Software Requirements Specification

for

A smart printing service for students at HCMUT

Version 1.0.0 approved

Prepared by:

1. **Bùi Phạm Thái An – 2052358**
2. **Cao Quế Phương – 2252652**
3. **Đặng Duy Tiến – 2252808**
4. **Đào Thị Hà An – 2252002**
5. **Đinh Bá Khánh – 2252323**

Department of Software Engineering

Faculty of Computer Science and Engineering

Ho Chi Minh City University of Technology – VNU-HCM

22/09/2024

Contents

Revision History

Name	Date	Reason For Changes	Version
Project 1	22/09/2024	Requirement elicitation	1.0.0
Project 1	01/10/2024	Use case diagram	1.1.0
Project 2	27/10/2024	System modeling	2.0.0
Project 3	10/11/2024	Architecture design	3.0.0
Project 4	22/11/2024	Online repository	4.0.0
Project 4	24/11/2024	Usability test	4.1.0

1. Task 1: Requirement elicitation (1.1, 1.2)

1.1 Domain Context

The Student Smart Printing Service (HCMUT_SSPS) is a web-based and mobile-based printing service that has an objective to provide students with a convenient way to print their documents with printers located around the campuses. Each printer has its own ID, brand name, printer model, short description and the location. The system allows students to print their documents by uploading their files onto the system, they can choose the printer and edit the printing properties, the permitted file types are also limited and configured by SPSO. The system also logs the printing actions for students including student ID, printer ID, file name, printing start and end time, number of pages for each paper size. Besides, SPSO can view the printing history of all students and each student can view their history. For each semester, each student will be given the default number of A4-size pages and they can buy additional printing pages through some online payment systems, each student can only print when the number of printing pages does not exceed their balance and each A3 page is equal to 2 A4 pages. Moreover, SPSO can also manage the printer states such as add, enable and disable and SPSO can also change the default number of pages, the dates the system will give the default number of pages to all students as well as change the permitted file types. SPSO can view the report monthly and annually which is stored in the system. Finally, to use the system, all users have to be authenticated by HCMUT_SSO service.

1.2 Stakeholders and Needs

Students:

- Print a document by uploading a document file onto the system
- View printing log for a time period together with a summary of number of printed pages for each page size
- Buy some more using the feature buy printing page

Student printing service officer:

- Limit and configure the permitted file types
- View the printing history of all students or a student for a time period (date to date) and for all or some printers
- Manage the printers such as add/ enable/ disable a printer
- Manage other configurations of the system such as changing the default number of pages, the dates that the system will give the default number of pages to all students, the permitted file types accepted by the system.
- View the reports of the using of the printing system

1.3 Benefits of the System

For Students:

1. Convenient Document Printing:

- Students can upload documents, choose printing properties, and print at any available campus printer from anywhere using a web or mobile app.

- Flexible options for specifying paper size, number of copies, double-sided printing, etc., offer personalized printing.

2. Cost Management:

- The system tracks each student's print quota (pages) per semester, ensuring they don't exceed their allowance.
- Students can purchase additional print pages as needed via an integrated online payment system like BKPay, making the process seamless.

3. Transparency and Tracking:

- Students can access their printing history, including the number of pages printed per size and logs of all previous printing actions.
- This helps students track their usage and ensure they stay within their budgeted print quota.

4. Multiple Access Points:

- The system is accessible via both web and mobile apps, providing students flexibility in managing their printing needs regardless of location.

5. Security and Authentication:

- Integration with the university's Single Sign-On (HCMUT_SSO) ensures that only authorized users access the system, protecting student information and transactions.

For University Administration (SPSO):

1. Centralized Printer Management:

- The SPSO can add, enable, disable, or configure printers across campuses, making printer management streamlined and ensuring better resource allocation.

2. Configurable System Settings:

- The system allows customization of key parameters such as the default print allowance for students, acceptable file formats, and important operational dates, making the system adaptable to evolving needs.

3. Comprehensive Logging and Reporting:

- All student printing actions are logged, allowing the university to monitor usage and create accurate reports on the print system's performance, enabling better resource management.
- Monthly and yearly reports are generated automatically, providing insights into system usage without manual effort.

4. Enhanced Accountability:

- The detailed logs ensure transparency, allowing the SPSO to monitor student printing history and troubleshoot any discrepancies.
- This also helps in preventing misuse or overuse of resources.

5. Efficient Payment Integration:

- The integration with an online payment system (BKPay) simplifies the process of buying additional print pages, reducing administrative overhead for managing student printing balances.

1.4 Functional Requirements

Students:

1. Students can print documents by uploading files, choosing printers and specifying printing properties.
2. Students can view their own printing log
3. Students can view their number of printed pages for each page size
4. Students can buy more pages to print
5. Students can view how many pages they have left in their balance

Student Printing Service Officer (SPSO):

1. SPSO can view printing history of student(s) or printer(s)
2. SPSO can add, remove, enable, disable a printer
3. SPSO can change the number of default pages given to student each period
4. SPSO can change the date students are givens pages
5. SPSO can change the permitted file types
6. SPSO can view the report of the printing system

1.5 Non- Functional Requirements

The printing system is implemented as a web-based app and a mobile app.
The system is available 24/7, except for maintenance periods.
The system is available for multiple concurrent users but must ensure that each printer serves only 1 user at a time.
The HCMUT_SSO authentication service must authenticate all users before they use the system.
The payment method is made online, like BKPay

2. Use case Diagrams (1.3)

2.1 Use case Diagram for the Whole System

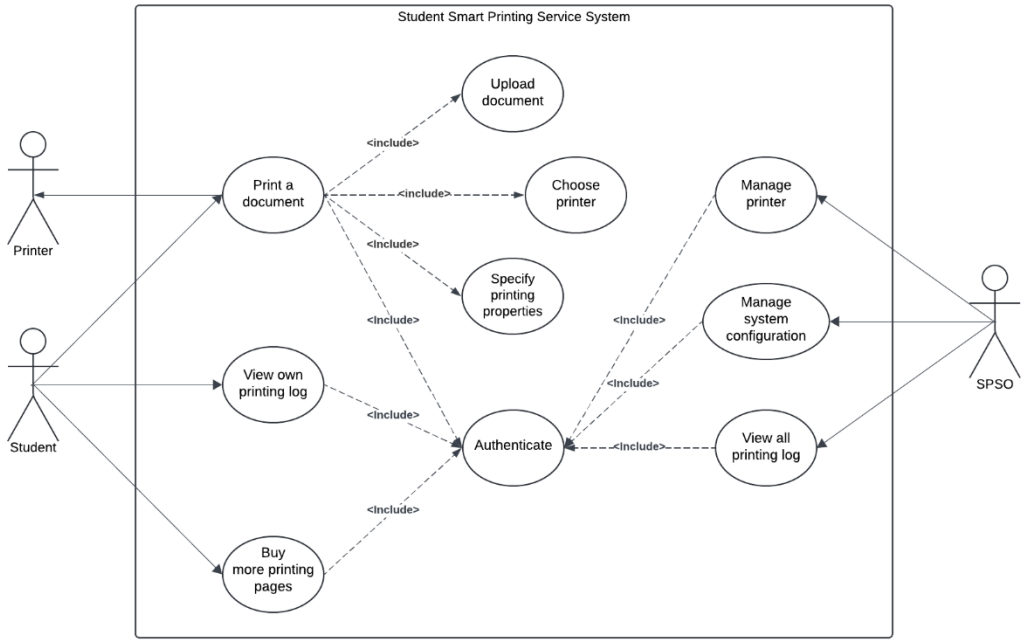

Figure 1: Use case diagram of the whole system

2.2 Use case Diagram for Printing Process Module

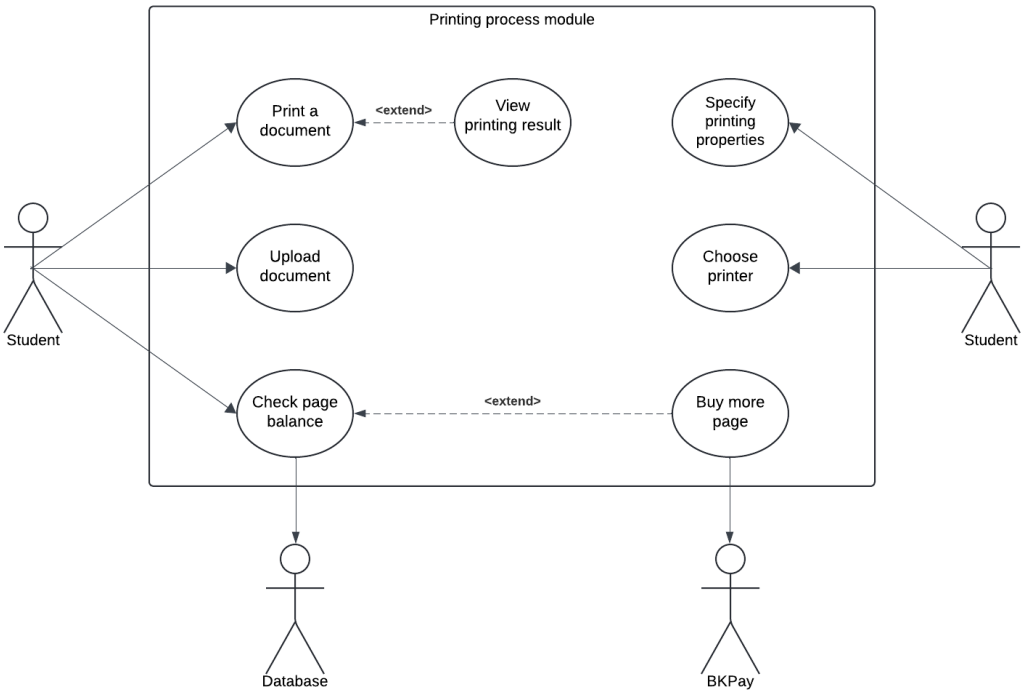

Figure 2: Use case diagram for printing process module

2.3 The Details of Use cases in Printing Process Module

1. Use case Print a document

Use case ID	UC-PPM-01
Use case name	Print a document
Created by	Đinh Bá Khánh
Created date	01/10/2024
Actors	Student, Database, BKPay
Description	The student requests the system to print his/her documents by choosing the printer based on its ID and then specify the printing properties such as paper size, number of pages, one-/double-size, number of copies, etc. The system will check if the student has enough available number of pages and the permitted file types and then it will print the documents for the student.
Trigger	The student wants to print the documents.
Preconditions	System checked page balance. Student must log into the system based on his/her ID. Student chose the printer. Student specified printing properties. Student uploaded documents onto the system
Postconditions	Printing histories are stored in the system. Documents are printed successfully.
Normal flow	1. The student logs into the system based on his or her ID. 2. The student uploads his or her documents onto the system. 3. The system checks if the file is uploaded successfully. 4. The student specifies printing properties. 5. The system checks the balance 6. Student choose the printer. 7. The system sends the document file to the selected printer. 8. The document is printed 9. The student views the result.
Alternative flow	Alternative 1: 5a. Student does not have sufficient page balance 5b. Student buys more pages
Exception	Exception 1: 5a. Student does not have sufficient page balance 5b. Student cancel the printing process => Use case stops

