Faculty of Technology, Design & Environment

Assessment cover

OXFORD
BROOKES
UNIVERSITY

Module No:	COMP5047	Module title:	Applied Software Engineering

Assessment title:	Resit Coursework - Software Engineering of a Modern Computer Application

Due date and time:	9:00am, 14th April, 2025

Estimated total time to be spent on assignment:	84 hours per student

LEARNING OUTCOMES

On successful completion of this assignment, students will be able to achieve the module's following learning outcomes (LOs):
1. Demonstrate an understanding of the role of requirements analysis and specification in software engineering and to be able to use this knowledge to create use case models and functional models of computer applications.
2. Demonstrate an understanding of the relationship between requirements and design and to be able to apply the knowledge to create structural and behavioural models of computer applications.
3. Critically evaluate and utilise design paradigms of object-oriented analysis and design, component-based design, and service-oriented design.
4. Use software modelling language such as UML and modelling tools in the context of model-driven software engineering.
5. Work in a group to apply the knowledge and skills developed in this module

Engineering Council AHEP4 LOs assessed	
C3	Select and apply appropriate computational and analytical techniques to model complex problems, recognising the limitations of the techniques employed
C5	Design solutions for complex problems that meet a combination of societal, user, business and customer needs as appropriate. This will involve consideration of applicable health & safety, diversity, inclusion, cultural, societal, environmental and commercial matters, codes of practice and industry standards
C6	Apply an integrated or systems approach to the solution of complex problems
C14	Discuss the role of quality management systems and continuous improvement in the context of complex problems
C16	Function effectively as an individual, and as a member or leader of a team

Student Name: nabhan **Student Id:** 19268021 **Subsystem:** operation

Statement of Compliance *(please tick to sign)*

🟩 I declare that the work submitted is my own and that the work I submit is fully in accordance with the University regulations regarding assessments *(www.brookes.ac.uk/uniregulations/current)*

School of Engineering, Computing & Mathematics

RUBRIC OR EQUIVALENT:

Marking grid and marking form are available on Moodle website of the module.

FORMATIVE FEEDBACK OPPORTUNITIES

(a) Discuss your work with your practical class tutor during practical classes; (b) Discuss your work with lecturer and/or practical class tutor in drop-in hours.

SUMMATIVE FEEDBACK DELIVERABLES

Deliverable content and standard description and criteria
Please see attached file of *COMP6030 Coursework Marking and Feedback* for feedbacks on your coursework, which include:
(a) Breakdown of marks on each assessment criterion
(b) Comments on each aspect of the assessment against assessment criteria
(c) Annotations on your submitted work

COMP5047 APPLIED SOFTWARE ENGINEERING
RESIT COURSEWORK
Task-1 Project Management

System operators can manage the mobile restaurant booking system for all the restaurants using the CloudTables-Operations subsystem. Such capabilities, including account activation/deactivation, cross system analytics, promotion setup, customer social features and loyalty systems, fall under this. The following tools and methodologies have been used for effective development and planning of this subsystem.

All development work was done in a GitHub repository and tracked:
https://github.com/Nabhan-ali20/rest-5047-operation

UML Modelling Tools:

- Lucidchart
- Draw.io
- Smartdraw.com

Task-2 Specification of Quality Requirements

Security and Privacy Protection:

A multi layered security architecture is very important for CloudTables-Operations subsystem which deals with sensitive system wide data (e.g. cross restaurant analytics, customer details, account control).

