# Software Requirements Specification

# for

# Student Smart Printing Service (HCMUT\_SSPS)

Version 1.0 approved

Prepared by:

1. Nguyễn Bá Nhật Quang– 2252669
2. Nguyễn Đỗ Gia Bảo– 2252069
3. Nguyễn Hồ Phi Ưng– 2252897
4. Nguyễn Huy Hoàng Anh- 2252033
5. Nguyễn Lê Khải Trọng- 2252850

Department of Software Engineering

Faculty of Computer Science and Engineering

Ho Chi Minh City University of Technology – VNU-HCM

17/09/2024

Table of Contents

Table of Contents [ii](#__RefHeading___Toc441230970)

Revision History [ii](#__RefHeading___Toc441230971)

1. Introduction [1](#__RefHeading___Toc441230972)

1.1 Purpose [1](#__RefHeading___Toc441230973)

1.2 Document Conventions [1](#__RefHeading___Toc441230974)

1.3 Intended Audience and Reading Suggestions [1](#__RefHeading___Toc441230975)

1.4 Product Scope [1](#__RefHeading___Toc441230976)

1.5 References [1](#__RefHeading___Toc441230977)

2. Overall Description [2](#__RefHeading___Toc441230978)

2.1 Product Perspective [2](#__RefHeading___Toc441230979)

2.2 Product Functions [2](#__RefHeading___Toc441230980)

2.3 User Classes and Characteristics [2](#__RefHeading___Toc441230981)

2.4 Operating Environment [2](#__RefHeading___Toc441230982)

2.5 Design and Implementation Constraints [2](#__RefHeading___Toc441230983)

2.6 User Documentation [2](#__RefHeading___Toc441230984)

2.7 Assumptions and Dependencies [3](#__RefHeading___Toc441230985)

3. External Interface Requirements [3](#__RefHeading___Toc441230986)

3.1 User Interfaces [3](#__RefHeading___Toc441230987)

3.2 Hardware Interfaces [3](#__RefHeading___Toc441230988)

3.3 Software Interfaces [3](#__RefHeading___Toc441230989)

3.4 Communications Interfaces [3](#__RefHeading___Toc441230990)

4. System Features [4](#__RefHeading___Toc441230991)

4.1 System Feature 1 [4](#__RefHeading___Toc441230992)

4.2 System Feature 2 (and so on) [4](#__RefHeading___Toc441230993)

5. Other Nonfunctional Requirements [4](#__RefHeading___Toc441230994)

5.1 Performance Requirements [4](#__RefHeading___Toc441230995)

5.2 Safety Requirements [5](#__RefHeading___Toc441230996)

5.3 Security Requirements [5](#__RefHeading___Toc441230997)

5.4 Software Quality Attributes [5](#__RefHeading___Toc441230998)

5.5 Business Rules [5](#__RefHeading___Toc441230999)

6. Other Requirements [5](#__RefHeading___Toc441231000)

Appendix A: Glossary [5](#__RefHeading___Toc441231001)

Appendix B: Analysis Models [5](#__RefHeading___Toc441231002)

Appendix C: To Be Determined List [6](#__RefHeading___Toc441231003)

Revision History

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

# Task 1: Requirement elicitation (1.1, 1.2)

## Domain Context

- The domain context of the smart printing service for students at HCMUT is in the education management domain. It is used to help students to print their documents and track their download files and history. It also helps the university to manage the printing of students by sending the logs (information when a student prints something at a printer.) to the Student Printing Service Officer (SPSO), which is under control of the university. The system will operate through a centralized platform that connects various printers across the campus. Each of the student is supplied with an account connected to their university ID, which can help them enter the printing services. It allows students to upload documents to this system from their devices by using wifi or their own mobile communications system.

## Stakeholders and Needs

- The relevant stakeholders of this system are: the HCMUT students, the SPSO:

+ The current needs of the HCMUT students: they need the system to give them access to the the printing service. That service has to be friendly and easy to use. They also need the service to be reliable, meaning that no files are lost in the process of printing and no errors in the printed documents. They also need the files they uploaded are secure and no students can see their uploaded documents. They need a system that allows most of students can access it and handles the scalability in order to print documents without any errors. For example, the system needs to maintain the fast speed during exam weeks to make sure students can print documents and revise for exam tests. Moreover, the system should be updated and maintained the security, new feature requests from students or fixed the errors contains in itself

+ The current needs of the SPSO: They need the system to be able to manage the printing of the HCMUT students as well as the printers in the campus. The application has to be friendly and easy to use. The system has to be secure and it has to make sure that no students can escalate their priviledge to get into the management side of the SPSP.

+ The current needs of the maintenance team: The maintenance team needs tools that allows them to manage the performance of both the software and hardware in real-time. They need to keep their eyes on printer statuses, paper or connectivity issues. The team has to receive the feedback from customers so that they can fix errors and post the new version to the public.

+ The current needs of the development team: The devlopers team needs a well-defined requests and clear objectives from HCMUT students and SPSO. They build a system to satisfy user expactations and their own goals. They also ensure that the print service system should be protected the students’ accounts, documents and information.

+ The current needs of the marketing team: They will create strategies that raise awareness of Smart Priting Service across campus. It is about devloping campaigns that the convenience, security, and cost-effectiveness of the service. They also produce content that educates students on how to use the service by using step-by-step video or make a infographics that demonstrate how to upload files, documents. The clear communication from the marketing team helps reduce confusion and support HCMUT students requests in order to ensure every students can use the system in their first experience.

## Benefits of the System

- The benefits HCMUT-SSPS give to each relevant stakeholder:

+ the HCMUT students: The HCMUT-SSPS will give them a way to print their document faster (because they do not have to go outside to go to a photography store to print their documents and therefore it is more economical), more secure (their documents will not be seen by anyone except for the SPSO).

