# Functional Requirements Document Airbnb Clone project

## 1. Introduction

### 1.1. Purpose

The purpose of this document is to define the functional requirements for the development of an Airbnb clone software application. It aims to provide a clear and detailed description of the functionalities and features that the system must possess to meet the needs of its users.

### 1.2 Project Summary

This project involves the development of a web and mobile application that replicates the core functionalities of Airbnb. The application will allow users to list, discover, and book accommodations worldwide, including unique experiences and adventures.

### 1.3 Background

Airbnb has revolutionised the travel industry by offering a platform for individuals to rent out their properties or spare rooms to travellers. This project aims to create a similar platform that can cater to a specific niche or demographic, enhancing the user experience with additional features and improved usability.

### 1.4 Scope

The scope of this project includes the development, testing, and deployment of the Airbnb clone application. The application will feature user registration and authentication, property listing, booking management, and reviews. The system will be designed to be scalable and secure, with support for both web and mobile platforms.

### 1.4 References

- Airbnb official website for reference on functionality.
- UX/UI design guidelines.

# 2. Overall Description

## 2.1 Product Perspective

The Airbnb clone will be a new, independent application that replicates the core features of Airbnb. It will be built using modern web and mobile technologies to provide a seamless user experience.

## 2.2 Product Functions

- User registration and authentication.
- Property listing and management.
- Search and filter functionality.
- Booking processing.
- Reviews and ratings.
- Messaging system.
- Admin management.

## 2.3 User Classes and Characteristics

- Guests: Users who book accommodations.
- Hosts: Users who list properties.
- Administrators: Users who manage the platform.

## 2.4 Operating Environment

- Web browsers (Chrome, Firefox, Safari, Edge).

## 2.5 Design and Implementation Constraints

- Compliance with data protection regulations.

## 2.6 Assumptions and Dependencies

- Reliable internet connection.

# 3. Specific Requirements

## 3.1 User Registration and Authentication

3.1.1 User Registration

- Users should be able to sign up using email.
- Required fields: Name, Email, Password, Phone Number.

*Validation Rules:*

- Email must be valid and unique.
- Passwords must be at least 8 characters long, contain at least one number, one uppercase letter, and one special character.

*Acceptance* Criteria

- Users can successfully register and log in using email

3.1.2 User Login

- Users should be able to log in using email.
- Password recovery option.

*Validation Rules:*

- Email must be in a valid format.
- Passwords must be at least 8 characters long, contain at least one number, one uppercase letter, and one special character.
- User account must be active and not blocked

*Acceptance* Criteria

- User should be successfully logged in, and have access to his profile, bookings, ..

3.1.3 Reset password

- Users can reset their password if forgotten
- The user must enter an email which already registered in the system
- The system will send an otp the entered email
- The user must enter a correct otp and valid password
- *Validation Rules*
  - Passwords must be at least 8 characters long, contain at least one number, one uppercase letter, and one special character.
- *Acceptance Criteria*
  - Users can reset their password and log in with the new password.

## 3.2 User Profile Management

3.2.1 View Profile Information

- Users can view their profile section
- *Acceptance Criteria*
  - User should see their profile details accurately displayed upon navigating to the profile section

3.2.2 Edit profile information

- Users should be able to update their profile information.
- Users should be able to upload a profile picture.
- *Acceptance Criteria*
  - User should be able to successfully save edited profile information without encountering validation errors
  - Edited profile information should be updated and displayed correctly on subsequent views

3.2.3 change password

- Users can change their account password.
- *Validation Rules*
  - New Passwords must be at least 8 characters long, contain at least one number, one uppercase letter, and one special character.
- *Acceptance Criteria*
  - User should be able to change their password successfully without any validation errors
  - New password should be immediately effective and allow user to log in using the new credentials

## 3.3 Property Listings

3.3.1 listing management

- Users (hosts) should be able to list new units with the needed details: Title, Description, Address, Photos, Price, Availability, Amenities.
- Hosts should be able to manage their listed properties (edit, delete, update availability).

**- Validation Rules:**

- Property title must be between 5 and 100 characters.
- Description must be at least 20 characters.
- At least one photo must be uploaded.

- Pricing must be a positive number.
- Availability dates must not overlap for the same property.

- **Acceptance criteria**
  - Users (Hosts) can create, update, and delete listings with all required fields.
  - Listings appear in search results according to filters and keywords.
  - Photos and descriptions are displayed correctly on listing pages.

3.3.2 View Property Listings

- Users (Guests) should be able to view property listings with details: Title, Description, Address, Photos, Price, Availability, Amenities, Host Information.

