Software Requirements Specification for

A smart printing service for students at HCMUT

Version 1.0 approved

Prepared by:

Group CC03-02

1. **Ha Chi Quyen – 2152933**
2. **Ha Kien Hoa – 2252225**
3. **Ho Khanh Nam – 2252500**
4. **Hoang Duc Chinh - 2252093**
5. **Huynh Nga – 2252508**

Department of Software Engineering

Faculty of Computer Science and Engineering

Ho Chi Minh City University of Technology – VNU-HCM

22/09/2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Project 1.1 -1.2	22/9/2024	Update task 1.1 and 1.2	1
Project 1.3	1/10/2024	Update task 1.3	2
Project 2	27/10/2024	Update task 2	3
Project 3	17/11/2024	Update task 3 Changing 1.3 because we want to focus on printing module, as a result changing 2.1, 2.2, 2.3	4
Project 4	24/11/2024	Update task 4 Sửa lại UI trên figma theo review đã nhận được từ các khâu testing	5
Project 5	16/12/2024	Update task 1.1 and 1.	

1. Task 1: Requirement elicitation

1.1 Domain Context

The domain context of the Student Smart Printing Service (HCMUT-SSPS) for students at HCMUT revolves around providing an efficient, accessible, and modernized solution for students to manage their printing needs across the campus. This service aims to address the need for a centralized system where students can upload, manage, and print their documents at various locations within the university. Through both web and mobile platforms, the system integrates existing infrastructure, like online payments via BKPay, and utilizes the HCMUT Single Sign-On (SSO) service for secure authentication, ensuring easy access and secure operations. The system offers flexibility in terms of managing printer settings, printing properties, and students' account (page) balance, ensuring that students are well-served throughout their academic journeys.

1.2 Stakeholders and Needs

The relevant stakeholders of the HCMUT-SSPS include students, the Student Printing Service Officer (SPSO) and the university. Students need a reliable and easy to access printing system that allows them to print their documents, view their printing log, and manage their account (page) balance. The SPSO is responsible for managing the system's configurations, monitoring usage, and ensuring the availability of printers across the campus. The university seeks a sustainable, scalable solution that can monitor and control printing usage while integrating with existing systems like BKPay and HCMUT_SSO for security and transparency.

1.3 Benefits of the System

HCMUT-SSPS provides distinct benefits to each stakeholder. For students, it simplifies the printing process by providing an easy-to-use platform, giving them better control over their printing limits and payment options, and ensuring transparency through log viewing. The SPSO benefits from centralized management, reducing the administrative burden of manually configuring printers and monitoring usage. They can also ensure system integrity by adjusting page quota for each page size and file-type restrictions as needed. They gain insights through automated reports on printing usage, enabling them to make data-driven decisions about resource allocation and environmental impact, fostering a more efficient campus-wide printing system. On the other hand, the university benefits by being able to monitor and control the amount of paper used each semester through the system's ability to set default page quotas for students. Additionally, the feature allowing students to buy more printing pages via the BKPay system creates an additional revenue stream for the university. Through the HCMUT_SSO authentication service, the university can also track how many students actively use the HCMUT-SSPS, allowing for better planning and scaling of the service. This combination of financial, administrative, and operational control helps the university optimize its printing infrastructure while supporting sustainability initiatives and enhancing student services

1.4 Functional Requirements

a. For Students:

- Upload and print documents: students can upload a document file onto the system, choose a printer, and specify the printing properties (such as page size, pages (of the file) to be printed, one-/double-sided, number of copies, etc.)
- View printing log: a student can check his/her printing log for a time period together with a summary of the number of printed pages for each page size.
- Buy extra pages: Students can use the "Buy Printing Pages" features to buy more A4-size pages.
- Pay online: Students can pay the extra pages through BKPay system of the university
- Authentication: Students must log in by using the HCMUT Single Sign-On (SSO) to access the system with their university account.

b. For SPSO:

- Manage Printers:
  - Add, enable, disable a printer.
  - Edit printer details (e.g., description, location, operational status).
- View students' printing log: 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.
- Manage configuration system settings:
  - Change the default number of pages per semester and the dates that the system will give the default number of pages to all students.
  - Configure permitted file types.
  - Modify the conversion rate between A3 and A4 pages.
  - Manage log data, including correcting logged errors (if any) or marking erroneous records.
- View system's reports: The SPSO can view the reports of the using of the printing system can be generated automatically by the system at the end of each month and each year
- Temporarily pause or lock the printer: The SPSO can temporarily pause or lock a printer in case the printer is defective or the students print the number of pages that exceeds the default number.
-

c. For the university:

- Default page's allocation: For each semester, the university can give each user (student) a default number of A4-size pages for printing and the university can adjust the default number.
- Monitor printer's status: The university can monitor all printer and printing activity or status, as well as the number of papers available in every printer.
- View financial reports: the system can view financial reports that include how much payment that students pay for extra pages, the amount of budgets for fixing defective printers.
- Analysis printer's usage: The system can generate reports of analysis on printer usage trends, such as making a statistical analysis on the most-used printers, peak usage time or the total number of pages used in a day, etc., in order for the university to plan and optimize the printer placement
- Manage access permissions: The university can control access of users or groups (e.g., students, SPSO, etc.) to specific printers or services in the system

1.5 Non- Functional Requirements

a. Usability:

- User-friendly interface: the user interface of the system must be intuitive, aesthetic, efficient and easy for students to navigate and interact with the system
- User-error protection: the user errors (such as data entry error, navigation error, configuration error, etc.) must be handled while using the system by using error prevention mechanism: validation checks, error messages, troubleshooting, and confirmations alert users to double-check and correct mistakes before causing issues
- Software: available on web-based apps and mobile apps.

b. Performance efficiency:

- Quick Response Time: The document upload and print action should be processed quickly (The system should response to the printing request within 2-3 seconds)
- Reliability: The system should be reliable with minimal downtime to ensure constant printer availability in the university
- Precise: The service should allow students to print equal or less pages than their limitation.

c. Security:

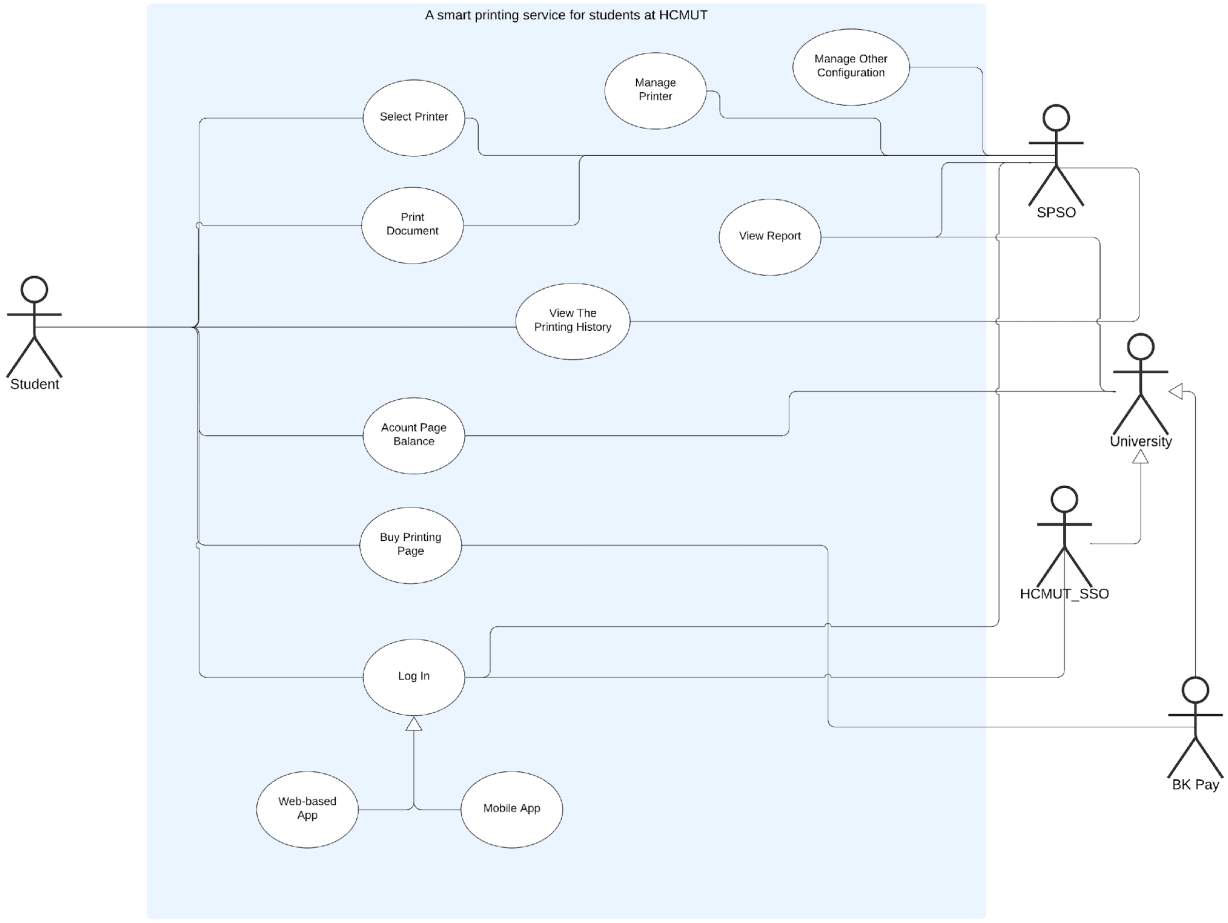
- Data privacy: All students information should be securely handled by the HCMUT_SSO system and not accessible by unauthorized users
- Accessibility: The system will ensure that some limited features of printing management and configuration settings are permitted for only SPSO.

d. Maintainability:

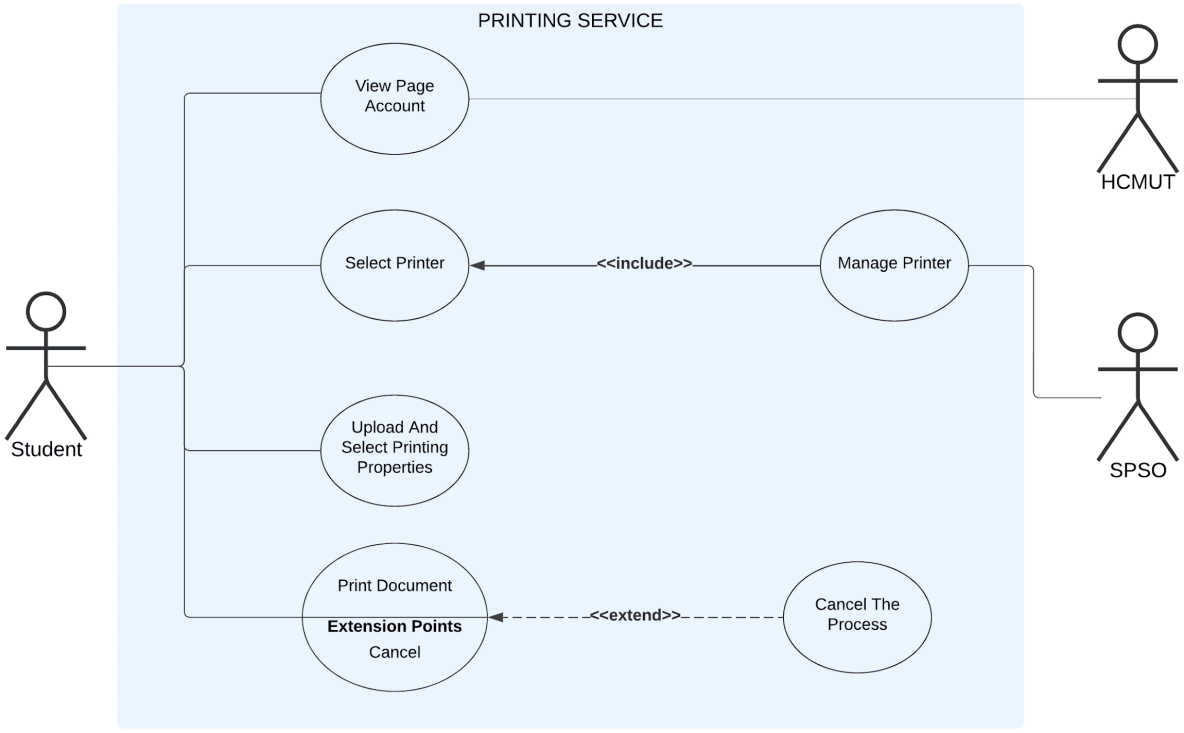
- Availability: The service should be accessible around the university open time length (from 6 AM to 9 PM)
- Testability: The system will be tested regularly for troubleshooting, refactoring, and introducing new features to improve performance.
- Reusability: The system will leverage existing services such as HCMUT_SSO for authentication and BKPay for payments.
- Error Correction Mechanism: The system should allow errors (e.g., in logging, configuration, or content) to be corrected efficiently by authorized users (SPSO) without causing further disruptions.
- Update Flexibility: The system must support easy updates to adapt to new requirements, including adding new file types, adjusting configurations, or revising content.
- Content Versioning: The system should keep track of configuration or content changes, enabling rollback to previous states when necessary.
- Documentation Standards: All changes made by the SPSO must follow established documentation standards to ensure clarity, consistency, and traceability.

2. Use-case Diagrams (1.3)

2.1 Use-case Diagram for the Whole System


A smart printing service for students at HCMUT
Manage Other Configuration
Manage Printer
Select Printer
SPSO
Print Document
View Report
View The Printing History
Student
University
Acount Page Balance
Buy Printing Page
HCMUT_SSO
Log In
BK Pay
Web-based App
Mobile App

2.2 Use-case Diagram for Printing Service Module



2.3 The Details of Usecases in Printing Service Module

2.3.1 Usecase Print Document

ID and name	UC-1 Print Document
Created by	Kien Hoa **Date created:** 29/09/2024
Primary Actor	Student **Secondary Actor:** HCMUT, SPSO
Description	After the student selected printer, uploaded and selected printing properties, student will be asked to print the document or to cancel the process.
Trigger	After student uploaded document and selected printing properties.
Preconditions	PRE-1: The student's identity has been authenticated by HCMUT SSO. PRE-2: The number of pages that the student intend to print exceed the student's account balance.

Postconditions	POST-1: The document is printed. POST-2: The system updates the student's printing page balance.
Normal flow	1. The system checks the student's printing page balance. 2. If the balance is sufficient, the system allows the student to upload document. 3. The student selects a document to print. 4. The student selects printing properties. 5. The printer prints the document. 6. The system deducts pages from the student's balance.
Alternative flow	After the student selected printing properties, student can cancel the process.
Exceptions	None

2.3.2 Usecase Cancel The Process

ID and name	UC-2 Cancel The Process
Created by	Kien Hoa **Date created:** 29/09/2024
Primary Actor	Student **Secondary Actor:** HCMUT, SPSO
Description	After the student selected printer, uploaded and selected printing properties, student will be asked to print the document or to cancel the process.
Trigger	After student uploaded document and selected printing properties.
Preconditions	PRE-1: The student's identity has been authenticated by HCMUT SSO. PRE-2: The number of pages that the student intend to print exceed the student's account balance.
Postconditions	POST-1: The document is printed. POST-2: The system updates the student's printing page balance.
Normal flow	1. The system checks the student's printing page balance. 2. If the balance is sufficient, the system allows the student to upload document. 3. The student selects a document to print. 4. The student selects printing properties.

	5. The student cancels the process.
Alternative flow	None
Exceptions	None

2.3.3 Use case Upload and Select Printing Properties

ID and name	UC-3 - Upload and Select Printing Properties
Created by	Khanh Nam **Date created:** 29/09/2024
Primary Actor	Student
Description	This use case allows students to upload documents into the SSPS and the system will display options for the students to modify the printing configuration
Trigger	The student selects the “Printing document” in the navigation bar
Preconditions	PRE-1: The student’s identity has been authenticated by HCMUT SSO PRE-2: The printer is available at campus PRE-3: The student’s page balance still remains
Postconditions	POST-1: The system will display a popup to confirm that the configuration is set up successfully and document is prepared to be printed
Normal flow	**Upload and select printing pages properties** 1. The system displays a file upload interface. 2. The user uploads the desired document file from the local computer, or chooses file from recent upload or from Google Drive. 3. The system verifies the uploaded file type and confirms it’s supported for printing (see 3.E) 4. The system presents options of available printers with their location in campus for the student to choose 5. After choosing a desired printer, user will modify the printing properties (number of copies, paper size, layout,...) and save the configuration (see 5.E)

	6. The system displays a summary of document and chosen properties for confirmation 7. User selects “Accept” to confirm the printing configuration
Alternative flow	None
Exceptions	3.E Student uploads inappropriate file type or format 3.E.1 The system notifies the error to inform the student 5.E Student selects “Return” 5.E.1 The system returns back to the Upload interface

2.3.4 Usecase Select Printer

ID and Name	UC-4 - Select Printer
Created by	Chinh Hoang **Date created** : 31/10/2024
Primary Actor	Student **Secondary Actor** : SPSO
Description	The Student selects a printer to use for printing a document based on available printers on campus.
Trigger	The Student initiates the action by choosing to print a document through the HCMUT_SSPS system.
Pre-conditions	PRE-1. The Student is authenticated through HCMUT_SSO. PRE-2. The system has available printers for selection.
Post-conditions	POS-The selected printer is set for the current print request.
Normal Flow	**4.0 Select Printer** 1. The Student accesses the printing service and chooses a document to print. 2. The system displays a list of available printers, including details (ID, location, status). 3. The Student selects a printer.
Alternative Flow	**4.1 No Printer Available** 1. The Student accesses the printing service and chooses a document to print. 2. The system informs that there are no printers left. 3. The Student leaves the system and ends the process.

