Software Requirements Specification

for

A smart printing service for students at HCMUT

Version 1.1 approved

Prepared by:

1. **Đoàn Thế Anh – 2252019**
2. **Lê Anh Khôi - 2252370**
3. **Hồ Gia Tường – 2252887**
4. **Huỳnh Văn Anh Hoàng - 2252229**
5. **Huỳnh Thanh Duy - 2252114**

Department of Software Engineering

Faculty of Computer Science and Engineering

Ho Chi Minh City University of Technology – VNU-HCM

18/9/2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Group meeting	9-2024	**Task 1: Requirement Elicitation**	**1.0**
Đoàn Thế Anh	9-2024	Benefit of the system, Use case: Upload Documents	1.0
Lê Anh Khôi	9-2024	Domain context, whole diagram, Use-case: Print Documents	1.0
Huỳnh Thanh Duy	9-2024	Functional requirements, use case Specify Printing Properties	1.0
Huỳnh Văn Anh Hoàng	9-2024	Non-functional requirements, Use-case: Cancel Printing	1.0
Hồ Gia Tường	9-2024	Stakeholders and Needs, Use-case: Choose a printer	1.0
Group meeting	30-9-2024	Reformatting, ensuring logical coherence	1.1
Đoàn Thế Anh	27-10-2024	Activity diagram	1.2
Hồ Gia Tường	27-10-2024	Sequence diagram	1.2
Huỳnh Thanh Duy	27-10-2024	Class diagram	1.2
Lê Anh Khôi & Huỳnh Văn Anh Hoàng	27-10-2024	Develope MVP (figma)	1.2
Huỳnh Thanh Duy	7-11-2024	Architectural diagram	1.3
Đoàn Thế Anh	8-11-2024	Component diagram	1.3

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

1.1 Domain Context

The Student Smart Printing Service (HCMUT_SSPS) is designed to provide an efficient document printing solution that allows students to deal with their printing tasks conveniently. In this system, students can upload their document, select from various printers around the campus by entering the printer ID and customize their printing such as paper size, number of copies, and the print format(one- or double-sided). Each student will have a default number of A4 -size pages for printing each semester, with the option to buy some more pages through Buy Printing Pages of the system and purchase through the BKPay system. The system stores all printing activities, logging details such as students ID, printers ID, printing time, file information,... and allows both students and Student Printing Service Officer (SPSO) to view the printing history. The SPSO also manages other configurations of the system such as permitted file types, changing default number of pages, and generates yearly and monthly reports of the usage of the system.

1.2 Stakeholders and Needs

The stakeholders of the HCMUT Smart Student Printing Service (HCMUT-SSPS) are students, the IT team, and the Student Printing Service Officer (SPSO), each with specific needs and criterias.

Students are the primary users of the printing service. As such, the baseline for the service is that it's able to facilitate printing for the students. They must be able to upload their documents for printing and access their uploaded documents, as well as view printing logs and track their remaining allotted resources. Additional functionalities include the ability to specify printing options (including paper size, page range, number of copies, and single- or double-sided printing); and querying functions to retrieve and reprint documents.

The system administrators are responsible for the security and reliability of the end product, which can be achieved through the implementation of several functionalities: Logging of user activities and the ability to authenticate users to restrict or authorize access to the service; Maintenance toolkit to perform regular system updates and maintenance smoothly and efficiently; Debugging & troubleshooting toolkit in case of system errors, for quick diagnosis and resolving to minimize downtime.

Student Printing Service Officer (SPSO) are the managerial bodies? providing service to students with the assistance of software /product/. Said software should help them manage and keep track of assets, in this case, printers and paper supply by tracking relevant information like details of the printer (ID, brand, model, etc.) and the amount of paper in stock, the amount assigned to any individual customer(student). It should also provide administrative functions such as logging customer(student) usage, monitoring & controlling requests, alteration of protocols, generation of monthly and yearly reports. And finally, facilitate transactions from the customer for more paper.

1.3 Benefits of the System

For every stakeholder, the HCMUT-SSPS will provide some benefits. For students, it provides a convenient self-service printing option that they can use from anywhere on campus. This allows them to manage and keep track of their print job. Beside that, students can also flexibly adjust their paper to fit their requirement online. The system facilitates the management of printers and print quotas for the Student Printing Service Officer (SPSO). It grants them instance access to print logs, enabling them to troubleshoot problems effectively and exert greater control over resources allocation. They can quickly resolve issues and guarantee optima service performance by managing and disabling printers with ease. The IT staff benefit from having fewer printers to manually monitor and maintain is made easier by the centralized design of the SSPS. Integration with already in-use authentication and payment systems (such as BKPay and HCMUT_SSO) facilitates simple maintenance and update while ensuring secure and seamless operations.This reduces the possibility of unexpected use. Moreover, the IT staff can also plan future enhancements and maintain system efficiency with the help of the system automatic report features.

1.4 Functional Requirements

Student:

- A student shall be able to upload document files for printing.
- A student shall be able to choose a printer and specify printing properties such as paper size, pages to print, one-/double-sided, and number of copies.
- A student shall be able to cancel a print job before it starts.
- A student shall be able to track when a print job starts and finishes.
- A student shall be able to view their printing history for a specified time period.
- A student shall be able to filter their printing log by printer, paper size, or date range.
- A student shall automatically receive a default number of A4 pages at the start of each semester.
- A student shall be able to buy additional printing pages using BKPay.
- A student shall be able to view their current page balance.
- A student shall be prevented from printing if the number of pages exceeds their available page balance.
- A student shall be able to view a summary of the number of printed pages for each page size.
- The system shall automatically calculate the correct number of pages from the student's balance based on the paper size chosen.

IT Team:

- The system shall require all users to authenticate via the HCMUT_SSO authentication service.
- The system shall be accessible through both a web-based app and a mobile app.
- The system shall allow the IT Team to monitor server load and resource usage.
- The system shall log system errors, failed user uploads, failed logins, unsuccessful print jobs.
- The system shall allow the IT Team to monitor printer availability and status.
- The system shall automatically generate and store reports on the usage of the printing service at the end of each month and year.

SPSO:

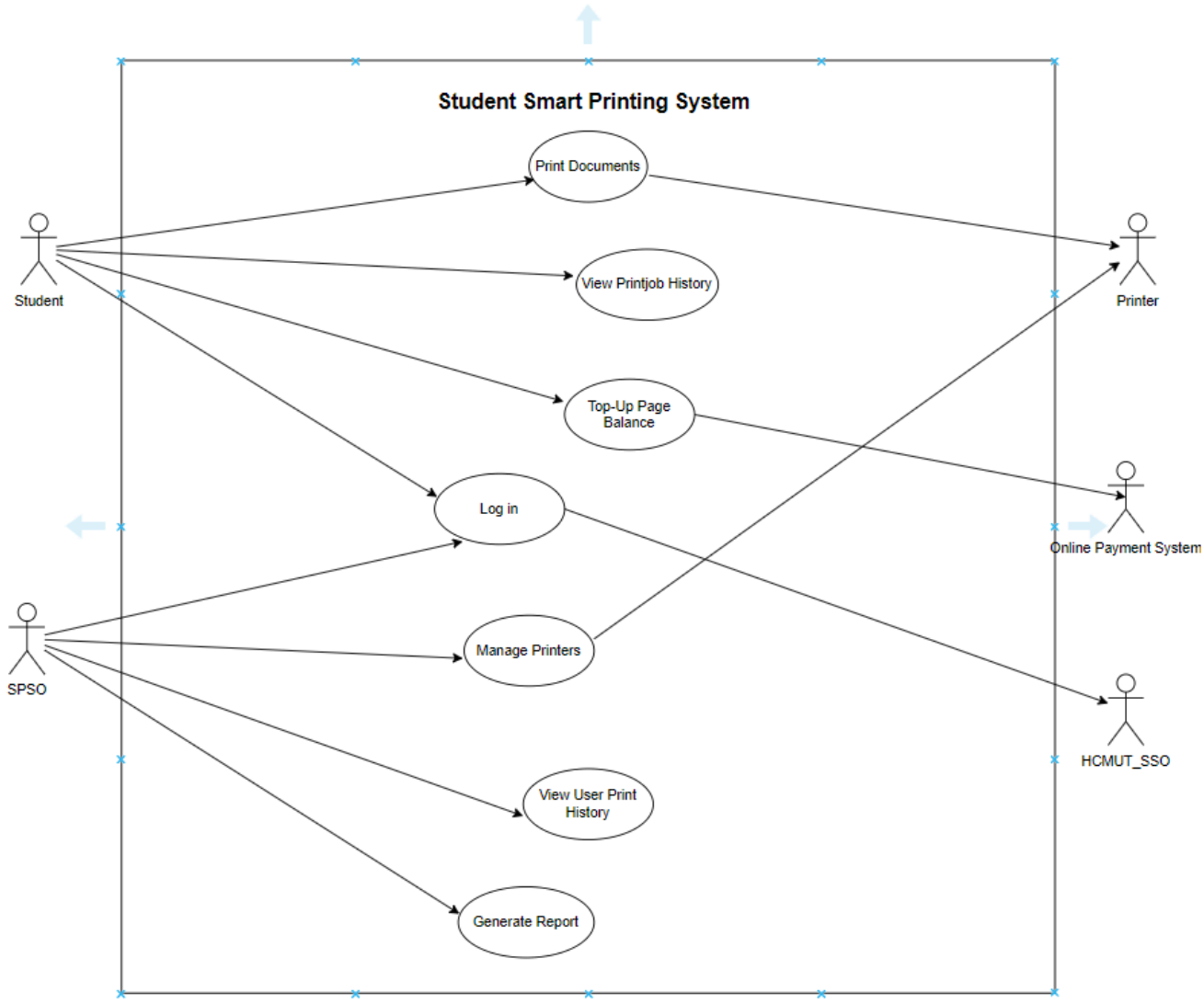
- The SPSO shall be able to add, enable, and disable printers in the system.
- The SPSO shall be able to configure the permitted file types for printing.
- The SPSO shall be able to set the default number of A4 pages assigned to students each semester.
- The SPSO shall be able to set the dates for crediting students with their default printing pages.
- The SPSO shall be able to view the printing logs for all students or a specific student, filtered by time period and printer.
- The SPSO shall be able to filter the printing logs by specific printers.
- The SPSO shall be able to view reports on system usage at any time.

