**Software Requirements Specification**

**for**

**A SMART PRINTING SERVICE FOR STUDENTS AT HCMUT**

**Version 1.0 approved**

**Prepared by:**

1. **Tran Chi Tai – 2252727**
2. **Phan Tran Y Nhi – 2252583**
3. **Phan Chi Vy – 2252938**
4. **Tran Dinh Dang Khoa – 2211649**
5. **Tran Nguyen Anh Khoi – 2252383**

**Department of Software Engineering**

**Faculty of Computer Science and Engineering**

**Ho Chi Minh City University of Technology – VNU-HCM**

**09/22/2024**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# **Task 1: Requirement elicitation (1.1, 1.2)**

## **Domain Context**

The Student Smart Printing Service (HCMUT\_SSPS) operates within the domain of university printing services at Ho Chi Minh City University of Technology (HCMUT). This system is designed to streamline document printing for students in the university.

Users can upload documents, select from available printers, and configure printing options such as paper size, page count, and number of copies. The system ensures that students can print efficiently while tracking their usage through comprehensive logs that include student IDs, printer IDs, and timestamps for each print job.

The service includes a page balance management feature, providing each student with a default number of A4 pages each semester and allowing them to purchase additional pages through an integrated online payment system like BKPay. Administrators, known as the Student Printing Service Officers (SPSOs), have the capability to manage printer settings, adjust page limits, and configure permitted file types.

Automated reports on printing activity are generated at the end of each month and year, giving the SPSO insights into usage trends. All users must authenticate through the HCMUT\_SSO service to ensure secure access to the system. The Student Smart Printing Service is available via both web and mobile applications.

## **Stakeholders and Needs**

**Students (Users)**

**Primary users** of the printing service.

They upload documents, select printing options, and manage their printing balance.

They can view their personal printing logs and summaries.

They can purchase additional printing pages via the system.

**Student Printing Service Officer (SPSO)**

**Administrators** of the system.

Responsible for managing printer configurations (adding, enabling, disabling printers).

Configures system settings like default pages, permitted file types, and printing limits.

Monitors printing activity logs and generates reports.

Manages the default number of pages provided to students each semester.

**HCMUT IT Department / System Administrators**

Manages the **technical infrastructure** of the system.

Ensures that the system is integrated with the university’s **SSO authentication service**.

Responsible for system updates, maintenance, and performance optimization.

**University Administration (Policy Makers)**

Oversees the allocation of resources and budget for the printing service.

Decides on policies like the default number of pages provided to students each semester.

Reviews system-generated reports for operational insights.

**Payment Service Provider (e.g., BKPay)**

Provides the **payment gateway** for students to purchase additional printing pages.

Manages payment transactions and ensures smooth integration with the university’s financial systems.

## **Benefits of the System**

Students: Can print their documents more comfortably through their phone or laptop with flexible timing on their ends. Further quality of life features like easy paying or usage tracking for their own profile.

HCMUT IT Department / System Administrators: The system helps keep track of the status of multiple printers at the same time so that it is easier to notice bugs and malfunctions for system fixing updates.

Student Printing Service Officers (SPSO): Easier to keep track of the printers’ activity as well as managing them through the system, reducing manual tasks.

University Administration (Policy Makers):: Automated reports provides insight on students usage and resource management, allows for further budgeting and optimization on the system

Payment service providers (BKPay, ZaloPay, MoMo, ...): Have more transaction opportunities from students buying pages for printing.

## **Functional Requirements**

**1.4.1. Students (Users)**

Students are the primary users of the system. They need features to upload documents, select printers, manage their balances, and view logs. The functional requirements for students are:

1. Students can **upload document files** (limited to specific file types) for printing through the web or mobile app.
2. Students can **select a printer** and configure print settings (e.g., number of copies, paper size, one-sided or double-sided).
3. Students can view **their personal printing logs** including the printer ID, file name, and time of printing.
4. Students can confirm/cancel the print settings before completing the printing process.
5. Students can **purchase additional printing pages** through the integrated payment system BKPay.

**1.4.2. Student Printing Service Officer (SPSO)**

SPSO administrators need to manage printers, monitor printing activity, and generate reports. The functional requirements for SPSO are:

1. SPSO can **add, enable, or disable printers** in the system.
2. SPSO can **set the permitted file types** that students can upload for printing (e.g., PDF, DOCX).
3. SPSO can **view printing logs** for all students, including detailed information such as student ID, printer ID, and time of printing.
4. SPSO can **adjust the default number of print pages** assigned to students per semester.
5. The system shall **automatically generate monthly and yearly reports** summarizing printing usage across all printers and students.

**1.4.3. HCMUT IT Department / System Administrators**

System administrators are responsible for the technical infrastructure and ensuring system performance. The functional requirements for IT/System Administrators are:

1. The system must **integrate with the HCMUT SSO** for user authentication, ensuring only authorized students and staff can access the system.
2. The system shall **monitor printer statuses** and report any malfunctions or errors to the IT department for timely maintenance.
3. IT administrators can **perform system updates** (e.g., adding new printers or updating features) without disrupting the ongoing printing services.
4. The system must **store all printing logs and reports securely**, ensuring data privacy and integrity.
5. The system must **manage user access rights**, allowing different levels of access for students, SPSO, and IT staff.

**1.4.4. University Administration (Policy Makers)**

The university administration needs insights from reports to allocate resources and manage budgets. The functional requirements for the university administration are:

1. The system shall **automatically generate usage reports**, summarizing printing activities and associated costs on a monthly and yearly basis.
2. University administrators can **access these reports** through the system dashboard for budgeting and resource planning purposes.
3. Administrators can **set printing policies**, such as adjusting the default number of free pages students receive each semester.
4. The system must allow administrators to **track the efficiency of printing resources**, such as printer usage statistics, to improve cost optimization.
5. The system shall **maintain a log of all report accesses** by the administration for auditing purposes.

**1.4.5. Payment Service Providers (e.g., BKPay, ZaloPay, MoMo)**

Payment providers handle transactions for students purchasing additional printing pages. The functional requirements for payment service providers are:

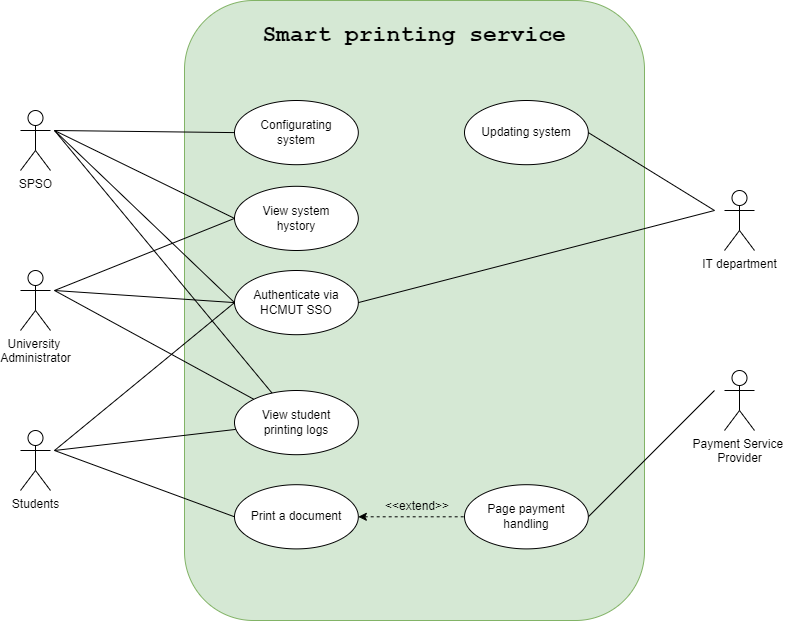
1. The system must **integrate with payment service providers** like BKPay, allowing students to securely purchase additional printing pages.
2. The system must **log each transaction**, including the amount paid, student ID, and time of transaction.
3. The payment system shall **notify the student and the SPSO** of successful transactions, automatically updating the student's page balance.
4. Payment service providers must **process refunds** in case of errors or canceled transactions.
5. The system shall allow students to **choose from multiple payment options**, ensuring convenience and flexibility.

