Software Requirements Specification

for

Student Smart Printing Service (HCMUT_SSPS)

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
TN01-03	18/09/2024	First version	1.0
TN01-03	15/11/2024	Major changes in system modeling diagrams	1.1

1.   Requirement elicitation

1.1   Domain Context

The HCMUT_SSPS is a smart printing service designed for the students of Ho Chi Minh University of Technology (HCMUT). It provides students with the capability to print their documents conveniently via laptops or smartphones. Additionally, HCMUT_SSPS enables students to easily monitor their printing history, ensuring greater control and transparency over their printing activities

1.2   Stakeholders and Needs

1.2.1. The University
They need a system that provides automatic document printing service for students in its campus.
1.2.2. Student Printing Service Officer (SPSO)
They need to manage and monitor the system's operations and resources, i.e., printers.
1.2.3. Students
They need to print documents by uploading document files onto the system, choosing a printer, and specifying the printing properties.
1.2.4. BKPay
They need to be integrated into the system to provide online payment services so that students can purchase pages online safely and conveniently.
1.2.5. HCMUT_SSO
They need to be capable of authenticating all users before giving them access to the system.

1.3   Benefits of the System

1.3.1. The University:
- Automate printing
- Reduce Workload
- Support sustainability
1.3.2. SPSO:
- Easily monitor and manage printers
- Ensuring smooth operation
1.3.3. Students:
- Print the documents from their devices
- Track their usage
- Customize print settings
1.3.4. BKPay:
- Allow students to pay for printing online safely
- Increase its usage
1.3.5. HCMUT_SSO:
- Ensure secure access for authorized users

- Make login and system access simple and safe

1.4   Functional Requirements

1.4.1. System requirements
- The system shall generate reports of the use of the printing system automatically at the end of each month and each year
- The system shall send notifications to SPSO via email when a printer has a problem

1.4.2. Student Printing Service Officer (SPSO)
- The SPSO has access to and views information on all printers available in the system, including transactions made, number of pages available, and status of current transactions.
- The SPSO can configure each printer, such as the start and end time of the day for a printer, and the file format the printer allows.
- The SPSO can enable or disable any printer in the system.
- The SPSO can add or delete printers in the system.
- The SPSO can manage system configuration, such as determining the file format that students are allowed to upload, etc.
- The SPSO can view students' transactions, including students' information.

1.4.3. Student
- Students can access and view transaction history, and information about printing times.
- Students can select files from the device and upload them to the system
- Students can view unpaid transactions.
- Students can choose the appropriate printer at the desired location.
- Students choose the time to receive the printout.
- Students can choose print-related customizations such as pages to print, paper size, number of prints, single-sided or double-sided.
- Students can view the status of the printing, including pending confirmation, printing, completed, or not completed.
- Students can view the printing status. In case of an error during printing, students will receive an email

1.4.4. BKPay
- The system stores the history of successful transactions, including student ID, amount, and date.
- Confirm successful payment, update the system history, and send the receipt to the student's email.
- Display recipient information (account number, QR)
- The printer confirms the transaction status as completed or not via BKPay
- Save unprocessed transactions.

1.4.5. HCMUT_SSO
- All students and SPSOs have to authenticate through the HCMUT_SSO authentication system.

1.5 Non- Functional Requirements

1.5.1. Performance efficiency
- Response time should not exceed 10 seconds
- The system must confirm successful payment within 1 minute
- The system should operate at its optimal performance even if 6000 users are accessing the system concurrently

1.5.2. Compatibility
- The system should support both web and mobile platforms

1.5.3. Usability
- The system should be foolproof: It takes at most 2 hours to learn how to use it properly

1.5.4. Reliability
- The system shall be available during normal working hours unless it is under maintenance
- The error rate when using the system must not exceed 2%
- During normal working hours, the system downtime must not exceed 1 hour should an error occur

1.5.5. Security
- The system must authenticate all users using the HCMUT_SSO service before allowing them to access the system
- Documents stored on the system can only be accessed by their owners

2. Use-case Diagrams

## 2.1 Use-case Diagram for the Whole System

2.1.1. Actors

Actor ID	Actor	Description
1	Student	
2	SPSO	Student Printing Service Officer
3	Printer	
4	HCMUT_SSO	HCMUT's Single Sign-On Service
5	BKPay	HCMUT payment service

2.1.2. Use case

Use case ID	Use case name	Description	Notes

UC-1	Authenticate	Students log in or log out to the service	
UC-2	Manage print operations	Students view printing history or print documents	Important
UC-2.1	Print documents	Students choose printer locations to print their documents, provided there is enough paper	
UC-2.2	View printing history	Students can view how many times and when they have printed documents	
UC-3	Manage transactions	View purchase history or purchase BKPay	Important
UC-3.1	Purchase papers	Students buy more papers	
UC-3.2	View purchase history		
UC-4	View monthly and annual reports	The SPSO views the reports automatically generated by the system	
UC-5	View students' transactions		
UC-6	Manage printers		

2.1.3. Diagram
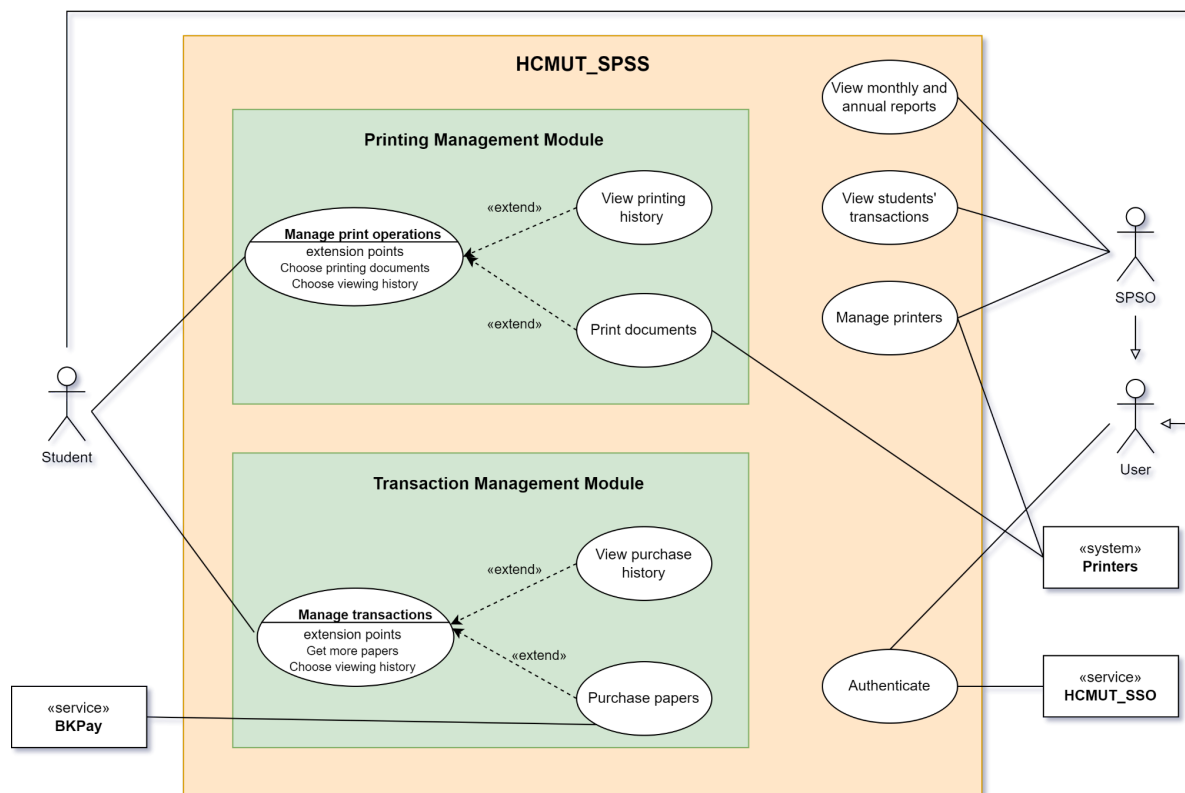
2.2   Use-case Diagram for Printing Management Module
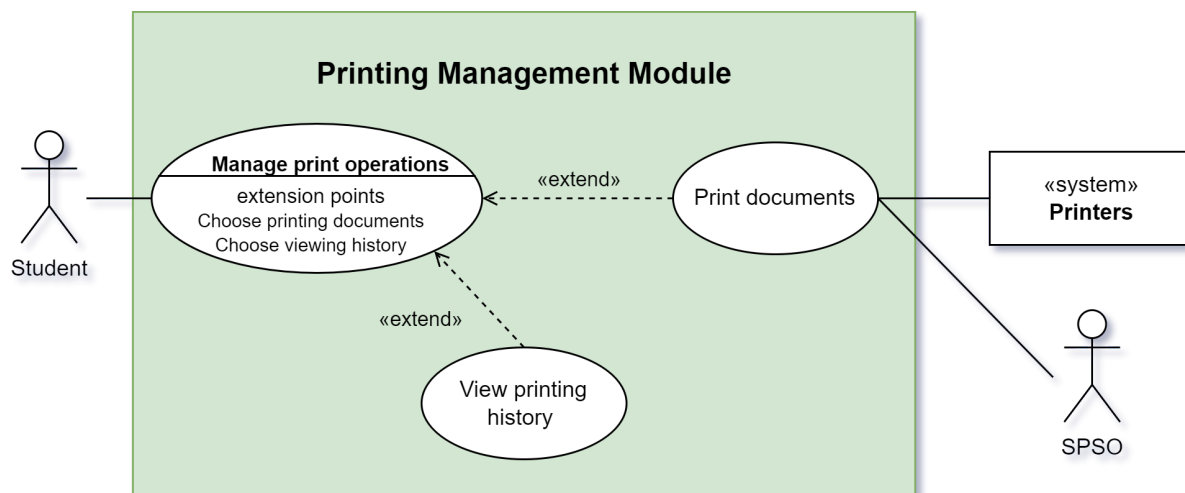
2.3   The Details of Use cases in Printing Management Module

