VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY

UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Software Engineering

A smart printing service for students at HCMUT

SYSTEM MODELING

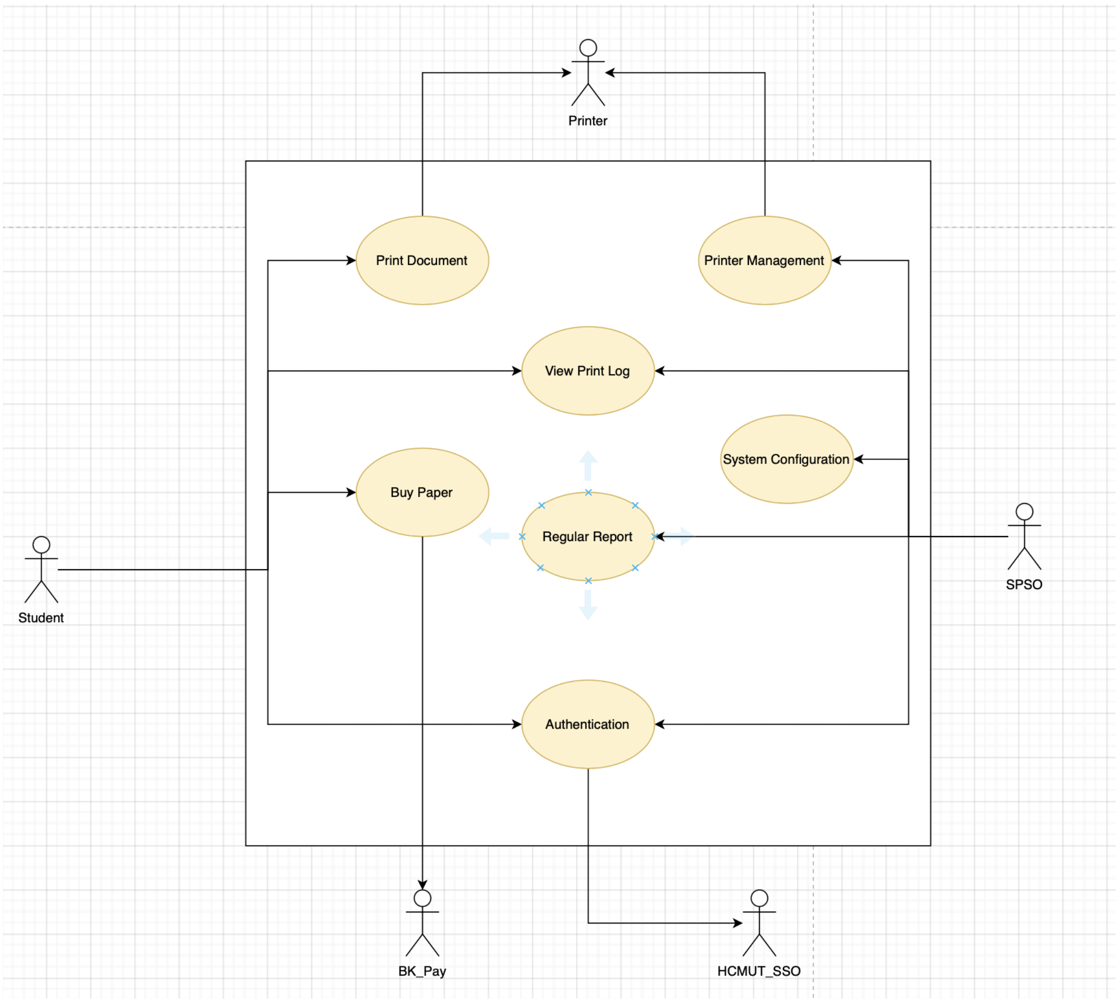
Advisor:	Trương Tuấn Anh
Students:	Phạm Đình Bảo Duy - 2252121
	Nguyễn Việt Hùng - 2252273
	Nguyễn Thành Minh – 2252481
	Nguyễn Sỹ Hùng – 2252271
	Nguyễn Tấn Huy – 2252258
	Nguyễn Thuỷ Tiên – 2252806

