



UNIVERSITY OF PIRAEUS

School of Information and Communication Technologies

Department of Informatics

Thesis

Thesis Title: Τίτλος Διατριβής:	Designing an Android Voice Assistant: A Development and Security Perspective Σχεδιασμός Ενός Φωνητικού Βοηθού για Android Συσκευές: μια Προσέγγιση Ανάπτυξης και Ασφάλειας
Student's name-surname:	IOANNIS AVDOULAS
Father's name:	SPYRIDON
Student's ID No:	Π18001
Supervisor:	ΕΦΘΗΜΙΟΣ ΑΛΕΠΙΣ

July 2024/ Ιούλιος 2024

1 COPYRIGHT

Copying, storing and distribution of this work is prohibited, ex in whole or in part, for a commercial purpose. Reprinting is permitted, storage and distribution for non-profit, educational or research purposes nature, provided the source of origin is indicated and the present message. The opinions and conclusions contained in this document express solely of the author and do not represent his official positions University of Piraeus. As the author of this paper, I declare that this paper does not constitute plagiarized and does not contain material from unlisted sources.

TABLE OF CONTENTS

1	Copyright.....	1
2	Table of Figures.....	4
3	Acknowledgements.....	5
4	Abstract.....	6
5	Introduction	7
5.1	Purpose and Requirements	7
6	Application Solution.....	8
6.1	Solution Overview	8
6.1.1	Introduction to the Application:	8
6.1.2	User Authentication System:	8
6.1.3	Communication Channels:	8
6.1.4	Voice Recognition and Processing:	8
6.1.5	Chatbot Integration:.....	8
6.1.6	Response Generation and Delivery:.....	8
6.2	Technologies used	9
6.3	Application Package Structure	9
6.4	Classes Explanation	12
6.4.1	Models.....	12
6.4.2	Views	12
6.4.3	Controllers.....	15
7	Application Proof of Concept (PoC)	18
7.1	User Register	18
7.2	User Login.....	20
7.3	User Chat	21
7.4	User Profile	23
8	Future improvements and thoughts.....	25
8.1	Origins.....	25
8.2	Future of Voice Assistants	26
8.3	Voice Assistants as Part of IoT.....	27
8.3.1	Historical Reference	27
8.3.2	How Voice Assistants Enhance User's Experience	28

8.3.3	Leverage Overall User's Routine	28
8.4	Security Concerns	29
8.4.1	Problem	29
8.4.2	Common Attacks on Voice Assistants Systems	29
8.4.3	Proposed solution	30
9	References	31

2 TABLE OF FIGURES

Figure 1: MVC Model Flow.....	9
Figure 2: MVC Package Structure	10
Figure 3: Project Resources.....	11
Figure 4: Activity Relationship	14
Figure 5: Voice Assistant Get Response Flow	17
Figure 6: User Register	18
Figure 7: User Register Select Profile Image	19
Figure 8: User Login	20
Figure 9: Chat Interface	21
Figure 10: Chat Example 1	22
Figure 11: User Profile Options.....	23
Figure 12: User Edit Profile	24
Figure 13: Voice Assistants Evolution Timeline	25
Figure 14: Voice Assistant with IoT.....	28

3 ACKNOWLEDGEMENTS

I would like to thank Professor Eythimios Alepis for entrusting me with the opportunity to deliver this bachelor thesis. I am also deeply grateful to my family for their unwavering support and encouragement throughout my studies, their love and understanding have been my greatest source of strength. I also wish to thank all those who have stood by my side, offering their assistance, encouragement, and friendship. Your support has been crucial to my success, and I am profoundly thankful for each one of you.

4 ABSTRACT

This bachelor thesis presents the development of an Android Voice Assistant designed to cater to the needs of Android users, offering assistance with basic queries throughout their daily routines. The application aims to enhance users' daily experiences by providing informative responses, motivational prompts, and entertainment options. Additionally, the thesis explores potential avenues for advancing Voice Assistants and their integration into people's lives, suggesting future enhancements to further streamline and personalize user interactions.

5 INTRODUCTION

Voice assistants have gradually occupied an important position in the daily life of most people. Artificial intelligence voice assistants can interpret human speech and respond. Users can ask their assistant questions and manage essential tasks. People can now immediately know the answer to a question by using a voice assistant. For example, when people use Siri or Google Assistant, they can query information just using some simple voice commands.

This interaction is mainly achieved through voice assistants and automatic speech recognition systems that output a set of words or sentences for text by recognizing and transforming the input speech fragments.

Nowadays, voice assistants have become a necessary application for most users. They are found not only in smartphones but also in smartwatches and cars, giving users the ability to interact easier with their device. This way, people can make their lives better by utilizing the functions of voice assistants. The relationship between artificial intelligence voice assistants and humans is getting closer and closer, and it has also led people to reflect on whether voice assistants can replace humans and whether humans will have feelings for voice assistants.

5.1 PURPOSE AND REQUIREMENTS

The current thesis aims in creating an android voice assistant application which will help users perform questions and provide them with the relevant response using different API calls. In order to accomplish this the following requirements are fulfilled:

- ❖ Android Chat Application.
- ❖ User Management (Sign-up, Sign-in, Sign-out).
- ❖ Voice Recognition.
- ❖ Retrieve response using chatbot APIs.
- ❖ Text-to-Speech Engine.
- ❖ Message History.

6 APPLICATION SOLUTION

6.1 SOLUTION OVERVIEW

The Android Voice Assistant application serves as an intelligent virtual assistant tailored to address the needs of Android users. This chapter provides an overview of the solution, detailing its key components, functionalities, and implementation strategies.

6.1.1 Introduction to the Application:

The Android Voice Assistant application is developed to provide users with seamless access to information, assistance, and entertainment through voice interactions.

Leveraging cutting-edge technologies such as chatbot APIs, voice recognition, and text-to-speech capabilities, the application offers a user-friendly interface for effective communication with the virtual assistant.

