**Project Name – Home Service Provider**

**Nikhil Chauhan- IIT2023217**

**Software Requirements Specification (SRS) for Service Provider-Client Platform**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to outline the detailed requirements for the development of a web-based platform designed to connect service providers with customers who require various services, such as carpentry, masonry, electrical work, laundry services, and construction. This platform will act as a digital marketplace, allowing service providers to offer their expertise to potential customers, and enabling customers to easily search, book, and review services according to their needs.

This Software Requirements Specification (SRS) serves multiple audiences, including the development team, project managers, stakeholders, and quality assurance teams. It is intended to provide a comprehensive overview of the functional and non-functional requirements necessary to build the platform, ensuring that all parties involved have a clear understanding of the project's objectives, scope, and expectations. The document will also serve as a baseline for validating and verifying the final product, ensuring it meets the requirements outlined herein.

**1.2 Scope**

The scope of this project includes the development of a fully functional web platform that facilitates interactions between service providers and customers. Key functionalities include user registration and profile management, service listing creation and management, advanced search and filtering capabilities, booking and scheduling systems, secure payment processing, and a review and rating system.

The platform will support multiple user roles, including service providers, customers, and administrators, each with specific permissions and access levels. It will be designed to be highly scalable, secure, and user-friendly, accommodating a wide range of devices, including desktops, tablets, and smartphones. The platform will also incorporate real-time communication features, such as chat and notifications, to enhance user experience.

This SRS document will cover both functional requirements (e.g., specific features and interactions) and non-functional requirements (e.g., performance, security, and scalability). The platform will adhere to industry standards and best practices, ensuring compliance with relevant regulations, particularly those related to data protection, online transactions, and consumer rights.

**1.3 Definitions, Acronyms, and Abbreviations**

To ensure clarity and consistency throughout this document, the following definitions, acronyms, and abbreviations will be used:

* SRS (Software Requirements Specification): A document that describes the functionalities, features, and constraints of a software system.
* UI (User Interface): The visual components of the platform through which users interact with the system, including buttons, forms, and menus.
* UX (User Experience): The overall experience and satisfaction of users when interacting with the platform, influenced by the design, usability, and accessibility of the UI.
* API (Application Programming Interface): A set of rules and protocols that allow different software components to communicate with each other, enabling integration with third-party services.
* DBMS (Database Management System): Software used to store, retrieve, and manage data in database MongoDB.
* SSL/TLS (Secure Sockets Layer/Transport Layer Security): Protocols that ensure secure communication over a computer network by encrypting data transmitted between the platform and its users.
* OAuth 2.0: An open standard for authorization, allowing users to grant third-party applications access to their information without sharing passwords.
* JWT (JSON Web Token): A compact, URL-safe means of representing claims to be transferred between two parties, commonly used for authentication and secure information exchange.
* REST (Representational State Transfer): An architectural style for designing networked applications, typically used in the development of APIs for web services.

**1.4 References**

This section provides references to relevant documents, standards, and regulations that must be adhered to during the development of the platform. These references ensure that the platform is built in compliance with industry norms and legal requirements. The following are key references to consider:

* **GDPR (General Data Protection Regulation):** European Union regulation on data protection and privacy for all individuals within the EU and the European Economic Area (EEA). The platform must comply with GDPR if it processes data of EU citizens.
* **ISO/IEC 27001:** An international standard for managing information security. This standard will guide the implementation of security measures to protect user data on the platform.
* **PCI DSS (Payment Card Industry Data Security Standard**): A set of security standards designed to ensure that all companies that accept, process, store, or transmit credit card information maintain a secure environment. This standard will be crucial for implementing the payment processing system on the platform**.**
* **WCAG (Web Content Accessibility Guidelines):** A set of guidelines developed by the World Wide Web Consortium (W3C) to ensure web accessibility for people with disabilities. The platform will adhere to these guidelines to ensure an inclusive user experience.
* **IEEE 830-1998**: A recommended practice for Software Requirements Specifications, providing guidance on how to write an SRS document. This document will follow the structure and content recommendations outlined in IEEE 830-1998.