1.5 Non- Functional Requirements

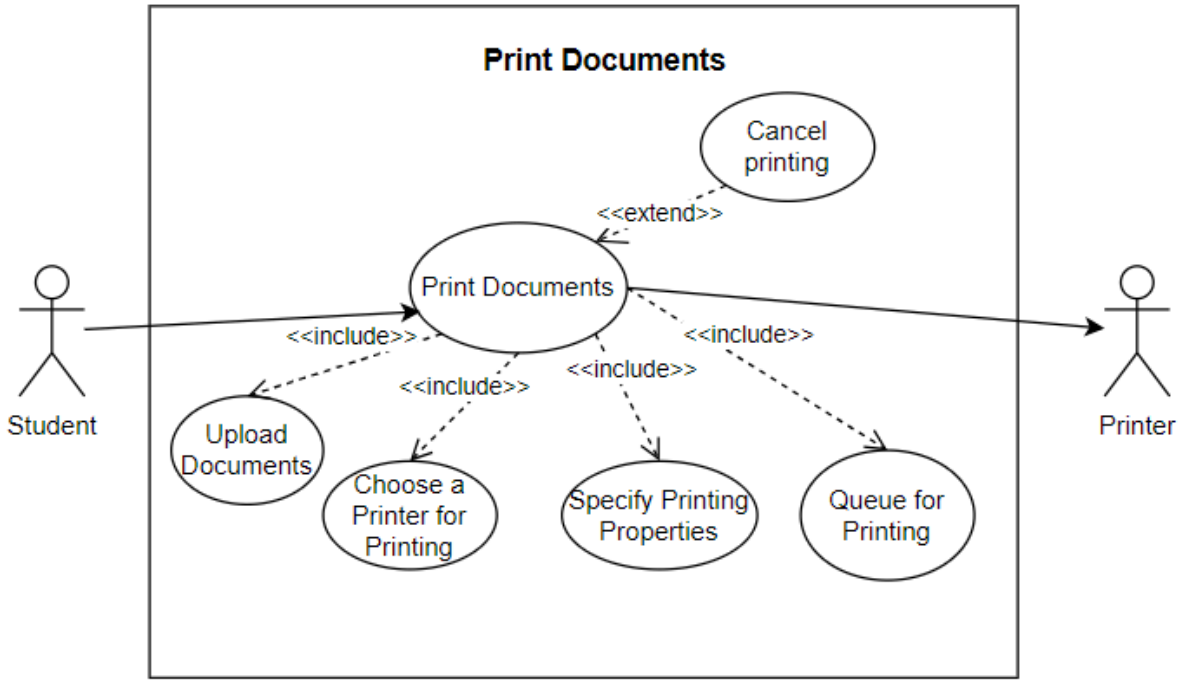
- Performance:
    + The loading time of the sending the request of user to the printer should be at most 5 seconds.
    + The system should maintain performance when handling up to 100 requests concurrently
- Availability:
    + The system should operate all the time (24/7 mode).
    + The system should not experience more than 1 hour of unplanned downtime per month.
- Maintainability:
    + The system's code should be modular to allow updates with minimal effort.
    + The code should follow a consistent naming convention and coding standard to ensure readability and maintainability.
    + The system architecture should allow for new features to be added with minimal changes to the existing codebase.
- Reliability:
    + The system must perform automatic data backups every hour to prevent data loss.
    + The system must ensure that no data is lost or corrupted during storage and retrieval operations.
- Synchronization:
    + One account should not be used in many printers at the same time.
- Security:
    + All users have to be authenticated by the HCMUT_SSO authentication service before using the system.
- Usability:
    + New users can learn to use basic functions in 10 minutes.
- Portability:
    + The system is provided through a web-based app and a mobile app.

2. Use-case Diagrams (1.3)

2.1 Use-case Diagram for the Whole System



2.2 Use-case Diagram for <Print Documents> Module



2.3 The Details of Usecases in <Print Documents> Module

1. **Usecase <Upload Documents>**

Name	Upload Documents
Create by	Thế Anh
Date Create	29/9/2024
Primary actor	Student
Secondary actor	N/A
Description	The student uploads the document to the system for later printing. The document type and size are also limited and managed previously by SPSO.
Trigger	The student chooses the upload function of the app on their device.
Precondition	PRE-1. Student identity has been authenticated. PRE-2. The SSPS database is still working.
Postcondition	POST-1. The document is saved on the local cache successfully
Normal Flow	1. The system will open a tab for students to drop in their documents from their device or drive. 2. The system will check the permitted file type and size of the document. 3. The system uploads to the local cache.
Alternative Flow	3.1 Invalid document(s): Notify the student and suggest re-upload. Return to step 1.
Exception	1. Network Failure: Notify the student and suggest refreshing the page or app.

2. **Usecase <Specify Printing Properties>**

Name	Specify Printing Properties
Created By	Thanh Duy
Date Created	29/9/2024
Primary Actor	Student
Secondary Actors	N/A
Description	The student specifies the size of paper they want to print in, the pages, number of copies, single-sided or double-sided, black and white or colored, and orientation (portrait or landscape).
Trigger	The student finishes uploading the document(s) to the cache.
Preconditions	PRE-1. Student identity has been authenticated. PRE-2. The documents for printing are ready and valid.
Postconditions	POST-1. The printing specifications will be selected.
Normal Flow	1. The student chooses the paper size. 2. The student chooses the pages they want to print. 3. The student chooses the number of copies 4. The student chooses if the document should be single-sided or double-sided. 5. The student chooses whether the document is colored 6. The student chooses the orientation of the documents 7. The system calculates the number of papers needed. 8. The student confirms the specifications.
Exceptions	1. Network Failure: Notify the student and suggest refreshing the page or app.

3. **Usecase <Choose a Printer for Printing>**

Name	Choose a Printer for Printing
Created By	Gia Tường
Date Created	29/9/2024
Primary Actor	Student
Secondary Actors	N/A
Description	The student selects a printer from a list of available printers within the system. The student can review printer details (e.g., location, status, and type of printer) before making a selection.
Trigger	The student finishes specifying properties.
Preconditions	PRE-1. Student identity has been authenticated. PRE-2. Printers have been configured and are available for selection. PRE-3. The printing properties are specified and valid.
Postconditions	POST-1. The selected printer is queued for the print job.
Normal Flow	1. The system displays a list of available printers to the student. 2. The student reviews the available printer details (location, status, type). 3. The student selects a printer from the list. 4. The system saves the selected printer for printing the document.
Alternative Flows	1.1 No Available Printers: Notify the student and suggest waiting or contacting support. End use-case. 3.1 Printer Unavailable(offline, out of paper): Notify the student and suggest alternative printers. Return to step 1.
Exceptions	1. Network Failure: Notify the student and suggest refreshing the page or app.

4. Usecase <Queue for Printing>

Name	Queue for Printing
Created by	Anh Khôi
Date Created	29/09/2024
Primary Actor	Student
Secondary Actor	Printer
Description	The student initiates the print job after selecting the desired printer and specifying the print properties (e.g., number of copies, paper size, double-sided). The system sends the document to the selected printer, tracks its progress, and updates the student's printing balance and printing history.
Trigger	The student selects the "Print" option after configuring the print properties and choosing a printer.
Preconditions	PRE-1. Student identity has been authenticated. PRE-2. The printing properties are specified and the selected printer is available.
Postconditions	POST-1. The document is successfully printed. POST-2. The student's printing log and balance are updated.
Normal Flows	1. The student reviews the documents for printing. 2. The student confirms the printing job. 3. The system verifies the student has sufficient page balance. 4. The system queues the documents for printing. 5. The system tracks the progress and notifies the student once printing is complete. 6. The system updates the student's page balance and printing history.
Alternative Flows	3.1 Insufficient Balance: Notify the student and suggest purchasing additional pages. Transfer to "Top-Up Page Balance" module.
Exception	Printer Disconnects: Attempt to resume printing or notify the student to choose another printer. Network Failure: Store the print job and retry once the connection is restored.