2. Use case Upload document

Use case ID	UC-PPM-02
Use case Name	Upload document
Created by	Đặng Duy Tiến
Created date	01/10/2024
Actors	Students
Description	The system allows students to upload documents for printing
Trigger	Students want to upload documents
Preconditions	Students logged in the system
Postconditions	The uploading actions is saved in the system Documents is uploaded successfully
Normal flow	1. Students log into the system 2. Students choose permitted file types 3. Students upload the documents 4. Students check the uploading action
Alternative flow	None
Exception	Exception 1: 2a. Students cancel uploading => Use case stops Exception 2: 3a. File types are not permitted => Use case stops

3. Use case Specify printing property

Use case ID	UC-PPM-03
Use case name	Specify printing properties
Created by	Đào Thị Hà An
Created date	01/10/2024
Actors	Student
Description	The student configures the printing properties of the selected printer according to his/her preferences in order to print the uploaded document.

Trigger	Printer is ready to print and student wants to set the printing mode.
Preconditions	Student has logged in Student uploads the printing file of permitted type Student has chosen an available printer.
Postconditions	All (changed) properties are successfully applied to the printer.
Normal flow	1. Student selects the default settings. 2. Student submits the settings. 3. System invokes the "Check page balance" use-case (UC-PPM-05). 4. System shows the notification that all settings are valid.
Alternative flow	Alternative 1: 1a. Student changes one or more printing properties. => Continue at step 2 Alternative 2: 3a. Student buys extra printing pages => Continue at step 2 Alternative 3: 2a. Student cancels the printing process. => Use case stops
Exception	Exception 1: 3a. Student doesn't have sufficient page balance 3b. Student changes some setting => Continue at step 2

4. Use case Choose printer

Use case ID	UC-PPM-04
Use case name	Choose printer
Created by	Bùi Phạm Thái An
Created date	01/10/2024
Actors	Student
Description	Student can choose which printer to use to print their document
Trigger	Student wants to print a document
Preconditions	Student has logged in Student is on printing window
Postconditions	Printer is selected

Normal flow	1. Student opens printer list 2. Student browses for printer 3. Student selects a printer 4. Student confirm the selection
Alternative flow	None
Exception	Exception 1: 3a. Student cancel the selection => Use case stops

5. Use case Check page balance

Use case ID	UC-PPM-05
Use case name	Check page balance
Created by	Cao Quế Phương
Created date	01/10/2024
Actors	Student, SPSS
Description	The student checks their remaining page balance for printing. The balance is displayed in terms of A4 pages, with conversions for A3 pages (1 A3 = 2 A4 pages). Students can use this information to determine if they need to purchase more pages.
Trigger	The student wants to print the documents.
Preconditions	Student has logged in The student has been allocated a page balance for the current semester.
Postconditions	The system displays the current page balance to the student, broken down by A4 pages (and A3 equivalents if applicable).
Normal flow	1. The student logs into the system. 2. The student navigates to the "Check Page Balance" feature. 3. The system retrieves the student's current page balance from the database. 4. The system displays the remaining page balance to the student. 5. The student can proceed with printing or purchasing additional pages.
Alternative flow	Alternative 1: 5a. Student has a balance of 0 pages 5b. System displays a message indicating that the student has no remaining pages and must purchase more to continue printing.
Exception	Exception 1: 5a. System is unable to retrieve the balance

	5b. An error message is displayed => Use case stops Exception 2: 5a. Student does not have enough page balance 5b. The system displays an error => Use case stops

3. Task 2: System modelling

3.1 Activity diagrams

1. Print document

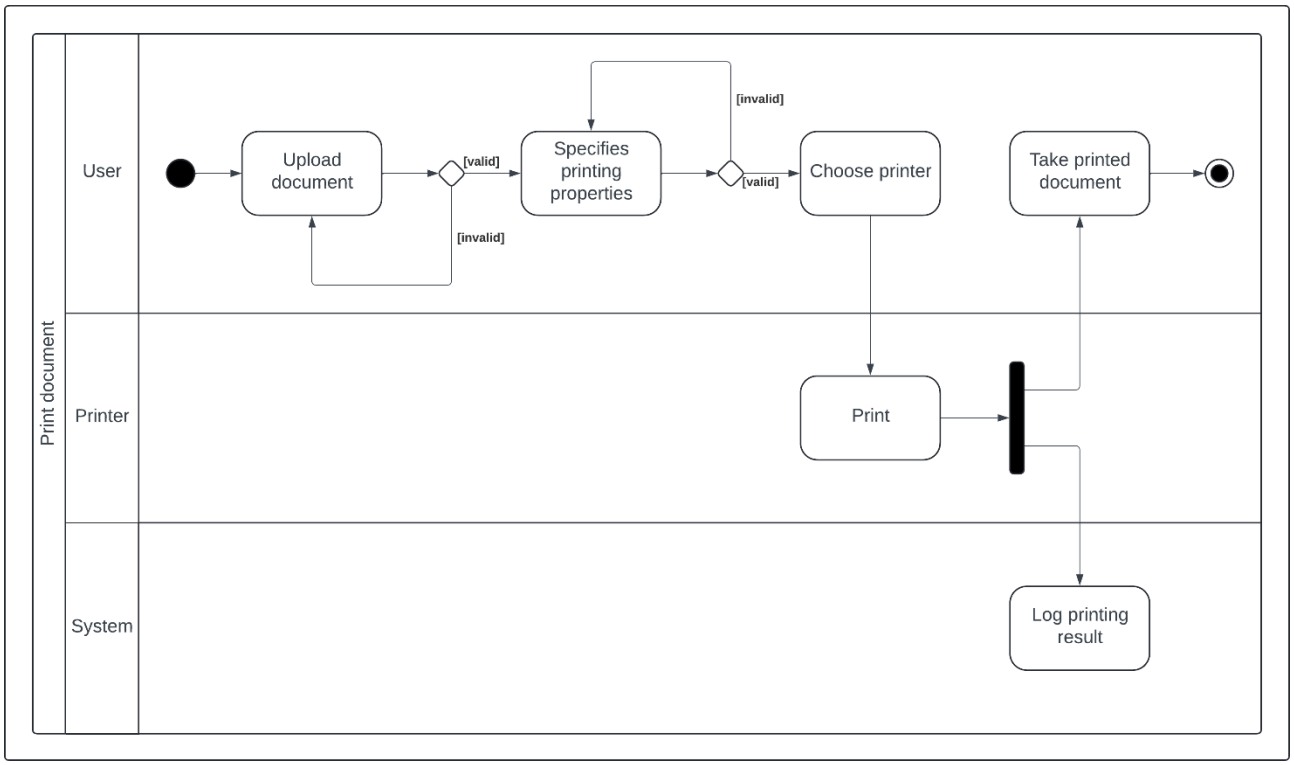

Figure 3: Activity diagram for print document

This diagram represents the steps which students need to follow to print a document. The process begins with a student uploading a document. If the document is valid, the student then needs to specify printing properties. If the properties are valid, the student then chooses a printer. While the printer is printing the document, the system will log this printing session. The process ends when the student has received the printed document.

