

PHENIKAA UNIVERSITY
PHENIKAA SCHOOL OF COMPUTING



COURSE: SOFTWARE ARCHITECTURE
DOC: SOFTWARE ARCHITECTURE ANALYSIS AND DESIGN
PRJ: YUMMY - ONLINE FOOD ORDERING & DELIVERY SYSTEM

Instructor	: M.Sc Vu Quang Dung
Class	: CSE703110 (N01)
Group	: 05

Performed by:	
Vu Thi Dieu Linh	23010545
Ta Vuong Bao Ngoc	23010644
Nguyen Minh Sang	23010888
Hoang Nhu Quynh	23010027

Hanoi, January 2026

Declaration

We hereby declare that this graduation project is the result of our own collective research and work. The content of this project has not been previously submitted for any degree or academic qualification at any institution.

All data, information, figures, images, and materials used from other sources have been properly acknowledged and fully cited in the reference section. Any similarities to existing works are purely coincidental or have been clearly referenced.

We take full responsibility for the accuracy, integrity, and originality of the content presented in this graduation project.

Acknowledgements

We would like to express our sincere gratitude to Mr. Vu Quang Dung, M.Sc., lecturer of the Software Architecture course, for his dedicated guidance, valuable feedback, and continuous support throughout the completion of this project. His insightful lectures and professional advice have greatly contributed to our understanding of software architecture principles and significantly improved the quality of our work.

We would also like to thank the Faculty and the University for providing a supportive academic environment and necessary resources that enabled us to successfully complete this project.

Table of Contents

Declaration	2
Acknowledgements	3
Table of Contents.....	Error! Bookmark not defined.
List of Figures	5
List of Tables	6
I. Project introduction.....	7
1. Development team information	7
2. Project information	7
II. Project overview	7
LAB 1 REPORT – REQUIREMENTS ELICITATION AND MODELING.	9
I. Summary	10
II. Technology and tool installation	10
III. Lab Specific Section: Requirements Elicitation & Modeling	10
1. Software Requirements Specification (SRS)	10
1.1. Identification of Actors	11
1.2. Functional Requirements (FRs)	11
1.4. Architecturally Significant Requirements (ASRs)	14
IV. Modeling Artifact: UML Use Case Diagram.....	14
V. Conclusion	17

List of Figures

Figure 1.1. UML Use Case Diagram for the Yummy system	16
---	----

List of Tables

Table 0.1: List of project team members	7
Table 1.1. List of supporting tools used in Lab 1.	10
Table 1.2. List of actors for the Yummy system.....	11
Table 1.3. List of functional requirements for the Yummy system.	13
Table 1.4. List of non-functional requirements for the Yummy system.....	14
Table 1.5. Architecturally Significant Requirements (ASRs).	14

I. Project introduction

1. Development team information

Full name	Student ID	Role	Contact
Vu Thi Dieu Linh	23010545	Member	23010545@st.phenikaa-uni.edu.vn
Ta Vuong Bao Ngoc	23010644	Member	<u>23010644@st.phenikaa-uni.edu.vn</u>
Nguyen Minh Sang	23010888	Member	23010888@st.phenikaa-uni.edu.vn
Hoang Nhu Quynh	23010027	Member	23010027@st.phenikaa-uni.edu.vn

Table 0.1: List of project team members

2. Project information

Topic: Yummy – Online food ordering and delivery system

Software Type: Mobile Application

This project is implemented within the framework of the Software Architecture course. The project focuses on developing a mobile application for online food ordering and delivery, supporting the entire process from searching for dishes, ordering, and tracking orders to delivery coordination. The system serves multiple user groups, including customers, restaurants, and delivery drivers. The project aims to build a lean, intelligent, and scalable food delivery platform, utilizing modern technology to enhance user experience, optimize operational efficiency, and support the business activities of restaurant partners.

II. Project overview

In recent years, online food ordering and delivery systems have become an indispensable part of modern urban life. The development of mobile technology, e-commerce, and logistics has strongly driven the trend of food consumption through digital platforms. Users are increasingly choosing food delivery apps to save time, increase convenience, and access more culinary options. Besides, the online food delivery model not only brings benefits to consumers but also creates new business opportunities for restaurants, eateries, and delivery drivers. This ecosystem forms a tight link between three main groups: customers – restaurants – drivers, in which the software system plays a central role in coordinating and optimizing the entire process.