6.1.2 User Authentication System:

To ensure secure access and personalized experiences, the application incorporates a user authentication system.

Users are required to register an account or log in using existing credentials (email and password) to access the features and services offered by the assistant.

6.1.3 Communication Channels:

The application offers two communication channels through which users can interact with the virtual assistant.

Users have the option to input their queries via text input or utilize voice commands for a mostly hands-free experience.

6.1.4 Voice Recognition and Processing:

Utilizing advanced voice recognition technology, the application accurately transcribes users' spoken queries into text format.

The processed text inputs are then analyzed and interpreted by the virtual assistant to generate relevant responses.

6.1.5 Chatbot Integration:

The core functionality of the virtual assistant is powered by integration with chatbot APIs.

Leveraging natural language processing capabilities, the assistant is equipped to understand and respond to a diverse range of user queries and requests.

6.1.6 Response Generation and Delivery:

The responses are delivered to users in real-time through text-to-speech conversion, enabling seamless communication and interaction.

6.2 TECHNOLOGIES USED

- ❖ Java
- ❖ Android Studio
- ❖ Firebase Authentication
- ❖ Firebase Firestore Database
- ❖ Firebase Storage
- ❖ Chatbot APIs
- ❖ Android Speech Recognizer
- ❖ Android Text to Speech

6.3 APPLICATION PACKAGE STRUCTURE

To develop the application a Model-View-Controller (MVC) architectural pattern was used in order to organize its code structure efficiently. In this design, the Model represents the necessary objects that needed to be created (Message, UserSettings), the View Android user interface actions and visualizations, and the Controller performs all the necessary tasks to respond to the user's prompts, managing user interactions and updating the Model accordingly. By adhering to the MVC pattern, the application achieves a clear separation of concerns, making it easier to maintain, scale, and extend.

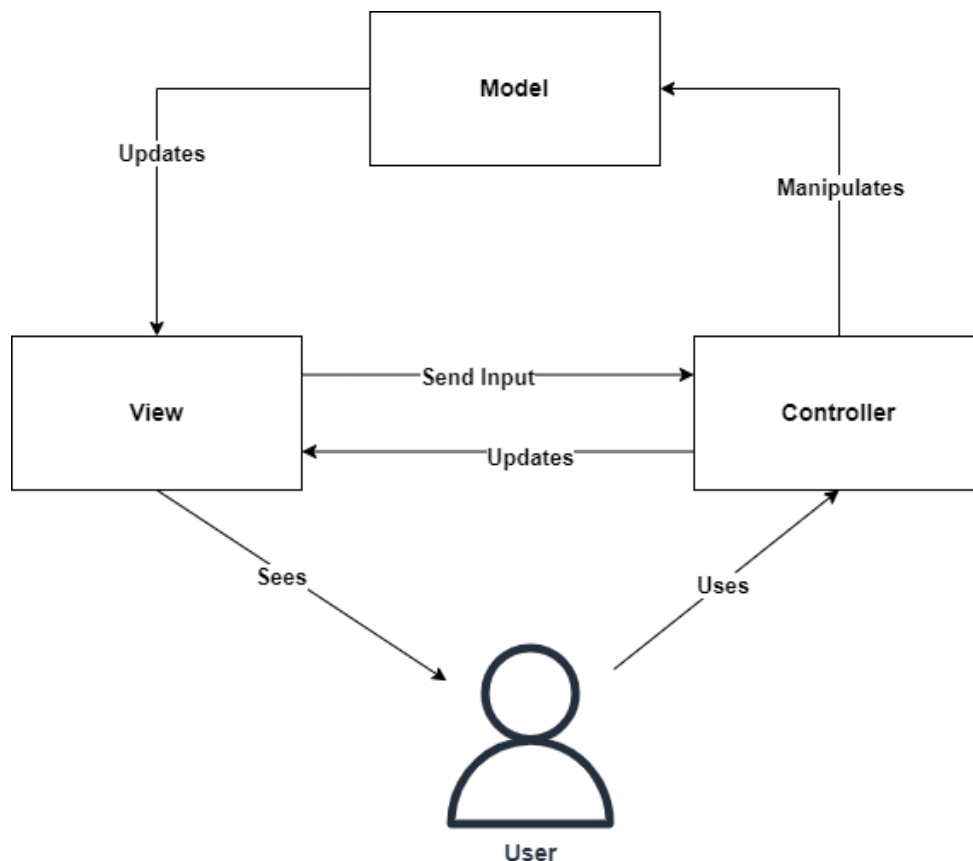
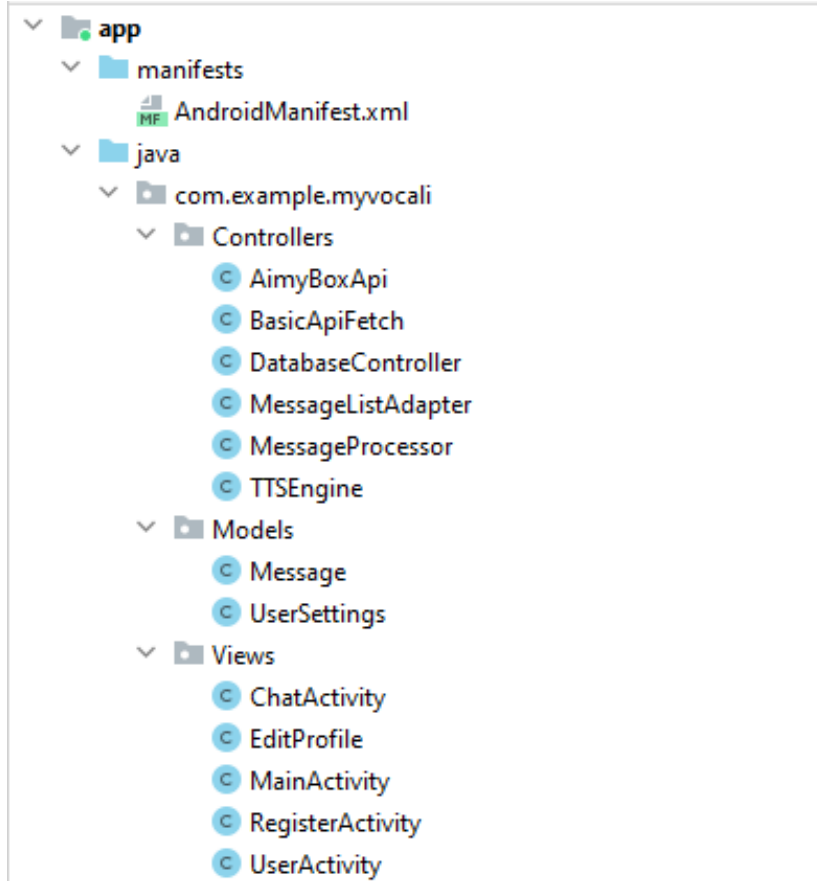


