

**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

# **GRADUATION THESIS**

**Building Information board using ePaper Display**

**Vu Le Nhat Minh**

minh.vln194333@sis.hust.edu.vn

**Major: School of Information, Communication and Technology**

**Supervisor:** ThS. Nguyen Duc Tien \_\_\_\_\_

Signature

**Department:** Computer Engineering

**School:** Information and Communications Technology

**HANOI, 12/2023**

# **ACKNOWLEDGMENTS**

The students' sincere thanks should send to their lover, family, friends, teachers, and themselves for their hard work and determination to achieve the best results. The acknowledgments should be written in a concrete manner and limited in the range of 100-150 words.

# ABSTRACT

Sinh viên viết tóm tắt ĐATN của mình trong mục này, với 200 đến 350 từ. Theo trình tự, các nội dung tóm tắt cần có: (i) Giới thiệu vấn đề (tại sao có vấn đề đó, hiện tại được giải quyết chưa, có những hướng tiếp cận nào, các hướng này giải quyết như thế nào, hạn chế là gì), (ii) Hướng tiếp cận sinh viên lựa chọn là gì, vì sao chọn hướng đó, (iii) Tổng quan giải pháp của sinh viên theo hướng tiếp cận đã chọn, và (iv) Đóng góp chính của ĐATN là gì, kết quả đạt được sau cùng là gì. Sinh viên cần viết thành đoạn văn, không được viết ý hoặc gạch đầu dòng.

Student

*(Signature and full name)*

## TABLE OF CONTENTS

<b>CHAPTER 1. INTRODUCTION.....</b>	<b>1</b>
1.1 Motivation .....	1
1.2 Objectives and scope of the graduation thesis .....	2
1.3 Tentative solution .....	2
1.4 Thesis organization .....	2
<b>CHAPTER 2. REQUIREMENT SURVEY AND ANALYSIS.....</b>	<b>4</b>
2.1 Status survey .....	4
2.2 Functional Overview .....	4
2.2.1 General use case diagram.....	4
2.2.2 Detailed use case diagram.....	4
2.2.3 Business process .....	6
2.3 Functional description.....	6
2.3.1 Description of use case A .....	6
2.3.2 Description of use case B.....	6
2.4 Non-functional requirement .....	6
<b>CHAPTER 3. METHODOLOGY.....</b>	<b>7</b>
3.1 Management UI (Front-end side) .....	7
3.1.1 NextJS .....	7
3.1.2 TailwindCSS .....	7
3.2 Server (Back-end side) .....	7
3.2.1 API Server .....	8
3.2.2 MQTT Broker .....	8
3.3 EPD device .....	9

<b>CHAPTER 4. DESIGN, IMPLEMENTATION, AND EVALUATION.....</b>	<b>10</b>
4.1 Architecture design.....	10
4.1.1 Software architecture selection .....	10
4.1.2 Overall design.....	11
4.1.3 Detailed package design .....	13
4.2 Detailed design.....	13
4.2.1 User interface design.....	13
4.2.2 Layer design .....	14
4.2.3 Database design .....	23
4.3 Application Building.....	24
4.3.1 Libraries and Tools.....	24
4.3.2 Achievement.....	25
4.3.3 Illustration of main functions .....	26
4.4 Testing.....	26
4.5 Deployment .....	27
<b>CHAPTER 5. SOLUTION AND CONTRIBUTION .....</b>	<b>28</b>
<b>CHAPTER 6. CONCLUSION AND FUTURE WORK .....</b>	<b>29</b>
6.1 Conclusion.....	29
6.2 Future work.....	29
<b>REFERENCE .....</b>	<b>32</b>
<b>APPENDIX .....</b>	<b>33</b>
<b>A. THESIS WRITING GUIDELINE.....</b>	<b>33</b>
A.1 General Regulations.....	33
A.2 Majoring .....	34
A.3 Bulleting and Numbering .....	34
A.4 Table insertion.....	35

A.5 Figure Insertion .....	35
A.6 Reference .....	36
A.7 Equations .....	36
A.8 Qui cách đóng quyển.....	36
<b>B. USE CASE DESCRIPTIONS .....</b>	<b>38</b>
B.1 Đặc tả use case “Thông kê tình hình mượn sách” .....	38
B.2 Đặc tả use case “Đăng ký làm thẻ mượn” .....	38



## LIST OF FIGURES

Figure 2.1	General use case diagram of the data management system . .	5
Figure 2.2	Detailed use case of User's Account Management function .	5
Figure 2.3	Detailed use case of Data Management function . . . . .	6
Figure 4.1	Overall architecture of the system . . . . .	10
Figure 4.2	MVCS architecture . . . . .	11
Figure 4.3	MQTT microservice . . . . .	11
Figure 4.4	General use case diagram of the data management system . .	12
Figure 4.5	General use case diagram of the data management system . .	13
Figure 4.6	General use case diagram of the data management system . .	13
Figure 4.7	General use case diagram of the data management system . .	14
Figure 4.8	Mockup display of authentication page . . . . .	14
Figure 4.9	Mockup display of dashboard page . . . . .	14
Figure 4.10	Mockup display of modal component . . . . .	14
Figure 4.11	Mockup display of creation page . . . . .	14
Figure 4.12	Detailed use case of Data Management function . . . . .	15
Figure 4.13	Caption . . . . .	15
Figure 4.14	Detailed use case of Data Management function . . . . .	16
Figure 4.15	Caption . . . . .	17
Figure 4.16	Detailed use case of Data Management function . . . . .	18
Figure 4.17	Detailed use case of Data Management function . . . . .	19
Figure 4.18	Detailed use case of Data Management function . . . . .	20
Figure 4.19	Detailed use case of Data Management function . . . . .	21
Figure 4.20	Detailed use case of Data Management function . . . . .	22
Figure 4.21	Detailed use case of Data Management function . . . . .	22
Figure A.1	Internet vạn vật . . . . .	35
Figure A.2	Quy cách đóng quyển đồ án . . . . .	37
Figure A.3	Quy cách đóng quyển đồ án . . . . .	37



## LIST OF TABLES

Bảng 4.1	Your Table Caption . . . . .	24
Bảng 4.2	Danh sách thư viện và công cụ sử dụng . . . . .	25
Bảng 4.3	Danh sách thư viện và công cụ sử dụng . . . . .	26
Bảng A.1	Table to test captions and labels. . . . .	35

## LIST OF ABBREVIATIONS

<b>Abriviation</b>	<b>Full Expression</b>
API	Giao diện lập trình ứng dụng (Application Programming Interface)
EUD	Phát triển ứng dụng người dùng cuối(End-User Development)
GWT	Công cụ lập trình Javascript bằng Java của Google (Google Web Toolkit)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
IaaS	Dịch vụ hạ tầng

## CHAPTER 1. INTRODUCTION

Lưu ý: **Trước khi viết ĐATN, sinh viên cần đọc kỹ hướng dẫn và quy định chi tiết** về cách viết ĐATN trong Phụ lục A. Sinh viên tuân theo mẫu tài liệu này để viết báo cáo đồ án tốt nghiệp, vì tài liệu này đã được căn chỉnh, chỉnh sửa theo đúng chuẩn báo cáo kỹ thuật đồ án tốt nghiệp (ISO 7144:1986). Sinh viên viết trực tiếp vào file này, chỉ chỉnh sửa nội dung, và không viết trên file mới.

**Khi đóng quyển ĐATN**, sinh viên cần lưu ý tuân thủ hướng dẫn ở phụ lục A.9

**SV cần đặc biệt lưu ý cách hành văn.** Mỗi đoạn văn không được quá dài và cần có ý tứ rõ ràng, bao gồm duy nhất một ý chính và các ý phân tích bổ trợ để làm rõ hơn ý chính. Các câu văn trong đoạn phải đầy đủ chủ ngữ vị ngữ, cùng hướng đến chủ đề chung. Câu sau phải liên kết với câu trước, đoạn sau liên kết với đoạn trước. Trong văn phong khoa học, sinh viên không được dùng từ trong văn nói, không dùng các từ phóng đại, thái quá, các từ thiếu khách quan, thiên về cảm xúc, về quan điểm cá nhân như “tuyệt vời”, “cực hay”, “cực kỳ hữu ích”, v.v. Các câu văn cần được tối ưu hóa, đảm bảo rất khó để thể thêm hoặc bớt đi được dù chỉ một từ. Cách diễn đạt cần ngắn gọn, súc tích, không dài dòng.

Mẫu ĐATN này được thiết kế phù hợp nhất với đa số các đề tài xây dựng phần mềm ứng dụng. Với các dạng đề tài khác (giải pháp, nghiên cứu, phần mềm đặc thù, v.v.), sinh viên dựa trên cấu trúc và hướng dẫn của báo cáo này để đề xuất và trao đổi với giáo viên hướng dẫn để thiết kế khung báo cáo đồ án cho phù hợp. Sinh viên lưu ý **trong mọi trường hợp, SV luôn phải sử dụng định dạng báo cáo này, và phải đọc kỹ toàn bộ các hướng dẫn từ đầu tới cuối.** Các hướng dẫn không chỉ áp dụng riêng cho đề tài ứng dụng, mà còn phù hợp với các dạng đề tài khác. Ngoài ra, trong mẫu ĐATN này đã được tích hợp một số hướng dẫn dành riêng cho đề tài nghiên cứu.

Chương 1 có độ dài từ 3 đến 6 trang với các nội dung sau đây

### 1.1 Motivation

Khi đặt vấn đề, sinh viên cần làm nổi bật mức độ cấp thiết, tầm quan trọng và/hoặc quy mô của bài toán của mình.

Gợi ý cách trình bày cho sinh viên: Xuất phát từ tình hình thực tế gì, dẫn đến vấn đề hoặc bài toán gì. Vấn đề hoặc bài toán đó, nếu được giải quyết, đem lại lợi ích gì, cho những ai, còn có thể được áp dụng vào các lĩnh vực khác nữa không. Sinh viên cần lưu ý phần này chỉ trình bày vấn đề, tuyệt đối không trình bày giải pháp.

## 1.2 Objectives and scope of the graduation thesis

Sinh viên trước tiên cần trình bày tổng quan các kết quả của các nghiên cứu hiện nay cho bài toán giới thiệu ở phần 1.1 (đối với đề tài nghiên cứu), hoặc về các sản phẩm hiện tại/về nhu cầu của người dùng (đối với đề tài ứng dụng). Tiếp đến, sinh viên tiến hành so sánh và đánh giá tổng quan các sản phẩm/nghiên cứu này.

Dựa trên các phân tích và đánh giá ở trên, sinh viên khái quát lại các hạn chế hiện tại đang gặp phải. Trên cơ sở đó, sinh viên sẽ hướng tới giải quyết vấn đề cụ thể gì, khắc phục hạn chế gì, phát triển phần mềm **có các chức năng chính gì**, tạo nên đột phá gì, v.v.

Trong phần này, sinh viên lưu ý chỉ trình bày tổng quan, không đi vào chi tiết của vấn đề hoặc giải pháp. Nội dung chi tiết sẽ được trình bày trong các chương tiếp theo, đặc biệt là trong Chương 5.

## 1.3 Tentative solution

Từ việc xác định rõ nhiệm vụ cần giải quyết ở phần 1.2, sinh viên đề xuất định hướng giải pháp của mình theo trình tự sau: (i) Sinh viên trước tiên trình bày sẽ giải quyết vấn đề theo định hướng, phương pháp, thuật toán, kỹ thuật, hay công nghệ nào; Tiếp theo, (ii) sinh viên mô tả ngắn gọn giải pháp của mình là gì (khi đi theo định hướng/phương pháp nêu trên); và sau cùng, (iii) sinh viên trình bày đóng góp chính của đề án là gì, kết quả đạt được là gì.

Sinh viên lưu ý không giải thích hoặc phân tích chi tiết công nghệ/thuật toán trong phần này. Sinh viên chỉ cần nêu tên định hướng công nghệ/thuật toán, mô tả ngắn gọn trong một đến hai câu và giải thích nhanh lý do lựa chọn.

## 1.4 Thesis organization

Phần còn lại của báo cáo đề án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về v.v.

Trong Chương 3, em/tôi giới thiệu về v.v.

**Chú ý:** Sinh viên cần viết mô tả thành đoạn văn đầy đủ về nội dung chương. Tuyệt đối không viết ý hay gạch đầu dòng. Chương 1 không cần mô tả trong phần này.

