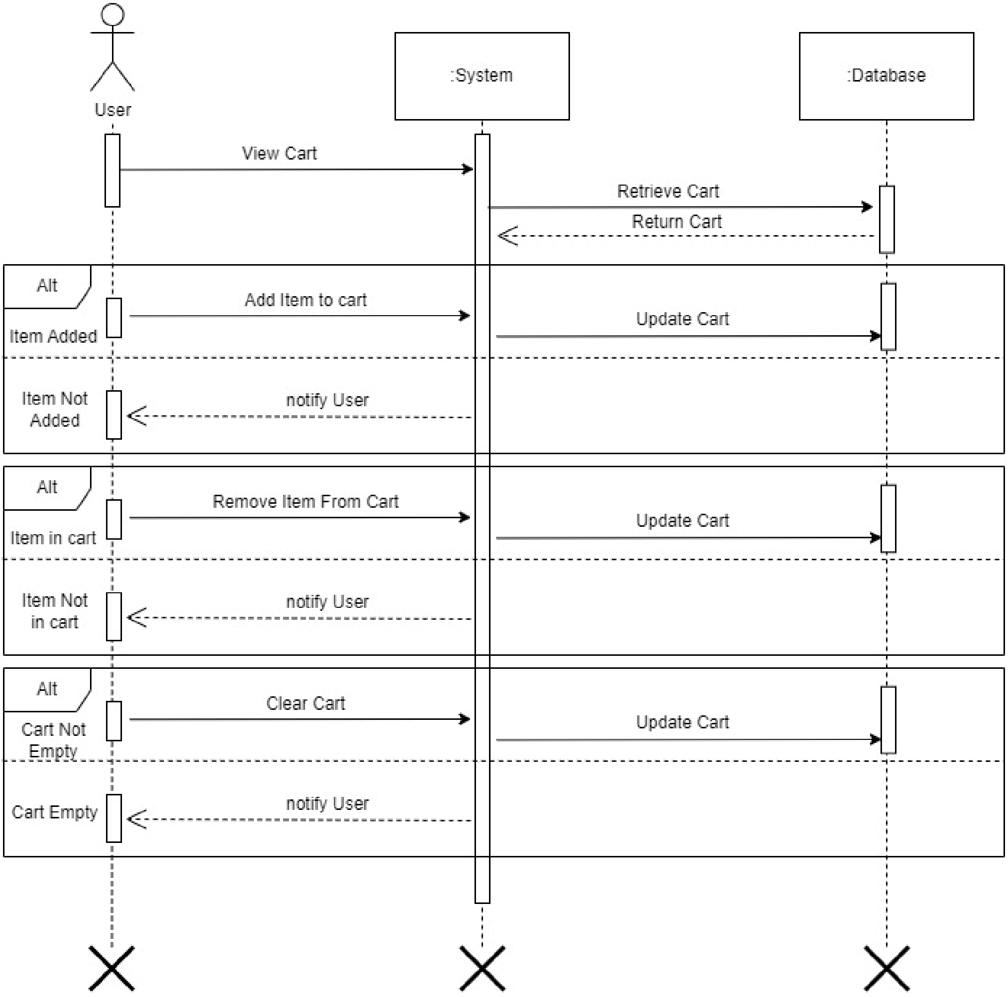
2.1.5.3. Search Restaurants



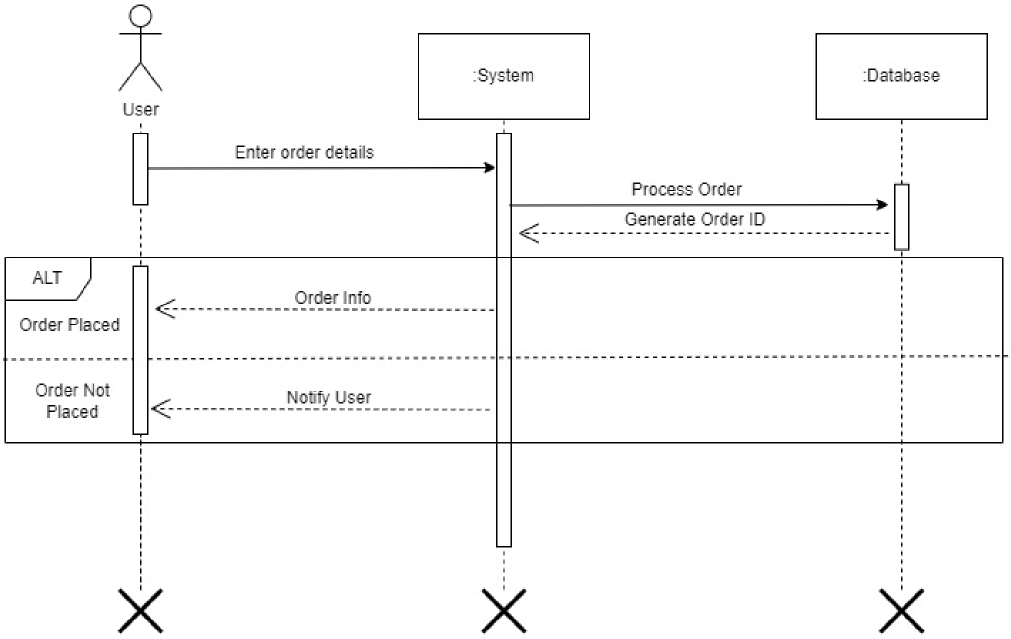
2.1.5.4. Explore Menu



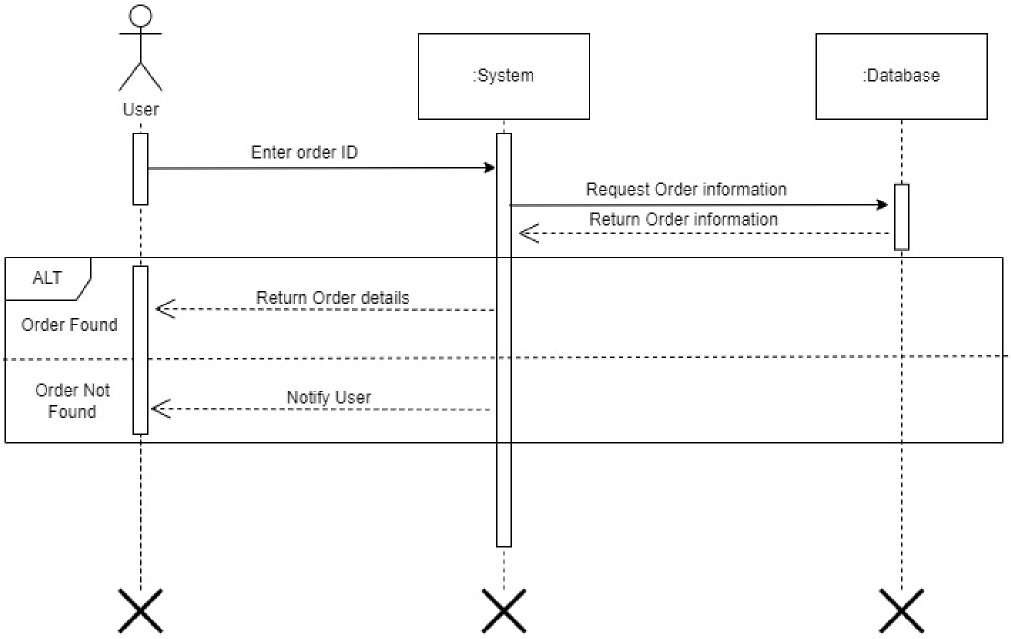
2.1.5.5. Manage Cart



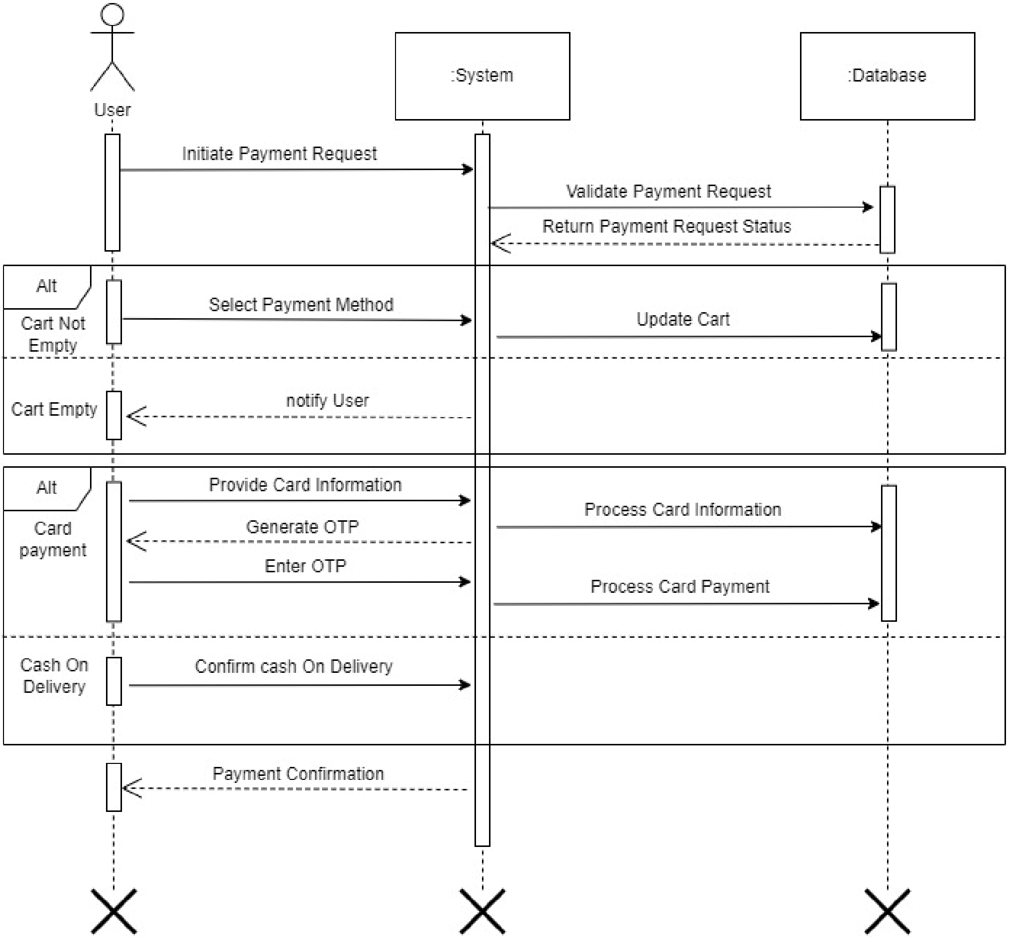
2.1.5.6. Place Order



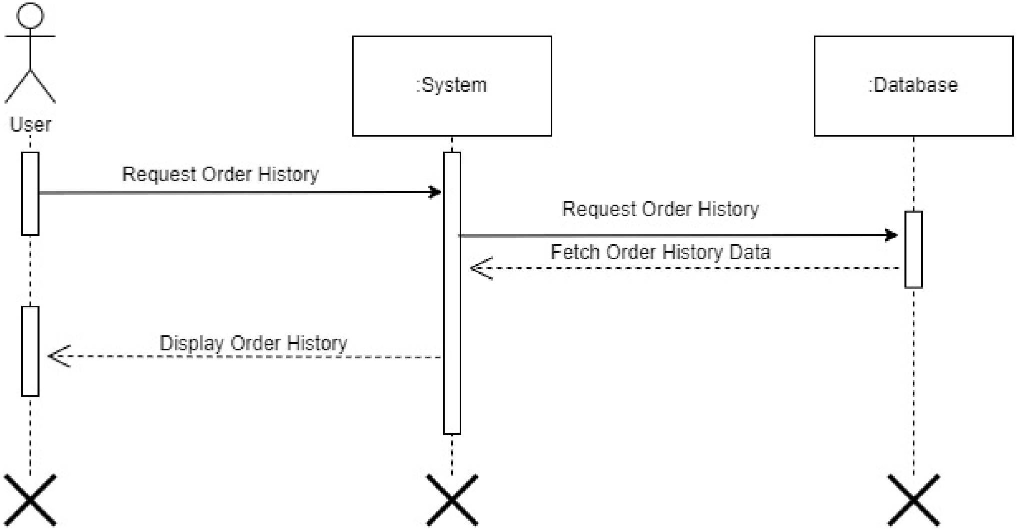
