Software Requirements Specification

for

A smart printing service for students at HCMUT

Version 1.0 approved

Prepared by:

1. Nguyễn Trương Thái Bảo – 2210251
2. Nguyễn Truyền Tuấn – 2213795
3. Nguyễn Tuấn Huy – 2211253
4. Nguyễn Văn Nhật Huy – 2211254
5. Phạm Bá Nhật Khang – 2211460

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

17/9/2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
All members	22/09/2024	Complete Requirement elicitation (Task 1.1, 1.2)	1.0
All members	29/09/2024	Complete Use-case Diagrams (Task 1.3)	1.0
All members	19/10/2024	Complete System Modeling (Task 2)	1.0
All members	08/11/2024	Complete Architecture Design (Task 3)	1.0

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

1.1 Domain Context

The Ho Chi Minh City University of Technology (HCMUT) is developing the HCMUT Student Smart Printing Service (HCMUT-SSPS) to provide a convenient and efficient solution for students to print their documents across various campuses. The service integrates multiple printers, a web-based application, a mobile app, and a management system, streamlining the printing process, reducing costs, and enhancing the overall student experience. Aimed at improving academic services and convenience, HCMUT-SSPS will enable students to print documents quickly and efficiently for their academic and research needs, offering a fast, time-saving printing service within the campus.

1.2 Stakeholders and Needs

- **HCMUT Students (Users):** As the primary users, students need a convenient, fast, and reliable way to print their academic documents across different campuses. They require easy access to printers through a web or mobile app, the ability to upload documents, choose printer settings, and track their printing history. Students also need a system that manages their printing quotas and allows for seamless online payments when additional pages are needed.
- **Student Printing Service Officers (SPSO):** These administrators manage the printing system, including the configuration of printers and student accounts. They need access to logs of student print jobs, the ability to adjust settings like allowed file types, manage printer statuses, and oversee the allocation of printing quotas. They also require automatic generation of reports on system usage for administrative purposes.
- **HCMUT Managers:** The university leadership requires the system to improve overall student services, enhance campus efficiency, and optimize resource usage. They need the system to be cost-effective, reduce administrative burden, and provide insights into printing system usage to make informed decisions regarding resource allocation and improvements.

1.3 Benefits of the System

- **For HCMUT Students:** The system provides a fast, convenient, and efficient way to print academic documents, saving time by allowing them to select printers and configure print settings directly through a web or mobile app. It also helps them manage their printing quotas easily and make online payments when necessary. Furthermore, students can track their printing history and usage, ensuring transparency and control over their print jobs.

- **For Student Printing Service Officers (SPSOs):** The system simplifies the management of printers and printing services. SPSOs benefit from having centralized control over printer configurations, monitoring student print activity, and accessing detailed reports on system usage. This automation reduces manual work and enables better service oversight and resource management.
- **For HCMUT Managers:** The system enhances overall campus service efficiency, providing a streamlined printing solution that reduces costs and administrative overhead. It also helps the university improve the student experience by offering a modern and user-friendly service, while the data collected from the system can be used to optimize printer distribution and resource allocation, ultimately enhancing campus operations

1.4 Functional Requirements

- **Students:**
    - Students are able to upload document files onto the system.
    - Students can choose which printer they want to print documents.
    - Students are allowed to customize printing properties such as paper size, number of printing pages, one-/doubled-sized, number of copies, etc.
    - Students can view their printing log for time period and information about the number of printed pages for each page size.
    - Students can buy more printing pages through the Buy Printing Pages feature of the system.
    - Students are able to pay for buying more printing pages through an online payment system like the BKPay system of the university.
    - A student will be not allowed to print if the printing action exceeds his/her account balance.
- **Student Printing Service Officer (SPSO):**
    - SPSO can configure file types allowed to be printed.
    - SPSO can modify the default number of pages.
    - SPSO can set the dates to update the default number of pages to all students.

  - SPSO is able to view the printing log of all students or a student for specified time (date to date) and for some printers or all of them.
  - SPSO can add a printer to the system.
  - SPSO can disable or enable a particular printer.
  - SPSO is able to view the activity reports of the system.
- **General requirements:**
  - The system should log all the printing actions (student ID, printer ID, file name, printing start and end time, number of pages for each page size) of all students.
  - The system should automatically generate activity reports at the end of each month and each year.
  - The reports should be stored in the system.
  - All users must be authenticated by HCMUT_SSO authentication before using the system.

1.5 Non- Functional Requirements

- **Speed:**
  - The system must ensure a response time of no more than 3 seconds for file uploads, submitting print requests, and printer selection.
  - Printing logs and history must be displayed to users within 3 seconds of their request.
  - Monthly and yearly reports must be automatically generated at the end of each period and should be accessible for viewing within 5 seconds upon request by the SPSO.
  - The system must be capable of handling at least 1,000 concurrent users without performance degradation.
- **Size:**
  - Each document uploaded must not exceed 100 MB in size, with total system storage scalable to 50 terabytes of documents and logs.
  - The system must be able to store printing logs for each student for at least 4 years (or until they graduate) without performance degradation.
- **Ease of Use:**
  - Users and SPSO can access the web-based app at any time.

    - The web-based app must adapt its user interface based on the screen size and orientation of the device.
    - The web-based app must function seamlessly across popular web browsers (Chrome, Safari, Firefox, and Edge) without requiring additional plugins.
- **Reliability:**
    - The printing service should be available from 7:00 to 20:00 on every working day (Monday to Friday) and from 7:30 to 17:00 on Saturday.
    - The system must operate properly (no errors, no crashes, no failures) during working days from Monday to Saturday. If there is an error from the printers, it will automatically try to restart up to three times.
- **Robustness:**
    - The system must maintain data integrity and accuracy in logging student activities and printer usage.
    - The system must back up data regularly, with a backup schedule of at least once daily to prevent data loss.
- **Portability:**
    - Printers must be placed in each faculty office and in the library.
    - Each faculty office must have at least one printer, and the library must have at least five.
    - The web-based app must be designed to be accessible from a variety of devices, including desktops, laptops, tablets, and smartphones.

2. Use-case Diagrams (1.3)

2.1 Use-case Diagram for the Whole System

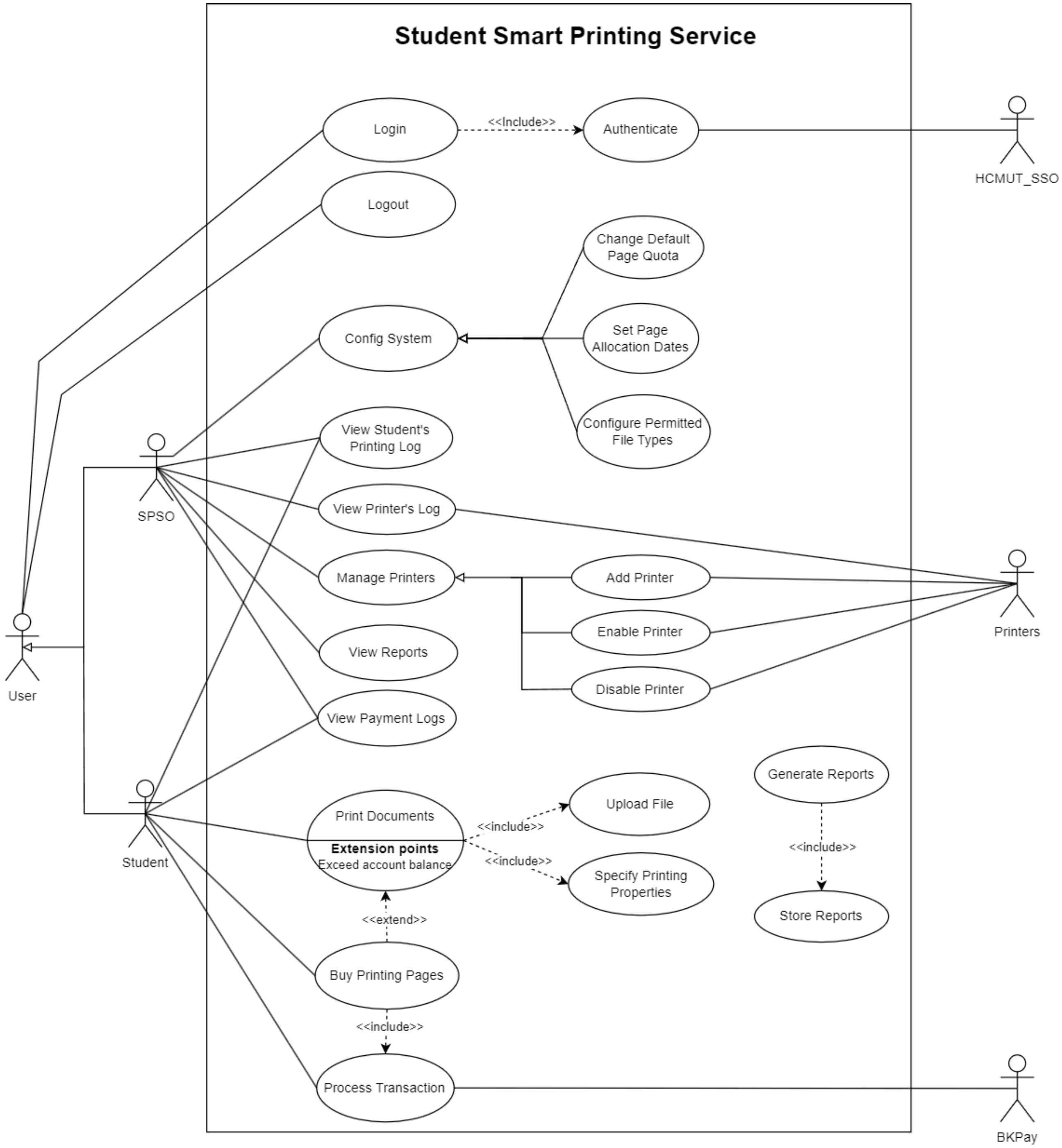


Figure 1. Use-case Diagram for the Whole System

2.2 Use-case Diagram for Printing Module

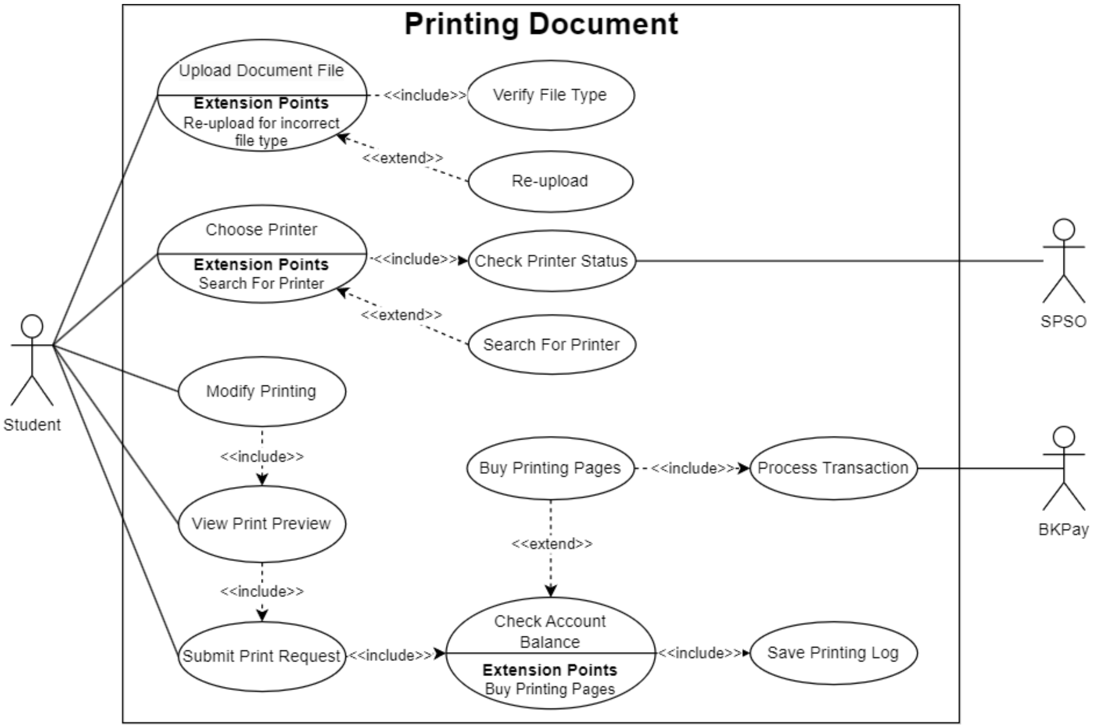


Figure 2. Use-case Diagram for Printing Module

2.3 The Details of Usecases in Printing Document Module

1. Usecase Upload Document File

Use Case ID	UC-PD01		
Use-case name	**Upload Document File**		
Created by	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Updated**	29/9/2024
Actor	Students		
Trigger	Student want to upload document file for printing		
Description	Student select file(s) on his device and upload them to the system for printing		

