Enroute Food Delivery

“If you need us, we’re on the way”

Team 2

Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

Enroute

Team 2

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

---

Table of Contents

Executive Summary

Our project focuses on making nutritious food more accessible to people in remote areas or those affected by natural disasters, war, or poor infrastructure. Many individuals struggle with nutritious food insecurity because traditional delivery methods can't reach them. To solve this problem, we are developing an autonomous drone delivery system that can transport food quickly and efficiently, even in difficult conditions. Using drones allows us to bypass obstacles and reach destinations that humans cannot, like damaged roads or dangerous terrain, ensuring that people in need get the food they require without delays.

This solution provides fast, cost-effective, and sustainable food delivery, providing access to nutritious food for vulnerable communities. Unlike traditional methods, drones reduce environmental impact by using less fuel and minimizing road traffic. They also offer critical support during emergencies, where time-sensitive food deliveries can make a major difference. Our drone concept not only helps people in need but also creates new opportunities for humanitarian organizations and businesses to explore drone-based logistics. By making food delivery more efficient and reliable, we hope to improve lives and contribute to a better, more sustainable future.

Problem Statement & Opportunity

Understanding the Problem

Food insecurity is a major global issue, particularly in disaster-stricken areas, remote locations, and impoverished communities where traditional food supply chains are unreliable or entirely unavailable. Our project aims to solve the following key challenges:

- **Cost of Food & Poverty:** Many individuals struggle to afford basic necessities, making food accessibility a persistent challenge.
- **Disaster Relief Bottlenecks:** Natural disasters frequently disrupt infrastructure, delaying or preventing timely food delivery.
- **Remote Locations & Poor Infrastructure:** Rural and hard-to-reach areas often lack proper roads or distribution networks, making traditional delivery methods inefficient or impossible.

Lean Canvas (*Figure 1*)

<table>
<tr>
<td rowspan="2">Problem

1. cost of food/poverty.

2. Disaster relief

3. Remote locations/bad infrastructure</td>
<td>Solution

1. No delivery fees

2. Focus on delivering to people in need

3. Energy efficient drones</td>
<td rowspan="2">Unique Value Proposition

Our companies focus is disaster relief and delivering food to people in need.</td>
<td>Unfair Advantage

Teams robotics experience and research on drone delivery.</td>
<td rowspan="2">Customer Segmentation

1. People in disaster zones

2. Remote individuals

3. Poor individuals</td>
</tr>
<tr>
<td>Key Metrics

1. Successful/Safe food deliveries

2. Drone response time

3. Cost efficient</td>
<td>Channels

1. Government

2. NGO/Disaster Response Organization

3. Marketing</td>
</tr>
<tr>
<td colspan="2">Cost Structures

Fixed: Development, Marketing, Office, Legal, Food logistics

Variable: Hosting</td>
<td colspan="3">Revenue Streams

1. Government Contracts

2. Food Suppliers

3. Individual Deliveries</td>
</tr>
</table>

Enroute

Team 2

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

---

Proposed Solution

To address these issues, our autonomous food drone delivery system provides:

1. No delivery fees, ensuring affordability for those in need.
2. A focus on disaster relief, prioritizing emergency food supply in crisis zones.
3. Energy-efficient drones, minimizing operational costs and environmental impact.

By utilizing autonomous navigation, real-time tracking, and optimized delivery routes, this system ensures food reaches its destination safely and efficiently.

Unique Value Proposition

Our project stands out due to its focus on humanitarian aid, leveraging drone technology to deliver food where traditional logistics fail. Unlike commercial delivery services, we prioritize speed, accessibility, and efficiency in disaster relief and food distribution efforts.

Understanding the Users

In designing the food drone delivery system, we have identified two key personas who represent our target audience: individuals in remote locations facing logistical challenges and businesses affected by natural disasters. These personas help us tailor our solution to meet real-world needs effectively.

Persona 1 - Katie McAlister (*Figure 2*)

KATIE MCALISTER - RETURNING CUSTOMER

DEMOGRAPHICS

Lives in remote location
Single mother
5 kids

MOTIVATIONS

Needs food delivered
wants healthy food
Loyal Customer

CONCERNS

SPEED OF DELIVERY

UNHEALTHY FOOD

DELIVERIES TAKING LONGER AS THE COMPANY GETS BIGGER

DESCRIPTION

Katie is a single mother of 5. She works remotely at a large tech company. amongst long work hours, taking care of her kids, and living an hour from any stores she has no time to go grocery shopping. She is not concerned about the costs, but knows it is hard to find a food delivery service willing to go out to her.

DELIVERY SERVICE DESIRES

ease of use

Healthiness of food

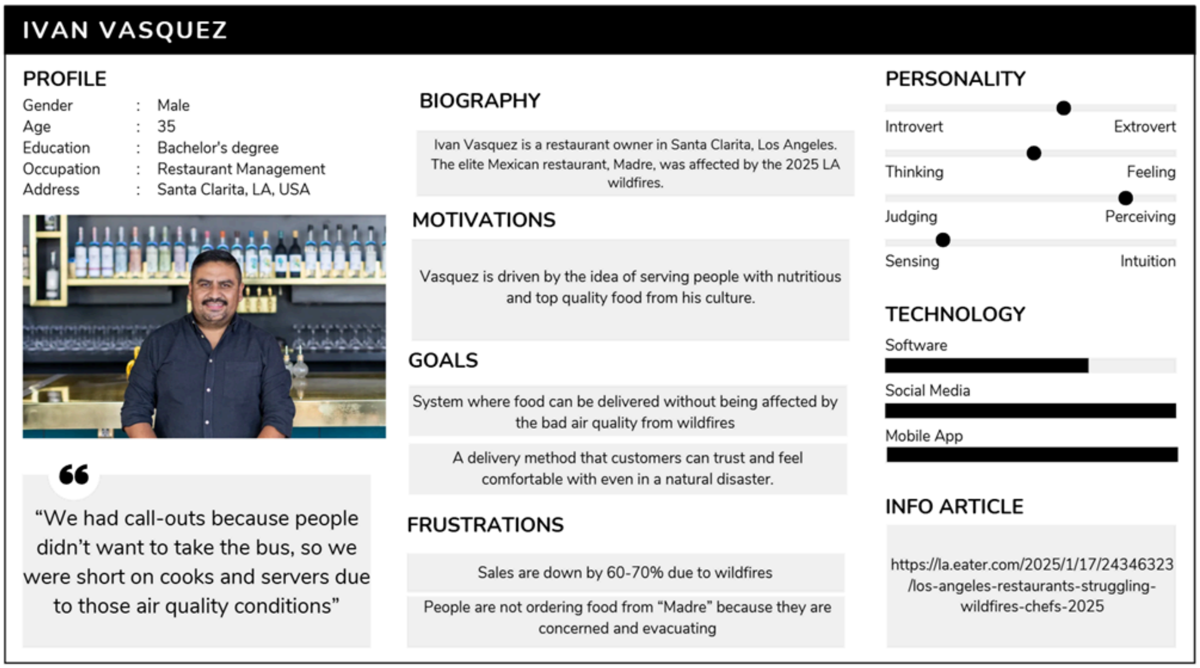
Cheap

KEY REALTIONS

Lives an hour from any stores

Has no extra time in her day to go grocery shopping

Persona 2 - Ivan Vasquez (*Figure 3*)



How These Personas Shape Our System

By addressing the needs of individuals and businesses, the food drone delivery system will:

1. Ensure food reaches people in remote locations efficiently.
2. Provide disaster-resilient delivery solutions for businesses and individuals.
3. Incorporate mobile and online platforms for easy accessibility.
4. Prioritize health-conscious and time-sensitive deliveries.

These personas illustrate the real-world impact of our project, reinforcing the necessity of an innovative, technology-driven solution for food accessibility.