+ The SPSO: The HCMUT-SSPS will give them a means of management to both the printing of the HCMUT students and the printers. The SPSO will be able to track the printing history of a student and to manage the printers by add/enable/disable them.

+ HCMUT Administration: HCMUT students can benefit from discounted printing because the university can subsidize costs for its students.

+ Environmental Protection: The university manages students' documents appropriately, preventing wasteful use of paper.

+ Copyright Protection: Copyrighted materials from the university can be printed for students, or personal documents can be printed on campus instead of externally, reducing the risk of copyright infringement and information theft.

+ Higher Quality Printing: Documents printed on campus are likely to be of better quality compared to external services.

+ SPSO Oversight: The SPSO can monitor the printing history of each student, allowing for data-driven decisions regarding resource allocation and service improvement.

## Functional Requirements

- The system allows a student to print a document by uploading a document file onto the system, choose a printer, and specifying the printing properties such as paper size, pages (of the file) to be printed, one-/double-sided, number of copies, etc.

- The system has to log the printing actions for all students, including student ID, printer ID, file name, printing start and end time, number of pages for each page size.

- The system allows the SPSO to view the printing history (log) of all students or a student for a time period (date to date) and for all or some printers.

- The SPSO is able to limit and configure the permitted file types.

- A student can view his/her printing log for a time period together with a summary of number of printed pages for each page size.

- The system only allows a student to print some number of pages when it does not exceed his/her account (page) balance.

- Students are allowed to buy more pages using the feature Buy Printing Pages of the system and pay the amount through some online payment system like the BKPay system of the university.

- The SPSO has a feature to manage printers such as add/enable/disable a printer.

- The SPSO can manage other configuration of the system such as changing the default number of pages, the dates that the system will give the default number of pages to all students, the permitted file types accepted by the system.

- The reports of the using of the printing system are generated automatically at the end of each month and each year and are stored in the system, and can be viewed by the SPSO anytime.

- All users have to be authenticated by the HCMUT\_SSO authentication service before using the system.

- SPSO Notification: The SPSO can receive notifications about the operational status of the printers to respond promptly to any issues.

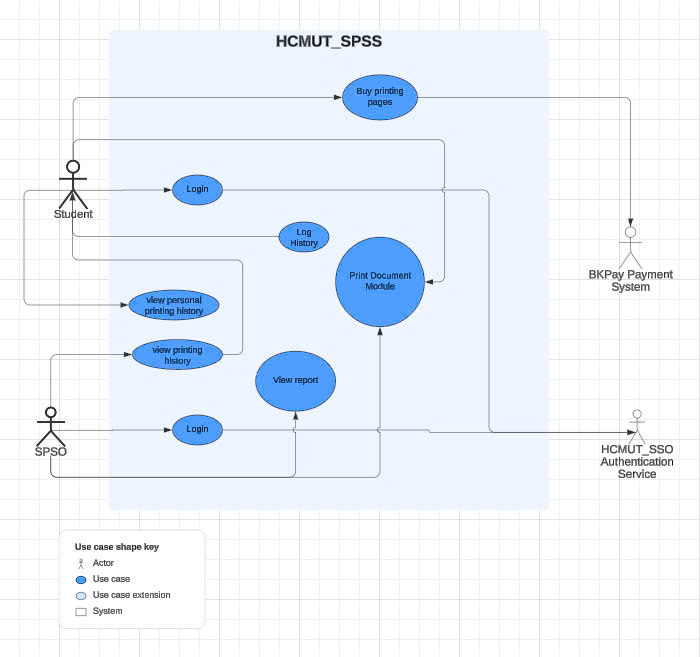
- Printing Limits: The SPSO can establish and adjust printing limits for each student as needed.

## Non- Functional Requirements

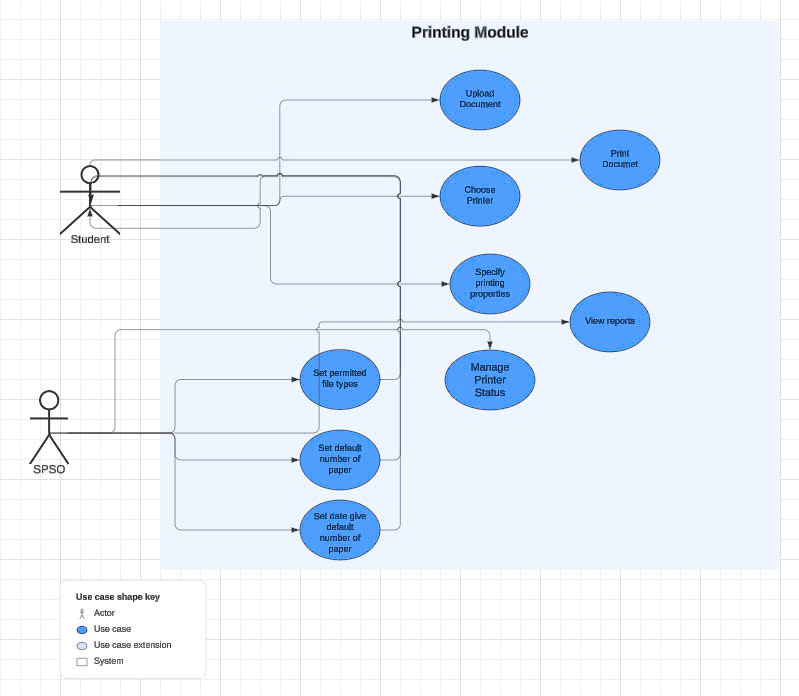
* The system must be available in office hours.
* The system must ensure the security and privacy of users' personal information and printing documents.
* The system must handle multiple simultaneous requests efficiently without significant delays.
* The system must comply with relevant university policies, legal regulations, and data protection laws
* The interface should be intuitive and user-friendly, allowing students to easily upload, preview, and print their documents with minimal training or instruction.
* The system needs to be able to withstand heavy traffic, especially at peak times such as early morning, before assignments or exams.