Exceptions	1. The system fails to retrieve available printers: an error message is displayed, and the process terminates.

2.3.5 Usecase Manage Printer

ID and name	UC-5 Manage Printer
Created by	Huynh Nga **Date created** : 31/10/2024
Primary Actor	Student **Secondary Actor** : SPSO
Description	The SPSO manages printer configurations, enabling or disabling printers as needed.
Trigger	The SPSO initiates printer management through the HCMUT_SSPS system interface.
Preconditions	PRE-1: SPSO is authenticated. PRE-2: SPSO has admin privileges. PRE-3: Printer management system is online.
Postconditions	POST-1: Printer list is updated in the system. POST-2: Notifications sent to users if any printers are taken offline.
Normal flow	1. 1. SPSO selects "Manage Printer" from the admin panel. 2. 2. System displays a list of existing printers. 3. 3. SPSO selects an action (add, update, or delete a printer). 4. 4. If adding, SPSO enters the printer details (name, location, status). 5. 5. If updating, SPSO selects a printer and modifies details. 6. 6. If deleting, SPSO confirms the removal. 7. 7. System saves changes and updates availability. 8. 8. System notifies users of changes.
Alternative flow	1A. SPSO tries to delete a printer currently in use. - System displays a warning and prevents deletion.
Exceptions	E1: System is offline: System notifies SPSO and queues the update for later.

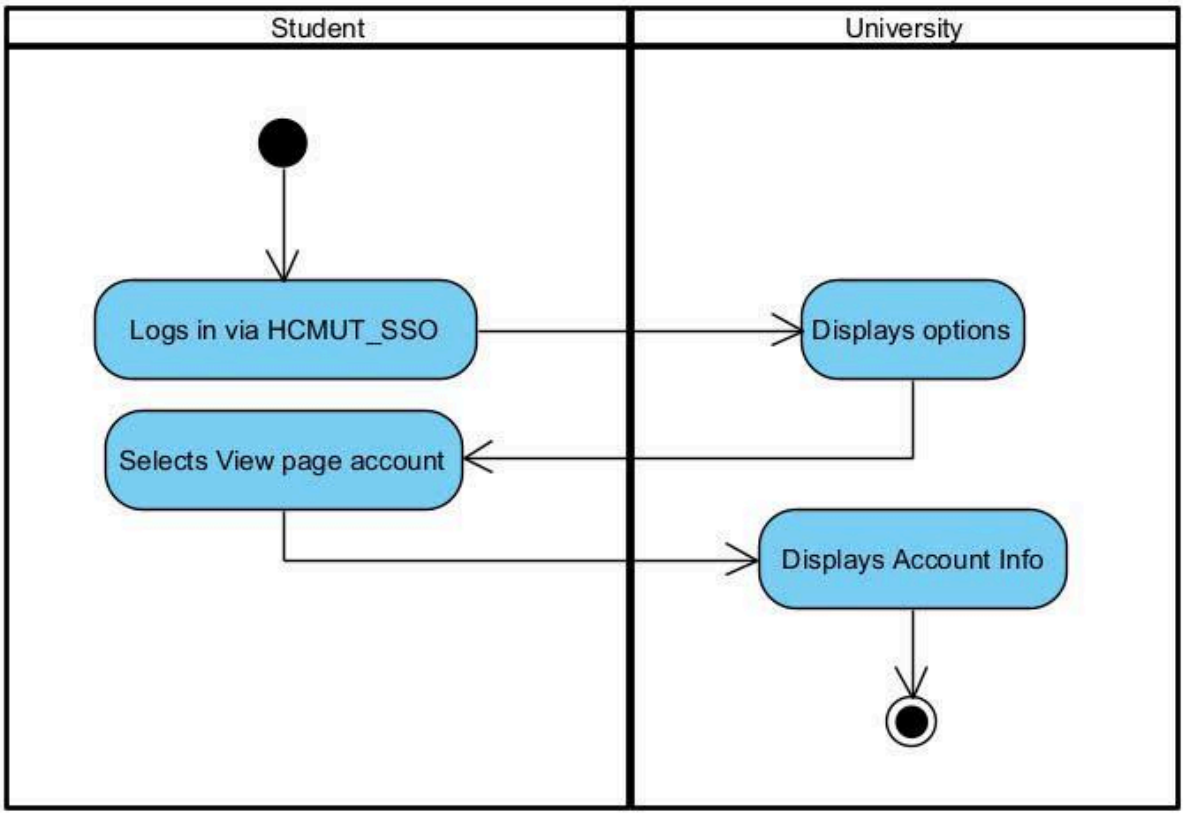
2.3.6 View Page Account

ID and name	UC-6 View Page Account
Created by	Huynh Nga **Date created** : 31/10/2024
Primary Actor	Student **Secondary Actor** : SPSO
Description	The student accesses their account page to view their printing balance, past print jobs, and remaining printing credits.
Trigger	Student logs in and selects "View Page Account".
Preconditions	PRE-1: Student is authenticated. PRE-2: Account information is stored in the system. PRE-3: Printing service system is online.
Postconditions	POST-1: Account information is displayed to the student. POST-2: Recent print jobs and balance are up to date.
Normal Flow	1. Student logs into the system. 2. Student selects "View Page Account". 3. System retrieves student's account details. 4. System displays balance, recent jobs, and remaining credits. 5. Student can navigate back to other sections.
Alternative Flows	2A: Student attempts to view account while system is under maintenance. - System displays a maintenance notification.
Exceptions	2E1: Account data retrieval fails. - System notifies the student of a temporary issue and retries.

3. System Modelling (Task 2)

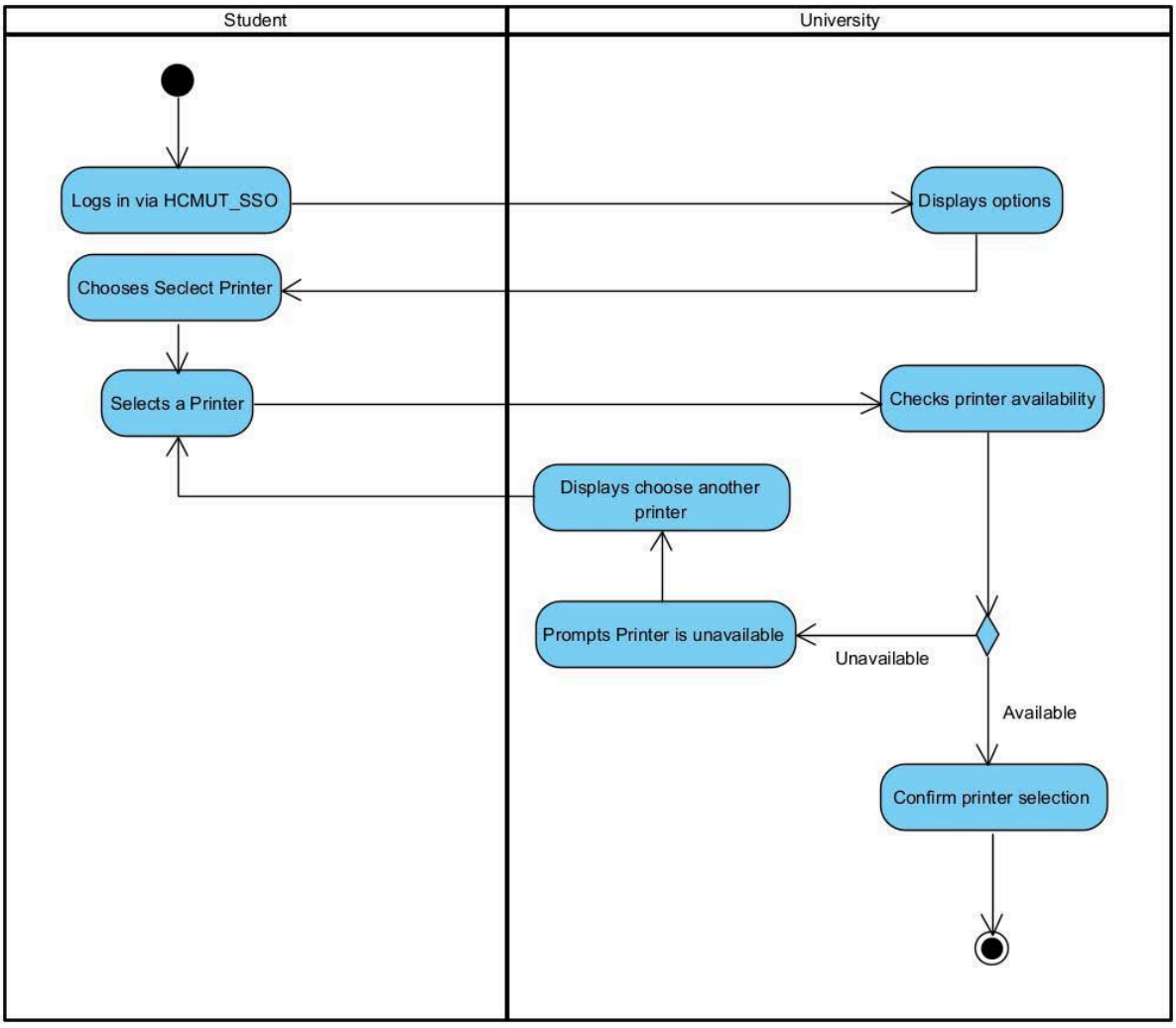
3.1 Activity Diagram

3.1.1 View Page Account



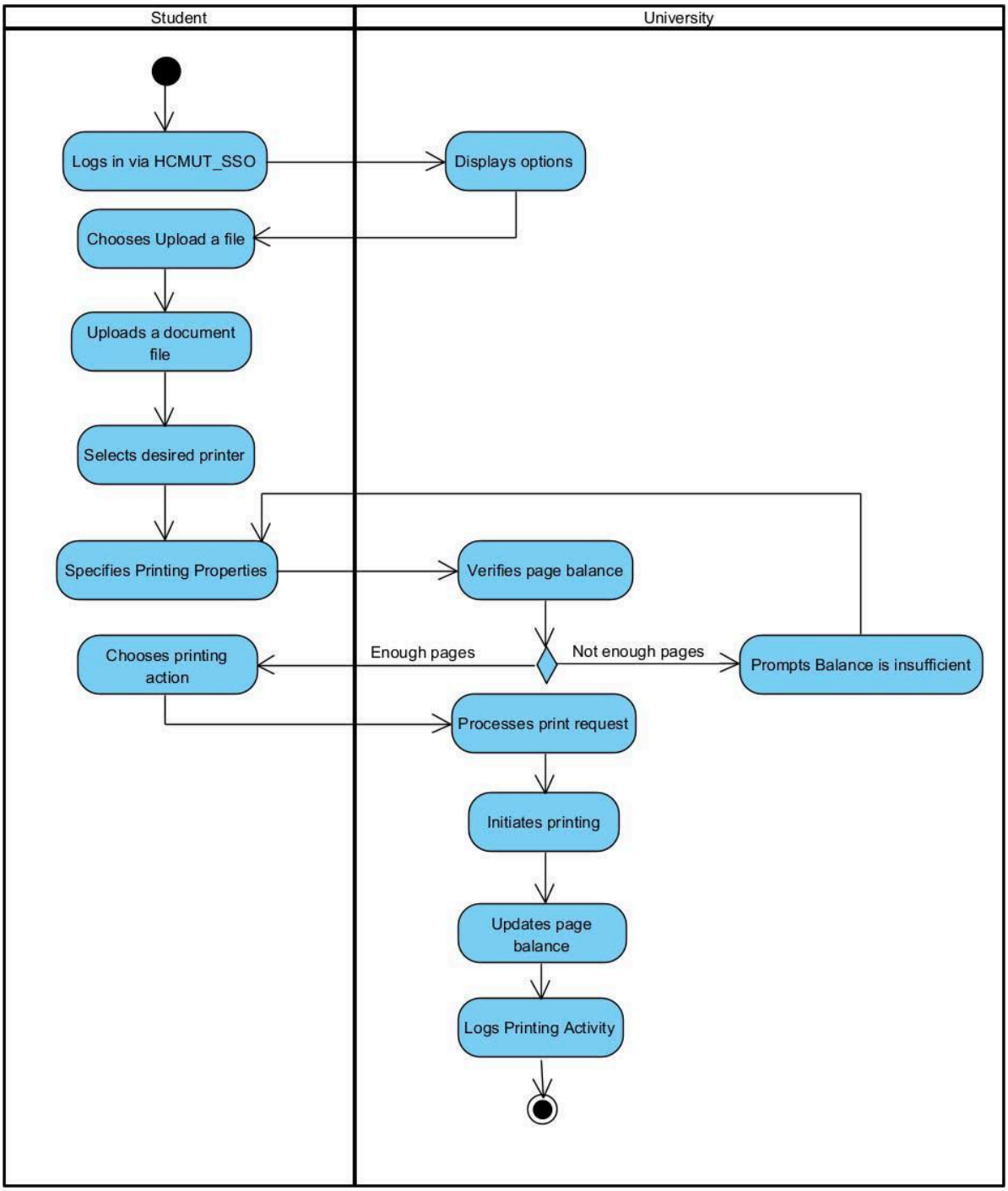
The View Page Account diagram illustrates the user's journey from accessing the main website to logging in and viewing their personal account details. It starts with the user entering their login credentials on the login page. Upon successful authentication, the system fetches the account data and displays it on the user's account page, providing an overview of their personal information, transaction history, and account settings. This process ensures that users can seamlessly access and manage their account details.

3.1.2 Select Printer



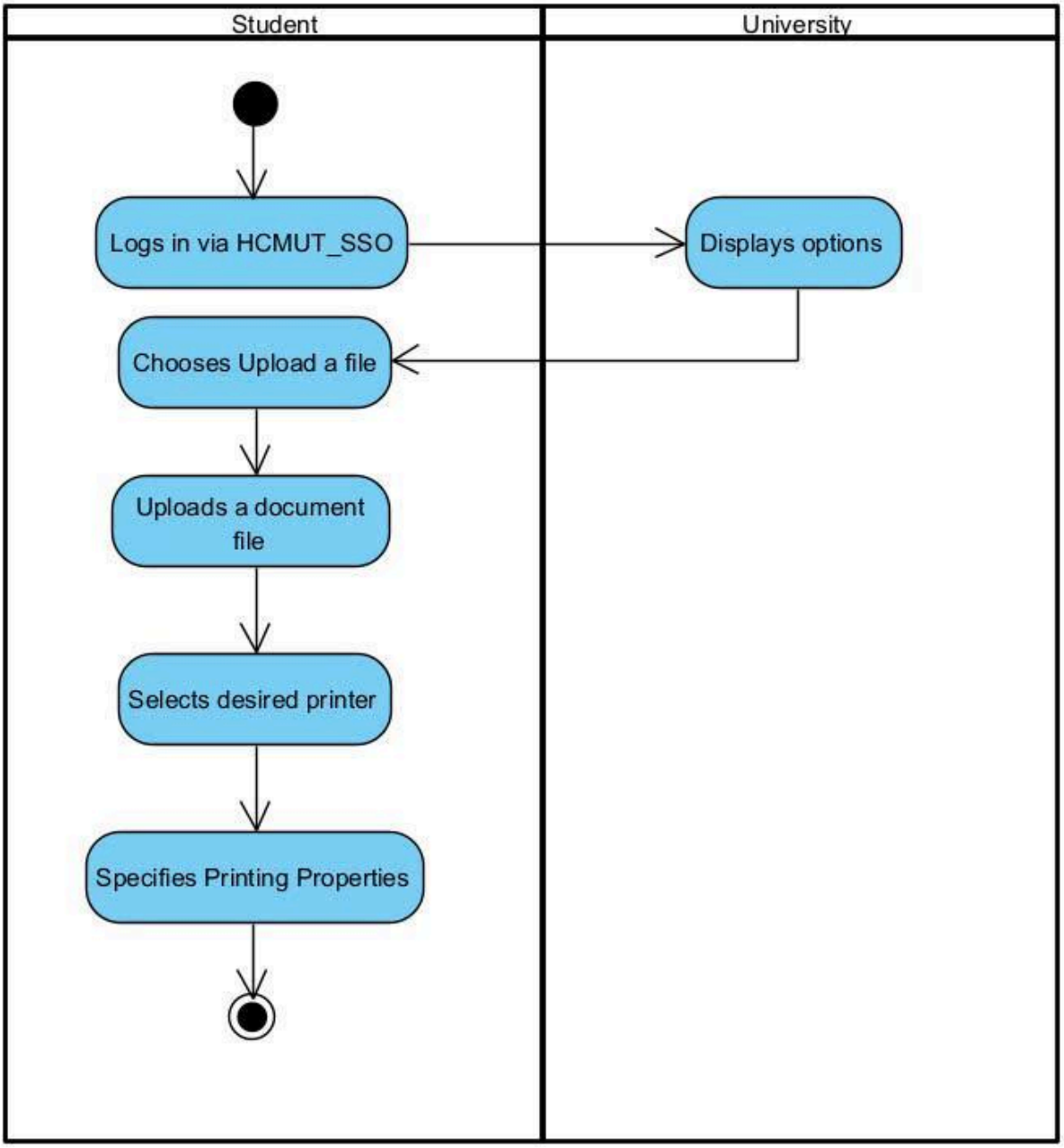
The Choose a Printer diagram outlines the steps a user takes to select a printer for their printing needs. Starting from the document view page, the user initiates the printing process by accessing a list of available printers. The user then reviews the options, considering factors such as printer location and status, before selecting the preferred printer. Once the printer is chosen, the system confirms the selection and prepares for the next steps in the printing process, ensuring a smooth transition to actual document printing.