System Overview & Interaction

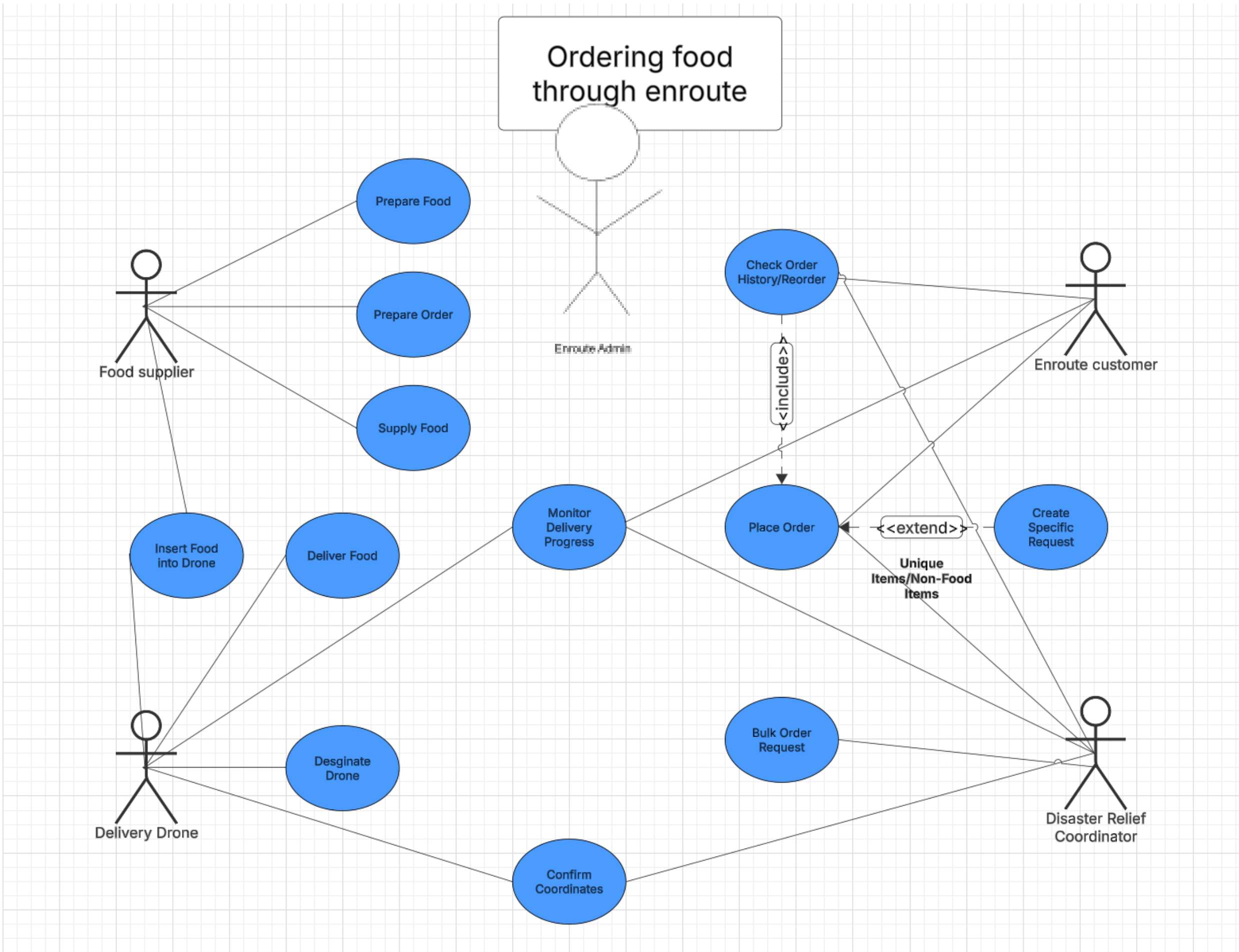
Looking at the actor-system interactions, we have identified five actors who play a crucial role in the drone delivery system. These actors include the Admin, customer, food supplier, delivery drone, and disaster relief coordinator. A couple of the use cases where these actors interact are with packing food and loading the drone (food supplier & drone) and place order (user, disaster relief coordinator, & admin). These actors and use cases give us a broader understanding of how the system will work and how actors will interact with each other in the system.

Enroute

Team 2

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

Use Case Diagram (*Figure 4*)



How the Use Case Diagram Addresses User Needs

The use cases that actors interact with provide the functional aspects of the application. These determine the features that users will need in order to meet their needs and wants. For example, the place order feature allows users to order the food they like, to the location they prefer, and in the time range that works best for them. The diagram shows other actors like the disaster relief coordinator and admin also interact with this feature, adding few constraints on the flexibility of the user choice.

Key Scenarios & Workflows

To truly understand our stakeholders and the potential problems they face we needed a way to connect with them on a more personal/emotional level. Decided use case scenarios would be the best way to achieve this to help our company understand our customers' challenges and mold them to the needs that come with those challenges. Use case scenarios give Enroute a small insight into their lives and how they live each day. Giving us the ability to make it just that little bit easier.

Scenario #1: Fatima’s Remote Village

Introduction: Fatima, a 34-year-old primary school teacher, lives in a remote mountain village in a developing country. Her community relies on seasonal farming, and harsh weather often cuts them off from nearby towns. Recently, a devastating landslide destroyed the only road to the village, leaving her and the other villagers stranded without essential supplies. Fatima, a caregiver to her elderly parents and two children, feels an overwhelming sense of responsibility to provide for her family but is at a loss for how to ensure their survival.

Situation: With local stores running out of food and water, Fatima has spent hours each day rationing meals and comforting her children, who don’t understand why their meals are suddenly smaller. She’s started walking hours to neighboring villages in hopes of finding supplies but often returns empty-handed, exhausted, and increasingly worried about her parents’ health. When rumors spread of a drone system delivering aid, she feels a faint glimmer of hope. Could this be the lifeline her family desperately needs?

<product gets released>

Outcomes: Within a few days, a drone lands in Fatima’s village carrying packages of non-perishable food, water filters, and basic medicines. Fatima watches with tears in her eyes as her children open the packages, thrilled at the sight of fresh food. Her parents regain strength after days of hunger, and Fatima feels a renewed sense of hope for the future. For the first time in weeks, she doesn’t have to worry about whether her family will eat. The drones continue to deliver supplies as her village begins the slow process of recovery. Fatima marvels at the speed, efficiency, and reliability of this system, grateful for how it turned her family’s despair into resilience.

Scenario #2: James' Disaster Relief Organization

James Carter, a logistics coordinator of the disaster relief organization, is deeply disturbed by this. A category 4 hurricane destroyed coastal areas, leaving thousands of people without food, water or shelter. James is determined to ensure the timely delivery of important food to remote areas, but he faces logistical challenges in coordinating the delivery. Traditional methods, such as trucks and ships, take too long because of flooding and damage to roads and infrastructure. James needs a quick and reliable solution to save lives.

When James starts a new day, he reviews the request list of several shelters. It is urgent that necessities need to reach stranded communities within a few hours. He logged into the application, delivered with a drone, ordered a large number of ready-to-eat food, and coordinated delivery to different locations. While navigating the app, James feels a mix of hope and anxiety. He configures bulk order requests, assigns priority levels to critical shelters, and tracks drone fleet availability in real-time. James also ensures he communicates with ground teams to prepare for the incoming supplies. The app's intuitive interface makes it easy to map drone routes, but James is also mindful of the weather and airspace restrictions.

As the drones took off, James monitored their progress via the live tracking feature. He noticed that one of the drones was experiencing delays due to high winds. However, the system automatically rerouted the drones and notified James of the adjusted delivery time. Throughout the day, James coordinated with local authorities, answered questions from shelter managers, and made sure the drones' payloads were fully utilized throughout the day. James was relieved to receive confirmation that a delivery had been made in the evening. The app provided a summary of more than 500 meals delivered to 10 locations within four hours. James was proud to see the positive impact of this new delivery system, knowing that lives were saved because of its efficiency and reliability.

The solution exceeded James' expectations by ensuring on-time, accurate, and safe deliveries, even in chaotic environments. The combination of real-time tracking and automatic rerouting reduced delays, allowing James to focus on other important relief efforts. His emotional state changed from stress and anxiety to gratitude and encouragement, knowing that he had a reliable ally in the disaster response.

Functional & Non-Functional Requirements

Enroute takes stakeholders' needs very seriously. To accommodate these needs we need to have functional and nonfunctional requirements to follow Through the development process. We connected with customers, food suppliers, and many other stakeholders to get some feedback on what their needs are when it comes to our food delivery system. We were then able to break down these requirements into functional requirements and non-functional requirements. We then broke up the nonfunctional requirements into subcategories of: performance requirements, safety requirements, security requirements, software quality attributes,and business rules.

Functional Requirements

REQ-1: (MUST) Authenticate User
The system must authenticate users before they can place an order.

REQ-2: (SHOULD) Estimate Delivery Time
The system should provide an estimated delivery time before payment confirmation.

REQ-3: (COULD) Modify and Cancel Order
The system could allow payers to modify or cancel orders before food preparation begins.

REQ-4: (WON'T) Delivery Location Change Not Accepted
The system won't allow users to change delivery locations once the order is placed (unless they cancel it).

REQ-5: (SHOULD) Fail-Safe System Mechanism
The system should have fail-safe mechanisms for unusual conditions (i.e. inclement weather, rerouting, etc.).