Ví dụ tham khảo mô tả chương trong phần bố cục đề án tốt nghiệp: Chương \*\*\* trình bày đóng góp chính của đề án, đó là một nền tảng ABC cho phép khai phá và tích hợp nhiều nguồn dữ liệu, trong đó mỗi nguồn dữ liệu lại có định dạng đặc thù riêng. Nền tảng ABC được phát triển dựa trên khái niệm DEF, là các module ngữ nghĩa trợ giúp người dùng tìm kiếm, tích hợp và hiển thị trực quan dữ liệu theo mô

hình cộng tác và mô hình phân tán.

**Chú ý:** Trong phần nội dung chính, mỗi chương của đề án nên có phần Tổng quan và Kết chương. Hai phần này đều có định dạng văn bản “Normal”, sinh viên không cần tạo định dạng riêng, ví dụ như không in đậm/in nghiêng, không đóng khung, v.v.

Trong phần Tổng quan của chương N, sinh viên nên có sự liên kết với chương N-1 rồi trình bày sơ qua lý do có mặt của chương N và sự cần thiết của chương này trong đề án. Sau đó giới thiệu những vấn đề sẽ trình bày trong chương này là gì, trong các đề mục lớn nào.

Ví dụ về phần Tổng quan: Chương 3 đã thảo luận về nguồn gốc ra đời, cơ sở lý thuyết và các nhiệm vụ chính của bài toán tích hợp dữ liệu. Chương 4 này sẽ trình bày chi tiết các công cụ tích hợp dữ liệu theo hướng tiếp cận “mashup”. Với mục đích và phạm vi của đề tài, sáu nhóm công cụ tích hợp dữ liệu chính được trình bày bao gồm: (i) nhóm công cụ ABC trong phần 4.1, (ii) nhóm công cụ DEF trong phần 4.2, nhóm công cụ GHK trong phần 4.3, v.v.

Trong phần Kết chương, sinh viên đưa ra một số kết luận quan trọng của chương. Những vấn đề mở ra trong Tổng quan cần được tóm tắt lại nội dung và cách giải quyết/thực hiện như thế nào. Sinh viên lưu ý không viết Kết chương giống hệt Tổng quan. Sau khi đọc phần Kết chương, người đọc sẽ nắm được sơ bộ nội dung và giải pháp cho các vấn đề đã trình bày trong chương. Trong Kết chương, Sinh viên nên có thêm câu liên kết tới chương tiếp theo.

Ví dụ về phần Kết chương: Chương này đã phân tích chi tiết sáu nhóm công cụ tích hợp dữ liệu. Nhóm công cụ ABC và DEF thích hợp với những bài toán tích hợp dữ liệu phạm vi nhỏ. Trong khi đó, nhóm công cụ GHK lại chứng tỏ thế mạnh của mình với những bài toán cần độ chính xác cao, v.v. Từ kết quả nghiên cứu và phân tích về sáu nhóm công cụ tích hợp dữ liệu này, tôi đã thực hiện phát triển phần mềm tự động bóc tách và tích hợp dữ liệu sử dụng nhóm công cụ GHK. Phần này được trình bày trong chương tiếp theo – Chương 5.

## CHAPTER 2. REQUIREMENT SURVEY AND ANALYSIS

Based on the real-world problems discussed in Chapter 1, Chapter 2 aims to delve deeper into the existing products and solutions that can help address these issues. Specifically, it will also analyze some detailed use cases of the Data Management System, as outlined in Section 2.2.

### 2.1 Status survey

Thông thường, khảo sát chi tiết về hiện trạng và yêu cầu của phần mềm sẽ được lấy từ ba nguồn chính, đó là (i) người dùng/khách hàng, (ii) các hệ thống đã có, (iii) và các ứng dụng tương tự. Sinh viên cần tiến hành phân tích, so sánh, đánh giá chi tiết ưu nhược điểm của các sản phẩm/nghiên cứu hiện có. Sinh viên có thể lập bảng so sánh nếu cần thiết. Kết hợp với khảo sát người dùng/khách hàng (nếu có), sinh viên nêu và mô tả sơ lược các tính năng phần mềm quan trọng cần phát triển.

### 2.2 Functional Overview

#### 2.2.1 General use case diagram

The general use case diagram of Data and EPD Devices Management System is illustrated in Figure 1. As per the diagram, the system involves three main agents, namely The Manager, The Administrator, and the EPD devices. The Manager has the ability to manage EPD devices, data and their own accounts. The Administrator inherits the functions of the Manager and can also manage and test the devices more advanced. On the other hand, the EPD devices act as end-users and receive and display data on the screen. They also interact with the system via MQTT protocol.

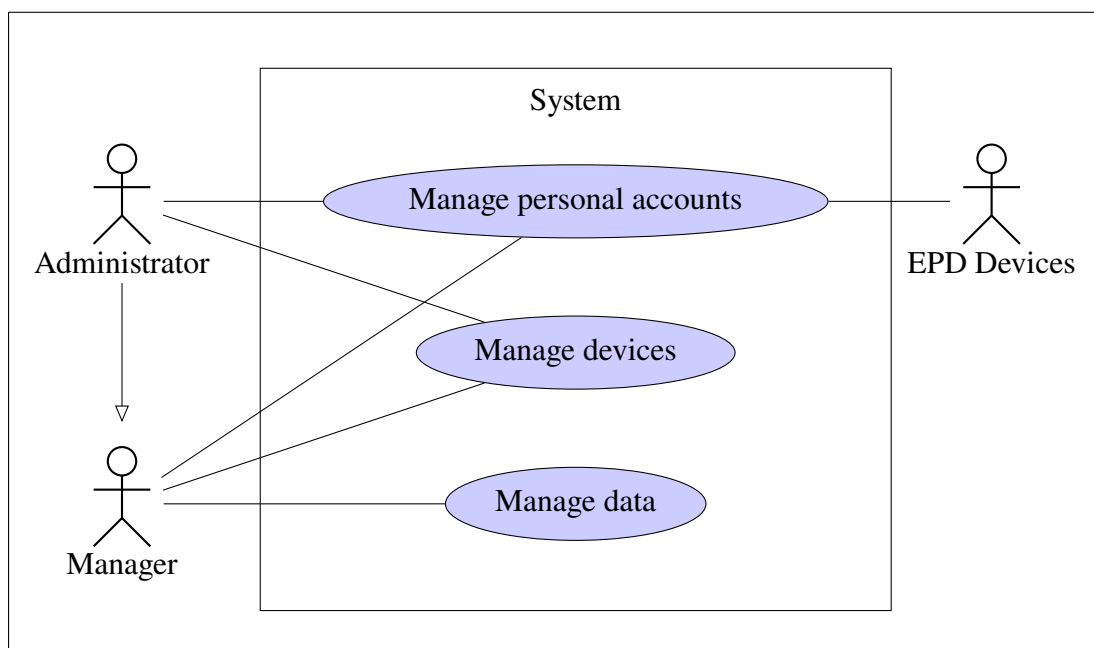
#### 2.2.2 Detailed use case diagram

##### a, Detailed use case of User's Account Management function

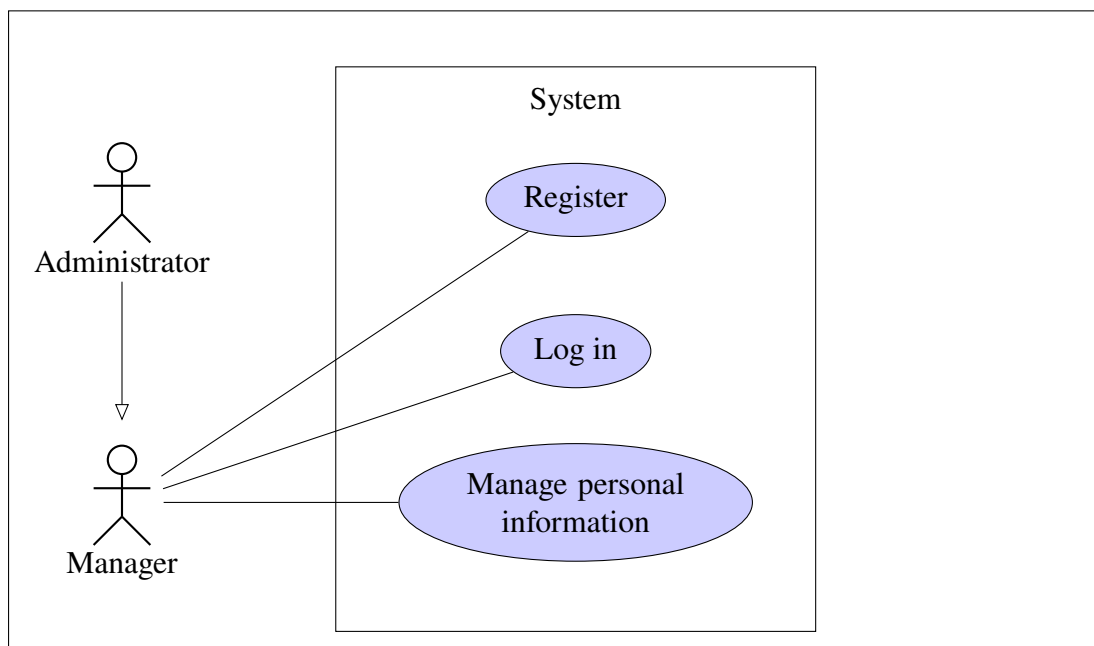
Figure 2 below describes the detailed use case diagram of the User's Account Management function, including the Manager and Administrator. Users can create a new account, log in, and modify personal information.

##### b, Detailed use case of Data Management function

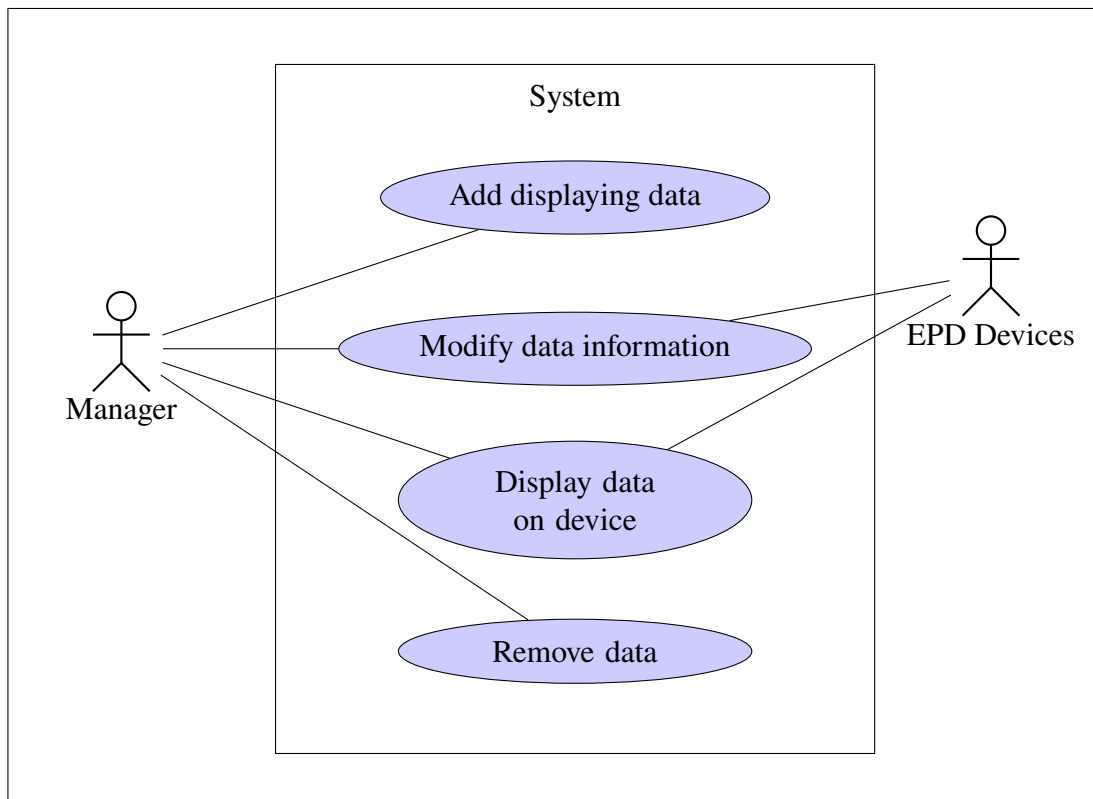
A detailed use case diagram of the Data Management function of the Manager is displayed in Figure 3. This function enables the Manager to view a list of data, add, modify, or remove data information, and choose whether to display the data on the device or not.



