

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



SOFTWARE ENGINEERING (CO3001)

SSPS SOFTWARE REPORT

Task 2

GROUP CC03-05

Students Nguyễn Hồ Đức An - 2252009
 Nguyễn Đức Quốc Kỳ - 2252415
 Nguyễn Châu Hoàng Long - 2252444
 Nguyễn Gia Phúc - 2252637
 Nguyễn Anh Quân - 2252678

Lecturer: Dr. Trương Tuấn Anh

Contents

| | | |
|----------|--|-----------|
| 1 | Task descriptions | 2 |
| 2 | Activity Diagrams | 3 |
| 2.1 | Add/Remove printer | 3 |
| 2.2 | Enable/Disable printer | 3 |
| 2.3 | Update Configuration | 5 |
| 2.4 | View printers logs | 6 |
| 3 | Sequence Diagrams | 7 |
| 3.1 | Display use case | 7 |
| 3.2 | Add printer use case | 7 |
| 3.3 | Delete/Enable/Disable printer use case | 8 |
| 3.4 | Update printer use case | 10 |
| 3.5 | View logs use case | 11 |
| 4 | Class Diagram | 12 |
| 5 | User Interface | 13 |
| 5.1 | Homepage | 13 |
| 5.2 | Login Page | 13 |
| 5.3 | Logs Page | 14 |
| 5.4 | Printer Information Page | 15 |
| 5.5 | Add Printer Add Printer | 16 |
| 5.6 | Configure Printer Page | 16 |



1 Task descriptions

- 2.1:** Draw an activity diagram to capture the business process between systems and the stakeholders in a particular module (Printer Management Module has been chosen)
- 2.2:** Draw a sequence diagram for a particular module (the same with the module used in task 2.1)
- 2.3:** Draw a class diagram of a particular module (the same with the module used in task 2.1) as comprehensive as possible
- 2.4:** Develop MVP 1 as user interfaces of either a Desktop-view central dashboard for a particular module (the same with the module used in task 2.1). (Figma has been used for this task)

2 Activity Diagrams

2.1 Add/Remove printer

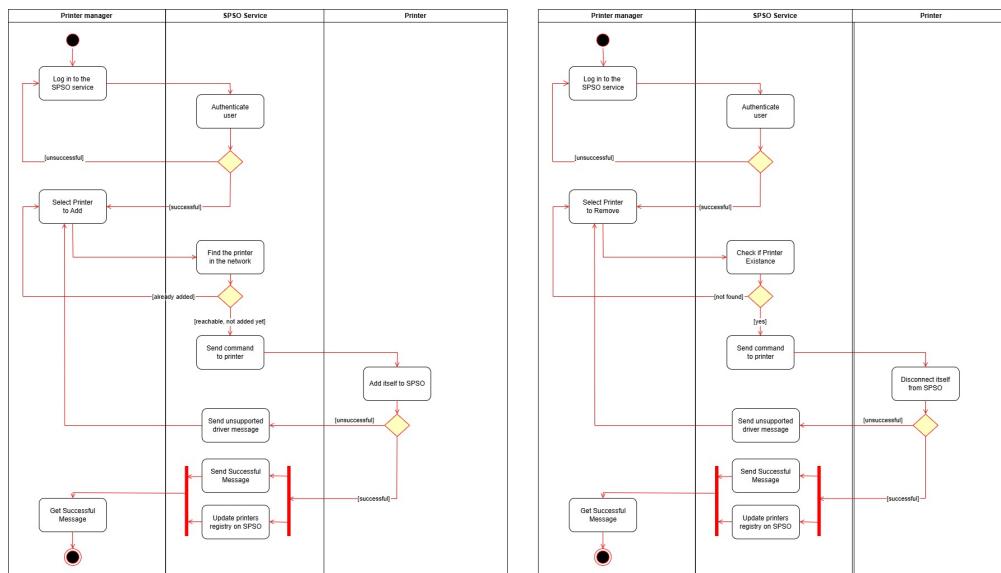


Figure 1: Add/Remove printer use case

The **Add/Remove Printer** use case involves a process where the user (SPSO) selects an option to either add or remove a printer from the system. When adding a printer, the system prompts the user for necessary details such as printer name and network information, while for removing, it shows a list of available printers for selection. The system then processes the request and updates the printer registry to reflect these changes, confirming the successful completion of the operation. If the user tries to add a printer that already exists or remove one that isn't found, the system responds with an error message. Additionally, an exception occurs if a printer driver is unavailable, causing the printer addition to fail.

2.2 Enable/Disable printer

The **Enable/Disable Printer** use case describes the process of modifying a printer's operational status. The user (SPSO) selects the option to enable or disable the printer, prompting the system to display its current status. After the user makes the desired change and submits it, the system updates the printer's status in the registry and confirms the successful modification.

In an alternative scenario, if the user attempts to enable a printer that is already enabled or disable one that is already disabled, the system will display a message indicating that the printer is already in the selected status.

In case of an exception, such as when the printer is not connected, the system will be unable to complete the action and will display an error message.

