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

SOFTWARE ENGINEERING (CO3001)

Software Document for:

A smart printing service for students at HCMUT

Version 1.0 approved

Instructor: Trương Tuấn Anh, CSE-HCMUT

Group: CC03-04
Student(s): Lý Tuấn Lộc - 2252456
Lê Thị Phương Thảo - 2252757
Lý Triều Uy - 2252889
Nguyễn Anh Khoa - 2252351
Lưu Quang Hoàng Cương - 2152032

HO CHI MINH CITY, NOVEMBER 2024

Contents

1 Revision History **3**

2 Task 1: Requirement elicitation (1.1, 1.2) **4**
2.1 Domain Context . . . 4
2.2 Stakeholders and Needs . . . 4
2.2.1 Students . . . 4
2.2.2 Student Printing Services Officer . . . 4
2.2.3 IT Developers . . . 5
2.2.4 BKPay Administrator . . . 5
2.2.5 The Authentication Service . . . 5
2.2.6 Printer Manufacturer . . . 5
2.3 Benefits of the system . . . 5
2.4 Functional Requirements . . . 6
2.4.1 Students . . . 6
2.4.2 Student Printing Services Officer . . . 6
2.5 Non-Functional Requirements . . . 7
2.5.1 User Interfaces . . . 7
2.5.2 Reliability . . . 7
2.5.3 Performance . . . 8
2.5.4 Security and Privacy . . . 8
2.5.5 System Organization . . . 8

3 Use case (1.3) **9**
3.1 Use case Diagram for the Whole System . . . 9
3.2 Use-case Diagram for Printing Module . . . 9
3.3 The Details of Use Cases in Print Module . . . 10
3.3.1 Use Case Print . . . 10
3.3.2 Use case Upload file . . . 10
3.3.3 Use case Specify properties . . . 12
3.3.4 Use case Choose printer . . . 13
3.3.5 Use case Confirm Print . . . 14
3.3.6 Use case Manage Printers . . . 15

4 Task 2: System modeling **16**
4.1 Activity Diagram . . . 16
4.1.1 Upload File . . . 16
4.1.2 Configure File . . . 17
4.1.3 Select Printer . . . 18
4.1.4 Confirm Print . . . 19
4.1.5 Manage Printers by SPSO . . . 20
4.2 Sequence Diagram . . . 21
4.2.1 Upload File . . . 21
4.2.2 Configure File . . . 22
4.2.3 Select Printer . . . 23
4.2.4 Confirm Print . . . 24
4.2.5 Printer Report by SPSO . . . 25
4.3 Class Diagram . . . 25
4.4 User Interface . . . 26
4.4.1 Student . . . 28
4.4.2 SPSO . . . 33

5 Architecture Design **35**
5.1 Layered Architecture . . . 35
5.2 Presentation Strategy . . . 35
5.3 Data storage approach . . . 36
5.4 API Management . . . 37
5.5 Deployment Diagram . . . 38

5.6 Component Diagram 39
5.6.1 Description of the Components in the Diagram 39
5.6.2 Relationships Between Components 40
5.6.3 Overall Functionality 40

1 Revision History

2 Task 1: Requirement elicitation (1.1, 1.2)

2.1 Domain Context

Printing is an essential part of student life, as students have diverse printing needs, including assignments, research papers, class notes, presentations, and other academic tasks.

The **HCMUT Smart Printing Service (HCMUT-SSPS)** is an automated printing solution designed to serve the students of HCMUT across multiple campuses. The primary goal of the system is to provide a convenient and efficient way for students to print their academic documents. The service involves a network of printers located strategically around campus buildings, which are accessible via a web-based and mobile application.

The system allows students to upload documents in specific formats, choose a printer, and set custom printing preferences such as paper size, page selection, single or double-sided printing, and the number of copies. It is integrated with the university's authentication system (HCMUT-SSO) to ensure secure access and with the university's payment system (BKPay) for page balance management. Students are allocated a default number of A4-size pages per semester, which they can track and manage, purchasing more pages as needed through the system.

The HCMUT-SSPS is designed to log all printing actions, capturing details like the student's ID, printer ID, file name, and the time taken for each print job. In addition, the system provides comprehensive logging and reporting capabilities for **the Student Printing Service Officer (SPSO)**, who can view and generate reports of printing activities for individual students, specific printers, or over specific time periods. The SPSO also manages system configurations, such as enabling/disabling printers, adjusting page limits, and modifying accepted file types.
This service aims to automate and simplify the printing process for students while giving administrators detailed control over printer usage, resource allocation, and reporting.

2.2 Stakeholders and Needs

2.2.1 Students

- An intuitive interface that is easy to navigate and use.
- A convenient online payment system for purchasing additional printing pages and checking their account balance.
- Secure authentication measures to protect student accounts from unauthorized access.
- The system should be readily available from various locations on campus.
- The system should have the ability to view their printing history at any time.

2.2.2 Student Printing Services Officer

- A comprehensive and intuitive dashboard for monitoring printer productivity and usage trends.
- The ability to configure system settings such as default page size, permitted file type, or default page number to manage many printers efficiently.
- Tools for managing printer inventory, including adding, enabling, disabling, removing, and tracking printer status.
- The authority to access printing logs of all students.
- A monitor device to ensure that printers are always accessible to students.

2.2.3 IT Developers

- Proficiency in programming languages, development tools, and system architecture.
- Design and implementation of the HCMUT_SSPS system features (e.g., printer configuration, printing history logs, printer management).
- Integration of the system with The Authentication Service to enable Single Sign-On (SSO) login.

2.2.4 BKPay Administrator

- Manage online payments for the HCMUT_SSPS.
- Ensure secure and efficient transaction processing by implementing OTP authentication via SMS.
- The ability to check student account balances before processing "Buy Printing Pages" purchases.

2.2.5 The Authentication Service

- Verify user identity with robust authentication methods in order to prevent unauthorized access.

2.2.6 Printer Manufacturer

- Ensure printer compatibility with HCMUT_SSPS software and hardware.
- Partner with the university to increase brand awareness and ensure a market share within the printing system.
- Provide warranties and reliable technical support.

2.3 Benefits of the system

The HCMUT-SSPS (Student Smart Printing Service) offers multiple benefits to various stakeholders. For students, the system provides convenience and efficiency by allowing them to print documents from any campus location, track their printing history, and manage their page quota. This service reduces the need to locate specific printers or manually handle print jobs, which streamlines their academic workflow. Additionally, students have the flexibility to purchase extra pages when needed, which ensures uninterrupted printing service. According to studies on self-service systems in education, such functionalities increase student satisfaction and reduce administrative burden.

For **the Student Printing Service Officer (SPSO), HCMUT-SSPS** offers a centralized management system, allowing real-time monitoring of printer performance and usage logs. The ability to manage printer configurations and access detailed reports helps the SPSO maintain efficiency and optimize resource distribution. Automated reporting tools reduce the workload of manual data collection, while configurable settings—such as page quotas and file types—ensure adaptability to evolving student demands. Effective management of such services has been shown to increase operational efficiency in university environments.

For **University administrators**, the system generates valuable insights through periodic reports on printer usage, which aid in decision-making processes, such as budgeting and resource allocation. By supporting seamless printing services, the system enhances the university's infrastructure and promotes a tech-enabled campus environment. This aligns with the broader trend of universities adopting smart technologies to improve service delivery and student satisfaction. Furthermore, integrating online payment systems like BKPay streamlines revenue collection for additional printing services, providing a financial model that supports both sustainability and growth.

For **Printer Manufacturers**, manufacturers benefit from the system's centralized printer management, which allows early detection of issues such as low ink levels or paper shortages. This ensures prompt maintenance and servicing, extending the lifespan of printers and reducing the risk of downtime. The system's usage data also helps manufacturers optimize their support and service offerings.
For **School IT Administrators**, IT administrators play a critical role in ensuring the HCMUT-SSPS

system's availability and security. By integrating the HCMUT_SSO authentication system, IT administrators ensure that only authorized users access the system. Additionally, they oversee system maintenance and troubleshooting, ensuring that both the web-based and mobile apps function efficiently. This ensures a smooth and secure printing experience for all users.

2.4 Functional Requirements

2.4.1 Students

- Students can log in by using the HCMUT_SSO authentication service.
- Students shall be able to upload files they want to print.
- Students shall be able to click on each printer to view details and submit print jobs.
- Students shall be able to choose a printer based on its location and availability.
- Students can choose properties such as paper size, pages, number of copies, one/double-sided.
- Students shall be able to view a summary of their number of printed pages for each page size (A4, A3).
- Students shall be able to purchase additional printing pages using the "Buy Printing Pages" feature and pay online through the BKPay system.
- The system should provide a print history feature that allows students to view their printing history within a specified time frame, including a summary of the number of pages printed for each paper size.
- The system should only allow students to print if the number of pages they want to print does not exceed their page balance.
- If a printer runs out of paper, that option should be hidden from the students. Only printers with sufficient paper will be displayed.
- The system should store a detailed print history for all students, including student ID, printer ID, file name, printing start and end time, and the number of pages for each page size.
- The system should automatically add a default number of pages to each student's page balance at the beginning of each semester, based on the date specified by the administrative office.
- The system should require students to log in using the HCMUT_SSO authentication service before accessing the system.
- Students should have individual accounts that allow them to log in and log out.
- Important actions such as printing, logging in, logging out, and modifying print settings should be accompanied by notifications or visual feedback to confirm successful completion.
- The system should restrict user access to their own uploaded files, preventing them from viewing other students' files.

2.4.2 Student Printing Services Officer

- SPSO can log in using the HCMUT_SSO authentication service.
- SPSO can see the logging action and information of all students who used the printer.
- The SPSO has the ability to manage printers, including adding, enabling, and disabling devices.
- SPSO has a feature to manage the system, such as changing the default number of pages, the date that the system will give printing pages for all students, and setting the default date for page allocation.
- SPSO can fix file upload permissions of the system.

- SPSO has the report to view the usage history of students in each period.
- Only permitted file types should be uploaded, limiting the size of uploaded files, and input files should be scanned for potential threats.
- No documents are lost during the print process; backups of the system data (logs, reports, documents) should be scheduled monthly.
- The system should monitor the status of printers and send notifications to the SPSO when issues arise, such as paper jams or empty paper trays.
- The system should be equipped with intrusion detection capabilities to alert the SPSO of any unauthorized access attempts.
- The SPSO will have limited access to student data, being able to view only file names and associated information such as file size and page count. The content of the printed files will not be accessible to the SPSO.
- The SPSO can access automated reports generated at the end of each month or year.

2.5 Non-Functional Requirements

2.5.1 User Interfaces

- The interface should be visually appealing, modern, and designed with students in mind.
- Fonts should be large, clear, easy to read, and have high contrast.
- Functional buttons and content dialog boxes should be arranged in a logical and user-friendly manner.
- The primary color scheme should be cool tones, with contrasting colors used to highlight notifications and tasks.
- The system should support multiple languages; currently, Vietnamese, with the potential to add more languages in the future.
- The interface should adapt to various screen sizes, including desktops, mobiles, and tablets.
- Information about active printers in different buildings should be displayed on a single screen, eliminating the need for scrolling.

2.5.2 Reliability

- The system must operate stably from 7:00 AM to 5:00 PM on all weekdays (excluding Sundays and public holidays).
- Printers must be operational for at least 12 consecutive hours daily.
- The probability of system failure should be less than 3
- When a printer encounters issues such as paper jams or empty paper trays, the system must automatically send a notification to the SPSO within 10 seconds. Support teams should address the issue within 5 minutes.
- Data on printer status and student accounts should be updated every 15 minutes with at least 95
- New notifications should be updated within 10 seconds of any changes.
- The backup and recovery functions must operate reliably and accurately 99
- The system must generate accurate reports on schedule, with a maximum delay of 2 days.
- The network connection between the server and printers must be reliable. Connections to external systems must also be reliable.

2.5.3 Performance

- The system should be able to handle concurrent requests from at least 1000 students at the same time without affecting overall performance.
- The system must be able to handle real-time data from at least 1000 students simultaneously.
- Printers should operate efficiently, with a minimum capacity of 5000 pages per day.
- Ensure that the system can store large amounts of data, even with high volumes of uploaded files and logs.
- Ensure each printer state and condition to prevent access by students when the printer is not working.
- The system database must clearly separate user functions for students and SPSO.
- Page load times should be less than 3 seconds.
- The system should process user actions within 2 seconds.
- System responses should be real-time, with a latency of less than 1 second.
- Due to the large size of files uploaded by students for printing, the system must optimize storage for these files. Each file uploaded by a student should not exceed 30MB.
- Responses from external systems such as HCMUT_SSO authentication service or BKPay (online payment system) should be received within 5 seconds when performing related functions.