1.   Use case: Manage print operations

Use case ID:	UC-2

Use case name:	Manage print operations

Created by:	Lê Văn Anh Khoa	Last updated by:	Lê Hồng Phúc
Date Created:	27/09/2024	Date Last Updated:	15/11/2024

Actors:	Student
Description:	Allows the student to manage printing operations, including viewing print history and printing documents.
Trigger:	The student opens the printing management page.
Extends use cases:	1. Print Documents 2. View Printing History
Preconditions:	1. The student is logged into the system
Postconditions:	1. Students' requests are processed successfully 2. The print history is updated and viewable.
Normal Flow:	1. The student opens the print management page. 2. The system displays 2 options: Viewing printing history or printing a document. 3. The student chooses to print a document. Use case continues in UC-2.1
Alternative Flow:	3a. The student chooses viewing history Use case continues in UC-2.2 3b. The student choose returning to main page Use case ends.
Notes and issues:	None

2. **Use case: Print documents**

Use case ID:	UC-2.1		
Use case name:	Print documents		
Created by:	Lê Phúc Hoàng	Last updated by:	Lê Hồng Phúc
Date Created:	29/09/2024	Date Last Updated:	15/11/2024
Actors:	Student		
Description:	Students upload and print documents		

Trigger:	The student requests to upload and print documents
Preconditions:	1. The student is logged into the system
Postconditions:	2. The documents are uploaded and printed
Normal Flow:	1. The student opens the print management software. 2. The student selects "Printing Documents". 3. A page appears letting students upload their documents. 4. The pages show available printers and printing properties. 5. The students choose a printer, and printing properties and confirm their printing. 6. The documents are printed and collected by the printer. 7. The operations are recorded in printing history.
Exception Flow	3a. If the file type is not permitted by SPSO, the system will deny that file and the students need to upload another file. 5a. If the students have not chosen printers, or printing properties or have not uploaded any files yet, the system shows a warning that the students need to complete the missing. 5b. If the printer does not have enough pages, the system shows a warning that the printer does not have enough pages. 5c. If the student does not have enough required pages, the system shows a warning that the student does not have enough required pages.

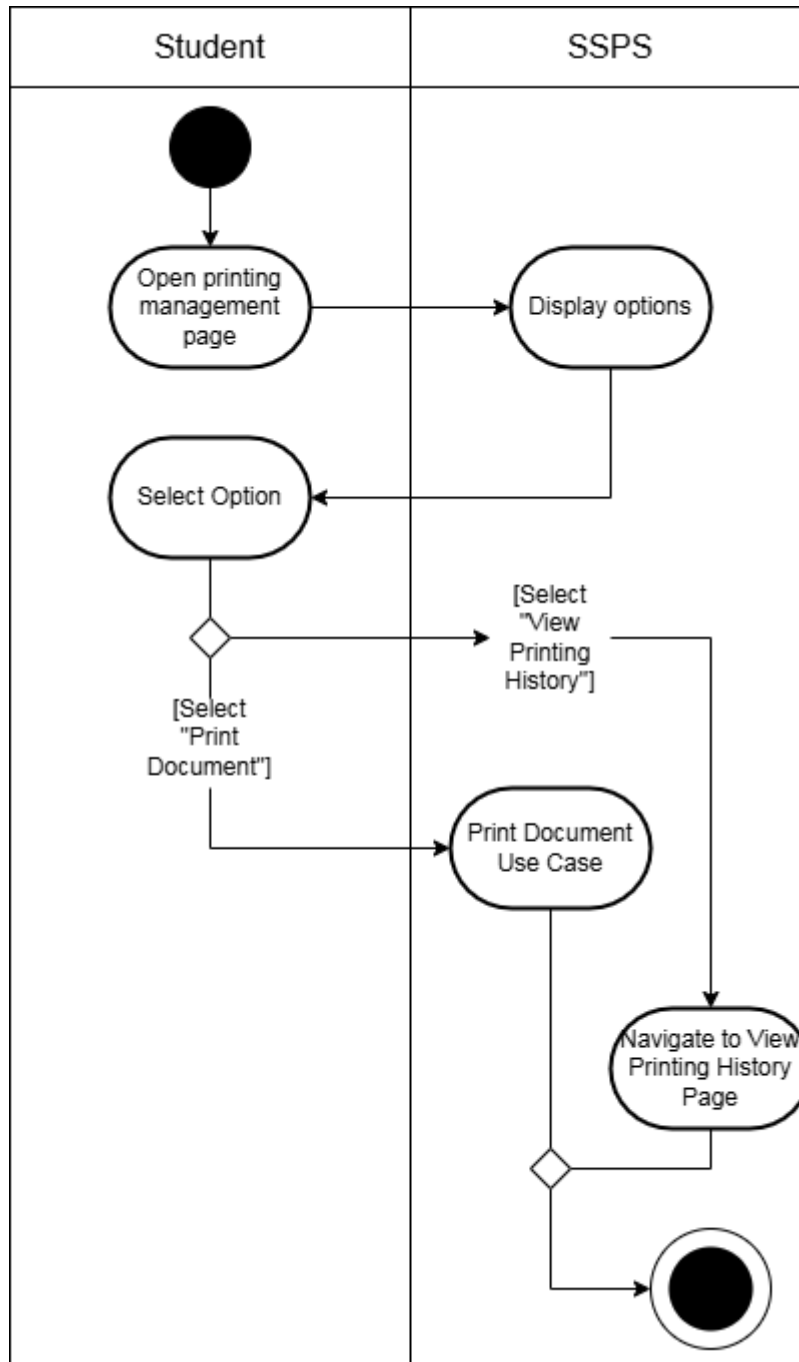## 3. Use case: View printing history

Use case ID:	UC-2.2		
Use case name:	View Printing History		
Created by:	Huỳnh Lê Đăng Khoa	Last updated by:	Lê Hồng Phúc
Date Created:	27/09/2024	Date Last Updated:	15/11/2024
Actors:	Student		
Description:	Students view information about print history		
Trigger:	The student opens the printing history page.		
Preconditions:	1. The student is logged into the system		
Postconditions:	1. The print history is updated and viewable.		

|	
Normal Flow:	1. The student opens the print management software. 2. The student selects "View Printing History" 3. A page appears showing all student printing. 4. The student chooses to see the printing history overview. 5. The page shows an overview of the prints that have been visualized.
Alternative Flow:	4a. The student chooses the sorting criteria and clicks confirm. 5a. A page appears showing all student printing sorted.

