# PROJECT AND TEAM INFORMATION

## Project Title

**FOODFAST:** A Food Delivery System with Route Optimization in Java

Student / Team Information

|  |  |
| --- | --- |
| *TeamName:* | CodeSmiths |
| **Team member 1 (Team Lead)**  *(Last Name, name: student ID: email, picture):* | *Singh, Utkarsh Prakash – 23011007:* [utkarshprakashsingh00078612](mailto:priyanshujibansal@gmail.com)*[@gmail.com](mailto:priyanshujibansal@gmail.com)*  *WhatsApp Image 2025-04-07 at 14.47.06_58c497ac* |
| **Team member 2**  *(Last Name, name: student ID: email, picture):* | *Gupta, Shambhavi- 23012017:*  *gshambhavi1311@gmail.com*  WhatsApp Image 2025-04-06 at 17.07.40_6a215edb |
| **Team member 3**  *(Last Name, name: student ID: email, picture):* | *Bhatt, Vishal- 2023321:*  *vishalsingh1432vai@gmail.com*  *WhatsApp Image 2025-04-07 at 10.34.10_3bbae8bf* |
| **Team member 3**  *(Last Name, name: student ID: email, picture):* | *Bhandari, Ayush- 230211350:*  *[230211350@geu.ac.in](mailto:230211350@geu.ac.in)*  *WhatsApp Image 2025-04-07 at 13.36.33_859a6eb2* |

# PROPOSAL DESCRIPTION

## Motivation :

In the modern digital era, food delivery services have become an essential part of urban living. With increasing reliance on platforms like Swiggy, Zomato, and Uber Eats, the need for efficient, accurate, and timely order fulfillment has never been greater. While end-users enjoy the convenience of receiving food at their doorstep, the underlying systems that power these services involve complex logistics and decision-making processes—particularly in terms of route optimization and delivery assignment.

This project aims to simulate a simplified food delivery system using Java, focusing specifically on the logic behind selecting the most efficient delivery route. The motivation stems from real-world challenges faced by delivery platforms, such as minimizing delivery time, reducing travel costs, and ensuring optimal utilization of delivery agents. These challenges, when broken down, provide a valuable learning opportunity to apply core programming principles and data structures in a practical context.

By implementing basic routing techniques such as Breadth-First Search (BFS) and incorporating object-oriented design, this project offers a hands-on approach to understanding how everyday applications function behind the scenes. It provides a platform to practice modular code design, logical flow, and algorithmic thinking, while remaining accessible to beginners.

In essence, the goal is to bridge the gap between theoretical learning and real-world application through a project that is both relevant and educational. By developing this beginner-level food delivery system, students can better appreciate the importance of optimized decision-making in software systems, even at a foundational level.

## State of the Art / Current solution:

In current food delivery systems, platforms rely heavily on real-time tracking, GPS navigation, and advanced algorithms to optimize delivery routes and reduce wait times. These systems integrate mapping APIs and dynamic location data to assign orders to the nearest available delivery agent while also accounting for traffic, restaurant prep time, and customer location. However, implementing such functionality requires a combination of advanced technologies, which can be overwhelming for beginner developers.

This project aims to replicate the core concept of food delivery route optimization using a simplified, beginner-friendly approach in Java. Instead of real-time data or external APIs, we simulate a grid-based environment where delivery agents are mapped to static locations. The shortest delivery path is determined using basic graph traversal algorithms such as Breadth-First Search (BFS).

## BANKPRO Goals and Milestones

* The goal of this project is to simulate a beginner-level food delivery system that focuses exclusively on route optimization from a restaurant to customer addresses (homes or apartments). The system will identify the shortest delivery path across a predefined map of connected locations.
* Initial milestones include defining a basic delivery map structure, determining input/output requirements, and setting up the core Java classes for handling routing logic. As the project progresses, additional milestones include integrating a relational database using JDBC to store map and location data, and organizing code into logical packages for better modularity and readability.
* The end goal is to demonstrate a clean and functional CLI-based application that takes two locations (restaurant and customer) as input and outputs the shortest delivery path and total distance.

