IS 4430 Final Group Project -

Seoul Foods

Table of Contents

Part 1: Team Organization

Team Name: Seoul Foods

Luke Foster: I am from San Tan Valley, Arizona. I am double majoring in Finance and IS with an emphasis in FinTech. I chose to go into this field because I like working with databases and want to combine it with my passion for Finance.

Luke Olson: I am a third-year and an accounting major. I am originally from Florida, but moved to Utah about two years ago. In my free time I enjoy working out, playing soccer, and watching movies.

Skylar Strubin: I am from West Jordan, Utah. I am majoring in Data Science and I chose to go into this major because I enjoy working hands-on through using code as well as the major offering good job security.

Jonathan Chan: Activist, Pro-Gun, Reddit Moderator, and most importantly, a father of 8. I started my journey in Korea, fighting with the Soviet Union. I had to fight for them to be promised my freedom, but they went back on their promise. I puppeteered the downfall of the nation using my background in accounting. After fleeing the nation, I lived in Thailand (Don't ask me why) for a couple of years. I then finally moved to Utah where I started my online course to rugpull cryptocurrency, which I targeted towards teenagers. After a short stint in jail, I finally settled down and started my family, and haven't looked back since.

Gavin Cahoon: I'm an IS major from American Fork, Utah. I chose to come on this trip to visit Korea and Japan because I wanted to meet others who have similar interests as I do, as well as they aren two of the countries I have wanted to see the most.

Executive Summary:

Seoul Food is a modern, app-based food delivery service that ensures Koreans have access to any restaurant to consume anywhere. By connecting restaurants, customers, delivery driver, and customer support staff, Seoul Food is able to deliver convenience, quality, and customer satisfaction to all its customers.
Restaurants manage their menus and promotions through our user management system to increase visibility and streamline order fulfillment.
Customers enjoy a user-friendly interface that allows easy ordering, real-time tracking, and secure payments. The platform also includes a robust feedback system, enabling customers to

rate food quality, delivery time, and overall satisfaction. This feedback enhances transparency and drives performance-based incentives for both drivers and restaurants.
Delivery drivers are the lifeline of the service. Through optimized route algorithms and real-time navigation, we ensure fast, reliable deliveries. Drivers receive ratings from customers, and high performers are rewarded through bonuses and scheduling flexibility.
Customer support staff ensures that customers can always have a memorable food experience. They handle inquiries, resolve issues, and ensure satisfaction. They provide timely, professional assistance to maintain trust and loyalty.
Seoul Food is not just a delivery app, it's a complete ecosystem focused on efficiency, feedback, and user experience, making it the best food delivery in South Korea.

Actors

Customer: End user who browses menus, places orders, provides feedback, and may request support.

Restaurant: Partner business that manages its own profile, uploads menu items, and fulfills incoming orders.

Driver: Delivery personnel responsible for picking up and delivering orders to customers.

Administrator: System overseer who manages approvals, user accounts, and handles escalated support issues.

System: Centralized digital platform handling data processing, communication, and transaction management.

As a…	I want to…	So that…	Achievement Criteria
Customer	Browse menus and place orders through the app	I can conveniently get food delivered from local restaurants	Customer can log in, browse restaurants, and view menus Orders can be placed and tracked in real-time Payment is processed securely through the app

Customer	Provide feedback or rate my order	I can share my experience and help improve service	Feedback form appears after delivery is complete Ratings and comments are stored and viewable by restaurants Low ratings can trigger customer service follow-up
Restaurant	Receive and manage orders through a dashboard	I can prepare food on time and ensure accurate delivery	Incoming orders are shown in real-time Restaurant can update order status (preparing, ready, etc.) Order changes notify customer and driver automatically
Restaurant	View customer feedback	I can improve food quality and service	Restaurant has access to historical order ratings Negative feedback includes order details Managers can respond or flag recurring issues
Driver	Accept delivery requests through the app	I can earn income by delivering food efficiently	Driver receives real-time delivery requests with location

			and order details Acceptance updates the system and assigns the order Navigation is integrated with real-time traffic data
Driver	Mark orders as delivered and collect tips	I can complete deliveries and be rewarded fairly	Driver can update delivery status in-app Customer has option to tip after delivery Delivery data is logged for performance tracking

Part 3: Use Case Scenarios

User Management: Allows customers to register, log in, and manage subscriptions.

Menu & Restaurant Management: Enables restaurants to register, edit profiles, and manage menus.

Search & Browse Restaurants: Customers can explore available restaurants and their menus.

Order Processing: Facilitates placing and validating food orders.

