
Software Requirements Specification for

Smart printing service for students at HCMUT

Version 1.0 approved

Prepared by:

- 1. Truong Minh Khang - 2252314**
- 2. Tran Tri Dung - 2252133**
- 3. Tran Trung Vinh - 2252914**
- 4. Võ Hoàng Phúc - 2252650**
- 5. Võ Nguyên Phát - 2252607**
- 6. Trinh Tri Bao - 2252077**

**Department of Software Engineering
Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology – VNU-HCM**

22nd September 2024

Table of Contents

1. Requirement elicitation (1.1, 1.2)	2
1.1 Domain Context	2
1.2 Stakeholders and Needs	3
1.3 Benefits of the System	4
1.4 Functional Requirements	4
1.5 Non- Functional Requirements	5
2. Use-case Diagrams (1.3)	6
2.1. Use-case Diagram for the Whole System	6
2.2. Use-case Diagram for Printer API Module	7
2.3. The Details of Use Cases in Printer API Module	7
3. Diagrams of Printer API module	8
a. Sequence diagram for parts of the Printer API module	8
b. Activity diagram	17
c. Class diagram	25
d. User interface	27

Revision History

Name	Date	Reason For Changes	Version

1. Requirement elicitation (1.1, 1.2)

1.1 Domain Context

In a university environment, managing documents is crucial for students as they rely on them to deepen their understanding and reference materials for assignments and reports. Therefore, having an app that allows students to efficiently print documents on campus is essential. This would save time and eliminate the need to visit off-campus photocopy centers, making the process more convenient and efficient.

To meet the documentary needs of students, HCMUT plans to develop the Student Smart Printing Service (HCMUT_SPSS) via web and mobile application for students' convenience. The primary goal of this system is to facilitate document printing on campus. It will be primarily utilized by students and the Students Printing Service Officer (SPSO). The system will include two main features: student printing management and SPSO printing management.

Student printing management encompasses several functionalities, such as uploading document files to the system, selecting a printer, and specifying printing properties (e.g., paper size, number of pages). Each student's actions, including student ID, printer ID, file name, printing start and end times, and the number of pages for each size, will be logged into the system. Students can view this information through the printing history feature. Additionally, students are allocated a default number of A4 pages for printing each semester. If they exhaust their allocated pages, they must purchase additional pages through the online transaction system - BKPay.

SPSO printing management includes functionalities such as adding, enabling, or disabling printers, managing the default number of pages allocated to students, setting the dates when the system will provide these default pages in each semester, and specifying the permitted file types accepted by the system. Additionally, SPSO can view usage reports of the printing system, which are automatically generated at the end of each month or year.

All users who access the system have to be authenticated by the HCMUT_SSO authentication service before using the system.

1.2 Stakeholders and Needs

1.2.1 Stakeholders:

Students: The primary users of the system, students need an efficient, cost-effective way to print documents on campus. They require easy access to printers, flexible printing options, and the ability to manage printing credits and history.

Student Printing Service Officer (SPSO): The SPSO manages the printing system, ensuring printers are functioning, configuring the system settings, and handling issues that arise. They need a system to monitor printer activity, manage settings, and generate reports.

University IT Department: Responsible for the technical maintenance of the system, the IT department needs access to logs, troubleshooting tools, and integration with the university's authentication service.

1.2.2 Needs:

- Students:
 - Access to multiple printers across campus.
 - Options to configure print jobs (paper size, one-/double-sided, number of copies).
 - A seamless online payment system for purchasing extra printing credits.
 - A clear overview of printing history, including credits remaining and pages printed.
- SPSO:
 - The ability to monitor and manage printer statuses (add, enable, disable printers).
 - View and generate reports on system usage.
 - Set system configurations, such as default page credits and file types.
- University IT:

- Ensure system integration with HCMUT's Single Sign-On (SSO) for secure authentication.
 - Troubleshoot system issues and maintain seamless operation.
- 1.2.3 Benefits of the HCMUT Smart Printing Service:
- For Students: They gain an easy-to-use, cost-effective printing service with flexible options for managing print jobs and credits.
 - For SPSO: They have centralized control over printers and access to detailed logs and reports, improving efficiency and accountability.
 - For the University: The system ensures the efficient use of printing resources, reduces waste, and integrates with existing infrastructure for smooth operations

1.3 Benefits of the System

For students of HCMUT, smart printing service offers convenience and flexibility as students can print their documents anywhere around the school campus without waiting. Students also don't have to pay a small amount of change just to print 1 or 2 pages of documents. It also helps students to choose the format for their document such as paper size, one-/double-sided, number of copies,... HCMUT_SSPPS also has a history function to track students' usage, ensuring that students still have enough pages for the next print. Mobile apps also enhance students' experience in using the service as it is portable and easy to access. For SPSO, having a management system helps the operation smoother in adding, removing or keeping track of the printer. Moreover, by controlling the input and output of documents, only compatible documents are processed, minimize errors and raise efficiency in using resources. The system also collects data for analysis, which helps the services better or normally data for students to study. For university administration, SSPPS helps in financial and operational benefits. The report from the service will help the university to make decisions about infrastructure and resources usage more reasonably for both students and university. The university can keep track of the report to know whether they should invest more or less on printers. In summary, all stakeholders will benefit from SSPPS, even more if the service continues to update and grow.

1.4 Functional Requirements

For student:

1. The system allows students to upload a document file for printing.
2. The system allows students to choose a printer from a list of printers available.
3. Students are allowed to specify printing properties, such as paper size, pages to be printed, single or double pages and number of copies.
4. The system logs all student printing actions, including student ID, printer ID, file name, printing start and end time and the number of pages printed.

5. Students are allowed to view their printing history.

For Students Printing Service Office (SPSO):

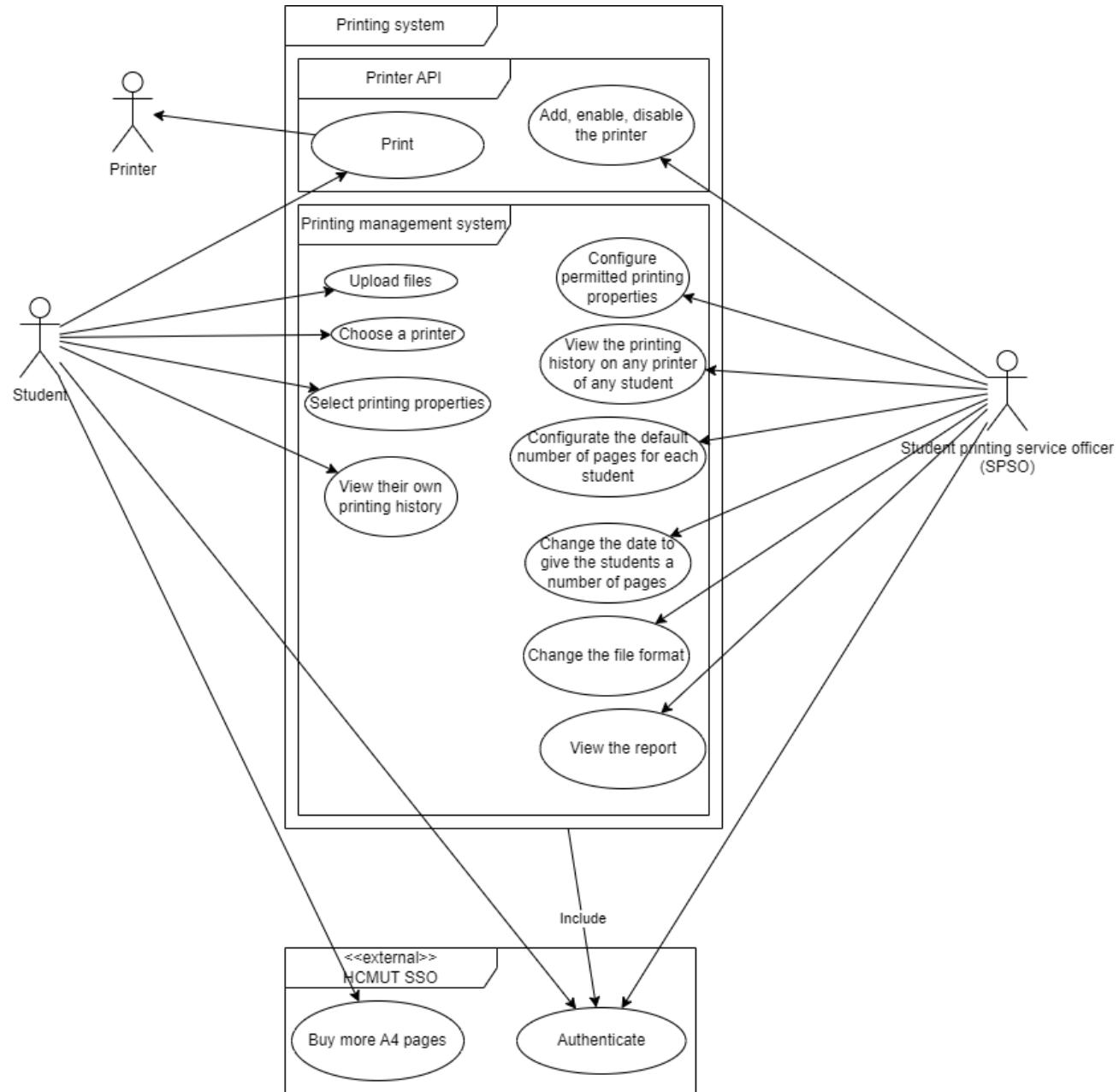
1. SPSO are allowed to manage printers such as add/enable/disable a printer.
2. SPSO can manage configuration of the system such as changing the default number of pages, the dates that the system will give the default number of pages to all students, the permitted file types accepted by the system.
3. SPSO are allowed 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.
4. The system allows the SPSO to configure and manage the permitted file types for printing.
5. SPSO are allowed to view the reports of the use of the printing system at the end of each month and each year at any time.