2.1.5.7. Track Order



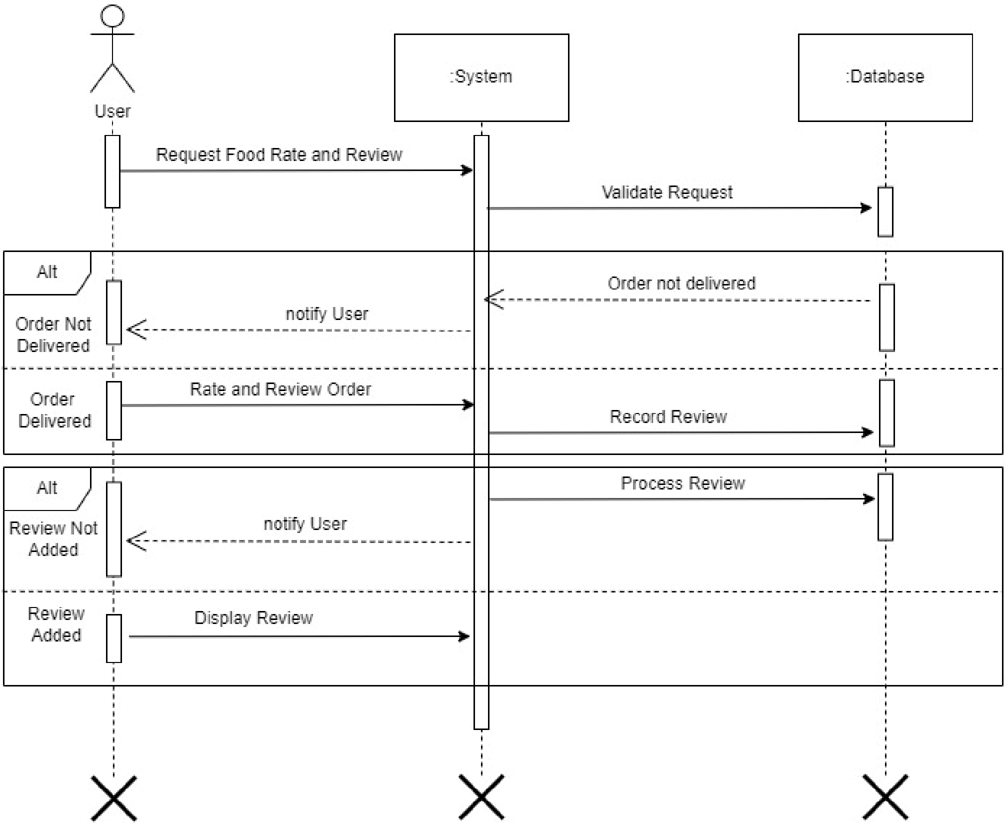
2.1.5.8. Process Payment



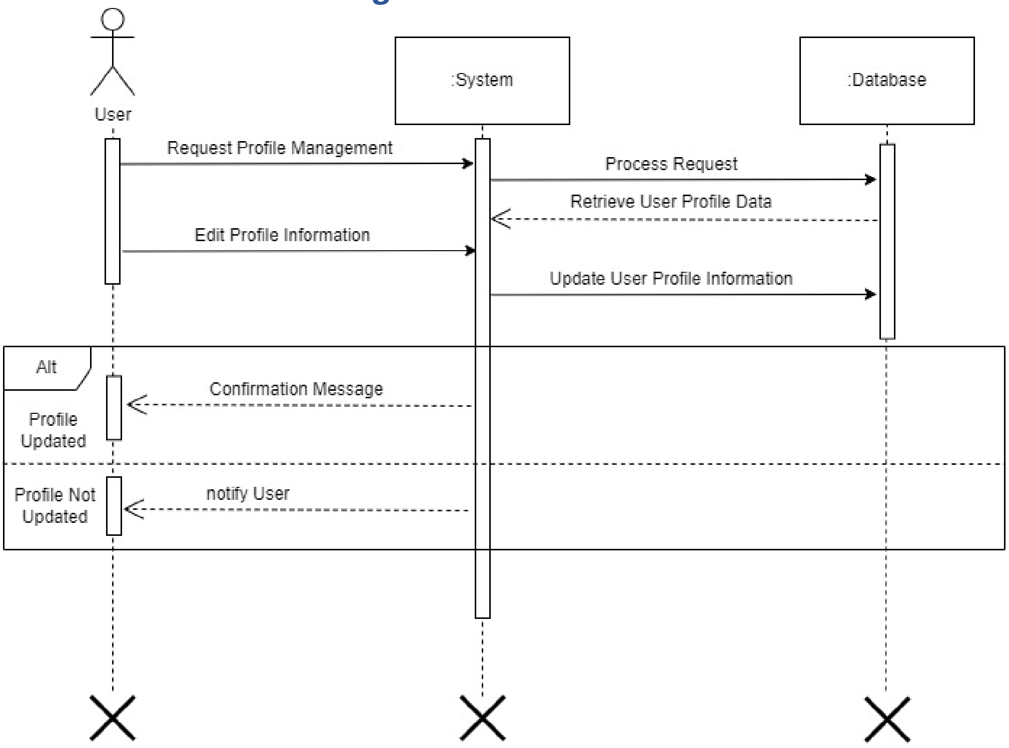
2.1.5.9. View Order History



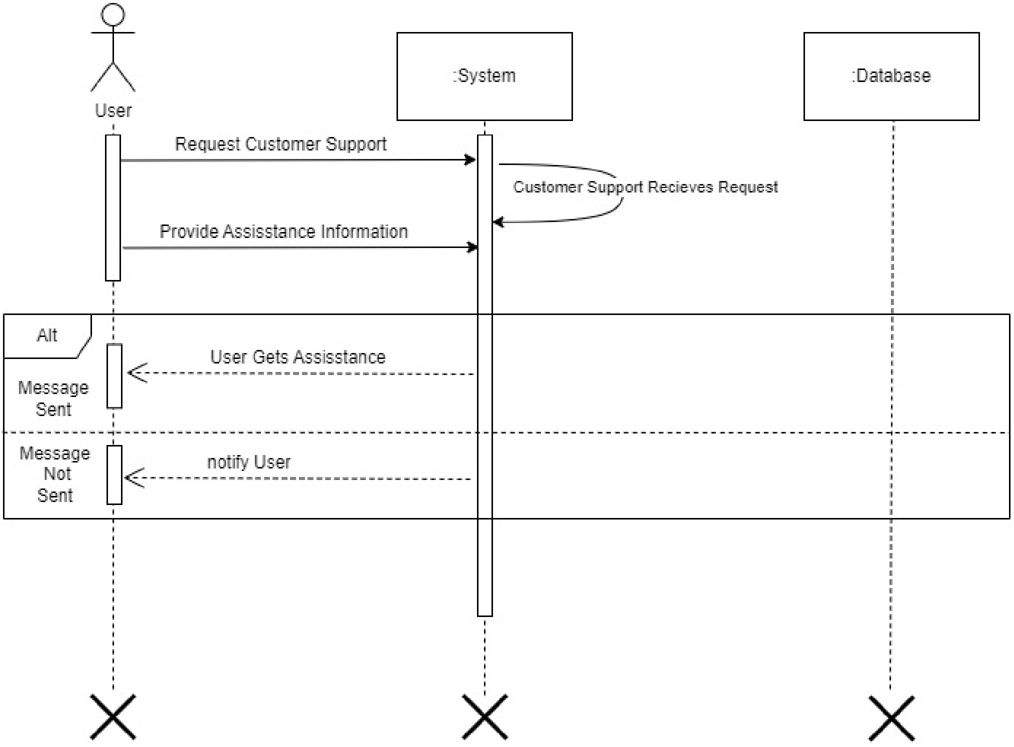
2.1.5.10. Rate and Reviews



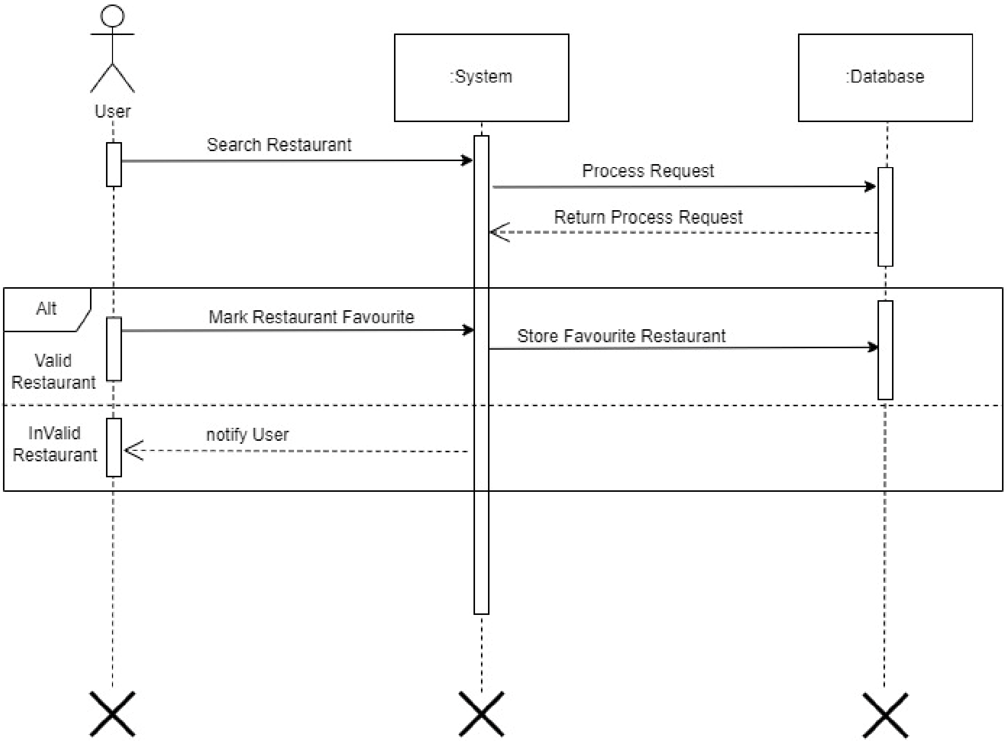