Preconditions	- The student has logged into the system - The student's device can connect to the internet and the system
Postconditions	A completely new file is uploaded to the system
Normal Flow	1) The student selects the "Upload file" button 2) The system opens the interface to select the file to upload 3) The student selects the file to upload 4) The student clicks the confirm button
Alternative Flows	1) In step 2.1, the student can access the web interface provided by the printing system and upload files directly from their computer or cloud storage (e.g., Dropbox, Google Drive, etc.) 2) In step 2.2, the student can drag and drop the file into the designated area on the web interface for a quick upload
Exceptions	Exception 1: At step 5, the file upload fails because the file size is too big

2. Usecase Re-upload

Use Case ID	UC-PD02		
Use-case name	**Re-upload**		
Created By	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Update**	29/9/2024
Actor	Students		
Trigger	The student wants to re-upload the desired file. Or the system requires re-uploading the file when he uploads the incorrect file type.		
Description	Select a new file and upload it to the system for printing		
Preconditions	- The student has logged into the system - The student's device can connect to the internet and the system		
Postconditions	A completely new file is uploaded to the system.		
Normal Flow	1) The student selects the "Re-upload" button		

	2) The system opens the interface to select the file to upload 3) The student selects the file to upload 4) The student clicks the confirm button
Alternative Flows	After step 2, the system displays 3 buttons: "Select," "Cancel," and "Upload new file": 1) In step 2.1, the student clicks "Cancel" if they cannot find the file to print. 2) In step 2.2, the student clicks "Upload new file" to upload a new file.
Exceptions	None

3. Usecase Verify File Type

Use Case ID	UC-PD03		
Use-case name	**Verify File Type**		
Created by	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Updated**	29/9/2024
Actor	Students		
Trigger	The file upload process by the student is completed and the system will verify file type.		
Description	Once the student uploads a document to the system, the system automatically initiates the "Verify File Type" use case to ensure that the uploaded file format is allowed by the system, based on the configurations set by the SPSO.		
Preconditions	- The student has selected a file to upload. - The file must be in a supported format (e.g., PDF, DOCX, JPEG).		
Postconditions	A completely new file is uploaded to the system		
Normal Flow	1) The student selects the file to upload. 2) The system checks the file type. 3) If the file type is supported, the system proceeds with the upload.		

	4) If the file type is invalid, the system prompts the user to select a valid file format.
Alternative Flows	The system offers a list of supported file types before uploading, so the student is aware of acceptable formats.
Exceptions	The system rejects the file because it is of an unsupported format (e.g., an executable file or an unknown format).

4. Usecase Search For Printer

Use Case ID	UC-PD04		
Use-case name	**Search For Printer**		
Created by	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Updated**	29/9/2024
Actor	Students		
Trigger	The student needs to find a specific printer on campus		
Description	The student can search for a specific printer based on information such as location, printer ID, printer brand, or model		
Preconditions	- The student has successfully logged into the system - The student has uploaded the document to be printed to the system		
Postconditions	The student has found a suitable printer		
Normal flow	1) The student selects the "Search printer" option if they need to find a suitable printer on campus 2) The system displays a search interface with multiple search criteria, such as location, printer ID, and wait time 3) The student enters the search criteria to find a printer 4) The system displays search results based on the student's entered criteria and provides a list of suitable printers 5) The student reviews the results in the list and selects the most appropriate printer		
Alternative flow	None		

Exceptions	Exception 1: At step 4, a system error occurs during the search process. The system displays an error message and asks the student to try again Exception 2: At step 4, the student cannot find a suitable printer in the list. The system displays a message stating that no printer matches the search criteria and suggests adjusting the search criteria

5. Usecase Choose Printer

Use Case ID	UC-PD05		
Use-case name	**Choose Printer**		
Created by	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Updated**	29/9/2024
Actor	Students		
Trigger	The student has uploaded the document to the system and needs to choose a printer		
Description	The student can select a printer from a list of available printers to print their document		
Preconditions	- The student has successfully logged into the system - The student has uploaded the document to be printed to the system		
Postconditions	- The student has successfully chosen a printer - The system confirms that the printer has been selected		
Normal flow	1) The system displays an interface with a list of available printers for the student 2) The student reviews the printer information from the list 3) The student selects a printer from the list 4) The student confirms the selected printer 5) The system records the student's selection		
Alternative flow	None		

Exceptions	Exception 1: At step 2, a system error occurs while displaying the list of available printers. The system displays an error message and asks the student to try again Exception 2: At step 3, the selected printer is already in use or encounters an error. The system notifies the student and offers options to either choose another printer or cancel the print job

6. Usecase Check Printer Status

Use Case ID	UC-PD06		
Use-case name	**Check Printer Status**		
Created by	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Updated**	29/9/2024
Actor	Students, SPSO		
Trigger	The student or SPSO wants to know the current status of a specific printer		
Description	The student or SPSO can check the status of a printer		
Preconditions	The student or SPSO has successfully logged into the system		
Postconditions	The student or SPSO has obtained the current status information of the selected printer		
Normal flow	1) The student selects the "Check printer status" option to view the current status of a specific printer 2) The system displays an interface with a list of printers that the student or SPSO can check 3) The student or SPSO selects a printer from the list to check 4) The system then displays the current status of the printer 5) Based on the status information, the student can decide which printer to use, or SPSO can proceed with maintenance or repairs 6) If the student or SPSO determines the printer to be used, they can select "Choose printer" to interact with the chosen printer		
Alternative flow	None		

Exceptions	Exception 1: If a system error occurs while retrieving the status information, the system displays an error message and asks the student or SPSO to try again

7. Usecase Modify Printing

Use Case ID	UC-PD07		
Use-case name	**Modify Printing**		
Created By	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Updated**	29/9/2024
Actor	Students		
Trigger	Student want to modify some printing properties		
Description	The student can change the printing properties (such as paper size, pages to print, number of copies, or one-/double-sided printing).		
Preconditions	1) The system is operational 2) The database is connected to SSPS 3) Internet connection is available 4) The student is successfully logged in and authenticated 5) The student has uploaded the file to be printed		
Postconditions	None		
Normal flow	1) The student selects the file for which they want to modify the print settings 2) The student selects the "Modify printing" button 3) The system displays a modification dialog box 4) The student updates and selects the desired print settings 5) The student clicks the "Finish" button to complete the modification process		
Alternative flow	1) In the step 5.1: After step 5, if the student wants to modify the settings again, the student selects the "Modify printing" button to continue adjusting the settings		

	2) In the step 5.2: The student selects the "Refresh" button, and the system displays a new modification dialog box, allowing the student to make further adjustments
Exceptions	None

8. Usecase View Print Preview

Use Case ID	UC-PD08		
Use-case name	**View Print Preview**		
Created By	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Updated**	29/9/2024
Actor	Students		
Trigger	The student wants to see how the document will look based on the selected printing settings		
Description	The student previews the print layout before submitting a print request and the system then generates and displays a visual representation of the document based on the selected printing options		
Preconditions	1) The system is operational 2) The database is connected to SSPS 3) Internet connection is available 4) The student is successfully logged in and authenticated 5) The student has uploaded the file to be printed		
Postconditions	None		
Normal flow	1) The student selects the file to preview 2) The student clicks the "Preview" button 3) The system presents the print preview 4) The student clicks the "Print" button to print the file		
Alternative flow	1) In step 1, the student can select a file that has already been uploaded or click "Preview" after completing the **Modify Print Settings**		

	2) After step 3, the system displays two buttons for selection: "Edit" and "Print": 3.1. The student selects the "Edit" button to modify the print settings 3.2. The student selects the "Preview" button to review the print layout again 3.3. The student selects the "Print" button to print
Exceptions	None

9. Usecase Buy Printing Pages

Use Case ID	UC-PD09		
Use-case name	**Buy Printing Pages**		
Created By	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Updated**	29/9/2024
Actor	Students, BK Pay		
Trigger	The student attempts to submit a print request but discovers an insufficient balance of printing pages		
Description	The student can purchase additional printing pages using BK Pay		
Preconditions	1) The system is operational 2) The database is connected to SSPS 3) Internet connection is available 4) The student is successfully logged in and authenticated 5) The student has uploaded the file and selected to print it		
Postconditions	None		
Normal flow	1) The system notifies the student of the number of additional pages needed for printing 2) The student selects the "Buy more pages" button 3) The system presents a dialog box for purchasing pages and informs the student of the required payment amount		

	4) The student enters the necessary information as required by the system to complete the transaction 5) The student selects the "Confirm" button to finalize the transaction
Alternative flow	After step 1, the system displays two buttons: "Buy more pages" and "Cancel": 2.1. The student selects the "Cancel" button to cancel the print job
Exceptions	None

10. Usecase Submit Print Request

Use Case ID	UC-PD10		
Use-case name	**Submit Print Request**		
Created by	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Update**	29/9/2024
Actor	Students		
Trigger	The student wants to send the print job request to the system		
Description	The student submits a request to print a document that has been successfully uploaded to the system.		
Preconditions	- The student has successfully uploaded a document to the system. - The student is logged into the system.		
Postconditions	A print request is submitted. Ready to check account balance.		
Normal Flow	1) The student selects the "Submit" button. 2) The system confirms the print request has been submitted successfully.		
Alternative Flows	None		
Exceptions	None		

11. Usecase Check Account Balance

Use Case ID	UC-PD11		
Use-case name	**Check Account Balance**		
Created By	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Update**	29/9/2024
Actor	Student		
Trigger	The student wants to verify their remaining printing pages or buying additional pages		
Description	Students check account balance and BK Pay ensures sufficient funds for printing or other services.		
Preconditions	- The student is logged into the system. - The student's account contains information regarding available funds.		
Postconditions	The system displays the student's account balance and notify students whether they have enough money to make the transaction or not.		
Normal Flow	1) The student selects the "Check Balance" button. 2) The system retrieves the account balance from the database. 3) The system displays the account balance to the student.		
Alternative Flows	None		
Exceptions	None		

12. Usecase Process Transaction

Use Case ID	UC-PD12		
Use-case name	**Process Transaction**		
Created By	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Updated**	29/9/2024

Actor	Students, BKPay
Trigger	The student requests to buy printing pages, prompting the system to process the payment.
Description	This use case describes the process of purchasing printing pages by interacting with the payment system (BKPay).
Preconditions	1) The student must have selected the option to buy printing pages. 2) The system must be connected to the BKPay payment gateway.
Postconditions	The printing pages are added to their account.
Normal flow	1) The student initiates the process to buy printing pages. 2) System checks the student's account balance via BKPay. 3) BKPay processes the payment transaction. 4) The system confirms the transaction and updates the account balance.
Alternative flow	In step 2, if there are insufficient funds, the student is prompted to add more funds.
Exceptions	Exception 1: If the payment fails, the system notifies the student, and the transaction is not processed.

13. Usecase Save Printing Log

Use Case ID	UC-PD13		
Use-case name	Save printing log		
Created By	Tuấn Huy	**Last Updated By**	Nhật Huy
Date Created	25/9/2024	**Date Last Updated**	29/9/2024
Actor	SPSO		
Trigger	The system successfully processes a print request		
Description	This use case describes the logging of print job information for future reference or auditing.		

