

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY  
UNIVERSITY OF TECHNOLOGY  
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



## SOFTWARE ENGINEERING (CO3001)

---

# SSPS SOFTWARE REPORT

### Part 3

---

#### GROUP CC03-05

**Students**    Nguyễn Hồ Đức An - 2252009  
                  Nguyễn Đức Quốc Kỳ - 2252415  
                  Nguyễn Châu Hoàng Long - 2252444  
                  Nguyễn Gia Phúc - 2252637  
                  Nguyễn Anh Quân - 2252678

**Lecturer:**    Dr. Trương Tuấn Anh

HO CHI MINH CITY, SEPTEMBER 2024

## Contents

<b>1</b>	<b>Task descriptions</b>	<b>2</b>
<b>2</b>	<b>Architectural Design</b>	<b>3</b>
<b>3</b>	<b>Presentation strategy</b>	<b>5</b>
<b>4</b>	<b>Database approach</b>	<b>6</b>
<b>5</b>	<b>API Management</b>	<b>7</b>
<b>6</b>	<b>Component Diagram</b>	<b>8</b>



## 1 Task descriptions

**3.1:** Use a layered architecture to design the HCMUT-SSPS system. Describe how will you present your User Interface. Describe how will you store your data. Describe how you will access to external services/APIs.