**Figure 2.1:** General use case diagram of the data management system



**Figure 2.2:** Detailed use case of User's Account Management function



**Figure 2.3:** Detailed use case of Data Management function

### 2.2.3 Business process

## 2.3 Functional description

Sinh viên lựa chọn từ 4 đến 7 use case quan trọng nhất của đề án để đặc tả chi tiết. Mỗi đặc tả bao gồm ít nhất các thông tin sau: (i) Tên use case, (ii) Luồng sự kiện (chính và phát sinh), (iii) Tiền điều kiện, và (iv) Hậu điều kiện. Sinh viên chỉ vẽ bổ sung biểu đồ hoạt động khi đặc tả use case phức tạp. ok kj ghj

### 2.3.1 Description of use case A

### 2.3.2 Description of use case B

## 2.4 Non-functional requirement

Trong phần này, sinh viên đưa ra các yêu cầu khác nếu'ck có, gjkkhghjghjgbao gồm các yêu cầu phi chức năng như hiệucsdfsdfsdfccccc năng, độ tjkl;kjl;jkl;in cậ, tính dễ dùng, tính dễ bảo trì, hoặc các yêu cầu về mặt kỹ thuật như về CSDL, công nghệ sử dụng, v.v. sdfsdfasd asda sdasd dgfdfg



## CHAPTER 3. METHODOLOGY

Based on the real-life surveyed situation and the detailed system analysis in Chapter 2, Chapter 3 will center on the technologies used to develop the Data Management System. This system comprises three main components, namely, the Management Frontend, Backend, and the EPD Devices that are part of the system. The Management UI is built using NextJS and TailwindCSS to provide an easy-to-use dashboard for managing multiple devices and data efficiently. The UI also incorporates protocols to connect to HTTP, MQTT servers, and EPD devices via a Serial port. All data is processed and stored on the server using MongoDB and ExpressJS. The server communicates with EPD devices through the MQTT protocol, with RabbitMQ acting as a broker and SSL/TLS ensuring a secure connection. The EPD devices connect to the broker and subscribe to a specific topic to receive and handle any information requested by users from the front-end side, as well as send the status back to the broker.

### 3.1 Management UI (Front-end side)

#### 3.1.1 NextJS

Note: because of its ..., NextJS help reduce an enormous amount of additional third-party components, which helps creating the UI in an efficient way and also help speeding up the UI's overall responsive time.

#### 3.1.2 TailwindCSS

because of its strength, TailwindCSS helps reducing a lot of redundant css syntax by implement it right into class, which minify JS packets when the website is rendering

### 3.2 Server (Back-end side)

Because of the complexity of the system that needs both processing data from client side, and also communicating with EPD devices, this Server section is divided into 2 main parts: API server handling user's requests and storing data in MongoDB; and RabbitMQ broker communicating with devices and sending data back to ExpressJS server to process. The server used is a dedicated server hosted in Hetzner, which is a server company located in Europe, with quality is ensured to be among of the best in the world. This dedicated server also ensures the privacy, security and durable span.

### 3.2.1 API Server

This part of the back-end server is in charge of receiving user requests, managing data to the MQTT Server, and storing data in the system. The server is written in MVCS design pattern for code transparency and management and leverages **ExpressJS**, a minimal and flexible Node.js web application framework, to help enhance the server's capabilities. It enables efficient API creation and middleware integration, streamlining development and improving maintainability.

To store and secure user data, the back-end system uses **MongoDB Community Edition for Linux** (hereafter referred to as '**MongoDB**'), which stands out from other database solutions due to its distinct advantages in storing and processing data. MongoDB is adept at handling vast amounts of user and device data, storing it securely across distributed systems, in this case, the *clusters*. This design enhances data redundancy and security and allows for horizontal scaling, accommodating large-scale data without compromising on system performance. In tandem with **ExpressJS**, this also ensures efficient, high-performance query and transaction processing in a timely manner while keeping the development and maintenance processes streamlined and minimal.

Utilized for API design and documentation in this project, Swagger from OpenAPI, known as OpenAPI Specification (OAS), offers a comprehensive, detailed description of the API structure, including endpoints, operations, and parameters. This documentation tool not only aids in API development and testing by allowing for interaction with API resources without implementing logic, but it also streamlines client SDK generation across various programming languages. Leveraging Swagger documentation in the project helps speed up the development cycle and enhance efficiency in identifying bugs and issues during testing, ultimately resulting in a more robust and user-friendly application.

### 3.2.2 MQTT Broker

To enable real-time communication and data transfer between users and devices, the system also operates as an MQTT broker, facilitating the exchange of information between parties. This is acquired by using RabbitMQ, an open-source message broker software that enables applications to communicate with each other and exchange information efficiently. It acts as an intermediary for messaging by accepting and forwarding messages, making it a critical tool for handling asynchronous communication between EPD devices and the server. Its reliability and scalability make it a preferred choice for enterprises needing to ensure message delivery without loss, even in high-throughput scenarios. RabbitMQ's ability to decouple

processes also leads to more resilient and manageable application architectures.

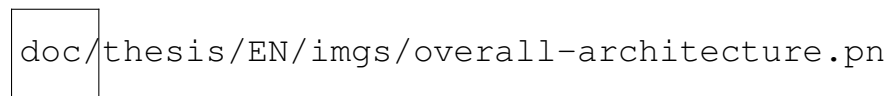
### **3.3 EPD device**

## CHAPTER 4. DESIGN, IMPLEMENTATION, AND EVALUATION

Given the distinctive characteristics of the system discussed in Chapter 3, it is essential to adopt an architecture or combination of multiple architectures to efficiently address these specific requirements while ensuring scalability, reliability, and seamless communication between different components. This chapter aims to deliver an in-depth exploration of the architecture design implemented in the project and provide a detailed understanding of each element in the system.

### 4.1 Architecture design

#### 4.1.1 Software architecture selection



**Figure 4.1:** Overall architecture of the system

**Figure 4.1** illustrates the system's general architecture and provides an overview of its data flow. The system strategically uses the combination of MVCS (Model-View-Controller-Service) and microservice architectures to take advantage of their unique strengths in specific aspects. This hybrid architecture leverages the benefits of both architectural styles, such as scalability, flexibility, and the ability to use different technologies and patterns within each service.

Microservices architecture is a software development method that structures an application as a collection of loosely coupled services. In this architecture, each service can be developed, deployed, and scaled independently, allowing greater scalability, flexibility, and agility than monolithic architectures, as teams can update or fix individual components without impacting the entire system in case of failure. Thanks to its ability to accommodate the dynamic and distributed natures of IoT applications, microservice architecture is frequently used in various IoT projects. It allows organizations to build more agile and adaptive applications to handle the complex demands of modern business environments.

On the other hand, the Model-View-Controller-Service (MVCS) architecture is an extension of the traditional Model-View-Controller (MVC) pattern. It introduces a service layer sitting between the Controller and the Model, responsible for containing business logic and rules. This layer abstracts complex business operations, allowing Controllers to focus on handling incoming requests and delegating the heavy lifting to Services. This design pattern is well-suited to many web applica-

tions, providing a structured and organized approach to developing complex applications and making it easier to manage, maintain, and scale the system over time.

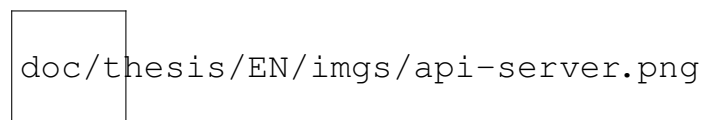
By combining these two architectures, the system is better equipped to handle the dynamic and distributed nature of IoT applications. It also enables organizations to build more agile and adaptive applications to address the complex demands of modern business environments while ensuring scalability, reliability, and seamless communication between different components.

#### **4.1.2 Overall design**

The system is divided into independently deployable services, including the API back-end server and MQTT Server. These services communicate via the MQTT protocol, and the MQTT Server also acts as a broker to relay data to the EPD devices (Figure 4.3). API server follows the MVCS pattern with three main components: Controllers handling incoming requests and coordinating responses, Services containing the business logic and rules of the application, and Models interacting with the database, managing data storage and retrieval, while Management UI and EPD devices act as a View component (Figure 4.2).

##### **a, MVCS architecture**

The system has three data types, and the packages of each type handling operation share the same structure: the Controller handles requests about the data type and delegates them to the Service, which processes the requests and uses the Model to interact with the database.



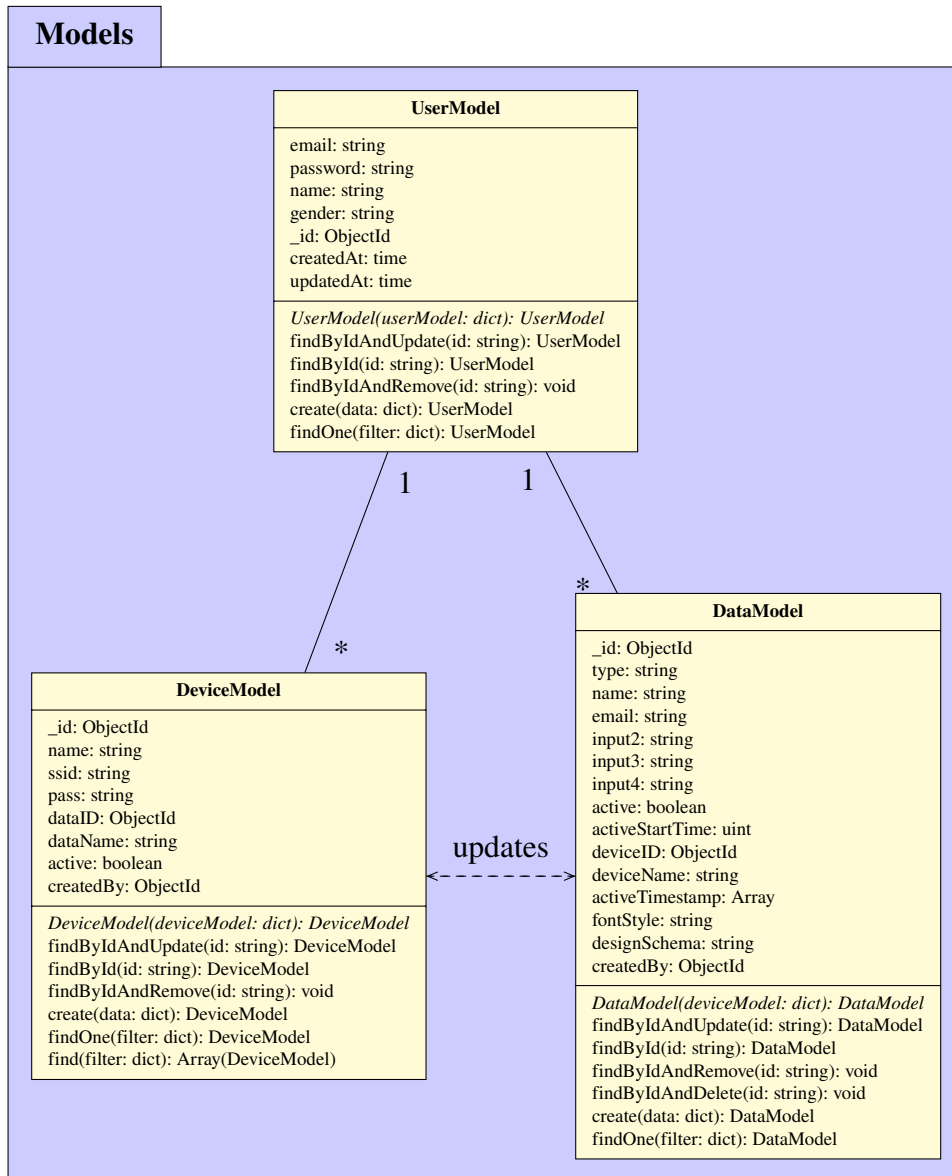
**Figure 4.2:** MVCS architecture

##### **b, MQTT Server**

This service in the system acts as an intermediary, managing the state of all MQTT client connections, subscriptions, and message exchanges.