2.5.4 Security and Privacy

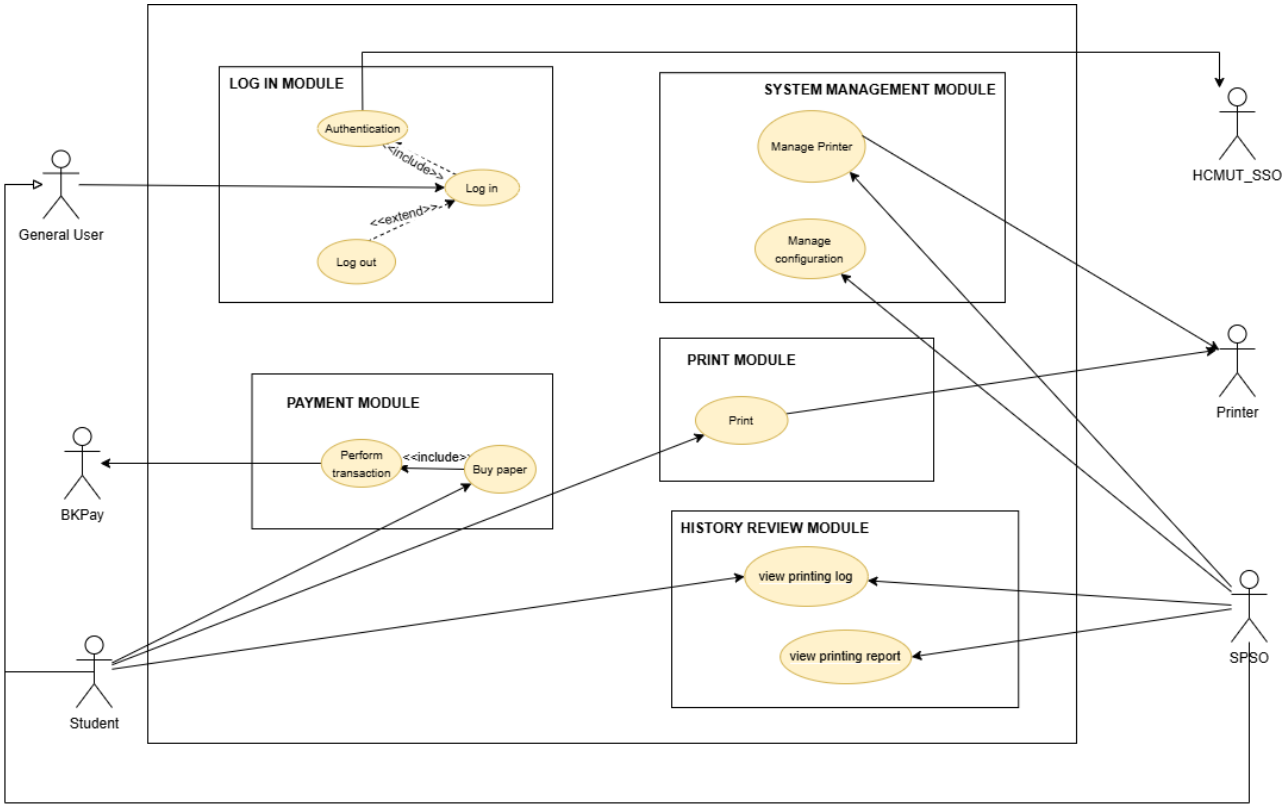
- Only student accounts from HCMUT_SSO are allowed to log in. User registration is not permitted.
- The system must notify the SPSO within 3 seconds of detecting any unauthorized access. The system should resolve the issue within 5 minutes.
- The system must enforce strict access controls to limit the information that each user can access, ensuring the confidentiality of all user data.

2.5.5 System Organization

- Printers and servers should be strategically located. Printers should not be placed more than 100 meters apart or less than 50 meters apart. At a minimum, each academic building should have one printer.
- The system should be scalable and flexible.

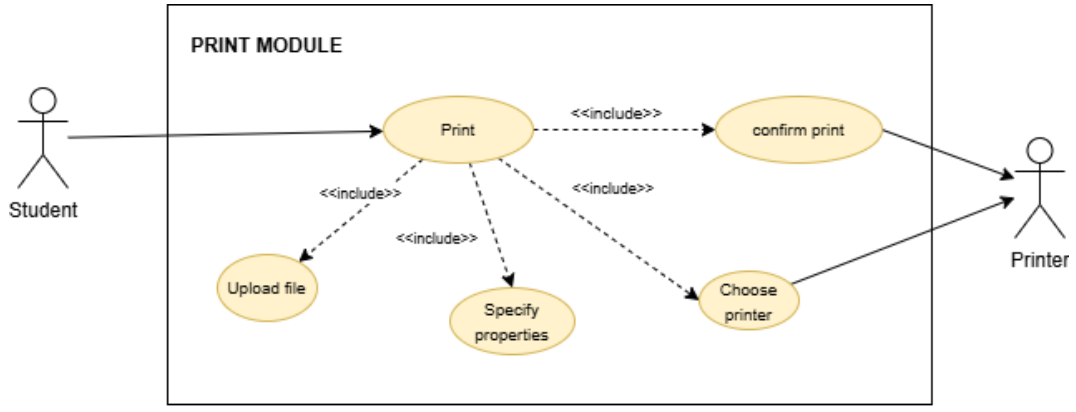
3 Use case (1.3)

3.1 Use case Diagram for the Whole System



Use-case Diagram for Whole System

3.2 Use-case Diagram for Printing Module



Use-case Diagram for Printing Module

3.3 The Details of Use Cases in Print Module

3.3.1 Use Case Print

Use case name	Print
Created by	Lý Triều Uy
Date created	29.09.2024
Description	The student initiates the print process. The system then displays a new interface where the student can upload a file, specify print properties, and select a printer.
Trigger	User clicks on “Print” option
Primary actor	Students
Pre-condition	User has successfully logged in HCMUT_SPSS System
Post-condition	The system displays the print interface where the student can proceed with uploading a file, specifying properties, and choosing a printer.
Normal flow	1. The student clicks the "Print" button. 2. The system displays the print interface with options for uploading a file, specifying properties, and choosing a printer.
Alternative flow	If the student clicks "Cancel" on the initial interface, the system returns to the previous state without taking further action.
Exception flow	If the system fails to load the print interface, it displays an error message and prompts the student to retry.

Table 1: Use case - Print

3.3.2 Use case Upload file

Use case name	Upload file
Created by	Lưu Quang Hoàng Cương
Date created	29.09.2024
Description	The student uploads a document file that needs to be printed on the new print interface.
Trigger	The student clicks the "Upload File" button on the print interface.
Primary actor	Students
Pre-condition	The student is on the print interface.
Post-condition	1. The file is successfully uploaded and ready for printing. 2. The file is saved in temporary storage for processing.
Normal flow	1. Student chooses the “Upload file” option. 2. The system prompts the student to select a file. 3. Student selects the document for printing and clicks the “Upload” button. 4. The system verifies the file type. 5. The system uploads the file to the server. 6. The system confirms that the file is uploaded and ready.
Alternative flow	At step 5: Student clicks the “Cancel” button to stop the upload process.
Exception flow	If there is an error in file uploading (invalid file type, network error, etc.), the system displays an error message asking the student to upload a compatible file format.

Table 2: Use case - Upload file

3.3.3 Use case Specify properties

Use case name	Specify properties
Created by	Lê Thị Phương Thảo
Date created	29.09.2024
Description	The student specifies the print properties for the uploaded file (pages to be printed, number of copies, size, orientation, number of pages per sheet)
Trigger	Student clicks the “Set print properties” button
Primary actor	Students
Pre-condition	1. The user has successfully logged in and confirmed the account. 2. The user uploaded the file successfully. 3. The system is working stably. 4. The internet connection is stable.
Post-condition	The system successfully sets the properties for the printed papers.
Normal flow	1. The student clicks the "Specify Properties" button. 2. The system displays property settings (e.g., paper sizes, copies, orientation, pages per sheet). 3. The student modifies the properties as needed: • Enter the number of pages to print. • Select paper orientation • Select 1-sided or 2-sided printing. • Select number of pages per sheet. • Select paper size. • Enter the number of copies of the print. 4. The student clicks "Save." 5. The system pops up a notification confirming that the properties are saved.
Alternative flow	1. If the student does not have enough paper balance to print, the system displays a pop-up notification informing them of the shortage. 2. The pop-up asks if the student wants to go to the "Buy Paper" screen. 3. If the student selects "Yes," the system navigates to the "Buy Paper" screen. 4. The student completes the purchase. 5. After purchasing, the student returns to the print interface to proceed with printing. 6. If the student selects "No," the system stays on the current screen, allowing the student to adjust properties or cancel the print process.
Exception flow	1. If the properties are invalid, an error message will pop up, and the user will be required to return to the settings interface to correct the properties. 2. If the payment for more paper fails, the system displays an error message and asks the student to retry or cancel.

Table 3: Use case - Specify Properties

3.3.4 Use case Choose printer

Use case name	Choose Printer
Created by	Nguyễn Anh Khoa
Date created	29.09.2024
Description	The student selects a printer for the print job from the list of available printers on the print interface.
Trigger	The student clicks the "Choose Printer" button on the print interface.
Primary actor	Students
Secondary actor	Printer
Pre-condition	• The student is on the print interface and has uploaded a file and specified properties. • The internet connection is stable for loading available printers.
Post-condition	The selected printer is assigned to the print job.
Normal flow	1. The student clicks the "Choose Printer" button. 2. The system displays a list of available printers which have enough paper for printing. 3. The student selects a printer from the list. 4. The student clicks "Confirm." 5. The system assigns the selected printer to the job.
Alternative flow	1. If the student clicks "Cancel," the system returns to the print interface without assigning any printer. 2. If no printers have enough paper for the print job, the system displays a message informing the student that no printers are available and prompts them to try again later.
Exception flow	If the selected printer runs out of paper after being chosen, the system displays an error message and prompts the student to select a different printer.

Table 4: Use case - Choose Printer

3.3.5 Use case Confirm Print

Use case name	Confirm Print
Created by	Lý Tuấn Lộc
Date created	30.09.2024
Description	The student reviews the selected file, print properties, and chosen printer. The student then confirms the print job to start the printing process.
Trigger	Click on the “Confirm Print” button.
Primary actor	Students
Secondary actor	Printer
Pre-condition	• The user has successfully logged in and confirmed the account. • The user uploaded the file successfully. • The user has selected specific properties. • The system is working stably. • The internet connection is stable. • The user have enough paper balance.
Post-condition	The system will print out the document with the specific properties that the user has input.
Normal flow	1. The user reviews the properties that they have selected. 2. The user clicks the "Confirm Print" button. 3. The system sends the print job to the selected printer. 4. A confirmation message is displayed.
Alternative flow	At Step 1, if the user isn’t satisfied with the properties, they can select the “Cancel” option, which will bring the user back to the “Specify Properties” section.
Exception flow	At Step 3, if the internet connection is unstable and causes an interruption between the system and the printer, the system will display an error message to inform the user and take the user back to Step 1.

Table 5: Use case - Confirm Print

3.3.6 Use case Manage Printers

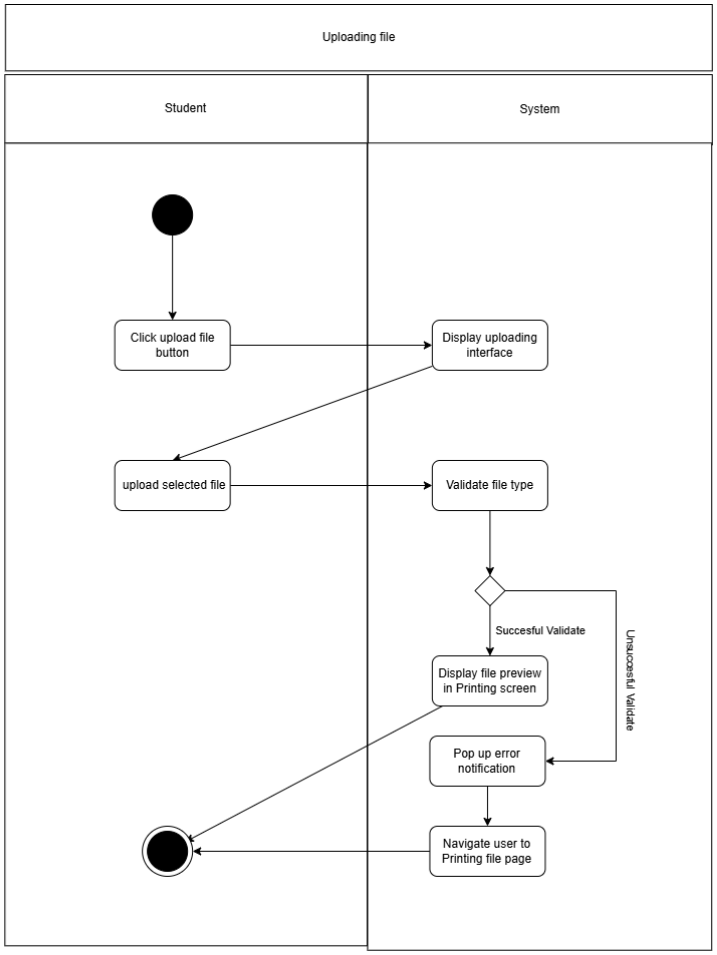
Use case name	Manage Printers
Created by	Lý Tuấn Lộc
Date created	30.09.2024
Description	The SPSO reiew the print order statistics and printer statistics. SPSO is able to view detail about selected printer.
Trigger	SPSO click on the "Printer Report" button.
Primary actor	SPSO
Secondary actor	Printer
Pre-condition	• The user has successfully logged in and confirmed the account. • The system is working stably. • The internet connection is stable.
Post-condition	The system will show the statistic with the specific numbers and charts that the user has chosen.
Normal flow	1. SPSO clicks on Printer Report button 2. The system display the printer system statistic including charts and tables. 3. The user clicks to specific printer button. 4. The system display the statistic about the selected printer such as printer usages, some recent print orders sent to this printer,...
Alternative flow	At Step 2, if the user want to view another time interval statistic, they can select time interval option, which will show the user different time interval statistic, for example: weekly, monthly,...
Exception flow	At Step 2, if the internet connection is unstable and causes an interruption between the system and the printer, the system will display an error message to inform the user and take the user back to Step 1.