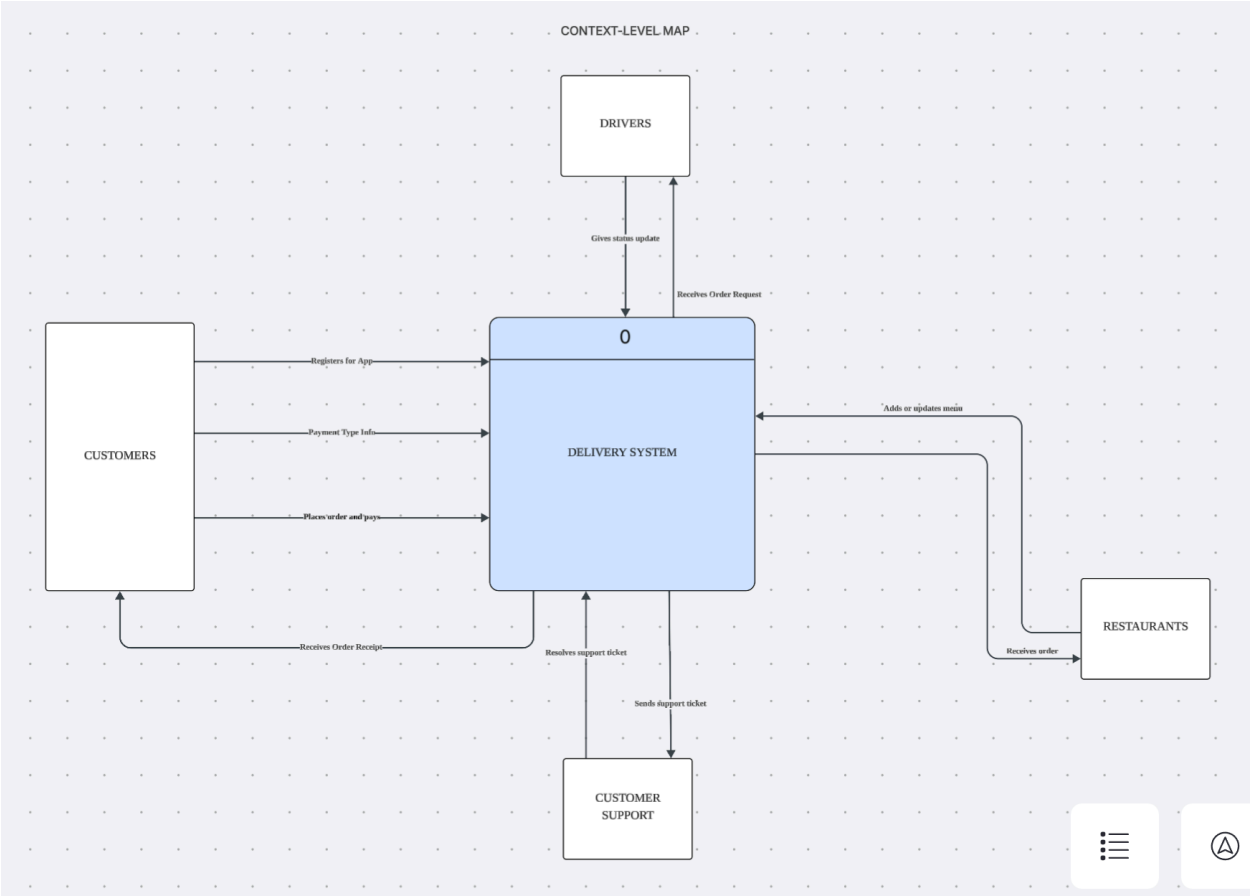
Delivery Dispatch: Assigns drivers and tracks order deliveries.

Customer Feedback: Collects customer feedback post-delivery.

Support Ticket Management: Lets users raise and resolve support tickets.

Part 4: Context-Level Diagram

The context diagram illustrates the Seoul Food system as a single process interacting with external entities: Customers, Restaurants, Drivers, and Administrators. All data flows in and out of the central system process, highlighting login, menu info, orders, deliveries, and feedback.

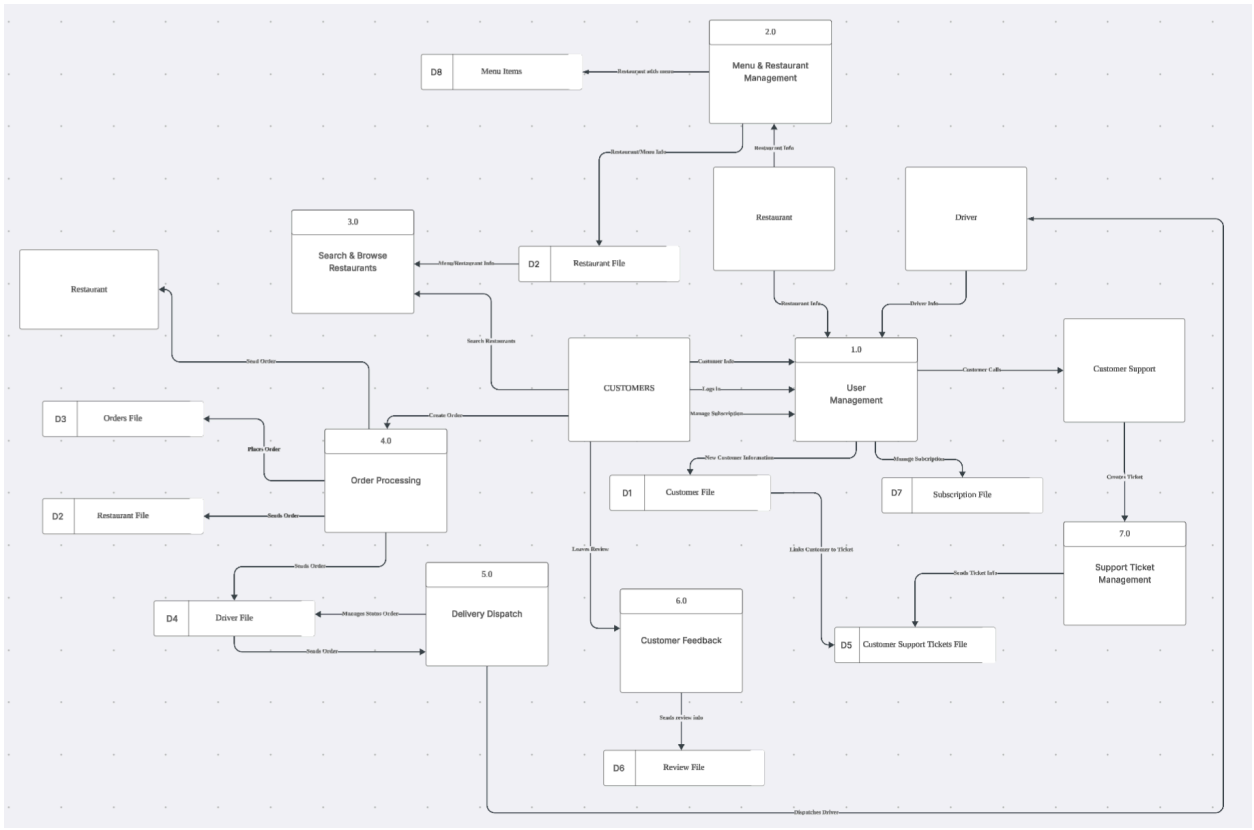


Part 5: Diagram 0

Diagram 0 expands the context diagram into seven major processes:

1.0 User Management 2.0 Menu & Restaurant Management 3.0 Search & Browse Restaurants 4.0 Order Processing 5.0 Delivery Dispatch 6.0 Customer Feedback 7.0 Support Ticket Management

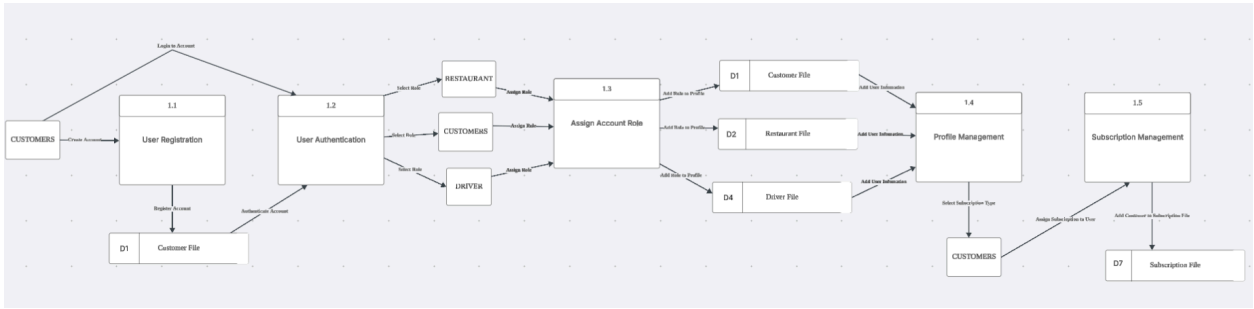
Each process interacts with specific data stores and external actors to fulfill system functionality.



Part 6: Level-1 Diagram Expansion

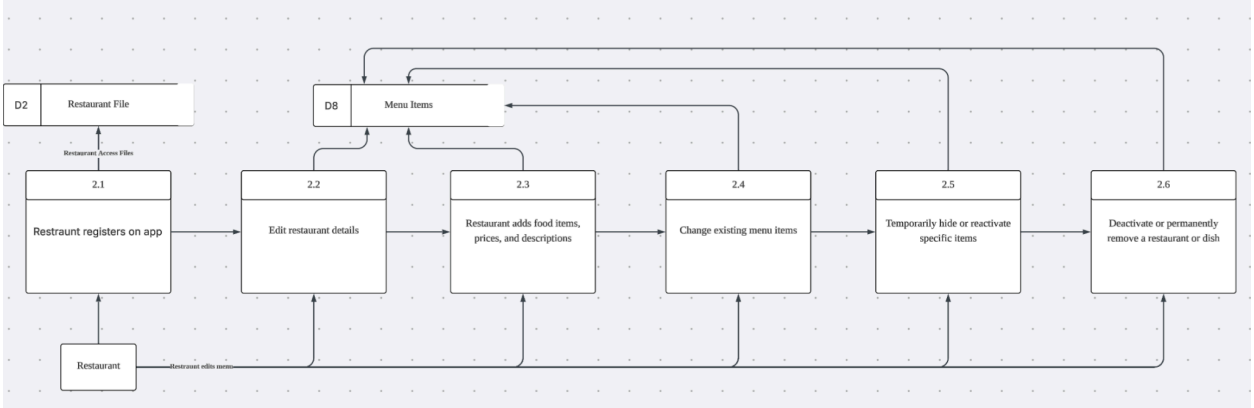
1.0 User Management

- 1.1 User Registration
- 1.2 User Authentication
- 1.3 Assign Account Role
- 1.4 Profile Management
- 1.5 Subscription Management



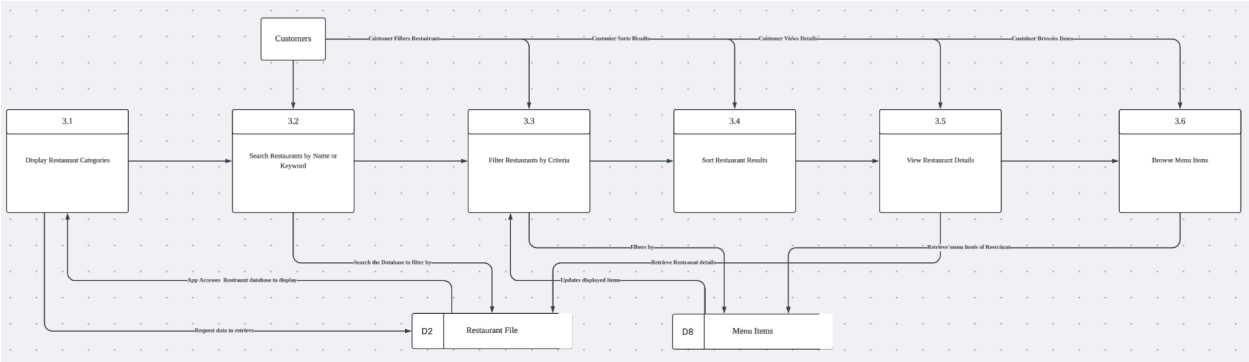
2.0 Menu & Restaurant Management

- 2.1 Restaurant Registers on App
- 2.2 Edit Restaurant Details
- 2.3 Add Menu Items
- 2.4 Change Existing Menu Items
- 2.5 Temporarily Hide/Reactivate Items
- 2.6 Deactivate/Remove Restaurant or Dish