REQ-6: (MUST) Real-Time Tracking Updates
The system must provide real-time tracking updates to the payer.

REQ-7: (SHOULD) Schedule for Future Deliveries
The system should allow payers to schedule orders for future delivery.

REQ-8: (MUST) Battery Management
The drone must monitor its battery level and return to a charging station if the battery falls below a critical threshold.

REQ-9: (MUST) Obstacle Avoidance
The drone must detect and avoid obstacles (trees, buildings, birds, etc.) using onboard sensors.

REQ-10: (SHOULD) Emergency Landing
The drone should have an emergency landing protocol in case of critical failures (e.g., battery depletion, motor failure).

REQ-11: (COULD) Voice Command Support
The system could allow customers to control basic functionalities via voice commands.

REQ-12: (MUST) Order Verification
The system must ensure that the correct food order is loaded onto the drone before dispatch.

REQ-13: (SHOULD) Dynamic Rerouting
The drone should dynamically change routes based on airspace restrictions or sudden weather changes.

REQ-14: (WON'T) Unauthorized Areas
The drone won't fly into restricted airspaces, such as airports, military zones, or no-fly zones.

REQ-15: (MUST) Order Handling Confirmation
The system must verify that the customer has received the package via QR code scan, facial recognition, or weight detection.

REQ-16: (COULD) Remote Manual Control
Engineers could take manual control of the drone in case of system failure.

REQ-17: (MUST) Identify Remote and Disaster-Affected Areas
The system must be able to identify remote and disaster-affected areas by analyzing geographical data to prioritize delivery routes.

REQ-18: (MUST) User Order Input
The system must allow users to input food orders through a mobile or web interface, specifying item details and delivery location.

REQ-19: (SHOULD) Enhanced Real-Time Tracking
The system should enable real-time tracking of drone deliveries, including estimated time of arrival and route updates.

REQ-20: (MUST) Automated Delivery Confirmation
The system must include an automated system to confirm successful delivery, sending notifications to both users and delivery personnel.

REQ-21: (COULD) Customer Rating System
The system could provide an option for customers to rate their delivery experience after the drone has completed the delivery.

REQ-22: (MUST) Multi-Drone Model Support
The system must support multiple drone models with varying capacities for different order sizes and delivery areas.

Non-Functional Requirements

<Performance Requirements>

REQ-NF1: (MUST) Scalability
The system must handle at least 100 simultaneous drone deliveries without performance degradation

REQ-NF2: (MUST) Delivery Time
The drone must be able to deliver orders within 1 hour for 98% of customers.

REQ-NF3: (SHOULD) Weight
The drone should be able to hold 30 pounds or more of supplies.

REQ-NF4: (MUST) Efficiency
The drone must deliver the correct supplies 100% of the time.

<Safety Requirements>

REQ-NF5: (MUST) Fault Tolerance
The system must be able to recover from minor software failures without stopping operations.

REQ-NF6: (MUST) Reliable Navigation
The system must have reliable navigation with GPS, live feed (only for drone engineers), and fail-safe mechanisms.

REQ-NF7: (SHOULD) System Redundancy
The system should have redundancy to ensure drone operation even if one or more servers fail.

<Security Requirements>

REQ-NF8: (MUST) Drone Security
The drone must have encrypted communication channels to prevent hacking or unauthorized takeovers.

REQ-NF9: (MUST) Data Privacy and Security
The system must ensure that data privacy and security measures comply with industry standards (e.g., encryption for sensitive user data).

<Software Quality Attributes>

REQ-NF10: (SHOULD) User-Friendly Interface
The system should have a user-friendly UI that is accessible on both mobile devices and desktops for ease of use.

REQ-NF11: (SHOULD) Energy Efficiency
The drone should optimize energy usage to maximize flight range and reduce operational costs.

REQ-NF12: (COULD) Noise Reduction
The drone could use quieter propellers to minimize noise pollution in residential areas.

REQ-NF13: (COULD) Eco-Friendliness
The system could use renewable energy sources or biodegradable packaging to promote sustainability.

REQ-NF14: (COULD) Multi-Language Support
The system could provide localization for at least two additional languages besides English.

<Business Rules>

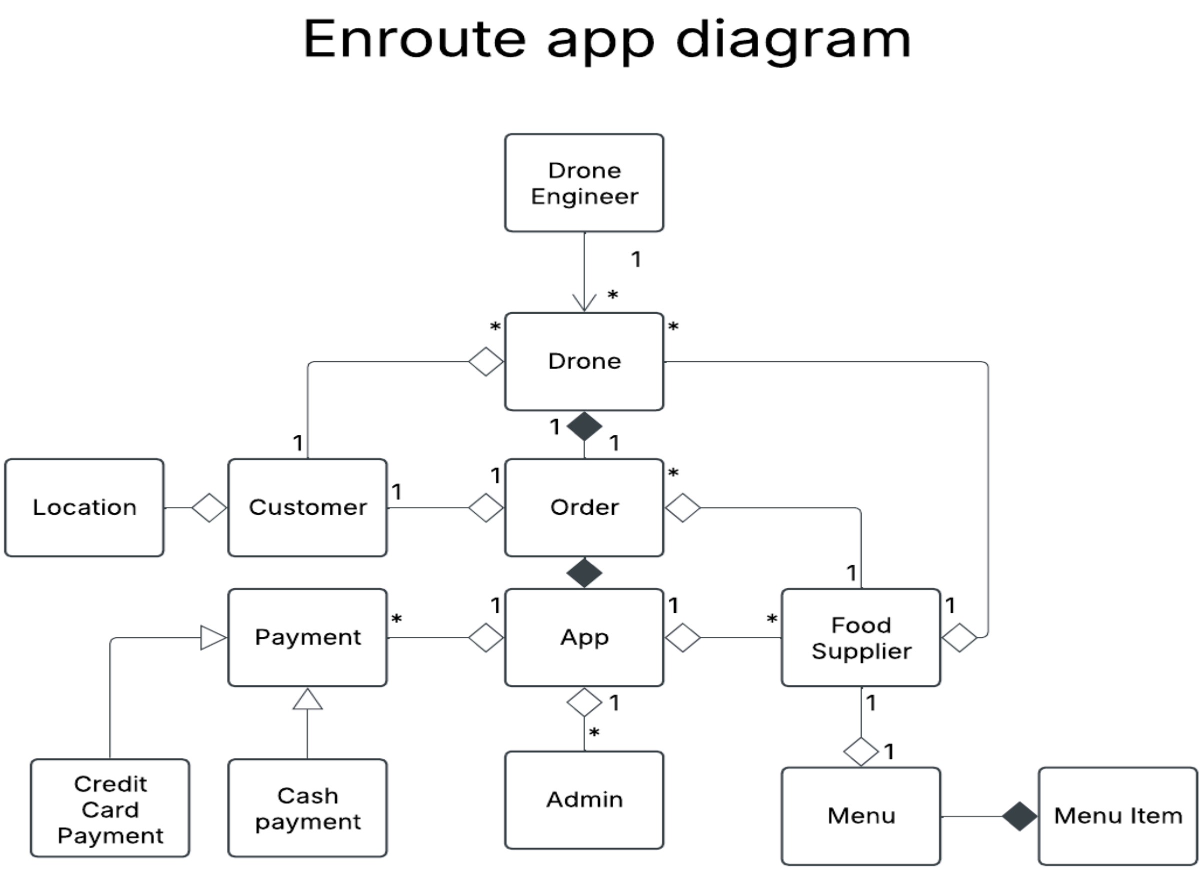
REQ-NF15: (WON'T) Human Pilots
The system won't rely on human pilots for regular delivery operations unless there is a critical failure.

REQ-NF16: (WON'T) Emergency Medical Deliveries
The system won't allow users to order emergency medical deliveries unless approved in a separate program prior.

4+1 View Models

All of the 4+1 diagrams play a key role in the development process of a business. It is essential to at least create a simple version of each of these diagrams. These diagrams help developers to understand what they are trying to accomplish, and what key aspects need to be accomplished. These diagrams also help later if the company gets off track or is wondering why something got done a certain way they can refer back to their models to see if there was a specific reason. Whether or not to update these diagrams can be more complicated sometimes it is not worth the time and effort that goes into keeping these diagrams up to date. If any large scale decisions are made about the business they should be added to the diagrams in case the diagrams are being looked back on like I previously mentioned. Another reason to put more effort into keeping these diagrams up to date has to do with the size of your business. If your business is large enough to have multiple departments, and high-level decisions are being made and discussions that all departments are not present for; then your business should highly consider keeping updated diagrams for departments to refer to for questions and so no progress or key pieces of the business are lost.