3.1.3 Manage Printer



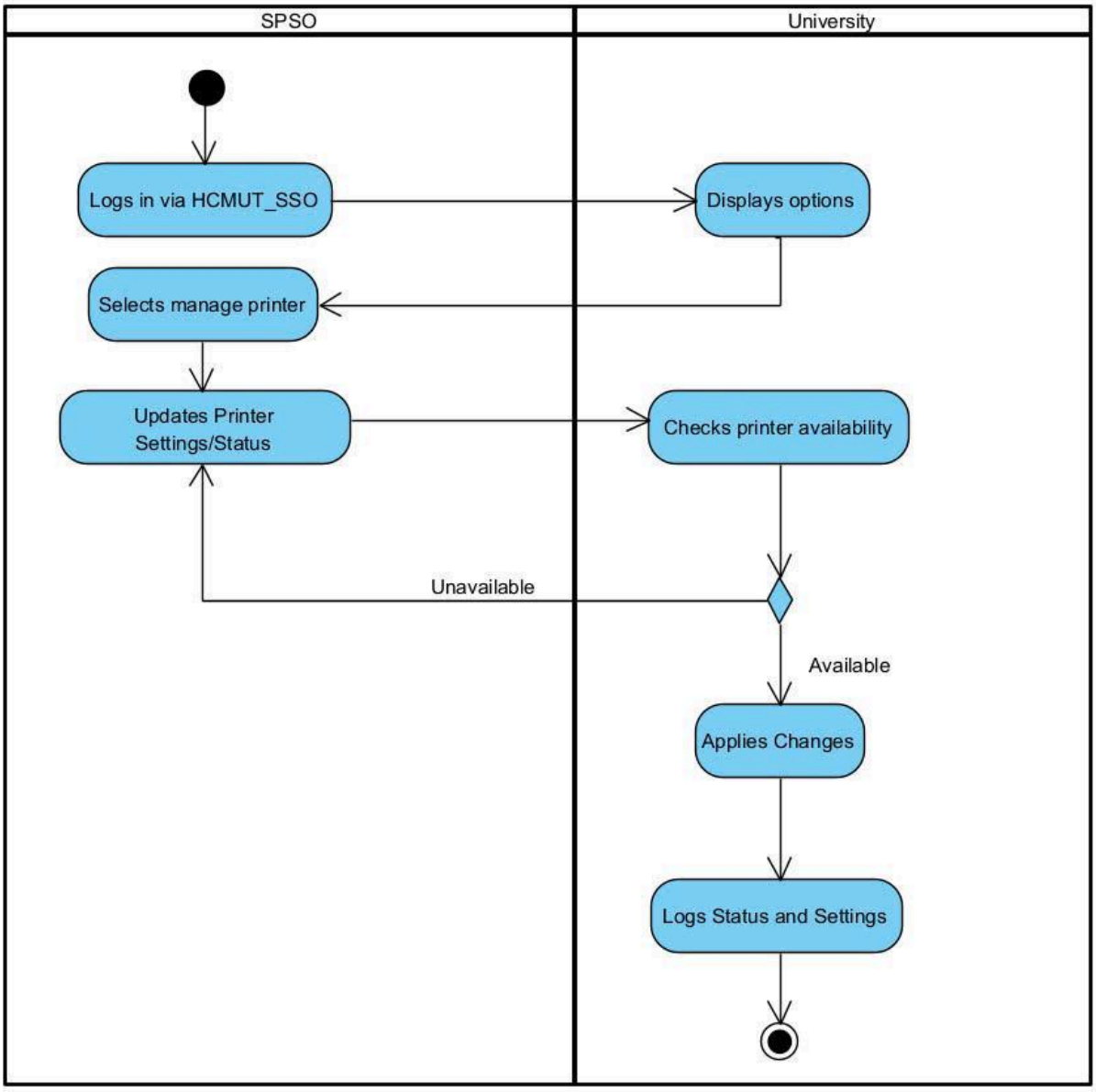
In the Manage Printer via SPSO diagram, the focus is on adjusting printer settings through the SPSO interface. The user navigates to the printer settings menu, where they can modify various preferences, such as paper size, print quality, and color settings. These adjustments are then saved and applied to the selected printer. The diagram highlights how users can personalize their printing experience by tailoring printer settings to their specific needs, ultimately enhancing efficiency and print output quality.

3.1.4 Upload and Select Printing Properties



This diagram shows the flow of uploading a document and specifying its printing properties. It begins with the user accessing the upload section and selecting the document they wish to print. Following the upload, the user is prompted to specify printing properties such as page layout, print quality, and number of copies. After finalizing these settings, the document is ready for printing. This diagram underscores the flexibility and control users have over the printing process, from document upload to the final print output.

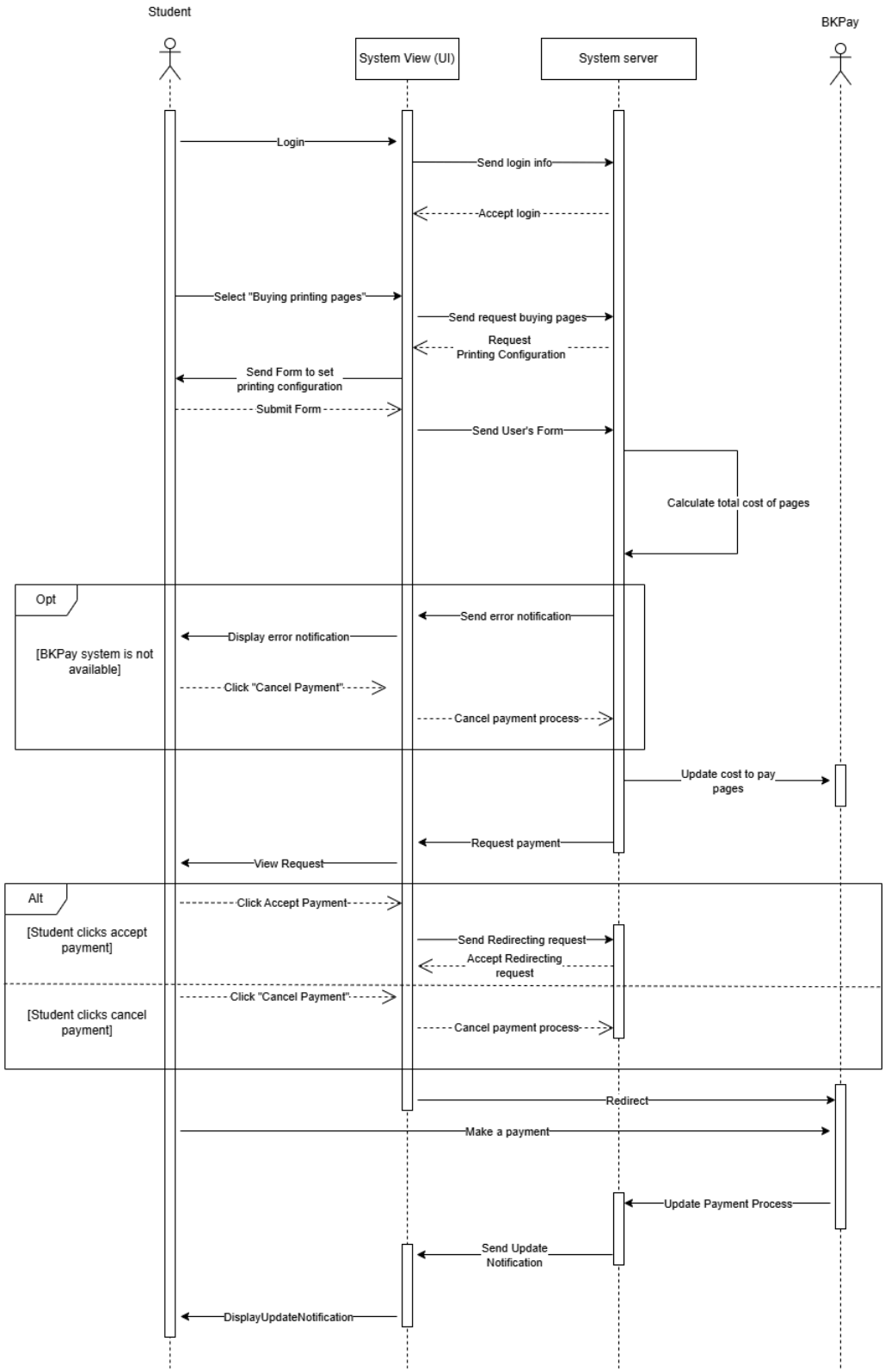
3.1.5 Mange Printer



In the Manage Printer via SPSO diagram, the focus is on adjusting printer settings through the SPSO interface. The user navigates to the printer settings menu, where they can modify various preferences, such as paper size, print quality, and color settings. These adjustments are then saved and applied to the selected printer. The diagram highlights how users can personalize their printing experience by tailoring printer settings to their specific needs, ultimately enhancing efficiency and print output quality.

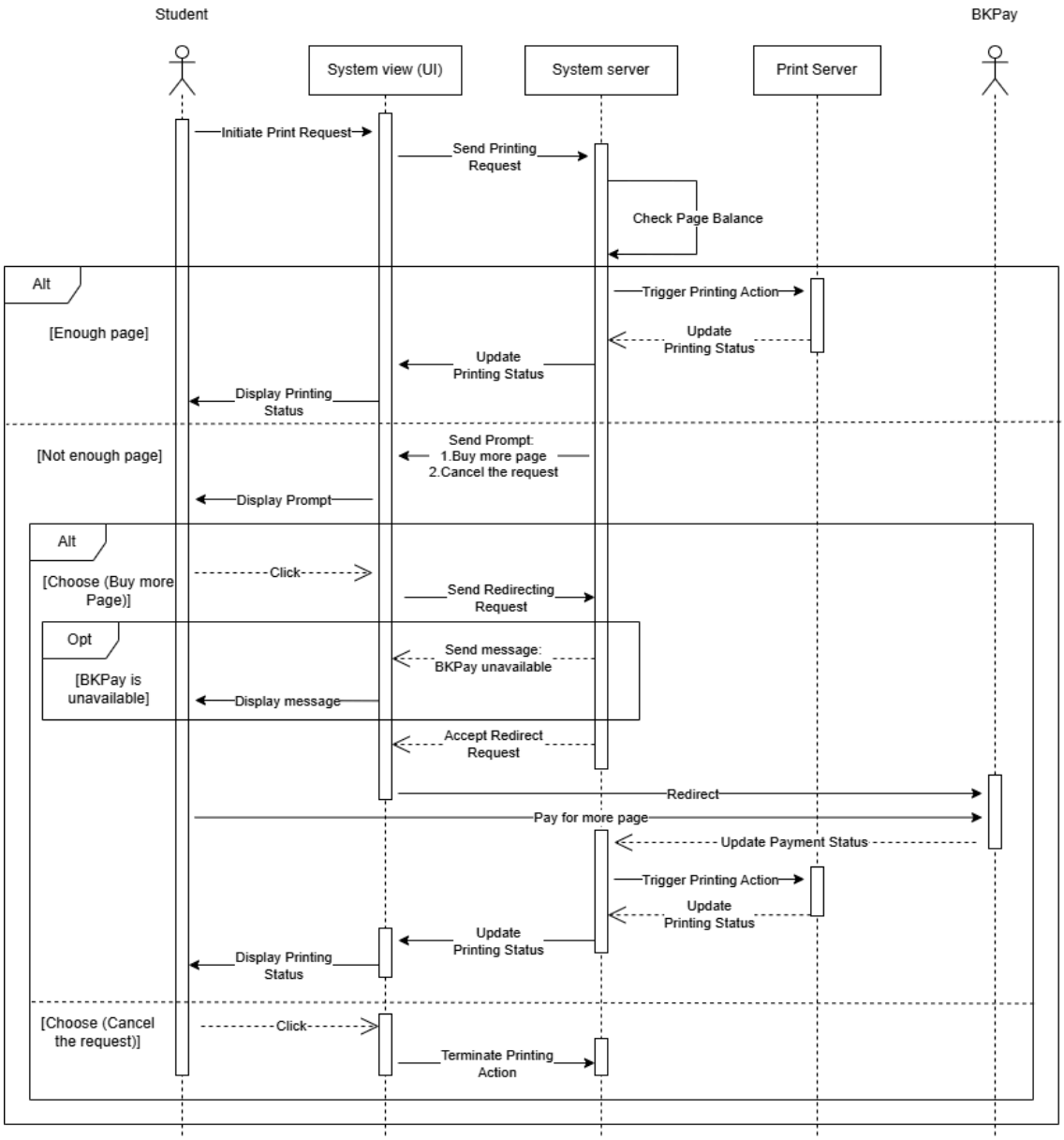
3.2 Sequence Diagram

3.2.1 Purchase Printing Pages


Student
System View (UI)
System server
BKPay
Login
Send login info
Accept login
Select "Buying printing pages"
Send request buying pages
Request
Printing Configuration
Send Form to set
printing configuration
Submit Form
Send User's Form
Calculate total cost of pages
Opt
[BKPay system is not
available]
Send error notification
Display error notification
Click "Cancel Payment"
Cancel payment process
Update cost to pay
pages
Request payment
View Request
Alt
[Student clicks accept
payment]
Click Accept Payment
Send Redirecting request
Accept Redirecting
request
[Student clicks cancel
payment]
Click "Cancel Payment"
Cancel payment process
Redirect
Make a payment
Update Payment Process
Send Update
Notification
DisplayUpdateNotification

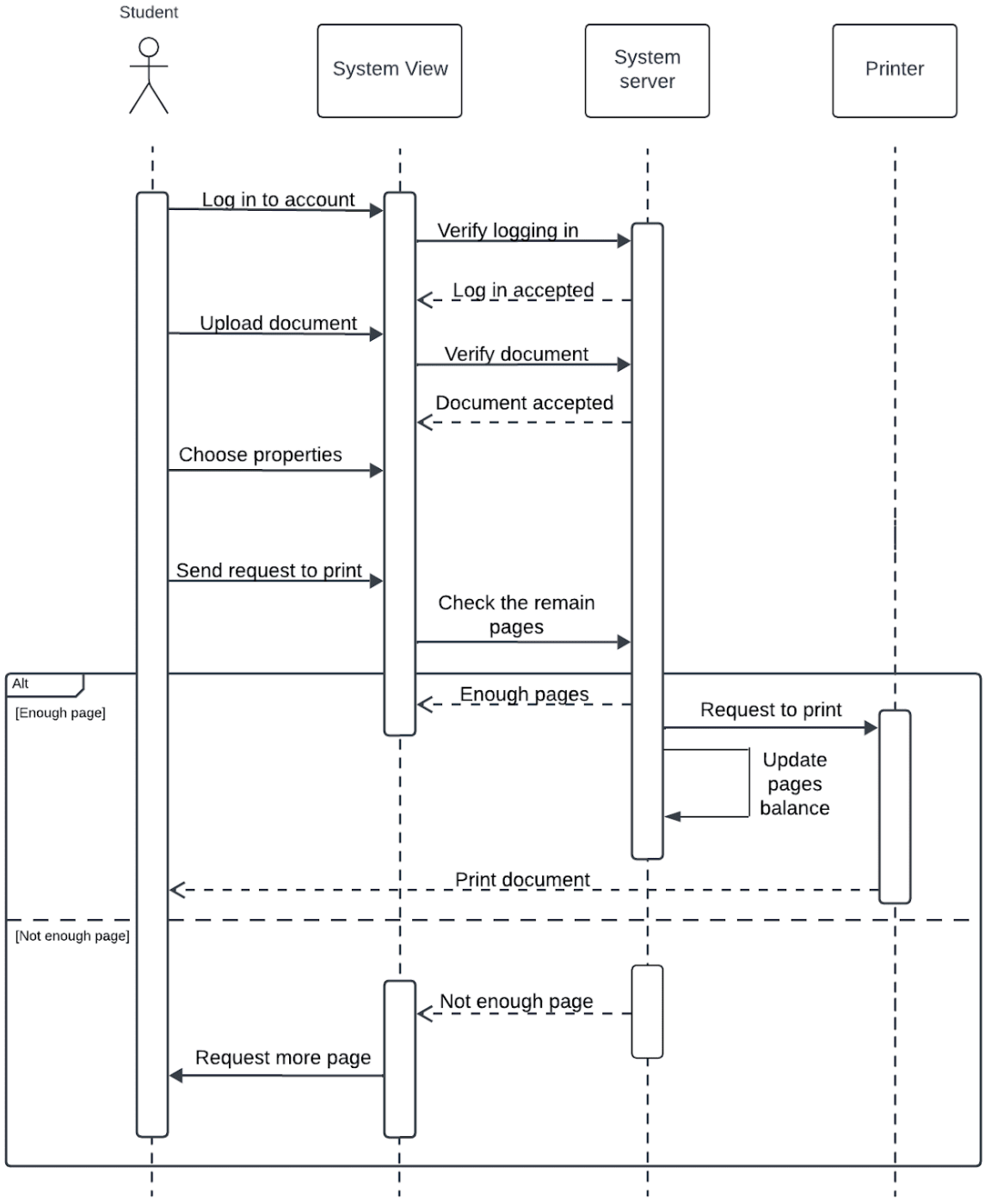
The student initiates the purchase by logging into the system and then selecting the "Buy printing pages" option, after which the system displays a form for setting printing configuration (choosing the number of pages, size, layout, etc.). The system calculates the total cost for paying pages and displays it to the student, who then submits the purchase request. The system sends a request to BKPay for payment, and if BKPay is available, it redirects the student to the payment gateway. In case the BKPay is not working, the system will send an error notification to the user to inform him/her that the BKPay is unavailable. When the student makes the payment in BKPay, the BKPay will check the student's bank account, which is registered in the BKPay system, to inspect the bank balance. Once the payment is successful, BKPay sends an update of the payment process to the system, which updates the student's page balance and payment logs and displays the update message.

