**EEX5362\_Deliverable 01**

**By: M.S.F.Nuha\_721434957**

**Food Delivery Dispatch System**

**System Overview**

The system being modeled is an urban food delivery dispatch system, similar to popular services like Uber Eats or PickMe. The main goal of the system is to efficiently match customer food orders with available delivery drivers. It seeks to balance speed, fairness, and resource use.

#### **Main Components**

* Customers - People who place food orders at different times throughout the day. Orders arrive randomly, just like in real life.
* Drivers - A limited group of delivery riders available to pick up and deliver food orders.
* Dispatch Algorithm - The “brain” of the system. It decides which driver gets which order.
* System Server - The central control unit that manages:  
  + Scheduling and tracking events (like new orders or completed deliveries)
  + Managing the queue of unassigned orders
  + Monitoring driver and order statuses in real time

**Key Performance Factors**

Several characteristics influence how well the system performs:

1. Order Inter-arrival Rate - How frequently customers place new orders, modeled using a Poisson process with random arrivals over time.
2. Service/Delivery Duration - How long each delivery takes, assumed to follow a uniform distribution that varies within a set range.
3. Driver Pool Size - The number of available drivers affects waiting times and delivery speed.
4. Dispatch Strategy - The algorithm used to assign drivers impacts efficiency and fairness.
5. Queue Length & Wait Time - How many orders are waiting to be assigned and how long customers wait before delivery starts, are key indicators of system performance.

### **Performance Objectives**

The food delivery dispatch system is designed and evaluated based on several key performance goals. These objectives ensure that the system not only delivers food efficiently but also remains reliable and scalable as demand increases.

| **Objective** | **Description** | **Performance Metric** |
| --- | --- | --- |
| **Minimize Delivery Time** | Ensure that customers receive their orders as quickly as possible once a driver is assigned. | Average Delivery Time (minutes) |
| **Minimize Queue / Wait Time** | Reduce the time an order spends waiting before being assigned to a driver. | Average Wait Time (minutes) |
| **Maximize Driver Utilization** | Keep drivers actively engaged in deliveries without excessive idle time, ensuring efficient resource use. | Driver Utilization (%) |
| **Ensure Scalability and Stability** | Maintain smooth system performance even as the number of orders increases significantly. | Throughput and Queue Growth Rate |
| **Identify Bottlenecks** | Detect system limitations or performance drops under different demand and load conditions. | *Queue Length vs. Order Arrival Rate* |

### **Dataset / Simulation Parameters**

The dispatch system was developed as a time-driven simulation using HTML and JavaScript (food.html). It models the real-world flow of food orders and deliveries, allowing key parameters to be adjusted through an interactive web interface.

Simulation Parameters

| **Parameter** | **Default Value** | **Description** |
| --- | --- | --- |
| Simulation Duration | 8 hours | Total time span for each simulation run. |
| Order Arrival Rate | 30 orders/hour | Average rate at which customer orders are placed. |
| Number of Drivers | 10 | Total number of available delivery drivers. |
| Average Delivery Time | 25 minutes | Mean time taken to complete a single delivery. |

#### **Simulation Outputs**

The system records several key performance metrics at the end of each simulation run:

* Total Deliveries Completed
* Average Delivery Time
* Average Wait Time
* Average Queue Length
* Driver Utilization (%)
* Throughput (orders/hour)

**Scenarios Evaluated**

To assess how the food delivery dispatch system performs under different conditions, several simulation scenarios were tested. Each scenario varies key parameters such as the number of drivers or order arrival rate to observe how the system handles changing demand levels.

| **Scenario** | **Parameters** | **Objective** | **Observed Behavior** |
| --- | --- | --- | --- |
| Scenario 1: Fewer Drivers | 5 drivers, 30 orders/hour | Test the system under limited driver availability (resource constraint). | Long order queues developed, and drivers were almost fully occupied with 97% utilization. |
| Scenario 2: Normal Load (Baseline) | 10 drivers, 30 orders/hour | Evaluate normal, balanced operating conditions. | System maintained steady performance with an average delivery time of 24.5 minutes and 80% driver utilization. |
| Scenario 3: High Demand | 10 drivers, 50 orders/hour | Test how the system performs under heavy customer demand (overload conditions). | Significant queue congestion occurred, and drivers operated near capacity with 99% utilization. |

### 

### **Bottlenecks & Recommendations**

Through simulation analysis, several performance bottlenecks were identified in the food delivery dispatch system.  
 Each bottleneck highlights a key limitation in the current setup and is paired with practical recommendations to enhance efficiency, scalability, and service reliability.

| **Identified Bottleneck** | **Suggested Improvement** | **Expected Impact** |
| --- | --- | --- |
| Driver Shortage at Peak Hours | Implement predictive scaling or flexible staffing (e.g., part-time or on-demand drivers during rush hours). | Helps maintain low queue times and prevent order backlogs during high-demand periods. |
| Static Dispatch Logic | Introduce a dynamic assignment algorithm that considers both driver distance and current availability instead of fixed or random matching. | Can reduce delivery delays by up to 25% and improve overall efficiency. |
| Random Demand Surges | Apply demand forecasting and zoning strategies to anticipate busy areas and allocate drivers proactively. | Enhances system stability and response time during unpredictable demand spikes. |
| Excessive Driver Utilization | Maintain driver utilization levels at or below 85% to allow flexibility for new orders and avoid burnout. | Prevents service collapse and supports scalable growth under varying workloads. |

### **Conclusion**

The study shows that the performance of an urban food delivery dispatch system is strongly influenced by two key factors: driver availability and order arrival rate. Keeping driver use between 80% and 85% provides the best balance of operational efficiency, delivery speed, and service stability.

The simulation-based approach was very effective for examining how the system behaves under different conditions. It helped identify potential bottlenecks and allowed for data-driven improvements in dispatch strategies and resource allocation before real-world deployment.

Overall, the findings emphasize that flexible staffing, dynamic dispatch algorithms, and demand forecasting are crucial for ensuring scalable, efficient, and customer-focused food delivery operations.