HO CHI MINH CITY, OCTOBER 2024

Contents

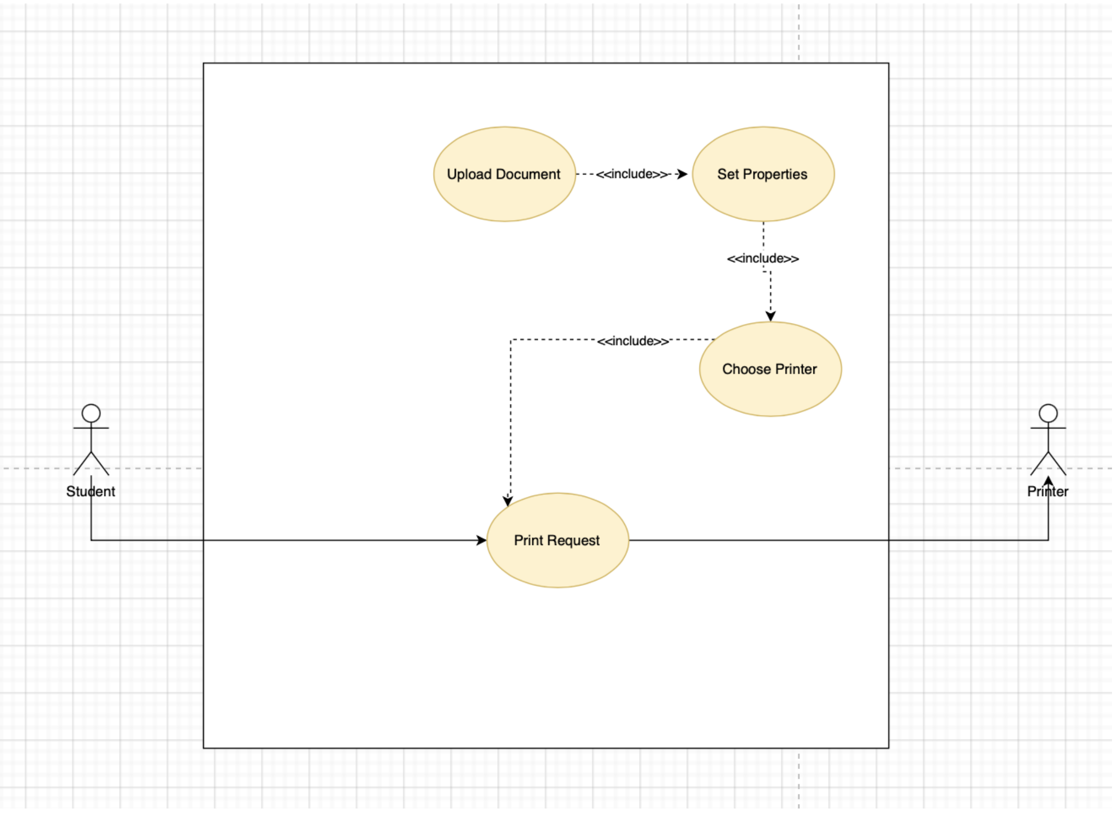
1 Use-case Diagram

1.1 Whole-System



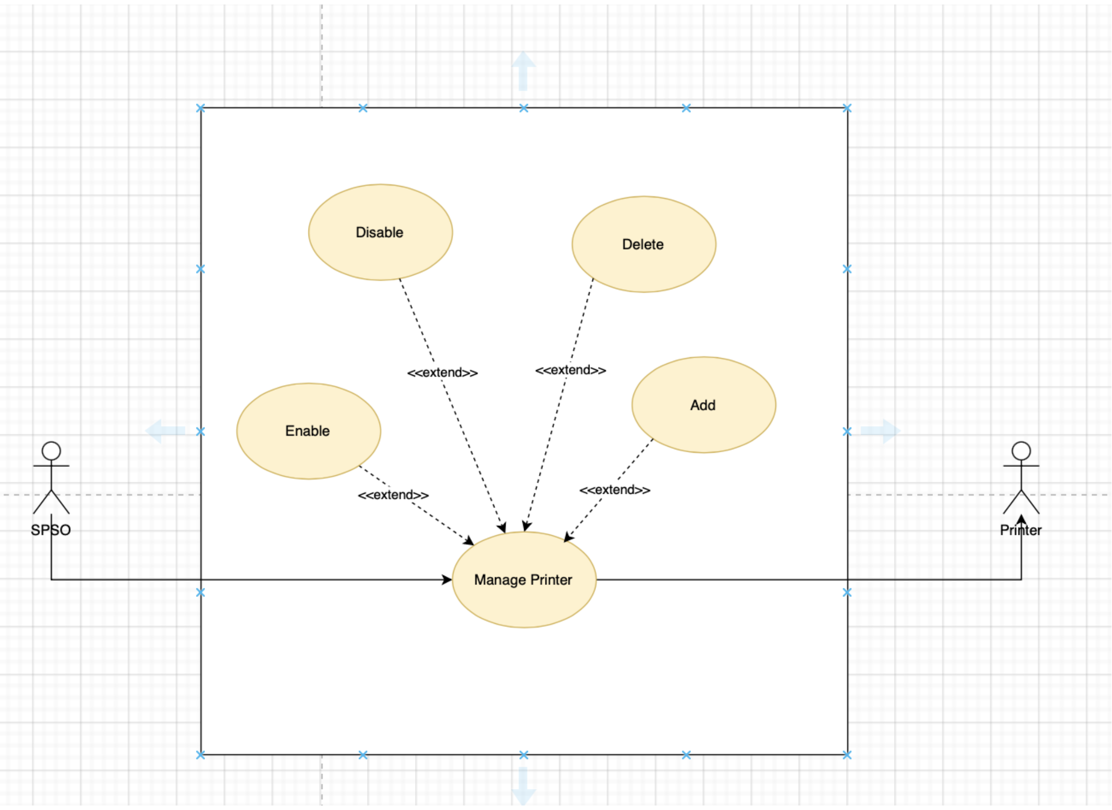
The use-case diagram includes five main actors interacting with the system, each playing a distinct role. The **Printer** interacts with the system through the **Printer Management** module, allowing it to manage and control its operations, and also provides access to the **View Print Log** functionality to review past print records. The **SPSO (System Print Service Operator)** oversees the **System Configuration** to manage and maintain the printing system's settings and ensure efficient operation. SPSO is also responsible for generating **Regular Reports** to monitor system performance and usage statistics. The **Student** primarily engages with the system to **Print Documents**, using the **Authentication** module to log in securely. Students can also access the **Buy Paper** module to ensure they have sufficient resources for printing and manage payments through **BK_Pay**. The **BK_Pay** actor facilitates transactions within the system, particularly for purchasing paper or credits needed for printing, and integrates with the **Buy Paper** module to handle secure payment processes. Lastly, **HCMUT_SSO (Single Sign-On)** provides centralized authentication services through the **Authentication** module, enabling seamless and secure access to the system for authorized users, including students and administrators. Each actor interacts with specific modules tailored to their roles, ensuring efficient operation and management of the printing system, while maintaining secure task execution and providing tools for system maintenance and reporting.