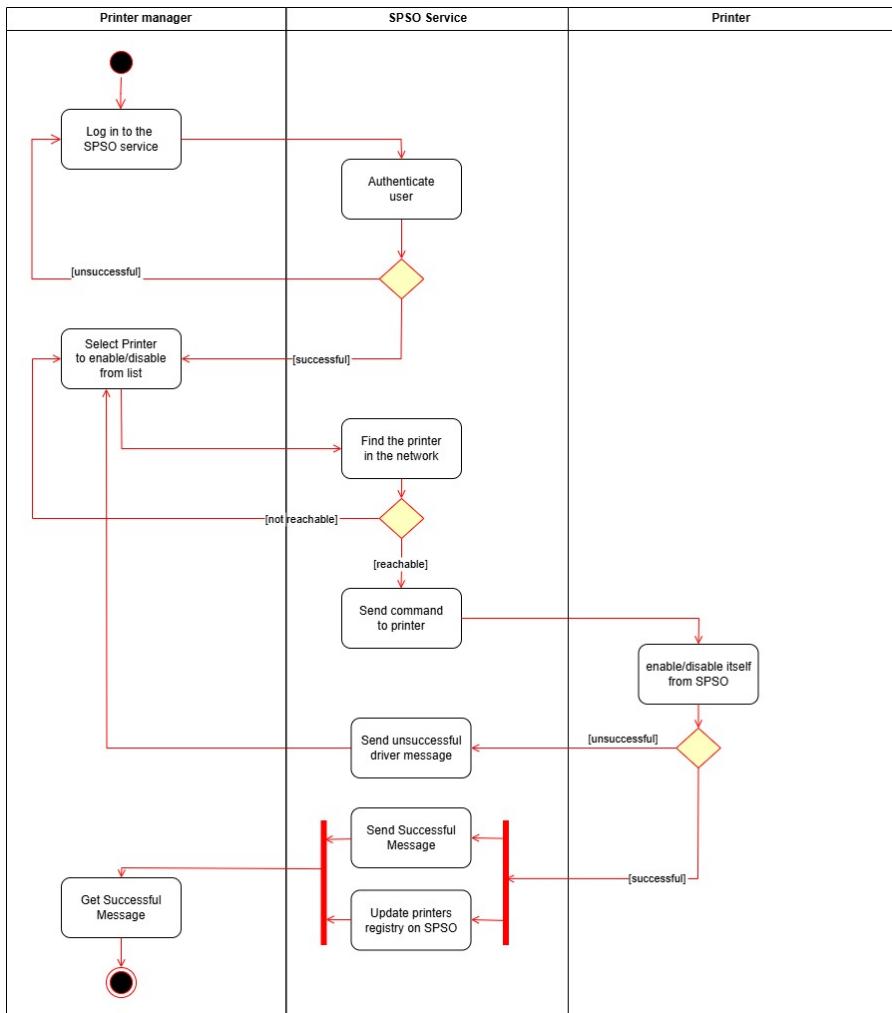


Figure 2: Enable/Disable printer use case

2.3 Update Configuration

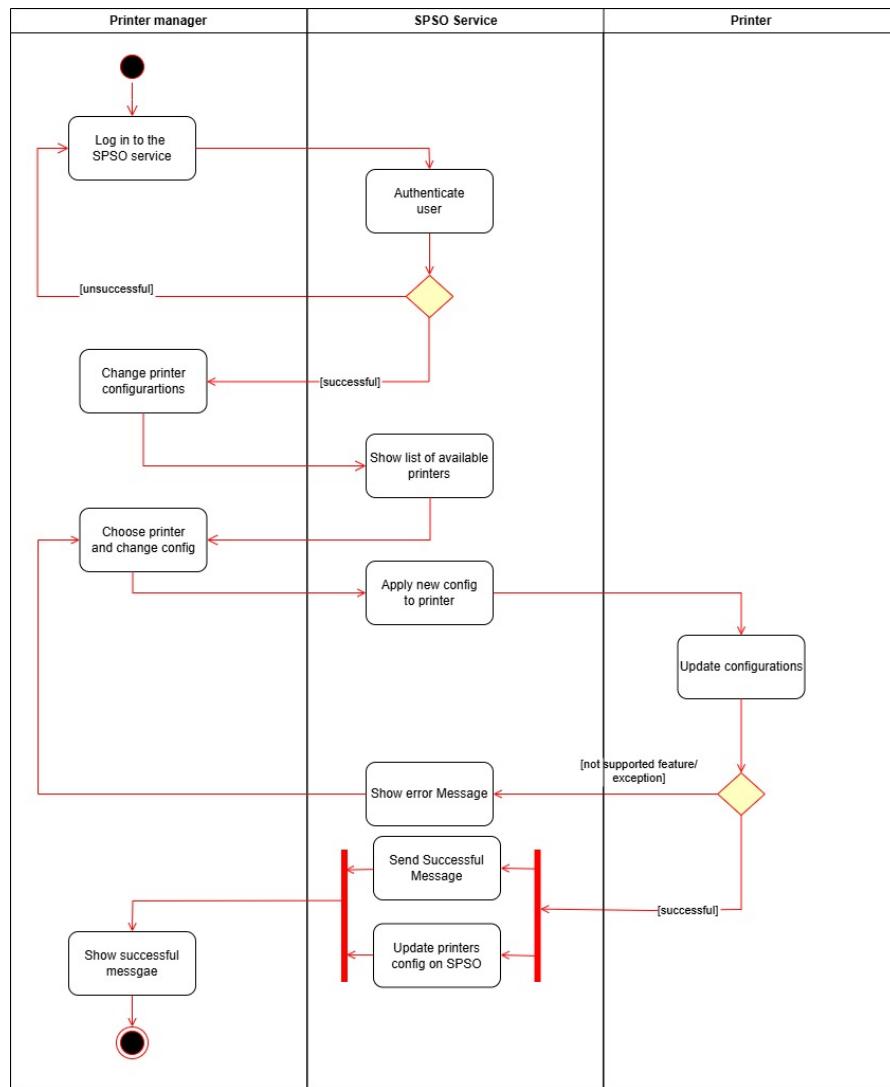


Figure 3: Change printer configuration use case

The **Change Printer Configuration** use case describes the process of modifying a printer's configuration settings. The user (SPSO) selects the option to change the printer's configuration, prompting the system to display available configuration options. After the user makes the necessary changes and submits them, the system updates the printer's configuration and confirms the successful modification.