3.2.2 Continue Printing Action



The student initiates the printing request by selecting the "Print" option in the system view, and the system checks the student's page account balance to determine if there are enough pages available. If there are, the printing action will proceed, and the printing status will be updated in the student's account and displayed in the system's UI after the printing action is successful. Otherwise, the student is prompted to either buy more pages through BKPay or cancel the request. If the student chooses to buy more pages and the BKPay system is available, the system redirects them to the payment gateway, and in exception, when the BKPay system is not available, the SPSS will send the message to inform the student that the BKPay is not working. After successful payment, the page balance is updated in the SPSS server and the printing action can continue.

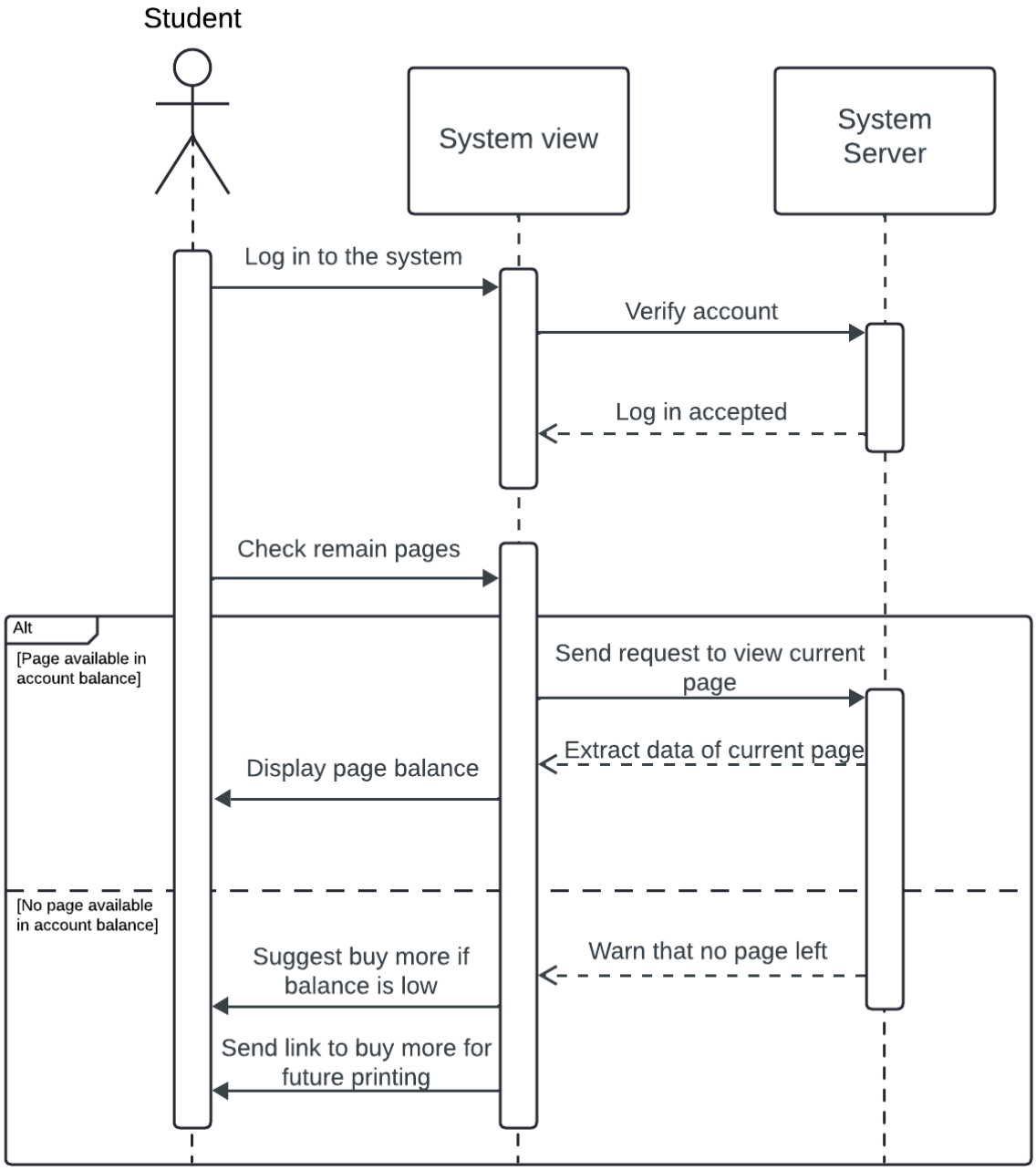
3.2.3 Receive Request for Printing Action



Initially, the student logs into their account through the HCMUT_SSO.Then, the system will check their logging in and accept it. Now, he/she can upload his/her document, which is verified by the system in a moment and choose the printing properties. After that, a request is sent through the HCMUT_SSO, and the system then checks the remaining page balance in the student's account.

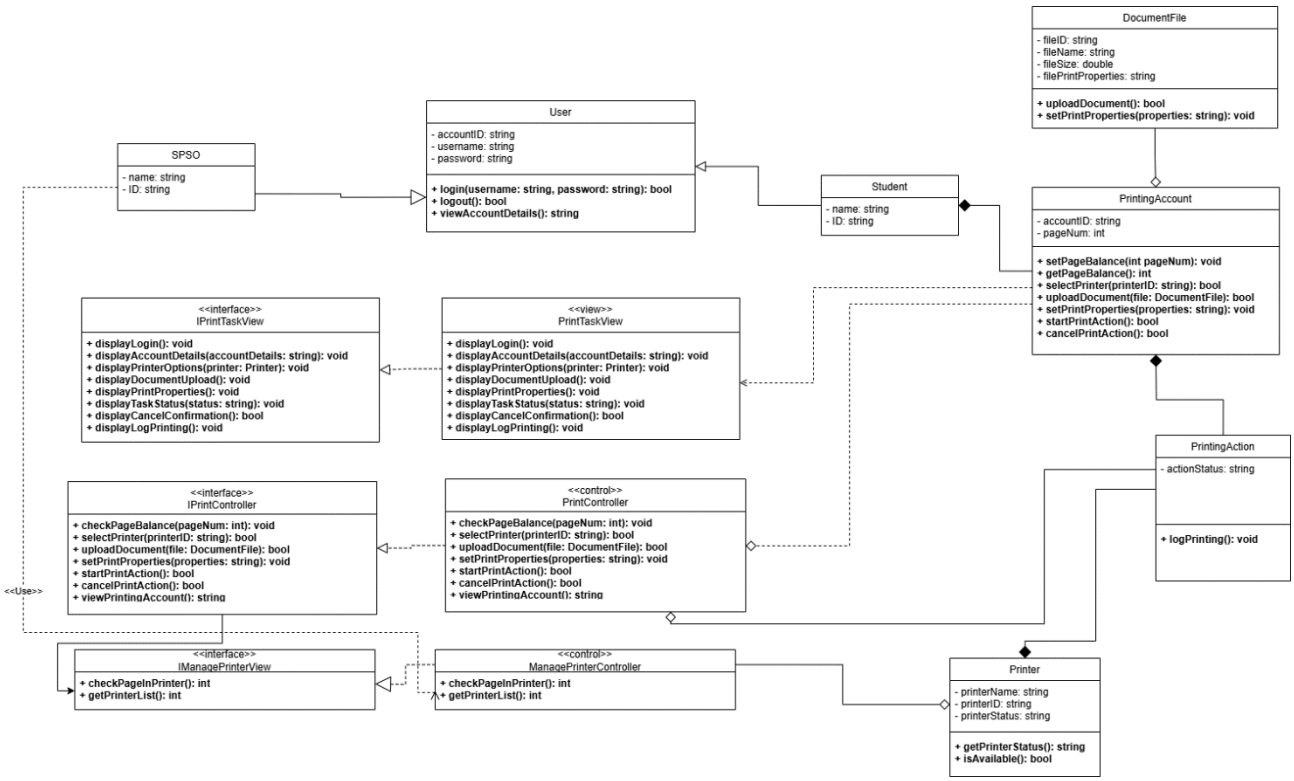
Normally, if the remaining page is enough for printing action, the system will send a request to the printer and update the new balance. Finally, the printer will print the document that the student needs. However, in an alternative flow where the student doesn't have enough pages, the system will announce it to the HCMUT_SSO, and it then will request the student to buy more pages.

3.2.4 Check Page Account



The student wants to check their page balance in their account. To do so, they must first log into the system via HCMUT_SSO. Once the login is verified and accepted, the student navigates to the "Check Page Balance" section. Typically, the current page balance is retrieved and displayed through HCMUT_SSO, which also provides a link for purchasing additional pages if the balance is low. If the balance reaches zero, HCMUT_SSO will directly send a link for the student to buy more pages to resume printing.

3.3 Class Diagram



The class diagram above illustrates the system components following the Model-View-Controller (MVC) pattern for a student printing a document.

SPSO Class

- **Attributes:**
  - name: string
  - id: string

User Class

- **Attributes:**
  - accountID: string
  - password: string
  - status: string

- **Methods:**
  - login(username: string, password: string): bool
  - logout(): void
  - viewAccountDetails(): string

This class manages user information and login functionality. Its attributes store account details, while its methods handle authentication and account management tasks.

Student (inherits from User)

- **Attributes:**
  - name: string
  - ID: string

The Student class inherits from the User class, implying that students are a specific type of user with additional attributes such as name and ID.

DocumentFile Class

- **Attributes:**
  - docID: string
  - docName: string
  - filePath: string
  - size: float
- **Methods:**
  - uploadDocument(): void
  - deleteDocument(): void
  - editFileProperties(fileID: string): void

This class represents documents to be printed. It allows uploading, deleting, and editing the properties of the document.

PrintingAccount Class

- **Attributes:**
  - accountID: string
  - accountPin: int
  - balance: float
- **Methods:**
  - getBalance(): float
  - addBalance(amount: float): void
  - deductBalance(amount: float): bool
  - createNewAccount(pin: int): void
  - cancelPrintAccount(): bool

This class is responsible for managing a student's printing account, including balance management and account creation or cancellation.

Printer Class

- **Attributes:**
  - printerID: string
  - printerName: string
  - printerType: string
  - printerStatus: string
- **Methods:**
  - getPrinterStatus(): string
  - isAvailable(): bool

The Printer class handles information related to printers in the system. It can check the printer status and availability.

PrintingAction Class

- **Attributes:**
  - actionStatus: string
- **Methods:**
  - logPrinting(): void

This class seems to handle the logging of printing actions, possibly for tracking purposes.

PrintTaskView

- **Methods**:
  - displayLogin(): Displays the login screen.
  - displayAccountDetails(accountDetails: string): Shows details of the user's account.
  - displayPrinterOptions(printer: Printer): Displays available printer options.
  - displayDocumentUpload(): Prompts the user to upload a document.
  - displayPrintProperties(): Shows print settings and properties.
  - displayTaskStatus(status: string): Displays the current status of the print task.
  - displayCancelConfirmation(): Confirms if the user wants to cancel the print action (returns a boolean).
  - displayLogPrinting():

This class defines the user interface methods related to printing tasks. It focuses on displaying relevant information to the user and handling user interactions.

PrintController

- **Methods**:

- checkPageBalance(pageNum: int): Checks if the user has enough page balance for the print job.
- selectPrinter(printerID: string): Allows the user to select a printer (returns a boolean).
- uploadDocument(file: DocumentFile): Handles the document upload process.
- setPrintProperties(properties: string): Sets the print properties like page size, orientation, etc.
- startPrintAction(): Initiates the printing process (returns a boolean).
- cancelPrintAction(): Cancels an ongoing print task (returns a boolean).
- viewPrintingAccount(): Provides details about the current printing account.

This class defines the control logic for managing print tasks, including handling document uploads, printer selection, and print actions.

ManagePrinterView

- **Methods**:
  - checkPageInPrinter(): Checks the number of pages left in a specific printer (returns an integer).
  - getPrinterList(): Retrieves the list of available printers (returns an integer)

This class defines methods related to managing printers, including checking available pages and retrieving printer lists

3.4 User Interfaces

Authentication

This is the login interface for the SSPS system. Users are greeted with the message "Welcome to SSPS." There are two login options:

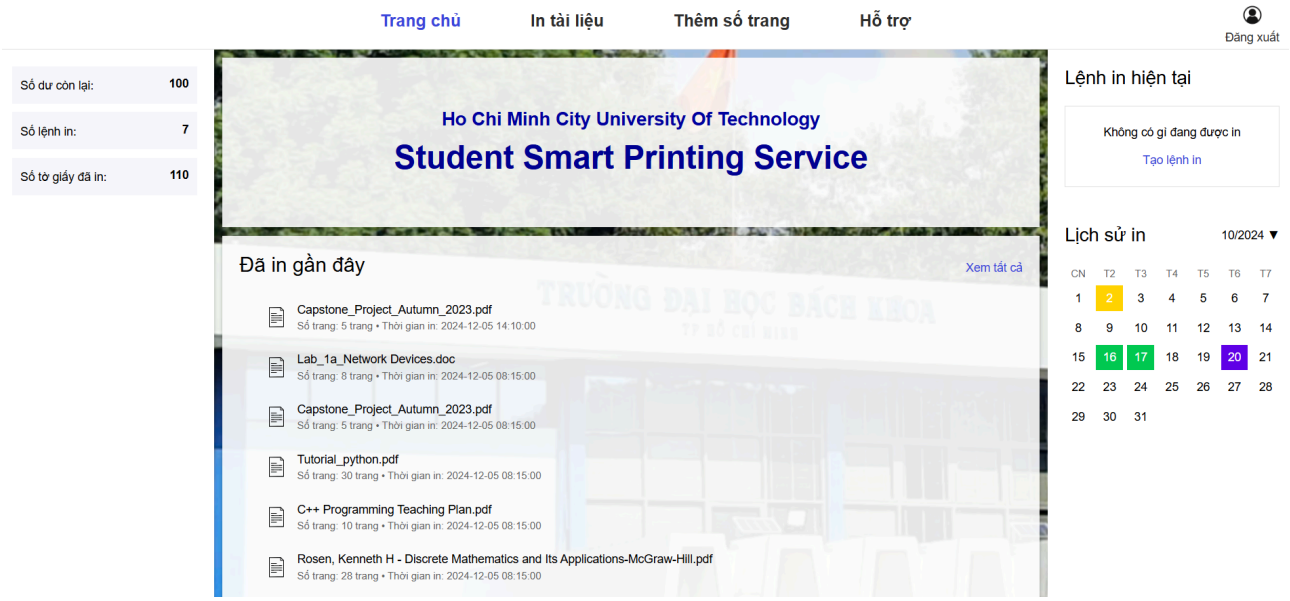
1. Student - If the user selects the "Student" button, the system will navigate to the student account interface.
2. Admin - If the user selects the "Admin" button, the system will redirect to the interface for the Student Printing Service Officer (SPSO).

After selecting to log in as a student or SPSO, the user will be redirected to this interface.

The interface includes:

1. Tài khoản (account) - An input field where users enter their email.
2. Mật khẩu (password) - An input field where users enter their password.
3. Đăng nhập (login button) - Once credentials are entered, users click this button to access the system.

