VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

SOFTWARE ENGINEERING (CO3001)

Assignment - Student Smart Printing Service (HCMUT_SSPS)

Advisor:	Trần Trương Tuấn Phát		
Group:	One		
Students:	Nguyễn Ngọc Quý	- ID: 2212870	- L01
	Nguyễn Huỳnh Đức Huy	- ID: 2211218	- L01
	Nguyễn Hoàng Quang Khánh	- ID: 2013458	- L01
	Trần Việt Hoàng	- ID: 2211122	- L01
	Nguyễn Trung Phú	- ID: 2212592	- L01
	Đỗ Đình Thiên Phúc	- ID: 2212610	- L01
	Nguyễn Quốc Khánh	- ID: 2211526	- L01
	Lê Vũ Thanh Hà	- ID: 2320007	- L01

HO CHI MINH CITY, September 2024

Member list & Workload

Task 1

No.	Fullname	Student ID	Tasks
1	Nguyễn Ngọc Quý	2212870	1.1 and 1.2
2	Nguyễn Huỳnh Đức Huy	2211218	1.1 and 1.2
3	Nguyễn Hoàng Quang Khánh	2013458	1.1, 1.2 and report
4	Trần Việt Hoàng	2211122	1.1 and 1.2
5	Nguyễn Trung Phú	2212592	1.1, 1.2 and 1.3
6	Đỗ Đình Thiên Phúc	2212610	1.1 and 1.2
7	Nguyễn Quốc Khánh	2211526	1.1, 1.2 and 1.3
8	Lê Vũ Thanh Hà	2320007	1.1 and 1.2

Task 2

No.	Fullname	Student ID	Tasks
1	Nguyễn Ngọc Quý	2212870	2.3
2	Nguyễn Huỳnh Đức Huy	2211218	2.2
3	Nguyễn Hoàng Quang Khánh	2013458	2.3 and 2.4
4	Trần Việt Hoàng	2211122	2.1
5	Nguyễn Trung Phú	2212592	2.1 and 2.4
6	Đỗ Đình Thiên Phúc	2212610	report
7	Nguyễn Quốc Khánh	2211526	report
8	Lê Vũ Thanh Hà	2320007	2.2

Contents

1 TASK 1

1.1 Introduction

1.1.1 The domain context of Student Smart Printing Service

The Smart Printing Service for Students (HCMUT-SSPS) is an advanced digital platform implemented at Ho Chi Minh City University of Technology (HCMUT). It is designed to streamline the printing process for students, providing them with easy access to printing services across multiple campus locations. This system allows students to submit their documents online and retrieve them from a network of strategically placed printers, reducing the need for students to visit external printing shops or use personal printers.

As universities move towards digitization and a more tech-driven environment, the HCMUT-SSPS fits within the larger framework of digital transformation in education. The system resolves logistical challenges faced by students, such as waiting in long queues or traveling off-campus to print documents, particularly during peak times like exam seasons.

By centralizing the printing services, HCMUT-SSPS also supports the university's efforts to provide a more integrated, efficient, and cost-effective academic support system. The service is accessible through both web and mobile applications, making it highly convenient for students to upload and manage their printing tasks remotely, without needing to be physically present near a printer.

1.1.2 Stakeholder and their needs

Stakeholder	**Expectations/Needs**
Students	Require a convenient, cost-effective, and reliable solution for printing academic documents (assignments, reports, posters). They need easy access from both web and mobile platforms to upload and print documents with little hassle.
University Administration	Needs a system that improves student satisfaction while ensuring resources like paper and printers are optimally used. The administration also requires the ability to generate detailed reports on printer usage to monitor operational efficiency and control costs.
Student Printing Service Officer (SPSO)	Requires tools to manage the system efficiently, including configuring printers, managing printer logs, and ensuring the printers are always functional. They also need real-time usage reports to make informed decisions.
Printer Vendors	Need to receive accurate data on printer usage and performance for timely maintenance and supply management.

1.1.3 Benefits of HCMUT_SSPS:

Stakeholder	Expectations/Needs
Students	A fast and reliable solution that offers transparency in terms of printing costs and allows them to print from any campus location without needing personal printers.
SPSO	The ability to manage and monitor the entire printing system, including handling reports and ensuring printers are in working condition.
University Administration	Enhanced ability to allocate resources effectively, optimize printer usage, and generate comprehensive reports on student printing habits.
Vendors	Improved efficiency in managing printer performance, leading to timely maintenance and reduced downtime.

1.2 Functional and non-functional requirements

1.2.1 Functional requirements

For Students:

- The system must allow document uploads for printing in specific file formats.
- Students should be able to select printers based on availability and location.
- Printing properties such as paper size, number of copies, and single-/double-sided options must be configurable.
- The system should display the remaining page balance and allow students to purchase additional pages through the payment gateway
- Students can view their printing history over a specified period, including details such as printer ID, document name, and timestamps.
- The system must have instructions for use for students.
- Students are allowed to create reports when problems occur with the printer

For SPSO:

- SPSO should be able to add, enable, or disable printers in the system.
- The system must allow configuration of file types for printing.
- SPSO can view and generate reports on printing usage, including detailed student and printer logs.
- The system must enable SPSO to manage the allocation of printing pages and set semester page quotas.
- SPSO must have the ability to send notifications to students if errors occur during printing or if a machine requires maintenance.

For University Administration:

- The system must generate and store monthly and annual reports on printing usage and associated costs.

- It must integrate with the university's Single Sign-On (SSO) service for secure user authentication.
- University administration should have tools to track overall printer usage, monitor print-related expenses, and enforce printing policies.
- Administration has a side where reports from students can be handled

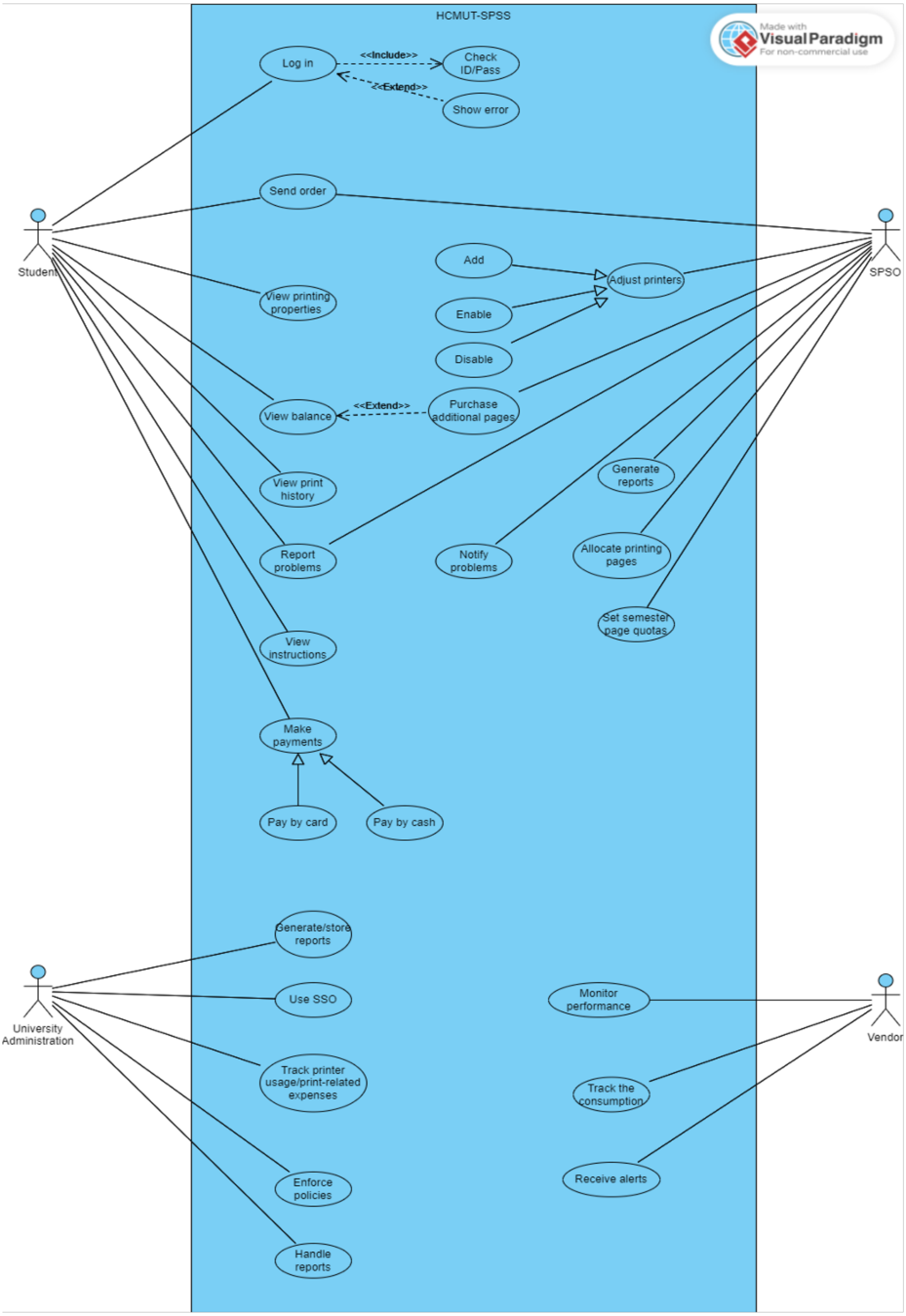
For Printer Vendors:

- Vendors should be able to monitor printer performance, track the consumption of resources (e.g., ink, paper), and receive alerts when maintenance is required.

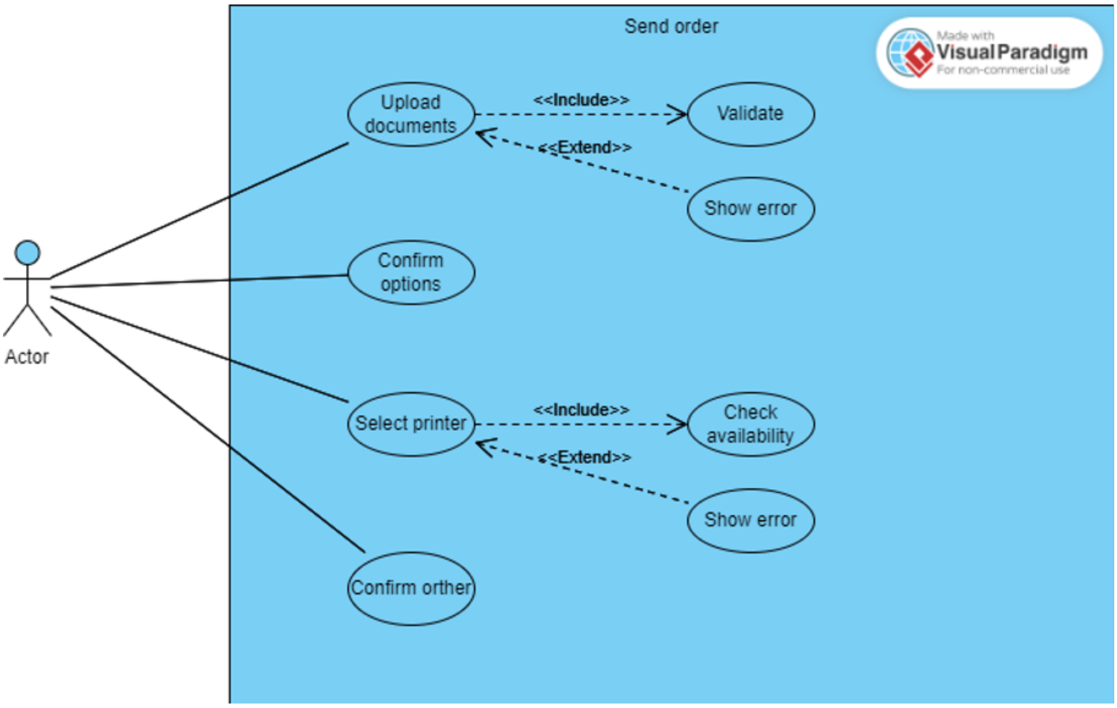
1.2.2 Non-functional requirements

- **Usability:** The system must provide an intuitive user interface, accessible via both web and mobile platforms, ensuring ease of use for students and SPSO.
- **Scalability:** The system should accommodate an increasing number of students and printers over time without significant performance degradation.
- **Security:** Secure data encryption must be implemented for sensitive transactions, such as payment processing and user authentication via HCMUT_SSO.
- **Performance:** The system must process print jobs and transactions within 5 seconds to provide a responsive user experience.
- **Availability:** The system must maintain a 99.9% uptime, ensuring continuous availability.
- **Reliability:** The system should maintain accurate logs and reports, with automatic backups to ensure data integrity.