In an alternative scenario, if the user attempts to modify settings that are not supported by the selected printer (e.g., unavailable features), the system displays an error message and prompts the user to select valid configuration options.

In case of an exception, such as a system error occurring during the configuration change, the system will display an error message and generate an error log for further review.

2.4 View printers logs

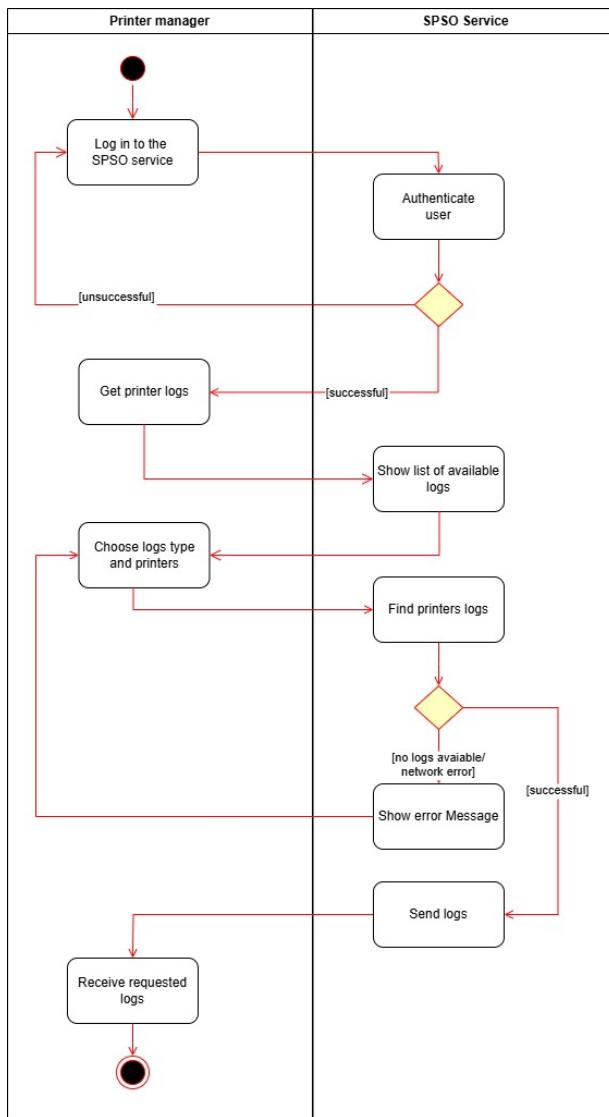


Figure 4: View printer logs use case

The **View Printer Logs** use case describes the process by which the user (SPSO) can view logs related to printer activity. Upon selecting the option to view logs, the system presents a list of available logs, such as error or maintenance logs. The user selects the desired logs, and the system retrieves and displays them accordingly. If no logs are available for the selected printer, the system notifies the user with a message. In case of a system error (e.g., database or connectivity issues), the system is unable to retrieve the logs and displays an error message to the user.

3 Sequence Diagrams

3.1 Display use case

When **PrinterService.View** in the SPSO system manages printers by selecting the "Printers" link on the header, it sends the request `get_all_printers()` to **PrinterService.View**. **PrinterService.View** executes this function and proceeds to call `get_printers_API()` in **PrinterService.Controller**. Similarly, **PrinterService.Controller** executes and then calls the `find_all_printers()` function in **PrinterService.Model**.

From here, **PrinterService.Model** executes the `find_all_printers()` function and returns a list of all printers, `printers_list`. After that, `printers_list` is returned asynchronously from **PrinterService.Model** through **PrinterService.Controller** and then **PrinterService.View**. Finally, **PrinterService.View** calls the `Display_Printer_management_interface()` function to display the "Printer Management Interface" view.