- **Data transmission:** All API communications will be encrypted using TLS 1.3 to secure system operator actions such as data reporting, promotions, account control.
- **Authentication and Authorization:** System operators will log in using OAuth 2.0, and sensitive actions like exporting data or disabling accounts will need Multi-Factor Authentication (MFA), using time-based codes (TOTP) for extra security.
- **Data Privacy Compliance:** GDPR Compliance Data Privacy Compliance: Operator dashboards will allow the data to be GDPR compliant by allowing access to identifiable customer data. Operator actions will be tracked in access log and audit trails. Datal exported and retained based on strict rules, auto deletion or anonymisation after 12 months of inactivity.
- **Access controls:** All endpoints will have token based access control and abuse or unauthorized automation will be limited by IP based throttling

Performance:

Benchmarks for Responses: 95% response time for Dashboards and analytics APIs will be 2 seconds. Pagination and pre aggregation will be used to optimize data visualizations and statistics queries.
Load handling: The system will support at least 100 concurrent operators during peak reporting times and auto-scaling backend services using AWS EC2 and RDS.

Caching: Redis will cache analytics results and common queries to reduce backend pressure.
Asynchronous Processing: Long running tasks such as report generation, data export will run asynchronously and in the background with the help of Celery or Rabbit MQ task queues.
Reliability:
The subsystem is designed to be high available, handle errors in a graceful manner, and have testable failover procedures, so that operations can continue without interruption.

- **Availability:** Operations Subsystem will be hosted on a multi-AZ cloud infrastructure with a target 99.9% uptime. Traffic rerouting will be verified during simulated failure events of AWS Load Balancers and Auto Scaling Groups as we move to production.
- **Error Handling**: Rollback mechanism and retry logic with exponential backoff will be enforced for critical workflows such as promotional setup or account deactivation and be tested via test cases. The edge cases for form inputs, submitting and submitting with failures and abnormal user actions will be covered by automated unit and integration tests.
- **Response Times Monitoring & Alerts:** Their response times will be monitored via AWS Cloud Watch and Sentry and alerts will be configured sent to developers and operators when response times spike or services fail. Unit testing and validation of alert thresholds will occur using artificially created stress scenarios.

Scalability
To support growing usage and new functionality across multiple restaurants and operators, the system is built for elastic scaling and modular deployment:

- **scalable Infrastructure:** Deployment and testing of Dockized services will be performed using scalable infrastructure. Metrics will drive resource allocation per pod that will be readjusted.
- **Elastic Cloud Deployment'**s EC2 and RDS Auto Scaling Policies will be tested by making it auto scale based on the user traffic spikes. Scaling will be confirmed in logs and dashboards within SLA defined thresholds
- **database Sharding & Indexing:** Before and after read replicas, sharding massive datasets (e.g. feedback or event logs) and applying indexes to highly computed fields is performance benchmarked
- **Microservices Architecture:** Functional services (analytics, promotions, loyalty programs) will be deployed independently, advanced independently and tested using their integration pipelines, to check that version updates and scalability of one service don't impact another

