**VIRTUAL AGENT (DESCRIPTION OF THE APPLICATION)**

Complete summary of the "Virtual Agent" application, integrating the Google Sheets API part

Complete Summary of the "Virtual Agent" Application

1. Registration and Login

Initial Screen:

The user registers with a username and a password.

Complete Registration:

They are asked to enter:

• Full name  
• Mobile phone: This number must support payments.  
• Nickname: It will be their identity within the app.

1. Property Entry

Selection of Property Type:

Available options: House, Land, and Commercial Premise.

Photograph Capture:

Photo 1: Close-up of the property's sign, where the phone number is included.

Artificial Intelligence: An OCR API (such as the Google Cloud Vision API) is used to automatically extract the number from the sign.

Photo 2: General image of the property for visual reference.

Location Capture:

Using the Google Maps API, the application detects and records the exact location of the property (displaying the user's current location).

1. Data Confirmation and Storage

Entry Confirmation:

Once the steps are completed, the app displays the message: "Property entered for analysis"

A unique identification number is generated for the property.

Integration with Google Sheets API:

All the property information is automatically saved in a master Google Sheets spreadsheet, previously set up by the administrator.

Fields in the Master Sheet:

• Username (who entered the property)  
• Nickname  
• Phone  
• Property type  
• Sign's phone number (extracted via OCR)  
• Location  
• Property ID

Thus, each entry from all users is centralized in a single table for easy tracking and analysis.

1. Generation and Management of KML Files

Creation of the KML File:

The application simultaneously exports the property's location in a KML file for Google Maps.

KML Details:

Each location is marked with the property ID as the placemark name.

Marker Color:

• Blue: House  
• Green: Land  
• Yellow: Commercial Premise

Custom Map for the Administrator:

The KML file is synchronized with a custom map in the administrator's Google account, allowing only them to view the distribution of the properties.

1. Technical Considerations and Tools

Mobile Platform:

Development with cross-platform frameworks such as Flutter or React Native, or native development.

Backend and Authentication:

Use of services such as Firebase Authentication to manage registration and login.

Image Processing:

Integration with the Google Cloud Vision API for optical character recognition (OCR) on the sign's photo.

Geolocation and Maps:

Implementation of the Google Maps SDK to capture the location and manage maps.

Integration of Google Sheets API:

Allows each property entry to be automatically added to the master Google Sheets spreadsheet, centralizing all the information.

Implementation Guide:

1. Planning and Design

Requirements Definition:

Review and document all the functionalities we want: registration and login, property entry, image processing with OCR, geolocation, integration with Google Sheets, and KML file generation.

User Flow and Wireframes:

Draw the navigation flows between screens (registration, property entry, confirmation, etc.).

Create sketches or wireframes using tools such as Figma or Adobe XD to visualize the interface and user experience.

Choice of Technology:

Decide whether to use a cross-platform framework (such as Flutter or React Native) or native development for each platform (Android/iOS).

1. Development Environment Setup

Tools Installation:

Install and configure the IDE or code editor (e.g., Visual Studio Code, Android Studio, Xcode, etc.).

Configure the corresponding SDK (Flutter SDK, Node.js for React Native, etc.).

Version Control:

Create a repository in Git (GitHub, GitLab, Bitbucket) for proper version control.

Emulators and Devices:

Set up emulators or use physical devices for testing during development.

1. Implementation of Registration and Login

Initial Screen:

Develop the interface for the user to register with a username and a password.

Complete Registration:

Create the screen where the user enters their full name, phone number (ensure this number can handle payments), and nickname.

Add validations for each field (e.g., phone format, required fields, etc.).

Authentication:

You can integrate services like Firebase Authentication to securely manage registration and login.

1. Development of Property Entry

Selection of Property Type:

Implement a screen where the user can choose between House, Land, and Commercial Premise.

Photograph Capture:

Sign Photo:

Integrate the camera to capture an image focused on the sign.

Use the Google Cloud Vision API (or another OCR API) to automatically extract the phone number from the sign.

General Photo:

Allow the user to capture an image of the entire property for reference.

Location Capture:

Implement the geolocation functionality using the Google Maps SDK or similar APIs to obtain the exact location at the time of capture.

Generation of Property ID:

Design and implement the logic to generate a unique identifier for each property (it can be a sequential number or a code generated using an algorithm).

1. Integration with Google Sheets API

Master Sheet:

Create, in the administrator's Google account, a spreadsheet (Google Sheet) with the following fields:

• Username  
• Nickname  
• Phone  
• Property type  
• Sign's phone number  
• Location  
• Property ID

Setup and Credentials:

Configure access to the Google Sheets API:

• Register the project in Google Cloud Console.  
• Generate the necessary credentials (API key, OAuth2) so that the app can interact with the spreadsheet.

Programming the Integration:

Develop the function in the backend or within the app that, once the property entry is completed, automatically sends and adds the data to the master spreadsheet.

Make sure to handle possible errors (e.g., connectivity issues) and validate that the data is stored correctly.

1. Generation and Management of the KML File

Creation of the KML File:

Develop the functionality to generate or update a KML file that contains the location of each property.

Each marker in the KML should:

• Have the property ID as the name.  
• Use a color coded according to the type:  
  – Blue: House  
  – Green: Land  
  – Yellow: Commercial Premise

Synchronization with the Administrator's Map:

Set up the system so that the KML file is uploaded or synchronized with a custom map in the administrator's Google account.

Only the administrator will have access to this map to view the distribution of the properties.

1. Testing and Debugging

Unit and Integration Testing:

Develop and run tests for each module of the application: registration, data capture, API integration, etc.

Device Testing:

Conduct tests on different devices and emulators to ensure that the app functions correctly under various conditions.

Bug Fixing:

Debug any errors that arise and optimize the application's performance.

1. Deployment and Publishing

Production Environment Setup:

Ensure that the backend, APIs, and integrations are configured for the live environment.

Publishing on Stores:

Prepare and submit the application to the app stores (Google Play Store, Apple App Store), following their guidelines and requirements.

Monitoring and Support:

Set up monitoring systems to detect potential issues and collect user feedback for future improvements.

1. Maintenance and Updates

User Feedback:

Listen to your users and make adjustments and improvements based on their suggestions and experiences.

Regular Updates:

Keep the app updated in terms of security, functionality, and compatibility with new versions of operating systems and APIs.

**APPLICATION VISUALIZATION REFERENCES** ──────────────────────────────