Figure 1: MVC Model Flow



As can be seen in the figure 2, the source content is separated in the MVC folders (Controllers, Models, Views).

Controllers' folder contains the Java classes responsible for handling the user's input and feeding the Views with the results.

Views folder contains all the Activity Classes which handle the visual and interactive part of the application. They provide the user with the result of his input and handle all the action made by the user such as button press, text input and voice input.

Finally, the model's folder contains the model which were constructed for the Voice Assistant application (Message Model and UserSettings Model)

Figure 2: MVC Package Structure

Also, all the Front-End components (Android Activities, Images) are stored in resources folder. These files contain all the User Interface components used in the related activity except from those that are dynamically generated (e.g. Message visualizations). The subfolder layout is used for activities components and the subfolder drawable is used for the custom images used in the Voice Assistant application.

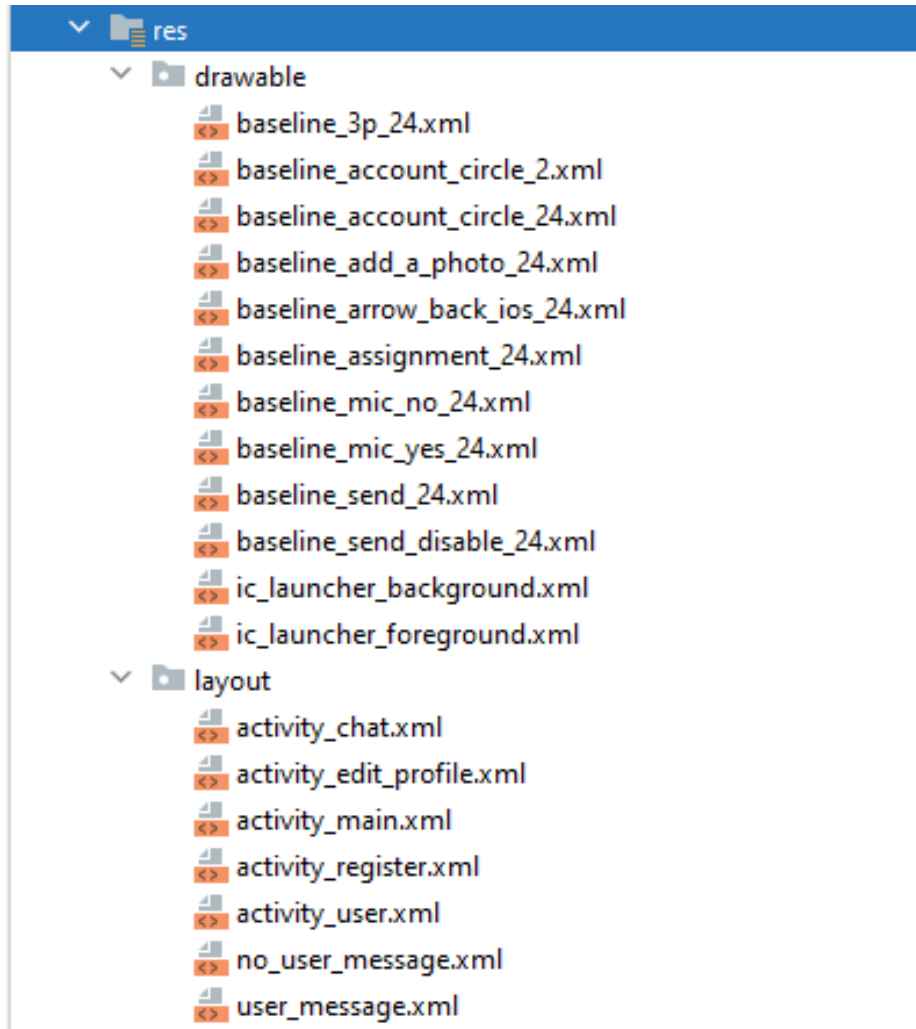


Figure 3: Project Resources

6.4 CLASSES EXPLANATION

6.4.1 Models

6.4.1.1 *Message Class*

Message class is used to represent a message. It contains the following required fields:

- ❖ `messageText`
- ❖ `sender`
- ❖ `createdAt`
- ❖ `senderID`

Also, it contains two constructors and getters for each field. Each message is stored in Firebase Firestore Database to maintain messaging history. Messages from database are retrieved when user open the application.

6.4.1.2 *UserSettings Class*

Most properties related to the user are saved in the Firebase Authentication (name, email, password). This class is used to keep extra information. This information is stored in the following fields:

- ❖ `playbackSpeed` (user's desired speed for text to speech engine)
- ❖ `gender`
- ❖ `birthday`

Also, it contains two constructors and getters for each field. User's settings are also stored in the Firebase Firestore Database.