1.5 Non- Functional Requirements

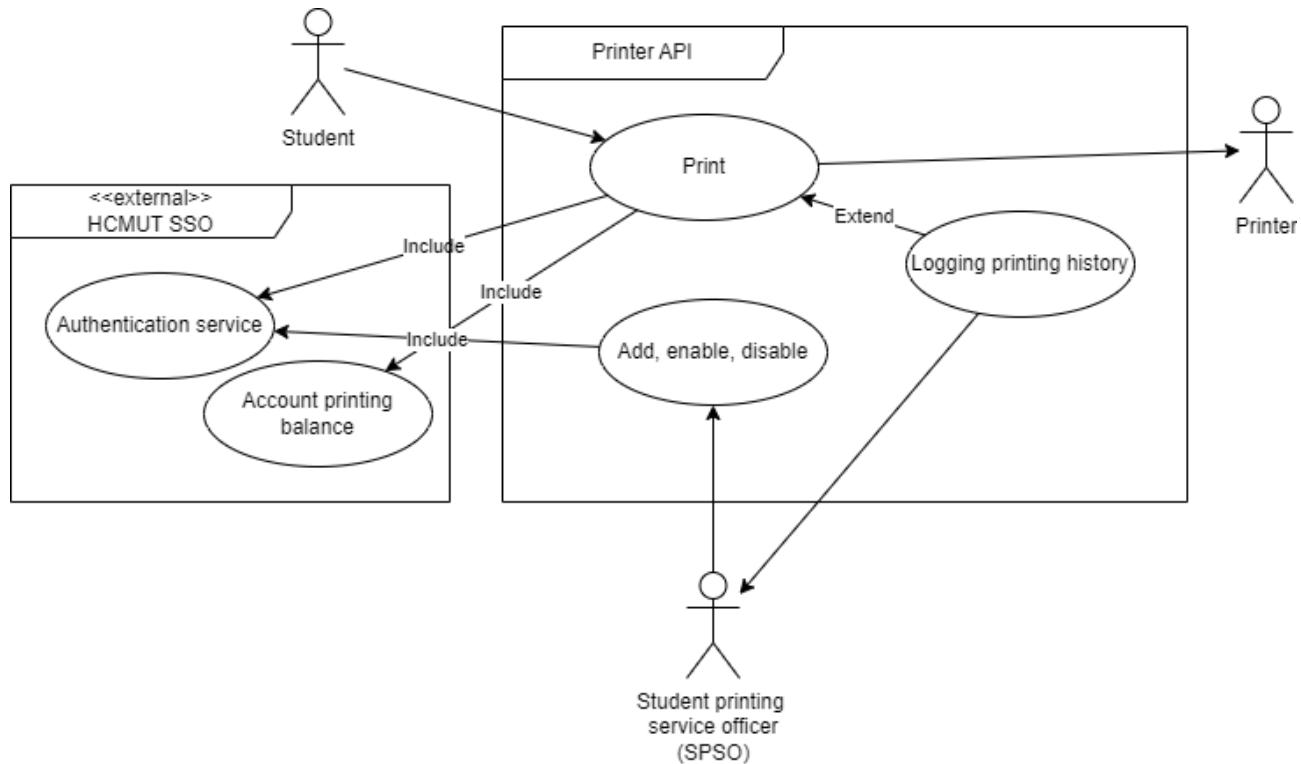
1. The system should maintain a log of printing actions for at least one year.
2. The system shall be scalable to accommodate multiple campuses.
3. The system must be compatible with the HCMUT SSO to allow the user to authenticate themselves using either their university student card or the application.
4. The system shall ensure the security of personal data and financial transactions, adhering to relevant data protection regulations.
5. The system shall be compatible with standard printers from multiple manufacturers.
6. The system must be capable of printing A4 and A3 paper.
7. The system must be capable of storing the information of all students.
8. The system must be capable of managing all printers, including the newly added ones.
9. The system can function on both web applications and mobile applications.

2. Use-case Diagrams (1.3)

2.1. Use-case Diagram for the Whole System



2.2. Use-case Diagram for Printer API Module



2.3. The Details of Use Cases in Printer API Module

1. Use Case Print

Use case	Print
Precondition	The user is authorized in the HCMUT SSO The user document is uploaded The printer is connected to the Internet
Postcondition	Print some paper

Normal flow	<ol style="list-style-type: none"> 1. The user specify the printing format 2. The user specify the document file 3. The user choose the printer 4. The printer print the document
Exception	<ol style="list-style-type: none"> 1. The document is not in the correct format 2. The balance of the user account is not sufficient 3. The printer is not available

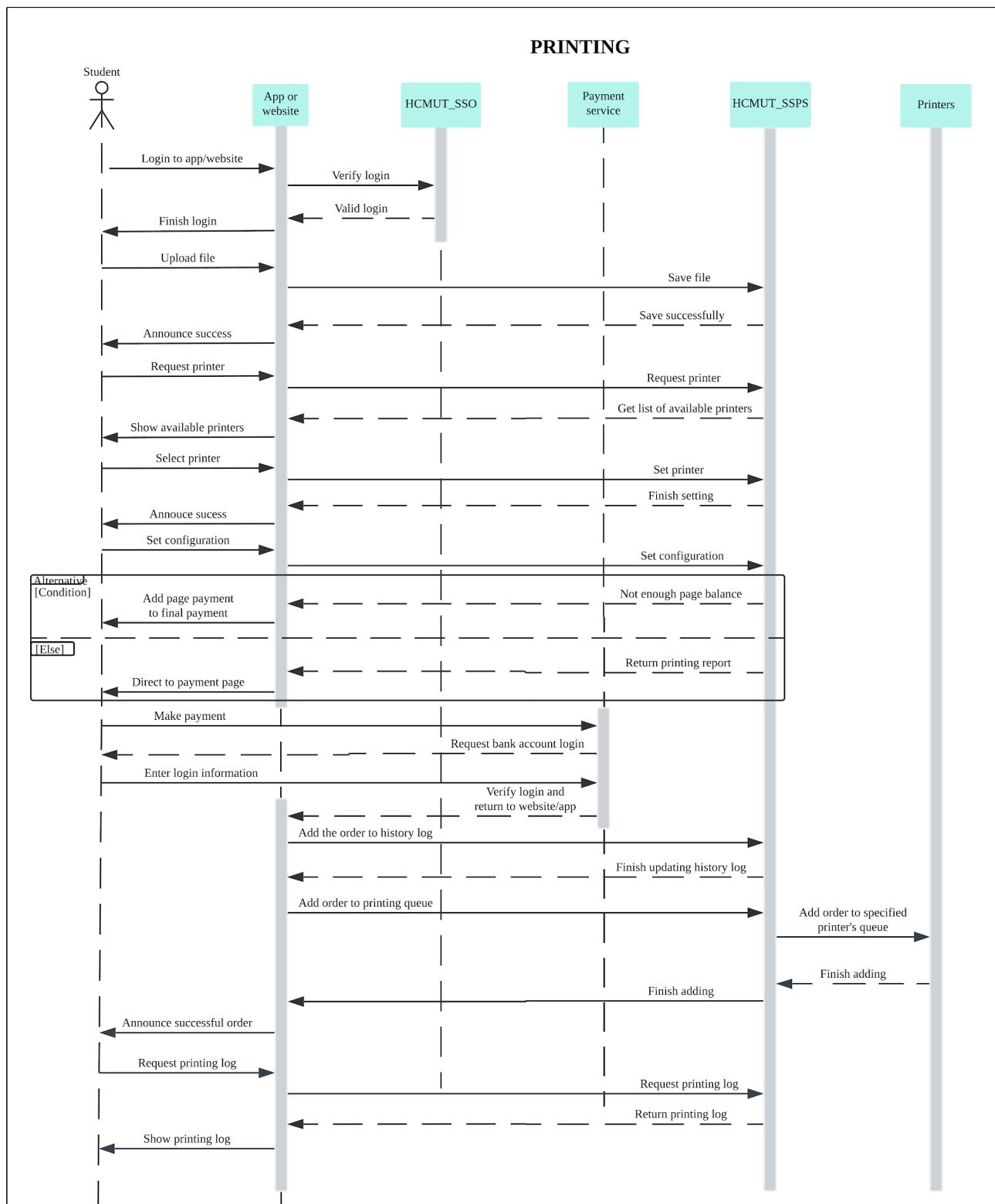
2. Add, enable, disable the printer

Use case	Add, enable, disable the printer
Precondition	<p>The user is authorized as SPSO in the HCMUT SSO</p> <p>The printer is connected to the Internet</p>
Postcondition	The printer is added or enabled or disabled
Normal flow	<ol style="list-style-type: none"> 1. The user accesses the application 2. The user specify the printer 3. The user specify action (add, enable or disable printer) 4. The system register the printer state
Exception	<ol style="list-style-type: none"> 1. The printer fails to connect to the system 2. The internet is disconnected on printer or the user device

3. Diagrams of Printer API module

a. Sequence diagram for parts of the Printer API module

i. Diagram for students



Description

This sequence diagram illustrates the process flow for a student initiating a print job through a printing service system that integrates login verification, payment processing, and queue management. The entities involved in this sequence include the Student, an App or Website interface, HCMUT_SSO for login authentication, a Payment Service, HCMUT_SSPO for additional configurations, and the Printers where the print job is executed.

The process begins with the student logging into the application or website. The login request is forwarded to HCMUT_SSO, which verifies the credentials. If successful, a valid login response is returned, allowing the student to proceed with the next steps in the printing process.

After login, the student uploads the file they wish to print. Once the file is uploaded, the application announces a successful upload. The student then requests a printer, prompting the system to show a list of available printers. Upon selection, the chosen printer is set, and the configuration process for the print job begins.

After selecting a printer, the student configures the print job settings. The system checks the student's page balance. If there is sufficient balance, the page count is added to the final payment. Otherwise, the system directs the student to the payment page to top up their balance.

