# **Software Design Specification (SDS)**

## **Project Name: GlowVibe Beauty Clinic Web Application**

### **Prepared By: Rofida, Menna, Abeer, Sara, Manar**

### **Date: 21/12/2024**

## **1. Introduction**

### **1.1 Purpose**

The purpose of this document is to describe the design, architecture, and technical specifications of the GlowVibe platform. It outlines the functionality, system components, and design decisions to be followed during development. The goal is to create a user-friendly and efficient platform that simplifies appointment scheduling and consultation management for beauty services, enhancing the experience for clients and professionals alike.

### **1.2 Scope**

This SDS covers the design and implementation details of the GlowVibe platform. The software will perform the following major tasks:

* **User Registration and Profile Management:** Allow clients and professionals to register, log in, and manage their profiles.
* **Appointment Booking and Scheduling:** Enable clients to view available times and book, modify, or cancel appointments with beauty professionals.
* **Service Catalog and Reviews:** Display a categorized list of services, allow clients to filter services by category and price, and enable clients to leave reviews and ratings for professionals.

## **2. System Overview**

### **Components**

* **Frontend:** React for a responsive and interactive user interface.
* **Backend:** Node.js for handling server-side logic.
* **Database:** MongoDB to manage customer data, bookings, and service information.

## **3. System Architecture**

### **3.1 Architectural Design**

This project follows a **client-server architecture**, where:

* **Frontend** communicates with the **backend** using RESTful APIs.
* HTTPS protocol ensures secure and reliable data transmission.
* The backend interacts with the database to manage and retrieve data.

### **3.2 Data Flow**

1. **User Interaction:** Users interact with the UI to perform actions (e.g., booking an appointment, viewing profiles).
2. **Request Processing:** The frontend sends API requests to the backend server.
3. **Data Handling:** Upon receiving requests, the backend validates the data, interacts with the database, and processes the request (e.g., checking appointment availability).
4. **Response:** The backend sends responses to the frontend, updating the UI (e.g., confirmation or error messages).

## **4. Database Design**

### **4.1 Database Schema**

The database is designed using the MongoDB NoSQL model, with collections representing various entities in the system. The schema ensures efficient querying, relationships between collections, and optimal storage of hierarchical data.

#### **Key Collections:**

1. **Users Collection:** Stores information about all users, including clients and professionals.
2. **Appointments Collection:** Tracks all bookings made by clients.
3. **Services Collection:** Contains details of services offered by beauty professionals.
4. **Reviews Collection:** Stores client feedback.
5. **Products Collection:** Manages information about products available for sale.

#### **Relationships:**

* **Users ↔ Appointments:** One-to-Many
* **Users ↔ Services:** One-to-Many
* **Users ↔ Reviews:** One-to-Many
* **Appointments ↔ Services:** Many-to-One

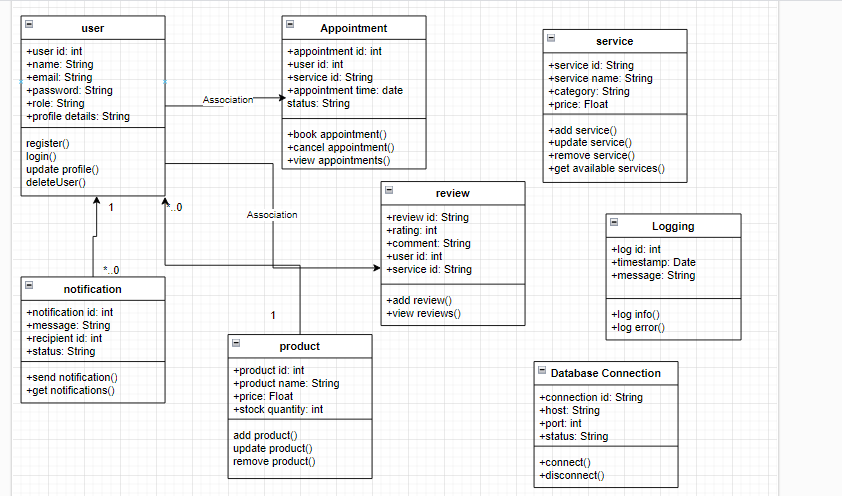
## **5. Design Patterns**

### **5.1 A Well-Structured Design**

* **Purpose:** Organize the system into clear and modular components, ensuring logical separation of concerns.
* **Implementation:** Applied throughout the project using modular architecture. Each service (e.g., Authentication, Appointment, Payment) operates independently but integrates seamlessly.
* **Benefits:**
  + Improves scalability and adaptability for future enhancements.
  + Facilitates understanding of the system's structure for new developers.