## **Non- Functional Requirements**

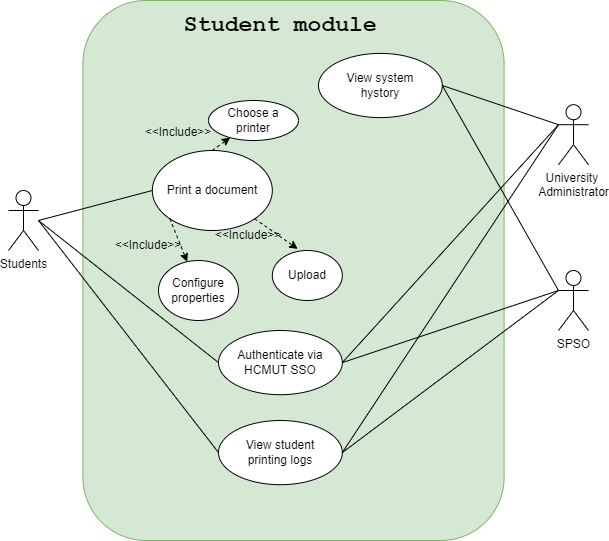
1. The system shall be available to all users from 7:00 AM to 5:00 PM, Monday to Saturday.
2. All users shall be authenticated by the HCMUT\_SSO service before using the system.
3. The system shall be interoperable with roughly 80% of printers currently used at the university.
4. The system shall be able to proceed approximately 500 users accessing the service at the same time without overloading.
5. The system shall allow users to upload up to 1GB of documents at a time.
6. The system shall be accessed through a website on Google and be provided on Android(8.1 and above) and iOS(8 and above) devices.
7. The users shall be able to understand and use the basic functions of the system by reading the manual or instructions provided by the system developers.
8. The printing log saved in the system shall be in read-only mode.

# **Use-case Diagrams (1.3)**

## **Use-case Diagram for the Whole System**



## **Use-case Diagram for Student Module**



## **The Details of Usecases in Students Module**

1. **Usecase Print a document**

| Name: UC-Print a document |
| --- |
| Created by: Tran Nguyen Anh Khoi Date created: 9/30/2024 |
| Primary Actors: Student Secondary Actors: None |
| Description: The student uploads the document file onto the system and choose a printer. The system then offers the student a list of printing properties, including paper size, pages to be printed, one/double-sided and number of copies, which they can either specify to their demand. |
| Trigger: After user is authenticated to the printing system |
| Preconditions: PRE-1. User’s identity is authenticated.  PRE-2. The system is connected to at least 1 printer. |
| Postconditions: POST-1. The printer receives the data to be printed |
| Normal Flow: 1. Student chooses the option to upload the documents for printing.   1. Student uploads the document files to the system 2. System offers the list of printers connected to the system, the choices of paper size, pages to be printed, one/double-sided and number of copies. 3. Students select the printer, paper size, pages to be printed, one/double-sided and number of copies through the options provided by the system. 4. Students choose the Print button in the interface to complete the setup and start printing. |
| Exceptions: **E1 There are no documents in the system.** |

1. **Usecase Authenticate via HCMUT SSO**

| Name: UC- Authenticate via HCMUT SSO |
| --- |
| Created by: Tran Dinh Dang Khoa Date created: 9/30/2024 |
| Primary Actors: Student, Stakeholders and Interests Secondary Actors: None |
| Description: Student: Wants to securely authenticate and access printing services.  University Administrator: Requires students to use their verified accounts for access. |
| Trigger: The student needs to access the printing system but is not yet logged in. This prompts the system to initiate the authentication process via HCMUT SSO. |
| Preconditions: The student is not logged in to the system.  The HCMUT SSO system is available. |
| Postconditions: The student is authenticated and gains access to their account and printing system. |
| Normal Flow: 1. The student navigates to the login page of the printing system.  2. The system redirects the student to the HCMUT SSO login page.  3. The student enters valid credentials on the SSO login page.  4. The system validates the credentials with the HCMUT SSO service.  5. The system successfully authenticates the student and redirects them back to the printing system dashboard.  6. The student is now able to access their printing logs and manage print jobs. |
| Exceptions: **E1: Invalid Credentials**  1. The student enters incorrect credentials.  2. The HCMUT SSO system displays an "Invalid Credentials" error message.  3. The student is prompted to re-enter the credentials.    **E2: SSO Service Unavailable**  1. The student attempts to log in, but the SSO service is down.  2. The system displays a message: "SSO service is currently unavailable, please try again later.".  3. The student is unable to log in until the SSO service is restored. |