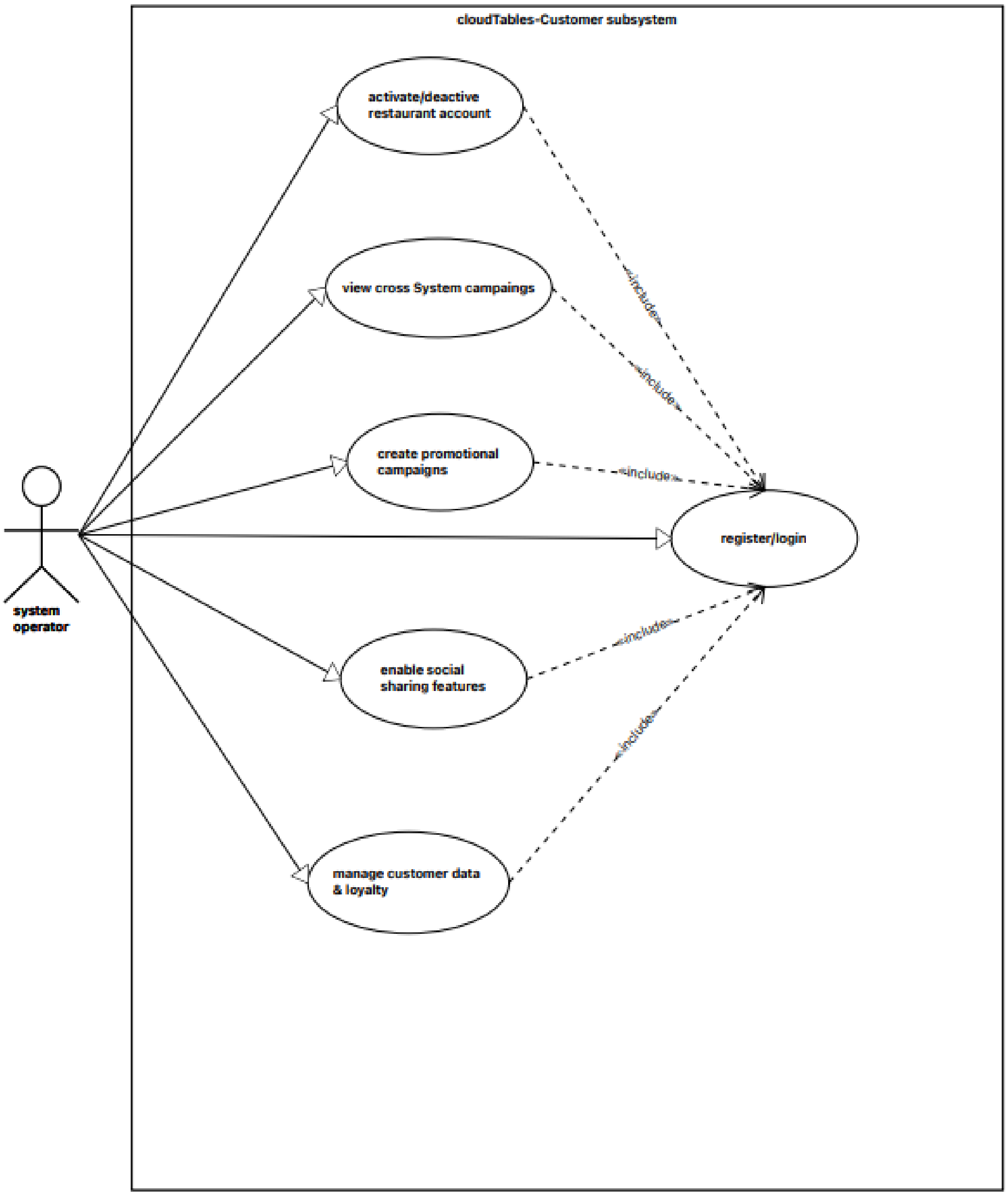
Verifiable implementation steps such as automated testing, performance monitoring, and scalability validation are used to prove the support for these quality attributes, which together keep the load of CloudTables-Operations secure, stable, and flexible in reality.

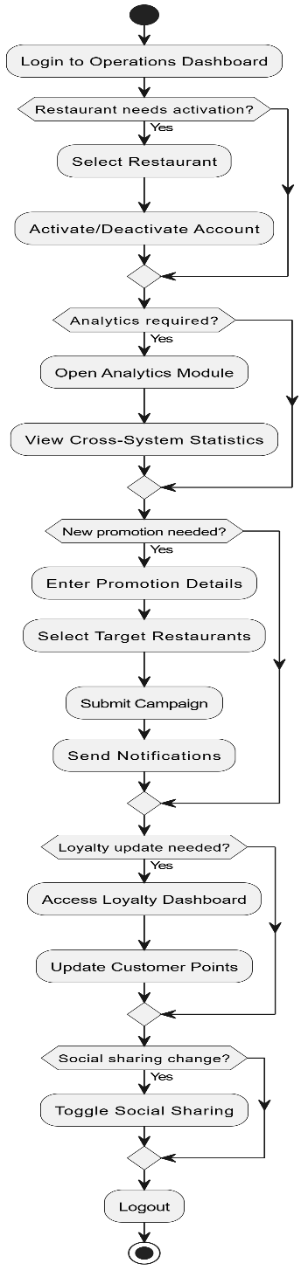
Task 3: Software Modelling and Specification