2.1.5.11. Manage Profiles



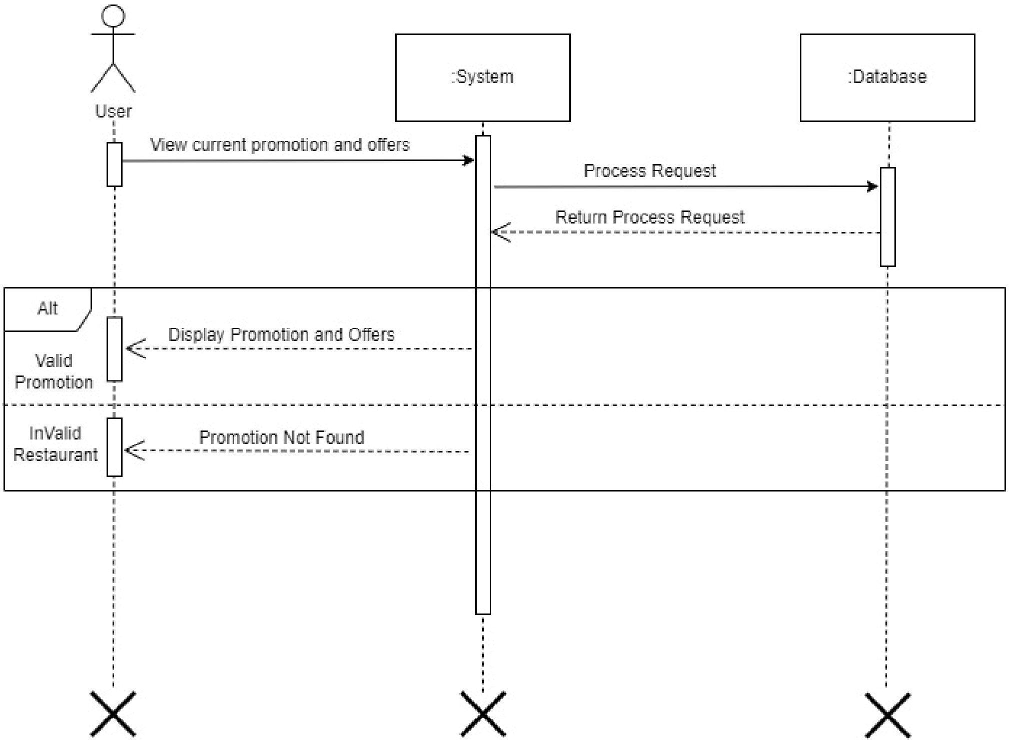
2.1.5.12. Access Customer Support



2.1.5.13. Mark Favorite Restaurants

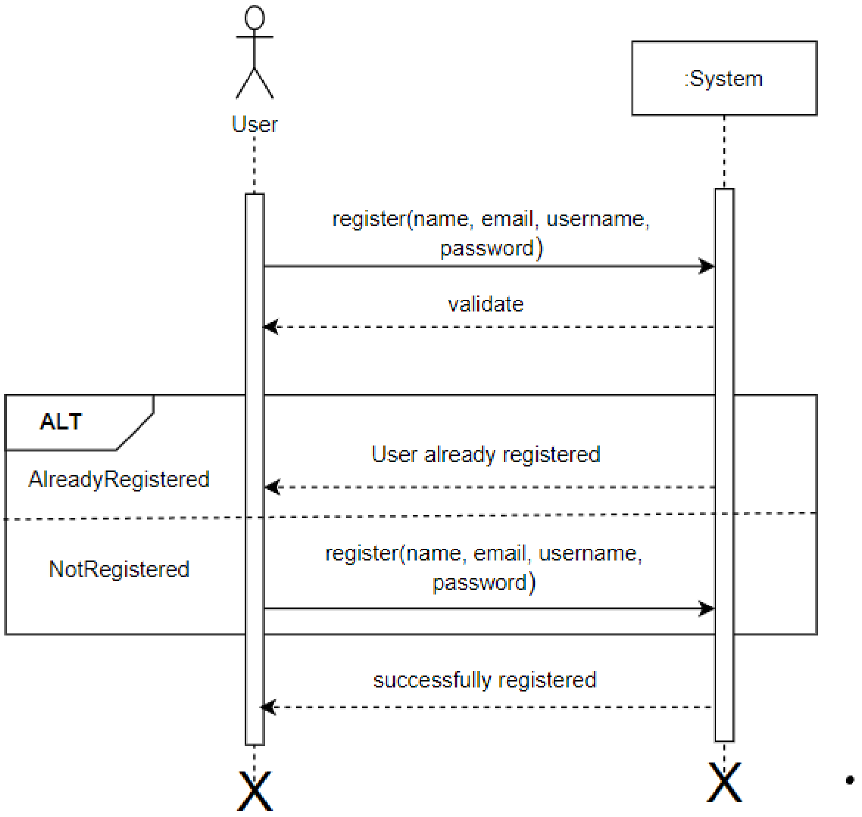


2.1.5.14. View Promotions and Offers

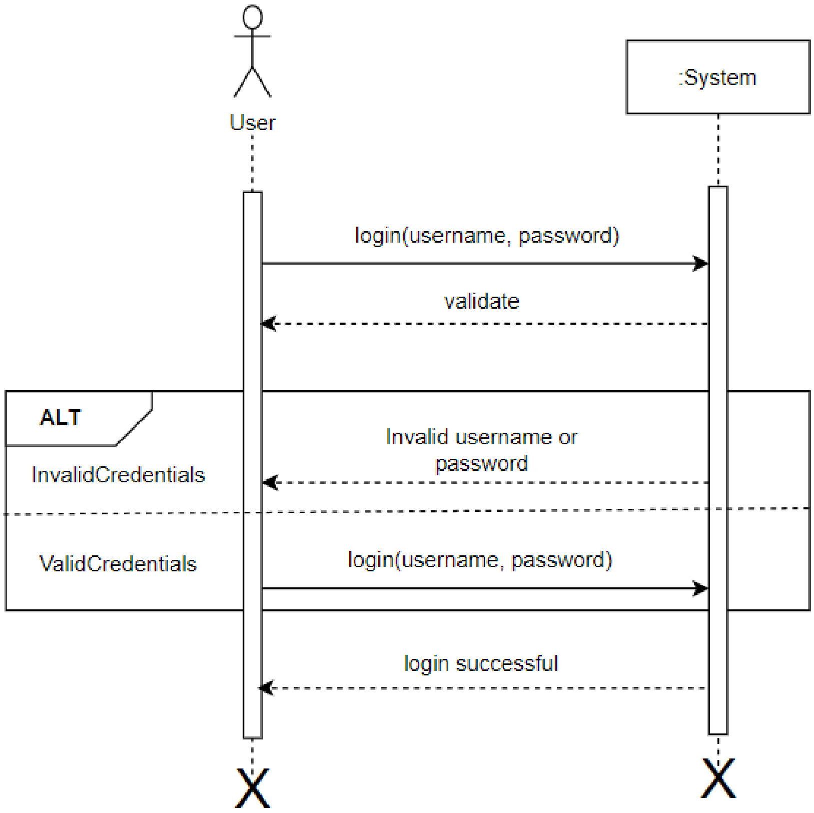


2.1.6. System Sequence Diagram

