Software Requirements Specification: BookForMe

Muhammad Ahmad Humayun Hanif, Jazib Waqas, Taha Hunaid Ali, Muhammad Taqi

November 5, 2025

Contents

1 Introduction

1.1 Purpose

This document defines the requirements for the **BookForMe** platform. The purpose of BookForMe is to address the significant inefficiencies and fragmentation inherent in booking informal, local services (such as sports courts, salons and gaming zones) within Karachi. The system aims to replace the current reliance on unstructured communication channels and manual record-keeping with a centralized, intelligent, and reliable platform.

It provides a streamlined booking interface for end-users, while also functioning as a social community hub for players and users. Simultaneously, it offers vendors an autonomous AI assistant to manage their bookings and availability across multiple channels, thereby reducing operational friction and potential revenue loss.

1.2 Scope

The BookForMe system comprises three primary components operating cohesively:

1. **User-Facing Web Application:** A responsive web platform enabling customers to discover vendors, view verified real-time service availability, and securely place bookings. This component also includes a comprehensive social layer, allowing users to create profiles, manage friends, participate in community forums, and join ranked/casual matches.
2. **Vendor Dashboard:** A dedicated web interface for service providers to register, manage their business profile (services, pricing, operating hours), view their consolidated booking calendar, and crucially, integrate their existing communication and tracking tools.
3. **AI Agent Backend:** The core intelligent engine of the platform. This component autonomously manages booking requests received via WhatsApp, facilitates payment confirmation, synchronizes availability data from linked Google Sheets, and maintains the central database (Firestore) as the definitive source of truth for service availability. It handles bilingual (Roman Urdu/English) natural language understanding, stateful conversation management, concurrency control for bookings, and multi-channel data synchronization.

2 Functional Requirements

This section describes each function/feature provided by our system. These functions are logically grouped into modules based on purpose and users.

2.1 Module: User (Booking & Discovery)

- `FR-CUST-01`: The system shall allow users to browse and search registered vendors based on service category (e.g., Futsal, Salon) and location area.

- `FR-CUST-02`: The system shall provide filtering options (e.g., price, rating, amenities).
- `FR-CUST-03`: The system shall display vendor profiles including name, description, location, services offered, pricing, user reviews, and ratings.
- `FR-CUST-04`: The system shall display any discounts or promotions offered by a vendor.
- `FR-CUST-05`: The system shall present users with a calendar interface displaying real-time availability slots.
- `FR-CUST-06`: The system shall allow users to select an available time slot and add it to a cart.
- `FR-CUST-07`: The system shall allow users to review their cart and confirm (checkout) their booking.
- `FR-CUST-08`: The system shall confirm successful bookings to the user via the web interface and an alerts/notifications page.

2.2 Module: User (Interaction & Feedback)

- `FR-CUST-09`: The system shall allow users to leave a star rating and a text review for a vendor after a completed booking.
- `FR-CUST-10 (Conversational Search Agent)`: The system shall provide an on-site chatbot that functions as an intelligent search assistant. This bot shall be capable of parsing user queries and performing advanced, cross-category filtered searches against the platform's database.
- `FR-CUST-11 (Notifications)`: The system shall provide an alerts/notifications page for booking updates, friend requests, and DMs.

2.3 Module: User (Social & Community)

- `FR-SOC-01`: The system shall provide a user registration and login mechanism.
- `FR-SOC-02`: The system shall allow users to create and manage a public profile (e.g., username, picture, match history).
- `FR-SOC-03`: The system shall allow users to search for other users on the platform.
- `FR-SOC-04`: The system shall provide a "friend" system, allowing users to send, accept, or deny friend requests.
- `FR-SOC-05`: The system shall allow users to send and receive private Direct Messages (DMs) to/from users on their friends list.

- FR-SOC-06: The system shall provide a community forum where users can create, view, and comment on public posts.

- FR-SOC-07: The system shall provide a mechanism for users to "Report" other users or posts for moderation.

2.4 Module: User (Matchmaking & Ranking)

- FR-MATCH-01: The system shall provide a mechanism for users to queue for a "Casual Match."

- FR-MATCH-02: The system shall provide a mechanism for users to queue for a "Ranked Match."

- FR-MATCH-03: The system shall calculate and update user ratings based on match outcomes.

- FR-MATCH-04: The system shall display a public leaderboard of top-ranked players.

2.5 Module: Vendor

- FR-VEND-01: The system shall provide a secure registration and login mechanism for vendors.

- FR-VEND-02: Vendors shall be able to create and manage their business profile (services, pricing, hours).

- FR-VEND-03: Vendors shall be able to view a consolidated calendar displaying all bookings from all sources (Web, WhatsApp, Sheets).

- FR-VEND-04: The system shall provide an interface for vendors to manually confirm or cancel bookings.

- FR-VEND-05: The system shall provide an interface for vendors to search, update, and delete bookings.

- FR-VEND-06: The system shall provide a secure interface for vendors to authenticate and link their official WhatsApp Business account.

- FR-VEND-07: The system shall provide a secure interface for vendors to authenticate and link their designated Google Sheet.