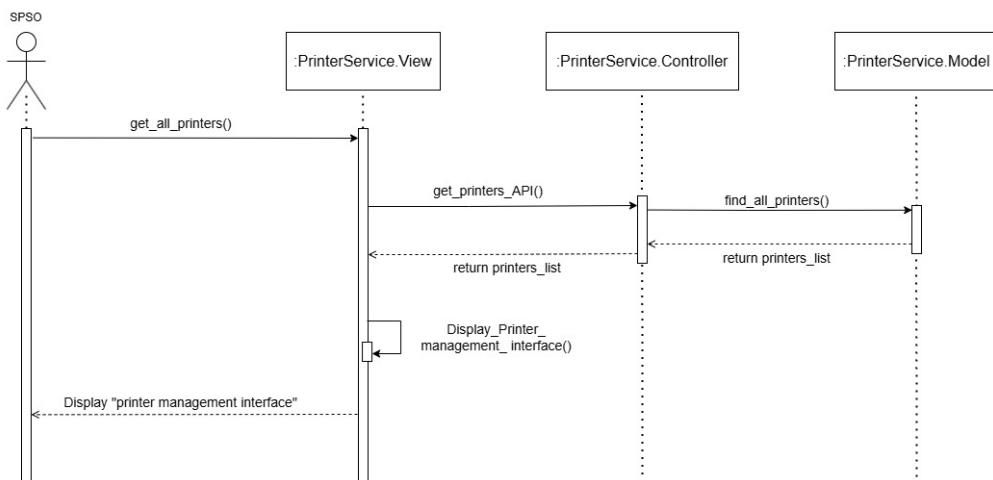


Figure 5: Display use case sequence diagram

3.2 Add printer use case

When SPSO aims to add more printers, it clicks on the "Add Printer" button to send the request `Addnewprinter()` to **PrinterService.View**, which executes the `Addnewprinter()` function. After that, **PrinterService.View** sends the request `Create` to **Printer Configuration**, where SPSO fills in the detailed information about the new printer in the additional form. During data entry, if SPSO clicks the "Cancel" button, **PrinterService.View** will call `CancelAddPrinter()` in **Printer Configuration** to cancel the process, returning the `cancel_status` to **PrinterService.View**, which then recalls the `Display_Printer_management_interface()` function to return to the Printer Management Interface.

Otherwise, SPSO will double-check the new printer information for accuracy before submitting the form. Once submitted, **Printer Configuration** calls the `AddPrinterAPI()` function with one parameter, `printer_data`, which contains the submitted data from the form, in **PrinterService.Controller**. After that, `Add_New_Printer(printer_data)` function is called by **PrinterService.Controller** in **PrinterService.Model**, which executes it and then returns asynchronously with `add_status` via **PrinterService.Controller** and then **PrinterService.View**.

Finally, **PrinterService.View** recalls the `Display_Printer_management_interface()` function to return to the Printer Management Interface.

Finally, **PrinterService.View** recalls the `Display_Printer_management_interface()` function to return to the printer management interface.

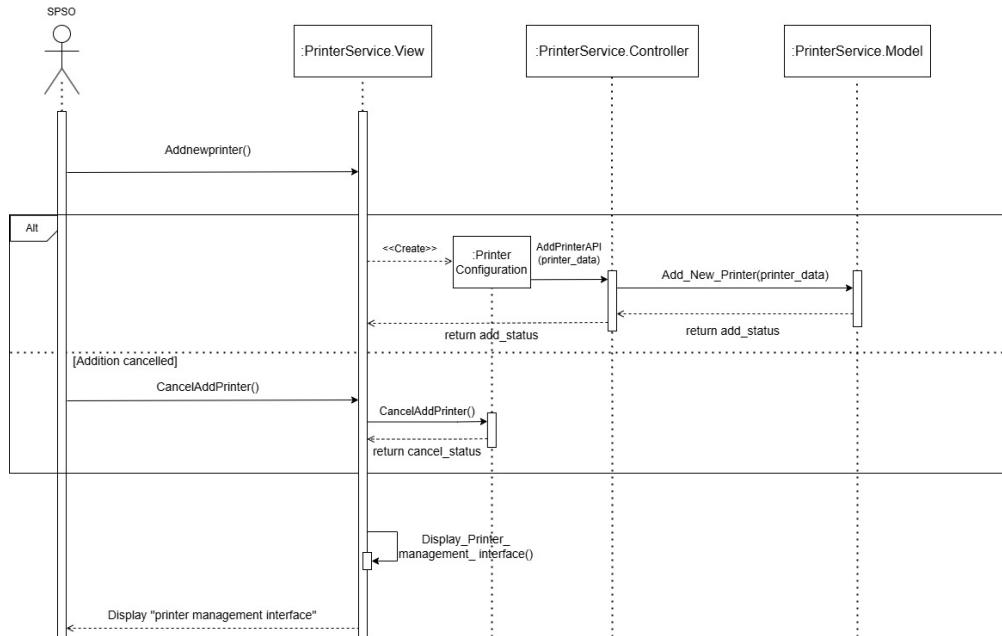


Figure 6: Add printer use case sequence diagram

3.3 Delete/Enable/Disable printer use case

When SPSO aims to remove a specified printer, it clicks on the "Recycle Bin" icon in the corresponding printer's "Action" field to send the request `DeletePrinter(id)` to **PrinterService.View**, where the `id` parameter represents the index of the specified printer. Afterward, a popup appears on the screen to confirm the deletion. If SPSO cancels the process, **PrinterService.View** calls the `CancelDeletePrinter()` function to stop the process and then recalls the `Display_Printer_management_interface()` function to return to the Printer Management Interface. Otherwise, **PrinterService.View** calls the `DeletePrinterAPI(id)` function in **PrinterService.Controller**, which then calls `Delete_Printer(id)` in **PrinterService.Model**. **PrinterService.Model** executes the deletion and returns the `delete_status` asynchronously via **PrinterService.Controller** and **PrinterService.View**. Finally, **PrinterService.View** recalls the `Display_Printer_management_interface()` function to return to the Printer Management Interface.