6.4.2 Views

6.4.2.1 *MainActivity Class*

This class implements functionality along with the `activity_main.xml` layout file. It is used as the home page of the application when a user first opens it. It visualizes a login screen and gives also the option to register.

When launched it checks if user is already logged in. If this condition is satisfied, then it launches the `ChatActivity`.

When user performs a log in action, text from input fields is collected. First, a check is performed if both fields contain a value (non-empty), then using `FirebaseAuth` object a login attempt is performed. If the credentials are correct the user is logged in and `ChatActivity` is launched, else an error message is displayed.

6.4.2.2 *RegisterActivity Class*

This class implements functionality along with the `activity_register.xml` layout file. It is used to perform the necessary actions needed to register a new user.

When user is registering, he must enter the following fields:

- ❖ First name
- ❖ Last name
- ❖ Gender
- ❖ Birthday
- ❖ Email
- ❖ Password
- ❖ Confirm Password
- ❖ Profile Image (Optional)

After, when user presses the Register button, necessary checks are performed to validate all the fields. This process includes that there are no empty fields and that the passwords match. If the conditions are met a new user is created using the `FirebaseAuth` object. When complete his display name is set, then its profile picture (if selected) is uploaded to the `Firebase Storage` and his extra information from `UserSettings` class are saved to `Firebase Firestore Database`. As an identifier to the `Firebase Storage` and `Firestore Database` the user's `UID` is used.

Finally, when the registration process is over, the `ChatActivity` is launched.

6.4.2.3 *ChatActivity Class*

This class implements functionality along with the `activity_chat.xml` layout file. This is where the core features of the voice assistant application are fulfilled. The activity contains a header with the user profile picture, clickable to open the profile activity, a `recycler View` to hold the messages from both the user and the assistant, a button which when pressed the application starts listening for user's speech, and a message field with a send button which can be used to send a message to the assistant using typing instead of speaking.

All the required permissions are checked and asked from the user during the operations of the specific activity.

The `ChatActivity` facilitates real-time communication between user via text and speech as described above. It serves as the interface for users to send and receive messages from the assistant, both through typing and speech input. When a new message is received from the user, `ChatActivity` then incorporates a custom message processing engine (`MessageProcessor`) to generate responses to user input, which are then converted to speech using a text-to-speech engine (`TTSEngine`).

Each message, both from user and the application are then added to the `recycler view` and stored to the `Firebase Firestore Database`.

6.4.2.4 *UserActivity Class*

This class implements functionality along with the activity_user.xml layout file. It is accessible from the ChatActivity when user presses the profile icon on top right.

It is used to visualize the user's profile by displaying his profile picture, full name, and his email. Finally, it gives the ability to the user to perform the following actions:

- ❖ Edit his Profile.
- ❖ Logout.
- ❖ Delete message history.
- ❖ Delete his account.

6.4.2.5 *EditProfile Class*

This class implements functionality along with the activity_edit_profile.xml layout file. As its described but the name the specific activity is used in order to user be able to edit his profile.

The user can change his profile picture, his first and last name just by changing the current values displayed in the input text fields. Also, the user can modify the playback speed used from the text to speech engine, giving him the ability to adjust the speed to his requirements. Finally, user can change his password by entering the current password and then entering the new password and confirming it.

All necessary validation checks are performed, to no empty values exist and to ensure that new password and new password confirmation match each other.

6.4.2.6 *Activities Relationship*

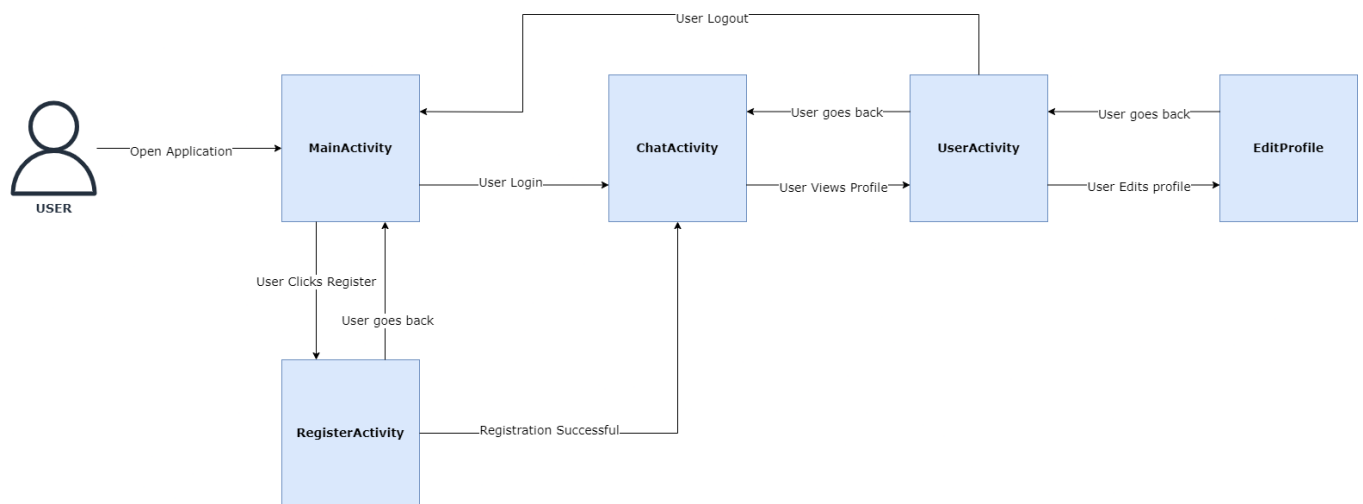


Figure 4: Activity Relationship

6.4.3 Controllers

6.4.3.1 *AimyBoxApi Class*

The *AimyBoxApi* controller serves as an asynchronous task to interact with the *AimyBox* API for natural language processing. It extends *AsyncTask* to perform network operations in the background in order to not interrupt user interface accessibility. The class constructs a JSON payload containing a user's query and sends it to the *AimyBox* API endpoint via an HTTP POST request. Then, after receiving a response from the API, it reads and processes the response, which typically contains the result of the natural language processing operation. The result is then returned to the *MessageProcessor* for further processing or enrichment based on the response.