Dashboard (Student View)



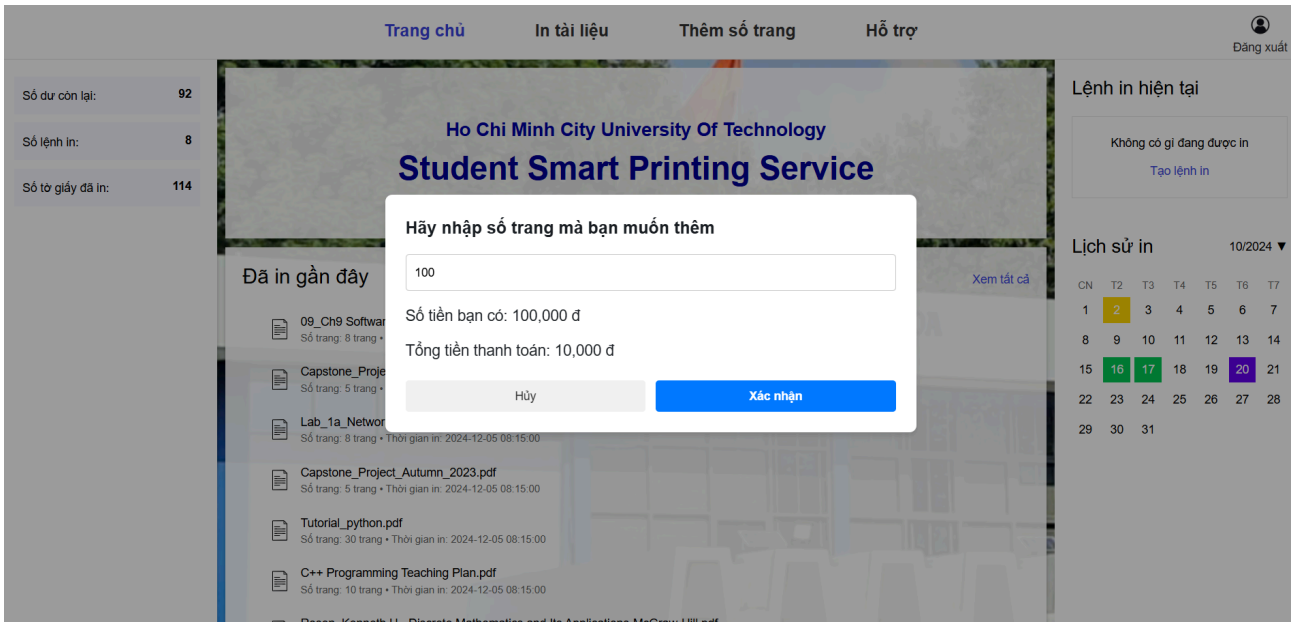
This is the Dashboard interface for students after successfully logging into the Student Smart Printing Service (SSPS) system at Ho Chi Minh City University of Technology. This dashboard will include the following main components:

1. Printing account information (left panel):
    a. Số dư còn lại (Remaining balance): Displays the number of pages left in the user's account.
    b. Số lệnh in (Print orders): Total number of print jobs completed.
    c. Số tờ giấy đã in (Total pages printed): Total number of printed pages.
2. Đã in gần đây (Recent print orders) (center):
    a. The student can view the printing history by viewing this interface. It will have a list of recently printed documents with details such as file name, number of pages printed and print time.
3. Lệnh in hiện tại (Current print orders) (right panel):
    a. Displays the current print status. If no orders are active, users can create a new order by selecting Tạo lệnh in (Create Print Order).
4. Print history calendar (right):

a. A monthly calendar highlights days with print activities. For example, days like the 2nd, 16th, 17th, and 20th are marked with distinct colors.

5. Navigation bar (top):
   a. Includes main options such as Trang chủ (Home), In tài liệu (Print Documents), Thêm Số Trang (Add Pages), and Hỗ trợ (Support).
   b. A Đăng xuất (Logout) button is located in the top-right corner for users to exit the system.

Add Page Balance

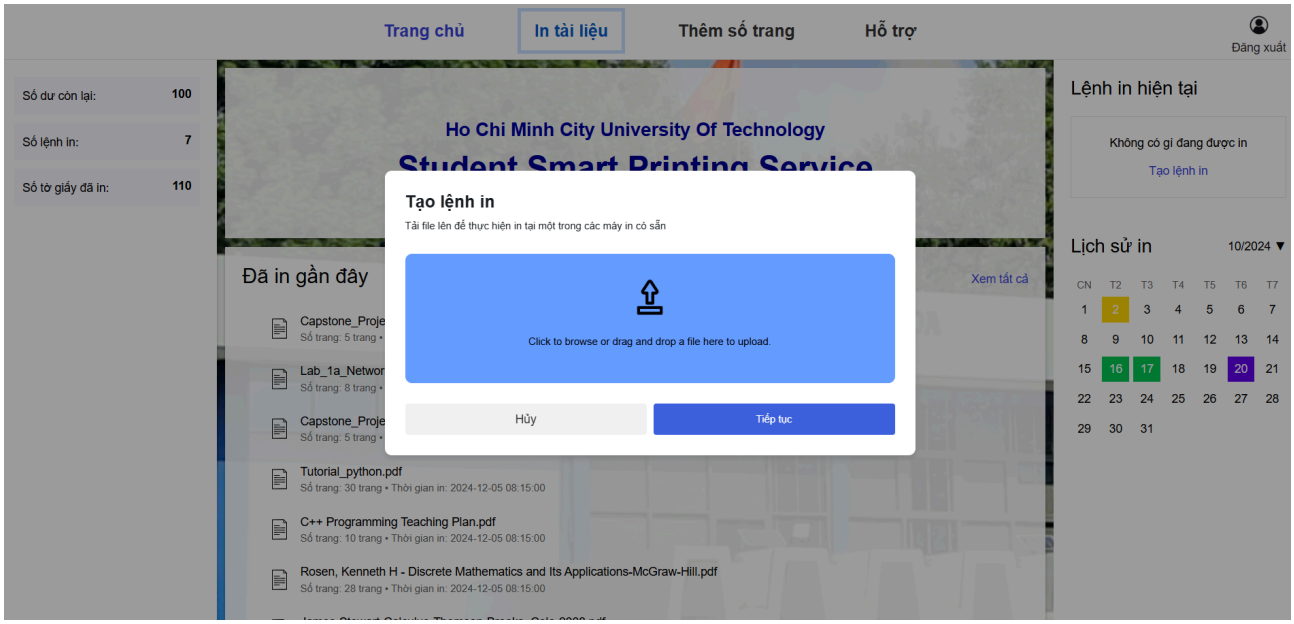


When clicking the “Thêm số trang” (Add Page Number) button, it will pop the dialog box that students can add the page to their page balance. It will have the field for students to input the number of pages to add. Below that input field are information about the current account balance and total cost for the pages being added. For example, the system has 8 “số lệnh in” (print commands), 92 “số dư còn lại” (sheets remaining), and 114 is “số tờ giấy đã in (sheets printed). In the pop-up window for purchasing additional paper, the user enters a number into the white bar under the line “Hãy nhập số trang mà bạn muốn thêm” (Enter the number of sheets you want to add) in order to specify the desired additional pages. Below that, there are two rows displaying the amount of money you have and the total amount to pay for purchasing additional printing paper.

At the bottom of the window, there are 2 functional button: “Hủy” (cancel the process) and “Xác nhận” blue button (confirm purchase)

Create Print Job (Student View)

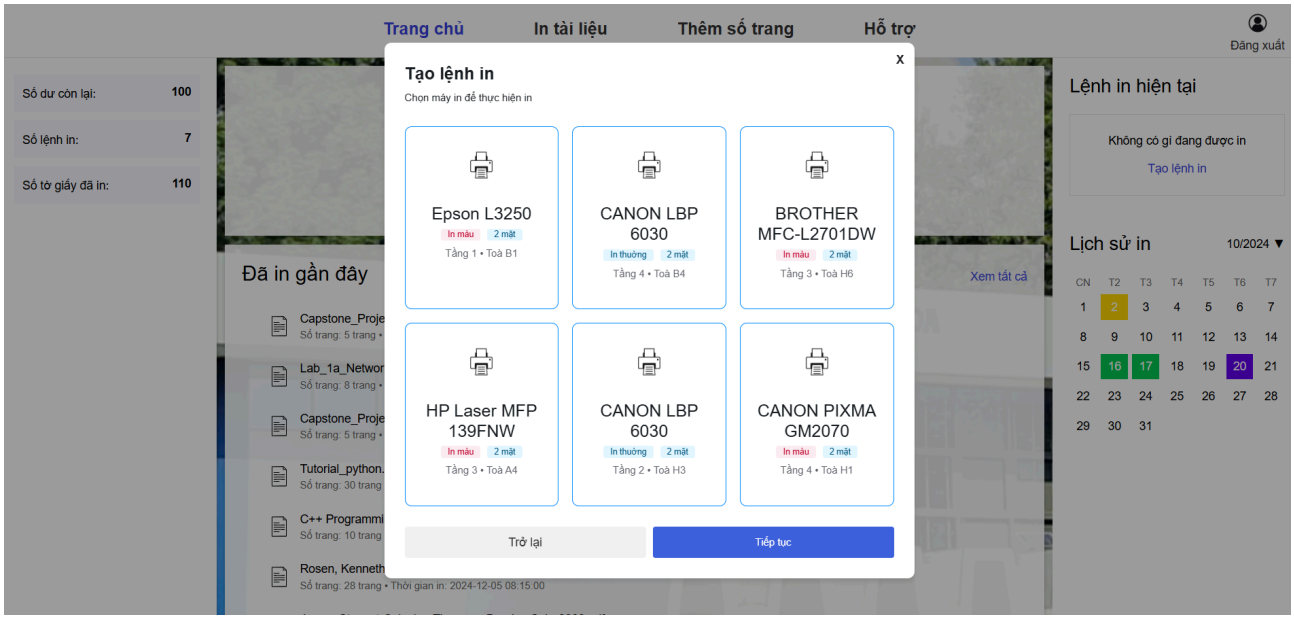
From the dashboard, clicking on "Tạo lệnh in" (create print job) on the top right or “In tài liệu” (print document) on the top center will pop the “Tạo lệnh in” (create print job) window.



The displayed window includes the following components:

Main pop-up window:

- Title: “Tạo lệnh in” (Generate print command").
- Description: “Tải file lên để thực hiện in một trong các máy in có sẵn” to inform the user in uploading a file to this window to initiate printing on the available printer.
- File upload area: within a navy round corner square with the line “Click to browse or drag and drop a file here to upload” at the bottom. It allows users to drag and drop files or click to browse and select a file. A file upload icon is displayed in the center.
- Two functional buttons: "Hủy" (cancel the printing process) and "Tiếp tục" (move to the next step).

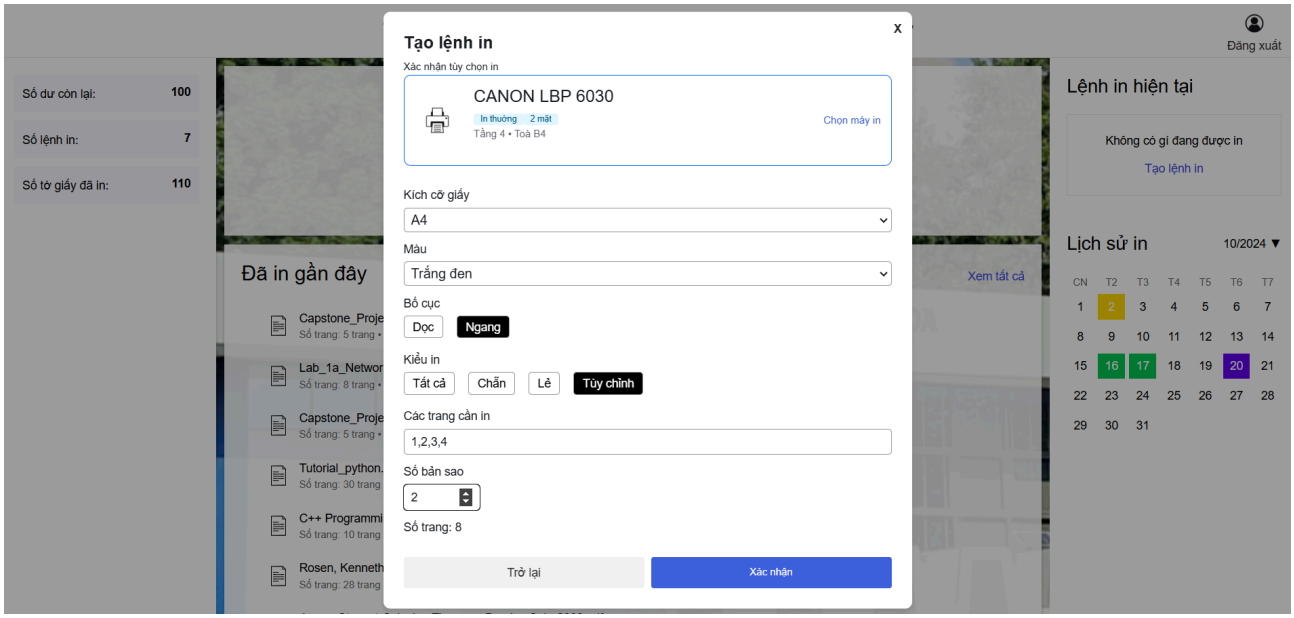


"Create Print Job" Pop-Up Window:

After completing the previous “Tạo lệnh in” step, user will be redirected to this pop-up window that displays the available printer list. It has the following components.

- **Printer icon**
- **Title**: "Tạo lệnh in " (create print job). The same as previous step.
- **Description**: "Chọn máy in để thực hiện in" to inform user to select an available printer on the window.
- **Available Printer List**:
    - Each card has a round corner square shape with a blue thin outline and a white inner background. It represents a printer and contains the following details:
        - Printer name (e.g., Epson L3250, Canon LBP 6030).
        - Supported print modes such as “In màu” (color print), "2 mặt" (duplex print), or "In thường" (standard print). More addition, “In màu” mode in red color unlike the rest mode in blue color.
        - Printer location in the building (e.g., “Tầng 1 - Tòa B1” Floor 1 - Building B1).

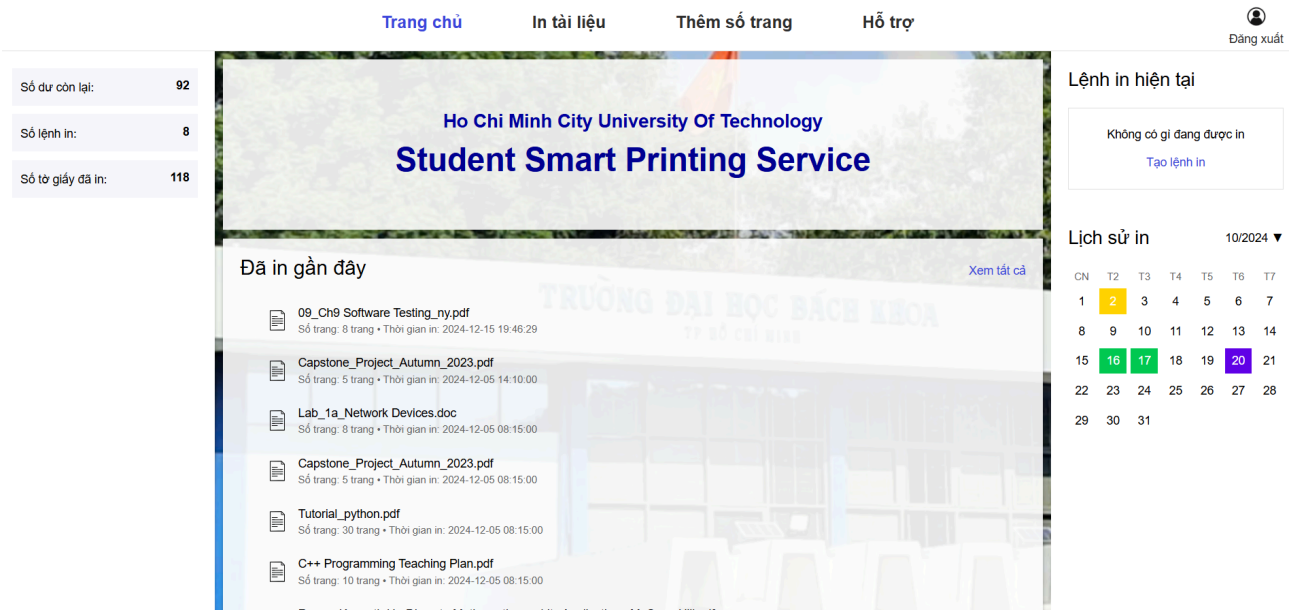
At the bottom of the window are 2 functional button: “Trở lại” (return to previous step) and blue “Tiếp tục” (continue the process) button.



This is the window demonstrating the final step of the printing process. It is an interface for specifying printing properties:

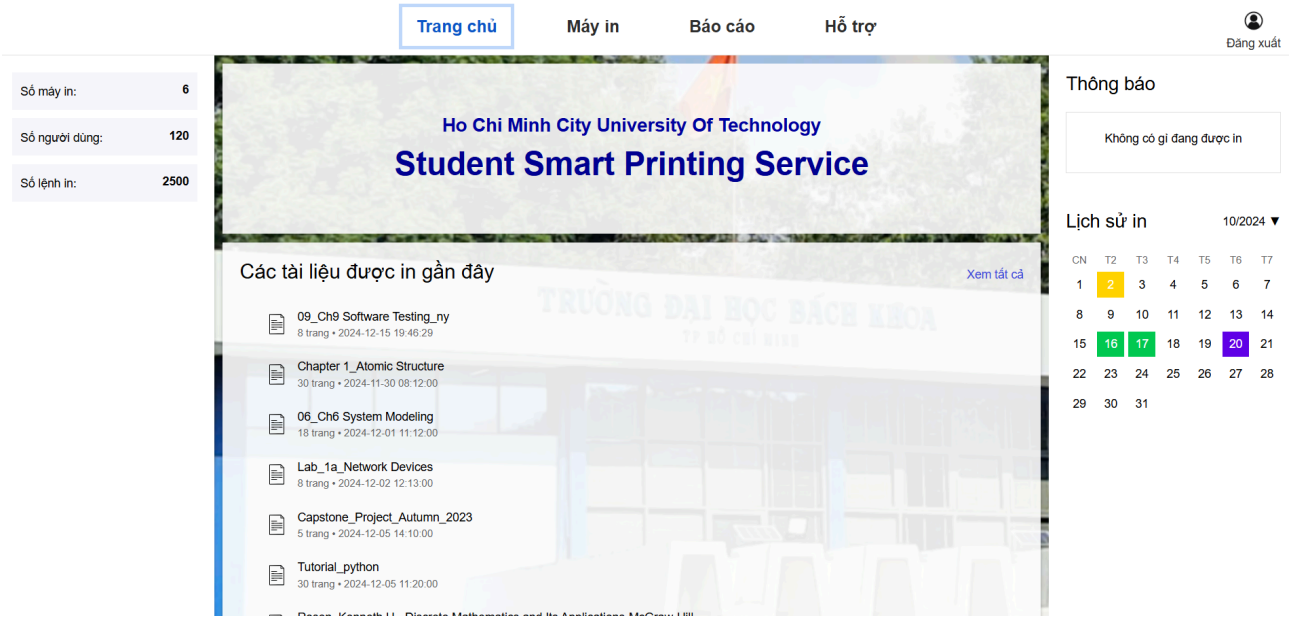
- **Chosen printer information** in a round corner square shape with blue outline. The area contains: Printer icon, Printer name (E.g, CANON LBP 6030), Printer's location (E.g, located on "Tầng 4- Tòa B4" the 4th Floor - Building B4), Printer mode (E.g, "In thường" (standard print), "2 mặt" (duplex print)). On the right hand side of the area is "Chọn máy in" (Select printer) option to redirect user to previous step if user want to choose another printer.
- **Kích cỡ giấy (paper size)**: e.g., A4.
- **Màu (color print option):** e.g., "Trắng đen" (Black and white).
- **Bố cục (layout)**: There are two options, "Dọc" và "Ngang" (Portrait and Landscape), with "Ngang" (Landscape) selected.
- **Kiểu in (print type)**: Options include "Tất cả" (All), "Chẵn" (Even), "Lẻ" (Odd), and "Tùy chỉnh" (Custom).
- **Các trang cần in (pages to print)**: each page separates each other by a comma. E.g., 1, 2, 3, 4.
- **Số bản sao (number of copies)**: e.g., 2 copies.
- **Số trang (total pages)**: 8 pages.