Table 6: Use case - Printer Report

4 Task 2: System modeling

4.1 Activity Diagram

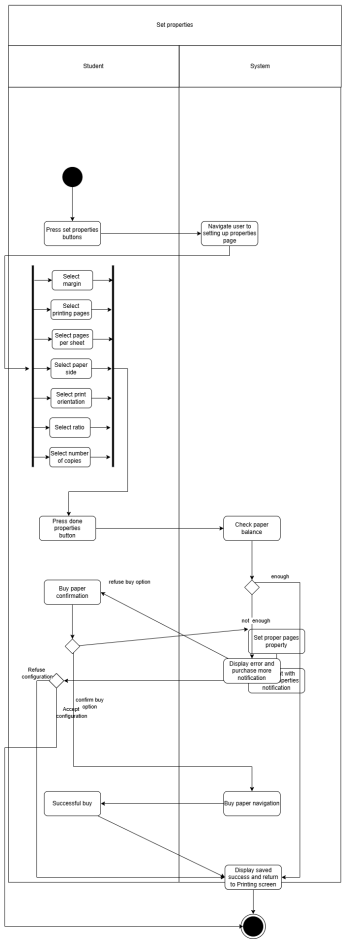
4.1.1 Upload File



Hình 1: Activity Diagram for Upload File

When a user is on the **Print Document** page, they can click the **Upload File** button to begin the upload process. After clicking, they are prompted to select a file from their device to upload for printing. Once the file is selected, the system checks the file type to ensure it is compatible. If the file type is unsupported, an error message appears, notifying the user of the issue. The user is then returned to the file upload interface to try again.

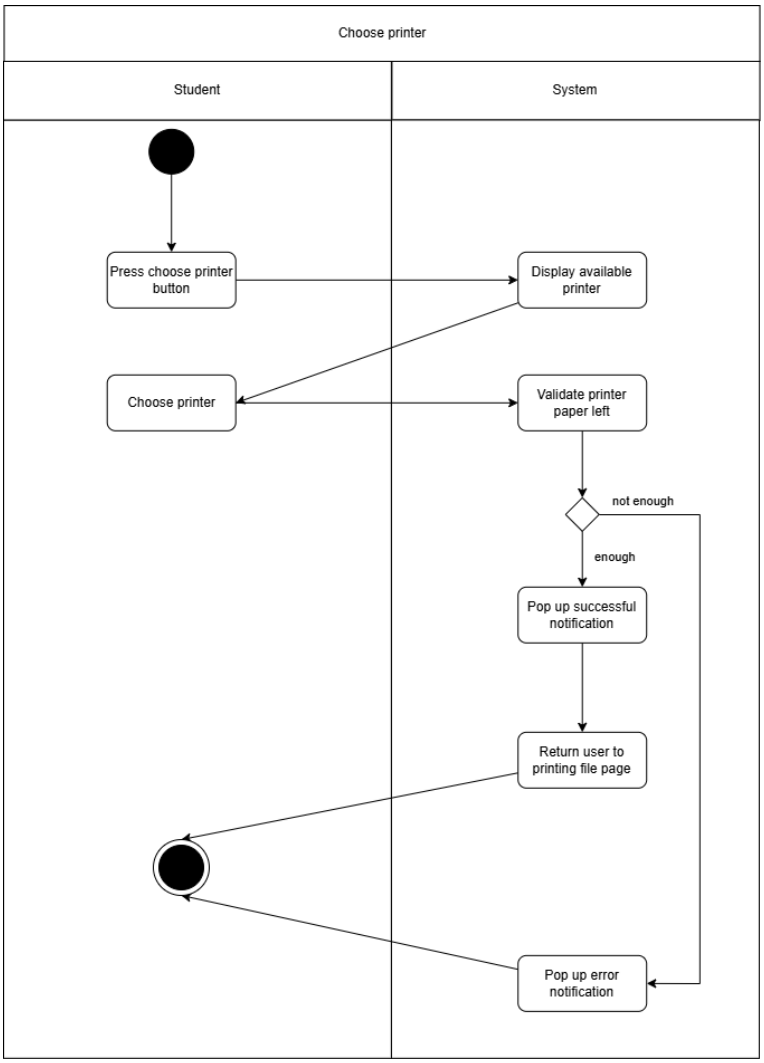
4.1.2 Configure File



Hình 2: Activity Diagram for Configure File

If the file type is supported, the system displays a file preview and allows the user to configure specific printing properties, including margin size, selected pages, paper sides (single or double-sided), print ratio, number of copies, and orientation. Each property has default settings, and the system ensures there are no conflicts between the selected properties and the uploaded file. The user can only proceed with submission when all properties are valid and compatible with the file. After the user submits the file with the selected properties, the system checks the student's paper balance to ensure they have enough pages for printing. If the balance is insufficient, the system displays an error message and redirects the user to the **BK Pay** system to purchase additional paper. If the user chooses not to buy more paper, they are sent back to the file preview and properties selection screen, where they can adjust the print settings to match their remaining paper balance. If the paper balance is sufficient, the system will pop up a successful setting of properties and redirect the user to the **Print Document** page.

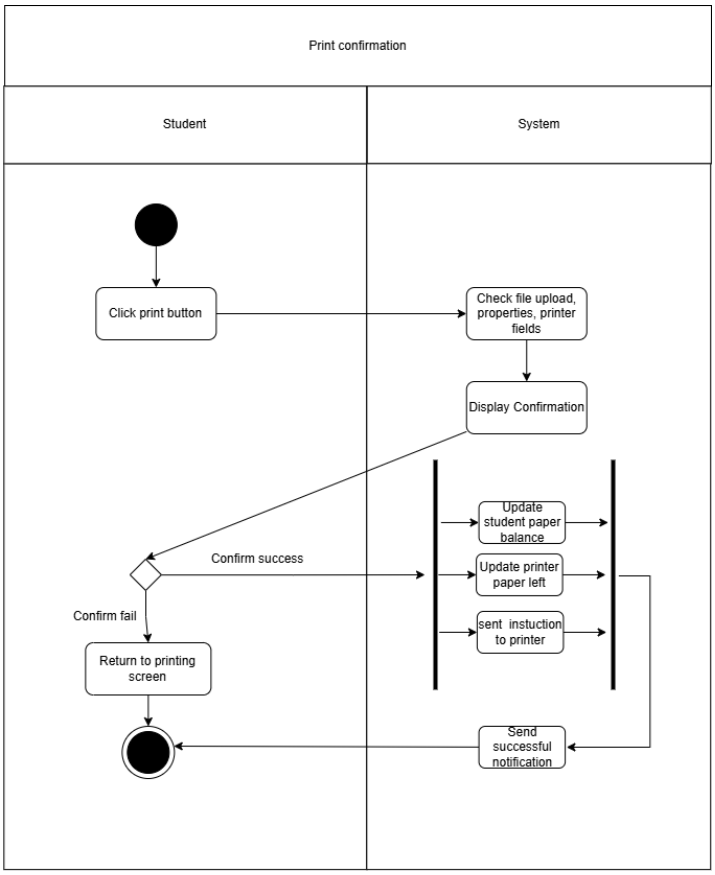
4.1.3 Select Printer



Hình 3: Activity Diagram for Select Printer

After setting the file properties, the user can select the printer system, which displays a list of available printers. Once the user chooses a printer, the system checks the remaining paper in the selected printer. If the selected printer does not meet the print properties requirements, the system will pop up an error message and return the user to the available printer.

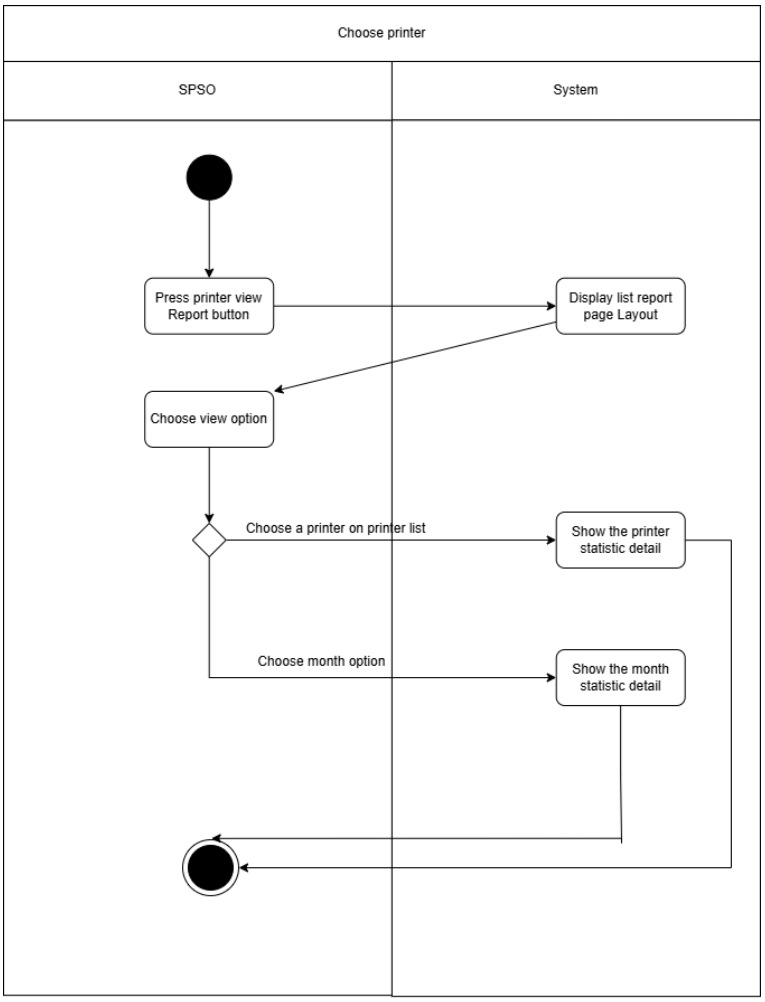
4.1.4 Confirm Print



Hình 4: Activity Diagram for Confirm Print

After completing the print setup, when the user clicks the **Done** button, the system checks if any required fields are incomplete. If everything is filled out, a confirmation pop-up appears, asking the user to verify the selected printer and print settings. If the user confirms, the system updates the printer's paper balance and sends the print instructions to the chosen printer, completing the process. If the user chooses not to confirm, they are given the option to either proceed with the printing or cancel it. If they decide to continue, the system redirects them back to the file preview and properties selection screen, allowing them to make any necessary adjustments before finalizing the print job.

4.1.5 Manage Printers by SPSO

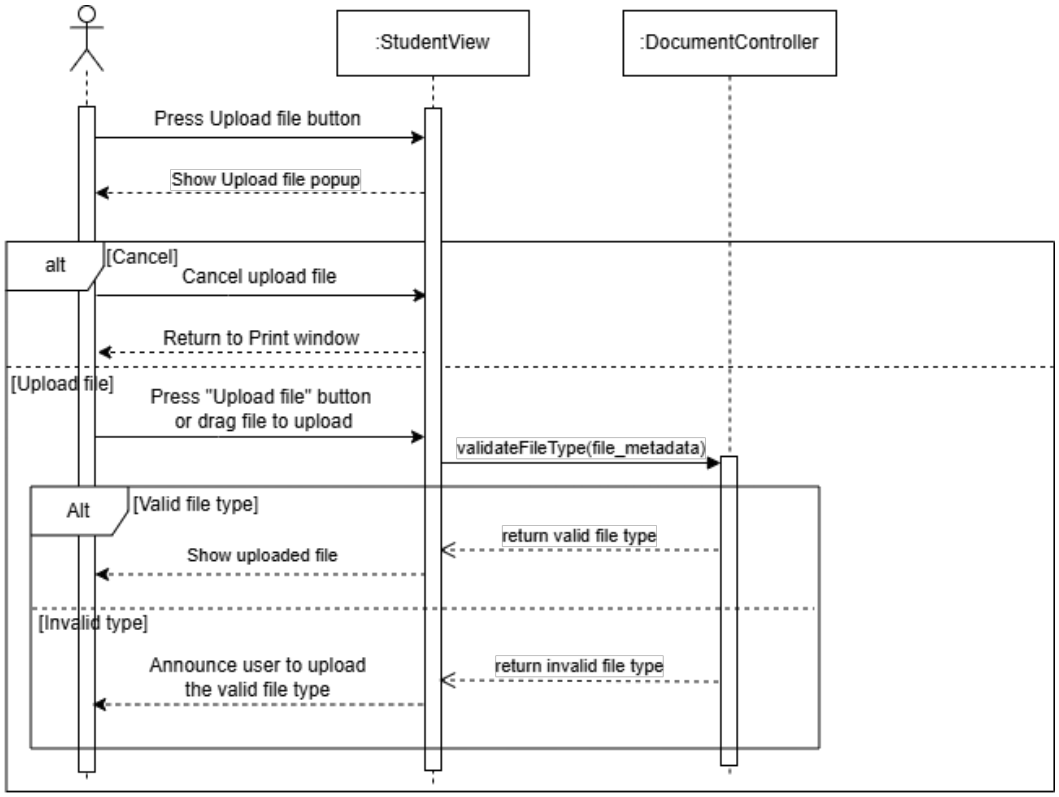


Hình 5: Activity Diagram for Manage Printers by SPSO

After logging into the SPSO account, the SPSO clicks on the "Report" button in the Navbar, and the system displays the report page with default print statistics, including a printer list and usage data for the last six months. If the user then clicks on a specific printer, the system navigates to that printer's statistics page, displaying details such as printer usage and the number of sheets of paper remaining. Alternatively, if the SPSO selects the monthly range report option, the system will display the report for the specified month range.

