Software Requirements Specification

for

Kami Foods Mobile Application

Version 1.0 approved

Prepared by

Duc Minh (Felix) Nguyen

Vansh Sethi

Anant Dubey

COMP2920_01

Created 28 Feb 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Felix Nguyen	20 Feb 2023	Initial document build	1.0
Felix Nguyen	21 Feb 2023	Developed user interface	2.0
Vansh Sethi	21 Feb 2023	Researched about the various software architecture	2.0
Vansh Sethi	22 Feb2023	Developed software architecture of the app	3.0
Vansh Sethi	23 Feb 2023	Developed Training plan	4.0
Vansh Sethi	23 Feb 2023	Wrote the software and hardware requirements	4.0
Felix Nguyen	24 Feb 2023	Wrote functional requirements for the app	5.0
Vansh Sethi	24 Feb 203	Wrote non functional requirements for the app	5.0
Felix Nguyen	25 Feb 2023	Segregated software and hardware requirements	6.0
Vansh Sethi	26 Mar 2023	Added more details in the training plan	7.0
Anant Dubey	1 March 2023	Developed Domain dictionary and features to be determined	8.0
Felix Nguyen	02 Mar 2023	Add diagrams	9.0
Anant Dubey	02 Mar 2023	Made Release plan and assumptions/dependencies	9.0
Vansh Sethi	02 Mar 2023	Reviewed Release plan and suggested more key points and other changes.	9.0
Vansh Sethi	02 Mar 2023	Wrote the Terms and conditions	9.0
Anant Dubey	03 Mar 2023	Made the required changes in the release plan as suggested	10.0
Vansh Sethi, Felix Nguyen, Anant Dubey	04 Mar 2023	Reviewed the whole SRS and did minor changes	11.0

1. Introduction

1.1 Purpose

This document was made to note the software requirement specification for the Kami Foods food app, version 1.0. This SRS only covers only the application aspect towards the mass users (user type Customers), not including other major stakeholders taking part in the operation of the business of Kami Foods.

1.2 Intended Audience and Reading Suggestions

This document is intended for project managers, product owners, developers, and testers of the client firm. Involved personnel are expected to understand the purpose of the product and have relevant technical knowledge on mobile application architecture.

The remainder of this SRS provides detailed descriptions of the system design, chosen architecture, and requirements for hardware and software in response to specific requirements documented from the client firm. Requirements are broken down into functional (the What) and non-functional (the How) categories. These are guidelines for the designs and decisions made in the development processes. Later versions may offer suitable changes.

1.3 Product Scope

Kami Foods is a food and beverage delivery application inspired by the globally recognized Uber Eats. Kami Foods aims to localize restaurants in Kamloops and supports students with tasty meals everyday through discounts and promotions exclusively for students. Using the app, customers select the restaurant they wish to order, take a look at menus, select quantities and customize their food, pay, and select the location of pickup.

There are 4 main user classes taking part in the operation of the business's system. Each class will have different user interfaces and versions of the application. During operation, all are connected towards a system of database, backend processing, and notifications.

This SRS focuses on the User's version of the system, consisting of only the mobile application. Relevant external systems include the banking system that provides payment functionalities, and learning institutes' student management system for student verification to be eligible for discounts.

1.4 References

For research purpose:

Software Architecture patterns. Retrieved from
https://www.geeksforgeeks.org/types-of-software-architecture-patterns/

Client-Server Architecture. Retrieved from https://www.sciencedirect.com/topics/computer-science/client-server-architecture

Software Release plans. Retrieved from https://www.peerbits.com/blog/software-release-planning.html

2. Overall Description

2.1 Product Perspective

The boom of mobile technology has directed the trend of users towards smartphone usage at an unprecedented rate. Since the beginning of the late 2010s, businesses have focused more on information delivery to a great number of users via their mobile phones. Taking advantage of this trend, tech giants integrating services onto mobile phone usage have achieved great success in both revenue and profit. One of the most profitable industries of the trend is Food and Beverage. Numerous companies have established a system of food order and delivery through mobile applications. This form of service is becoming the most dominant method of dining thanks to its convenience and simplicity.

2.2 Product Functions

Main features for user class "Customers" include:

- Mandatory login
- Restaurant selection (highly dynamic filters)
- Food ordering (with time-guarantee delivery)
- Payment (Cash, Card)
- Order tracking using Maps systems for courier's real-time location and restaurants in range of User

2.3 User Classes and Characteristic

Code	User Class Name	Description
U1	Customer	Mass users who browse restaurants, order food, and make payments via the application. Customers are the largest number of users expected to sign on. The business and services centralize around bringing more Customers into the system so as to attract revenue.
U2	Courier	This user class consists of business applicants who wish to get food from restaurants and deliver to the corresponding Customers. Couriers are an essential part of the system and must have their GPS and

		4G/5G on at all times. As these personnel are considered employees, it is mandatory that they register both on the app and have face-to-face evaluations at company branches.
U3	Restaurant	Restaurants are the other endpoints of the service. Restaurants are responsible for taking orders from the system and preparing meals. Progress update is also required for each order. Once registered within the system, multiple restaurant employees may sign in with the same Restaurant’s account and perform their functions. Restaurants are also able to post their own promotions/discounts on their page. Each new program must be registered accordingly into the system so as to modify costs accordingly.
U4	Admins	Admins are Kamfood’s business employees who manage all other account types. It is the Admins’ responsibility to: ● Manage, verify, and update relevant information of Couriers, ● Register, verify, train, and manage Restaurant accounts, ● Post Kamfood’s exclusive promotions/discounts on the app, and ● Carry out Customer Service

2.4 Design and Implementation Constraints

- Data security and privacy: To safeguard user data, including payment and delivery information, the Kami Foods app must adhere to data protection laws and use secure authentication and encryption protocols.
- Integration with third-party services: The Kami Foods app may need to connect with a number of third-party services, including payment gateways, delivery tracking systems, and restaurant management systems. Additional development work and possible technical problems could be introduced by integration with these services.
- Performance and scalability: Since a large number of customers will use the Kami Foods app, it is crucial to make sure that it is scalable and capable of handling high traffic levels. To make sure that the app can manage peak usage times, this may necessitate the use of cloud-based infrastructure and load testing.
- Time and financial restrictions: The development team will have to stay within the Kami Foods app's development budget and schedule. To make sure that the most crucial features are created within the allocated timeline and budget, this may require prioritizing features and making trade-offs.

- Platform restrictions: The Kami Foods app's design must take into account the platform, such as Android, iOS for which it is being created. Depending on the platform selected, different design and interface components will be used.

2.5 Assumptions and Dependencies

Assumptions:

- Database already available and secure
- Online banking system is established
- The app will comply with local laws and regulations regarding food delivery services.
- There will be enough demand for the app in the target market.
- Payment gateway integration has been established with major payment providers.

Dependencies:

- GPS systems from iOS and Android
- Sufficient database solutions to hold account info
- Wifi connectivity
- Verified working phone number
- Mobile phone camera system

2.6 Release Plan

Release 1 (First Cycle): (4 weeks)

- Allow the users to create an account
- Helps the users to browse restaurants and menus
- Allows the users to place an order for delivery or pickup
- Users can pay for their order securely through the app
- Provide the basic customer support

Release 2 (Enhancements): (6 weeks)

- Save the users favorite restaurants and dishes
- Provide personalized recommendations based on users` order history
- Allow the users to track their order status in real-time
- Ask the users to rate and review restaurants and dishes
- Provide additional customer support options (e.g., live chat)

Release 3 (Understanding Client View): (2 weeks)

Objective: To understand the client's feedback, needs, and expectations to improve the app's user experience.

Tasks:

- Conduct user interviews and surveys to gather feedback and insights from users.

- Analyze user data to identify areas for improvement and new features.
- Refine the app's design and user interface based on user feedback and data analysis.
- Update the app's content and restaurant listings to reflect user preferences and feedback.
- Implement user-requested features, such as dietary restrictions and allergen filters.

Release 4 (Expansions): (8 weeks)

- Expand the app's coverage area to more cities and regions
- Allow the users to schedule future orders
- Allow the registered restaurants to offer promotions and discounts through the app
- Allow the restaurants to customize their menus and manage their orders through a merchant portal
- Provide more advanced customer support options (e.g., phone support)

Release 5 (Innovations): (10 weeks)

- Introduce a subscription service for frequent users
- Allow the users to order groceries and household essentials through the app
- Integrate with third-party apps and services (e.g., ride-sharing apps)
- Provide more advanced order tracking and delivery logistics features
- Introduce new payment methods (e.g., cryptocurrency, Loyalty points etc.)

2.7 Development Plan

Project Planning and Preparation (1-2 weeks)

During this sprint, the team will establish the project scope, create a product backlog, and identify the initial set of requirements. The team will also set up the development environment, establish the communication channels, and prepare for the upcoming sprints.

User Research and Prototyping (2-4 weeks)

During this sprint, the team will conduct user research to identify user needs and preferences. The team will also create user personas, user stories, and wireframes for the application. By the end of this sprint, the team will have a solid understanding of the user needs and a prototype that can be used for user testing.

User Interface Design (2-4 weeks)

During this sprint, the team will design the user interface for the application based on the wireframes created in Sprint 1. The team will also create a style guide and design assets that can be used throughout the application. By the end of this sprint, the team will have a polished user interface design that is ready for implementation.

Front-end Development (4-8 weeks)

During these sprints, the team will develop the front-end of the application using the design assets created in User Interface Design. The team will also implement user authentication and basic functionality such as browsing restaurants and menus, adding items to the cart, and placing orders. By the end of these sprints, the team will have a functional front-end that can be used for user testing.

Back-end Development (4-8 weeks)

During these sprints, the team will develop the backend of the application, including the database, APIs, and server-side logic. The team will also implement payment processing and order fulfillment. By the end of these sprints, the team will have a functional back-end that can be integrated with the front-end developed.

Testing and Bug Fixing (4-8 weeks)

During these sprints, the team will conduct extensive testing to identify and fix any bugs or issues in the application. The team will also perform user acceptance testing to ensure that the application meets the user requirements. By the end of these sprints, the team will have a fully functional application that is ready for deployment.