3. System Modelling

3.1. Activity diagrams

3.1.1. Manage print operations

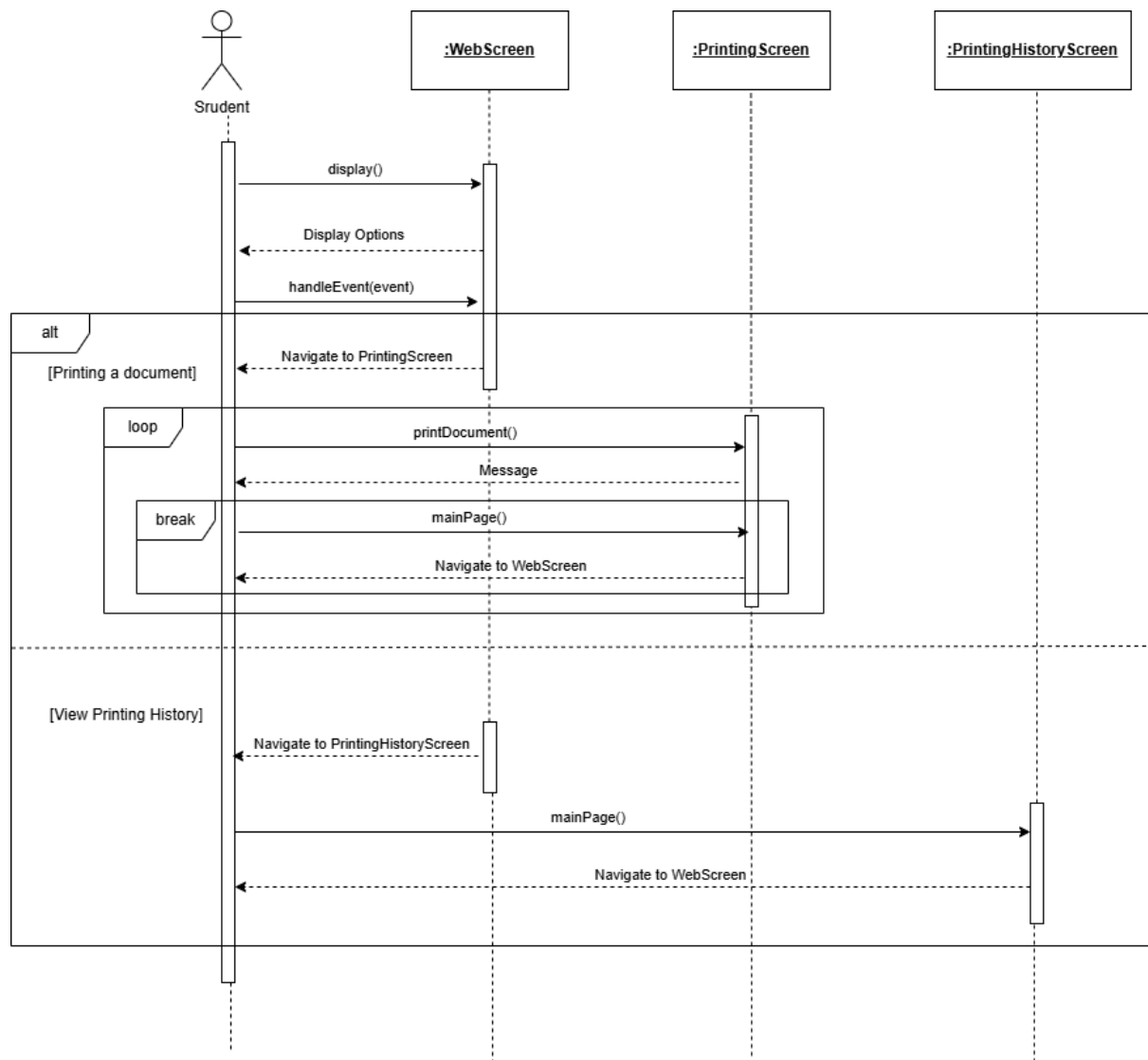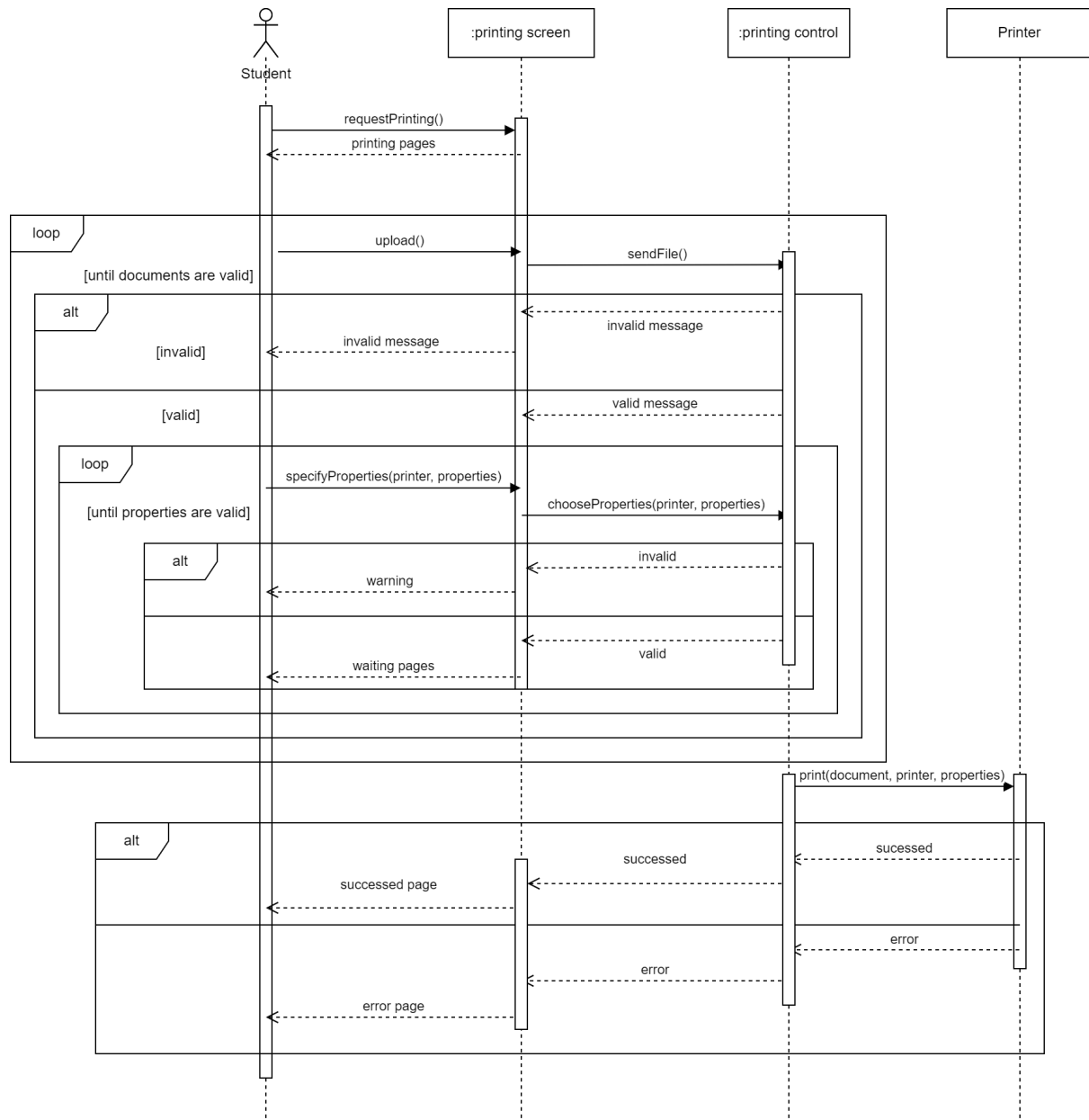### 3.1.2. Print documents
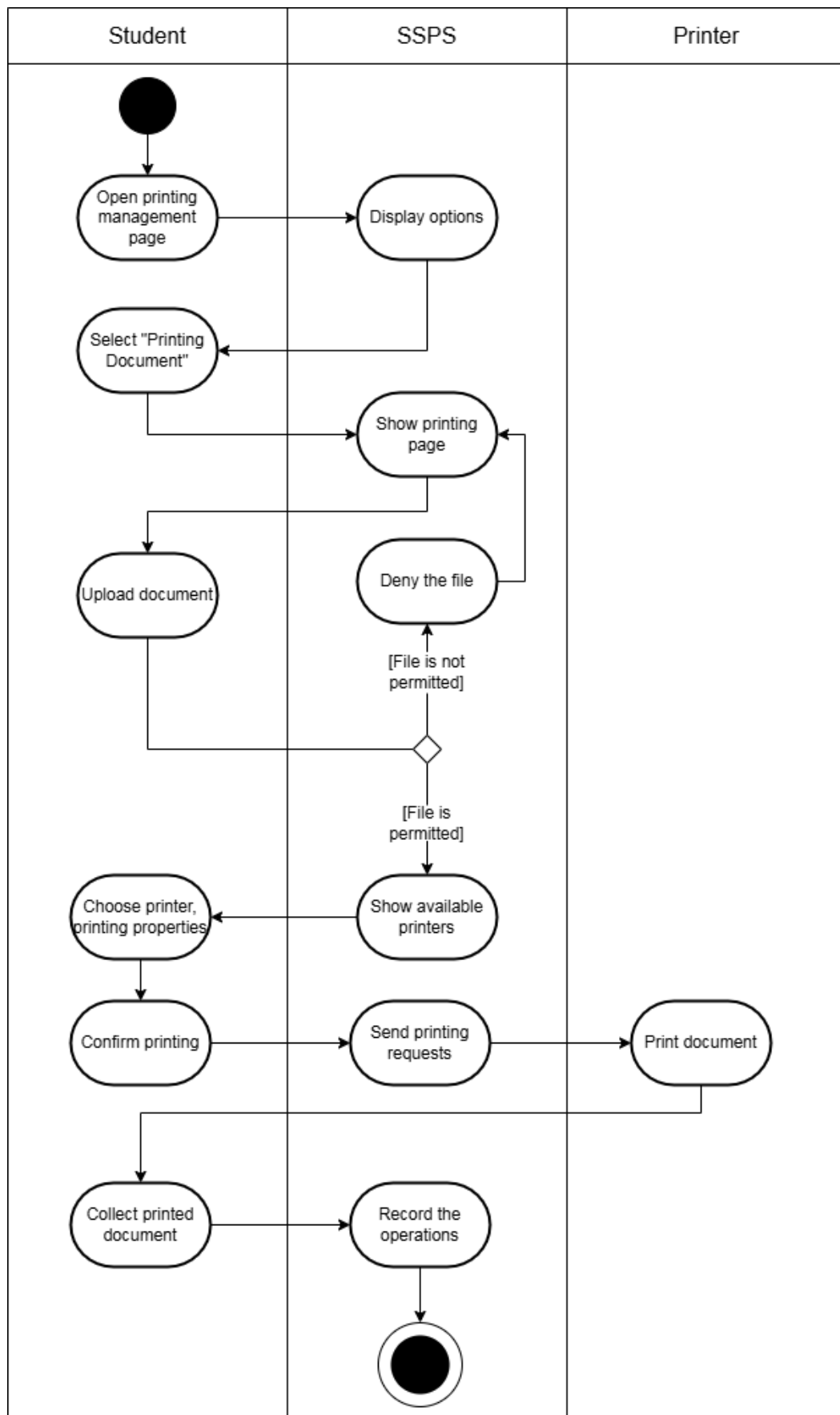