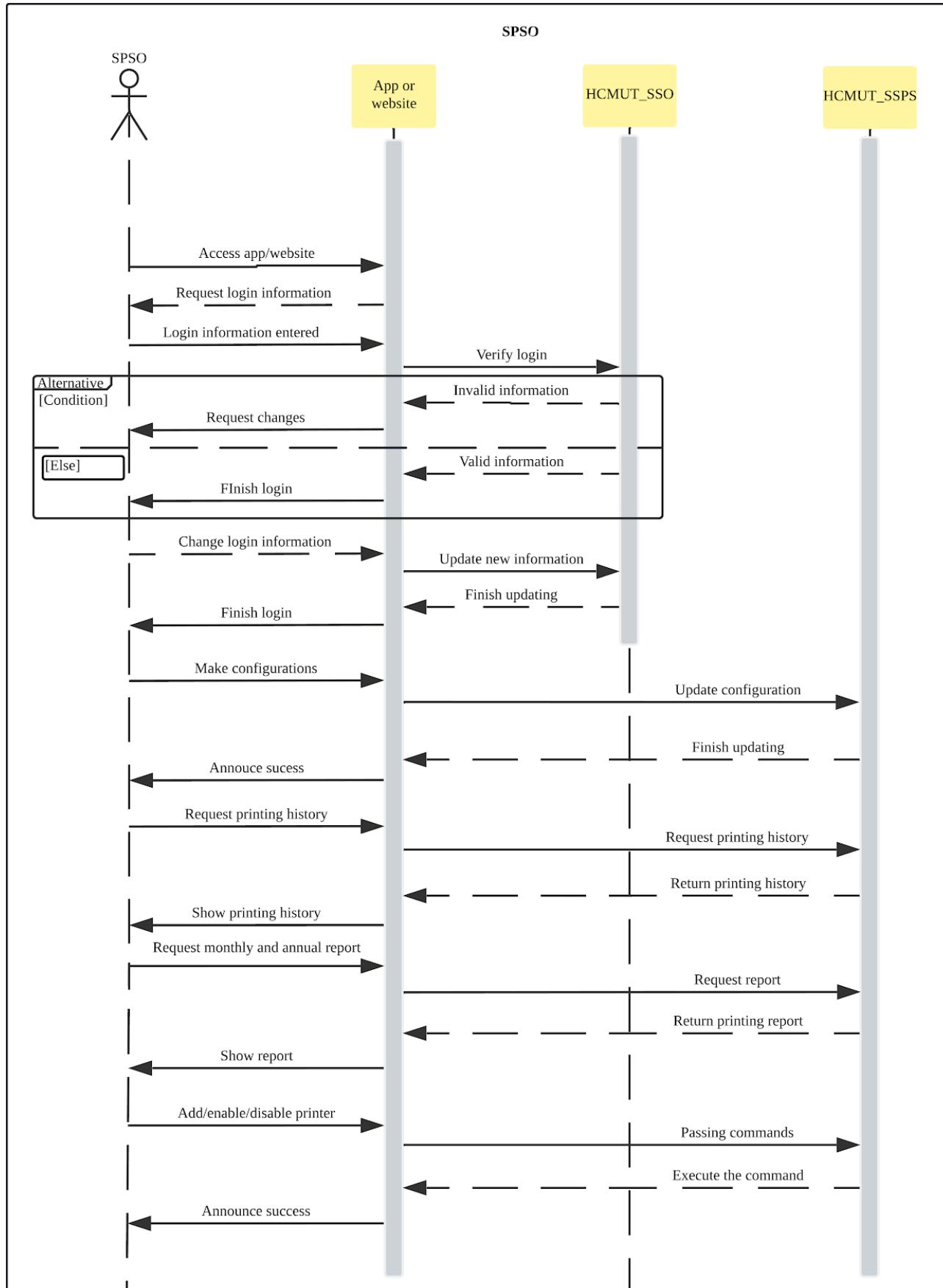
For insufficient balance cases, the student is directed to the payment page. Here, they enter their payment information, which the Payment Service verifies by prompting a bank account login. Once the login is verified, the transaction is completed, and control is returned to the application.

Once payment is confirmed, the system logs the order in the student's history log for record-keeping. The print job is then added to the printer queue, with confirmation messages announcing successful queueing. The system adds the order to the specific printer's queue, where it awaits execution.

Upon completion of the print job, the system allows the student to request the printing log. This log is returned to the student, providing a summary of the job's status, costs, and any other relevant details.

This sequence diagram effectively models the real-world interactions and dependencies in a student printing system, capturing each entity's role, and detailing the communication across multiple services. The use of conditional paths (for payment verification) adds depth to the diagram, indicating potential branching based on user circumstances (such as insufficient funds). Overall, this sequence is comprehensive and aligns well with the operational flow expected in a student-focused printing management system.

ii. Diagram for SPSO



Description

This sequence diagram represents the interactions within the SPSO (School Printing Service Operations) system, where a user (SPSO administrator) accesses a web application to manage login and printing tasks, handle configurations, view print history, generate reports, and manage printer settings. The main actors in the sequence are the SPSO user, the App or Website interface, HCMUT_SSO for single sign-on (SSO) authentication, and HCMUT_SSPPS for school-specific printing service operations.

The diagram begins with the SPSO administrator accessing the application or website. Upon accessing, the system prompts the user to enter login information. Once the login information is provided, the app forwards the credentials to the HCMUT_SSO system, which verifies the validity of the login attempt. If the information is valid, the user is granted access and can proceed with additional tasks. However, if the login information is invalid, the system requests the user to make necessary changes before attempting login again. This alternate condition flow ensures secure access and provides feedback for incorrect login attempts.

After a successful login, the SPSO administrator may need to update login information or adjust personal settings. The user changes the login information if required, and the application forwards this update request to HCMUT_SSO, which updates the user's information in its database. Once the update is completed, a confirmation is sent back to the app, allowing the user to continue with further actions. This sequence highlights the system's ability to manage user credentials and ensure they remain accurate, providing flexibility for users to maintain updated profiles.

Once logged in and authenticated, the SPSO administrator can proceed to make configurations for the printing setup. This involves setting parameters within HCMUT_SSPPS, ensuring that the configuration is aligned with the printing service's operational requirements. After configurations are set, the system sends a success message, allowing the administrator to move forward. The administrator can then request printing history to review previous activities. The request is forwarded to HCMUT_SSPPS, which retrieves and returns the printing history, providing a detailed log of prior print jobs.

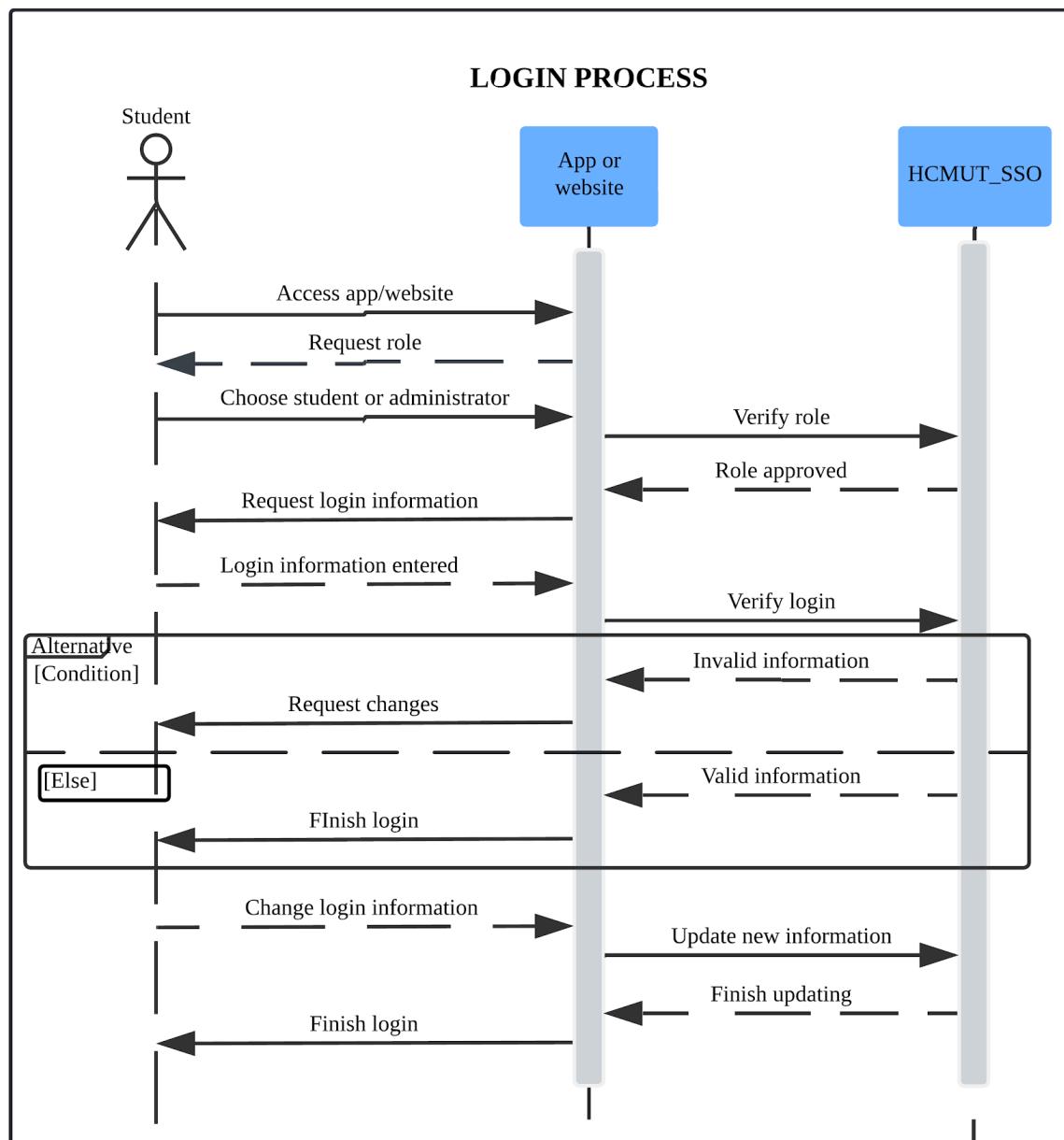
In addition to viewing print history, the administrator has the option to generate monthly and annual reports. The administrator requests these reports, and HCMUT_SSPPS processes the request, pulling the necessary data from its records. The report, once generated, is returned to the administrator through the app interface. This feature supports SPSO administrators in analyzing printing trends, understanding usage patterns, and managing resources effectively.