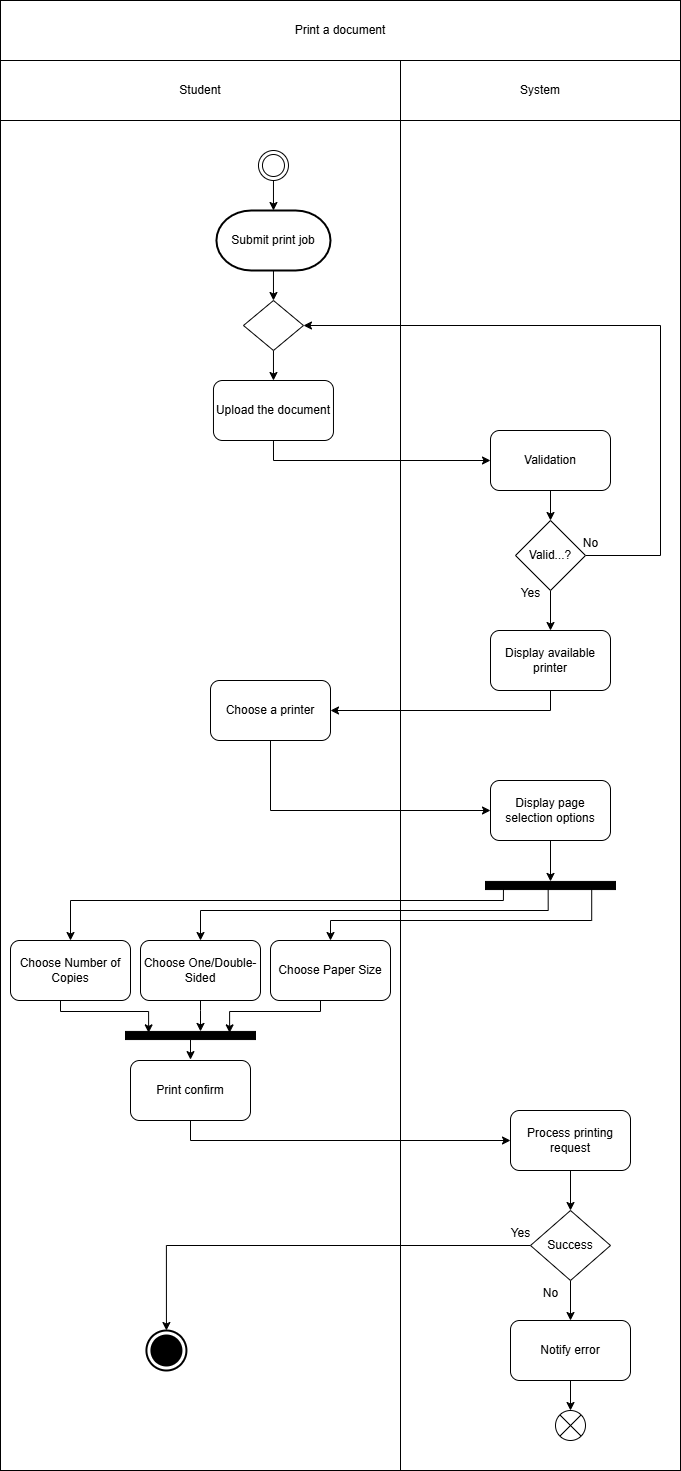
1. **Usecase View student printing logs**

| Name: UC-5 View Student Printing Logs |
| --- |
| Created by: Nhi Date created: 30/9/2024 |
| Primary Actors: Student, SPSO Secondary Actors: None |
| Description: Students can view their own printing logs for a specific period, and the SPSO can view printing logs of all students for any time and any printer. |
| Trigger: The Student or SPSO clicks the “Search” button in the “View Printing Logs” section. |
| Preconditions: PRE-1. The user must be authenticated via HCMUT\_SSO.  PRE-2. The user must have appropriate access. |
| Postconditions: POST-1. The Student or SPSO can view printing logs based on their access level and selected filters (time, printer). |
| Normal Flow: **5.0 View Student Printing Logs**  1. The Student or SPSO logs into the system.  2. The user navigates to the “View Printing Logs” section.  3. The user selects the date range and, optionally, specific printer(s) to filter logs. SPSO can additionally filter by student ID.  4. The system retrieves and displays the printing logs based on the selected filters.  5. For students, only their own logs are shown. For SPSO, all relevant logs are displayed based on the filters.  6. Students see a summary of printed pages by page size, while SPSO has access to detailed logs for all students and printers.  Alternative Flows: **5.1 No Logs Found for the Selected Filters**  1. The user selects a date range, printers, or student ID (for SPSO) that returns no results.  2. The system displays a message stating that no logs were found for the selected criteria and suggests adjusting the filters.  3. The user modifies the filters or broadens the search criteria.  **5.2 No Filters Applied**  1. The user or SPSO did not enter any filters (e.g., date range, printers, or student ID).  2. The system displays all available printing logs.  For students, the system shows all of their own printing logs.  For SPSO, the system displays all available logs across all students and printers.  3. The user reviews the results and can choose to apply filters if needed for further refinement. |
| Exceptions: **5.1.E1 The student attempts to access logs other than their own** |

# **3. System modelling**

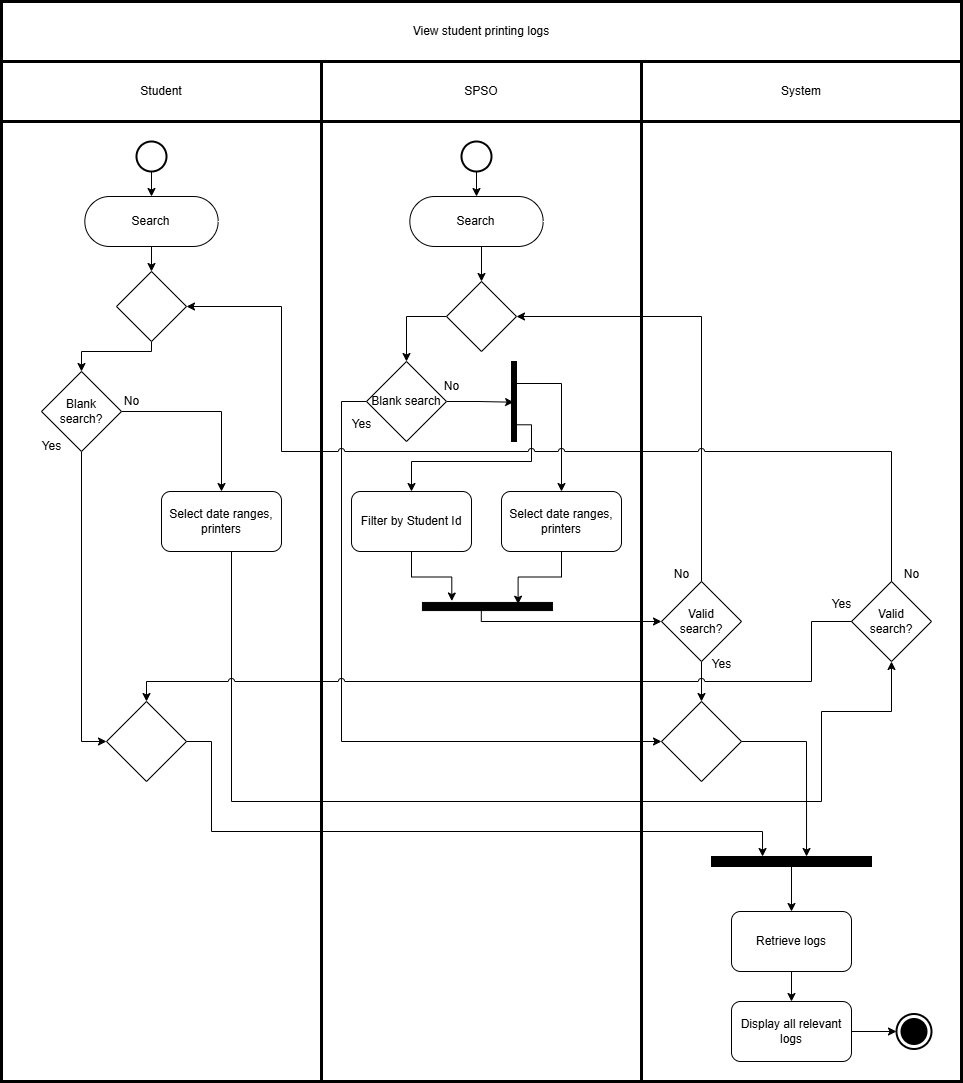
## **3.1 Activity Diagrams for Key Processes**