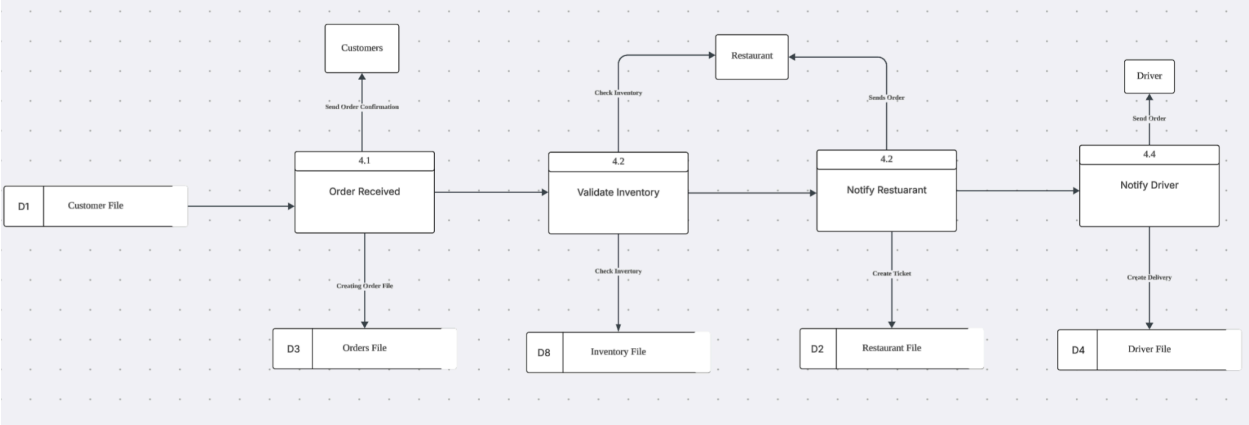
3.0 Search & Browse Restaurants

- 3.1 Display Restaurant Categories
- 3.2 Search by Name or Keyword
- 3.3 Filter by Criteria
- 3.4 Sort Results
- 3.5 View Restaurant Details
- 3.6 Browse Menu Items



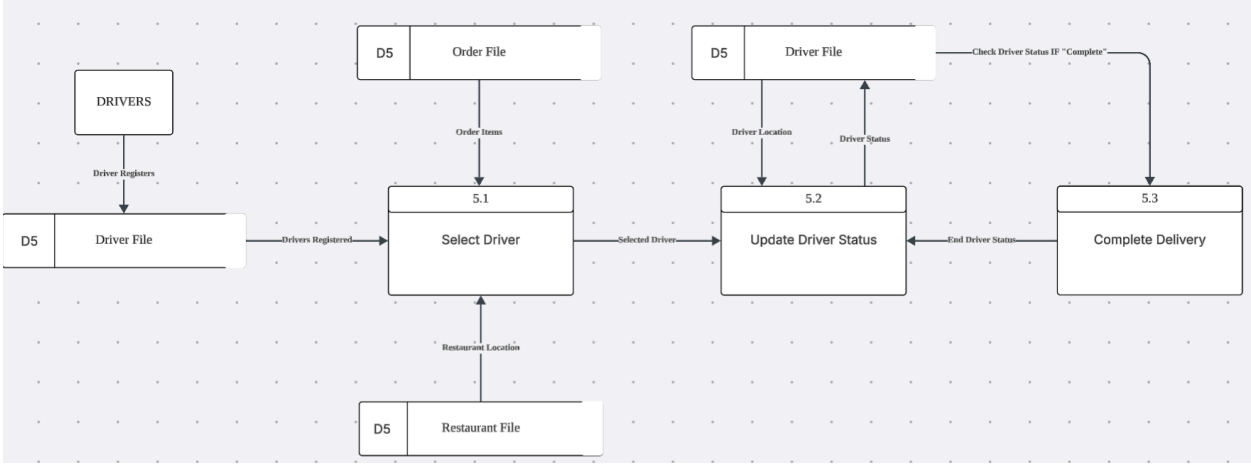
4.0 Order Processing

- 4.1 Order Received
- 4.2 Validate Inventory
- 4.3 Notify Restaurant
- 4.4 Notify Driver