2. Upload document

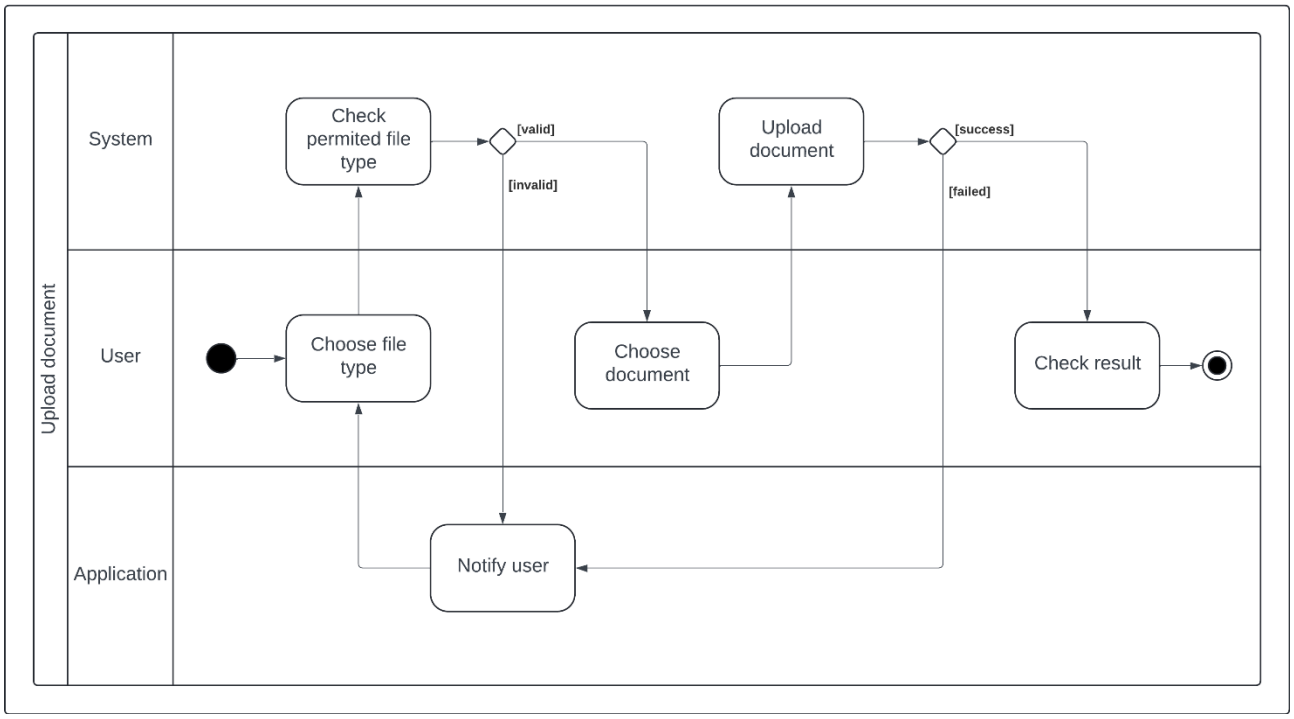

Figure 4: Activity diagram for upload document

This diagram shows the activities performed by the student and system while uploading documents. The process begins with a student choosing the file type. If the file type is allowed (configured by SPSO), the student can then choose and upload the document. The process ends after the student checks and confirms the document has been uploaded.

3. Specify printing properties

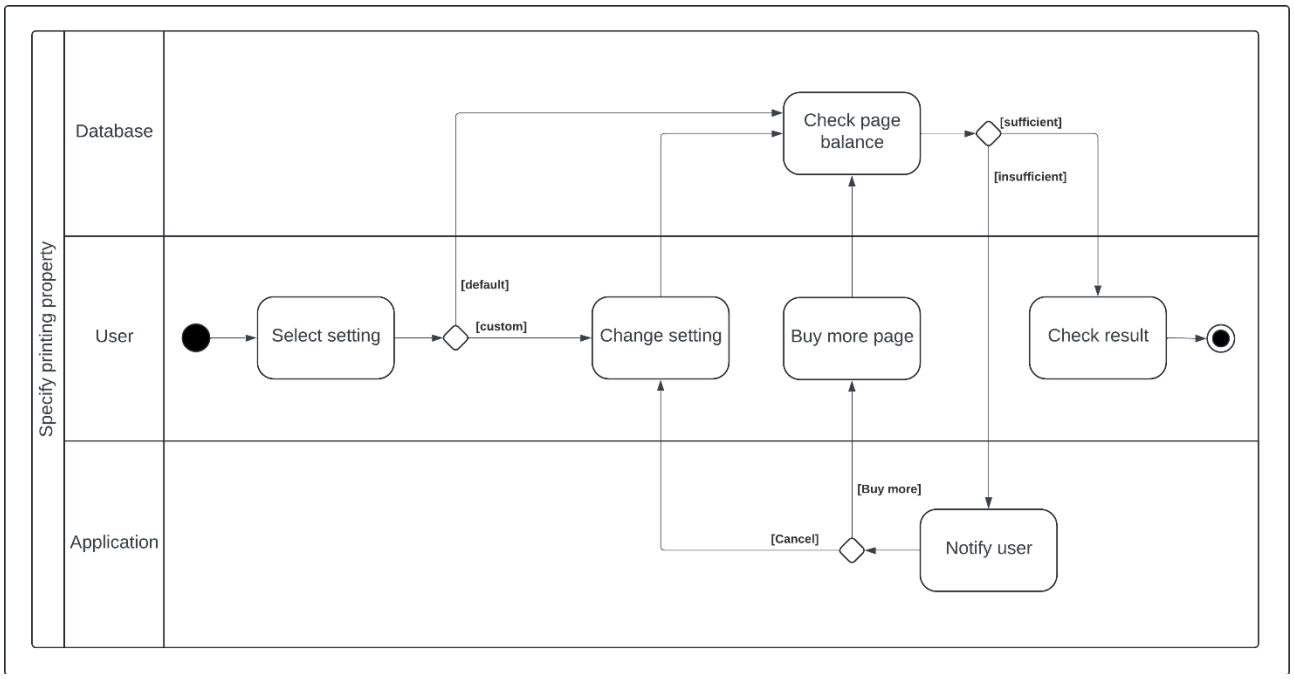

Figure 5: Activity diagram for specify printing properties

This diagram represents the specify printing properties process. The process begins when a student chooses which setting to apply, if they choose default setting, the system will invoke the check page balance process, otherwise they need to specify the reference (page size, page number, number of copies, etc.) before the system checks page balance. The process ends with the student confirming the correct properties.

4. Check page balance

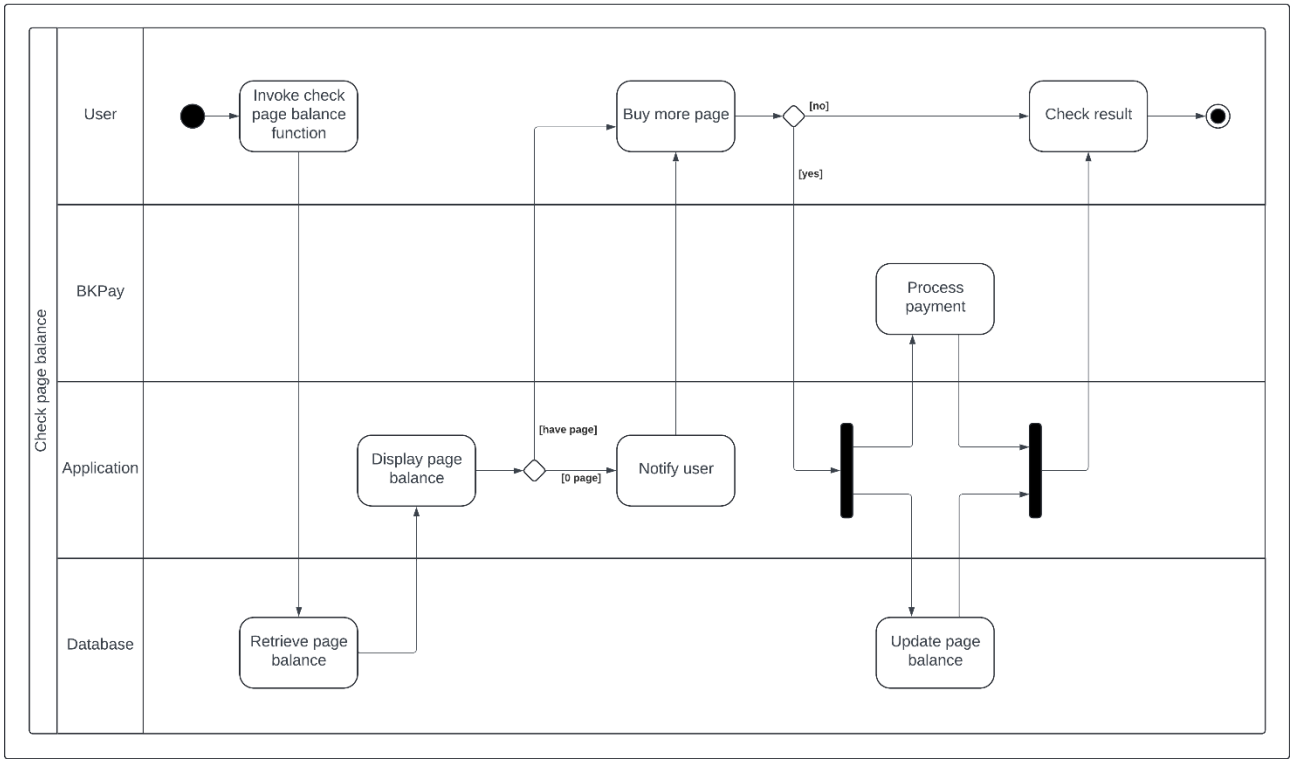

Figure 6: Activity diagram for check page balance

This diagram represents the process the system checks and the process to buy more pages from students. The activity starts when the check page balance function is invoked. The application then retrieves page balance of the students and displays it. If the page balance is 0 it will also notify the student. The student can then choose to buy more pages. If they do, the BKPay system will process the billing and the system will update the student's page balance. The process ends after the student has checked the result.