4.2 Sequence Diagram

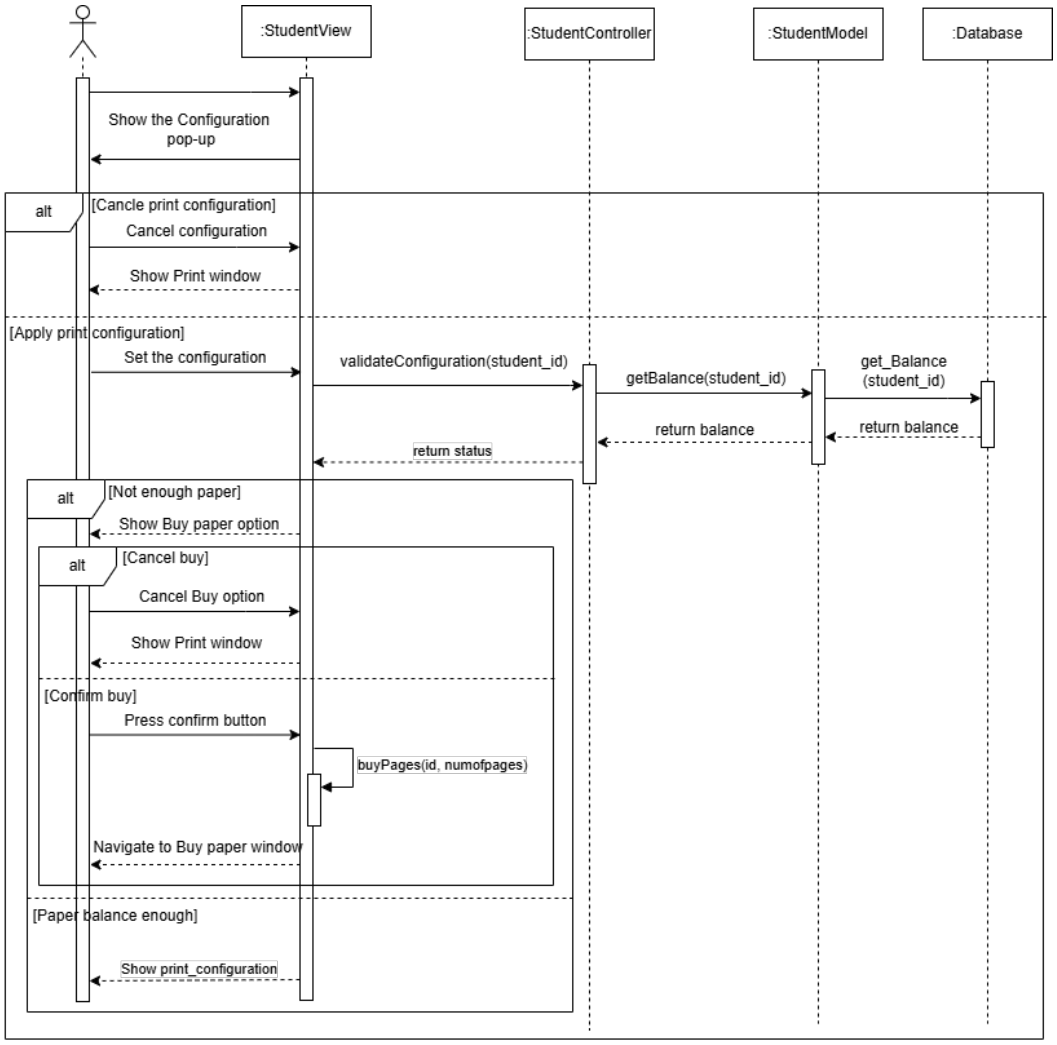
4.2.1 Upload File



Hình 6: Sequence Diagram for Uploading File

When the student clicks on the "Upload file" option in the "Print Document" interface, the system will display a pop-up for the user to upload a file for printing. If the student clicks the "Cancel" button, the system will return to the "Print Document" interface. The student can either drag and drop a file or click and select a file to upload. Once the upload is complete, **StudentView** will call **DocumentController** to check the file type through the **validateFileType(file_metadata)** function, and **DocumentController** will return the status to **StudentView**. If the file is in the correct format, the "Print Document" interface will display the uploaded file. If the file is not in the correct format, a message will inform the student of the valid file types.

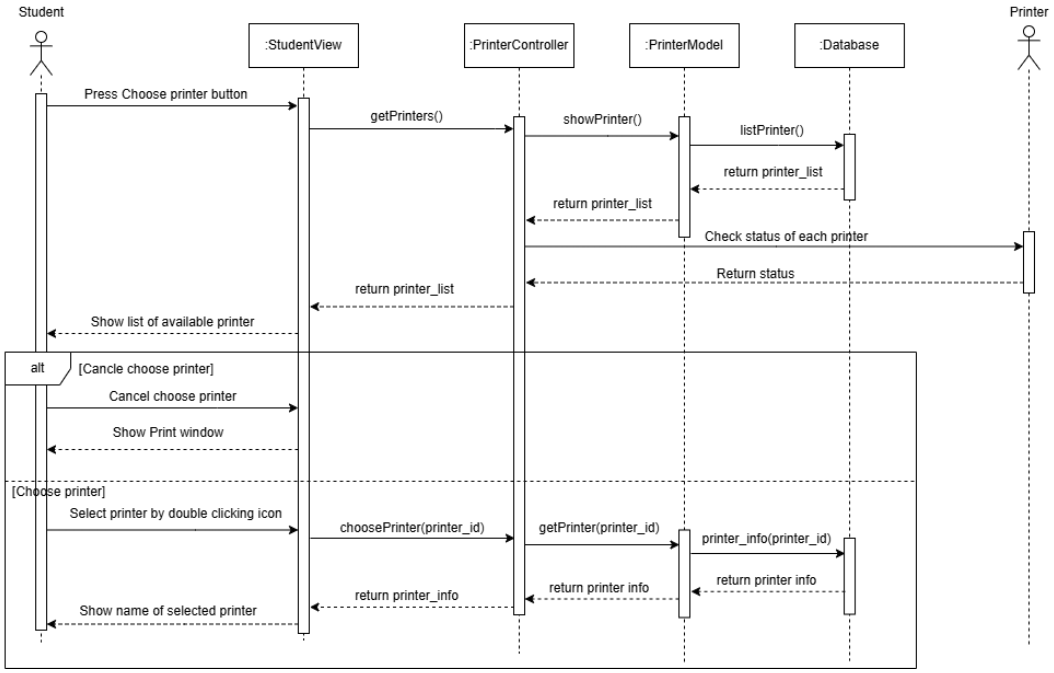
4.2.2 Configure File



Hình 7: Sequence Diagram for Configure File

When the student clicks on the "Set up print properties" option, the system will display a pop-up for the student to configure the file. If the student clicks the "Cancel" button, the system will return to the "Print Document" interface. If the student configures the file and clicks the "Done" button, **StudentView** will call **StudentController** to execute the **validateConfiguration(student_id)** function to check if the student has sufficient paper balance to print the chosen number of pages. **StudentController** will call **StudentModel** to execute the **getBalance**(student_id) function, and **StudentModel** will call the Database to retrieve the student's balance through the get_Balance(student_id) function. The paper balance data will be returned sequentially through each call. If the student does not have enough paper to print, **PrinterView** will display a message to the student. The student can choose not to buy more paper by clicking the "Cancel" button, and the system will return to the "Print Document" interface. If the student chooses to confirm the purchase, **StudentView** will call the **buyPages(id, numberofpages)** function to navigate to the "Buy Paper" window. If the student has enough paper to print, the application will display the print configuration in the "Print Document" interface.

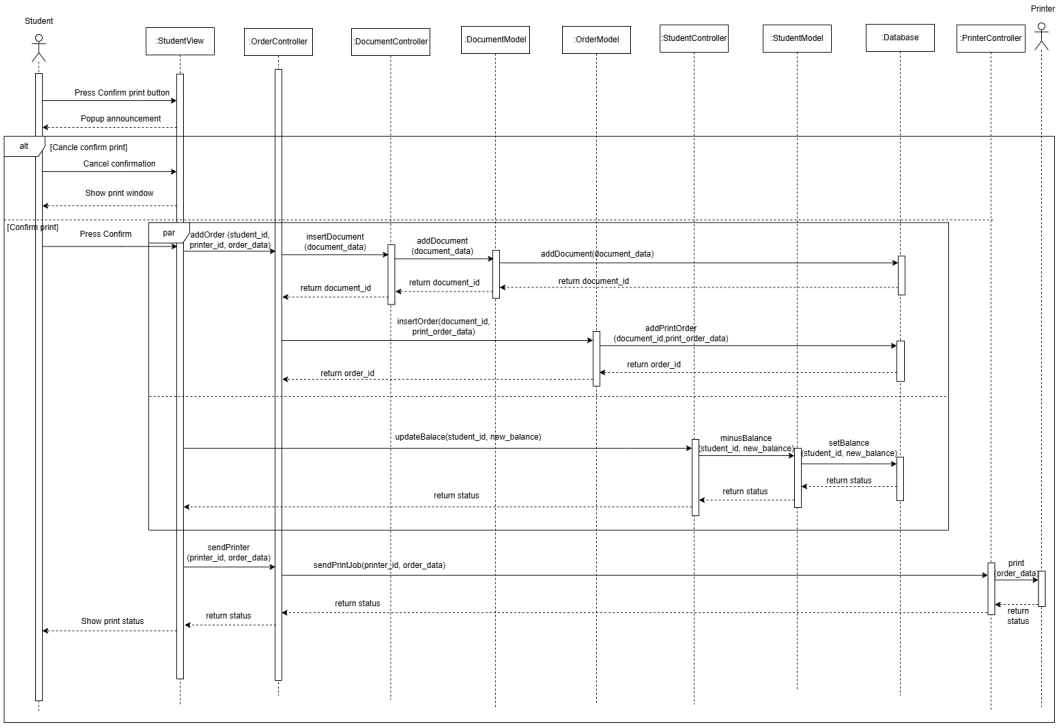
4.2.3 Select Printer



Hình 8: Sequence Diagram for Select Printer

When the student clicks on the "Choose printer" option in the "Print Document" interface, the system will display a pop-up for the user to select from the available printers. **StudentView** will call **PrinterController** to execute the **getPrinters()** function to retrieve the list of printer information. **PrinterController** will call **PrinterModel** to execute the **showPrinter()** function, and **PrinterModel** will call the Database to run the **listPrinter()** function. The list of printer information will be returned sequentially through each call. **PrinterController** will be connected to the printers to check if they are currently being used by other students. **StudentView** will then display the list of available printers for the student to choose from. If the student clicks the "Cancel" button, the system will return to the "Print Document" interface. If the student selects a specific printer, **StudentView** will call **PrinterController** to execute the **choosePrinter(printer_id)** function to retrieve information about the selected printer. **PrinterController** will call **PrinterModel** to execute the **getPrinter(printer_id)** function, and **PrinterModel** will call the Database to run the printer_info() function. The printer information will be returned sequentially through each call. **StudentView** will display the "Print Document" interface with the selected printer's information.

4.2.4 Confirm Print



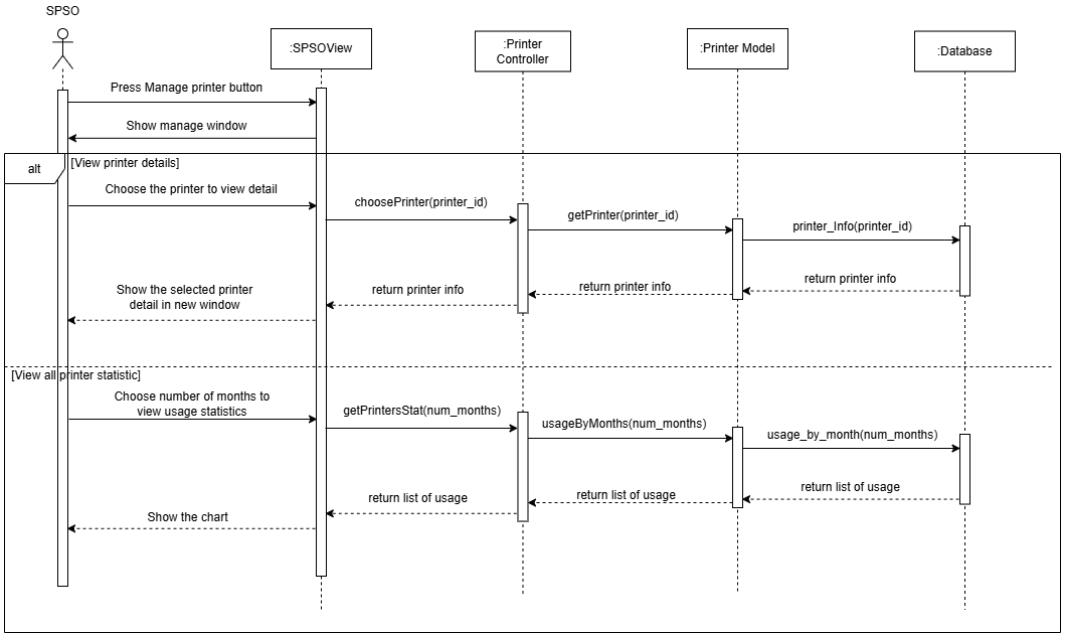
Hình 9: Sequence Diagram for Confirm print

When the student clicks on the "Print" option in the "Print Document" interface, the system will display a pop-up to confirm the print information. If the student clicks the "Cancel" button, the system will return to the "Print Document" interface. If the student clicks the "Confirm" button, two actions occur in parallel:

1. **StudentView** will call **OrderController** to execute the **addOrder(student_id, printer_id, order_data)** function. **OrderController** requires the document_id, so it will call **DocumentController** to execute the **insertDocument(document_data)** function. **DocumentController** then calls **DocumentModel** to execute the **addDocument(document_data)** function, and **DocumentModel** calls the Database to add the document via the **addDocument(document_data)** function. The order_id is returned once the document is successfully added to the **Database** and is passed back through each call. When **OrderController** receives the document_id, it calls **OrderModel** to execute the **insertOrder(document_id, print_order_data)** function. **OrderModel** then calls the **Database** to execute **addPrintOrder(document_id, print_order_data)**. The order_id is returned if the insert operation is successful.

2. **StudentView** will call **StudentController** to execute the **updateBalance(student_id, new_balance)** function. **StudentController** then calls **StudentModel** to execute **minusBalance(student_id, new_balance)**, and **StudentModel** calls the **Database** to update the student's paper balance using the **setBalance(student_id, new_balance)** function. After completing the database updates, **StudentView** will call **OrderController** to execute the **sendPrinter(printer_id, order_data)** function. **OrderController** then calls **PrinterController** to execute the **sendPrintJob(printer_id, order_data)** function. **PrinterController** sends the print job to the selected Printer using the **print(order_data)** function. The Printer returns the print status to **PrinterController**, and the print status is passed back through each function call until it reaches **StudentView**, where it is displayed to the student.