3.1.3. View printing history
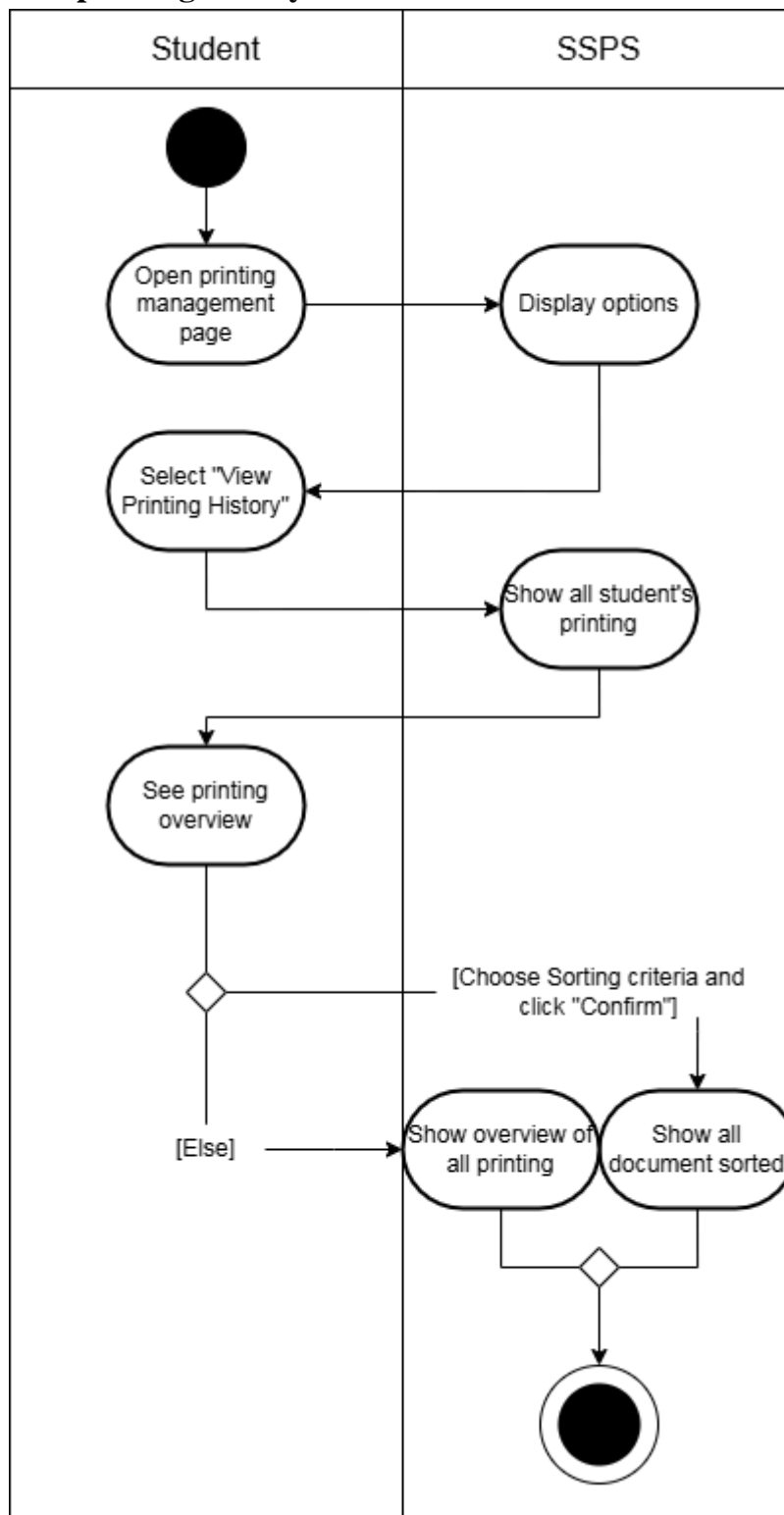
3.2. Sequence diagrams
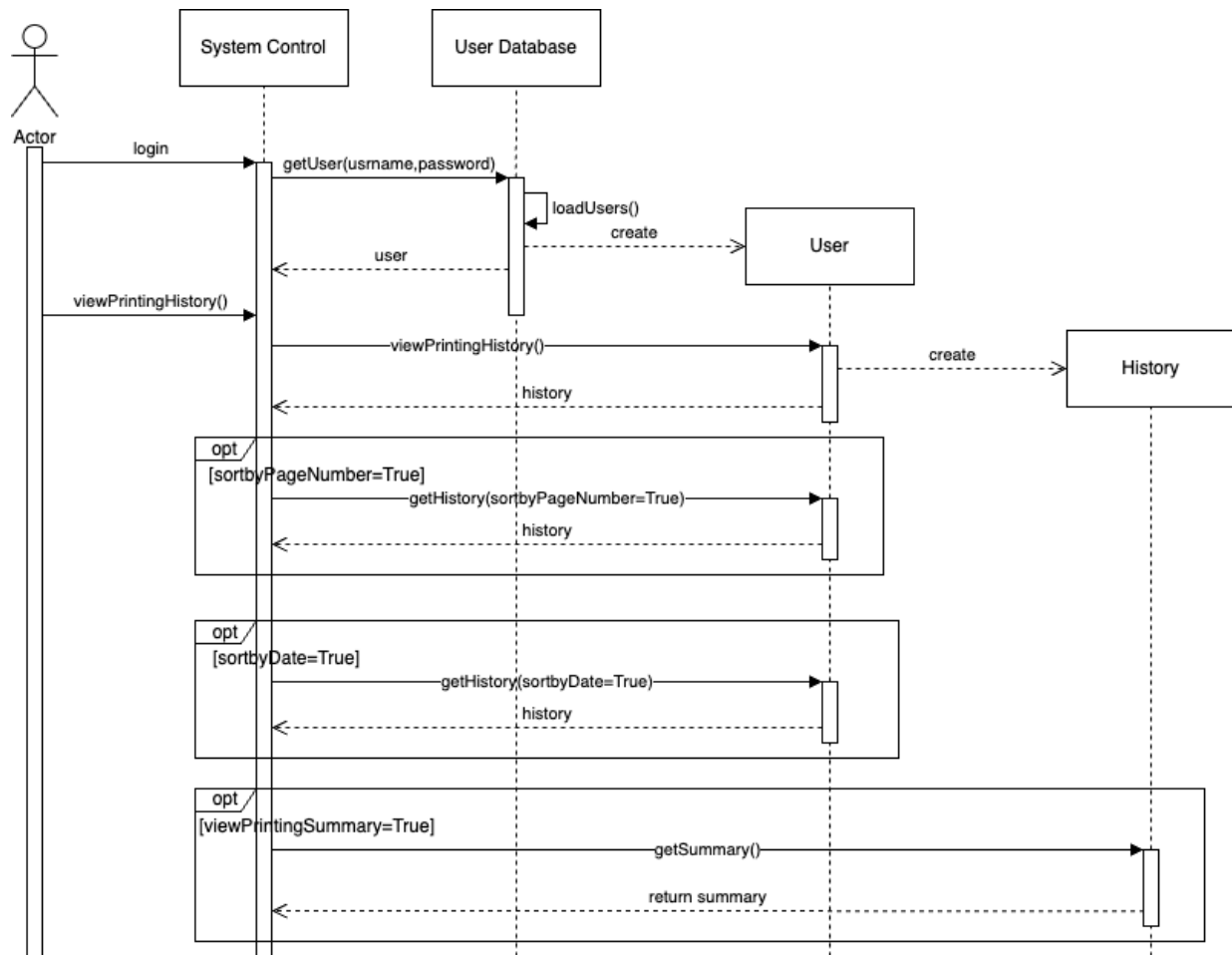
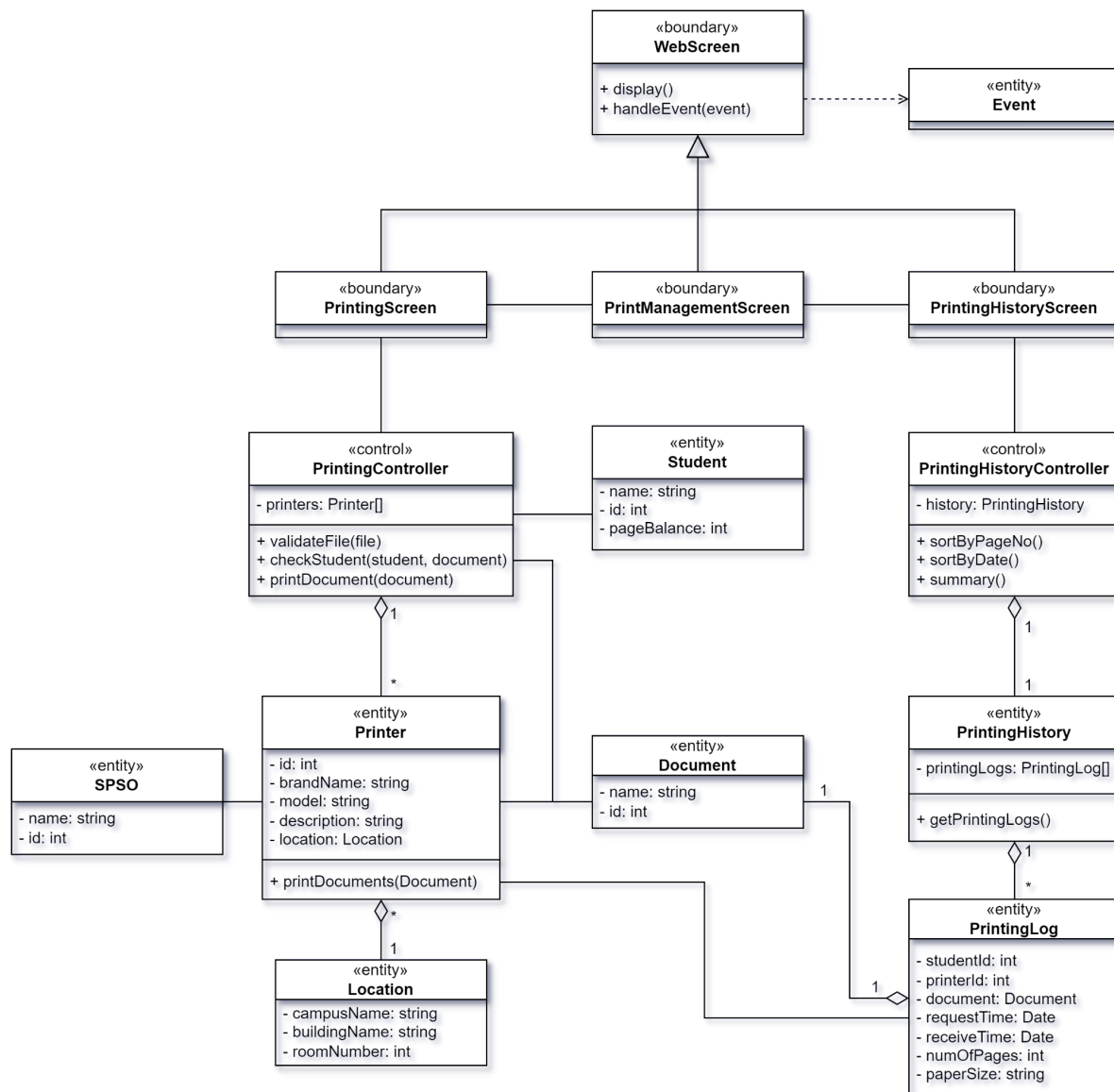### 3.2.1. Manage print operations

