**EEX5362 - PERFORMANCE MODELLING   
Mini Project – Deliverable 01**

**HelaServe Food Delivery System**

**═══════════════════════════════════  
STUDENT INFORMATION  
═══════════════════════════════════  
Course: EEX5362 Performance Modelling  
Name: H.L. Nagolla  
Registration Number: 422513890  
Student ID: s22010108**  
**═══════════════════════════════════**

**Table of Contents**

[**1. System Overview 3**](#_Toc212827919)

[**1.1 Real-World Operational Context 5**](#_Toc212827920)

[**1.2 Key Performance Challenges 5**](#_Toc212827921)

[**1.3 Dispatch Strategy 6**](#_Toc212827922)

[**2. Data Set Description 6**](#_Toc212827923)

[**2.1 Dataset Overview and Structure 6**](#_Toc212827924)

[**2.2 Variable Definitions 7**](#_Toc212827925)

[**2.3 Dataset Files 7**](#_Toc212827926)

[**3. Sources 7**](#_Toc212827927)

# 1. System Overview

HelaServe is a food delivery system that helps to connect with everyone involved in the process. Customers place orders, restaurants make the food, couriers pick it up and deliver it and the platform keeps everything running smoothly. The system works step by step when a new order comes in, when food is ready, when a courier gets assigned or when the delivery is done.

The entire system of the HelaServe food delivery system can be considered as a four-sided market and the technology platform acts as the central agent and market-creator. The success of the platform is based on its ability to create value for and balance the conflicting interests of the other three sides.

1. Customers - As the source of all demand, customers are driven by a desire for comfort, variety, speed and value. Their experience is the final measure of service quality and it is based on a multitude of factors like usability of the mobile application, the span of restaurant choices, the accuracy of the delivered order and the total delivery time. Customer satisfaction is heavily based on the delivery time, with even small delays seriously impacting perception and loyalty.
2. Restaurants - These platforms represent a powerful way to extend their customer base and increase sales without investing in to own delivery infrastructure for restaurants. However, participation has some challenges like restaurants must agree with a commission fee, the logistical complexity of integrating delivery orders into their current workflow and the challenge of maintaining food quality and temperature during the delivery period.
3. Couriers (delivery Riders) - When managing the mobile and on-demand workforce, the couriers are the physical link of the logistics chain. Their primary task is to maximize earnings and it is based on the number of deliveries completed per hour, the distance travelled and customer tips and recommendations. They desire efficient routes, minimal waiting time between assignments and a fair and transparent system for order allocation. As independent agents their individual objectives, like preferring shorter and profitable trips are not always compatible with the platform's optimization goals of serving all customers efficiently.
4. The Platform - The platform is the backbone of the entire system and it provides the technological infrastructure and the algorithmic intelligence that arrange the entire operations. Its primary objectives are to increase market share, overall system efficiency and profitability. This is achieved by developing and managing the user-friendly applications, processing and developing payment gateways and implementing the dispatching, batching, and routing logic that is based on the system's performance.

