
Software Requirements Specification

for

SMART PRINTING SERVICE FOR STUDENTS AT HCMUT

Version 1.5 approved

Prepared by:

- 1. Phan Thanh Son – 2252718**
- 2. Tran Dang Hien Long – 2252449**
- 3. Tran Hoang Phuc – 2252647**
- 4. Tran Nguyen Anh Khoa – 2252364**
- 5. Tran Thi Hong Hanh – 2252193**
- 6. Tran Thien Phu – 2252623**

Department of Software Engineering

Faculty of Computer Science and Engineering

Ho Chi Minh City University of Technology – VNU-HCM

September, 2024

Table of Contents

Table of Contents.....	ii
Revision History	iii
1. Requirement elicitation	1
1.1 Domain Context	1
1.2 Stakeholders and Needs	1
1.3 Benefits of the System.....	4
1.4 Functional Requirements.....	7
1.5 Non - Functional Requirements.....	11
2. Use-case Diagrams.....	12
2.1 Use-case Diagram for the Whole System	12
2.2 Use-case Diagram for Printing Documents Module	13
2.3 The Details of Use Cases in Printing Documents Module	13
3. System Modelling.....	20
3.1 Activity Diagram.....	20
3.2 Sequence Diagram.....	25
3.3 Class Diagram	31
3.4 User Interface	33
4. Architecture Design	41
4.1 Architectural Diagram	41
4.2 Component Diagram	53
5. Implementation – Sprint 1	55
5.1 Version control.....	55
5.2 Usability Test	55

Revision History

Name	Date	Reason For Changes	Version
Tran Thi Hong Hanh	September 22 nd , 2024	Complete domain context, stakeholders and needs, benefits of each stakeholder of the system, functional requirements and non-functional requirements.	1.0
Tran Hoang Phuc	September 30 th , 2024	Use case diagram for the whole system, use case diagram and the detail of each use case of Printing Documents Module.	1.1
Tran Thi Hong Hanh	October 27 th , 2024	Create activity diagram, sequence diagram, class diagram and user interface using figma for Printing Document Module	1.2
Tran Nguyen Anh Khoa	November 13 th , 2024	Create architectural diagram, component diagram and their descriptions	1.3
Tran Thi Hong Hanh	November 23 th , 2024	Create github link, usability test	1.4
Tran Thi Hong Hanh	December 06 th , 2024	Update domain context, use case description, activity diagram, sequence diagram and their description. Format the document	1.5

1. Requirement elicitation

1.1 Domain Context

The lack of an on-campus printing system has long been a challenge for printing study materials in the Ho Chi Minh City University of Technology (HCMUT). Using off-campus vendors is both inefficient and unpredictable and leads to unnecessary delays, hidden costs, and frustration. Not only are these vendors not integrated with the university's academic ecosystem but they also don't offer sufficient flexibility to accommodate students who have a printing need to obtain materials for their classes, assignments, final project submissions, etc. The burden of having to travel out to and from these vendors is disrupted because of the inconsistencies in pricing and when students are studying.

To address these shortcomings, HCMUT Student Smart Printing Service (SSPS) functionalities was developed to be directly applicable to HCMUT's students and faculty. It supports the possibility of digitally uploading our documents through a deployed platform through a web and a mobile device. Further, this platform seamlessly integrates with HCMUT's existing infrastructure including Single Sign On (SSO) and BKPay for secure access, and easy payment processing. The real-time printer is what sets the SSPS apart, students can see the progress of their print job from page view to full print job and receive a notification when their documents are ready for collection. This means there is no guessing and little disruption to students' schedules.

The SSPS also extends far beyond basic printing to provide students with a highly customizable, location-aware experience. Students have access to a campus-wide network of printers to choose a location nearest to themselves, thereby reducing unnecessary waiting in queues or travel. The system is also readily configurable, with configuration options like paper size, double-sided printing, all without having to calculate costs. Each student gets a page limit (equivalent to A4) per semester, that can be tracked, and refillable. Students are able to place their work where it's needed and get it as soon as possible without having to worry about offline or resource-depleted printers because of the real-time printer status updates on the system.

1.2 Stakeholders and Needs

1.2.1 Student Printing Service Officer (SPSO)

- Description: SPSO is the one who manages and monitors the printing activities in the system
- Needs:

- + Manage all printers across the campus.
- + Configure system-wide settings about files, printing pages, policies.
- + Access logs and reports of all students using printing service.
- + Monitor the status of the printing system.
- + Resolve students' complaints and issues related to printing.
- + Maintenance requirements, refilling paper/ink.

1.2.2 University Administration (HCMUT)

- Description: Manages the overall operation and efficiency of the printing service, ensures student satisfaction, and monitors financial and operational performance.
- Needs:
 - + Keep track and manage the resources and financial efficiency of the service.
 - + Ensure a reliable printing service is provided for students.
 - + Ensure data security and privacy.
 - + Control over system policies, such as page limits, semester-based allocations, and pricing.
 - + Generate reports on overall system usage for budget planning and service improvements.

1.2.3 Students

- Description: Main users of the system's printing services. They are the ones who create printing and download requests, upload documents, select printers, and make payments
- Needs:
 - + Easy access to print documents via both web and mobile platforms, user-friendly and reliable.
 - + Ability to create requests to print documents quickly and can access affordable prices for students.
 - + Ability to upload files, select printer locations, and configure print settings.
 - + View and keep track printing history, page summary and available balance.
 - + Can buy more printing pages through the BK-Pay system.

1.2.4 Student Management System

- Description: This system is responsible for storing and managing student data, such as student ID, name, major, faculty, and status (active, graduated, on leave,...). It provides

this information to the HCMUT_SSPPS to facilitate smooth authentication, tracking, and management of student-related activities within the printing service.

- Needs:
 - + Seamless integration with the HCMUT_SSPPS to ensure accurate and up-to-date student information.
 - + A secure and reliable connection to transfer sensitive student data without compromising privacy.
 - + Regular updates to student records that reflect any changes in student status (graduation, withdrawal, etc.) in real-time.
 - + The ability to log access requests for auditing purposes to ensure data security.

1.2.5 Payment System (BKPay system)

- Description: processes online payments for students buying additional printing pages, ensuring secure and reliable payment transactions.
- Needs:
 - + Secure and process online payments made by students to buy more printing pages.
 - + Ensure the reliability and uptime of the payment platform.

1.2.6 Finance Department

- Description: Manages the financial aspects of the printing service, including tracking payments made by students and generating financial reports on the service's usage.
- Needs:
 - + Integration with the BKPay for student transactions.
 - + Track and manage payments made by students when purchasing additional printing pages.
 - + Generate financial reports related to printing usage and income from printing services.

1.2.7 Guests

- Description: Guests are users who do not log in to the system but can view general information about the HCMUT Student Smart Printing Service
- Needs:
 - + Access to basic information about the printing service, such as available printer locations, operating hours, and pricing details.
 - + Information on how to use the system and any frequently asked questions

- + Security to ensure that while viewing general information, no sensitive data is exposed or accessed.

1.2.8 Development Team

- Description: This team is responsible for maintaining the technical infrastructure of the system, ensuring secure authentication, integration with payment systems, and resolving any technical issues.
- Needs:
 - + Develop the printing system.
 - + Maintain system security and data integrity, including student data, printing logs, and payment details.
 - + Ensure system scalability and reliability for large numbers of students and printers.
 - + Handle any system issues, security threats, or technical problems quickly.

1.2.9 Printer Manufacturers

- Description: Supply and maintain the printers used in the system, ensuring they are compatible and functioning properly, and providing technical support as needed.
- Needs:
 - + Provide, install and maintain the printer across the campus.
 - + Ensure technical support related to the printers and repair when needed.

1.3 Benefits of the System

1.3.1 Student Printing Service Officer (SPSO)

- Streamlined Printer Management: Simplifies the management of printers across campus with centralized controls and monitoring.
- Policy Configuration: Provides flexibility to set printing policies such as page limits, file types.
- Efficient Issue Resolution: Quickly address student complaints and resolve printing issues.
- Access to Reports: Automatically generated monthly and yearly reports for improvement.
- Maintenance Tracking: Helps manage paper and ink refill needs, ensuring minimal downtime for printers.

1.3.2 University Administration (HCMUT)

- Operational Efficiency: Automates student page allocations and usage tracking, allowing better resource management and cost control.
- Data Security: Ensures secure handling of sensitive student data with HCMUT_SSO authentication, preventing unauthorized access.
- Financial Planning: Provides insights through usage reports, which help in budgeting and resource allocation for the printing service.
- Service Reliability: Increases student satisfaction by maintaining a stable, reliable printing service across campuses.

1.3.3 Students

- Convenience and Accessibility: Easy access to printing services via both web and mobile platforms, offering flexible usage from anywhere on campus.
- User-Friendly Interface: Allows students to upload files, select printers and have personalized settings.
- Flexible Payment Choices: The ability to purchase additional pages through BKPay ensures affordability and flexibility in managing printing needs.
- History Tracking: Students can view their printing logs, track usage, and monitor their available balance, promoting accountability.

1.3.4 Student Management System

- Seamless Integration: Results in up-to-date student data, facilitating accurate and smooth authentication and tracking within the HCMUT_SSNS.
- Data Privacy: The secure transfer of sensitive information protects student data, following privacy guidelines.
- Real-Time Updates: Reflects any changes in student status (e.g., graduation, leave) instantly, resulting in system accuracy.
- Audit Logs: Logs access requests for auditing purposes, ensuring data security and regulatory compliance.

1.3.5 Payment System (BKPay System)

- Secure Payments: Facilitates secure, real-time online payments for students purchasing additional printing pages.
- Reliability: Ensures high availability and uptime, which is crucial for uninterrupted transactions and student convenience.

- Integration: Streamlined with the HCMUT-SSPS to provide seamless payment experiences.

1.3.6 Finance Department

- Transaction Tracking: Tracks payments made by students for additional printing pages, providing accurate financial records.
- Report Generation: Generates financial reports on the system's usage and income, aiding in financial planning and analysis.
- BKPay Integration: Facilitates easy integration with the BKPay system for reliable tracking and management of financial data related to printing services.

1.3.7 Guests

- Access to Information: Provides basic information like printer locations, pricing, and operational hours, ensuring transparency.
- Guides and FAQs: Offers helpful resources on how to use the system and common queries, making it easier for newcomers to understand.
- Security: Results in no sensitive data is accessible while browsing general information, protecting the integrity of the system.

1.3.8 Development Team

- Streamlined System Development: Enables efficient development, updates, and customization of the printing system, reducing development time and improving system functionality.
- Security and Data Integrity: Ensures the protection of student data, payment details, and printing logs through secure authentication (HCMUT_SSO) and data management.
- Scalable and Reliable Infrastructure: Provides a scalable system that can support a large number of students and printers, reducing the need for major overhauls and ensuring long-term stability.
- Proactive Issue Resolution: Enables faster detection and resolution of system issues or security threats, minimizing downtime and improving overall system reliability.
- Seamless System Integration: Ensures smooth integration with third-party systems like BKPay and the Student Management System, simplifying technical maintenance and improving user experience.

1.3.9 Printer Manufacturers

- Maintenance and Support: Enables manufacturers to maintain and provide technical support for printers effectively, ensuring ongoing service quality.
- System Compatibility: Ensures that printers are fully compatible with the system, leading to a smoother operation and fewer issues.
- Repair Services: Streamlines the process for identifying and addressing technical problems, resulting in quick turnaround times for repairs.

1.4 Functional Requirements

1.4.1 Student Printing Service Officers (SPSOs)

- SPSOs can access and view student account information by using the system's administrative interface.
- SPSOs can view the status and information of all printers in the system by accessing the printer management dashboard.
- SPSOs can add, activate, or deactivate printers by configuring the system's printer settings.
- SPSOs can manage system configurations, including adjusting the default number of pages allocated to students and selecting file formats for upload through the system's settings panel.
- The system must generate monthly and yearly statistical reports, which only SPSOs can access through the reporting function.

1.4.2 University Administration (HCMUT)

- The university administration can access analytics on printing resource usage through a reporting dashboard for budget planning and resource allocation.
- The system must enforce user authentication through HCMUT_SSO by integrating the university's Single Sign-On (SSO) system.
- The university administration can set policies regarding print quotas and pricing for students by accessing the system's policy configuration tools.
- The system must implement data security measures to ensure the protection of user information and printing logs by using encryption protocols.
- The system should facilitate communication between students and SPSOs by enabling a feedback mechanism within the platform.

1.4.3 Students

- Students can access and view personal information by logging into the system through their account.
- Students can select a file from their device and upload it to the system by using the file upload feature.
- Students can select a specific printer based on their desired location by using the printer selection tool within the system.
- Students can choose a time to receive their print jobs by scheduling through the system's time management feature.
- The system must notify students via email when their print job is completed by sending an automated email.
- Students can adjust printing settings (e.g., page size, number of copies, single or double-sided) by configuring options within the system.
- The system must store all print history, including student ID, printer ID, start and end times, and page counts by logging each print activity.
- Students can access and review their previous print jobs by using the system's history feature and filtering based on time or printer location.
- The system must allocate a specific number of printing pages to each student at the beginning of each semester.
- Students can purchase additional printing pages through the integrated electronic payment system (BKPay).

1.4.4 Student Management System

- The student management system can provide real-time access to student data for authentication purposes by integrating with HCMUT-SSPS.
- The student management system must support secure data transfer for student information by using encrypted communication protocols.
- The student management system should automatically update student records (e.g., graduation or withdrawal) by synchronizing data with HCMUT-SSPS.
- The student management system must log access requests by maintaining detailed logs of who accessed student data and when for auditing.
- The student management system must allow efficient retrieval of student information by HCMUT-SSPS for operational tasks through a secure API.

1.4.5 Payment system (BKPay)

- The payment system can securely process payments for students purchasing additional printing pages by using the integrated payment gateway.

- The payment system must ensure uninterrupted transaction processing by maintaining high availability and reliability.
- The payment system should provide instant confirmation of successful transactions by sending notifications through the integrated user interface.
- The payment system must support refunds and transaction history retrieval by enabling access to payment logs within the system.
- The system must send email receipts to students after successful payments through the automated email function.

1.4.6 Finance Department

- The finance department can access detailed reports on student payments for printing services by generating financial reports from the system.
- The finance department can track and manage all financial transactions related to the printing service by accessing system transaction logs.
- The finance department should generate financial summaries for budget planning by using the system's financial reporting tools.
- The finance department must ensure compliance with financial regulations by auditing student transactions through the system.
- The finance department must provide financial oversight by conducting regular audits of the printing service's operations through detailed reports.

1.4.7 Guest

- The system provides a homepage containing general information for guests by displaying system details.
- Guests can view information about the system by accessing the homepage.
- Guests can review key features of the system by navigating through the homepage.
- Guests can view usage instructions by accessing the dedicated help section on the system.
- Guests can contact support by using the contact information provided on the system.
- Guests can log in to the system by following the authentication process to access services.
- The system must assign roles (students, SPSO, staff) upon successful login through the authorization module.

1.4.8 Development Team

- The development team can deploy new system features and updates by using a continuous integration/continuous deployment (CI/CD) pipeline.

- The system must allow the development team to monitor system performance and detect errors in real-time by using integrated logging and monitoring tools.
- The maintenance team can schedule system maintenance windows by configuring downtime periods within the system's maintenance module.
- The development team must implement security patches and updates by following secure coding standards and best practices.
- The system must allow the maintenance team to back up and restore data by using an automated backup solution.
- The development team can track and resolve bugs by using an integrated issue-tracking system linked to the development environment.
- The system must notify the development and maintenance team of critical system issues through automated alerts via email or messaging platforms.
- The development team must document all system changes and updates by maintaining version control and update logs.
- The maintenance team can roll back system updates or changes by using the version control system when issues arise.
- The system must facilitate communication between development, maintenance, and other stakeholders by providing a ticketing and communication platform (BKSI) for issue reporting.

1.4.9 Printer Manufacturers

- Printer manufacturers must provide installation and setup support for all printers deployed on campus by dispatching technical teams as needed.
- Printer manufacturers must ensure all printers are compatible with HCMUT-SSPS by providing technical documentation and system integration support.
- Printer manufacturers must supply technical specifications for each printer by including documentation within the system for reference.
- Printer manufacturers must offer ongoing technical support and maintenance services by providing a dedicated support team for troubleshooting and repairs.
- Printer manufacturers must supply updates or patches for printer firmware by releasing them, when necessary, through the system's update management tool.
- Printer manufacturers must ensure that parts and supplies are readily available for maintenance and repairs by maintaining stock at local service centers.