**2. Overall Description**

**2.1 Product Perspective**

The platform will be a comprehensive web-based application designed to serve as a digital marketplace, seamlessly connecting service providers with potential customers. It aims to bridge the gap between individuals or businesses offering specialized services and those seeking such services. This platform will facilitate interactions, transactions, and communications between both parties, making the process of finding and booking services more efficient and user-friendly.

Architecture: The platform will be built using a three-tier architecture, comprising the presentation layer (frontend), application logic layer (backend), and data layer (database). The presentation layer will be responsible for the user interface and will be built using modern web technologies such as HTML, CSS, and JavaScript, along with frontend frameworks like React.js or Vue.js for enhanced interactivity and responsiveness. The application logic layer will handle business logic, service orchestration, and API integration, using backend technologies like Node.js with Express.js or Django (Python). The data layer will consist of a robust database management system, such as PostgreSQL or MongoDB, to store and manage all platform data, including user information, service listings, and transaction records.

Integration with Third-Party Services: The platform will integrate with several third-party services to provide additional functionalities such as payment processing (e.g., Stripe, PayPal), real-time communication (e.g., WebSockets, Firebase), geolocation services (e.g., Google Maps API), and email/SMS notifications (e.g., SendGrid, Twilio). These integrations will enhance the platform’s capabilities, offering users a seamless and efficient experience.

Deployment and Hosting: The platform will be deployed on a cloud infrastructure such as Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure. This choice of hosting will ensure scalability, high availability, and reliability. The use of cloud services will also facilitate the implementation of automated scaling, load balancing, and backup mechanisms, ensuring that the platform can handle varying levels of user traffic and data storage requirements without compromising performance.

**2.2 Product Functions**

The platform will offer a wide range of functionalities tailored to meet the needs of its users, including service providers, customers, and administrators. Key product functions include:

* **User Registration and Profile Management:**
  + Users can register on the platform using their email addresses, social media accounts, or phone numbers.
  + Upon registration, users can create and manage their profiles, including uploading profile pictures, updating contact information, and specifying their preferences or service offerings.
  + Service providers will have the ability to create detailed service profiles, including descriptions, pricing, availability, and other relevant information.
* **Service Listing and Management for Providers:**
  + Service providers can create and manage listings for the services they offer, with options to categorize services, set prices, and indicate availability.
  + Listings can include multimedia elements such as images, videos, and documents to provide customers with a comprehensive understanding of the services offered.
  + Providers can update or remove listings as needed, and track the performance of their listings through analytics provided in the admin dashboard.
* **Search and Filter Functionality for Customers:**
  + Customers can search for services using keywords, categories, or specific filters such as location, price range, ratings, and availability.
  + Advanced filtering options will allow customers to narrow down search results based on criteria such as service type, distance from their location, and customer reviews.
  + The search engine will be powered by a robust search algorithm, possibly using Elasticsearch or Algolia, to ensure fast and accurate search results.
* **Real-Time Chat Between Customers and Providers:**
  + A real-time chat feature will enable customers and service providers to communicate directly through the platform, allowing them to discuss service details, negotiate terms, and confirm bookings.
  + The chat system will support text messaging, file sharing, and potentially voice or video calls, depending on the platform's requirements.
  + Notifications will be sent to users for new messages or updates, ensuring prompt communication.
* **Booking System with Calendar Integration:**
  + The platform will offer a booking system that allows customers to schedule services directly through the platform. Service providers can manage their availability through an integrated calendar, which customers can view to select convenient time slots.
  + The booking system will handle confirmations, rescheduling, and cancellations, with automated notifications sent to both parties.
  + The calendar integration will sync with popular calendar applications like Google Calendar or Outlook, ensuring that service providers can manage their schedules efficiently.
* **Payment Processing and Invoicing:**
  + Secure payment processing will be facilitated through integration with payment gateways such as Stripe or PayPal, allowing customers to pay for services directly through the platform.
  + The platform will support multiple payment methods, including credit/debit cards, digital wallets, and potentially cryptocurrency.
  + Automated invoicing will be provided to customers upon payment, with options for generating receipts and tracking transaction history.
* **Review and Rating System:**
  + Customers will be able to leave reviews and ratings for service providers based on their experiences. This feedback will be visible to other customers and contribute to the overall rating of the service provider.
  + Providers can respond to reviews, allowing for transparency and engagement with customers.
  + The review system will be moderated to ensure that feedback is constructive and adheres to platform guidelines.
* **Admin Dashboard for Platform Management:**
  + The admin dashboard will provide platform administrators with tools to manage users, service listings, transactions, and platform content.
  + Administrators can monitor platform activity, track user engagement, and generate reports for business analysis.
  + The dashboard will include features for handling disputes, managing platform settings, and ensuring compliance with legal and regulatory requirements.