Preconditions	The print job must have been successfully submitted
Postconditions	The print job information is saved in the system logs
Normal flow	1) After the student submits a print request, the system completes the printing process. 2) The system records the details of the print job (e.g., file name, pages printed, timestamp, printer used). 3) The print log is saved in the system for future auditing or user reference. 4) The system sends a confirmation to the student indicating that the print job was successful and the log has been saved.
Alternative flow	In step 1.1, if the print job fails, the system logs the error details.
Exceptions	None

3. Task 2: System Modeling

3.1 Activity Diagrams

3.1.1 Printer selecting diagram

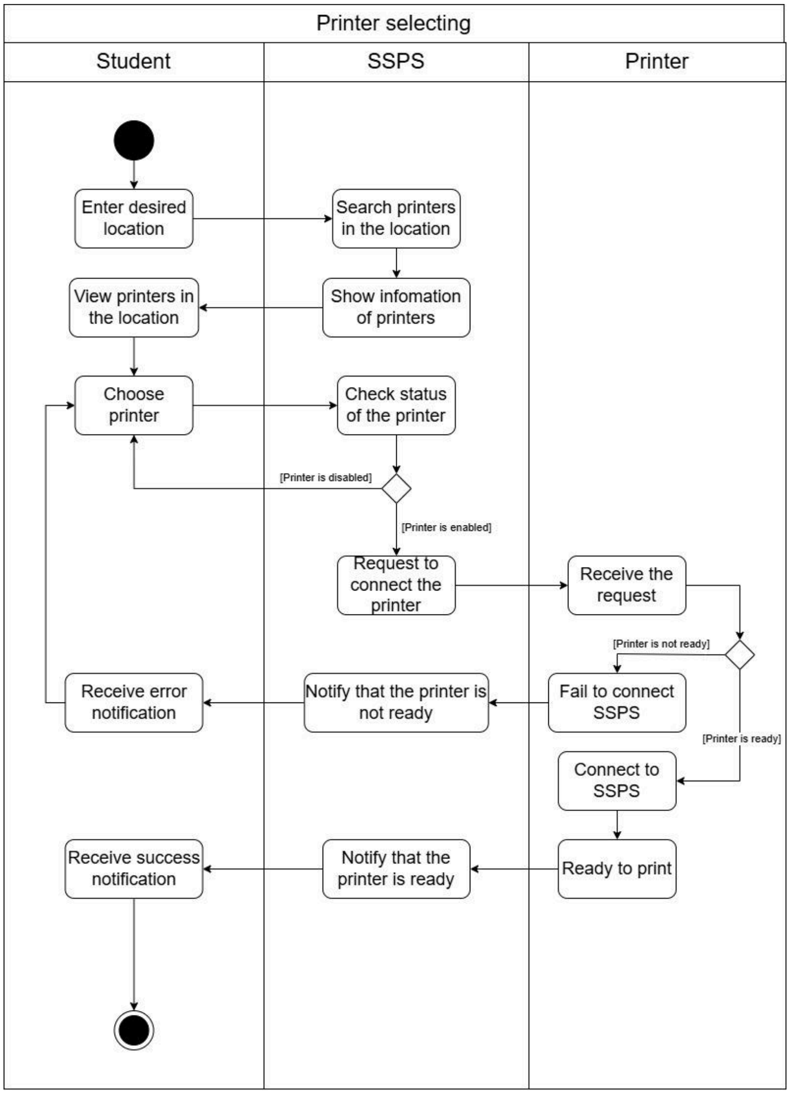

Printer selecting
Student
SSPS
Printer
Enter desired location
Search printers in the location
View printers in the location
Show infomation of printers
Choose printer
Check status of the printer
[Printer is disabled]
[Printer is enabled]
Request to connect the printer
Receive the request
[Printer is not ready]
Receive error notification
Notify that the printer is not ready
Fail to connect SSPS
[Printer is ready]
Connect to SSPS
Receive success notification
Notify that the printer is ready
Ready to print

Figure 3. Printer selecting - Activity Diagram

The printer choosing diagram totally contains fifthteen steps for a student to select his/her desired printer. In the first stage, the student will have to enter an expected location and the SSPS will do a search in this area to find nearby printers. All information of the printers will be shown up to the student and then he/she has to choose a desired printer. After selecting a printer, SSPS continues checking the status of it, in case that printer has been disabled by SPSO before, the system will make the student choose another printer in the location. If the status is "enabled", the SPSO will send a request to the printer to make sure it is available to print. The printer receives the request and if there are some problems with the printer such as broken one, out of printing ink, failure transmission, etc, the SSPS cannot connect to it, then it will inform the student that the printer is not ready and make him/her choose another one. The student will receive the failure notification and change the printer. If the printer is working, it will connect to SSPS and be ready to print, then the SSPS recognizes that successful signal and notify the students that the printer choosing process is completed. The student will receive the notification and then the process ends.

3.1.2 File uploading diagram

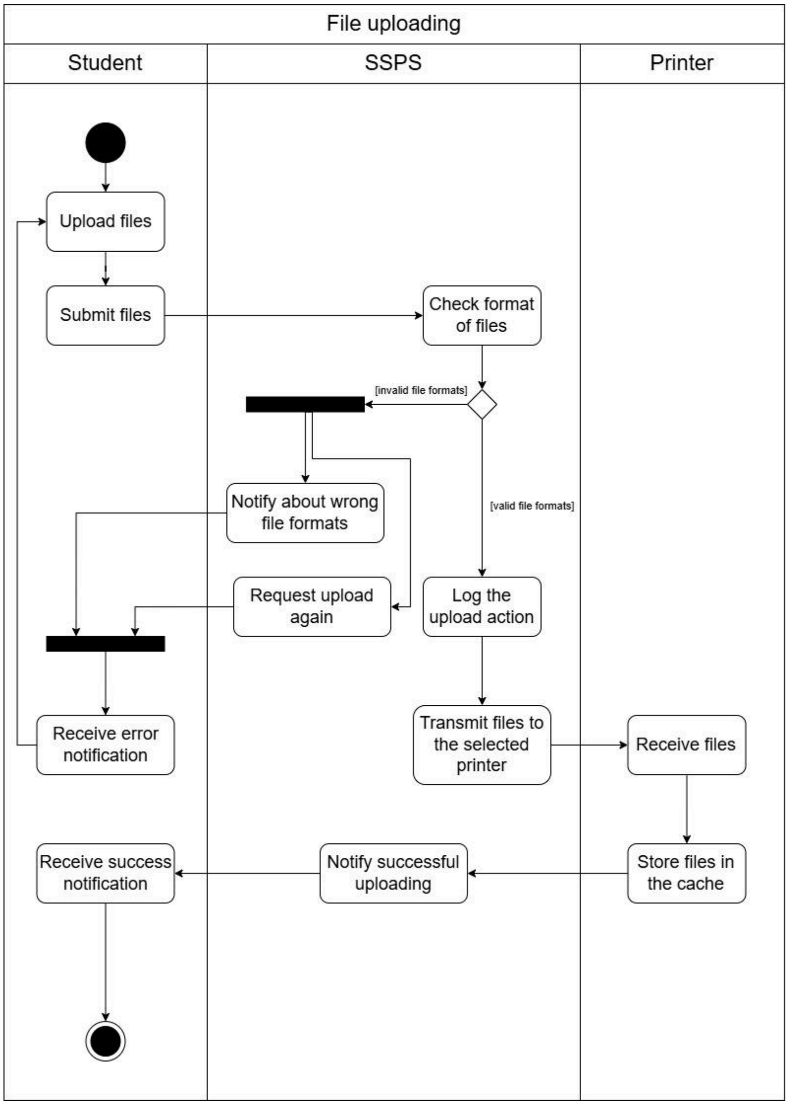


Figure 4. File uploading - Activity Diagram

The activity diagram of file uploading use case has twelve steps in the uploading procedure. Firstly, the student will start with uploading their desired files, documents to print and then they have to click the submit button to confirm the process. The SSPS system will receive the files and start to check the format of them to make sure they are valid types which are determined by the SPSO. In case of invalid formats, the SSPS will simultaneously notify the student about the errors and request he/she to upload again files with acceptable formats. The student will receive the failure notification and come back to the first stage to upload files again. If the formats are appropriate, the SSPS recognizes the uploading action of the student and logs it into the stored record, then the files will be transmitted to the selected printer which is decided by the student in the printer choosing phase. The printer receives the files, stores them in its cache and sends back to the SSPS a signal that the printer is ready to print. The SSPS recognizes the signal and informs the student that the uploading process is successfully finished, then the file uploading procedure ends.

3.1.3 Printing submission

The printing submission diagram totally consists of fifteen stages. Firstly, the user will configure the properties for the printer such as paper sizes, number of pages, one/doubled-sized. The SSPS will recognize the settings and the student has to click on the submission button to confirm these configurations. After that, the SSPS will check the account balance of the student, in case the number of printing pages exceeds the balance, it will request the user to purchase more pages. Therefore, the student has to enter his/her desired number of pages and the system will show the total amount of money needed for purchasing. To confirm the purchase, the student will send a request to the BKay and then it will handle the transaction. If the number of pages does not exceed the balance, the SSPS will send the configured properties which were determined by the user previously to the printer. The printer will receive them and start printing files with the desired settings. After the file printing stage, SSPS will log the printing action into the report and inform the student that the process is successfully completed. The student will receive this notification and the file printing process ends.

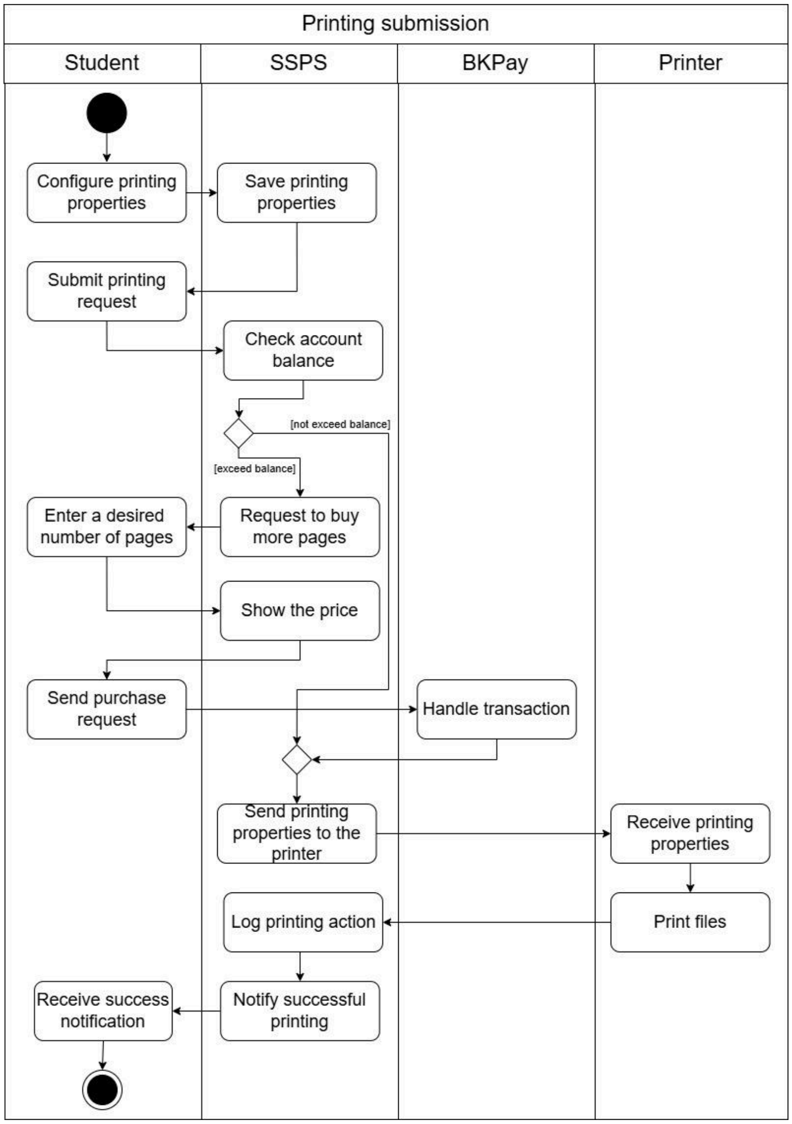


Figure 5. Printing submission - Activity Diagram

3.2 Sequence Diagrams

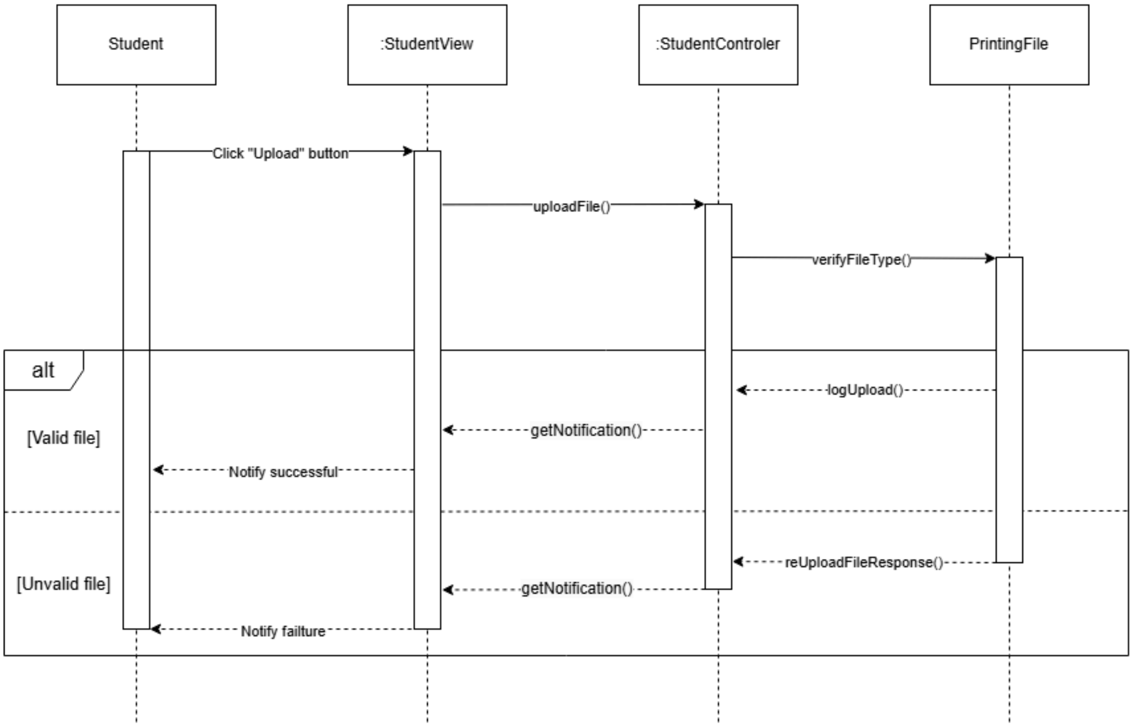


Figure 6. Upload file - Sequence Diagram

Describe upload file diagram (Figure 6):

1. The student initiates the process by clicking the "Upload" button in the StudentView.
2. The StudentView sends a request with the selected file (*uploadFile()*) to the StudentController to begin the file upload process.
3. The StudentController communicates with the PrintingFile to verify the type of the file being uploaded (*verifyFileType()*).
4. There will be 2 cases happening here:
   - If the file is valid, the system logs the upload (*logUpload()*), and the StudentController sends a notification back to the student indicating a successful upload (*getNotification()*).
   - If the file is invalid, the system will request a re-upload (*reUploadFileResponse()*) and notify the student of the failure (*getNotification()*).
5. The student receives a notification from the system about the result (either success or failure).

