**OrderBuzz Requirement Document**

**Project Overview:** OrderBuzz is a subscription-based order notification system designed for restaurant owners to efficiently manage and notify customers when their orders are ready for pickup. The system will be deployed on the cloud, allowing restaurant owners to access the platform from anywhere. The application consists of a backend built with Flask for handling server-side logic and a frontend developed using React.js for the user interface.

**Key Features:**

1. User Registration: Restaurant owners can register for an account to access the OrderBuzz platform.
2. Order Management: Restaurant owners can manually enter customer orders into the system, including customer name, phone number, and order details.
3. Order Notification: Once an order is ready for pickup, the system automatically sends an SMS notification to the customer's phone number.
4. Multi-Tenant Architecture: Each restaurant owner has their own set of customers and orders, ensuring data segregation and privacy.
5. Subscription-Based Model: The platform operates on a subscription-based model, where restaurant owners subscribe to the service to access its features.

**Database Design:**

* **Owners Table:**
  + **owner\_id** (INT, Primary Key): Unique identifier for each restaurant owner.
  + **email** (VARCHAR(120)): Email address of the restaurant owner.
  + **password** (VARCHAR(60)): Password for the owner's account.
* **Restaurants Table:**
  + **restaurant\_id** (INT, Primary Key): Unique identifier for each restaurant.
  + **owner\_id** (INT, Foreign Key): Foreign key referencing the owner who owns the restaurant.
  + **name** (VARCHAR(255)): Name of the restaurant.
* **Customers Table:**
  + **customer\_id** (INT, Primary Key): Unique identifier for each customer.
  + **restaurant\_id** (INT, Foreign Key): Foreign key referencing the restaurant where the customer placed the order.
  + **name** (VARCHAR(255)): Name of the customer.
  + **phone\_number** (VARCHAR(20)): Phone number of the customer.
* **Orders Table:**
  + **order\_id** (INT): Unique identifier for each order.
  + **restaurant\_id** (INT): Foreign key referencing the restaurant where the order was placed.
  + **owner\_id** (INT): Foreign key referencing the owner who owns the restaurant.
  + **customer\_name** (VARCHAR(255)): Name of the customer who placed the order.
  + **phone\_number** (VARCHAR(20)): Phone number of the customer who placed the order.
  + **order\_status** (VARCHAR(20)): Current status of the order (e.g., "Pending", "Ready").
  + **other\_order\_details** (TEXT): Additional details about the order.
  + PRIMARY KEY (order\_id, restaurant\_id),
  + FOREIGN KEY (restaurant\_id) REFERENCES Restaurants(restaurant\_id),
  + FOREIGN KEY (owner\_id) REFERENCES Owners(owner\_id)

**System Architecture:**

* Backend: Implemented using Flask, a lightweight Python web framework, for handling server-side logic, API endpoints, and database interactions.
* Frontend: Developed using React.js, a JavaScript library for building user interfaces, to create an interactive and responsive user experience.
* Database: Utilizing a relational database management system (e.g., SQLite, PostgreSQL) to store and manage data, ensuring data integrity and scalability.
* Cloud Deployment: The application will be deployed on cloud platforms (e.g., AWS, Azure) to provide accessibility and scalability for restaurant owners.

**Next Steps:**

1. Design and implement the backend APIs for user registration, order management, and order notification.
2. Develop the frontend components for user registration, order entry, and dashboard display.
3. Integrate the backend and frontend components to create a seamless user experience.
4. Test the application to ensure functionality, performance, and security.
5. Deploy the application on cloud infrastructure for production use by restaurant owners.

**Conclusion:** OrderBuzz aims to streamline order management and notification processes for restaurant owners, enhancing customer satisfaction and operational efficiency. By providing a user-friendly and scalable platform, OrderBuzz empowers restaurant owners to deliver timely and personalized service to their customers.