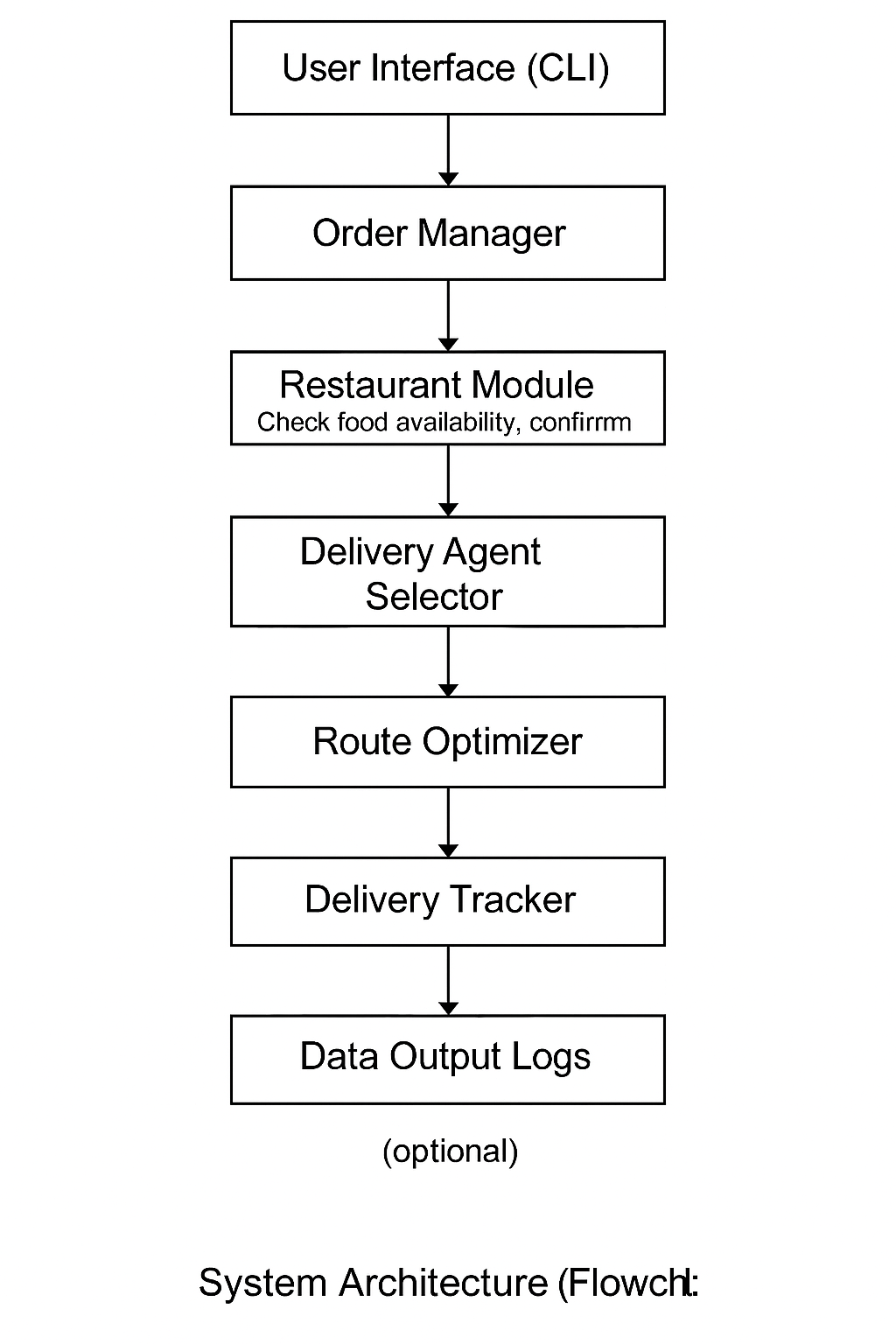
## 

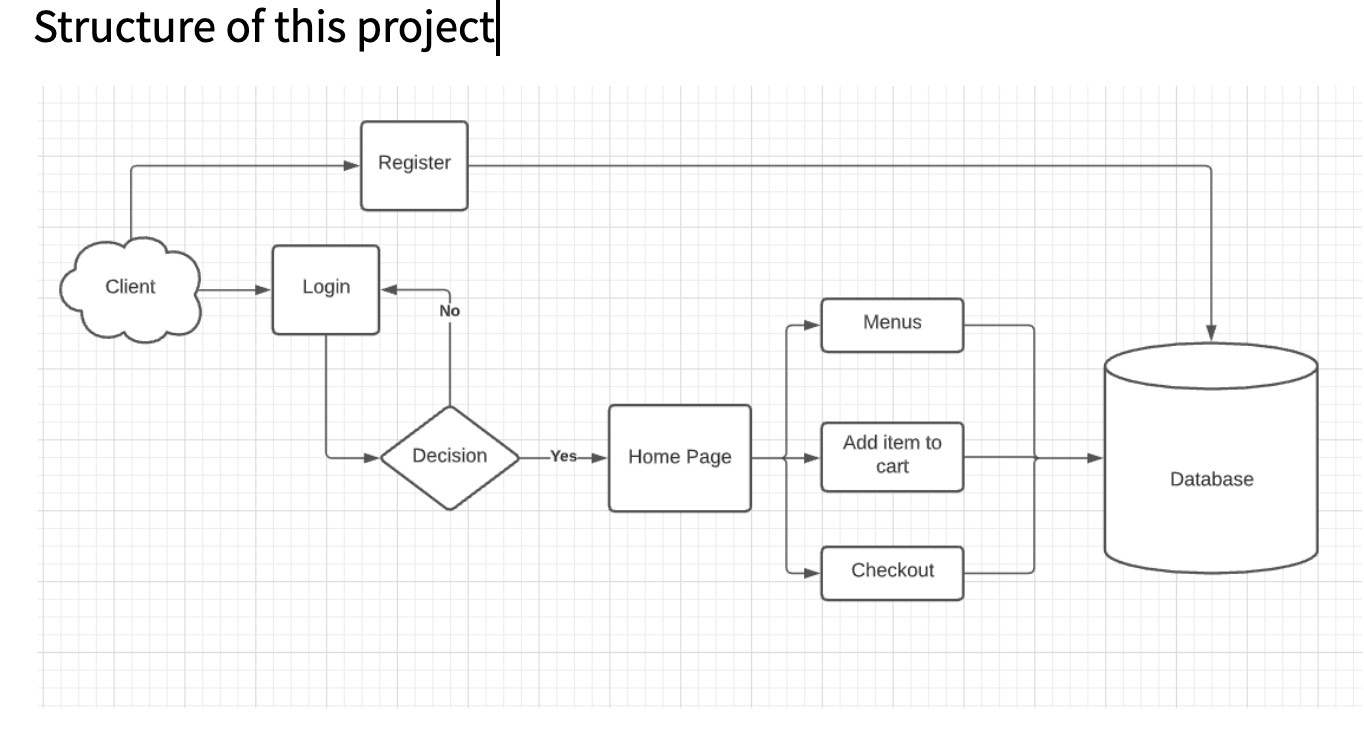
## Project Approach

* Requirement Analysis:  
  Define project scope: simulate delivery from a restaurant to customer homes using shortest path logic. Focus on static restaurant and customer data stored in a database.
* Database Setup (JDBC):  
  Use JDBC to connect the Java application to a relational database (e.g., MySQL or SQLite).  
  Store location data, delivery paths (edges), and customer address mappings in tables.
* OOP Design & Modular Structure (Packages):  
  Organize the project into packages such as:
* foodfast.db → Database connection and query handling
* foodfast.routing → Route optimization logic (e.g., BFS or Dijkstra)
* foodfast.model → POJO classes for Location, Route, Node
* foodfast.main → CLI launcher and main driver class
* Route Optimization Module:  
  Implement graph-based pathfinding logic using data retrieved via JDBC.  
  Input: Restaurant location and customer location  
  Output: Shortest path and delivery distance
* Command-Line Interface (CLI):  
  Build a simple CLI to allow user input of source and destination nodes (e.g., “Restaurant\_A” to “Apartment\_12”), and display the shortest route.
* Testing & Debugging:  
  Use sample map data to verify route accuracy, database integration, and edge cases.
* Final Documentation & Demo:  
  Document the system architecture, package layout, database schema, and sample executions for demonstration.

## 

System Architecture (High Level Diagram)





## 

## Project Outcome / Deliverables

* The primary outcome of this project is a functional, beginner-level Java application that simulates food delivery route optimization using the shortest path logic. The application will allow users to input a source (restaurant) and destination (home/apartment), and it will compute and display the most efficient delivery route along with the total distance.