**2.3 User Classes and Characteristics**

The platform will cater to three primary user classes, each with distinct characteristics and roles:

* **Service Providers:**
  + Service providers are individuals or businesses offering services such as carpentry, masonry, electrical work, laundry services, or construction. They will use the platform to list their services, manage bookings, communicate with customers, and receive payments.
  + Providers may vary in size and capacity, ranging from independent contractors to large service companies.
  + They require tools to effectively manage their service offerings, monitor customer feedback, and track their business performance.
* **Customers:**
  + Customers are individuals seeking services listed on the platform. They will use the platform to search for services, communicate with providers, make bookings, and submit payments.
  + Customers may have varying levels of familiarity with digital platforms, so the user interface must be intuitive and easy to navigate.
  + They will benefit from features such as detailed service listings, customer reviews, and secure payment options.
* **Administrators:**
  + Administrators are responsible for managing the platform, ensuring that it operates smoothly and that users adhere to platform policies.
  + Their tasks include user management, content moderation, transaction monitoring, and resolving disputes between users.
  + Administrators require access to comprehensive tools for overseeing platform activity, generating reports, and making data-driven decisions.

**2.4 Operating Environment**

The platform will be designed to operate in a flexible and scalable cloud environment, ensuring high availability, reliability, and performance. Key aspects of the operating environment include:

* **Cloud Hosting**: The platform will be hosted on a cloud service provider such as Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure. This will provide the necessary infrastructure for scalable computing resources, data storage, and network services.
* **Web Accessibility:** The platform will be accessible via modern web browsers (e.g., Chrome, Firefox, Safari, Edge) on various devices, including desktops, laptops, tablets, and smartphones. The user interface will be fully responsive, adapting to different screen sizes and resolutions.
* **Database Management:** The platform will use a robust DBMS such as PostgreSQL, MySQL, or MongoDB, depending on the data requirements. The database will store user information, service listings, transaction records, and other critical data.
* **API Integration:** The platform will integrate with various third-party APIs for payment processing, geolocation, communication, and other services. These integrations will be secure and optimized for performance.
* **Security Protocols:** The platform will implement SSL/TLS encryption to secure data transmission, along with other security measures such as firewalls, intrusion detection systems, and regular security audits.

**2.5 Design and Implementation Constraints**

The design and implementation of the platform will be subject to several constraints, which must be considered to ensure the system’s security, compliance, and performance:

* **Security Best Practices:**
  + All data transmitted between users and the platform must be encrypted using SSL/TLS to protect against eavesdropping and man-in-the-middle attacks.
  + The platform must implement robust authentication and authorization mechanisms, including OAuth 2.0 and JWT, to ensure that only authorized users have access to specific resources.
  + User input must be validated and sanitized to prevent common security vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
* **Data Privacy Regulations:**
  + The platform must comply with data privacy laws such as the General Data Protection Regulation (GDPR) for European users and other relevant regulations for users in different regions.
  + Users must be informed about the collection, processing, and storage of their personal data, and must be given options to control their data, including the ability to delete their accounts.
* **Performance Constraints:**
  + The platform must be designed to handle high volumes of concurrent users without significant degradation in performance. This includes optimizing database queries, minimizing server response times, and using caching strategies.
  + The platform must be capable of scaling horizontally to accommodate increasing user numbers, with load balancing and automated scaling mechanisms in place.