There are two buttons at the bottom of the pop-up window: "Trở lại" (return to previous step) and "Xác nhận" (confirm the specifying printing properties).



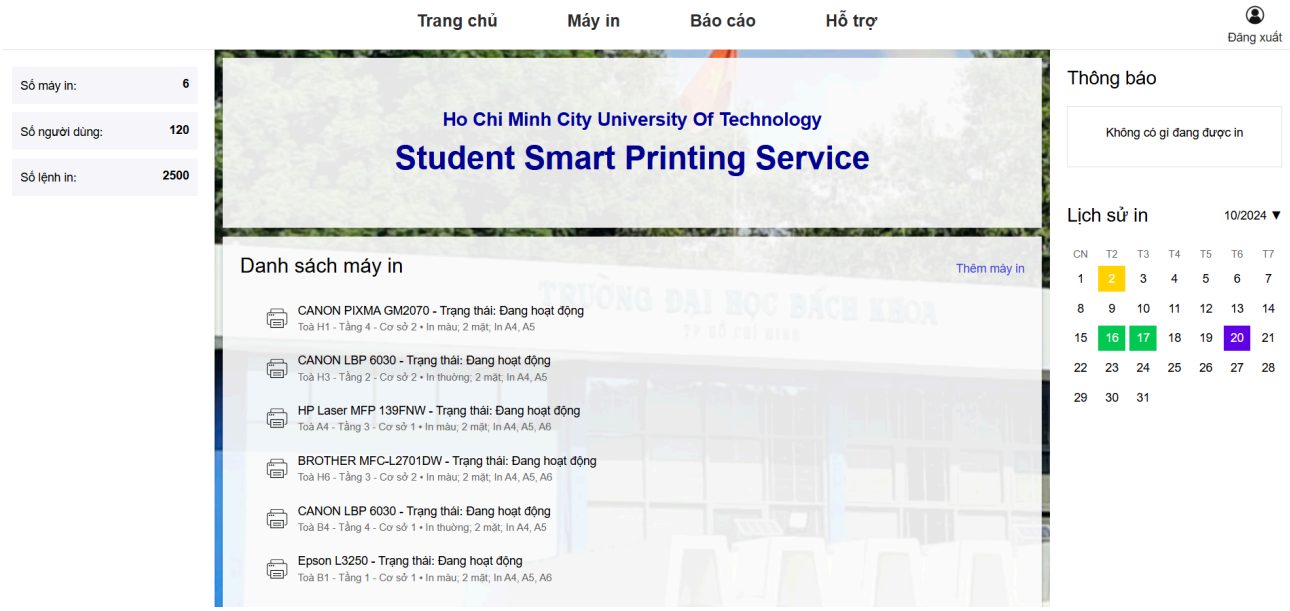
After printing is completed, "Số dư còn lại" (the number of pages) decreases by 8 (100 drop down to 92), "Số lệnh in" (the number of print commands) increases by 1 (7 rises up to 8), "Số tờ giấy đã in" (the number of printed sheets) increases by 8, and the print history displays the printed document, for example, the document "09 - Ch9 Software Testing_ny.pdf".

Dashboard (SPSO View)

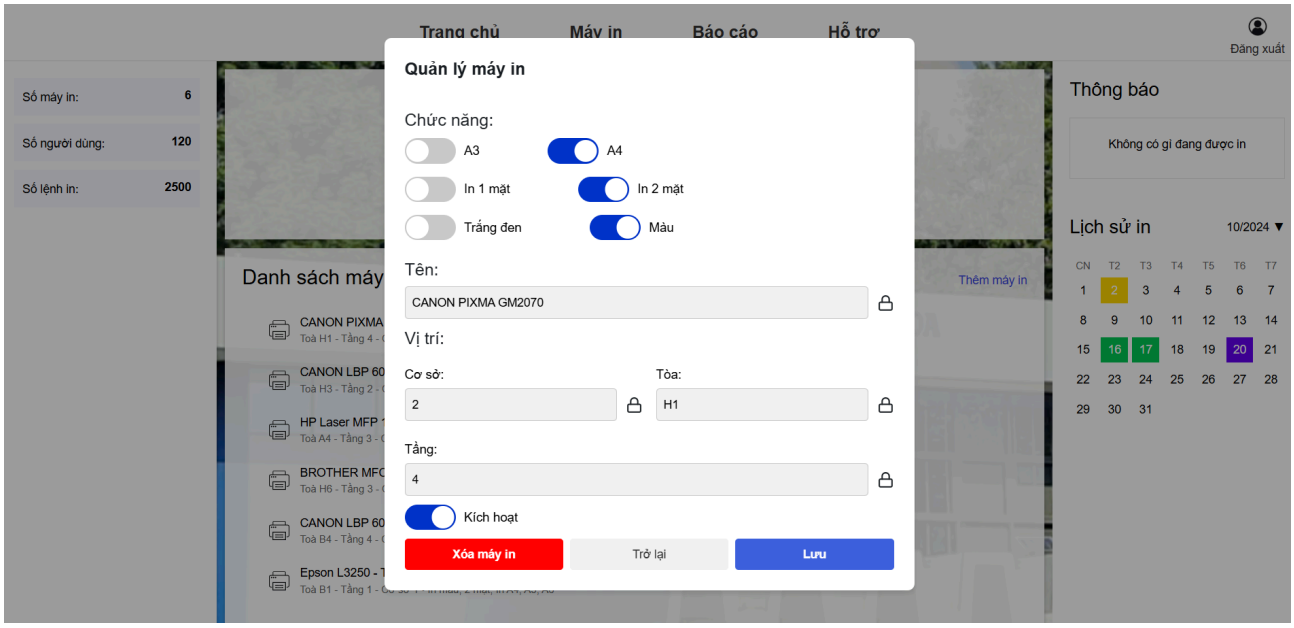


This is the main interface of SPSO, which is similar to the student interface but replaces the options "Thêm trang in" (to add a print page) and "In tài liệu" (print documents) on the navigation bar (top) with "Báo cáo" (report) and "Máy in" (printer) options.

Printer Management



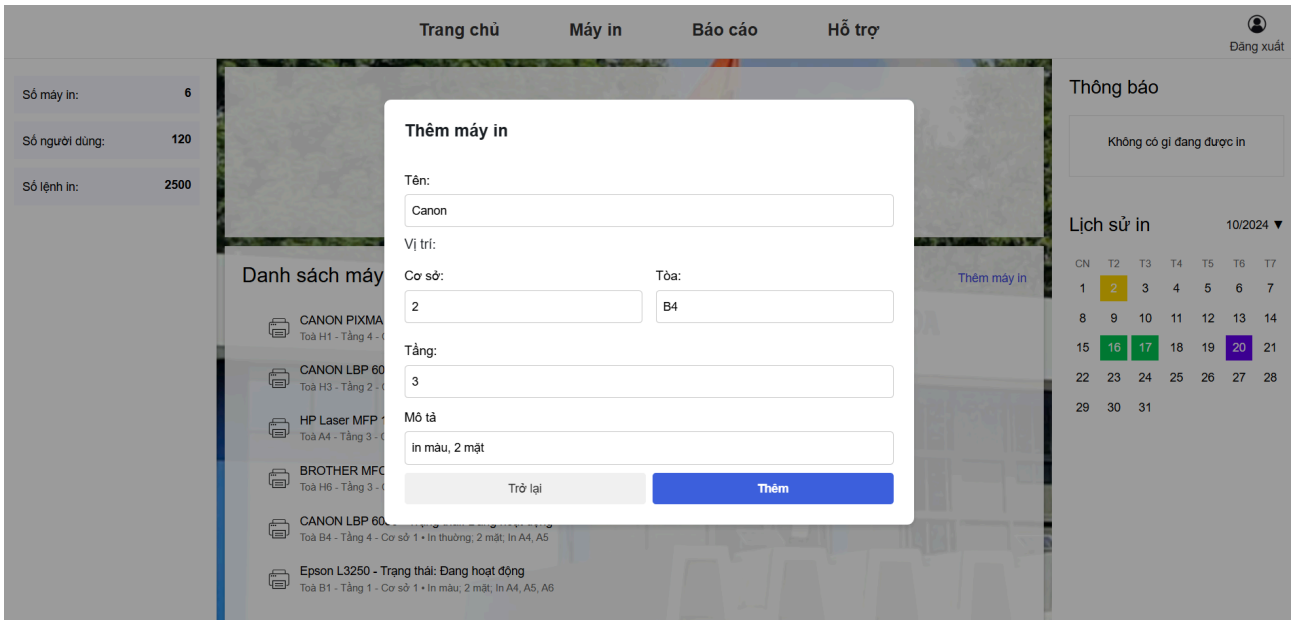
SPSO chooses the printer ("Máy in") option in the navigation bar and will receive the list of printers for monitoring printers' status. In the printer list ("Danh sách máy in") section, the printer list shows the information of each printer's information, including the printer's model name (e.g., "CANON PIXMA GM2070"), status ("Trạng thái"), location (e.g., "Tòa H1 - Tầng 4 - Cơ sở 2"), Features (e.g., "In màu; 2 mặt; In A4, A5"). There is an Add a printer ("Thêm máy in") option in the top right corner of the printer list section. This makes SPSO easy to add printers and makes their job easier to complete.



When clicking a printer, a form will be popped up on the screen to configure the information of that printer. The form includes the following fields:

- **Chức năng (Function)** has toggles switches to configure the page size (A3, A4), page side ("In 1 mặt", "In 2 mặt"), page color ("Trắng đen", Màu).
- **Tên (Model name)**
- **Vị trí (Location)** includes: "Cơ sở" (campus), "Tòa" (building), "Tầng" (floor)
- **Printer status** has a toggle switch that demonstrates that if the printer is enabled ("Kích hoạt") or disabled ("Vô hiệu hóa")

The control buttons include **"Xóa máy in" - Delete Printer** (red button) to remove a printer out of the system, **"Trở lại" - Back** (gray button) to return printer management step, and **"Lưu" - Save** (blue button) to save adjustments.



This is the interface for adding a new printer to the system after SPSO chooses the "Thêm máy in" (add a printer) option. The adding printer form includes the following fields:

- **Tên (model name):** Enter the name of the printer (e.g., Canon).
- **Vị trí (Location):**
    - **Cơ sở (Campus):** Enter the facility number (e.g., 2).
    - **Tòa (Building):** Enter the building name (e.g., B4).
    - **Tầng (Floor):** Enter the floor number (e.g., 3).
- **Mô tả (Description):** Enter the description of the printer feature (e.g., "In màu" (color printing), "2 mặt" (double-sided)).

There are two buttons at the bottom of the pop-up window:

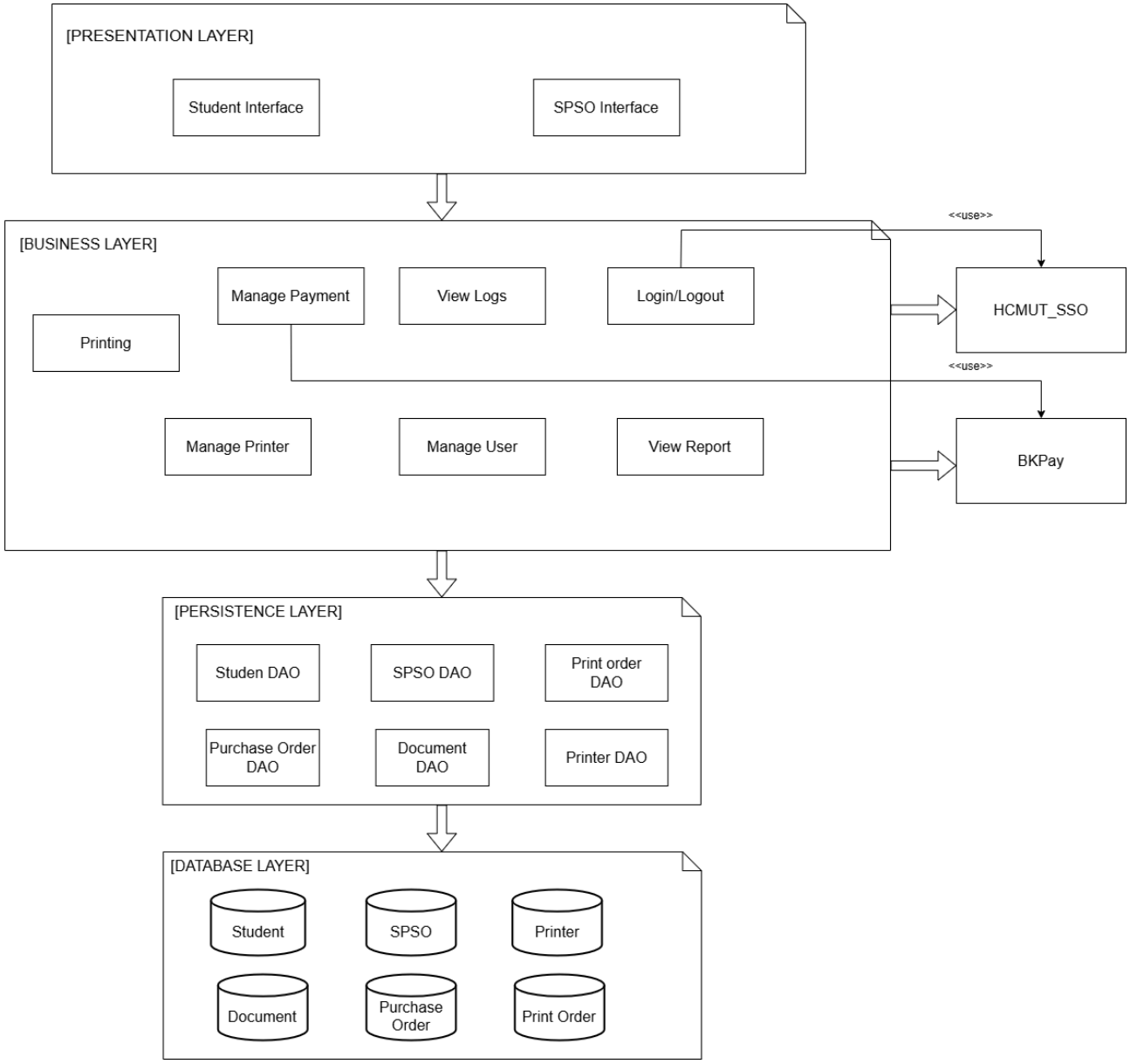
Trở lại (Back): This button is to go back to the home page.

Thêm (Add): This button is to add the new printer to the system and completes the add printer process.