4.2.5 Printer Report by SPSO

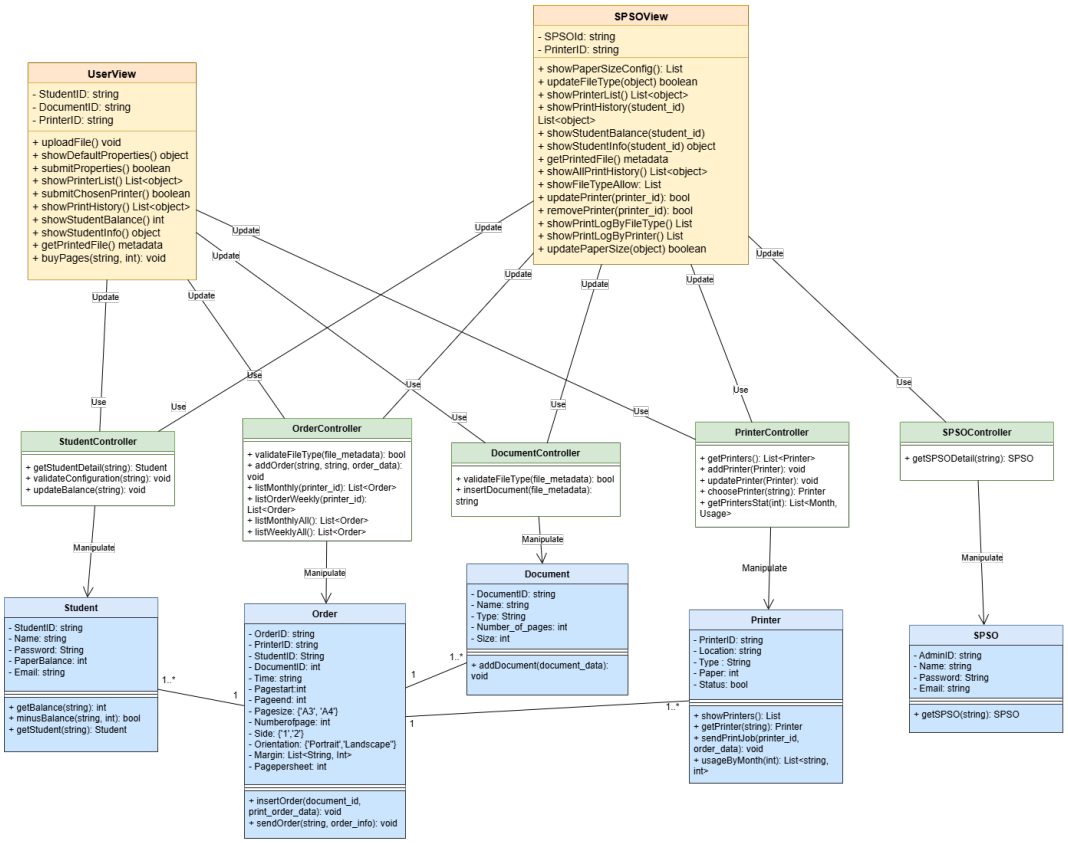


Hình 10: Sequence Diagram for Manage Printers by SPSO

When the **SPSO** clicks the "Printer Report" button on the main interface, the system will display the "Printer Report" interface, which contains information about all printers used in the application. If the **SPSO** wants to view details of a specific printer, they will select the printer, and **SPSOView** will call **PrinterController** to execute the **choosePrinter(printer_id)** function to retrieve information about the selected printer. **PrinterController** will call **PrinterModel** to execute the **getPrinter(printer_id)** function, and **PrinterModel** will call the Database to run the printer_info() function. The printer information is returned sequentially through each call, and **SPSOView** will display an interface containing the details of the printer. If the **SPSO** wants to view printer usage statistics, they will select the type of chart they wish to view. **SPSOView** will call **PrinterController** to execute the **getPrintersStat(num_months)** function to retrieve a statistical chart for the selected time interval. **PrinterController** will call **PrinterModel** to execute the **usageByMonths(num_months)** function, and **PrinterModel** will call the **Database** to run the **usage_by_month(num_months)** function. The statistical chart data is returned sequentially through each call.

4.3 Class Diagram

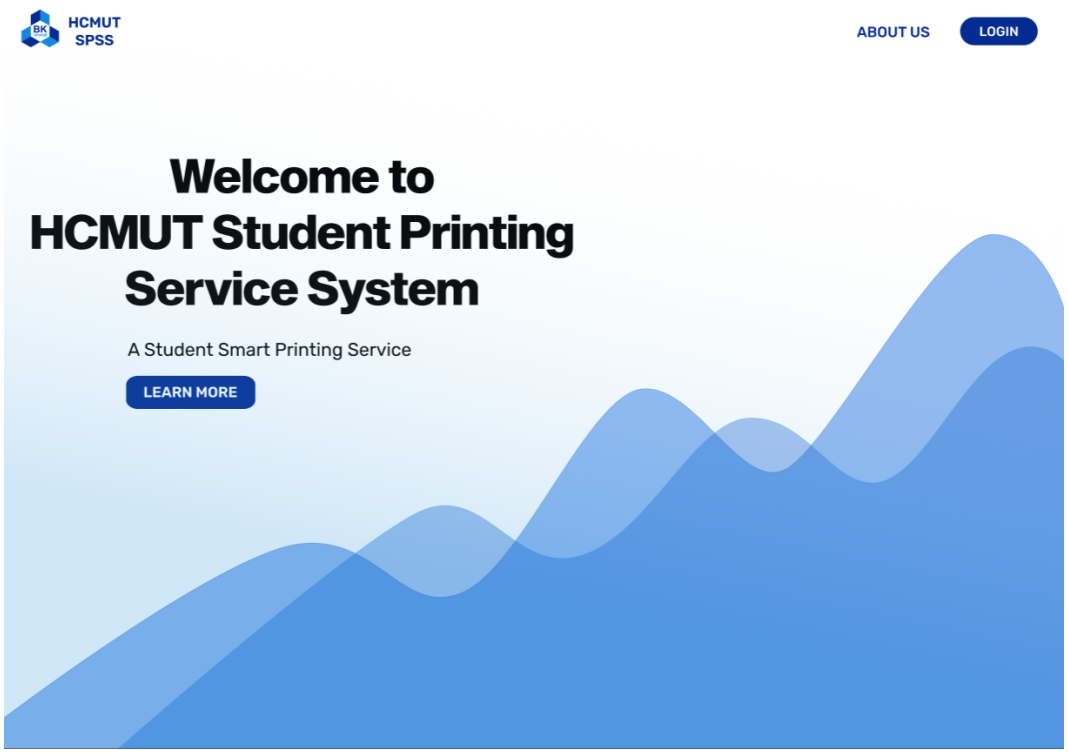
Link to the Class Diagram is available here: Here!



Class Diagram

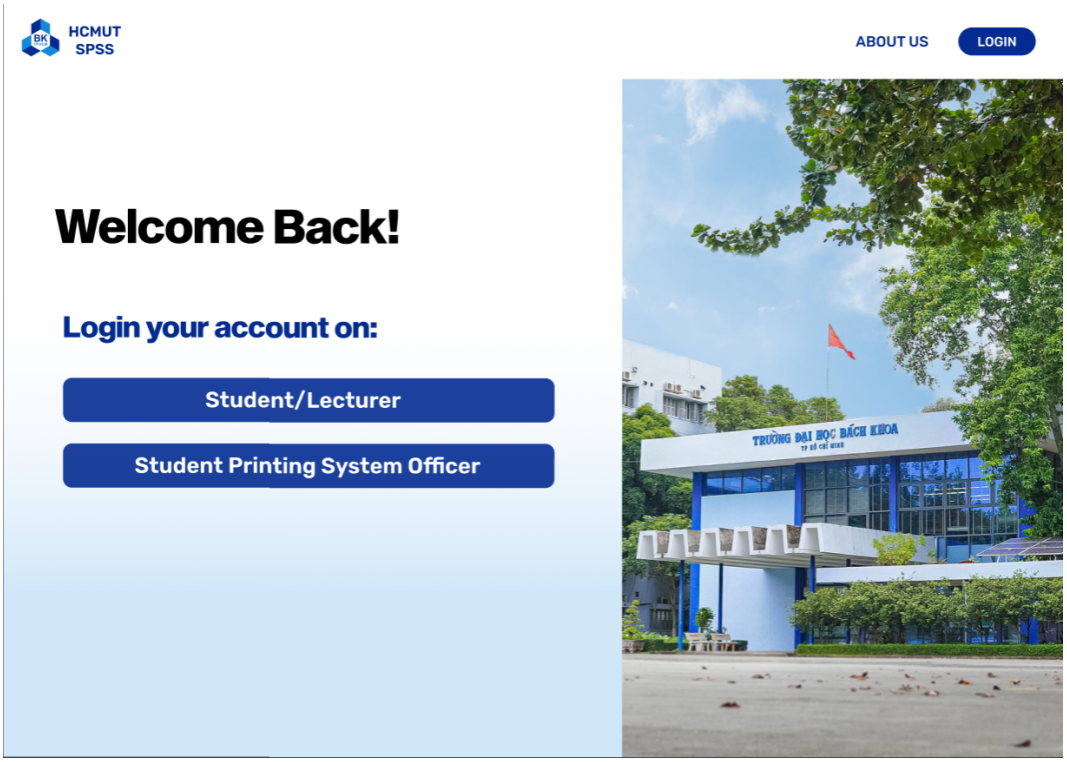
4.4 User Interface

Link to the Figma reference is available here: Figma Link.



Homepage

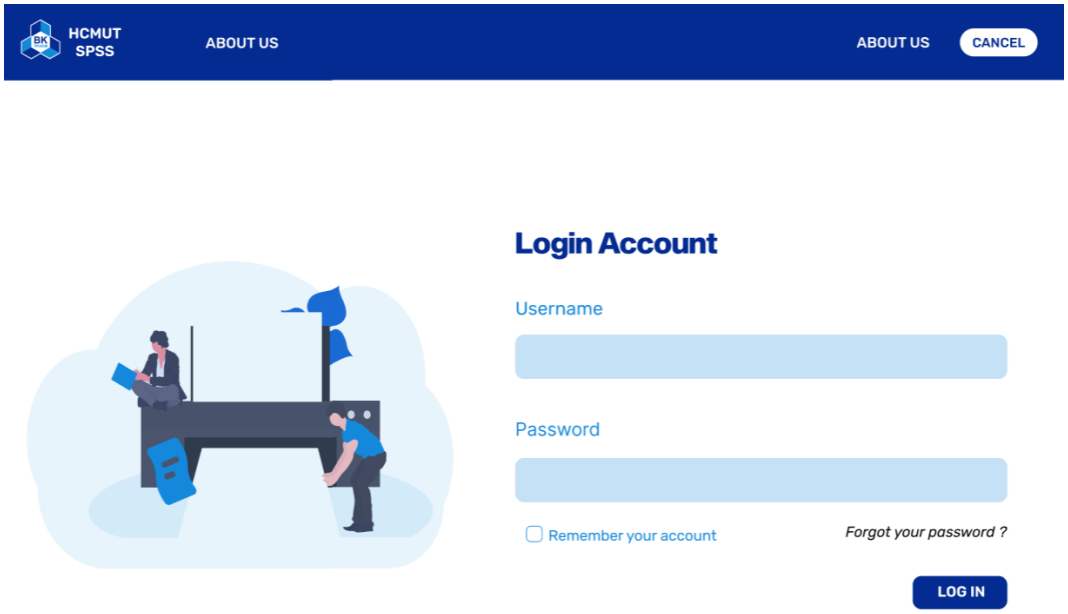
When accessing the website, users (both students and SPSO) will see the homepage screen. At this point, clicking the login button will lead to the desktop screen displaying two options: SPSO or students/lecturers.



Login Page

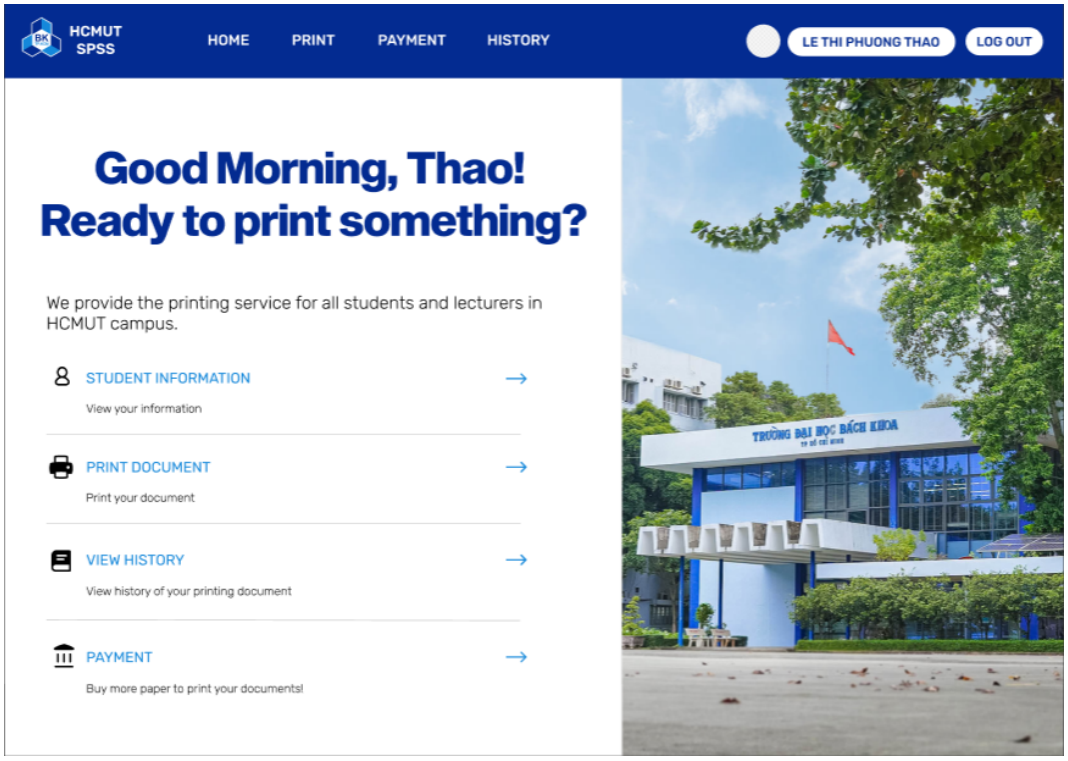
Users and SPSO will select their respective role to access their personal page.