- FR-VEND-08 : The Vendor Dashboard shall provide an interface for the vendor to manually confirm a payment has been received or to reject a pending payment.

2.6 Module: AI Agent (System)

- FR-AGENT-01 : The system shall receive incoming message from the linked vendor's WhatsApp API via a configured webhook.
- FR-AGENT-02 : The system shall be able to send outgoing messages (replies, confirmations) via the WhatsApp API.
- FR-AGENT-03 : The system shall process incoming WhatsApp messages (Roman Urdu/English) to extract intents and entities.
- FR-AGENT-04 : The system shall maintain conversational context (state) for ongoing WhatsApp interactions.
- FR-AGENT-05: Based on NLU data and availability, the system shall determine appropriate actions (e.g., confirm, offer alternatives).
- FR-AGENT-06: The system shall implement mechanisms (e.g., Firestore Transactions) to prevent conflicting bookings for the same time slot originating simultaneously from any channel (Web, WhatsApp, Sheets).
- FR-AGENT-07: The system shall parse the sheet to identify manual bookings and update the central database.
- FR-AGENT-08: The system shall update the central database upon successful booking confirmation from any channel.
- FR-AGENT-09: The system (AI Agent) shall, after a booking is verbally agreed upon, prompt the user on WhatsApp to submit a payment screenshot to confirm the booking (which will be marked as "Pending Approval").
- FR-AGENT-10 (Payment Facilitation): The system shall attach the received screenshot to the booking record and notify the vendor via the Vendor Dashboard.
- FR-AGENT-11 (Payment Confirmation): Upon vendor approval (from FR-VEND-08), the system (AI Agent) shall update the booking status to "Confirmed" and send a final confirmation message to the user on WhatsApp.

2.7 Module: Stretch Goals (Future Scope)

- FR-STRETCH-01: The system may provide a payment gateway for users to pay for bookings online.
- FR-STRETCH-02: The AI Agent may be enhanced to transcribe and understand WhatsApp voice notes.

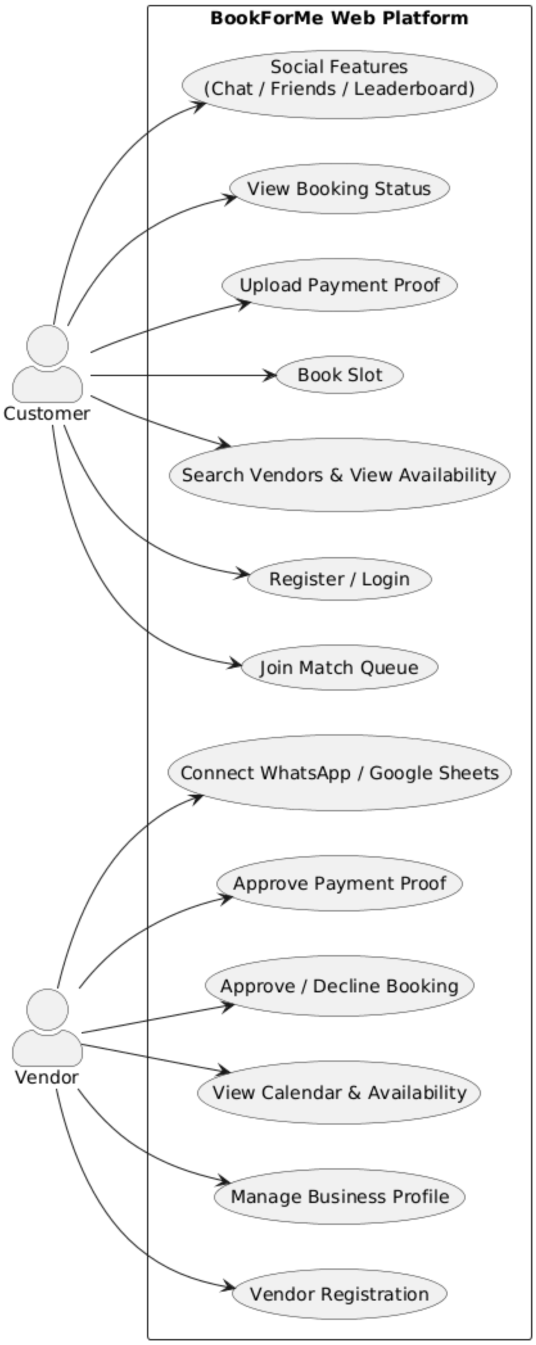
- FR-STRETCH-03: The system may implement a recommendation algorithm to suggest vendors or activities to users.

- FR-STRETCH-04: The system may use an OCR/image detection service to automatically scan and validate payment screenshots to confirm transactions.

- FR-STRETCH-05: The system may be enhanced to provide a user-facing "agent mode," allowing users to provide natural language booking goals (e.g., "Find me a futsal court for Friday night") and have the agent autonomously search, present options, and finalize the booking.

2.8 Use Case Diagrams

The following diagrams illustrate the core actors and interactions in the BookForMe system, covering both the customer-facing web platform and the AI receptionist that handles WhatsApp-based booking flows.