AimyBoxApi is the main chatbot API used to fetch responses for the user's query. If response fails to be fetched a fallback error message is returned, which is then handled by the *MessageProcessor* class.

6.4.3.2 *BasicApiFetch Class*

The *BasicApiFetch* class is responsible for fetching data from a given API endpoint asynchronously. It extends *AsyncTask* to perform network operations in a background thread and not interrupt the user interface accessibility. Upon instantiation, it accepts a URL string representing the API endpoint to fetch data from.

It then establishes an HTTP connection to the specified URL, retrieves the response stream, and reads it using a *BufferedReader*. The response data is then accumulated into a *StringBuilder* and returned as a string.

If any exceptions occur during the network operation, such as *IOException*, the method returns a fallback error message which is then handled by the *MessageProcessor* class.

This class is a general reusable mechanism for interacting with the APIs used to implement the whole assistant. It communicates with the given API's URL sending a query and fetching the response.

6.4.3.3 *DatabaseController Class*

The *DatabaseController* class serves as a mediator between the voice assistant application and the *Firebase Firestore Database*. It handles the following operations related to user data and settings:

- ❖ Add messages to the message history.
- ❖ Delete message history.
- ❖ Delete user data.
- ❖ Read message history.
- ❖ Save user settings.
- ❖ Update playback speed settings.
- ❖ Read user settings.

It provides all the methods necessary to add messages to the message history, delete message history, delete user data, read message history, write user settings, update playback speed settings, and read user settings from the *Firestore* database. These methods enable seamless interaction between the user interface and the database, ensuring smooth functionality and data management within the voice assistant application.

6.4.3.4 *MessageListAdapter Class*

MessageListAdapter class provides the functionality needed to efficiently manage and display messages in the chat interface of the application.

It is responsible for displaying of messages within a RecyclerView in the ChatActivity. It extends the RecyclerView.Adapter class and handles different types of message views based on whether the message is sent by the user or received from the chatbot as response to the user's query.

The adapter inflates appropriate layout files for user (user_message.xml) and non-user (chatbot) messages (no_user_message.xml) and binds message data to the corresponding views. For received messages, it displays the sender's name, message content, and timestamp, along with the profile image if available. For sent messages, it shows the message content and timestamp.

The adapter also formats the timestamp and date of each message to display either the time or date when the message was sent, depending on whether the message is part of a consecutive series of messages sent on the same date or time. This ensures a clean and concise display of messages while maintaining chronological order to the message history.

6.4.3.5 *MessageProcessor Class*

MessageProcessor class serves as the central component in the voice assistant's functionality. It processes user messages to generate appropriate responses, incorporating various chatbot functions. Upon receiving a message, the class determines the user's intent through rule-based analysis. It then maps the intent to corresponding chatbot functions and executes them asynchronously using CompletableFuture function.

The class includes functions to handle different user intents such as requesting a motivational quote, seeking an activity when bored, asking for a Chuck Norris joke, or initiating an exit from the application. Each intent is associated with a specific chatbot function that fetches data from external APIs or generates responses based on predefined rules. For requests that do not correspond to the above functions a default function is used which first uses the AimyBox API. If no response is given from that API, then the brainshop API is used as a failover.

Additionally, the class includes error handling to manage exceptions during message processing and response generation. If an intent is not recognized or an error occurs, it provides a fallback response to ensure a seamless user experience and that a response is return to the user interface at all cases.

6.4.3.6 TTSEngine Class

The TTSEngine class is responsible for managing text-to-speech functionality within the voice assistant application. It initializes the TextToSpeech engine with the given application context and sets up an initialization listener to handle the engine's status. Upon successful initialization, it sets the language to US English and displays a message if the language is not supported. The class provides a method `speak()` to vocalize the input message using the TextToSpeech engine, and a `shutdown()` method to stop and release the resources used by the engine. Additionally, it includes a `setSpeechRate()` method to adjust the speech rate if needed based on the value the user has defined at his settings (default is 1). Overall, the TTSEngine class encapsulates text-to-speech functionality, providing seamless speech synthesis capabilities.