**Figure 4.3:** MQTT microservice



**Figure 4.4:** General use case diagram of the data management system

4.1.3 Detailed package design

a, Models

b, Services

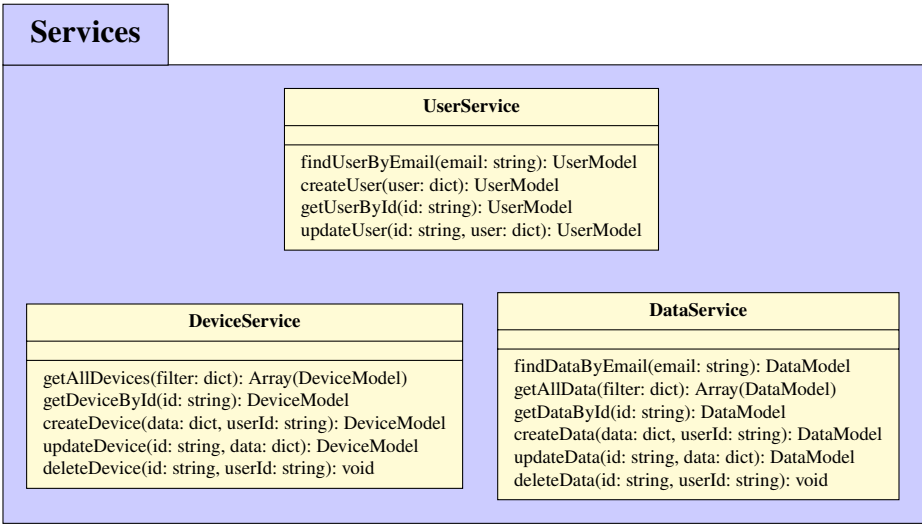


Figure 4.5: General use case diagram of the data management system

c, Controllers

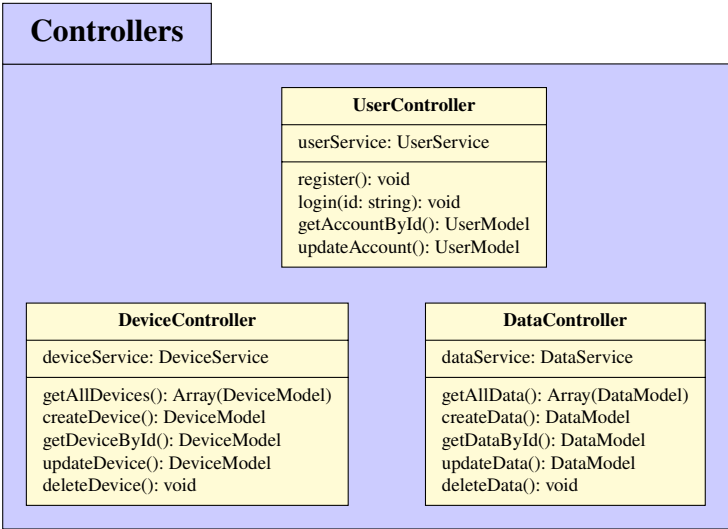


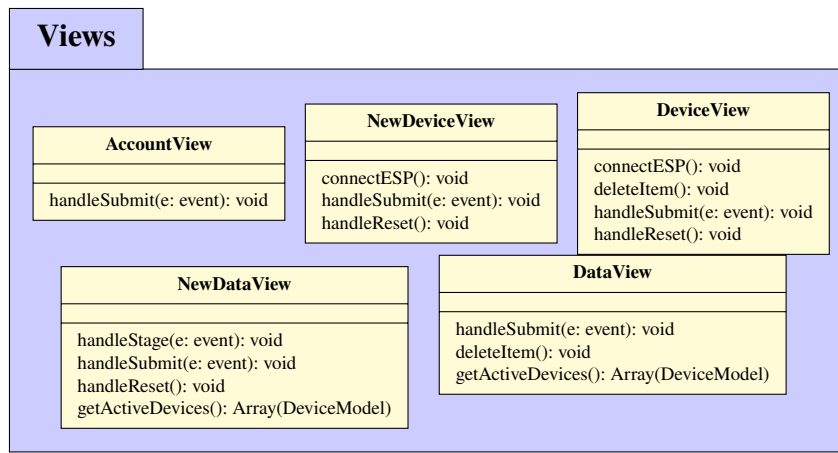
Figure 4.6: General use case diagram of the data management system

d, Views

4.2 Detailed design

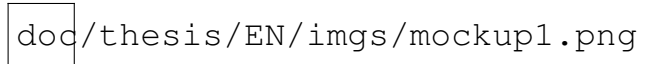
4.2.1 User interface design

The architecture and user interface of this system are meticulously tailored for computer displays, ensuring optimal functionality and visual experience on larger screens. The design is customized to leverage the expansive real estate of computer monitors, facilitating ease of use and comprehensive information display. This fo-

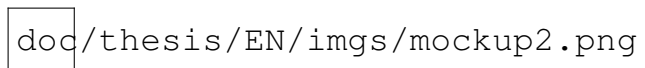


**Figure 4.7:** General use case diagram of the data management system

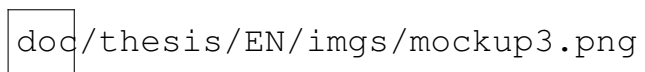
cus on computer-targeted design means that the system is not intended for mobile use, and as such, it may not provide an ideal user experience or full functionality on smaller mobile device screens. The decision to specialize in computer displays stems from a commitment to delivering a high-quality, immersive experience that fully utilizes the capabilities and advantages of larger screens typically associated with desktop or laptop computers.



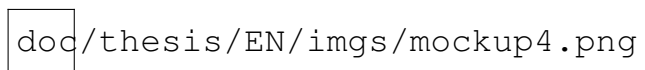
**Figure 4.8:** Mockup display of authentication page



**Figure 4.9:** Mockup display of dashboard page



**Figure 4.10:** Mockup display of modal component



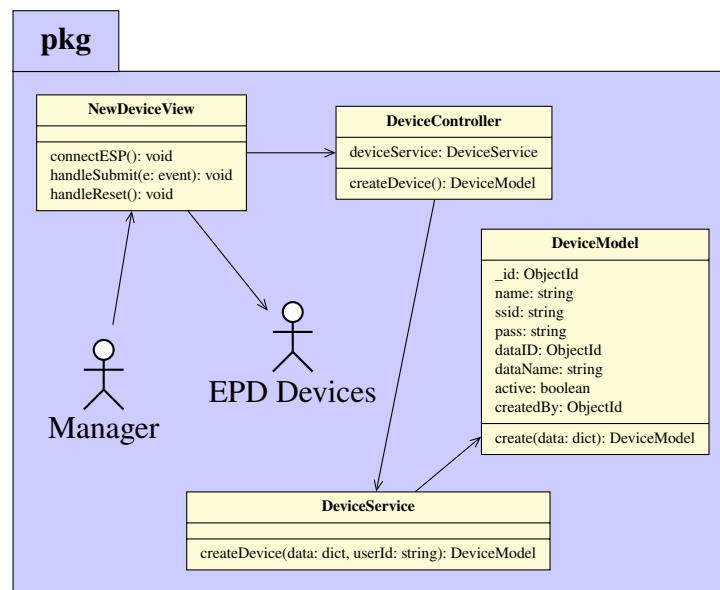
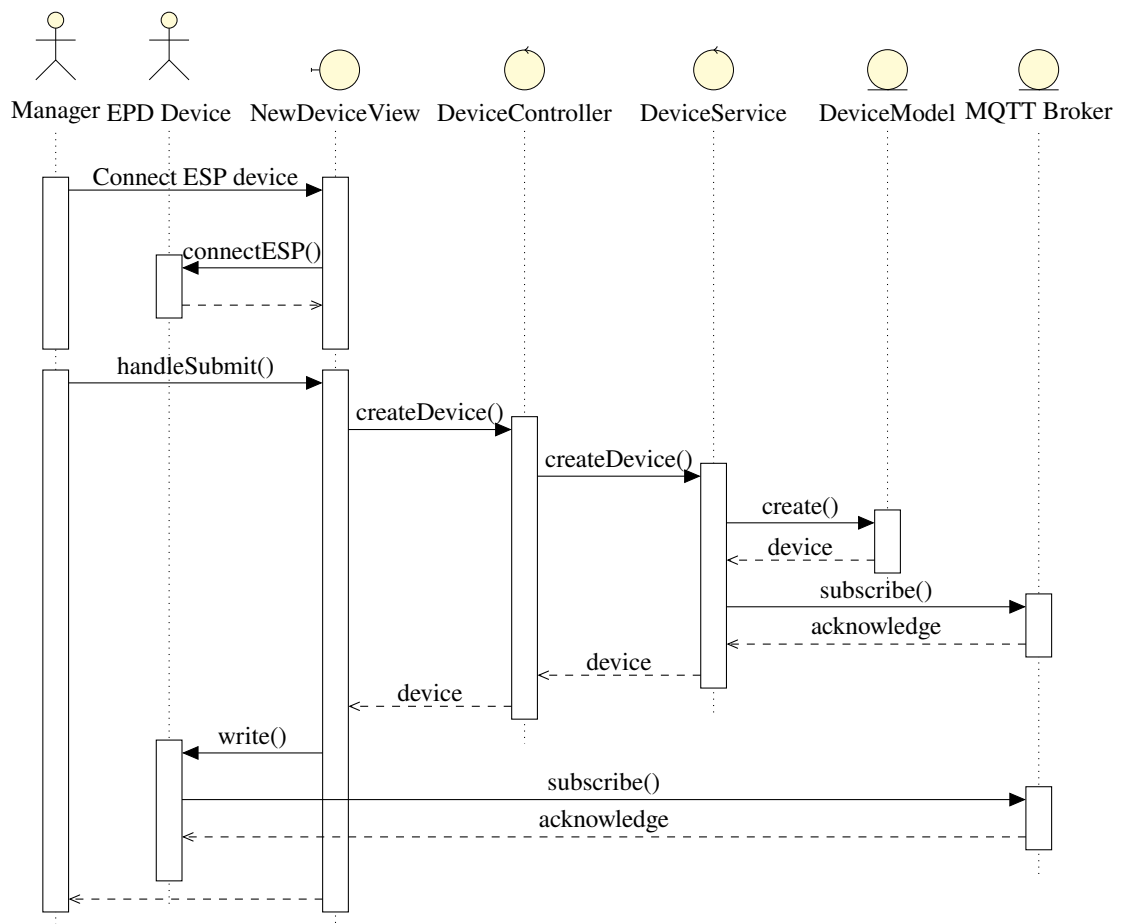
**Figure 4.11:** Mockup display of creation page