# Use-case Diagrams (1.3)

## Use-case Diagram for the Whole System



## Use-case Diagram for Printing Module



## The Details of Usecases in Printing Module

### Usecase Print Document

|  |  |
| --- | --- |
| **Use Case Name** | Print Document |
| **Primary Actor** | Student |
| **Secondary Actor** | SPSO (System Printing System Operator) |
| **Description** | The student uploads a document file, selects a printer, and specifies printing properties before printing. The system logs the printing actions. |
| **Trigger** | Student initiates the print action. |
| **Precondition** | Student must be authenticated and have sufficient page balance. |
| **Postcondition** | Document is printed, and printing action is logged. |
| **Normal Flow** | 1. Student logs into the system.  2. Student uploads a document file.  3. Student selects a printer.  4. Student specifies printing properties (paper size, pages, one-/double-sided, number of copies).  5. Student initiates printing.  6. System checks page balance.  7. Document is printed and action is logged. |
| **Alternative Flow** | If the student exceeds the page limit:  1. System alerts the student about insufficient page balance.  2. Student can choose to buy more pages. |
| **Exception Flow** | If the document file type is not permitted:  1. System displays an error message.  2. Student can upload a different file.   If the printer is unavailable:  1. System notifies the student.  2. Student can select a different printer. |

### Usecase View Reports

|  |  |
| --- | --- |
| **Use Case Name** | View Reports |
| **Primary Actor** | SPSO (System Printing System Operator) |
| **Secondary Actor** | None |
| **Description** | Allows the SPSO to view system-generated reports of printing activity at the end of each month or year. |
| **Trigger** | SPSO requests to view system reports. |
| **Precondition** | SPSO is authenticated and reports are available. |
| **Postcondition** | System report is displayed based on the SPSO's request. |
| **Normal Flow** | 1. SPSO selects the report viewing feature. 2. SPSO specifies the desired time period (month or year). 3. System displays the report. |
| **Alternative Flow** | No report data available: 1. System notifies SPSO of no available data for the selected period. |
| **Exception Flow** | System fails to generate or display reports: 1. SPSO is notified of the error. 2. System retries fetching the report. |

### Usecase Manage System Configuration

|  |  |
| --- | --- |
| **Use Case Name** | Manage System Configuration |
| **Primary Actor** | SPSO (System Printing System Operator) |
| **Secondary Actor** | None |
| **Description** | SPSO manages system configuration (page limits, permitted file types, default page balance). |
| **Trigger** | SPSO requests system configuration changes. |
| **Precondition** | SPSO has administrative privileges to configure the system. |
| **Postcondition** | Configuration is updated, affecting future print requests. |
| **Normal Flow** | 1. SPSO modifies system settings (e.g., default page limits, file types, etc.). 2. System updates settings and applies them to all users. |
| **Alternative Flow** | None |
| **Exception Flow** | 1. Configuration changes fail due to system error. 2. System notifies SPSO and logs the failure. |

### Usecase Manage Printer Status

|  |  |
| --- | --- |
| **Use Case Name** | Manage Printer Status |
| **Primary Actor** | SPSO (System Printing System Operator) |
| **Secondary Actor** | Printers |
| **Description** | Allows SPSO to manage printers, including adding, enabling, or disabling printers. |
| **Trigger** | SPSO requests to manage printers. |
| **Precondition** | SPSO is authenticated. |
| **Postcondition** | System configuration is updated based on the SPSO's changes. |
| **Normal Flow** | 1. SPSO selects the printer management feature. 2. SPSO can add, enable, or disable printers. 3. System updates the printer status accordingly. |
| **Alternative Flow** | Configuration action fails: 1. SPSO is notified of the failure. 2. System retries the action. |
| **Exception Flow** | System fails to update configuration: 1. SPSO is notified of the error. 2. SPSO can retry the action later. |