**3.2:** Draw a component diagram for an important module (the same with the module used in task 2.1)

## 2 Architectural Design

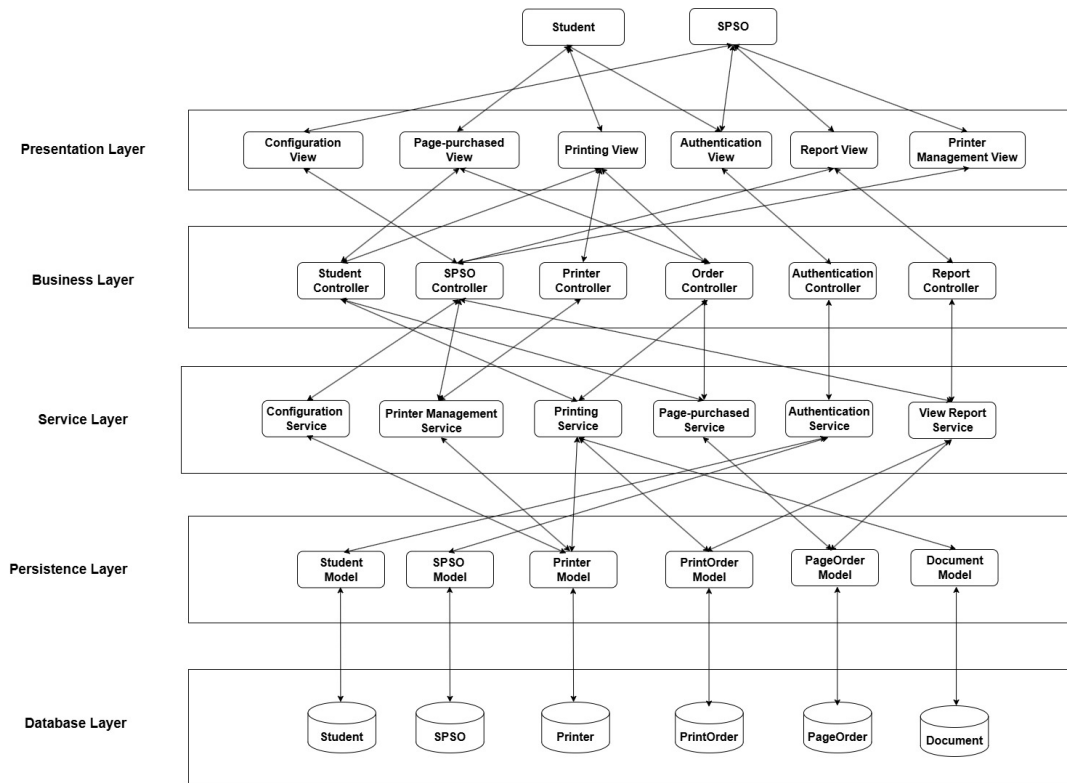


Figure 1: Box-Line Diagram

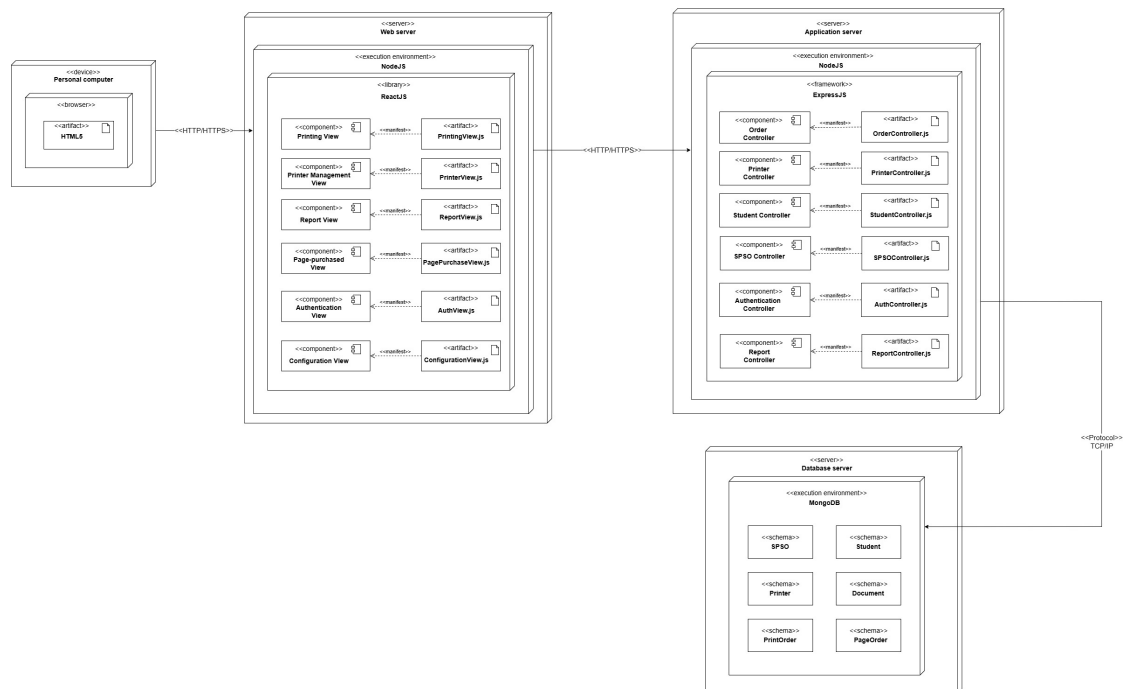


Figure 2: Deployment Diagram

### 3 Presentation strategy

This is the first layer in the architecture. We focus on simplicity, ease of use, ensuring that users, whether students or staff, can quickly access the system's functionalities without steep learning curves. To achieve this, we've selected technologies and design principles that prioritize both visual appeal and technical efficiency:

- **Front-end library and framework:** We use front-end user interface development technologies like React. React's component-based structure enables us to develop adaptable, reusable user interface components that maintain high performance and responsiveness.
- **Responsive Design:** Ensuring compatibility with all devices currently used by students and staff. We will integrate with various devices and screen sizes. Flexible forms and interface components will be implemented to provide a good user experience on computers, mobile phones, and tablets.
- **User-Friendly Features:** We consider visual elements such as buttons, forms, and easy-to-use menus. The aim is to allow first-time users to access the system effortlessly. Each interaction is designed to be straightforward, with minimal steps to accomplish tasks, ensuring that the learning curve remains low while the user experience remains high.

By using React for the user interface, along with a flexible design and a focus on user experience, we will create an impressive interface for the HCMUT\_SPSS system. This approach not only enhances the functional aspects of the system but also ensures a consistent, enjoyable experience for all users.

## 4 Database approach

In layered architecture, the database will be the bottom layer, responsible for storing and managing all data. Application data will be stored here, and operations such as search, insert, update, and delete will be performed frequently to interact with the data through the database management system. For the HCMUT\_SPSS, the team will use layered architecture with the database layer stored in a NoSQL database, specifically MongoDB, which means that application data will be stored as documents in collections without the need for tables and complex relationships as in relational databases.

In the HCMUT\_SPSS system, we will create entities as documents with flexible structures, allowing for scalable and adaptable data structures according to the needs of the application. Here are the required entities:

- Student: Save student's information such as Student ID, Name (First Name and Last Name), Password, email, remaining page, last logged in session.
- SPSO: Save information such as SPSO ID, Name, password, email, phone number, last login.
- Document: Including Document ID, Name, Format, Number of Pages.
- Order: Include information such as: Order ID, printing configuration (Number of copy, Paper Size, Type face, Vector), Start Date, Completed Date.
- Page order: Including ID purchase, Price, Number of pages, Date Purchase
- Printer: Including information such as: Printer code, Brand, Model, Description, Location (Campus, Building, Room), Status.