Domain Diagram (*Figure 5*)



Domain Diagram Overview

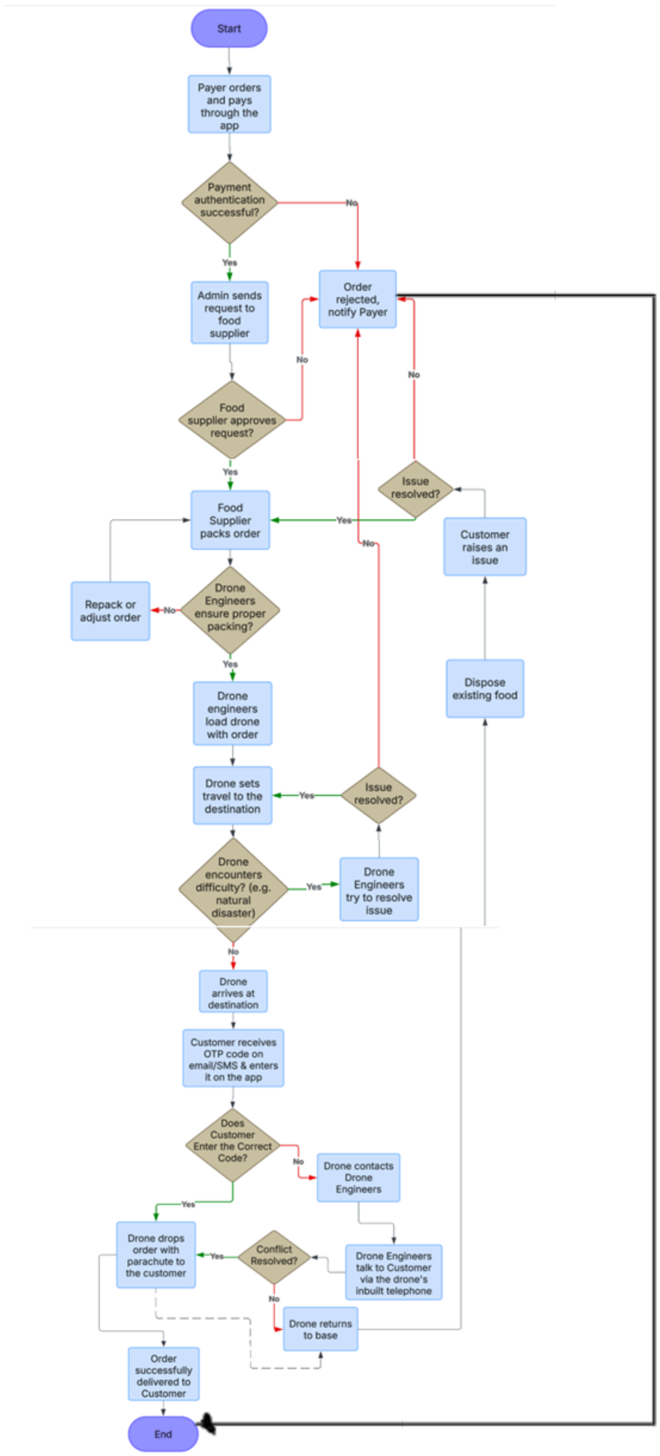
This domain diagram shows how the drone system, food supplier, and customer are able to seamlessly interact together through our app which is available on mobile and desktop devices. The customer and food supplier are given a different view of the enroute app. The customer is able to go to any store or restaurant that is also using enroute. The customer is then able to see any items they offer and select one they want delivered. An enroute drone is then sent to the location and the location is alerted that they have an order. The order is then fulfilled by the food supplier and delivered to the customer by drone. All payments or transactions are then completed on the app and the drone returns.

Enroute

Team 2

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

Activity Diagram (*Figure 6*)



Activity Diagram Overview

The main concern in this activity diagram is making sure we can deliver healthy food to remote or disaster-affected areas efficiently and reliably. This means we need to handle things like proper packing of enough food for the customer(s), overcoming obstacles such as natural disasters, ensuring timely deliveries, making sure the proper customer receives the order, and allowing the company and food suppliers an option to reject orders. This is why I had those things included in my activity diagram. This diagram shows how different actors, like the payer, customer, drone engineers, drones, and admin interact with the system and each other.

A key architectural decision in this diagram is the implementation of a reliable drone delivery system. This also improves delivery security and reduces failed/illegitimate deliveries. By having both, the admin and food suppliers verify if they can supply the order, the delivery of the accepted orders is assured. Additionally, validating that the correct customer receives the correct food also increases the system's reliability. Having drone engineers step in when intervention is required allows the drones to navigate through difficulties and further increases their chances of reaching the target location, thus improving reliability. Here are some key phases:

1. **Order Processing**

   The Payer places an order and completes payment from the app. The App Admin receives the request and decides whether to approve it. If the request is rejected, the Payer is notified and the process stops.

2. **Food Packaging**

   The food supplier packs the order. The food supplier head ensures correct food packaging. If improperly packed, then repack the food or adjust the order. Once correctly packed, load the drone with the package.

3. **Drone Deployment**

   Once the package is loaded, the drone sets off to the target location. If the drone encounters difficulties like natural disasters, then Drone Engineers attempt to resolve it. If they are unsuccessful, the order is called off and the customers are either re-delivered something else or the order is cancelled depending on the conflict.

4. **Delivery Verification**

   After reaching the target location, the customer is sent an OTP through email or SMS. They are tasked with entering this in the app when the drone arrives. They also enter the visible drone number. If the OTP and drone number match correctly, then the drone drops the food with a parachute so the customer can easily receive it. Otherwise, the customer is allowed to retry 3 times in total. After 3 failed attempts, the drone engineers' support team is called to talk with the customer. If the issue is still not resolved, the drone returns without delivering the food, assuming an error from the customer end.

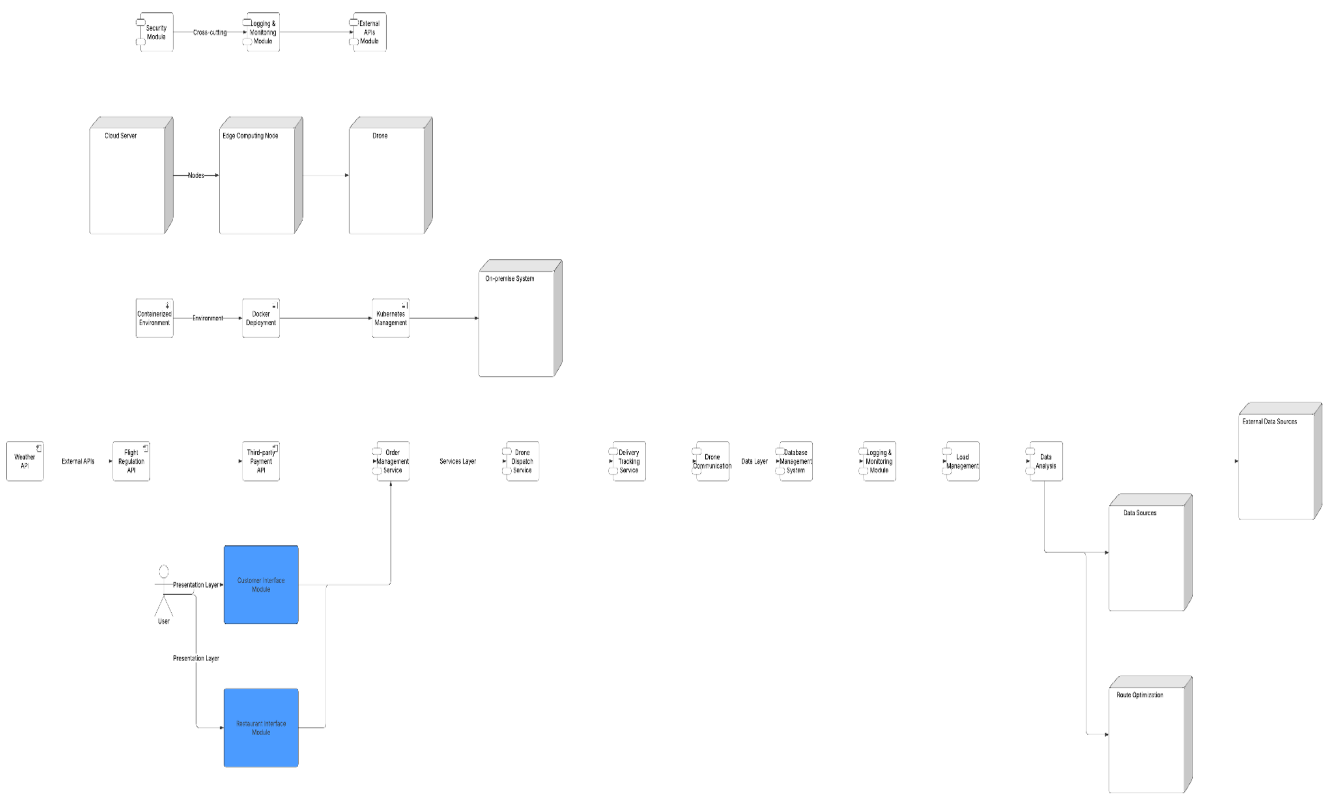
5. **Customer Support**

   If the drone returns without delivering and the customer believes there was an error on the company end, they may raise an issue. The customer support team attempts to resolve the issue. If the customer is at fault, they are marked with a point in the admin system. Too many points (3) result in stricter measures taken against the customer. If the customer is indeed correct, the company re-delivers on priority without additional fees. The company also provides compensation for the mistake.