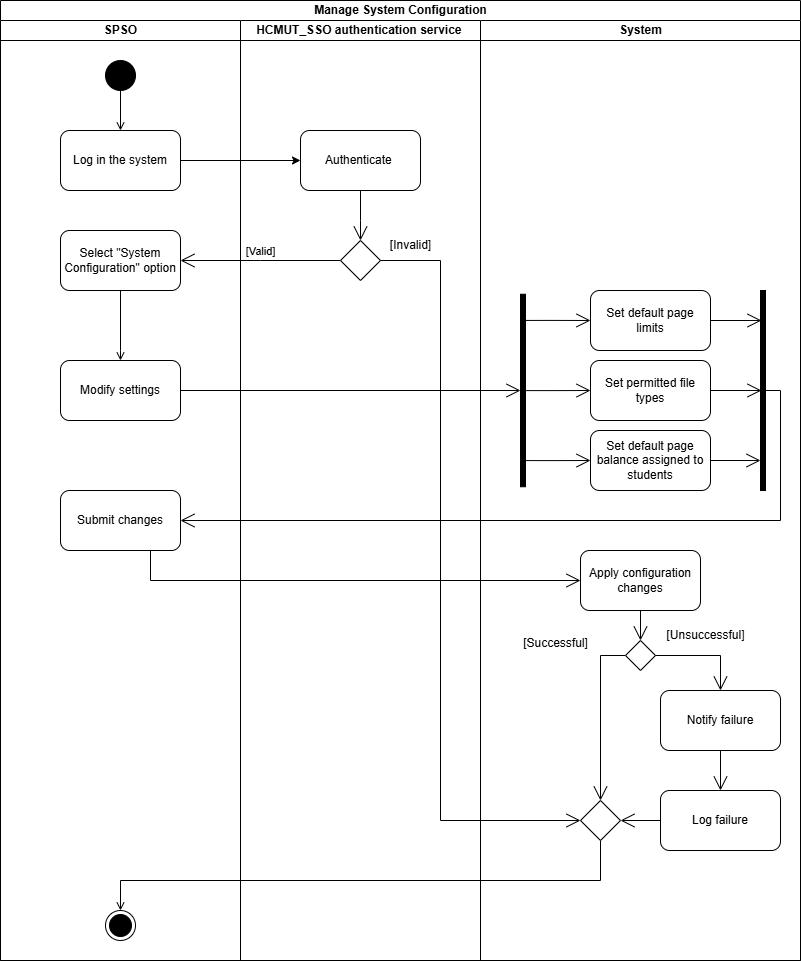
# Activity Diagram for Printing Module

## Use case Print Document

**Description:**   
The activity diagram for the "Print Document" use case captures the process a student follows to print a document, with swimlanes to represent interactions between the student, system, SPSO, HCMUT\_SSO authentication and BKPay system. The normal flow shows the student logging into the system, uploading a document, selecting a printer, specifying printing options, and initiating the print. The system verifies the file type, printer availability, and the student’s page balance.

## Usecase View Reports

**Description:**

The activity diagram for the "View Reports" use case illustrates the process followed by the SPSO to view system-generated reports of printing activities. The diagram uses two swimlanes to separate the actions performed by the SPSO and the system. The SPSO logs into the system, selects the report viewing feature, and specifies the desired time period (monthly or yearly). The system checks the availability of the requested report and, if successful, displays the report to the SPSO. If no data is available, the system notifies the SPSO accordingly. In case of a system error, the system notifies the SPSO and retries the process of fetching the report.

## Manage System Configuration

**Description:**The activity diagram for the "Manage System Configuration" use case shows how the SPSO can modify key system settings such as default page limits, file types allowed for printing, and default page balances. The swimlanes represent the actions performed by the SPSO and the system. After logging in with administrative privileges, the SPSO selects the system configuration option and makes the necessary changes. The system applies the updates and confirms success. In case of a system error while applying the changes, the system notifies the SPSO and logs the failure, prompting further investigation

## Manage Printers

**Description:**

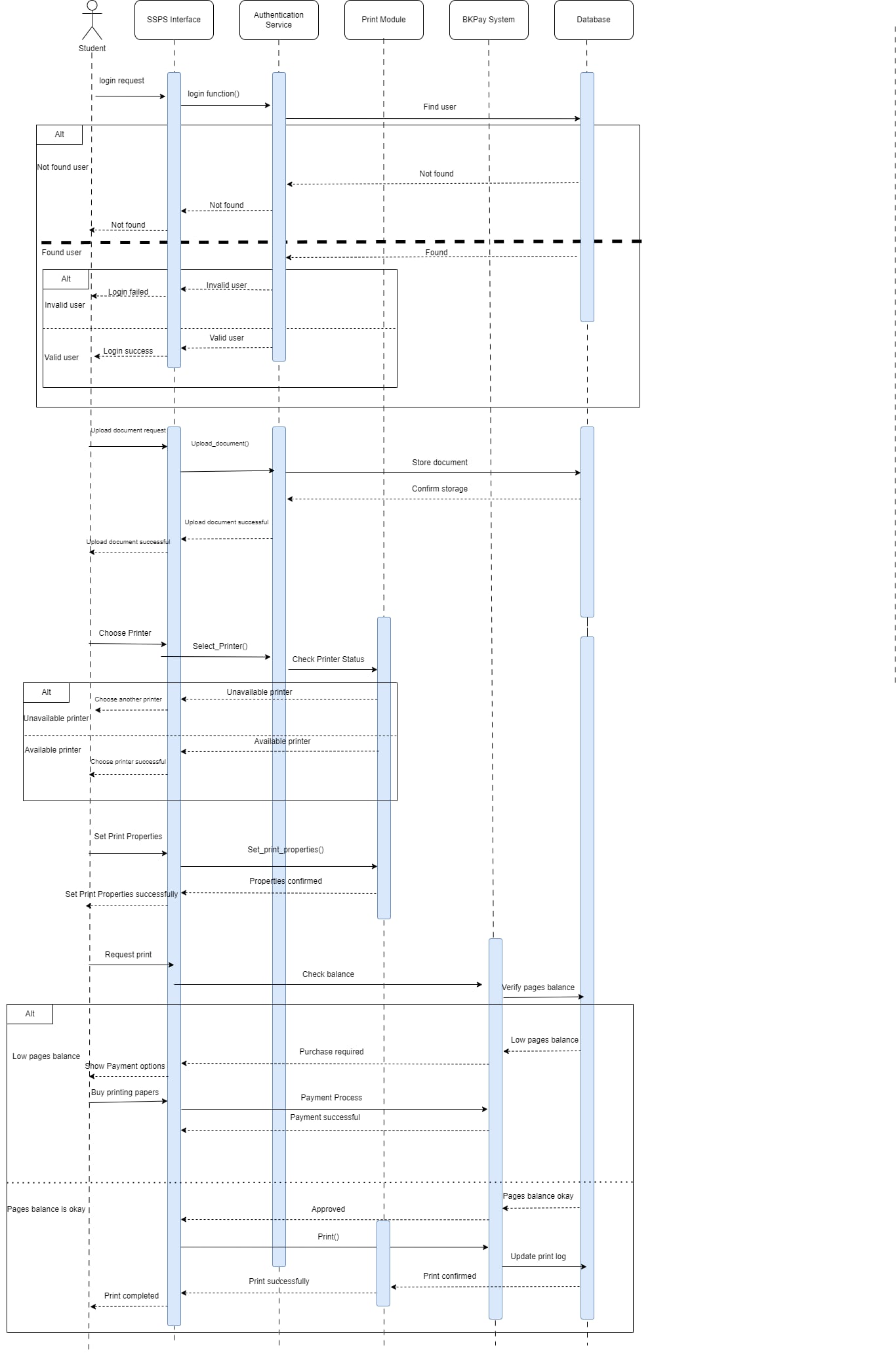
The activity diagram for the "Manage Printers" use case captures the process where the SPSO manages the system's printers. The SPSO can add new printers or enable/disable existing ones. The swimlanes differentiate the SPSO's actions from the system's responses. After logging in, the SPSO selects the printer management option, performs the required action, and submits the changes. The system processes the request and confirms the status change if successful. In case of failure, the system retries the action, and if it continues to fail, it notifies the SPSO and logs the error.