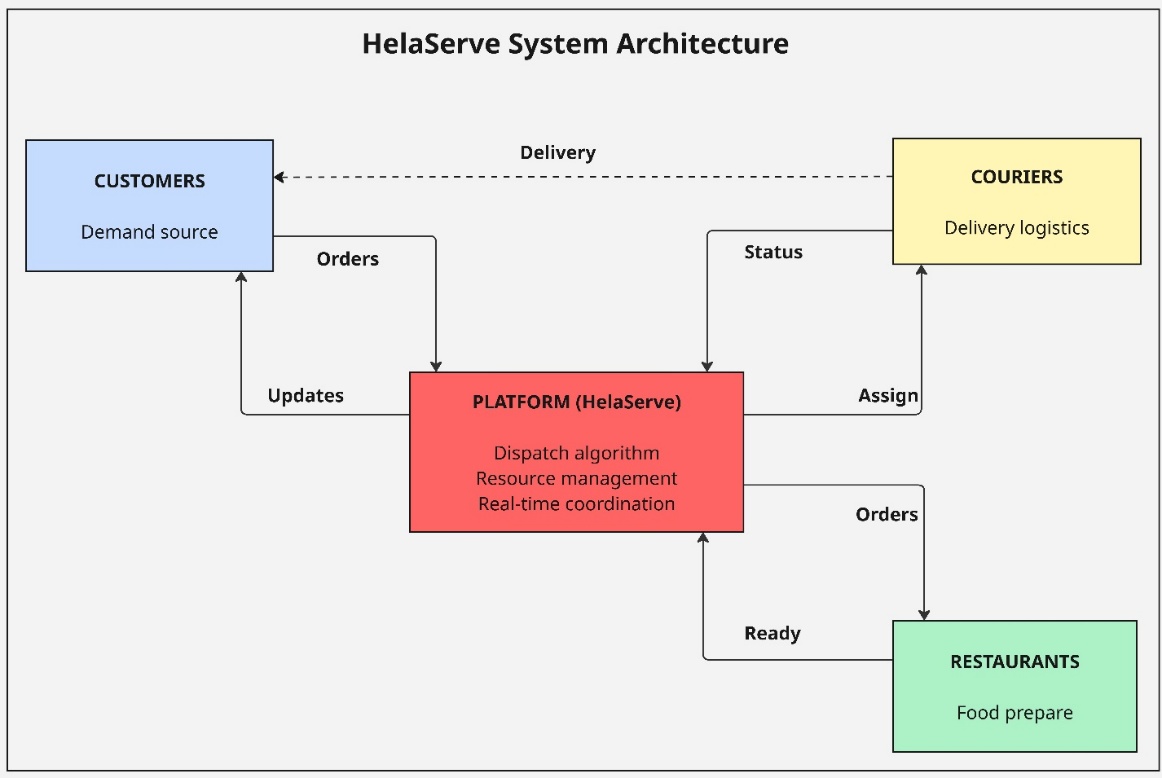


Figure : HelaServe System Architecture

## 1.1 Real-World Operational Context

Food delivery platforms deal with tricky problems because customer demand changes a lot, there are not always enough couriers and people want their food fast. HelaServe works in this kind of setting and tries to meet these real-world goals.

* Good service - Keep delivery times short and steady.
* Smart use of couriers - Make sure couriers are busy enough without being overworked.
* Fairness - Share the work fairly so no courier gets too tired.
* Growth ready - Manage busy times smoothly, even when lots of orders come in.

The system processes each order through a sequential process.

|  |  |  |
| --- | --- | --- |
| Stage | Description | Typical Duration |
| 1. Order Arrival | Customer places order via platform | Instant |
| 2. Food Preparation | Restaurant prepares ordered items | 10 - 20 minutes |
| 3. Courier Assignment | System assigns first available courier | 0 - 60+ minutes |
| 4. Delivery Transit | Courier travels to customer | 15 - 25 minutes |

The total delivery time equals to sum of preparation time, courier waiting time and travel time. This provides the primary performance metric for system evaluation.

## 1.2 Key Performance Challenges

1. Courier capacity Problem - When only 3 couriers handling all the orders, courier availability becomes a big issue. When all couriers are out delivering, new orders that are ready have to wait until someone is free. This waiting time often longer than cooking or driving, and it becomes the main reason for delays.
2. Order arrival changes - Customer orders come in at random times. Sometimes many orders show up quickly one after another. This can overload the couriers and cause a short-term back. Other times, there is a gap with no orders and couriers are idle, which wastes resources.
3. Service time uncertainty - Food prep usually takes 10 to 20 minutes and delivery takes about 15 to 25 minutes, but these times can change. This makes things harder. If food takes longer to prepare, a courier need to wait. If delivery takes longer, that courier can not pick up the next order in time. if These delays stack up. even when the system is not too busy but delivery times still change a lot.