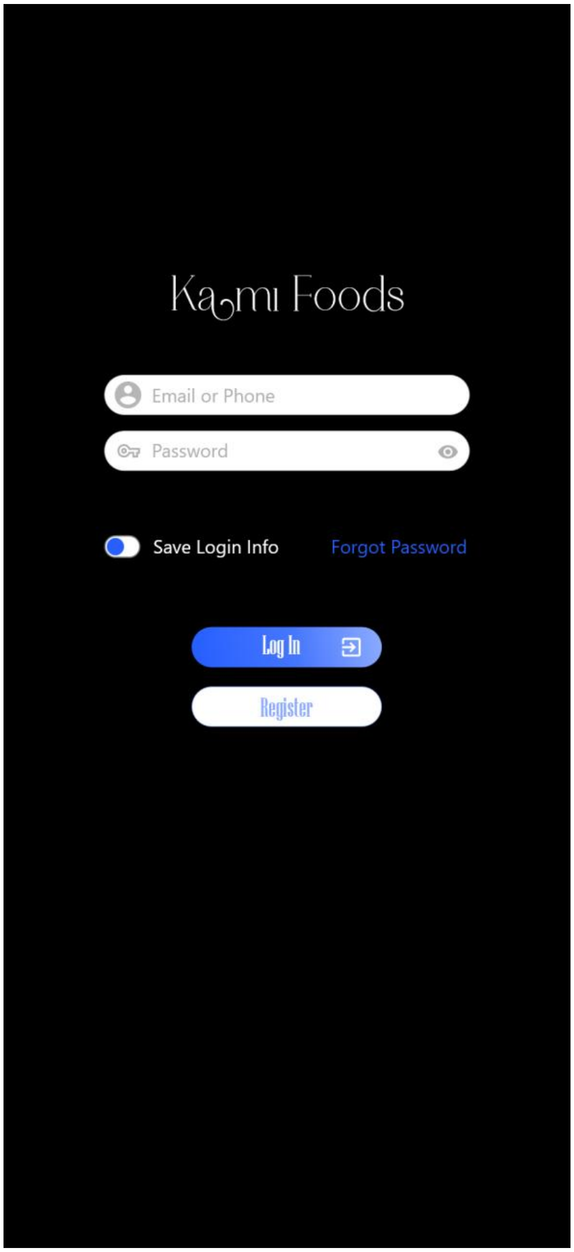
Deployment and Launch (1-2 weeks)

During this sprint, the team will deploy the application to a production environment and perform final testing to ensure that the application is working as expected. The team will also prepare for the launch of the application, including marketing and user acquisition efforts. By the end of this sprint, the application will be launched and available to the public.

.

3. External Interface Requirements

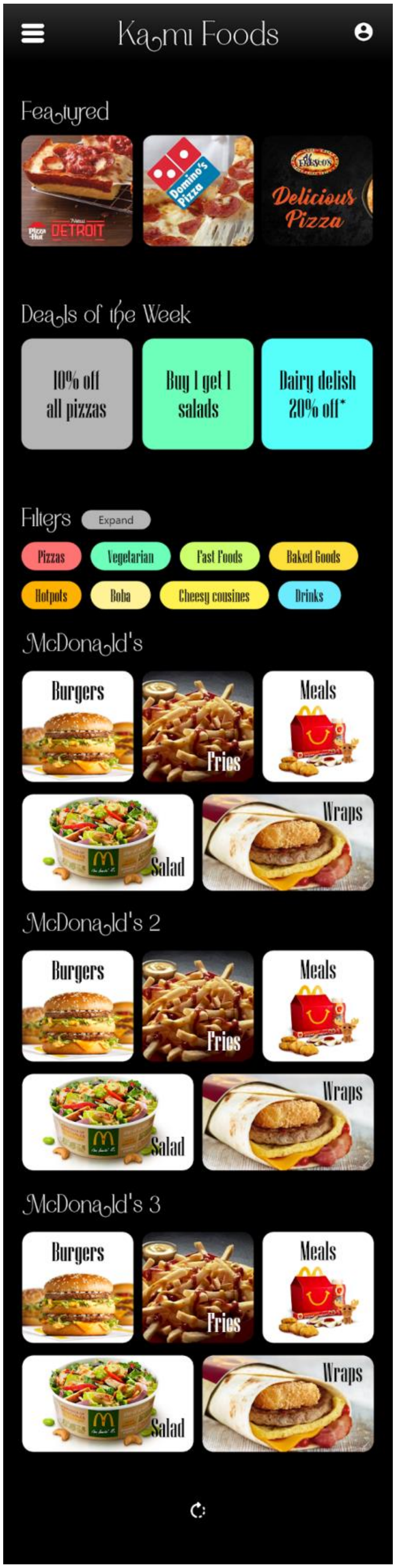
3.1 User Interfaces



UI 1: Customer Login screen

When first opening the app after installation on mobile phones, Customers are prompted with this login screen.

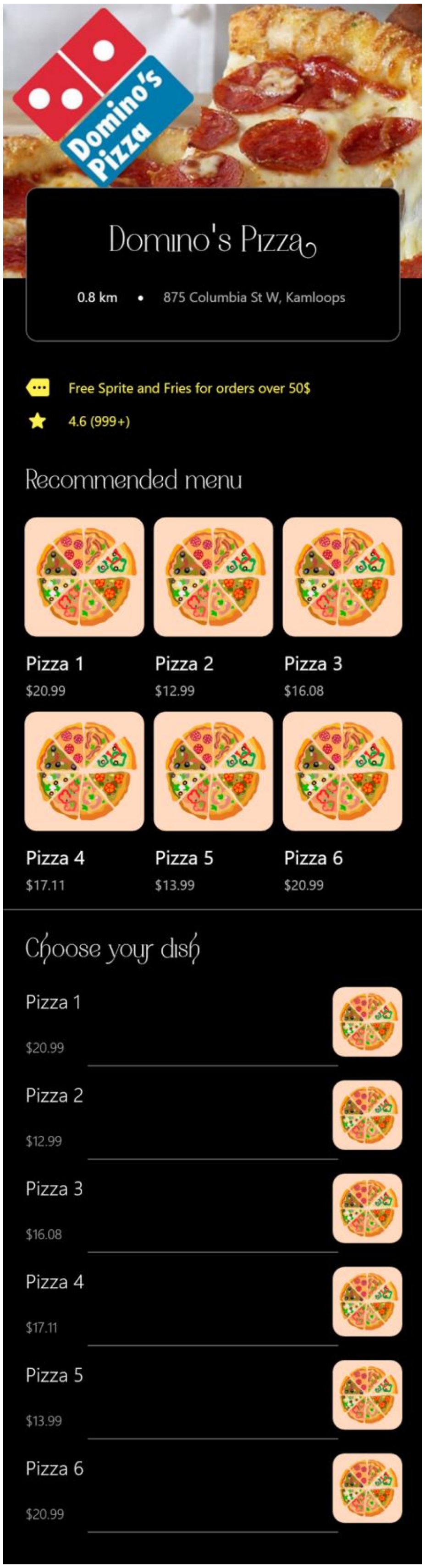
Customers must create accounts before being able to use other features of the app.



UI 2: Home screen

After successfully creating an account and logging in, Customers arrive at the Home screen, which displays a variety of Restaurants, Deals, and Promotions.

Customers may also apply the application’s filters and searches to have more customized dining options.



Domino's Pizza

0.8 km • 875 Columbia St W, Kamloops

Free Sprite and Fries for orders over 50$

4.6 (999+)

Recommended menu

Pizza 1
$20.99

Pizza 2
$12.99

Pizza 3
$16.08

Pizza 4
$17.11

Pizza 5
$13.99

Pizza 6
$20.99

Choose your dish

Pizza 1	$20.99
Pizza 2	$12.99
Pizza 3	$16.08
Pizza 4	$17.11
Pizza 5	$13.99
Pizza 6	$20.99

UI 3: Restaurant Menu screen

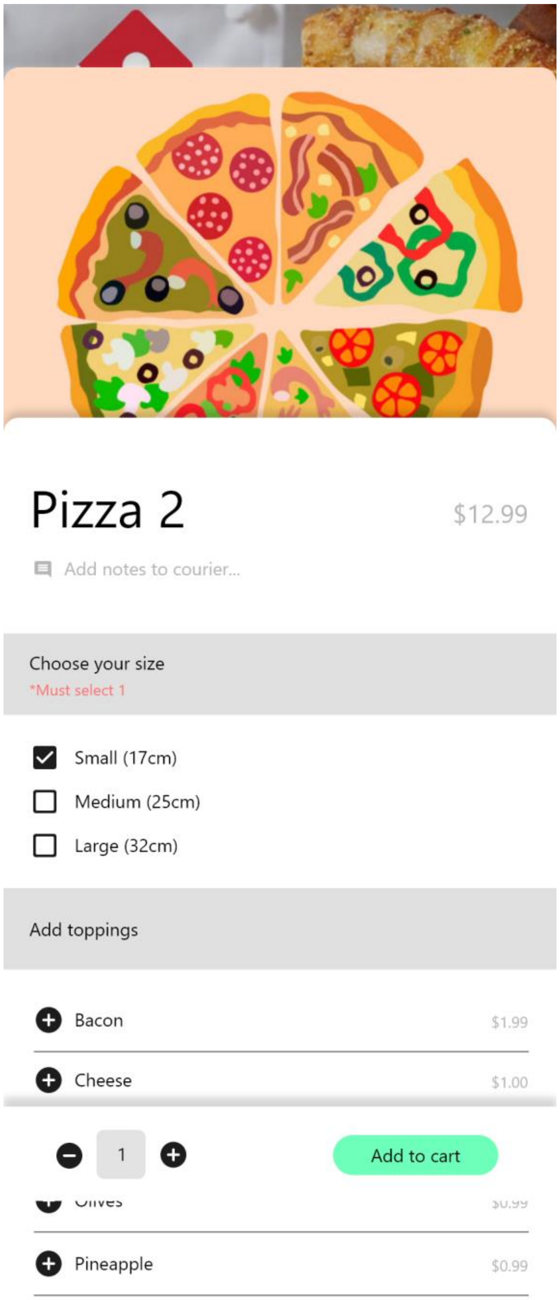
Registered Restaurants will have their own screen that they can customize their menus. Designs follow specific structures set out by the system. This screen is viewed from the perspective of Customers ordering food. Here, Customers can browse and select, make an order, and customize their dishes.