4.4.1 Student



Student Login Page

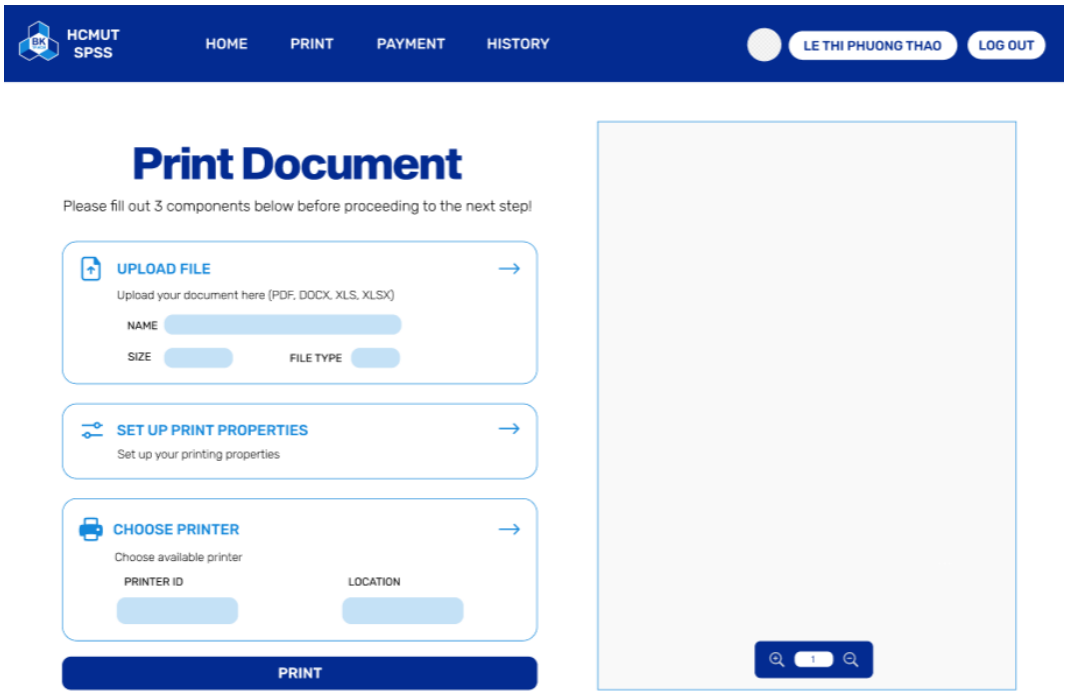
The student will enter their username and password to access their corresponding personal page. There are two buttons available: one to remember the password and another to handle cases where the user forgets their password, linked to the HCMUT account of the university.



Student Home Page

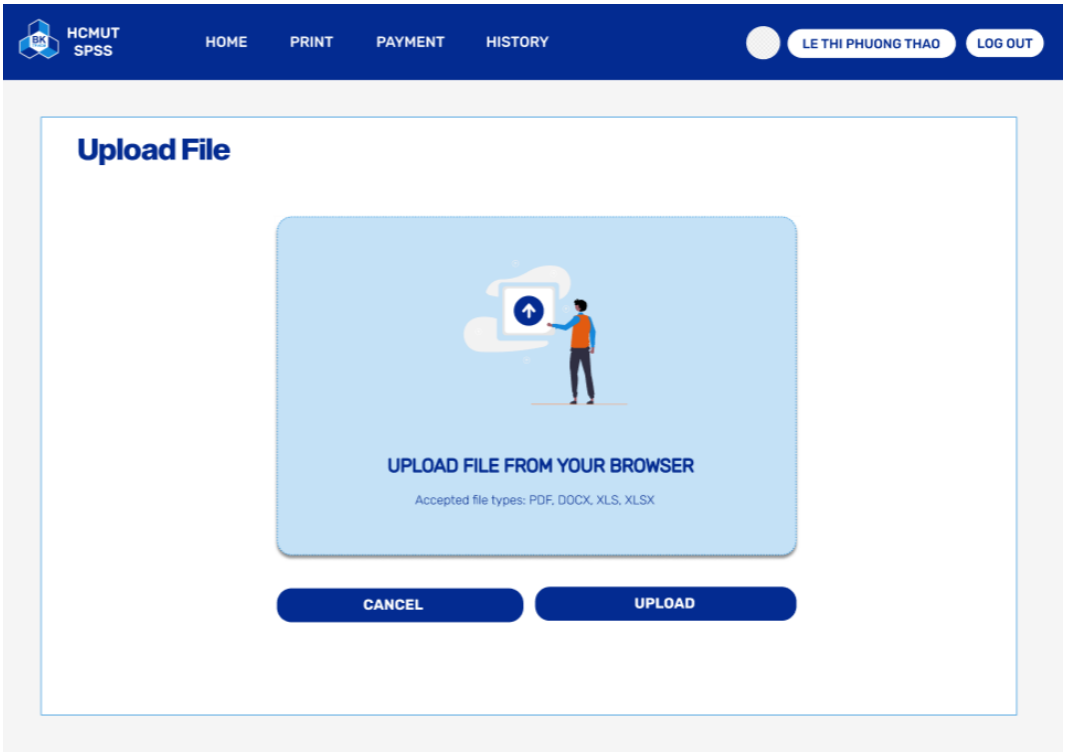
When users log into their account, they access a main page featuring four functions:

- **Student Information** - Allows users to view and update their personal information, set print properties and print documents, view their history (including files and payment), and make payments if they run out of printing credit.
- **Print Document** - Enables users to upload documents, set printing properties, upload files and select a printer for document printing.
- **View History** - Provides access to a record of previous actions, such as past print jobs, login sessions, and other account activities.
- **Payment** - Offers options for viewing and managing payment history, tracking balances related to printing and other services, and making payments when funds are low.



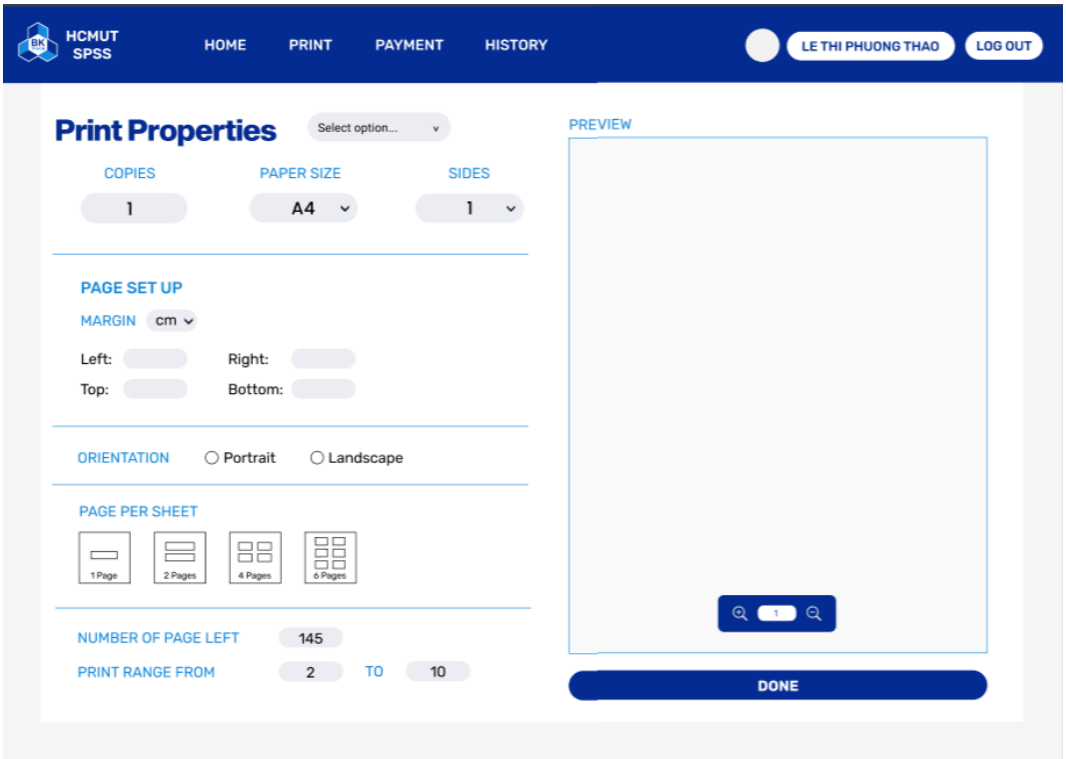
Print Document

When clicking on the 'Print Document' box, users are taken to a page that requires them to complete three actions: upload printed files, set up print properties, and select the appropriate printer. A preview of the file is displayed on the left, showing the name and file type if a file has been uploaded. Note that once each action is completed, users will return to this page. There is a 'Print' button available to finalize printing after completing all three actions



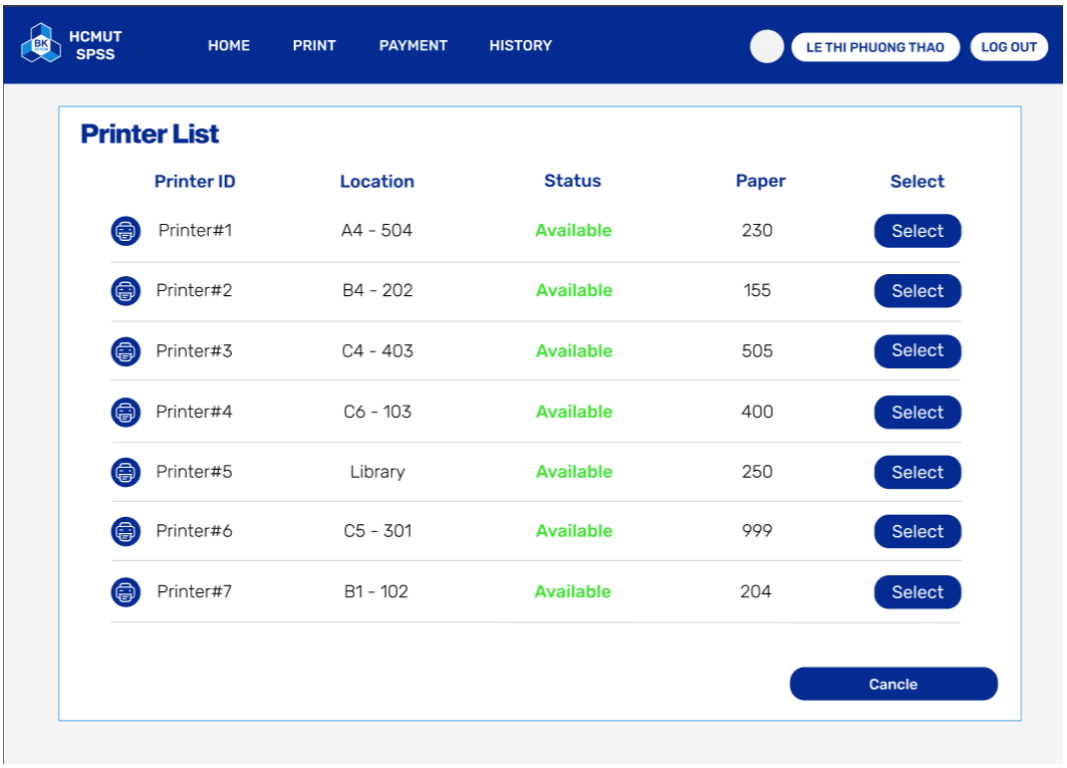
Upload File

The user can upload a file with the appropriate file type using the 'Upload' button. If the user decides not to proceed with printing, they can click the 'Cancel' button. Students can choose to upload their files at any time.



Print Properties

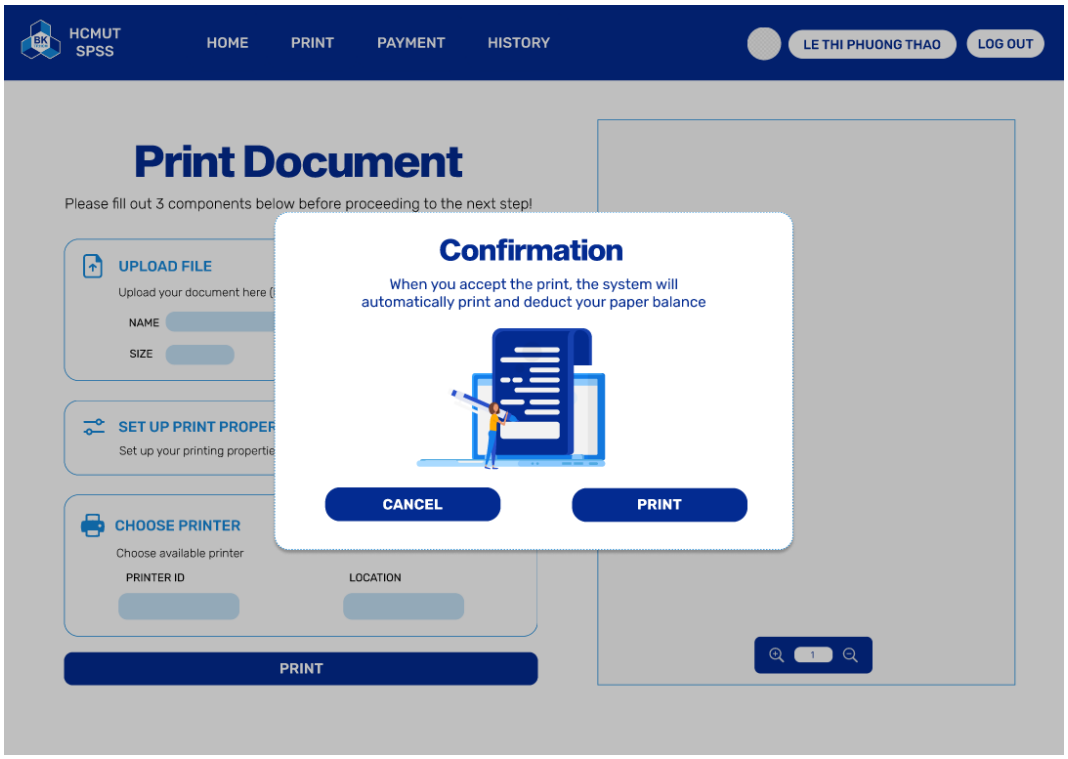
Print properties allow users to set the paper type, number of copies, and format of the printed file. A preview of the file is provided, allowing users to view and adjust these settings as needed.