**3.1.1 Print a Document**

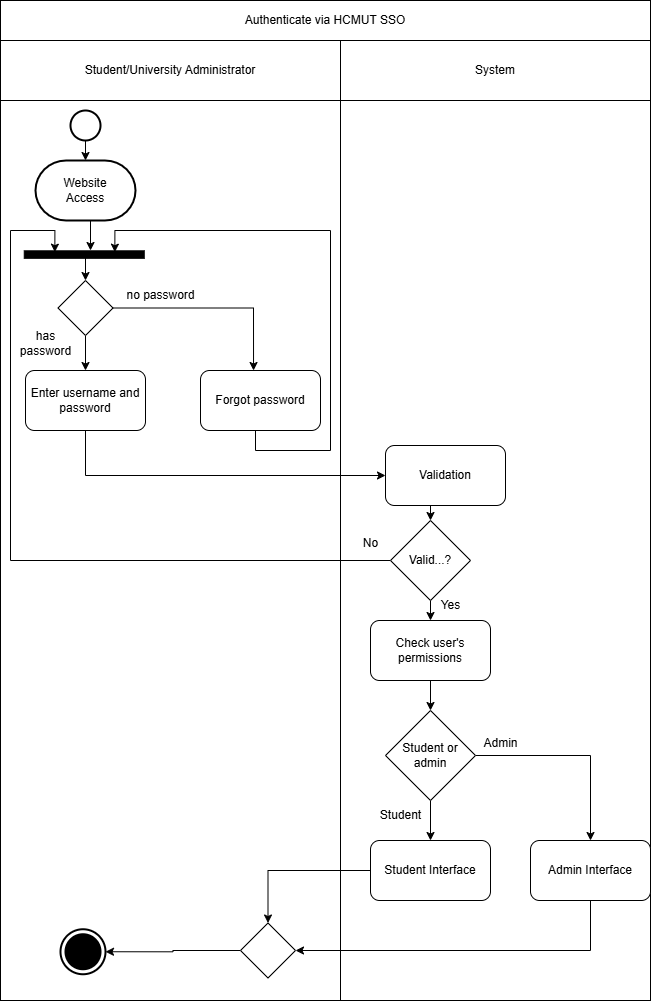


This activity diagram represents the process where a student submits a print job using the Student Smart Printing Service. The actor involved is the student. The process starts when the student uploads a document, which the system validates. If the document is invalid, the system prompts the student to upload it again. Once validated, the system shows the available printers for the student to select from. The student configures print settings like the number of copies, paper size, and whether to print single- or double-sided. After confirming the settings, the system processes the print request. If the request is processed successfully, the print job is executed; otherwise, the system notifies the student of an error.

**3.1.2 View Student Printing Logs**

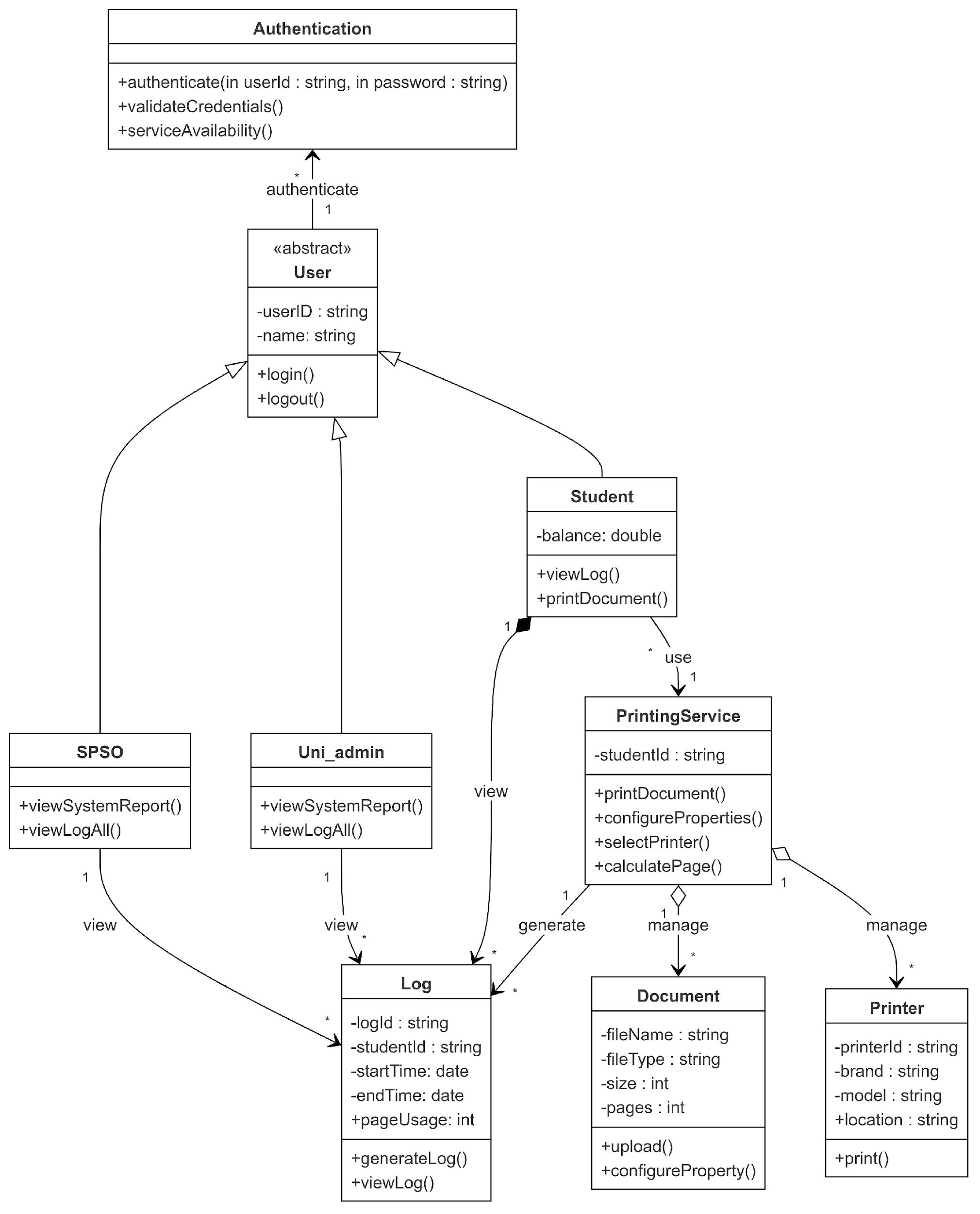


This activity diagram depicts the process of viewing printing logs in the Student Smart Printing Service. Two actors are involved: the student and the SPSO (Student Printing Service Officer). If a blank search (no filters applied) is performed, the system displays all results. If specific filters (e.g., date ranges, printers) are applied, the system retrieves and displays the relevant logs. The SPSO can also filter by student ID to view logs for a particular student. If no logs are found or the search is invalid, the user is prompted to re-enter the search criteria.