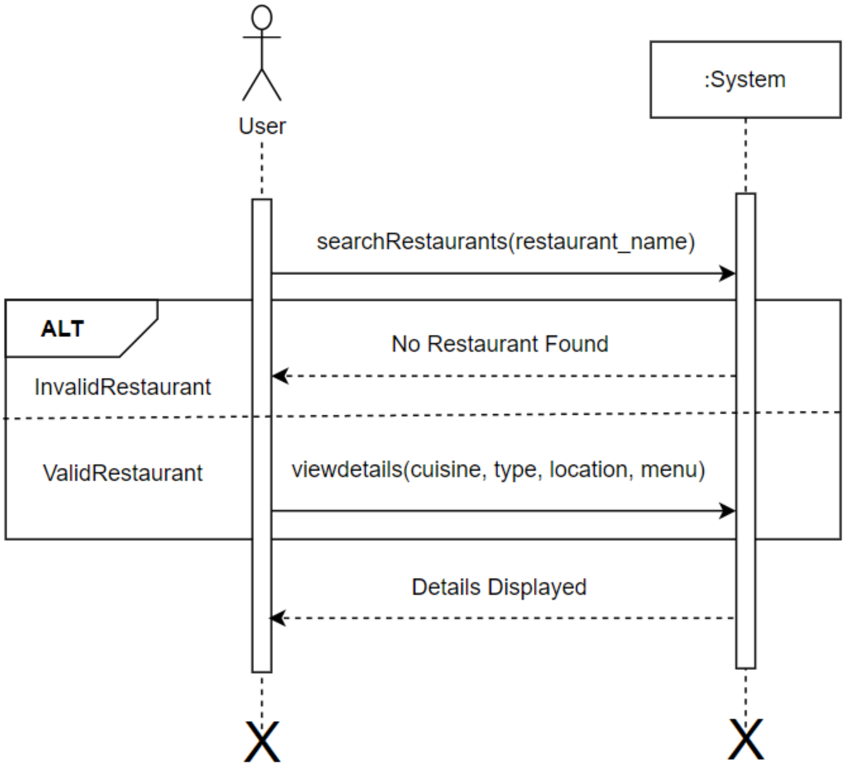
2.1.6.1. User Registration



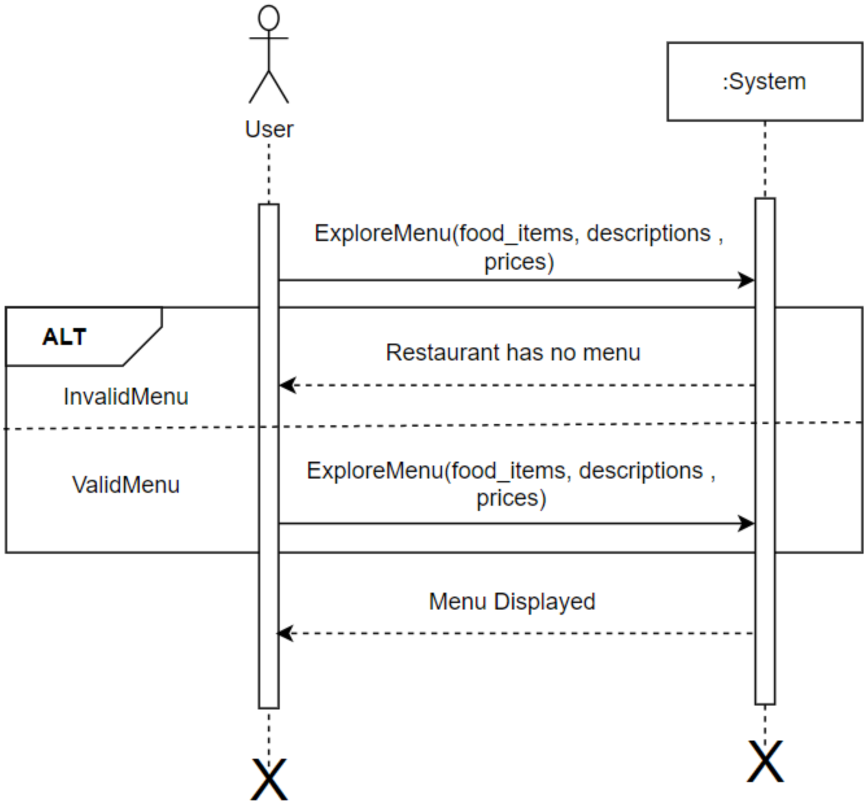
2.1.6.2. User Login


User
:System
login(username, password)
validate
ALT
InvalidCredentials
Invalid username or password
ValidCredentials
login(username, password)
login successful
X
X