3.2 Sequence diagrams

1. Print document

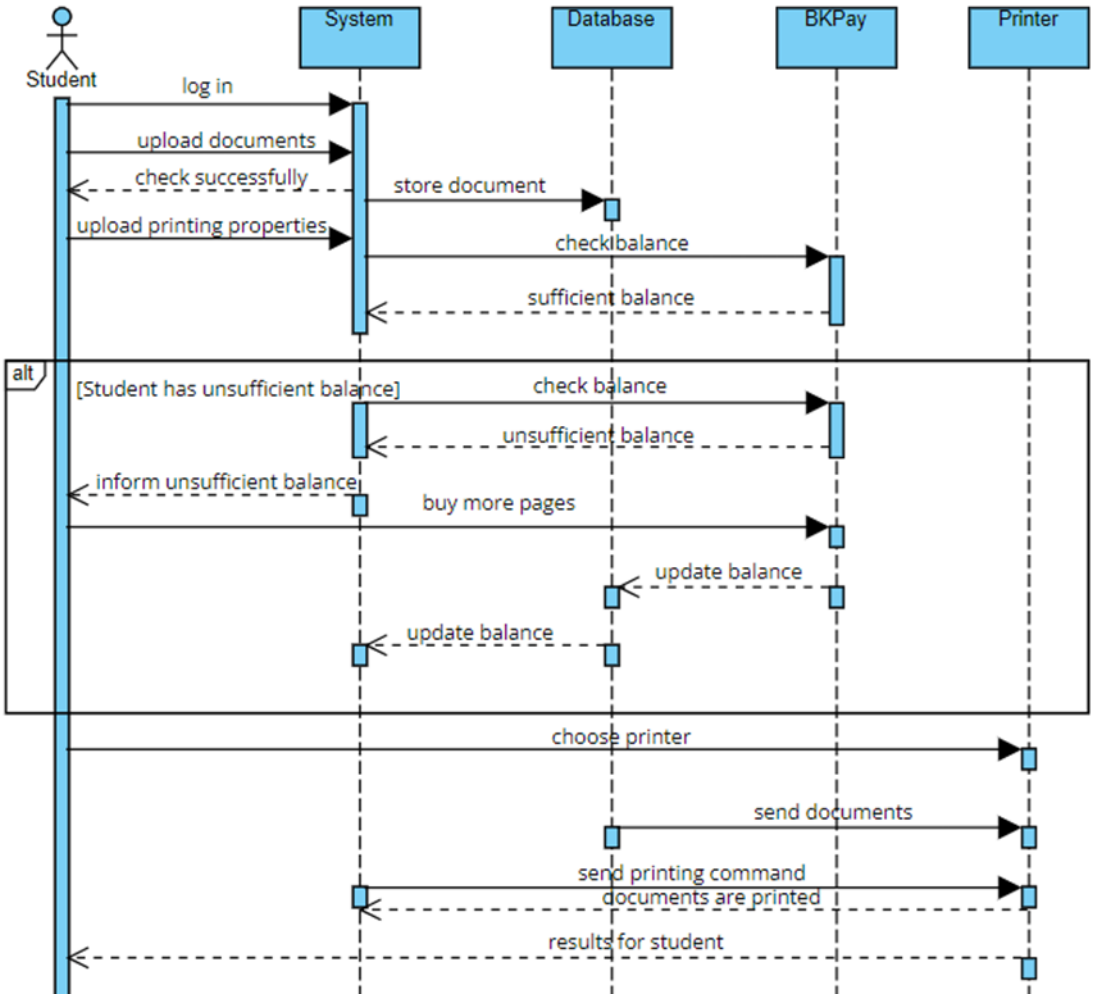

Figure 7: Sequence diagram for print document

This sequence diagram illustrates the use case of printing a document. The key participants are Student (User), System, Database, BKPay and Printer. The sequence starts when the student logs in to the system. Then the system will check the information, if correct then the student subsequently uploads printing properties and then the system will check the balance and if it is sufficient, it will print the document, completing the use case.

2. Upload document

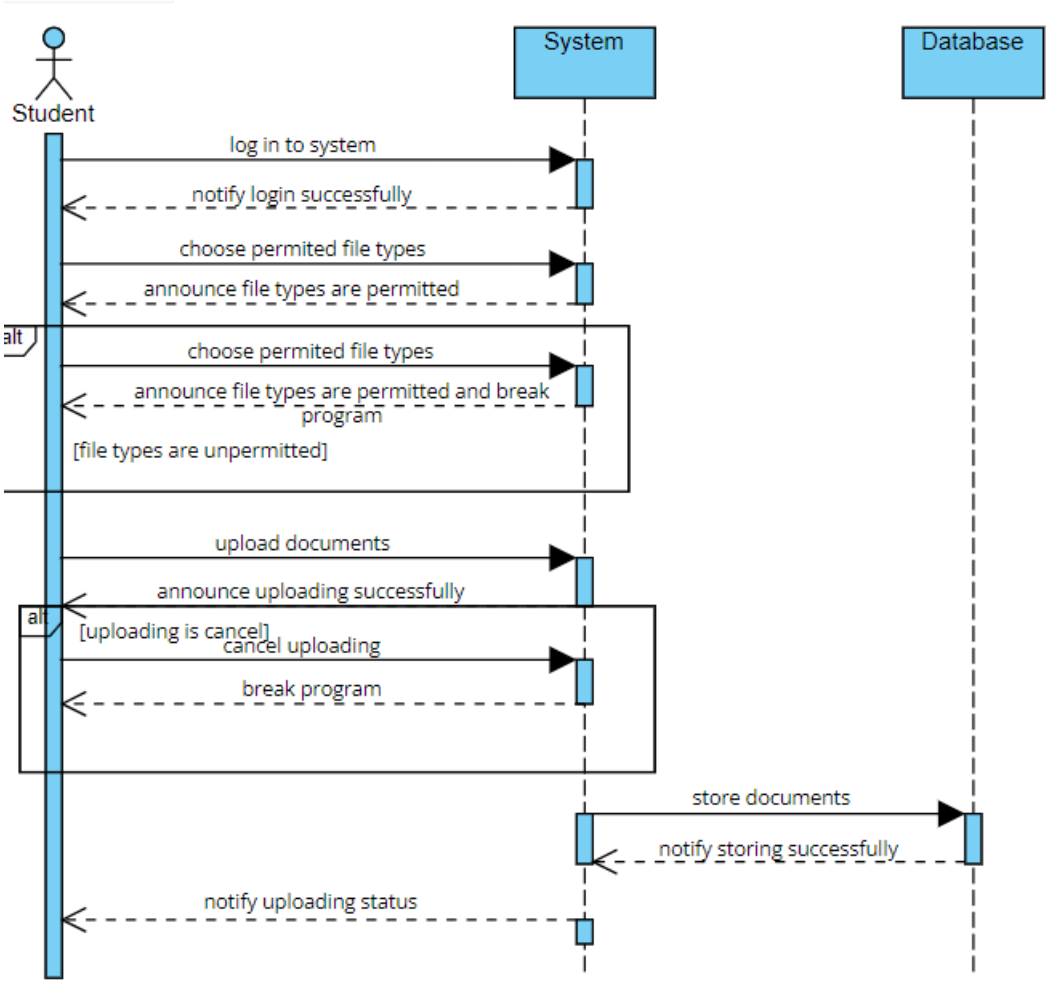

Figure 8: Sequence diagram for upload document

This sequence diagram illustrates the use case of uploading documents. The key participants are Student, System and Database. The sequence starts when the student logs in to the system. If log in successfully, then they choose the permitted file types and then upload the document, completing the use case.

3. Specify printing properties

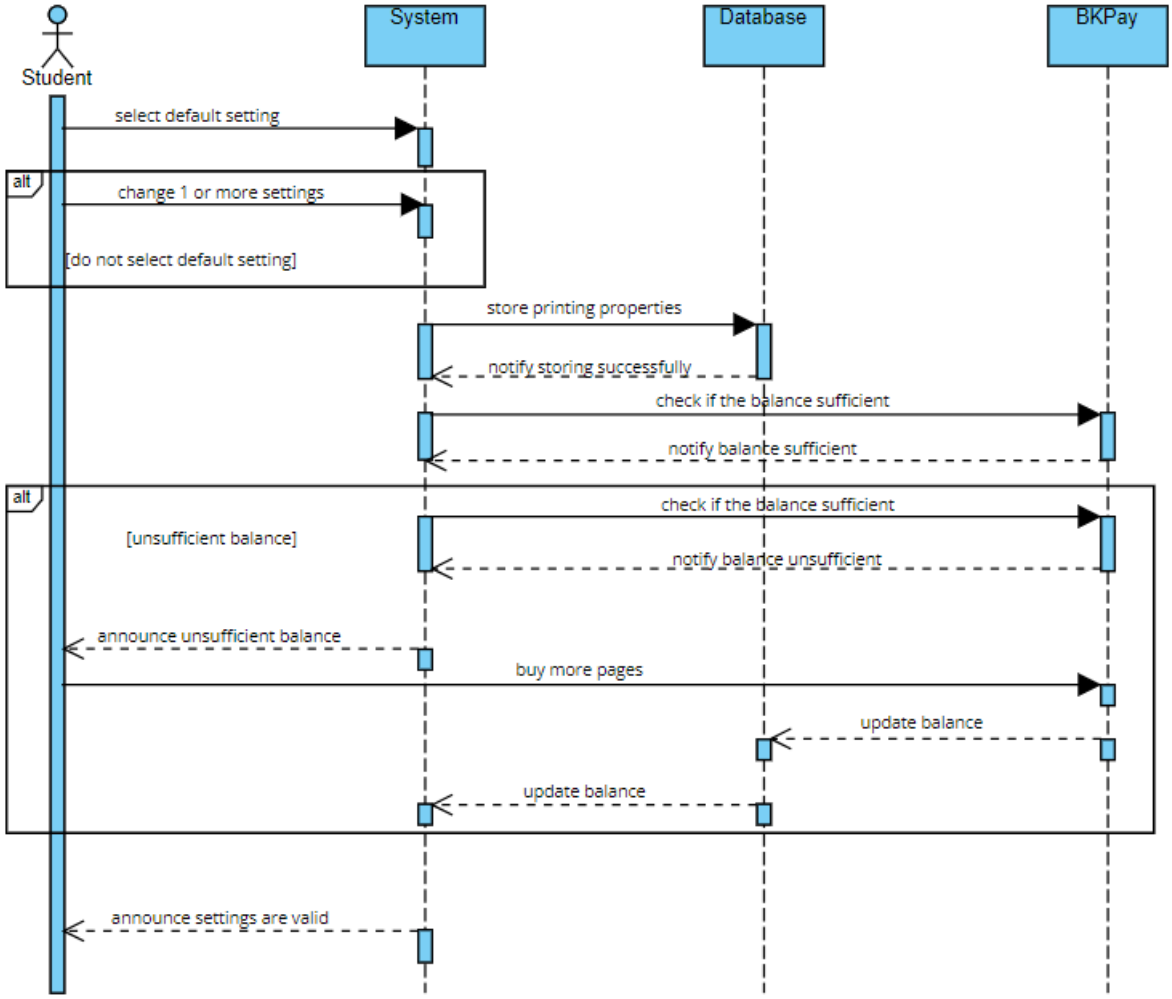

Figure 9: Sequence diagram for specify printing properties

This sequence diagram illustrates the use case of specifying printing properties. The key participants are Student, System, Database and BKPay. The sequence starts when the student selects settings and the system will store the properties into the database, the system will also check the balance, if the balance is sufficient, they will announce the settings are valid, completing the use case.