1.2 Printing Module



The use-case diagram illustrates the process of **Printing Module**, involving two actors: **Student** and **Printer**. The **Student** initiates the process by interacting with the system to **Upload Document**, which includes subsequent steps to **Set Properties** such as print quality, page orientation, and color preferences. This step is critical to ensuring that the document meets the desired specifications. After setting the properties, the student proceeds to **Choose Printer**, selecting the most suitable printer available in the network. This ensures that the system can route the print job to the selected printer effectively. Finally, the process culminates with the **Print Request**, where the selected printer processes the request and begins printing the document. The interaction between the **Student** and **Printer** through the system ensures an organized and user-friendly workflow for completing print tasks.

1.3 Printer Management

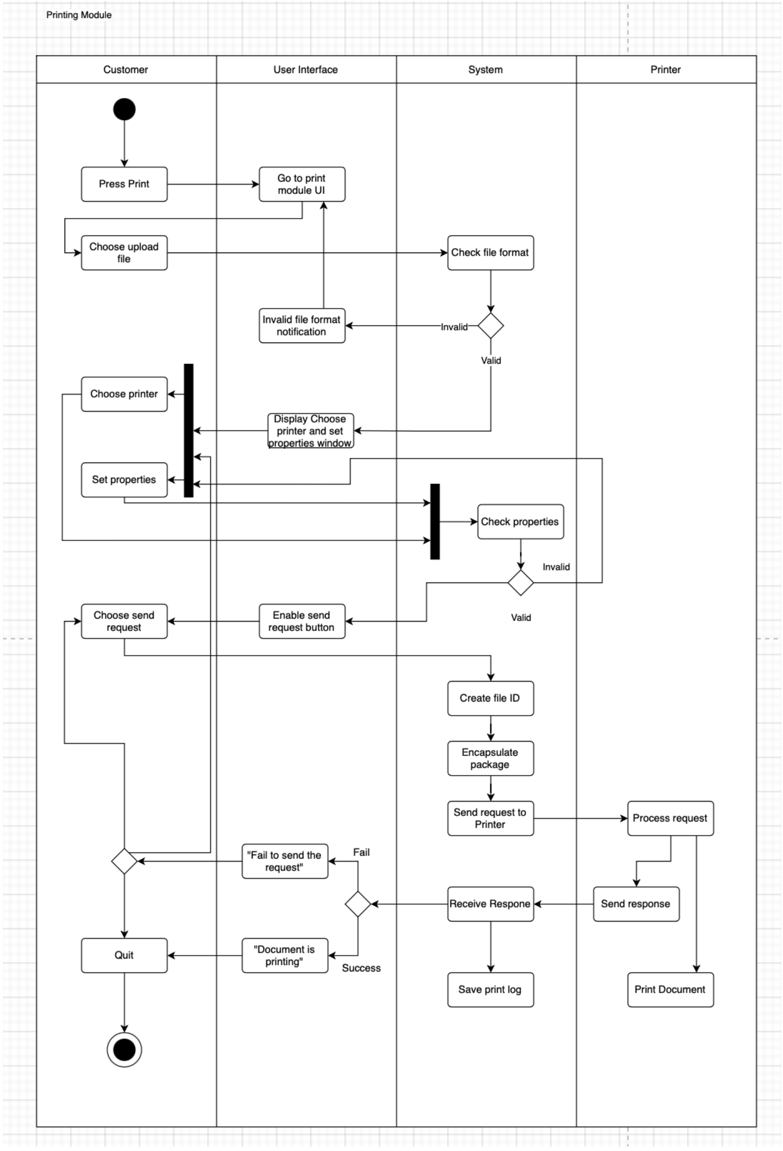


The use-case diagram focuses on the **Printer Management** module, which facilitates the management of printers by two primary actors: **SPSO (System Print Service Operator)** and **Printer**. The **SPSO** interacts with the system to perform key printer management tasks, which are modeled as extensions of the central **Manage Printer** use case. These tasks include the ability to **Add** new printers to the system, **Delete** printers that are no longer in use, **Enable** printers to make them operational, and **Disable** printers when maintenance or deactivation is required. Each of these actions ensures that the printers are effectively managed and maintained, contributing to seamless system operations. The **Printer** actor is indirectly involved, as it is the entity being managed within the system. This setup allows the **SPSO** to maintain control over the printing infrastructure, ensuring optimal performance and availability.

