**Functional Required Document: Homeowner Onboarding & Villa Registration**

**Project Name:** Wezo.ae   
**Document Version:** 1.0   
**Date:** 11-Aug-25   
**Prepared by:** skyve.ae

**1. Introduction**

* **Purpose:** This document details the functional requirements and user flow for the Homeowner Onboarding and Villa Registration feature of the wezo.ae application. It serves as a guide for development and a reference for the client, outlining how a homeowner will register and list their property.
* **Scope:** This document focuses exclusively on the user journey from initial sign-up to the point where their villa listing is live and ready for bookings. It does not cover features like booking management, guest reviews, or payment processing.
* **Reference:** The workflow for homeowners to onboard and register their villas has been designed to mirror the established process on Booking.com. This ensures a familiar and efficient user experience.

**2. High-Level Overview**

The Homeowner Onboarding feature allows individuals to create an account and list a property for rent. The process is broken down into a multi-step, guided workflow to ensure all necessary information is collected accurately. The primary goal is to empower homeowners to manage their own listings with minimal friction.

**3. User Roles**

* **Homeowner:** A user who owns a property (in this case, a villa) and wants to list it on the platform to generate rental income. They need a simple, intuitive process to get their property live.

**4. Functional Requirements & User Flow**

This section details the step-by-step process a homeowner follows to list a villa. The wizard is designed to collect all the necessary information, from basic property details to complex pricing and policy rules.

**Step 1: Property Type & Location**

* **Description:** The homeowner begins by identifying the property type and its physical location.
* **User Interface (UI):**
  + A dropdown to select "Villa" as the property type.
  + An address field with an **autocomplete feature** powered by a mapping service (e.g., Google Maps API).
  + A map display that visually confirms the entered address.
* **System Logic:**
  + Validate the address and convert it into geocoded coordinates (latitude, longitude) for map display and search functionality.

**Step 2: Villa Details, Amenities, and Host Information**

* **Description:** This step gathers all the core information about the villa and the host.
* **User Interface (UI):**
  + **Property Name** and **Description** fields.
  + Inputs for number of bedrooms, bathrooms, and guest capacity.
  + **Services and Amenities:** A comprehensive list of checkboxes for amenities (e.g., Wi-Fi, Pool, Parking) and services (e.g., Airport Shuttle, Breakfast included, Laundry service).
  + **Languages Spoken:** A multi-select dropdown for the host to specify languages they can communicate in (e.g., English, Spanish, French).
* **System Logic:**
  + All core fields are mandatory.
  + Store all selected amenities and languages in the database associated with the property.

**Step 3: Pricing & Rate Policies**

* **Description:** This is where the homeowner sets up their pricing structure, including different rate types.
* **User Interface (UI):**
  + **Base Nightly Rate:** An input field to set the default price per night.
  + **Weekly Rate:** An option to set a discounted weekly rate. The system should calculate and display the total weekly cost to the homeowner.
  + **Non-Refundable Rate:** A checkbox or toggle to enable a non-refundable rate option. This rate is typically a percentage discount off the base rate (e.g., a 10% discount).
* **System Logic:**
  + Validate that all rates are positive numbers.
  + The system must be able to calculate and apply discounts for non-refundable and weekly rates automatically.

**Step 4: Promotions & Availability**

* **Description:** This section allows the homeowner to create special offers and manage their calendar.
* **User Interface (UI):**
  + **Promotions:** A tool to create limited-time promotions (e.g., "Early bird discount: 20% off for bookings made 30 days in advance"). This should have fields for the discount percentage, promotion name, and active dates.
  + **Availability Calendar:** An interactive calendar where the homeowner can manually block off specific dates or date ranges when the villa is not available for booking.
* **System Logic:**
  + Apply promotions and calculate discounted prices on the front end for guests.
  + Store blocked dates in the database and ensure the booking system cannot process reservations on these dates.

**Step 5: House Rules & Cancellation Policy**

* **Description:** The homeowner defines the rules for their property and the rules for canceling a booking.
* **User Interface (UI):**
  + **House Rules:** A text area for the homeowner to write specific rules for guests (e.g., "No smoking," "No parties," "Quiet hours after 10 PM").
  + **Cancellation Policy:** A dropdown to select a pre-defined policy template (e.g., "Flexible," "Moderate," "Strict"). Each policy should have a clear description of its terms.
* **System Logic:**
  + Store the selected cancellation policy and house rules in the database.
  + The booking system must adhere to these rules when processing cancellations and calculating refunds.

**Step 6: Upload Photos & Final Review**

* **Description:** The final step involves uploading photos and a last review before publishing the listing.
* **User Interface (UI):**
  + A drag-and-drop file upload interface for images.
  + The ability to set a primary display image.
  + A final summary page that presents all the information entered across all steps.
* **System Logic:**
  + Resize and compress images for optimal performance.
  + Upon clicking "Publish Listing," the system changes the property status to published and makes it visible to guests.

**6. Assumptions & Constraints**

* The homeowner has valid contact information (email address).
* Images uploaded by the homeowner do not contain explicit or inappropriate content (this may require an image moderation service in the future).
* The system assumes a single currency for pricing initially, with potential for multi-currency support in later phases.
* The mapping service (e.g., Google Maps API) will be integrated and requires an API key.