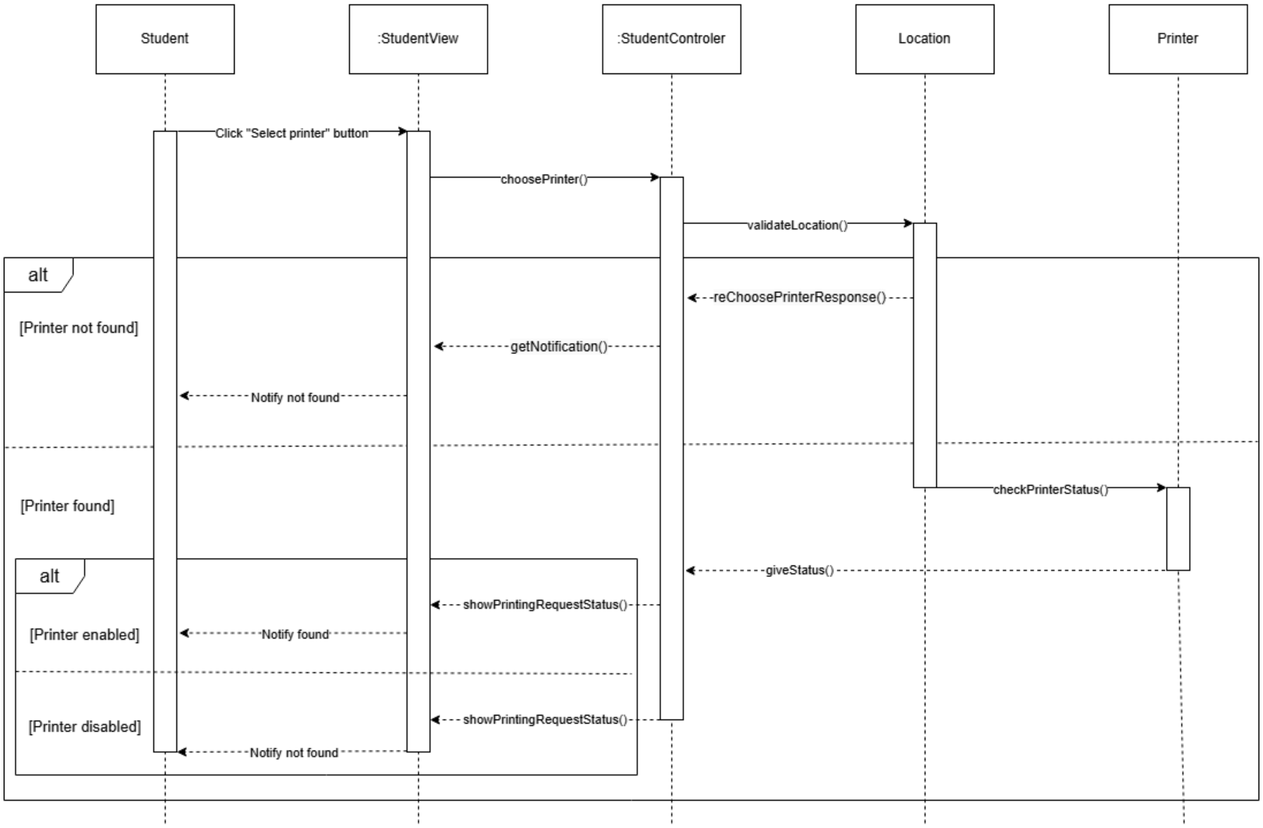


Figure 7. Choose printer - Sequence Diagram

Describe choose printer diagram (Figure 7):

1. The student clicks the "Select printer" button in the StudentView, triggering the process.
2. The StudentView sends a request (*choosePrinter()*) to the StudentController to select the desired printer.
3. The StudentController sends a request to Location to validate the location of the selected printer (*validateLocation()*).
4. There will be 2 cases happening in validate the location:
    - If the printer is not found, the system sends a "Not found" notification back to the student.
    - If the printer is found, the system checks the status of the printer (*checkPrinterStatus()*).
5. Check Printer Status: The StudentController sends a request to the Printer to check its status (checkPrinterStatus()).
6. There will be 2 cases happening in check printer status:
    - If the printer is enabled, the system notifies the student that the printer is available.

- If the printer is disabled, the system sends a notification to the student that the printer is unavailable.

7. The student receives a notification from the system about the printer status (either enable or disable).

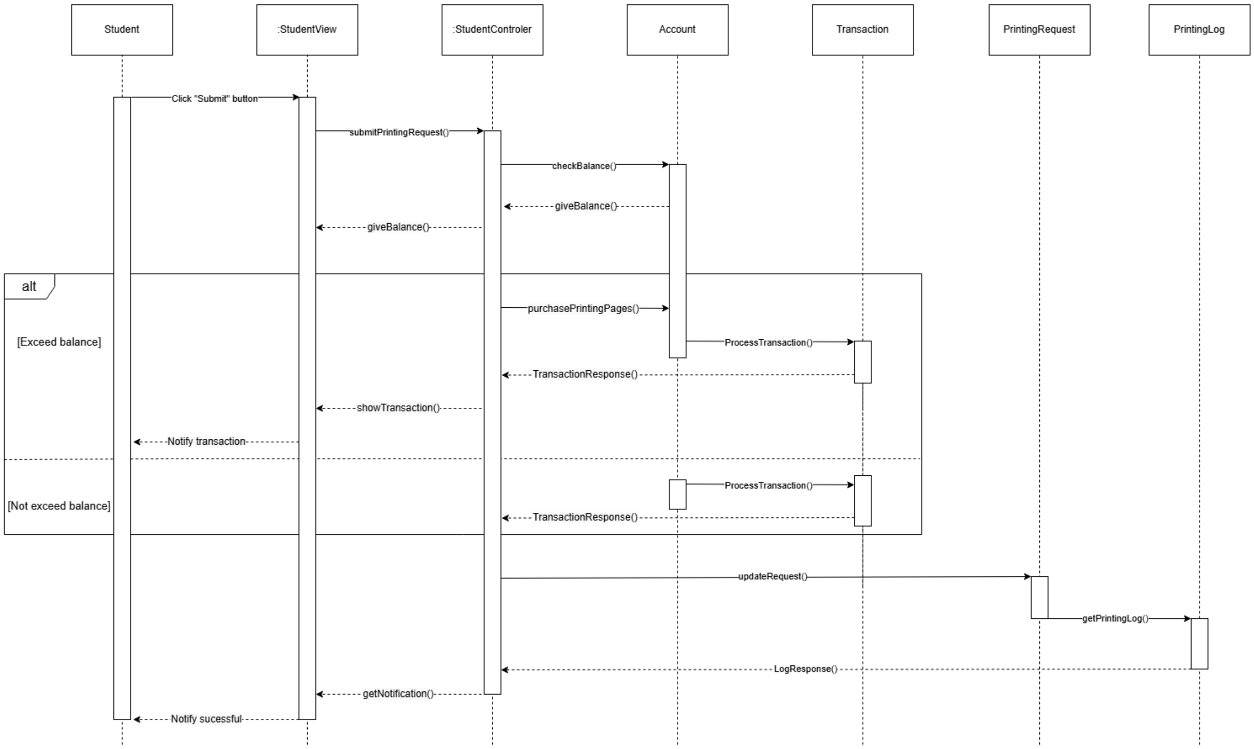


Figure 8. Submit printing file - Sequence Diagram

Describe submit printing file (Figure 8):

1. The student initiates the process by clicking the "Submit" button in the StudentView.
2. The StudentView sends a request (*submitPrintingRequest()*) to the StudentController.
3. The StudentController communicates with the Account to check the balance (*checkBalance()*).
4. There will be 2 cases happening here:
   - If not exceeding balance, the Account communicates with the Transaction and conducts transaction (*processTransaction()*).
   - If exceeding balance, the StudentController communicates with the Account to purchase pages(purchasePrintingPages). After that, the Account communicates with the Transaction and conducts transaction (*processTransaction()*).

5. The StudentController communicates with the PrintingRequest to update requests (*updateRequest()*).
6. The PrintingRequest communicates with the PrintingLog to log the requests (*getPrintinglog()*).
7. The PrintingRequest returns the response to StudentController.
8. The StudentController returns notification to StudentViewer.
9. The student receives a notification from the system about the result from StudentViewer.

3.3 Class Diagram for Whole module

Here are some key operations of the printing module *(See in Figure 9)*:

1. Submitting a printing request:
   - A Student uses the *StudentView* to access the *displayPrintingRequestForm()*.
   - The *StudentController* manages the process, allowing the student to *uploadFile(), choosePrinter()*, and *submitPrintingRequest()*.
   - The *PrintingRequest* is created with details like number of copies, file type, and printer selection.
   - Then the *PrintingRequest* is sent to the *Printer* to print.
   - *PrintingLog* records all *PrintingRequest* activities, accessible via *getPrintingLog()*.
   - Students receive updates and notifications on request status through *getNotification()* in *StudentView*.
2. Printer management:
   - *SPSOController* manages printer operations and can *refillPrinter()* when necessary.
   - *Printer* objects maintain their status, availability, and remaining pages.
   - *SPSO* staff can *checkPrinterStatus()* to ensure functionality.
3. Buying printing pages:
   - Each student owns an *Account* to access the system, each *Account* contains information about its *balance*, *currentPage* available and a list of *Transactions.*
   - A *Transaction* is created when the student wants to buy more printing pages. It contains an *ID*, the *amount* of money, *date*,...

Workflow:

- A student logs into the system, selects the files to print, and submits a request.
- The system checks the student's account current printing page availeble.
- There are 2 cases that can happen:

  - ○ If the account does have enough pages, the system will process the request.
  - ○ If the account does not have enough pages, the student must create a transaction to buy more printing pages before continuing printing.
- The request is sent to the chosen printer, and the system logs the details.
- Staff manage printer operations, ensuring they are ready and refilled as needed.
- The system notifies the student about the request status and any required actions (e.g., re-uploading files).