2 Printing Module

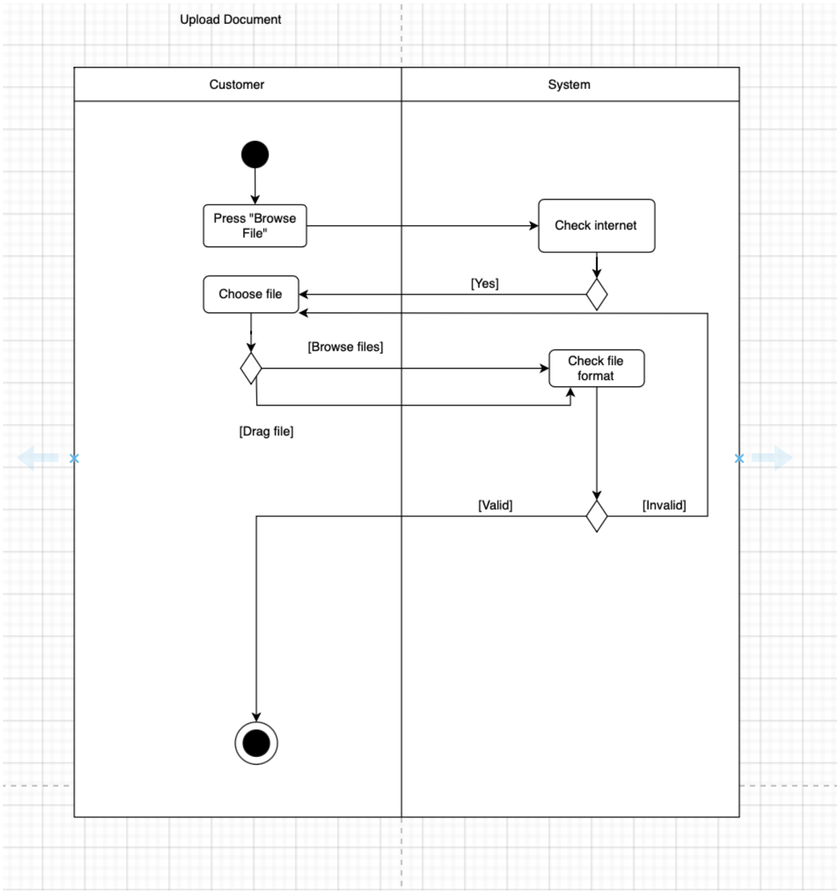
2.1 Activity Diagram

Printing Module Activity

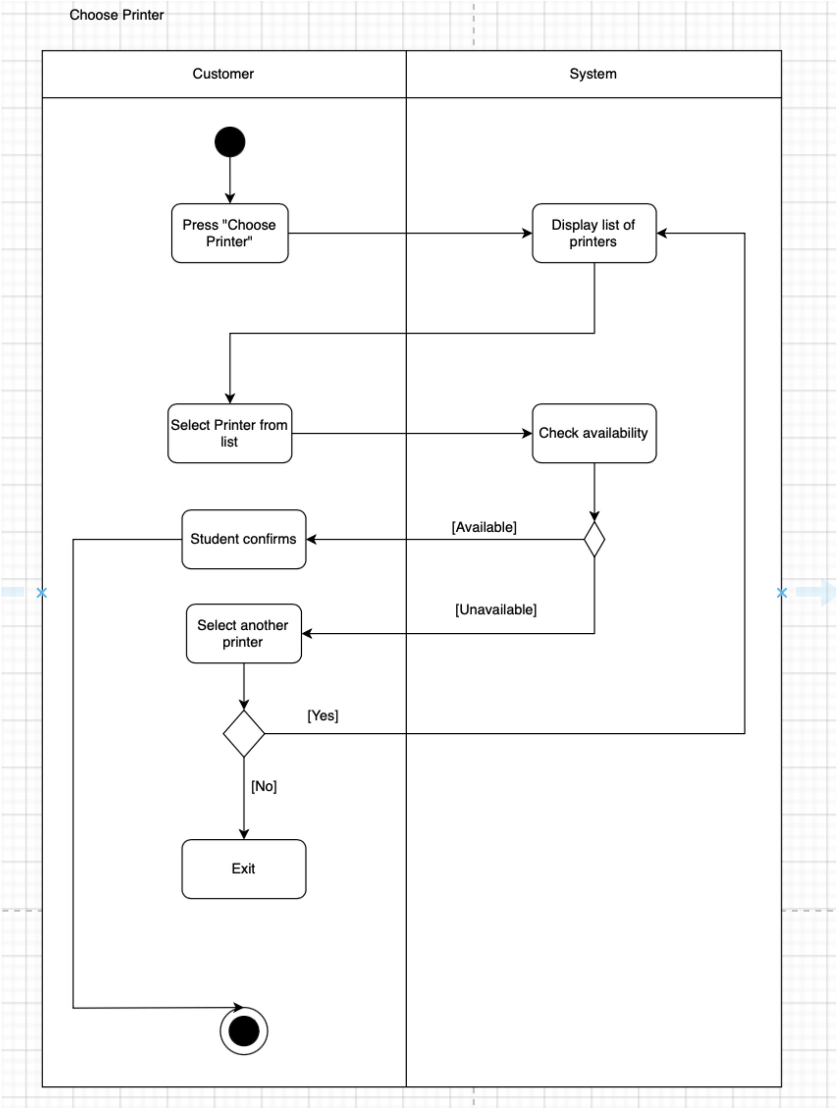


The following is an activity diagram for a Printing Service at HCMUT. Characterizes the interaction between the customer, the user interface, the system, and the printer. Here, the customer has to press the print button and then select a file to upload, which the system verifies to verify the integrity of the valid format. Check whether the format of the file is invalid; if so, inform the user. Otherwise, via the GUI, the customer chooses a printer and selects the properties of the printing. Then, the system verifies the settings of the printer. Once verified, the system allows activation of the send request button. The customer sends the request, and the system generates a file ID, encapsulates it into a package, and sends the request to the printer. The printer processes the document, prints it, and sends a response to the system. If successful, the system logs the print and informs the customer that their document is printing. If it fails, the system notifies the customer that it cannot send the request. This diagram outlines the entire workflow involved in printing, including validation steps and possible outcomes.

