# Hotel Room Booking System via WhatsApp Bot: Workflow and Development Structure

## 1. Project Overview

This system integrates a WhatsApp bot with a web application to enable seamless hotel room booking. New users are guided by the bot to a dynamic web application link where their name and phone number are pre-filled. Returning users can view, modify, or cancel existing bookings directly through the bot. The backend manages booking logic, while the frontend web application collects additional details for the booking. The system uses the Meta WhatsApp Business API for communication and Google Cloud Platform (GCP) for deployment.

## 2. Key Features

### Customer Features

- New User Workflow: Book rooms via a WhatsApp bot-generated dynamic link to the web application.  
- Returning User Workflow: View, modify, or cancel bookings directly through the bot.  
- Automated Notifications: Booking confirmations, reminders, follow-ups, and feedback requests sent via WhatsApp.  
- Mobile-Friendly: The web application is designed with a mobile-first approach for ease of use.

### Admin Features

- Booking Management: Admins can view, modify, or cancel bookings from the admin panel.  
- Manual Payment Handling: Record offline payments in the admin interface.  
- Restricted Access: Admin features are accessible only to authorized users.  
- Analytics and Reports: View trends in bookings, revenue, and room availability.

### Integration Features

- Meta WhatsApp Business API: For messaging and dynamic interactions.  
- Dynamic Web Application Link: Pre-fills user details (name and phone) extracted from WhatsApp payload.  
- Google Cloud Platform: Hosts backend, frontend, and database for a scalable, secure deployment.

## 3. System Architecture

### Frontend (Web Application)

- Built with Next.js for a mobile-first, responsive user experience.  
- Handles the booking workflow, collecting additional details like room type, dates, and number of guests.  
- Dynamic data binding to prefill user details (name and phone number) extracted from WhatsApp.  
- Interfaces with the backend to fetch room availability and submit bookings.

### Backend (Node.js)

- Built with Node.js and Express to handle API requests.  
- Manages the booking workflow, user data, and room availability.  
- Sends messages and notifications through the Meta WhatsApp Business API.  
- Validates booking data and handles reminders, follow-ups, and feedback requests.

### Database (SQLite3)

- Users Table: Stores user details (name, phone number).  
- Bookings Table: Records booking information, including room type, dates, and guest count.  
- Rooms Table: Tracks room types, availability, and pricing.  
- Hosted on Google Cloud Persistent Disk for data persistence and scalability.

## 4. Workflow

### WhatsApp Bot Interaction

1. User Sends 'Hi':  
 - New users see options to book a room or contact the hotel.  
 - Returning users can view, modify, or cancel their bookings.  
  
2. Booking Link Generation:  
 - For new users, the bot generates a link to the web application with prefilled name and phone number.  
 - Returning users can interact directly with the bot for existing bookings.

### Web Application Workflow

1. Prefilled User Data:  
 - The web application extracts user details from query parameters or secure tokens.  
  
2. Collect Booking Details:  
 - Room type, check-in and check-out dates, number of guests, and special requests are collected.  
  
3. Confirmation:  
 - After successful booking, the user sees a confirmation page, and a WhatsApp message is sent.

### Admin Panel Workflow

1. View and Manage Bookings:  
 - Admins can search, view, and update bookings.  
  
2. Room Inventory Management:  
 - Admins can add, update, or remove room types and adjust availability.  
  
3. Payment Handling:  
 - Offline payments can be manually marked as paid.

## 5. API Endpoints

- POST /api/bookings: Save new booking.  
- GET /api/bookings/:id: Fetch booking details.  
- PATCH /api/bookings/:id: Modify booking.  
- DELETE /api/bookings/:id: Cancel booking.  
- GET /api/rooms: Fetch room availability.  
- POST /api/admin/login: Admin login for restricted access.

## 6. Database Schema

Users Table:  
```sql  
CREATE TABLE users (  
 id INTEGER PRIMARY KEY AUTOINCREMENT,  
 name TEXT NOT NULL,  
 phone TEXT NOT NULL UNIQUE  
);  
```  
  
Bookings Table:  
```sql  
CREATE TABLE bookings (  
 id INTEGER PRIMARY KEY AUTOINCREMENT,  
 user\_id INTEGER NOT NULL,  
 room\_type TEXT NOT NULL,  
 check\_in\_date TEXT NOT NULL,  
 check\_out\_date TEXT NOT NULL,  
 guest\_count INTEGER NOT NULL,  
 status TEXT DEFAULT 'confirmed',  
 FOREIGN KEY (user\_id) REFERENCES users (id)  
);  
```  
  
Rooms Table:  
```sql  
CREATE TABLE rooms (  
 id INTEGER PRIMARY KEY AUTOINCREMENT,  
 type TEXT NOT NULL,  
 price REAL NOT NULL,  
 availability INTEGER NOT NULL  
);  
```

## 7. Deployment Plan

### Frontend Deployment

- Deploy the Next.js frontend on GCP Cloud Run.  
- Use Docker for containerization and scalability.  
- Ensure mobile responsiveness for a seamless user experience.

### Backend Deployment

- Deploy the Node.js backend on GCP Cloud Run.  
- Connect the backend to Meta WhatsApp Business API for messaging.  
- Host the SQLite3 database on Google Cloud Persistent Disk for persistent data storage.

### Meta API Integration

- Configure Meta API for sending and receiving WhatsApp messages.  
- Set up webhooks for real-time interaction with user messages.  
- Register message templates for booking confirmations, reminders, and feedback requests.

## 8. Future Enhancements

- Online Payment Integration: Add payment gateways for secure transactions.  
- Restaurant Booking: Enable users to reserve tables for their stay.  
- Loyalty Programs: Offer rewards for frequent guests.  
- AI Recommendations: Suggest room types or packages based on user preferences.