**3. System Features**

**3.1 User Registration and Profile Management**

The platform will offer a robust and flexible user registration system, allowing users to create and manage their profiles with ease.

* **User Registration:**
  + Users can sign up using multiple methods, including email, social media accounts (e.g., Google, Facebook, LinkedIn), or phone numbers.
  + The registration process will include email or SMS verification to ensure the authenticity of user accounts.
  + During registration, users will agree to the platform’s terms of service and privacy policy, ensuring compliance with legal requirements.
* **Profile Customization:**
  + Upon successful registration, users can customize their profiles by adding personal details such as name, profile picture, contact information, and service preferences.
  + Service providers will have additional options to create detailed service profiles, including business name, service categories, service descriptions, pricing structures, and availability schedules.
  + Users can update their profiles at any time, allowing them to keep their information current and relevant.
* **Account Management:**
  + Users can manage their accounts, including changing passwords, updating email addresses or phone numbers, and managing notification preferences.
  + The platform will also include options for account deactivation or deletion, with appropriate confirmation steps to prevent accidental loss of data.

**3.2 Service Listing and Management**

Service providers will have comprehensive tools to manage their service offerings effectively, ensuring that potential customers have access to accurate and up-to-date information.

* **Service Creation:**
  + Providers can create service listings through a user-friendly interface, with fields for service name, category, detailed description, pricing, and availability.
  + Listings can include multimedia elements such as images, videos, and documents to better showcase the services offered.
* **Service Management:**
  + Providers can easily edit or update their service listings, allowing them to modify service details, adjust pricing, or change availability based on demand or other factors.
  + Providers can also delete listings that are no longer relevant or have been discontinued, ensuring that the platform remains clutter-free and up-to-date.
* **Service Categorization:**
  + Services will be organized into categories (e.g., carpentry, masonry, electrical work, laundry services, construction), making it easier for customers to find the services they need.
  + Providers can assign their services to multiple categories if applicable, increasing visibility and reach.

**3.3 Search and Filter**

The platform will provide powerful search and filtering capabilities, enabling customers to find the services they need quickly and efficiently.

* **Search Functionality:**
  + Users can search for services using keywords, service categories, or specific criteria such as provider name or service location.
  + The search engine will be optimized for speed and accuracy, ensuring that users receive relevant results promptly.
* **Advanced Filtering Options:**
  + Users can refine their search results using a variety of filters, including service category, location (with support for geolocation-based searches), price range, ratings, and availability.
  + Filters will help users narrow down their choices, making it easier to find services that meet their specific requirements.
* **Sorting Options:**
  + In addition to filtering, users can sort search results based on criteria such as price (low to high or high to low), ratings (highest first), or distance (nearest first).
  + This functionality will enhance the user experience by allowing customers to prioritize the factors most important to them.

**3.4 Booking System**

The platform’s booking system will streamline the process of scheduling services, making it easy for customers to book and manage appointments**.**

* **Integrated Calendar:**
  + Service providers can manage their availability through an integrated calendar, which customers can view to select convenient time slots for bookings.
  + The calendar will sync with popular calendar applications like Google Calendar or Outlook, helping providers manage their schedules efficiently.
* **Booking Process:**
  + Customers can book services directly through the platform, selecting their preferred date and time based on the provider’s availability.
  + The booking system will handle real-time updates, ensuring that time slots are reserved as soon as a booking is confirmed.
* **Confirmation and Reminders:**
  + Once a booking is made, both the customer and the provider will receive a confirmation email or SMS, including details of the appointment.
  + The platform will also send automated reminders to both parties prior to the scheduled service, reducing the likelihood of missed appointments.
* **Rescheduling and Cancellations:**
  + Customers can reschedule or cancel bookings through the platform, with any changes immediately reflected in the provider’s calendar.
  + The system will enforce any cancellation policies set by the provider, ensuring fairness and transparency.