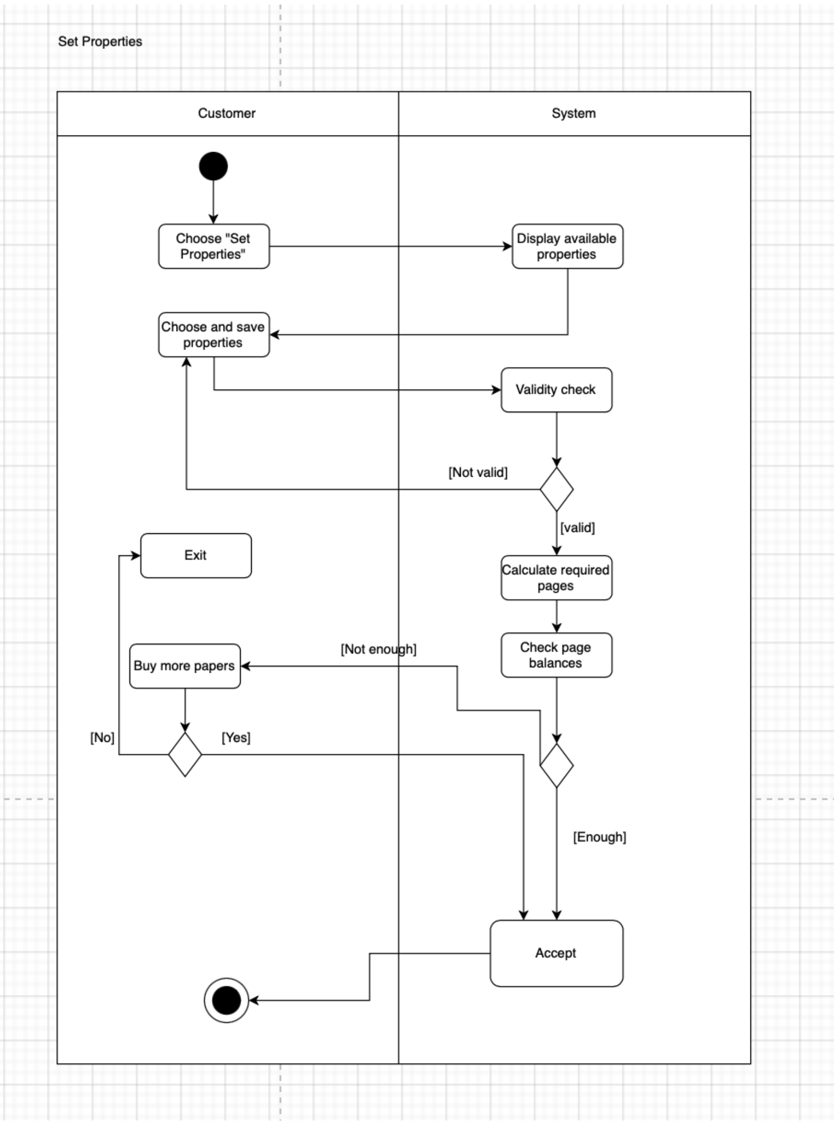
Upload Documents Activity



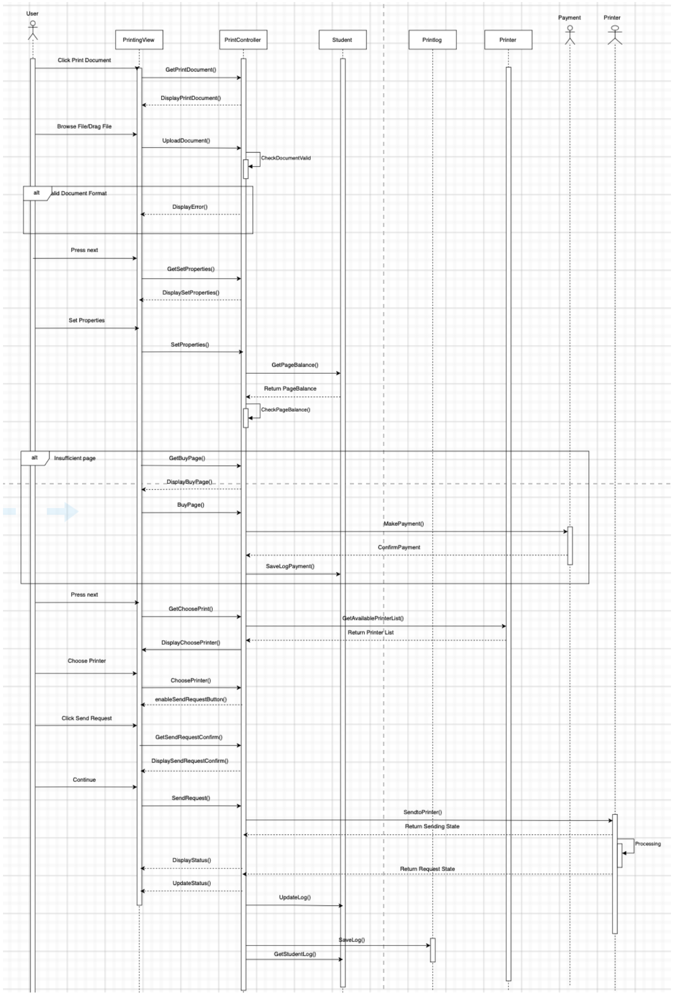
Choose Printer Activity



Set Properties Activity

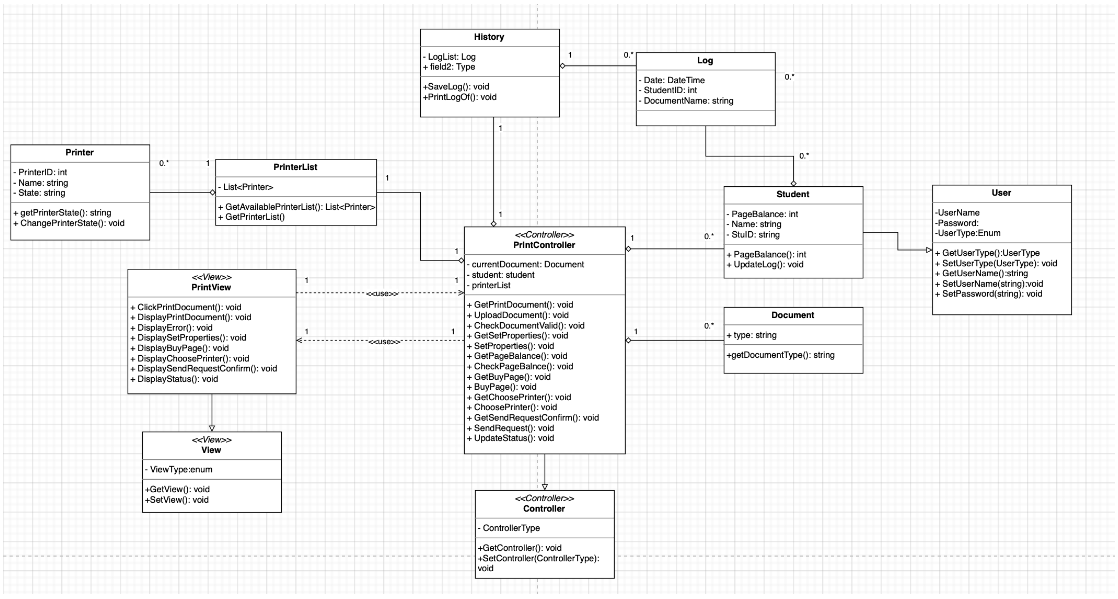


2.2 Sequence Diagram



This Printing sequence diagram illustrates the interactions between various components in a system over time. The diagram includes entities such as “User,” “UI Layer,” “Controller,” and others, each represented by a lifeline. Arrows between these lifelines indicate messages or function calls, showing the flow of control and communication. For example, the “User” interacts with the “UI Layer,” which in turn communicates with the “Controller” to perform specific actions.