When SPSO aims to enable or disable a specified printer, it clicks the enable/disable button to send the request to **PrinterService.View**. If the printer is disabled, clicking the button makes it available, and vice versa. **PrinterService.View** calls the `ToggleStatusAPI()` function with the parameter `id`, which is the index of the specified printer, in **PrinterService.Controller**. Then, **PrinterService.Controller** calls the `Toggle_Status_State(id)` function in **PrinterService.Model**, which executes the function and returns the `toggle_status` asynchronously via **PrinterService.Controller** and **PrinterService.View**. Finally, **PrinterService.View**

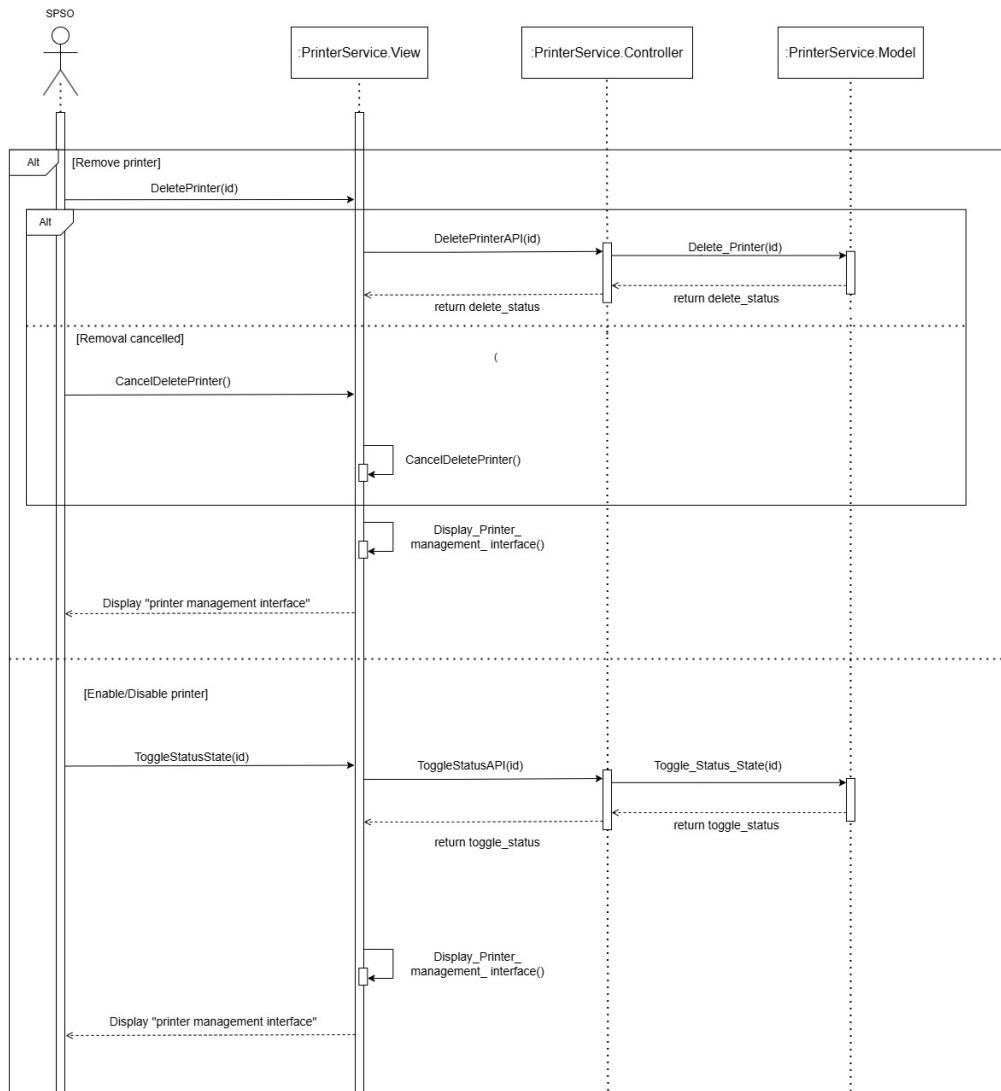


Figure 7: Delete/Enable/Disable printer use case sequence diagram

recalls the `Display_Printer_management_interface()` function to return to the Printer Management Interface.

3.4 Update printer use case

When SPSO aims to update a specified printer, it clicks on the "Edit" icon in the corresponding printer's "Action" field to send the request `Updateprinter(id)` to **PrinterService.View**, where the `id` parameter represents the index of the chosen printer. At this point, SPSO has two options: to update the printer information or to update the printer drivers for maintenance.