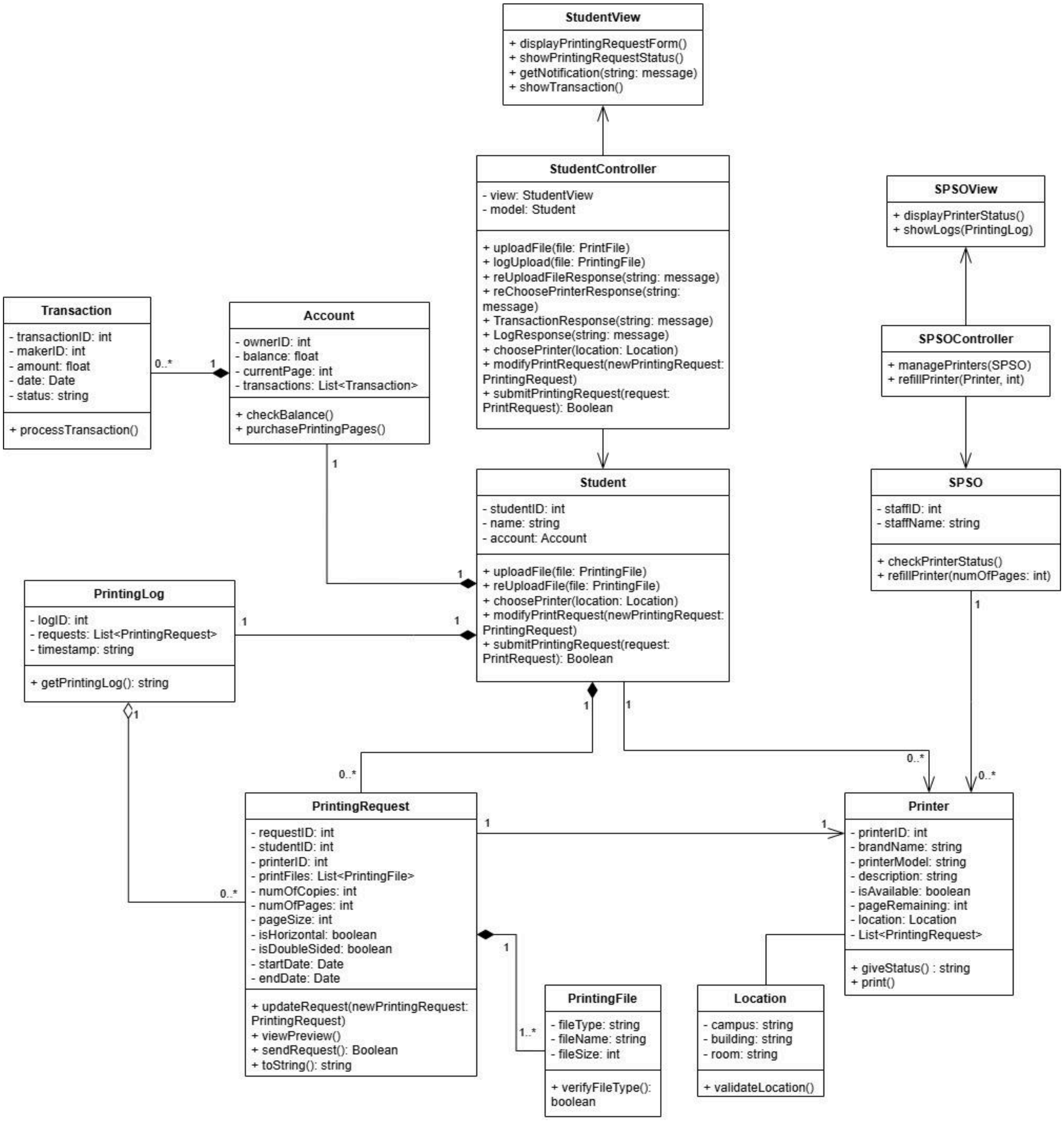
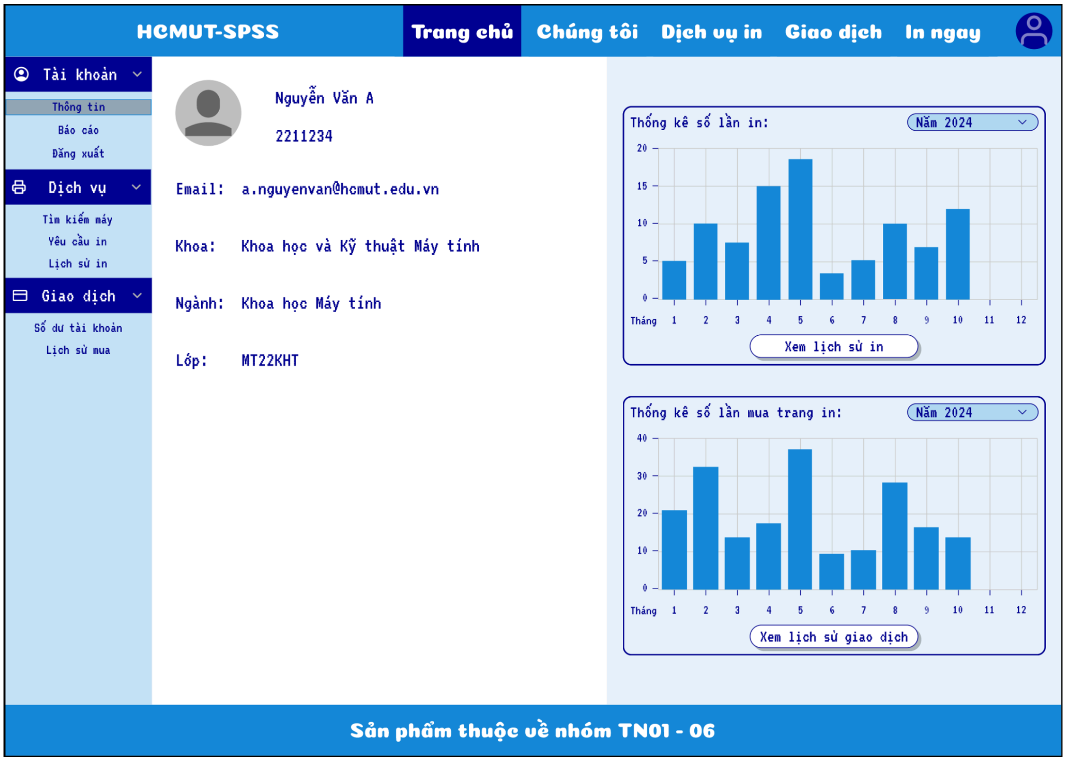


Figure 9. Class Diagram for whole Printing module

3.4 User Interfaces - MVP1



Nguyễn Văn A

2211234

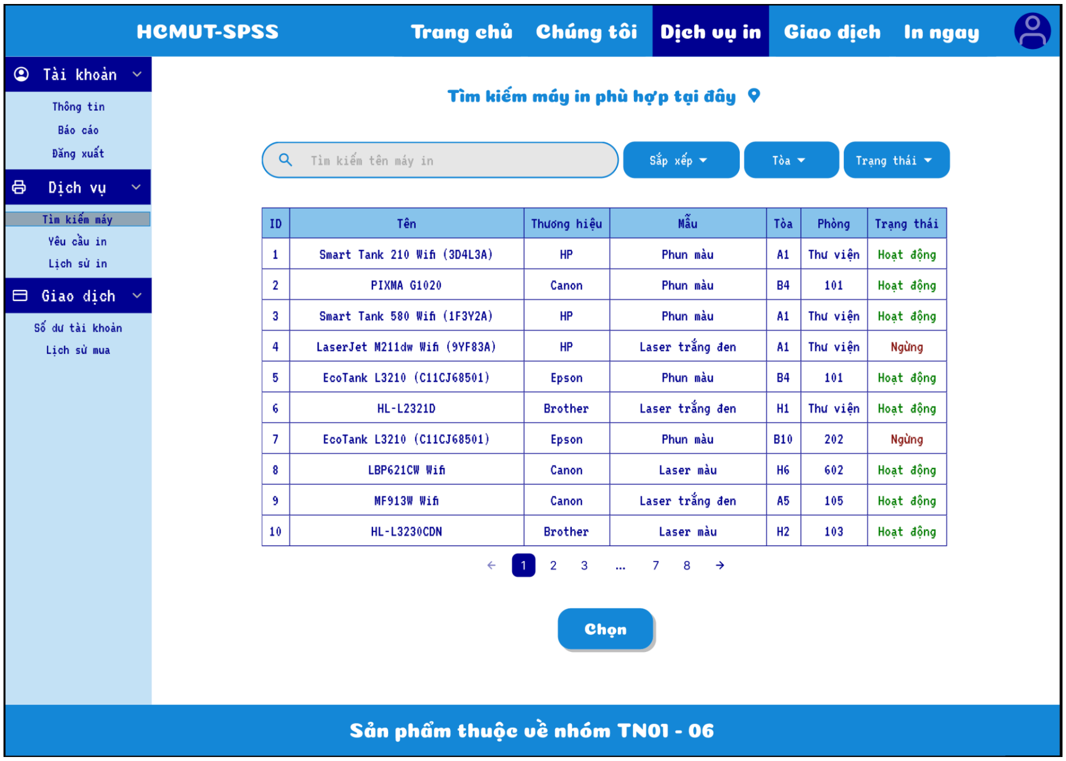
Email: a.nguyenvan@hcmut.edu.vn

Khoa: Khoa học và Kỹ thuật Máy tính

Ngành: Khoa học Máy tính

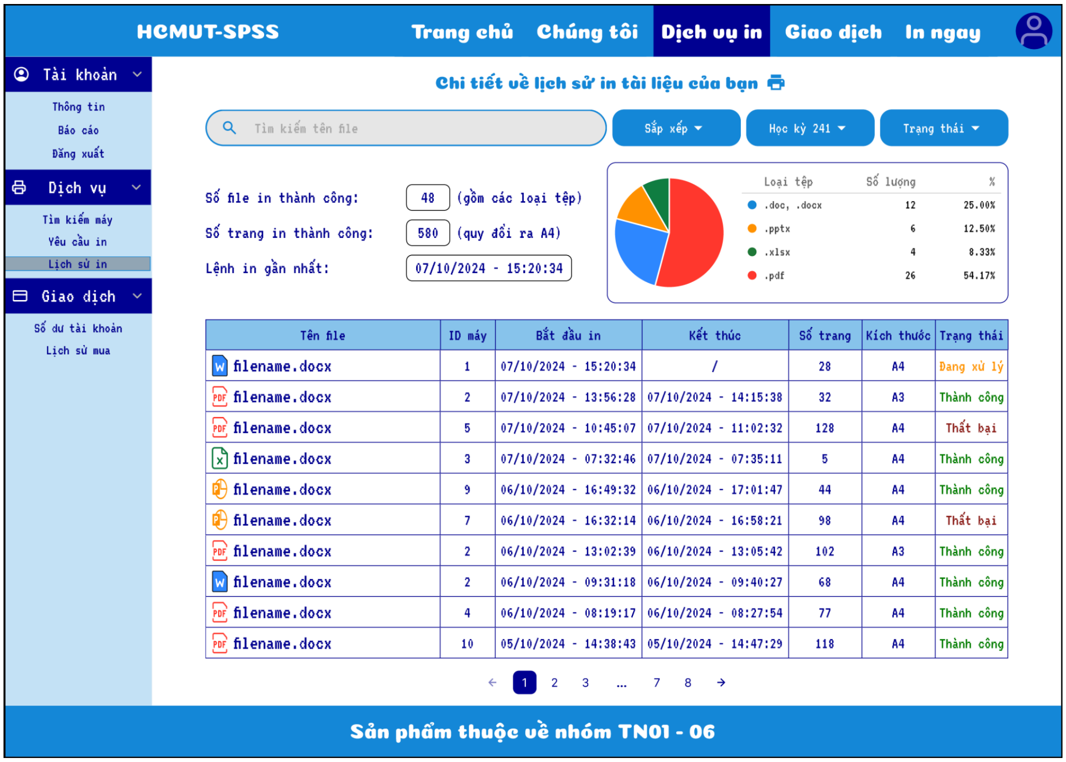
Lớp: MT22KHT

Figure 10. Account information webpage (MVP 1)



ID	Tên	Thương hiệu	Mẫu	Tòa	Phòng	Trạng thái
1	Smart Tank 210 Wifi (3D4L3A)	HP	Phun màu	A1	Thư viện	Hoạt động
2	PIXMA G1020	Canon	Phun màu	B4	101	Hoạt động
3	Smart Tank 580 Wifi (1F3Y2A)	HP	Phun màu	A1	Thư viện	Hoạt động
4	LaserJet M211dw Wifi (9YF83A)	HP	Laser trắng đen	A1	Thư viện	Ngừng
5	EcoTank L3210 (C11CJ68501)	Epson	Phun màu	B4	101	Hoạt động
6	HL-L2321D	Brother	Laser trắng đen	H1	Thư viện	Hoạt động
7	EcoTank L3210 (C11CJ68501)	Epson	Phun màu	B10	202	Ngừng
8	LBP621CW Wifi	Canon	Laser màu	H6	602	Hoạt động
9	MF913W Wifi	Canon	Laser trắng đen	A5	105	Hoạt động
10	HL-L3230CDN	Brother	Laser màu	H2	103	Hoạt động

Figure 11. Searching printer webpage (MVP 1)



Chi tiết về lịch sử in tài liệu của bạn

Số file in thành công: 48 (gồm các loại tệp)