Use case model:

- **User Login and Registration:** this is used by all other use cases as it is needed to access the application.
- **Activate/Deactivate Restaurant Account:** Operators can update customer profiles, adjust loyalty points, and handle redemptions through this feature.
- **View Cross-System Statistics:** enables the operator to view performance data and usage trends across the system
- **Create Promotional Campaign:** They can also create time-limited promotions and apply them to selected restaurants.
- **Manage Customer Data & Loyalty:** allows operator to handle customer related information and loyalty point system

The below diagram was made using Apollon (available on IOS App store)


cloudTables-Customer subsystem
activate/deactive restaurant account
view cross System campaings
create promotional campaigns
register/login
enable social sharing features
manage customer data & loyalty
system operator
«include»
«include»
«include»
«include»
«include»



Activity model diagram

- The system operator logs in to Dashboard using secure credentials.
- If needed, the operator chooses the restaurant and toggles its active status.
- The operator can view real time cross system statistics such as booking trends or customer engagement.
- If a new campaign is needed, they fill in promo details, select restaurants and notify them.
- Customer loyalty points or rewards can be manually updated by the system operator.
- Social networking features can be enabled or disabled globally (FR-SO-4).
- Logout: Ends the operator’s session securely.

Diagram made using(plantUml feature in draw.io)

<u>Task 4 software architectural design</u>

CloudTables-Operations Subsystem Component Diagram

Overview

This is the component diagram of the CloudTables-Operations Subsystem, which is responsible for the backend operations of the CloudTables system. The diagram depicts the different components interacting with each other focusing on user authentication, management of restaurants, promotions, statistics, loyalty programs, notifications and data storage.

Components

Authentication Service: This service checks user login credentials and only grants access if the user is on the authorized list.. It verifies login credentials, succeed, and other components that allows to access to other components.

Restaurant Account Manager: It manages the restaurant specific data like account creation, update, deletion. It communicates with the Database Service to save the restaurant information, so that the system holds the current statuses of the restaurant.

Statistics Engine: It processes and produces statistical data about the performance of the restaurants and user interactions. This component retrieves data from the Database Service to gain insights into the restaurant activity and performance metrics.

Promotion Manager: It manages promotional campaigns like creation, configuration and tracking. It notifies the relevant parties about the promotions and interacts with the Notification Service.

Social Sharing Service: It allows users and restaurants to share promotional offers or other important details of the restaurant through social media platforms. It also talks to the Notification Service to send notifications to the users about social sharing opportunities.

Loyalty Management Service: Manages loyalty points, reward offers and redemption options for the customer loyalty programs. It guarantees that customer engagement is recorded in the Database Service and can be used in the future. Notification Service Through certain actions in the system (promotions, restaurant updates, social media shares), it sends notifications to users or restaurant operators. It works in close cooperation with the Promotion Manager and the Social Sharing Service to let the right people know.