1.5 Non - Functional Requirements

1.5.1 Performance Requirements

- Response Time: System should process requests within 2 seconds under normal load.
- Scalability: Support up to 10 000 concurrent users without performance degradation.
- Throughput: Handle up to 100 print job submissions per minute.
- System Load: Maintain optimal performance for up to 100 000 active users daily.

1.5.2 Safety Requirements

- Data Backup: Automated daily backups to ensure data safety.
- Data Recovery: Restore lost data within 30 minutes in case of failure.
- Failure Handling: Ensure safe shutdown to prevent data loss during system failures.

1.5.3 Security Requirements

- Authentication: Integrate with HCMUT's Single Sign-On (SSO) for secure user authentication.
- Data Encryption: Use SSL/TLS for data transmission and AES-256 for data storage.
- Access Control: Implement role-based access control (RBAC) for system functionalities.
- Logging: Log access and data changes for auditing, retaining logs for 2 years.

1.5.4 Software Quality Attributes

- Usability: Provide a user-friendly interface requiring minimal training.
- Reliability: Ensure 99.9% uptime to minimize service disruptions.
- Maintainability: System updates and patches should be easy to deploy with minimal service impact.
- Compatibility: Ensure cross-browser support (Chrome, Firefox, Safari) and mobile compatibility.

1.5.5 Business Rules

- Print Quotas: Students receive 100 pages per semester, with additional pages available for purchase.
- Refunds: System failures will trigger automatic refunds or equivalent solutions.

2. Use-case Diagrams

2.1 Use-case Diagram for the Whole System

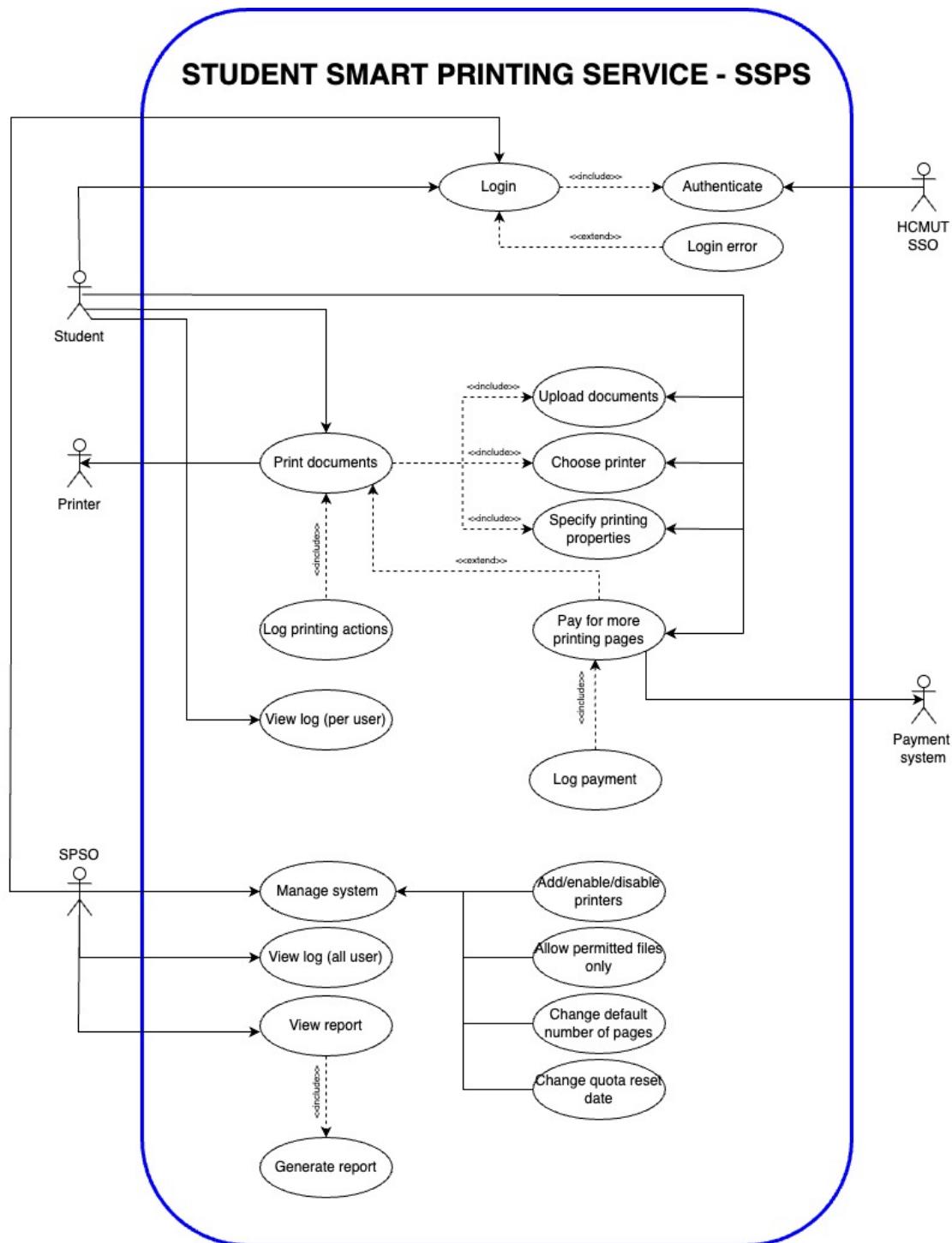


Figure 1: Use Case Diagram for The Whole System

2.2 Use-case Diagram for Printing Documents Module

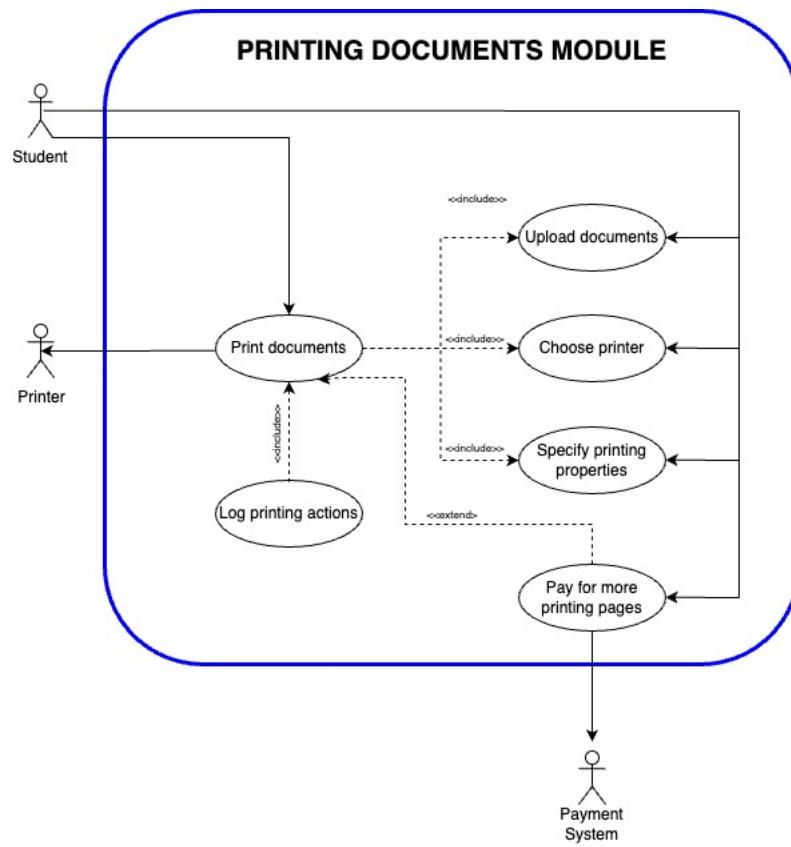


Figure 2: Use Case Diagram for Printing Documents Module

2.3 The Details of Use Cases in Printing Documents Module

2.3.1 Use case Print Document

ID and Name	UC-1 Print Document
Created By	Tran Thi Hong Hanh
Date Created	September 28th, 2024
Primary Actor	Student
Secondary Actors	Printer
Description	The student prints documents uploaded into the HCMUT_SPSO system.

Trigger	The student wants to print the document.
Preconditions	PRE-1: The device has to be connected to the internet. PRE-2: The student has to log in to the system and be authenticated by HCMUT_SSO. PRE-3: The student has enough available pages to print. PRE-4: At least one printer is available.
Postconditions	POST-1: The document is printed successfully. POST-2: The student's page balance is updated. POST-3: The printing activity is logged in the system history.
Normal Flow	1. The student clicks the Print button to start the printing process. 2. The system receives the print request and sends it to the selected printer. 3. The system displays the printing status on the interface (e.g., "Printing in progress"). 4. The system updates and shows the printing progress in real-time. 5. Once printing is complete, the system logs the printing activity. 6. The History page updates and displays the completed printing record.
Alternative Flow	None
Exceptions	E2a. Invalid file type or exceeds size limit. E2b. No printer available. E2c. Insufficient page balance. E5. Printer malfunction (e.g., jam, no paper)

Table 1: Detail of Use Case Print Document

2.3.2 Use case Upload Documents

ID and Name	UC-2 Upload Documents
Created By	Phan Thanh Son
Date Created	September 28th, 2024
Primary Actor	Student
Secondary Actors	None

Description	The student uploads a document to the HCMUT_SPSO system. The system validates the document, and if successful, the document becomes available for printing.
Trigger	The student clicks to the “In tài liệu” button to start.
Preconditions	PRE-1: The device is connected to the internet. PRE-2: The student is logged in and authenticated by HCMUT_SSO.
Postconditions	POST-1: The document is successfully uploaded and stored. POST-2: The student can preview the uploaded document. POST-3: The document is available for printing.
Normal Flow	<ol style="list-style-type: none"> 1. The student selects a document from their device and initiates the upload. 2. The system checks the document's file type and size for validity. 3. If the document passes validation, the system confirms the successful upload. 4. The system opens the uploaded document for preview.
Alternative Flow	None
Exceptions	E2a: Unsupported file format. E2b: File size exceeded.

Table 2: Detail of Use Case Upload Documents

2.3.3 Use case Specify Printing Properties

ID and Name	UC-3 Specify Printing Properties
Created By	Tran Hoang Phuc
Date Created	September 28th, 2024
Primary Actor	Student
Secondary Actors	None
Description	The student specifies their preferred printing properties, such as paper size, page range, single/double-sided printing, and number of copies, to configure the document for printing.

Trigger	After uploading document, user clicks to the button “Tiếp tục” to specify properties.
Preconditions	PRE-1: The student is logged in and authenticated by HCMUT_SSO. PRE-2: The student has a valid document selected for printing.
Postconditions	POST-1: The document is configured with the specified properties. POST-2: The configured properties are displayed for review.
Normal Flow	1. The student selects and configures the desired printing properties (e.g., paper size, page range, copies). 2. The system verifies that the student has sufficient page balance. 3. If the balance is sufficient, the system confirms and saves the printing settings.
Alternative Flow	None
Exceptions	E2. Insufficient page balance.

Table 3: Detail of Use Case Identify Printing Properties

2.3.4 Use case Choose Printer

ID and Name	UC-4 Choose Printer
Created By	Tran Dang Hien Long
Date Created	September 28th, 2024
Primary Actor	Student
Secondary Actors	None
Description	The student selects a printer from the list of available printers to proceed with printing a document.
Trigger	The student clicks the Choose Printer button to select a printer for the printing job.
Preconditions	PRE-1: The student is logged in and authenticated by HCMUT_SSO. PRE-2: The student has chosen a valid document for printing. PRE-3: The student has specified the desired printing properties.
Postconditions	POST-1: The selected printer is assigned to the printing job.

	POST-2: The printer details are saved and linked to the printing job in the system.
Normal Flow	<ol style="list-style-type: none"> 1. The system displays a list of available printers. 2. The student selects the desired printer from the list. 3. The student confirms the printer selection. 3. The system validates and confirms the selected printer.
Alternative Flow	None
Exceptions	E1. No printer is available.

Table 4: Detail of Use Case Choose Printer

2.3.5 Use case Pay More Printing Pages

ID and Name	UC-5 Pay More Printing Pages
Created By	Tran Nguyen Anh Khoa
Date Created	September 29th, 2024
Primary Actor	Student
Secondary Actors	Online Payment System
Description	The student purchases additional printing quota when their current allocation is insufficient.
Trigger	The student clicks the Buy More Pages button to purchase additional printing pages or the student's account does not have enough pages to process the printing request.
Preconditions	<p>PRE-1: The student is logged in and authenticated by HCMUT_SSO.</p> <p>PRE-2: The student's online payment system account is active and linked.</p> <p>PRE-3: The student has sufficient funds in the payment account.</p> <p>PRE-4: The student has an existing page balance allocation.</p> <p>PRE-5: The current printing quota is insufficient.</p>
Postconditions	<p>POST-1: The student's printing quota is increased by the purchased pages.</p> <p>POST-2: The transaction details are recorded in the system.</p> <p>POST-3: A payment confirmation is sent to the student.</p>

	<p>POST-4: The updated page balance is displayed in the student's account.</p> <p>POST-5: The payment history is updated and accessible to the student.</p>
Normal Flow	<ol style="list-style-type: none"> 1. The student specifies the number of additional pages to purchase (between 50 and 2000). 2. The system validates the selected number of pages. 3. The system processes the payment request through the linked payment system. 4. The system sends a payment confirmation to the student. 5. The system displays payment details and confirms a successful transaction. 6. The student's page balance is updated.
Alternative Flow	None
Exceptions	<p>E2. Invalid number of pages.</p> <p>E4. Payment confirmation not received.</p>

Table 5: Detail of Use Case Pay For More Printing Pages

2.3.6 Use case Log Printing Actions

ID and Name	UC-6 Log Printing Actions
Created By	Tran Thien Phu
Date Created	September 28th, 2024
Primary Actor	Student
Secondary Actors	None
Description	After successfully completing a print job, the system logs details about the printing action. This includes information such as the document name, print properties, printer used, and timestamp.
Trigger	The system logs the printing action automatically once the print job is successfully completed.
Preconditions	PRE-1: The student is logged in and authenticated via HCMUT_SSO.

	<p>PRE-2: The document has been successfully uploaded, and print properties are specified.</p> <p>PRE-3: The selected printer is online and ready to accept print jobs.</p> <p>PRE-4: The system has available storage to log print actions.</p>
Postconditions	<p>POST-1: The print action is recorded in the system log.</p> <p>POST-2: Details such as student ID, document name, printer ID, and timestamp are saved for future reference.</p> <p>POST-3: The print log is accessible to both the student and the HCMUT_SPSO for review.</p>
Normal Flow	<ol style="list-style-type: none"> 1. The system captures the details of the completed print job (e.g., student ID, document name, printer ID, and print properties). 2. The system logs the print action with a timestamp. 3. The system updates the student's print history, making it available for viewing.
Alternative Flow	None
Exceptions	E2. Insufficient log storage.

Table 6: Detail of Use Case Log Printing Actions

3. System Modelling

3.1 Activity Diagram

3.1.1 Upload Document

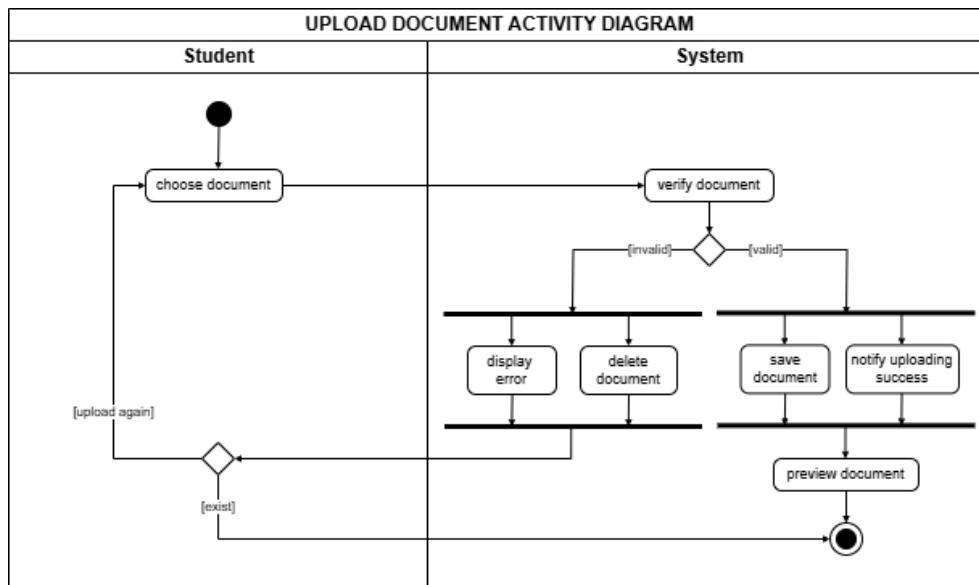


Figure 3: Activity Diagram of Upload Document Use Case

The activity diagram outlines the process of uploading a document by a student through a system. It begins with the student selecting a document to upload. Once chosen, the system verifies the document's validity. If the document is invalid, an error is displayed, and the document is deleted, prompting the student to upload a new document. If the document is valid, the system proceeds to save the document and notifies the student of the successful upload. Following the successful upload, the system offers the student a preview of the uploaded document. The process completes once the document is either successfully uploaded and saved, or an error occurs and the document must be re-uploaded.