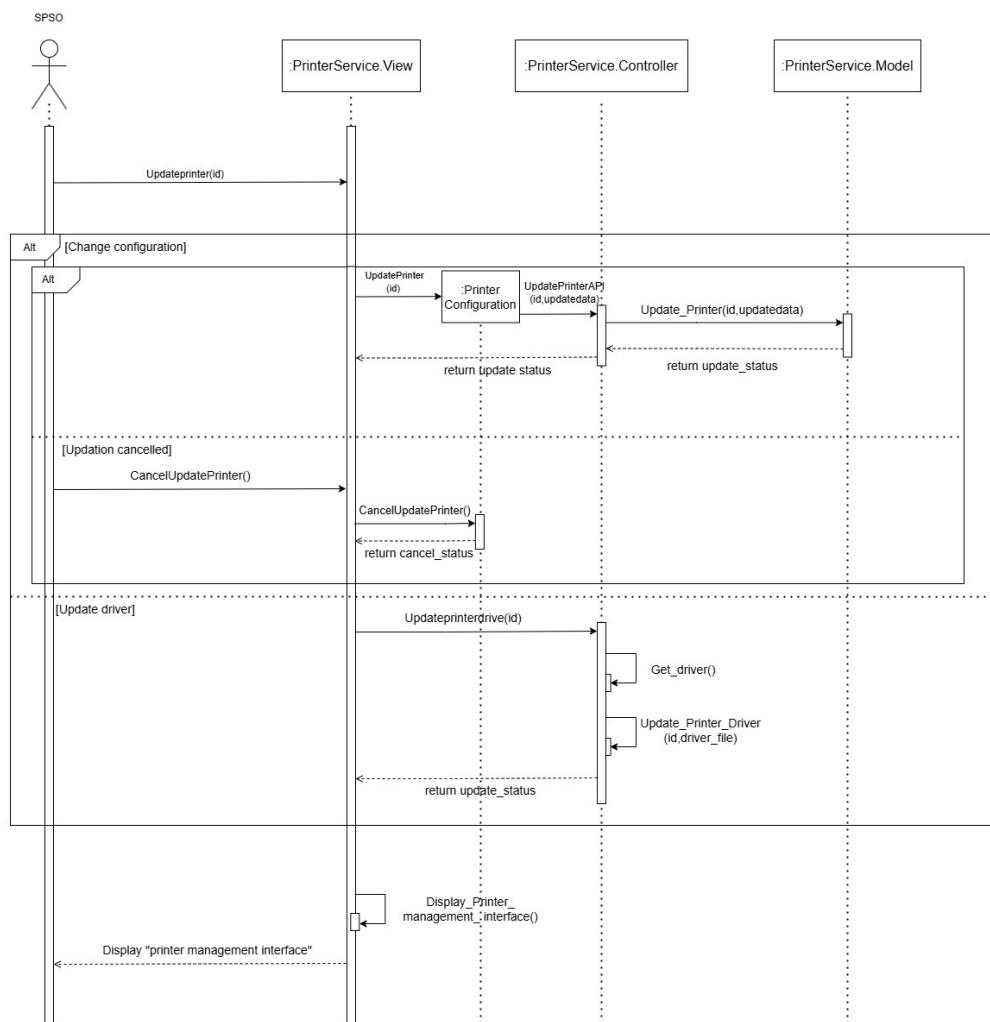


Figure 8: Update printer use case sequence diagram

In the first case, when SPSO chooses to update printer information, **PrinterService.View** calls the `UpdatePrinter(id)` function in **Printer Configuration**, where SPSO updates the latest information for the printer. During data updating, if SPSO clicks the "Cancel" button, **PrinterService.View** calls the `CancelUpdatePrinter()` function in **Printer Configuration**,

which returns the `cancel_status` to `9extbfPrinterService.View`. This action prompts **PrinterService.View** to recall the `Display_Printer_management_interface()` function, returning to the Printer Management Interface.

Otherwise, SPSO will double-check the updated information before submitting the form. Upon submission, **Printer Configuration** calls the `UpdatePrinterAPI` function with two parameters: `updatedata` (the submitted data from the form) in **PrinterService.Controller**. Similarly, **PrinterService.Controller** calls the `Update_Printer(updatedata, id)` function in **PrinterService.Model**, which executes this function and then asynchronously returns the `update_status` via **PrinterService.Controller** and **PrinterService.View**. Finally, **PrinterService.View** recalls the `Display_Printer_management_interface()` function to return to the Printer Management Interface.

In the other case, when SPSO performs printer maintenance, **PrinterService.View** calls `Updateprinterdrive()` with one parameter (the index of the chosen printer) in **PrinterService.Controller**. Here, **PrinterService.Controller** sequentially calls the `Get_driver()` function to retrieve the latest drivers from the internet, with the result, to apply the update, `Update_Printer_Driver(id, driver_file)` function is called. After this, **PrinterService.Controller** returns the `update_status` to **PrinterService.View**. Similar to the previous case, **PrinterService.View** recalls the `Display_Printer_management_interface()` function to return to the Printer Management Interface.

3.5 View logs use case

When SPSO aims to view the operation history, it clicks on the "Log" dropdown and selects "Printers." After that, **PrinterService.View** calls the `ViewPrinterLogsAPI()` function in **PrinterService.Controller**, which then calls the `ViewPrinterOperationHistory()` function in **PrinterService.Model**. **PrinterService.Model** processes this request and returns the list of all logs, `history_list`. The `history_list` is then returned sequentially from **PrinterService.Model** through **PrinterService.Controller** and finally to **PrinterService.View**. Here, **PrinterService.View** calls the `Show_history()` function to display all the operation history on the screen.

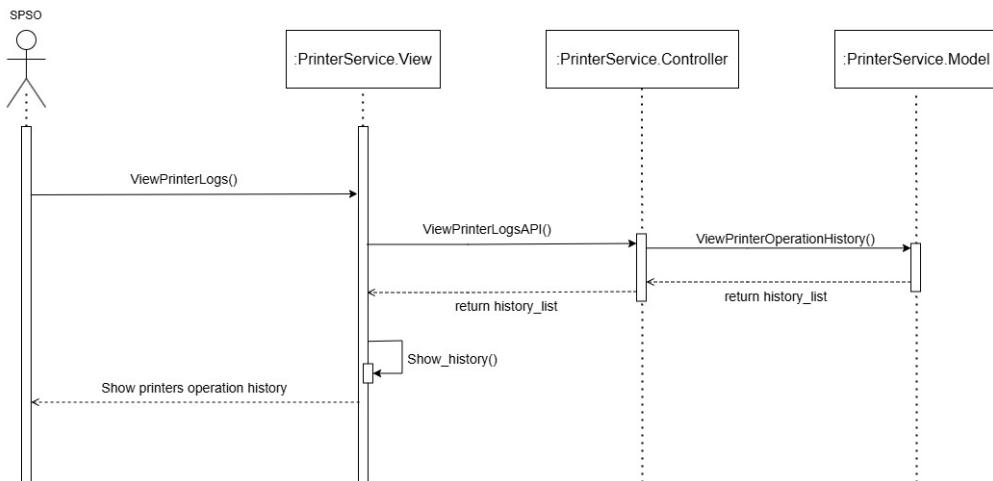


Figure 9: View logs use case sequence diagram

4 Class Diagram

Below is the class diagram of the system following MVC design pattern.