1.3 Use-case diagram of the system


HCMUT-SPSS
Made with Visual Paradigm For non-commercial use
Log in
<<Include>>
Check ID/Pass
<<Extend>>
Show error
Send order
Student
SPSO
Add
Adjust printers
View printing properties
Enable
Disable
View balance
<<Extend>>
Purchase additional pages
View print history
Generate reports
Report problems
Notify problems
Allocate printing pages
Set semester page quotas
View instructions
Make payments
Pay by card
Pay by cash
Generate/store reports
Use SSO
University Administration
Monitor performance
Vendor
Track printer usage/print-related expenses
Track the consumption
Enforce policies
Receive alerts
Handle reports

1.4 Use-case diagram



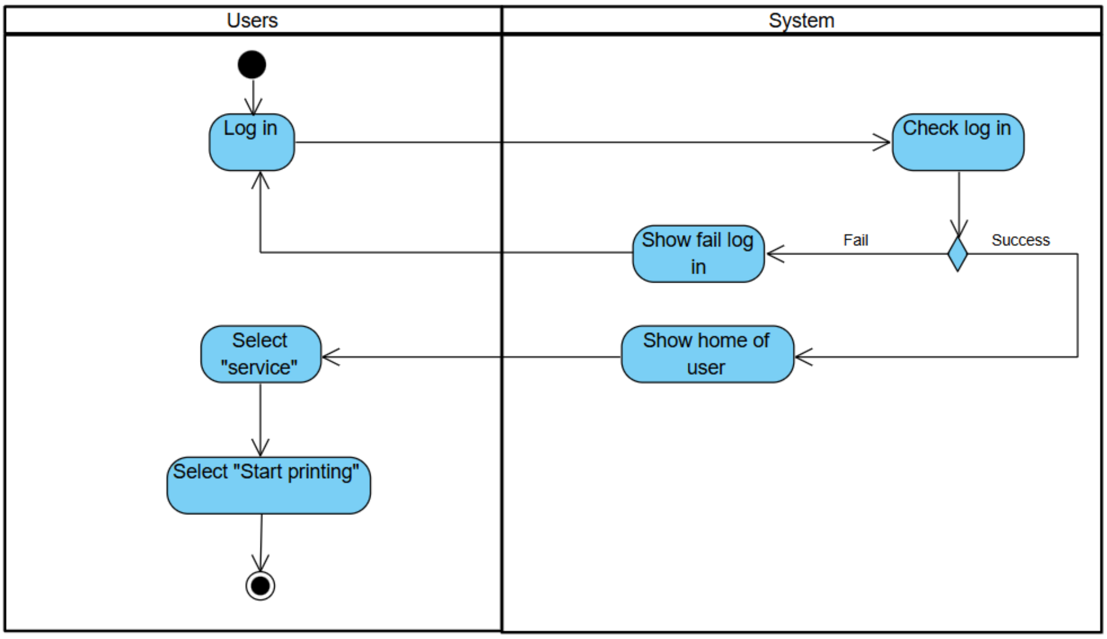
Use-case ID	U1
Use-case name	Send order
Use-case description	Students submit documents through the system for printing, select print options, confirm information, and select printers
Actor	Students
Preconditions	The student is logged in The document to be printed is ready for upload.
Trigger	Click "upload documents" button in Print page
Steps	1. Upload the document to be printed. 2. The system displays the printing options. 3. Students choose the options (color, number of plates, duplex). 4. Students select printers based on availability and location. 5. Students confirm information. 6. The system confirms the print order and provides the receiving time.
Post Condition	The print order has been sent and the student receives a confirmation notification.
Exception flow	1. If students select the printer that is not available, the system will notify and return to the confirmation step. 2. If the uploaded file is invalid, the system asks the student to redownload the document in the correct format.

2 TASK 2

2.1 Activity diagram to capture the business process between systems and the stakeholders in Task Assignment module

To understand the "send order" process in this system, we can divide the process into three main steps: log in and select service, upload file, and select option.

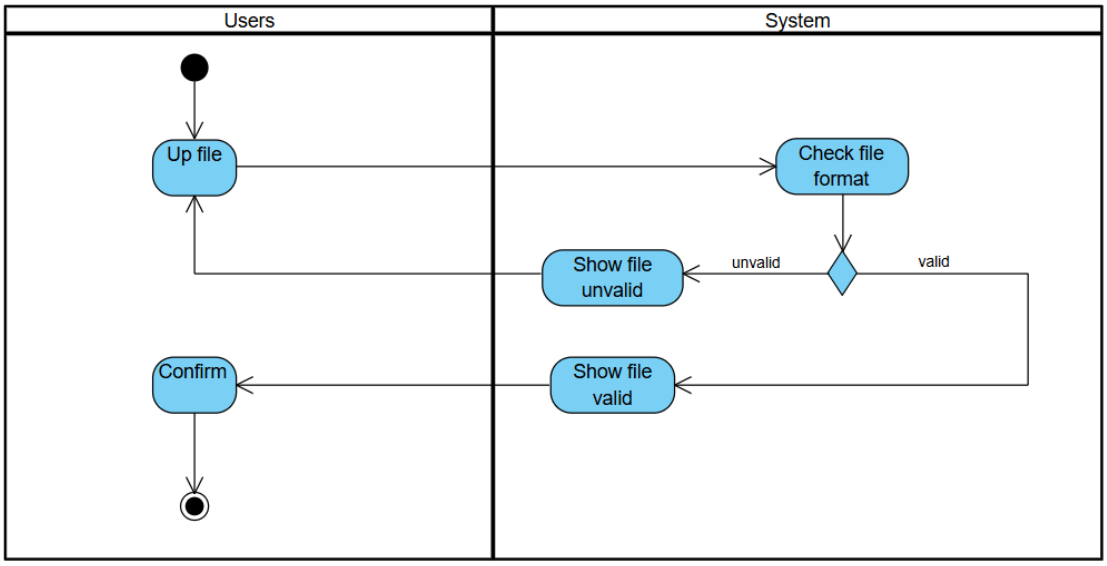
2.1.1 Log in and select service



Description

The activity diagram describes the process of user login and service usage in a system. First, the user performs the "Log in" operation. The system then checks the login information. If the login fails, the system displays the message "Show failed log in" and the user needs to try again. If the login is successful, the system takes the user to the "Show home of user" main screen. From here, the user can select "Select service", followed by "Select Start printing" to start using the printing service. The process ends when the selected service is completed.

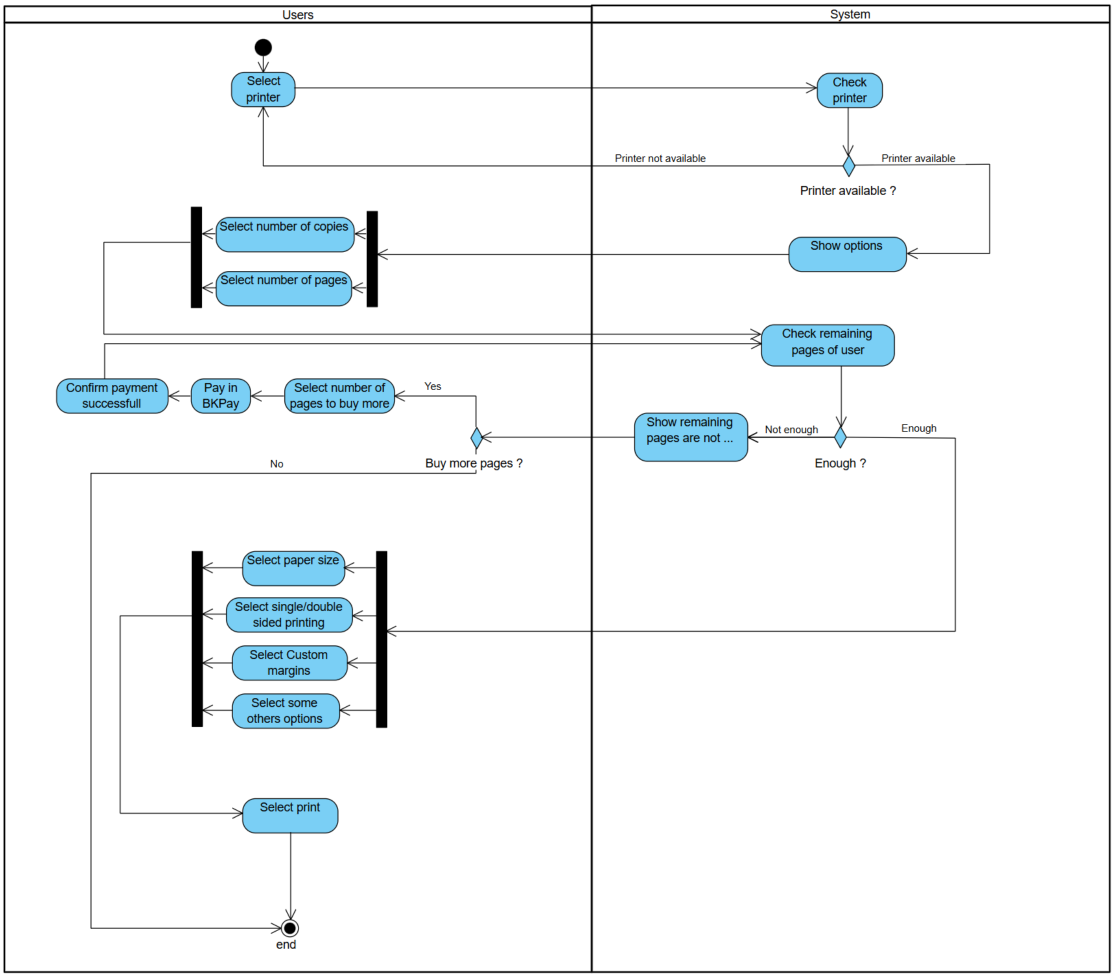
2.1.2 Upload file



Description

This activity diagram describes the process of uploading and checking files in the system. First, the user performs the "Up file" operation. The system will check the file format with "Check file format". If the file is invalid, the system displays the message "Show file invalid" and the user needs to upload the file again. If the file is valid, the system displays "Show file valid", then the user can proceed to the "Confirm" step to complete the process. The process ends after confirmation.

2.1.3 Select options



Description

The activity diagram illustrates the process of a user selecting a printer and printing in the system. First, the user starts by selecting a printer. The system checks the availability of the printer. If the printer is not available, the user needs to select it again. Otherwise, if the printer is available, the system displays options to the user. Next, the user selects the number of copies and pages to print. The system checks if the user has enough pages to print. If not, the system notifies and asks the user if they want to purchase more pages. If the user decides to purchase more, they select the number of pages to purchase and make the payment via BKPay. After the payment is successful, the process returns to the step of selecting the number of copies and pages. Once the number of pages is sufficient, the user can continue by selecting the paper size, single-sided or double-sided printing mode, custom margins, and other options. Finally, after completing the selections, the user performs the print command and the process ends after printing is complete.

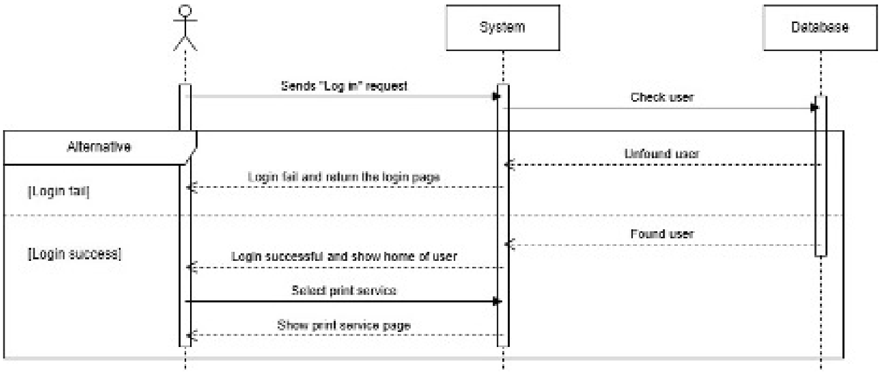
2.2 Sequence diagram

Using module “Send order” in task 2.1, we have 3 sequence diagram : log in, upload file, and select option.

2.2.1 Log in

This sequence diagram illustrates the login process for a user on a system that communicates with a database to validate credentials.

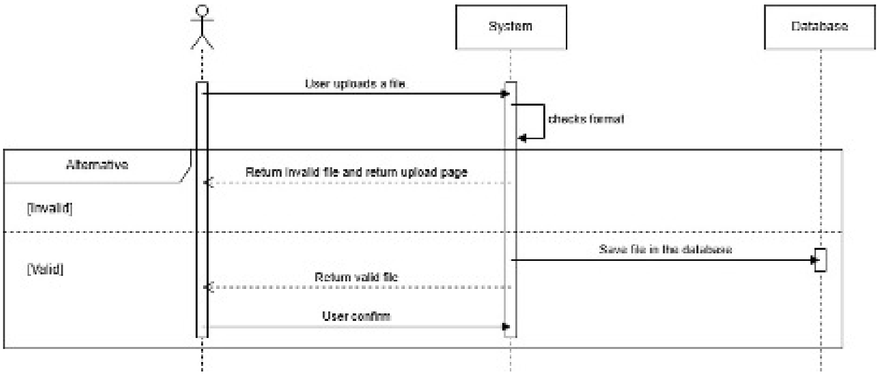
The user sends a "Log in" request to the system, signaling an attempt to access their account. Upon receiving the login request, the system queries the database to check for the existence of the user’s credentials. If the user is found in the database, the database responds to the system with a "Found user" message. If the user is not found, the database responds with an "Not found user" message. If the user is not found, the system returns a "Login fail" message and redirects the user back to the login page. If the user is found, the system returns a "Login successful" message to the user and directs to home of user page. User select print service. System shows print service page to user.