3.1.2 Specify Printing Properties

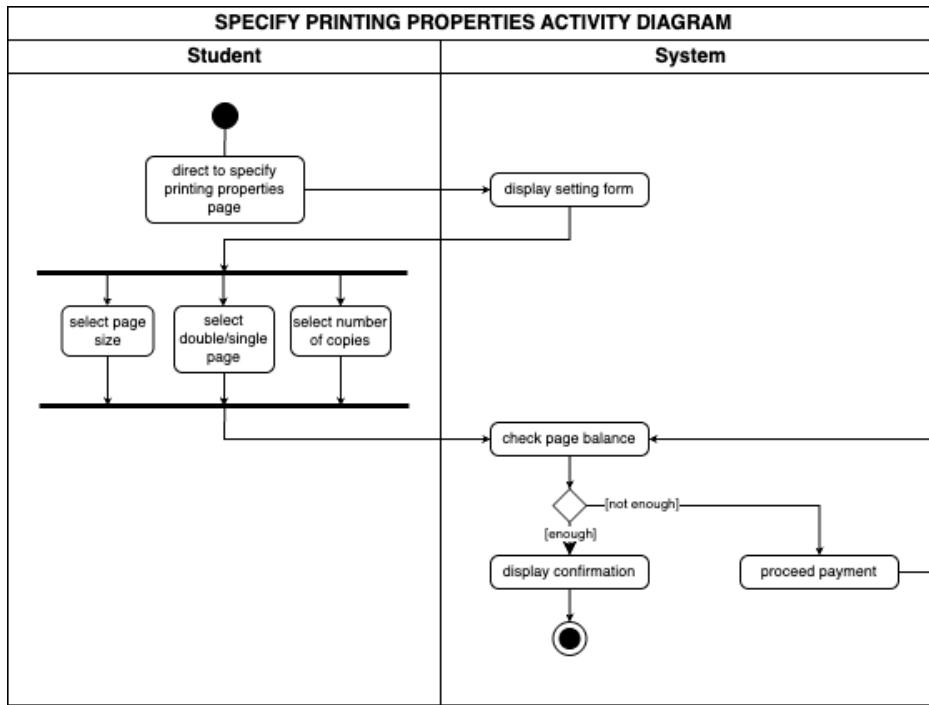


Figure 4: Activity Diagram of Specify Printing Properties Use Case

The activity diagram illustrates the process of specifying printing properties for a student using a system. The activity begins with the student being directed to a page where they can specify printing options. The system displays a form that allows the student to input these settings. The student then chooses from various printing settings, including page size, double or single-sided printing, and the number of copies. Once the selections are made, the system checks the student's page balance to determine enough pages to complete the printing request. If the balance is insufficient, the system prompts the student to proceed with payment. After payment is made, the system checks page balance again to guarantee that the student has enough pages to print. If the balance is sufficient, a confirmation is displayed to the student and the process is completed.

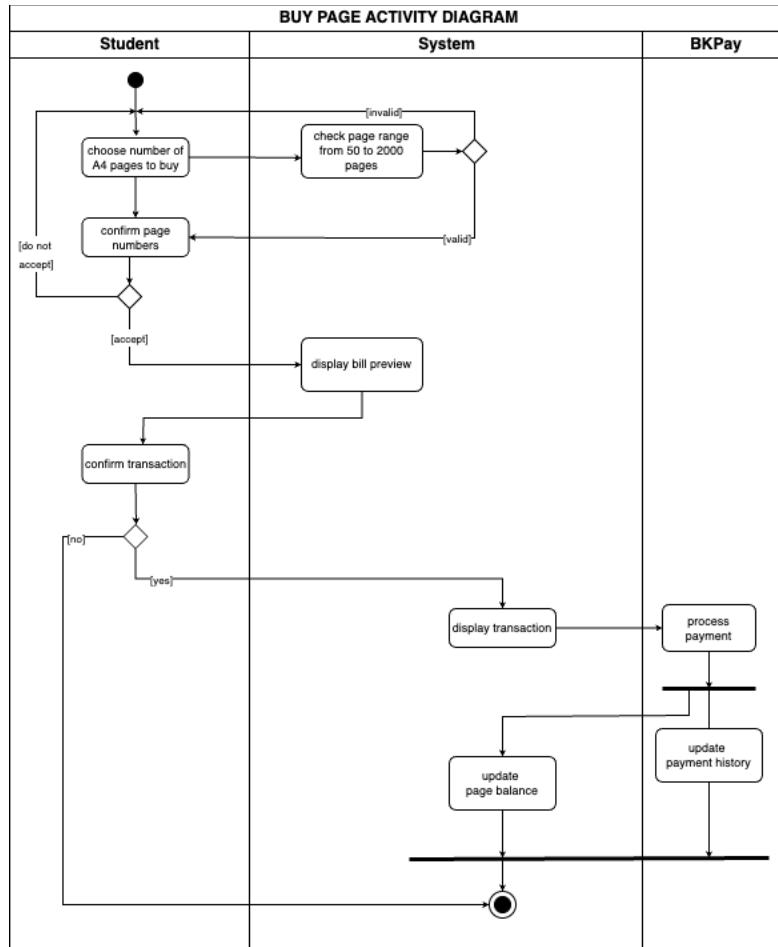


Figure 5: Activity Diagram of Buy Pages Use Case

The activity diagram represents the process of a student purchasing pages through a system that interfaces with a payment system. The process starts with the student choosing number of A4 pages to buy. The system then checks the validity of the page range. If the number of chosen pages is between 50 and 2000, it continues to ask for the confirmation. If the number is out of range, the system will ask the student to choose again. If the student accepts the page numbers, the bill preview will be displayed and the system asks for transaction confirmation. If the student agrees, the system will display transaction and navigate to BKPay system to process payment. Once the transaction is successful, the system updates the student's page balance, reflecting the newly purchased credits and also updates payment history. The activity ends after the balance and the payment history are updated.

3.1.3 Choose Printer

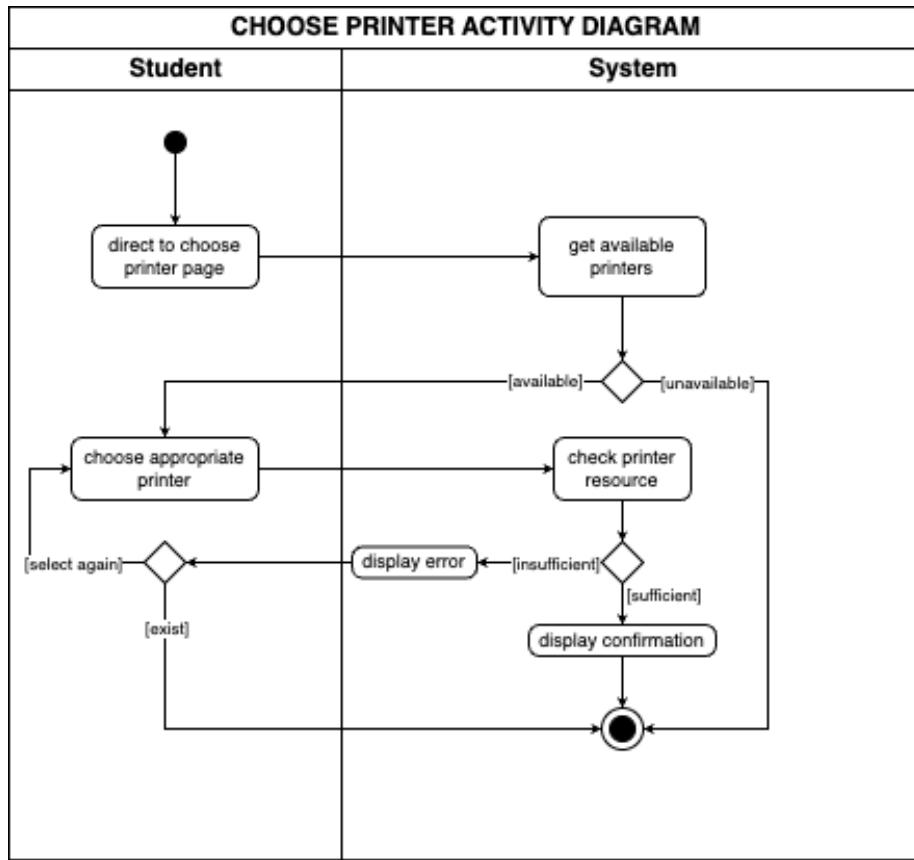


Figure 6: Activity Diagram of Choose Printer Use Case

The activity diagram outlines the process a student follows to choose an appropriate printer for their printing needs. The student is directed to the "Choose printer" page, where the system displays all available printers. The system checks the printer's availability, and if the printer is unavailable, the system will terminate the choice. Once an available printer is selected, the system checks the printer's resources to ensure it has sufficient capacity to handle the printing job. If the printer's resources are insufficient, an error is displayed, prompting the student to select again. However, if the printer has enough resources, a confirmation message is displayed, marking the completion of the process.

3.1.4 Print Document

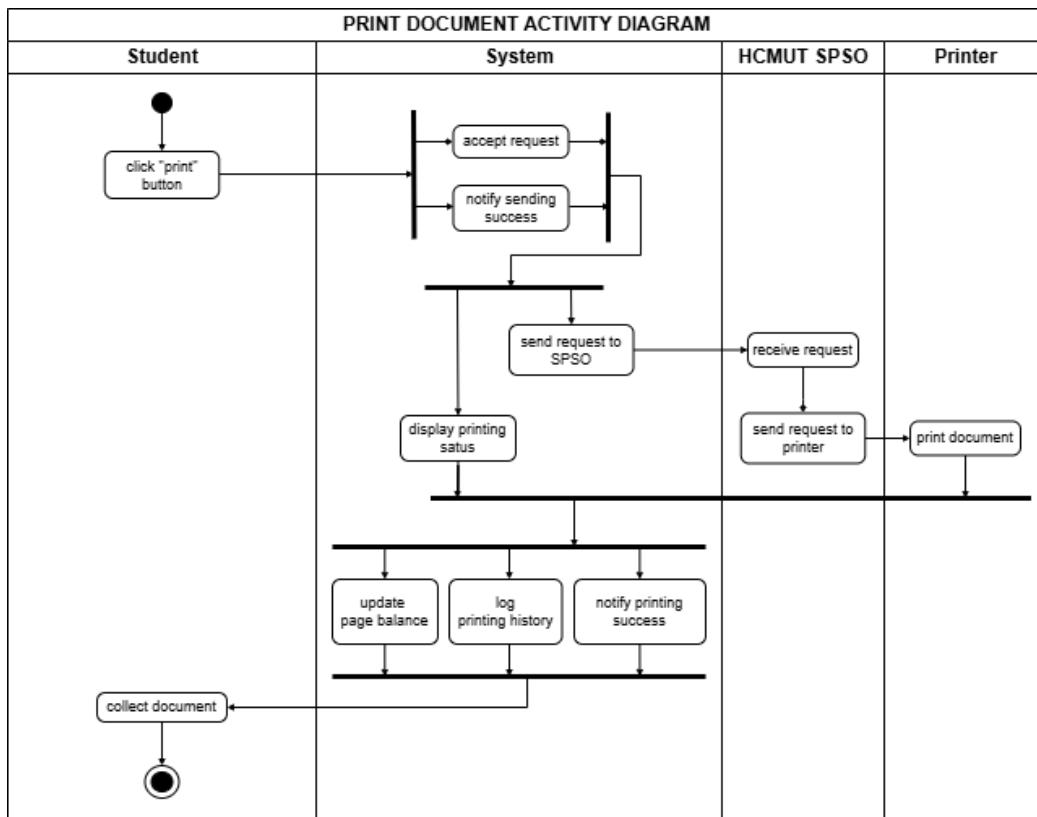


Figure 7: Activity Diagram of Print Document Use Case

The activity diagram outlines the process of printing a document initiated by a student. The student begins by clicking the "print" button, which sends a request to the system. The system accepts the request and notifies the student that the request was successfully sent. The system then forwards the request to the HCMUT SPSO, which receives the request and passes it to the designated printer. Once the printer receives the request, it proceeds to print the document. The system concurrently displays the printing status at regular intervals, allowing the student to track the progress of their print job. After printing successfully, the system updates the page balance for the student, logs the printing history, and notifies the student of the successful printing. The final step involves the student collecting the printed document.

3.2 Sequence Diagram

3.2.1 Upload Document

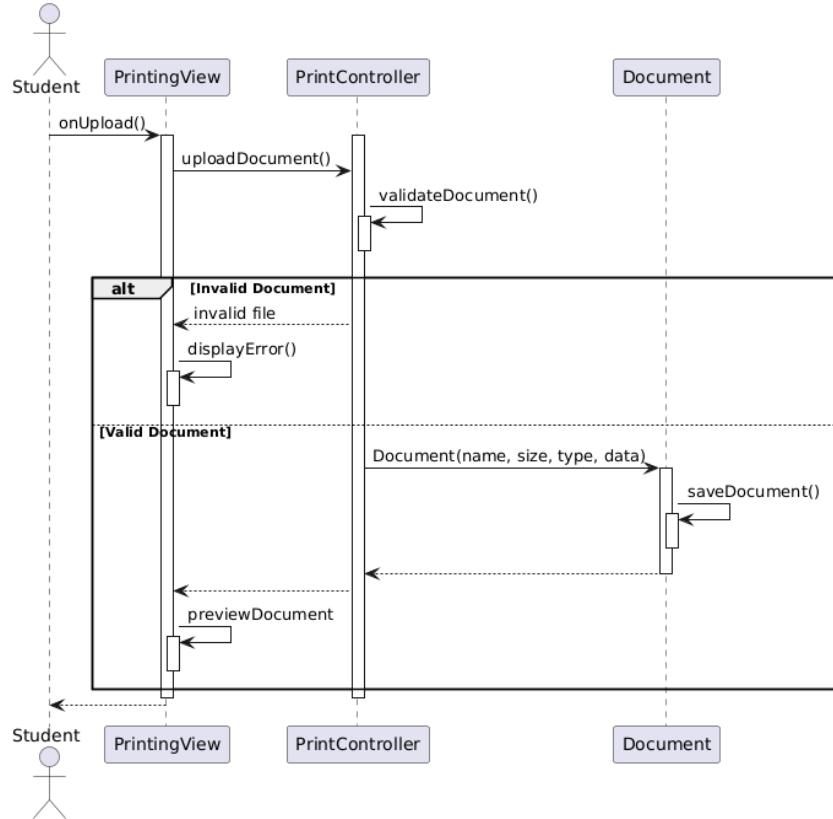


Figure 8: Sequence Diagram of Upload Document Use Case

The student triggers the upload action by interacting with the PrintingView via the `onUpload()` method. The PrintingView sends an `uploadDocument()` message to the PrintController, initiating the document upload process. The PrintController calls `validateDocument()` to ensure the uploaded document meets the required file type, size, and other constraints. This validation is handled by the system.

Conditional handling (alt block):

- Invalid document path: If the document is invalid (e.g., wrong format, type, size, ...), the system marks the document as `invalidDocument`. The PrintController sends an `invalid file` response, and the PrintingView calls `displayError()` to show an error message to the student.
- Valid document path: If the document is valid, the system sends back a `Document(name, size, type, data)` response, and then saves the Document, and the PrintingView proceeds to call `previewDocument()` to display a preview of the document for the student to review.