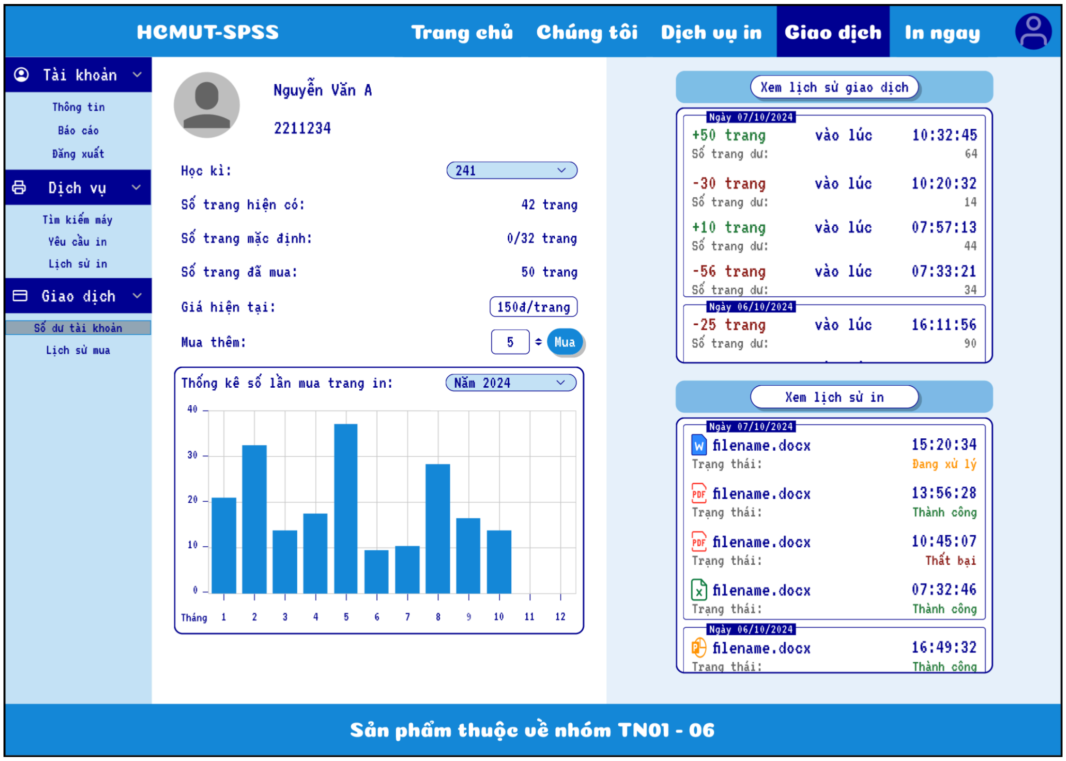
Số trang in thành công: 580 (quy đổi ra A4)

Lệnh in gần nhất: 07/10/2024 - 15:20:34

Loại tệp	Số lượng	%
.doc, .docx	12	25.00%
.pptx	6	12.50%
.xlsx	4	8.33%
.pdf	26	54.17%

Tên file	ID máy	Bắt đầu in	Kết thúc	Số trang	Kích thước	Trạng thái
filename.docx	1	07/10/2024 - 15:20:34	/	28	A4	Đang xử lý
filename.docx	2	07/10/2024 - 13:56:28	07/10/2024 - 14:15:38	32	A3	Thành công
filename.docx	5	07/10/2024 - 10:45:07	07/10/2024 - 11:02:32	128	A4	Thất bại
filename.docx	3	07/10/2024 - 07:32:46	07/10/2024 - 07:35:11	5	A4	Thành công
filename.docx	9	06/10/2024 - 16:49:32	06/10/2024 - 17:01:47	44	A4	Thành công
filename.docx	7	06/10/2024 - 16:32:14	06/10/2024 - 16:58:21	98	A4	Thất bại
filename.docx	2	06/10/2024 - 13:02:39	06/10/2024 - 13:05:42	102	A3	Thành công
filename.docx	2	06/10/2024 - 09:31:18	06/10/2024 - 09:40:27	68	A4	Thành công
filename.docx	4	06/10/2024 - 08:19:17	06/10/2024 - 08:27:54	77	A4	Thành công
filename.docx	10	05/10/2024 - 14:38:43	05/10/2024 - 14:47:29	118	A4	Thành công

Figure 12. Printing history of a user webpage (MVP 1)



Nguyễn Văn A

2211234

Học kì: 241

Số trang hiện có: 42 trang

Số trang mặc định: 0/32 trang

Số trang đã mua: 50 trang

Giá hiện tại: 150đ/trang

Mua thêm: 5

Ngày 07/10/2024

+50 trang vào lúc 10:32:45
Số trang dư: 64

-30 trang vào lúc 10:20:32
Số trang dư: 14

+10 trang vào lúc 07:57:13
Số trang dư: 44

-56 trang vào lúc 07:33:21
Số trang dư: 34

Ngày 06/10/2024

-25 trang vào lúc 16:11:56
Số trang dư: 90

Ngày 07/10/2024

filename.docx 15:20:34
Trạng thái: Đang xử lý

filename.docx 13:56:28
Trạng thái: Thành công

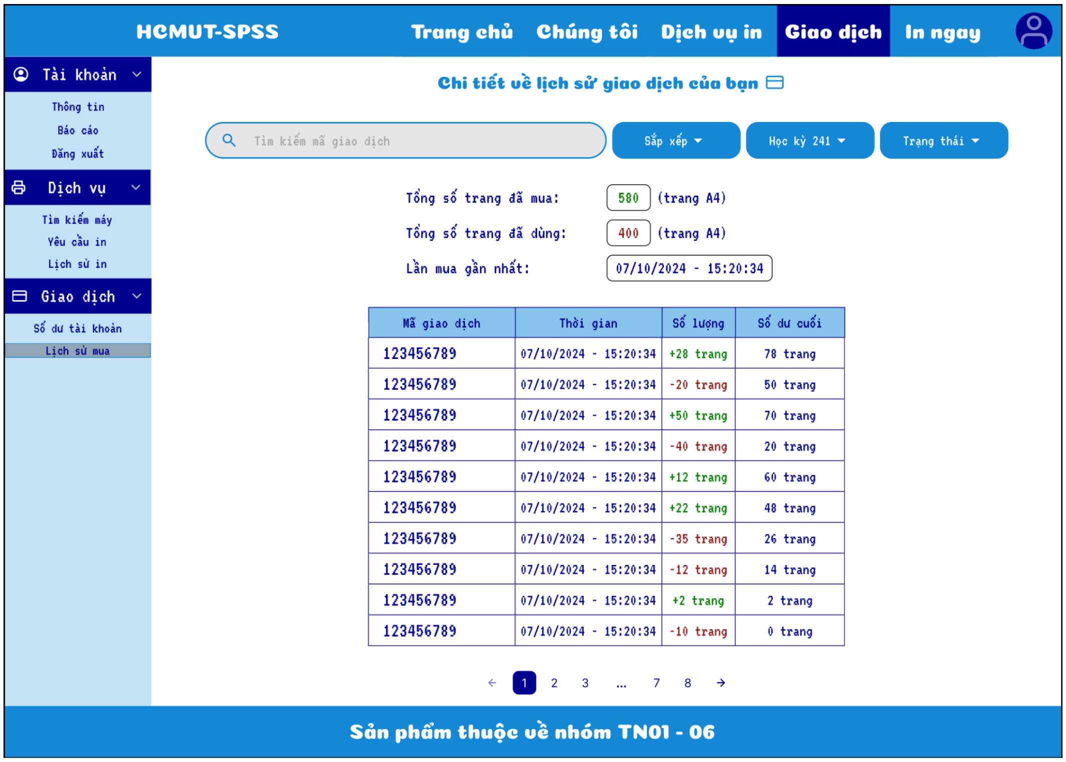
filename.docx 10:45:07
Trạng thái: Thất bại

filename.docx 07:32:46
Trạng thái: Thành công

Ngày 06/10/2024

filename.docx 16:49:32
Trạng thái: Thành công

Figure 13. Account balance webpage (MVP 1)



Chi tiết về lịch sử giao dịch của bạn

Tổng số trang đã mua: 580 (trang A4)

Tổng số trang đã dùng: 400 (trang A4)

Lần mua gần nhất: 07/10/2024 - 15:20:34

Mã giao dịch	Thời gian	Số lượng	Số dư cuối
123456789	07/10/2024 - 15:20:34	+28 trang	78 trang
123456789	07/10/2024 - 15:20:34	-20 trang	50 trang
123456789	07/10/2024 - 15:20:34	+50 trang	70 trang
123456789	07/10/2024 - 15:20:34	-40 trang	20 trang
123456789	07/10/2024 - 15:20:34	+12 trang	60 trang
123456789	07/10/2024 - 15:20:34	+22 trang	48 trang
123456789	07/10/2024 - 15:20:34	-35 trang	26 trang
123456789	07/10/2024 - 15:20:34	-12 trang	14 trang
123456789	07/10/2024 - 15:20:34	+2 trang	2 trang
123456789	07/10/2024 - 15:20:34	-10 trang	0 trang

Sản phẩm thuộc về nhóm TN01 - 06

Figure 14. Payment history webpage (MVP 1)

Interface designs are implemented on Figma *(see all screens at the following link)*.

4. Task 3: Architecture Design

4.1 Layered Architecture Diagram

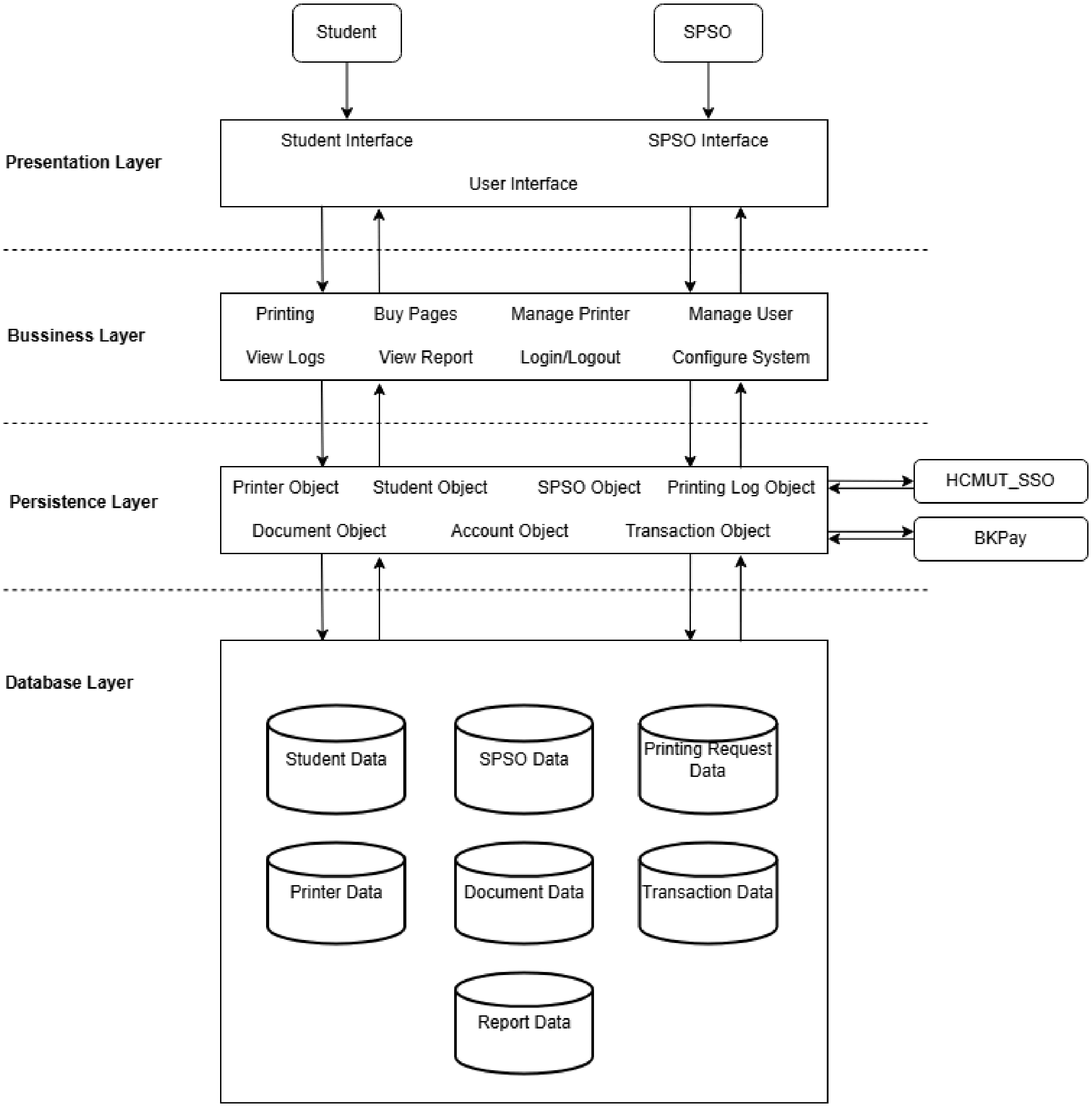


Figure 15. Box-line diagram for architecture of application

4.1.1 Presentation strategy

Here is the first layer in the architecture. We will apply a strategy focused on simplicity, ease of use, and user experience. To achieve this, we will leverage several modern and specific technologies:

- **Front-end library and framework:** We are using front-end development technology such as React. React allows us to create a flexible and efficient interface.
- **Responsive Design:** Ensuring compatibility with all devices used by students and university staff. We will integrate the system with various devices and screen sizes, incorporating flexible forms and interface components to ensure a great user experience across computers, mobile phones, and tablets.
- **User-Friendly Features:** We consider intuitive elements like buttons, forms, and easy-to-use menus, aiming to make the system easily accessible to first-time users.