2.2.2 Upload file

This sequence diagram illustrates the upload process for a user on a system that communicates with a database to save file

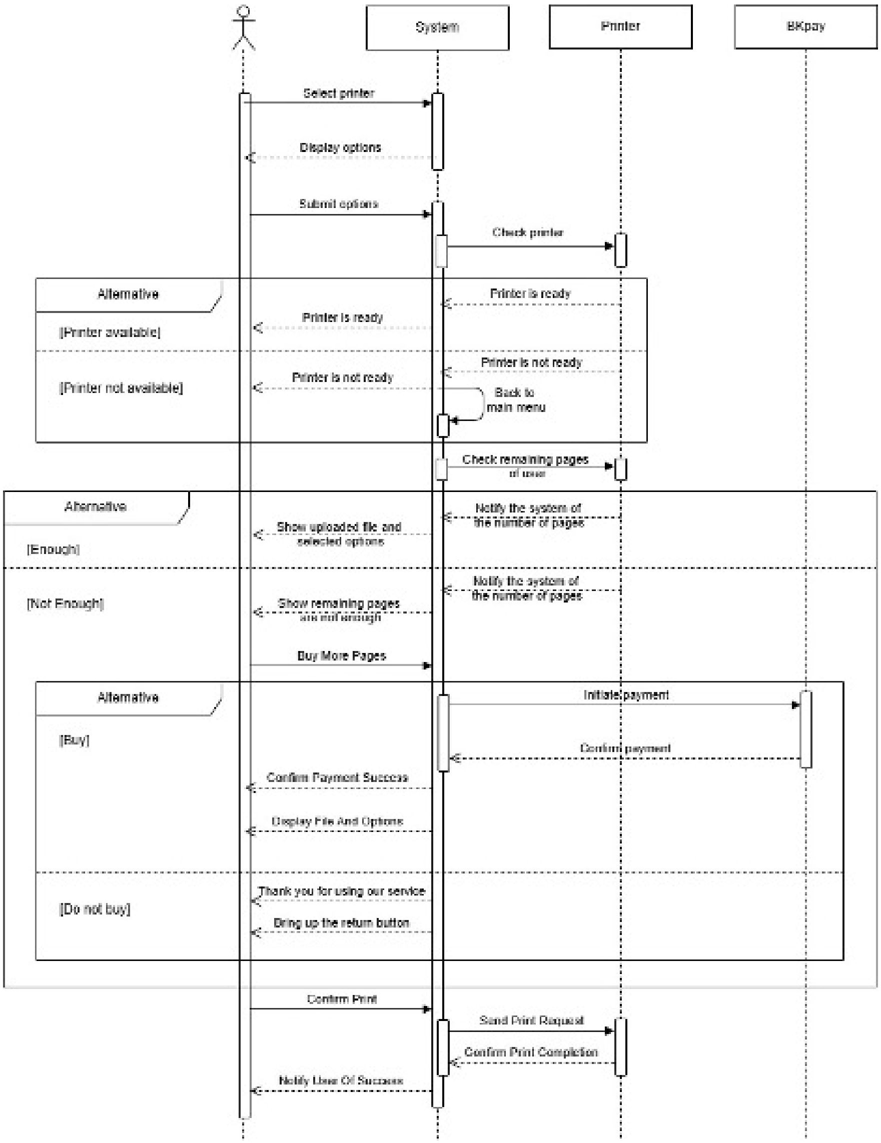
User upload a file to system. Then system check if the format of the file is valid. If the format is invalid return the message invalid file to user and redirects the user back to the upload page. If the format is valid, save the file in the database then return the message valid file to user. User confirm to end process.



2.2.3 Select options

This sequence diagram illustrates the select option to print process for a user on a system that communicates with printer and bkpay.

User select printer from system then system displays printing options to User (e.g., paper size, number of copies, single/double-sided printing, number of pages, etc.). User sets their desired options and submits the request. System checks the status of the Printer to verify if it is available. If it is unavailable, system redirects the user back to the choose printer page. Then system check the user's remaining print page. If the System finds not enough pages, System displays a message to User remaining pages are not enough. System redirects User to BKPay for payment. If user is not buy more page, return message "Thank you" and bring up the home button. If user is buy more page, User completes the payment through BKPay. After a successful payment, BKPay confirms the payment with the System. System updates the User's page balance and allows User to proceed with printing. If the System finds enough pages and confirms the Printer is working, System displays the uploaded file along with the selected printing options. User reviews and confirms to print. System sends the print job to the Printer. Printer executes the print job. After printing is complete, System confirms successful printing to User.


System
Printer
BKpay
Select printer
Display options
Submit options
Check printer
Alternative
[Printer available]
Printer is ready
Printer is ready
[Printer not available]
Printer is not ready
Printer is not ready
Back to
main menu
Check remaining pages
of user
Alternative
[Enough]
Notify the system of
the number of pages
Show uploaded file and
selected options
[Not Enough]
Notify the system of
the number of pages
Show remaining pages
are not enough
Buy More Pages
Alternative
[Buy]
Initiate payment
Confirm payment
Confirm Payment Success
Display File And Options
[Do not buy]
Thank you for using our service
Bring up the return button
Confirm Print
Send Print Request
Confirm Print Completion
Notify User Of Success

2.3 Class diagram of Task Assignment module

Description

A. Login

1. User

   (a) Attitude

   - fileId: String
     Unique identifier for the file.
   - fileName: String
     Name of the file.
   - fileSize: Integer
     Size of the file in bytes.
   - fileType: String
     Type or format of the file (e.g., .pdf, .jpg).
   - uploadedBy: String
     Identifier of the user who uploaded the file.
   - uploadedAt: Date
     Date and time when the file was uploaded.

   (b) Method

   - uploadFile(fileName: String, fileSize: Integer, fileType: String, uploadedBy: String): Boolean
     Uploads a file to the system and returns true if the upload is successful, otherwise returns false.
   - deleteFile(fileId: String): Boolean
     Deletes the specified file from the system and returns true if the deletion is successful, otherwise returns false.
   - getFileDetails(fileId: String): File
     Retrieves the details of the specified file and returns a File object containing its attributes.

2. Session

   (a) Attitude

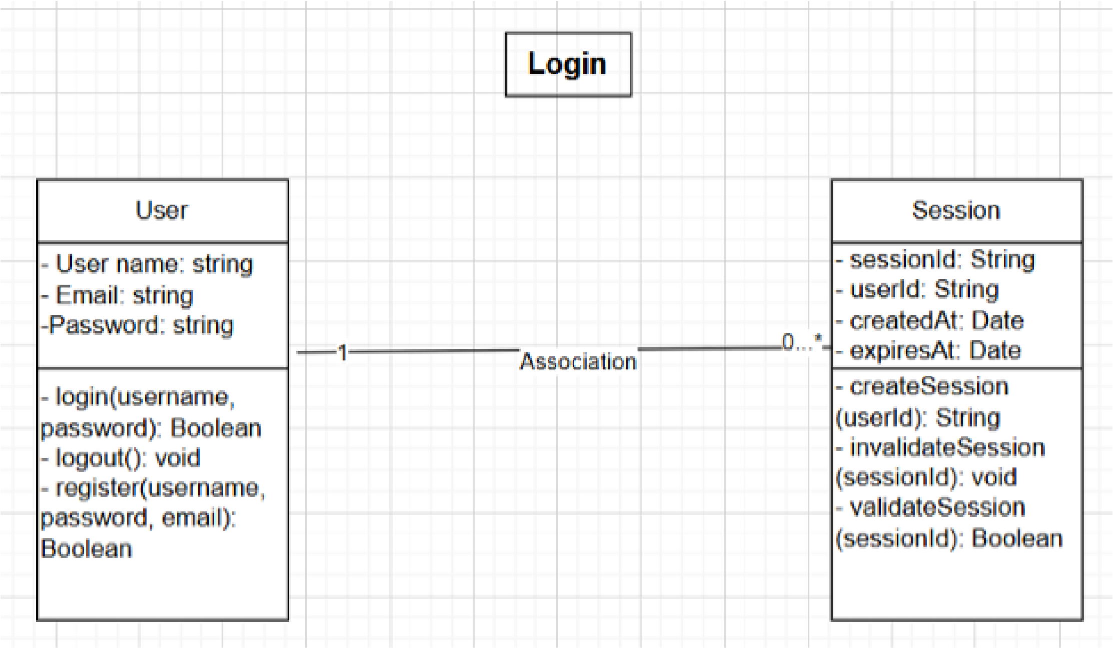
   - sessionId: string: A unique identifier for the session. This ID is generated when a session is created and is used to track the user's active session.
   - userId: string: The unique identifier of the user associated with this session. This links the session to a specific user in the system.
   - createdAt: Date: The date and time when the session was created. This is used to determine the age of the session.
   - expiresAt: Date: The date and time when the session will expire. This is set based on the application's session timeout policy and is used to enforce session validity.

   (b) Method

   - createSession(userId: String): string: This method creates a new session for the specified user. It generates a unique session ID, records the creation time, and sets the expiration time based on predefined rules. The method returns the newly created session ID.

- invalidateSession(sessionId: void): void: This method terminates the specified session, effectively logging the user out. It removes the session from active sessions, ensuring that the session ID can no longer be used for authentication.
- validateSession(sessionId: Boolean): string: This method checks if the provided session ID is valid. It verifies whether the session exists and has not expired, returning true if the session is valid and false otherwise.

Class diagram


Login
User
- User name: string
- Email: string
-Password: string
- login(username, password): Boolean
- logout(): void
- register(username, password, email): Boolean
1
Association
0...*
Session
- sessionId: String
- userId: String
- createdAt: Date
- expiresAt: Date
- createSession (userId): String
- invalidateSession (sessionId): void
- validateSession (sessionId): Boolean

B. Upload File

1. File

   (a) Attributes

   - fileId: String
     Unique identifier for the file.
   - fileName: String
     Name of the file.
   - fileSize: Integer
     Size of the file in bytes.
   - fileType: String
     Type or format of the file (e.g., .pdf, .jpg).
   - uploadedBy: String
     Identifier of the user who uploaded the file.
   - uploadedAt: Date
     Date and time when the file was uploaded.

   (b) Methods

   - uploadFile(fileName: String, fileSize: Integer, fileType: String, uploadedBy: String): Boolean
     Uploads a file to the system and returns true if the upload is successful, otherwise returns false.
   - deleteFile(fileId: String): Boolean
     Deletes the specified file from the system and returns true if the deletion is successful, otherwise returns false.
   - getFileDetails(fileId: String): File
     Retrieves the details of the specified file and returns a File object containing its attributes.

2. Storage

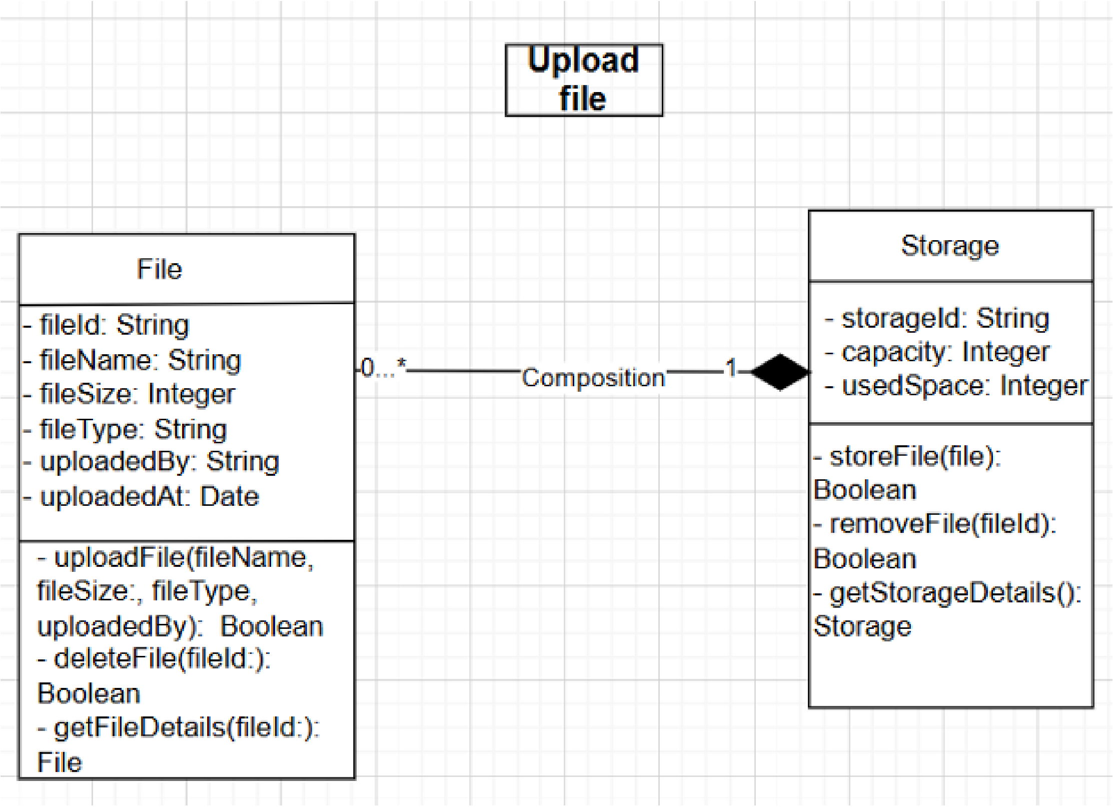
   (a) Attributes

   - storageId: String
     Unique identifier for the storage.
   - capacity: Integer
     Total capacity of the storage in bytes.
   - usedSpace: Integer
     Amount of space currently used in the storage in bytes.

   (b) Methods

   - storeFile(file: File): Boolean
     Stores the specified file in the storage and returns true if the operation is successful, otherwise returns false.
   - removeFile(fileId: String): Boolean
     Removes the file with the specified ID from the storage and returns true if the operation is successful, otherwise returns false.
   - getStorageDetails(): Storage
     Retrieves the details of the storage and returns a Storage object containing its attributes.

Class diagram


Upload file
File
- fileId: String
- fileName: String
- fileSize: Integer
- fileType: String
- uploadedBy: String
- uploadedAt: Date
- uploadFile(fileName, fileSize:, fileType, uploadedBy): Boolean
- deleteFile(fileId:): Boolean
- getFileDetails(fileId:): File
0...*
Composition
1
Storage
- storageId: String
- capacity: Integer
- usedSpace: Integer
- storeFile(file): Boolean
- removeFile(fileId): Boolean
- getStorageDetails(): Storage

C. Select Option

1. User

   (a) Attitude

   - userID: string - A unique identifier for each user.
   - pageBalance: int - Tracks the number of pages the user has available for printing.

   (b) Method

   - selectPrinter(): Printer - Allows the user to choose a printer from the list of available printers.
   - selectOptions(options: Object): void - Lets the user choose print options (e.g., number of copies, number of pages, single/double-sided printing).
   - confirmPrint(): Boolean - Submits the print request after all options have been selected and payment (if required) is made.
   - payForPages(amount: double): void - Redirects to the payment process if the user needs more pages for printing.
   - buyMorePages(pages: int): void - A method to increase the user's page balance when their current balance is insufficient.