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

---

Block Diagram (*Figure 7*)



Block Diagram Overview

The *drone-based food delivery system* is designed to provide efficient, real-time, and automated food delivery services, particularly targeting remote areas and disaster zones. This system integrates cloud computing, edge computing, containerized environments, real-time tracking, and AI-driven route optimization to ensure scalability, efficiency, and regulatory compliance.

The architecture is structured in multiple layers, including:

- **Presentation Layer**: User interaction via the Customer and Restaurant Interface.
- **Services Layer**: Core operations of order management and drone communication.
- **Data Layer**: Manages data storage, logging, load balancing, and analytics.
- **Nodes & Execution Environment**: Includes cloud servers, edge computing nodes, and on-premise systems for managing computation and drone coordination.
- **Cross-cutting Services**: Security, logging, monitoring, and external API management.
- **External APIs**: Integration with weather services, flight regulations, and payment gateways.

Enroute

Team 2

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

---

Stakeholders & Their Concerns

Stakeholder	**Primary Concerns**
Customers	Real-time order tracking, delivery reliability, and easy-to-use interface.
Restaurants	Efficient order processing, integration with dispatch, and accurate delivery tracking.
Drone Operators	Safety of drones, compliance with aviation regulations, and remote monitoring.
Regulatory Bodies	Compliance with flight regulations and weather safety measures.
NGOs/Disaster Relief	Reliable food supply chain to remote/disaster-struck areas.

Key Non-Functional Concerns Addressed

Scalability

- **Solution**: Cloud and edge computing nodes ensure the system can handle large-scale food deliveries efficiently.
- **Justification**: Enables smooth operations even when demand spikes in emergencies or high-order situations.

Real-Time Performance

- **Solution**: AI-driven route optimization and drone communication services ensure timely deliveries.
- **Justification**: Optimized routing and low-latency data processing improve delivery time.

Security

- **Solution**: A dedicated Security Module ensures secure payment transactions and data encryption.
- **Justification**: Protects customer payment information and prevents cyber threats.

Regulatory Compliance

- **Solution**: Flight Regulation APIs ensure drone routes comply with aviation safety standards.
- **Justification**: Avoids legal issues and ensures safe drone operation in restricted airspace.

Enroute

Team 2

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

---

Key Architectural Decisions & Rationale

1. **Cloud & Edge Computing for Distributed Processing**
   - **Decision**: A hybrid architecture with cloud servers and edge computing nodes.
   - **Rationale**: Cloud servers handle high-level processing, while edge nodes handle real-time drone communication for low-latency performance.
2. **Containerized Deployment Using Docker & Kubernetes**
   - **Decision**: Docker deployment for microservices with Kubernetes for orchestration.
   - **Rationale**: Ensures scalability, portability, and efficient resource management across different computing environments.
3. **AI-Powered Route Optimization**
   - **Decision**: Implement a route optimization module powered by AI.
   - **Rationale**: Ensures efficient drone navigation, minimizing travel time and avoiding no-fly zones.
4. **External APIs for Weather & Regulations**
   - **Decision**: Integrate Weather API and Flight Regulation API.
   - **Rationale**: Prevents drones from flying in hazardous weather and ensures legal compliance.
5. **Data-Driven Decision Making**
   - **Decision**: Implement a Data Analytics module for predictive analysis.
   - **Rationale**: Helps optimize drone dispatch schedules, order volume predictions, and route adjustments based on historical data.

Diagram-Specific Explanations

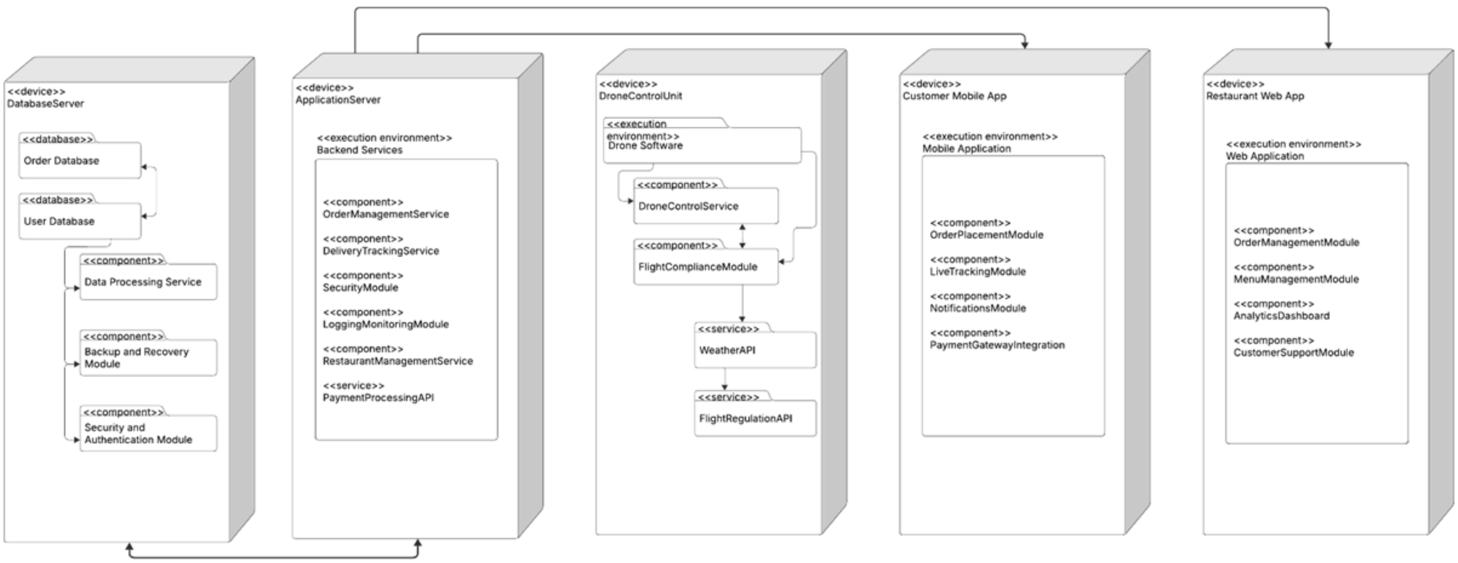
1. **Nodes & Execution Environment**:
   - **Cloud Server**: Handles overall system logic, coordination, and order processing.
   - **Edge Computing Node**: Manages real-time drone flight operations and sensor data processing.
   - **On-Premise System**: Handles localized order processing where cloud access is limited.
2. **Containerized Deployment**:
   - Introduces Docker and Kubernetes for scalable microservice management.
3. **Enhanced Data Layer**:
   - Data Analytics Module added for better decision-making and reporting.
4. **Route Optimization**:
   - Optimized AI-driven path planning added to improve delivery efficiency.

Conclusion

This architecture is designed to **scale, optimize, and secure drone-based food deliveries** while ensuring compliance with aviation regulations and real-time performance expectations. The **integration of cloud computing, edge processing, AI-driven routing, and secure APIs** ensures efficient delivery operations for both commercial and humanitarian applications.

Overview of the System Architecture

Deployment Diagram (*Figure 8*)



This deployment diagram represents the architecture of a drone-based food delivery system, detailing how different software components are deployed on specific hardware devices. The architecture ensures scalability, reliability, and security while enabling efficient order management, real-time tracking, and autonomous drone operations.

Motivation & Key Non-Functional Requirements Addressed

1. **Scalability** – The system is designed to support a growing number of users, restaurants, and drones without major architectural changes.
2. **Security & Compliance** – Ensuring authentication, authorization, and regulatory compliance for drone operations.
3. **Fault Tolerance & Data Integrity** – The system incorporates backup, recovery, and monitoring mechanisms.
4. **Real-Time Operations** – The architecture facilitates real-time order tracking and drone flight management.