With the use of React for the user interface, along with flexible design and a focus on user experience, we will create an impressive interface for the HCMUT_SPSS system while ensuring seamless integration with the system's layered architecture.

This layer is dedicated to user interface and interaction, handling user commands and displaying information to enhance user experience. It consists of two main components: the Student Interface and the SPSO Interface. Both share common elements like a Home page, which serves as the entry point when users access or log into SPSS, and a Login page, which adapts based on whether the user is a student or an administrator.

- **Students Interface:**

  The students interface contains some main pages:

  - Home page;
  - Log in;
  - Printer location;
  - Executing transaction page;
  - Printing Page.
- **SPSO Interface**

  The admin interface contains some main pages:

  - Home page;
  - Log in;
  - Dashboard page;
  - Printing page.

The Student Interface provides features tailored to student needs. It includes a Printer Location page that displays available printers and their statuses, helping students quickly locate functional, nearby printers. The Printing page allows students to upload documents, select a specific printer, and set printing options, ensuring an efficient printing process. If students need additional paper, they can seamlessly navigate to the Buying page to top up their paper balance.

The SPSO Interface, designed for SPSO, includes advanced tools for managing the BKPrint system. The Dashboard page serves as the command center, giving SPSO an overview of user activities, such as print history, printer availability, and real-time printer status. It also allows them to make immediate adjustments to printer settings. SPSO can export periodic reports from this interface, making it invaluable for monitoring system performance and usage trends.

In summary, the User Interface and Interaction layer bridges users with the system's functionalities, optimizing the efficiency and effectiveness of the printing service for both students and administrators.

4.1.2 Data storage approach

In the layered architecture, the database will be the bottom layer, responsible for storing all data and processing it. The application's data will be stored here, and operations like create, search, insert, update, and delete will be frequently performed to interact with the data through a database management system. For the Smart Printing Service project, the team will use a layered architecture, with the database layer stored in a relational database and managed by the MySQL database management system. This means that the application's data will be stored in tables and relationships between them. For the Smart Printing Service system, we need to have the following types of entities:

1. **User**
   - Attributes:
     - User_ID: Unique identifier for each user.
     - Fullname: The full name of the user.
     - Username: The username for user login.
     - Password: The password for user authentication.
   - Relationships:
     - A User can be either a Student or an SPSO.
2. **Student**
   - Attributes:
     - Balance: The current balance of the student.
   - Relationships:

    - A Student can Buy and Order printing services.
    - A Student can Send multiple Reports.

3. **SPSO (Student Printing Service Officer)**
   - This entity inherits from User and represents the operator role in the system.
4. **Transaction**
   - Attributes:
     - Transaction_ID: Unique identifier for each transaction.
     - Amount: The amount of money involved in the transaction.
     - Balance_after: The balance after the transaction.
     - Timestamp: The date and time of the transaction.
     - Type: The type of transaction.
   - Relationships:
     - A Student can perform multiple Transactions through Buy.
5. **Report**
   - Attributes:
     - Report_ID: Unique identifier for each report.
     - Header: The title or header of the report.
     - Description: Details about the report.
     - Rate: Rating or evaluation related to the report.
   - Relationships:
     - A Student can send multiple Reports.
6. **Printer**
   - Attributes:
     - Printer_ID: Unique identifier for each printer.
     - Brand_name: Brand of the printer.
     - Printer_model: Model of the printer.
     - Description: Additional details about the printer.
     - IsAvailable: Status indicating if the printer is available.
     - Page_remaining: Number of pages left for printing.

- Location Composite attribute:
  - Campus, Building, Room: Details of the printer's physical location.
- Relationships:
  - A Printer can be used for multiple Print operations.

7. **Printing_request**
   - **Attributes:**
     - Request_ID: Unique identifier for each printing request.
     - Num_of_copies: Number of copies requested.
     - Page_size: Size of the pages requested.
     - IsHorizontal: Orientation of the print (horizontal or vertical).
     - IsDoubledSide: Indicates if printing is double-sided.
     - startDate: Start date of the printing request.
     - endDate: End date of the printing request.
   - **Relationships:**
     - A Printing_request can have multiple Print operations associated with a Printer.
     - A Student can create multiple Printing_request orders.
8. **File**
   - **Attributes:**
     - File_ID: Unique identifier for each file.
     - Size: Size of the file.
     - Type: Type or format of the file.
     - Name: Name of the file.
   - **Relationships:**
     - A File is associated with a Printing_request.

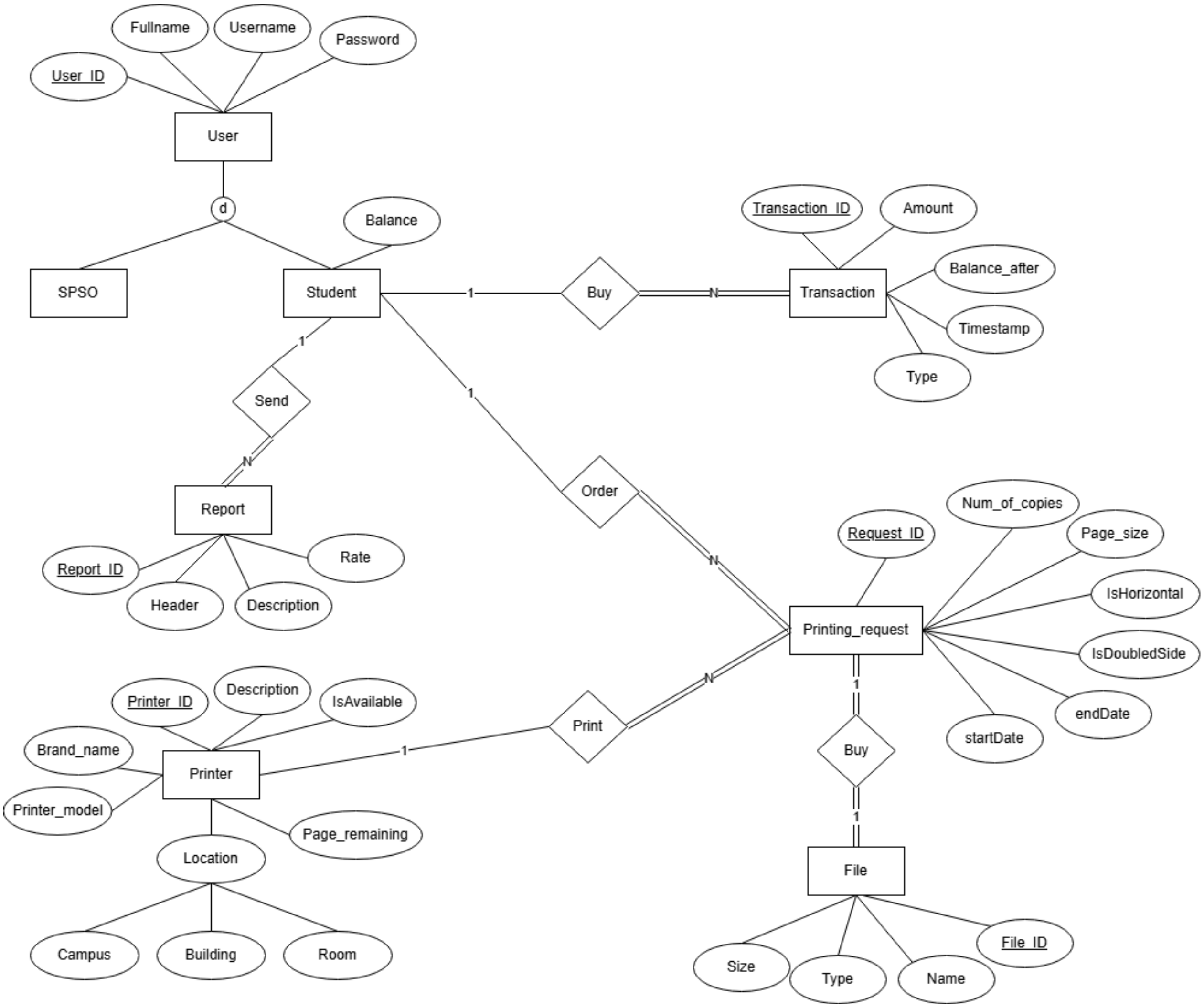


Figure 16. (E)ERD design for database system

4.1.3 API management

APIs (Application Programming Interfaces) are methods and protocols for connecting with libraries and other applications. APIs provide access to a set of commonly used functions, allowing data exchange between applications. The APIs for the HCMUT_SSPS automated printing system include:

- **Security and Authentication API:** Ensures the security of communications and printing operations by providing authentication through API tokens. These tokens are issued via HCMUT-SSO, granting secure, time-limited access. Additionally, access rights are managed by API token to control permissions and ensure authorized usage of the system, used for Student and SPSO.

- **Get information API**: Retrieve specific data from database. This type of API is typically read-only and focuses on providing requested information without modifying data, used for Student and SPSO.
- **Data Formatting and Input Processing API:** Allows applications to send input data (such as print files, images, or text) to the automated printing system and ensures that this data meets specific formatting and quality requirements.
- **Print Job Management API:** Provides APIs for creating, managing, and tracking print jobs. This can include scheduling prints, monitoring job status, checking available page counts, and canceling print jobs, used for Student.
- **Payment API:** Enables payment functionalities (for document printing and purchasing print pages) via BKPay, used for Student.
- **Custom Print Template API:** Allows for the creation and management of custom print templates, enabling applications to generate dynamic print templates based on specific needs, used for Student, used for Student.
- **Printer Control API:** Allows applications to interact directly with the printer, like create Printer, used for SPSO.
- **Reporting and Statistics API:** Provides data on completed print jobs, print times, page counts, and other relevant information for monitoring and reporting on printing activities, used for Student.