2. Printer

   (a) Attitude

   - printerID: string — A unique identifier for each printer.
   - status: string — The current status of the printer (e.g., "available", "unavailable").
   - availablePages: int — The total number of pages that can be printed on this printer before needing a refill.

   (b) Method

   - checkAvailability(): Boolean — Checks if the printer is currently available. Returns true if available.
   - printDocument(document: Document): void — Sends a document to the printer if it's available and ready for printing.

3. Payment

   (a) Attitude

   - paymentID: string — A unique identifier for each payment.
   - amount: double — The amount of money required for the transaction (e.g., purchasing more pages).

   (b) Method

   - processPayment(): Boolean — Executes the payment process, updating the user's page balance once the payment is successful. Returns true if payment is processed successfully.

4. Document

   (a) Attitude

   - documentID: string — A unique identifier for each document.

- file: File — Represents the file that the user wants to print (e.g., PDF, DOC).
- pageCount: int — The number of pages in the document.
- printOptions: Map<String, String> — Stores options selected by the user (e.g., paper size, single/double-sided printing, margins).

(b) Method

- uploadFile(file: File): void — Uploads the file that the user wants to print.
- selectPrintOptions(options: Map<String, String>): void — Allows the user to choose specific options for printing the document (e.g., paper size, single-sided or double-sided printing).

5. PrintRequest

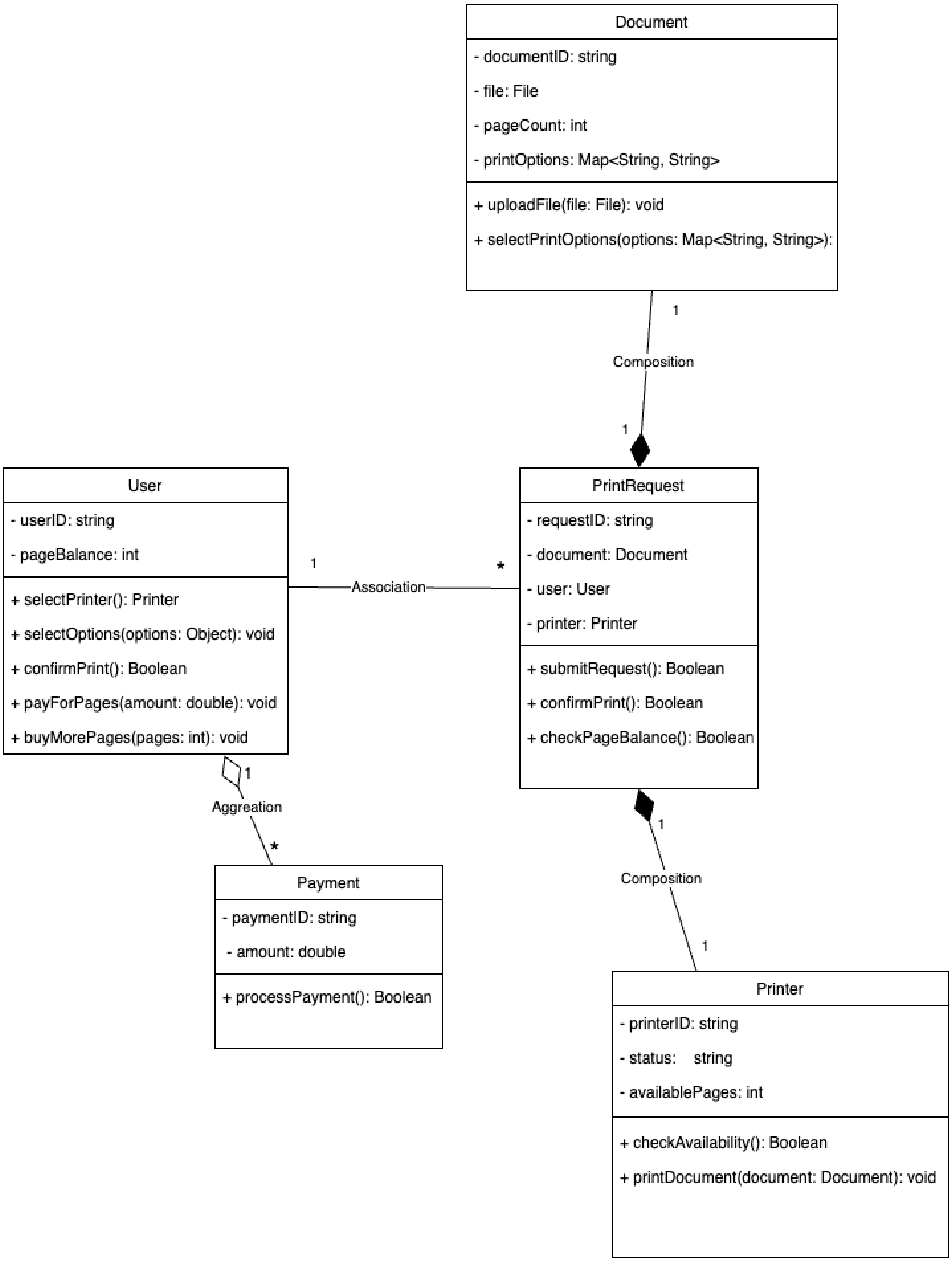
(a) Attitude

- requestID: string — A unique identifier for each print request.requestID: String — A unique identifier for each print request.
- document: Document — The document that needs to be printed.
- user: User — The user who initiated the print request.
- printer: Printer — The printer assigned to handle this print request.

(b) Method

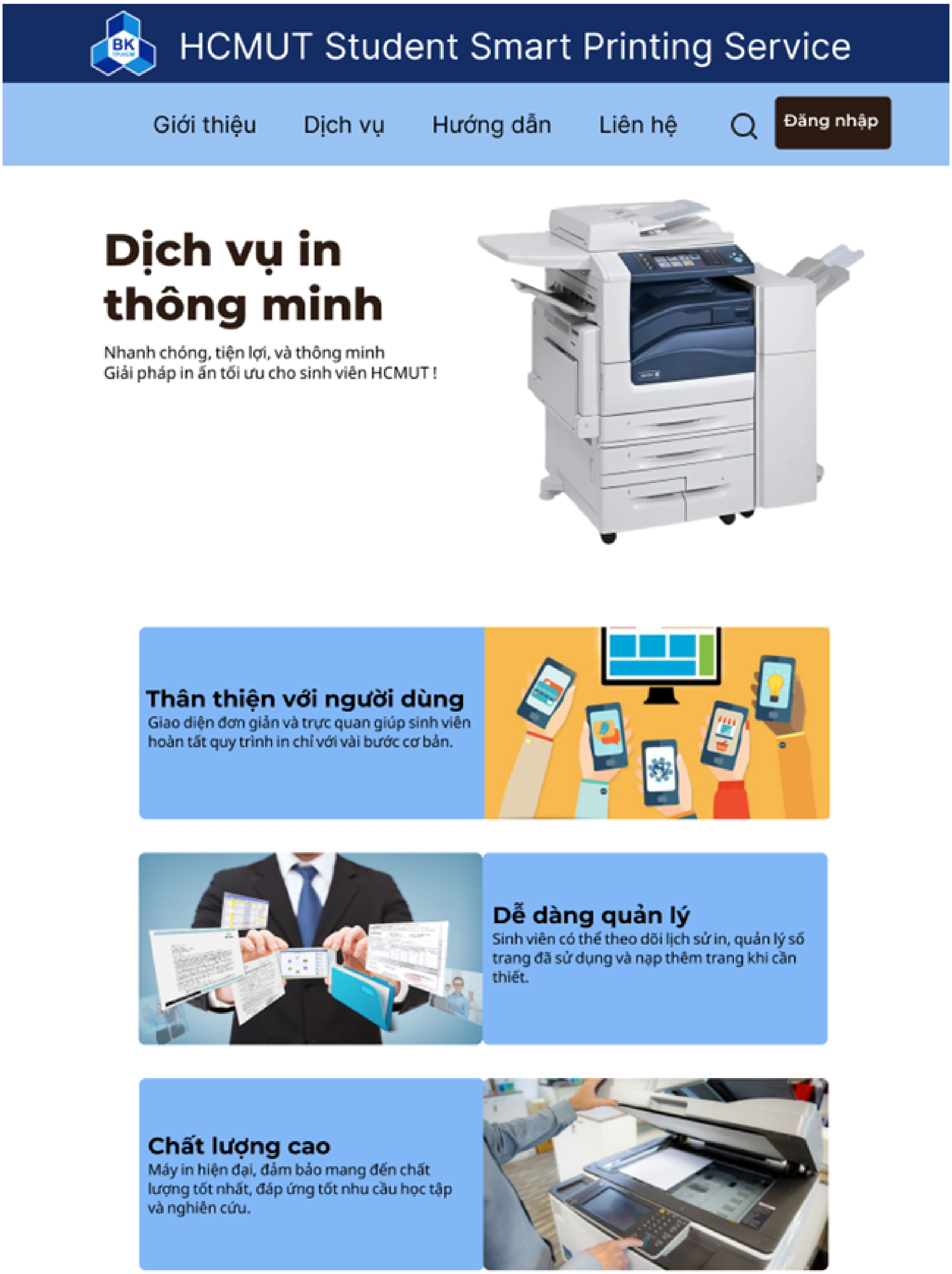
- submitRequest(): Boolean — Submits the print request after the user has chosen all options. Returns true if submission is successful.
- confirmPrint(): Boolean — Confirms that the print job is ready to be processed by the printer. Returns true if confirmed.
- checkPageBalance(): Boolean - Check whether the user's remaining page count is greater than or equal to the number of pages they requested to print

Class diagram


Document
- documentID: string
- file: File
- pageCount: int
- printOptions: Map<String, String>
+ uploadFile(file: File): void
+ selectPrintOptions(options: Map<String, String>):
1
Composition
1
User
- userID: string
- pageBalance: int
+ selectPrinter(): Printer
+ selectOptions(options: Object): void
+ confirmPrint(): Boolean
+ payForPages(amount: double): void
+ buyMorePages(pages: int): void
1
Association
*
PrintRequest
- requestID: string
- document: Document
- user: User
- printer: Printer
+ submitRequest(): Boolean
+ confirmPrint(): Boolean
+ checkPageBalance(): Boolean
1
Aggregation
*
Payment
- paymentID: string
- amount: double
+ processPayment(): Boolean
1
Composition
1
Printer
- printerID: string
- status: string
- availablePages: int
+ checkAvailability(): Boolean
+ printDocument(document: Document): void

2.4 User Interface

2.4.1 Home_page



HCMUT Student Smart Printing Service

Giới thiệu Dịch vụ Hướng dẫn Liên hệ Đăng nhập

Dịch vụ in thông minh

Nhanh chóng, tiện lợi, và thông minh
Giải pháp in ấn tối ưu cho sinh viên HCMUT !

Thân thiện với người dùng

Giao diện đơn giản và trực quan giúp sinh viên hoàn tất quy trình in chỉ với vài bước cơ bản.

Dễ dàng quản lý

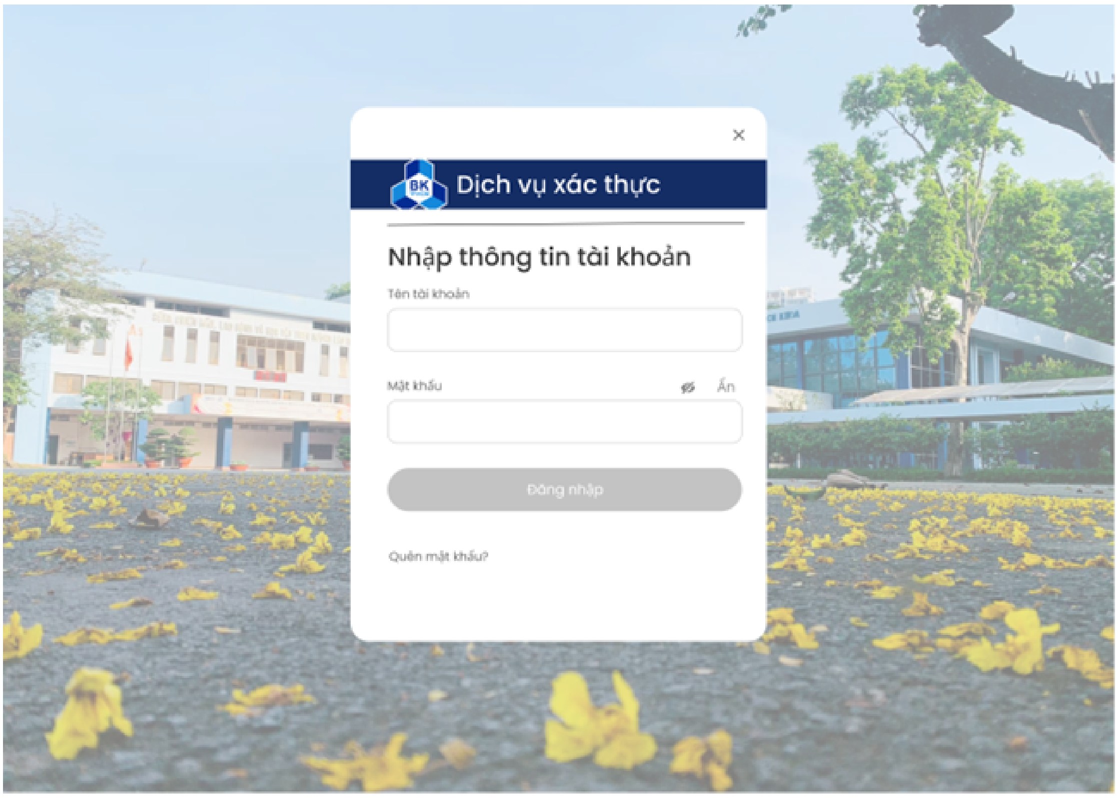
Sinh viên có thể theo dõi lịch sử in, quản lý số trang đã sử dụng và nạp thêm trang khi cần thiết.