4. Choose printer

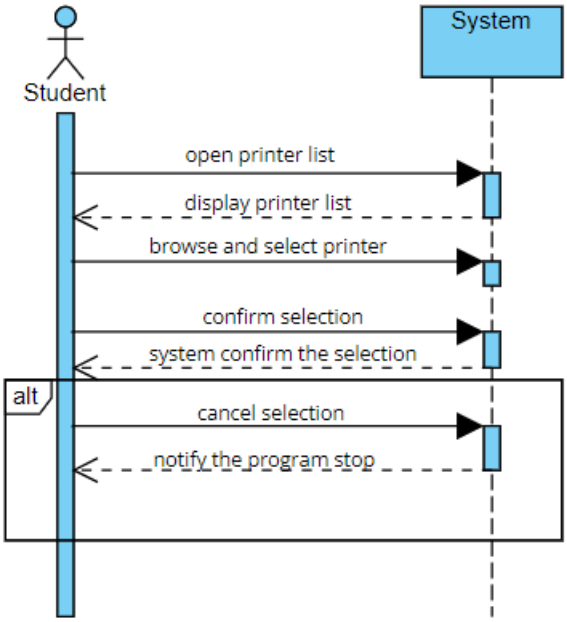

Figure 10: Sequence diagram for choose printer

This sequence diagram illustrates the use case of choosing a printer. The key participants are Student and System. The sequence starts when the student opens the printer list, and browses then selects the printer. The system then announces that they confirm the selection, completing the use case.

5. Check page balance

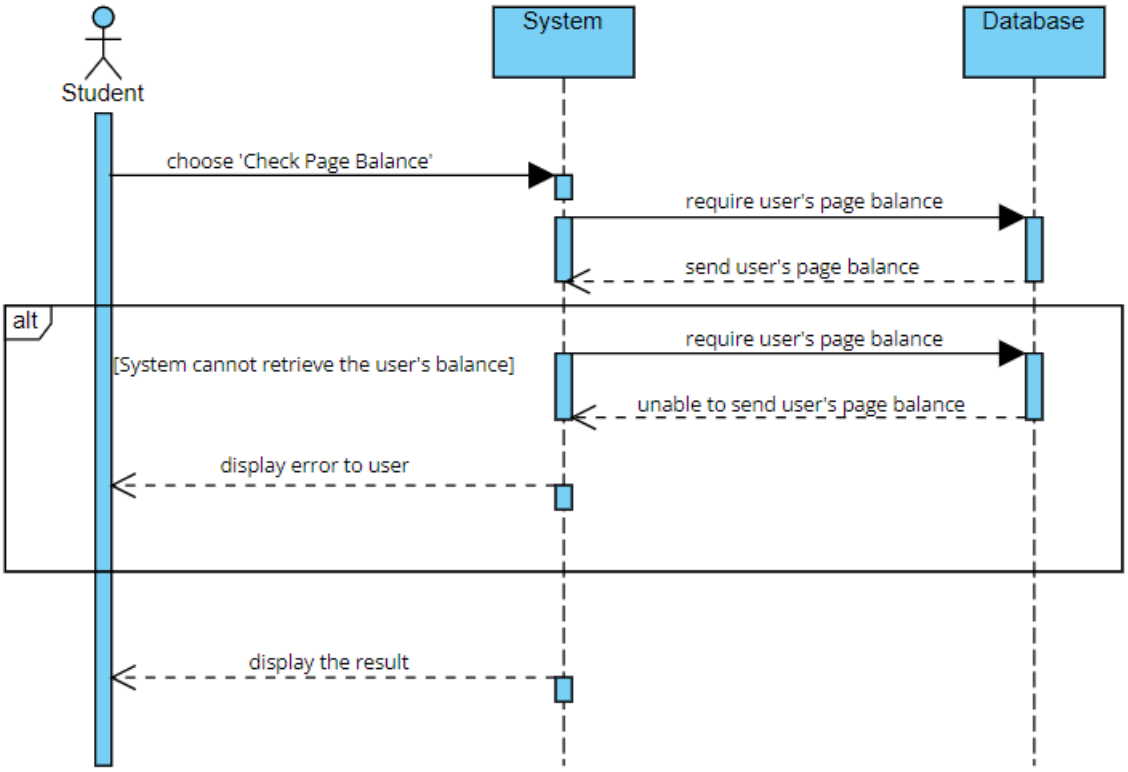

Figure 11: Sequence diagram for check page balance

This sequence diagram illustrates the use case of checking page balance. The key participants are Student, System and Database. The sequence starts when the student chooses the 'Check Page Balance' option. The system will require the database to send the user's balance, after they send the balance, the system will display it to the user, completing the use case.