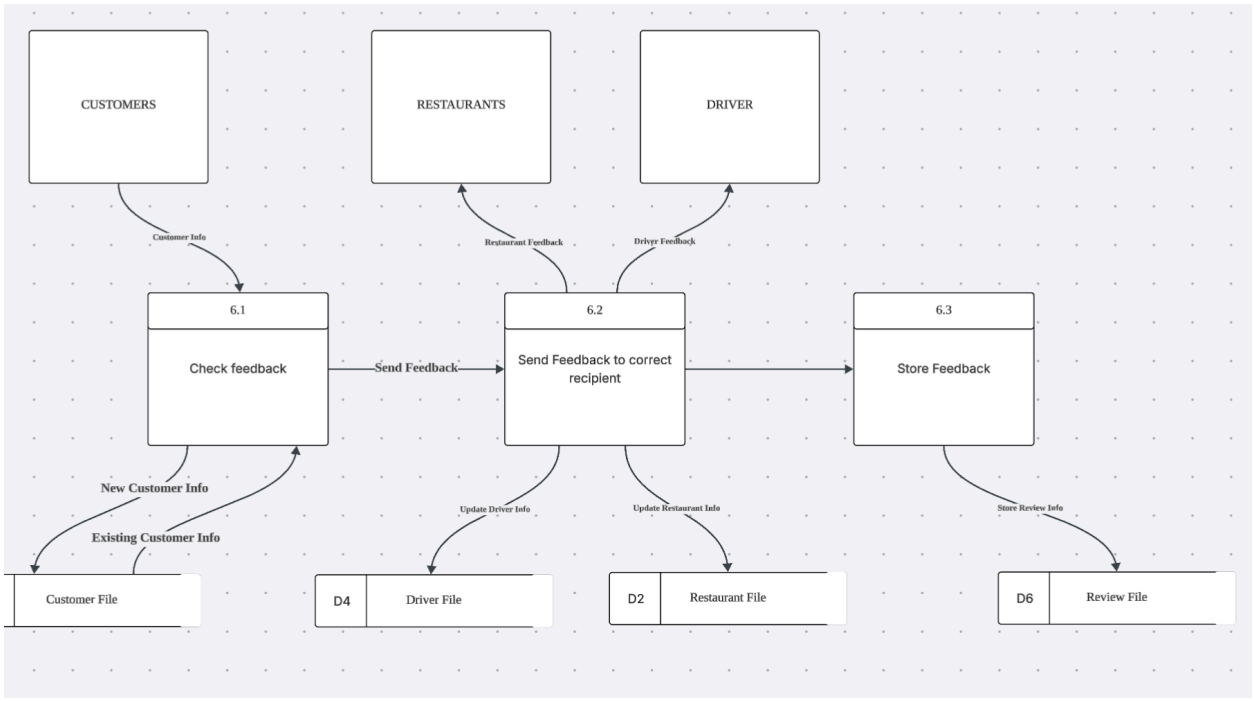
5.0 Delivery Dispatch

- 5.1 Select Driver
- 5.2 Update Driver Status
- 5.3 Complete Delivery



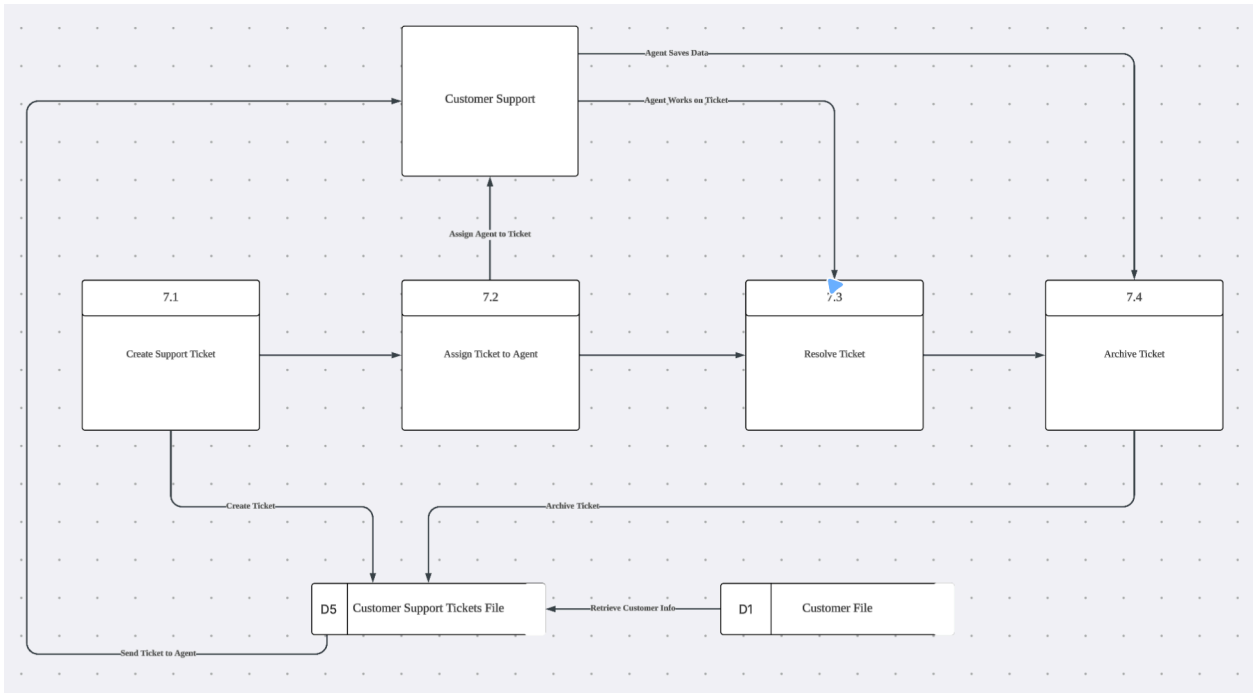
6.0 Customer Feedback

- 6.1 Check Feedback
- 6.2 Route Feedback
- 6.3 Store Feedback



7.0 Support Ticket Management

- 7.1 Create Support Ticket
- 7.2 Assign Ticket to Agent
- 7.3 Resolve Ticket
- 7.4 Archive Ticket