Despite the presence of many large platforms in the market, user experience still faces certain limitations. Most current applications rely heavily on keyword searches or traditional category browsing, which causes users to spend a lot of time when they do not know the exact name of a dish. In reality, the "see it, want it" demand is becoming more popular, especially when users encounter dishes through images on social media or in real life. Additionally, feature overload on some major platforms makes the experience complex and unsuitable for users who only need to perform basic operations such as searching, ordering, and tracking orders quickly.

For restaurants, especially small and medium-sized businesses, joining online delivery platforms offers the opportunity to reach a broader customer base. However, they still face difficulties in managing orders, updating menu information, and optimizing their visibility on the system. Therefore, a simple, easy-to-use platform that still ensures business efficiency is a practical need. On the driver's side, the most critical factors are stability in order distribution, transparent income information, and route

optimization. An intelligent coordination system will help shorten delivery times, increase work productivity, and improve income for drivers.

Arising from the aforementioned needs and challenges, the Yummy project - an online food ordering and delivery application - is proposed to build a lean, intelligent system focused on core user experiences. Yummy aims to optimize key steps in the food ordering journey, including: searching for dishes, selecting items, ordering, and tracking order status. A standout feature of the system is the integration of AI technology, specifically Computer Vision, to support food searching via images. This feature allows users to find dishes more intuitively and naturally, contributing to shorter search times and an enhanced user experience.

The Yummy online food ordering and delivery system is not just a consumer application, it is also a technological solution contributing to the digital transformation of the food service industry. By operating based on data, integrating real-time technology, and utilizing a flexible system architecture, Yummy aims to build a platform capable of scaling to meet the needs of users both now and in the future.

YUMMY – ONLINE FOOD ORDERING AND DELIVERY SYSTEM
LAB 1 REPORT – REQUIREMENTS ELICITATION AND MODELING.

I. Summary

Lab 1 focuses on the initial stage of the software architecture design process for the "Yummy" online food ordering and delivery application. The main content involves business requirement analysis, identifying critical requirements directly affecting the system architecture, and modeling external system behavior through a UML Use Case Diagram.

In this lab, the team collected and clarified Architecturally Significant Requirements (ASRs), including performance, scalability, availability, real-time capabilities, and the ability to integrate independent components within the system. These requirements serve as the foundation for selecting an appropriate architecture, especially given that the system serves multiple user groups such as customers, restaurants, and delivery drivers.

Next, the lab identified the primary (Actors) of the Yummy system, including customers, restaurant partners, delivery drivers, and external payment systems. Based on these actors, core Use Cases were analyzed and described to reflect the system's main functionalities, such as food searching, ordering, order management, delivery tracking, and payment processing.

Finally, the relationships between actors and system functions were modeled using a UML Use Case Diagram, clearly illustrating the system scope, external interactions, and how the system meets business needs. The results of Lab 1 provide an essential baseline for proceeding with subsequent software architecture design steps in following labs.

II. Technology and tool installation

Tool	Purpose	Installation/Setup Instructions
Google Docs Microsoft Words	Drafting and storing requirement documents and ASRs.	Standard word processing software. Google Docs is used online; Microsoft Word requires local installation.
draw.io	Creating professional UML diagrams (Use Case Diagram, Component Diagram, etc.).	Accessed online via web browser; no local installation required.

Table 1.1. List of supporting tools used in Lab 1.

III. Lab Specific Section: Requirements Elicitation & Modeling

This section presents the deliverables of Lab 1, developed according to the report requirements of the *Software Architecture* course. The content focuses on collecting, analyzing, and modeling requirements for the Yummy system, serving as the basis for the architectural design process in subsequent stages.

1. Software Requirements Specification (SRS)

The Software Requirements Specification (SRS) document describes the set of requirements for the Yummy system, including functional and non-functional requirements, reflecting the business needs of stakeholders like customers, restaurants, and drivers. These requirements were gathered through business analysis, use case scenarios, and the overall architectural direction of the system. They serve as the foundation for identifying key architectural components, communication mechanisms between services, and future scalability.

1.1. Identification of Actors

STT	Actor	Description
1	Customer	Customer is the end-user of the system, using the application to search for restaurants, select dishes, order, pay, and track delivery status. This is the Actor that initiates most of the core business flows of the system. Customers expect the ordering process to be fast, convenient, and accurate. Additionally, they need the ability to track orders in real-time to understand delivery status, as well as view order history to easily reorder favorite dishes.
2	Restaurant	Restaurant is the partner providing food on the platform. This Actor is responsible for managing store information, menus, and processing orders placed by Customers through the system. Restaurants need a tool to help manage menus flexibly, update dish information, and process orders promptly. Furthermore, Merchants also have a need to track revenue, sales performance, and implement promotion programs to attract customers.
3	Driver	Driver is the person directly performing delivery activities, receiving orders from the system and transporting food from the Merchant to the Customer. Drivers expect to receive suitable orders, with clear navigation instructions to optimize delivery time. At the same time, Drivers need to track income, delivery history, and bonus amounts transparently.
4	Admin	Admin is the Actor responsible for operating and monitoring the entire system. Admin does not participate directly in the ordering process but plays a role in ensuring the system operates stably and complies with regulations. Admin needs the ability to manage users and partners participating in the system, including Customers, Restaurants, and Drivers. Additionally, Admin must monitor overall system metrics such as order volume, revenue, and activity levels to serve operational and reporting tasks.