3.2.2. Print documents
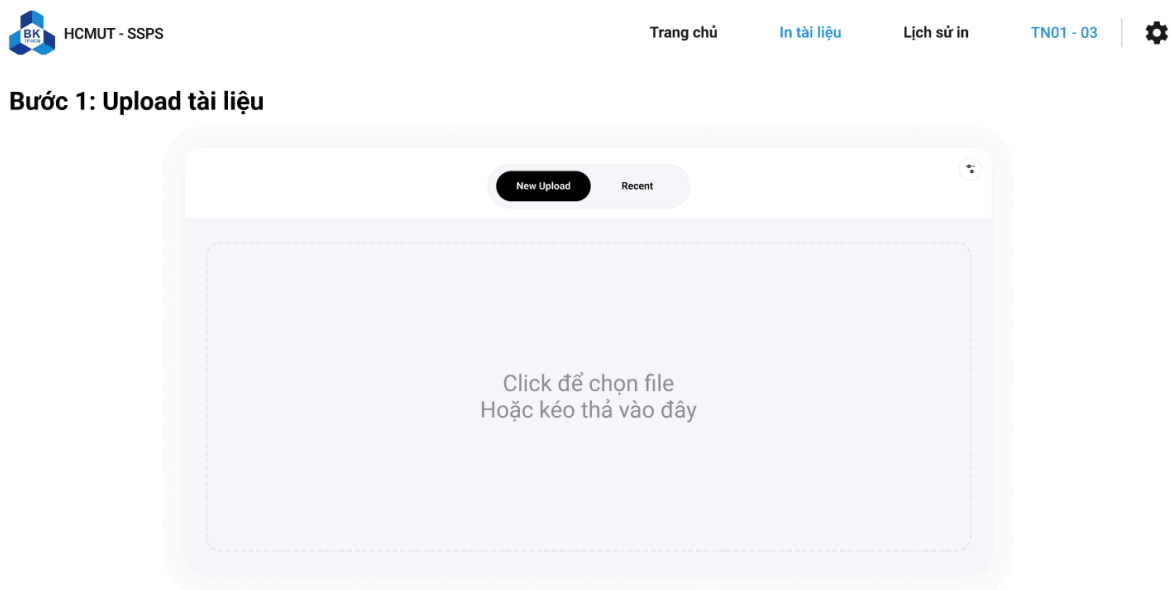
3.2.3. View printing history

3.3. Class diagrams

3.4. First minimum viable product (MVP 1)

The design of the project: Figma Link

Screen 1: Landing Page

Screen 2.1: Upload documents

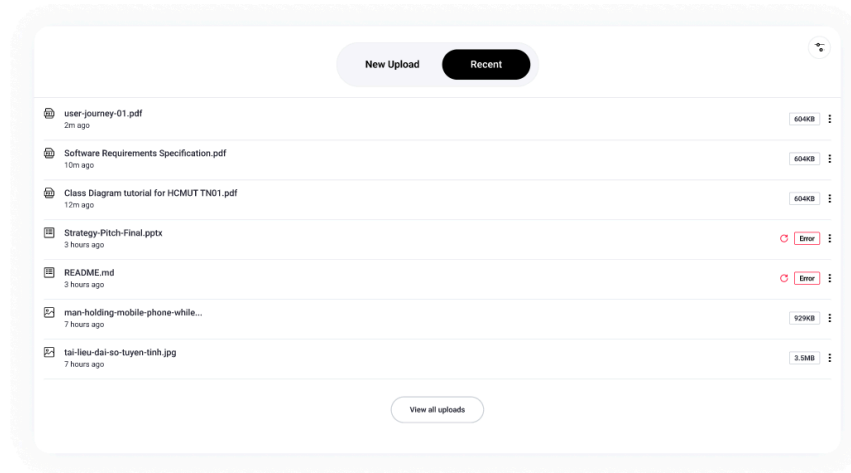*Screen 2.2: Recent files*

Screen 3: Format configurations and choosing printer's location
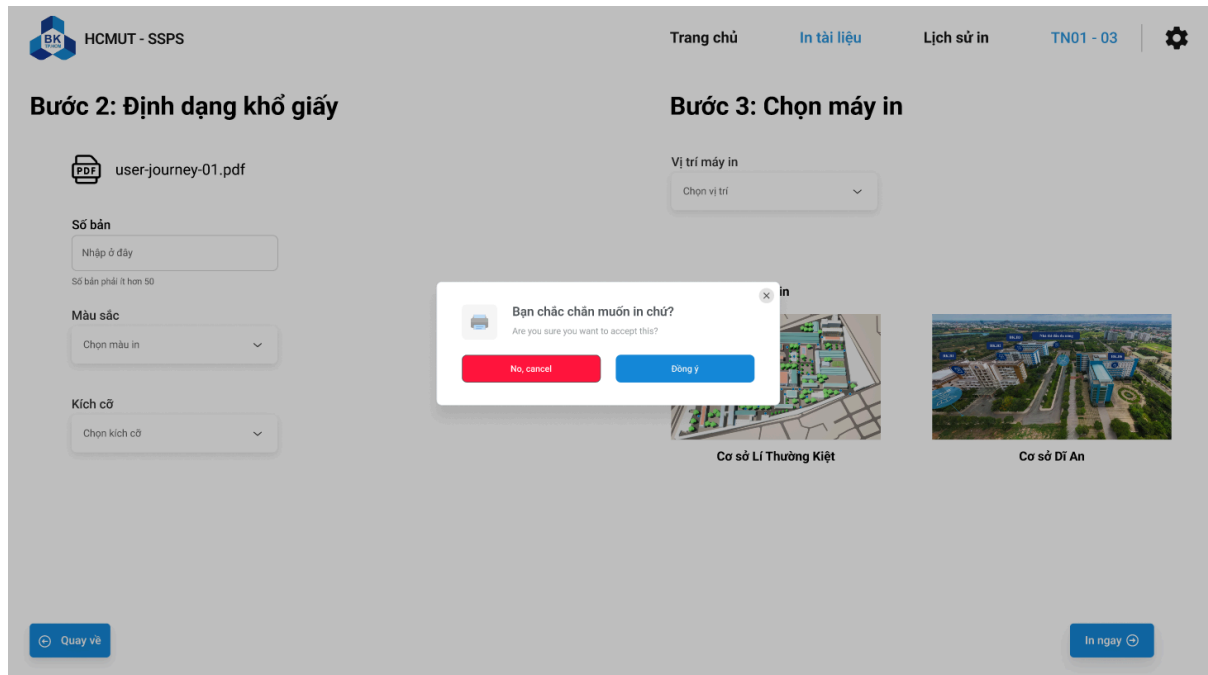
Screen 3.1: Confirmation of printing request
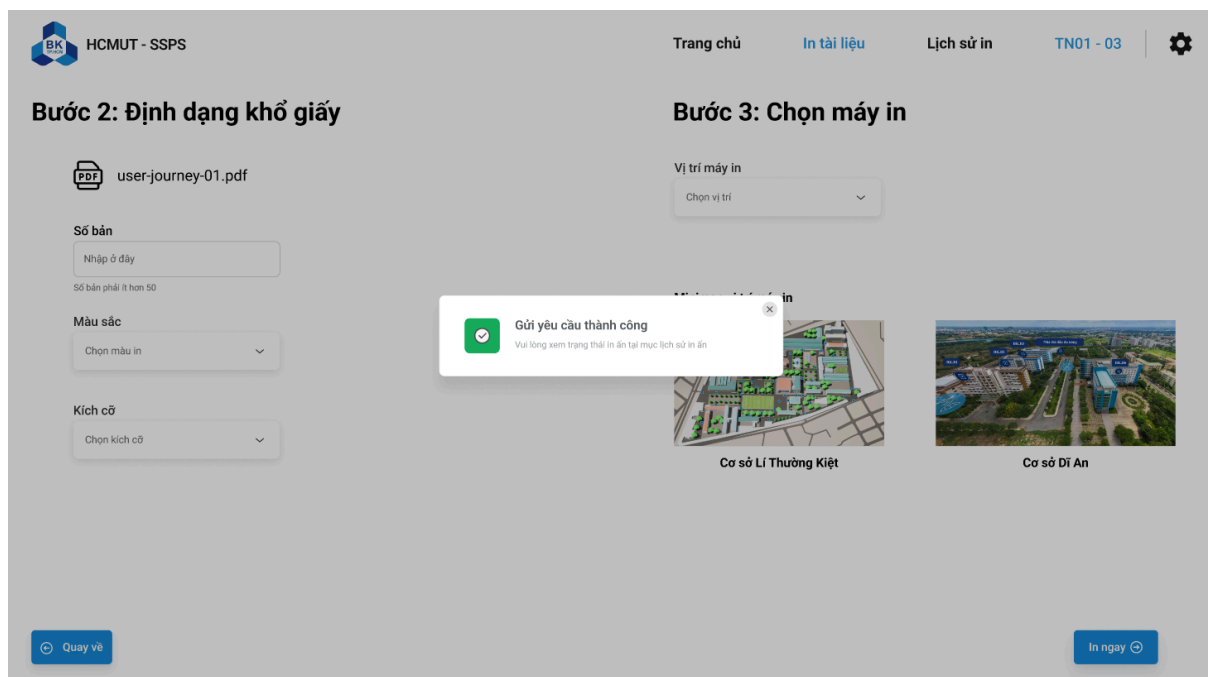
Screen 3.2: Successfully sending request

HCMUT - SSPS

Trang chủ   In tài liệu   Lịch sử in   TN01 - 03   ⚙

Lịch sử trang in
Thống kê

Tìm kiếm

Lần in	Máy in	Thời gian bắt đầu ↑↓	Thời gian kết thúc ↑↓	Tên file ⌄	Kích thước trang	Số trang ↑↓	Trạng thái ⌄
1	114H6	27/10/2023 7:00 AM	27/10/2023 7:00 AM	cnpm.pdf	A4	30	HOÀN THÀNH
2	504H3	23/10/2023 1:00 PM	23/10/2023 1:00 PM	btl.docx	A5	15	HOÀN THÀNH
3	101A3	06/10/2023 7:00 AM	06/10/2023 7:00 AM	ssps.png	A4	1	HOÀN THÀNH