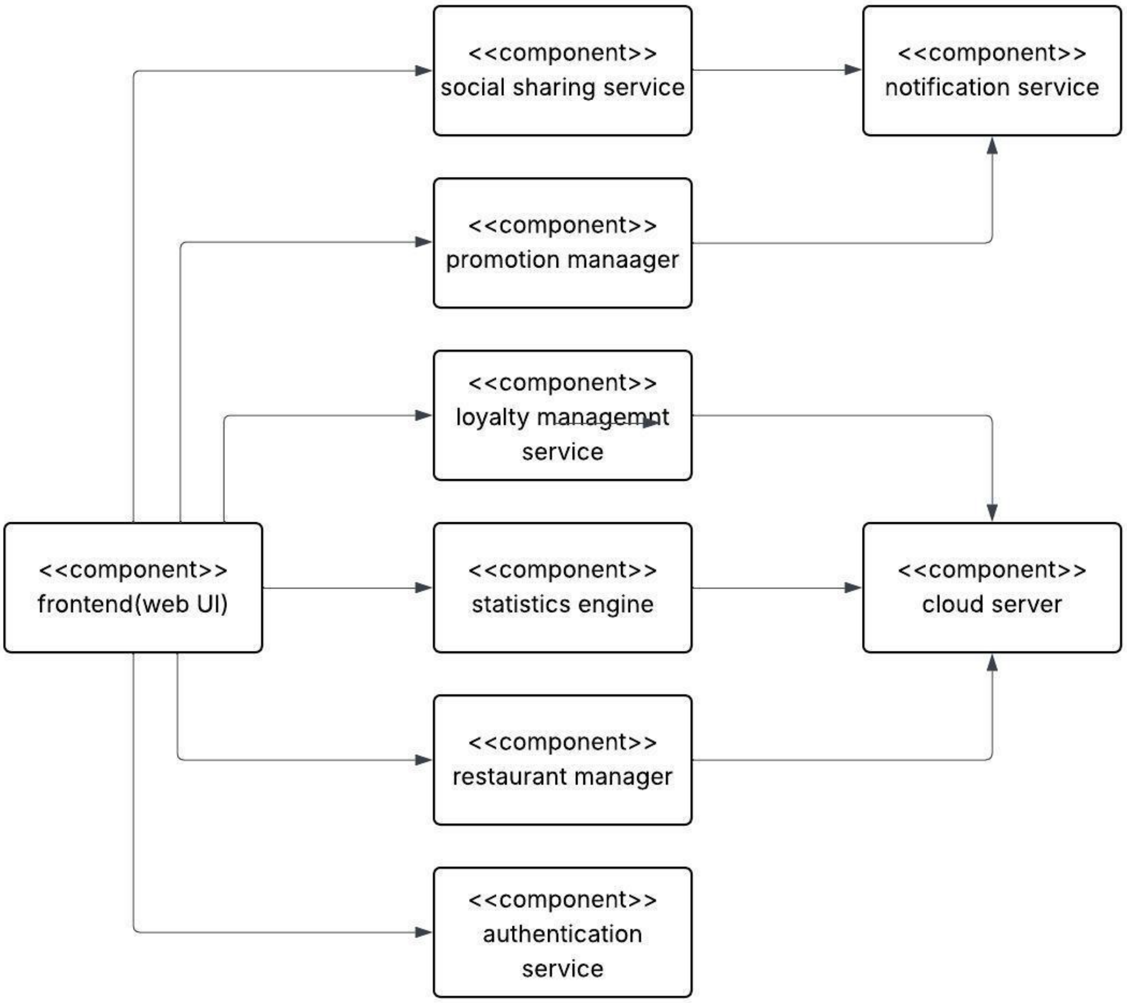
Component Interactions

Frontend(WebUI): It uses the Authentication Service for the user credential verification. it can communicate with the Restaurant Account Manager to manage restaurant data and with the Statistics Engine to provide performance data. It also works with the Promotion Manager and allows users to create and track promotional campaigns. In addition, the service can promote through social media via Social Sharing Service and interact with Loyalty Management Service for customer loyalty information management.

Internal Service Dependencies: To send notifications about promotional campaigns to both restaurants and users, the Promotion Manager uses the Notification Service. It is also used by the Social Sharing

Service to send out notifications on social media sharing. The Database Service is used by the Restaurant Account Manager, Statistics Engine, and Loyalty Management Service to store and retrieve data, so that the important data pertaining to these components is kept safe and can be accessed quickly.