3.2.2 Specify Printing Properties

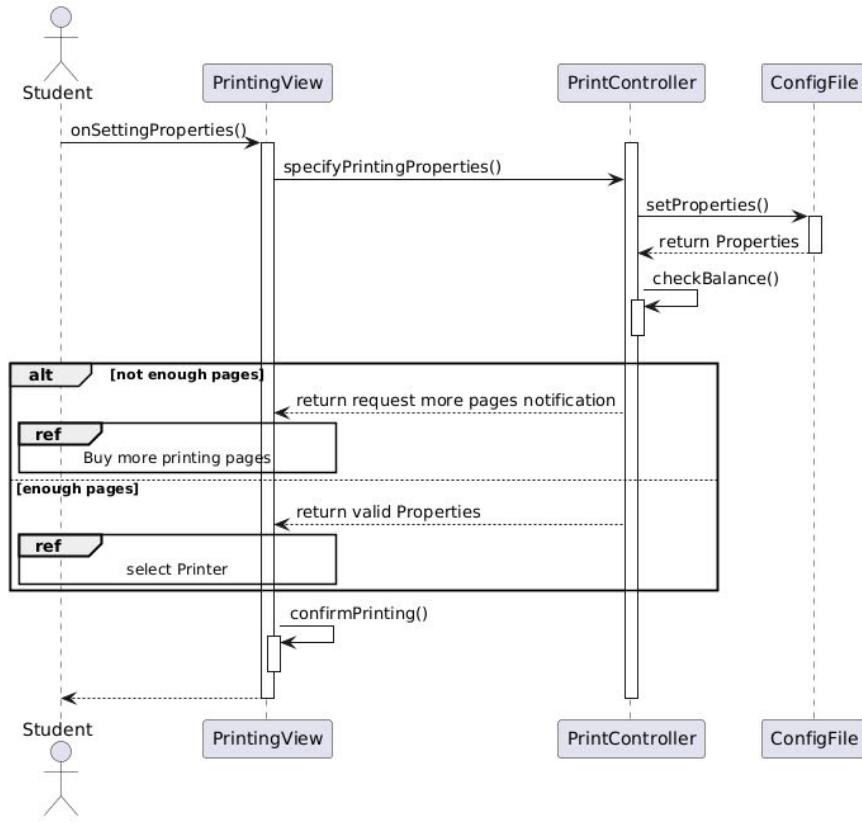


Figure 9: Sequence Diagram of Specify Printing Properties Use Case

The printing process begins with the student interacting with the printing interface (PrintingView). Here, they can customize their print job by selecting options such as page size (Letter, A4, etc.), number of copies, and single or double-sided printing, using standard interface elements like drop-down menus and checkboxes.

Once the student has configured their preferences, the PrintingView relays these settings to the PrintController. The PrintController acts as a coordinator, preparing these settings for the system.

The PrintController then consults the ConfigFile to ensure the chosen settings are valid and compatible with the available printing resources. The ConfigFile contains details like printer specifications, paper tray mappings, and default margins.

Next, the PrintController performs a balance check using the `checkBalance()` method. This method calculates the print job's cost, considering factors like the number of pages and printing style, and compares it against the student's current printing quota.

Based on the balance check, two scenarios are possible:

- Insufficient Quota: If the student's quota is insufficient, the PrintController notifies the PrintingView. The PrintingView then informs the student with a clear message and provides an option to purchase additional quota, effectively pausing the printing process until the quota is replenished.
- Sufficient Quota: With sufficient quota, the PrintController confirms the validity of the settings to the PrintingView. The PrintingView then prompts the student to select their desired printer from the available options.

Finally, after printer selection, the PrintingView presents a confirmation screen summarizing the print job details, including settings, printer, and cost. The student can review and confirm or cancel the job. Upon confirmation, the print job is dispatched to the selected printer, completing the process.

3.2.3 Buy Pages

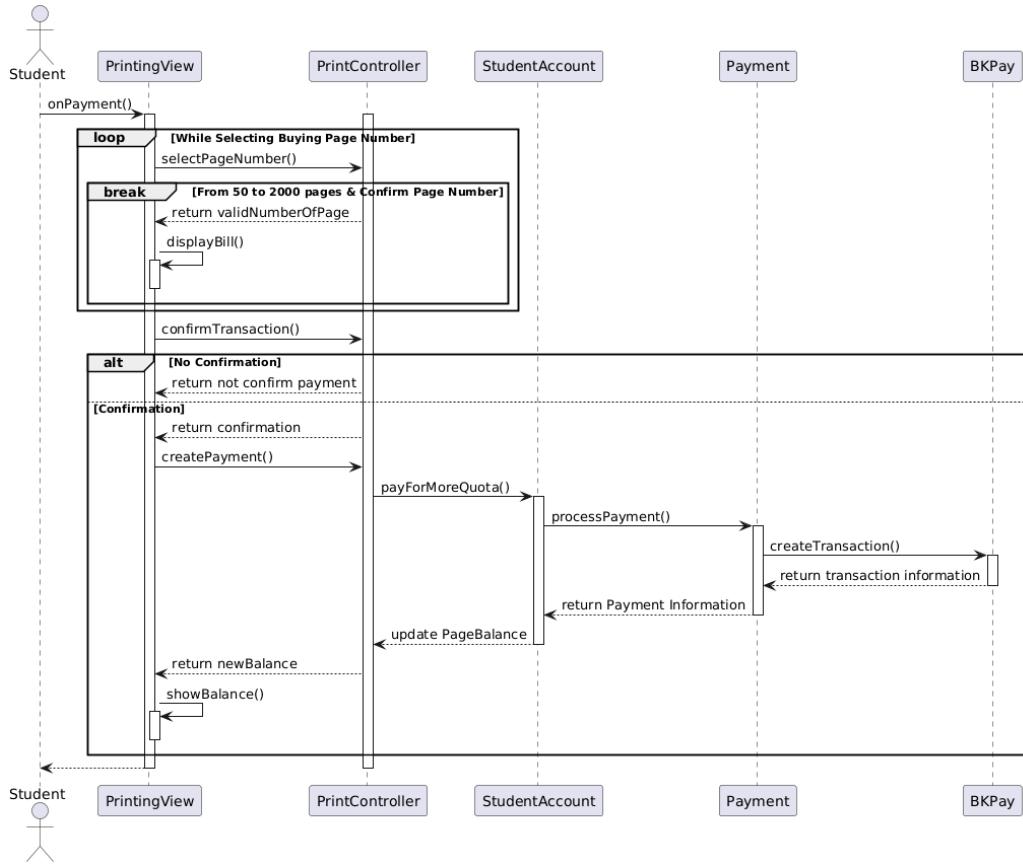


Figure 10: Sequence Diagram of Buy Pages Use Case

The sequence diagram depicts a student purchasing additional printing pages. The student initiates the process via the `onPayment()` function in the `PrintingView`. The student then enters a loop to select the desired number of pages, constrained between 50 and 2000. Upon confirming a valid page number, the `PrintController` returns a confirmation and the `PrintingView` displays the calculated bill.

Next, the student confirms the transaction. If they decline, the `PrintController` returns a "not confirm payment" message. If confirmed, the `PrintingView` initiates payment creation via `createPayment()`. The `PrintController` then requests the `StudentAccount` to `payForMoreQuota()`.

The `StudentAccount` processes the payment through the `Payment` component by the function `processPayment()`, which in turn interacts with the `BKPay` payment gateway to create a transaction by the function `createTransaction()`. `BKPay` returns transaction information to `Payment`, which then relays payment information back to the `StudentAccount`. The `StudentAccount` updates the page balance and informs the `PrintController`.

Finally, the `PrintController` returns the updated balance to the `PrintingView`, then the `PrintingView` displays the updated balance to the student using the `showNewBalance()` method. The process concludes with the `PrintingView` notifying the student of the completed transaction.

3.2.4 Choose Printer

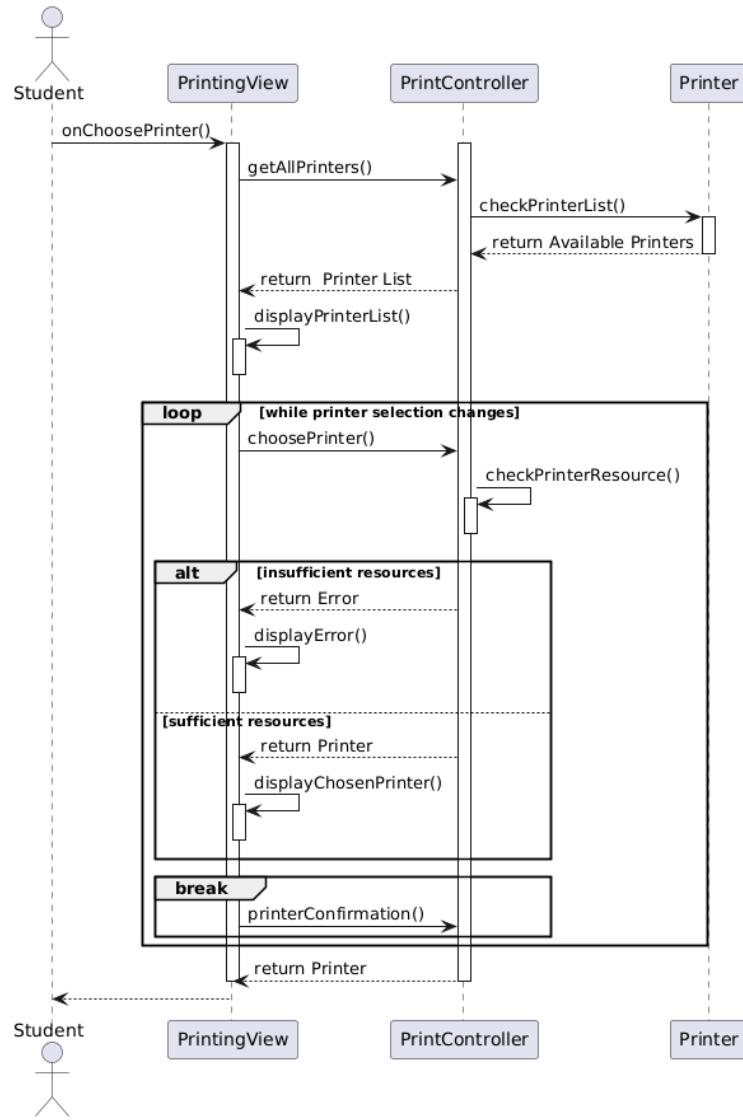


Figure 11: Sequence Diagram of Choose Printer Use Case

The student initiates the printing process through the PrintingView's `onChoosePrinter()` method. The PrintingView requests available printers from the PrintController using `getAllPrinters()`. The PrintController then queries the Printer via `checkPrinterList()` and returns the list to the PrintingView, which displays it to the student using `displayPrinterList()`. The student selects a printer using `choosePrinter()`, which will initiate the loop that allows the student to pick the printer and be able to change their mind. Following choosing the printer, the PrintController checks the printer's resources using `checkPrinterResources()`. If resources are insufficient, an error is returned and displayed by the PrintingView using `displayError()`; otherwise, if resources are sufficient, the printer is returned and displayed using `displayChosenPrinter()`, and then the loop will break if student confirm with `printerConfirmation()`.

3.2.5 Print Document

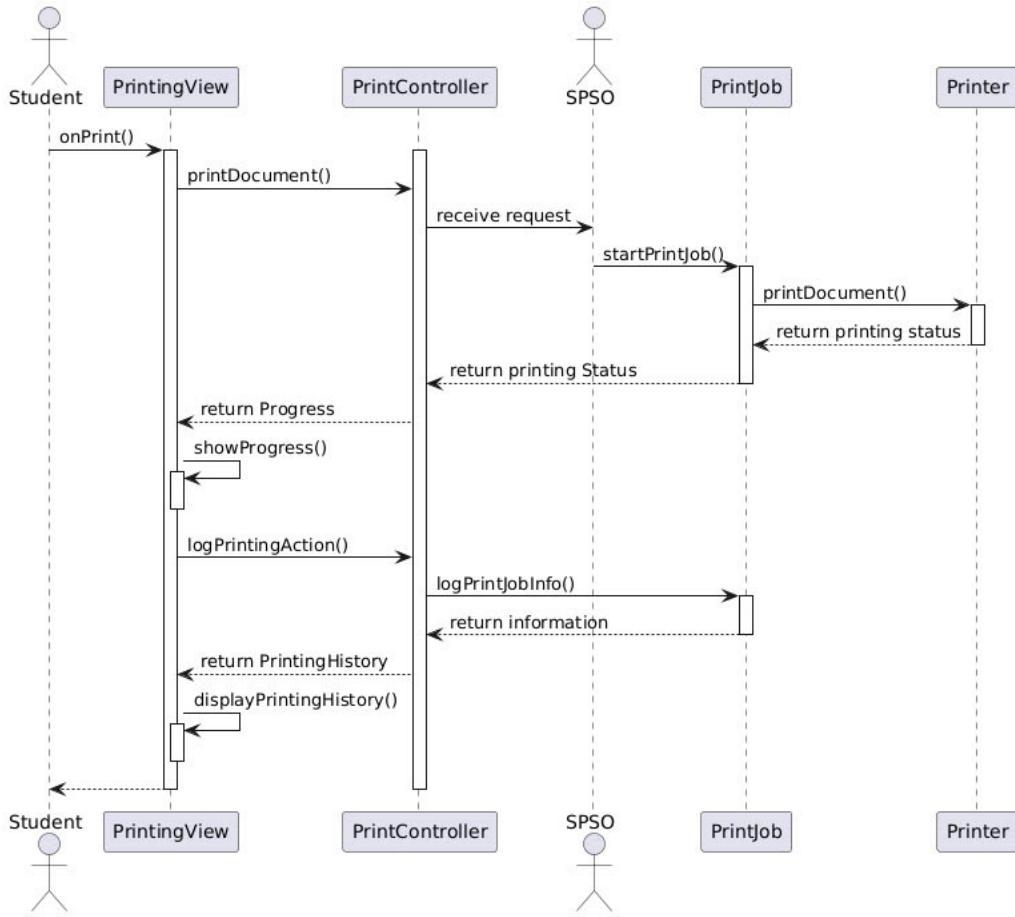


Figure 12: Sequence Diagram of Print Document Use Case

The student starts the process by triggering the `onPrint()` method from the `PrintingView`, indicating they want to print the document. The `PrintingView` sends a `printDocument()` request to the `PrintController`. After that, the `PrintController` forwards the request to the `SPSO`. The `SPSO` initiates the print job by calling `startPrintJob()` to begin the process of sending the document to the printer. The `PrintJob` receives the request and calls the `printDocument()` to print the document via the selected `Printer`. The `PrintJob` returns the current printing status of the document to the controller, indicating the progress of the print job. Next, the controller sends the progress update (`return Progress`) to the `PrintingView`. The `PrintingView` displays the progress to the student using the `showProgress()` method. After the document is printed, the `PrintController` logs the student's printing action with `logPrintingAction()`. The controller calls `logPrintJobInfo()` from `PrintJob` to get the printing information. The logged data is then returned to the `PrintController` with a `return information` message. Finally, the `PrintController` returns the student's printing history (`return Printing History`), which is displayed to the student via the `displayPrintingHistory()` method in the `PrintingView`.

3.3 Class Diagram

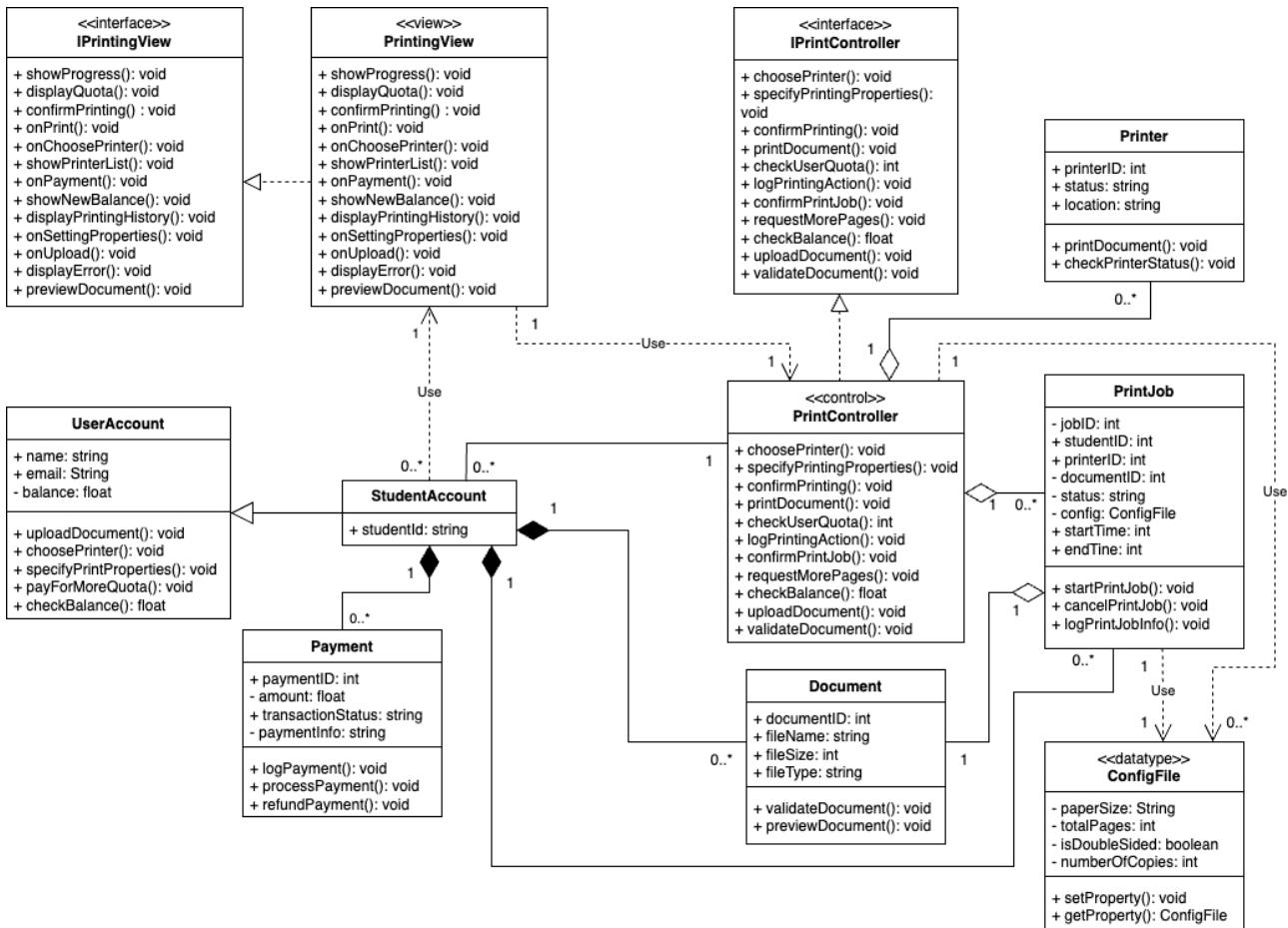


Figure 13: Class Diagram of Printing Document Module

The class diagram represents HCMUT_SSPPS using the Model-View-Controller (MVC) pattern, with defined relationships between the key components. The diagram involves interfaces, controllers, views, and models that interact to manage the printing process for students.