## Usecase View Printer Status Notification

**Description:**

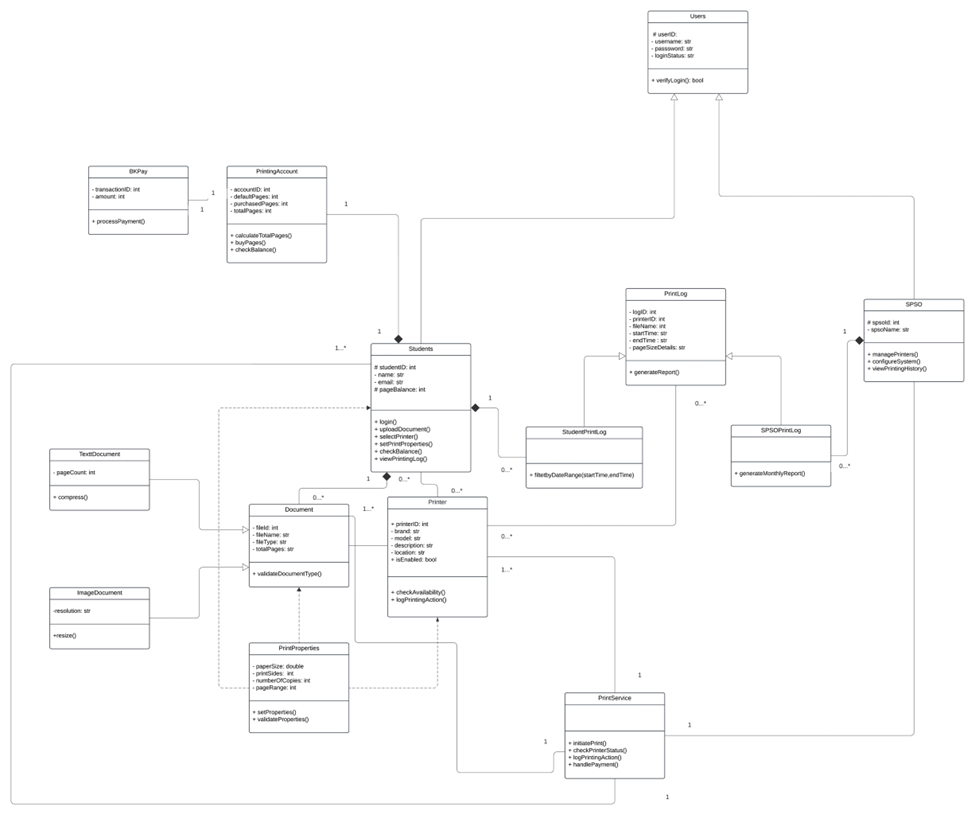
The activity diagram for the "View Printer Status Notification" use case shows how the SPSO is notified when a printer encounters operational issues, such as running out of paper or becoming jammed. The swimlanes distinguish between the actions of the system and the SPSO. The system continuously monitors the printers and sends notifications to the SPSO when issues arise. The SPSO receives these notifications and can take corrective action. If there is a failure in delivering the notification, the system logs the error and the SPSO may need to check the printer status manually.

# Sequence Diagram for Print Module

  
**Description:**

The login process starts when a student sends an SSPS interface-specific request to start the login, which reaches Authentication Service through one of its interfaces. It references the user in database sizes. This ends in "Login failed" as no user is found and results into the passing back of “Not Found” messages. It checks the credentials if a user is found. If the credentials are correct, it finally just reaches Login success and if wrong it terminates with login failed. After that you will see the file uploading process and choosing a printer in flowchart form. One student uploads a document then the documents are processed and saved to the database. When the upload is complete, a “Success” message prompts all students to choose their printer. The system checks for the availability of a printer and either produces an "Unavailable printer" message (which tells students to try another selection) or confirms that there is an available printer with this prompt. Lastly, the print execution and payment process is shown. This is done to set the print properties and to check the balance of the student through BKPay system. But if this condition is not met then a payment is involved. It remains in the system until a certain level of balance has been reached, it is implemented, monitored and recorded in the database if needed. Apart from confirming that the print job has been processed, it is necessary to display a print completed badge to the student after the print job has successfully been completed.

# Sequence Diagram for Printing module



**Description:**

The diagram above describes a printing service management system, where the central object of the system is **Students**, representing the students participating in the printing service. Students can perform operations related to printing and managing their printing accounts.

The system analyzes the relationships between classes as follows: The **Students** class has attributes such as student ID (**studentID**), email (**email**), and available balance for printing fees (**payableBalance**). The behaviors of this class include login (**login()**), document submission (**submitDocument()**), checking printer status (**checkPrinterStatus()**), and checking balance and printing history (**viewPrintingLog()**).

The **PrintingAccount** class allows each student to link with one or more printing accounts (1..\*), managing the student's printing account and storing information such as account ID (**accountID**), number of printed pages, and total amount payable (**totalPages**, **totalPay**). The methods of this class help students check and pay for printed pages through the functions **calculateTotalPages()** and **checkBalance()**.