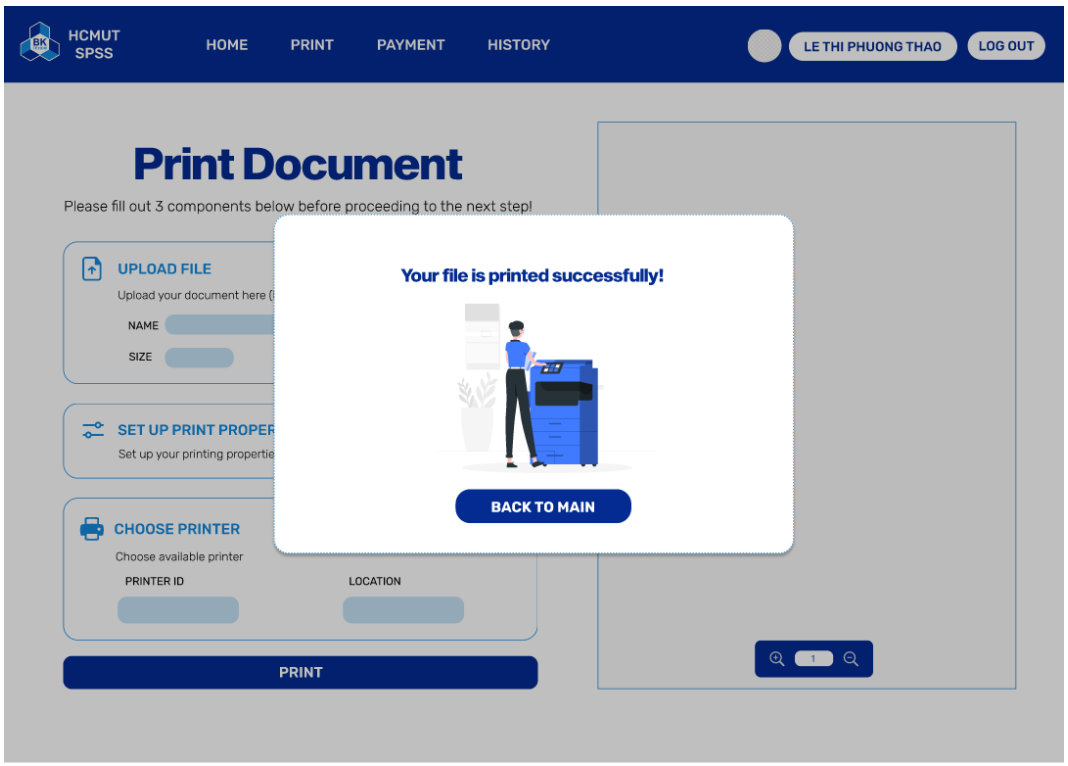
Select Printer

The printer list displays the printer ID, location, status, available paper, and a selection option. Based on the printer's status, the user can select the appropriate printer. After making a selection, the user can click the 'Cancel' button to return to the 'Print Document' page or choose to leave without making a selection.

After completing the three actions in the 'Print Document' section, when the user clicks the 'Print' button, a pop-up will appear with 'Print' and 'Cancel' options. It will ask users if they are sure they want to print the document with the current settings. If they confirm by clicking 'Print,' a pop-up will notify them of a successful print. If they choose to cancel or if an error occurs during printing, another pop-up will display the error message.

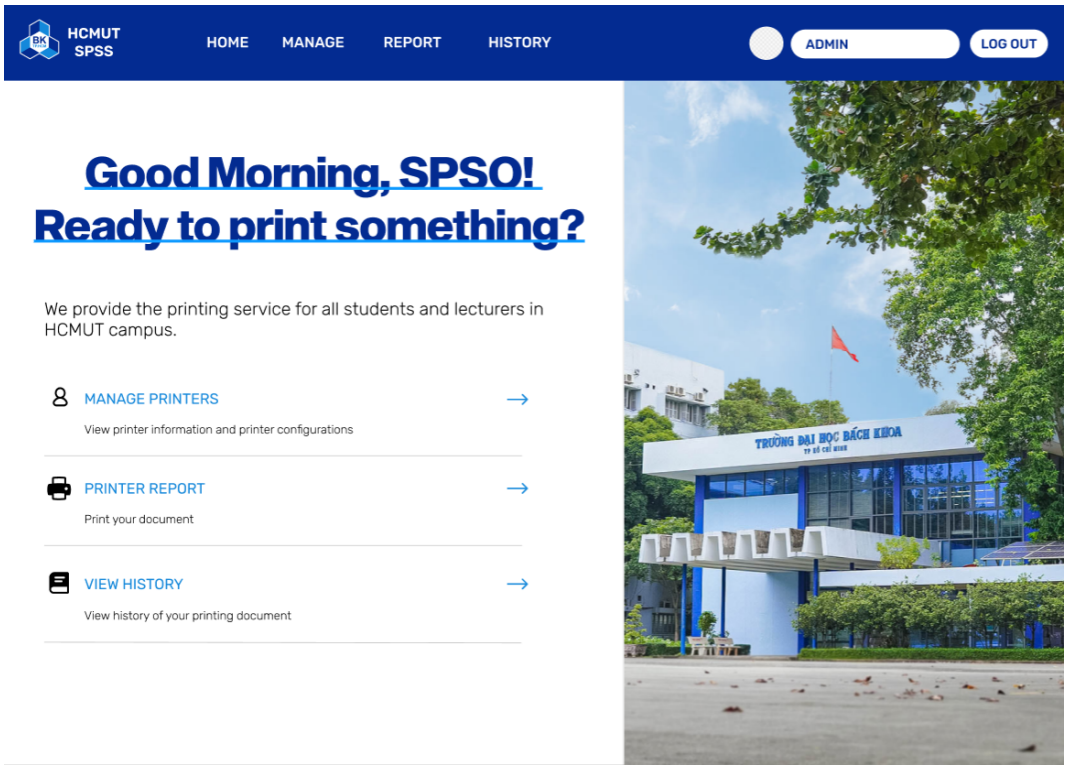


Confirm Print



Print Successfully

4.4.2 SPSO



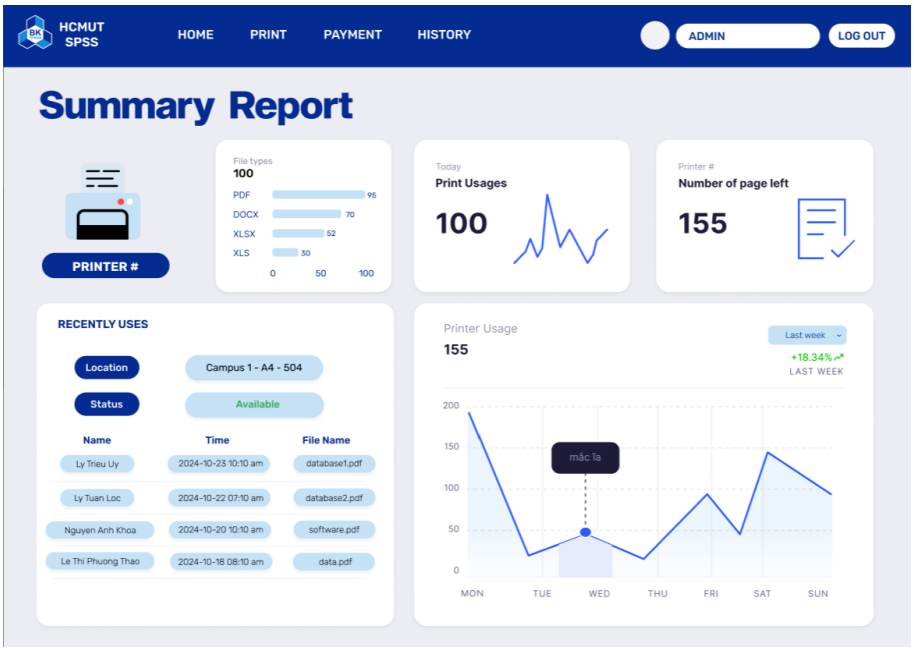
SPSO Homepage

When the SPSO selects their respective role, they will be directed to their admin homepage, which displays three options:

- **Manage Printers**: Enables or disables printers, and allows changes to configurations and settings.
- **Printer Reports**: Allows checking the reports of each printer as well as the overall system.
- **View History**: Provides access to the history of all system activities and each student's interactions.

Printer Report

The Printer Reports section will display all usage statistics, currently active users, transaction details for print requests, and a list of printers on campus. Each printer will have a 'View' button that leads to a detailed report for that specific printer.

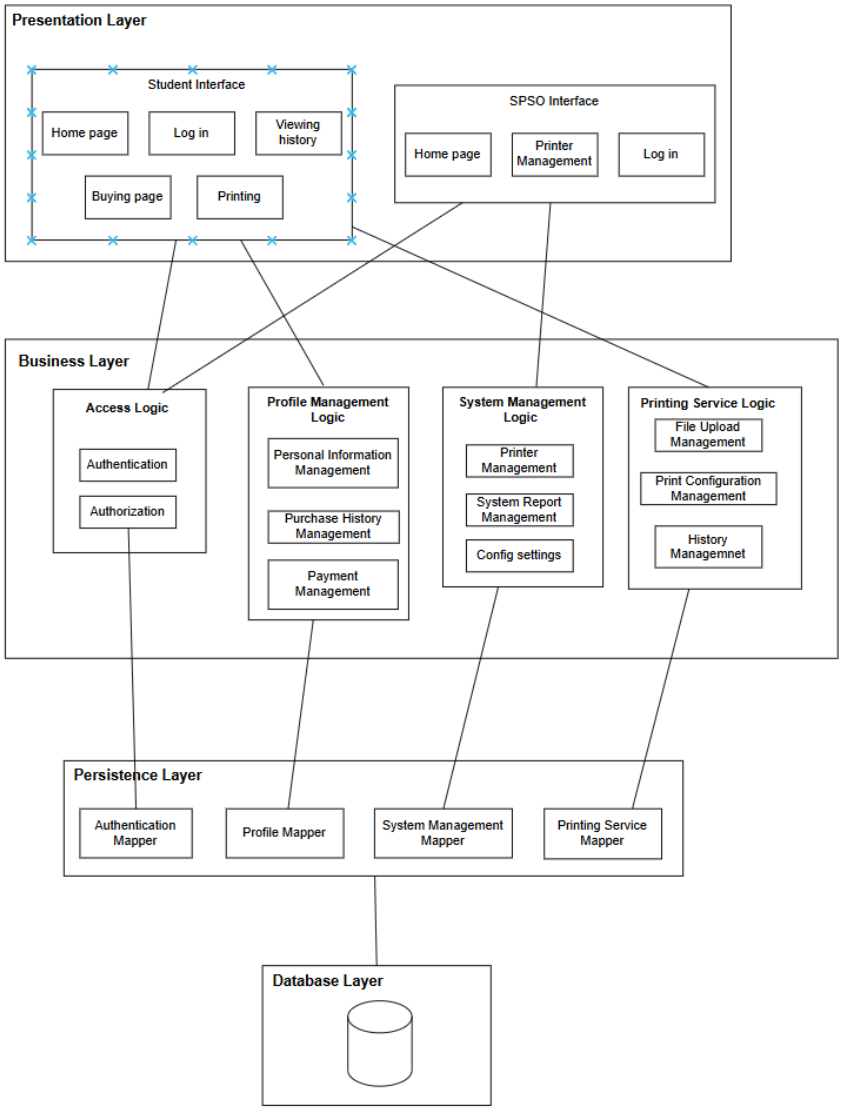


Summary Report

The summary report will display specific information, including the history, usage statistics, and printed files for specific printer.