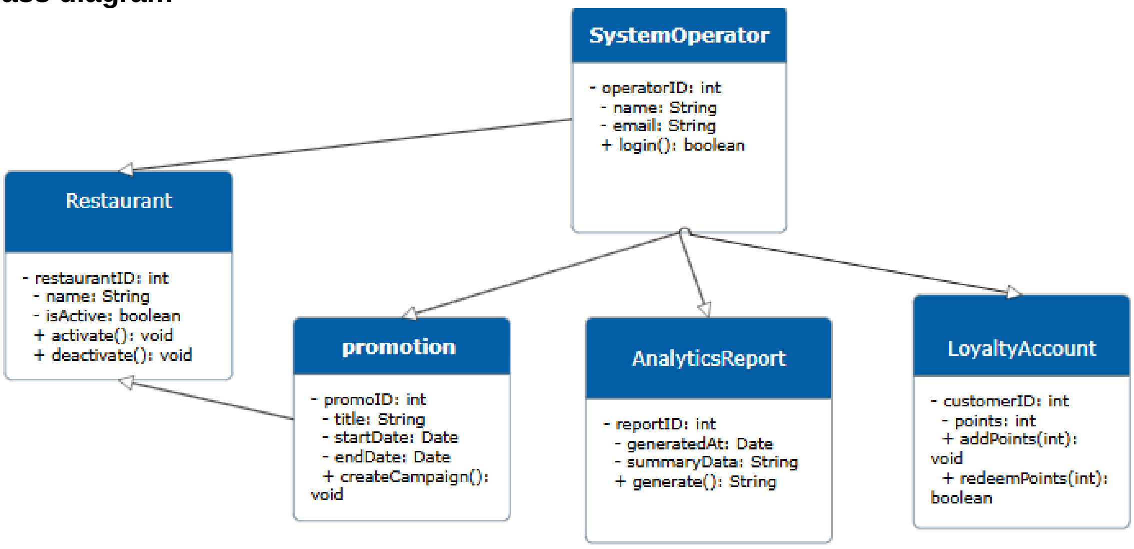
(The below diagram was made using lucid chart)



<<component>> social sharing service
<<component>> notification service
<<component>> promotion manaager
<<component>> loyalty managemnt service
<<component>> frontend(web UI)
<<component>> statistics engine
<<component>> cloud server
<<component>> restaurant manager
<<component>> authentication service

Task 5-Software detailed design

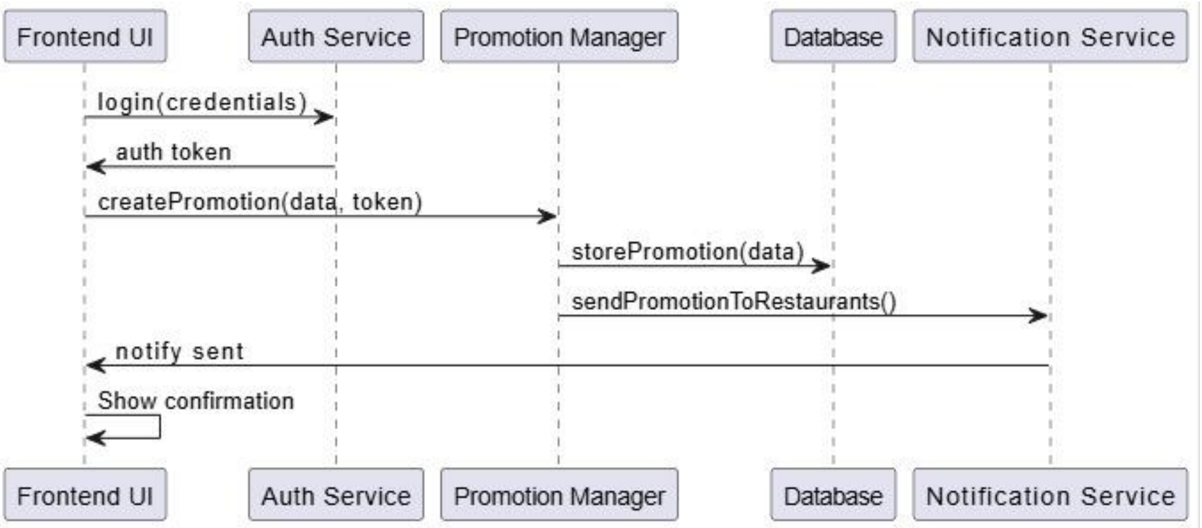
Class diagram



Made using (smartdraw.com)