Note: Brands wishing to have their pages displayed in unique styles may contact Kamfood’s business department for detailed discussion and additional charges.

UI 4: Ordering screen

After selecting a desired item, the ordering screen pops up for that specific item. Customers are asked to select mandatory relevant customizations and quantities. Other optional modifications to dishes are added by Restaurants, but not necessary for Customers to choose.

The process ends when Customers add items to cart.



Pizza 2 $12.99

Add notes to courier...

Choose your size

*Must select 1

☑ Small (17cm)

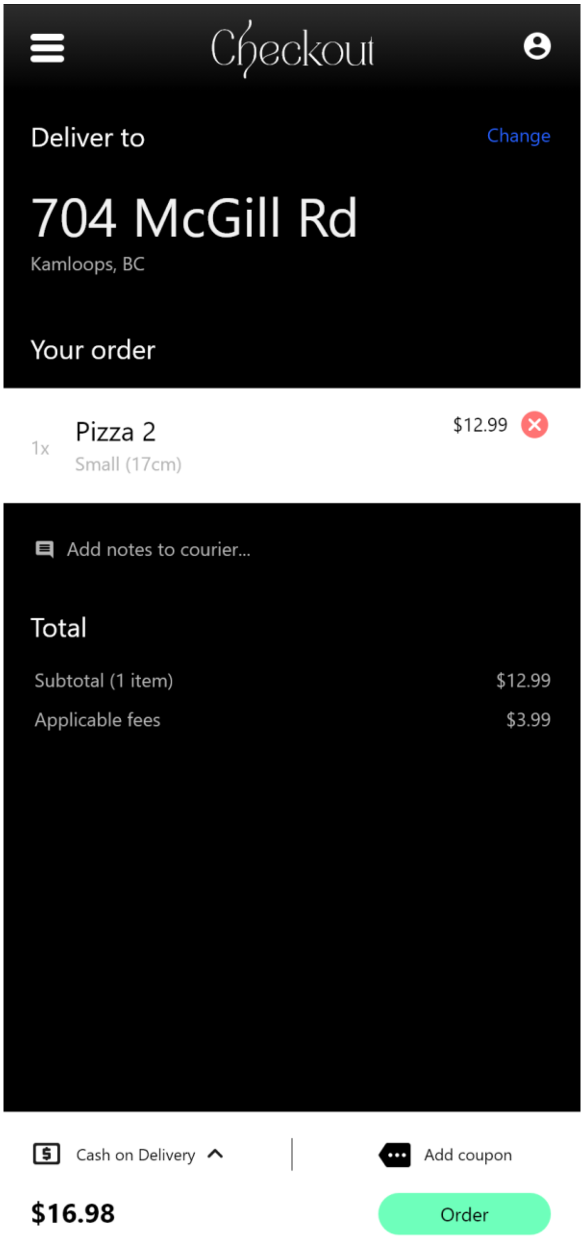
□ Medium (25cm)

□ Large (32cm)

Add toppings

⊕ Bacon	$1.99
⊕ Cheese	$1.00
⊕ Olives	$0.99
⊕ Pineapple	$0.99

⊖ 1 ⊕ Add to cart

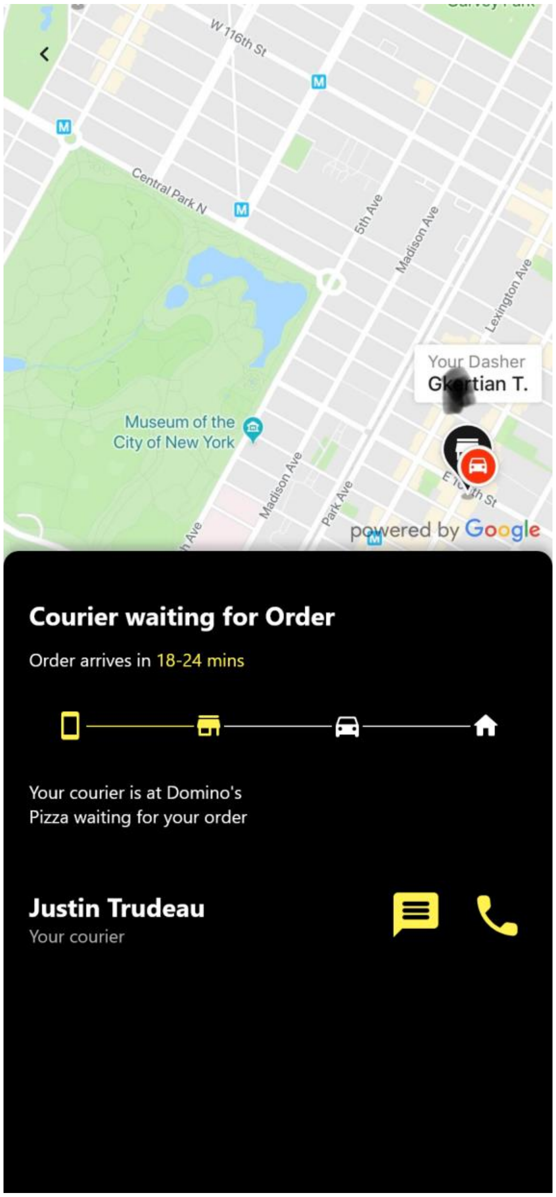


UI 5: Checkout screen

After finishing adding all items to cart, Customers may proceed to Checkout. Here, Customers can review their selections one last time, remove items if necessary, and see the overall price for their orders.

Add Coupon option is also available for eligible Customers.

The process finishes when users hit the Order button.



Courier waiting for Order

Order arrives in 18-24 mins

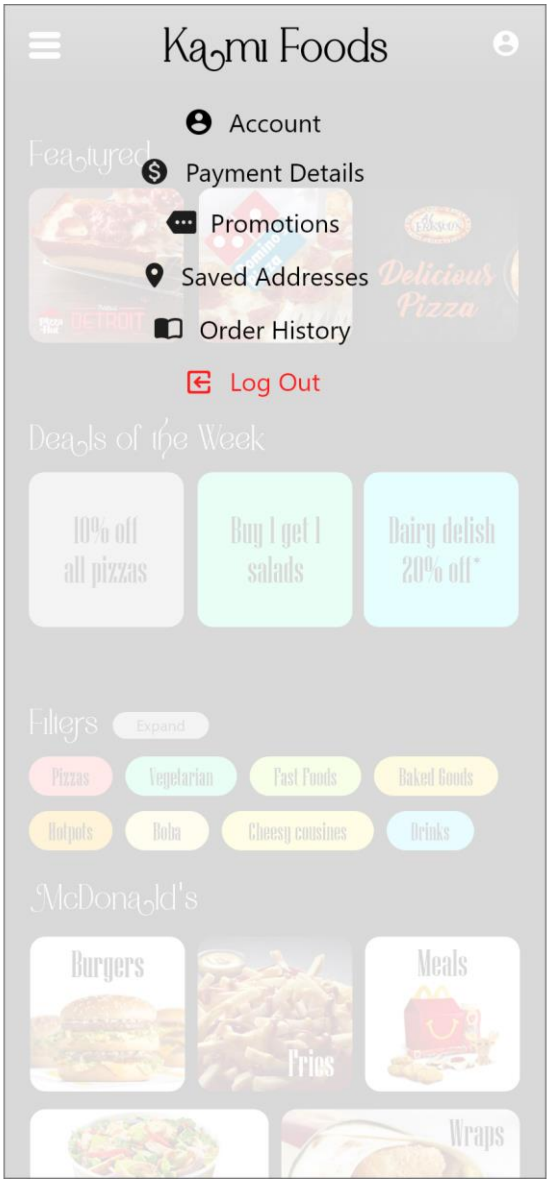
Your courier is at Domino's Pizza waiting for your order

Justin Trudeau

Your courier

UI 6: Order Tracking screen

After placing orders and payments have been processed, Customers can track their orders on the map. Progress is also shown so that users know which phase their order is in.



UI 7: Side user menu

Clicking on the three bars on the top left of the app, the Side User Menu will appear for Customers to navigate to other sections of the app.

3.2 Hardware Requirements

For Android:

- At least 1.5GB of RAM
- Internet connectivity (WIFI or cellular data)
- Screen resolution of at least 1280 x 720 pixels

For IOS:

- At least 1.5GB of RAM
- iPhone 5S or later
- Internet connectivity (WIFI or cellular data)
- Screen resolution of at least 1136 x 640 pixels

Additions for both Android and iOS:

- A rear-facing camera with autofocus and flash is recommended, to enable features such as barcode scanning or taking photos of food orders.
- A front-facing camera is also recommended, to enable features such as user profile pictures.
- The device should have a reliable internet connection, with at least 4G data speeds or access to WIFI.
- The device should support Bluetooth 4.0 or higher, to allow for features such as contactless payments or connecting to Bluetooth-enabled devices.
- The device should support NFC, to enable features such as mobile payments.

3.3 Software Requirements

For iOS:

- A device running iOS 13 or later is recommended, to ensure compatibility with the latest features and security updates.
- At least 60 MB of free storage space

For Android:

- A device running Android 30 or later is recommended, to ensure compatibility with the latest features and security updates.
- At least 50 MB of free storage space

3.4 Software Architecture

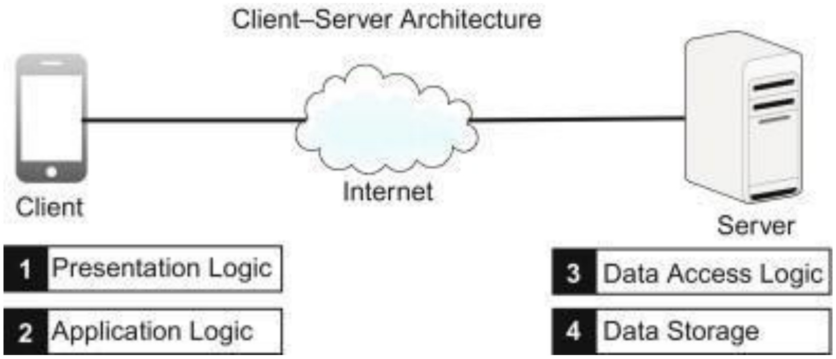
Kamfood’s app is built using a client-server architecture. Our app is split into two components in this architecture: a client-side application and a server-side application.