**BPPay** is the payment processing class that has a "1-1" relationship with the printing account to perform payment processing through the method **processPayment()**. The **Document** class represents the documents that students submit for printing, with attributes such as file ID (**fileID**), file name (**fileName**), and size (**fileSize**). This class has two subclasses: **TextDocument** (managing text documents) and **ImageDocument** (managing image documents), with attributes such as page count and resolution. Documents must be validated before printing through the method **validateDocumentType()**.

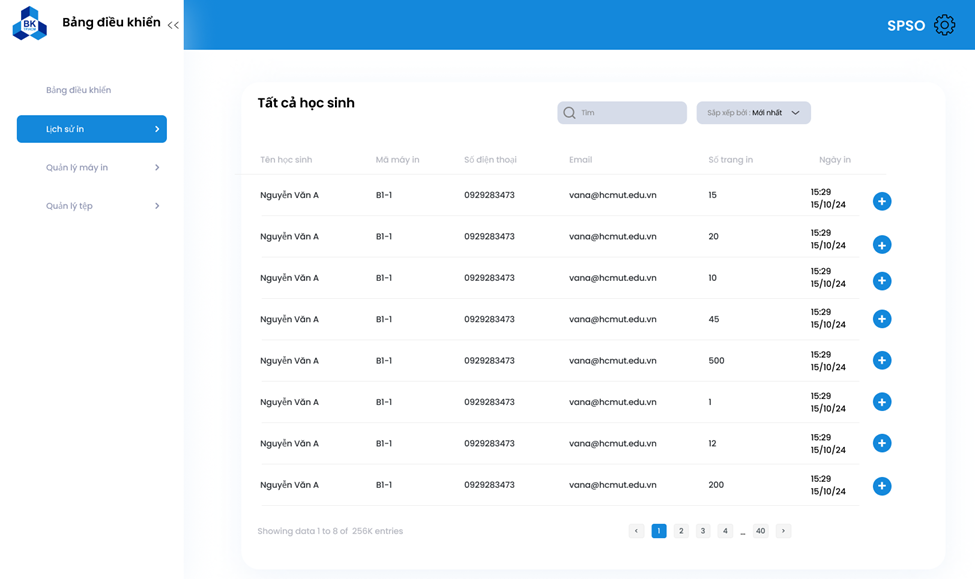
The printing machines are managed by the **Printer** class, which can link to multiple documents. Each printer has attributes such as printer ID (**printerID**), location (**location**), availability status (**isAvailable**), and special features like duplex printing support (**isDuplex**). Printers can check status and log printing activities through the methods **checkAvailability()** and **logPrintingAction()**. The **PrinterProperties** class manages printer attributes such as paper size, number of copies, duplex support, and maximum page capacity (**pageCapacity**). The methods **setProperties()** and **validateProperties()** ensure that documents are printed according to requirements.

The **PrintService** class is responsible for coordinating the entire printing process, from initiating print jobs (**initiatePrint()**), monitoring printer status (**checkPrinterStatus()**), to processing payments for printing (**handlePayment()**). The **PrintLog** class manages the printing history of the system, with detailed information about print sessions, including start and end times, number of pages, and document details (**pagesSizeDetails**). The subclass **StudentsPrintLog** manages the printing history of students, while **SPFSOPrintLog** tracks the printing history of the SPFSO system.

Finally, **SPFSO** is a special part of the printing system that manages printers belonging to SPFSO, ensuring that SPFSO printers operate correctly through the method **managePrinter()**, and can also receive and manage data from other systems using **retrieveSystem()** and **notifySystem()**. In summary, this system is built around the process of printing documents for students, where each student has a printing account and can perform management, checking, and payment operations for their print jobs. The documents, which can be either text or images, will be validated before printing. Printers can serve multiple students simultaneously, while printing services are closely managed through a logging system that records printing and payment history. The SPFSO system provides additional specialized printer management capabilities, enhancing the overall system support.

# User Interface for Printing Module

## View Reports



## A screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedManage Printer Status

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

## A screenshot of a computer Description automatically generatedManage System Configuration

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## A screenshot of a computer Description automatically generatedPrint Document