## 3.4 Search and Filter

3.4.1 Search and filter Properties

- Guests should be able to search for properties, dates, and number of guests.
- Search results should display property details and prices.
- Guests should be able to filter search results based on price range, property type, amenities, ratings.

- **Acceptance criteria**
  - Search results match the query and applied filters.
  - Users can sort results by relevance, price, or rating.

## 3.5 Booking System

3.5.1 Check Availability

- Guests should be able to check property availability based on selected dates.

3.5.2 Book Property

- Guests should be able to request to book available properties.
- Booking details: Property, Dates, Number of Guests, Total Price.

- **Validation Rules:**
  - Check-in date must be before check-out date.
  - Number of guests must not exceed the property's maximum capacity.
  - Booking requests cannot be made for dates that are already booked.
- **Acceptance criteria**
  - Users can request bookings and see their request status (pending, accepted, declined).
  - Hosts receive notifications for new booking requests.

3.5.3 Booking Confirmation

- Guests should receive a booking confirmation via ***email and*** in-app notification.

- **Acceptance criteria**
    - Booking confirmations and cancellations update availability and notify both parties.

3.5.4 Booking Management

- Guests should be able to view and manage their bookings.
- Hosts should be able to view and manage bookings for their properties.

## 3.6 Reviews and Ratings

3.6.1 Leave a Review

- Guests should be able to leave a review and rating for properties they have stayed in.
- Hosts should be able to leave a review and rating for guests.

- **Validation Rules:**
    - Reviews must be between 20 and 1000 characters.
    - Ratings must be between 1 and 5 stars.
    - ***Reviews must be submitted within a certain period after the stay.***
- **Acceptance criteria**
    - Guests can submit reviews and ratings.

3.6.2 View Reviews

- Reviews and ratings should be displayed on property listings.

- **Acceptance criteria**
    - Reviews are displayed on property listings.

## 3.7 Messaging System

***3.7.1 Send Messages***

- ***Guests and hosts should be able to send and receive messages within the app.***
- Notifications for new messages.

3.7.2 View Message History

- Users should be able to view their message history.

## 3.8 Admin Panel

### 3.8.1 User Management

- Admins should be able to manage users (***view, edit, delete***).

### 3.8.2 Property Management

- Admins should be able to manage properties (***view, edit, delete***).

### 3.8.3 Booking Management

- Admins should be able to manage bookings (view, edit, cancel).

### 3.8.4 Review and Rating Management

- Admins should be able to moderate reviews and ratings.

### 3.8.5 Reports and statistics

Admins should be able to generate reports, and view system statistics.

## 4. Non-Functional Requirements

### 4.1 Performance Requirements

- The system should handle up to 10,000 concurrent users.
- Page load time should be less than 2 seconds.

### 4.2 Security Requirements

- Data encryption for sensitive information.
- Secure user authentication and authorization.
- Compliance with GDPR.

### 4.3 Usability Requirements

- Intuitive and user-friendly interface.
- Responsive design for web and mobile.

### 4.4 Reliability Requirements

- 99.9% uptime.
- Regular backups of data.

### 4.5 Scalability Requirements

- The system should be scalable to support future growth and additional features.

## 5. Appendices

### 5.1 Assumptions

- Users have access to a stable internet connection.
- Third-party services (payment gateways) are reliable.

### 5.2 Glossary

- Host: A user who lists their property for booking.
- Guest: A user who books a property.

## Description

The template provides explanations and sample documentation for all of the relevant sections including but not limited to:

1. Introduction
   - 1.1 Purpose of Document
   - 1.2 Project Summary
   - 1.3 Background
   - 1.4 Project Scope
   - 1.5 System Purpose
   - 1.5.1 Users
   - 1.5.2 Location
   - 1.5.3 Responsibilities
   - 1.5.4 Need
   - 1.6 Overview of Document
2. Functional Objectives
   - 2.1 High Priority
   - 2.2 Medium Priority
   - 2.3 Low Priority
3. Non-Functional Objectives
   - 3.1 Reliability
   - 3.2 Usability
   - 3.3 Performance
   - 3.4 Security
   - 3.5 Supportability
   - 3.6 Online user Documentation and Help
   - 3.7 Purchased Components
   - 3.8 Interfaces
4. The Context Model
   - 4.1 Goal Statement
   - 4.2 Context Diagram
   - 4.3 System Externals
5. The Use Case Model
   - 5.1 System Use Case Diagram
   - 5.2 Use Case Descriptions (for selected cases)
6. User Stories
7. Appendix
   - Glossary