5. Usecase <Cancel printing>

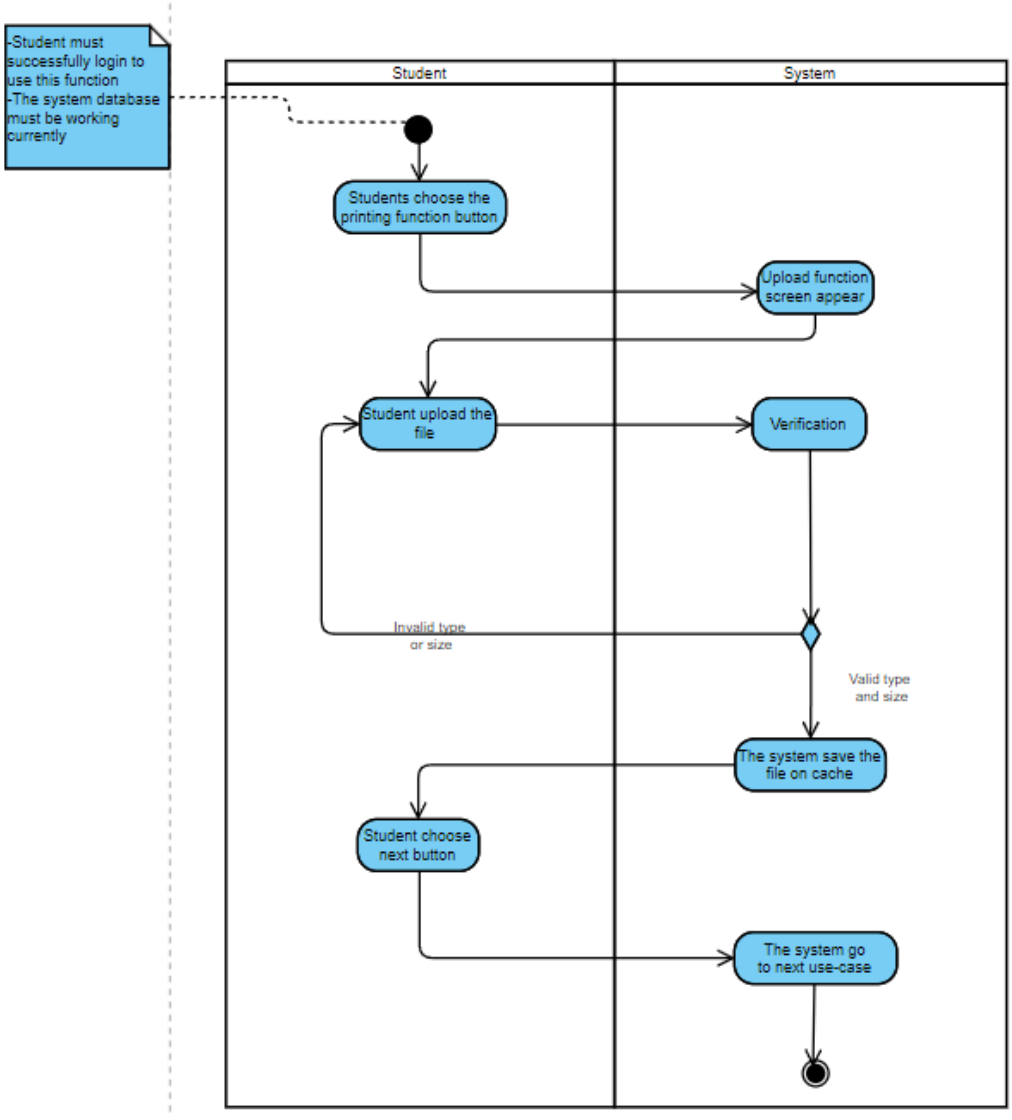
Name	Cancel Printing
Created by	Anh Hoàng
Date Created	29/09/2024
Primary Actor	Student
Secondary Actor	
Description	The student can cancel queued printing request(s).
Trigger	The student views sent requests and selects the "Cancel" option.
Preconditions	PRE-1. Student identity has been authenticated. PRE-2. The chosen printing request(s) must be in waiting state.
Postconditions	POST-1: The chosen request will be removed from the printer's waiting queue.
Normal Flows	1. The student tracks queued printing requests. 2. The student selects the request(s) they want to cancel. 3. The system asks the student to confirm cancellation. 4. The system removes the requests from the waiting queue.
Alternative Flows	3.1 Student refuses confirmation: End use-case. 4.1 Request Already Canceled: Notify the student. End use-case. 4.2 Request Already Completed: Notify the student. End use-case.
Exception	Network Failure: Notify the student and suggest refreshing the page or app.

3. System modeling

3.1 Activity diagrams

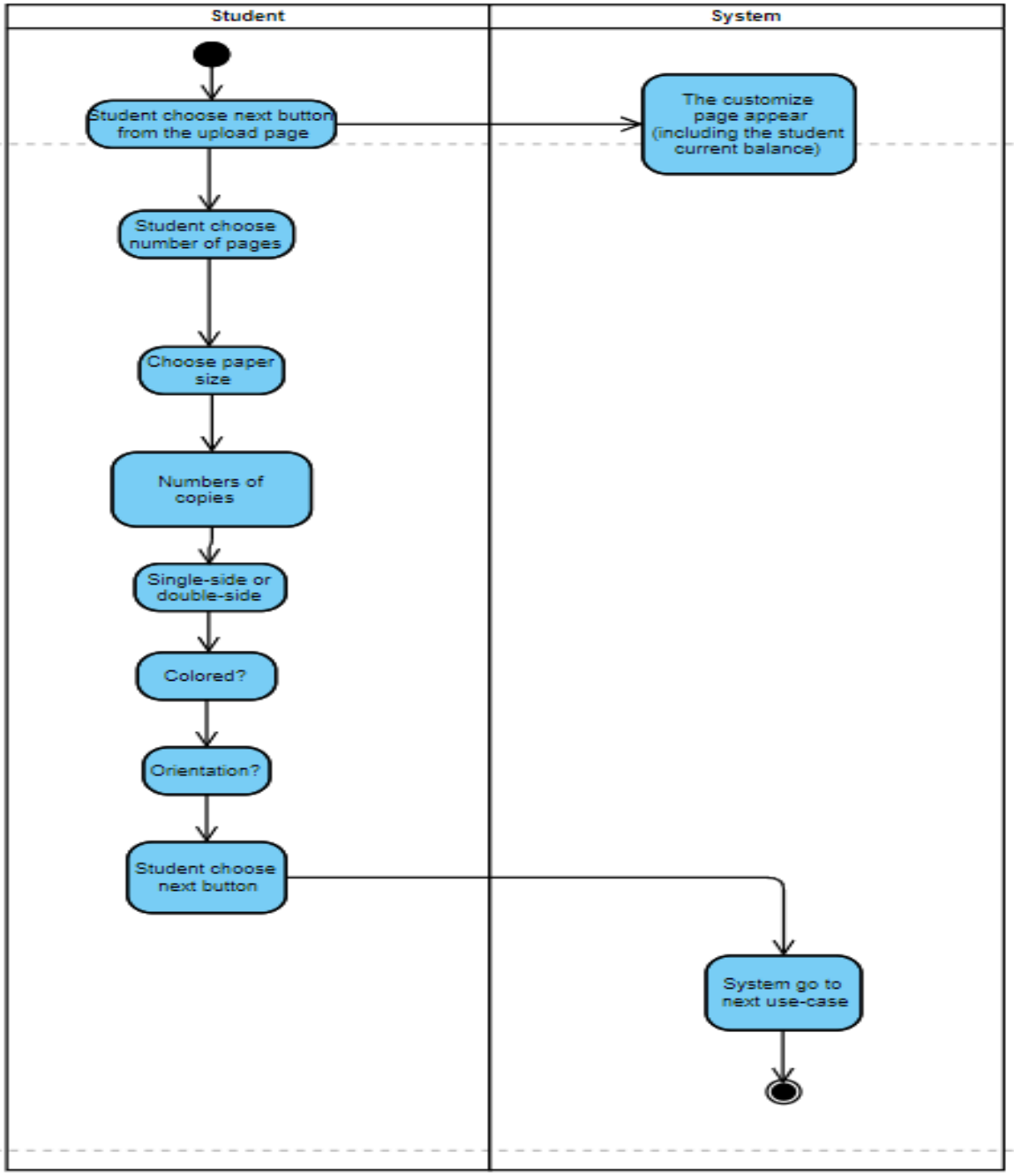
1/ Upload Documents use-case

- The activity diagram represents the "Upload Documents" use-case in a Smart Student Printing Service (SSPS). To access the printing function, a student must be logged in, and the system's database must be operational. The process begins with the student selecting the printing function, prompting the system to display the upload screen. The student then uploads a file, which undergoes verification by the system to ensure it meets the acceptable type and size requirements. If the file is invalid, the student is directed back to re-upload an appropriate file. If the file passes verification, it is temporarily stored in the system's cache. The student can then proceed by selecting the "Next" button, which transitions the system to the Specify-propertiy use-case.