First, the **inheritance relationship** is demonstrated between StudentAccount and UserAccount. StudentAccount inherits attributes and methods from UserAccount, such as the user's name, email, and balance. It also adds a new attribute, studentId, specific to student users.

The **realization relationship** connects PrintController and IPrintController, as well as PrintingView and IPrintingView. PrintController implements the methods defined in IPrintController, such as selecting printers and specifying print properties,... while PrintingView implements the interface IPrintingView, providing essential actions like displaying quotas, confirming printing, showing errors,...

The **aggregation relationship** is used in several places. PrintController aggregates multiple PrintJob and Printer objects. This means PrintController manages the lifecycle of PrintJob and

Printer, but they can still exist independently of the controller. A single PrintController can manage multiple print jobs and printers, but each job or printer belongs to one specific controller. Similarly, PrintJob also aggregates with Document to manage all documents uploaded by students.

The **composition relationship** illustrates a stronger connection. For instance, StudentAccount has a composition relationship with PrintJob, meaning that when a StudentAccount is deleted, its related PrintJob instances are also removed. Additionally, StudentAccount compositions exist with Payment and Document to ensure all related payments and documents are removed when the account is deleted.

The **dependency relationships** are shown between the controller and view components. PrintingView depends on the PrintController to fetch and display necessary data (e.g., showing the available printers or updating the quota after a print job). There is also a dependency between StudentAccount and PrintingView, where the student account uses the view to display the user's printing history and account balance. In addition, PrintController uses ConfigFile to manage and store print job settings that need to be applied when executing the printing task. The same is applied for PrintJob which accesses this configuration to ensure the printing follows the user's preferences. This ensures that every print job adheres to the specific settings provided by the user and stored in ConfigFile.

The one-to-many relationship exists between StudentAccount and Document, PrintJob, and Payment, indicating that each student account can be associated with multiple documents, print jobs, and payments, while each of these entities is linked to a single student account. Similarly, PrintController and PrintingView each have a one-to-many relationship with StudentAccount, allowing them to manage and display information for multiple student accounts. Furthermore, the PrintController has a one-to-many relationship with both ConfigFile and PrintJob, enabling the controller to manage multiple configurations and print jobs simultaneously. On the other hand, the Document and PrintJob entities are connected through a one-to-one relationship, ensuring that each document corresponds to a specific print job. Additionally, PrintJob also establishes a one-to-one relationship with ConfigFile, ensuring that each print job adheres to a unique set of configuration settings tailored to the user's preferences. These relationships collectively illustrate how the various components within the MVC architecture interact to facilitate the printing process efficiently and effectively.

3.4 User Interface

3.4.1 Student's Home Page

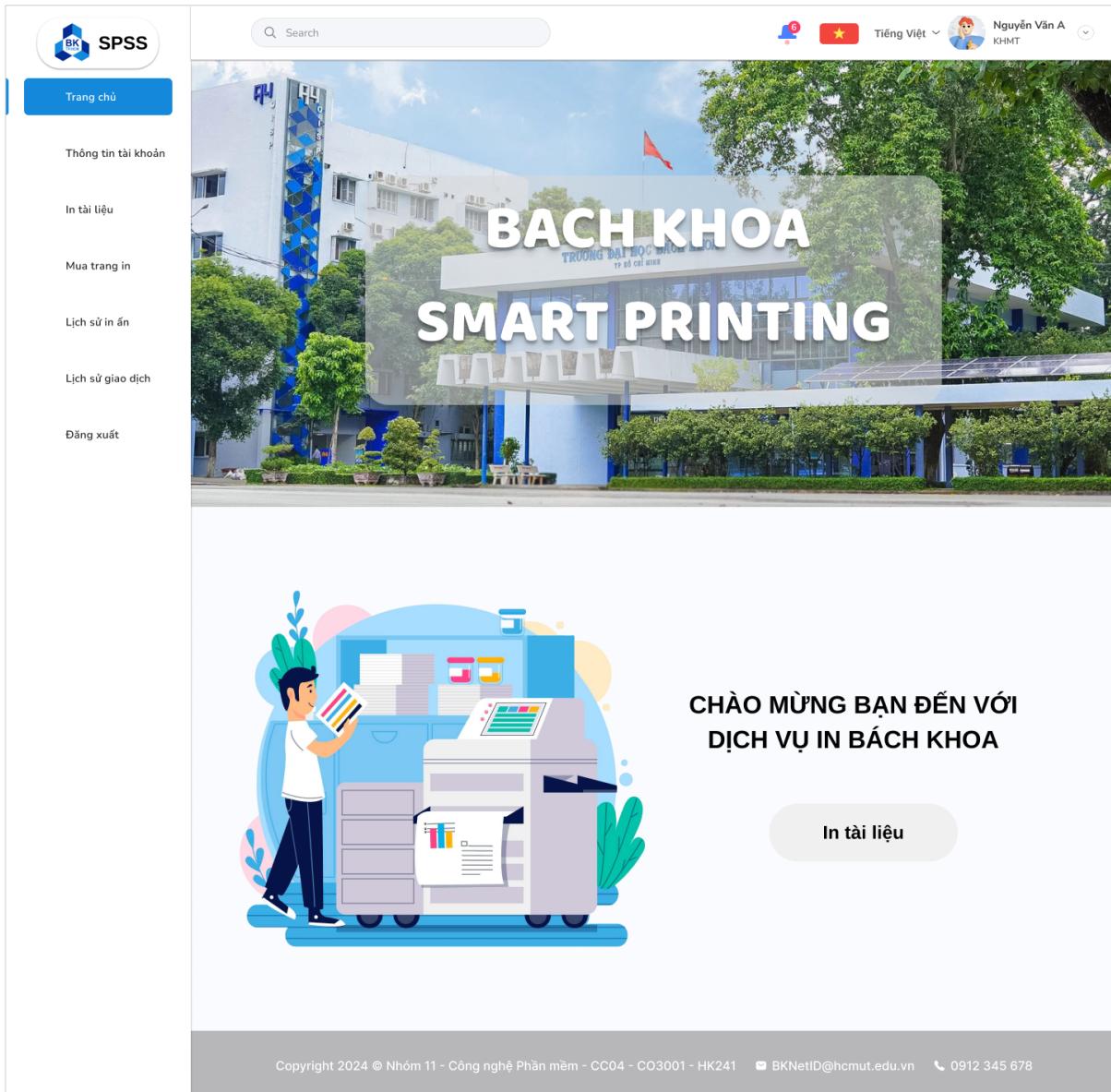


Figure 14: Student's Home Page (picture source: HCMUT)

3.4.2 Admin's Home Page

The screenshot displays the Admin's Home Page with the following key components:

- Header:** SPSS logo, search bar, notifications (6), flag (VN), language (Tiếng Việt), user (Moni Roy Admin).
- Left Sidebar:** Trang chủ, Quản lý máy in, Quản lý người dùng, Lịch sử in ấn, Lịch sử giao dịch, Cài đặt, Đăng xuất.
- Top Metrics:**
 - Tổng sinh viên: 5,423 (↑ 16% tháng này)
 - Tổng số lần in: 1,893 (↓ 1%)
 - Doanh thu: 189,9 Tr (ICOIN)
- Table:** Yêu cầu gần đây (Showing data 1 to 8 of 256K entries)

Họ và tên	MSSV	Khoa	Email	Số máy	Ngày
Nguyễn Văn A	2252678	KH&KTMT	a.nguyen@hcmut.edu.vn	01-LTK-A4-402	14/10/2024
Trần Đăng Dương	2152678	KTHH	duong.tran@hcmut.edu.vn	01-LTK-A4-402	14/10/2024
Trần Minh Hiếu	2252254	CK	hieu.tranminh@hcmut.edu.vn	01-LTK-A4-402	14/10/2024
Nguyễn Quang Anh	2152353	XD	anh.nguyen@hcmut.edu.vn	01-LTK-A4-402	14/10/2024
Hoàng Đức Duy	2357683	KHUD	duy.hoang@hcmut.edu.vn	01-LTK-A4-402	14/10/2024
Lê Quang Hưng	2352345	KH&KTMT	hung.le@hcmut.edu.vn	01-LTK-A4-402	14/10/2024
Nguyễn Thanh Pháp	2452123	Đ-ĐT	phap.nguyen@hcmut.edu.vn	01-LTK-A4-402	14/10/2024
Bùi Anh Tú	2252238	KTHH	tu.bui@hcmut.edu.vn	01-LTK-A4-402	14/10/2024
- Line Chart:** Doanh thu (Revenue) over time (T5 to T10).

Thời gian	Doanh thu
T5	~55
T6	~62
T7	~75
T8	~76
T9	~68
T10	~38
- Bar Chart:** Số lần in (Number of prints) by location.

Địa điểm	Số lần in
LTK A4-402	~60
LTK B10-301	~65
LTK B1-212	~75
ĐI An H1-201	~78
ĐI An H2-203	~70
ĐI An H6-104	~40
- Bar Chart:** Số lượng sử dụng máy in ở 2 cơ sở (Number of printer usage at 2 locations) over months.

Tháng	Cơ sở LTK	Cơ sở ĐI An
Tháng 4	~11k	~10k
Tháng 5	~14k	~8k
Tháng 6	~3k	~21k
Tháng 7	~13k	~3k
Tháng 8	~8k	~7k
Tháng 9	~13k	~10k
Tháng 10	~20k	~7k
- Page Footer:** Copyright 2024 © Nhóm 11 - Công nghệ Phần mềm - CC04 - CO3001 - HK241 | BKNetID@hcmut.edu.vn | 0912 345 678