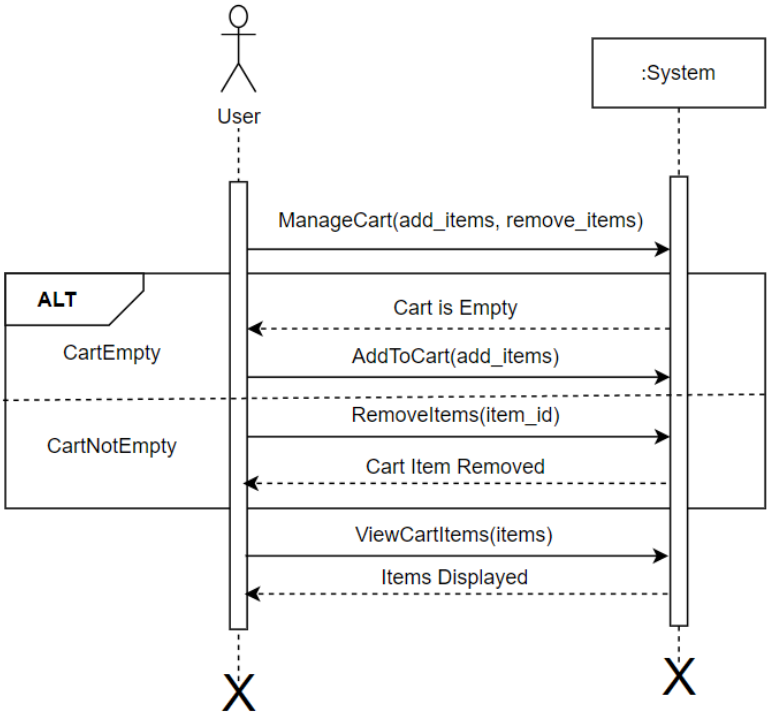
2.1.6.3. Search Restaurants



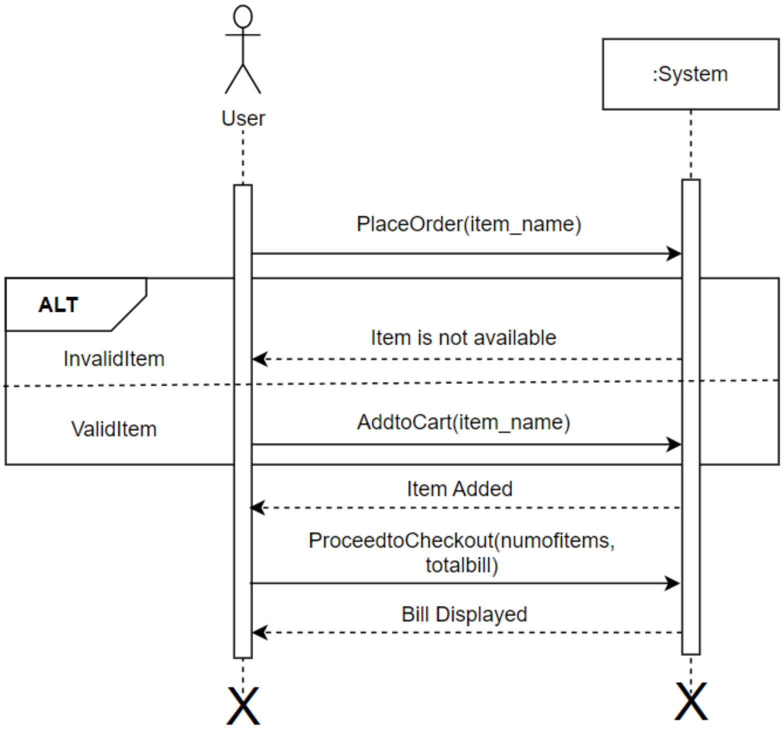
2.1.6.4. Explore Menu


User
:System
ExploreMenu(food_items, descriptions , prices)
ALT
Restaurant has no menu
InvalidMenu
ValidMenu
ExploreMenu(food_items, descriptions , prices)
Menu Displayed
X
X