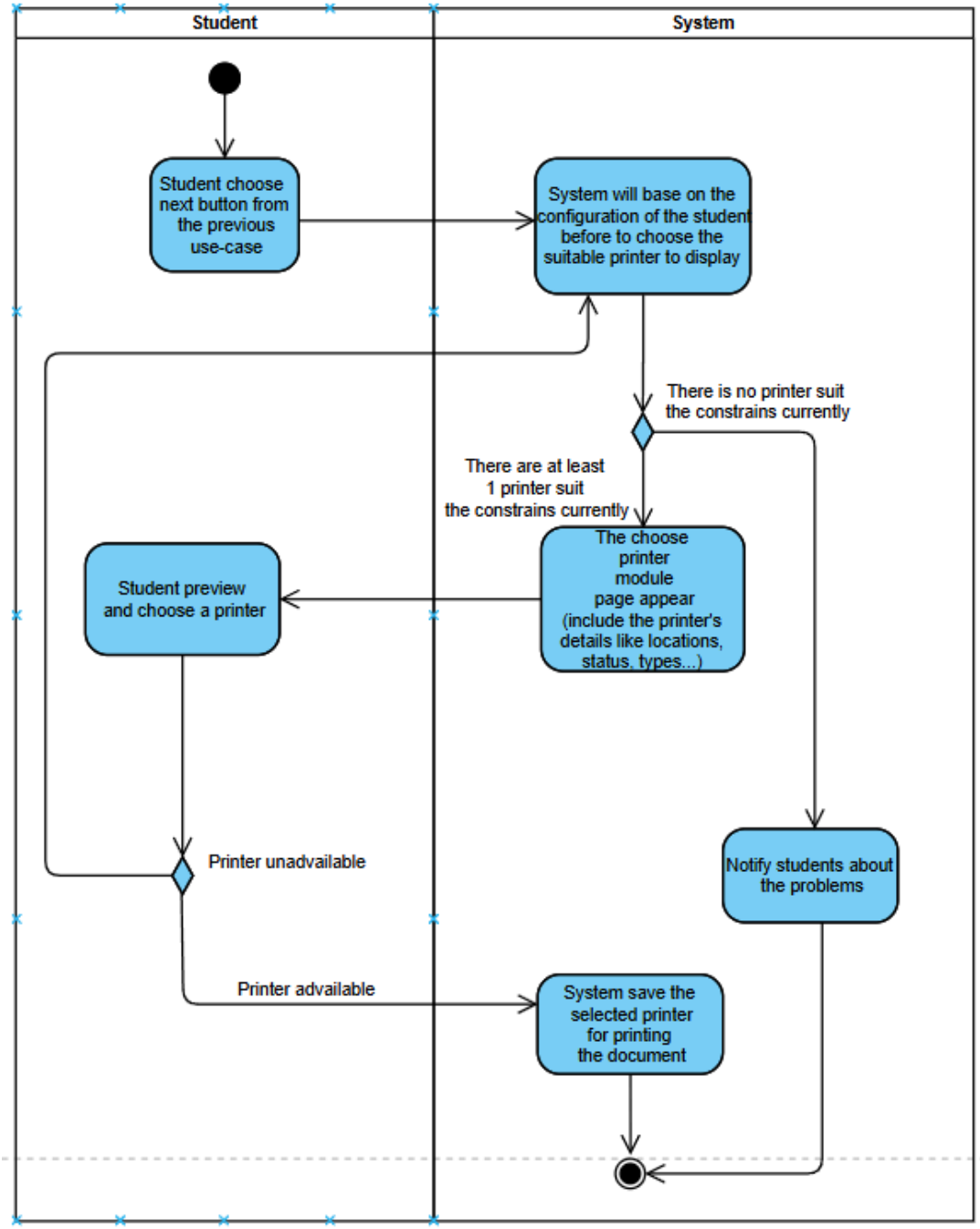
2/Specify printing properties use-case

- After successfully uploading a verified file, the student proceeds by clicking the "Next" button. The system then displays a customization page, which includes details such as the student's current balance. The student is prompted to specify various print settings, beginning with the number of pages to print. Additional customizable properties include selecting the paper size, the number of copies, single-sided or double-sided printing, color options, and the print orientation (portrait or landscape). Once all preferences are set, the student clicks "Next" to proceed, allowing the system to transition to the choose-printer use-case.



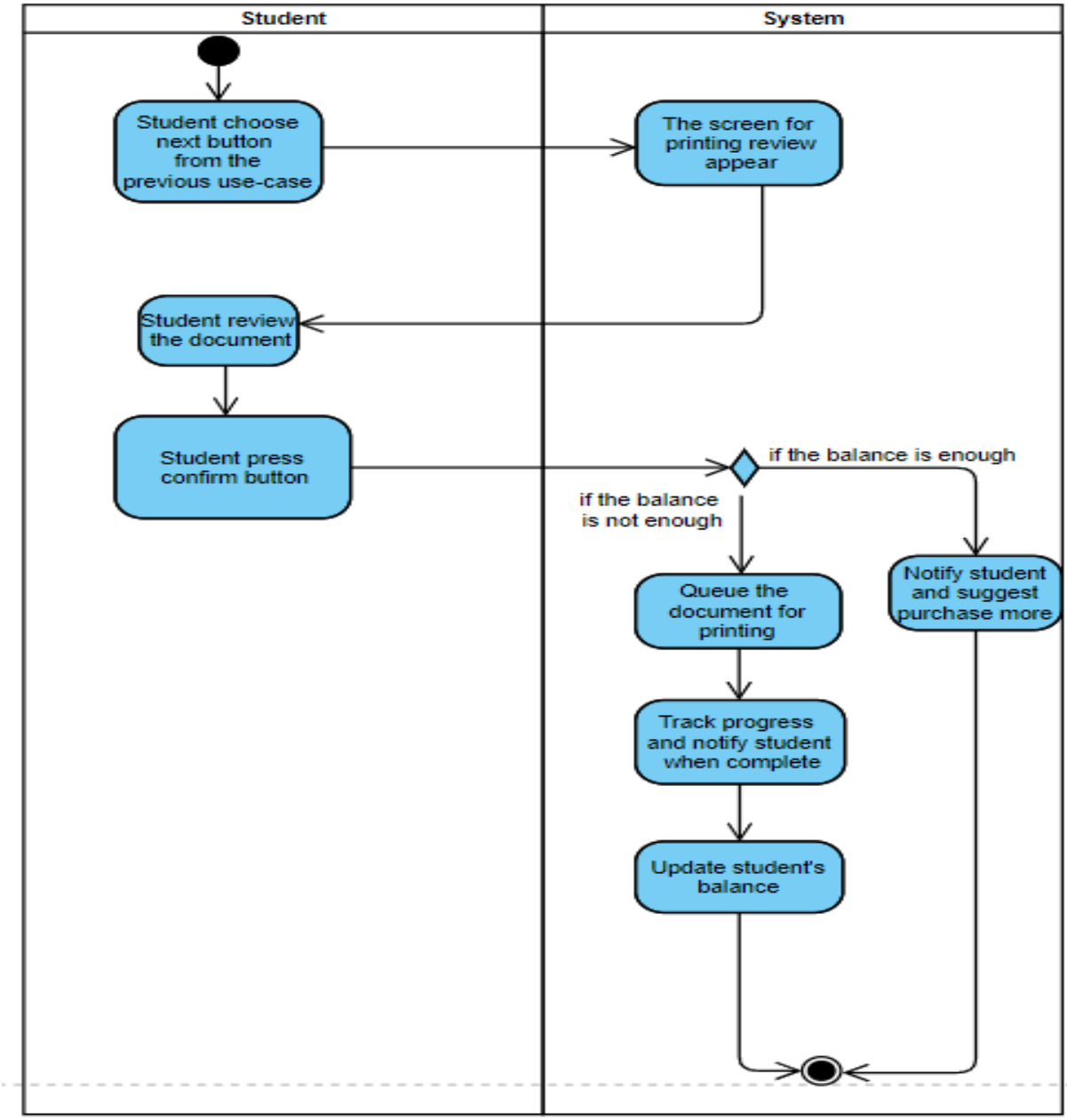
3/Choose a Printer for Printing use-case

- After setting the print properties, the student clicks the "Next" button to proceed to the printer selection stage. The system uses the specified print configurations to display a list of suitable printers based on their availability, location, and type. If no printers meet the current requirements, the system notifies the student about the unavailability. However, if at least one printer matches the specified constraints, the "Choose Printer" page appears, displaying details for each printer, such as location and status. The student can preview the options and select a printer. If the chosen printer becomes unavailable during this process, the student is prompted to select another option. This use-case ensures that students have access to compatible printers that meet their specific requirements:



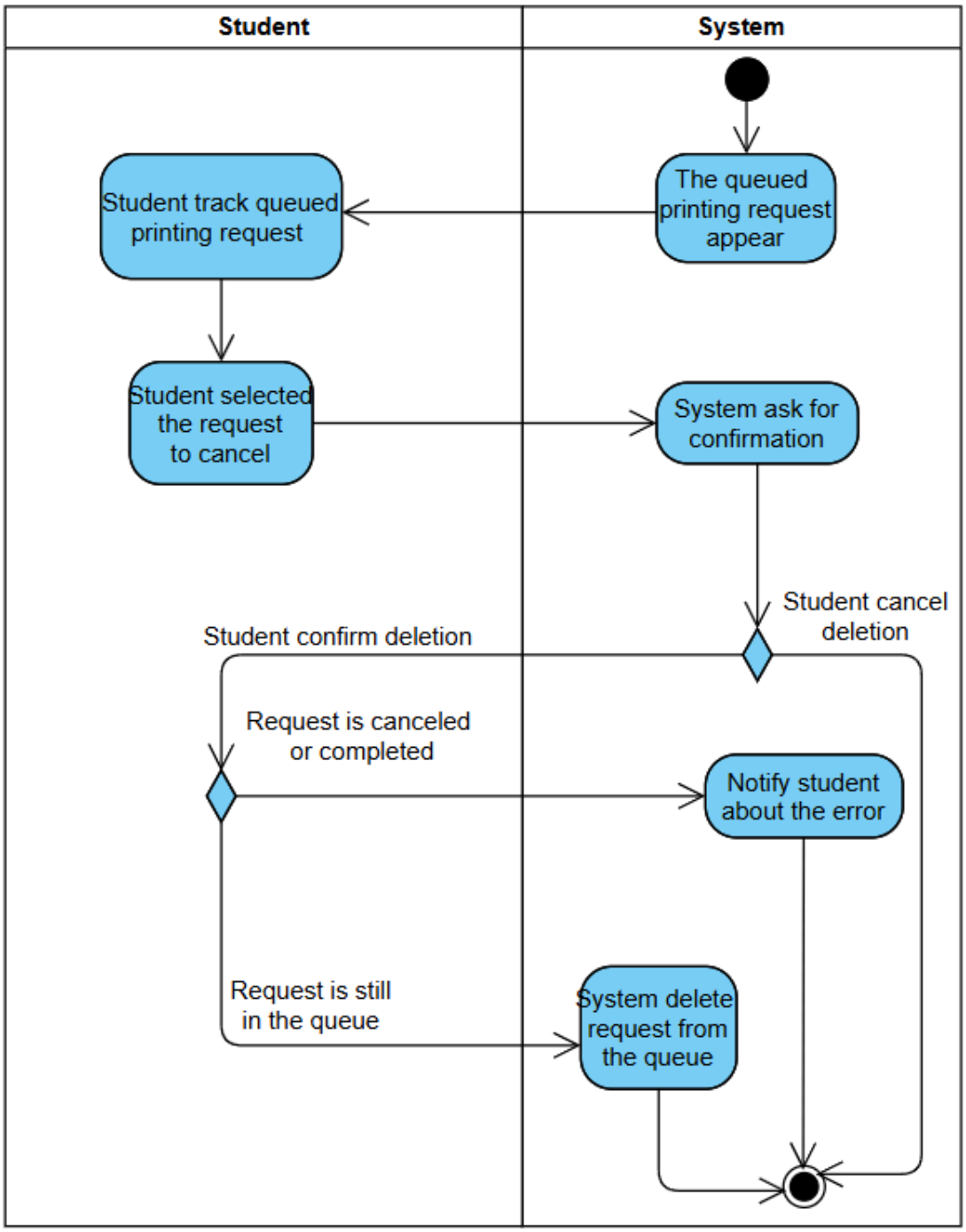
4/ Print Documents use-case

- After selecting a printer, the student clicks "Next" to reach the printing review screen. Here, the student can review the document details before finalizing the print job. Upon confirming, the system checks if the student's balance is sufficient for the printing costs. If the balance is insufficient, the system notifies the student and suggests purchasing additional credit. If the balance is adequate, the document is queued for printing, and the system tracks the printing progress. Once printing is complete, the system notifies the student and updates their balance accordingly. This step ensures that students are informed of any balance issues in advance, providing a seamless experience as they complete the printing process:

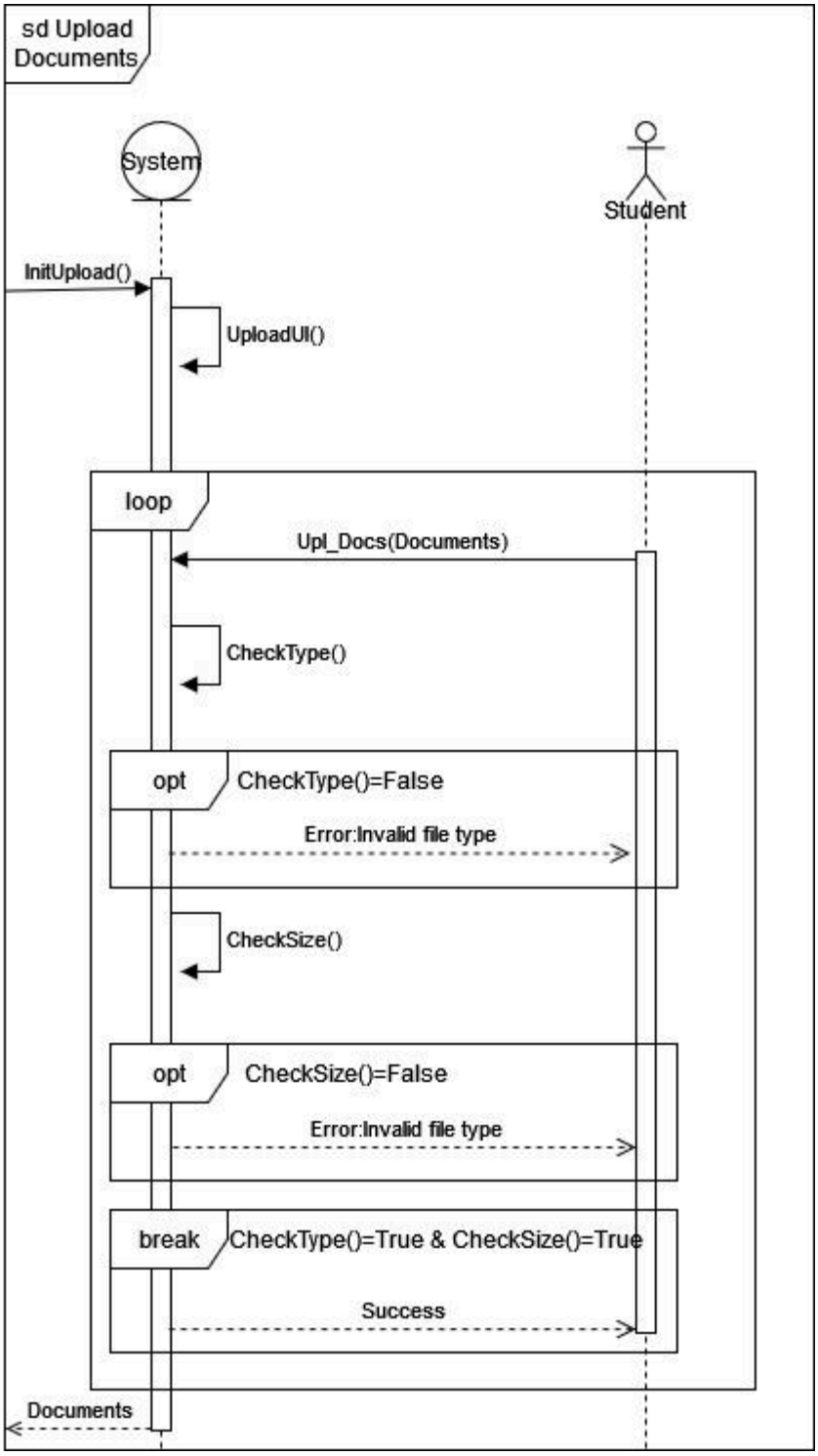


5/ Cancel Printing use-case

- The process begins with the student tracking their queued print jobs. If they choose to cancel a specific request, the system prompts them for confirmation. Once the student confirms the cancellation, the system checks the status of the request. If the request is already canceled or printing or already printed, the system will notify students about this error. Otherwise, the system will delete the request in the queue. This use-case provides students with control over their queued print jobs, ensuring they can manage their print tasks effectively:

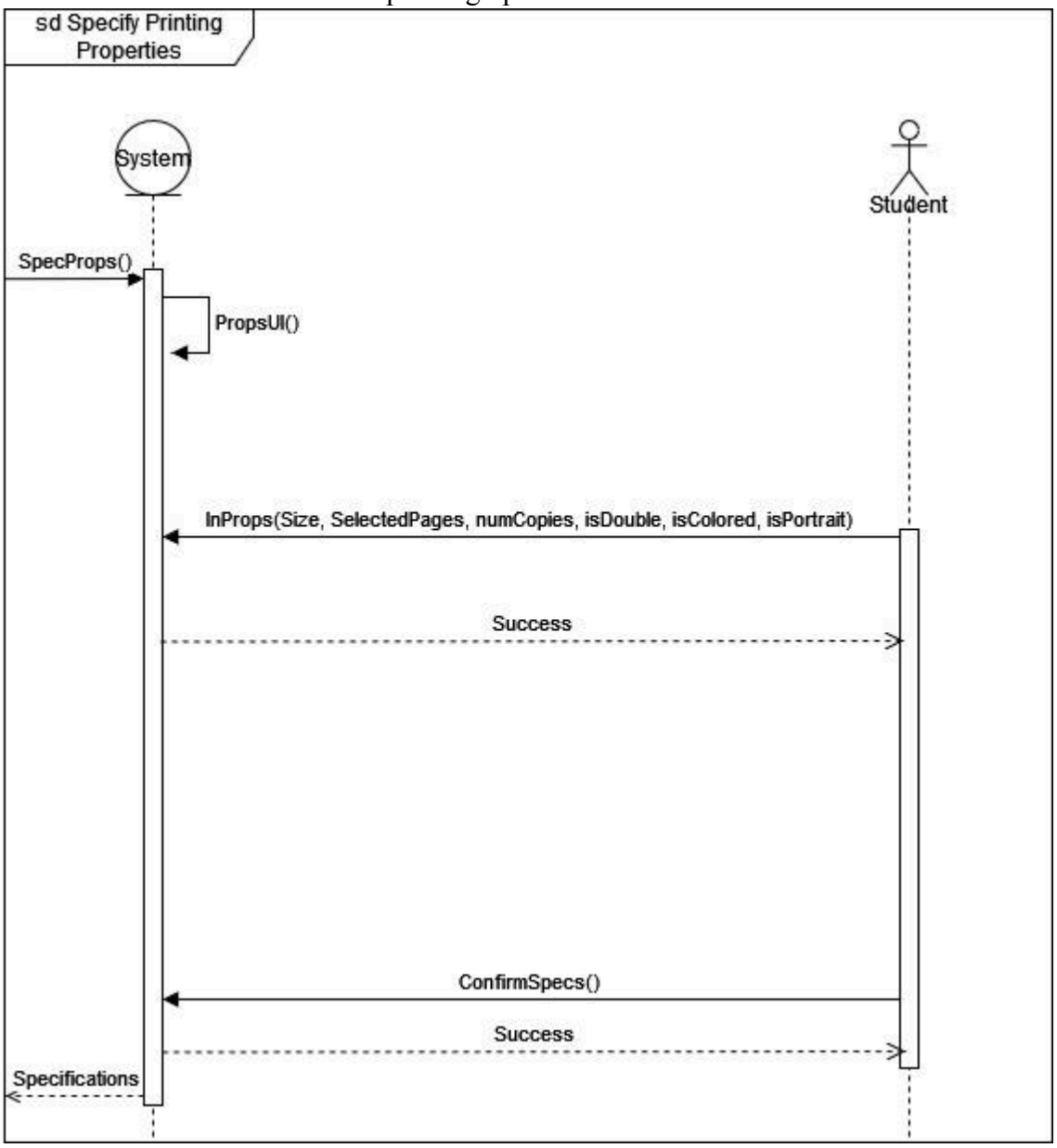


3.2 Sequence diagram

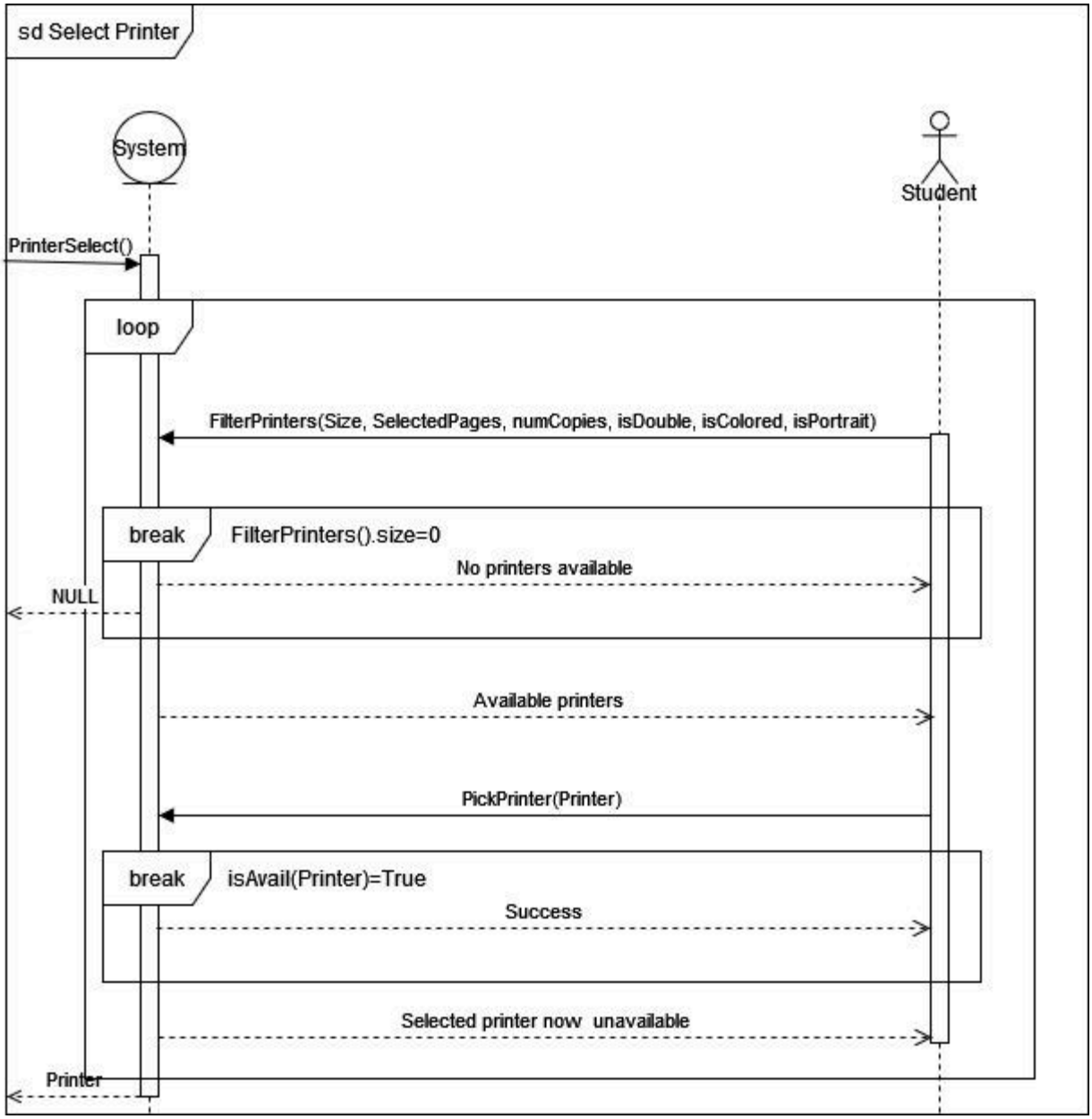


The "Upload Documents" sequence diagram begins with the student selecting a document to upload through the system interface. Upon receiving the document, the system validates the file type, then file size to ensure compatibility with supported formats and meets uploading limits. If the file type or file size is invalid, the system notifies the student with the respective error message and restarts the upload process. Once both validations are successfully completed, the system uploads the

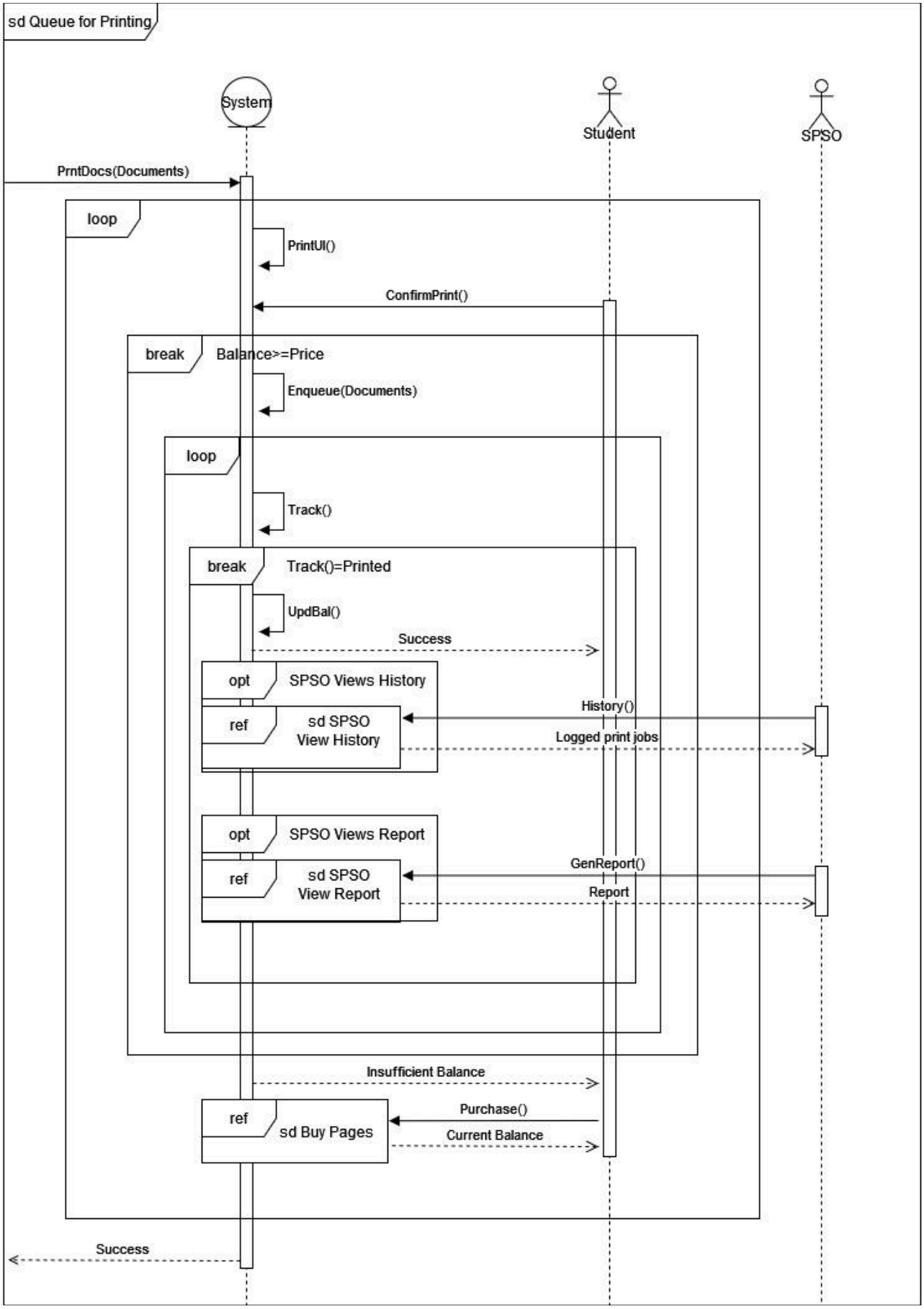
document and provides confirmation of the successful upload, which is then reflected in the student's document list for future printing options.