### **5.2 Maintainable Architecture**

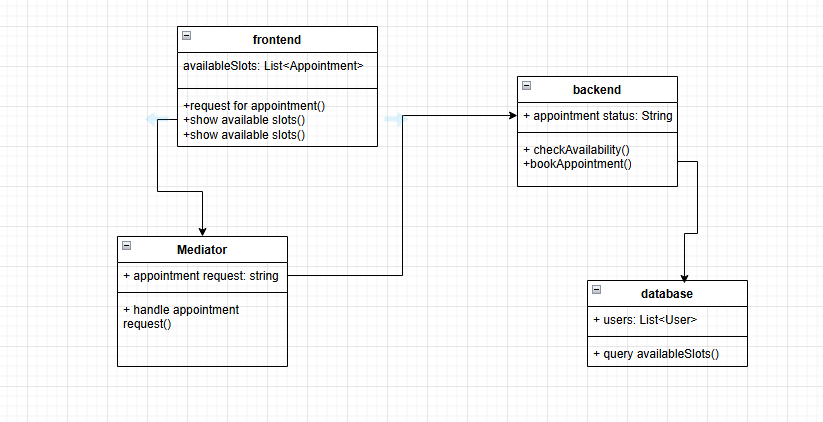
* **Purpose:** Ensure the codebase is easy to update, debug, and expand over time.
* **Implementation:**
  + Followed coding standards and best practices (e.g., reusable components in React, RESTful APIs in Node.js).
  + Used dependency injection for decoupling modules.
* **Benefits:**
  + Reduces technical debt by avoiding hard-coded dependencies.
  + Simplifies the process of fixing bugs or adding features.

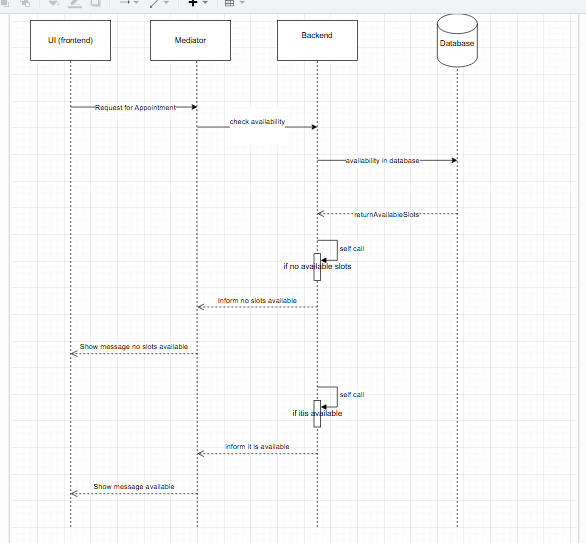


### **5.3 User-Friendly Interfaces**

* **Purpose:** Enhance user experience with intuitive and responsive interfaces.
* **Implementation:**
  + Built using React for dynamic and interactive UI components.
  + Designed user flows based on usability principles, such as simplicity and consistency.
* **Benefits:**
  + Increases user satisfaction and retention.
  + Provides seamless navigation and functionality across devices.

### **5.4 Mediator Pattern**

* **Purpose:** Simplify communication between components by centralizing interactions.
* **Implementation:** Used in the Notification Service to manage message dispatching between users and other services (e.g., appointment confirmation).
* **Benefits:**
  + Reduces direct dependencies between components, improving maintainability.
  + Enhances modularity by isolating interaction logic.
* 



## **6. Technology Stack**

* **Frontend:** React.
* **Backend:** Node.js.
* **Database:** MongoDB.
* **Hosting:** Azure, Heroku.

## **7. Testing Plan**

### **7.1 Unit Testing**

* Tools: Jest for frontend (React), Mocha for Node.js.
* **Scope:** Test individual UI components, backend logic, and database interactions independently.

### **7.2 Integration Testing**

* Tools: Postman for API testing and Supertest for HTTP assertions.
* **Scope:** Validate interactions between frontend, backend, and database.

### **7.3 User Acceptance Testing (UAT)**

* Tools: Test scripts, questionnaires, and Google Forms for feedback.
* **Scope:** Involve end users to test real-world scenarios (e.g., booking appointments).

### **7.4 Performance Testing**

* Tools: Apache JMeter or Gatling for load testing.
* **Scope:** Evaluate system performance under high concurrency.

## **8. Conclusion**

GlowVibe is designed to satisfy the functional and non-functional requirements detailed in this document. Its system architecture ensures reliability, user-friendliness, and adaptability. By simplifying the appointment booking and service management processes, GlowVibe aims to offer a valuable and intuitive platform for beauty and wellness clients and professionals.