Figure 15: Admin's Home Page

3.4.3 Printing Module

3.4.3.1 Upload Document Page

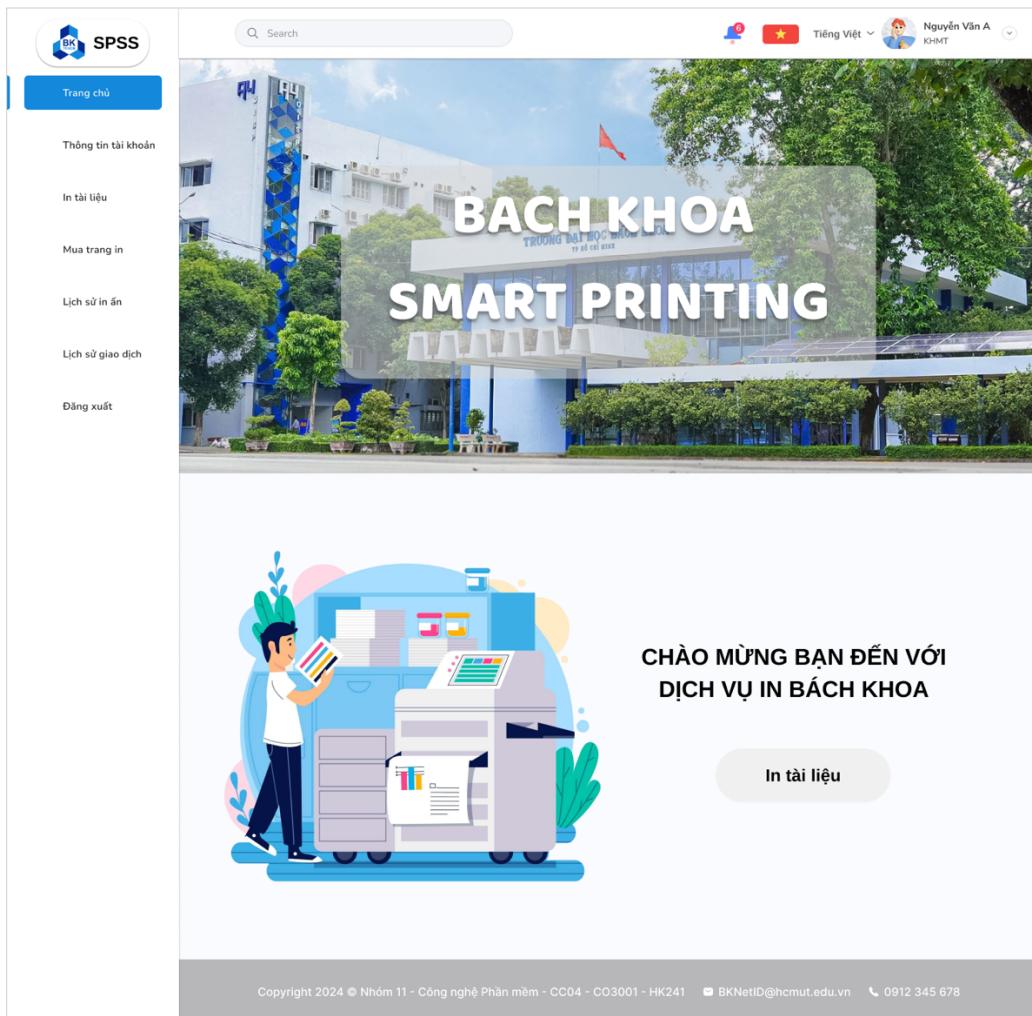


Figure 16: Home Page

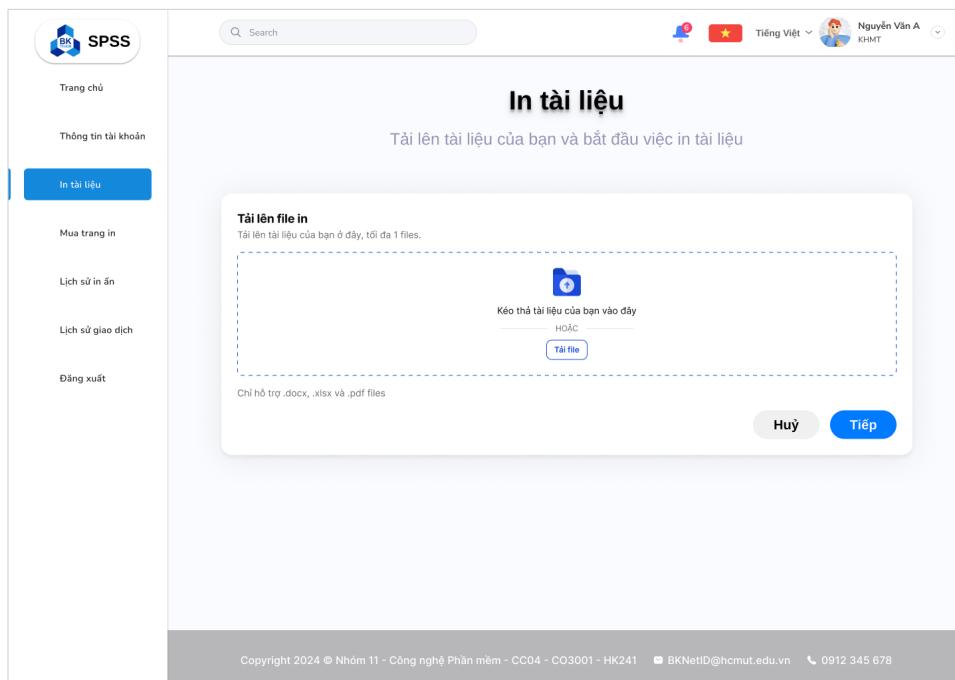


Figure 17: Upload Document

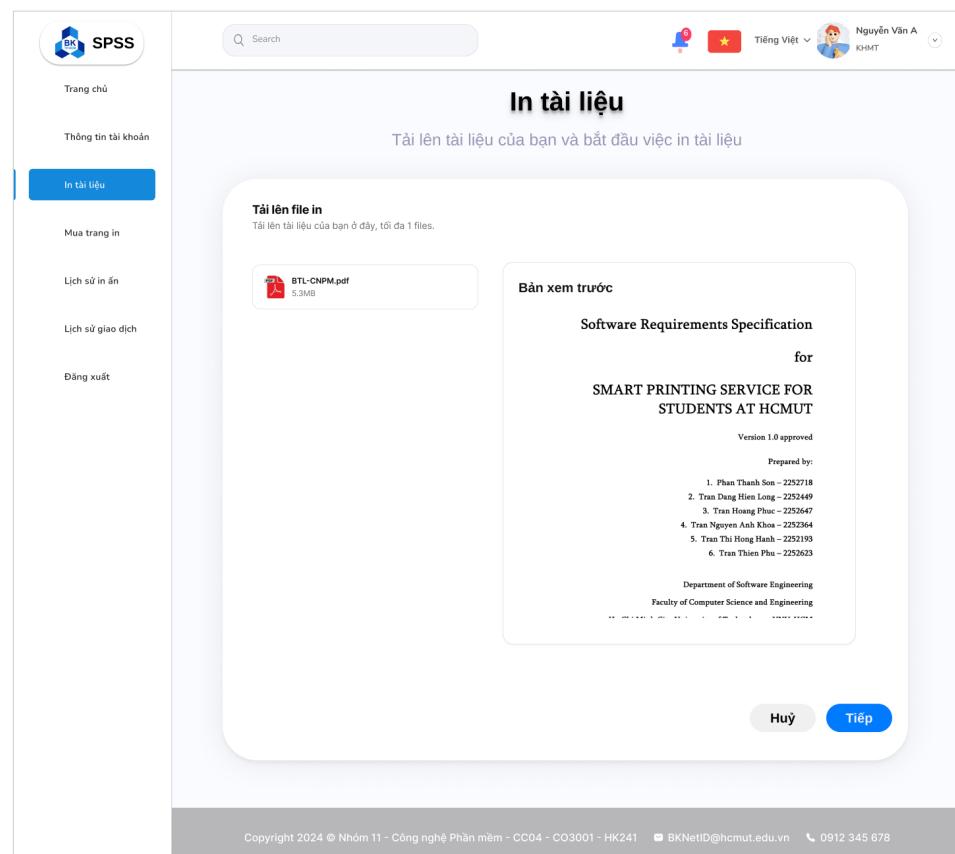


Figure 18: Document Preview

3.4.3.2 Specify Printing Property Page

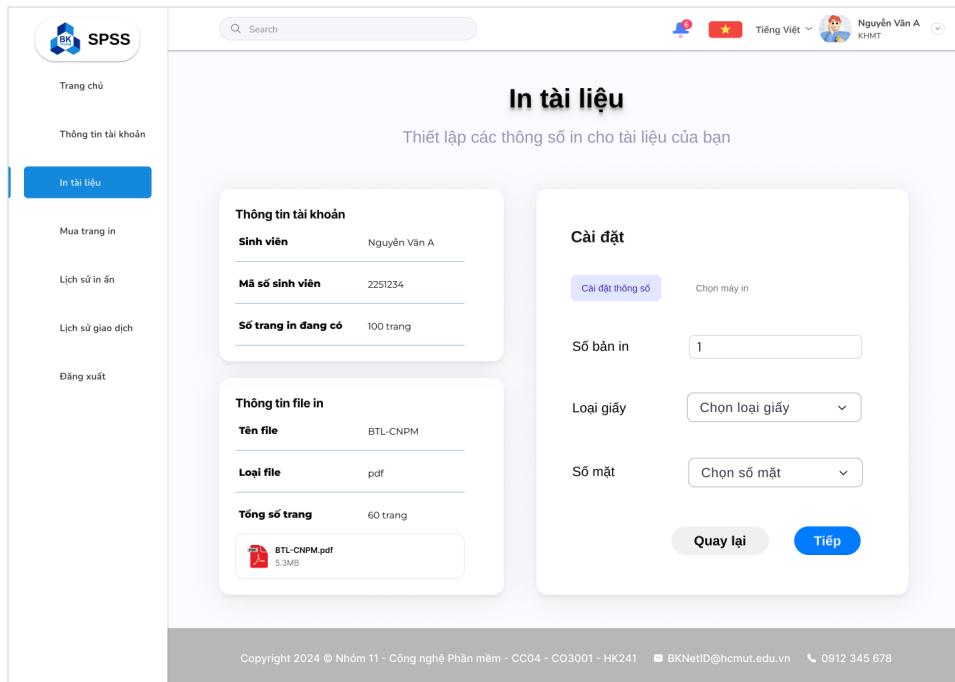


Figure 19: Specify Printing Property

3.4.3.3 Choose Printer Page

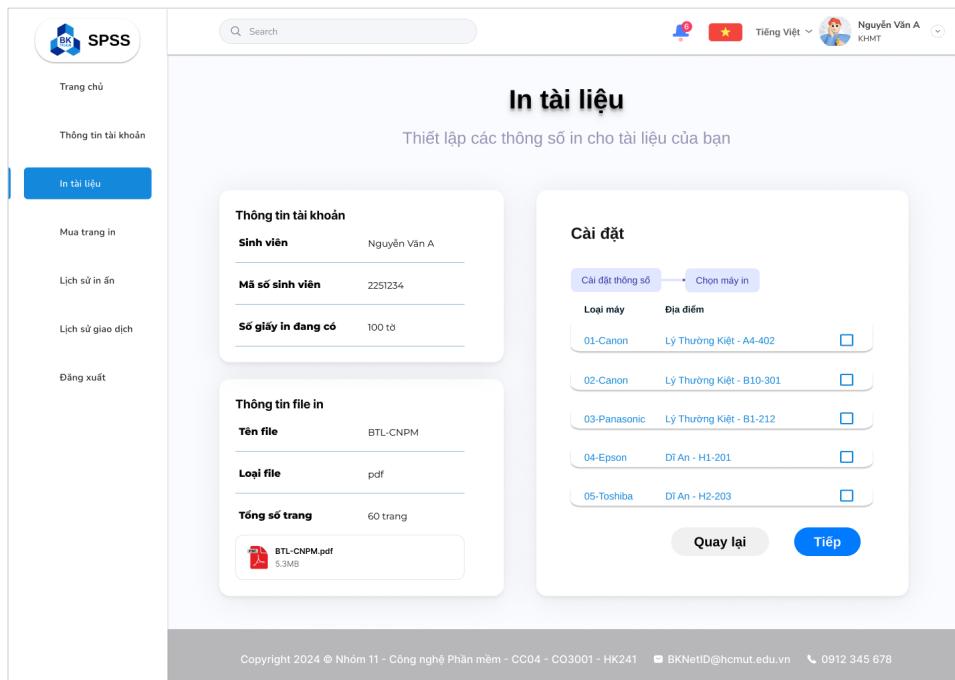


Figure 20: Choose Printer

3.4.3.4 Confirmation Printing Request Page

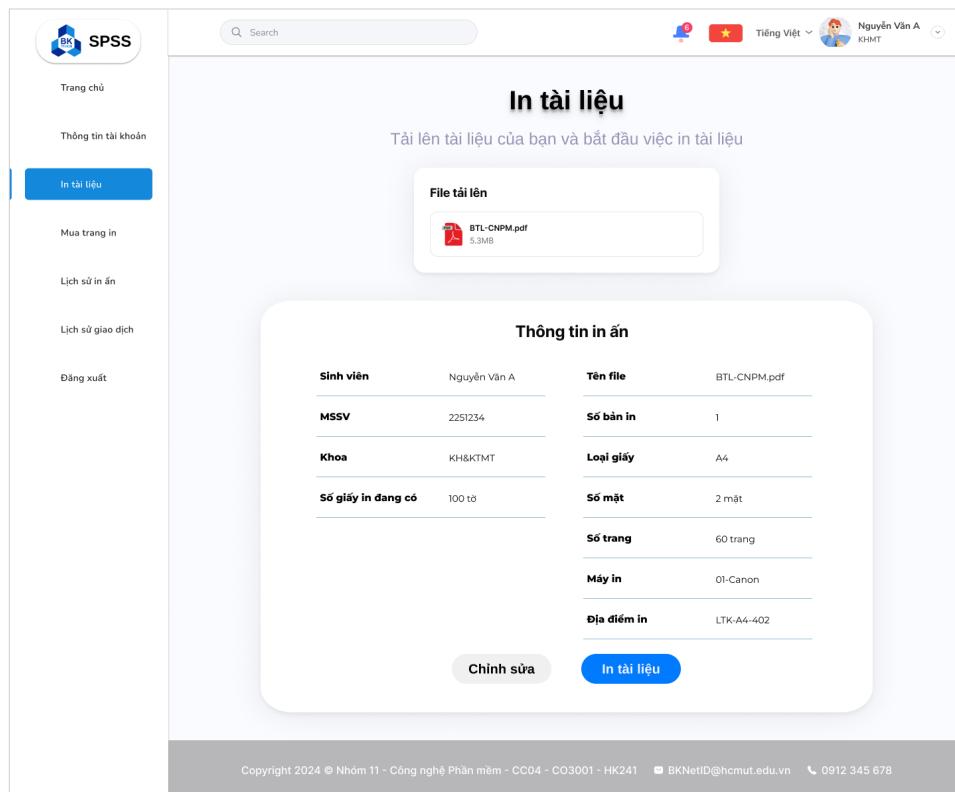


Figure 21: Printing Information Preview

3.4.3.5 Sending Request Successfully Page

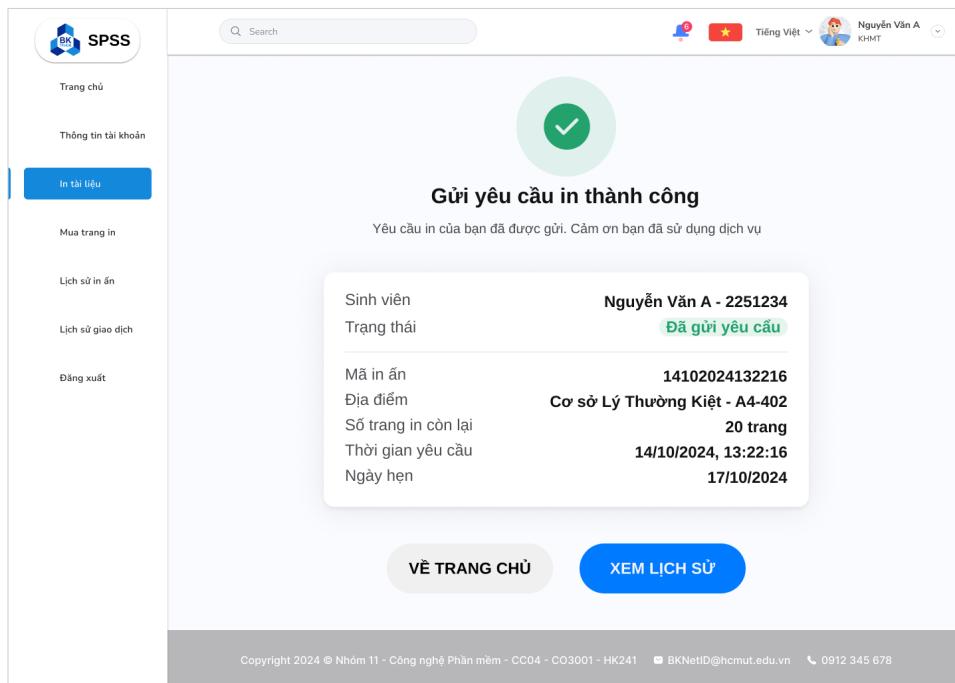


Figure 22: Sending Printing Request Successfully

3.4.3.6 Printing History Page

The screenshot shows a web-based application interface for managing printing history. On the left, a sidebar lists navigation options: Trang chủ, Thông tin tài khoản, In tài liệu, Mua trang in, Lịch sử in ấn (which is highlighted in blue), Lịch sử giao dịch, and Đăng xuất. The main content area is titled "Lịch sử in ấn". It displays a table of print jobs with columns: Mã in ấn, Ngày yêu cầu, Ngày hẹn lấy, Tên tài liệu, Địa điểm in, and Trạng thái. The table contains several entries, each with a status indicator (e.g., Đang xử lý, Đang in, Đã in xong). At the bottom right of the table, there is a pagination control with buttons for 1, 2, ..., 9, 10, and >. The footer of the page includes copyright information: Copyright 2024 © Nhóm 11 - Công nghệ Phần mềm - CC04 - CO3001 - HK241, an email address BKNetID@hcmut.edu.vn, and a phone number 0912 345 678.

Figure 23: Printing History

3.4.3.7 Pay More Printing Balance Page

The screenshot shows a web-based application interface for purchasing additional printing balance. On the left, a sidebar lists navigation options: Trang chủ, Thông tin tài khoản, In tài liệu, Mua trang in (which is highlighted in blue), Lịch sử in ấn, Lịch sử giao dịch, and Đăng xuất. The main content area is titled "Cổng thanh toán online" and features a sub-section "Tuỳ chọn trang in". It asks to choose the number of A4 pages to buy, with a default value of 50. To the right, there is a "Hướng dẫn" (Instructions) box containing a numbered list of steps for buying paper. Below the input field, it says "Thành tiền:" (Amount) and shows "10.000 VND". At the bottom is a large blue "Thanh toán" (Pay) button. The footer of the page includes copyright information: Copyright 2024 © Nhóm 11 - Công nghệ Phần mềm - CC04 - CO3001 - HK241, an email address BKNetID@hcmut.edu.vn, and a phone number 0912 345 678.