The end user's device (such as a smartphone or tablet) will run the client-side application, which will provide an interface for users to interact with the app, place orders, monitor deliveries, and carry out other operations. Data such as order information, user information, and menu data will be sent and received by the client-side application through communication with the server-side application.

The backend logic of our program will be handled by the server-side application, which will also be in charge of handling user accounts, processing orders, storing data, and carrying out other tasks. The client-side program will send data to the server, which will process it, and send a response back to the client-side application.

For example, when a user places an order using the client-side application, the client-side application will send a request to the server-side application with the details of the order. The server-side application will process the request, which may involve updating a database with the order information and generating a confirmation message. The server-side application will then send a response back to the client-side application with the confirmation message, which will be displayed to the user.

Overall, the client-server architecture will allow for the separation of concerns between the client-side application and the server-side application, so each component can be developed and maintained independently. This can make it easier to update and improve the app over time thereby providing a robust and scalable foundation for Kami Foods.



There are various other reasons why using a client-server architecture for our app is a good approach:

1. Real-time order tracking: Users must be able to receive updates regarding the status of their orders in real-time through our app. Given that the server-side application can handle the processing and storing of order data and offer real-time updates to the client-side application, a client-server architecture can help ensure that this information is updated promptly and accurately.

2. Secure payment processing: Since that users will be disclosing sensitive financial information; our app needs to be able to manage safe payment processing. The server-side application can process payments and offer secure payment data storage by employing a client-server architecture.

3. Complex business logic: Our app will include complex business logic, such as handling promotions and discounts or calculating delivery prices based on location or the time of day. The server-side application may manage these complicated computations and make sure they are carried out consistently and accurately by establishing a client-server architecture.

3.5 Communications Interfaces

For operations on the app, HTTP protocol will be used for all internet-based communication between the client-side application and the server-side. The client-side application will use the HTTP protocol to make a request to the server-side application whenever the user interacts with the app on the client-side. The request will be handled by the server-side application, which may entail retrieving information from a database, running calculations, and producing a response. The client-side application will receive the response via the HTTP protocol and adjust the user interface as necessary.

Contacting between Customer Service and Courier and Customers will be done via normal SMS and MMS protocols.

3.6 Training Plans

TRAINING PLAN FOR USERS

An in-app training for the Customers is available the first time Customers sign in the app or when users choose the Help option. The training demo is in the form of live, step-by-step software-controlled demonstrations that will walk customers through different screens. The goal of the training will be to show consumers how to place food orders, pay, use coupons, and track deliveries. Customers will learn where to click and how to use each feature after following this tutorial.

The training tutorial will only last 30 seconds in total. This training concentrates on the key features and capabilities of the software. This is due to the fact that most users lack the patience or free time to sit through protracted training sessions. Also, a condensed training schedule lowers the chance of confusing users with too much information, which can be detrimental and make the app seem extremely difficult to use.

Some of the key features of the training will include: -

The user will only receive training the first time they use the app, and it will be as basic and user-friendly as possible to guarantee that users can easily navigate through it and place orders without any problems.

1) Restaurant navigation: The training will start by teaching visitors how to look up certain restaurants they enjoy.
2) Ordering: Following that, it will demonstrate how to use the app to place an order, including choosing products from the menu, selecting a delivery or pick-up option, establishing the delivery address, and, if desired, entering any cooking instructions.
3) Payment: The app will take users through the process of making payments, including entering credit card information, applying any available coupons, discounts, or promos, and confirming their orders.
4) Tracking delivery or progress: It will show end users how to monitor the delivery or development of their order. Customers can look up the anticipated delivery window and track the delivery driver's location.

5) Customer care service: At the end users will be guided with how to use customer care help option in order to report any grievances and how they can provide the feedback on their experience using the app, including rating the quality of the food, the accuracy of the order, and the overall service provided.

TRAINING PLAN FOR DEVELOPERS

Our app development process will have the 5 days training for the developers as follows that will provide brief information regarding the basic motive and functionality of the app. This training will be more of the planning for the development of the application. The concepts discussed on each day of training are as follows:

Day 1

- A description of the Kami Foods delivery app's characteristics
- a summary of the project's aims and objectives
- Overview of the technology stack and app architecture

Day 2

- Reviewing the app's user interface and user experience in great detail
- explanation of the main features and usage cases for the software
- Evaluation of the data access layer and database schema for the application
- An introduction to the design and implementation of the app's API

Day 3

- A summary of the security criteria and implementation for the app
- An explanation of how payments are processed by the app and how it integrates with payment gateways
- An introduction to the capabilities for processing orders and tracking deliveries in the app
- Evaluation of the performance and scaling needs for the

Day 4

- Outline of the testing and quality assurance plan for the application
- Description of the framework and test suites used for the app's automated testing
- Overview of the pipeline for continuous integration and deployment of the application
- Description of the monitoring and logging requirements for the application

Day 5

- Discussion of potential areas for improvement and further development
- Q&A session to address any remaining questions or concerns
- Wrap-up and next steps for the development team.

4. System Features

4.1 Login

4.1.1 Description and Priority

This feature corresponds to UC1 in section 5. Login is a mandatory stage for all users before being able to carry out their functionalities. Therefore, it is of **High** priority and must be developed with great attention to details. Section 6 specifies the non-functional requirements applicable for Login.

4.1.2 Stimulus/Response Sequences

- For Customers, the first time accessing the app after registration, the mobile application prompts the login screen.
- Customers stay logged in until sign out or there are changes made to the account from Kamfood's office.

4.1.3 Functional Requirements

REQ-UC1-01: Customers shall have options of Remembering Login Information
REQ-UC1-02: Customers shall be able to Register accounts on unregistered phone numbers only.
REQ-UC1-03: Customers must always log in, no guest users are allowed

4.2 Browse Food

4.2.1 Description and Priority

This feature corresponds to UC2 in section 5. Customers can begin to browse foods in restaurant menus and deals after successfully logging in. This use case is the first step to setting up a food order. This feature is **High** quality and must be developed for the first release.

4.2.2 Stimulus/Response Sequences

- Customers only reach this feature after successfully registering an account and performing verifications.

4.2.3 Functional Requirements

REQ-UC2-01: System displays Deals and Promotions accordingly to set up by Admins
REQ-UC2-02: Restaurants are ranked to display from top to bottom
REQ-UC2-03: There is a Featured and Recommended Restaurant section in the top part of the screen

REQ-UC2-04: Customers shall be able to apply filter at the home page to get the most suitable restaurants
REQ-UC2-05: Restaurants in the systems shall be sorted into specific categories to be filtered by Customers
REQ-UC2-06: Nutrition information is displayed in all dishes, provided by restaurants

4.3 Order Food

4.3.1 Description and Priority

This feature corresponds to UC3 in section 5. After making selections of what to eat, Customers move to the ordering phase. Here, the cart of items is confirmed, Customers choose the pickup location, and be presented the amount to pay. The flow can not continue if User cancels during this feature. It is of **High** priority.

4.3.2 Stimulus/Response Sequences

- Customers only proceed into payment after choosing options in this stage of the flow.

4.3.3 Functional Requirements

REQ-UC3-01: Customers shall be able to customize their dishes before adding to cart
REQ-UC3-02: Quantity must be selected before adding to cart
REQ-UC3-03: Customers shall be able to choose utensil types at checkout
REQ-UC3-04: Customers shall be able to choose delivery address, or pickup from restaurants to reduce delivery costs.

4.4 Make payment

4.4.1 Description and Priority

This feature corresponds to UC4 in section 5. Payment is where transactions are carried out between Customers and business. This feature requires more precision than speed. Business talks with banks must be carried out in great detail. Connecting Kami Foods system to banking transactions is of **High** priority.

4.4.2 Stimulus/Response Sequences

- Customers are prompted with payment methods after checking out.
- Cash on Delivery requires no extra processing time in the system, but only notifies the Courier
- Payment by Cards requires confirmation of success before moving the order on to preparation stages.

4.4.3 Functional Requirements

REQ-UC4-01: Customers shall be able to choose their coupon to calculate a new subtotal
REQ-UC4-02: Customers shall be able to input the amount to tip for Couriers before payment

REQ-UC4-03: Multiple payment methods is available to choose from
REQ-UC4-04: Payment info such as cards can be saved at this stage
REQ-UC4-05: Student discounts are always automatically deducted from subtotal, but shown to Customers
REQ-UC4-06: Online receipt shall be sent to Customers' email address to confirm order

4.5 Track Order

4.5.1 Description and Priority

This feature corresponds to UC5 in section 5. Order tracking is a user-friendly bonus feature to have on the application. Its features help create a better food ordering experience, and provides convenience for both clients and businesses. However, as it is only a bonus after other crucial steps, it has **Medium** priority in the development process.

4.5.2 Functional Requirements

REQ-UC5-01: Customers shall be able to track location of their order's Courier
REQ-UC5-02: Order status is updated by Restaurants in the food preparation process
REQ-UC5-03: Customers shall be able to contact their Courier either by phone call or SMS, options provided in-app.
REQ-UC5-04: Notifications shall be sent to Customers for every milestone reached in the order preparation and delivery process
REQ-UC5-05: With penalties applied, Customers shall be given the chance to cancel orders the Courier picks up food from Restaurants.