5 Architecture Design

5.1 Layered Architecture



Hình 11: Layer Architecture Diagram

5.2 Presentation Strategy

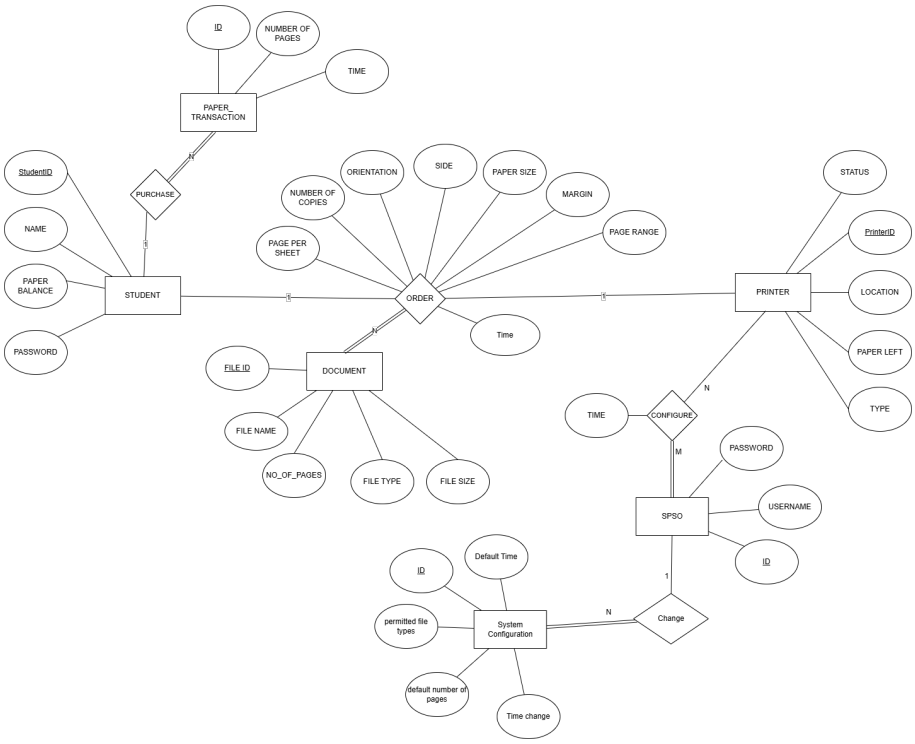
This is the first layer in the architecture of the HCMUT-SSPS system. We will adopt a strategy that prioritizes simplicity, ease of use, and an exceptional user experience. The goal is to create an intuitive and accessible interface that ensures seamless interactions with the system, regardless of the user's technical expertise. To achieve this, we will utilize several modern technologies and design principles, specifically:

- **Front-end library and framework**: We will use React as the core framework for developing the front-end of the application. React is known for its component-based architecture, allowing us to build reusable UI elements, which increases both development efficiency and scalability. React's virtual DOM enhances performance by optimizing updates to the user interface, ensuring the application remains responsive even under heavy load. Additionally, React's extensive ecosystem, including libraries like React Router for navigation and Redux for state management, will allow us to build a highly dynamic, maintainable, and performant front-end.
- **Responsive Design**: Ensuring accessibility across a wide range of devices is a key priority. The

system will be built with responsive design principles to ensure that it functions seamlessly on desktops, mobile devices, and tablets. We will use CSS frameworks like Bootstrap or Tailwind CSS to implement flexible layouts that adapt to different screen sizes and orientations. This will allow students and faculty to access the system from any device they prefer, whether they are in the computer lab or on the go with their smartphones. Responsive design will ensure that the system is equally user-friendly on large screens and mobile screens, providing a consistent and optimized experience for all users.

- **User-Friendly Features**: A major focus will be placed on creating a visually appealing and intuitive interface. We will design elements like buttons, forms, and menus that are simple to navigate, even for users who are not familiar with technology. Features like tooltips, clear labeling, error messages, and progress indicators will provide immediate feedback, reducing any confusion. For new users, a guided walkthrough or tutorial can be incorporated to help them quickly understand how to submit print jobs, check print balances, and view their printing history. The goal is to eliminate any barriers to using the system and make the user experience as straightforward as possible.

5.3 Data storage approach



Hình 12: Entity Relation Diagram

In layered architecture, the database resides at the bottom layer, responsible for storing and processing all data. Application data is stored here, and operations such as retrieve, insert, update, and delete are frequently performed to manipulate data through the database management system. For the Smart Printing Service project, the team will employ layered architecture with the database layer stored using a relational database and MySQL as the database management system. This means that application data will be stored as tables and their relationships. For the Smart Printing Service system, we need to define the following entity types:

- Student: StudentID, name, paper balance, password.
- Paper transaction: ID, Time, Number of pages, Student ID.
- Order: Student ID, Printer ID, Document ID, Time, page range, margin, paper size, side, orientation, page per sheet, number of copies.

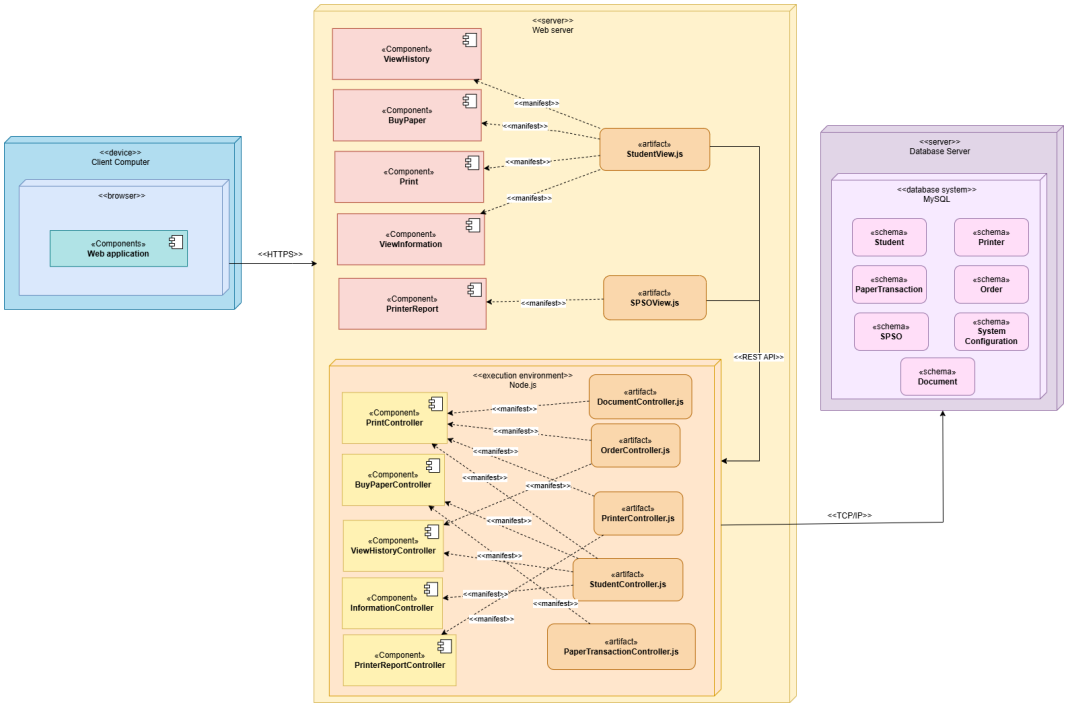
- Document: File ID, file name, number of pages, file type, file size.
- Printer: Printer ID, Location, paper left, type, status.
- SPSO: ID, Username, password.
- System Configuration: ID, SPSO ID, time change, default number of pages, permitted file types, default time.
- Printer Configure: Printer ID, SPSO ID, time.

5.4 API Management

An **API (Application Programming Interface)** is a set of rules and protocols that allows different software applications to communicate with each other. It enables systems to share data, functionality, and services without needing to understand the internal workings of each other. For instance, in a web-based system, an API allows one application to request and retrieve information from another application over the internet. For a system like HCMUT SPSS, critical features like authentication, access control, performance monitoring, and error handling are necessary to protect sensitive data and ensure smooth communication between services and that is why API management comes into play.
API management is the process of creating and publishing web application programming interfaces (APIs), enforcing their usage policies, controlling access, nurturing the subscriber community, collecting and analyzing usage statistics, and reporting on performance. API Management components provide mechanisms and tools to support developer and subscriber communities. The API needed for HCMUT SPSS consist of:

- **API Security and Authentication**: APIs ensure secure access to the system by enforcing authentication via HCMUT-SSO (Single Sign-On), so students must log in using their university credentials. API management also implements access control through the SPSO, determining which users (students, staff, or administrators) can perform specific tasks like managing printers or viewing reports.
- **Input Data Formatting and Processing API**: This API handles input validation by ensuring that documents uploaded for printing meet specific format and quality standards. API management oversees the process to ensure that these APIs consistently maintain proper formatting rules and document compatibility, preventing errors during print jobs.
- **Print Job Management API**: The Print Job API allows students to create, manage, and track their print jobs, such as choosing a printer, setting properties like paper size, and monitoring job progress. API management monitors the performance of these APIs to ensure that they handle requests efficiently and keep track of job statuses in real-time. It also ensures print jobs are executed within the allowed page limits.
- **Payment API**: The Payment API connects the printing system to BKPay, enabling students to purchase additional print pages or pay for print jobs. API management ensures the security and reliability of these transactions, tracking payments and integrating with the student's printing account to update page balances accordingly.
- **Custom Print Template API**: This API allows students or staff to create custom print templates, which can be particularly useful for printing documents with specific layouts (e.g., posters or reports). API management monitors these templates to ensure they follow the system's rules and standards, while also ensuring that they are dynamically generated based on user needs.
- **Printer Manage API** The Printer Manage API enables SPSO to interact with printers, including enabling/disabling printers, adding a new printer to the system or handling errors (like paper jams or low ink). API management ensures that printers across different campuses are accessible, functional, and properly configured.
- **Reporting and Statistics API**: This API provides detailed reports on print jobs, including statistics like the number of pages printed, printing time, and usage per printer. API management tracks this data to generate monthly or yearly reports for the SPSO, ensuring that the system's performance is regularly monitored and optimized.

5.5 Deployment Diagram

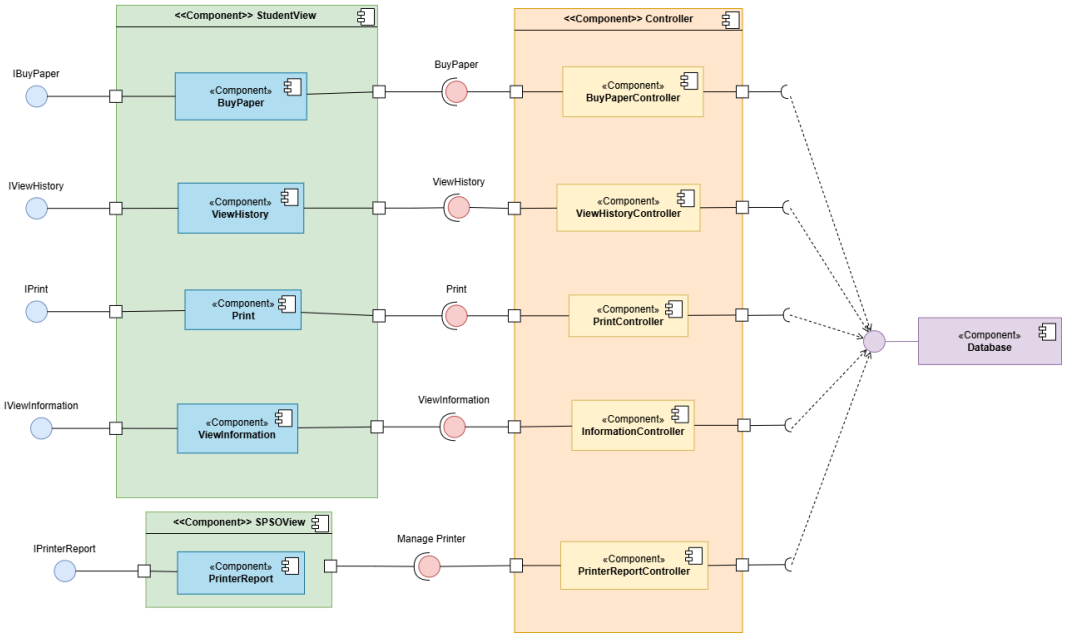


Hình 13: Deployment Diagram

The software is developed in a web application with MVC architecture.

- Client-Server Communication: The client will connect to the server via HTTPS protocol. The received data from the server will be interface realization files, which will be displayed on the user's browser.
- Server-Side Environment: The server-side execution environment consists of the system's main functional modules. The back-end execution environment is NodeJS, using the Express.js framework, and includes 3 main components. The front-end user interface is built using ReactJS library, and includes 3 main components. Components will be implemented from corresponding JavaScript source files. Through REST APIs, the front-end and back-end can interact with each other to update, display the interface, and receive/respond to user actions.
- Server-Database Communication: The server will connect to the database server via TCP/IP protocol to retrieve data for display on the interface or update data whenever the user interacts with the application. The updated data will then be transmitted via TCP/IP to the server and then to the client to update the interface.
- Database Server: The database server includes a MySQL relational database. The database server stores information in structured tables and their relationships.

5.6 Component Diagram



Hình 14: Component Diagram

Figure 14 is a **Component Diagram**, which is used to describe the structure of a software system by showing its main components, their relationships, and how they interact with each other.

5.6.1 Description of the Components in the Diagram

1. **StudentView:**
   - Includes the interface components designed for students, such as:
     - **BuyPaper:** Handles the functionality for purchasing paper.
     - **ViewHistory:** Allows viewing of history.
     - **Print:** Handles printing documents.
     - **ViewInformation:** Displays information to the user.
   - Each interface component is connected to a specific interface, for example: `IBuyPaper`, `IViewHistory`, `IPrint`, `IViewInformation`.
2. **SPSOView:**
   - Includes the **Printer Report** component, which handles printer management, connected to the `IPrinterReport` interface.
3. **Controller (yellow column):**
   - Represents the control layer, which manages requests from the interface components, including:
     - **BuyPaperController:** Controls the purchase paper functionality.
     - **ViewHistoryController:** Controls the view history functionality.
     - **PrintController:** Controls the printing functionality.
     - **InformationController:** Handles information display.
     - **ManagePrinterController:** Manages printer-related operations.
   - Each component in the **Controller** layer is linked to a corresponding component in **StudentView** or **SPSOView**.
4. **Database:**
   - Represents the data storage component, connected to all the components in the **Controller** layer through dashed lines.

5.6.2 Relationships Between Components

- **Interfaces:** The blue and red circles represent the input and output interfaces for the components.
- **Interaction:** Each component in the **StudentView** or **SPSOView** layer communicates with its corresponding component in the **Controller** layer. The **Controller** layer then connects to the **Database** to perform data-related operations.

5.6.3 Overall Functionality

The diagram clearly illustrates how the system processes user requests:

1. Users interact with the interface components (**StudentView** or **SPSOView**).
2. Requests are sent to the **Controller** layer for logic processing.
3. The **Controller** interacts with the **Database** to retrieve or store data as needed.