Figure 24: Choosing Number of Pages to Purchase

3.4.3.8 Payment Success Page

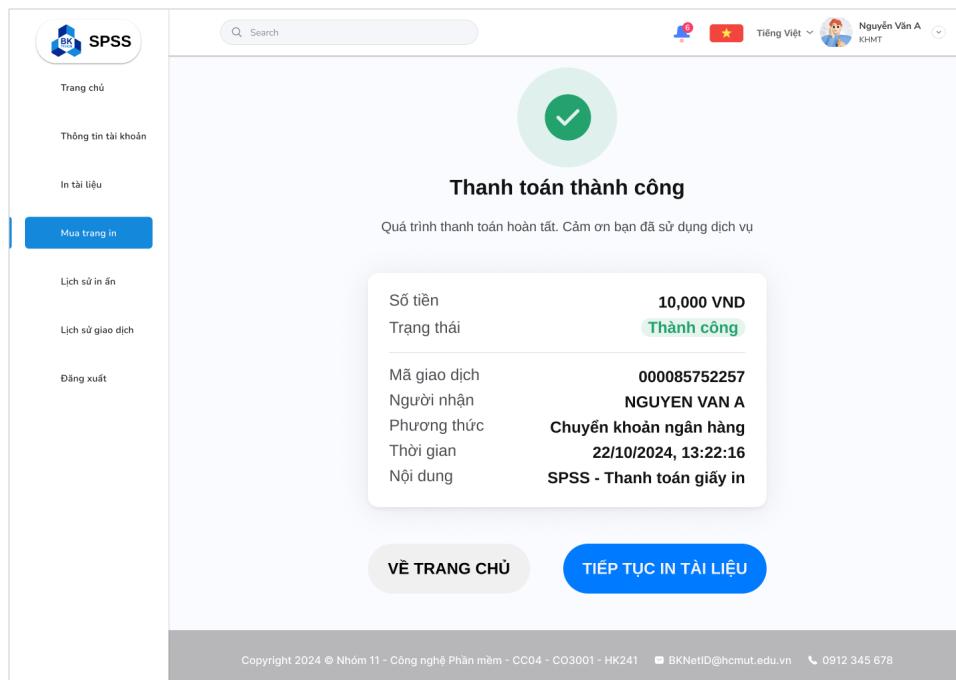


Figure 25: Payment Success

3.4.3.9 Payment History Page

Lịch sử giao dịch				
Mã giao dịch	Ngày giao dịch	Số trang mua	Số tiền	Trạng thái
#23456	10/10/2024	100 trang	20,000 VND	Thành công
#56489	10/10/2024	100 trang	20,000 VND	Thành công
#56489	10/10/2024	100 trang	20,000 VND	Thành công
#98380	10/10/2024	100 trang	20,000 VND	Thành công
#90394	10/10/2024	100 trang	20,000 VND	Thành công
#929348	10/10/2024	100 trang	20,000 VND	Thành công
#48394	10/10/2024	100 trang	20,000 VND	Đang xử lý
#48394	10/10/2024	100 trang	20,000 VND	Đang xử lý

Copyright 2024 © Nhóm 11 - Công nghệ Phần mềm - CC04 - CO3001 - HK241 BKNetID@hcmut.edu.vn 0912 345 678

Figure 26: Payment History

4. Architecture Design

4.1 Architectural Diagram

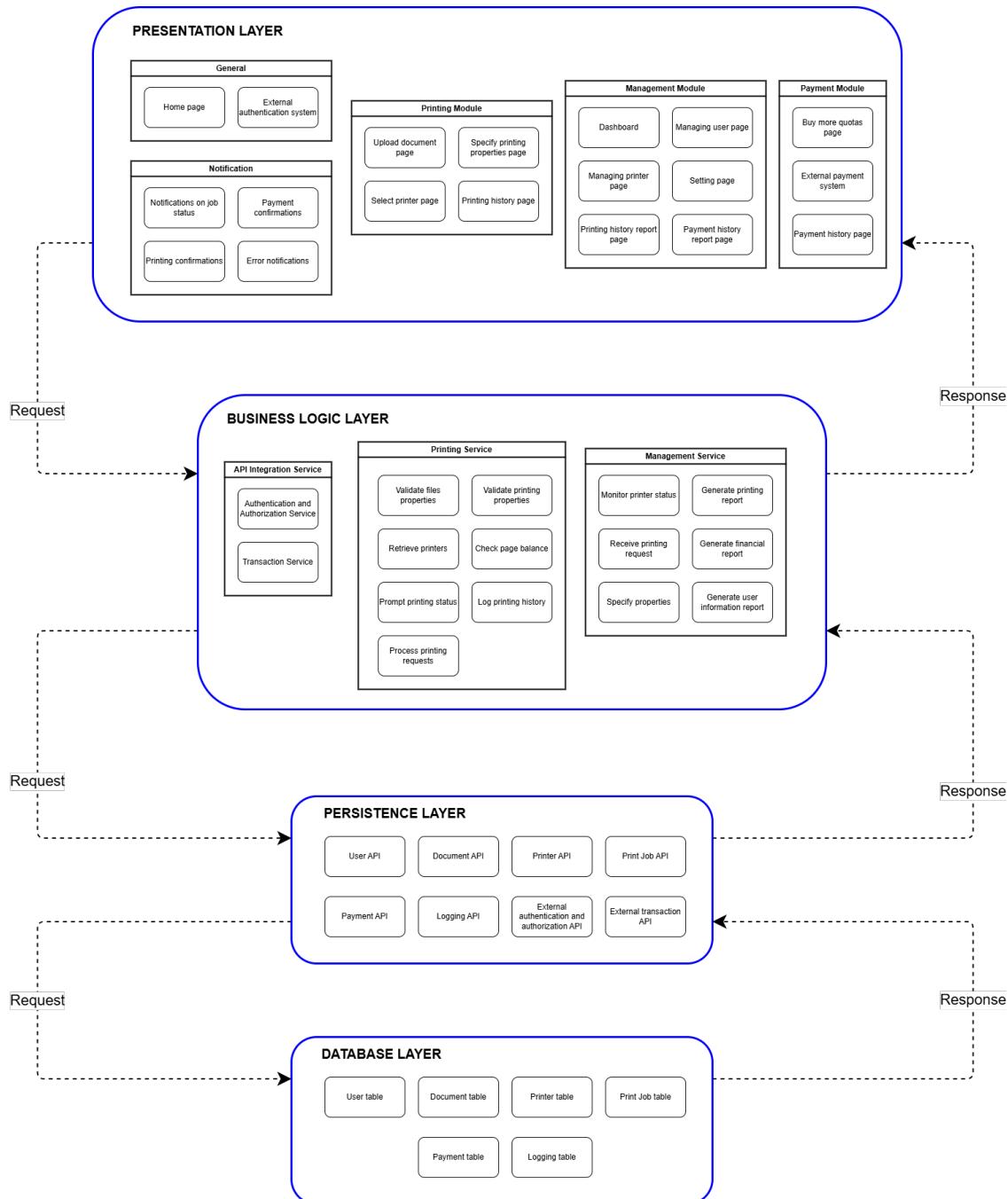


Figure 27: Architectural Diagram of Printing Module

4.1.1 Presentation Strategy

The presentation layer of this software system is organized into several modules, each designed to handle specific user-facing functionalities within the application. This layer is structured to deliver a seamless user experience by organizing components logically according to their functionalities. Here's a breakdown of each module and its purpose:

4.1.1.1 General Module

Contains the essential entry points and authentication mechanisms for the application.

Components:

- Home Page: The main landing page of the application.
- External Authentication System: Integrates with external authentication providers to allow secure user login.

4.1.1.2 Notification Module

Manages notifications related to various system events, providing users with real-time updates on their actions and system status.

Components:

- Notifications on Job Status: Alerts users about the current status of their jobs.
- Payment Confirmations: Notifies users upon successful payments.
- Printing Confirmations: Provides confirmation after print jobs are completed.
- Error Notifications: Alerts users about any errors encountered during processing.

4.1.1.3 Printing Module

The core module dedicated to managing the printing functionality, guiding users through document upload to specifying print properties.

Components:

- Upload Document Page: Allows users to upload documents for printing.
- Specify Printing Properties Page: Enables users to set specific printing options for their documents.
- Select Printer Page: Allows users to choose a printer for their job.
- Printing History Page: Displays the history of previous print jobs.

4.1.1.4 Management Module:

Provides administrative tools for managing various aspects of the application, including user and printer settings.

Components:

- Dashboard: A centralized interface for monitoring key metrics and data.

- Managing User Page: Manages user profiles and access permissions.
- Managing Printer Page: Administers printer settings and availability.
- Setting Page: Allows customization of various application settings.
- Printing History Report Page: Generates reports based on the printing history.
- Payment History Report Page: Provides a record of past payments.

4.1.1.5 Payment Module:

Manages the purchasing and payment processes within the application.

Components:

- Buy More Quotas Page: Allows users to purchase additional print quotas.
- External Payment System: Integrates with external payment processors for secure transactions.
- Payment History Page: Shows a history of completed payments.

4.1.2 Data Storage Approach

4.1.2.1 User table

Attribute	Type	Description
UserID	String	Unique identifier for the user.
Username	String	Full name of the user.
Email	String	Email address of the user.
RemainingPages	Integer	Count of remaining pages the user can print.

The User contains basic information about the users of the system (students or other personnel). The key interaction points for the User collection are:

- Documents: A User uploads documents, which are stored in the Document collection. The UserID is used as a foreign key in the Document collection to link the uploaded documents to the user.
- Print Jobs: A User initiates print jobs, which are recorded in the Print Job collection. The UserID is used to associate a print job with the user who initiated it.
- Payments: A User can make payments for additional pages, which are recorded in the Payment collection. The UserID is used to track which user made the payment.
- Logging: Every action performed by the User is recorded in the Logging collection, such as document uploads, print job executions, and payment actions. The UserID links the log entry to the specific user.

4.1.2.2 Document table

Attribute	Type	Description
DocumentID	String	Unique identifier for the document.
UserID	String	ID of the user who uploaded the document (foreign key).
Filename	String	Name of the document file.
FileType	String	File format of the document (e.g., PDF, DOCX).
FileSizeMB	Float	Size of the document in MB.

The Document stores details about documents uploaded by users. Each document has the following interactions:

- User: Each document is uploaded by a User, and the UserID (foreign key) associates the document with that user.
- Print Jobs: When a User initiates a print job, the DocumentID is used to link the document being printed with the specific print job. This helps track which document was printed, how many pages were printed, and where the document was printed.

4.1.2.3 Printer table

Attribute	Type	Description
PrinterID	String	Unique identifier for the printer.
PrinterBrand	String	Brand name of the printer.
PrinterName	String	Model name of the printer.
Location	Object	Embedded object detailing the printer's location.
Location.Facility	String	Facility where the printer is located.
Location.Building	String	Building where the printer is located.
Location.Room	String	Room number where the printer is located.
Status	Boolean	Printer's operational status (enabled/disabled).
PrintedPages	Integer	Total number of pages printed by this printer.

The Printer stores information about the printers available in the system. The interactions for the Printer collection include:

- Print Jobs: Each Print Job references a PrinterID, which identifies the specific printer used to print the document. This allows tracking of which printer was used for each print job.

- Status and Printed Pages: The Status of a printer (enabled/disabled) and the PrintedPages count help the system monitor printer availability and usage.

4.1.2.4 PrintJob table

Attribute	Type	Description
PrintJobID	String	Unique identifier for each print job.
UserID	String	ID of the user initiating the print job (foreign key).
DocumentID	String	ID of the document being printed (foreign key).
PrinterID	String	ID of the printer used (foreign key).
PrintingTime	Date	Date and time when the print job was executed.
PrintedPages	Integer	Number of pages printed in this job.
PaperType	String	Type of paper used for printing (e.g., A4, A3).

The Print Job records every print job initiated by users. Each print job contains the following interactions:

- User: Each print job is linked to a User by the UserID, indicating which user initiated the job.
- Document: Each print job references the document being printed through the DocumentID, helping track which document was printed.
- Printer: Each print job references the PrinterID to specify which printer was used for the print job. This interaction also helps to update the PrintedPages count on the corresponding printer.
- Logging: Print jobs are logged in the Logging collection to track the action, success, or failure, as well as the timestamp and any other relevant details.

4.1.2.5 Payment table

Attribute	Type	Description
PaymentID	String	Unique identifier for each payment transaction.
UserID	String	ID of the user making the payment (foreign key).
Time	Date	Date and time when the payment was made.
Method	String	Method used to pay for additional pages (e.g., credit card, PayPal).

Amount	Float	Total amount paid for purchasing additional pages.
PurchasedPages	Integer	Number of additional pages purchased in the transaction.

The Payment records payments made by users for purchasing additional printing pages. The interactions for the Payment collection are as follows:

- User: Each payment is linked to a User through the UserID, showing which user made the payment for extra pages.
- Logging: Payment transactions are logged in the Logging collection to track when and how much the user paid, and the status of the payment.

4.1.2.6 Logging table

Attribute	Type	Description
LogID	String	Unique identifier for each log entry.
UserID	String	ID of the user associated with the log entry (foreign key).
ActionType	String	Type of action performed (e.g., print, payment, upload).
Description	String	Detailed description of the action performed.
Timestamp	Date	Date and time when the action was logged.
Status	String	Status of the action (e.g., successful, failed, pending).

The Logging tracks various actions performed by users and the system, allowing for auditing and monitoring. These are the main interactions for the Logging collection:

- User: The UserID associates each log entry with a specific user, helping identify who performed the action (e.g., printing a document, making a payment).
- ActionType and Description: The ActionType describes what action was taken (e.g., document upload, print job initiation, payment) and the Description provides more details about the action.
- Timestamp and Status: The Timestamp captures when the action occurred, and the Status indicates whether the action was successful, failed, or still pending.

4.1.3 API Management

The Persistence Layer in this system handles communication between the Business Logic Layer and the Database Layer via APIs, providing standardized interfaces for accessing and manipulating data. Below are descriptions of each API within this layer

4.1.3.1 User API

Purpose: Manages user-related information in the database, supporting authentication, authorization, and user profile management.

Methods:

Method	Return Type	Description
getUserDetails(userID: String)	User	Retrieves user information (e.g., name, email, role) for a specified userID.
updateUserInfo(userID: String, userInfo: Object)	Boolean	Updates specific details in the user's profile (e.g., email or faculty), based on the provided userInfo object.
getUserPermissions(userID: String)	List<String>	Retrieves a list of permissions associated with the user to verify access rights for various system modules.
getUserInfo(userID: int)	UserAccount	Retrieves the account details (name, email, balance) of a specified user.
addUser(user: UserAccount)	int	Adds a new user to the system, returning the unique userID for the user.
updateUserBalance(userID: int, balance: float)	Boolean	Adjusts the user's balance, typically after a payment or quota purchase.
deleteUserAccount(userID: int)	Boolean	Deletes a user account from the system

Use Cases:

- Used during login to fetch user details and permissions.
- Called when administrators update user profiles or permissions.

4.1.3.2 Document API

Purpose: Handles storage, retrieval, and management of document data for printing operations.

Methods:

Method	Return Type	Description

getDocumentDetails(docID: String)	Document	Retrieves metadata (e.g., file name, type, size) for a document identified by docID.
storeDocument(document: Object)	String	Saves a new document in the database and returns a unique document ID.
deleteDocument(docID: String)	Boolean	Deletes a document after it has been printed or reaches a specified retention period.
updateDocument(docID: String, documentData: Object)	Boolean	Updates the properties of an existing document, such as file name or print settings.

Use Cases:

- Called when a user uploads a document for printing.
- Allows updates or deletions based on retention policies or user actions.
- Enables administrators to retrieve document details for auditing or troubleshooting.

4.1.3.3 Printer API

Purpose: Provides access to printer information, including status, location, and configuration settings.

Methods:

Method	Return Type	Description
getPrinterStatus(printerID: String)	String	Returns the current status of a printer (e.g., "Available", "Busy", "Out of Order").
getPrinterLocation(printerID: String)	String	Retrieves the building and room information for a specific printer.
updatePrinterConfig(printerID: String, config: Object)	Boolean	Updates printer configuration details such as paper type, default print quality, and operational status.
getAvailablePrinters()	List<String>	Retrieves a list of all available printers, optionally filtered by location or printer type.

resetPrinter(printerID: String)	Boolean	Resets a printer's status to "Available" in case of error or failure.
---------------------------------	---------	---

Use Cases:

- Used by students to check printer availability before sending print jobs.
- Called by administrators to monitor printer locations and configurations.

4.1.3.4 Print Job API

Purpose: Manages print job lifecycle, from submission to completion, and provides status tracking.

Methods:

Method	Return Type	Description
submitPrintJob(jobDetails: Object)	String	Submits a new print job with details such as userID, docID, and printerID, and returns a unique job ID.
getPrintJobStatus(jobID: String)	String	Retrieves the current status of a specific print job (e.g., "In Queue", "Printing", "Completed").
cancelPrintJob(jobID: String)	Boolean	Cancels a print job if it has not yet started, freeing up resources.
updatePrintJob(jobID: String, newDetails: Object)	Boolean	Updates details of a print job, such as changing the printer or priority level.
getPrintHistory(userID: String)	List<PrintJob>	Retrieves the printing history for a specific user, showing details of past print jobs.

Use Cases:

- Used by the Printing Module when a user initiates a print request.
- Allows students or administrators to check job status or cancel jobs if needed.

4.1.3.5 Payment API

Purpose: Facilitates payment processing for students to purchase additional printing quotas.

Methods:

Method	Return Type	Description
processPayment(userID: String, amount: Double)	TransactionID	Processes a payment transaction and returns a TransactionID upon success.
getPaymentHistory(userID: String)	List<Transaction>	Retrieves the history of all payment transactions made by a user.
refundTransaction(transactionID: String)	Boolean	Initiates a refund for a specified transaction if applicable.
verifyPayment(transactionID: String)	Boolean	Verifies if a payment was processed successfully.

Use Cases:

- Called when students purchase additional pages or quotas.
- Administrators can review payment history or process refunds when required.

4.1.3.6 Logging API

Purpose: Records system activities and user actions for tracking and auditing purposes.

Methods:

Method	Return Type	Description
logActivity(userID: String, action: String, details: Object)	Boolean	Logs a specific user action or system event, including details such as timestamps and action type.
getActivityLogs(filterParams: Object)	List<LogEntry>	Retrieves logs based on filter criteria (e.g., by userID, date range, or action type).
deleteOldLogs(retentionPeriod: Integer)	Boolean	Periodically deletes logs older than a specified retention period.

Use Cases:

- Called whenever a significant user action occurs, such as document printing or system changes.
- Allows administrators to access activity logs for security and troubleshooting.