**3.1.3 Authenticate via HCMUT SSO**

This activity diagram shows the authentication process via HCMUT Single Sign-On (SSO). The actors involved are the student and the University Administrator. Both access the website to log in. If the user doesn't have a password, they select the "Forgot Password" option to reset it. After entering the username and password, the system validates the credentials. If validation fails, the user must retry. Upon successful validation, the system checks if the account belongs to a student or an administrator. Depending on the user type, the system presents either the student interface or the admin interface.

## **3.2 Class Diagram for the Student Smart Printing Module**



This class diagram illustrates two main actions students can perform:

Print a Document:

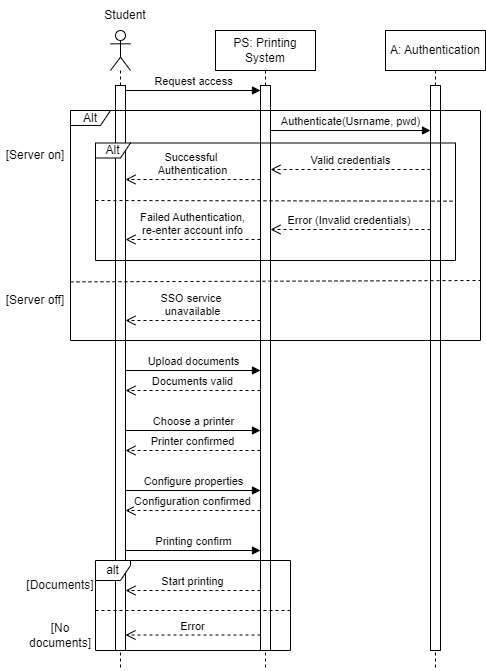
A student submits a print job through the printDocument() function. The PrintingService processes the request, manages the printer, and applies configuration changes made by the student (e.g., paper size, number of copies). After confirming the print request, a log is generated containing details like the student ID, page usage, printer start time, and end time.

View Printing Logs:

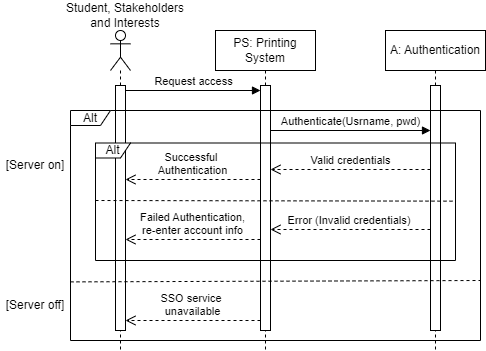
A student can view their printing history using the viewLog() function. The PrintingService retrieves logs associated with the student's ID. The SPSO and University Administrator can view logs for all students, while individual students only have access to their own logs.

## **3.3 Sequence Diagram for Print a Document, Authenticate via HCMUT SSO, and View Printing Logs**

**3.3.1 Print a Document**

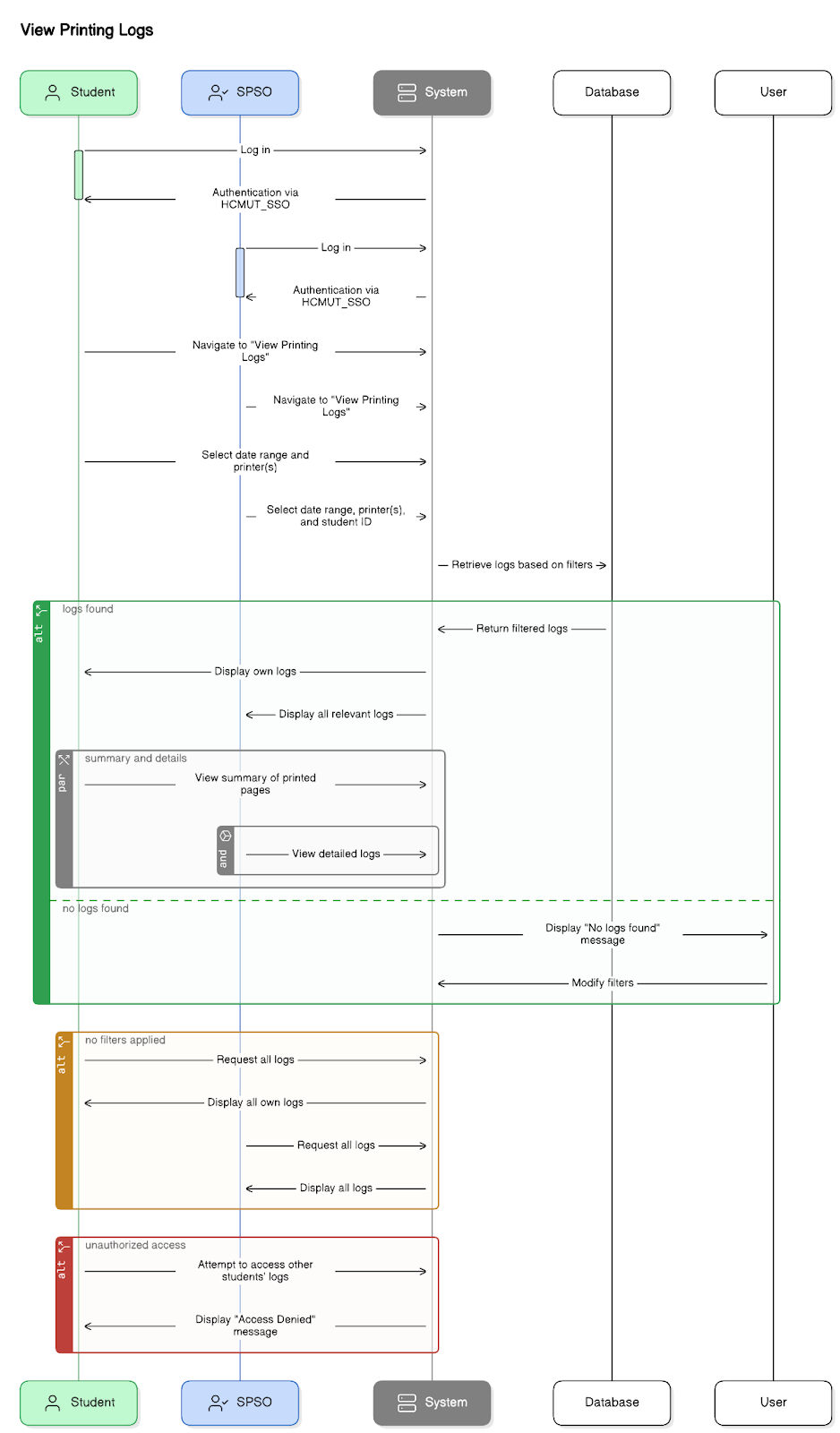


The sequence diagram illustrates how the printing process works for a student. The student first authenticates via the HCMUT SSO service. Once authenticated, the student can upload documents and select from available printers. After configuring printing properties (e.g., paper size, single- or double-sided printing, number of copies), the system starts printing if the document is loaded; otherwise, an error is generated.

**3.3.2 Authenticate via HCMUT SSO**

The sequence diagram shows how a user gains access to the printing system using HCMUT SSO. The user must provide valid credentials (username and password). If the credentials are correct, the user is authorized to access system functions. Invalid credentials trigger an error, prompting the user to re-enter the information.