The SPSO administrator is also responsible for managing the printers. This may involve adding new printers, enabling or disabling existing ones, or updating printer statuses. The administrator issues these commands through the app interface, which sends them to HCMUT_SSPPS. The HCMUT_SSPPS system executes the commands, adjusting the printer

configurations as requested, and sends a success notification back to the administrator, ensuring the desired changes have been made.

Overall, this sequence diagram provides a structured view of the interactions and dependencies among various components of the SPSO system. The diagram outlines a comprehensive process flow for secure login, user information management, configuration settings, historical data retrieval, report generation, and printer management, all of which are crucial for effective printing service operations within the school system. This structured approach allows SPSO administrators to efficiently manage their responsibilities and supports a smooth, streamlined experience for managing the printing system.

iii. Diagram for Authentication process



Description

The provided sequence diagram offers a detailed visualization of the login process within a system that leverages the HCMUT_SSO service for authentication. This process is essential for ensuring secure access to the system's resources and functionalities.

The login process commences when a student initiates access to the application or website. The system prompts the student to select their role, either as a "student" or "administrator." This initial step is crucial as it determines the level of access and privileges granted to the user.

Once the role is selected, the system forwards the request to the HCMUT_SSO service for verification. The HCMUT_SSO system, acting as a centralized authentication authority, validates the requested role against its records. This step ensures that the student is authorized to assume the specified role.

Upon successful role verification, the system prompts the student to input their login credentials, typically consisting of a username and password. These credentials are then transmitted to the HCMUT_SSO service for authentication. The HCMUT_SSO service rigorously scrutinizes the provided credentials against its database to determine their validity.

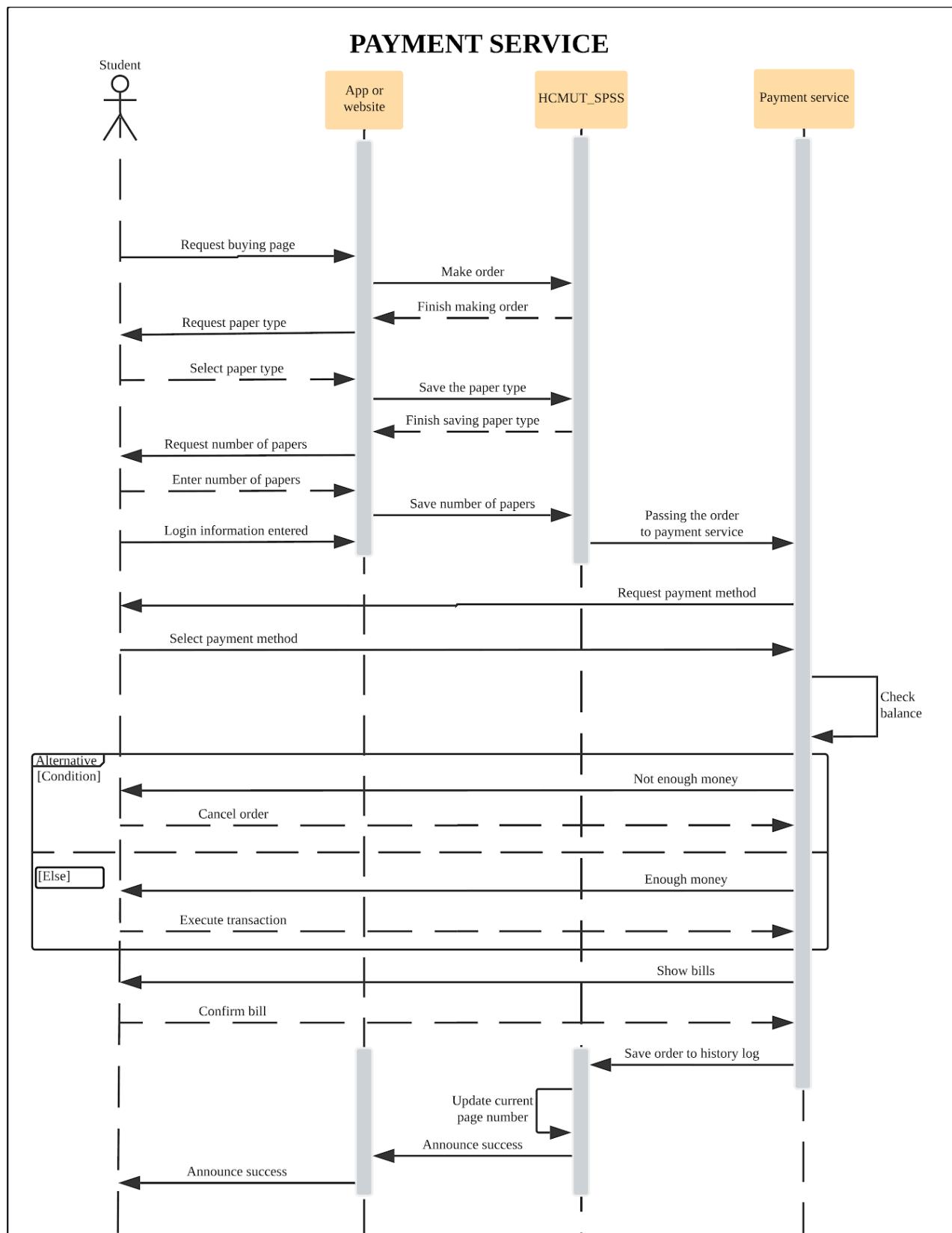
The HCMUT_SSO service returns a verification outcome, which can be either "Invalid Information" or "Valid Information." This situation is determined by HCMUT_SSO authentication.

After successful authentication, the system offers the student an optional step to modify their login information. This feature allows students to update their credentials, such as their password, to enhance security and convenience. If the student chooses to update their information, the changes are transmitted to the HCMUT_SSO service for verification and updating.

Once the login process is completed, either with or without information updates, the system grants the student access to the appropriate functionalities based on their verified role. The student can now navigate the system, perform tasks, and access resources as permitted by their role.

The login process depicted in the sequence diagram is a robust and secure mechanism that relies on the HCMUT_SSO service to validate user identities and authorize access. By following these steps, the system ensures that only authorized individuals can access its resources, safeguarding sensitive information and preventing unauthorized access attempts.

iv. Diagram for Payment



Description

The payment process begins when a student requests a buying page. The system responds by creating a new order and prompts the student to specify the desired paper type and quantity. This information is essential for calculating the total cost of the print job.

Before proceeding with the payment, the system requires the student to enter their login information. This step is crucial for verifying the student's identity and ensuring that the payment is associated with the correct account. Once the student's identity is verified, they are presented with various payment options, such as credit card, debit card, or mobile payment. The student selects their preferred payment method, and the system initiates the payment process.

The selected payment method is transmitted to the payment service, which is responsible for handling the transaction. The payment service performs a balance check to ensure that the student has sufficient funds to cover the cost of the print job. This payment method is handled by a third party such as banks or digital wallet.

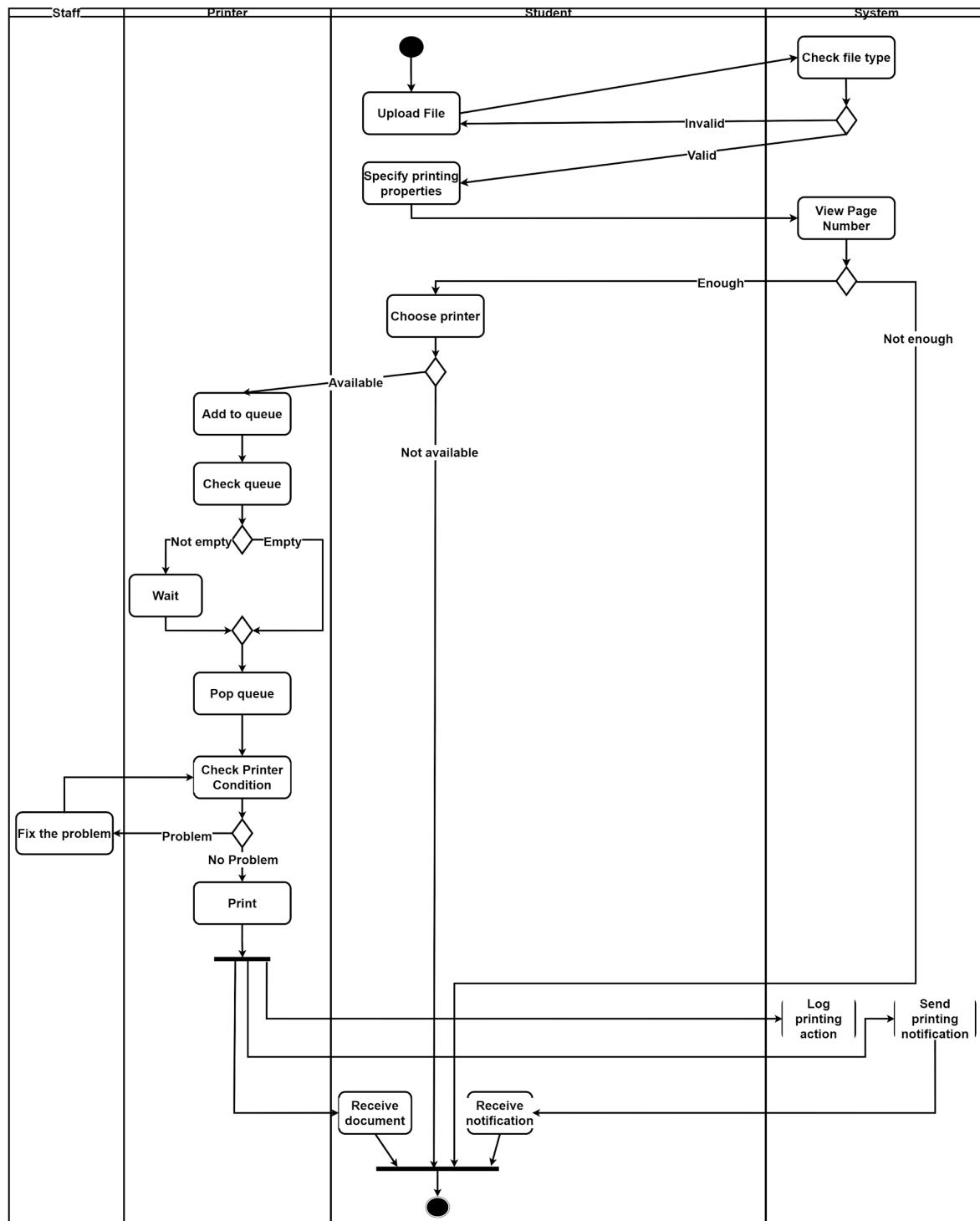
Upon successful payment processing, the system generates a detailed bill for the print job. The bill includes information such as the order ID, student name, paper type, quantity, total cost, and payment method. The system confirms the bill with the student, ensuring that they are aware of the charges and have approved the transaction. To maintain a comprehensive record of all transactions, the system saves the bill to the history log.