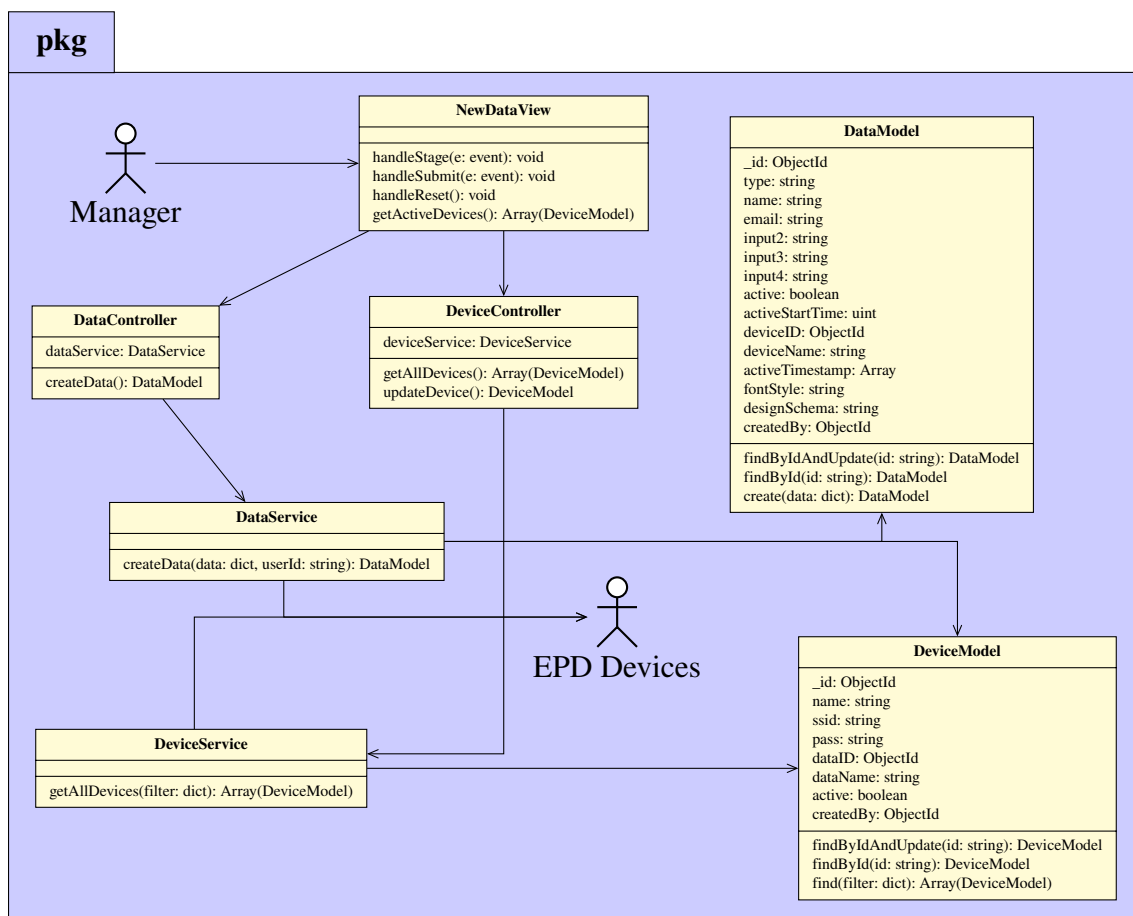
### 4.2.2 Layer design

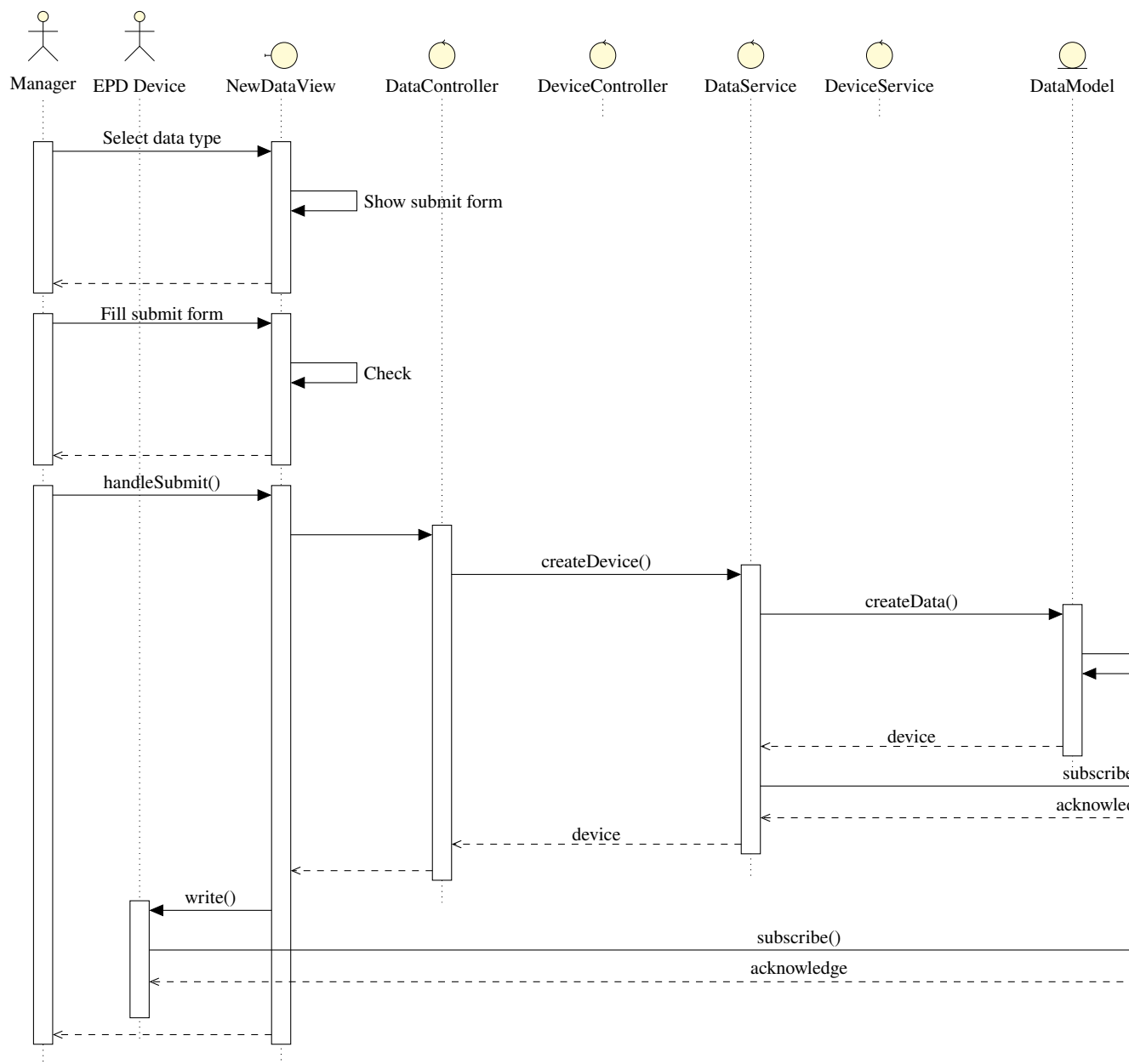
Each sub-section below demonstrates the workflow of each class participating in each use case in the form of a class diagram and a responding sequence diagram. Each class only shows relevant methods used in the use case.



## a, Class and sequence diagram of use case "Register a new device"

**Figure 4.12:** Detailed use case of Data Management function**Figure 4.13:** Caption

**b, Class and sequence diagram of use case "Add a new data"****Figure 4.14:** Detailed use case of Data Management function



**Figure 4.15:** Caption

c, Class and sequence diagram of use case "Change a device information"

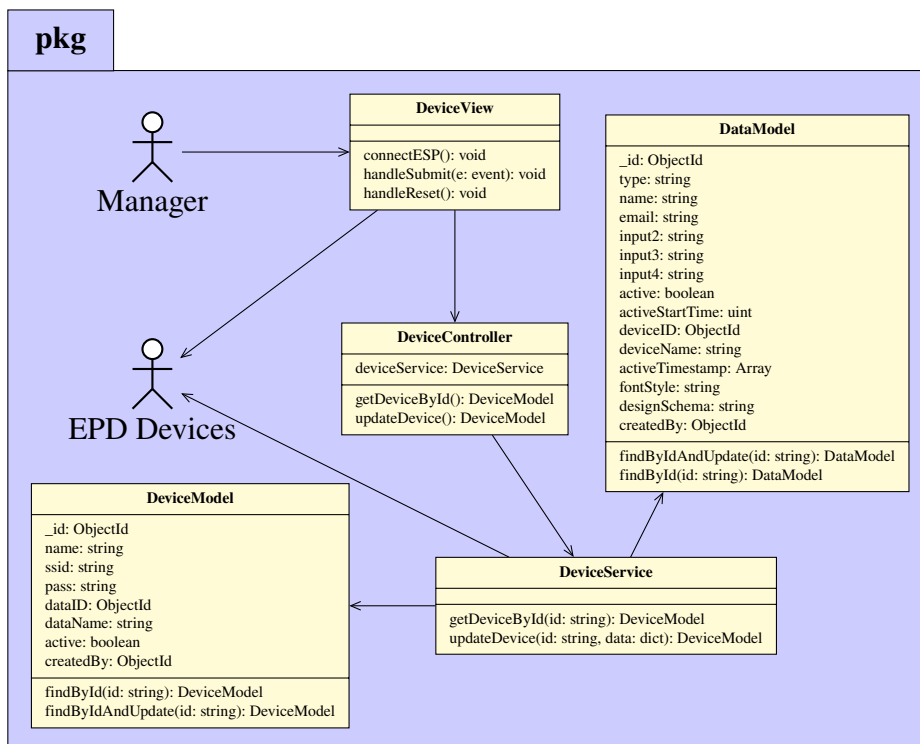
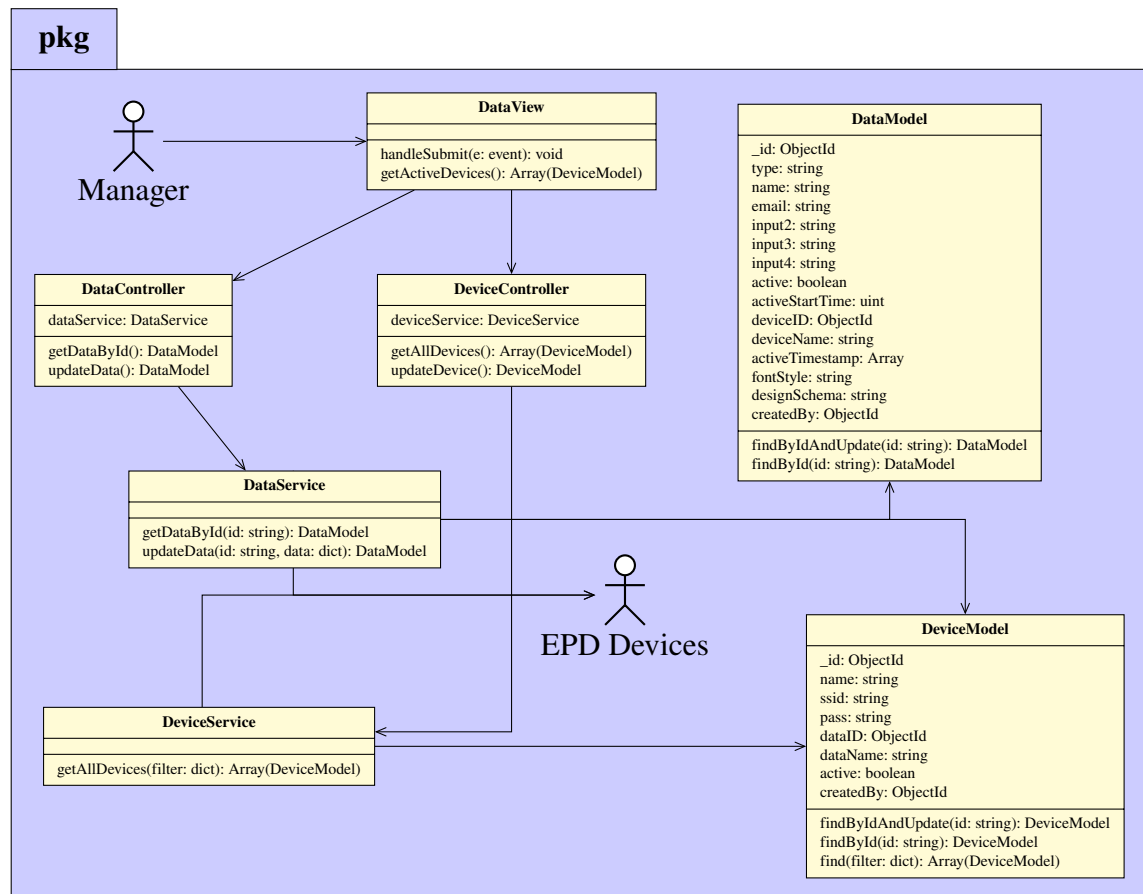