**3.3.3 View Printing Logs**



This diagram shows the detailed process for viewing printing logs, involving interactions between the student, SPSO, system, and database. The system filters logs based on criteria such as date range, printer, and student ID. Logs are displayed as either a summary or detailed view. Unauthorized access attempts are blocked, ensuring that only the relevant user or authorized personnel (SPSO) can view specific logs.

## **3.4 Develop MVP 1: User Interfaces for a Desktop-View Central Dashboard**

**3.4.1 Introduction**

MVP 1 focuses on building the user interface for the HCMUT Smart Printing Service dashboard, used by students, SPSO, and university administrators. It covers essential tasks like printing document submission, log retrieval, and authentication via HCMUT SSO.

**3.4.2 Key Features**

HCMUT SSO Authentication: Students, SPSO, and admins log in via HCMUT SSO, with role-based access.

Document Upload and Print Configuration: Students can upload documents, choose printers, and configure settings (e.g., paper size, number of copies).

View Print Logs: Both students and SPSO can view and filter print logs (date range, printers, student ID).

Navigation: Simple top navigation bar and main content area for easy access to functions.

**3.4.3 User Interface Design**

Top Navigation Bar: Quick access to "Upload Document," "View Logs," etc.

Main Content Area: Displays features like document upload, log filters, and results.

Design Principles: Focus on simplicity and user-friendliness. Based on the [**Figma prototype**](https://www.figma.com/proto/i5VVQU95f0NkWhQcD1Cg2L/HCMUT-SPS?node-id=30-2244&starting-point-node-id=30%3A2244).

**3.4.4 User Experience**

Ease of Use: Designed for quick, intuitive access to printing services.

Error Handling: Clear messages for invalid uploads or incorrect credentials.

Real-Time Feedback: Notifications on successful document upload or empty log results.

**3.4.5 Future Enhancements**

Mobile Support: Expand interface for mobile use.

Advanced Filtering: More detailed search criteria for print logs.

This MVP ensures a streamlined, user-friendly dashboard for the HCMUT Smart Printing Service.

**4.1 Architecture design**

**4.1.1 Layered architecture**

The layered architecture pattern, also known as the n-tier architecture pattern, is one of the most commonly used designs. It is the default standard for many Java EE applications, making it highly familiar to architects, designers, and developers alike. This pattern aligns well with the traditional communication flows and organizational structures found in most companies, which makes it a natural fit for business application development.

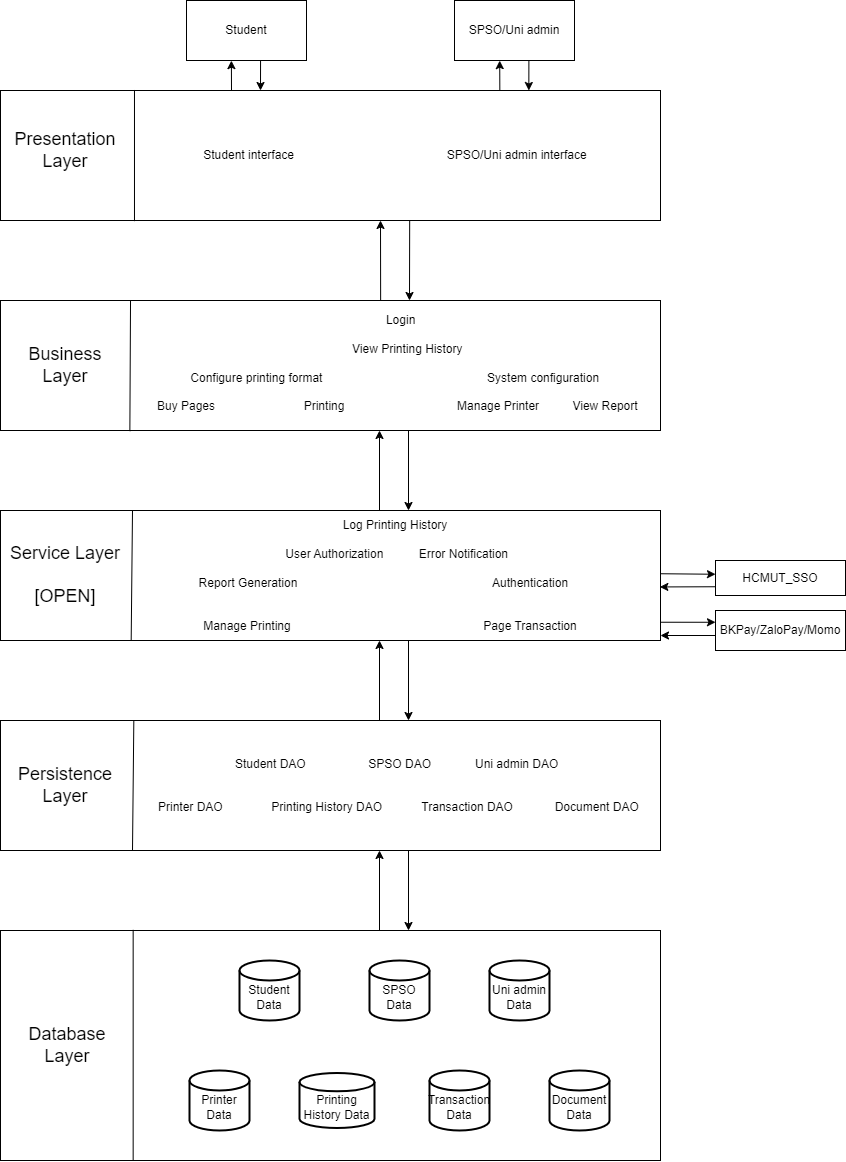
We choose this pattern because it is widely recognized and relatively straightforward to implement. In the layered architecture pattern, components are arranged into horizontal layers, each dedicated to a specific function within the application. For instance, the presentation layer manages user interface and browser interactions, while the business layer handles the business logic needed to process requests. With the separation of concerns among components, it simplifies the development, testing and maintenance process. The isolation from the layered architecture also allows us to make changes in one layer without affecting the other layer.

However, this structure also has its own disadvantage:

- Rigid structure: The strict layer hierarchy can be limiting, especially if the application needs to bypass certain layers for efficiency or services that frequently change dynamically. Since our printing system is a simple system and only focuses on providing printing utility to the students and the university staff, this problem won’t affect as much as high scalability applications.

- High overhead: While it is true some layered architectures can perform well, the pattern does not lend itself to high-performance applications due to the inefficiencies of having to go through multiple layers of the architecture to fulfill a business request especially for more simple requests that do not add any values or performing operations at every layer. Because HCMUT\_SSPS size and scale is not very large so we can accept this disadvantage

So, even though there are disadvantages to consider, its benefit outweighs the impact caused by the shortcomings



The system is being used by two main types of users: students and the university staff (SPSO, Uni admin). So, the presentation layer respectively have two interfaces one for the students and one for the staff.