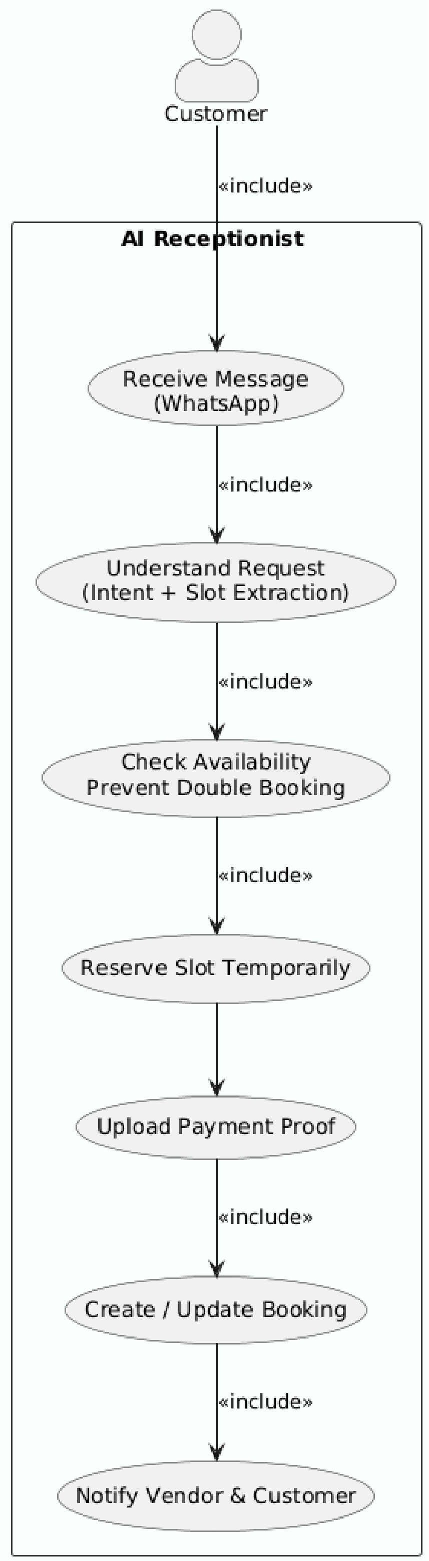


Figure 1: Customer and Vendor Use Case Diagram

Figure 2: AI Receptionist Use Case Diagram

3 Non-functional Requirements

This section mentions specific non-functional requirements for the system.

3.1 Performance

- `NFR-PERF-01 (Response Time - Web)`: The User Website's key pages (e.g., vendor search, calendars) shall load completely within **30 seconds** on an average stable connection.
- `NFR-PERF-02`: The AI Agent's NLU processing for a standard WhatsApp message shall complete within **60 seconds**.
- `NFR-PERF-03`: The total time from receiving a simple WhatsApp availability query to sending the initial response via the WhatsApp API shall not exceed **60 seconds**.
- `NFR-PERF-04`: The WhatsApp webhook endpoint shall be capable of processing at least **4 simultaneous incoming webhook requests** without failure.

3.2 Security

- `NFR-SEC-01`: All user and vendor access shall be protected via secure password authentication.
- `NFR-SEC-02`: All external API keys (WhatsApp, Google Sheets, Gemini) shall be stored securely as environment variables or secrets, and not hardcoded in the application source.
- `NFR-SEC-03`: The WhatsApp webhook endpoint shall verify the integrity and authenticity of all incoming requests from Meta using the Verify Token.
- `NFR-SEC-04`: All data transmission between client browsers, the backend server, and external APIs shall use HTTPS/TLS encryption.

3.3 Reliability

- `NFR-REL-01`: The system shall aim for **99.0% uptime** during peak operating hours.
- `NFR-REL-02`: The AI Agent shall handle external API errors (e.g., NLU timeouts) and log them appropriately without crashing the main service.
- `NFR-REL-03`: The Google Sheet synchronization process shall be fault-tolerant, logging API and parsing errors.

3.4 Scalability

- `NFR-SCAL-01`: The backend application shall be designed to support scaling (e.g., using serverless functions) to handle increased load.

- `NFR-SCAL-02`: The system's Firestore database structure shall be designed to support baseline functionality for at least **100 concurrent users**.
- `NFR-SCAL-03`: The system architecture shall support onboarding and baseline operation for at least **10 vendors** without significant performance degradation.

3.5 Usability

- `NFR-USAB-01 (User Website)`: The User Website shall feature a clean, intuitive, and mobile-first design. A new user shall be able to complete a booking in fewer than **8 primary steps** (e.g., Search -> Select Vendor -> View Calendar -> Select Slot -> Cart -> Confirm).
- `NFR-USAB-02`: The Vendor Dashboard shall provide clear interfaces, with all key functions (e.g., View Calendar, Connect APIs) accessible within **6 clicks** from the main dashboard.
- `NFR-USAB-03`: The AI Agent's WhatsApp responses shall be clear, concise, and contextually appropriate in Roman Urdu/English.