Figure 4.16: Detailed use case of Data Management function

**d, Class and sequence diagram of use case "Change data information"**



**Figure 4.17:** Detailed use case of Data Management function

## e, Class and sequence diagram of use case "Remove a device"

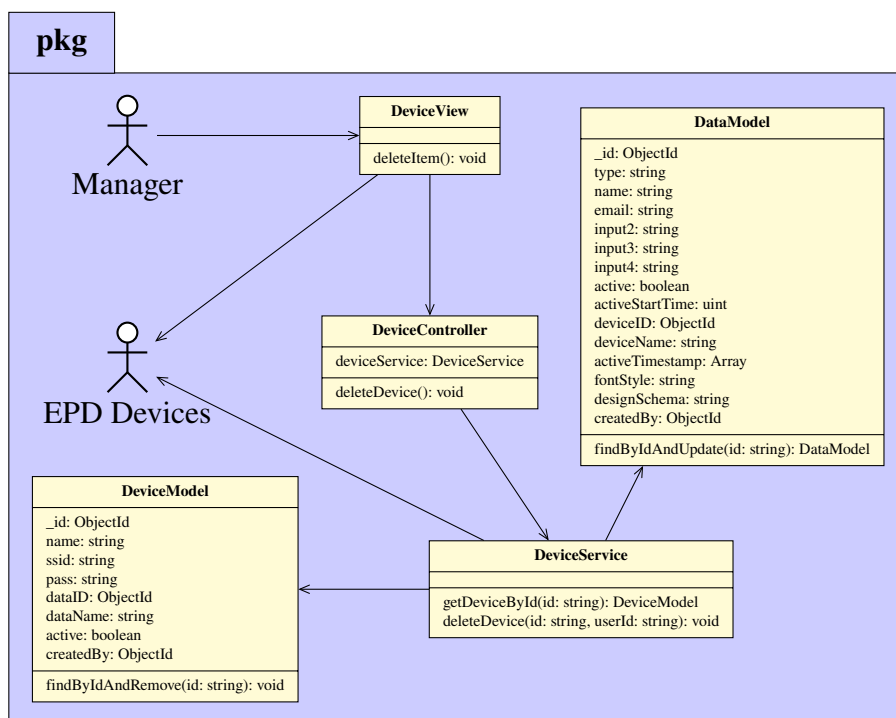


Figure 4.18: Detailed use case of Data Management function

## f, Class and sequence diagram of use case "Remove a data"

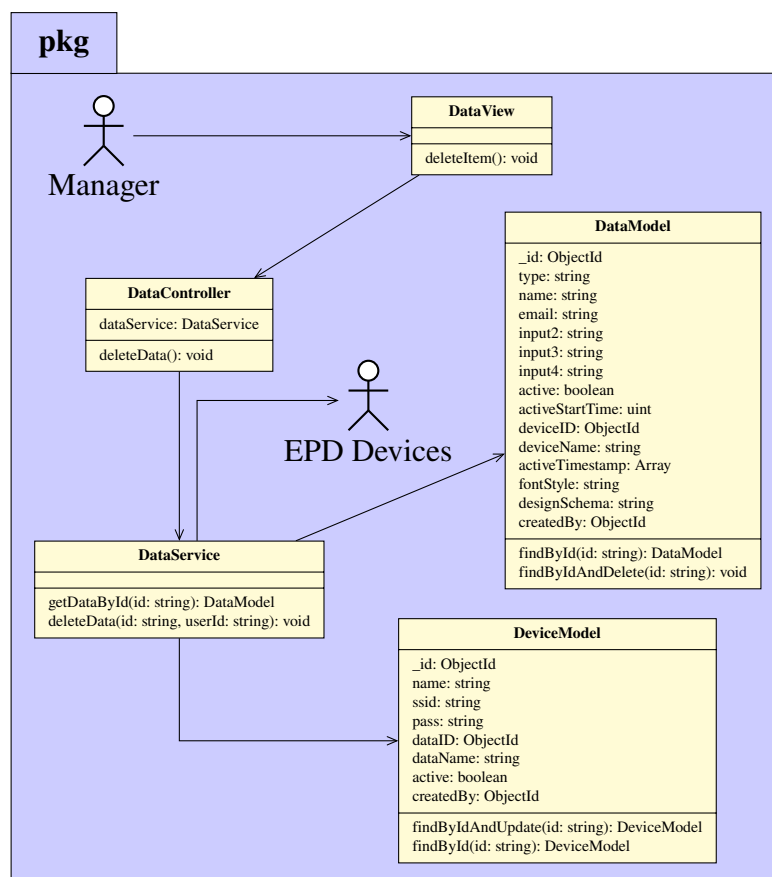
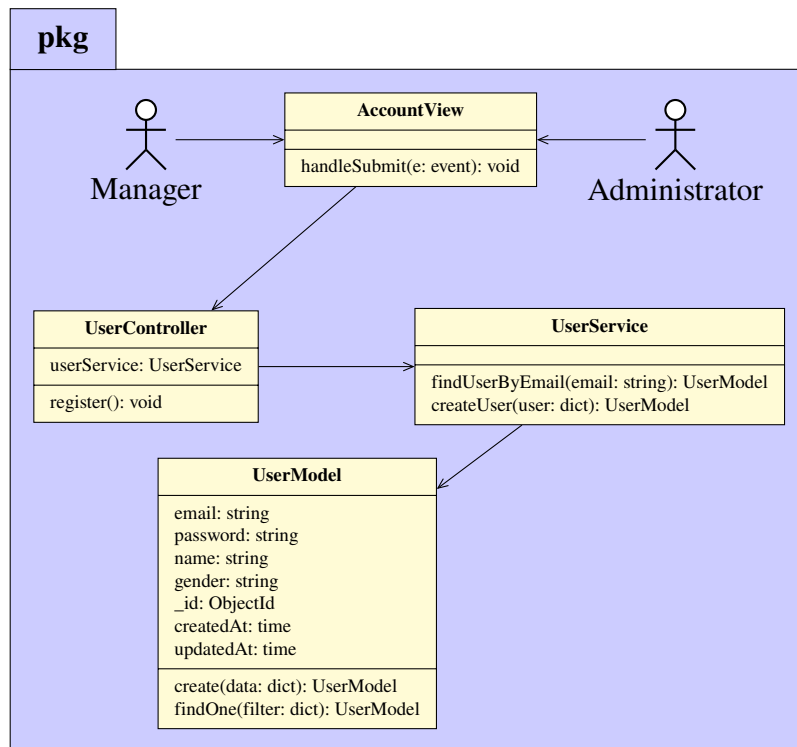
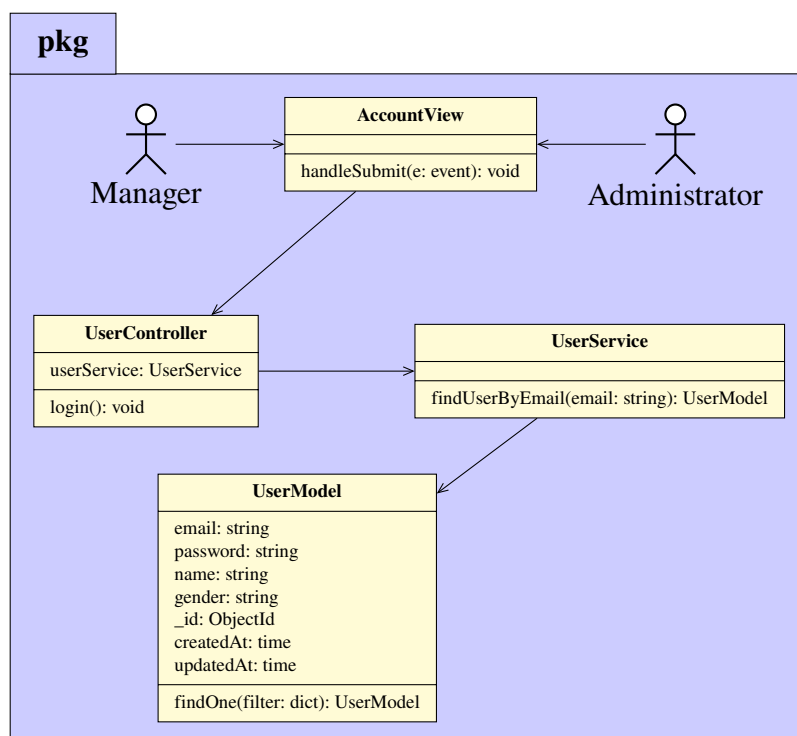


Figure 4.19: Detailed use case of Data Management function

**g, Class and sequence diagram of use case "Register new account"****Figure 4.20:** Detailed use case of Data Management function**h, Class and sequence diagram of use case "Sign in"****Figure 4.21:** Detailed use case of Data Management function



### 4.2.3 Database design

Phần này có độ dài từ hai đến bốn trang. Sinh viên thiết kế, vẽ và giải thích biểu đồ thực thể liên kết (E-R diagram). Từ đó, sinh viên thiết kế cơ sở dữ liệu tùy theo hệ quản trị cơ sở dữ liệu mà mình sử dụng (SQL, NoSQL, Firebase, v.v.)

#### a, Entity-Relationship diagram

#### b, Detail database design

Collection	Field Name	Field Type	Required	Description
User	_id	ObjectId	*	User's ID used in the system
	email	String	*	User's personal email
	password	String	*	Encrypted password of user's account
	name	String	*	User's name
	gender	String		User's gender
	createdAt	Time	*	Time of account creation
	updatedAt	Time	*	Last account update time
Data	_id	ObjectId	*	Data's ID used in the system
	type	String	*	Data type
	name	String	*	Data name
	email	String		First information: email
	input2	String		Second information
	input3	String		Third information
	input4	String		Fourth information
	active	boolean	*	Display status on EPD device
	activeStartTime	uint	*	The display start time
	deviceID	ObjectId		ID of displayed device
	deviceName	String		Name of the displayed device
	activeTimestamp	Array	*	A list of display time range
	fontStyle	String		Custom display font style
	designSchema	String		Custom display theme
	createdBy	ObjectId	*	ID of the user creating the data
Device	_id	ObjectId	*	Device's ID used in the system
	name	String		Device name
	ssid	String	*	SSID of the network the device is connecting to
	pass	String	*	Password of the network the device is connecting to

Collection	Field Name	Field Type	Required	Description
	dataID	ObjectId		ID of displayed data on the device
	dataName	String		Name of displayed data on the device
	active	boolean	*	Connection status of device
	createdBy	ObjectId	*	ID of the user creating the data

**Table 4.1:** Your Table Caption

## 4.3 Application Building

### 4.3.1 Libraries and Tools

In the development of this project, a suite of sophisticated tools was employed to ensure efficiency and quality in the coding process. Central to this toolkit was Visual Studio Code (VS Code), a versatile and powerful code editor known for its user-friendly interface and wide range of extensions. Alongside VS Code, a variety of other tools were integral to the workflow, such as Mongo Compass to manage the MongoDB database and PlatformIO to work with ESP devices. These also included version control systems for tracking changes and collaborating with team members, debugging tools for identifying and resolving issues, and project management software to keep the development process streamlined and on schedule. Table 4.3 below lists all the tools and applications used during the development process and also the systems used in the project.

Tools used	Type	Version	Description	URL
Visual Studio Code - Insiders (VSCode)	Tools	1.86.0-insider	Main development environment	<a href="https://code.visualstudio.com/insiders">https://code.visualstudio.com/insiders</a>
PlatformIO	VSCode Plugins	v3.3.2	A development ecosystem for embedded and IoT applications	<a href="https://platformio.org">https://platformio.org</a>
Remote Development	VSCode Plugins	v0.4.1	A plugin allowing VSCode to develop on remote systems	<a href="https://code.visualstudio.com/docs/remote/remote-overview">https://code.visualstudio.com/docs/remote/remote-overview</a>
Wokwi Simulator	VSCode Plugins	v2.3.2	Simulator for Embedded & IoT Systems	<a href="https://wokwi.com">https://wokwi.com</a>
NodeJS	Language	v20.9.0	A JavaScript runtime for building scalable network applications	<a href="https://nodejs.org">https://nodejs.org</a>
NextJS	Framework	v13.4.5	A framework for server-side rendered React applications	<a href="https://nextjs.org">https://nextjs.org</a>
TailwindCSS	Framework	v3.3.2	A utility-first CSS framework for rapid UI development	<a href="https://tailwindcss.com">https://tailwindcss.com</a>
ExpressJS	Framework	v4.18.2	A web application framework for Node.js	<a href="https://expressjs.com">https://expressjs.com</a>
Mongoose	Library	v8.0.0	A library for MongoDB and Node.js data modeling	<a href="https://mongoosejs.com">https://mongoosejs.com</a>
MQTT NPM Package (mqtt)	Library	v5.3.0	A library for implementing MQTT protocol in Node.js	<a href="https://www.npmjs.com/package/mqtt">https://www.npmjs.com/package/mqtt</a>
WaveShare EPD Library	Library	v1.0	A library for interfacing with e-paper displays	
Node Version Manager (nvm)	Tools	v0.39.2	A tool for managing multiple Node.js versions	<a href="https://github.com/nvm-sh/nvm">https://github.com/nvm-sh/nvm</a>
Node Package Manager (npm)	Tools	v10.1.0	A package manager managing dependencies in Node.js projects	<a href="https://www.npmjs.com">https://www.npmjs.com</a>
MongoDB Community Edition for Linux	Tools	v7.0.2	An open-source document database for Linux systems	<a href="https://www.mongodb.com/try/download/community">https://www.mongodb.com/try/download/community</a>
MongoDB Compass	Tools	v1.40.4	A GUI for MongoDB, simplifying data visualization and management	<a href="https://www.mongodb.com/products/compass">https://www.mongodb.com/products/compass</a>
RabbitMQ	Tools	v3.12.10	An open-source message broker software	<a href="https://www.rabbitmq.com">https://www.rabbitmq.com</a>
ESP32 C3-Supermini	Device		A compact microcontroller module for EPD devices	
WeAct Studio E-paper 2.9inch display	Device		A low-power display module for high-contrast, readable content	<a href="https://www.weact-tc.cn">https://www.weact-tc.cn</a>
Ubuntu Server	System		A dedicated server in Hetzner, for Web hosting and MQTT Broker	<a href="https://www.hetzner.com">https://www.hetzner.com</a>
Nginx	Tools	nginx/1.24.0	A high-performance web server and reverse proxy	<a href="https://nginx.org">https://nginx.org</a>
Cloudflare	Tools		A service for website performance optimization and security	<a href="https://www.cloudflare.com">https://www.cloudflare.com</a>

Table 4.2: Danh sách thư viện và công cụ sử dụng

### 4.3.2 Achievement

Sinh viên trước tiên mô tả kết quả đạt được của mình là gì, ví dụ như các sản phẩm được đóng gói là gì, bao gồm những thành phần nào, ý nghĩa, vai trò?