The students main functionalities are Log in (authenticated by HCMUT\_SSO in the service layer), View Printing History (read data from the Database layer), Configure printing format of their documents, Buying page for printing (through payment service providers in the service layer) and Printing itself. The printing will access the manage printing service and interact with the persistence layer for read/write data purposes.

For the SPSO/Uni admin functionalities, there are Log in, View students printing log, View annual reports, System configuration, and Managing the printers. Some functions have similar working with the students. Other notable functions like viewing the reports will access the report generation service and read corresponding data from the database. Another functionality, printers management will access the Manage Printing service. This service will read printers attributes and allow editing their attributes and statuses from the database layer through the persistence layer.

In addition, the service layer in our diagram is marked as open, meaning requests are allowed to bypass this open layer and go directly to the layer below it. This makes sense because some functions like View Printing History don’t require additional processing from the service layer and benefit from direct access to the persistence layer for faster, simpler data retrieval.

**4.1.2 Presentation strategy**

This is the foundational layer in the system architecture. We will prioritize simplicity, ease of use, and user experience. To achieve this, we are leveraging modern and specific technologies as follows:

* Front-end Library and Framework: For designing and developing the user interface, we are using Figma for initial layout and prototyping, paired with HTML for building the actual interface. This combination allows us to create user-friendly, visually consistent, and functional designs.
* Responsive Design: Our design will adapt seamlessly to various devices used by students, staff, and faculty. We will integrate compatibility across a range of devices and screen sizes, ensuring a smooth user experience on desktops, mobile phones, and tablets. The interface will incorporate flexible components and forms to enhance accessibility and usability.
* User-Friendly Features: We are paying careful attention to intuitive elements, including buttons, forms, and easily navigable menus. Our goal is to ensure that even first-time users can navigate the system effortlessly.

With Figma as our prototyping tool and HTML for interface development, alongside a flexible, user-centered design approach, we aim to create an impressive, well-integrated interface for the HCMUT\_SPSS system that fits seamlessly within the system’s architectural layers.

**4.1.3 Data Storage Approach**

The data storage approach for the HCMUT Smart Printing Service will utilize MySQL as the database management system, designed to efficiently manage a variety of data types, including user profiles, document uploads, print logs, printer configurations, and transaction records. The key components of this approach are as follows:

**Database Structure:**

* The relational MySQL database is used to store structured data, such as user information, printer details, and transaction records. Tables are designed with relationships to optimize data retrieval and maintain data integrity.
* Indexing is applied to frequently queried fields, such as user IDs and printer IDs, to ensure quick access to frequently requested data, like individual user print logs or printer statuses.

**Document Storage:**

* Blob Storage (Binary Large Object) or a dedicated document storage service is used to securely store uploaded documents. This storage solution allows for handling various file formats and sizes while ensuring quick access for the printing process.
* Documents are stored with a unique identifier linking them to the corresponding user and print job in the MySQL database, allowing for efficient cross-referencing between document storage and user data.

**Data Security and Privacy:**

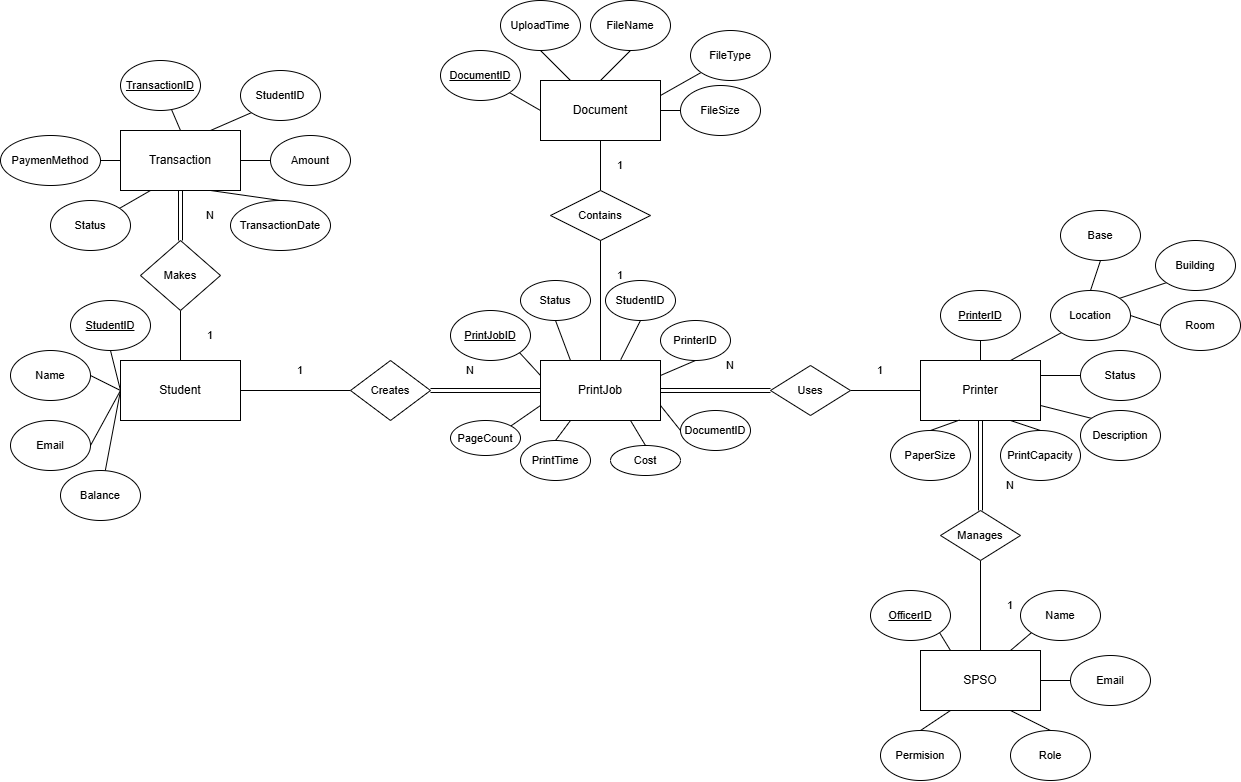
* All sensitive data, including user details and transaction records, is encrypted both at rest and in transit to ensure data confidentiality and integrity.
* Role-based access control is implemented to restrict access to specific data types. For example, only authorized Student Printing Service Officers (SPSOs) can view all print logs, while students can access only their own logs.

**Data Archiving and Retention:**

* Automated data archiving for historical print logs and transaction records is implemented to maintain system performance. Archived data can be retrieved for reporting and auditing purposes but is stored separately from active data.
* A data retention policy is enforced, automatically deleting data that exceeds the retention period unless flagged for long-term storage, helping optimize storage usage while adhering to data privacy requirements.

This data storage approach with MySQL ensures that the system is scalable, secure, and efficient, capable of handling high data volumes and providing quick, reliable access to information within the HCMUT Smart Printing Service.

The **Entity-Relationship Diagram (ERD)** for the **Student Printing Service** system, which is designed to manage and track various aspects of the printing process, including student print jobs, transactions, printer management, and document handling.

****

#### 1. Student Table

The **Student** table represents the students utilizing the printing service. It contains the following fields:

* **StudentID**: A unique identifier for each student (Primary Key).
* **Name**: The student's full name.
* **Email**: The student’s contact email.
* **Balance**: The balance available in the student's printing account.