4	315B3	02/10/2023 12:00 PM	02/10/2023 12:00 PM	img.jpg	A4	2	HOÀN THÀNH
5	315B3	02/11/2024 12:00 PM	.	img.jpg	A4	2	ĐANG IN
6	315B3	02/11/2024 12:00 PM	.	img.jpg	A4	2	KHÔNG HOÀN THÀNH

← Previous **1** 2 3 ... 67 68 Next →

Screen 4: Printing history

HCMUT - SSPS

Trang chủ   In tài liệu   Lịch sử in   TN01 - 03   ⚙

Lịch sử trang in
Thống kê

Tìm kiếm

Lần in	Máy in	Thời gian bắt đầu ↑↓	Thời gian kết thúc ↑↓	Tên file ⌄	Kích thước trang	Số trang ↑↓	Trạng thái ⌄
1	315B3	02/11/2024 12:00 PM	.	img.jpg	A4	2	ĐANG IN
2	315B3	02/11/2024 12:00 PM	.	img.jpg	A4	2	KHÔNG HOÀN THÀNH
3	114H6	27/10/2023 7:00 AM	27/10/2023 7:00 AM	cnpm.pdf	A4	30	HOÀN THÀNH
4	504H3	23/10/2023 1:00 PM	23/10/2023 1:00 PM	btl.docx	A5	15	HOÀN THÀNH
5	101A3	06/10/2023 7:00 AM	06/10/2023 7:00 AM	ssps.png	A4	1	HOÀN THÀNH
6	315B3	02/10/2023 12:00 PM	02/10/2023 12:00 PM	img.jpg	A4	2	HOÀN THÀNH

← Previous **1** 2 3 ... 67 68 Next →

Screen 4.1: Sort descending by start time
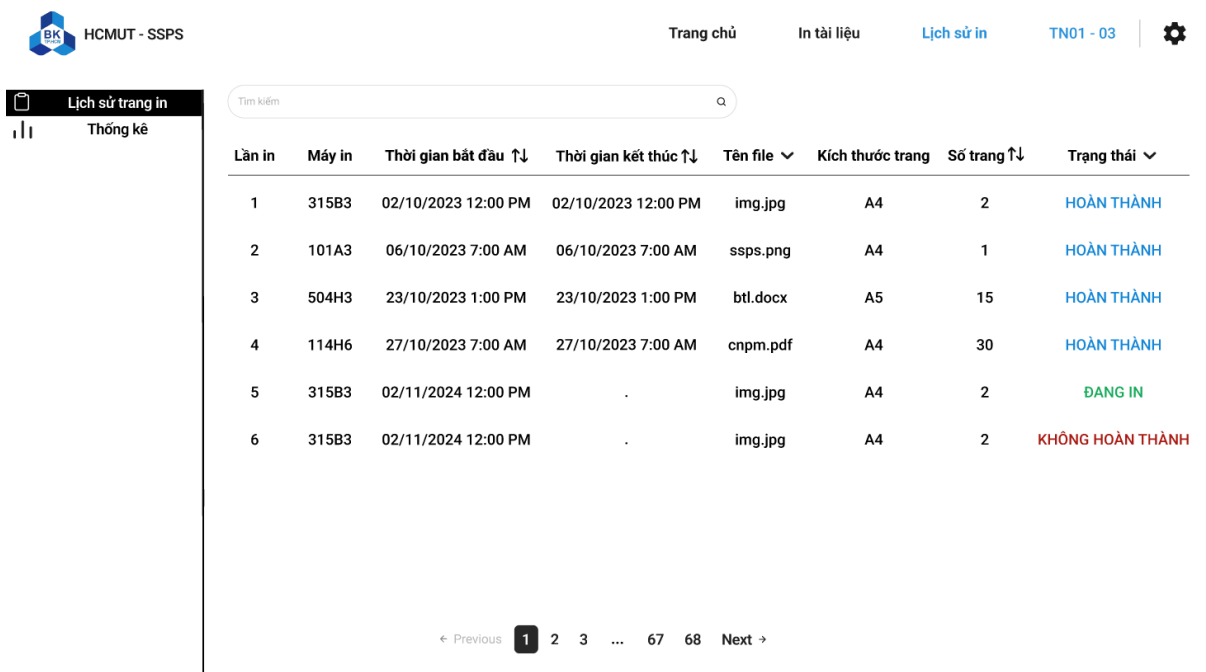
Screen 4.2: Sort ascending by start time