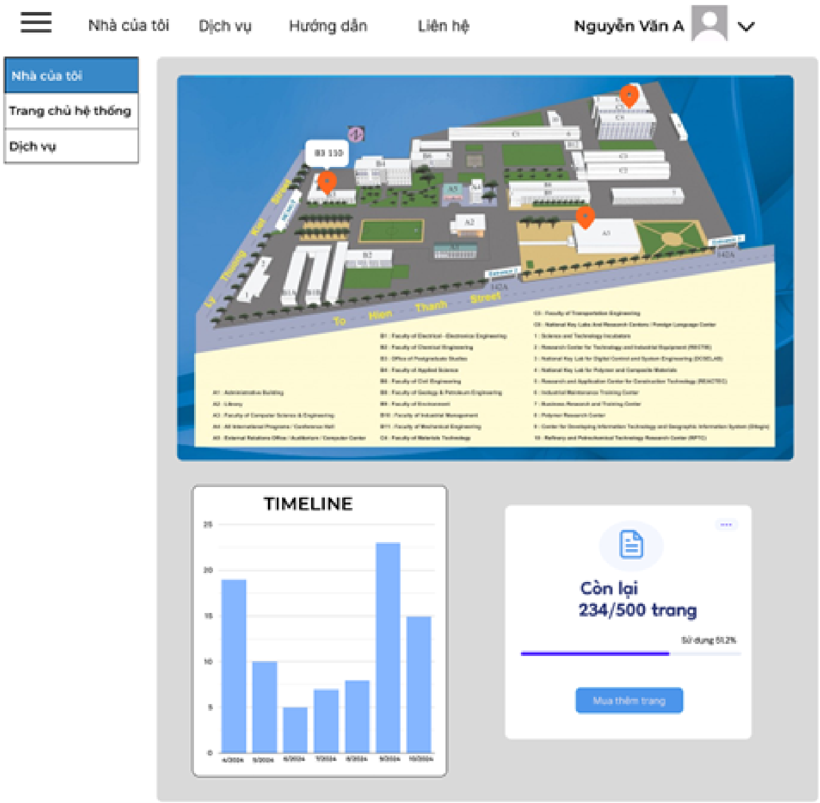
Chất lượng cao

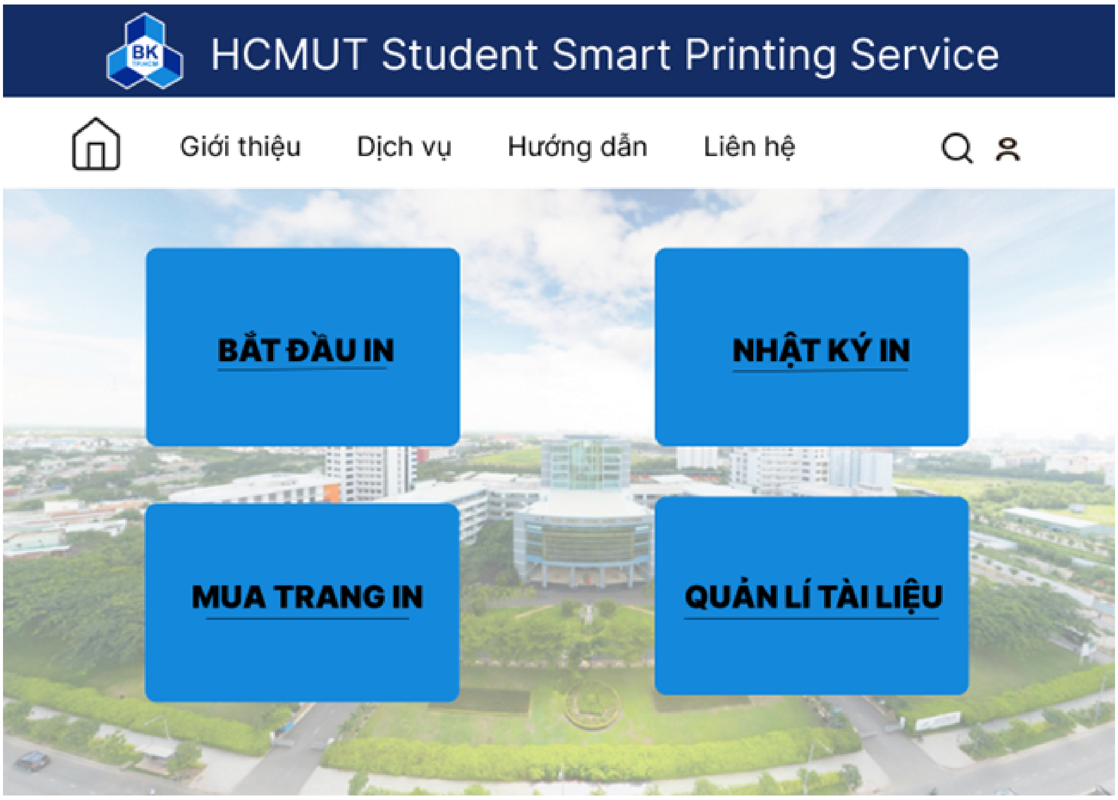
Máy in hiện đại, đảm bảo mang đến chất lượng tốt nhất, đáp ứng tốt nhu cầu học tập và nghiên cứu.

This website provides smart printing services for HCMUT students, with the aim of providing optimal printing solutions. The interface is designed to be friendly and intuitive, helping students easily operate and manage the printing process with just a few basic steps. In addition, students can easily monitor and manage printing status, allowing them to adjust the number of documents when necessary. The service ensures high quality printing, well meeting the learning and research needs of students. The website also includes sections such as Introduction, Services, Instructions, Contact and has a Login button for easy access.

2.4.2 Log in and select service

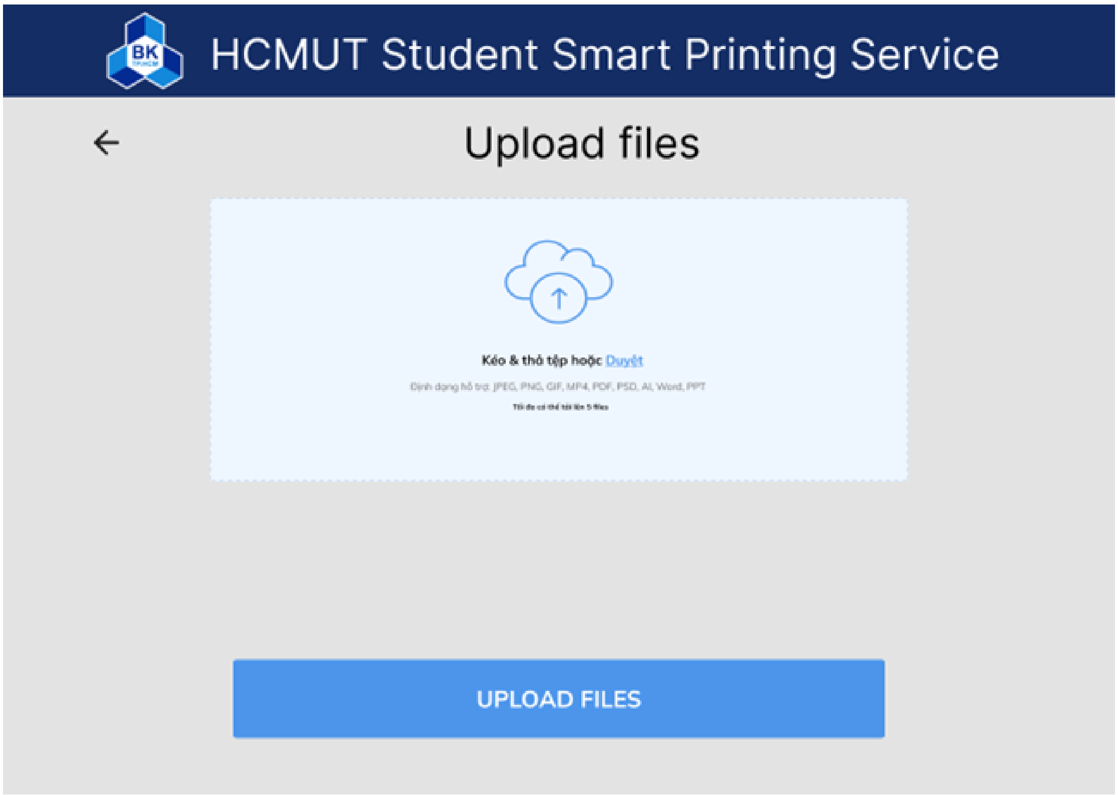



Nhà của tôi
Dịch vụ
Hướng dẫn
Liên hệ
Nguyễn Văn A
Nhà của tôi
Trang chủ hệ thống
Dịch vụ
B3 110
Ly Thuong Kiet Street
To Hien Thanh Street
TIMELINE
25
20
15
10
5
0
4/2024
5/2024
6/2024
7/2024
8/2024
9/2024
10/2024
Còn lại
234/500 trang
Sử dụng 51.2%
Mua thêm trang



The login page allows users to enter their account information to access the system. After logging in, users are taken to their personal home page, which displays a map of printers, usage frequency, and tracking of pages used. Next, in the service section, there are a number of services such as starting printing, printing logs, purchasing additional pages, and document management, helping students easily perform printing-related operations effectively.

2.4.3 Upload files



HCMUT Student Smart Printing Service

Upload files

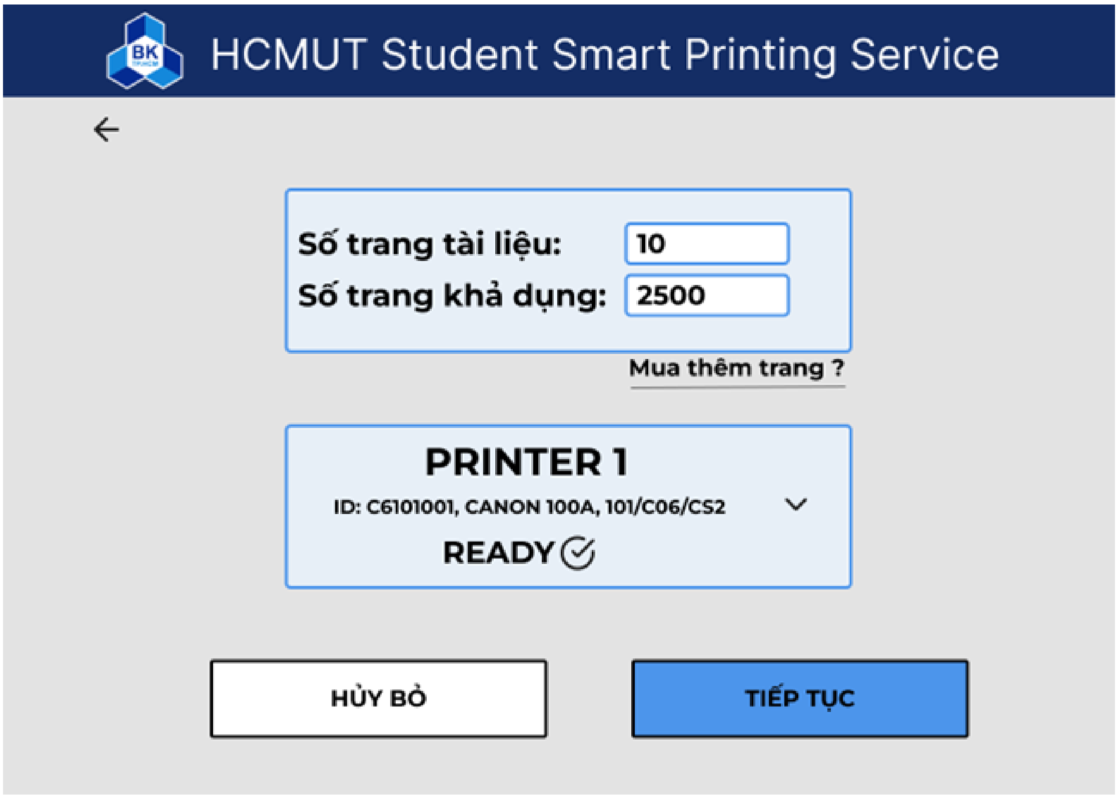
Kéo & thả tệp hoặc Duyệt

Định dạng hỗ trợ: JPEG, PNG, GIF, MP4, PDF, PSD, AI, Word, PPT

Tối đa có thể tải lên 5 files

The website allows users to upload files to be printed in various formats such as JPEG, PNG, PDF, Word, PowerPoint, etc. Students can drag and drop files directly into the designated area or click the "Browse" button to select files from their computer. Currently, the system allows users to upload up to 5 files. After uploading, students can proceed to the next steps to complete the printing process.