Sinh viên cần thống kê các thông tin về ứng dụng của mình như: số dòng code, số lớp, số gói, dung lượng toàn bộ mã nguồn, dung lượng của từng sản phẩm đóng gói, v.v. Tương tự như phần liệt kê về công cụ sử dụng, sinh viên cũng nên dùng bảng để mô tả phần thông tin thống kê này.

With the help of tools and libraries in the table 4.3 above, the project has reached significant milestones and also released a minimum viable product (MVP) including the Management UI, the back-end server, and a couple of EPD devices that can be implemented in many small business environments. While having some minor performance issues, this MVP still proved its usefulness in various use cases, such as mini-markets, schools, and offices, unveiling the enormous potential of the system in a broader spectrum of the service industry.

The whole system is running on the website so the users don't need to install any additional applications. Also, the users just need to plug the EPD device into the computer via a USB port when needed without additional drivers, and the website will automatically recognize the device. Details of the running system are shown in the table below

Description	URL
Management UI	<a href="https://epaper.artsakh.ventures">https://epaper.artsakh.ventures</a>
MongoDB Databases	<a href="mongodb://epaper.artsakh.ventures:27017">mongodb://epaper.artsakh.ventures:27017</a>
Back-end server	<a href="https://epaper.artsakh.ventures/api">https://epaper.artsakh.ventures/api</a>
MQTT Broker	<a href="mqtt://mqtt.epaper.artsakh.ventures">mqtt://mqtt.epaper.artsakh.ventures</a>
MQTT Management UI	<a href="https://mqtt.epaper.artsakh.ventures/dashboard">https://mqtt.epaper.artsakh.ventures/dashboard</a>

**Table 4.3:** Danh sách thư viện và công cụ sử dụng

### 4.3.3 Illustration of main functions

To manage EPD Devices and data, users need to log in with credentials as illustrated in the picture ???. After logging in, the user can choose to create data/device or go to the dashboard to view from the sidebar on the left.

When creating data/device, the user can easily follow on-screen instructions and fill in the required information. All the steps are annotated in detail, and the rendered display is showing in real-time, which helps improve user experience.

The device is running on battery, which can last for 5 - 6 hours before needing to recharge. In the scope of the project, this is still reasonable and acceptable as the product is still in the prototype stage, and not optimized for higher endurance in a real-life environment.

## 4.4 Testing

Phần này có độ dài từ hai đến ba trang. Sinh viên thiết kế các trường hợp kiểm thử cho hai đến ba chức năng quan trọng nhất. Sinh viên cần chỉ rõ các kỹ thuật kiểm thử đã sử dụng. Chi tiết các trường hợp kiểm thử khác, nếu muốn trình bày, sinh viên đưa vào phần phụ lục. Sinh viên sau cùng tổng kết về số lượng các trường

hợp kiểm thử và kết quả kiểm thử. Sinh viên cần phân tích lý do nếu kết quả kiểm thử không đạt.

### **4.5 Deployment**

All of the project's source code is hosted and managed on Github, using Continuous Integration and Continuous Delivery (CI/CD) to automate the deployment. The

Sinh viên trình bày mô hình và/hoặc cách thức triển khai thử nghiệm/thực tế. Ứng dụng của sinh viên được triển khai trên server/thiết bị gì, cấu hình như thế nào. Kết quả triển khai thử nghiệm nếu có (số lượng người dùng, số lượng truy cập, thời gian phản hồi, phản hồi người dùng, khả năng chịu tải, các thống kê, v.v.)

## CHAPTER 5. SOLUTION AND CONTRIBUTION

Chương này có độ dài tối thiểu 5 trang, tối đa không giới hạn.<sup>1</sup> Sinh viên cần trình bày tất cả những nội dung đóng góp mà mình thấy tâm đắc nhất trong suốt quá trình làm ĐATN. Đó có thể là một loạt các vấn đề khó khăn mà sinh viên đã từng bước giải quyết được, là giải thuật cho một bài toán cụ thể, là giải pháp tổng quát cho một lớp bài toán, hoặc là mô hình/kiến trúc hữu hiệu nào đó được sinh viên thiết kế.

Chương này **là cơ sở quan trọng** để các thầy cô đánh giá sinh viên. Vì vậy, sinh viên cần phát huy tính sáng tạo, khả năng phân tích, phản biện, lập luận, tổng quát hóa vấn đề và tập trung viết cho thật tốt. Mỗi giải pháp hoặc đóng góp của sinh viên cần được trình bày trong một mục độc lập bao gồm ba mục con: (i) dẫn dắt/giới thiệu về bài toán/vấn đề, (ii) giải pháp, và (iii) kết quả đạt được (nếu có).

Sinh viên lưu ý **không trình bày lặp lại nội dung**. Những nội dung đã trình bày chi tiết trong các chương trước không được trình bày lại trong chương này. Vì vậy, với nội dung hay, mang tính đóng góp/giải pháp, sinh viên chỉ nên tóm lược/mô tả sơ bộ trong các chương trước, đồng thời tạo tham chiếu chéo tới đề mục tương ứng trong Chương 5 này. Chi tiết thông tin về đóng góp/giải pháp được trình bày trong mục đó.

Ví dụ, trong Chương 4, sinh viên có thiết kế được kiến trúc đáng lưu ý gì đó, là sự kết hợp của các kiến trúc MVC, MVP, SOA, v.v. Khi đó, sinh viên sẽ chỉ mô tả ngắn gọn kiến trúc đó ở Chương 4, rồi thêm các câu có dạng: “Chi tiết về kiến trúc này sẽ được trình bày trong phần 5.1”.

---

<sup>1</sup>Trong trường hợp phần này dưới 5 trang thì sinh viên nên gộp vào phần kết luận, không tách ra một chương riêng rẽ nữa.

## **CHAPTER 6. CONCLUSION AND FUTURE WORK**

### **6.1 Conclusion**

Sinh viên so sánh kết quả nghiên cứu hoặc sản phẩm của mình với các nghiên cứu hoặc sản phẩm tương tự.

Sinh viên phân tích trong suốt quá trình thực hiện ĐATN, mình đã làm được gì, chưa làm được gì, các đóng góp nổi bật là gì, và tổng hợp những bài học kinh nghiệm rút ra nếu có.

### **6.2 Future work**

Trong phần này, sinh viên trình bày định hướng công việc trong tương lai để hoàn thiện sản phẩm hoặc nghiên cứu của mình.

Trước tiên, sinh viên trình bày các công việc cần thiết để hoàn thiện các chức năng/nhiệm vụ đã làm. Sau đó sinh viên phân tích các hướng đi mới cho phép cải thiện và nâng cấp các chức năng/nhiệm vụ đã làm.

## SHORT NOTICES ON REFERENCE

Lưu ý: Sinh viên không được đưa bài giảng/slide, các trang Wikipedia, hoặc các trang web thông thường làm tài liệu tham khảo.

Một trang web được phép dùng làm tài liệu tham khảo **chỉ khi** nó là công bố chính thống của cá nhân hoặc tổ chức nào đó. Ví dụ, trang web đặc tả ngôn ngữ XML của tổ chức W3C <https://www.w3.org/TR/2008/REC-xml-20081126/> là TLTK hợp lệ.

Có năm loại tài liệu tham khảo mà sinh viên phải tuân thủ đúng quy định về cách thức liệt kê thông tin như sau. Lưu ý: các phần văn bản trong cặp dấu < > dưới đây chỉ là hướng dẫn khai báo cho từng loại tài liệu tham khảo; sinh viên cần xóa các phần văn bản này trong ĐATN của mình.

<**Bài báo đăng trên tạp chí khoa học:** Tên tác giả, tên bài báo, tên tạp chí, volume, từ trang đến trang (nếu có), nhà xuất bản, năm xuất bản >

[1] E. H. Hovy, "Automated discourse generation using discourse structure relations," *Artificial intelligence*, vol. 63, no. 1-2, pp. 341–385, 1993

<**Sách:** Tên tác giả, tên sách, volume (nếu có), lần tái bản (nếu có), nhà xuất bản, năm xuất bản>

[2] L. L. Peterson and B. S. Davie, *Computer networks: a systems approach*. Elsevier, 2007.

[3] N. T. Hải, *Mạng máy tính và các hệ thống mở*. Nhà xuất bản giáo dục, 1999.

<**Tập san Báo cáo Hội nghị Khoa học:** Tên tác giả, tên báo cáo, tên hội nghị, ngày (nếu có), địa điểm hội nghị, năm xuất bản>

[4] M. Poesio and B. Di Eugenio, "Discourse structure and anaphoric accessibility," in *ESSLLI workshop on information structure, discourse structure and discourse semantics*, Copenhagen, Denmark, 2001, pp. 129–143.

<**Đồ án tốt nghiệp, Luận văn Thạc sĩ, Tiến sĩ:** Tên tác giả, tên đồ án/luận văn, loại đồ án/luận văn, tên trường, địa điểm, năm xuất bản>

[5] A. Knott, "A data-driven methodology for motivating a set of coherence relations," Ph.D. dissertation, The University of Edinburgh, UK, 1996.

<**Tài liệu tham khảo từ Internet:** Tên tác giả (nếu có), tựa đề, cơ quan (nếu có), địa chỉ trang web, thời gian lần cuối truy cập trang web>

[6] T. Berners-Lee, *Hypertext transfer protocol (HTTP)*. [Online]. Available:



`ftp://info.cern.ch/pub/www/doc/http-spec.txt.Z` (visited on 09/30/2010).

[7] Princeton University, *Wordnet*. [Online]. Available: `http://www.cogsci.princeton.edu/~wn/index.shtml` (visited on 09/30/2010).

## REFERENCE

- [1] E. H. Hovy, “Automated discourse generation using discourse structure relations,” *Artificial intelligence*, vol. 63, no. 1-2, pp. 341–385, 1993.
- [2] L. L. Peterson and B. S. Davie, *Computer networks: a systems approach*. Elsevier, 2007.
- [3] N. T. Hải, *Mạng máy tính và các hệ thống mở*. Nhà xuất bản giáo dục, 1999.
- [4] M. Poesio and B. Di Eugenio, “Discourse structure and anaphoric accessibility,” in *ESSLLI workshop on information structure, discourse structure and discourse semantics, Copenhagen, Denmark*, 2001, pp. 129–143.
- [5] A. Knott, “A data-driven methodology for motivating a set of coherence relations,” Ph.D. dissertation, The University of Edinburgh, UK, 1996.
- [6] T. Berners-Lee, *Hypertext transfer protocol (HTTP)*. [Online]. Available: <ftp://info.cern.ch/pub/www/doc/http-spec.txt>. Z (visited on 09/30/2010).
- [7] Princeton University, *Wordnet*. [Online]. Available: <http://www.cogsci.princeton.edu/~wn/index.shtml> (visited on 09/30/2010).

# APPENDIX

## A. THESIS WRITING GUIDELINE

### A.1 General Regulations

Dưới đây là một số quy định và hướng dẫn viết đồ án tốt nghiệp mà bắt buộc sinh viên phải đọc kỹ và tuân thủ nghiêm ngặt.