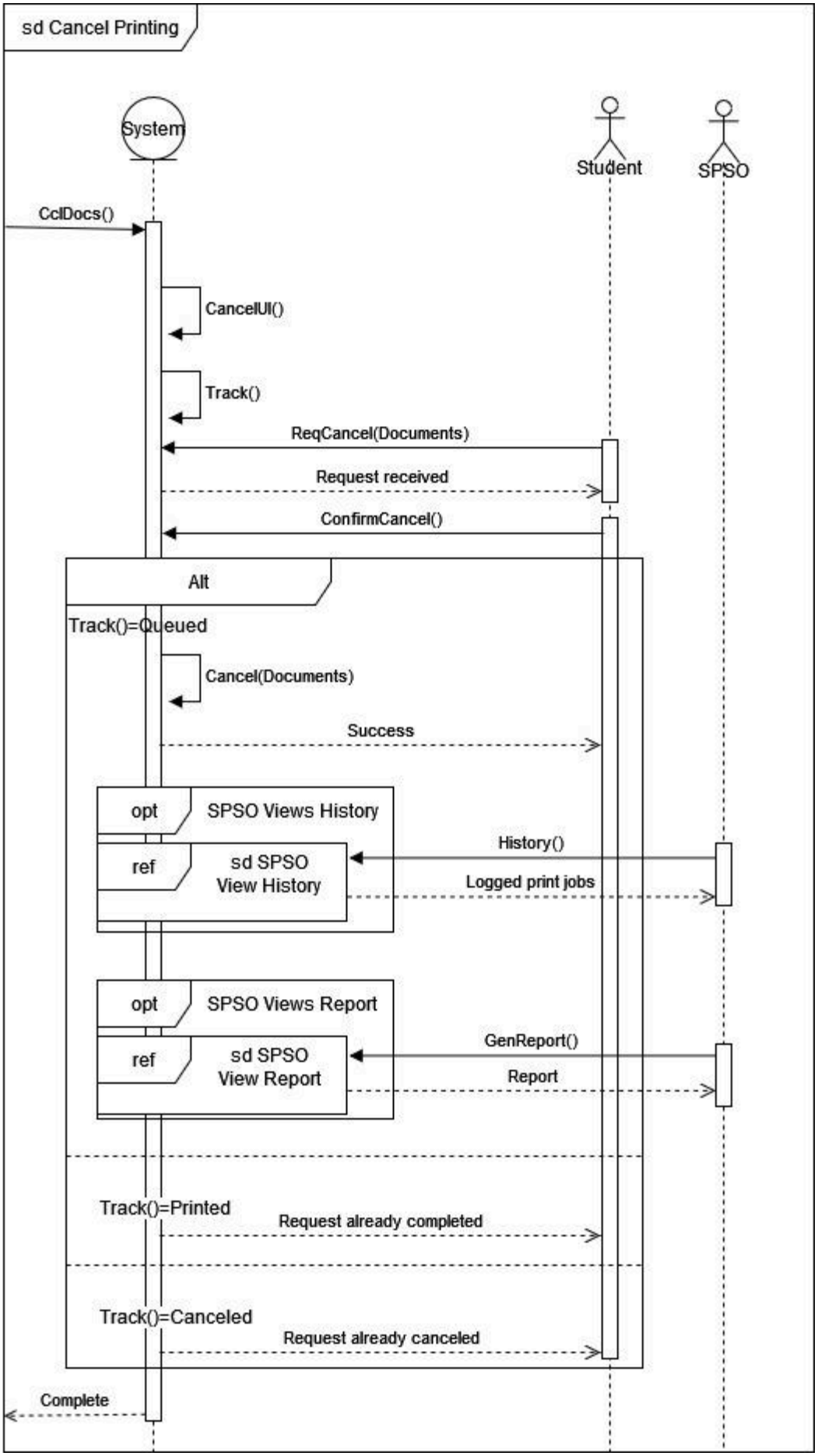
The "Specify Printing Properties" sequence diagram begins with the student initiating a request to input various printing parameters, such as paper size, which page to print, number of copies, whether the pages are printed on both sides, whether it’s colored, and orientation. The system then checks whether these inputs are valid, if they are, the system asks the student to confirm this and saves it for printing. Should any input be invalid, the system notifies the student, prompting them to re-enter the necessary information.



The "Choose Printer for printing" sequence diagram begins with the student initiating the "Select Printer" option within the system interface. The system then retrieves a list of available printers and conducts a filtering process to identify suitable options. If no printers are found, a notification informs the student that no printers are available, prompting them to act accordingly. If printers are available, the system presents this list to the student, who then chooses their desired printer. Upon selection, the system again verifies the printer's availability through a function call. If the selected printer becomes unavailable during the process, the system notifies the student that the printer is no longer accessible, thus requiring them to select a different one. If the selection is successful, a success message is displayed, and the system returns the chosen printer.

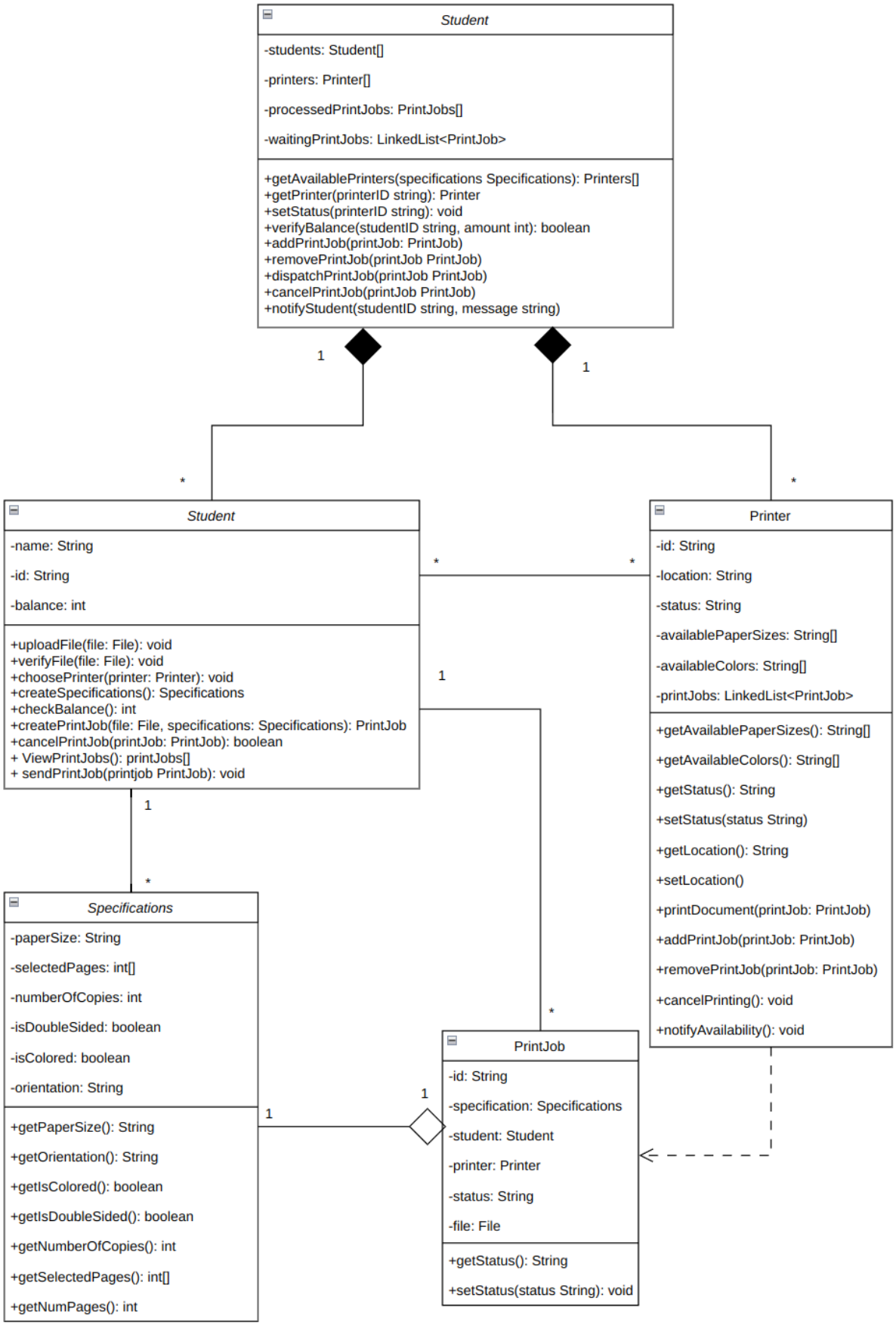

sd Queue for Printing
System
Student
SPSO
PrntDocs(Documents)
loop
PrintUI()
ConfirmPrint()
break
Balance>=Price
Enqueue(Documents)
loop
Track()
break
Track()=Printed
UpdBal()
Success
opt
SPSO Views History
ref
sd SPSO View History
History()
Logged print jobs
opt
SPSO Views Report
ref
sd SPSO View Report
GenReport()
Report
Insufficient Balance
ref
sd Buy Pages
Purchase()
Current Balance
Success