5. Use Case Diagrams

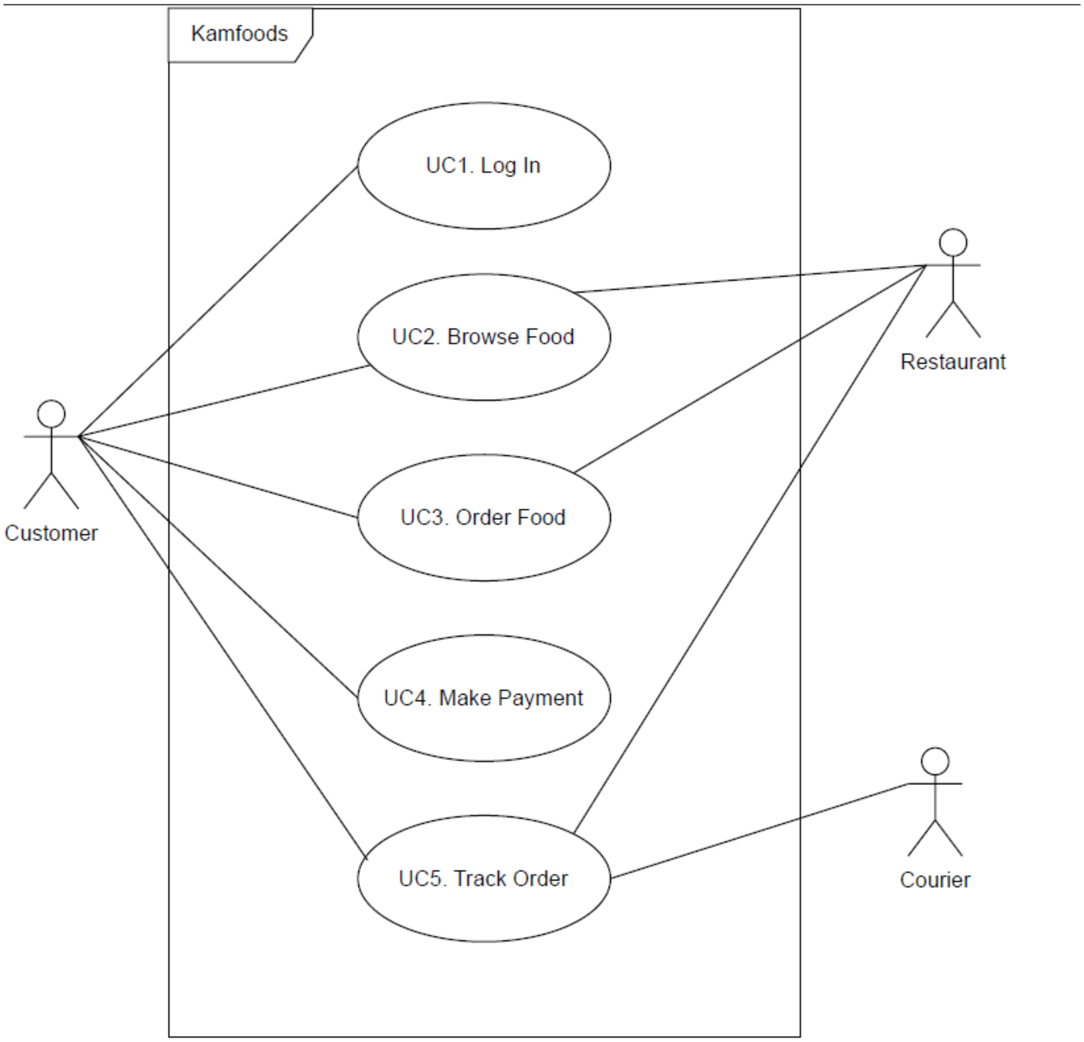


Figure 5.1: General Use Cases

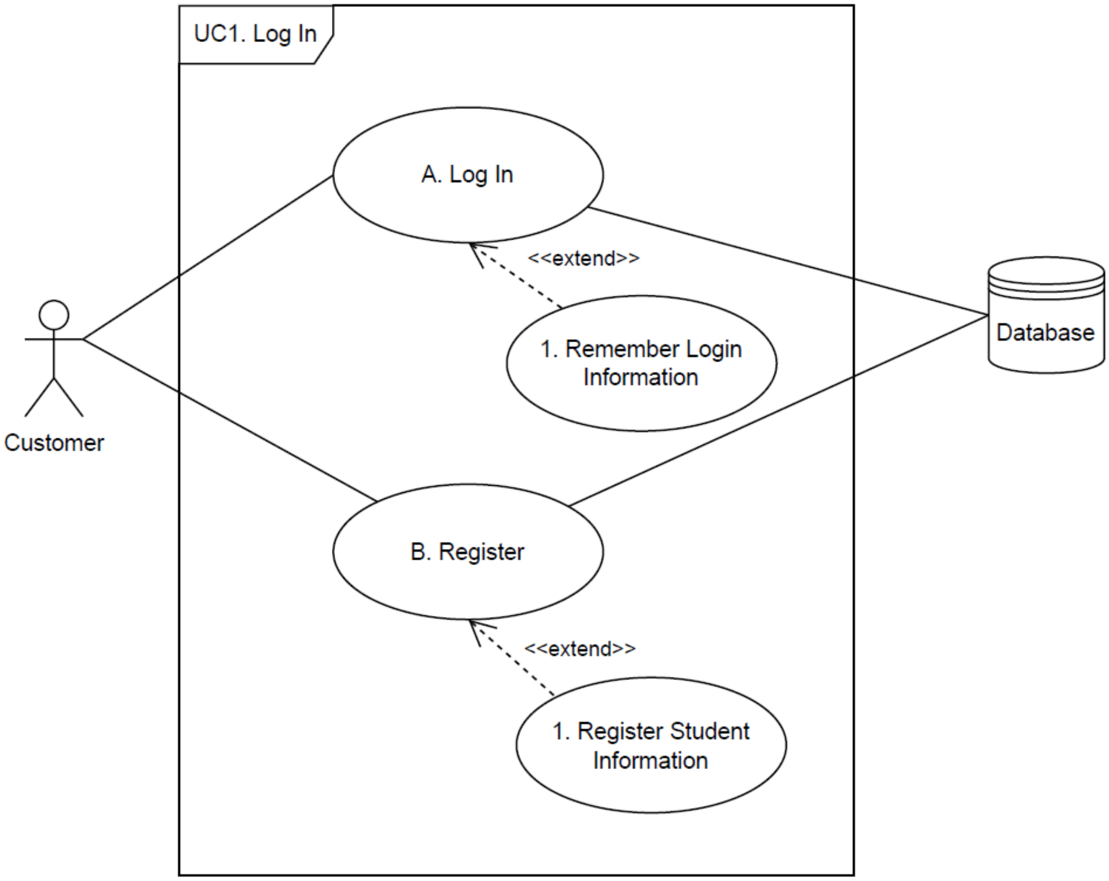


Figure 5.2: Use Case 1 - Log In

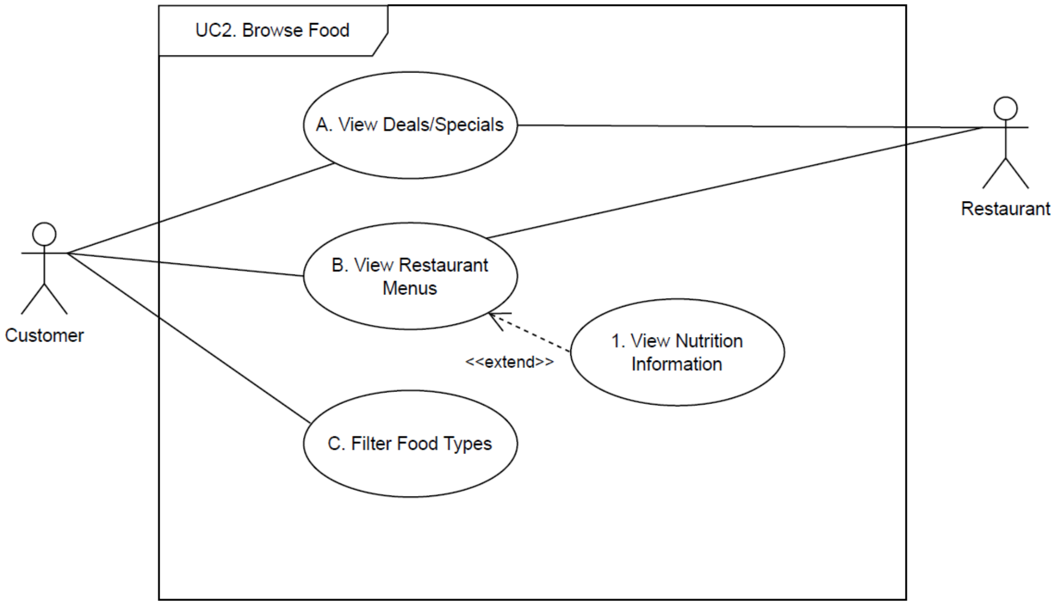


Figure 5.3: Use Case 2 - Browse Food

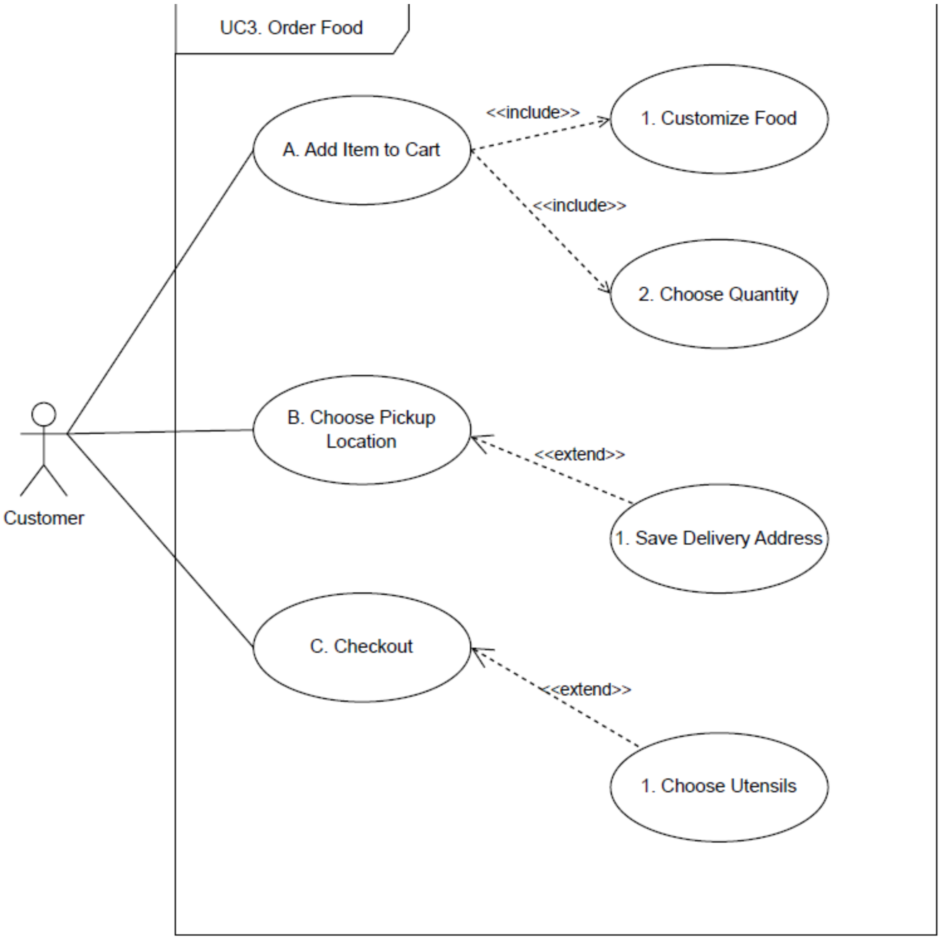


Figure 5.4: Use Case 3 - Order Food

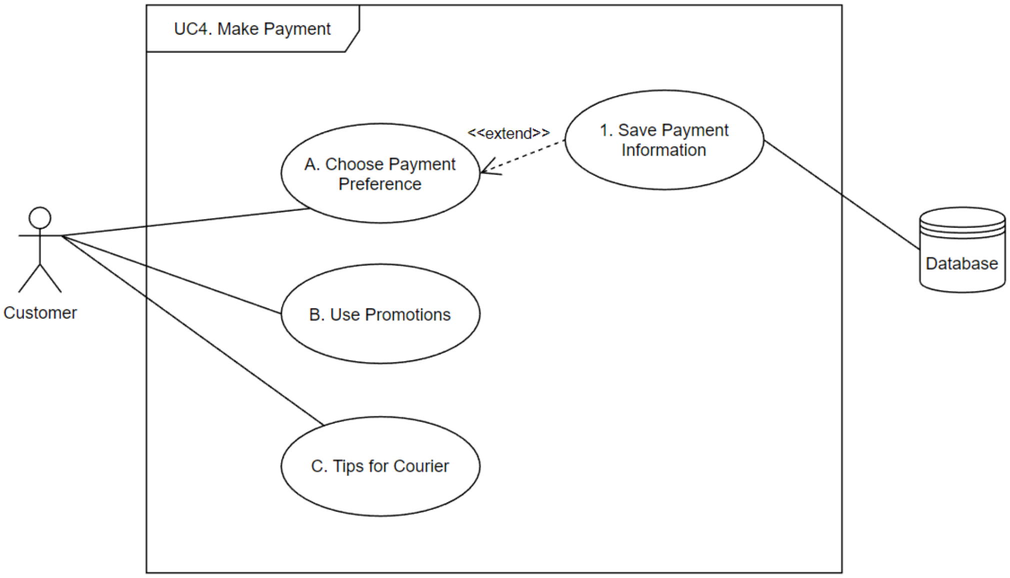


Figure 5.5: Use Case 4 - Make Payment

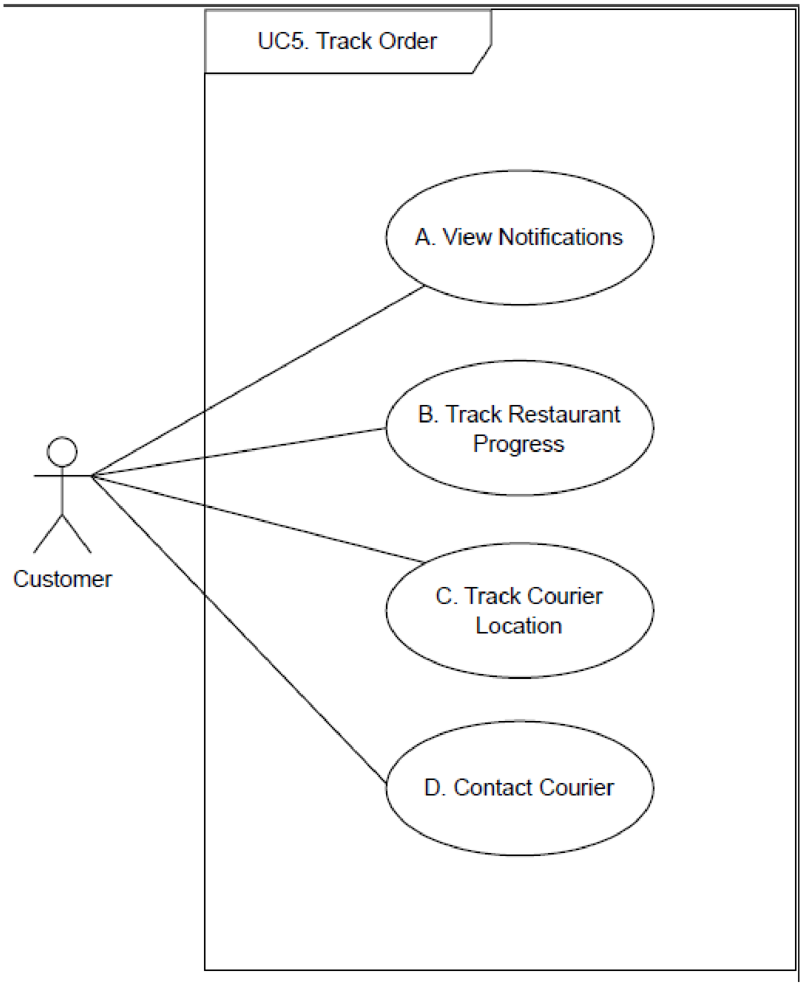


Figure 5.6: Use Case 5 - Track Order

6. Non-functional Requirements

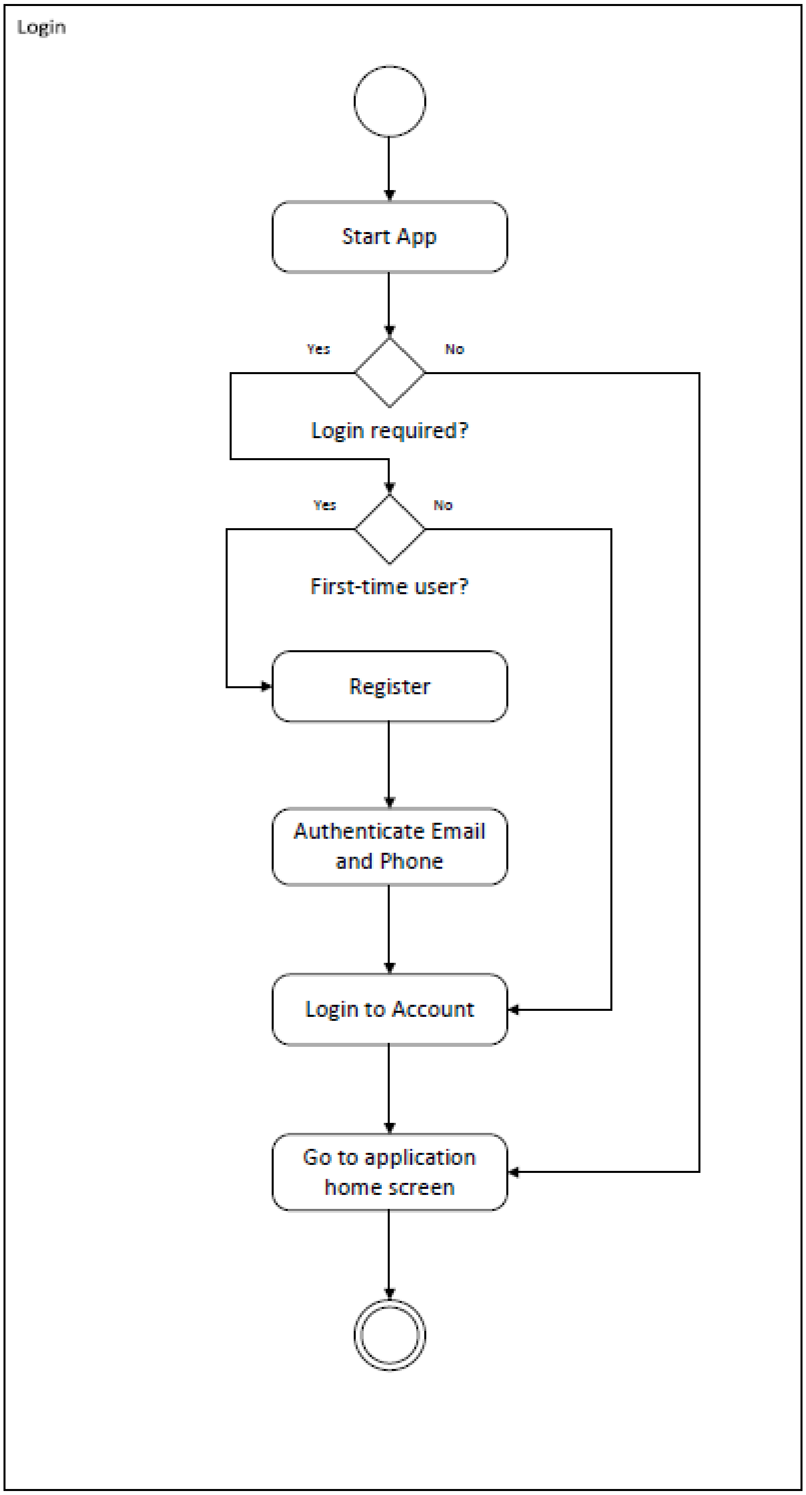
Requirement no.	Type	Description
NFR001	Fast Delivery	The app shall provide fast delivery options that allow users to receive their orders within a reasonable timeframe. The app shall provide a discount of 50% on the total order value if the order is not delivered within 30 minutes of the order time (this will be applicable only if the delivery is within 5km of distance radius).

NFR002	Food/Delivery Status Updates	The app shall provide food status updates to users, showing the progress of their orders in real-time. The delivery status updates shall be updated by the restaurant and delivery partners end-to-end. Also the customers who have selected the delivery option of the food shall be able to see the live location of the delivery person who is coming to deliver the food to them.
NFR003	Container Options	The app shall provide users with the option to select the type of container they would like their food to be delivered in, including recyclable containers that are free and plastic containers that may incur additional charges.
NFR004	Promotions and Discounts	The app shall provide users with various promotions and discounts, including general discounts, student discounts, and seasonal promotions. The promotions and discounts shall be applied automatically to eligible orders. Student discount will only be applicable in pick-up orders in which student need to show their student card in order to avail it
NFR005	Payment	The app shall support various payment options like credit cards, debit cards, net banking, e-wallets, and cash on delivery. The users shall be able to make payment in a separate window using two-step verification by entering the OTP received on the phone number and also by approving the payment on Gmail. There will be an option to remember the card details of the user so that they need not to enter the card details again and again each time they order the food.
NFR006	Nutrition Information Display	The Kami Foods app shall display nutrition information, including calories, fat content, protein content, and other relevant information for each food item on the menu. The app shall also update nutrition information regularly to ensure accuracy and relevancy for customers. Also, the app shall provide customers with the ability to view a summary of their order's nutrition information before checkout to make informed decisions about their food choices.