6.4.3.7 Voice Assistant Flow Diagram

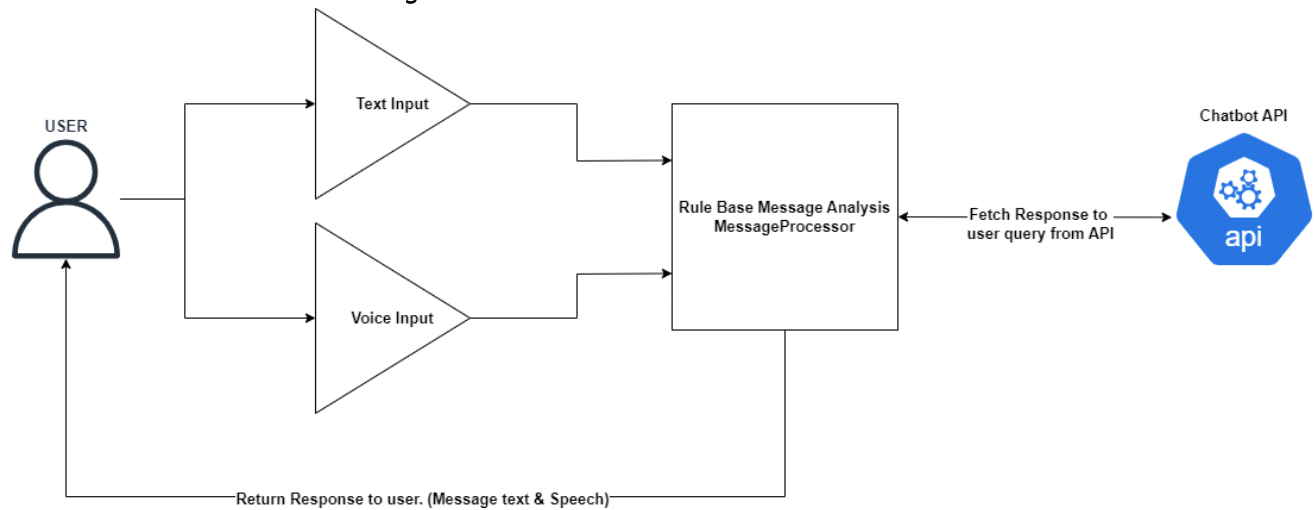
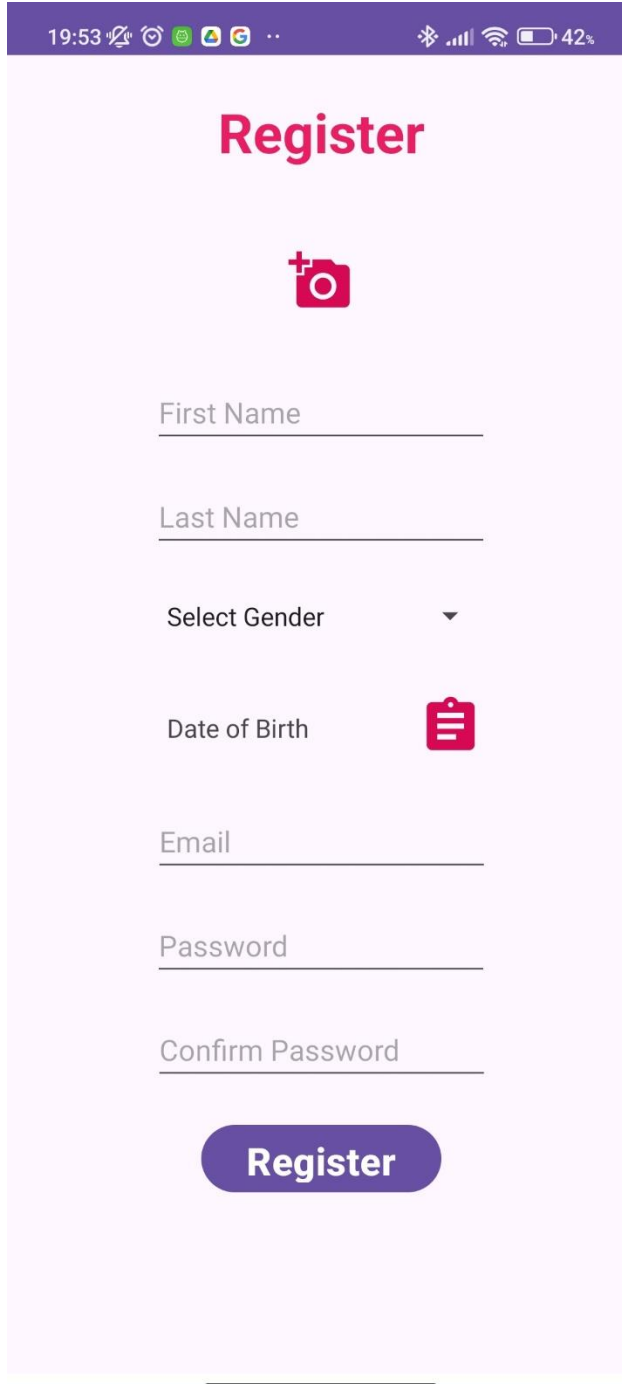


Figure 5: Voice Assistant Get Response Flow

7 APPLICATION PROOF OF CONCEPT (POC)

7.1 USER REGISTER



The screenshot displays the 'Register' activity. At the top, the word 'Register' is written in a large, pink font. Below it is a pink camera icon with a white plus sign. The form consists of the following elements from top to bottom:

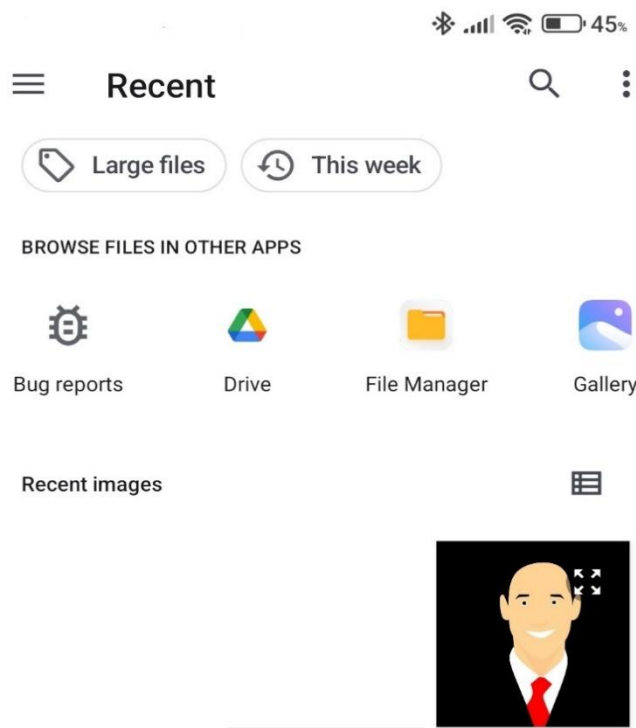
- A text input field labeled 'First Name'.
- A text input field labeled 'Last Name'.
- A dropdown menu labeled 'Select Gender' with a downward arrow.
- A text input field labeled 'Date of Birth' with a pink calendar icon to its right.
- A text input field labeled 'Email'.
- A text input field labeled 'Password'.
- A text input field labeled 'Confirm Password'.
- A large, rounded, purple button with the text 'Register' in white.