Once the payment is confirmed, the system updates the current page number to reflect the newly printed pages. This ensures accurate tracking of page usage and helps in managing resource allocation. Finally, the system sends a success notification to the student, informing them that the payment was successful and the print job has been processed.

The payment process depicted in the sequence diagram is a robust and secure mechanism that ensures accurate billing, efficient payment processing, and proper record-keeping. By following these steps, the system provides a seamless payment experience for students while safeguarding the integrity of financial transactions.

b. Activity diagram

i. Diagram for student



Description

The activity diagram illustrates the process of printing a document within a system, involving interactions between the Staff, Student, and System components.

The process begins when a Student initiates the printing process by uploading a file. The system then checks the file type for compatibility. If the file type is valid, the student is prompted to specify printing properties such as paper size, orientation, and number of copies.

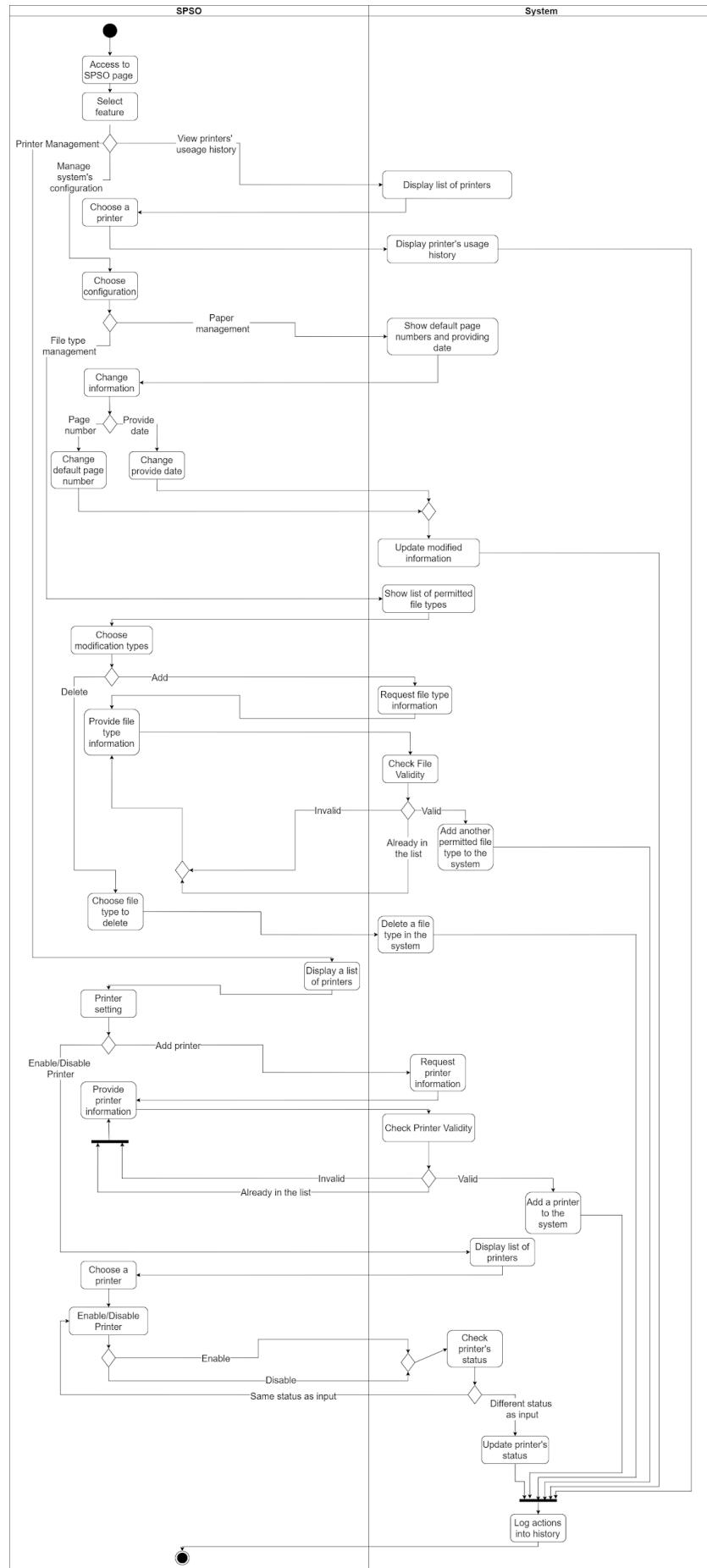
Once the printing properties are defined, the system checks the available page number. If there are enough pages available, the system moves to the printer selection phase. The student chooses a printer from the available options. If the chosen printer is available, the print job is added to the print queue.

If the queue is not empty, the system waits until the previous print job is completed. Once the queue is empty, the system pops the current print job from the queue and checks the printer's condition. If there's a problem with the printer, the system notifies the staff to fix the issue. If the printer is working correctly, the system proceeds with the printing process.

During the printing process, the system logs the printing action and sends a printing notification to the student. Once the printing is complete, the system sends a notification to the student confirming the successful completion of the print job.

This activity diagram provides a clear visual representation of the printing process, highlighting the key steps involved and the potential decision points based on file type, page availability, printer status, and queue status. It also illustrates the interactions between the different system components and the notifications sent to the student at various stages of the process.

ii. Diagram for SPSO



Description

The diagram begins with the user initiating the process by selecting an action. This could be adding a printer, enabling or disabling a printer, or managing print requests. Based on the user's choice, the system proceeds with the corresponding actions.

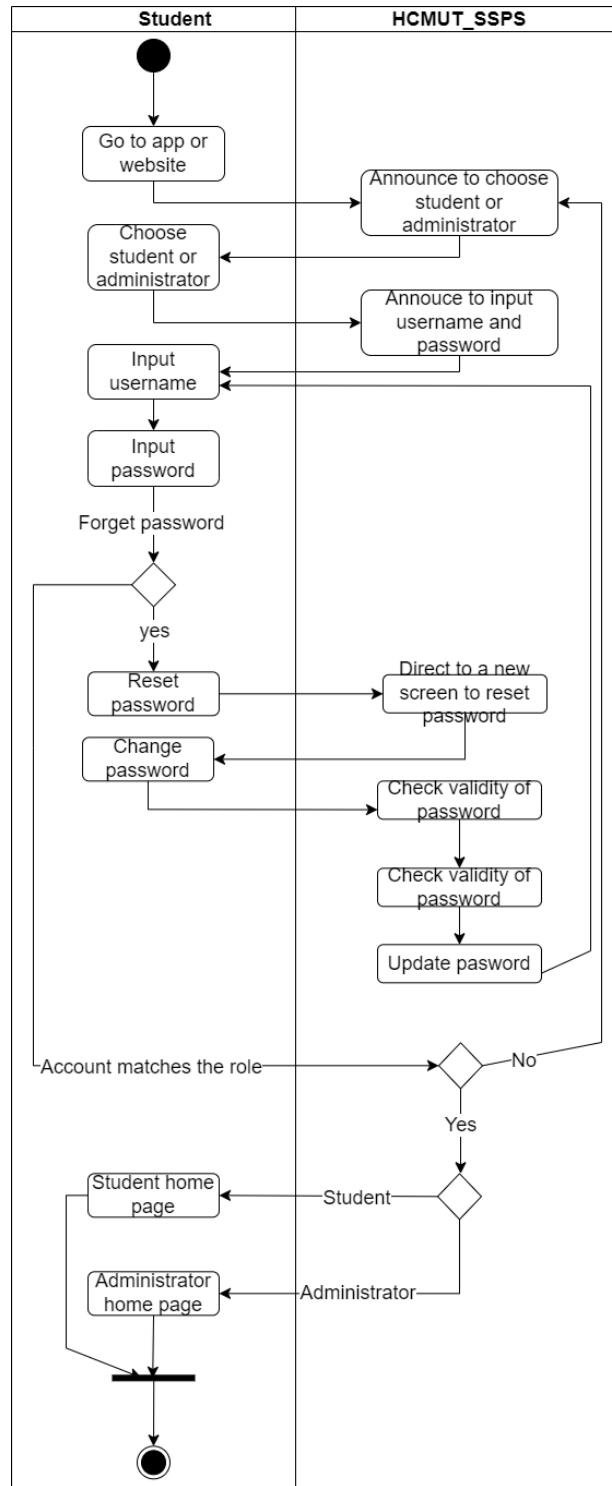
Adding a Printer: If the user chooses to add a printer, the system prompts the user to provide details such as printer ID, brand, model, and location. Once the details are provided, the system validates the information and adds the printer to the system's database.

Enabling/Disabling a Printer: To enable or disable a printer, the user selects the desired action and chooses the printer from the list. The system then updates the printer's status in the database accordingly.