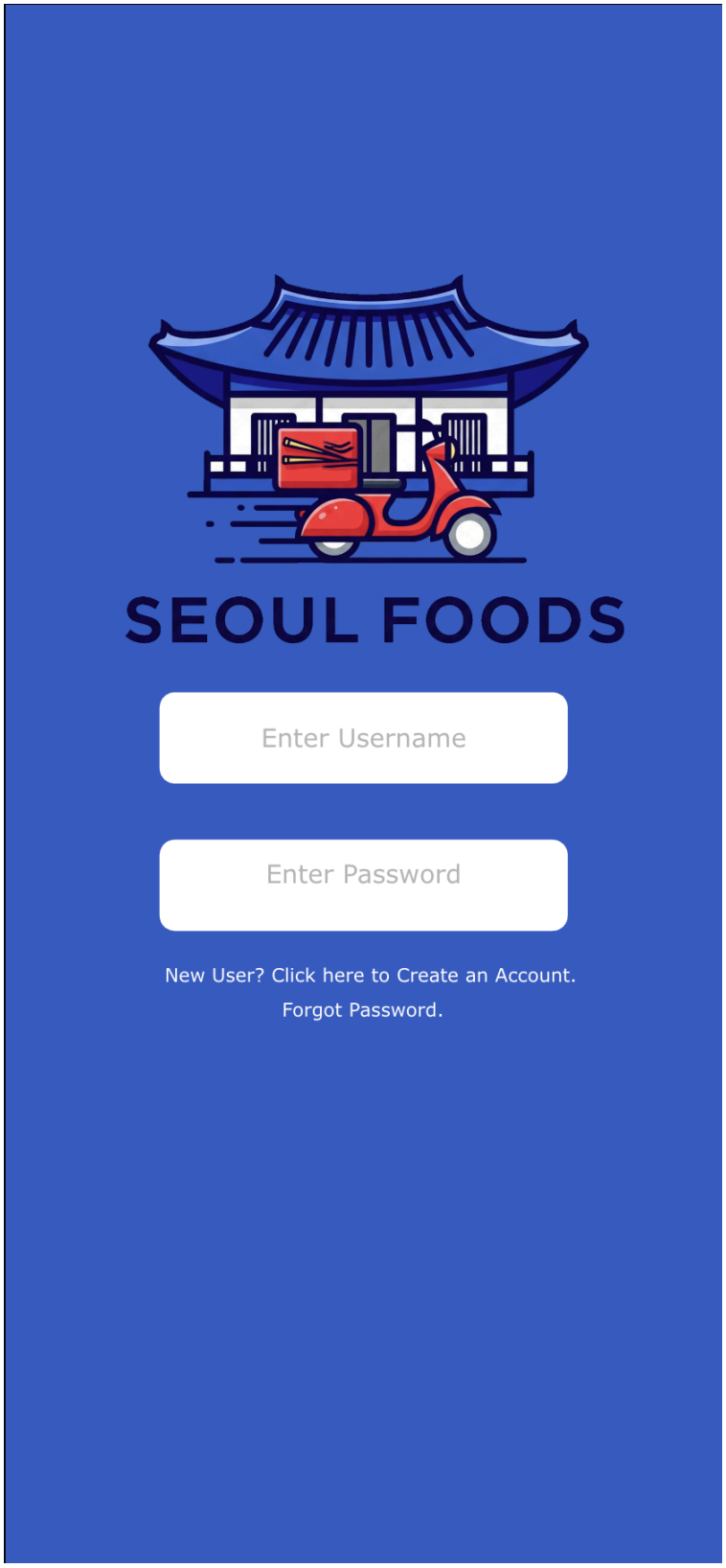
Part 7: Mockup Screens

Included are UI mockups showcasing:

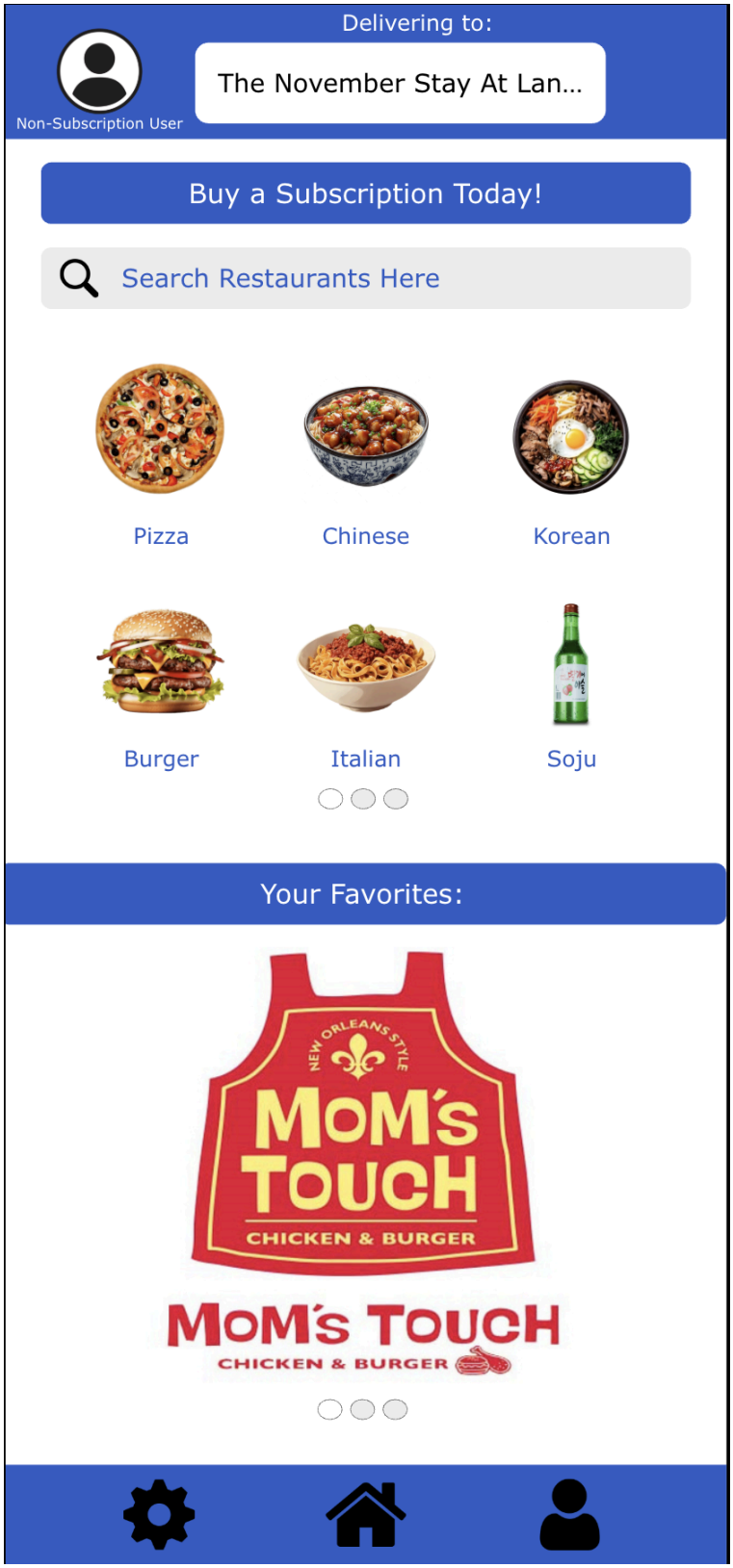
- Customer sign-up/login
- User Dashboard
- Delivery status tracker
- Feedback form and support ticket submission

The design prioritizes usability, clean layout, minimal steps, responsive layout, and clear feedback messages. Improvements over existing apps include more transparency in delivery tracking and simpler subscription management.