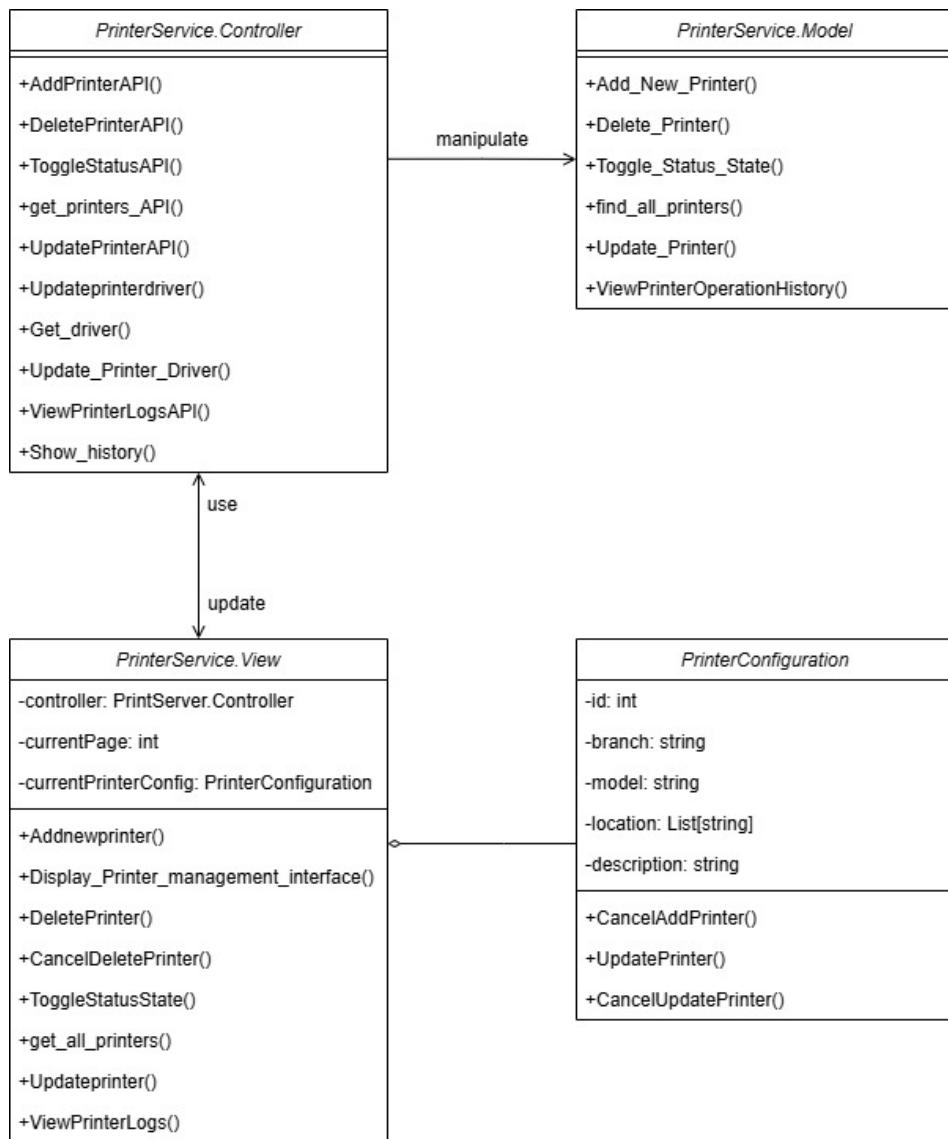


Figure 10: Class diagram



5 User Interface

5.1 Homepage

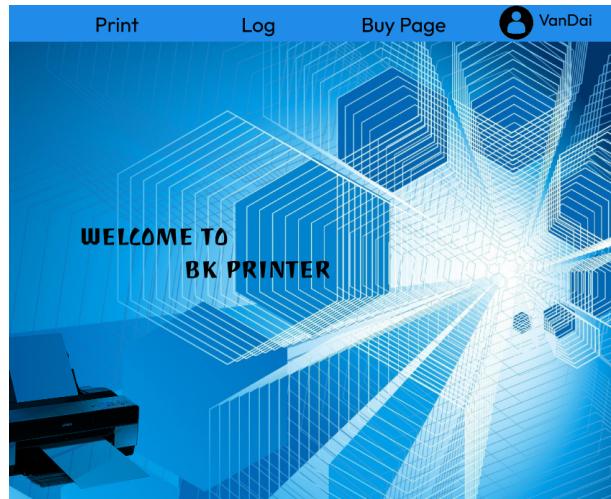


Figure 11: HomePage

The **Homepage** of the HCMUT-SSPS website serves as the main entry point for users. At the top, a header provides quick navigation with options like "Print Log" to view past print jobs, "Buy Page" for purchasing additional print credits, and a username display that, when clicked, reveals account options like "Profile" and "Logout."