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

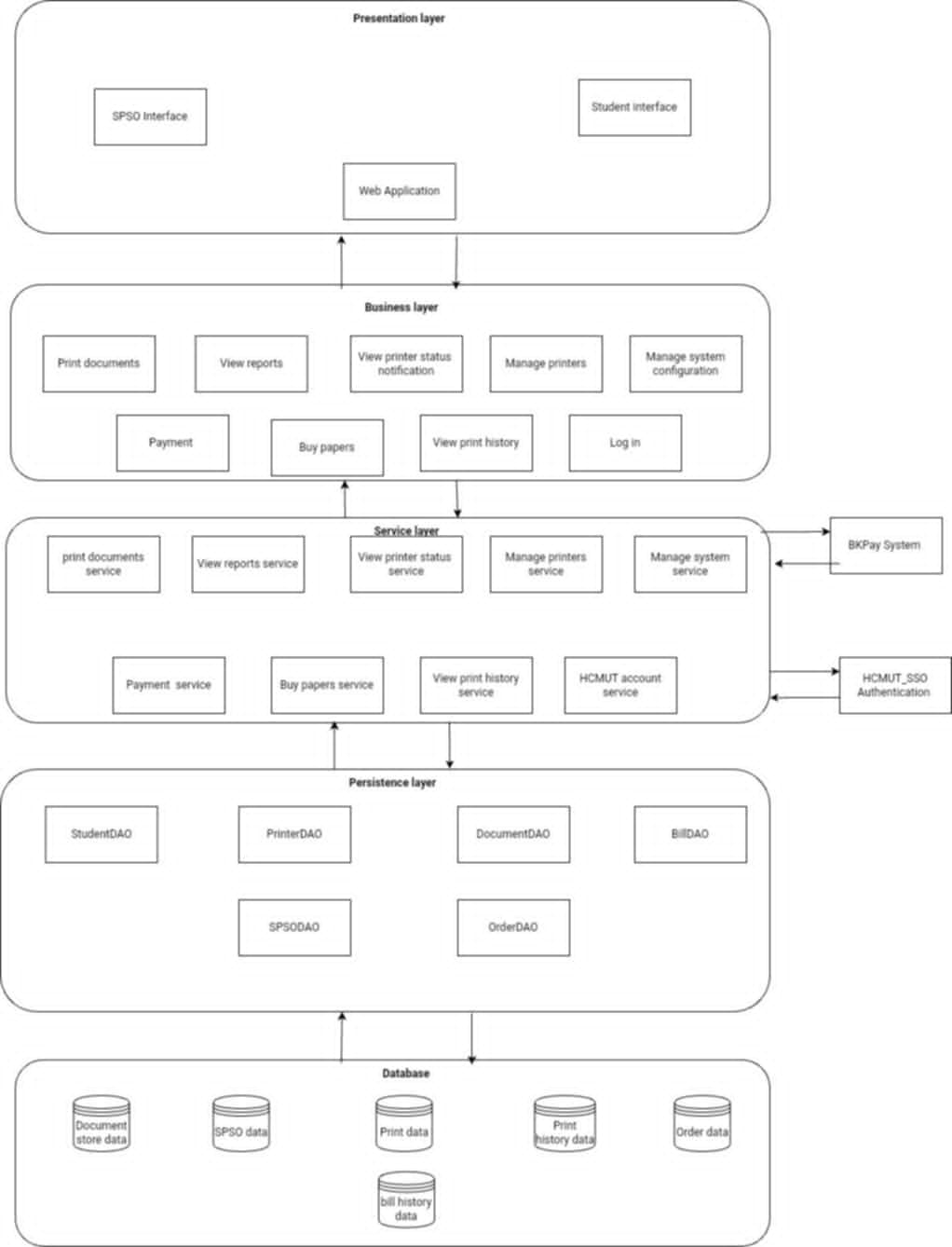
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Architecture Design**
   1. **The architetural diagram of the HCMUT\_SPSS System**



* + 1. **Presentation strategy**

This is the first layer in the architecture. We will adopt a strategy focused on simplicity and ease of use while enhancing the user experience. To achieve this, we will use Tailwind CSS, a modern CSS framework that allows for quick and efficient interface design.

Tailwind CSS enables us to build visually appealing interfaces with utility classes, allowing for flexible styling. We will create interface components such as buttons, forms, and menus with a clean design that meets the needs of users.

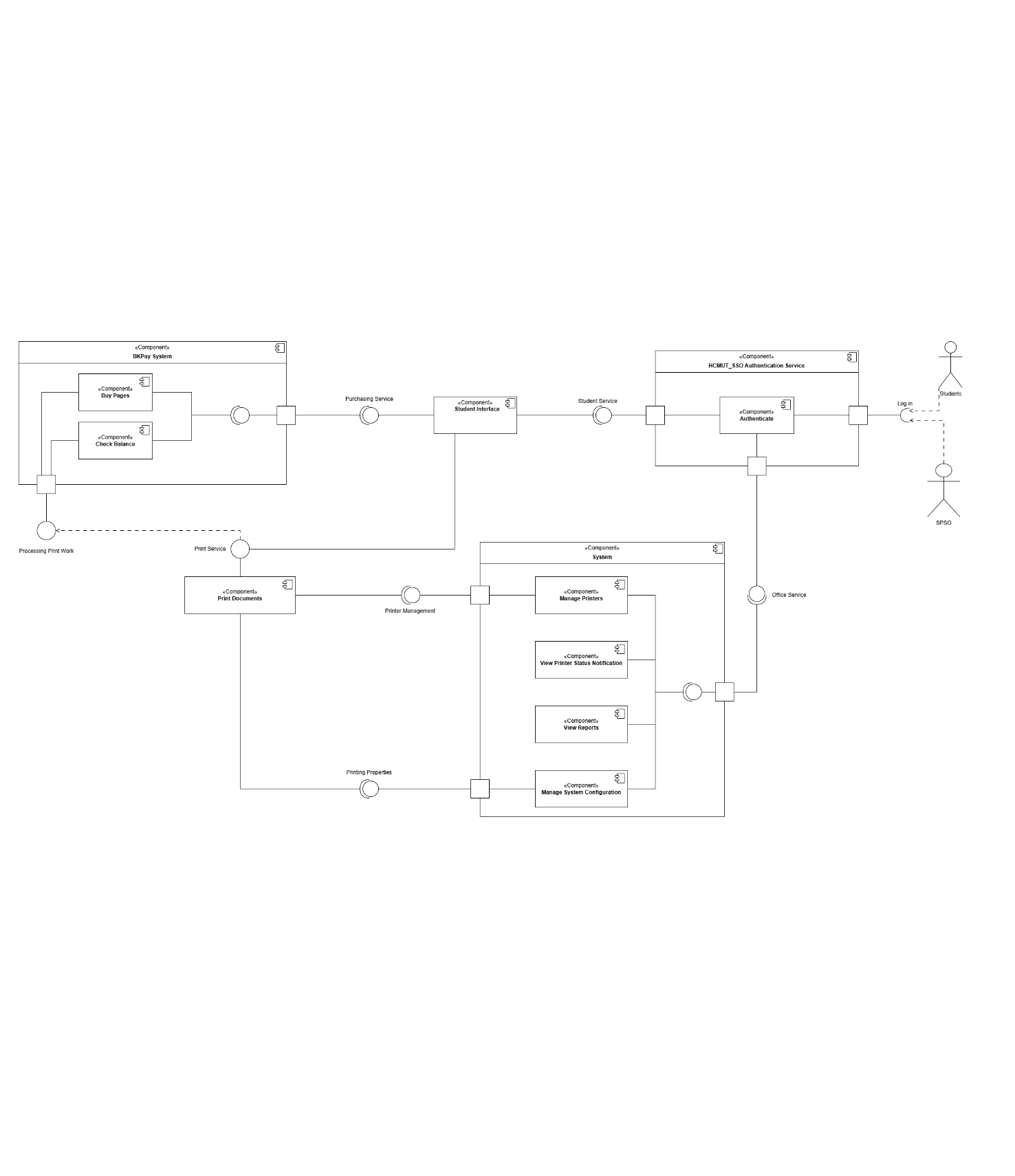
With Tailwind CSS, creating a consistent and easily understandable interface has never been easier. We will ensure that the HCMUT\_SPSS system is not only impressive in appearance but also supports users during interactions, helping them easily access the features of the system.