4. Architecture design (Task 3)

4.1 Layered architecture

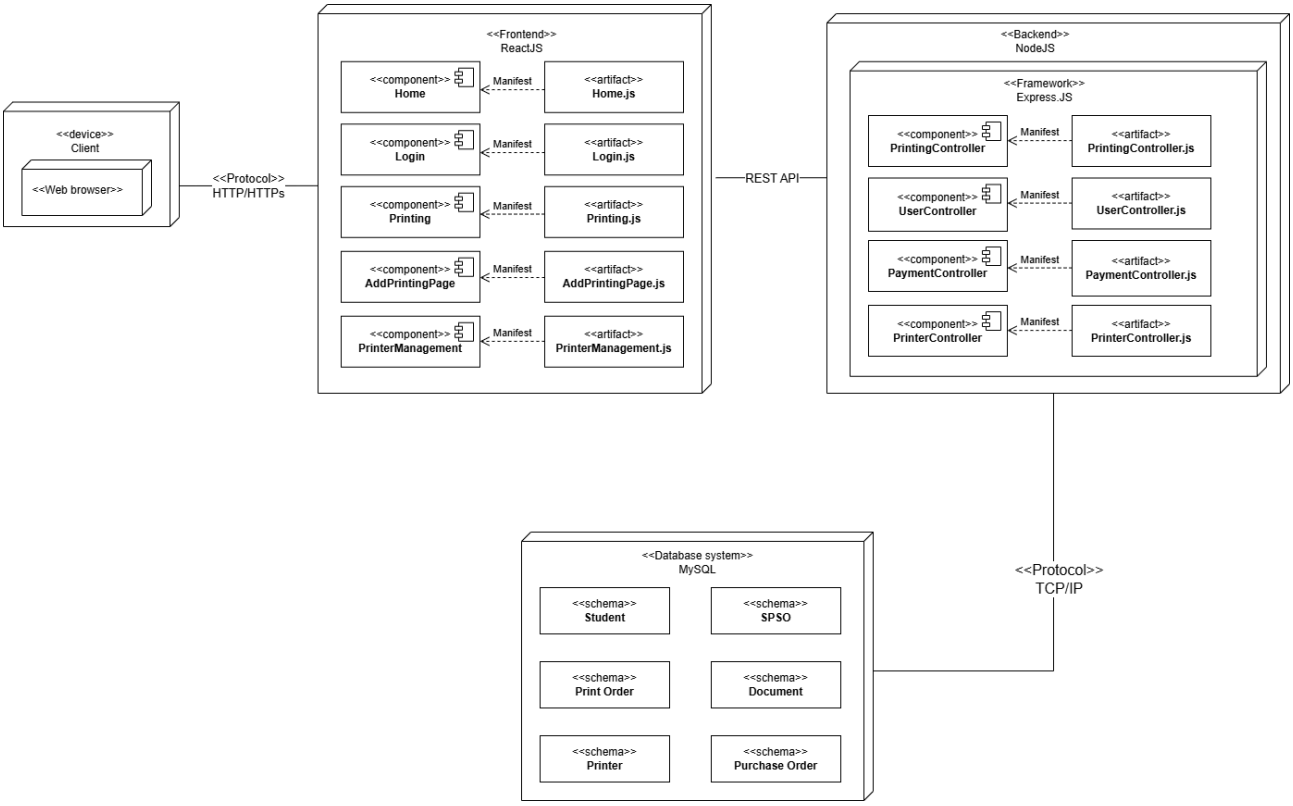
4.1.1 Layered Architecture Diagram



The diagram illustrates a four-layered architecture designed for SSPS. The **Presentation Layer** provides user interfaces for 2 types of user (Student and SPSO). The **Business Layer** serves as the core of the application, handling various business logic through different components. For example, the Printing manages document uploads and printing configurations, while the Login/Logout with HCMUT_SSO deal with authentication and user information display. The Manage Payment with BKPay handle purchase processes, ensuring that the business logic is well-separated into distinct modules. The **Persistence Layer** consists of several data access objects (DAO), each corresponding to key entities like Students, Printer, and Print order. This layer acts as an intermediary between the business logic and the actual data storage. The **Database Layer** stores structured data across multiple tables, supporting efficient access and data integrity for entities such as Student, Document, and Orders.

This architecture promotes scalability and maintainability by enforcing a unidirectional flow of data and separating concerns across layers, which prevents cyclic dependencies and facilitates easier updates to individual system components.

4.1.2 Deployment Architecture Diagram



When the user (client) accesses the system via web browser, which connects to the Front-end through HTTP/HTTPS protocols. The Front-end, built with ReactJS, consists of 5 components: Home, Login, Printing, AddPrintingPage, and PrinterManagement, each represented by corresponding JavaScript files. These components will render the site's layout and the user can interact with the system. The Front-end communicates with the Back-end via REST API, which sends HTTP requests (GET, POST, DELETE, POST methods) from the client. The Back-end, implemented in NodeJS and using the Express.js framework, includes 4 controllers for different functionalities: PrintingController, UserController, PaymentController, and PrinterController. The Back-end then connects to the database server (using MySQL as the Database Management System) through TCP/IP protocol, where data is physically stored in tables corresponding Data access objects, including: Student, SPSO, Print Order, Document, Printer, Purchase Order, and the relationship between entities. This deployment structure ensures efficient data flow and interaction across layers while maintaining a clear separation of Front-end, Back-end, and data storage components, facilitating scalability and maintainability of the system.

4.1.3 Presentation Strategy

Both students and staff at the Student Printing Service Office (SPSO) will find the Presentation User Interface strategy for the HCMUT-SSPS system to be smooth and easy to use. Each user group has its own unique module in the UI, guaranteeing that features are suited to their particular requirements. With further tools to add and manage print jobs, students can access the system via a login page, home page, and printing view page. On the other side, SPSO staff can effectively monitor and manage the printing services because they have access to a specialized login page, home page, and printer management page. By providing a user-friendly and effective interface that satisfies the various needs of all SSPS stakeholders, this dual interface design increases user satisfaction.

4.1.4 Data Storage approach

In this smart printing service for students at HCMUT, there are two distinct **user** types: **SPSO** and **students**. Each user has a unique **ID**, a **name** consisting of both **first** and **last names**, and login credentials using an **email** and **password**. However, students have an additional attribute called **available pages**.
When a student wants to purchase more printing pages, they make a payment through a **Purchase Order** entity, which includes a unique **ID**, transaction **date**, the **quantity of pages** purchased, and the cost. Furthermore, students can interact with a **Print Order** entity, which also has a unique **ID**, a **printing log**, and stores the printing history, including start time, completion, or cancellation.
Next is the **printing configuration,** where students can select the following options:

- Number of copies
- Printing color
- One/Double-sided printing
- Paper size
- Paper type

The **Print Order** entity contains the **Document** entity, which has a unique **ID**, **name**, **format**, and the **number of pages** required for printing. Finally, the **Document** entity connects to the **Printer** entity. The **Printer** entity includes a unique **ID**, **brand**, **short description**, **status**, **model**, and **location**, which specifies the **campus**, **building**, and **room** details.

Entity:

- User :
    - ID : Unique Identifier
    - Login :
        - Email
        - Password
    - Name:
        - F_Name
        - L_name
- Student :
    - Available Pages
    - Subclass of User
- SPSO :

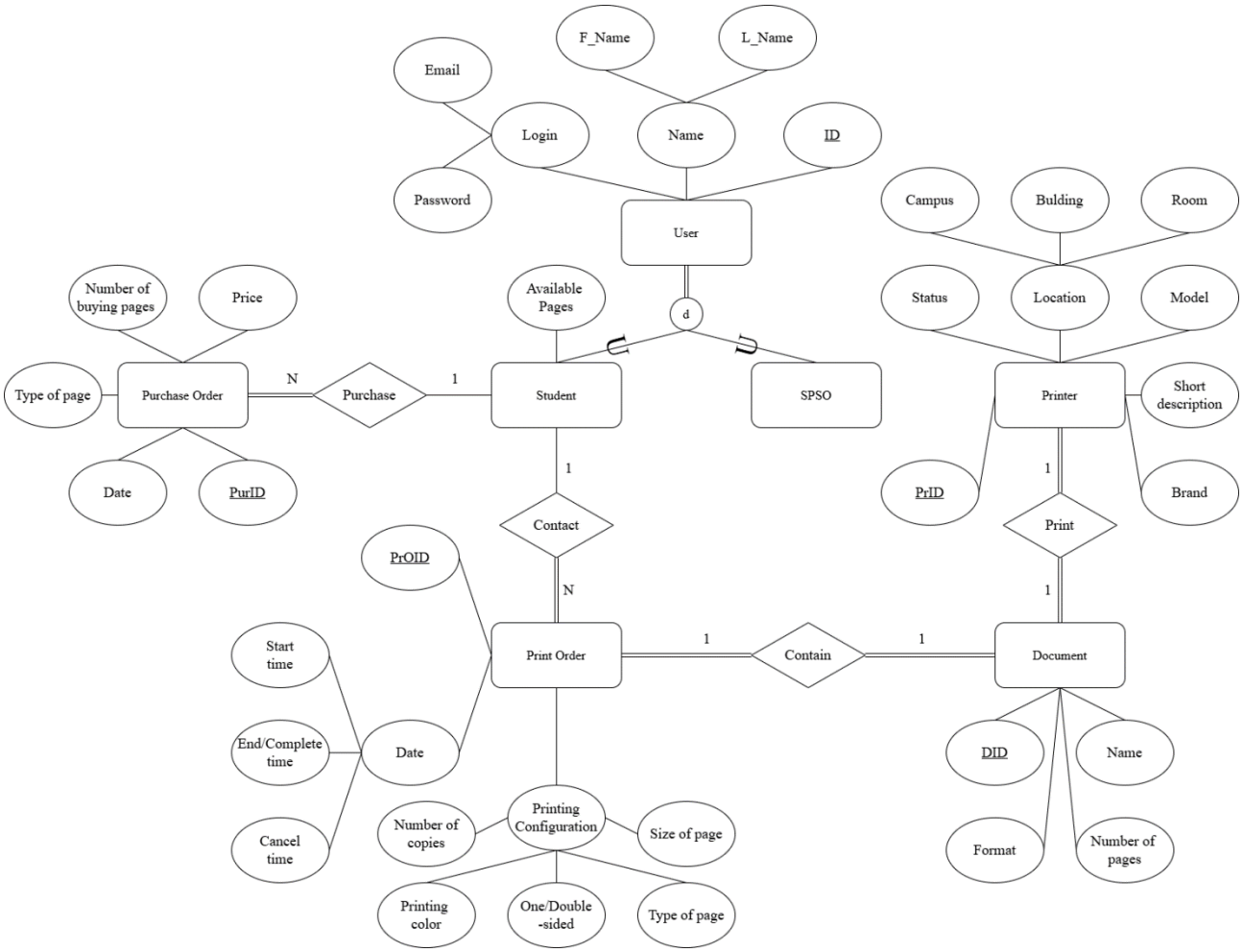
    - Subclass of User
- Purchase Order :
    - PurID : Unique Identifier
    - Price
    - Date
    - Number of buying pages
    - Type of page
- Print Order :
    - PrOID : Unique Identifier
    - Date :
        - Start time
        - End/Complete time
        - Cancel time
    - Printing Configuration :
        - Number of copies
        - Printing color
        - One/Double - sided
        - Type of page
        - Size of page
- Document :
    - DID : Unique Identifier
    - Name
    - Format
    - Number of pages
- Printer :
    - PrID : Unique Identifier
    - Status
    - Model
    - Location :
        - Campus
        - Building
        - Room
    - Short description
    - Brand

Relationship:

- Student - Purchase Order : One-to-many
    - A student can make many trades and must purchase for them.
- Student - Print Order : One-to-many

  - A student can contact many properties of the printing configuration and must choose it for the next process.
- Print Order - Document : One-to-one
  - A print order must contain only one document.
- Document - Printer : One-to-one
  - Only one document can be printed by a printer at the same time.

Entity Relationship Diagram



4.1.5 API Management

API for Authentication: This API handles user authentication using the HCMUT_SSO (HCMUT Single Sign-On) system. It ensures that only authorized users, such as students or SPSO administrators, can access the platform. Upon successful login, an access token is generated, which is required for accessing other APIs.

API for Document Upload: This API allows users to upload documents that they intend to print. Validates uploaded documents to confirm that they are in supported formats like PDF or DOCX.

This API also checks for file integrity to prevent errors during the printing process and ensures compatibility with the printers.

API for Document Formatting: Converts uploaded documents into a standardized format suitable for printing. If the user modifies print settings, this API reprocesses the document accordingly, providing a preview before printing to confirm user preferences.

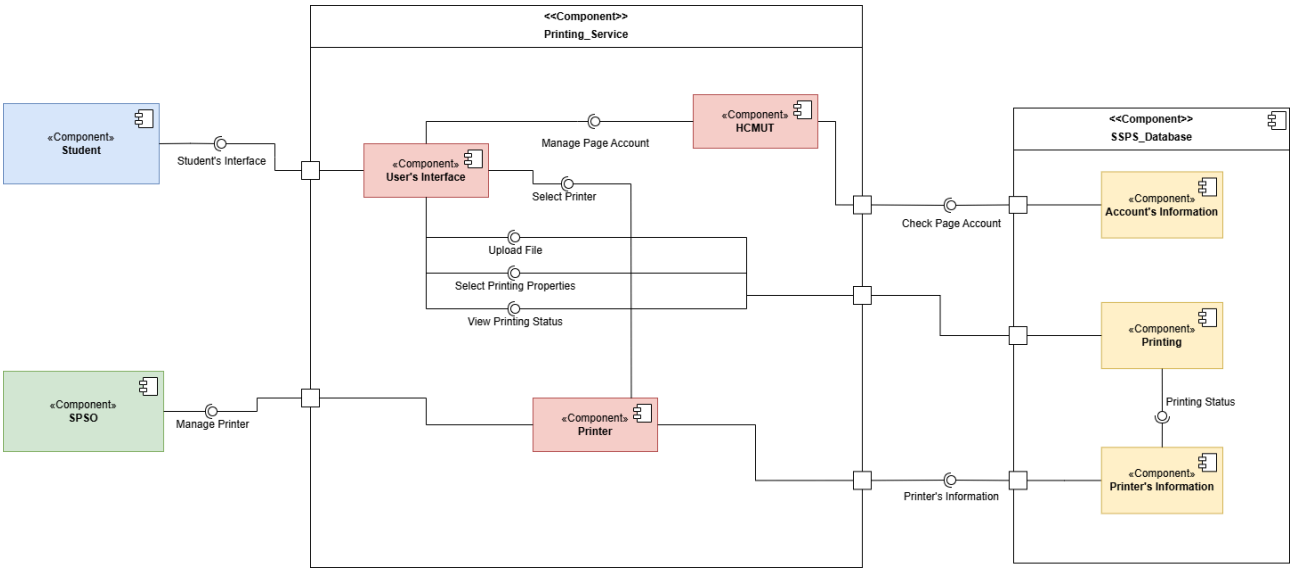
API for Printing Documents: Handles the actual printing job by sending user-configured documents to selected printers. It manages settings like page range, color options, and print quality to ensure the output matches user specifications.

API for Printer Management: Allows the SPSO to manage printer resources, including adding new printers, deleting outdated ones, and disabling malfunctioning units. It also tracks printer statuses to optimize availability and usage.

API for Purchasing Printing Pages: Provides a method for users to purchase additional printing pages using the BKPay system.

API for Printing History and Reports: Generates detailed logs of all printing activities for both users and administrators. It includes data on print jobs, user activities, and system performance, allowing for effective monitoring and auditing of the printing service.

4.2 Component Diagram



There are two components represent types of user such as Student and SPSO:

- **Student:** The Student component represents the user (typically a student) interacting with the printing service. It is provided with the "Student's Interface" as its main interaction point, allowing students to access printing functionalities through the User's Interface component like managing accounts and submitting print requests.

- **SPSO:** The SPSO (Student Printing System Operator) component is responsible for managing the printers. It interacts with the system to handle printer configurations, monitor printer status, and perform administrative tasks for the printing service through the Printer component.

The **Printing Service** layer is the central module coordinating all printing-related activities. It acts as the bridge between user interfaces, the database, and printers, ensuring smooth communication between the components. It includes three components which are User's Interface, HCMUT and Printer.

- **User's Interface:** The User's Interface component provides the Student component the interface through which students interact with the printing system. It supports various actions like managing page accounts, selecting printers, uploading files, setting printing properties, and checking printing statuses.

- **HCMUT:** The HCMUT component serves as an intermediary, facilitating specific account managing tasks. It connects with the SSPS Database component to assist in activities like account management made from User's Interface component.

- **Printer:** The Printer component represents the physical or virtual printers connected to the system. It handles tasks from User's Interface component such as receiving print jobs, executing print commands, and updating the printer's status in the system.

The **SSPS (Student Printing System) Database** layer stores and manages the essential data required for the printing service. It includes three components which are Account's Information, Printing and Printer's Information.

- **Account's Information:** This Account's Information Component manages student account data, including page credits and usage history. It supports the HCMUT component with the "Check Page Account" functionality.

- **Printing:** The Printing component is responsible for storing and retrieving details about ongoing and completed print jobs. It interacts with the Printing Service layer, specifically the User's Interface component to facilitate print operations.

- **Printer's Information:** This Printer's Information component manages metadata about printers, such as their configurations, statuses, and capabilities. It supports the "Printer's Information" functionality, ensuring the system as well as the Printer component always has up-to-date printer details.

5. Implementation – Sprint 1 (Task 4)

5.1 Github Repository

Our public repository is followed link: https://github.com/namhk4311/CO3001-SSPS

5.2 Usability Test

Usability Testing is the process of evaluating the user experience of a designed interface. This method is used with the following purposes: identifying design errors; uncovering areas for potential design improvements in the future; gathering feedback and observing user behavior to assess and refine the interface to better align with user needs and trends. Testing is conducted iteratively and regularly to continually enhance the UI and ensure it delivers the best possible user experience.

Designing a modern user interface involves numerous variables, and the complexity is amplified by the countless factors in human cognition. The sheer number of possible combinations is immense. The only reliable way to perfect UX design is through thorough testing.