The **StudentID** is referenced in other tables, establishing relationships with transactions and print jobs.

#### 2. Transaction Table

The **Transaction** table stores records of financial transactions made by students. Its key attributes include:

* **TransactionID**: A unique identifier for each transaction (Primary Key).
* **StudentID**: The student who made the transaction (Foreign Key referencing **Student**).
* **Amount**: The monetary value of the transaction.
* **PaymentMethod**: The payment method used (e.g., credit card, cash).
* **Status**: The current status of the transaction (e.g., completed, pending).
* **TransactionDate**: The date and time the transaction was completed.

Each student may have multiple transactions, establishing a one-to-many relationship with the **Student** table.

#### 3. Document Table

The **Document** table tracks documents uploaded by students for printing. It includes:

* **DocumentID**: A unique identifier for each document (Primary Key).
* **FileName**: The name of the uploaded document.
* **FileType**: The type of the document (e.g., PDF, Word).
* **FileSize**: The size of the document.
* **UploadTime**: The timestamp when the document was uploaded.

This table is connected to the **PrintJob** table through the **DocumentID**, as each print job corresponds to a specific document.

#### 4. Printer Table

The **Printer** table manages printer details, such as:

* **PrinterID**: A unique identifier for each printer (Primary Key).
* **Location\_Base**, **Location\_Building**, **Location\_Room**: The physical location of the printer.
* **Status**: The current operational status of the printer (e.g., online, offline).
* **Description**: A description of the printer’s specifications or features.
* **PaperSize**: Supported paper sizes (e.g., A4, Letter).
* **PrintCapacity**: The printer's monthly print capacity.

Each printer is referenced in the **PrintJob** table, where a specific printer is used for each job.

#### 5. PrintJob Table

The **PrintJob** table logs each print job requested by students:

* **PrintJobID**: A unique identifier for each print job (Primary Key).
* **StudentID**: The student who created the print job (Foreign Key referencing **Student**).
* **PrinterID**: The printer used for the print job (Foreign Key referencing **Printer**).
* **DocumentID**: The document being printed (Foreign Key referencing **Document**).
* **PageCount**: The number of pages in the document.
* **PrintTime**: The timestamp of when the print job was processed.
* **Cost**: The cost associated with the print job.
* **Status**: The current status of the print job (e.g., completed, failed).

Each print job is associated with a student, a printer, and a document, creating a many-to-one relationship between the **PrintJob**, **Student**, **Printer**, and **Document** tables.

#### 6. SPSO (Student Printing Service Officer) Table

The **SPSO** table contains details of officers who manage the printers:

* **OfficerID**: A unique identifier for each officer (Primary Key).
* **Name**: The officer's name.
* **Email**: The officer’s contact email.
* **Permission**: The officer’s access level within the system.
* **Role**: The officer’s role (e.g., administrator, support).

### Key Relationships

* **Student-Transaction**: A student can have multiple transactions.
* **Student-PrintJob**: A student can initiate multiple print jobs.
* **PrintJob-Document**: Each print job corresponds to one document.
* **PrintJob-Printer**: Each print job uses a specific printer.
* **SPSO-Printer**: An SPSO officer can manage multiple printers.

### Conclusion

This ERD represents a well-organized structure for managing the Student Printing Service. By utilizing primary and foreign key relationships, we ensure data integrity and efficient querying across tables such as **Students**, **Transactions**, **PrintJobs**, and **Printers**. The relationships between tables help streamline the management of print services, transactions, and document handling, ensuring a smooth and efficient operation of the system.

**4.1.3 API Management**

The Smart Printing Service System (HCMUT\_SSPS) uses different APIs to make the printing process smooth, secure, and easy to manage.:

1. Security and Authentication APIs:

* These APIs ensure all users are verified and secure using the HCMUT\_SSO system. They control who can access the service and keep all interactions safe.

1. Data Formatting and Processing APIs:

* These APIs allow users to upload documents, images, or text files for printing. They check and prepare the files to make sure they meet the required format and quality for printing.

1. Print Job Management APIs:

* Help create, manage, and track print jobs. Users can schedule printing, check the status, manage available print pages, and cancel print jobs when needed.

1. Payment APIs:

* These APIs handle payments through systems like BKPay, allowing students to buy more print pages or pay for printing services quickly and securely.

1. Custom Print Template APIs:

* Allow users to create and manage personalized print templates based on their specific needs, making the printing process more flexible.

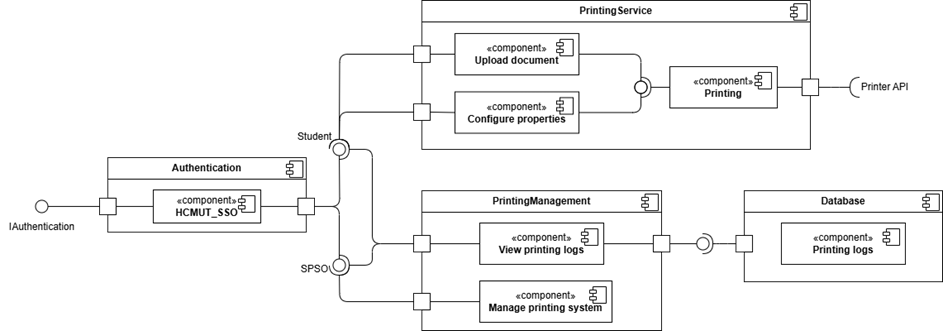
1. Printer Control APIs:

* These APIs let users interact directly with printers, such as managing printer settings, checking status, and making adjustments.

1. Reporting and Analytics APIs:

* Provide data on completed print jobs, including print times, number of pages, and other useful stats for tracking and reporting.

**4.2. Component Diagram:**

****

The components in the diagram:

- Authentication: linked to HCMUT\_SSO service to authenticate users for access to the system

- Printing Service: includes “Printing” which shall require “Upload document” and “Configure properties”, then connects with PrinterAPI

- PrintingManagement: includes “View printing logs” accessible by both Student and SPSO, along with “Manage printing system” only used for SPSO

- Database: contains printing logs history of all the users

**Relationships between components:** The Authentication component connects to the HCMUT\_SSO component, enabling student and SPSO users to authenticate via the HCMUT single sign-on system. Once authenticated, students can access the PrintingService component, which includes the Upload document, Configure properties, and Printing subcomponents. When a student uploads a document, the Upload document subcomponent initiates the process and allows the configuration of print properties through the Configure properties subcomponent. After configurations are set, the Printing subcomponent sends a print request to the external Printer API to execute the physical printing process. The PrintingManagement component is accessible to the SPSO and includes two subcomponents: View printing logs and Manage printing system. The View printing logs subcomponent retrieves printing log data from the Printing logs subcomponent within the Database component, allowing the SPSO to monitor and manage printing records. Additionally, the Manage printing system subcomponent is responsible for overseeing system settings, coordinating with other subcomponents to ensure proper functioning of the printing system. The Printing logs in the Database component serves as the central repository for all print-related records, providing data access for logging and management purposes.