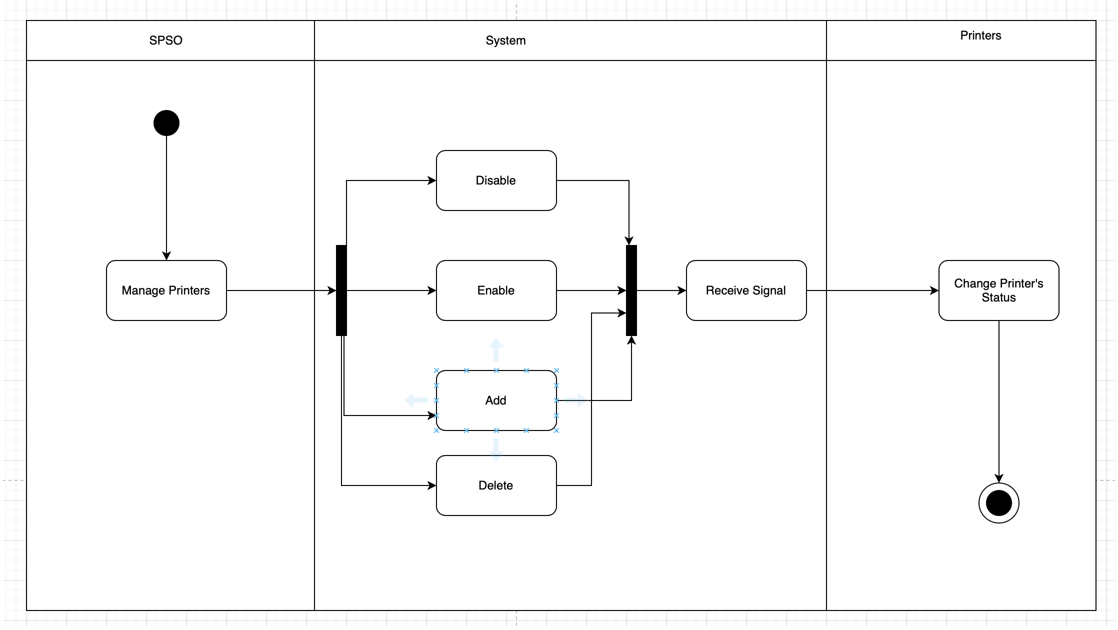
2.3 Class Diagram



The MVC class diagram includes several classes and their relationships. The ‘Printer‘ class has attributes for ID, name, and state, and is aggregated into the ‘PrinterList‘ class, which manages a list of printers and provides methods for retrieving available printers. The ‘PrintController‘, inheriting from the ‘Controller‘ class, interacts with ‘PrintView‘ (a subclass of ‘View‘), ‘PrinterList‘, ‘Document‘, and ‘Student‘. It manages printing actions, document uploads, property settings, and printer selection. The ‘Student‘ class, associated with ‘Log‘, includes attributes like page balance and student ID and is linked to the ‘History‘ class, which aggregates multiple ‘Log‘ entries and provides methods to save or print logs. The ‘User‘ class, connected to ‘Student‘, manages user credentials and types. The ‘Document‘ class includes methods to retrieve document types. The relationships in the diagram reflect the aggregation and associations among these classes, ensuring a structured interaction between views, controllers, and models.