Some main functions for these above API:

getPrinterName	Integer (ID)	Pass the printer ID to get the information of printer’s brand from the database
getPrinterBrand	Integer (ID)	Pass the printer ID to get the information of printer’s name from the database
getStudentFullName	Integer (ID)	Pass the student ID to get the information of the student’s name from the database.
getAllPrinters	Integer(ID)	Pass the LocationID to get all the information of

		the printers in this location.
getAccount	Integer(ID)	Pass the StudentID to get all the information of the account of the student.
checkNamePassword	String(name), String(password)	Pass the username and password to authenticate create token for this user
createPrinter	String(name), String(brand), String(description)	Create printer
getPrintingLog	Integer(ID)	Pass the StudentID to get all the information of the printing log of the student.
createPrintingRequest	MultipartFile(file), Integer(copies)	Create PrintingRequest

4.2 Component Diagram

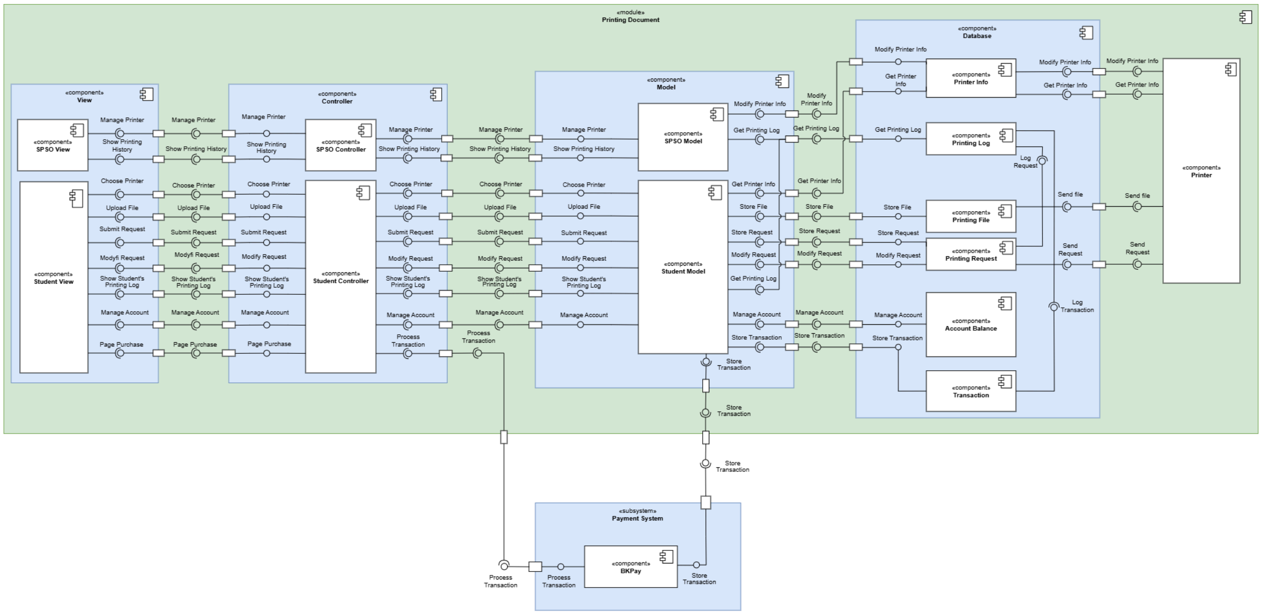


Figure 17. Component Diagram design for Printing module

- **View Component**:

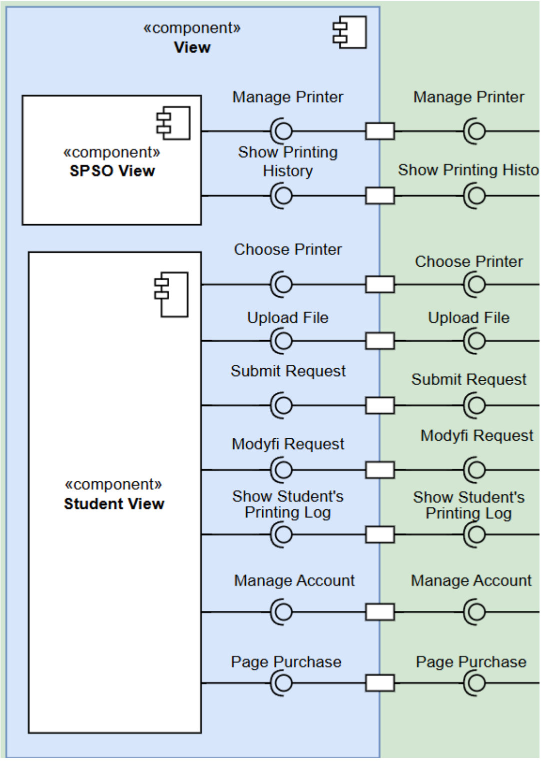


Figure 18. View component in Component Diagram

  - SPSO View allows users to manage printers and view printing history.
  - Student View allows users to choose the printer, upload file, submit request, modify request, view student's printing log, manage account, and page purchase.

- **Controller Component**:
  - SPSO Controller and Student Controller: These components handle the business logic for the SPSO and student views, processing user requests and managing interactions between the view and model layers.
  - The controllers are responsible for actions corresponding to the view's user interface, including printer management, file upload, request submission, account management, and transaction processing.

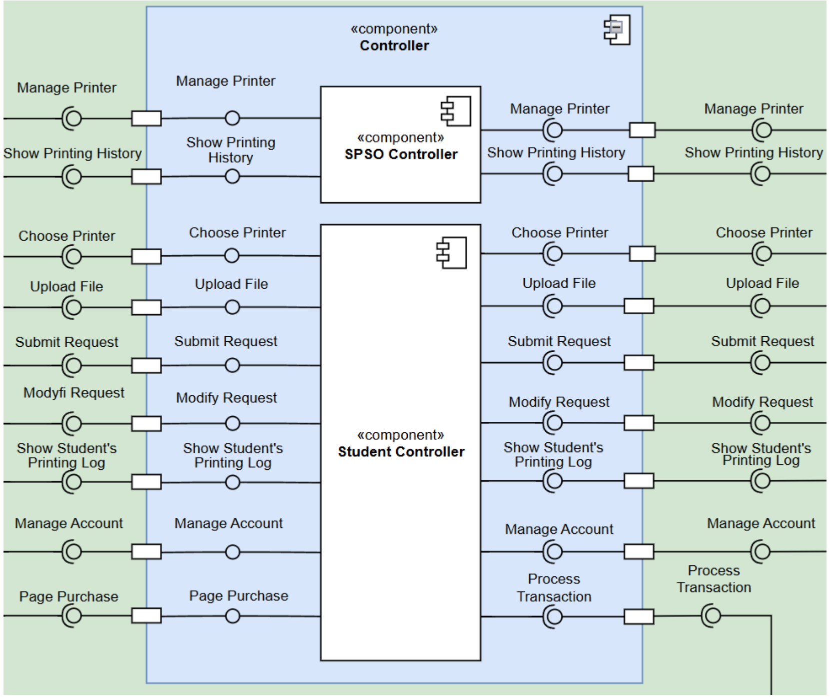


Figure 19. Controller component in Component Diagram

- **Model Component**:

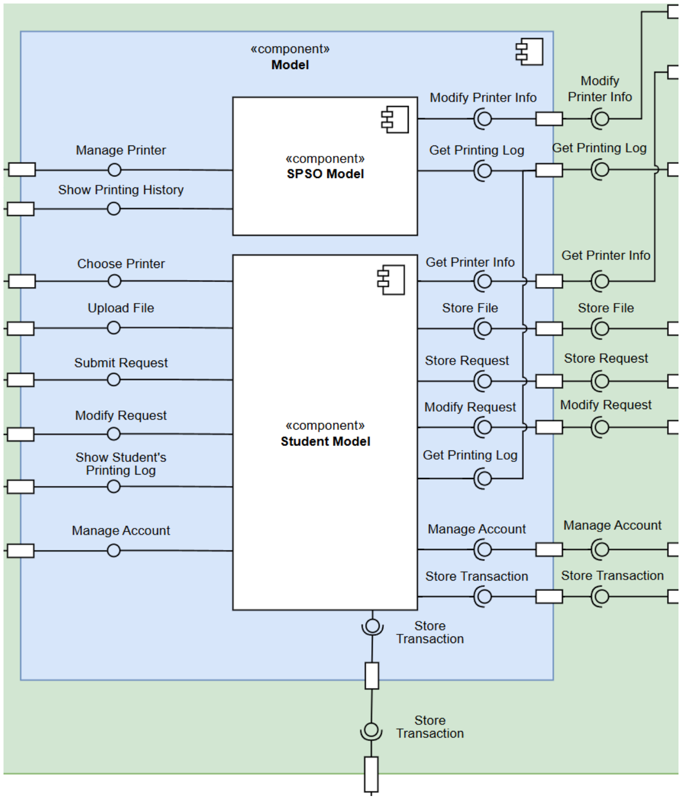


Figure 20. Model component in Component Diagram

- ○ SPSO Model and Student Model: These components represent the underlying data and processing logic for the SPSO and student functionalities.

- ○ The model components handle actions like store transaction, get printing log, store file, store request, modify request, etc, interacting with the database to store and retrieve information.

- **Database Component**:

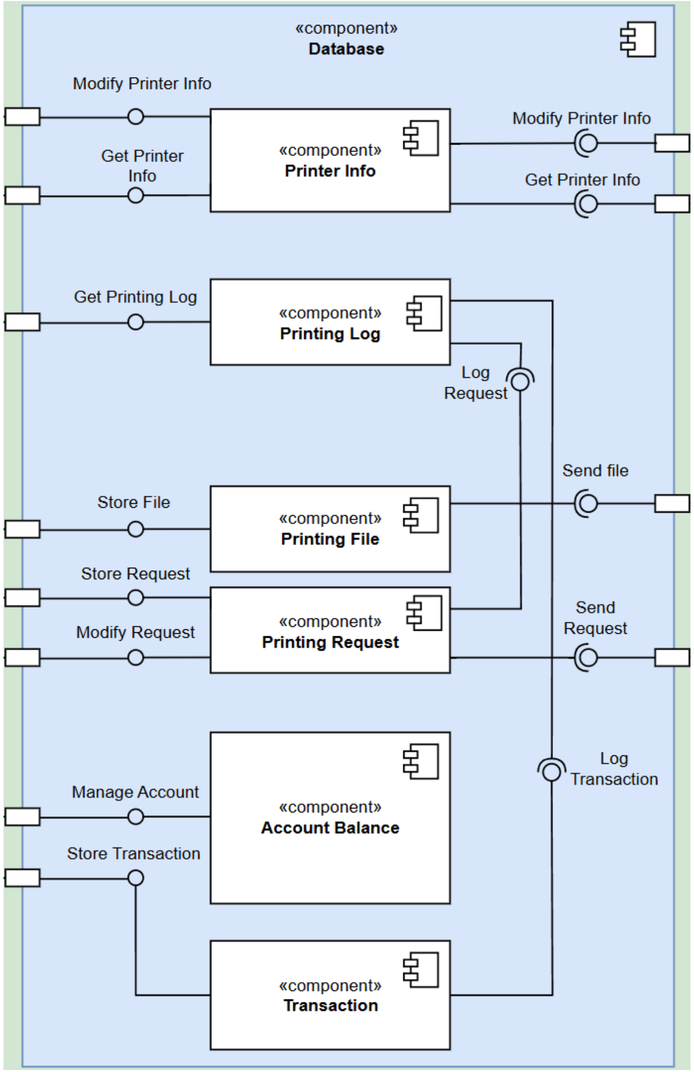


Figure 21. Database component in Component Diagram

- ○ This section contains components like printer info, printing log, printing file, printing request, account balance, transaction, etc.

  - These database components store information about printer configurations, user printing logs, files for printing, user requests, account balances, and transaction records.
  - The data is accessed and modified by the model layer as needed for processing requests and maintaining logs.

- **Printer Component**:

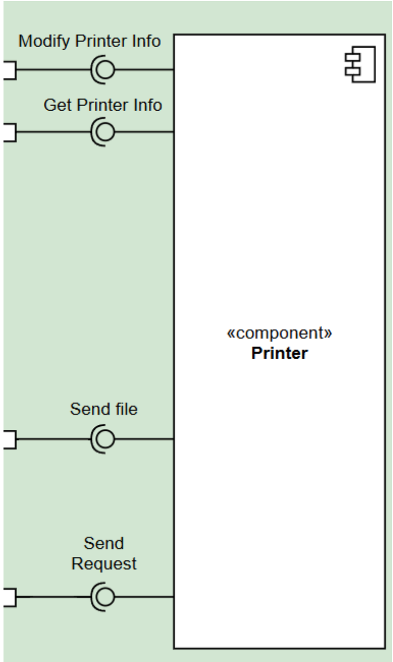


Figure 22. Printer component in Component Diagram

  - Printer component interacts with the printing request and printing file components, facilitating the actual print requests and file transfers to the printer hardware.

- **Payment Subsystem**:
  - A separate payment subsystem module, with a component labeled BKPay, is responsible for handling financial transactions related to printing charges.
  - This module interacts with the transaction component in the database layer to log and process payments for print requests.

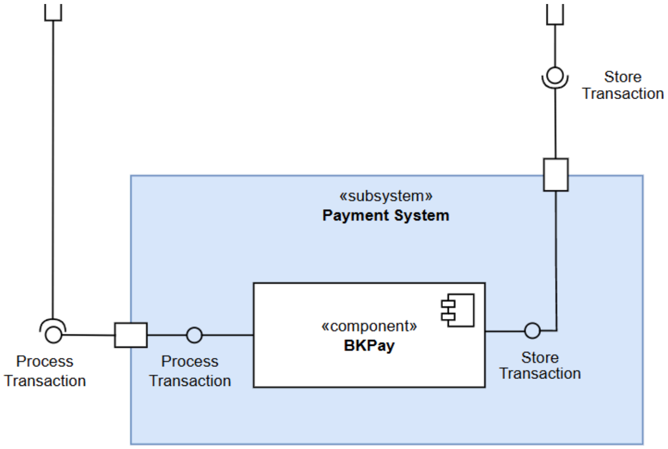


Figure 23. Payment subsystem in Component Diagram