* Key deliverables include:

1. A modular Java codebase organized using packages for routing logic, database operations, and models.
2. A working database (MySQL or SQLite) integrated via JDBC to store location and distance data between connected nodes.
3. A command-line interface (CLI) that provides user interaction for selecting delivery routes and viewing results.
4. Implementation of a shortest path algorithm (BFS or Dijkstra) to demonstrate core routing logic.
5. Sample test data and scenarios to verify routing accuracy.
6. Documentation including system architecture, class diagrams, database schema, and instructions for running the application.

* This project provides a foundational understanding of how data structures and algorithms can be applied to real-world problems and serves as an educational resource for learning routing, modular Java design, and basic database integration.

## References

Oracle Java Documentation  
[https://docs.oracle.com/javase/tutorial/](https://docs.oracle.com/javase/tutorial/" \t "_new)

GeeksforGeeks – Dijkstra’s Algorithm in Java  
https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-in-java/

GeeksforGeeks – Breadth-First Search (BFS) in Graph  
https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/

Java JDBC Tutorial – JavaPoint  
<https://www.javatpoint.com/java-jdbc>

Java Code Examples for Graph Representation  
https://www.programiz.com/dsa/graph-adjacency-list

Object-Oriented Programming Principles – Java OOP Concepts  
https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/

·