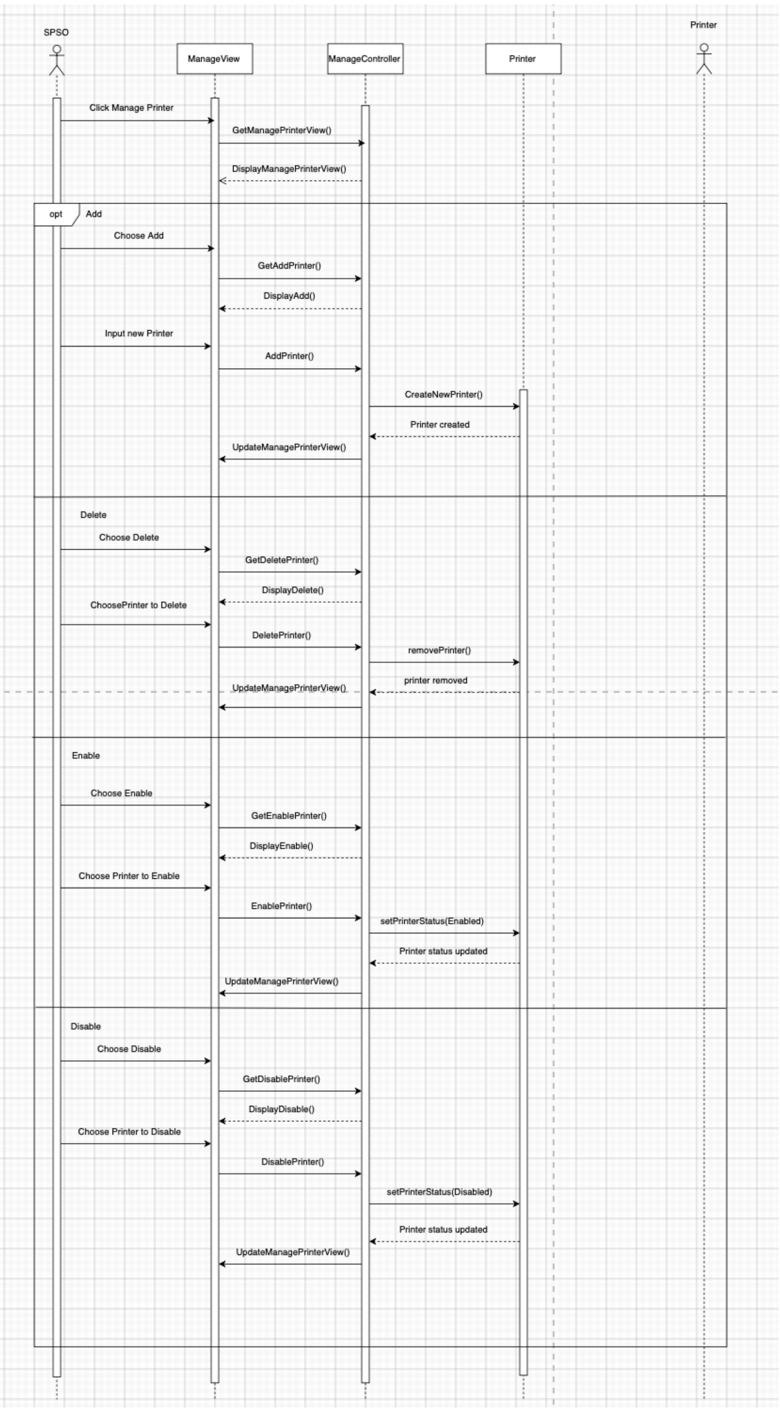
3 Printer Management

3.1 Activity Diagram



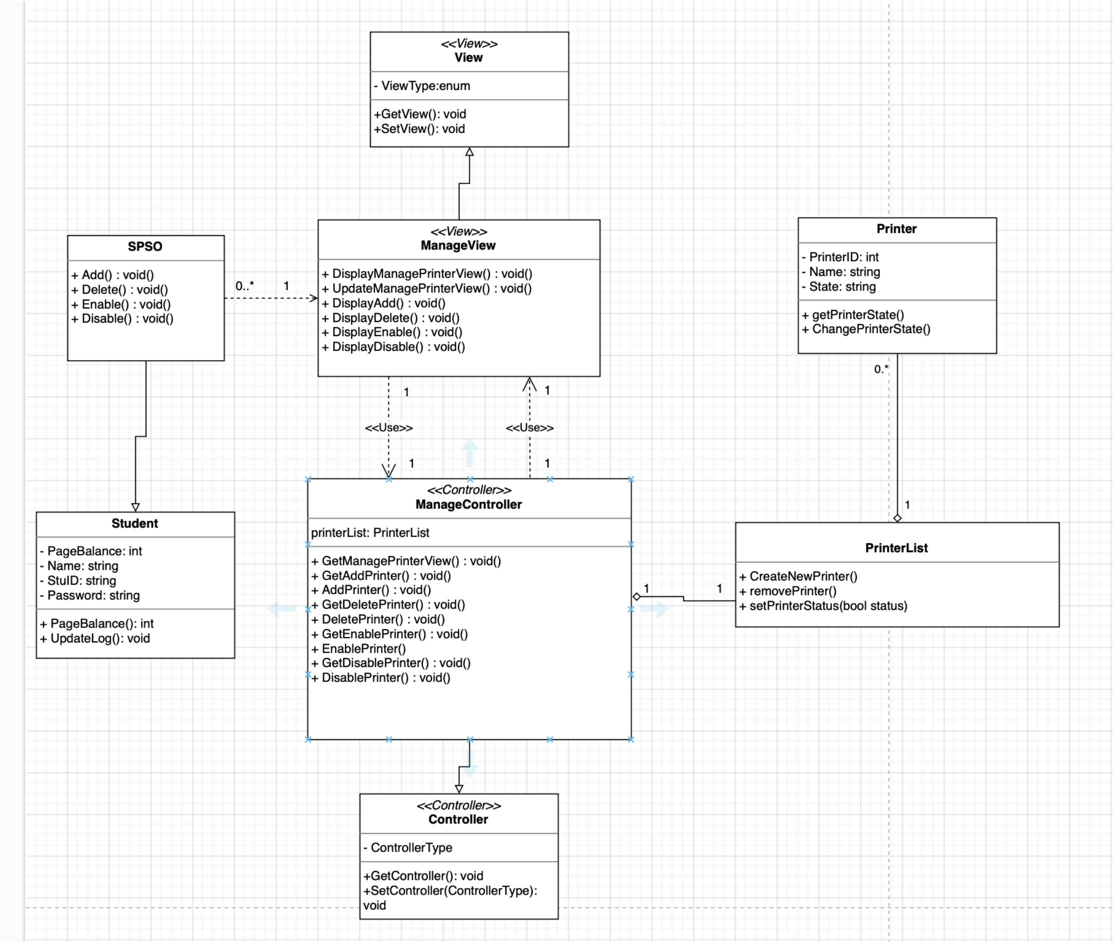
The activity diagram for Manage Module starts from an initial node labeled “SPSO” and branches into three parallel paths: “Manage Printers,” “System,” and “Printers.” Within the “System” path, activities such as “Decide,” “Enable,” “Add,” and “Delete” are depicted, showing the decision-making process and subsequent actions. The “Printers” path includes activities like “Receive Signal” and “Change Printer’s Status,” indicating how the system interacts with printers. The diagram concludes with a final node, representing the end of the process. This visual representation helps in understanding the dynamic behavior of the system, highlighting decision points and concurrent processes essential for managing printers effectively.

3.2 Sequence Diagram



This sequence diagram illustrates the interactions between various components in a system over time. The diagram includes entities such as “User,” “UIProcessManager,” “DataManager,” and others, each represented by a vertical lifeline. Horizontal arrows between these lifelines indicate messages or interactions, showing the flow of communication. The sequence starts with the "User" initiating an action, which is processed by the "UIProcessManager" and then passed to the "DataManager" for further handling. This step-by-step depiction helps in understanding the dynamic behavior of the system, making it easier to analyze and design the interactions between different components. This type of diagram is essential for visualizing the sequence of operations and ensuring that the system functions as intended.

3.3 Class Diagram



This class diagram illustrates a Printer Management System using the MVC (Model-View-Controller) design pattern. The SPSO class provides basic operations for adding, deleting, enabling, and disabling printers. The ManageView class handles the user interface, displaying relevant information and actions. ManageController acts as the intermediary, coordinating between ManageView and the model classes Printer and PrinterList. The Printer class represents individual printers, including attributes like ID, name, and state, while PrinterList manages a collection of printers, enabling batch operations. This structure promotes separation of concerns, with each class focused on specific functionalities for managing printers effectively.