4.1.3.7 External Authentication and Authorization API

Purpose: Integrates with an external authentication system to validate user credentials and retrieve user roles.

Methods:

Method	Return Type	Description
validateUserCredentials(username: String, password: String)	Boolean	Authenticates user login credentials with an external system.
fetchUserRoles(userID: String)	List<String>	Retrieves roles and permissions for a user to enable role-based access control.
resetPassword(userID: String)	Boolean	Initiates a password reset for a user via the external authentication system.

Use Cases:

- Called during login to authenticate and authorize users, particularly for those with roles requiring restricted access.

4.1.3.8 External Transaction API

Purpose: Interfaces with an external payment service to handle secure transactions for purchasing additional pages or printing services.

Methods:

Method	Return Type	Description
initiateTransaction(userID: String, amount: Double)	TransactionID	Starts a new transaction and returns a TransactionID for tracking.
validateTransaction(transactionID: String)	Boolean	Confirms the transaction's success with the payment service before granting additional printing pages.
getTransactionStatus(transactionID: String)	String	Checks the current status of a payment (e.g., "Pending", "Completed", "Failed").

cancelTransaction(transactionID: String)	Boolean	Cancels a transaction if it's still pending.
--	---------	--

Use Cases:

- Used by the Payment Module to securely handle transactions for purchasing print credits.
- Allows the system to verify and record completed transactions before updating user quotas.
- Each API in the Persistence Layer serves as a bridge between the Business Logic Layer and the Database Layer, ensuring modular and secure interactions with the underlying data. These APIs allow for clear separation of functionalities, making the system scalable and easier to maintain.

4.2 Component Diagram

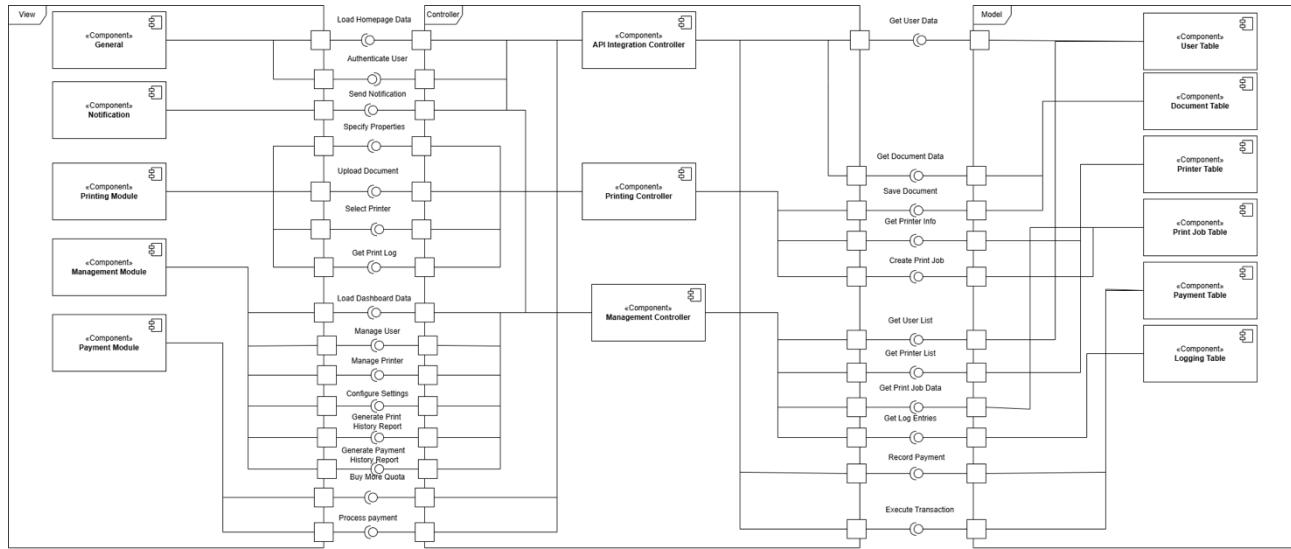


Figure 28: Component Diagram for Printing Module

In the component diagram, these different modules, controllers and tables are connected, showing how they relate to other system parts. Left, five main modules including **General**, **Notification**, **Printing Module**, **Management Module** and **Payment Module** respectively perform user-oriented functionalities. With that, the General module has functions that include **Load Homepage Data** and **Authenticate User** so it establishes how the system is going to talk to the system. The Notification module consists of the **Send Notification** which sends updates to the user. **Upload Document**, **Select Printer**, and **Get Print Log** are features of this Printing Module, allowing document management and printing operations. **Load Dashboard Data**, **Manage User**, **Manage Printer** and **Configure Settings** are all functions found in the Management module which is an administrative module. It, also, contains reporting functions entitled **Generate Print History Report** and **Generate Payment History Report**. User quota and payment operations handled by the Payment module include **Buy More Quota** and **Process Payment**.

Across from the center are a few controllers connecting modules to the database table on the right. It provides functions such as **Get User Data**, **Get Document Data** and **Save Document** to handle interactions between systems with external or internal data sources. The printing controller has the function of creating a **Print Job** and a **Get Printer Info** which implies that it is responsible for managing and executing printing tasks. Functions such as **Get User List**, **Get Printer List**, and **Get Print Job Data**, are imperative for system management and monitoring, and are what the Management Controller is responsible for, retrieving various system lists and data. Financial transactions are performed and consist of **Record Payment**, **Execute Transaction** etc. to support and log payment activities within the system via the Payment Controller.

Several tables on the right store and organize system data. The User Table contains the User List & User Data while Document Table is linked with saved documents. Information on the printers and the individual print jobs are managed by the Printer Table and Print Job Table, respectively. Supporting functions Record Payment and Execute Payment in beta, the Payment Table to log payment-related data. The Logging Table documents system activities, and is updated to record in the Logging Table a record of system activities, which is useful in the Get Log Entries function.

5. Implementation – Sprint 1

5.1 Version control

GITHUB: <https://github.com/tthonghanh/HCMUT-smart-printing-service>

5.2 Usability Test

5.2.1 Recruit testers

Name	Role
Tran Hoang Phuc	Facilitator
Phan Thanh Son	Observer
Tran Thi Hong Hanh	Analyst
Tran Dang Hien Long	Tester
Tran Nguyen Anh Khoa	Tester
Tran Thien Phu	Tester

5.2.2 Define tasks

- Log In: Log in with HCMUT_SSO and verify successful access.
- Upload Document: Upload a file (e.g., PDF/Word) and check for acceptance or error messages.
- Select Printer: Choose your preferred printer and confirm its availability.
- Set Printing Properties: Configure paper size, pages, one-/double-sided, and number of copies.
- Check Balance: View printing balance and verify sufficient pages for the job.
- Submit Print Job: Confirm settings, submit the job, and check status updates.
- Retrieve Document: Collect the document from the printer and verify correctness.
- View Log: Access your printing history and check details for accuracy.
- Buy Pages: Purchase additional pages using BKPay and confirm the updated balance.

- Test Errors: Attempt unsupported actions (e.g., upload invalid file, print beyond balance) and verify error handling.

5.2.3 Define test strategy

5.2.3.1 Define role

Facilitator (Member 1):

- Prepares and manages the testing session.
- Monitors participant progress and ensures quality in tasks and responses.
- Resolves participant issues during sessions.

Observer (Member 2):

- Reviews task recordings and user feedback.
- Notes user behavior, usability issues, and recurring patterns.
- Shares observations with the analyst to support insights.

Analyst (Member 3):

- Aggregates and analyzes data from all sessions.
- Categorizes usability issues by severity and frequency.
- Prepares a report with actionable recommendations.

Testers (Members 4, 5, 6):

- Act as mock users during initial test runs to validate tasks.
- Provide qualitative feedback based on their experience.
- Rotate roles to facilitate or observe in subsequent sessions.

Rotation Plan: Rotate roles every session (10 total):

- Member 1 becomes the analyst.
- Member 2 becomes the facilitator.
- Member 3 becomes the observer
- Members 4, 5 and 6 alternate between facilitator/observer/analyst or continue as testers.

5.2.3.2 Testing Approach

Scope:

- Focus on key tasks such as printing a document, buying more pages, and selecting a printer.
- Address potential user pain points (e.g., difficulty locating features).

Test Setup:

- Use a remote testing platform (e.g., Figma or prototype testing tool).
- Define clear and realistic tasks (e.g., “Upload a document and set printing preferences”).
- Include follow-up questions for qualitative insights.

Execution Workflow:

- Facilitator: Prepares the platform, monitors progress, and handles participant queries.
- Observer: Watches live or recorded sessions, notes user actions, and identifies usability challenges.
- Testers: Complete tasks as mock users, mimicking realistic participant interaction.
- Analyst: Analyzes early data trends and adjusts test strategies if needed.

5.2.3.3 Data Collection

- Screen Recordings: Capture user interactions via the testing platform.
- Feedback: Gather verbal or written responses from participants after tasks.
- Task Metrics: Record task completion rates, time taken, and any errors encountered.

5.2.3.4 Analysis Workflow

Facilitator Responsibilities:

- Monitors session flow and prepares session summaries.
- Ensures data collection consistency and resolves any testing issues.

Observer Responsibilities:

- Reviews session recordings, highlighting usability patterns.
- Categorizes feedback into actionable insights.

Analyst Responsibilities:

- Compiles data from all sessions, focusing on recurring issues.
- Prepares a final report with prioritized recommendations.

Tester Contributions:

- Provide firsthand feedback from completing tasks.
- Highlight unclear or challenging parts of the process.

5.2.3.5 Deliverables

- Usability report highlighting key findings, user struggles, and improvement suggestions.
- Role-specific feedback, ensuring perspectives from all members.
- Actionable recommendations to refine the product.

5.2.4 Conduct the test

5.2.4.1 Testing as a Student

Log In as a Student:

- Navigate to the Home Page of the application.
- Click on the “Đăng nhập” button located in the top-right corner of the page.
- Log in using your MyBK username and password.
- Ensure that the login process works smoothly and provides clear feedback for success or failure.

Test the Navigation Bar:

- After logging in, you will land on the “Trang chủ” (Home Page).
- Begin browsing the pages using the side navigation bar:
- Verify that each menu item is clickable and leads to the correct page.
- Note any misalignment, unclear labels, or confusing navigation patterns.

Test the “Thông tin tài khoản” Page:

- Go to the “Thông tin tài khoản” page from the side navigation bar.
- Evaluate the following:
 - + Line spacings and font sizes: Check if the information is well-spaced, readable, and visually appealing.
 - + Edit Buttons: Test the functionality of all “Chỉnh sửa” buttons (e.g., email, password) and ensure they work correctly.
- Look for any inconsistencies in layout or design compared to other pages.

Test the “In tài liệu” Page:

- Navigate to the “In tài liệu” (Print Documents) page.
- Perform the following tasks:
 - + Upload a file supported by the service (e.g., PDF, Word).
 - + Select your printing preferences, such as: Paper type, Printing options, Number of copies.
 - + Verify that the system correctly displays the selected printing information.
- Start the printing process and collect the documents at the designated printer.
- Check the printed documents for correctness and quality.

Test the “Mua trang in” Page:

- Go to the “Mua trang in” (Purchase Print Pages) page.
- Complete the following actions:

- + Select your desired paper type (e.g., A4).
- + Choose the number of pages to purchase.
- + Complete the payment transaction.
- Verify the following after the transaction:
 - + The remaining pages have been updated correctly.
 - + The account balance reflects the transaction accurately.

Test the “**Lịch sử in ấn**” Page

- Access the “Lịch sử in ấn” (Print History) page.
- Verify if the printing request you made in the “In tài liệu” step is:
 - + Accurately recorded.
 - + Displaying all relevant details (e.g., file name, printing preferences, timestamp).

Test the “**Lịch sử giao dịch**” Page

- Navigate to the “Lịch sử giao dịch” (Transaction History) page.
- Verify if the page correctly registers the purchase you made in the “Mua trang in” step.
- Check if the transaction details (e.g., amount paid, number of pages purchased, payment method) are displayed accurately.

5.2.4.2 Testing as an Admin

Log In as an Admin

- In the guests' homepage, select the login button to switch to the login page.
- Select “Quản Trị Viên Hệ Thống” to login as admin.

Test the Side Bar

- After logging in, we will be redirected to the “Trang chủ” (Home Page). In there, we can see the recent print request, income of the month, number of printers and how many times have students printed in this month.
- Begin browsing the pages using the side bar:
 - + Verify that each menu item is clickable and leads to the correct page.
 - + Note any misalignment, unclear labels, or confusing navigation patterns.

Test the “**Quản Lý Máy In**”

- Access the “Quản Lý Máy In” page
- Check the information about each column and test the “Thêm Máy In” button.

Test the “**Quản Lý Người Dùng**”

- Access the “Quản Lý Người Dùng” page
- Check the information of each column.

Test the “Lịch Sử In Án”

- Access the “Lịch Sử In Án” page.
- Try the action column and check the information.

Test the “Lịch Sử Giao Dịch”

- Access the “Lịch Sử Giao Dịch” page.
- Try the action column and check the information.

5.2.5 Feedback from testers

Pages	Strengths	Weaknesses	Suggestions
“Trang chủ” (Student)	- Clear font size and well-organized layout that feels professional and easy to navigate.	- Navigation bar text alignment feels slightly off when hovering over items like "Thông tin tài khoản."	- Align text slightly to the left for better balance.
		- The search bar seems unnecessary and adds potential clutter to the interface.	- Consider removing the search bar if it does not serve a specific purpose.
		- The student's field of study (Ngành học Sinh viên) is abbreviated, which may confuse first-time users.	- Use full names for fields of study to ensure clarity.
“Thông tin tài khoản”	- Logical grouping of user data makes it easy to find information.	- Navigation bar placement is inconsistent with the Home Page design, causing confusion.	- Make the navigation bar layout consistent across pages (e.g., match the Home Page layout).
		- Personal information (name, date of birth, student ID) lacks proper spacing, making it harder to read.	- Add more spacing between lines for better readability.

		<ul style="list-style-type: none"> - Fields like Student ID and Field of Study are scattered across multiple sections. 	<ul style="list-style-type: none"> - Group Student ID and Field of Study with name and date of birth for a more logical structure.
		<ul style="list-style-type: none"> - The Edit button under account balance and printing balance is misleading, as these balances cannot be edited. 	<ul style="list-style-type: none"> - Remove the Edit button for account and printing balances. Instead, add Edit buttons for email and password for consistency.
“In tài liệu”		<ul style="list-style-type: none"> - Provides clear functionality for selecting printers and setting preferences. 	<ul style="list-style-type: none"> - Inconsistent spacing between elements in the Settings → Choose Printer section when a “-” symbol is used.
		<ul style="list-style-type: none"> - The tick mark for selecting printers lacks visual distinction. 	<ul style="list-style-type: none"> - Use a darker green color for the tick mark to make selections more noticeable and satisfying.
“Mua trang in”		<ul style="list-style-type: none"> - Straightforward interface for purchasing pages. 	<ul style="list-style-type: none"> - Limited payment options may frustrate users who lack access to the available methods.
		<ul style="list-style-type: none"> - Labeling in the Invoice section (“Giấy in A4”) is unclear and may confuse users. 	<ul style="list-style-type: none"> - Rename “Giấy in A4” to “Number of A4 pages” for better clarity.
		<ul style="list-style-type: none"> - The lack of variety in paper types limits the service's appeal to users with diverse needs. 	<ul style="list-style-type: none"> - Introduce more paper types (e.g., heavier paper for formal documents) to meet different requirements.

“Trang chủ” (Guest)	Clear font-size, good alignment, easy to interact.	The notification icon and search bar is not necessary	Remove the bell icon which is for notification, and remove the search bar. The search bar and the notification should only appear in the Student Home page.
“Trang chủ” (Admin)	The sections are set orderly, the font-size is clear and	The Notification is not necessary for the admin interface	Remove the notification.
“Quản Lý Máy In”	Well aligned header	The information in the table is not completed yet.	Complete the information in the table.
“Lịch Sử In Án”	Well aligned header	The information in the table is not completed yet.	Complete the information in the table.
“Lịch Sử Giao Dịch”	Well aligned header	The information in the table is not completed yet.	Complete the information in the table.
Overall Thoughts	- Clear layout and logical design create a solid foundation for user interactions.	- Inconsistencies in navigation and spacing can disrupt the overall user experience.	- Address inconsistencies in navigation and spacing to provide a seamless experience.

		<ul style="list-style-type: none">- Limited flexibility (e.g., payment options, paper types) may not cater to all users' needs.	<ul style="list-style-type: none">- Expand customization options (payment methods, paper types) to better meet user expectations.
--	--	---	---