The objectives of usability testing can differ depending on the study but typically include:

- Detecting issues in the design of a product or service
- Discovering opportunities for enhancement
- Gaining insights into the target user's behavior and preferences

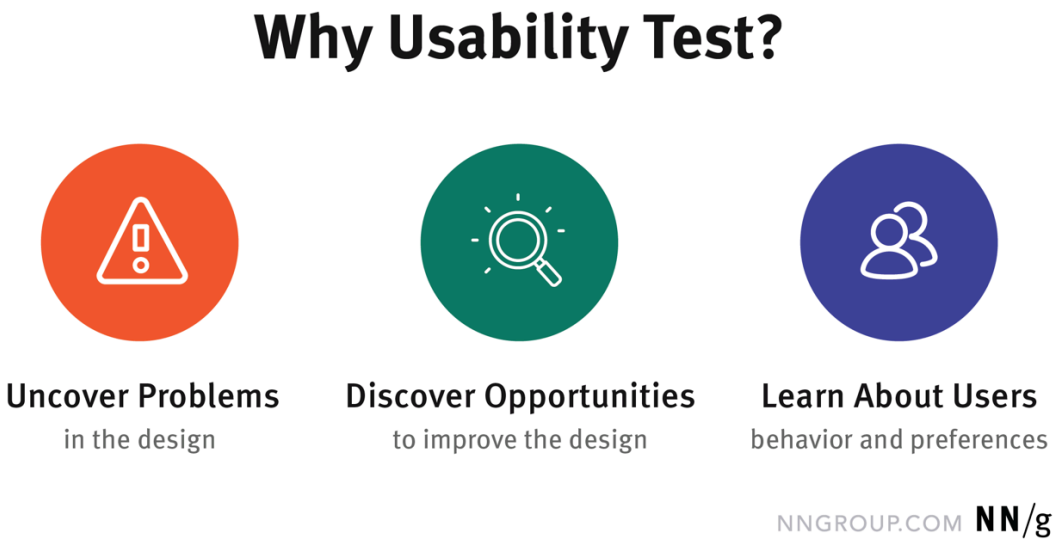


Image 1: Purpose of Usability Testing

The usability testing team conducts the process through the following steps:

5.2.1 Selecting the Tester and Participants

The tester (also referred to as the facilitator or moderator) is responsible for guiding participants through the evaluation steps, answering their questions during the testing process, recording observations, and analyzing the collected data to improve the product. This individual assigns tasks to participants and oversees their execution.

The participants are individuals recruited to test the product by completing tasks set by the tester. Tasks may be communicated in two forms: orally or in written instructions. For both methods, participants are asked to articulate their thoughts and actions, enabling the tester to better understand their behaviors, expectations, thought processes, and motivations. These participants act as representatives of the end-user group.

For this usability test, aimed at identifying the majority of interface issues in HCMUT_SSPS, the team will involve one tester and ten participants.

5.2.2 Defining the Tasks

Tasks are practical activities designed to evaluate participants' ability to use the interface. They can range from highly specific to open-ended questions, depending on the study's objectives.

The tasks must be written clearly with unambiguous and easy-to-understand language, avoiding confusion or misinterpretation. This ensures accurate results by preventing discrepancies between participants' understanding and the intended meaning of the instructions.

The team will test the document printing function and will therefore design the following tasks:

Task	**Task Description**
Test the printing function as a user or SPSO.	Test the login interface as a regular user or SPSO.
Test the print command initialization process.	Test the interface for creating a print command, uploading a file, and selecting a printer.
Test the ability to set print options.	Set up and confirm different print options for each participant.
Test the functions dedicated to SPSO.	Display the printer list and configure printer settings in the printer management interface.
Test print command completion	Test the homepage status after completing the print command.

5.2.3 Define the Testing Strategy

The testing strategy is the approach that the tester wants to evaluate through various testing methods.

(a) **Testing Approach**

There are two testing approaches for usability: qualitative and quantitative.
• **Qualitative Testing**: Focuses on the thoughts, emotions, and actions of participants as they use the product. Qualitative testing is most suitable for identifying issues in the user experience.
• **Quantitative Testing**: Focuses on metrics that describe the user experience, including the number of tasks completed and the time taken to complete each task.

(b) **Testing Method**

The evaluation can be conducted using two methods: direct or remote. Direct evaluation is a face-to-face method between the tester and the participant. Remote evaluation is divided into two types: "Supervised" and "Unsupervised."
• **Supervised Method**: Similar to direct evaluation, both methods require direct communication between the developer and the participant. However, remote supervision differs from direct evaluation in that the developer and the participant are in different geographical locations and must communicate using supporting software such as Google Meet, Zoom, etc.
• **Unsupervised Method**: Uses supporting application software to build tasks. This software acts as the publisher, providing, guiding, and assisting participants in completing tasks.

Based on the theoretical foundations of these methods, the group chooses the unsupervised remote testing method to evaluate product quality due to its speed, convenience for participants, no concerns about geographical barriers, and flexibility in timing. The group will also use qualitative testing to comprehensively understand the user experience and partially apply quantitative methods by tracking the number of tasks completed.

5.2.4 Conducting Testing

After identifying the testers, participants, tasks, and suitable strategies, the team proceeds with the testing process. To maximize efficiency and save time and costs, the team will conduct usability testing with 5 participants, up to a maximum of 3 iterations. Feedback from participants will be collected through Google Form following this link: https://docs.google.com/forms/d/e/1FAIpQLSfWVCvHNl9L2hy6OT35nsboFLXIgraO-EBn7nsIIJNALyKfpA/viewform. After the user testing group provides feedback in the previous round of testing, the team will revise the MVP1 for the next round of testing. This process will be repeated until all parties are satisfied with the assigned tasks or until a maximum of 3 iterations is reached.

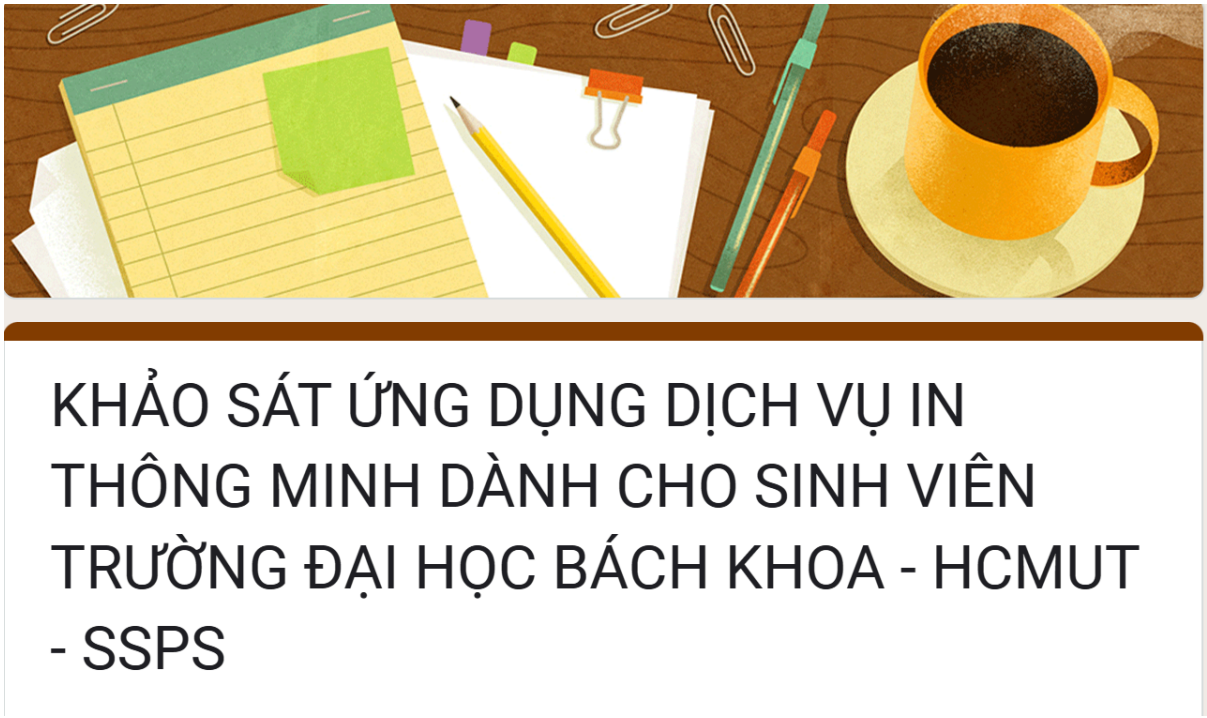


Image 2: SURVEY ON SMART PRINTING SERVICE APPLICATION FOR STUDENTS OF HCMUT - SSPS (for participants)

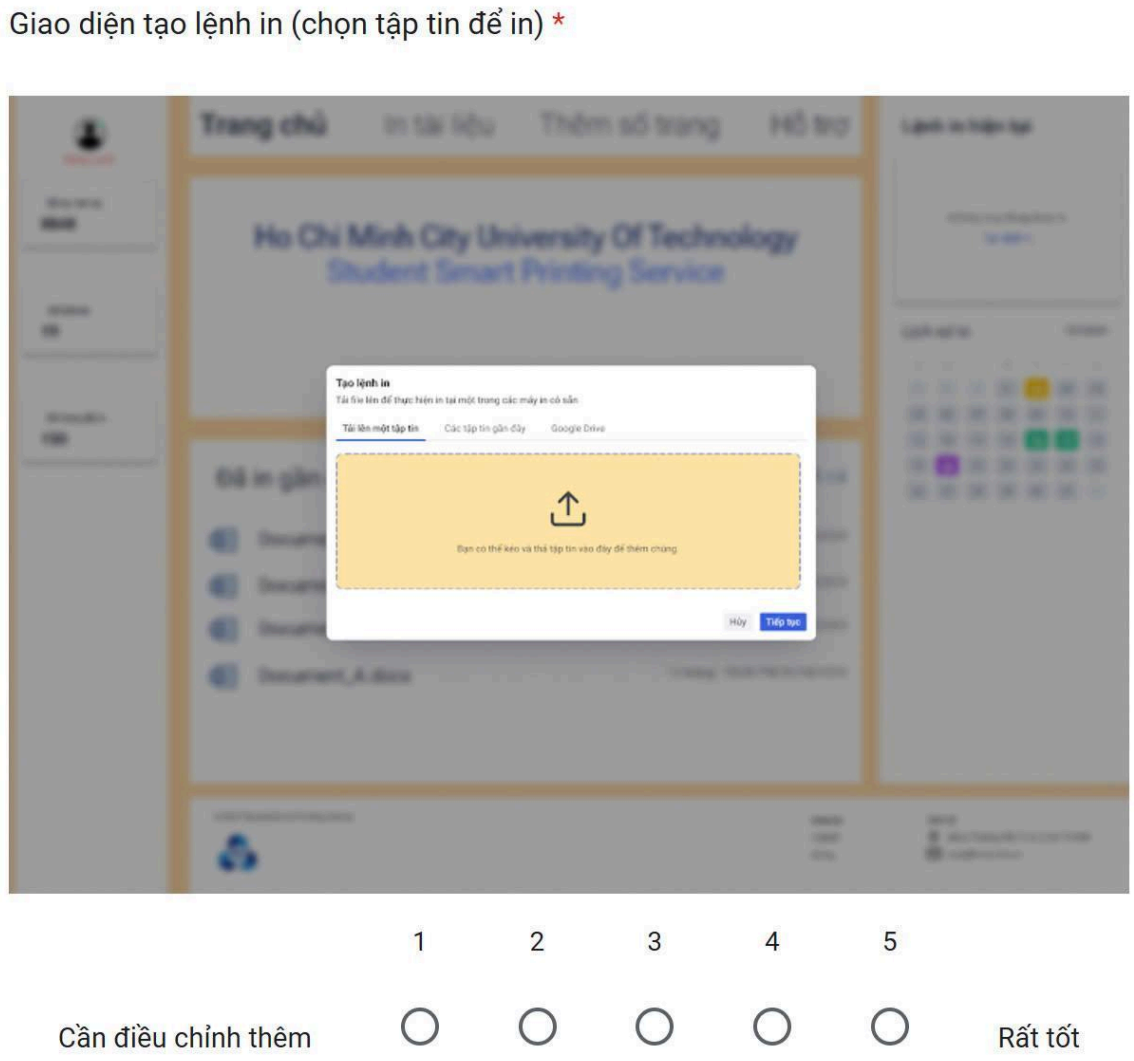


Image 3: Illustration one of the Tasks needs to be completed

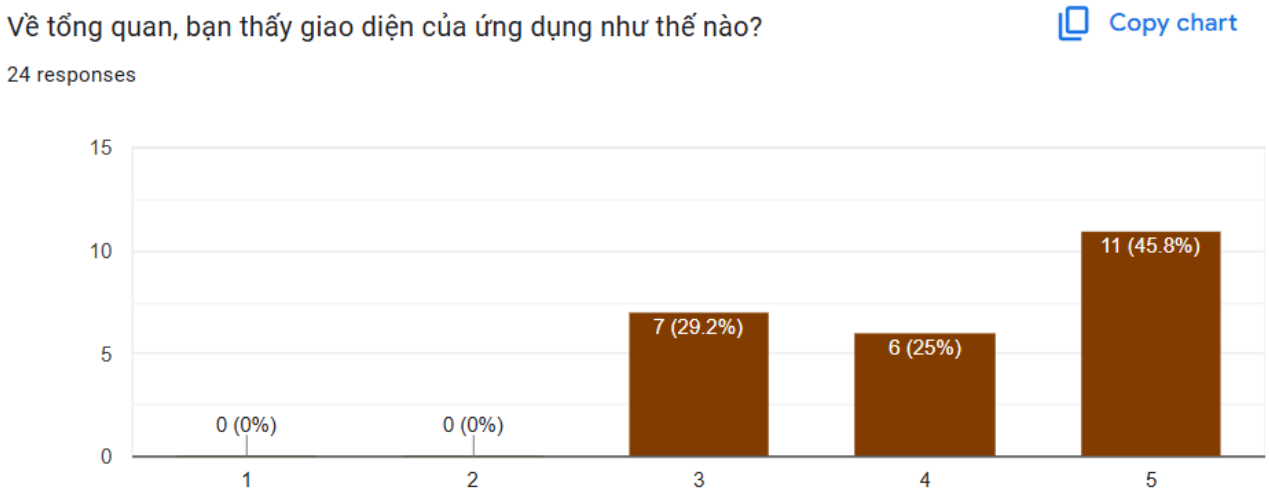
5.2.5 Collecting Feedback and Reporting Results

After conducting usability testing based on the predefined tasks and strategies, the team focuses on gathering feedback from the participants. This process helps the team gain a deeper understanding of user experiences and identify areas for improvement in the printing system. Based on the results, the team can adjust the HCMUT_SSPS interface to ensure it best meets the users' requirements and expectations.

In the first round of testing, the general feedback from participants showed satisfaction with the HCMUT_SSPS interface. However, there were still some concerns from a few participants regarding the login section, printer management, which needs to be enlarged, and the print command creation section, which requires further review.

The tester actively received feedback and made specific adjustments to meet the needs of the participants. Subsequently, the printer management, login, and print command creation sections were enlarged a bit more.

After the next round of testing, the results showed that all participants expressed satisfaction with the improvements that had been made. This demonstrates that the adjustment and enhancement process effectively met the users' expectations, creating a positive user experience for all test participants. The bar chart below illustrates the overall user feedback on the application's interface, based on a survey of 24 respondents. No respondents rated the interface as 1 or 2. Meanwhile, 7 users (29.2%) rated it as 3, 6 users (25%) rated it as 4, and the majority, 11 users (45.8%), rated it as 5, indicating a generally positive perception of the application's interface.



Information about the job descriptions, the number of participants for each task, and their dissatisfaction feedback (if any) has been recorded by the team in the picture below for easy tracking and to identify which features have met or not met the requirements of the website.

Task	Task Description	Participants	Desired result	Recorded results					Current results	Reason for dissatisfaction (if any)
				First test	Implementation date	Second test	Implementation date	Third test		
Test the printing function as a user or SPSO.	Test the login interface as a regular user or SPSO.	24	Users are satisfied with the interface and the login process.	F	23/11/2024	P	24/11/2024		P	"The username and password fields on the login page should be slightly enlarged."
Test the print command initialization process.	Test the interface for creating a print command, uploading a file, and selecting a printer.	24	Users can easily create print commands and upload files.	F	23/11/2024	P	24/11/2024		P	The print command creation section should be reviewed.
Test the ability to set print options.	Set up and confirm different print options for each participant.	24	Users can easily customize print settings and adjust configurations as desired.	P	23/11/2024	P	24/11/2024		P	
Test the functions dedicated to SPSO.	Display the printer list and configure printer settings in the printer management interface.	24	SPSO finds the interface user-friendly, making functional operations easier to perform.	F	23/11/2024	P	24/11/2024		P	"The printer management section should be enlarged as well."
Test print command completion	Test the homepage status after completing the print command.	24	SPSO finds the interface user-friendly, making functional operations easier to perform.	P	23/11/2024	P	24/11/2024		P	

From the results of the usability testing process, the team has gained more experience and made further adjustments to ensure the system runs more smoothly.

6. Lesson from doing this project

After completing the project, we gained many valuable lessons throughout the process. First of all, we learned how to use and create Function and Non-functional diagrams, as well as designing the Activity, Sequence, and Class diagrams to detail the system's behavior and structure. This gave us a comprehensive and clear understanding of the operational flow of the system from the initial design stage. Additionally, applying the Layered Architecture and Deployment Architecture to design the user interface (UI) significantly improved flexibility and reusability in various UI components.

In terms of system programming, our group utilized ReactJS, using Typescript for the front-end, which provided a flexible and robust framework for error checking. Furthermore, NodeJS with the Express Framework helped us build the backend efficiently. At the database layer, we chose MySQL for the database server to manage data with high consistency and reliability. This allowed us to enhance our skills in developing a full-stack environment.

Moreover, the group learned how to organize data, manage versions, and collaborate effectively using Git and GitHub as the versioning control system for the project. Clear task delegation and defined roles within the group helped us complete the work on schedule. However, there are some challenges that arise, such as synchronizing the front-end and backend or resolving conflicts on GitHub. These were gradually overcome through good communication and teamwork.

Overall, the project helped our team improve both technical and soft skills, from programming and design to time management and group communication. This experience has laid a valuable foundation for future projects.