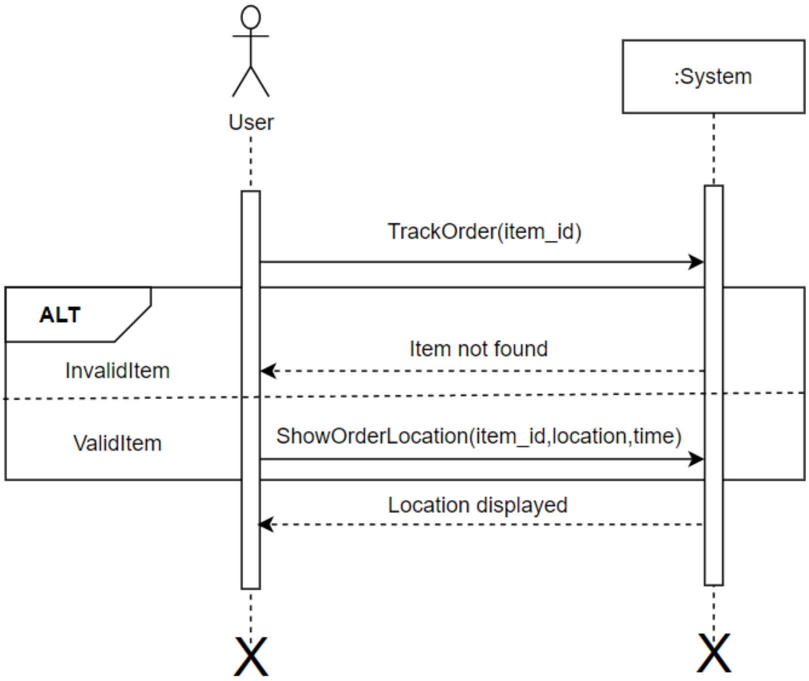
2.1.6.5. Manage Cart



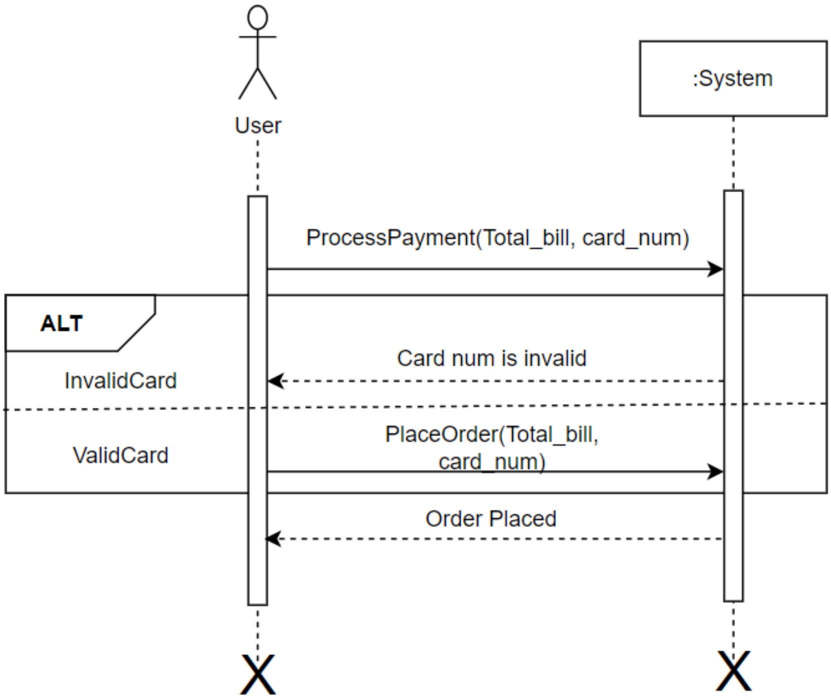
2.1.6.6. Place Order



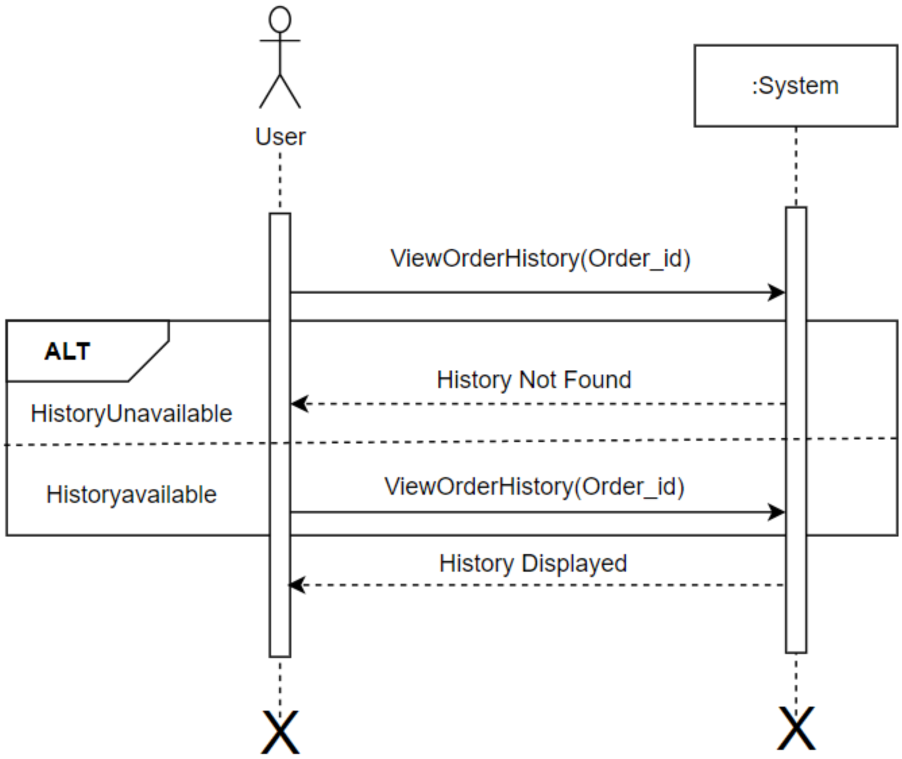
2.1.6.7. Track Order



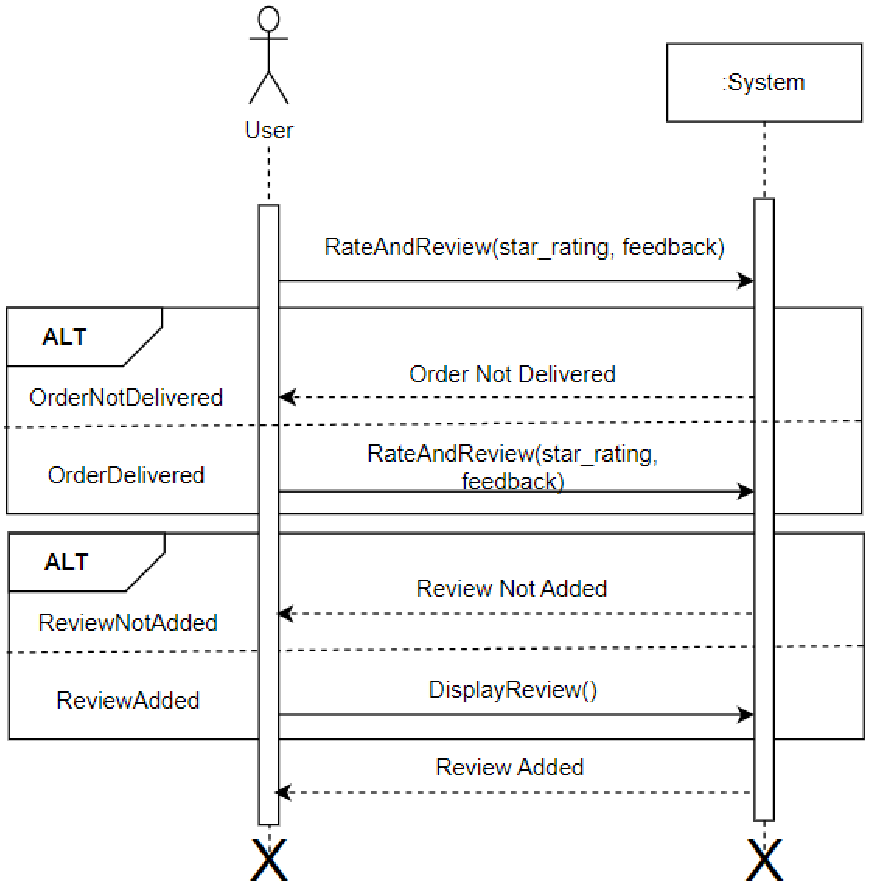
2.1.6.8. Process Payment


User
:System
ProcessPayment(Total_bill, card_num)
ALT
InvalidCard
Card num is invalid
ValidCard
PlaceOrder(Total_bill, card_num)
Order Placed