**3.5 Payment Gateway**

The platform will integrate with secure and reliable payment gateways, facilitating seamless transactions between customers and service providers.

* **Secure Payment Processing:**
  + Customers can make payments through trusted payment gateways such as Stripe or PayPal, ensuring that transactions are secure and compliant with industry standards.
  + The platform will support multiple payment methods, including credit/debit cards, digital wallets, and potentially cryptocurrency.
* **Invoice Generation:**
  + Upon successful payment, the platform will automatically generate and send an invoice to the customer, detailing the services provided and the amount paid.
  + Providers can also access and download invoices for their records, helping them track their earnings and manage their finances**.**
* **Payment Tracking:**
  + The platform will maintain a comprehensive record of all transactions, allowing customers and providers to view their payment history and monitor the status of pending payments.
  + Payment tracking will include details such as payment method, transaction ID, and date of payment, ensuring transparency and accountability.

**3.6 Review and Rating System**

The platform will include a review and rating system, allowing customers to provide feedback on the services they receive and helping providers build their reputation**.**

* **Customer Reviews:**
  + After a service is completed, customers will have the opportunity to leave a review, rating the provider on factors such as quality, punctuality, and professionalism.
  + Reviews will be visible to other users, helping them make informed decisions when selecting service providers.
* **Provider Responses:**
  + Service providers will have the option to respond to reviews, allowing them to address any concerns or thank customers for positive feedback.
  + This feature will promote open communication and help resolve potential misunderstandings or disputes**.**
* **Moderation and Guidelines:**
  + The platform will implement moderation mechanisms to ensure that reviews are constructive and adhere to community guidelines.
  + Inappropriate or offensive reviews will be flagged and removed, maintaining a respectful and professional environment for all users.

**3.7 Real-Time Communication**

To facilitate seamless interactions between customers and service providers, the platform will offer real-time communication features**.**

* **Live Chat:**
  + The platform will include a live chat feature, enabling customers and providers to communicate directly through the platform.
  + The chat system will support text messaging, file sharing, and potentially voice or video calls, depending on the platform's requirements.
* **Notifications:**
  + Users will receive notifications for new messages, updates, or changes to their bookings, ensuring they stay informed and can respond promptly.
  + Notifications will be delivered via email, SMS, or push notifications, depending on the user’s preferences.
* **Communication History:**
  + The platform will maintain a record of all communications between users, allowing them to refer back to previous conversations if needed.
  + This history will be accessible through the user’s profile or booking details, ensuring transparency and accountability in all interactions.

**3.8 Admin Dashboard**

The platform will include a comprehensive admin dashboard, providing administrators with the tools they need to manage and monitor the platform effectively.

* **User Management:**
  + Administrators will have the ability to manage user accounts, including creating, updating, suspending, or deleting accounts as needed.
  + The dashboard will include tools for monitoring user activity, verifying service provider credentials, and resolving disputes between users.
* **Transaction Management:**
  + The admin dashboard will provide detailed insights into platform transactions, including payment processing, invoicing, and refunds.
  + Administrators can track financial metrics, monitor payment gateway integration, and ensure compliance with financial regulations.
* **Analytics and Reporting:**
  + The dashboard will feature analytics tools, allowing administrators to monitor platform activity, user engagement, and service performance.
  + Reports can be generated on demand, providing insights into key metrics such as user growth, transaction volume, and customer satisfaction.
* **Content Management:**
  + Administrators will have control over platform content, including managing service categories, setting platform policies, and curating featured services.
  + The dashboard will also include tools for managing platform notifications, promotional content, and user communications.
* **Security and Compliance:**
  + The admin dashboard will include features for managing platform security, including monitoring for potential security breaches, managing user permissions, and enforcing compliance with legal and regulatory requirements.