User Register Activity implements the functionality of user registration in the Voice Assistant Application. Registration is required for users to access the Voice Assistant.

While registering users must fill the required fields (first and last name, gender, date of birth, email and password). Optionally, they can select a profile picture which will be used.

Error handling is implemented ensuring that all fields are filled with the proper values and throwing error to the user.

Figure 6: User Register

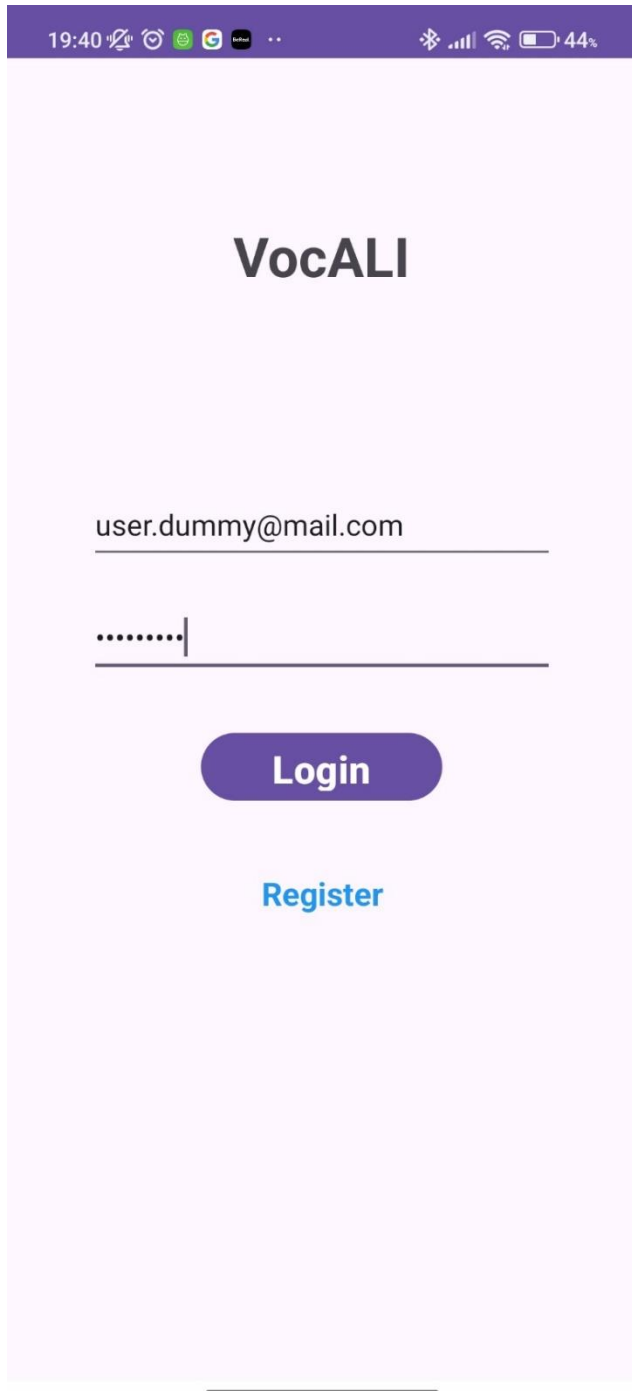


Using the system's library user can select any photo from his library and set it as its profile picture. The photo is then uploaded to the Google Storage and retrieved when the user logs in to his account.

The profile picture can be changed later through the user settings activity.

Figure 7: User Register Select Profile Image

7.2 USER LOGIN



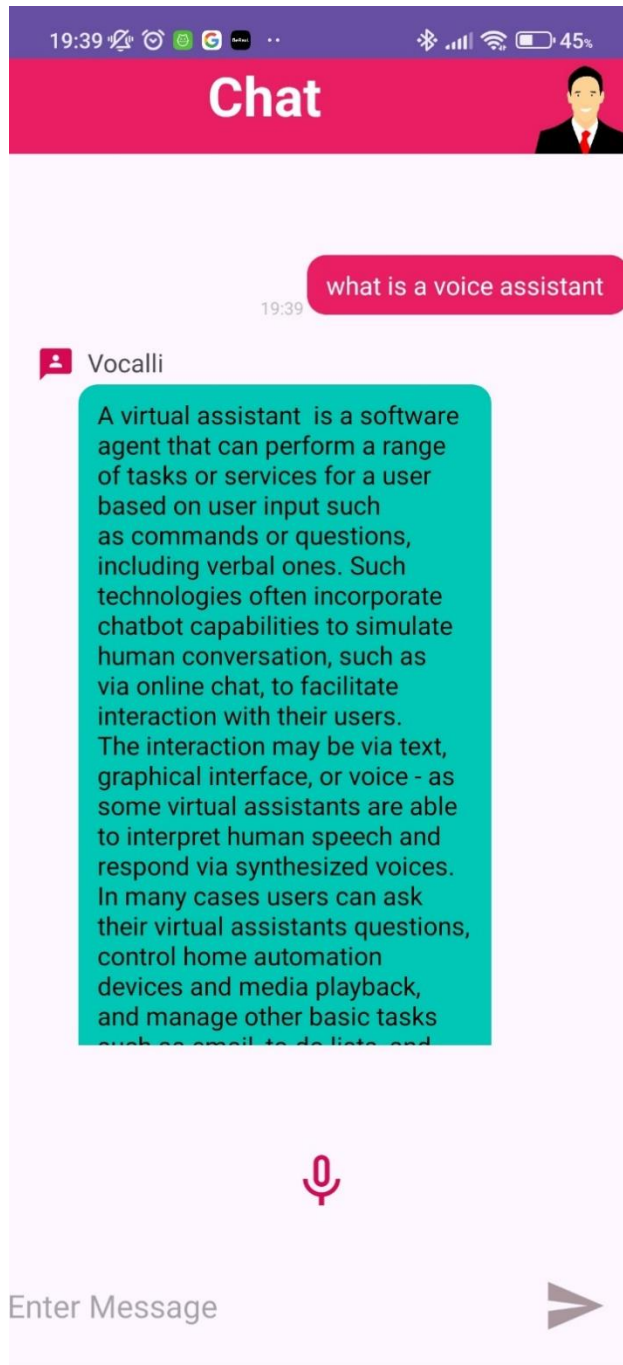
User Login Activity implements the functionality of the user login to the Voice Assistant application.