Screen 5: Statistic Summary
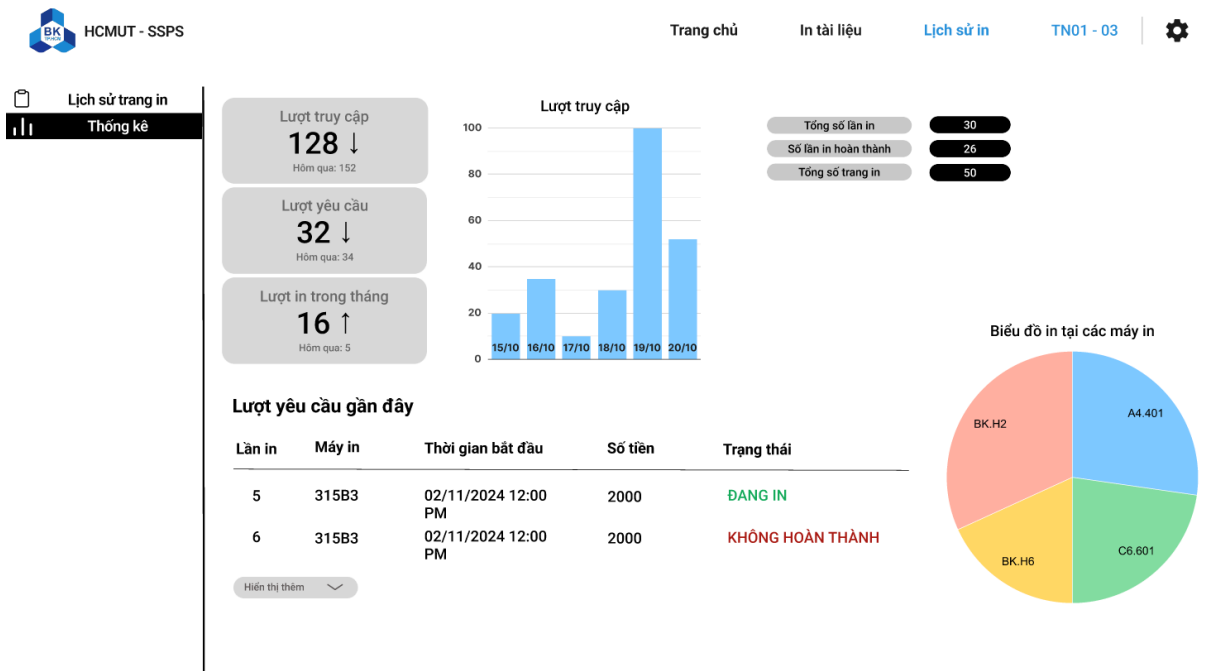
4. Architecture design

4.1. Layered architecture
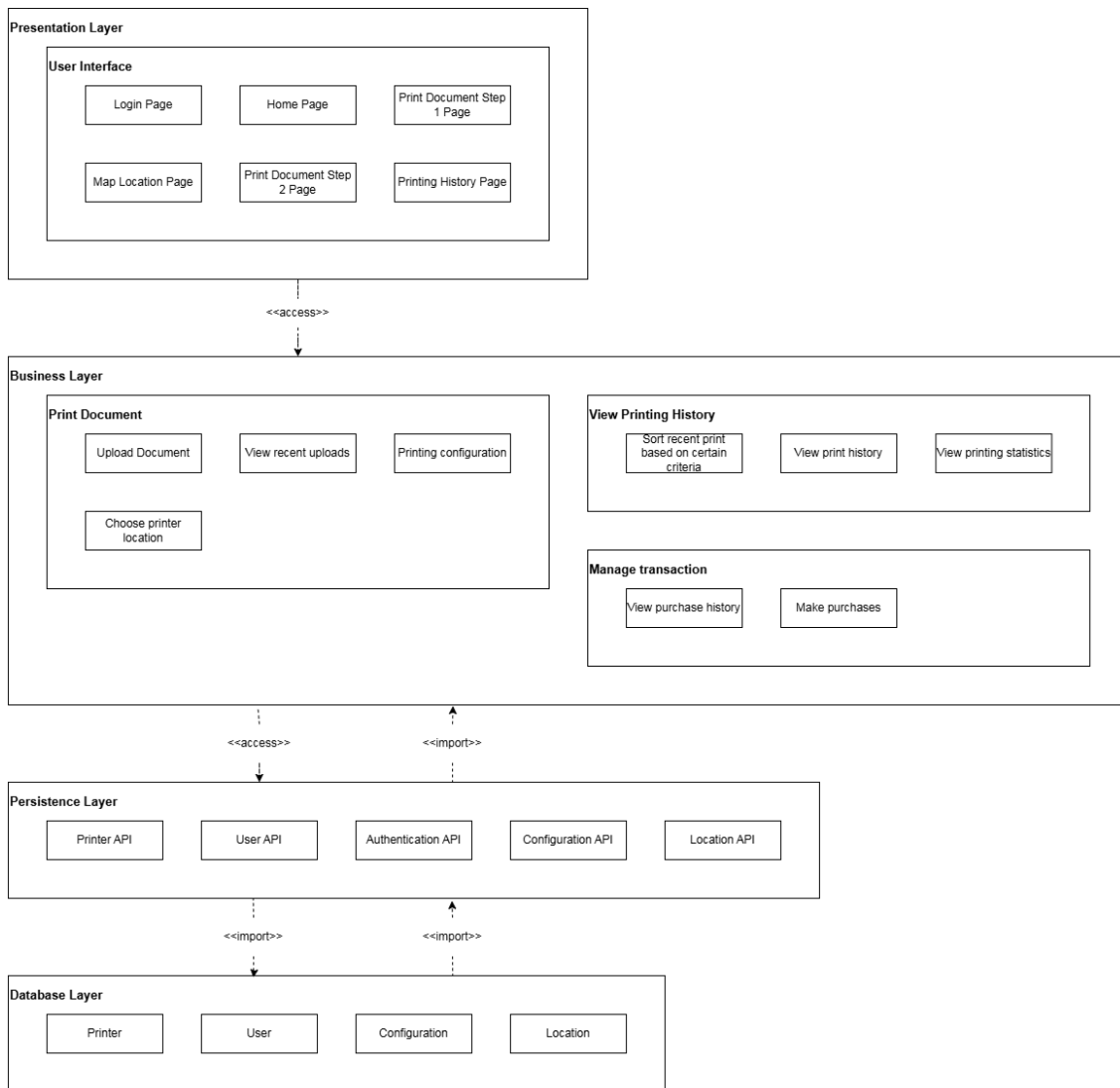
Presentation Strategy

The User Interface (UI) will be provided through a web application. React, which is a responsive design framework, will be used to implement the web app.

Data storage approach

The HCMUT-SSPS system will use Microsoft SQL Server as the database to manage user's accounts, printer configurations, and printing logs.

API management approach

The HCMUT-SSPS system will integrate several APIs to support core functionalities, including User, Printer, Authentication, Location, and Configuration. The Authentication API will be integrated with HCMUT-SSO to enable users to log in. The User API is used to get a user's balance. The Location API gets the printers' locations. The Configuration API will get the printer's configuration.