**4. External Interface Requirements**

* **User Interfaces**:
  + Intuitive and responsive UI/UX design.
  + Accessible on mobile and desktop browsers.
* **Hardware Interfaces**:
  + No special hardware requirements; standard web access.
* **Software Interfaces**:
  + APIs for payment processing, geolocation, and messaging.
  + Integration with third-party services like Google Maps, Stripe, etc.
* **Communication Interfaces**:
  + HTTPS for secure data transmission.
  + WebSockets for real-time communication.

**5. System Attributes**

* **Performance Requirements**:
  + The system should handle up to 10,000 concurrent users.
  + Response times should be under 2 seconds for all user actions.
* **Security Requirements**:
  + Data encryption at rest and in transit.
  + Regular security audits and vulnerability assessments.
* **Software Quality Attributes**:
  + **Usability**: The platform must be easy to use with minimal training.
  + **Reliability**: 99.9% uptime with automated backups.
  + **Scalability**: The platform should be able to scale horizontally to accommodate growing user numbers.
* **Other Requirements**:
  + Compliance with local regulations regarding data privacy and online payments.

**6. Other Nonfunctional Requirements**

* **Safety Requirements**: Ensure user data is securely stored and protected from unauthorized access.
* **Security Requirements**: Detailed logging and monitoring of all transactions and communications.
* **Software Quality Attributes**: Regular updates and maintenance to keep the platform secure and performant.
* **Business Rules**: Define the terms of service, user conduct policies, and dispute resolution procedures.

### Use Case Diagram

#### Purpose:

The use case diagram provides a high-level overview of how different users (actors) interact with the system. It highlights the primary functionalities of the platform and the roles performed by each type of user.

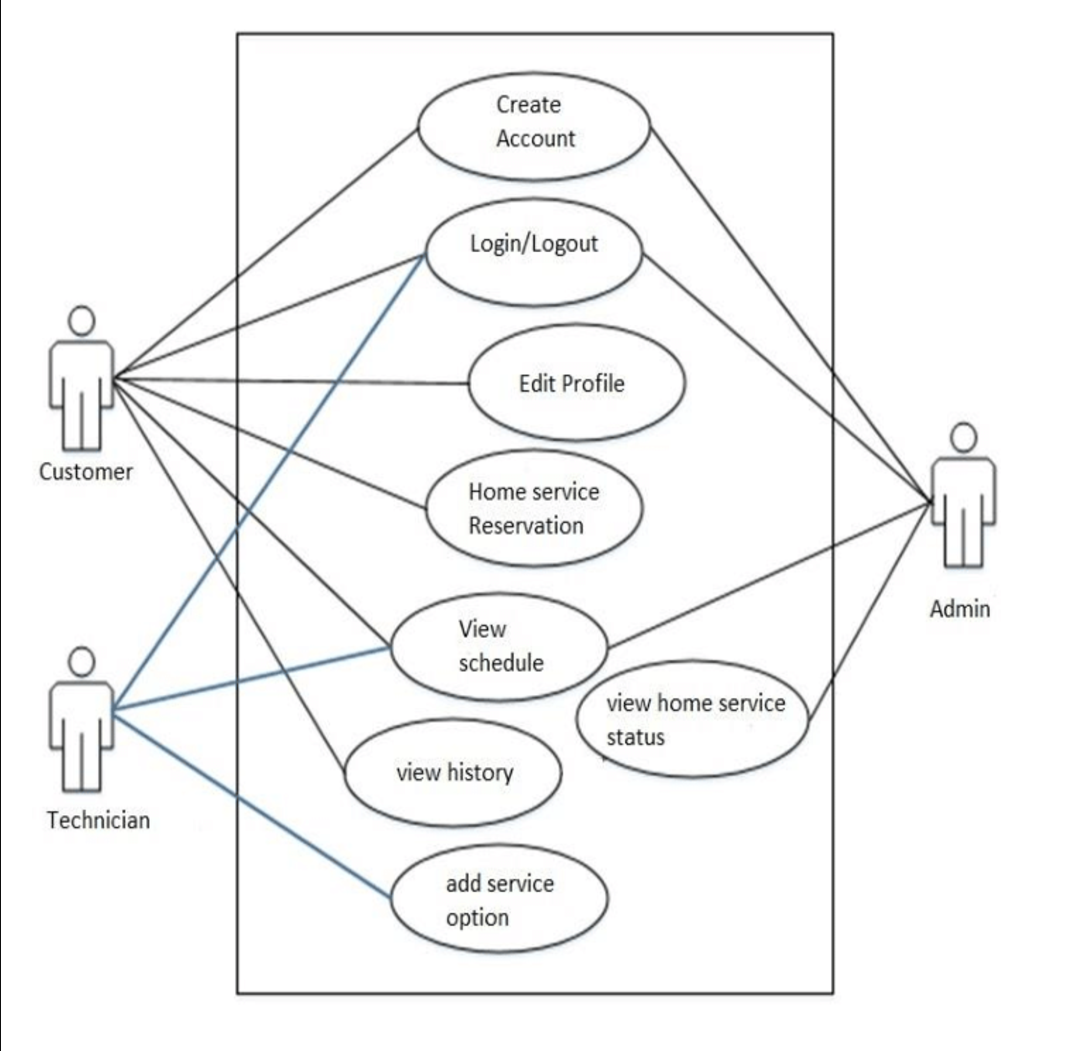
#### Description:

**Actors:**

1. **Customer:**
   * Registers or logs into the system.
   * Searches for services using filters.
   * Books services from listed options.
   * Provides reviews and ratings for completed services.
2. **Service Provider:**
   * Registers or logs into the system.
   * Lists their services for customers.
   * Manages bookings, including accepting or rejecting requests.
3. **Admin:**
   * Manages users (customers and service providers).
   * Monitors platform activities to ensure compliance.
   * Resolves disputes between customers and service providers.

**Use Cases:**

1. **User Registration/Login:**
   * Enables customers and service providers to create and manage accounts securely.
2. **Service Search and Filter:**
   * Allows customers to search and filter available services based on specific criteria (e.g., location, price, service type).
3. **Service Booking:**
   * Facilitates the booking process for customers to hire a service provider.
4. **Payment Processing:**
   * Provides a secure payment gateway for customers to complete transactions.
5. **Feedback Submission:**
   * Allows customers to submit reviews and ratings for services they have used.
6. **Admin Moderation:**
   * Enables the admin to monitor platform activities, manage user accounts, and handle disputes.



**State Diagram Specification (SDS)**

### Introduction to SDS

**Title:** State Diagram Specification Overview  
**Content:**

* **Purpose:**
  + Illustrates the lifecycle of the service booking process.
  + Provides a visual representation of states and transitions for clarity.
* **Focus:**
  + Tracks the journey of a service from availability to customer feedback.

### Key States in the Process

**Title:** States in the Service Booking Process  
**Content:**

* **Service Available:** The service is listed on the platform and visible for booking.
* **Service Requested:** A customer initiates a booking request.
* **Booking Confirmed:** The service provider confirms the booking request.
* **Payment Pending:** Awaiting payment from the customer.
* **Payment Completed:** Payment is successfully processed.
* **Service Delivered:** The task is completed by the service provider.
* **Feedback Received:** The customer provides feedback and ratings.

### Transitions Between States

**Title:** Transitions Between States  
**Content:**

* **Service Available → Service Requested:**Trigger: Customer selects and initiates a booking.
* **Service Requested → Booking Confirmed:**Trigger: Provider accepts the booking.
* **Booking Confirmed → Payment Pending:**Trigger: System prompts for payment.
* **Payment Pending → Payment Completed:**Trigger: Customer completes payment.
* **Payment Completed → Service Delivered:**Trigger: Service provider delivers the service.
* **Service Delivered → Feedback Received:**Trigger: Customer submits feedback.