Table 1.2. List of actors for the Yummy system.

1.2. Functional Requirements (FRs)

Functional Requirements (FRs) describe the specific behaviors and functions the Yummy system must provide to meet user needs. These requirements include core functions such as food searching, ordering, order management, delivery status tracking, and supporting interaction between actors in the system.

The FRs are identified for each actor group and serve as the basis for building Use Case Diagrams, helping to model the external behavior of the system and clearly defining the functional scope that the system needs to perform.

ID	Name Function	Description	Priority
Customer Functions			
FR-C01	Search & Discovery	The system shall allow Customers to search for restaurants or food items by keyword and apply filters such as distance, category, rating, and price range	High

FR-C02	Restaurant Browsing	The system shall allow Customers to view restaurant details including menu, opening hours, service fee, distance, and available promotions.	High
FR-C03	Cart Management	The system shall allow Customers to add food items to a cart and manage cart contents according to single-restaurant ordering rules.	High
FR-C04	Voucher & Promotion Application	The system shall allow Customers to apply platform vouchers, merchant vouchers, and free-shipping vouchers during checkout.	Medium ⁶⁵
FR-C05	Checkout & Payment	The system shall support multiple payment methods including Cash on Delivery (COD), E-wallets, and Credit/Debit cards, and verify payment before order confirmation.	High
FR-C06	Real-time Order Tracking	The system shall allow Customers to track order status in real time, including order preparation, driver pickup, delivery progress, and completion.	High
FR-C07	Rating & Review	The system shall allow Customers to submit ratings and reviews for food quality and delivery service after order completion.	Low
FR-C08	Image-based Food Search	The system shall allow Customers to upload or capture food images to search for similar dishes or restaurants with a confidence score.	Medium
<i>Merchant Functions</i>			
FR-M01	Menu Management	The system shall allow Merchants to create, update, or remove menu items, toppings, images, and prices.	High
FR-M02	Availability Management	The system shall allow Merchants to toggle food items or restaurant availability (Out of Stock / Busy Mode) in real time.	High
FR-M03	Order Processing	The system shall allow Merchants to receive, accept, and process orders through defined statuses (Accepted, Cooking, Ready for Pickup).	High
FR-M04	Revenue Reporting	The system shall allow Merchants to view revenue reports by day, week, or month, including completed and cancelled orders.	Medium
<i>Driver Functions</i>			
FR-D01	Delivery Request Handling	The system shall assign delivery requests to nearby Drivers and allow Drivers to accept or decline requests within a defined time limit.	High
FR-D02	Navigation & Routing	The system shall provide Drivers with navigation guidance from their current location to the restaurant and then to the customer's address.	High

FR-D03	Delivery Status Update	The system shall allow Drivers to update delivery status, including arrival, pickup confirmation, delivery in progress, and successful delivery with proof.	High
FR-D04	Driver Wallet Management	The system shall allow Drivers to view income history and request withdrawals to a linked bank account.	Medium
Admin & Platform Functions			
FR-A01	User & Partner Management	The system shall allow Administrators to manage Customer, Merchant, and Driver accounts, including activation, suspension, and verification.	High
FR-A02	Commission Configuration	The system shall allow Administrators to configure commission rates for individual Merchants or merchant categories.	High
FR-A03	Global Promotion Management	The system shall allow Administrators to create and manage platform-wide promotional campaigns.	Medium
FR-A04	System Dashboard & Reporting	The system shall allow Administrators to view system-wide statistics, including order volume, revenue, active users, and export reports in CSV/PDF format.	High

Table 1.3. List of functional requirements for the Yummy system.

1.3. Non-Functional Requirements (NFRs)

Non-Functional Requirements (NFRs) describe quality attributes and constraints that the Yummy system must meet during operation. These requirements do not focus on specific functions but reflect how the system operates to ensure performance, reliability, scalability, security, and user experience. The NFRs are determined based on the specific nature of an online ordering and delivery system serving multiple simultaneous actor groups. These requirements play a crucial role in directing the software architecture design and are considered the basis for selecting suitable architectural solutions for the Yummy system.

