

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

ARCHITECTURE LAYER

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 Thủy Tiên - 2252806

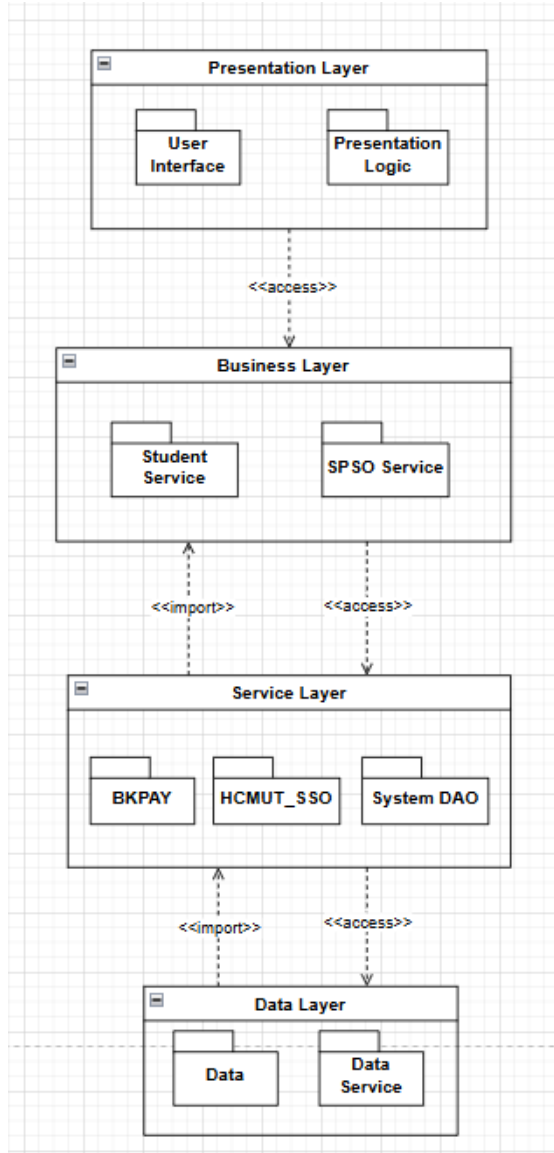
HO CHI MINH CITY, OCTOBER 2024



Contents

1	Presentation Layer Diagram	2
2	Component Diagram	4
2.1	Printing	4
2.2	Printer Management	5
3	Deployment Diagram	6

1 Presentation Layer Diagram



Presentation Strategy

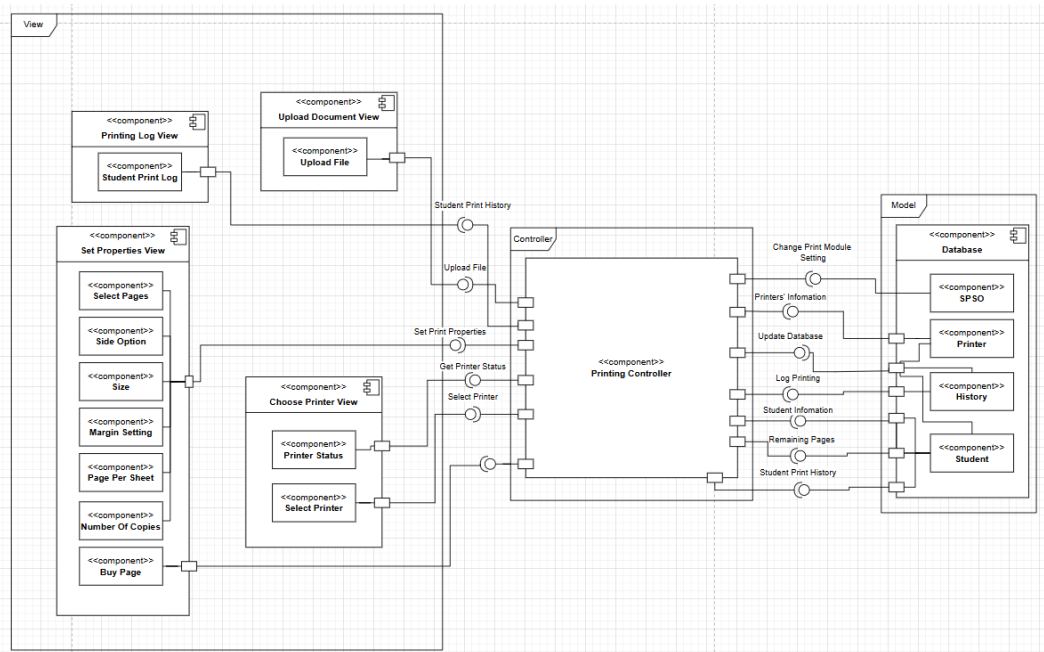
The system architecture is structured into four layers: Presentation Layer, Business Layer, Service Layer, and Data Layer, promoting modularity, scalability, and maintainability. The Presentation Layer serves as the user interaction point, consisting of the User Interface and Presentation Logic. It handles inputs, displays outputs, and processes data flows between the interface



and the underlying business logic. This layer communicates directly with the Business Layer, which encapsulates core functionality through its Student Service and SPSO Service components. The Student Service manages student-related operations like document printing and purchase transactions, while the SPSO Service handles system configurations, access controls, and report generation for administrators. The Business Layer processes user requests by leveraging the Service Layer to access data and external systems. The Service Layer acts as a bridge between the Business Layer and Data Layer, including specialized services like BKPAY for payment processing, HCMUT_SSO for authentication, and System DAO for handling database interactions. These services manage external integrations and abstract complexities, allowing the Business Layer to focus on core operations without delving into data access specifics. The Data Layer at the foundation comprises the Data component and Data Service, which handles persistent data storage and retrieval, ensuring data integrity and consistency. It supports CRUD operations efficiently and maintains secure data communication via protocols like TCP/IP.