### Visual Representation of SDS

**Title:** State Diagram Representation  
**Content:**

* **Diagram Description:**
  + The diagram shows all states and transitions.
  + States are represented as circles.
  + Transitions are represented as labeled arrows connecting the states.
* **Service Available → Service Requested → Booking Confirmed → Payment Pending → Payment Completed → Service Delivered → Feedback Received**

**Class Diagram Specification**

### Introduction to Class Diagram

**Title:** Introduction to Class Diagram

**Content:**

* **What is a Class Diagram?**
  + A class diagram is part of the Unified Modeling Language (UML).
  + It visually represents the **structure** of a system, focusing on **classes**, their **attributes**, **methods**, and their **relationships**.
* **Why Use a Class Diagram?**
  + To provide a **high-level view** of the system for stakeholders.
  + To define the **static architecture** for developers.
  + To **clarify relationships** and avoid implementation issues.

**Visual Suggestion:**

* Add an image or icon of a generic class diagram for visual appeal.

### Slide 3: Key Classes in the System

**Title:** Key Classes in the System

**Content:**Break down the classes as follows:

1. **User Class:**
   * **Attributes:** userID, name, email, password, role (Customer/Provider/Admin).
   * **Methods:** register(), login(), updateProfile().
   * **Purpose:** Manages user profiles and access to the system.
2. **Service Class:**
   * **Attributes:** serviceID, name, description, price, category, availability.
   * **Methods:** addService(), updateService(), removeService().
   * **Purpose:** Represents services offered by providers and their availability.
3. **Booking Class:**
   * **Attributes:** bookingID, customerID, serviceID, providerID, status, date.
   * **Methods:** createBooking(), updateStatus(), cancelBooking().
   * **Purpose:** Tracks and manages booking information.
4. **Payment Class:**
   * **Attributes:** paymentID, bookingID, amount, status, paymentDate.
   * **Methods:** processPayment(), refund().
   * **Purpose:** Handles payment transactions securely.
5. **Feedback Class:**
   * **Attributes:** feedbackID, customerID, providerID, rating, comments, date.
   * **Methods:** submitFeedback(), viewFeedback().
   * **Purpose:** Allows customers to provide reviews and ratings for services.

**Visual Suggestion:**

* Use bullet points with icons representing each class for clarity.

### Slide 4: Relationships Between Classes

**Title:** Relationships Between Classes

**Content:**

**Key Relationships:**

* **User ↔ Booking:**
  + A **user** can either create a booking (customer) or accept a booking (provider).
* **Booking ↔ Service:**
  + Each booking is linked to a specific service.
* **Booking → Payment:**
  + A payment is tied to a booking and processed upon confirmation.
* **User ↔ Feedback:**
  + Feedback connects customers and providers. Customers leave reviews based on their service experience.
* **Admin ↔ User and Service:**
  + The admin monitors, edits, and resolves issues related to users and services.

**Diagram Tips:**

* Use arrows to depict associations, aggregations, or dependencies.
* Annotate each relationship briefly to clarify its purpose.

**Additional Content:**

* Explain **relationship types** like:
  + **Association:** Straightforward connection between two classes.
  + **Aggregation:** Indicates a "whole-part" relationship.
  + **Composition:** Strong ownership where the part cannot exist without the whole.

### Slide 5: Class Diagram Representation

**Title:** Visual Class Diagram

**Content:**

* This slide displays the **actual class diagram** for your system.
* Include the classes (**User, Service, Booking, Payment, Feedback**) and show their:
  + **Attributes:** Listed inside the class boxes.
  + **Methods:** Placed below the attributes.
  + **Relationships:** Connect the classes with arrows.

**Diagram Suggestions:**

* Use UML notation with proper labeling.
* Clearly differentiate relationships like **aggregation**, **association**, and **composition**.
* Highlight primary classes (e.g., User, Service) for better visibility.

**Summary**

**Title:** Summary of Class Diagram

**Content:**

**Key Takeaways:**

* The class diagram **defines the system’s structure**, including core entities, their attributes, and methods.
* **Relationships** between entities help to understand system interactions.
* Serves as a **blueprint** for developers during implementation.
* Provides **clarity** on data flow and system behavior.