Managing Print Requests: When a user submits a print request, the system checks the file type, page count, and paper type. The system then selects an available printer and adds the print job to the queue. Once the printer is available, the system retrieves the print job from the queue and initiates the printing process.

The diagram also shows the system's ability to generate reports on printer usage and print history. This information can be used to analyze printing trends and optimize resource allocation.

iii. Diagram for login



Description

The login process commences when a student initiates access to the application or website. The system prompts the student to select their role, either as a "student" or "administrator." This initial step is crucial as it determines the level of access and privileges granted to the user.

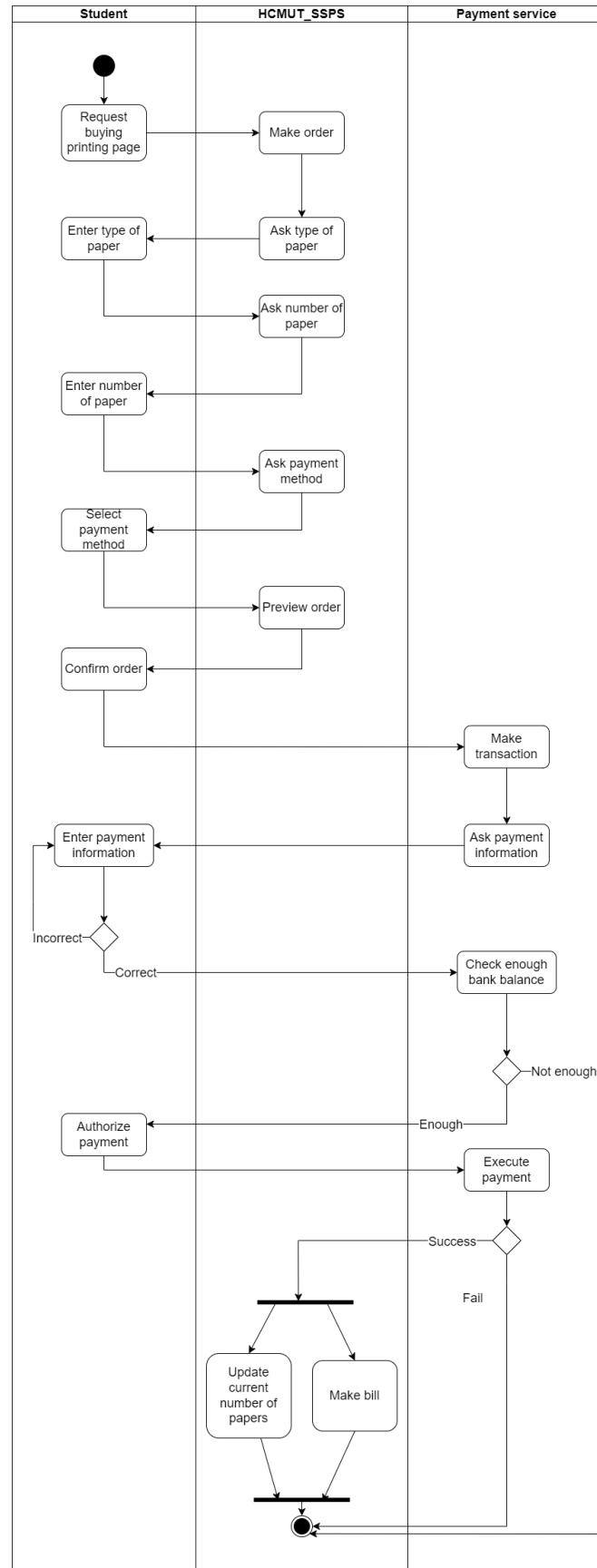
Once the role is selected, the system prompts the student to input their login credentials, typically consisting of a username and password. These credentials are then transmitted to the HCMUT_SSPPS for authentication. The HCMUT_SSPPS scrutinizes the provided credentials against its database to determine their validity.

If the student forgets their password, they can initiate a password reset process. The system directs the student to a new screen where they can provide necessary information to verify their identity. Upon successful verification, the system allows the student to reset their password.

After successful authentication, the system verifies whether the provided account matches the selected role. If the account matches the role, the system grants the student access to the appropriate home page.

- Student Home Page: If the student role is verified, the system directs the student to the student home page, which provides access to student-specific features and functionalities.
- SPSO Home Page: If the SPSO role is verified, the system directs the administrator to the administrator home page, which grants access to administrative tools and controls

iv. Diagram for payment process



Description

The payment process begins when a student requests a printing page. The system prompts the student to specify the desired paper type and quantity. This information is essential for calculating the total cost of the print job.

Once the paper details are confirmed, the system prompts the student to select their preferred payment method. This could include options like credit card, debit card, or mobile payment. The system then guides the student through the necessary steps to provide the required payment information.

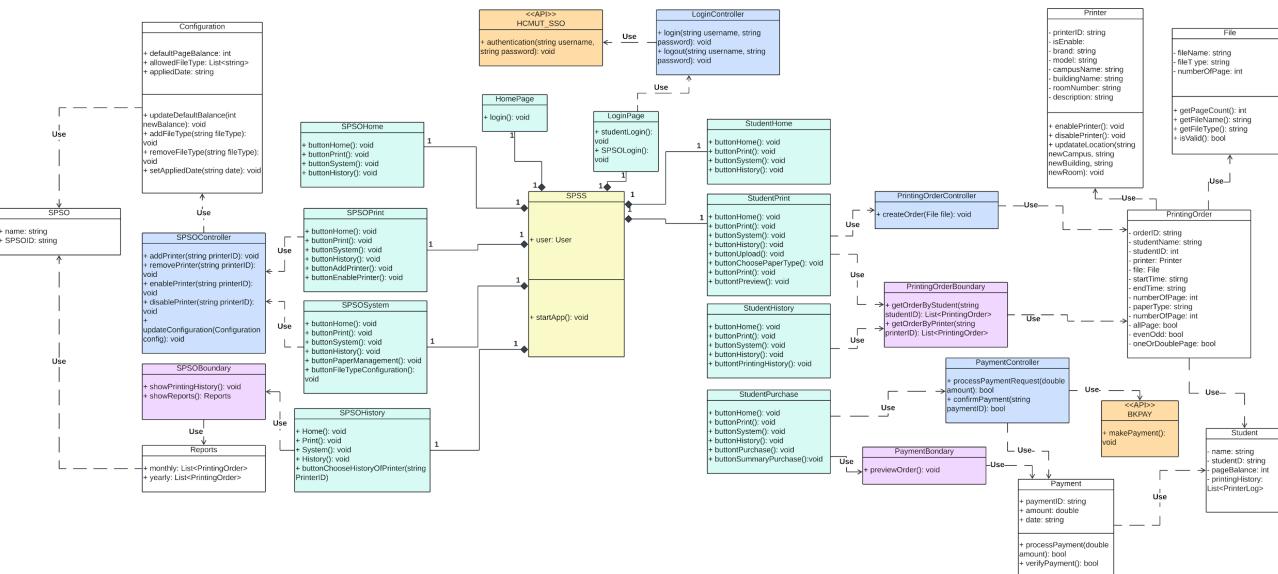
The selected payment method is transmitted to the payment service, which is responsible for handling the transaction. The payment service performs a balance check to ensure that the student has sufficient funds to cover the cost of the print job. Two scenarios may arise at this stage:

1. Insufficient Funds: If the student's balance is insufficient, the payment service rejects the transaction, and the order is canceled. The system notifies the student of the insufficient funds and prompts them to either add more funds to their account or cancel the order.
2. Sufficient Funds: If the student has sufficient funds, the payment service proceeds with the transaction. The payment service securely transmits the necessary payment information to the relevant financial institution and processes the payment.

Upon successful payment processing, the system generates a detailed bill for the print job. The bill includes information such as the order ID, student name, paper type, quantity, total cost, and payment method. The system confirms the bill with the student, ensuring that they are aware of the charges and have approved the transaction. To maintain a comprehensive record of all transactions, the system saves the bill to the history log.

Once the payment is confirmed, the system updates the current page number to reflect the newly printed pages. This ensures accurate tracking of page usage and helps in managing resource allocation. Finally, the system sends a success notification to the student, informing them that the payment was successful and the print job has been processed.

c. Class diagram



Description:

This class diagram represents a sophisticated printing and payment management system, likely tailored for an educational or organizational setting. It organizes classes into distinct layers that cover core functionalities, user management, printer operations, payment processing, and configurations. Each component plays a specific role, and the interactions between these components are mapped out to ensure seamless user experiences and efficient system operations.

User and Authentication Management

The system features a robust user management framework. The User class encapsulates information about each user, such as name, studentID, and pageBalance, along with a printingHistory attribute that holds a list of PrinterLog entries. This structure allows the system to track each user's printing history, maintain their account balance, and potentially restrict access based on their balance or usage. Authentication is managed through the HCMUT_SSO API interface, which provides an authentication method for secure login. The LoginPage and LoginController classes facilitate user login and logout, accommodating different login flows through studentLogin() and SPSOLogin(). This separation of concerns enhances security by restricting unauthorized access and offering different login paths for students and system administrators.

SPSO Components and Printer Management

At the core of the printing management functionality is the SPSO class and its derivatives. The SPSOController acts as the main controller for printer operations, with methods to add, remove, enable, and disable printers, as well as update configurations. The SPSOBoundary class provides a user interface for displaying printing histories and generating reports. This modular approach ensures that printer control is centralized in one component (SPSOController), while the user interface and interaction layers are separated (SPSOBoundary). The Printer class stores detailed information about each printer, including its ID, brand, model, and location, and provides methods

for enabling and disabling the printer as well as updating its location. This encapsulation allows for streamlined printer management, with each printer being easily configurable and manageable by the system's controllers.

