

Product Requirements Document: Multi-Agent Real Estate Assistant Chatbot

Project Overview

A multi-agentic chatbot system specialized in real estate, with two primary agents: one for visual issue detection and troubleshooting, and another for tenancy-related questions and support. The system incorporates router logic to direct queries to the appropriate agent.

Requirements Specification

Req. #	Requirement Description	User Story	Expected Behavior	Implementation Approach
1. Core System Architecture				
1.1	Multi-agent architecture with router component	As a user, I want my query to be automatically directed to the right specialist agent without having to select one myself.	System should analyze input and determine whether to route to the Issue Detection or Tenancy FAQ agent automatically.	Implement using Agno's multi-agent architecture with "route" mode or ADK's LLM-driven dynamic routing.
1.2	Seamless agent switching	As a user, I want to switch between discussing property issues and tenancy questions in a single conversation.	System should maintain context when switching between agents and provide a cohesive conversation experience.	Leverage Agno's session storage or ADK's state management to maintain conversation context across agents.
1.3	Fallback mechanism for ambiguous queries	As a user, I want the system to ask clarifying questions when my query could be handled by either agent.	System should detect ambiguity and ask targeted questions to determine the appropriate agent.	Implement confidence threshold in router component with pre-defined clarification questions.
2. Issue Detection & Troubleshooting Agent				
2.1	Image upload capability	As a user, I want to upload photos of property issues for analysis.	System should accept and process common image formats (JPEG, PNG, etc.) with reasonable file size limits.	Implement multimodal input handling using Agno's native multimodal capabilities.
2.2	Detection of common property issues	As a user, I want the system to identify problems like water	System should analyze images and highlight visible	Train or use pre-trained computer vision models integrated as

Req. #	Requirement Description	User Story	Expected Behavior	Implementation Approach
		damage, mold, cracks, or electrical issues from my photos.	issues with reasonable accuracy.	tools within the Agno agent framework.
2.3	Context-aware issue analysis	As a user, I want to provide additional context about an issue through text alongside my images.	System should combine image analysis with textual context for more accurate problem identification.	Implement prompt engineering to combine visual analysis results with textual context.
2.4	Detailed troubleshooting guidance	As a user, I want specific recommendations for addressing the identified issues.	System should provide actionable steps or professional referrals based on issue severity.	Create a comprehensive knowledge base of property issues and solutions, integrated with Agno's knowledge features.
2.5	Follow-up question capability	As a user, I want the system to ask clarifying questions to better understand complex property issues.	System should identify when additional information is needed and ask specific, relevant questions.	Utilize Agno's ReasoningTools to implement diagnostic decision trees.
2.6	Severity assessment	As a user, I want to know if an issue requires immediate professional attention or can be addressed DIY.	System should classify issues by severity and recommend appropriate action urgency.	Develop classification system with clear criteria for different severity levels.
3. Tenancy FAQ Agent				
3.1	Comprehensive tenancy knowledge base	As a user, I want accurate answers to a wide range of tenancy-related questions.	System should provide legally accurate information	Create extensive knowledge base using Agno's knowledge

Req. #	Requirement Description	User Story	Expected Behavior	Implementation Approach
			covering common tenancy scenarios.	features with reliable legal resources.
3.2	Location-specific guidance	As a user, I want advice tailored to the tenancy laws in my specific region.	System should adjust responses based on jurisdiction when location is provided.	Implement location detection and maintain region-specific legal databases.
3.3	Legal disclaimer provision	As a user, I want to understand the limitations of the advice provided.	System should clearly indicate it's providing general guidance, not legal advice.	Add appropriate disclaimers to responses involving legal matters.
3.4	Landlord/tenant perspective toggle	As a user, I want responses that consider my role as either a tenant or landlord.	System should adapt responses based on the user's stated role in the tenancy relationship.	Implement role-based response templates with appropriate perspective shifts.
3.5	Document explanation capability	As a user, I want help understanding sections of my lease or tenancy agreement.	System should be able to interpret common lease clauses and explain their implications.	Create a specialized tool for lease analysis with common clause recognition.
4. User Experience & Interface				
4.1	Intuitive chat interface	As a user, I want a simple, easy-to-use interface for interacting with the system.	Interface should be clean, responsive, and clearly indicate when I can upload images.	Develop responsive UI with clear visual cues for multimodal input options.
4.2	Conversation history persistence	As a user, I want to reference previous parts of my conversation.	System should maintain and display conversation history within a session.	Implement session management using Agno's memory capabilities.