## 1.3 Dispatch Strategy

The system uses a simple method to assign orders. When food is ready it gives the order to the first free courier. This nearest driver strategy focuses on quick action instead of finding the best overall plan. The workflow is,

When an order is ready

1. The system checks if any courier is free.
2. If one is free, the order goes to that courier.
3. If all are busy, the system picks the one who finishes next.
4. The order waits until that courier can pick it up.

**Strategy Advantages**

* Simple - Easy to use and does not need much computing.
* Fast - Makes quick decisions when couriers are free.
* Fair - Gives orders out in the order they are ready first come, first served.

**Strategy Limitations**

* No planning ahead - Does not move couriers to areas with future demand.
* Short term thinking - Focuses on one delivery at a time, not the whole system.
* Ignore distance - Does not look at how far couriers are from pickups or drop offs.
* Uneven work - Some couriers may get more orders just because of timing.

This method gives a basic way to measure how well other smarter strategies work. More advanced systems like ones that group orders, predict demand or balance workloads can be compared to this one to see how much better they do.

# 2. Data Set Descr**iption**

## 2.1 Dataset Overview and Structure

The HelaServe dataset has full order records from a simulated food delivery system that uses the nearest-driver dispatch method.

|  |  |
| --- | --- |
| Dataset Characteristic | Value / Description |
| Total Orders | 55 complete transactions |
| Number of Variables | 16 attributes per order |
| Start Time | 2025-10-28 11:00:00 AM |
| End Time | 2025-10-28 05:21:21 PM |
| Number of Couriers | 3 delivery persons |
| File Format | CSV |
| File Size | 8.20 KB |

## 2.2 Variable Definitions

The dataset contains 16 variables into four categories. identifiers, temporal data, operational metrics and business attributes.

|  |  |  |
| --- | --- | --- |
| Variable Name | Data Type | Description |
| Order\_ID | Integer | Unique order identifier (1-55) |
| Arrival\_Time\_Min | Float | Order arrival time (minutes from start) |
| Arrival\_DateTime | DateTime | Order arrival timestamp |
| Prep\_Time\_Min | Float | Food preparation duration (minutes) |
| Ready\_Time\_Min | Float | Time when order ready (min from start) |
| Ready\_DateTime | DateTime | Order ready timestamp |
| Courier\_ID | Integer | Assigned courier identifier (1-3) |
| Courier\_Wait\_Min | Float | Time order waits for courier (minutes) |
| Pickup\_Time\_Min | Float | Courier pickup time (min from start) |
| Pickup\_DateTime | DateTime | Courier pickup timestamp |
| Travel\_Time\_Min | Float | Delivery travel duration (minutes) |
| Distance\_KM | Float | Delivery distance (kilometers) |
| Delivery\_Time\_Min | Float | Delivery completion (min from start) |
| Delivery\_DateTime | DateTime | Delivery completion timestamp |
| Total\_Delivery\_Time\_Min | Float | End to end time (arrival to delivery) |
| Order\_Value\_LKR | Float | Order cost (Sri Lankan Rupees) |

## 2.3 Dataset Files

|  |  |  |
| --- | --- | --- |
| File Name | Format | Description |
| HelaServeSimulationDataset.csv | CSV | Primary dataset |
| Deliverable Report 01 | Word | This document (problem and dataset description) |

The dataset file (CSV) can open with Microsoft Excel, Google Sheet or other spreadsheet applications.

# 3. Sources

* GitHub Repository – [HelaServe GitHub](https://github.com/hasindu-nagolla/HelaServe-MiniProject)
* Dataset – [CSV File](https://drive.google.com/file/d/1nz8yRDHLT84_8eJAlK8qle6fHD37zF20/view?usp=sharing)
* Images – [System Architecture Diagram](https://drive.google.com/file/d/1e7tYSYW8Cuvewq5OunG5DclWvKGkhqWf/view?usp=sharing)