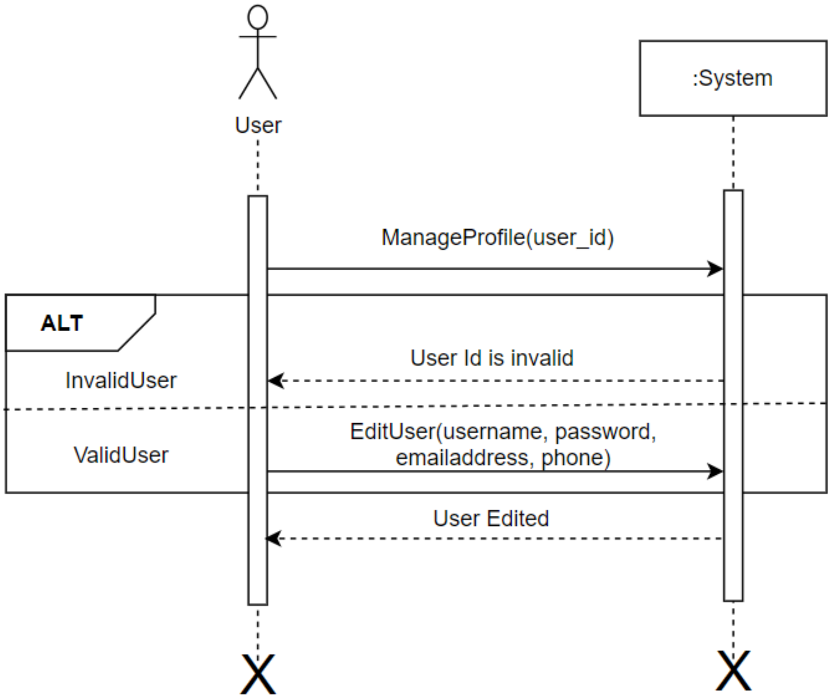
2.1.6.9. View Order History



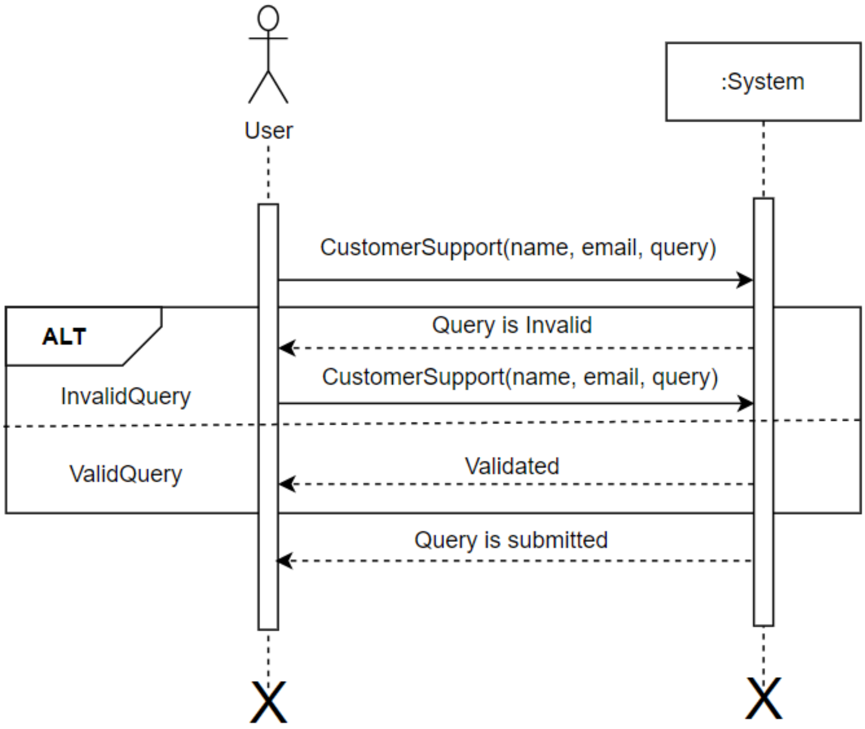
2.1.6.10. Rate and Reviews



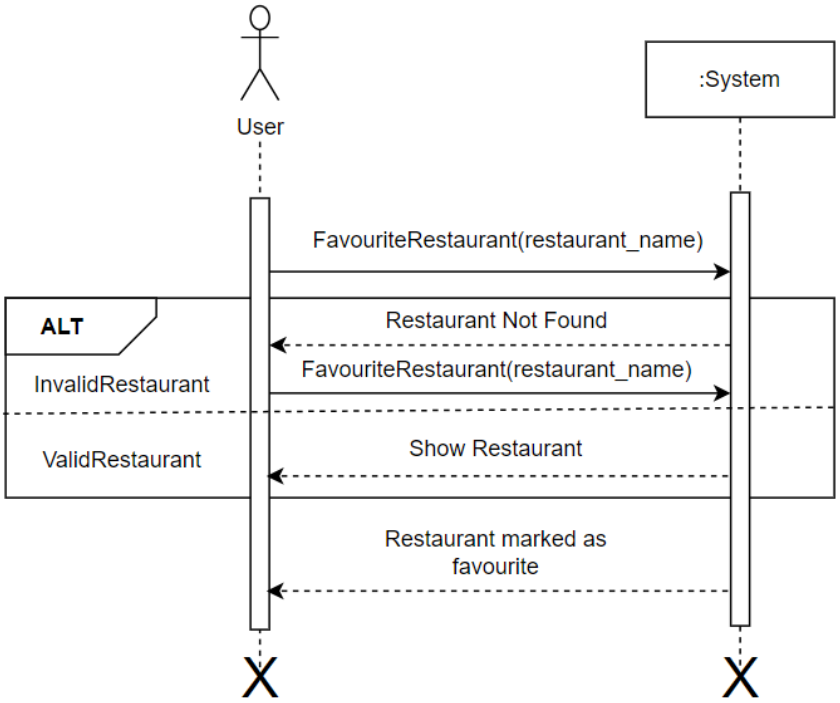
2.1.6.11. Manage Profiles



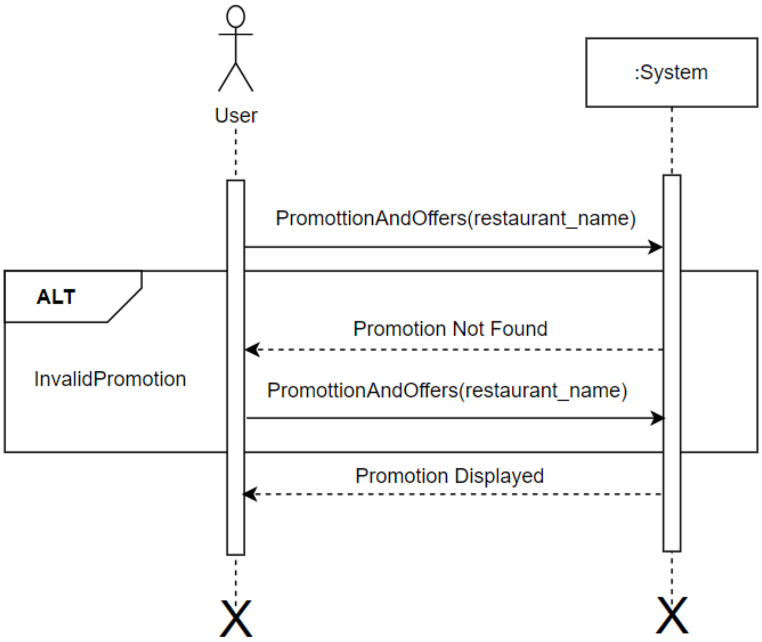
2.1.6.12. Access Customer Support



2.1.6.13. Mark Favorite Restaurants

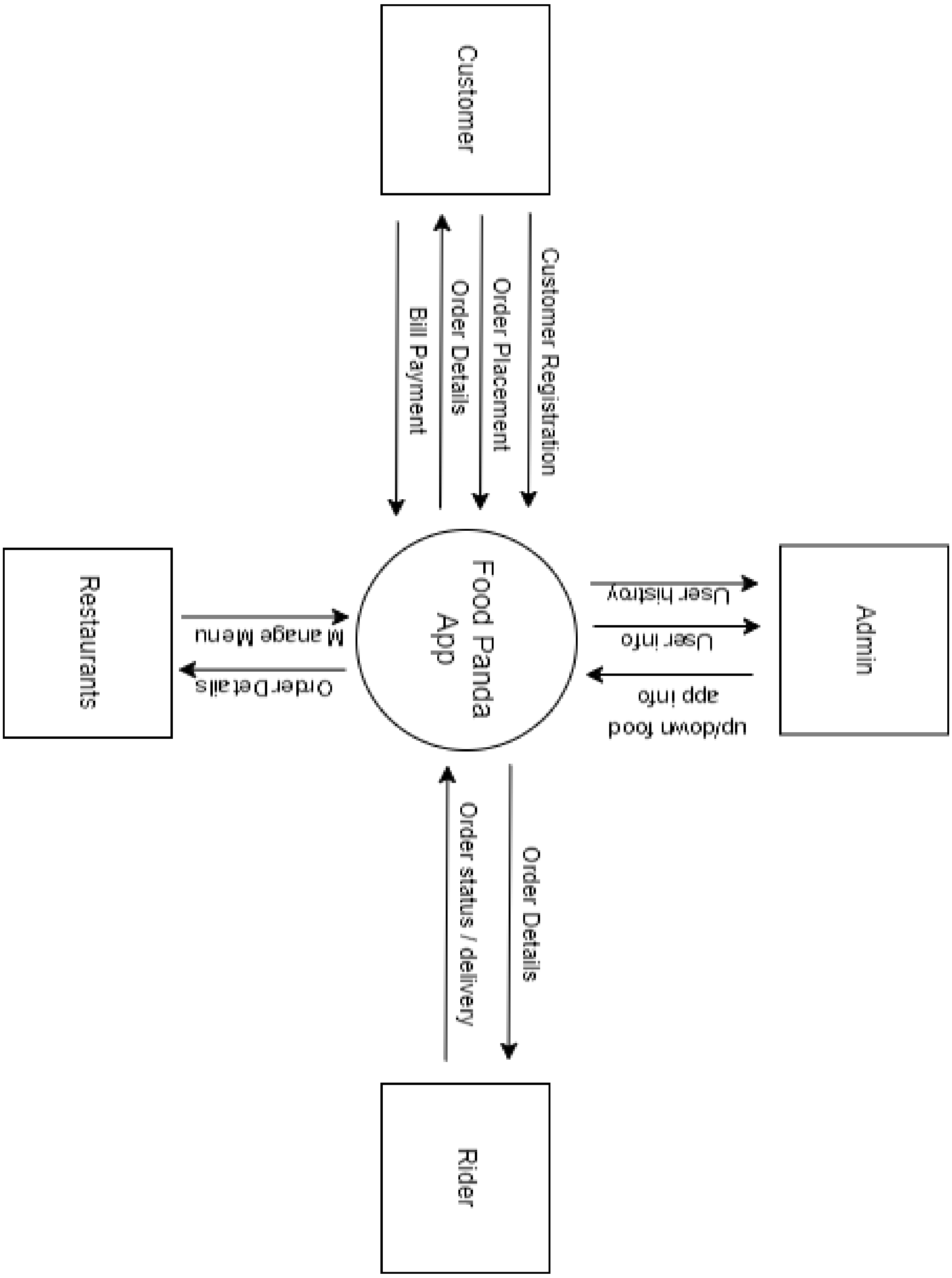

User
:System
FavouriteRestaurant(restaurant_name)
ALT
Restaurant Not Found
InvalidRestaurant
FavouriteRestaurant(restaurant_name)
ValidRestaurant
Show Restaurant
Restaurant marked as favourite