Req. #	Requirement Description	User Story	Expected Behavior	Implementation Approach
4.3	Response formatting optimization	As a user, I want information presented in an easily digestible format.	System should use appropriate formatting (lists, bold text, etc.) to enhance readability.	Enable markdown formatting in agent responses (supported by Agno).
4.4	Visual indicators for agent switching	As a user, I want to know which specialized agent is currently responding.	Interface should subtly indicate which agent is active in the conversation.	Implement visual cues in the UI when router switches between agents.
5. Technical Requirements				
5.1	Performance optimization	As a user, I want quick response times, even for image analysis.	System should respond within acceptable timeframes (< 5 seconds for text, < 10 seconds for images).	Leverage Agno's "lightning fast" architecture and optimize image processing pipeline.
5.2	Scalability	As a system administrator, I want the system to handle multiple concurrent users efficiently.	System should maintain performance under load and scale resources as needed.	Implement containerized deployment with auto-scaling capabilities.
5.3	Privacy compliance	As a user, I want assurance that my property images and conversation data are handled securely.	System should implement appropriate data handling practices and clearly communicate privacy policies.	Implement encryption, secure data storage, and privacy controls.
5.4	Integration capabilities	As a business owner, I want to integrate this chatbot with existing	System should provide API endpoints for	Develop a well-documented API layer for external integration.

Req. #	Requirement Description	User Story	Expected Behavior	Implementation Approach
		property management systems.	integration with third-party services.	
6. Monitoring & Improvement				
6.1	Performance monitoring	As a system administrator, I want insights into system performance and usage patterns.	System should log and analyze key metrics like response times, query types, and user satisfaction.	Leverage Agno's monitoring capabilities via agno.com.
6.2	Continuous improvement mechanism	As a product owner, I want the system to improve over time based on user interactions.	System should collect feedback and failed queries to identify improvement opportunities.	Implement feedback collection and analysis pipeline.
6.3	Model retraining pipeline	As a data scientist, I want to periodically update the underlying models with new data.	System should support model versioning and smooth update deployment.	Create model management system with version control.
7. Administrative Functions				
7.1	Knowledge base management	As a content manager, I want to update the knowledge base with new information.	System should provide tools to add, edit, and remove knowledge base entries.	Create an admin interface for knowledge base management.
7.2	Response customization	As a business owner, I want to customize certain responses to align with company policies.	System should allow authorized users to edit response templates.	Implement template management system with version control.
7.3	Usage analytics	As a marketing manager, I want insights	System should provide aggregated	Develop analytics dashboard with

Req. #	Requirement Description	User Story	Expected Behavior	Implementation Approach
		into most common user questions and pain points.	analytics on query types and frequency.	relevant metrics and visualization.

Technical Stack Recommendations

Based on the analysis of Agno and Google ADK, the following approach is recommended:

- Core Framework:** Agno for its specific focus on agent development with built-in support for:
 - Multi-modal capabilities (critical for image analysis)
 - Multi-agent architecture with routing
 - Memory and knowledge integration
 - Reasoning tools
- Integration Points:**
 - Computer vision models for property issue detection
 - Vector database for knowledge management
 - Frontend framework for chat interface
- Deployment:**
 - Containerized deployment for scalability
 - Monitoring via Agno's built-in tools

Implementation Phases

Phase 1: Foundation (4-6 weeks)

- Basic router implementation
- Simple text-only versions of both agents
- Core knowledge base development

Phase 2: Core Functionality (6-8 weeks)

- Image analysis integration
- Enhanced knowledge bases
- Improved routing logic

Phase 3: Refinement (4-6 weeks)

- UI/UX improvements
- Performance optimization
- Advanced context handling

Phase 4: Production Readiness (4 weeks)

- Security hardening
- Scalability testing
- Documentation and training