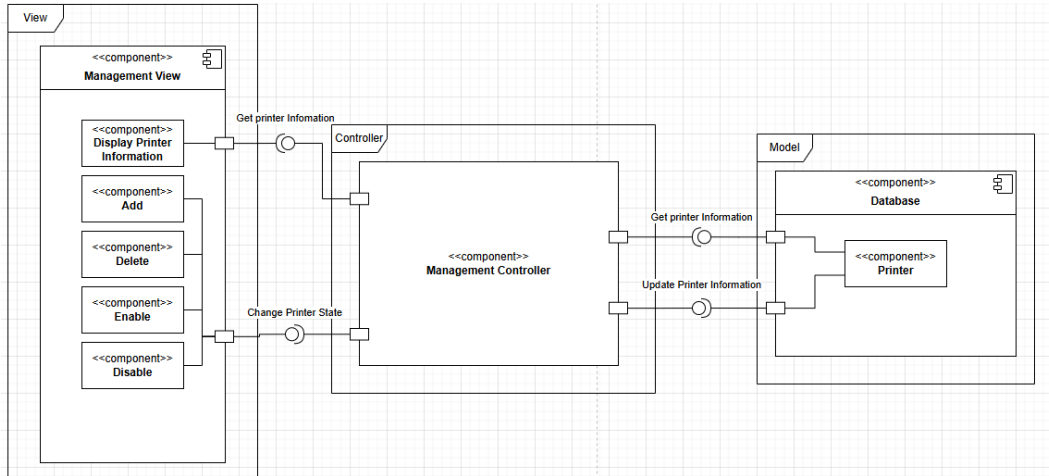
2 Component Diagram

2.1 Printing



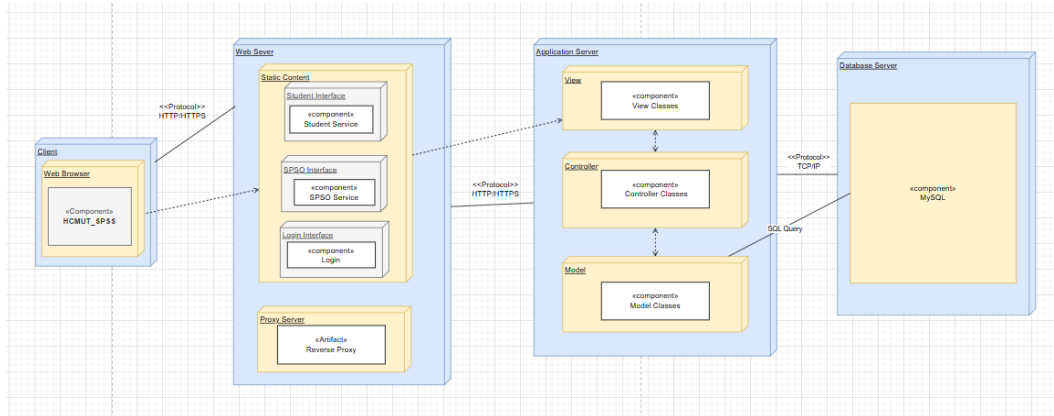
The system architecture utilizes a Model-View-Controller (MVC) design pattern to manage a printing service, ensuring modularity and maintainability. The View Layer interacts with users, providing functionalities like uploading documents, customizing print settings, viewing print logs, and selecting printers. The Controller Layer, primarily through the Printing Controller, processes these user inputs, manages business logic, and coordinates actions such as setting print properties, retrieving printer status, and handling user print histories. The Model Layer focuses on data management, with components for storing user information, printer settings, print logs, and system configurations within the database. The workflow begins with users interacting through the View, with the Controller processing these inputs and the Model ensuring data is accurately updated, logged, and stored, thus maintaining system integrity.

2.2 Printer Management



Printer Management System using the Model-View-Controller (MVC) architecture. The View Layer includes a Management View interface that allows users to manage printers through options like displaying printer information, adding, deleting, enabling, or disabling printers. The Management Controller serves as the intermediary, processing user actions from the view and interacting with the Model Layer. It retrieves printer information, updates printer states, and manages changes based on user input. The Model Layer consists of a Database that stores details in the Printer component, ensuring that printer information is consistently updated and accurately reflects the current system status.

3 Deployment Diagram



The Client Layer features a Web Browser interacting with the system via HTTP/HTTPS protocols, utilizing the HCMUT_SPSS component for authentication. The browser communicates directly with the Web Server, which hosts Static Content such as the Student Service Interface, SPSO Service Interface, and Login Interface. A Reverse Proxy is utilized to manage incoming requests efficiently, handling SSL termination and directing traffic appropriately. The Application Server is structured using the Model-View-Controller (MVC) pattern. It includes View Classes to handle presentation logic, Controller Classes to manage the business logic, and Model Classes for data management. This separation ensures modularity, allowing for easy maintenance and scalability. The Application Server connects to the Database Server via TCP/IP protocols, executing SQL Queries on a MySQL Database to store and retrieve application data securely.