2.4.4 Select options

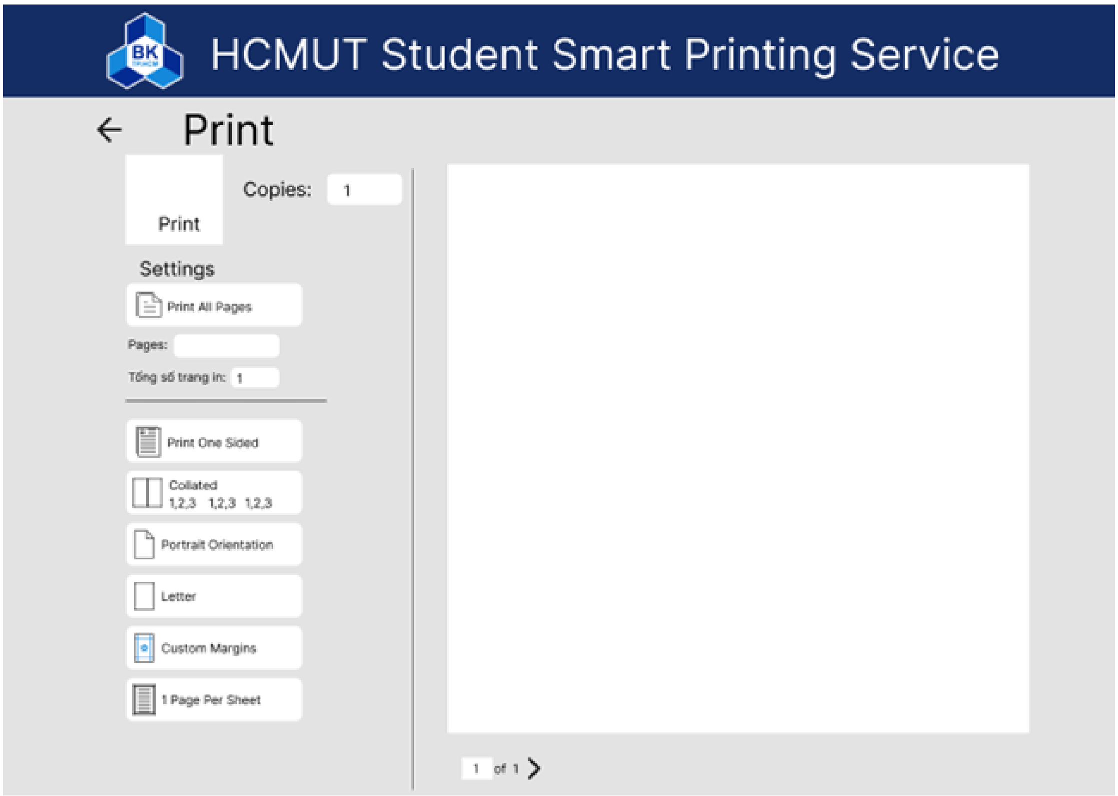


HCMUT Student Smart Printing Service

Số trang tài liệu: 10

Số trang khả dụng: 2500

This is the next interface in the process of using the smart printing service. After the user has uploaded the file, the user will be transferred to the screen showing the number of pages in the file, the number of pages remaining for the user. If you want to buy more, you can also do it at this step. Next, choose a printer that is suitable for the user's location



HCMUT Student Smart Printing Service

Print

Copies: 1

Settings

Print All Pages

Pages:

Tổng số trang in: 1

Print One Sided

Collated
1,2,3 1,2,3 1,2,3

Portrait Orientation

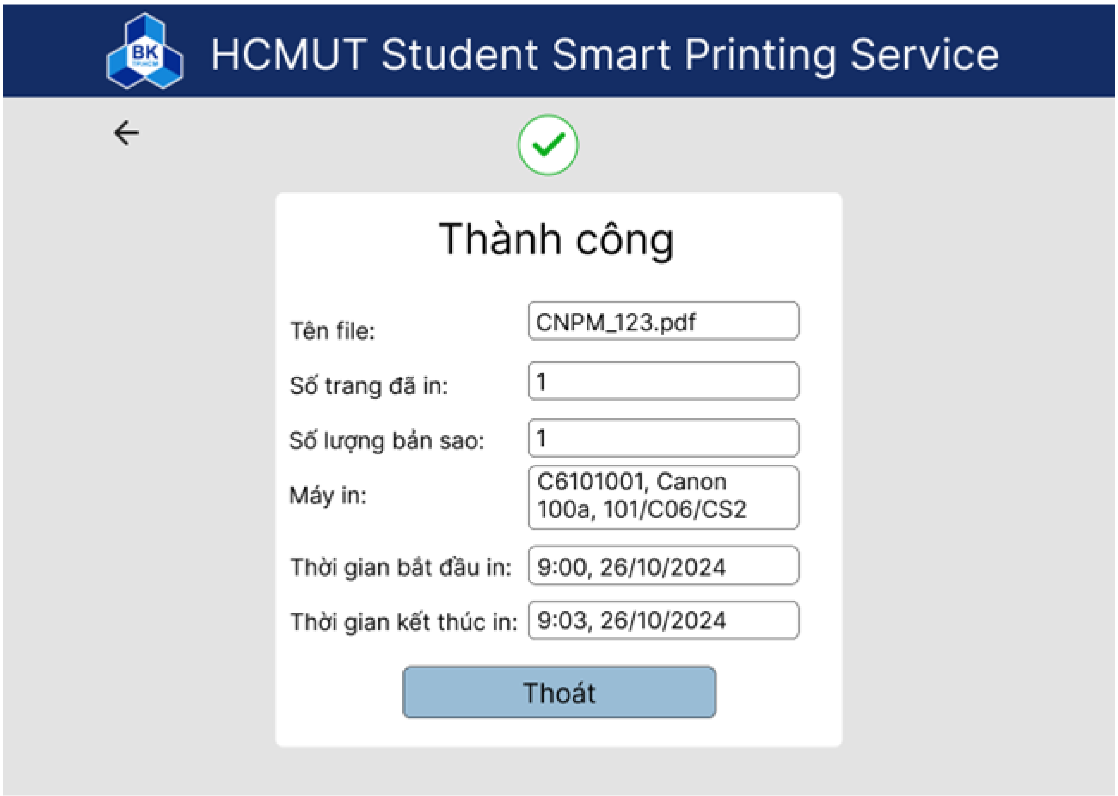
Letter

Custom Margins

1 Page Per Sheet

1 of 1

The user will be transferred to this page to set up printing parameters such as: number of copies, page selection, single-sided or double-sided printing, page arrangement, paper format, print margins, and number of pages per sheet of paper. These options help the user customize the print according to their requirements before sending it for printing



HCMUT Student Smart Printing Service

Thành công

Tên file: CNPM_123.pdf

Số trang đã in: 1

Số lượng bản sao: 1

Máy in: C6101001, Canon 100a, 101/C06/CS2

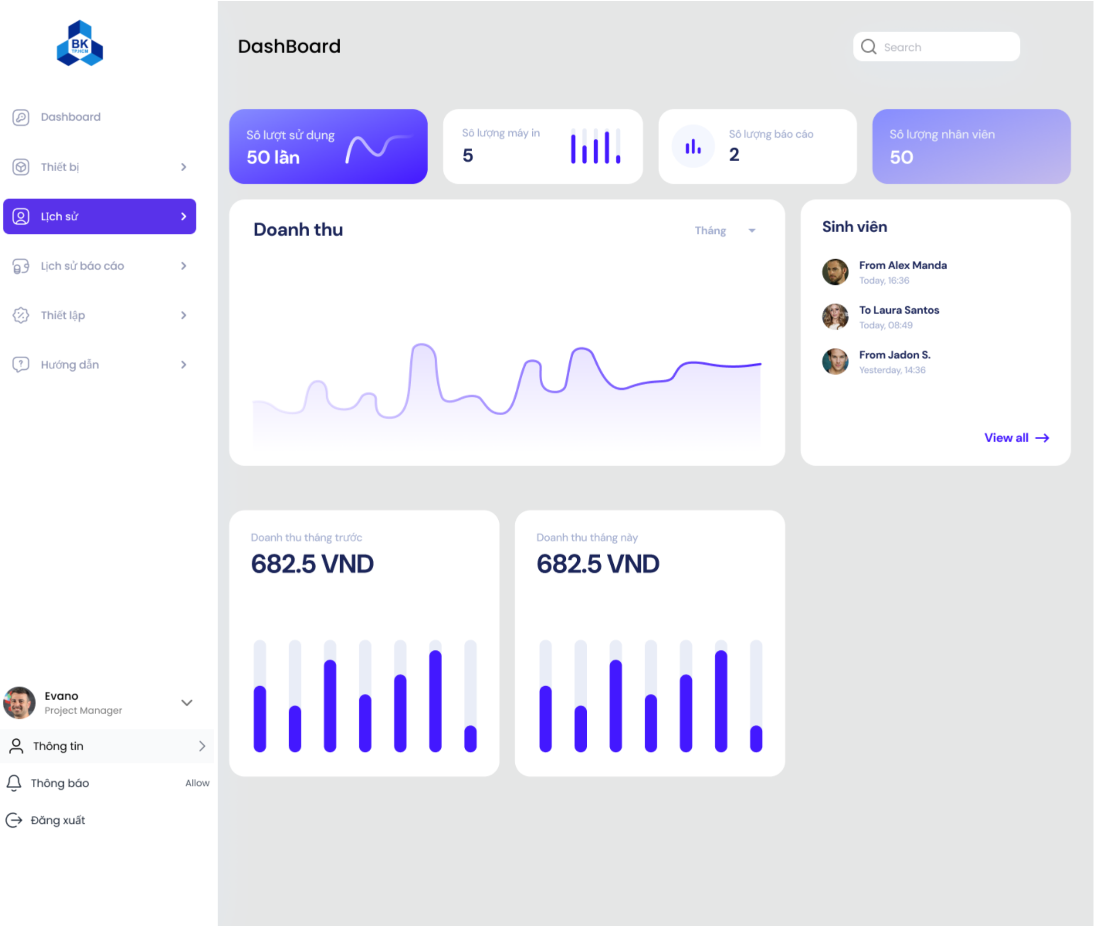
Thời gian bắt đầu in: 9:00, 26/10/2024

Thời gian kết thúc in: 9:03, 26/10/2024

After completing the printing process, the user will receive a confirmation message on the screen. This interface will provide full information about the order, including the file name, number of printed pages, number of copies, the printer used, and the start and end time of the printing process. This allows users to easily review all information and ensure that the document has been printed correctly. When they want to view other information or perform further operations, they simply press to return to the home page.

2.4.5 UI of admin

1. UI Dashboard: With this interface, managers can see charts representing statistical data about the software's performance.