Student Functionality and User Interactions

The StudentHome, StudentPrint, StudentHistory, and StudentPurchase classes represent distinct user interfaces designed to facilitate different user activities. Each class has specific button methods, such as buttonHome, buttonPrint, and buttonHistory, allowing users to navigate between sections and interact with the system. For example, the StudentPrint class contains unique methods like buttonUpload, buttonChoosePaperType, and buttonPreview, supporting file uploads, paper type selection, and print previews before the final print order is placed. This design ensures that users can intuitively access and manage their print jobs, review their history, and make payments. By dividing these functionalities into separate classes, the system achieves high modularity and flexibility, allowing easy adjustments to individual components without impacting the entire structure.

Printing Order Management

The printing order functionality is handled by several classes, ensuring detailed tracking and management of print jobs. The PrintingOrderController is responsible for creating new printing orders via the createOrder method, while the PrintingOrderBoundary class allows retrieval of orders by student ID or printer ID. The PrintingOrder class encapsulates the attributes of each print job, including orderID, studentName, studentID, file, numberOfPage, and options for paper type and duplex printing. This detailed breakdown allows the system to track individual orders accurately, ensuring each print job is processed according to the user's specifications. Additionally, the system's order retrieval methods facilitate easy access to historical data, useful for reporting, analytics, and user history tracking.

Payment Processing and External API Integration

The system includes a comprehensive payment processing component with the PaymentController and PaymentBoundary classes. PaymentController is responsible for handling payment requests and confirmations, with methods like processPaymentRequest and confirmPayment. This ensures that each transaction is validated before being recorded in the system. The PaymentBoundary class offers a preview of orders before finalizing payments, helping users understand costs beforehand. Furthermore, the integration with an external API (BKPAY) via the makePayment method allows for secure transactions handled by a trusted third-party payment service. This integration reduces the security risks associated with handling sensitive financial data directly and adds an extra layer of security and reliability to the system's payment handling.

System Configuration and Report Generation

Configuration management is handled by the Configuration class, which stores key settings like defaultPageBalance, allowedFileType, and appliedDate. It includes methods for updating balances, allowed file types, and applied dates, enabling system administrators to configure and update the system as needed. The Reports class allows the system to generate and store monthly and yearly reports based on printing orders, useful for tracking usage trends and managing resources.

effectively. This ability to configure system settings and generate reports gives administrators control over system parameters and access to valuable usage data, enabling informed decision-making.

Analysis of System Design and Modularity

This class diagram illustrates a highly modular design where each component is encapsulated with clear responsibilities, promoting reusability and maintainability. The separation of concerns is evident, with distinct classes for managing user interactions, printing operations, payment processing, and configuration. Furthermore, the use of API interfaces, such as HCMUT_SSO for authentication and BKPAY for payments, allows the system to leverage external services, enhancing its functionality and security without reinventing the wheel. The design also promotes scalability, as new features or components can be added without extensive modifications to existing classes, making it adaptable to future needs.

d. User interface

i. Welcome screen



ii. SPSO view



iii. SPSO Management view

The screenshot shows a web-based application for managing a smart printing service. At the top, there is a blue header bar with the logo of Ho Chi Minh University of Technology (BK TECH) and a 'Log out' button. Below the header is a sidebar on the left containing links for Home, Print, System, and History. The main content area displays four printers, each labeled 'Available' with a vertical bar icon. To the right of the printers is a 'Paper Management' section with 'File Type Configuration' options. A three-line menu icon is located in the top right corner.

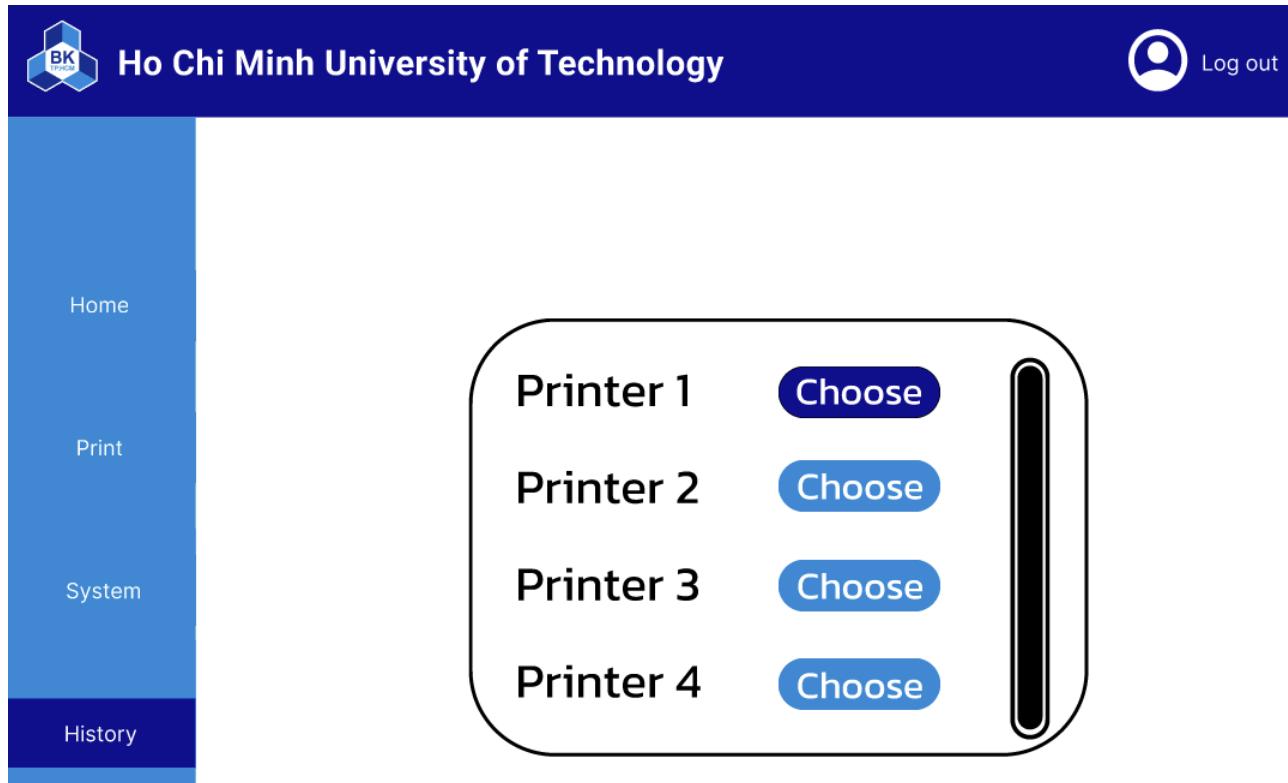
Printer	Status
Printer 1	Available
Printer 2	Available
Printer 3	Available
Printer 4	Available

The screenshot shows a continuation of the smart printing service interface. The top header bar includes the university logo and 'Log out' button. The sidebar on the left has links for Home, Print, System, and History. In the main content area, there is a section titled 'Monthly Paper Provision' with fields for 'Number of papers' set to 5 and a 'Change' button. Below it is a field for 'Next Providing Date' set to 30/10/2024, also with a 'Change' button. A large blue rectangular area is present in the center of the page.

Monthly Paper Provision

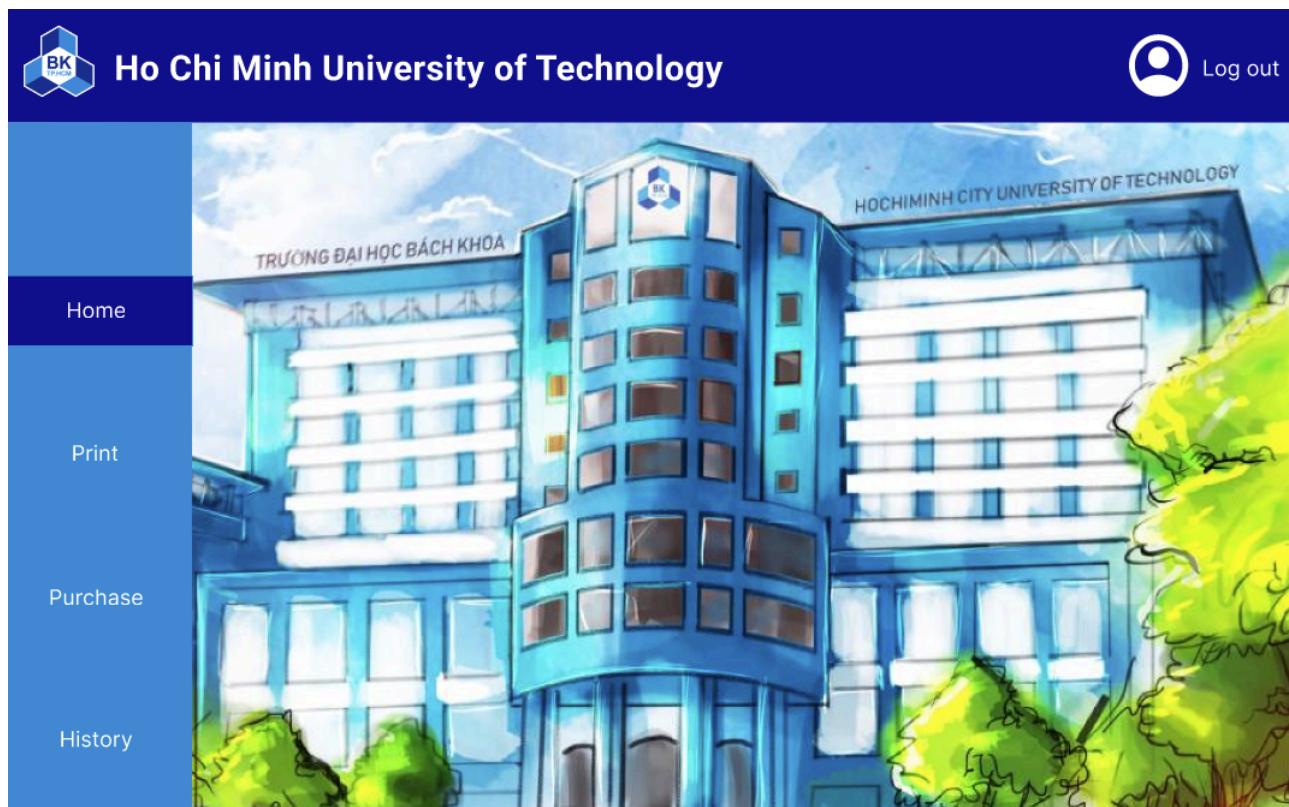
Number of papers:

Next Providing Date:

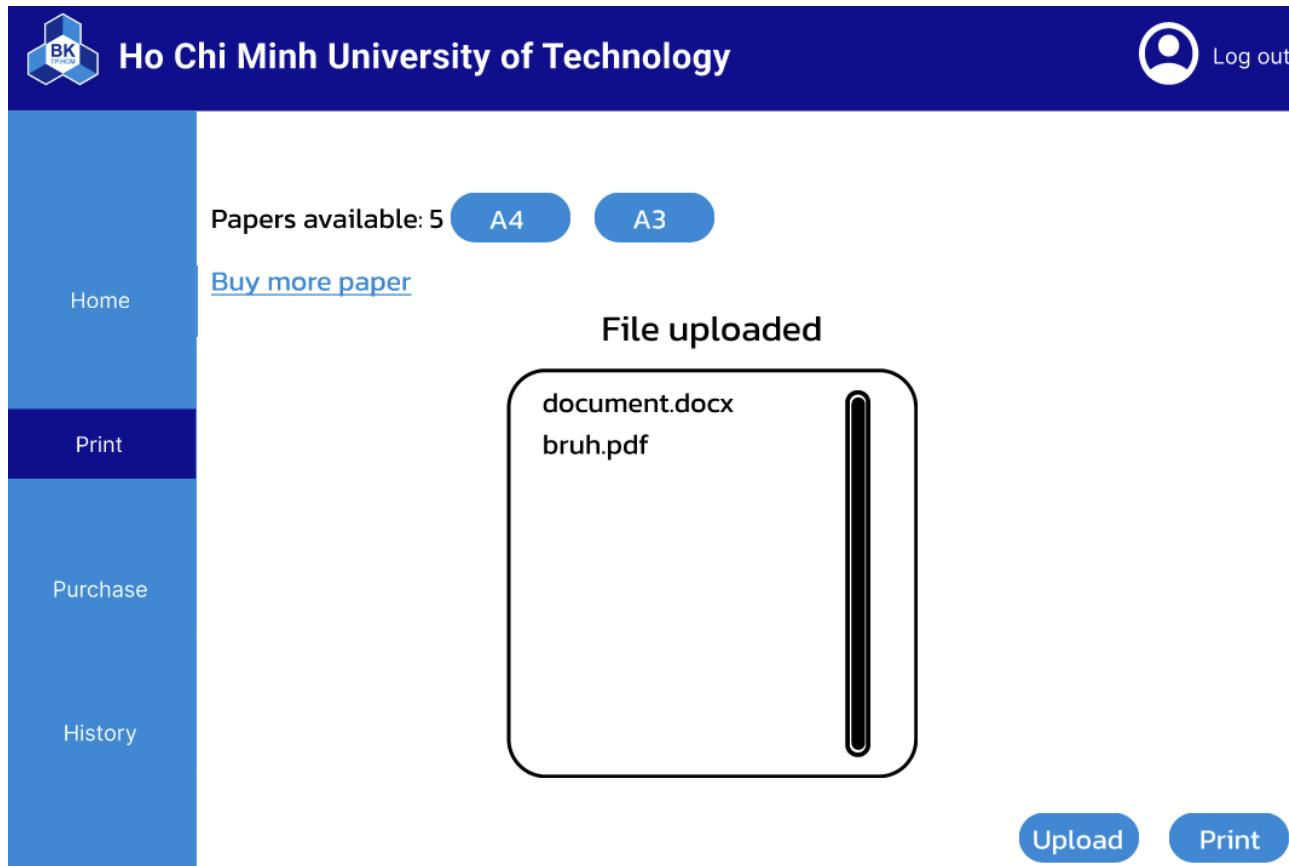


Printer History								
Job ID	User ID	Username	Document Name	Pages	Print Time	Completion Time	Status	
1234	0001	NGUYEN VAN A	document.docx	3	12:00 AM 10/10/2024	12:05 AM 10/10/2024	Finished	
1235	0001	NGUYEN VAN A	lmao.pdf	2	12:30 AM 10/10/2024	12:35 AM 10/10/2024	Finished	
1236	0002	TRAN THI B	bruh.png	1	12:00 AM 11/10/2024	12:05 AM 11/10/2024	Finished	
1237	0003	TRAN VAN C	gyat.jpg	1	12:00 AM 12/10/2024	12:05 AM 12/10/2024	Finished	
1238	0004	DOAN THI D	fanum.docx	4	12:00 AM 13/10/2024	12:05 AM 13/10/2024	Finished	

iv. Student view



v. Student printing view



vi. Student purchasing paper view

The screenshot shows a user interface for purchasing printing services. At the top, there's a blue header bar with the logo of Ho Chi Minh University of Technology (BK TECH) and a 'Log out' button. On the left, a vertical sidebar has buttons for 'Home', 'Print', 'Purchase' (which is highlighted in dark blue), and 'History'. The main content area displays the text 'Pages to buy: 10' and 'Sum amount: 5000 VND'. Below this, two buttons are shown: one for 'Purchase using momo' with the momo logo, and another for 'Purchase using BK Pay' with the BK Pay logo.

Pages to buy: 10

Sum amount: 5000 VND

Purchase using

Purchase using

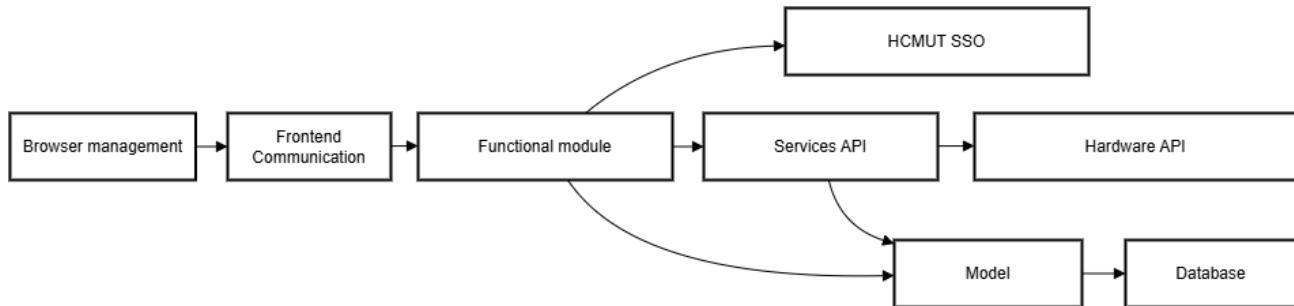
vii. Student history view

The screenshot shows a user interface for viewing print history. At the top, there's a blue header bar with the logo of Ho Chi Minh University of Technology (BK TECH) and a 'Log out' button. On the left, a vertical sidebar has buttons for 'Home', 'Print', 'Purchase', and 'History' (which is highlighted in dark blue). The main content area displays the text 'Print History' above a table. The table lists five completed print jobs with details like document name, pages, print time, completion time, and status.

Document Name	Pages	Print Time	Completion Time	Status
document.docx	3	12:00 AM 10/10/2024	12:05 AM 10/10/2024	Finished
lmao.pdf	2	12:30 AM 10/10/2024	12:35 AM 10/10/2024	Finished
bruh.png	1	12:00 AM 11/10/2024	12:05 AM 11/10/2024	Finished
gyat.jpg	1	12:00 AM 12/10/2024	12:05 AM 12/10/2024	Finished
fanum.docx	4	12:00 AM 13/10/2024	12:05 AM 13/10/2024	Finished

4. Architecture

a. SPSS Architecture diagram



The SPSS architecture was designed to be implemented in layered architecture. There are some ends that need to be considered.

- The user interface, or user endpoints is the browser, backed by the browser management module, which is supported by the frontend communication module.
- The skeleton of the operation is the Functional module. It includes communication with both internal and external modules. This module is responsible for providing high level API.
- Services API responsible for every inner operation. The it does the low level operations, including modeling data, interacting with databases and store data, or communicating with hardwares

i. User interface

To present the User Interface, we will create 2 streamlines for students and SPSO, which is accessible through a web and mobile application. First is the homepage, which our team makes just slightly different in the system and purchase button at the left of the homepage. The students-facing interface when changing to printing will be simple and easy to use as there are only a few main options for students to choose and upload the document they demand to print. The UI then comes to the next step is payment, which we will let them choose 2 options, through MOMO or using BKPay. We also design a history page for students to view their history of printing. Second is SPSO, which will have printer manage and paper manage pages. Both will have some tools to manage printers and also the system of the SSPS for example Printer Management, where they can add, enable, or disable printers and view their locations and statuses, and Paper Management, to monitor and update available printing resources.

ii. Data storage

For the data storage approach of the SSPS, a hybrid system will be implemented to ensure data integrity, scalability, and secure access. We will use a relational database (e.g., MySQL or PostgreSQL) to store structured data such as user profiles, printer details, print job logs, and payment transactions. To manage uploaded documents, we'll integrate a cloud storage solution that stores files securely. The storage system will also include role-based access control to secure data visibility—students can view only their own printing history, while SPSOs have access to system-wide records. This approach balances performance, data security, and ease of maintenance, making the SSPS both reliable and scalable for university-wide use.

iii. Service API

Technically speaking, API (Application Programming Interface) refers to Programming Interface Application, a software intermediary mechanism that enables data transmission and interaction between two distinct applications. There are numerous applications for APIs, such as web-based systems, software libraries, database systems, computer hardware, or operating systems. In basic form

Most significantly, APIs offer an interface that enables requests and responses between two applications. Depending on the type of API—for example, SOAP or REST—the manner in which data is accessed via the API might also differ.

We implement two terms of API, Backend API and Service API. For Service API, it is implemented in the Service API module, which is responsible for all operations with the inner of the system. The Backend API, is Service API come along with utilities API calls or calls for external services, including Authentication services and Transaction services. Backend API module is responsible for all API of the system, being the communication point between Backend and Frontend.