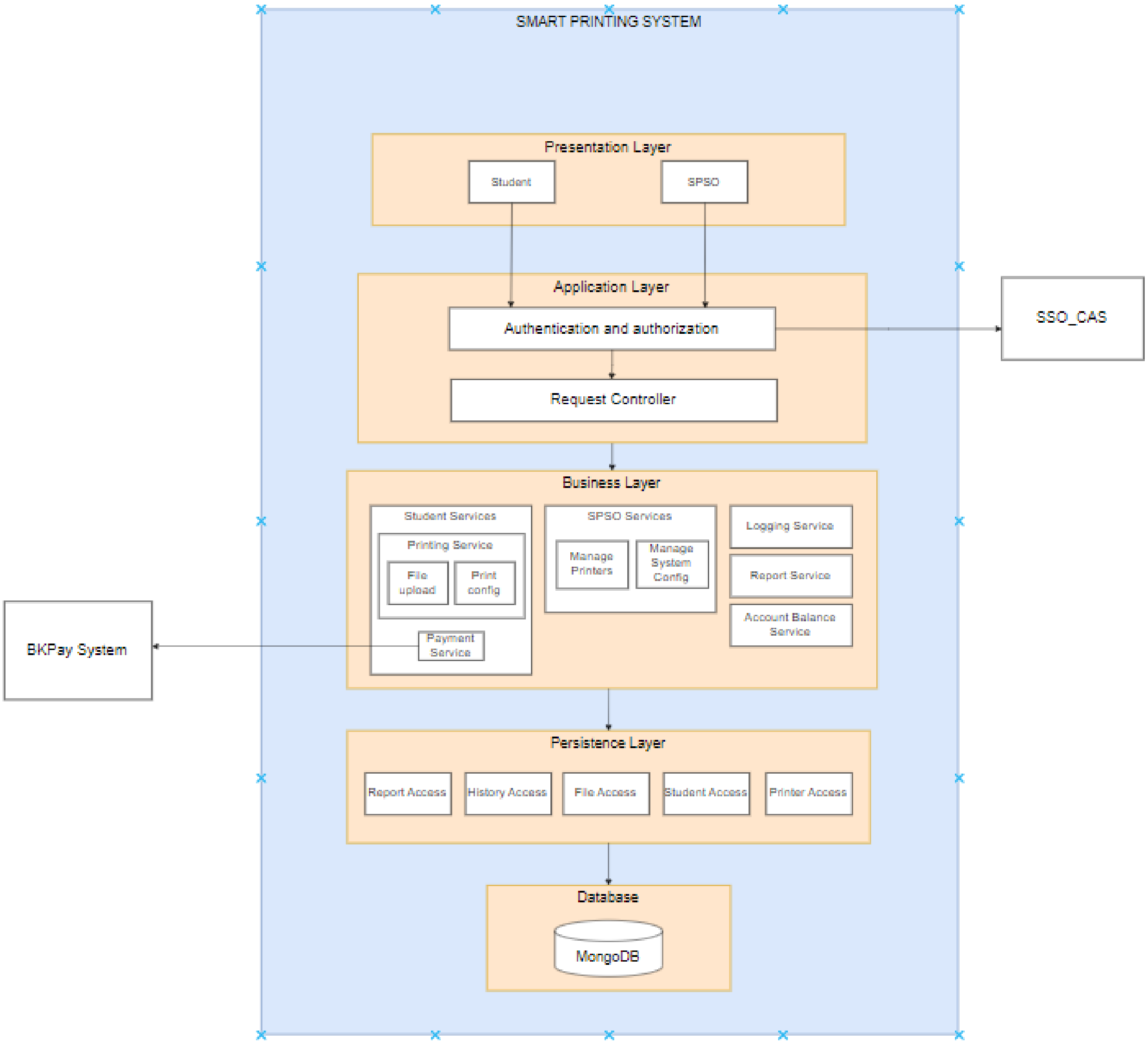

Reason:

The reason we choose this type of architecture is that layered architecture allows for clear separation of concerns, scalability, maintainability, and flexibility in your smart printing service system, ensuring that each layer performs its designated tasks while providing a structured and efficient overall system design.

Layered Architecture Description:

Presentation Layer:

The presentation layer serves as the outermost layer of the system, encompassing both the customer and administrator interfaces. This layer is responsible for delivering a userfriendly and visually appealing experience to end-users. It includes web-based interfaces

that allow customers to upload files, select printers, and submit print jobs.

Simultaneously, administrators have access to specific features and controls for managing printers, user accounts, and system settings. The presentation layer is designed using web technologies and ensures a seamless and intuitive user interaction with the application.

Application Layer:

The application layer sits between the presentation layer and the underlying business logic. It plays a pivotal role in processing and managing requests from the user interfaces. Within this layer, you'll find components responsible for request routing and control (request controller), handling user views and interactions (view handler), as well as implementing authentication and authorization mechanisms to secure system access. It acts as a bridge between the presentation layer and the business layer, orchestrating user interactions and enforcing access control policies.

Business Layer:

The business layer represents the heart of the smart printing service system, where the core functionality and logic of the application reside. It encapsulates the essential operations required for managing user accounts, handling print job submissions, printer selection, and any other business-specific processes. This layer is designed to be independent of the presentation layer and focuses on delivering the system's key functionality while maintaining flexibility and reusability.

Persistence Layer:

The persistence layer is responsible for managing data storage and retrieval. It abstracts the interaction with the database and ensures efficient data access. In this layer, you'll find components dedicated to querying, updating, and maintaining the database's integrity, making use of MongoDB in your case. It separates the business logic from the underlying data storage, promoting data consistency and scalability.

Database:

At the lowest level of the architecture, the database layer stores and manages all the system's data, including user profiles, printer configurations, print job histories, and related information. MongoDB is used as the database system, providing a NoSQL, document-oriented database suitable for handling both structured and unstructured data efficiently. The database layer ensures data durability and provides a structured and organized storage solution to support the application's functionality.

Presentation Strategy:

Our presentation strategy for the smart printing service system will prioritize a userfriendly web interface that offers seamless login and file upload functionality. We'll employ modern web technologies to create an intuitive, responsive, and visually appealing front-end using React.js, ensuring a smooth user experience. By utilizing a combination of HTML and CSS, or bootstrap, we will enable users to easily select from a variety of available printers and submit their print jobs with minimal friction. The presentation layer will be designed for both desktop and mobile devices, providing a consistent experience across different platforms and screen sizes.

Data Storage Approach:

To efficiently manage user accounts, print job details, and printer information, we will employ MongoDB as our database solution. MongoDB's NoSQL architecture is wellsuited for our needs, as it can handle unstructured data such as user profiles, print job histories, and printer configurations. We will organize data into collections that are logical and easy to query, ensuring fast retrieval and updates. This approach will enable us to scale the system smoothly as more users and printers are added, providing a reliable and flexible data storage solution for our smart printing service.

API Management:

API management is critical for ensuring the smooth interaction between the front-end and the back-end components of our smart printing service. We will create RESTful APIs using Express.js to handle user authentication, file uploads, printer selection, and job submission. To manage these APIs effectively, we will implement authentication and authorization mechanisms to secure user data and print job information. Additionally, we will document our APIs comprehensively to facilitate easy integration for potential third-party services in the future, promoting extensibility and collaboration within the printing ecosystem. Using API management tools, we will monitor and optimize API performance, ensuring a responsive and reliable service for our users.