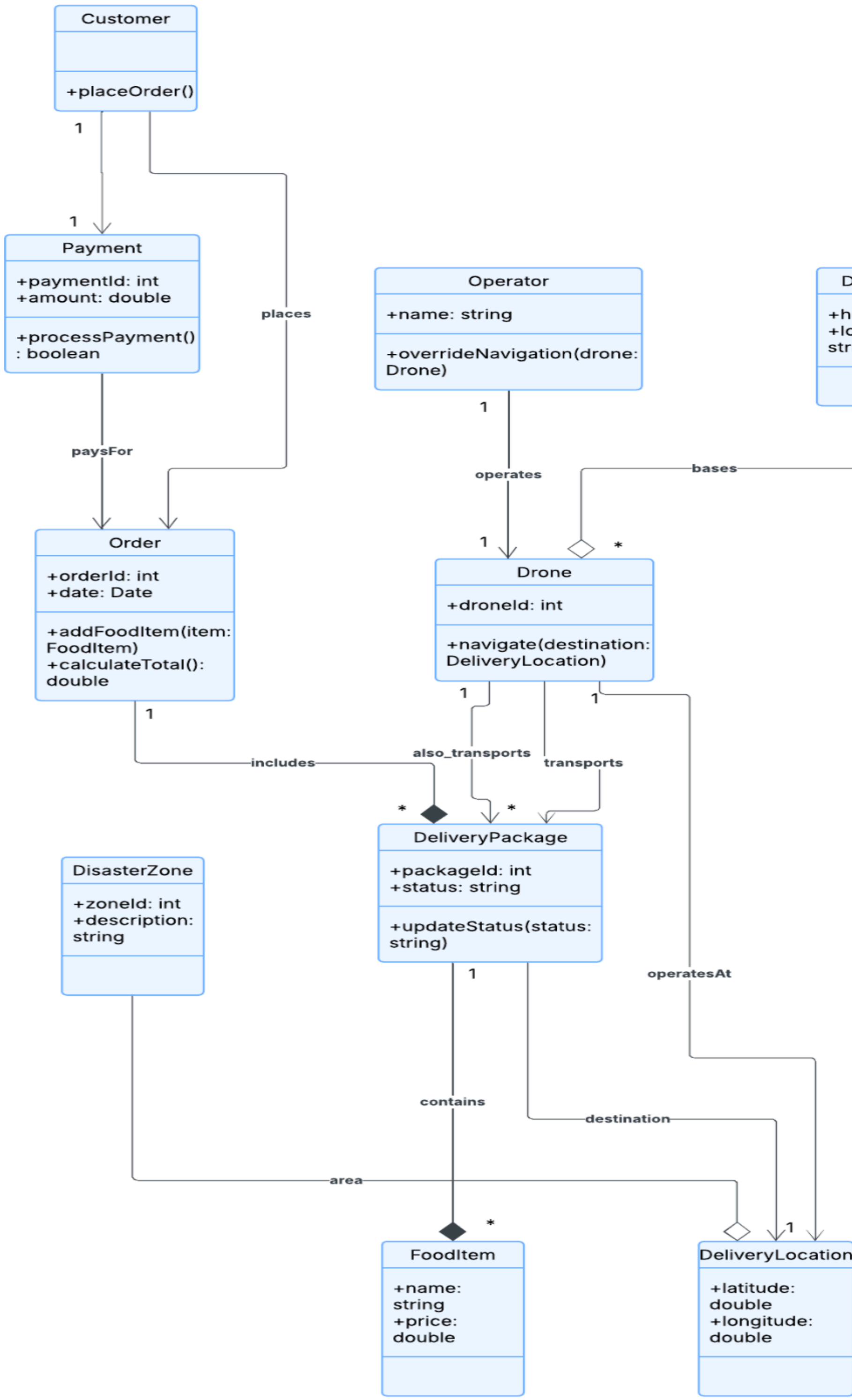
Quality Concerns of 4+1 Models for Software Developers

Quality Concerns	Approach
Scalability	In the block diagram it shows how a server communicates to the drones, and this easily can be scalable.
Accessibility	In the block diagram it shows how the user interface is a separate system, and this allows it to be made accessible without impacting the rest of the system.
Ease-of-Use	The domain diagram shows how the customer has simple steps: give location, give payment, and confirm order; to place an order.
Security	In the deployment diagram it shows how we store data and how to access important modules, so we can ensure authentication before granting access to sensitive information or resources.
Feasibility	In the activity diagram it shows the feasibility of the system and easily demonstrates how all of the actors interact.

Architectural Models

For the use case of **Placing an Order** specifically, we have created a class architecture diagram, robustness diagram, and sequence diagram. The **class diagram** shows the architecture of key actors and their attributes, such as the Drone with its ID attribute and navigates (destination: deliveryLocation) functionality. The **robustness diagram** helps visualize how objects interact when the customer is placing an order, with the help of actors, controllers, boundaries, and entities. The **sequence diagram** visually displays a timeline of how objects interact over time along with the order of messages.

Class Diagram (*Figure 9*)



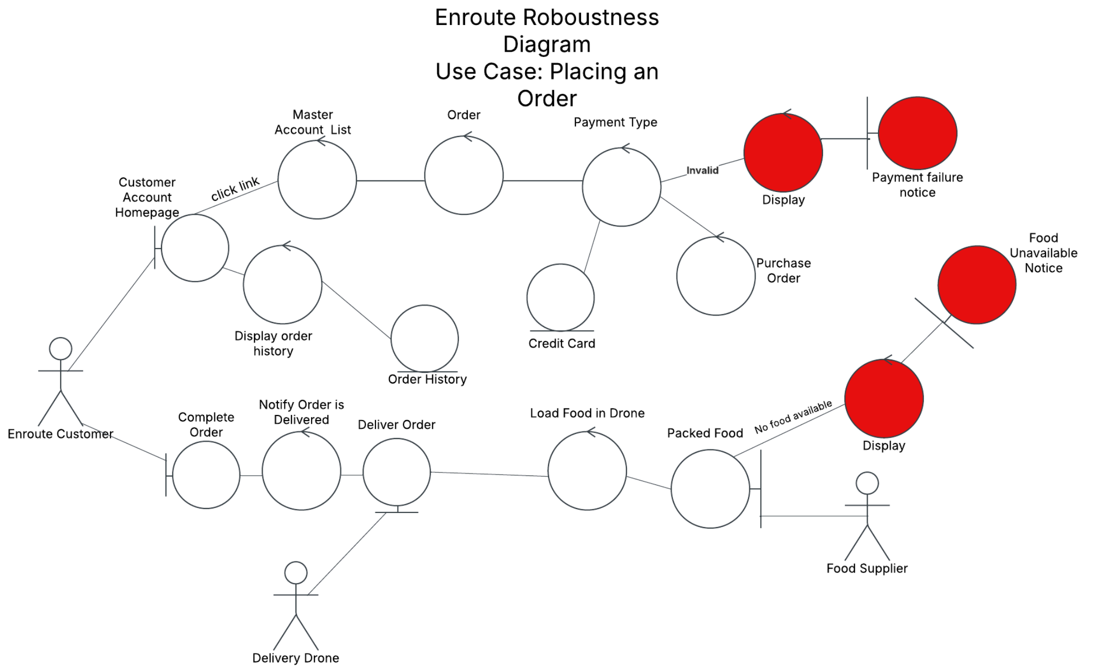
Enroute

Team 2

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

---

Robustness Diagram (*Figure 10)*



Sequence Diagram (*Figure 11*)

Detailed Proposal

App Design

The app design is further detailed in *Figure 11.*

Home

- Favorites menu with quick option selection for favorite restaurants or food items.
- Favorite restaurants show restaurant location & estimated time to deliver from there.
- Favorite food items show item pic, ingredients, restaurant making it, & estimated delivery time.
- Clicking on a restaurant/food item leads to a food menu.
- What's hot menu shows new promos & items (food ads) from restaurants.

Menu

- Filter based on restaurant, cuisine, & estimated delivery time.
- Click cart button to add to cart. Click as many times as desired quantity.
- Click on the green box for more information regarding the food item. Give restaurant food ratings from this page.

Cart

- Left bubble shows the quantity. Tapping increases. Double-tapping decreases count.
- Confirming order shows address & estimated delivery page. Can choose from saved addresses or add new.

Profile

- No UI design made yet, but this shows up as the customer picture if customized.
- Contains customer information such as name, demographics, user/password, address, friends, and past orders.

Settings

- No UI created yet but this shows the general app and account settings, & legal information regarding the business & app.