3.3 Class diagram

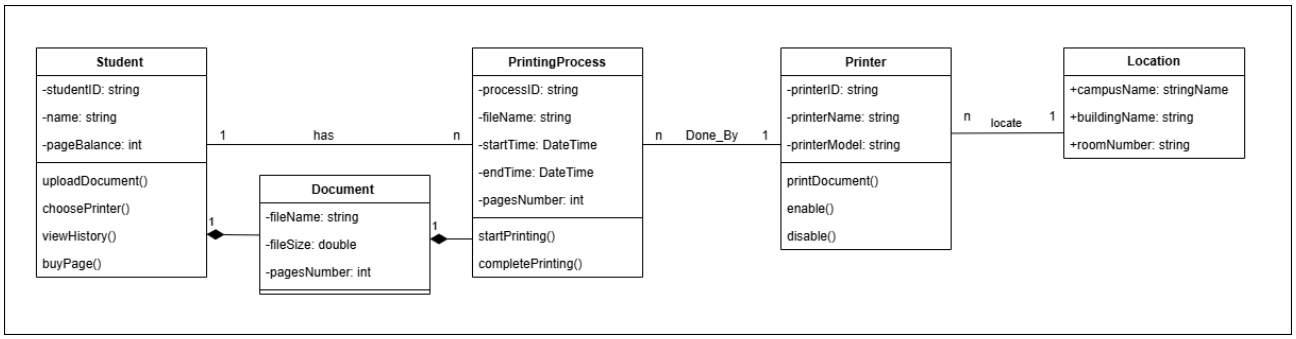

Figure 12: Class diagram

3.4 Wireframe

The following designs are UI for students' account

Figure 13: Main menu for students

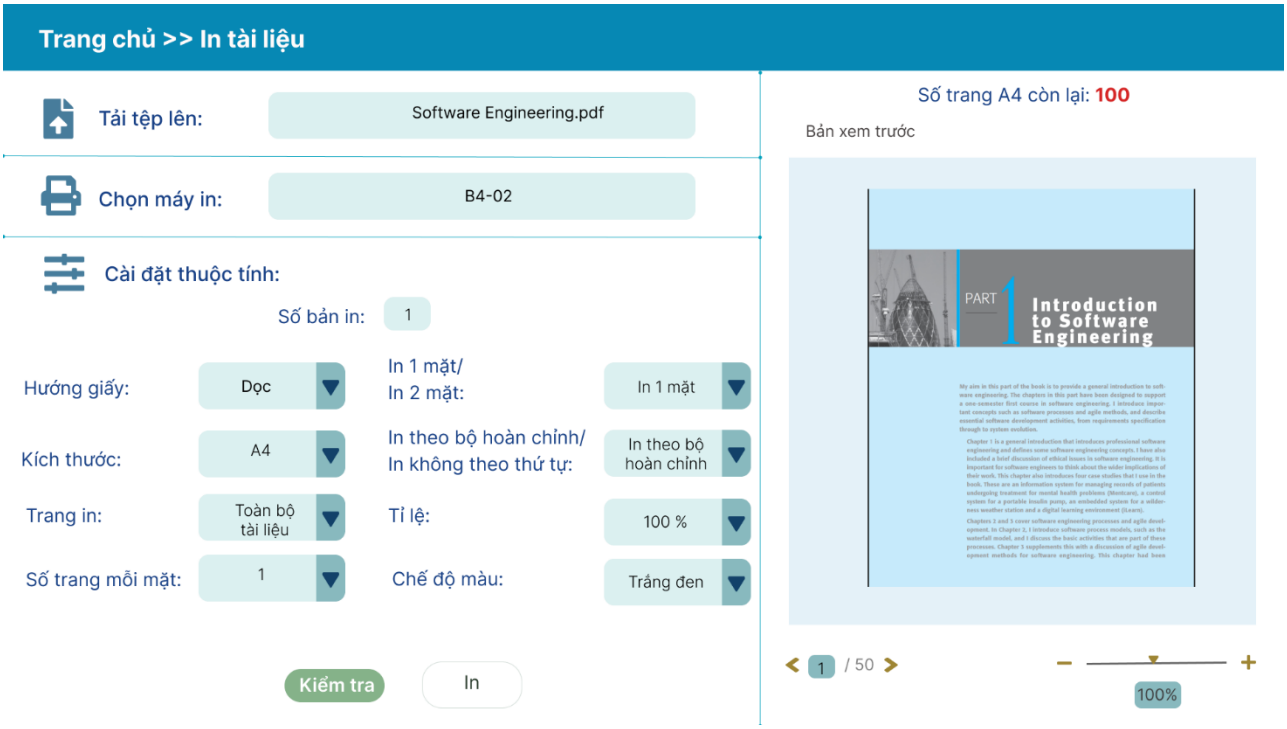

Figure 14: Printing screen for students

Figure 15: Buy more page

Figure 16: Upload document

Figure 17: Choose printer

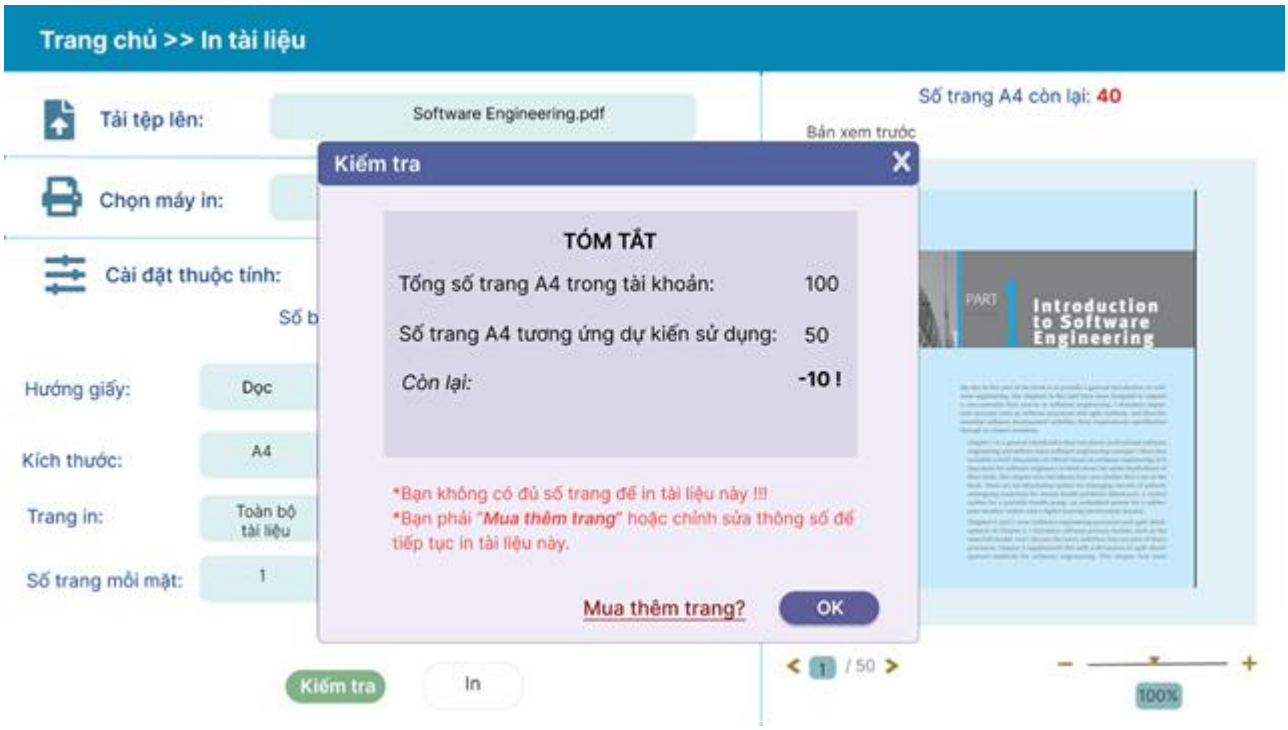

Figure 18: Review printing

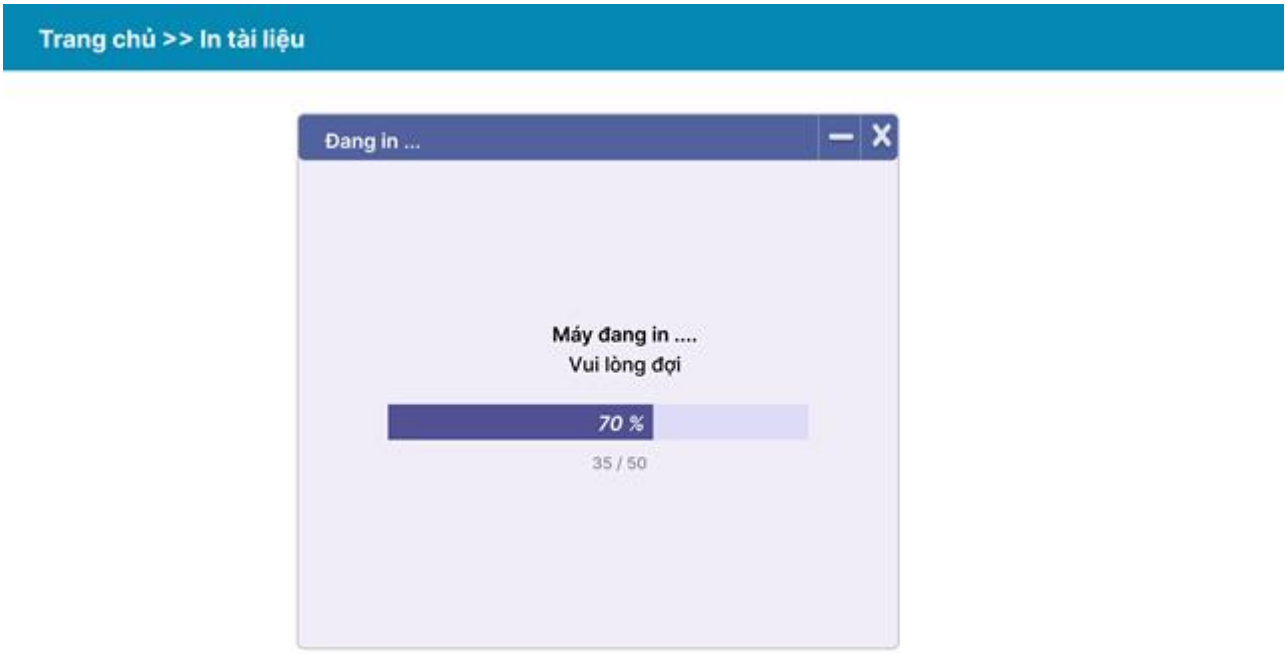

Figure 19: Print progress

The following designs are for SPSO to manage students and printers

Figure 20: SPSO main menu

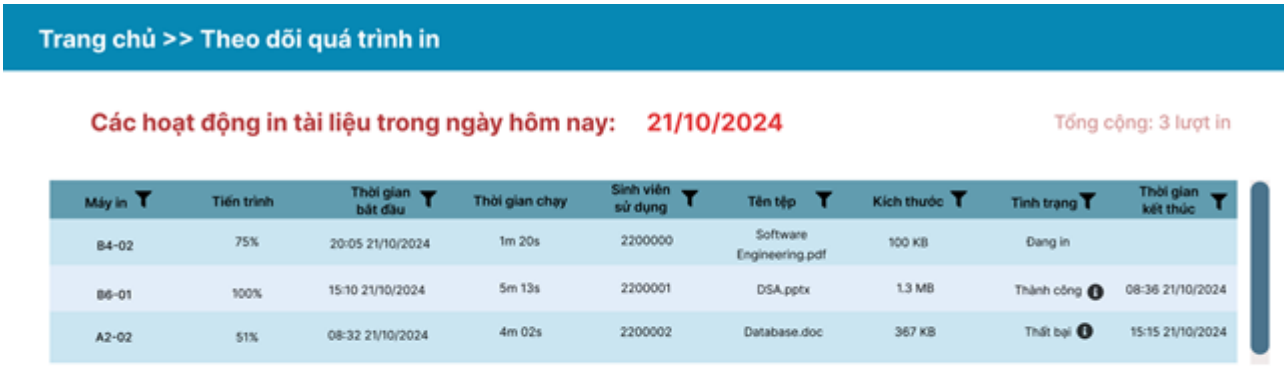

Figure 21: Printing log

Users need to authenticate using the central HCMUT_SSO authentication service

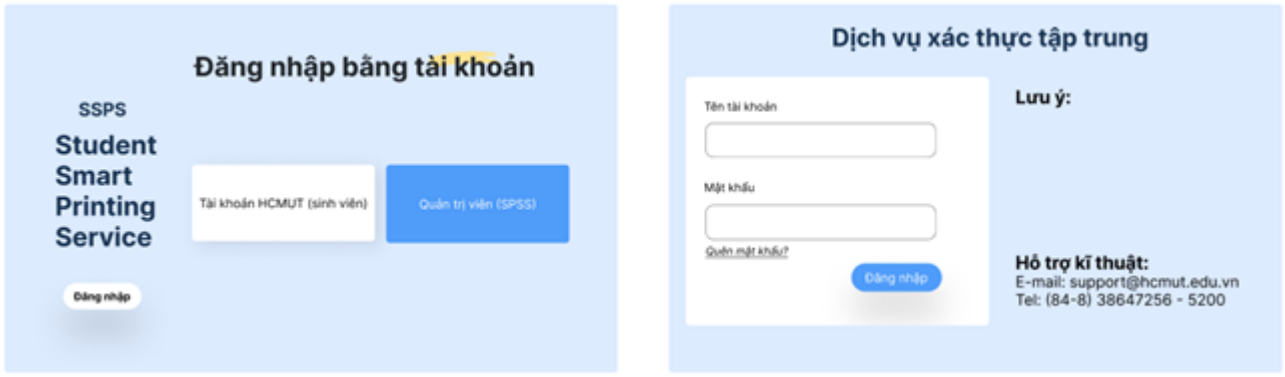

Figure 22: Authentication

4. Task 3: Architecture design

4.1 Layered architecture

1. Architecture diagram

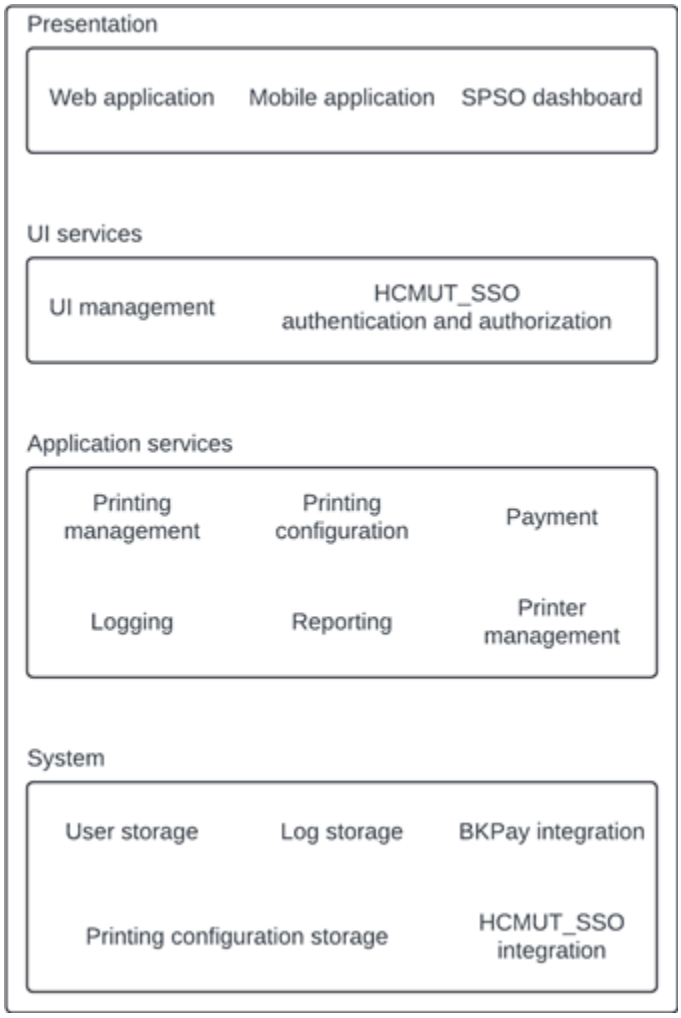

Figure 23: Layered architecture diagram

2. Presentation strategy

Our SPSS design focuses on the smooth experience of users as students and SPSO in both website and mobile app versions, hence the interface is flexible according to devices from small screens of smartphones to large screens of desktops. We apply the column-grid layout for consistent content organization throughout page navigation. The structure mainly follows the minimalist style, using white with various shades of blue, green, and purple as the primary colors, but contrasting colors like red and orange will be used to highlight components such as errors, warnings, etc. We also place icons next to some textual items for a better visual effect. On each page, a header bar shows where users are visiting as a text title and has a “selection” icon with a list of all pages that provide users with quick access to other

main sections without returning to the Home page. Typography will be chosen for readability across devices, with larger fonts for headings and titles, and smaller legible fonts for supporting text and description The buttons are divided into 2 types: active and inactive. The former is represented with full background color and the latter with empty background. We also create an instruction page in case users find it difficult or confusing. Besides some fields that need to be completely inputted from the keyboard, we add many combo boxes as a quicker way to select the inputs. For students, we provide default printing property settings to save time. There is also a "filter" area to choose a list of items with wanted properties from the list. To enhance user experience, there is a link to a "Report issue" form where users can describe their problems in detail and insert photos or videos as well.

3. Data storage approach

During production, the system should have its API service to ensure that our system is secured. The requests endpoint our services are hidden behind a proxy layer implemented in the API gateway, and every request requires a token for authentication.

- BK Pay: Our system does not access any information (the system just sends a request for money that students need to pay for their use of service), and the payment information should be preceded by BKPay department, also the system simply requires a navigation capability from BKPay service.