* + 1. **Data storage approach**

The database layer in the layered architecture plays a crucial role in storing and processing all the information of the application. Here, all data will be stored, and operations such as searching, inserting, updating, and deleting will be performed through the database management system. For the Smart Printing Service project, the team will implement a layered architecture using PostgreSQL as the database management system. The data will be organized into tables, defining the relationships between them. The Smart Printing Service system will include the following entity types:

* **Customer:** Contains detailed information about customers, including customer ID (student ID or staff ID ), full name, password (encrypted), user type (student or staff), email, balance , and the last usage time.
* **SPSO:** Stores information such as SPSO ID, full name, username (for login), password (encrypted), date of birth, email, phone number, and the last usage time.
* **Printer:** Records information such as printer ID, printer name, brand, model, description, location (facility, building, room), and operational status.
* **Document:** Stores information about documents, including document ID, document name, file format, and page count.
* **Print Order:** Includes properties such as print order ID, print configuration (including paper orientation, paper size, type of face, number of pages per face, and scaling), start time, end time, status, and number of pages used.
* **Page Purchase Order:** Contains attributes such as transaction ID, transaction time, number of pages purchased, purchase price, and current status.
  + 1. **API management**

API (Application Programming Interface) is a set of methods and protocols that allows for connection with other libraries and applications. APIs provide access to a collection of commonly used functions, making data exchange between applications easier. The automated printing system HCMUT\_SSPS includes the following types of APIs:

* **Print Queue Management API**: Provides tools for managing the print queue, including adding, removing, and reordering print jobs to optimize the printing process.
* **Document Management and Storage API**: Supports the uploading, storing, and managing of print documents, allowing users to easily search for and access stored documents.
* **Device Status Monitoring API:** Monitors the operational status of printers and related devices, helping to detect and address issues promptly to ensure continuous operation.
* **Notification and Alert API:** Sends notifications to users about important events, such as when a print job is completed or when a printer encounters an issue.
* **Print Job Management API:** Provides tools for creating, monitoring, and managing print jobs, including print scheduling and checking job status.
* **Reporting and Statistics API:** Offers detailed information on completed print jobs, print duration, and the number of printed pages, aiding in effective monitoring and reporting of printing activities.
* **Custom Print Template API:** Allows for the creation and management of custom print templates, assisting applications in generating dynamic print formats based on specific needs.
* **Payment API:** Facilitates the payment process for printing documents and purchasing print credits through BKPay.
* **Security and Authentication API:** Ensures the safety of communications and printing processes by utilizing authentication methods from HCMUT-SSO and access control via SPSO.
* **Printer Control API:** Enables applications to interact directly with printers, from sending print commands to adjusting printer settings.
* **Input Data Formatting and Processing API:** Assists applications in sending data (such as print files, images, or text) to the automated printing system while ensuring that the data meets specific formatting and quality requirements.
  1. **The component diagram of the Printing module**

This component diagram models our printing system. The system is designed to provide secure, efficient printing services with purchasing and reporting functionality for students and administrators. It includes the following main components: SSO\_Authentication, BKPay, Print Document, Student interface ,and System. The System component further contains several subcomponents that manage printing, printer status, reporting, and system configuration.

* **SSO\_Authentication:** This component handles user login and authentication, allowing both students and administrators to access the system securely using single sign-on (SSO) functionality. It ensures that each user has the appropriate permissions for their role. This component provides the difference interfaces based on the role of the user. (student\_interface for students and office\_interface for admin, or SPSS).
* **Student interface:** This component is an intermediary for the activities of students. This component helps the students to access to the BKPay system for purchasing more pages as well as checking their balance, to upload and print documents.
* **BKPay:** The BKPay component manages the purchasing and balance-checking functionalities. Students can use this component to purchase printing credits for A4 and A3 pages and view their remaining balance. It also provided a Processing Print Work interface, for checking whether a student is able to print a document (i.e. it compares the number of pages the student has and the number of pages needed to print the document) before the Print document component start printing.
* **Print document:** This component handles the printing process. It is the abstract representation for the printers.
* **System:** The core System component contains several subcomponents that coordinate various printing system operations:
* **Manage Printer:** This subcomponent allows administrators to add, remove, and configure printers in the network. It ensures that only configured printers are available to users.
* **View Printer Status:** SPSS and administrators can check the availability and operational status of printers through this component.
* **View Report:** Each time a student prints a document, a report detailing the printing activity, including details such as the user, document type, number of pages, and timestamp is generated. The report is then sent to the System for record-keeping and viewing.
* **Manage System Configuration:** This subcomponent allows administrators to manage the overall system settings, such as specifying the types of documents allowed for printing (e.g., PDF, DOCX) and configuring other properties such as default pages given to students and default date that the System will give out the pages.