NFR007	Filter Options	The app shall provide users with filter options that allow them to sort and search for food items based on various criteria, including price, vegetarian/non-vegetarian, and sort by franchise of the restaurant, distance, food type, etc. The app shall also provide customers with the option to filter food items based on nutritional criteria, such as low calorie or low fat.
NFR008	Scalability	Depending on the number of transactions and traffic, the Kami Foods app shall be able to scale up or down without degrading performance. As the user base expands, the app shall be able to support a growing number of restaurants and menu items.This app will be using the amazon web services for that.
NFR009	Feedback mechanisms	Customers shall be able to provide feedback on their experiences through the app, which will enable the business to gradually enhance its offerings. They shall be able to score their overall satisfaction with the app, delivery efficiency, meal quality, and customer service using a straightforward rating system that will be put in place. Moreover, customers can provide written comments outlining their scores.
NFR010	Cooking Instructions	The app shall have the options to select the cooking instructions for the people ordering food. Also To ensure that their needs and preferences are met, the app should allow users to add custom cooking instructions to their orders. There shall also be an option for the user to specify any food allergies they have so that the restaurants can take care while making the food for them. Some of the cooking instruction option shall be included are Crispy cooked, Light cooked, Double-cut, Gluten-free, etc.

7. Activity Diagrams

Figure 7.1: Login flow from Customer standpoint



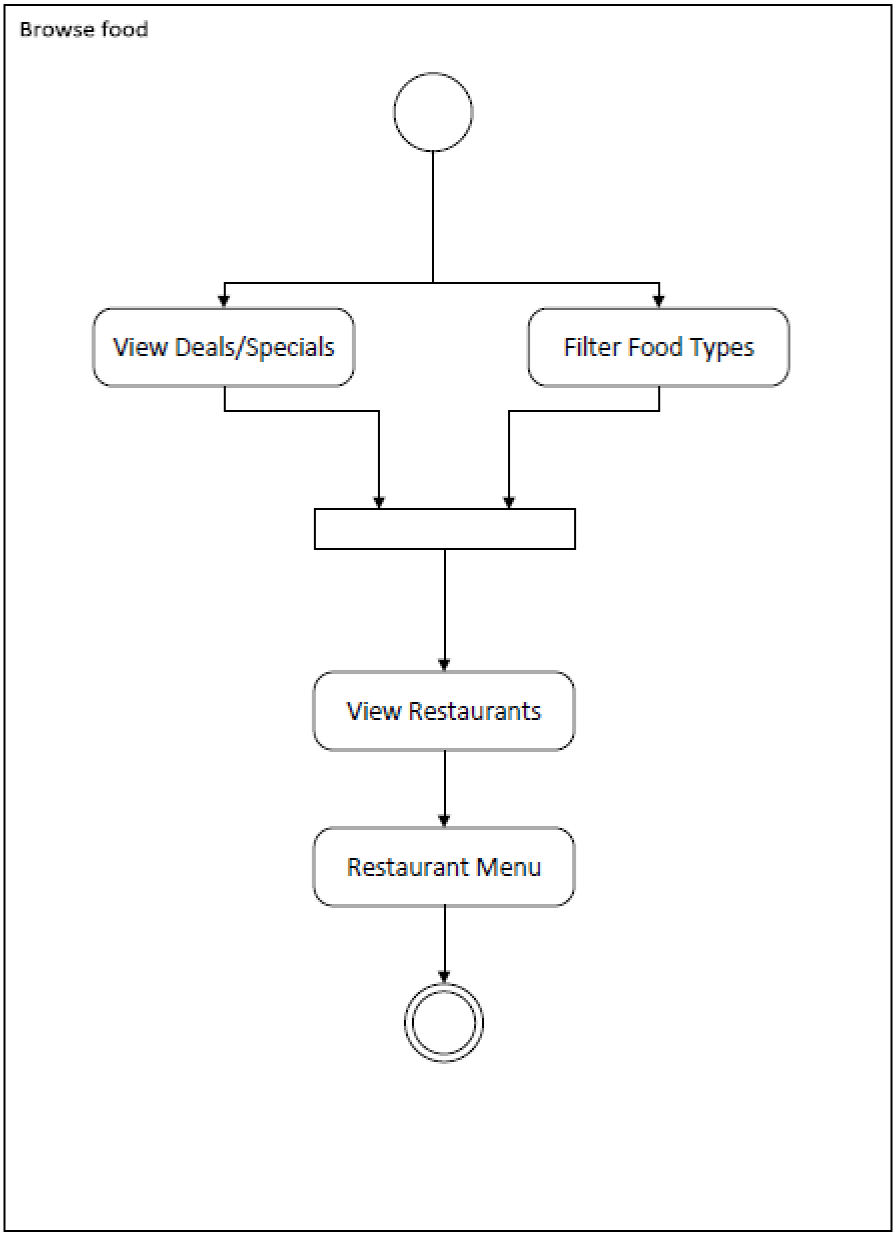


Figure 7.2: Browse food flow from Customer standpoint

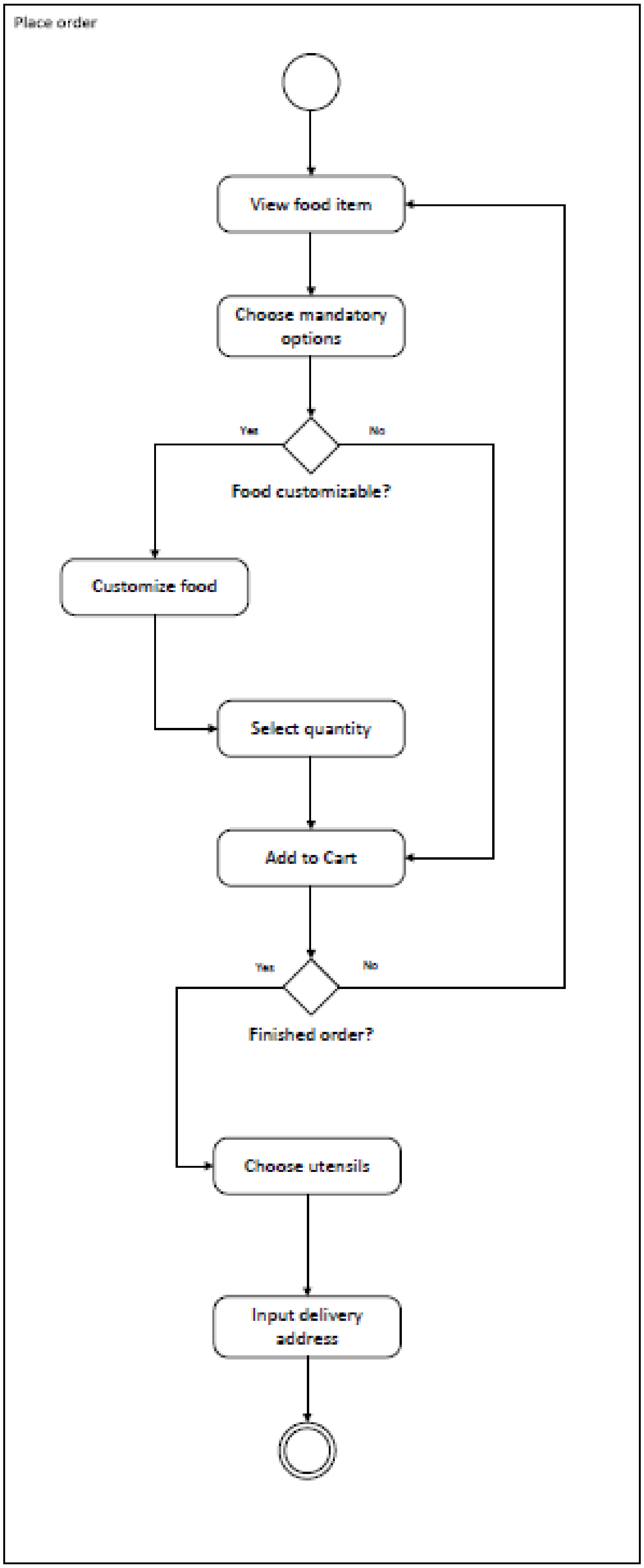


Figure 7.3: Place order flow

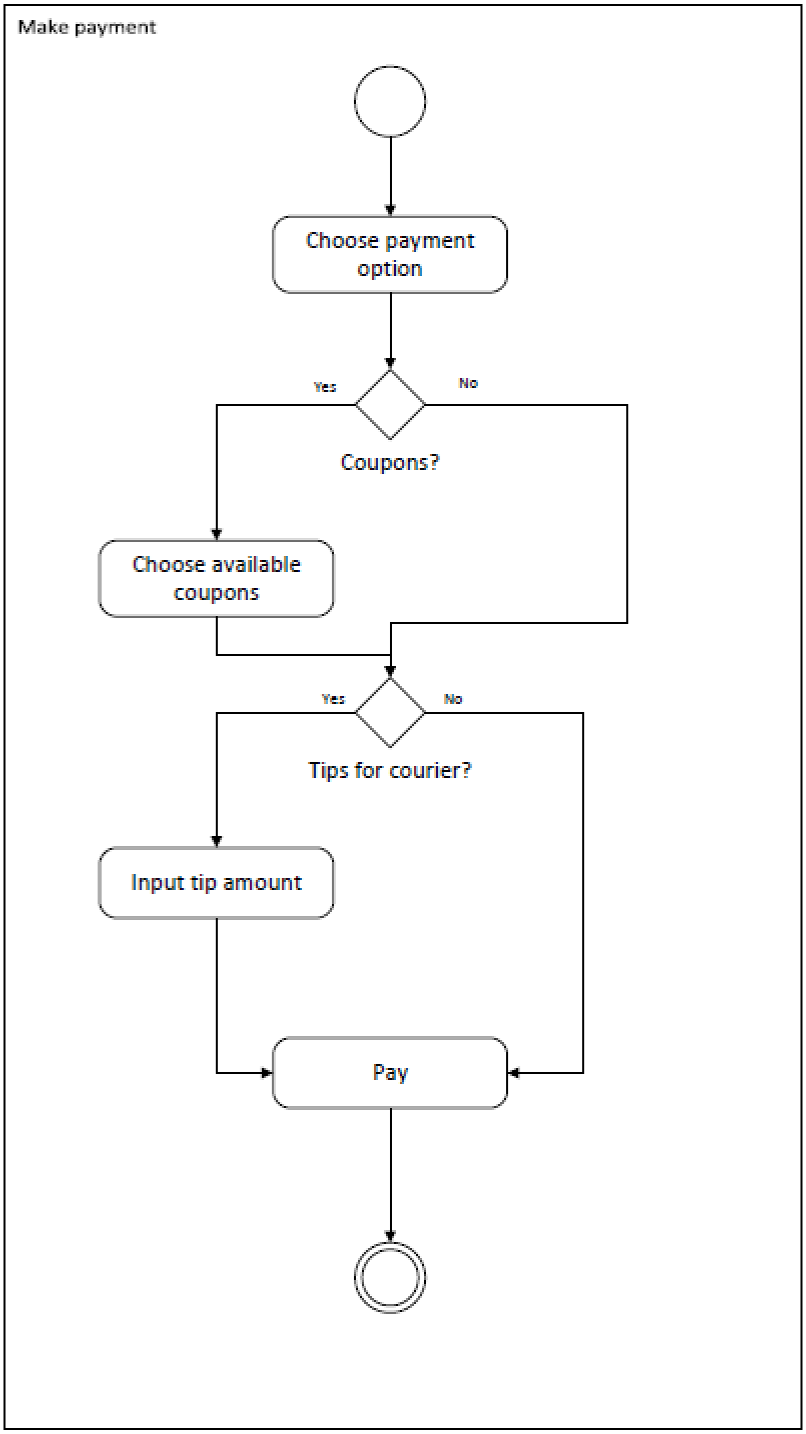


Figure 7.4: Payment flow

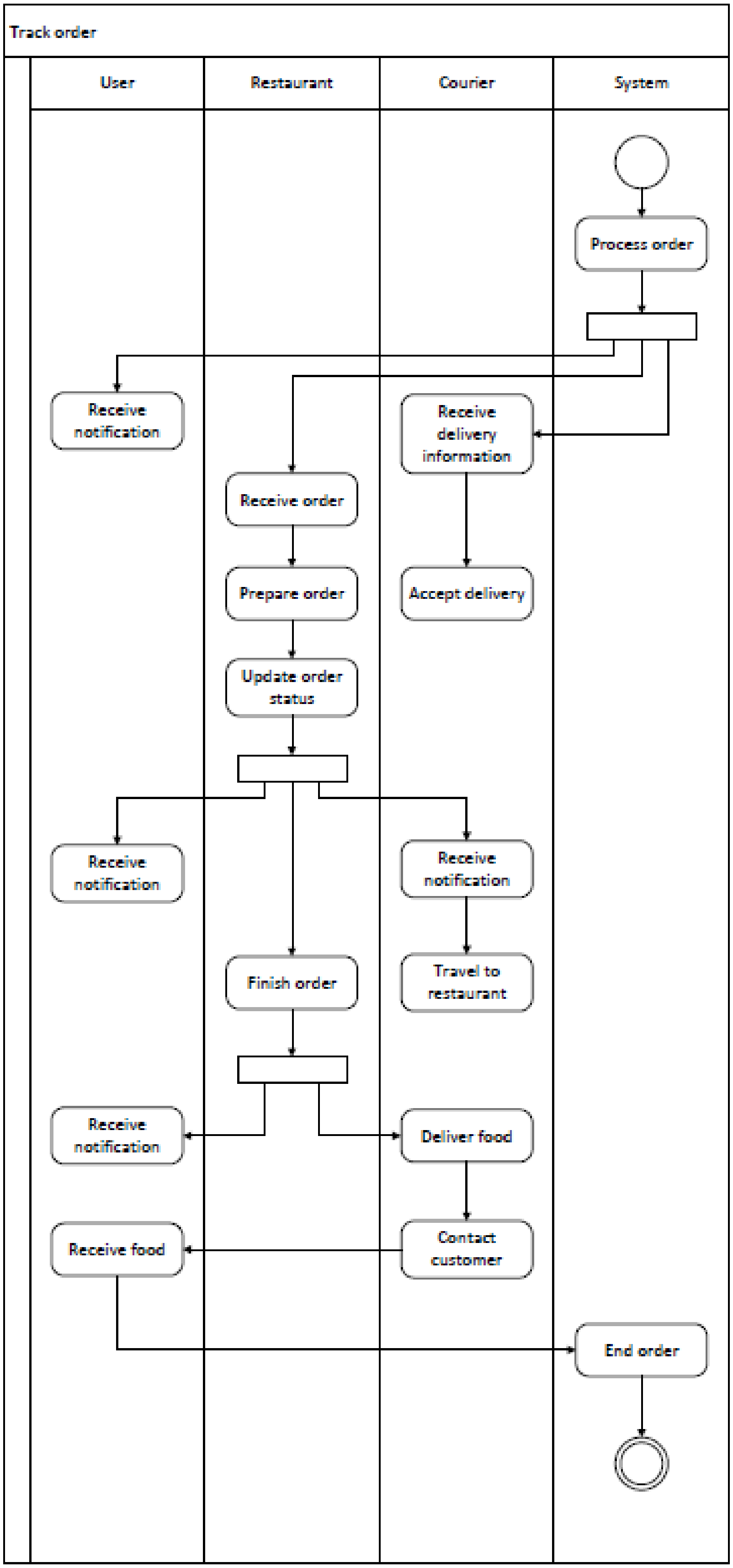


Figure 7.5: Order Processing flow

8. Other

Appendix A: Domain Dictionary

Term	Definition
Sign-up	The process of creating a new account in the app, typically requiring the user's name, email address, phone number, and payment information.
Sign-in	The process of logging into the app using an existing account, typically requiring the user's email address or phone number and password or other authentication method.
Sign-out	The process of logging out of the app, typically initiated by the user to end their session and prevent unauthorized access to their account.
Profile	A section of the app that displays and allows users to manage their personal information, such as their name, email address, phone number, payment information, and order history.
Restaurant	A business that offers food and drinks for delivery or pickup through the app, typically with a menu, pricing, and ratings and reviews from other users.
Menu	A list of food and drink items offered by a restaurant through the app, typically with prices, descriptions, and options for customization or special requests.
Cart	A section of the app that allows users to add and remove items from their current order, view their total cost and estimated delivery time, and proceed to checkout.
Checkout	The process of finalizing and submitting an order, typically requiring the user to confirm their delivery or pickup details, review their order summary, and provide payment information and a delivery tip if applicable.
Order	A record of a user's purchase of food and drinks through the app, typically including the items ordered, the restaurant, the total cost, and the delivery or pickup details
Delivery	A service offered by the app that delivers food and drinks from restaurants to users, typically with an estimated delivery time and a fee based on distance and other factors.
Pickup	A service offered by the app that allows users to order food and drinks from restaurants for pickup, typically with an estimated pickup time and no delivery fee.