2.1.6.14. View Promotions and Offers

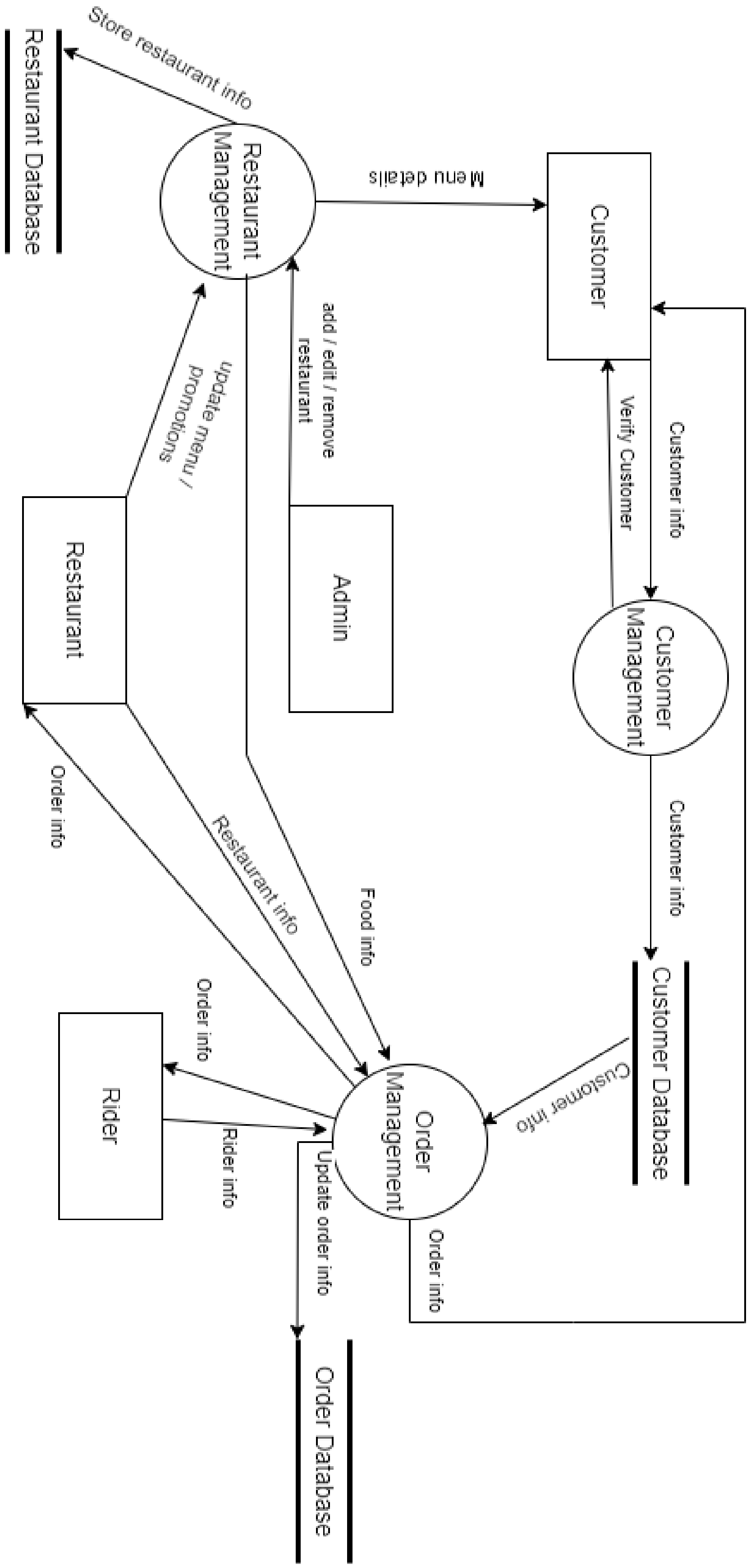

User
:System
PromottionAndOffers(restaurant_name)
ALT
Promotion Not Found
InvalidPromotion
PromottionAndOffers(restaurant_name)
Promotion Displayed

2.1.7.Data Flow Diagram

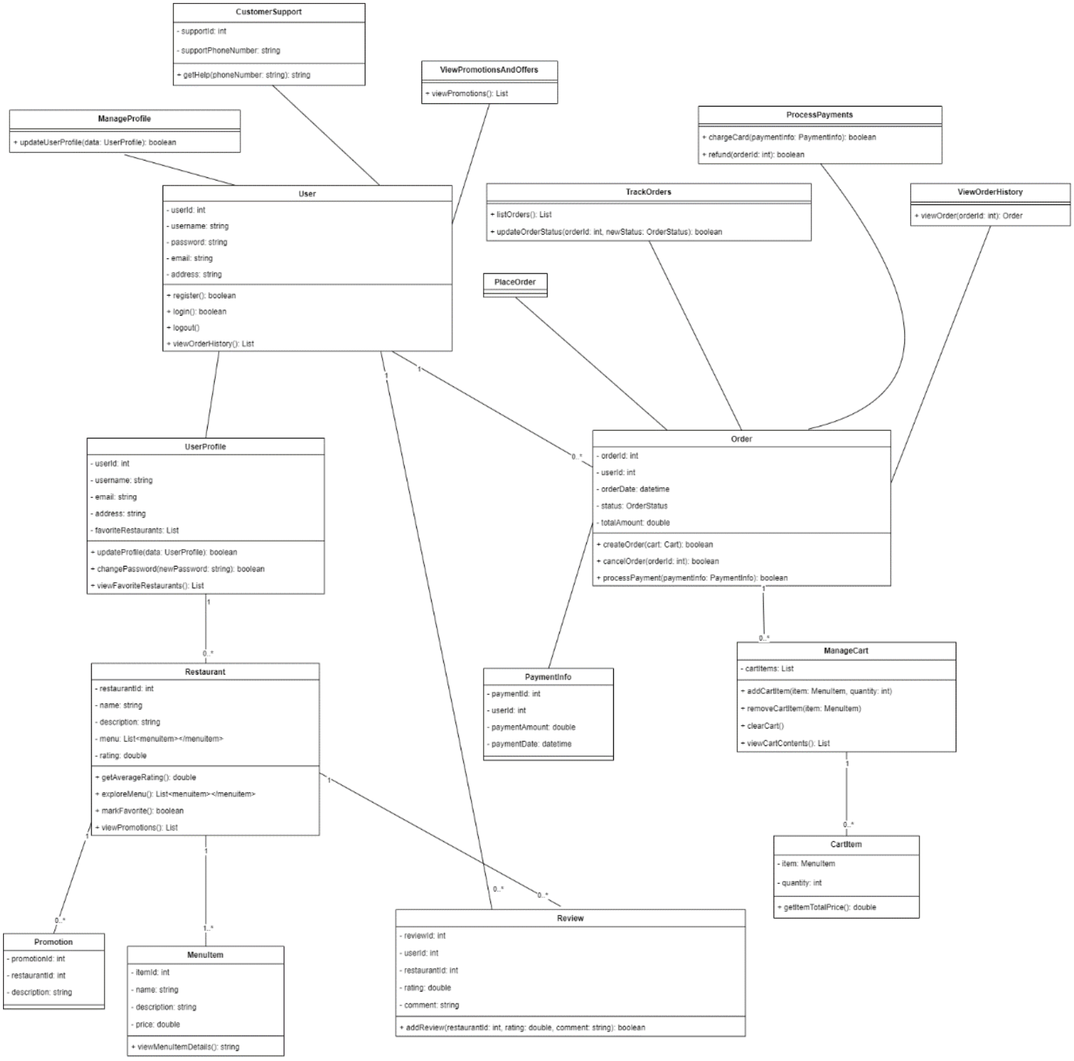
2.1.7.1. Level 0


Customer
Customer Registration
Order Placement
Order Details
Bill Payment
Food Panda App
Restaurants
Manage Menu
Order Details
Admin
User histroy
User info
up/down food app info
Order Details
Order status / delivery
Rider

2.1.7.2. Level 1


Restaurant Database
Store restaurant info
Restaurant Management
Menu details
Customer
add / edit / remove restaurant
update menu / promotions
Restaurant
Admin
Customer info
Verify Customer
Customer Management
Customer info
Customer Database
Customer info
Order info
Restaurant info
Food info
Order Management
Order info
Rider
Rider info
Update order info
Order info
Order Database

2.1.8. Class Diagram



2.2. Architecture Style

For a food delivery app, the most suitable architectural style would be the “Distributed Software Architecture”, specifically the “Client-Server Architecture model”.

2.2.1. Why

Client-Server Architecture:

1) In a food delivery app, you have clients (the users' mobile devices or web browsers) that request information about restaurants, menus, and place orders.

2) Servers can handle these requests, process orders, manage user accounts, and communicate with restaurant partners.

3) The client-server architecture allows for efficient distribution of tasks and data processing. It ensures that the client (user interface) and server (business logic and data storage) can operate independently.

2.2.2. Architecture Style

1. Client Side (User Interface):

- Mobile App or Web App
- User Registration and Authentication
- Restaurant Search and Selection
- Menu Browsing and Item Selection
- Order Placement and Payment
- Real-time Order Tracking
- User Notifications

2. Server Side (Backend):

- Web Servers (for serving the app)
- Application Servers (business logic)
- Database Servers (for data storage)
- User Management:
  - User Authentication
  - User Profiles
- Restaurant Management:
  - Restaurant Listings
  - Menu Management