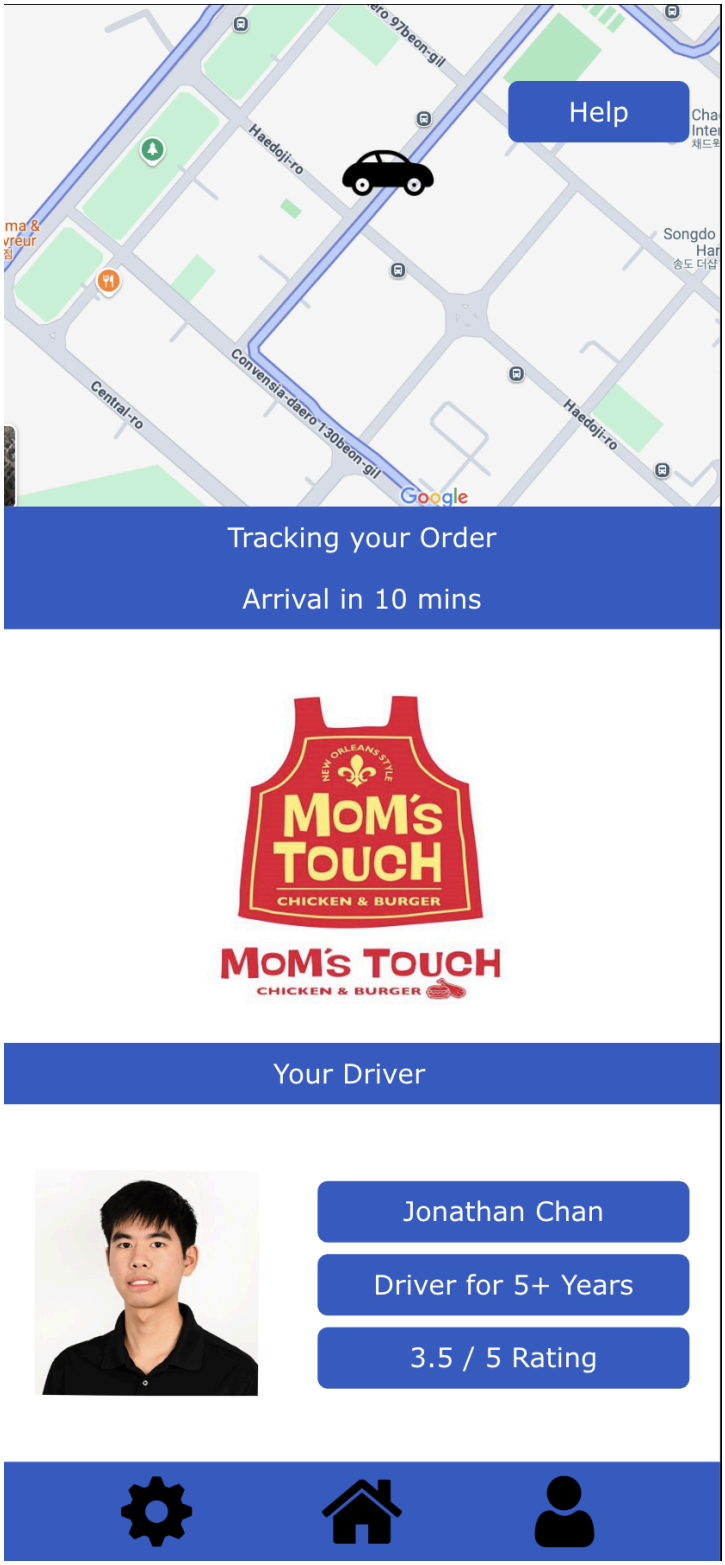
Customer Sign-up/Login:



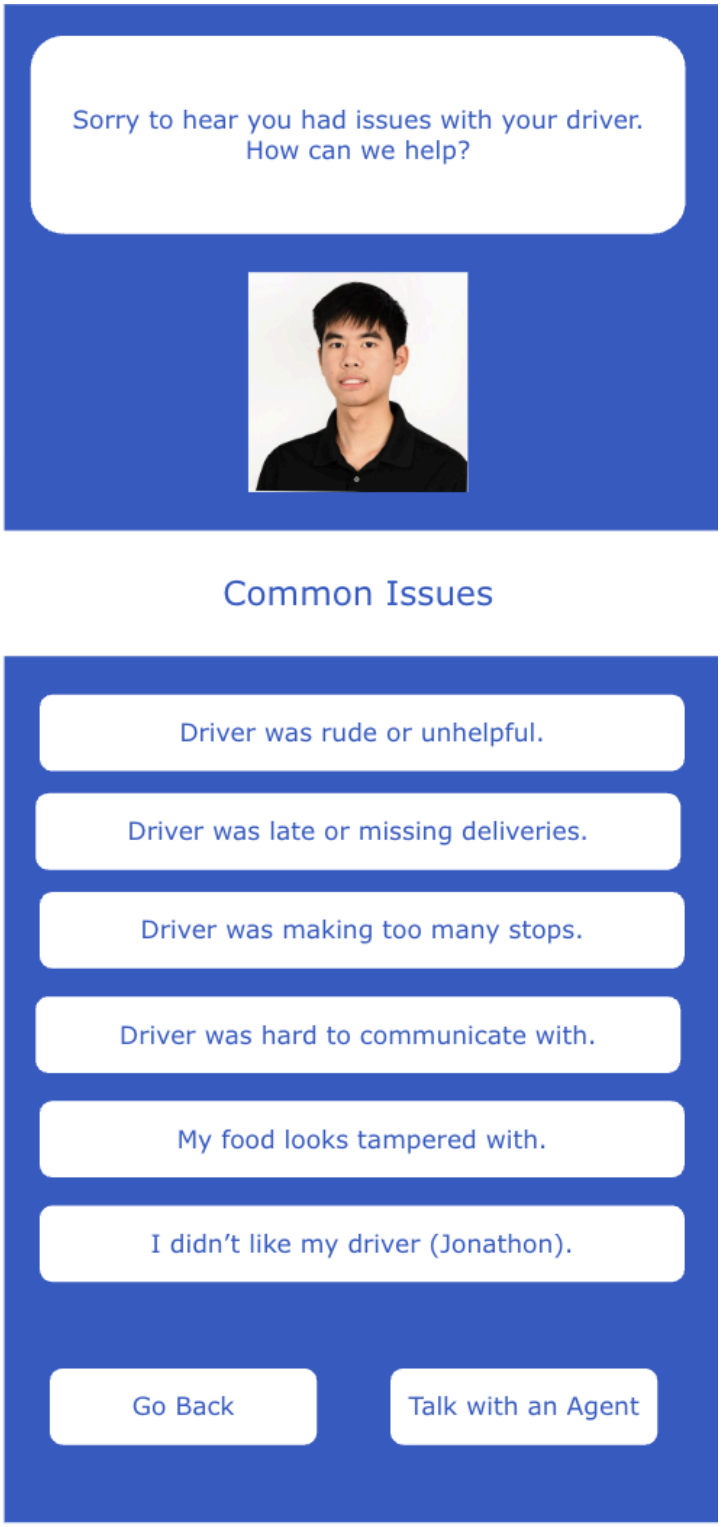
User Dashboard:



Delivery Status Tracker:



Feedback form and support ticket submission:



Part 8: References

- Uber Eats Interface Study
- DoorDash System Architecture
- IS 4430 System Design Lecture Notes

Part 2: Project Selection & Requirements Analysis

Possible Survey Questions:

- How often do you use our food delivery service?
- How would you rate the quality of the food you received?
- Are there any features you would like to see in the app?
- Did the delivery arrive within the estimated time?

Part 3: Project Plan

Use case name: User Management	
Actors: Customer, System Administrator	
Description: Manages customer registration, login, and subscription details.	
Steps Performed	1. Customer accesses the system. 2. Registers or logs into their account. 3. Provides necessary personal information. 4. System validates and stores the data. 5. Users can manage their subscription preferences.
Preconditions: Customer must access the application or website.	
Postconditions: Customer account is active, and subscription settings are updated.	
Assumptions: Customer has internet access and valid credentials.	
Success Guarantee: Customer is logged in and can manage their information.	
Outstanding Issues: Handling invalid credentials and data validation compliance.	

Use case name: Menu & Restaurant Management	
Actors: Restaurant	
Description: Restaurants provide and manage menu items and restaurant profiles.	
Steps Performed	1. Restaurant registers with the platform. 2. Admin verifies and approves the restaurant profile. 3. Restaurant uploads menu and details. 4. System stores and updates menu information.
Preconditions: Restaurant must be registered and approved.	
Postconditions: Menu items and restaurant details are live in the system.	
Assumptions: Restaurant data is accurate.	
Success Guarantee: Restaurant and menu are correctly displayed to customers.	
Outstanding Issues: Menu accuracy and timely updates.	

Use case name: Search & Browse Restaurants	
Actors: Customer	
Description: Customers search and browse through listed restaurants and their menus.	
Steps Performed	1. Customer logs in and accesses the browsing feature. 2. System retrieves restaurant list and menu items. 3. Filters and search options are applied. 4. Restaurant pages are presented to the user.
Preconditions: Restaurants and menus are already added.	
Postconditions: Customers view restaurant and food options.	
Assumptions: Menu and restaurant data is current.	
Success Guarantee: Customers are able to browse and find restaurants.	
Outstanding Issues: Search performance and accuracy of results.	

Use case name: Order Processing	
Actors: Customer, System	
Description: Handles customer order creation and links it to the restaurant and delivery system.	
Steps Performed	1. Customer selects menu items and places an order. 2. System validates and creates the order. 3. Order is sent to the restaurant. 4. System updates the order status. 5. Order is prepared and ready for dispatch.
Preconditions: User is logged in and selects a valid menu.	
Postconditions: Order is processed and prepared for delivery.	
Assumptions: Restaurant is open and has capacity to fulfill orders.	
Success Guarantee: Order is successfully created and forwarded.	
Outstanding Issues: Delays in order updates or missing items.	

Use case name: Delivery Dispatch	
Actors: Driver	
Description: Manages assignment and dispatch of drivers to fulfill customer orders.	
Steps Performed	1. System selects or assigns a driver. 2. Driver receives notification and order details. 3. Driver picks up the order from the restaurant. 4. Order is marked as out for delivery.
Preconditions: Order is ready from the restaurant.	
Postconditions: Driver is assigned and order is dispatched.	
Assumptions: Driver availability.	
Success Guarantee: Order reaches the customer.	
Outstanding Issues: Driver delays or miscommunication.	

Use case name: Customer Feedback	
Actors: Customer	
Description: Captures customer feedback and satisfaction data.	
Steps Performed	1. After order completion, customer receives a feedback form. 2. Customer rates the experience and adds comments. 3. Feedback is submitted to the system. 4. Feedback is linked to specific order and customer record.
Preconditions: Customer received an order.	
Postconditions: Feedback is stored and associated with the order.	
Assumptions: Customer is willing to provide feedback.	
Success Guarantee: Feedback is captured and stored.	
Outstanding Issues: Low participation rate in surveys.	

Use case name: Support Ticket Management	
Actors: Customer, Customer Support	
Description: Allows customers to raise issues or support tickets.	
Steps Performed	1. Customer initiates support request via the system. 2. System logs and assigns a ticket. 3. Support agent reviews and responds. 4. Ticket is resolved and marked complete.
Preconditions: Customer is logged in and has an issue.	
Postconditions: Ticket is logged and processed.	
Assumptions: Support team is active and available.	
Success Guarantee: Customer issue is resolved.	
Outstanding Issues: Response time and ticket closure tracking.	