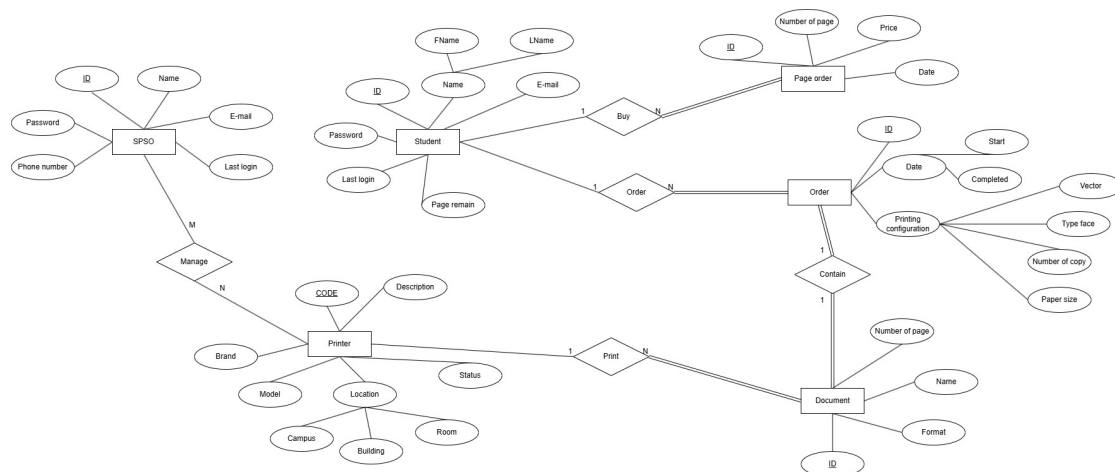


Figure 3: ER Diagram

## 5 API Management

An API, or Application Programming Interface, is a set of rules and protocols that allows different software applications to communicate with each other. It defines how data and functionality can be exchanged between these applications. This enables developers to integrate existing services and data into their own applications, saving time and effort.

In software engineering, APIs are widely used to build complex systems and web applications. They facilitate the creation of microservices architectures, where different components of an application can be developed and scaled independently. Additionally, APIs power many popular web services and platforms, allowing developers to access and utilize their functionality.

Effective API management is crucial for several reasons. It ensures that APIs are reliable, secure, and performant. Proper management involves creating clear documentation, setting access controls, monitoring usage, and implementing security measures. By managing APIs effectively, organizations can improve their software's efficiency, scalability, and user experience.

For the HCMUT-SPSS, there are several crucial APIs listed as below:

- Authentication API: Ensuring security as well as using the school account for tracking personal page quota, printer configuration, etc.. through HCMUT-SSO.
- Payment API: Providing a mean to purchase the services of the system with BKPay.
- Printer managing APIs: helping the SPSO to manage the devices efficiently with viewing users' activity with individual printer, checking the properties of the printers.
- Printer manipulating APIs: including APIs for adding, deleting, enable or disable printers or updating the driver
- Printer configuring API: for the purpose of convenience and personalization, each student can have their own configuration for the printers.
- Printer controlling API: with this, student can interact with the printers, uploading the documents and begin to print their materials.



## 6 Component Diagram

The Printer Management component diagram for the printing service consists of three main components: Printer Management, Controller, and Database. The Printer Management component serves as the main interface for admins, allowing them to manage printer operations. The Controller component includes several utility modules that handle specific tasks, such as viewing logs, updating printer drivers, displaying printer information, changing printer configurations, enabling or disabling printers, and adding or removing printers. These modules interact with the Database to retrieve necessary data via the Logs and Printer API interfaces. The Database component stores essential information in the User and Printer subcomponents, which supply user info and printer info as needed. The Logs subcomponent keeps track of activity records, which the Controller references for logging and status updates. Together, these components ensure efficient management and monitoring of printer services by admins.

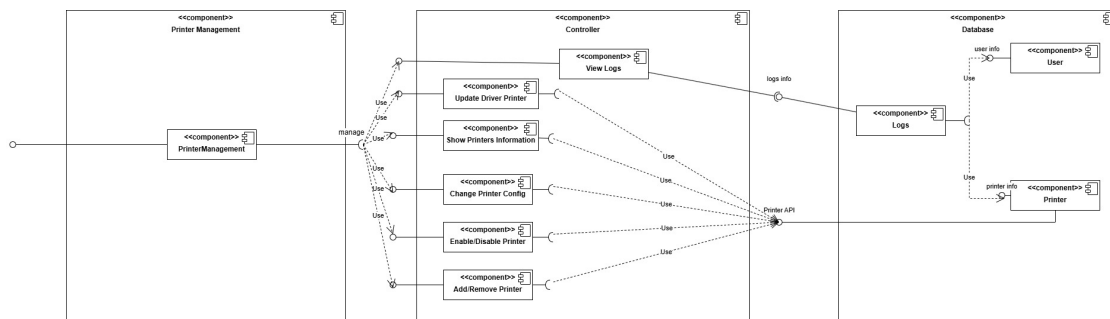


Figure 4: Component Diagram