ID	Name Function	Description	Priority
NFR-01	Performance (Concurrency)	The system must be able to handle a minimum of 10,000 concurrent active users (CCU) during peak hours (11:00–13:00 and 18:00–20:00) without crashing or timing out.	Critical
NFR-02	Performance (Latency)	The system must return the list of restaurants and menus for 90% of requests within 1.0 second under normal load conditions.	High
NFR-03	Data Consistency	Order status must be accurately synchronized across customer, merchant, and driver applications.	Critical
NFR-04	Security (Payment & Data Protection)	All sensitive data (payment details, phone numbers, addresses) must be encrypted in transit (TLS 1.3) and at rest (AES-256). Payment processing must comply with PCI-DSS standards via a secure payment gateway.	Critical
NFR-05	Reliability (Availability)	The system must maintain a minimum 99.9% uptime during operational hours.	High

NFR-06	Scalability	The backend architecture must support horizontal auto-scaling to handle sudden traffic spikes, such as flash sales or promotional campaigns, without service degradation.	High
NFR-07	Accuracy	The system's calculation of distance and delivery fees must have an error margin of no more than 50 meters compared to actual map data to ensure fair delivery fee calculation.	Medium
NFR-08	Maintainability	The system architecture must allow integration of new third-party services (e.g., alternative payment gateways) without requiring refactoring of core business logic.	High
NFR-09	Availability (Maintenance Handling)	Scheduled maintenance must be announced at least 24 hours in advance and executed during off-peak hours (03:00–05:00) to minimize user impact.	Medium
NFR-10	Usability (Mobile Experience)	The user interface must be optimized for mobile devices (iOS & Android).	High

Table 1.4. List of non-functional requirements for the Yummy system.

1.4. Architecturally Significant Requirements (ASRs)

ASR ID	Quality Attribute	Requirement Statement	Architectural Rationale
ASR-1	Scalability	The architecture must support rapid horizontal scaling to accommodate sudden load spikes (1k to 10k users) during premieres.	Dictates a Stateless Architecture for backend services to facilitate load balancing. It also necessitates the use of a Content Delivery Network (CDN) for media offloading and efficient caching strategies at the Data layer.
ASR-2	Security	Access to protected resources (Purchases, Profile) must be strictly controlled via token validation; Payment logic must be isolated.	Requires a centralized Security/Authorization Component (e.g., AuthGuard) within the Business Logic layer to intercept and validate requests. Payment integration must be encapsulated to prevent leakage of sensitive data.
ASR-3	Modifiability	Replacing third-party integrations (Stripe, Firebase, Video Provider) must not impact the core Domain Logic or UI.	Enforces Separation of Concerns via a Layered Architecture. Service interfaces (e.g., IPaymentService) must be defined to decouple implementation details from the business logic, enabling independent evolution of components.

Table 1.5. Architecturally Significant Requirements (ASRs).

IV. Modeling Artifact: UML Use Case Diagram

The UML Use Case Diagram of the Yummy App describes the overall functional scope of the system through actors and their corresponding use cases. The System Boundary represents the full scope of the application, including functions related to ordering, restaurant management, delivery, and system administration.

The system identifies primary actors including User, Customer, Restaurant, Driver, and Admin. Among these, User is a generalized actor, inherited by specialized actors to represent different permissions and roles when interacting with the system. Customers perform functions such as food searching, cart management, and order tracking. Restaurants manage store information and menus and process orders. Drivers handle receiving and delivering orders and updating delivery status. Admins manage users, restaurants, system categories, and monitor summary statistics.

The diagram utilizes <<include>> and <<extend>> relationships to clarify business logic. Mandatory functions in the ordering process are modeled with <<include>> while optional functions, such as applying vouchers, are represented by <<extend>>. Overall, the Use Case Diagram provides a full and clear reflection of the Yummy App's core functions, aligning with the project requirements.



Figure 1.1. UML Use Case Diagram for the Yummy system

V. Conclusion

Lab 1 has completed the critical initial steps in the software architecture design process for the Yummy online food ordering and delivery system. Through requirement gathering and analysis, the team has identified core functional and non-functional requirements, several of which significantly impact the system architecture.

Additionally, the primary actors and system use cases have been clarified and modeled using a UML Use Case Diagram, representing the full functional scope and external interactions. The results of Lab 1 create a solid foundation for subsequent labs, particularly in proposing the overall architecture and designing the components of the Yummy system.