The key entities of the CloudTables-Operations Subsystem and their relationships are represented by this class diagram. Attributes of the SystemOperator class (operatorID, name, email) and a login() method for authentication are used by the SystemOperator class (represents the operator managing the system). The restaurantID, name and isActive attributes are included in the Restaurant class along with methods to activate() and deactivate() a restaurant. Promotion class is a class representing promotional campaigns as well as a method createCampaign() for creating campaigns targeting specific restaurants. This class, AnalyticsReport, is used to generate performance reports and store reportID, generatedAt, and summaryData, which has a generate() method to produce reports. Customer loyalty points are managed by the LoyaltyAccount class, which has methods addPoints and redeemPoints. SystemOperator is depicted as a class which can handle promotions, restaurants, analytics reports and loyalty accounts, and Promotion class which is connected to Restaurant to target some restaurants for campaigns.

Behaviour Model



(Made using draw.io)

The sequence diagram depicts how a promotion is created in the CloudTables system from the inside. The Frontend UI starts with the user credentials sent to the Auth Service for authentication. The Frontend UI will then make a request to the Auth Service that asks for a token, on successful authentication, the Auth Service will return an auth token to the Frontend UI, that will be used then to authorize further actions. Promotion Manager receives the request from Frontend UI for creating a promotion with the details of the promotion and authentication token. The promotion data is stored in the Database by the Promotion Manager for persisting and the Notification Service is triggered to notify the targeted restaurants on the new promotion. The Notification Service sends the notification and when the notification is sent successfully it confirms this action to the Frontend UI, which in turn displays a confirmation message to the user. This diagram shows how different parts of the system work together to create a system that handles all the basic functionalities from logging in and authenticating the user, to storing the campaign and notifying restaurants.

Reference

- *Cloudflare (2024). What Is Transport Layer Security? Cloudflare. [online] Available at*: https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/.
- Fortinet. (n.d.). *What is OAuth? Definition and How It Works*. [online] Available at: https://www.fortinet.com/resources/cyberglossary/oauth.
- Wolford, B. (2025). *What is GDPR, the EU's new data protection law?* [online] GDPR.eu. Available at: https://gdpr.eu/what-is-gdpr/.
- Microsoft (2024). *What is: Multifactor authentication - microsoft support*. [online] support.microsoft.com. Available at: https://support.microsoft.com/en-gb/topic/what-is-multifactor-authentication-e5e39437-121c-be60-d123-eda06bddf661.
- AWS. (2023). *Amazon API Gateway quotas and important notes.* [online] Available at: https://docs.aws.amazon.com/apigateway/latest/developerguide/limits.html
- Celery Project. (2024). *Celery Documentation.* [online] Available at: https://docs.celeryq.dev/en/stable/
- Amazon Web Services. (2024). *What is Auto Scaling?* [online] Available at: https://aws.amazon.com/autoscaling/ Amazon Web Services. (2024). *Elastic Load Balancing.* [online] Available at: https://aws.amazon.com/elasticloadbalancing/
- Docker. (2024). *What is Docker?* [online] Available at: https://www.docker.com/resources/what-container/
- Lucidchart. (2024). *Lucidchart for UML.* [online] Available at: https://www.lucidchart.com/pages/uml-diagram\
- Draw.io. (2024). *Free Online Diagram Tool.* [online] Available at: https://www.draw.io
- SmartDraw. (2024). *SmartDraw UML Tools.* [online] Available at: https://www.smartdraw.com/uml-diagram/
- *https://miro.com/. (2024). Quick Guide to PlantUML: Diagrams, Syntax & Best Practices* | *Miro. [online] Available at:* https://miro.com/diagramming/what-is-plantuml/.