5.2 Login Page



Figure 12: Login Page

The **Login** page features a simple and user-friendly layout, providing two main fields: one for the BKNetID and another for the password. These fields allow students to securely enter their



university credentials to access the printing service. A "Login" button below the forms completes the process, enabling students to authenticate and proceed to their account and printing options on the platform.

5.3 Logs Page

The **Logs** page displays a structured list of past print jobs, each entry showing key details to help Managers track their students printing history. Each record includes the student's name, the printer name used, the date of the print job, the file name of the document printed, and the number of pages. This organized view makes it easy for Manager to review their students past prints, verify completed jobs, and manage their printing activity efficiently.

Moreover, there are other options for viewing log of other events such as the purchasing log or the operation done on different printers

| Name | Printer | Start Date | Complete Date | File Name | Page |
|------|---------|--------------------|--------------------|-----------|------|
| Kỳ | CA1 | 22/10/2024 7:00 AM | 27/10/2023 7:00 AM | cnpm.doc | |
| Quân | EP1 | 27/10/2023 7:00 AM | 27/10/2023 7:00 AM | cnpm.pdf | |
| Phúc | CA2 | 27/10/2023 7:00 AM | 27/10/2023 7:00 AM | btl.docx | |
| Long | HP1 | 27/10/2023 7:00 AM | 27/10/2023 7:00 AM | btl.docx | |
| An | CA2 | 27/10/2023 7:00 AM | 27/10/2023 7:00 AM | btl.docx | |

Figure 13: Print Logs Page

| ID | Date | Operation | Description |
|-----|--------------------|-----------|-------------|
| CA1 | 01/11/2024 7:00 AM | Remove | |
| EP1 | 29/10/2024 7:00 AM | Configure | |
| CA1 | 28/10/2024 7:00 AM | Enable | |
| EP1 | 27/10/2024 8:00 AM | Add | |
| CA1 | 27/10/2024 7:00 AM | Add | |

Figure 14: Printers Logs Page



| Name | Page | Date | Payment Method | Total Cost |
|------|------|--------------------|----------------|------------|
| Kỳ | 100 | 22/10/2024 7:00 AM | Momo | 10.000 |
| Quân | 100 | 27/10/2023 7:00 AM | BKPay | 10.000 |
| Phúc | 100 | 27/10/2023 7:00 AM | BKPay | 10.000 |
| Long | 100 | 27/10/2023 7:00 AM | CreditCard | 10.000 |
| An | 200 | 27/10/2023 7:00 AM | Momo | 20.000 |

Figure 15: Purchase Logs Page

5.4 Printer Information Page

The **Printer information** page provides a comprehensive list of all available printers, each entry displaying detailed information such as the printer ID, branch, model, description, campus, building, and room location. At the top of the list, there's a search button that allows users to quickly locate specific printers based on their attributes. Additionally, there are action buttons for adding a new printer, as well as options to edit, delete, enable, or disable existing printers. This setup offers administrators full control and easy management over the printer fleet across campuses.

| ID | Branch | Model | Description | Campus | Building | Room | Action | Enable/Disable |
|------|--------|---------------|-------------|--------|----------|------|--------|----------------|
| CA1 | Canon | Pixma IX6770 | | 1 | A4 | 402 | | |
| EPI1 | Epson | L3250 | | 2 | H6 | 203 | | |
| CA2 | Canon | Pixma IX6770 | | 1 | B4 | 505 | | |
| HP1 | HP | M454NW W1Y43A | | 2 | H6 | 306 | | |

Figure 16: Printer Information Page



5.5 Add Printer Add Printer

The **Add Printer** page provides a form where administrators can input essential details to register a new printer. The form includes fields for Code Number (ID), Branch, Model, Campus, Building, Room, Location, and a Description of the printer. Once all fields are completed, the administrator can finalize the addition by clicking the Confirm button, which saves the new printer's details into the system for future use. This form ensures that each printer is accurately identified and easily locatable within the system.

The screenshot shows a web-based application interface for adding a new printer. At the top, there is a navigation bar with icons for home, log out, and user profile, followed by the text "Printers" and "Log". On the right side of the header is a dropdown menu labeled "Admin". Below the header, a large blue button on the right says "Add Printer". The main content area contains a form with the following fields:

- Code number (ID): An input field.
- Branch: An input field.
- Model: An input field.
- Location:
 - Campus: A dropdown menu set to "1".
 - Building: A dropdown menu set to "A4".
 - Room: A dropdown menu set to "110".
- Description: A text input field.

At the bottom of the form are two buttons: "Confirm" (blue) and "Cancel" (red).

Figure 17: Add Printer Page

5.6 Configure Printer Page

The **Configure Printer** page resembles the add printer page, providing a form with fields for Code Number (ID), Branch, Model, Campus, Building, Room, Location, and Description. However, this page allows users to edit and update the information of an existing printer. Once modifications are made, clicking the Confirm button saves the updated details, ensuring that the printer's information remains accurate and up-to-date in the system. This feature is essential for managing changes to printer locations or descriptions over time.

The screenshot shows a web-based application interface for configuring an existing printer. The layout is identical to the Add Printer page, featuring a header with "Printers", "Log", and "Admin" options, and a central "Configure Printer" button. The form fields are:

- Code number (ID): An input field.
- Branch: An input field.
- Model: An input field.
- Location:
 - Campus: A dropdown menu set to "1".
 - Building: A dropdown menu set to "A4".
 - Room: A dropdown menu set to "110".
- Description: A text input field.

At the bottom are "Confirm" and "Cancel" buttons.

Figure 18: Configure Printer Page