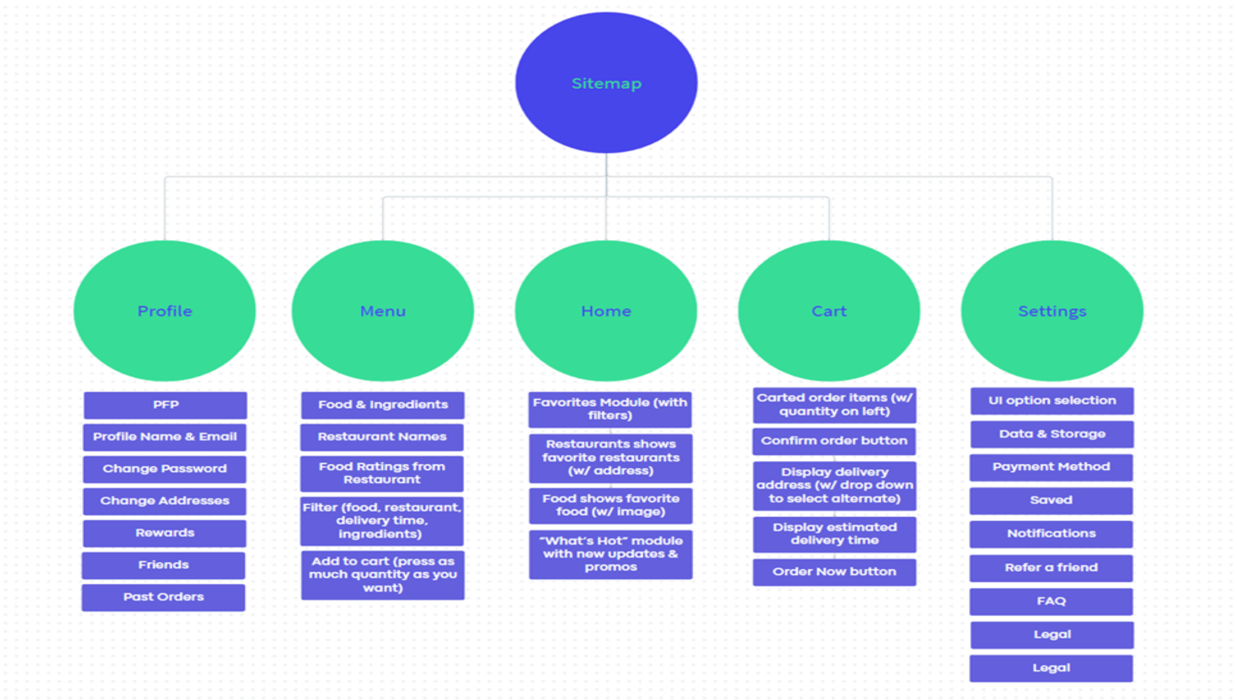
Enroute

Team 2

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

---

App UI/UX (*Figure 11*)



Enroute

Team 2

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

---

Performance Cost

Project Overview

The Drone-Based Food Delivery System aims to provide faster, cost-efficient, and eco-friendly food delivery services by using drones instead of traditional delivery methods. This system is designed to reduce delivery times, lower operational costs, and enhance customer satisfaction. It can also be expanded to include a government disaster relief contract, where drones are used to deliver emergency food, medical supplies, and essentials to disaster-affected areas.

The benefits are calculated based on expected demand, cost savings, and revenue generation over 5 years.

Benefit Category	**Estimated Value**	**Assumptions**
Faster Relief Delivery	50-70% reduction in response time	Compared to road-based aid delivery
Cost Savings per Emergency Operation	\$5,000 - \$20,000	Reduced reliance on helicopters/trucks
Government Contract Revenue	\$2M - \$5M annually	Based on federal/state funding for disaster relief
Commercial Order Growth	15-25% increase	Social impact improves brand image
Lives Impacted	50,000+ annually	Faster response reduces food/water shortages
Environmental Impact	40% reduction in CO_2 emissions	Less ground transport for aid

Estimated Additional Revenue per Year: \$3M - \$7M [1]
Estimated Cost Savings per Year: \$500K - \$1M [1]

Total Estimated Benefits (5 years) = \$17M - \$40M in revenue and savings. [1]

Enroute

Team 2

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

---

Estimated Project Costs

Initial Development Costs (Year 1)

Cost Category	**Estimated Cost ($)**
Software Development (AI for emergency routing, fleet management)	$750,000 - $1M
Hardware (specialized drones, solar charging hubs)	$1M - $2M
Regulatory Compliance (FAA, FEMA, government approvals)	$200,000 - $400,000
Training for Emergency Teams	$100,000 - $300,000
Total Initial Development Costs	$2.05M - $3.7M

Deployment & Operations (Annual)

Cost Category	**Estimated Annual Cost ($)**
Drone Maintenance & Emergency Readiness	$300,000 - $500,000
Server & Cloud Infrastructure	$200,000 - $300,000
Disaster Response Operations	$500,000 - $1M
Government Liaison & Compliance	$150,000 - $300,000
Marketing & Public Awareness	$250,000 - $500,000
Total Annual Costs	$1.4M - $2.6M

Total Estimated Costs (5 years) = $9M - $16M [2][3]

Cost-Benefit Comparison

Factor	**Value (5 years)**
Total Estimated Benefits	$17M - $40M
Total Estimated Costs	$9M - $16M
Net Profit / Savings	$8M - $24M

Break-even Point: 1.5 - 2.5 years
ROI (Return on Investment): 50% - 150% over 5 years [4]

Project Justification

Tangible Benefits

- **Increased Revenue**: Government contracts provide stable funding and commercial growth.
- **Faster Emergency Response**: Drones deliver aid within 30-60 minutes, compared to hours/days by trucks.
- **Reduced Costs**: Eliminates fuel and labor costs of traditional disaster logistics.
- **Regulatory Advantages**: Partnership with FEMA, Red Cross, and other agencies strengthens industry credibility.

Intangible Benefits

- **Corporate Social Responsibility (CSR):** Brand improves through humanitarian efforts.
- **Scalability:** System can extend to medical deliveries, military applications, and rural areas.
- **Public Trust & Media Coverage:** Positive impact increases support from government and customers.

Conclusion: By securing government contracts, this project achieves financial sustainability while making a humanitarian impact which will create a win-win situation for both business and society.

Security Analysis

This section covers the risks the companies face. We have a table that addresses a lot of the threats our company might face and evaluates them with the STRIDE and DREAD criteria *fig 12.* We also have a DFD that interprets the data flow throughout the ordering process of Enroute; it also shows the trust boundaries *figure 13.* Lastly we addressed some of our more serious threats in depth explaining the threats and how we would handle them.

Having possible threats can completely change a whole company's mission. Threats take the highest precedent because their fallout can be detrimental to the project. This would change a lot of the other diagrams by adding more steps to verify, block, avoid, and mitigate possible threats. Once the threats are identified the company must decide how crucial these threats are and take proper levels of precautions to avoid them. Some threats are very low-level and can just be ignored. These threats would likely not change any of the other artifacts. The larger threats can force the company to create completely new processes or change the missions of the company.

Threats and Risk Assessment (*Figure 12*)

STRIDE	Threat Example	How It May Occur	D	R	E	A	D	Rate	Decision
Spoofing	The attacker impersonates the customer to place an unauthorized order	Hacker steals customer credentials (via phishing, data leaks, etc.) and places order using that data for self	7	6	8	0	7	5.6	transfer
	Hacker spoofs a food supplier on a drone system to accept or modify orders	Hacker intercepts communication between the system & food supplier and sends fake confirmation messages (i.e. order is ready when it's not)	10	2	0	8	0	4	mitigate

STRIDE	Threat Example	How It May Occur	D	R	E	A	D	Rate	Decision
Tampering	Hacker modifies drone GPS data and changes drone's flight path	Hacker intercepts customer address data and modifies drone request to instead send food to self	9	0	0	9	1	3.8	avoid
	Hacker changes food options (adds/removes items) to manipulate a menu	Hacker gains access to food suppliers' database or UI & tampers the menu to remove items so no one can order it and hacker can have it	4	5	6	10	6	6.2	accept
Repudiation	Food supplier denies fulfilling an order, causing disputes	Food supplier logs in without authentication and falsely claims they didn't receive an order to avoid responsibility for delays, missing items, or other problems	2	7	10	5	5	5.8	transfer
	Customer fraudulently claims they never placed an order	Customer places an order & receives food but disputes the charge, claiming they never ordered (payment fraud)	4	10	10	1	7	6.4	accept
Information Disclosure	Customer personal data (name, email, phone, etc.) is leaked	Hacker gains access to the database & exposes sensitive customer details like names, emails, etc.	8	2	2	7	1	5	mitigate