- HCMUT SPSS: the endpoint is used together with SSPS application ID, which is registered through HCMUT central authentication department, alongside with the callback url to HCMUT SSO web- site. The application ID, also known as the client in the client - server of CAS authentication service, acts like an API key to access the features of HCMUT SSO service.

The system will store its user interface components (also known as the Frontend part) on a separate server. Since our frontend framework is chosen to be ReactJS, the bundled files (consists of App.vue, index.js and style.css) which are created in the build process will be sent to the client on request to the server (on loading page). The frontend server is built with Docker.

On the other hand, the frontend will communicate with the other components (services, database, etc.) via an API gateway. By implementing this, the system will gain the following benefits:

- Security: The API gateway acts as a proxy to prevent external sources from accessing the actual endpoints of our system, which prevent the APIs from (D)DOS attacks, or exposing sensible contents from the APIs.
- Rate-limiting and Caching: The SSPS service is estimated to handle a large quantity of requests at a time, it is valuable to have the capability to persist information from requests for an extended time period. Rate-limiting also helps in (D)DOS protection.

4. API management

In this project, API management will concentrate on secure, organized and scalable API interactions. About API Gateway Platform, which helps us traffic management as well as security, besides, it helps in managing access control, rate limiting, routing, load balancing, authentication and request throttling. In this project, we want to use Kong Gateway, this is one of the most widely used in the world right now. This is a lightweight and highly scalable platform for developers. Subsequently, talking about functionality tools, for secure access to the system, OAuth is one of the suitable tools for managing access authorization request, grant and asking to access the user resources. Then, to monitor, alert and observe in the system, Prometheus together with Grafana are the good solution. Prometheus works in the monitoring and alerting toolkit, which is developed by SoundCloud, it also tracks the usage of CPU, memory and other performances in the system. Grafana works in data visualization and analytics, it is used to visualize data from many sources including Prometheus and other Database Management Systems such as MySQL, SQL Server, …

4.2 Component diagram

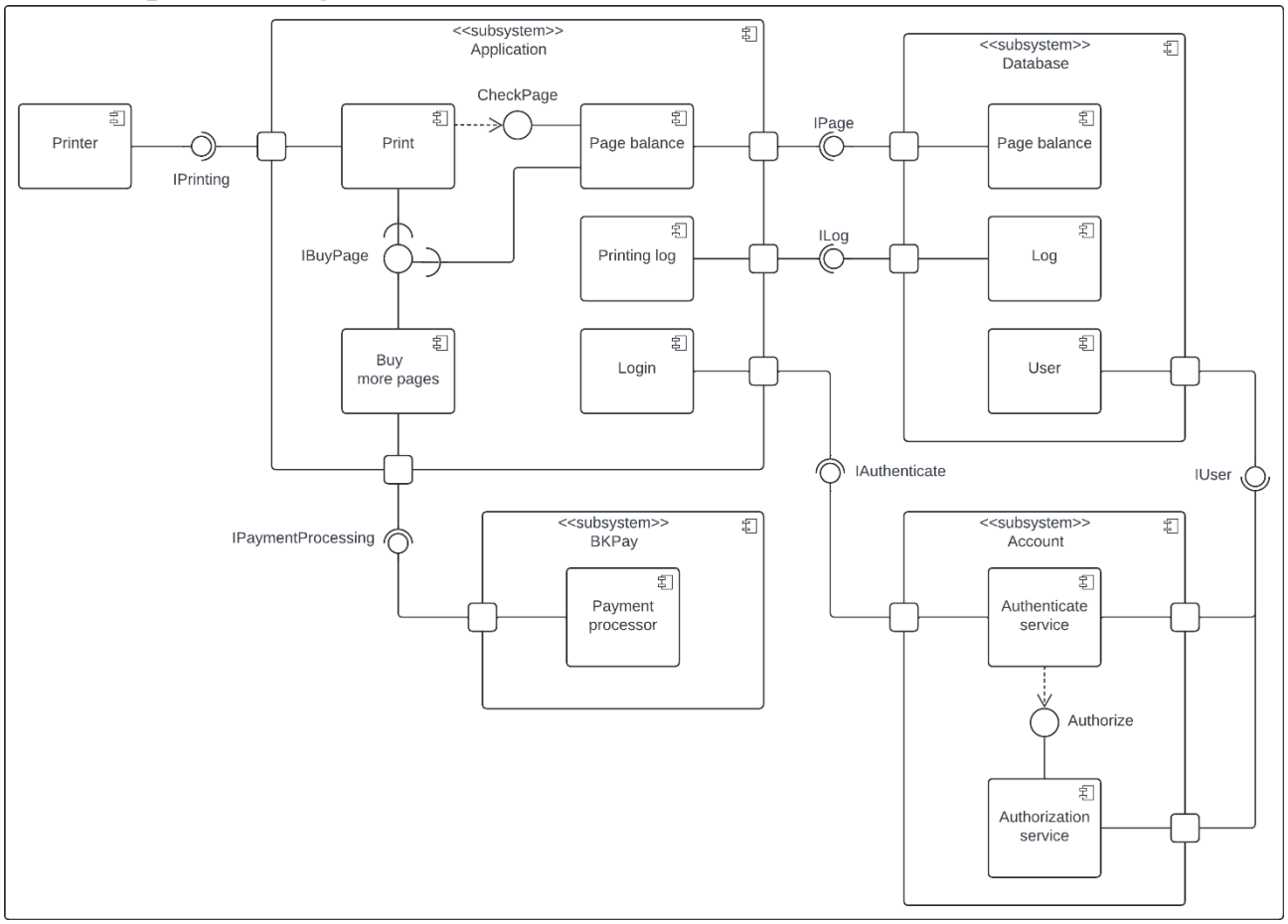

Figure 24: Component diagram

This component diagram illustrates the architecture of the Student Smart Printing Service (HCMUT_SSPS), detailing how various components work together to enable students to print documents on campus.

Components Overview:

Printer: Represents physical printers available on campus. It provides an interface (IPrinting) for interacting with the print functionality within the application.

Print: Part of the Application subsystem, responsible for managing print requests, including verifying page balance and sending print jobs to the selected printer.

Buy More Pages: Allows students to purchase additional pages, deducting costs via the online payment system.

Page Balance: Tracks students' remaining pages for printing and is part of the Database subsystem.