DashBoard

Số lượt sử dụng
50 lần

Số lượng máy in
5

Số lượng báo cáo
2

Số lượng nhân viên
50

Doanh thu

Sinh viên

From Alex Manda
Today, 16:36

To Laura Santos
Today, 08:49

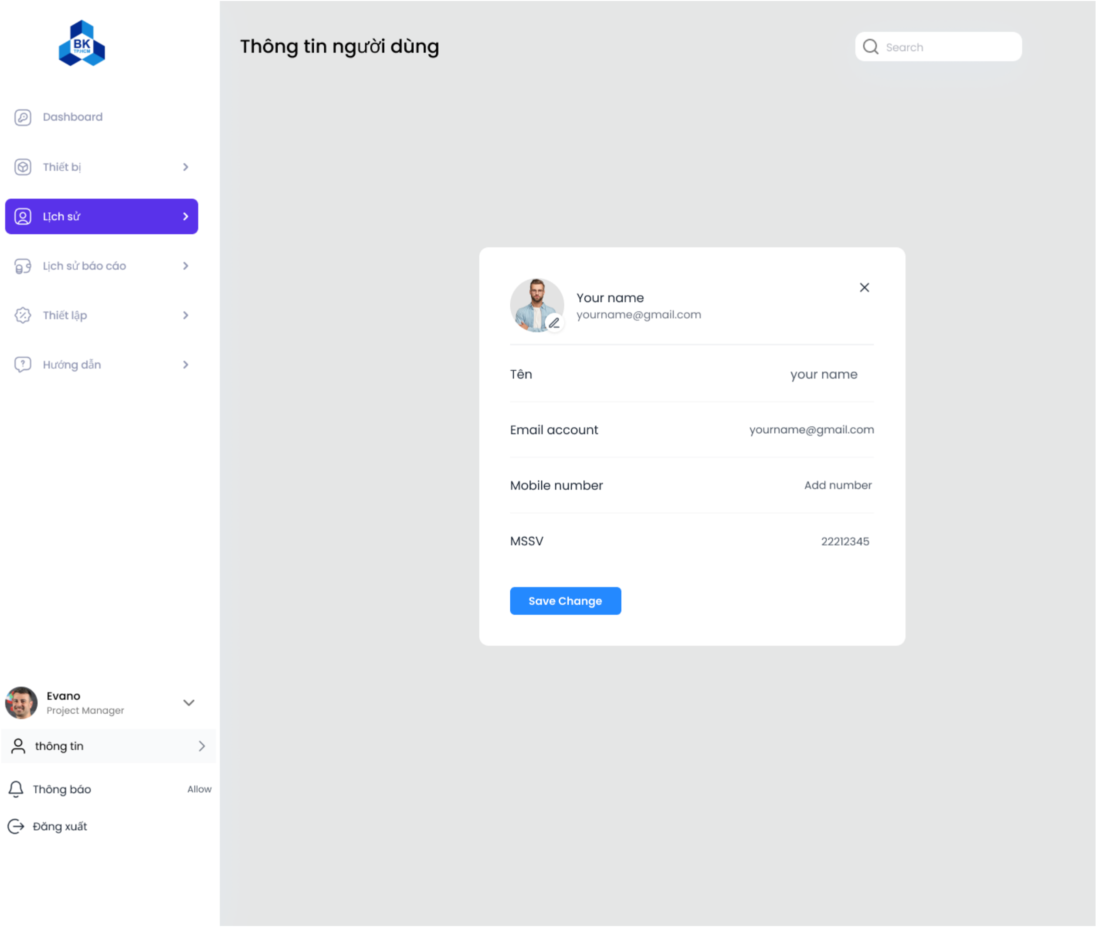
From Jadon S.
Yesterday, 14:36

Doanh thu tháng trước
682.5 VND

Doanh thu tháng này
682.5 VND

Evano
Project Manager

2. UI profile: With this interface, managers can view and edit their information



Thông tin người dùng

Your name
yourname@gmail.com

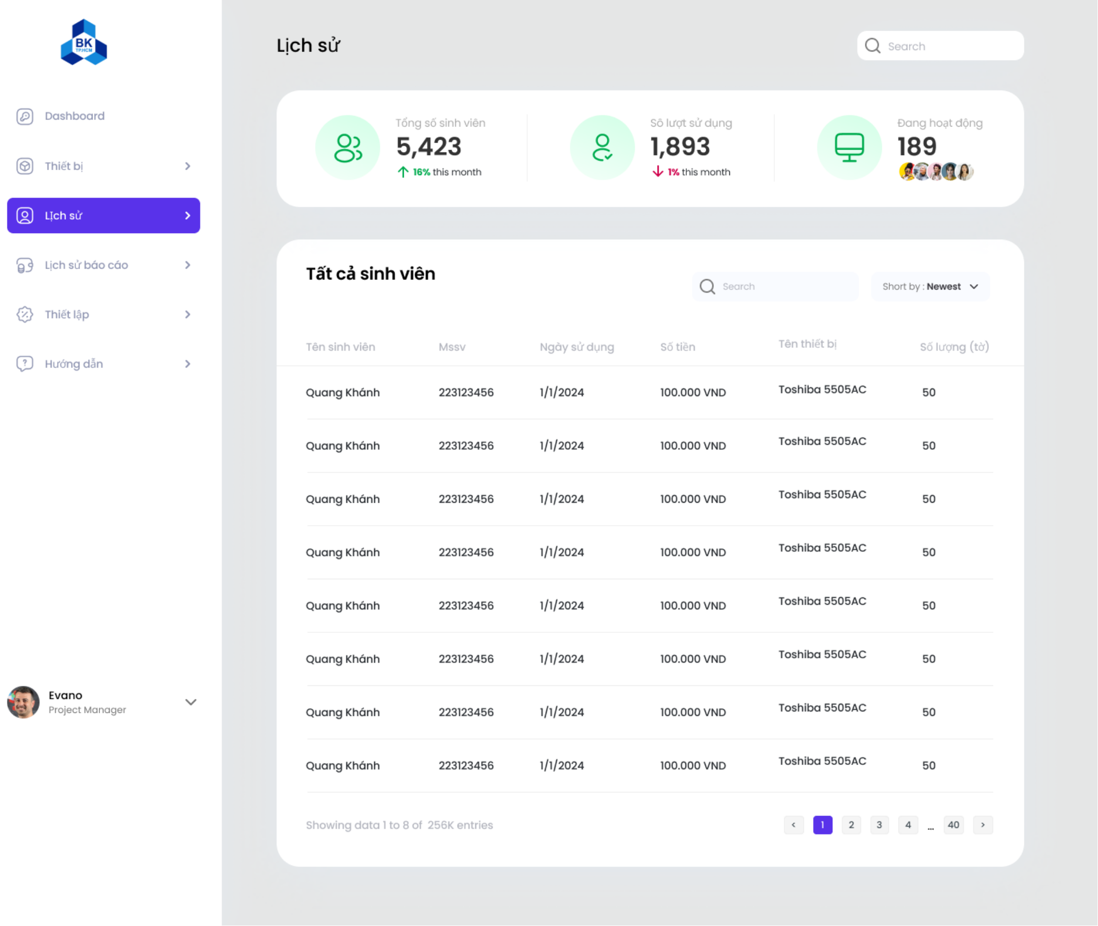
Tên: your name

Email account: yourname@gmail.com

Mobile number: Add number

MSSV: 22212345

3. UI History: With this interface, administrators can review activities where students have interacted with the system



Lịch sử

Tổng số sinh viên 5,423 ↑ 16% this month

Số lượt sử dụng 1,893 ↓ 1% this month

Đang hoạt động 189

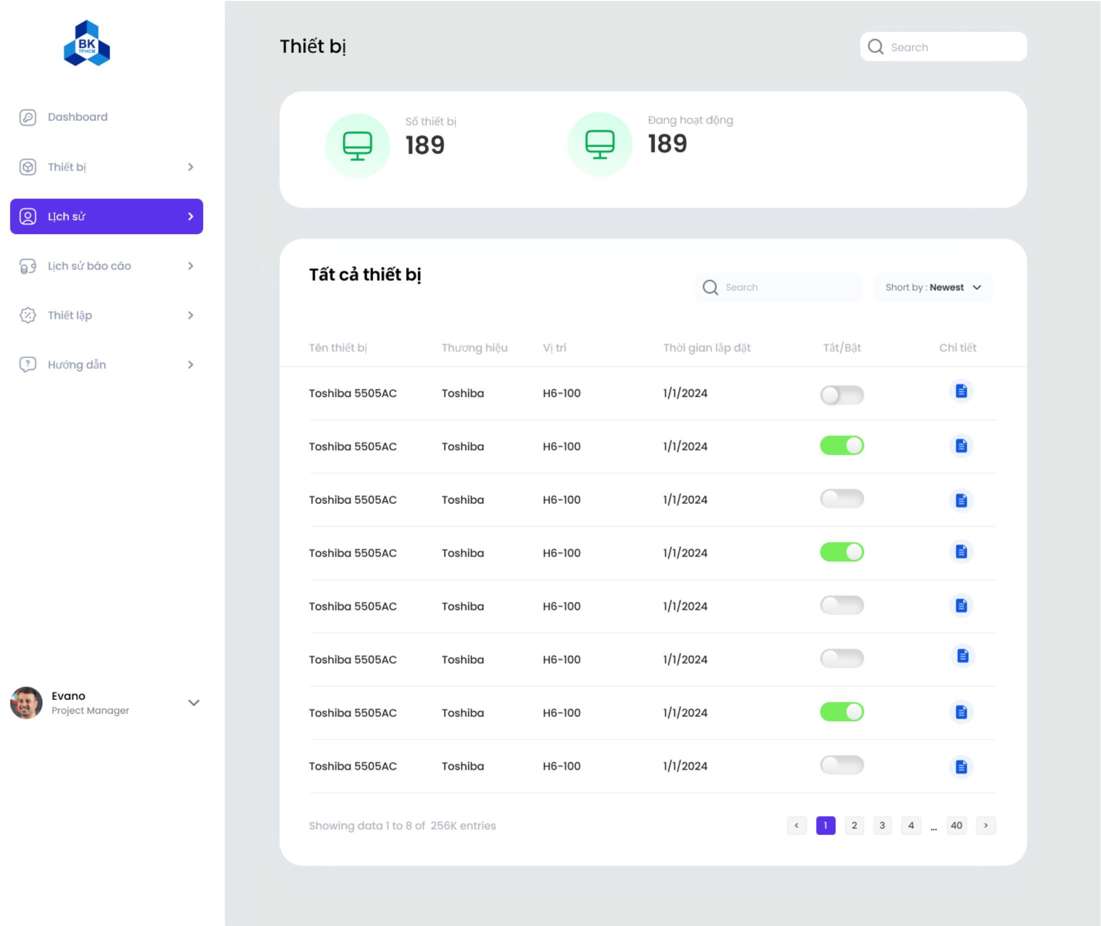
Tất cả sinh viên

Tên sinh viên	Mssv	Ngày sử dụng	Số tiền	Tên thiết bị	Số lượng (tờ)
Quang Khánh	223123456	1/1/2024	100.000 VND	Toshiba 5505AC	50
Quang Khánh	223123456	1/1/2024	100.000 VND	Toshiba 5505AC	50
Quang Khánh	223123456	1/1/2024	100.000 VND	Toshiba 5505AC	50
Quang Khánh	223123456	1/1/2024	100.000 VND	Toshiba 5505AC	50
Quang Khánh	223123456	1/1/2024	100.000 VND	Toshiba 5505AC	50
Quang Khánh	223123456	1/1/2024	100.000 VND	Toshiba 5505AC	50
Quang Khánh	223123456	1/1/2024	100.000 VND	Toshiba 5505AC	50
Quang Khánh	223123456	1/1/2024	100.000 VND	Toshiba 5505AC	50

Showing data 1 to 8 of 256K entries

< 1 2 3 4 ... 40 >

4. UI Device: With this interface, managers can view the number of devices as well as detailed information of the devices connected to the system. In addition, managers can also turn these devices on and off



Thiết bị

Số thiết bị
189

Đang hoạt động
189

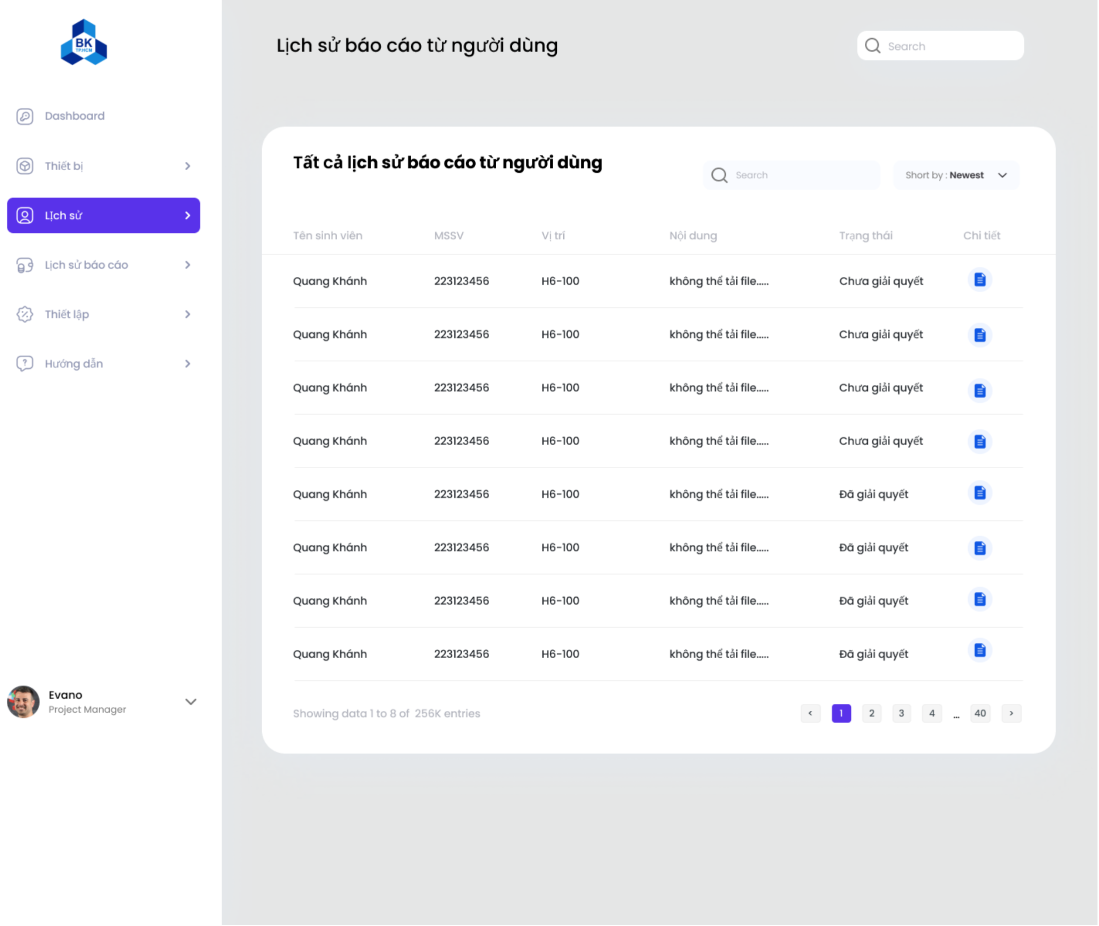
Tất cả thiết bị

Tên thiết bị	Thương hiệu	Vị trí	Thời gian lắp đặt	Tắt/Bật	Chi tiết
Toshiba 5505AC	Toshiba	H6-100	1/1/2024	○	📄
Toshiba 5505AC	Toshiba	H6-100	1/1/2024	●	📄
Toshiba 5505AC	Toshiba	H6-100	1/1/2024	○	📄
Toshiba 5505AC	Toshiba	H6-100	1/1/2024	●	📄
Toshiba 5505AC	Toshiba	H6-100	1/1/2024	○	📄
Toshiba 5505AC	Toshiba	H6-100	1/1/2024	○	📄
Toshiba 5505AC	Toshiba	H6-100	1/1/2024	●	📄
Toshiba 5505AC	Toshiba	H6-100	1/1/2024	○	📄

Showing data 1 to 8 of 256K entries

‹ 1 2 3 4 ... 40 ›

5. UI Report receiving history: With this interface, managers can view reports that the system has received from users and can then give reasonable assessments and feedback to users.