Enroute

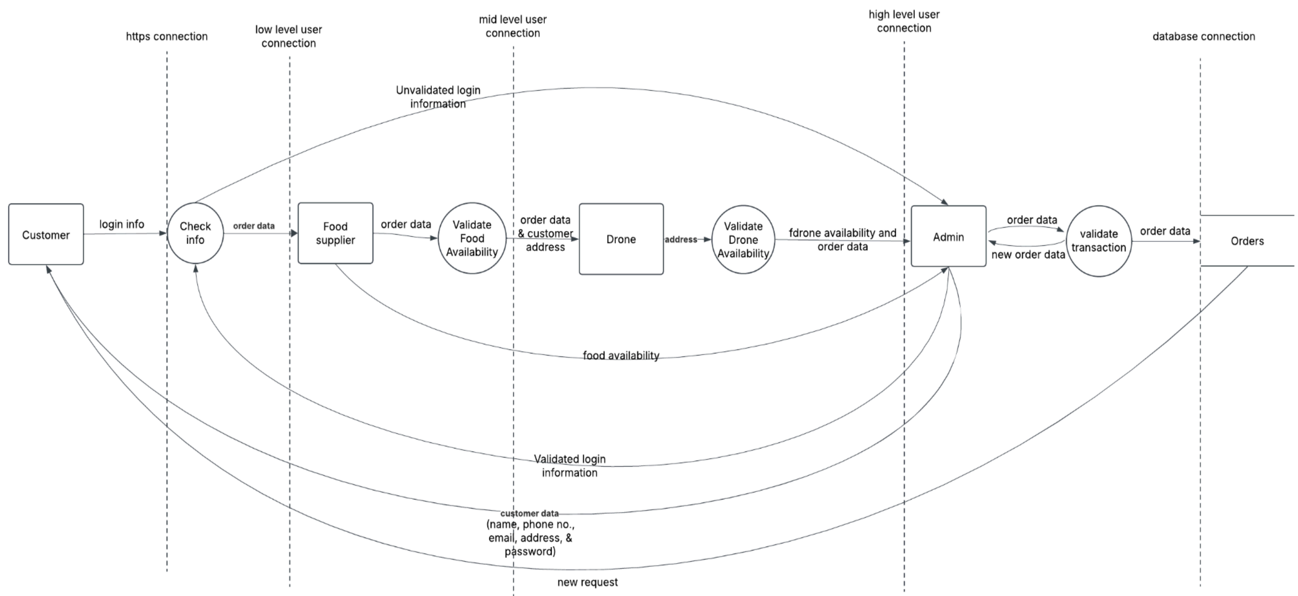
Team 2

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

---

STRIDE	Threat Example	How It May Occur	D	R	E	A	D	Rate	Decision
Information Disclosure	Hackers intercept drone GPS logs and track deliveries	Hackers intercept GPS logs in real time and track information such as customer addresses, drone routes, & drone bases to sell	10	0	0	10	0	4	mitigate
Denial Of Service	Fake orders flood & overload the system	Hacker sends multiple fake orders to cause server overload and timeout, preventing real customers from accessing the service	5	5	5	10	5	6	mitigate
	Customer faces service denial due to verification failure	Hacker blocks the verification process & prevents legit customers from verifying identity before placing orders	3	7	10	5	6	6.2	mitigate
Elevation of Service	Hacker gains admin access and modifies system settings	Hacker exploits admin login credentials to modify prices	10	0	0	10	0	4	avoid
	Hacker gains unauthorized access to drone operations & manipulates deliveries	Hacker exploits the drone control system to use them for malicious activities (i.e. terrorist attacks)	10	0	0	5	0	3	avoid

Data Flow Diagram (DFD) for Ordering Food (*Figure 13*)



Prioritization of Threats

1. **Tampering -** The hacker modifies drone GPS data and changes the drone's flight path

Decision: Avoid

Rationale: This threat is highly critical because it compromises the fundamental functionality and security of our drone delivery system. If an attacker can modify drone flight paths, they could redirect deliveries, steal food, or use the drones for malicious purposes. Avoidance is the best approach because mitigating or transferring the risk may not be sufficient to prevent such attacks, because mitigation would involve implementing security controls such as encrypted communication and GPS spoofing detection which might help, but may not fully stop advanced attacks. Transferring to insurance won't prevent service disruption or reputational damage. Acceptance won't work because the consequences are too severe, & drones could be misused.

2. **Repudiation -** Customer fraudulently claims they never placed an order

Decision: Accept

Rationale: While this threat can cause financial loss, it is difficult to fully prevent it without negatively impacting the customer experience. We can use fraud detection mechanisms(mitigation) such as verification codes, and order confirmation logs to help minimize disputes, but some level of fraudulent claims will likely still occur. Meanwhile, a transfer might help with the payment process, but fraud will still occur too. Avoidance such as strict verification would frustrate legitimate customers, so we will acknowledge this problem, but the financial impact is manageable and does not justify extreme security measures that could drive away our legitimate customers.

3. **Denial of Service -** Fake orders flood & overload the system
Decision: Mitigate
Rationale: Fake orders & overloading of the system cause severe service disruptions, thus it has a high DREAD rate of 6. It prevents real customers from using the drone food delivery service. Even though it is not as dangerous as the GPS hijacking attack, it severely impacts the business, customer base, suppliers, as well as natural resources like food. Mitigation is the best option in this case because the system can easily implement authentication to prevent automated order flooding and limit order rates to reduce risk without excessive costs. Avoidance would hinder even the real customers from accessing the service. Delegating/transferring to a 3rd party would be costly and instead, having a built-in mitigation system would be much more effective. Simply accepting the threat is the worst option because it would shut down the platform and servers. That is why mitigating would be the best course of action.

4. **Elevation -** Hacker gains admin access and modifies system settings
Decision: Avoid
Rationale: If a hacker gains admin access, they control the entire system and can do whatever they want. The hacker could manipulate prices, request drones to go wherever, delete records, or disable security protocols. This would harm the company's finances and reputation. As a result, this threat is the biggest one and should be avoided completely. The high-priority risk can be dealt with by enforcing strong access control mechanisms like MFA to prevent unauthorized access. Mitigating won't help prevent the attack in the first place, transferring admin security to a third party won't work because there needs to be security within the system, and accepting the threat would not work because it would compromise the whole system. Thus, avoiding it is the best course of action.

References:

[1] https://www.flyzipline.com/about/zipline-fact-sheet
[2] https://www.mckinsey.com/industries/logistics/our-insights/how-customer-demands-are-reshaping-last-mile-delivery
[3] https://www.nasa.gov/aeronautics/drones/delivering-closer-to-reality/
[4] https://www.wfp.org/wfp-drones
[5] https://www.swiggy.com/
[6] https://www.doordash.com/?msockid=0b63a66a9adb6d7e2f05b3279b616cf8

Artifact Change History

Enroute

Team 2

Members: Maxx McArthur, Clayton McArthur, Ishaan Shete, Suphachot Ratchworapong

---

Artifact	**Last Date of Change**	**Author**
Lean Canvas	January 16th, 2025	Maxx McArthur, Clayton McArthur, Suphachot Ratchworapong, Ishaan Shete
Persona 1	January 23rd, 2025	Maxx McArthur
Persona 2	January 23rd, 2025	Ishaan Shete
Use Case Diagram	January 28th, 2025	Maxx McArthur, Clayton McArthur, Suphachot Ratchworapong, Ishaan Shete
Use Case Scenario 1	January 28th, 2025	Clayton McArthur
Use Case Scenario 2	January 28th, 2025	Suphachot Ratchworapong
Functional & Non-Functional Requirements	February 4th, 2025	Maxx McArthur, Clayton McArthur, Suphachot Ratchworapong, Ishaan Shete
Domain Diagram	February 16th, 2025	Maxx McArthur
Activity Diagram	February 16th, 2025	Ishaan Shete
Block Diagram	February 16th, 2025	Clayton McArthur
Deployment Diagram	February 16th, 2025	Suphachot Ratchworapong
Class Diagram	March 12th, 2025	Ishaan Shete
Robustness Diagram	March 12th, 2025	Clayton McArthur
Sequence Diagram	March 12th, 2025	Clayton McArthur
Ux Design of App	March 12th, 2025	Ishaan Shete
Performance Cost	March 12th, 2025	Suphachot Ratchworapong
Security Analysis	March 12th, 2025	Maxx McArthur, Clayton McArthur
Data Flow Diagram	March 3rd, 2025	Maxx McArthur, Ishaan Shete