Promo code	A code that provides users with a discount or other special offer on their order, typically entered during checkout and subject to specific terms and conditions.
Customer support	A service offered by the app to assist users with questions, issues, or feedback related to their use of the app or their orders, typically accessible through a contact form, email, or phone number and staffed by trained support agents.

Term	Definition
Rating and review	A feature of the app that allows users to rate and provide feedback on restaurants and their food and drinks, typically displayed as a numerical rating and written comments from other users.

Appendix B: To Be Determined List

- Voice **instructions for putting** orders or navigating the app
- Integration with **clever domestic** devices (e.g., Alexa, Google Home) for **smooth** ordering
- In-app chatbot for **customer service**
- Option for **clients** to pay with cryptocurrencies
- Integration with loyalty **packages** for **eating places** and **clients**
- Option for **clients to reserve** from **more than one eating places withinside the identical** order
- Integration with augmented **truth generation** for **digital** menu **surfing** and **meals** visualization
- Integration with social media **systems** for sharing **meals pictures** and **critiques**
- Real-time **transport monitoring the usage of** GPS and geofencing **generation**
- Advanced **seek** and filtering **alternatives** for menus and **eating places** (e.g., **nutritional** restrictions, **delicacies** type)
- Integration with public transit or ride-sharing **offerings** for **quicker transport**
- Virtual **meals gala's** or **occasions withinside the** app
- Option for **clients** to request **custom** menu **gadgets** from **eating places**
- In-app **rankings** and **critiques** for **transport companions**
- Integration with **meals** donation or waste **discount tasks**
- Option for **clients** to pre-order **food and feature** them **added** at **a selected** time.

Appendix C: Terms and Conditions

Intellectual Property Rights Ownership: Our company is the only owner of all intellectual property rights relating to the app, including the source code, supporting materials, and designs. The software and its components may only be used by the client for the purposes specified in the SRS, and this license is non-exclusive and non-transferable.

Payment: The client consents to pay the app development company the agreed-upon fee for creating the app as well as any extra expenses incurred throughout the development process.

User Agreement: The use of the Kami Foods app is at the user's own risk. We are not responsible for any damages or losses resulting from the use of the app.

Privacy Concern: Kamfood takes user privacy seriously and will take reasonable measures to protect user information. However, we are not responsible for any privacy breaches or information leaks caused by third-party entities or hackers.

Users Payment: We are not responsible for any financial transactions between users and restaurants. The app provides a platform for users to order food and make payments, but We are not liable for any issues related to user's payment processing or refunds.

Liability: For a period of 30 days following delivery, the app development business guarantees that the app will be free from faults in materials and workmanship. The customer acknowledges and agrees that the app development company is not responsible for any damages, including lost profits or interruption of business, resulting from the use or inability to use the app.

Settlement of Disputes: The client and the app development company will negotiate to settle any disagreements originating from the SRS or the app development. Both parties agree to bring the matter to binding arbitration if negotiations fail to resolve it.

Governing Law: The laws of the country where the app development business is based shall be used to interpret and govern the SRS and all connected actions.

By signing below, the Company acknowledges that they have read, understood, and agree to the above terms and conditions for the development of the app.

Signature :