Printing Log: Records all printing activities for auditing and history tracking. This log data can be accessed by both students and SPSO.

User: Stores basic information about each user, managed within the Database subsystem.

Login: Manages user login within the Application subsystem by interfacing with the Account subsystem for authentication.

Payment Processor: Part of the BKPay subsystem, handles financial transactions when students buy more pages.

Authenticate Service: A component within the Account subsystem that verifies user identities and interfaces with the "Login" component.

Authorization Service: Controls access to system resources.

Interfaces and Dependencies:

IPage and ILog: Interfaces used by the Application to interact with the Database for retrieving page balances and printing logs.

IPrinting: Interface allowing the Print component to send print jobs to available printers.

IPaymentProcessing: Interface connecting the Application to the BKPay subsystem, allowing students to make payments for additional printing pages.

IAuthenticate and IUser: Interfaces within the Account subsystem for managing user authentication and authorization requests.

Components communicate primarily within the application layer and database, while the Account subsystem ensures all interactions are secure and authenticated.

External Dependencies:

The BKPay subsystem serves as an external dependency, managing payment transactions when students purchase more pages.

The Account subsystem relies on an external authentication service, HCMUT_SSO, to authenticate users before granting access to the system.

5. Task 4: Implementation – Sprint 1

5.1 Online repository

We decide to use Github as our online repository.

Github link: [241 SE Assignment Github]

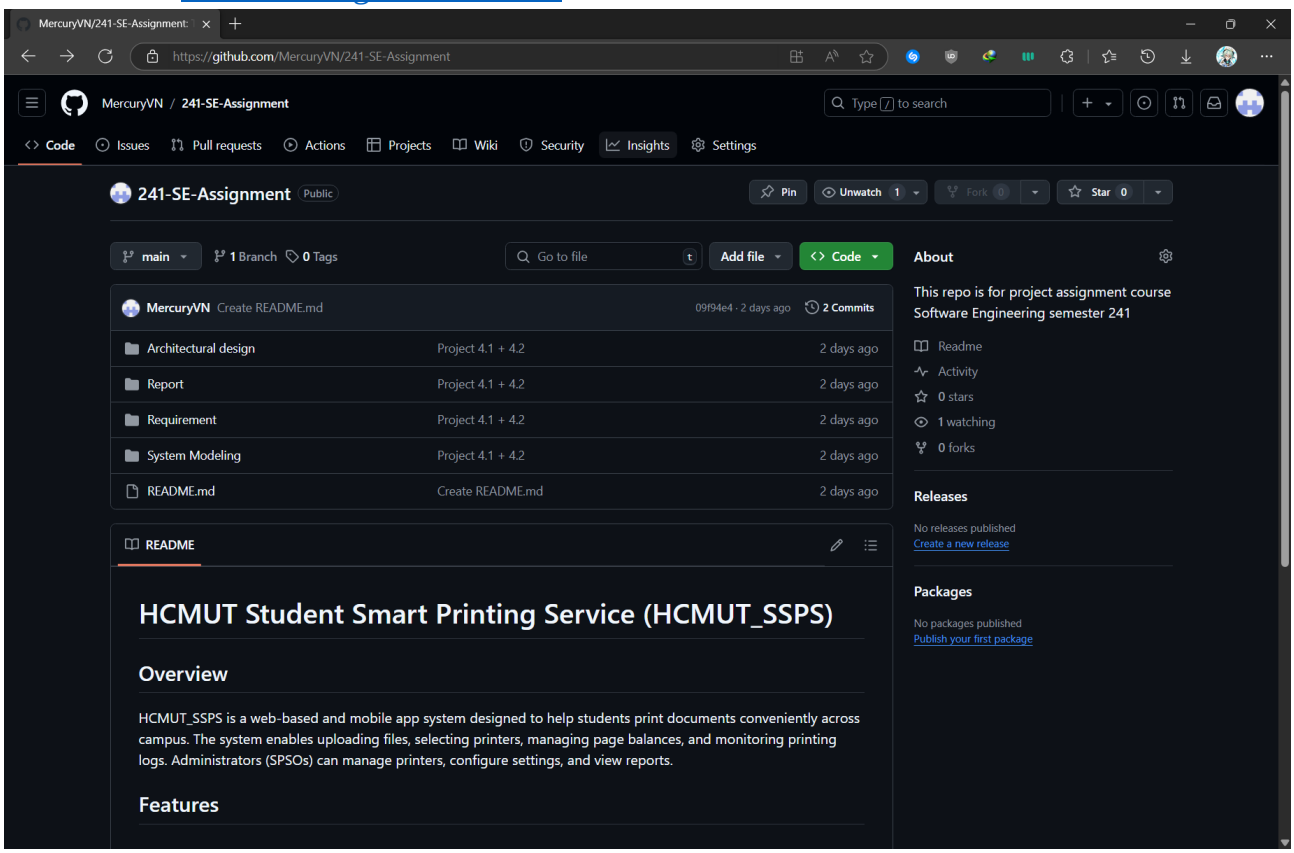

Figure 25: Github dashboard

5.2 Usability test

5.2.a Define tasks

General:

- Login to the system using your credentials

SPSO:

- View printing log
- Filter printing log
- Change permitted file types
- Enable/Disable printer

Students:

- Open printing page

- Upload document
- Set printing properties
- Buy more pages

5.2.b Test strategy

Testing goal:

- Qualitative: Observe user behavior while performing tasks, note challenges, confusion, or feedback.
- Quantitative: Measure task completion times, success rates, and error rates.

Testing Method:

- Remote Testing: Students can test on their own devices, providing insights into how the system performs in varied environments.
- In-Person Testing: Complex task, like printing process, need to be tested in controlled environment to observe detailed interactions

Participants:

- Students: Diverse group with different majors, levels of tech proficiency, and printing needs.
- SPSOs: Staff familiar with the service's administrative functions.

Devices:

- Extensive testing on web-based apps to ensure a more comfortable experience.

Feedback Collection:

- Use screen recording tools (for remote tests) or observation sheets (for in-person tests).
- Collect direct feedback through surveys.

Evaluation Criteria:

- Task Efficiency: Time taken to complete tasks.
- Effectiveness: Task success rate and frequency of errors.
- User Satisfaction: Ratings of ease of use, clarity of interface, and overall satisfaction.

5.2.c Documentation

Overview:

This usability testing is to validate the functionality, usability, and compatibility of the MVP 1 (the aforementioned mockup user interface of the printing module) as a part of the Student Smart Printing Service Web across desktop and mobile views. The testing scope includes features with specific types of users as follows:

- Student and SPSO: Log in.
- Student: Print document (including upload file, select printer, set properties, check page balance, confirm print action), and buy more pages.
- SPSO: view the printing status of the current day.

Participant demographic:

1. Students:
   From different academic years and faculties.
   Have accounts authenticated by the HCMUT_SSO authentication service.
2. Administrators staff
   Manage and test the backend features like monitoring or troubleshooting.

Task tested:

1. For students
   a) Account
      Log into the system using the university's account
      View Printing Log
      Buy more printing pages
   b) Printing process
      Upload the document with permitted format
      Choose available printers on the campus
      Specifying printing properties (e.g., single/double-sided, color/black-and-white, number of copies)
   c) Receiving the document
      Collect printed documents after submission at the chosen printer
   d) Error handling
      Resolve a failed print attempt or retrieve a refund for a failed job
2. For Administrators
   View all printing logs
   Mark a specific printer as unavailable for maintenance

Feedback:

1. User Interface
   Positive: The UI was visually intuitive with clear layouts and easily accessible navigation.
   Negative: Some participants found the printer selection page overly cluttered, especially when there were multiple printers listed.
2. Functionality
   Positive: The flow for uploading documents and setting printing properties was smooth and clear.
   Negative: The feedback when a file type was not supported was delayed, causing some confusion.
3. Usability

Positive: Students appreciated the preview function for ensuring correct printing formats.

Negative: The option to buy more pages required multiple redirects, which some users found inconvenient.

4. System Responsiveness
   Positive: System response time for basic operations like logging in and loading the dashboard was satisfactory.
   Negative: Minor delays were observed during remote testing in displaying printer statuses.
5. Accessibility
   Positive: The system supported screen readers effectively and catered to different user groups.
   Negative: Users suggested having tooltips or guides for first-time users.

Issue:

1. Printer Status Display
   Issue: Real-time updates for printer availability were inconsistent.
   Impact: Could lead to users selecting unavailable printers.
2. Error Handling
   Issue: Error messages for unsupported file types lacked clarity and timing.
   Impact: Users had to retry without clear guidance.
3. Payment Integration
   Issue: Buying more pages through BKPay was cumbersome due to multiple steps.
   Impact: Potential frustration among users needing immediate printing credits.
4. Mobile Optimization
   Issue: Some buttons and input fields were misaligned on smaller screens.
   Impact: Affected usability for students accessing the system on mobile devices.
5. Filtering Options
   Issue: Filter options in the printer list were not intuitive for first-time users.
   Impact: Users required additional time to locate specific printers.

Recommendation:

1. User Interface:
   Improve the printer selection interface by adding sorting or categorizing options or searching filter functions. This may help users find their desired printer easily.
2. Functionality:
   Improve by adding error handling, it must display immediately, clear feedback when an unsupported file type is uploaded.
3. Usability:

Improve by reducing the number of redirects, we can use the modal window for more efficiency.

4. System Responsiveness:
   Enhance real-time updates for printer statuses by implementing a more robust polling mechanism or using WebSockets for continuous updates. This would reduce delays and prevent users from selecting unavailable printers.
5. Accessibility:
   Adding tooltips and adding guides for first-time users.

Conclusion:

Our MVP 1 has met the basic requirements that the normal flows of action can be executed without errors. Our web must, however, slightly alter the UI layout to improve visual appeal, especially in small screens, and when certain exceptions arise, we need to provide users with straightforward warnings and recommendation to correct issues.