Sinh viên cần đảm bảo tính thống nhất toàn báo cáo (font chữ, căn dòng hai bên, hình ảnh, bảng, margin trang, đánh số trang, v.v.). Để làm được như vậy, sinh viên chỉ cần sử dụng các định dạng theo đúng template ĐATN này. Khi paste nội dung văn bản từ tài liệu khác của mình, sinh viên cần chọn kiểu Copy là “Text Only” để định dạng văn bản của template không bị phá vỡ/vi phạm.

Tuyệt đối cấm sinh viên đạo văn. Sinh viên cần ghi rõ nguồn cho tất cả những gì không tự mình viết/vẽ lên, bao gồm các câu trích dẫn, các hình ảnh, bảng biểu, v.v. Khi bị phát hiện, sinh viên sẽ không được phép bảo vệ ĐATN.

Tất cả các hình vẽ, bảng biểu, công thức, và tài liệu tham khảo trong ĐATN nhất thiết phải được SV giải thích và tham chiếu tới ít nhất một lần. Không chấp nhận các trường hợp sinh viên đưa ra hình ảnh, bảng biểu tùy hứng và không có lời mô tả/giải thích nào.

Sinh viên tuyệt đối không trình bày ĐATN theo kiểu viết ý hoặc gạch đầu dòng. ĐATN không phải là một slide thuyết trình; khi người đọc không hiểu sẽ không có ai giải thích hộ. Sinh viên cần viết thành các đoạn văn và phân tích, diễn giải đầy đủ, rõ ràng. Câu văn cần đúng ngữ pháp, đầy đủ chủ ngữ, vị ngữ và các thành phần câu. Khi thực sự cần liệt kê, sinh viên nên liệt kê theo phong cách khoa học với các ký tự La Mã. Ví dụ, nhiều sinh viên luôn cảm thấy hối hận vì (i) chưa cố gắng hết mình, (ii) chưa sắp xếp thời gian học/choi một cách hợp lý, (iii) chưa tìm được người yêu để chia sẻ quãng đời sinh viên vất vả, và (iv) viết ĐATN một cách cẩu thả.

Trong một số trường hợp nhất thiết phải dùng các bullet để liệt kê, sinh viên cần thống nhất Style cho toàn bộ các bullet các cấp mà mình sử dụng đến trong báo cáo. Nếu dùng bullet cấp 1 là hình tròn đen, toàn bộ báo cáo cần thống nhất cách dùng như vậy; ví dụ như sau:

- Đây là mục 1 – Thực sự không còn cách nào khác tôi mới dùng đến việc bullet trong báo cáo.
- Đây là mục 2 – Nghĩ lại thì tôi có thể không cần dùng bullet cũng được. Nên tôi sẽ xóa bullet và tổ chức lại hai mục này trong báo cáo của mình cho khoa học hơn. Tôi muốn thầy cô và người đọc cảm nhận được tâm huyết của tôi

trong từng trang báo cáo ĐATN.

## A.2 Majoring

Sinh viên lưu ý viết đúng ngành/chuyên ngành trên bìa và trên gáy theo đúng quy định của Trường. Ngành học hay chuyên ngành học phụ thuộc vào ngành học mà sinh viên đăng ký. Sinh viên có thể đăng nhập trên trang quản lý học tập của mình để xem lại chính xác ngành học của mình.

Một số ví dụ sinh viên có thể tham khảo dưới đây, trong trường hợp có chuyên ngành thì sinh viên không cần ghi chuyên ngành:

- Đối với kỹ sư chính quy:
  - Từ K61 trở về trước: Ngành Kỹ thuật phần mềm
  - Từ K62 trở về sau: Ngành Khoa học máy tính
- Đối với cử nhân:
  - Ngành Công nghệ thông tin
- Đối với chương trình EliteTech:
  - Chương trình Việt Nhật/KSTN: Ngành Công nghệ thông tin
  - Chương trình ICT Global: Ngành Information Technology
  - Chương trình DS&AI: Ngành Khoa học dữ liệu

## A.3 Bulleting and Numbering

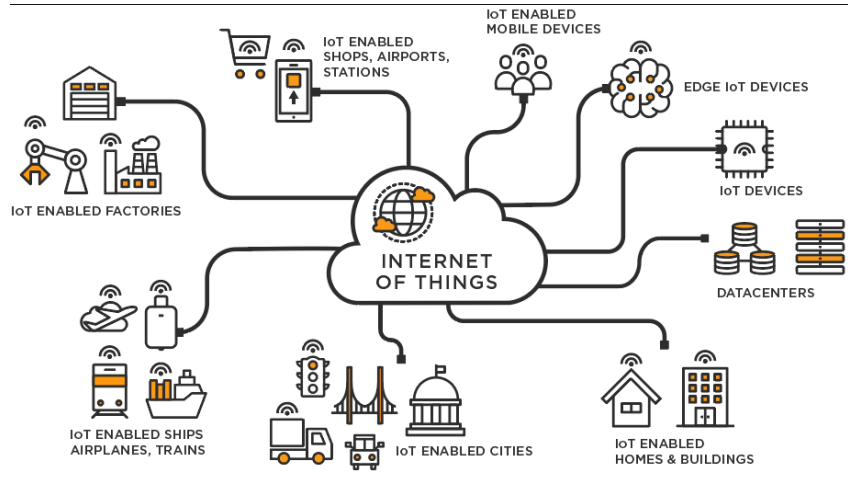
Việc sử dụng danh sách trong LaTeX khá đơn giản và không yêu cầu sinh viên phải thêm bất kỳ gói bổ sung nào. LaTeX cung cấp hai môi trường liệt kê đó là:

- Đánh dấu (bullet) là kiểu liệt kê không có thứ tự. Để sử dụng kiểu liệt kê đánh dấu, chúng ta khai báo như sau

```
\begin{itemize}
\item Nội dung thứ nhất được viết ở đây.
\item Nội dung thứ hai được viết ở đây.
\item ...
\end{itemize}
```

- Đánh số (numering) là kiểu liệt kê có thứ tự. Để sử dụng kiểu liệt kê đánh số, chúng ta khai báo như sau

```
\begin{enumerate}
\item Nội dung thứ nhất được viết ở đây.
\item Nội dung thứ hai được viết ở đây.
\item ...
\end{enumerate}
```

**Figure A.1:** Internet vạn vật

\end{enumerate}

Chú ý các nội dung trình bày trong cả hai môi trường liệt kê theo sau lệnh `\item`. Ngoài ra LaTeX còn cung cấp một số kiểu liệt kê khác, sinh viên có thể tham khảo tại <https://www.overleaf.com/learn/latex/Lists>

#### A.4 Table insertion

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

**Table A.1:** Table to test captions and labels.

Bảng A.1 là ví dụ về cách tạo bảng. Tất cả các bảng biểu phải được đề cập đến trong phần nội dung và phải được phân tích và bình luận. Chú ý: Tạo bảng trong LaTeX khá phức tạp và mất thời gian, vì vậy sinh viên có thể sử dụng các công cụ hỗ trợ tạo bảng (Ví dụ: <https://www.tablesgenerator.com/>). Sinh viên có thể tìm hiểu sâu hơn về cách chèn ảnh trong LaTeX tại link <https://www.overleaf.com/learn/latex/Tables>.

#### A.5 Figure Insertion

Hình A.1 là ví dụ về cách chèn ảnh. Lưu ý chú thích của hình vẽ được đặt ngay dưới hình vẽ. Sinh viên có thể tìm hiểu sâu hơn về cách chèn ảnh trong LaTeX tại [https://www.overleaf.com/learn/latex/Inserting\\_Images](https://www.overleaf.com/learn/latex/Inserting_Images).

Chú ý, tất cả các hình vẽ phải được đề cập đến trong phần nội dung và phải được

phân tích và bình luận.

## A.6 Reference

### Listing

Áp dụng cách liệt kê theo quy định của IEEE. Ví dụ của việc trích dẫn như sau **scott2013sdn**. Cụ thể, sinh viên sử dụng lệnh `\cite{}` như sau **ashton2009internet**. Chỉ những tài liệu được trích dẫn thì mới xuất hiện trong phần Tài liệu tham khảo. Tài liệu tham khảo cần có nguồn gốc rõ ràng và phải từ nguồn đáng tin cậy. Hạn chế trích dẫn tài liệu tham khảo từ các website, từ wikipedia.

### Types of Reference

Các nguồn tài liệu tham khảo chính là sách, bài báo trong các tạp chí, bài báo trong các hội nghị khoa học và các tài liệu tham khảo khác trên internet.

## A.7 Equations

Các gói `amsmath`, `amssymb`, `amsfonts` hỗ trợ viết phương trình/công thức toán học đã được bổ sung sẵn ở phần đầu của file `main.tex`. Một ví dụ về tạo phương trình (A.1) như sau

$$F(x) = \int_b^a \frac{1}{3}x^3 \quad (\text{A.1})$$

Phương trình A.1 là ví dụ về phương trình tích phân. Một phương trình khác không được đánh số thứ tự (gán nhãn)

$$x[t_n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[f_k] e^{j2\pi nk/N}$$

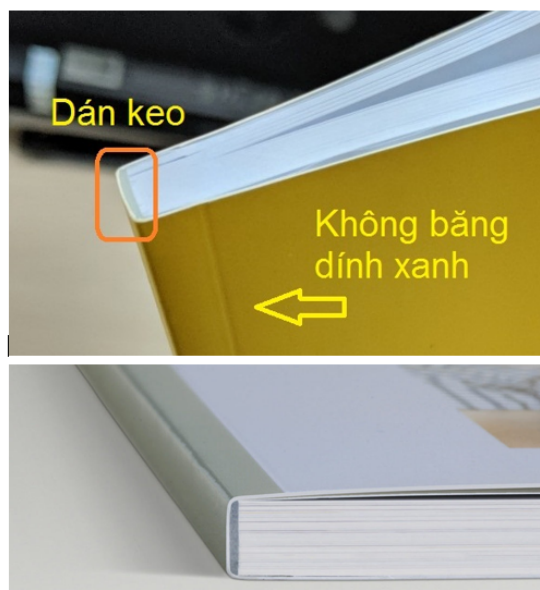
Phương trình này thể hiện phép biến đổi Fourier rời rạc ngược (IDFT).

## A.8 Qui cách đóng quyển

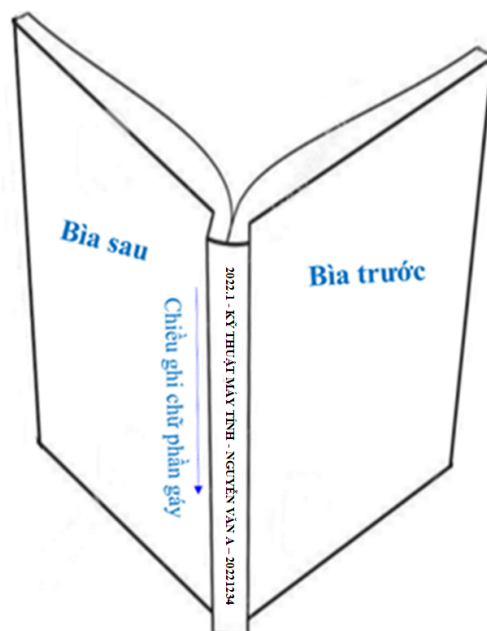
Phần bìa trước chế bản theo qui định; bìa trước và bìa sau là giấy liền khổ. Sử dụng keo nhiệt để dán gáy khi đóng quyển thay vì sử dụng băng dính và dập ghim như mô tả ở imgs A.3 Phần gáy ĐATN cần ghi các thông tin tóm tắt sau: Kỳ làm ĐATN - Ngành đào tạo - Họ và tên sinh viên - Mã số sinh viên. Ví dụ:

2022.1 - KỸ THUẬT MÁY TÍNH - NGUYỄN VĂN A - 20221234

Qui cách ghi chữ phần gáy như hình dưới đây:



**Figure A.2:** Qui cách đóng quyển đồ án



**Figure A.3:** Qui cách đóng quyển đồ án



## **B. USE CASE DESCRIPTIONS**

Nếu trong nội dung chính không đủ không gian cho các use case khác (ngoài các use case nghiệp vụ chính) thì đặc tả thêm cho các use case đó ở đây.

### **B.1 Đặc tả use case “Thống kê tình hình mượn sách”**

...

### **B.2 Đặc tả use case “Đăng ký làm thẻ mượn”**

...