Lịch sử báo cáo từ người dùng

Tất cả lịch sử báo cáo từ người dùng

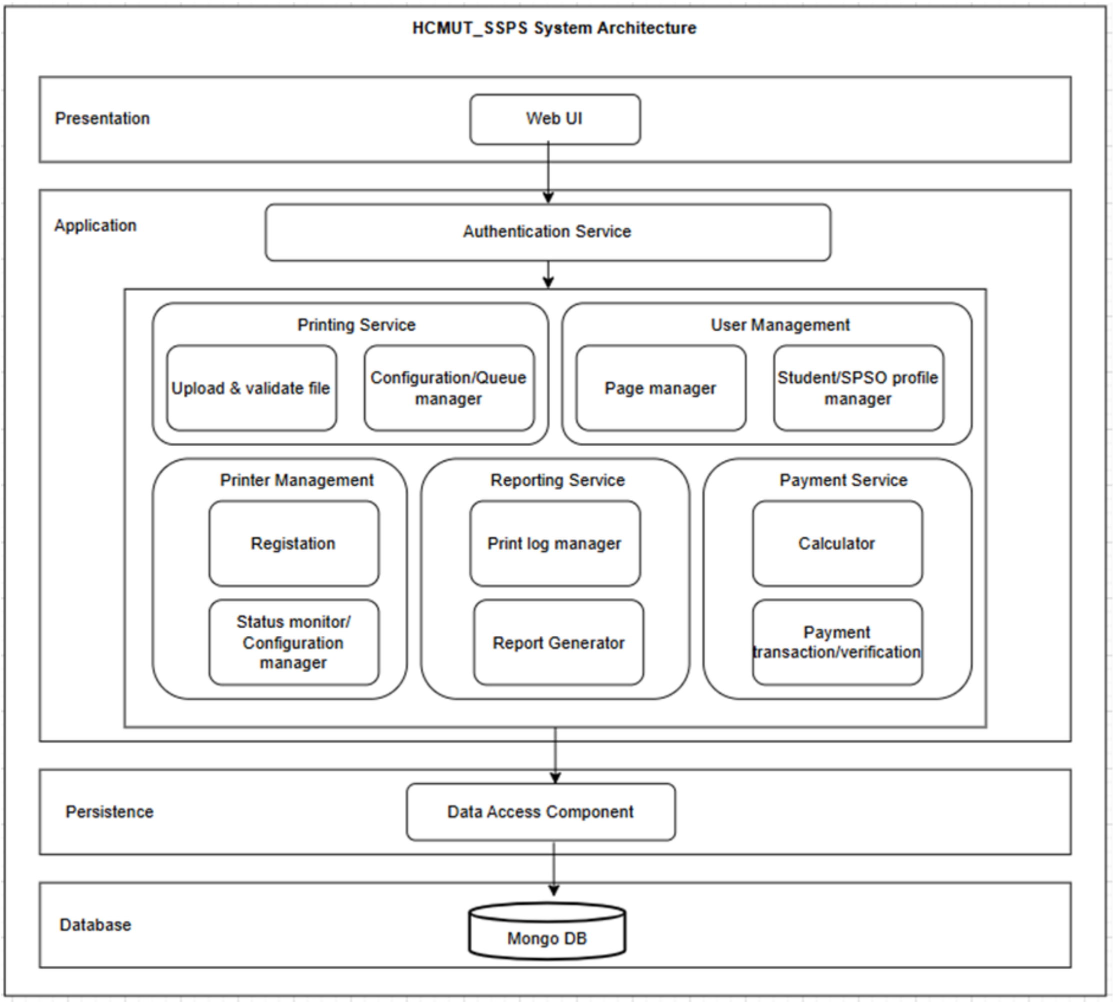
Tên sinh viên	MSSV	Vị trí	Nội dung	Trạng thái	Chi tiết
Quang Khánh	223123456	H6-100	không thể tải file.....	Chưa giải quyết	📄
Quang Khánh	223123456	H6-100	không thể tải file.....	Chưa giải quyết	📄
Quang Khánh	223123456	H6-100	không thể tải file.....	Chưa giải quyết	📄
Quang Khánh	223123456	H6-100	không thể tải file.....	Chưa giải quyết	📄
Quang Khánh	223123456	H6-100	không thể tải file.....	Đã giải quyết	📄
Quang Khánh	223123456	H6-100	không thể tải file.....	Đã giải quyết	📄
Quang Khánh	223123456	H6-100	không thể tải file.....	Đã giải quyết	📄
Quang Khánh	223123456	H6-100	không thể tải file.....	Đã giải quyết	📄

Showing data 1 to 8 of 256K entries

3 TASK 3

3.1 Architecture Design

3.1.1 Layer architecture for HCMUT SSPS



The system is designed in a layered architecture, with separate functions for each layer. The architecture was chosen consists of four layers: **presentation**, **business**, **persistence**, and **database**. In particular, each class performs a specific role in the application:

- **Presentation layer:** This layer is responsible for user interaction, displaying the interface, sending requests, and receiving responses. It consists of the Web UI, which allows users to interact with the HCMUT SSPS printing services.

- **Business layer:** This layer handles the core logic of the HCMUT SSPS system. Upon receiving a request from the presentation layer, it interacts with the persistence layer to retrieve and process data according to the system's rules. The results are then returned to the presentation layer. This layer includes various services: an Authentication Service for user login and access control; a User Management service for managing user profiles and permissions (including a Page manager for user-specific print settings and a Profile manager for student and administrator details); a Printing Service to handle file uploads, validation, print queues, and printer management (including printer registration, status monitoring, and configuration); a Reporting Service for generating print logs and usage reports; and a Payment Service for calculating print costs and processing transactions.

- **Persistence layer:** This layer acts as an intermediary between the application and database layers, providing a secure and controlled way to access data. It focuses solely on data retrieval without involving HCMUT SSPS specific logic. It consists of the Data Access Component, which handles database interactions.

- **Database layer:** This layer is responsible for storing all HCMUT SSPS system data, including printer information, user details, print logs, and configurations. It uses MongoDB, a NoSQL database, for flexible and scalable data storage.

Each layer in this architecture represents a distinct group of components designed to fulfill specific HCMUT SSPS requirements. This layered approach provides isolation and encapsulation, enhancing maintainability and security. Requests flow sequentially from the upper layers to the lower layers without bypassing any layer. This structure ensures that changes within one layer have minimal impact on other layers, promoting stability and limiting data access vulnerabilities.

Notable Advantages and Disadvantages:

The layered architecture offers several advantages, including simplicity, ease of modification, controlled information access, and adaptability for future upgrades. However, it can also lead to performance overhead for simple requests that don't require complex HCMUT SSPS processing. For instance, retrieving basic user information like name or ID might involve unnecessary traversal through all layers, consuming resources and increasing response time.

Despite this potential drawback, the layered architecture is well-suited for this HCMUT SSPS printing service system. The core functionalities, such as print job management, reporting, and payment processing, involve substantial data processing and HCMUT SSPS specific logic. Therefore, the benefits of modularity, maintainability, and security offered by the layered approach outweigh the potential performance overhead for simpler requests.

3.1.2 Describe activity of system

a) **Describe how you will present your User Interface**

The user will perform operations on the Web UI (Contains Student UI and Admin UI), then the presentation layer will send those requests to the business layer. In the business layer, user requests will have to be authenticated through the authentication service before being passed to functional modules in the same layer. For example, if a user wants to access the printing service, the authentication service will verify whether the user is student or admin to grant access to upload & validate file or configuration/queue manager. Next, functional modules will request the necessary data by accessing the Data Access module in the persistence layer. At this module, it will forward the data request to the data component in the Database layer. Finally, the data will be sent back from the Database layer to Persistence, then to Business for functional modules to process the logic and send the data to the presentation layer to display for the user.

b) Describe how will you store your data

In the database layer, data storage is designed for efficient retrieval and management, following the principles of a layered architecture. Since the database layer only communicates directly with the persistence layer, after going through the logical processing steps, the data layer will handle user requests through data access modules on the persistence layer. For data storage, basic user data that can be stored in the form of a table with appropriate keys and indexes to allow quick access and retrieval the information. In addition, print jobs, printer settings, print queue statuses, and detailed print logs are maintained in collections that facilitate efficient querying, especially for reporting and monitoring. After completing the requirements, it will automatically check, send a notification to the object that has changed or has just updated new information, and then will update the user's data with the information previously received.

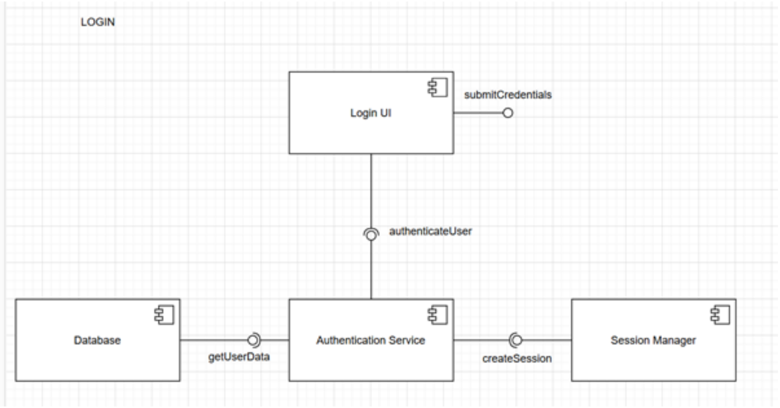
c) Describe how you will access external services/APIs

The system is designed to leverage APIs for communication and data exchange between the different layers of the HCMUT SSPS system. With the layered architecture above. We created APIs to be used in the following ways: The presentation layer communicates with the application layer using APIs. User inputs, such as printing requests or viewing print history, are sent to the application layer's endpoints, which handle these requests by processing data and returning responses to the user interface. The business logic layer uses APIs to request data from the data access layer or to store data in the system. For example, when a user requests print logs or printer status, the application layer retrieves this data from the persistence layer using its internal service APIs and applies the necessary business logic before passing it back to the presentation layer. APIs be used to facilitate communication between different components of the system.

3.2 Component Diagram for the Task Assignment Module

3.2.1 Login Module

Diagram:

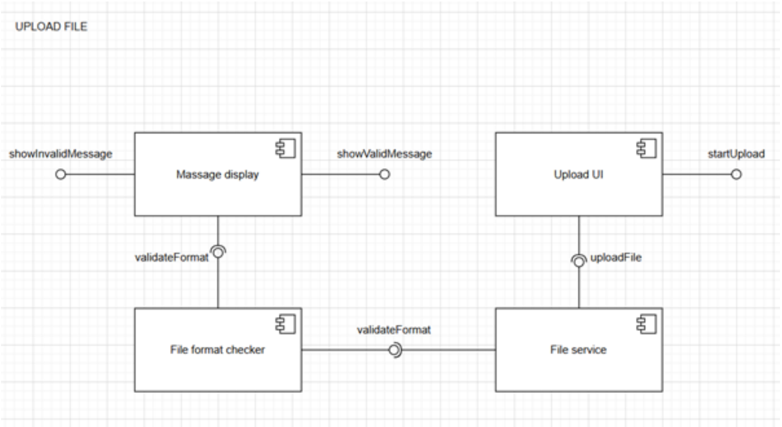


Description:

The Login Module is responsible for handling user authentication in the system. When a user enters their login credentials in the Login UI, the module sends these details to the Authentication Service, which checks the credentials against the records in the Database. If the credentials are valid, the Session Manager creates a new session, allowing the user to access their account and continue using the system. If the credentials are invalid, the Authentication Service will reject the login attempt, and the user will need to try again. This module ensures that only verified users are granted access to secure parts of the system.

3.2.2 Upload File Module

Diagram:

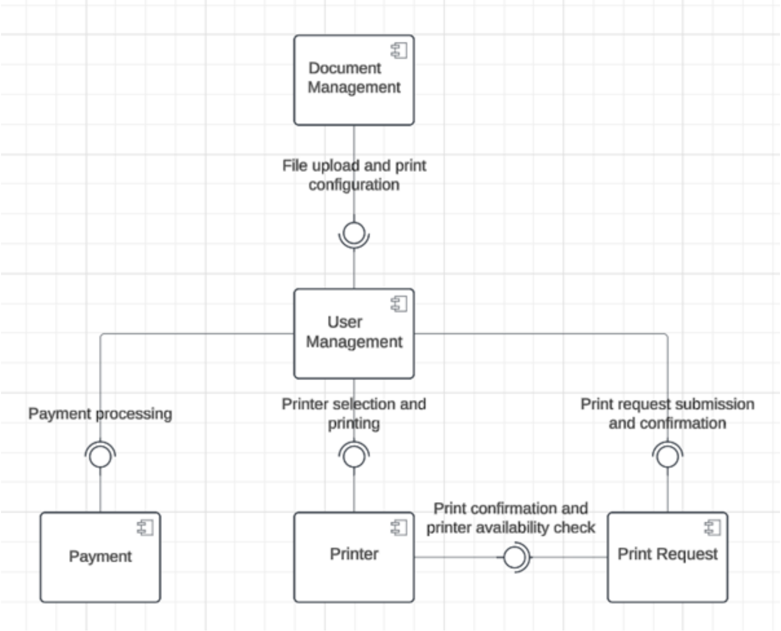


Description:

The File Upload Module handles the process of uploading and validating user files. When a user uploads a file, this module uses the File Format Checker to determine the file's validity. If the file is invalid, Message Display shows an error message and prompts the user to re-upload. Conversely, if the file is valid, the system displays a "File is valid" message, allowing the user to proceed with confirmation to complete the process. The File Upload Module ensures that only files meeting the system's requirements are processed further, enhancing data safety and consistency.

3.2.3 Select Option Module

Diagram:


Document Management
File upload and print configuration
User Management
Payment processing
Printer selection and printing
Print request submission and confirmation
Payment
Printer
Print confirmation and printer availability check
Print Request

Description:

The component diagram represents the architecture of a printing system that includes five primary components: User, Printer, Payment, Document, and PrintRequest. These components interact with each other through well-defined interfaces, ensuring smooth communication and functionality.

- The **User** Component interacts with the **Printer** Component to select a printer and initiate print jobs. It also connects with the **Payment** Component to handle any necessary transactions, such as purchasing additional pages when the user's page balance is insufficient. Furthermore, the User Component communicates with the **Document** Component to upload documents and select print options.
- The **Printer** Component is responsible for handling print requests from the User Component. It checks its availability and processes print jobs once confirmed by the **PrintRequest** Component.
- The **PrintRequest** Component plays a central role by facilitating the print request process. It communicates with both the User and Printer Components to submit print jobs, confirm

the readiness of the printer, and manage user interactions.

- The **Payment** Component ensures that users can pay for additional pages if required. It is connected to the User Component, allowing users to process payments seamlessly.

Each component is linked via assembly connectors, which represent required and provided interfaces. These connectors illustrate the flow of information between the components, such as how the User Component requests printing from the Printer Component, or how the PrintRequest Component ensures that the user's print job is confirmed and processed. The diagram does not require explicit ports unless there is a need for a more detailed distinction between different types of interfaces, as the assembly connectors adequately convey the necessary interactions.