4.2. Component diagram
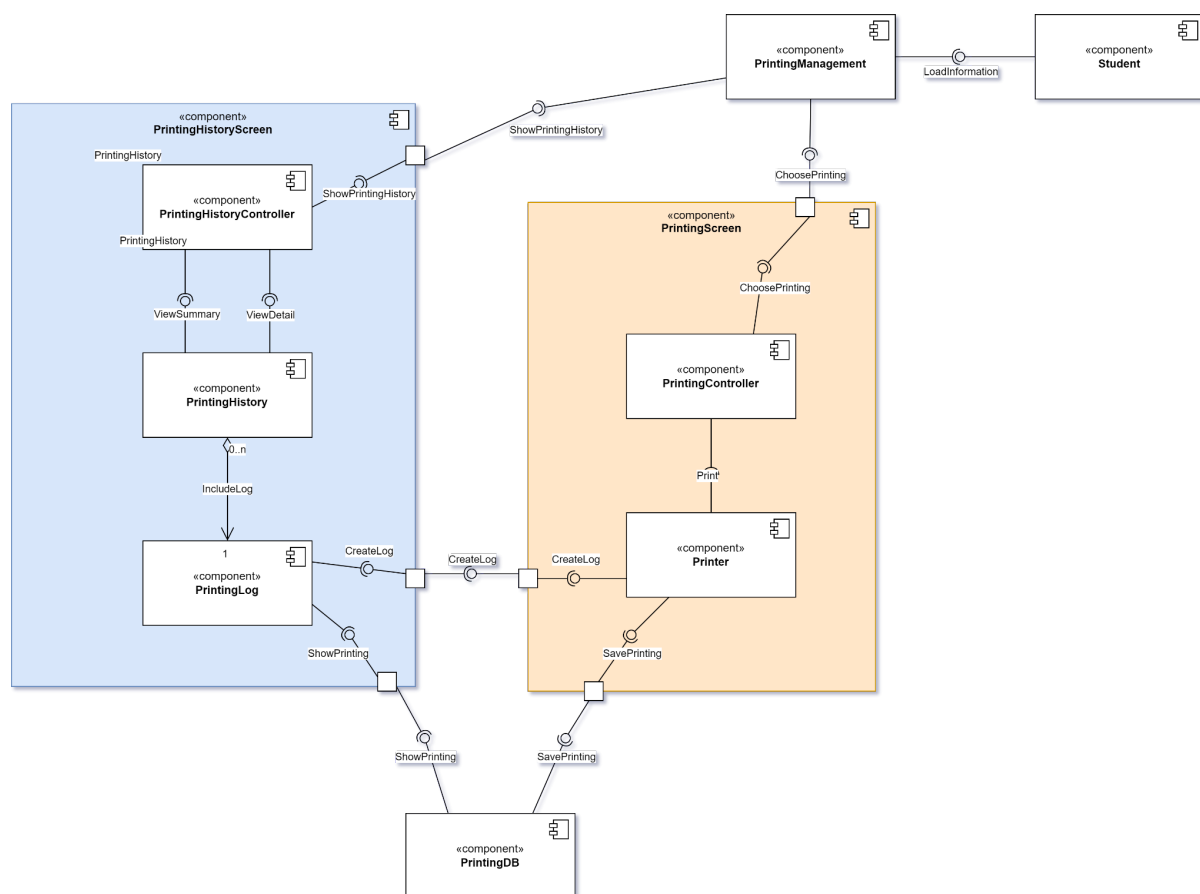
This component diagram shows the architecture of a printing management system, divided into modules for handling printing operations and viewing print history.

- **PrintingManagement** is the main module that links students to the printing system, allowing them to initiate printing tasks or view their printing history.
- **Student** interacts with **PrintingManagement** by loading information and choosing print options.

On the **Printing Operations** side:

- **PrintingScreen** and **PrintingController** allow students to select print jobs and control printing.
- **Printer** executes the printing process.
- After each print job, a log entry is created and sent to **PrintingLog** and stored in **PrintingDB**.

On the **Printing History** side:

- **PrintingHistoryScreen** and **PrintingHistoryController** let students view summaries or details of their print history.
- **PrintingHistory** maintains a collection of logs (0 to many) and works with **PrintingLog** to display past printing activities.

The **PrintingDB** component serves as the central database, storing both new print logs and history data, and supporting the retrieval of records for viewing.

This modular design separates the system into clear components for printing tasks, history tracking, and data storage, making it easier to maintain and extend.

5. Implementation - Sprint 1

5.1. Version control system

In this project, GitHub was employed as the primary Version Control System (VCS) to effectively manage code changes, facilitate team collaboration, and maintain a clear development history. Two repositories were created:
- Front-end repository: [Front-end repo](#)
- Back-end repository: [Back-end repo](#)

5.2. Documentation

All the documents are stored in the *Documents* folder of the [front-end repository](#). Documentation includes:
- The software requirements specification
- Diagrams
  - Use case diagrams
  - Activity diagrams
  - Sequence diagrams
  - Class diagram
  - Layer architecture design
  - Component diagram

5.3. Usability testing

General

Every member has conducted a usability test, using Figma prototype at the following link
Link Figma. The testing session took place directly with the participation of all members, including the host (team leader), the secretary and the tester. During the testing session, the team recorded the solutions, task completion rate, comments, overall evaluation, questions and feedback from all members.

Participants and Roles

We invited 5 people as testers to perform the usability testing and document their feedback. Here is the information of the participants.

Name	Roles
Lê Hồng Phúc	Host/Leaders/Member
Lê Phúc Hoàng	Secretary/Member
Võ Thành Khoa	Tester
Trần Anh Đức	Tester
Lê Phúc Khánh Linh	Tester
Lê Thanh Hồng Khánh	Tester
Trần Anh Khôi	Tester

Testing method

The testing session was divided into several sessions, each lasting about 30 minutes and focusing on a part of the system. During a session, the team leader explained the testing process and asked the participants to fill in a form with questions related to the tasks corresponding to the scenarios. Each participant would try to read the scenarios of these tasks and try to manipulate the application to achieve that scenario.

After each task, the team leader asked the participants to rate the user interface on a 5-point scale, ranging from "Strongly Disagree" to "Strongly Agree". The evaluation factors in these post-task scenarios include:
- Ease of finding information on the application
- Flexibility of operation
- Accuracy in predicting which part of the application will contain a specific piece of information
- After the final task was completed, the team leader asked the members to evaluate the system overall

- on the same 5-point scale as above, with the following factors:
- Ease of use
- Frequency of use
- Ability to learn how to use the application
- Supportability - the ability of users to find information easily
- Attractive interface - the ability of the website interface to make users want to explore further
- Content - the ability of the website content to attract users
- Website layout

In addition, the team leader also asked the members to answer the questions:

- Which components of the application did they like the most
- Which components of the application did they like the least
- Suggestions for improving the application

Testing strategy

Test participants will attempt to perform the following tasks:

1. Successfully register for printing service
2. Fail to register for printing service due to lack of printed pages and purchase new printed pages, then return to register for printing service again successfully
3. View student printing history at printer 114H6
4. View statistics of student's website visits this month

Result

- Task completion rate

Participant	Khoa	Đức	Linh	Khánh	Khôi	Completion rate
Task 1	x	x	x	x	x	100%
Task 2	x	x	x	x	x	100%
Task 3	x	x	x	x	x	100%
Task 4	x	x	x	x	x	100%

Task evaluation

- Ratings are on a scale of 0-5. The following scores are the average scores of the test participants.

Task	Easy to find information	Easy to navigate	Easy to operate	Average rating
1	4	5	5	4.67

2	3	4	3	3.33
3	4	5	4	4.33
4	5	5	4	4.67

Performing time

- Time is measured in seconds

Task	Khoa	Đức	Linh	Khánh	Khôi	Trung bình
1	21	15	12	14	17	15.8
2	20	50	28	23	31	30.4
3	12	18	21	16	19	17.2
4	23	26	17	29	15	22

Total evaluation

- Here is the average vote result of every tester.

	Average score
Website is easy to use	4.8
Will use the site more regularly	4.0
Easy to navigate between elements of the site	4.2
Easy to learn how to use the website	4.4
Can find information quickly	3.6
Website has a good layout	3.4
The content of the site makes me want to come back	3.2

Suggestion for system improvement

- Add statistical features for each printer: Need to add statistics features for each printer to have more reporting information to improve statistics.

6. Implementation - Sprint 2

6.1. Develop MVP 2

MVP 2 is developed with web applications.

Technology is used in the process of implementation:
- Front-end:
  - Framework: ReactJS
  - Main programming language: TypeScript
  - UI tools: TailwindCSS and Flowbite / Flowbite React
  - Project builder: Vite
  - Error detector: ESLint
  - Formatter: Prettier
- Back-end
  - Framework: ExpressJS
  - Language: TypeScript
  - Authentication framework: PassportJS
  - ORM: Prisma
  - Error detector: ESLint
  - Formatter: Prettier
- DBMS: Microsoft SQL Server

Deployment:
- Front-end: https://hcmutssps.vercel.app/
- Back-end: https://hcmutssps.azurewebsites.net/

6.2. Project report presentation

The presentation slides of the project report:
https://docs.google.com/presentation/d/1WdDBWO45q7YHnNYYen7ih1TBnlWVXCPmrzdmSfKVEtU/edit?usp=sharing