The "View Printing History" sequence diagram begins with the student selecting the "Print Documents" option in the system interface. The system promptly displays the documents queued for printing and the specifications the student made previously and asks for student confirmation. If the student has insufficient balance for the print order, they have the option to purchase additional balance using a different module in the system “Buy Pages”. Otherwise, the system queues the documents for printing at the user-specified printer(s), during which time, the system tracks the completion of the order. Once the order is completed, the student’s balance is updated to reflect the transaction, and the system logs the print history for archiving, which can be accessed by SPSO using their dedicated modules.

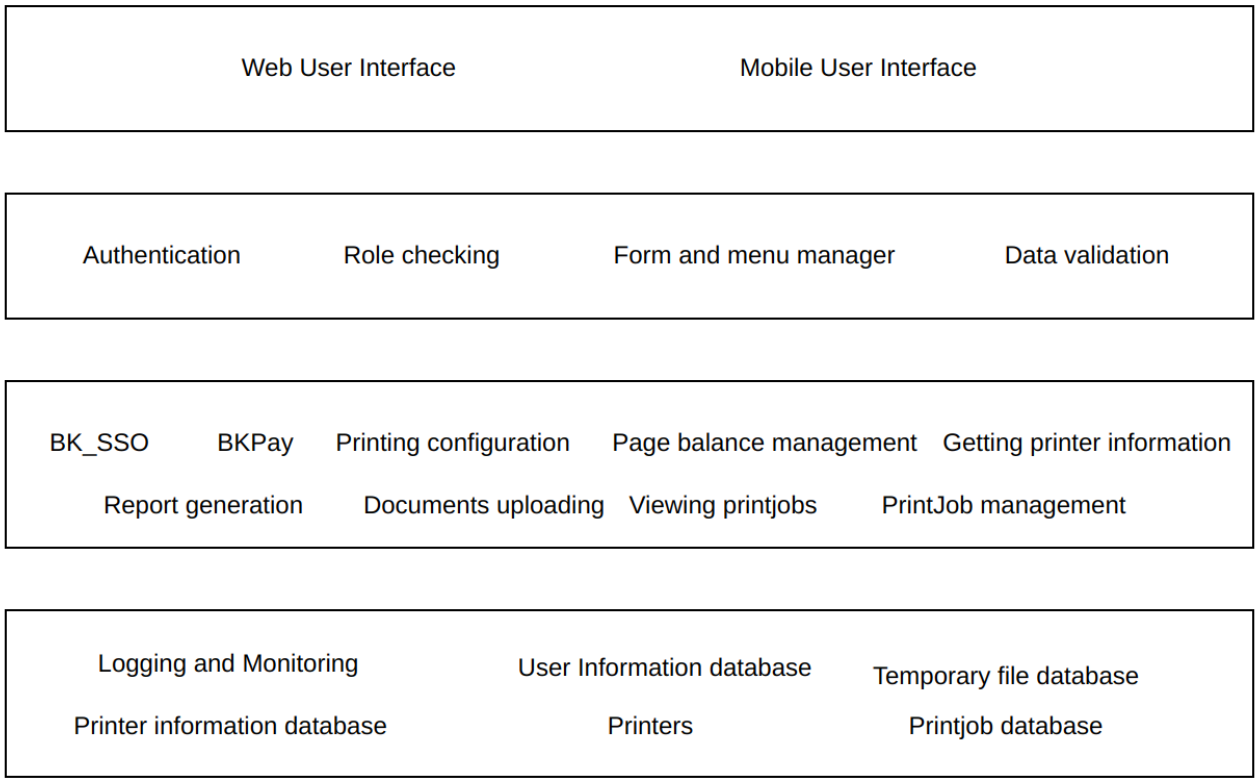

sd Cancel Printing
System
Student
SPSO
CclDocs()
CancelUI()
Track()
ReqCancel(Documents)
Request received
ConfirmCancel()
Alt
Track()=Queued
Cancel(Documents)
Success
opt
SPSO Views History
ref
sd SPSO View History
History()
Logged print jobs
opt
SPSO Views Report
ref
sd SPSO View Report
GenReport()
Report
Track()=Printed
Request already completed
Track()=Canceled
Request already canceled
Complete

The "Cancel Print Request" sequence diagram begins with the student initiating a cancellation request through the system interface. Upon receiving this request, the system verifies the job's status to ensure it is still cancelable. If the print job has not yet begun, the system updates the status to "canceled". The system then updates the student's print history to reflect the canceled status and provides confirmation of the successful cancellation, at which point, the Student Printing Service Officer (SPSO) can view the changes made through their dedicated modules. Should the cancellation not be feasible, the student will be notified.

3.3 Class diagram


Student
-students: Student[]
-printers: Printer[]
-processedPrintJobs: PrintJobs[]
-waitingPrintJobs: LinkedList<PrintJob>
+getAvailablePrinters(specifications Specifications): Printers[]
+getPrinter(printerID string): Printer
+setStatus(printerID string): void
+verifyBalance(studentID string, amount int): boolean
+addPrintJob(printJob: PrintJob)
+removePrintJob(printJob PrintJob)
+dispatchPrintJob(printJob PrintJob)
+cancelPrintJob(printJob PrintJob)
+notifyStudent(studentID string, message string)
1
1
*
*
Student
-name: String
-id: String
-balance: int
+uploadFile(file: File): void
+verifyFile(file: File): void
+choosePrinter(printer: Printer): void
+createSpecifications(): Specifications
+checkBalance(): int
+createPrintJob(file: File, specifications: Specifications): PrintJob
+cancelPrintJob(printJob: PrintJob): boolean
+ ViewPrintJobs(): printJobs[]
+ sendPrintJob(printjob PrintJob): void
Printer
-id: String
-location: String
-status: String
-availablePaperSizes: String[]
-availableColors: String[]
-printJobs: LinkedList<PrintJob>
+getAvailablePaperSizes(): String[]
+getAvailableColors(): String[]
+getStatus(): String
+setStatus(status String)
+getLocation(): String
+setLocation()
+printDocument(printJob: PrintJob)
+addPrintJob(printJob: PrintJob)
+removePrintJob(printJob: PrintJob)
+cancelPrinting(): void
+notifyAvailability(): void
*
*
1
1
*
Specifications
-paperSize: String
-selectedPages: int[]
-numberOfCopies: int
-isDoubleSided: boolean
-isColored: boolean
-orientation: String
+getPaperSize(): String
+getOrientation(): String
+getIsColored(): boolean
+getIsDoubleSided(): boolean
+getNumberOfCopies(): int
+getSelectedPages(): int[]
+getNumPages(): int
1
1
PrintJob
-id: String
-specification: Specifications
-student: Student
-printer: Printer
-status: String
-file: File
+getStatus(): String
+setStatus(status String): void

3.4 MVP development
(the attached file)
3.5 Architectural diagram



Presentation strategy:
The User Interface is a web-based platform designed to be simple and easy to use, built with a minimalist design philosophy, it emphasizes intuitive navigation, where users can quickly locate functions like initiating print jobs or accessing their account details. The chosen method of implementation uses a modern front-end framework. The UI will have to call the backend server to do actions.
Data Storage Strategy:
For our data storage strategy, we will use hard-coded classes and data structures like arrays to store information and queues to store print jobs. There would be 2 queues used, a waiting queue and print jobs that are processed (like printing, finished or canceled) will be stored in an array. Reports are generated on demand and downloaded directly to the SPSO's device after being generated.
API strategy:
Our application interacts with external services via secure API calls, enabling smooth integration with systems like BKSSO and BKPay. For authentication, we send the username and password to the BKSSO API, receiving a session for access. Similarly, for payments, we use the BKPay API, sending check information and pricing details and awaiting confirmation before proceeding with any further processing.

3.6 Component diagram

The component diagram represents the Student Smart Printing Service, illustrating how students interact with the system to print documents. The Student Interface serves as the access point for uploading documents, setting print specifications, and managing requests. Uploaded files are handled by the Documentation Management component, while the Specification Management allows customization of print settings like color and paper size. Once configured, the Print Job component queues the request, coordinating with Printer Hardware to execute the print. The Cancellation Management handles job cancellations, and the Student Balance Management ensures sufficient funds before processing. This setup enables efficient and streamlined printing for students.