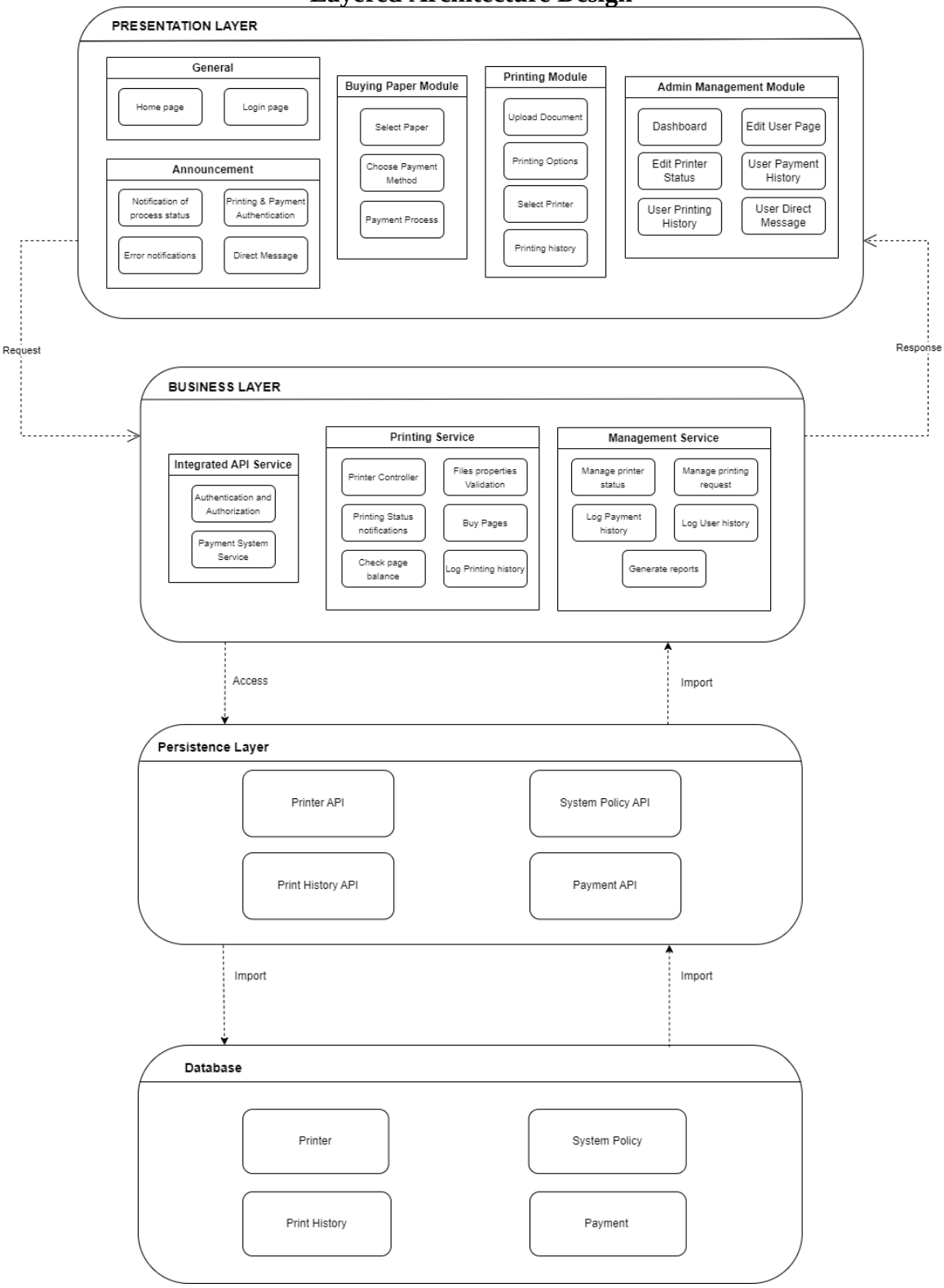
3.1 Layered Architecture

Layered Architecture Design



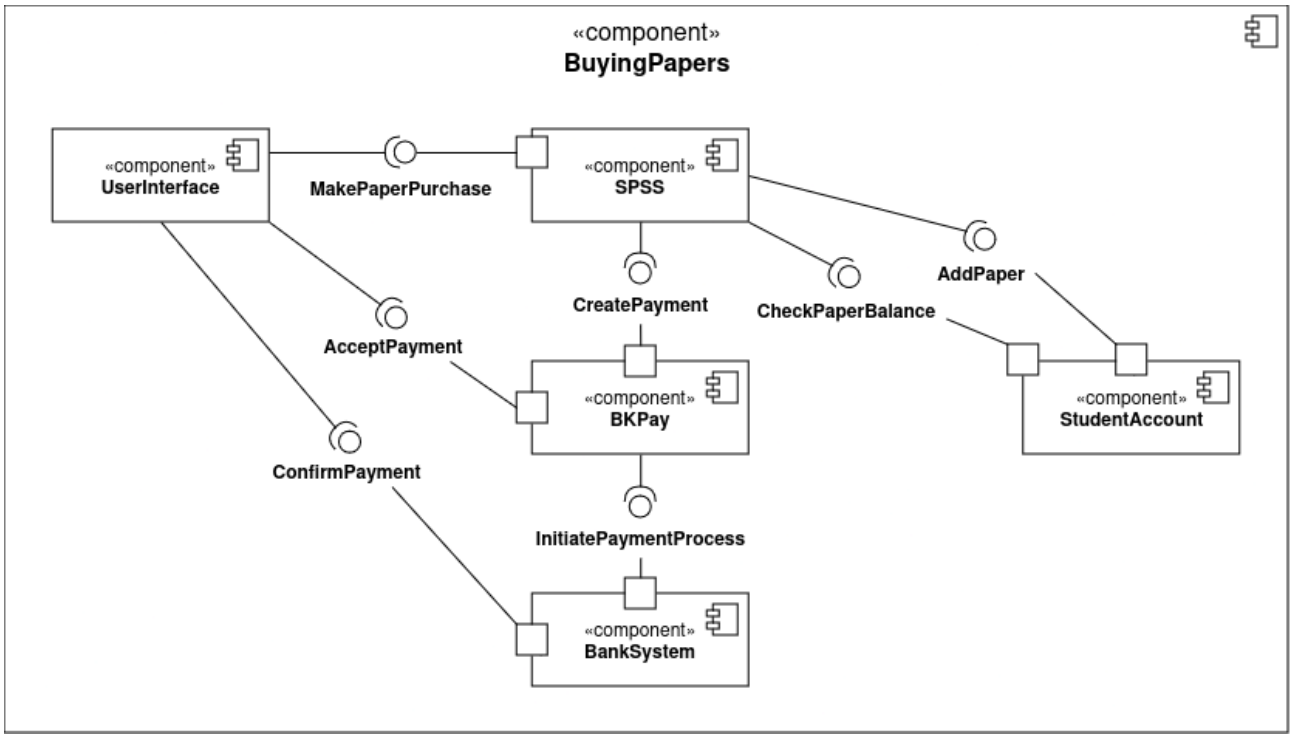
Presentation Strategy

The presentation layer of the system architecture is designed to provide a user-friendly interface for both students and administrators. By structuring this layer into multiple modules: General, Announcement, Buying Paper, Printing Module, and Admin Management Module, we create an intuitive and efficient user experience. This approach allows for scalability, enabling future expansion and easy maintenance. The UI/UX is designed for modular applications, this strategy facilitates a smooth workflow and promotes high user satisfaction.

Data Storage Approach
For the HCMUT_SSPS system, our team is about to use MySQL. This SQL database is renowned for its user-friendliness, scalability, performance, and flexibility. It's a favorite among beginners due to its ease of use and familiarity. Additionally, MySQL's document-oriented data model supports unstructured data with its flexible schema, making it ideal for systems with changing data needs. Furthermore, MySQL's built-in replication and sharding features enable horizontal scaling as data volume grows, ensuring high availability and fault tolerance.

External services/ API
According to requirements, our system must integrate with the university's authentication system, which employs the Central Authentication Service (CAS) protocol to offer single sign-on (SSO) functionality. Consequently, we must adhere to the HCMUT_SSO service protocols to authenticate and retrieve student accounts so that our application can function properly.

○ 3.2 Component Diagram



The BuyingPapers component diagram depicts a modular system for managing student's buying paper transactions. There are multiple components, including a User Interface for student

interaction, the SPSS, a Student Account component maintaining student data (such as paper balance...), the BKPay component managing payments, and the bank system component where students execute transactions. The User Interface interacts with SPSS to make a paper purchase, and then the SPSS checks the paper balance in the student account. The SPSS interacts with BKPay to create a payment and then provides an interface for students to accept. After payment is agreed upon, BKPay will initiate the transaction process, and navigate the student to the bank system, which provides an interface to confirm the payment. Finally, SPSS will update the paper balance in the student account.