SE3020 – Distributed Systems
BSc (Hons) in Information Technology Specialized in Software Engineering
Year 3 – Semester 1
2025 – Assignment 1

Assignment Duration – 5 Weeks

Title: Building a Cloud-Native Food Ordering & Delivery System using Microservices

Note: This assignment carries a weightage of **25%** towards the final grade. It is a **group project** with **3 to 4 members**.

You have been asked to develop a **food ordering and delivery platform** similar to **PickMe Food, UberEats,** which allows customers to order food from multiple restaurants and get deliveries efficiently. Below are the requirements provided by the client and/or the Business Analyst.

Requirements:

- **Web/Mobile Interface:** Develop a **web/mobile interface** where customers can browse restaurants, add food items to a cart, and place orders. Ensure the interface is **user-friendly and supports various devices**.
- **Restaurant Management Service:** Implement a service where restaurant owners can add, update, and delete menu items. Restaurants should be able to manage order availability.
    - **Restaurant role** – Add, update, and delete menu items, set restaurant availability, manage incoming orders.
    - **Admin role** – Manage user accounts, verify restaurant registrations, handle financial transactions.

- **Order Management Service:** Develop a service where customers can place orders, modify them before confirmation, and track the status of their orders.
- **Delivery Management Service:** Implement a system that **automatically assigns delivery drivers** based on order location and availability. Customers should be able to **track deliveries in real-time**.
- **Payment Integration:** Integrate **secure online payment gateways** to facilitate order payments. Use **Sri Lankan third-party services** such as **PayHere, Dialog Genie, or FriMi** (or an internationally recognized service such as **Stripe or PayPal (sandbox environment)**).
- **Confirmation and Notification:** Upon successful order placement, customers should receive **confirmation via SMS and email.** Delivery personnel should also receive **real-time notifications** regarding assigned orders. Utilize **third-party SMS and email services** for sending notifications.

You may add new functionalities other than those mentioned in the description.

Implementation:

1. **Based on the provided requirements, develop a set of RESTful web services** to implement the food delivery platform. You may choose any technology stack to implement the services. Ensure the services are designed following **REST principles**, maintaining **scalability, security, and performance**.
2. You must use the **Microservices architecture** to develop/integrate the API. Ensure that you use **Docker and Kubernetes**. If you are using any other tool for **Microservices orchestration/integration**, you may justify that in the report and during the viva.

3. Develop an **asynchronous web client** using any **JavaScript framework** that supports asynchronous programming (such as **Angular, React, etc.**) or use **regular jQuery + AJAX**. However, for the scope of this assignment, implementing just an **asynchronous web client** is sufficient.
4. Use appropriate **security/authentication mechanisms** to uniquely identify each user and authenticate them. There should be **three roles:**
    - **Customer** – Browse, order food, track deliveries.
    - **Restaurant Admin** – Manage menus, accept orders, handle payments.
    - **Delivery Personnel** – Accept and fulfill deliveries.

Deliverables:

1. A text file called **submission.txt**, containing a **GitHub repository link** with all the source code. The source code should contain the **backend services, client, and any other relevant source/resource files** (e.g., database scripts), arranged in a proper directory structure.
2. The **submission.txt** should contain a **YouTube video link** of a presentation/demo of the project. Each member may use **a maximum of 3 minutes** to explain their contribution, so the total video length **should not exceed 12 minutes**.
3. A **readme.txt** document, listing down the steps to deploy the above deliverables.
4. A **members.txt** file, containing the names, registration numbers, and the IDs of the group members.

5. A **report in PDF format (report.pdf)**. The report should include:
    - A **high-level architectural diagram** showing the services and their interconnectivity.
    - A list of **interfaces (NOT the user interfaces, but the service interfaces)** exposed by each service.
    - A brief explanation of **each workflow** used in the system (you may use design diagrams of your choice to do this).
    - **Details about authentication/security mechanisms** adopted.
    - **Individual contributions** of each group member.

You may use **code snippets** in the report to explain the above.

The **report must have an appendix** with all the code that you have written (**excluding the auto-generated code**). **Do not paste screenshots of the code in the appendix**; instead, copy the **code as text**. If screenshots are added in the appendix, **only the minimum mark may be offered**.

Note: All reports will be uploaded to **Turnitin** for plagiarism checking. If the **Turnitin similarity** is **above 20%**, marks will be penalized.

Submission Details:

- **All files should be uploaded in a single ZIP archive**. The ZIP file name should be **Group ID_DS-Assignment.zip**. **Only one member** needs to upload the submission.
- **Submission Deadline: 11th week of the semester**.
- **Viva Session: 13th** and **14th week of the semester**.

Marking Rubric

Criteria	**Allocated Mark**
Application of **Microservices principles** in the architecture and the design	**20**
Having clearly defined **interfaces**, that facilitate reusability	**10**
Use of proper **payment integration mechanism and SMS, Email Notification**	**10**
Quality and readability of the code, with meaningful and detailed comments	**10**
Integration/Orchestration of services using **Docker + Kubernetes**	**20**
Adoption of appropriate **authentication/security mechanisms**	**10**
Comprehensiveness and quality of the report	**20**
Total Marks	**100**