To log in the user, have to enter his email and his password. Then, pressing the Login Button he will be directed to the Main Activity if his credentials are correct. If not, error messages are displayed to the user.

The user management process is handled with the Google Firebase Authentication using the email & password sign in method.

Figure 8: User Login

7.3 USER CHAT



In the Main Activity is where the Voice Assistant core features are implemented.

Using the chat interface users can interact with the assistant to ask questions and have a conversational flow. The interaction occurs through voice and text input.

This way the user can use the assistant either with his voice using the voice recognition feature of the application or by typing his message in the text input field and pressing the send button.

Figure 9: Chat Interface

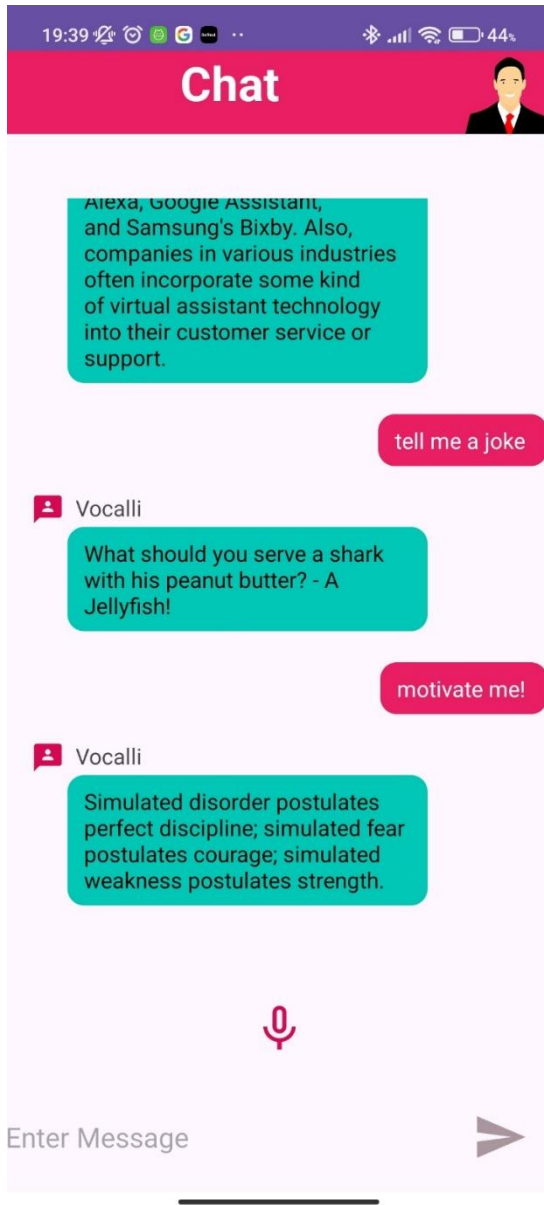
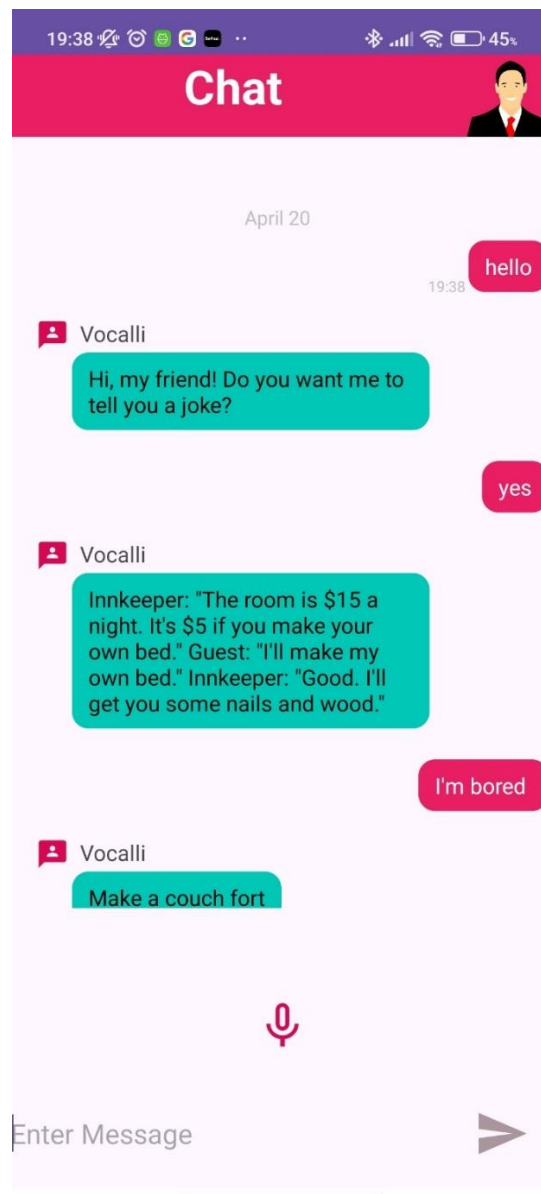


Figure 10: Chat Example 1

The voice assistant can be used by the user for entertainment purposes such as asking for jokes or by motivating the user and offering tasks to perform.

The chat interface visualizes the user messages to the right side of the user interface and the Vocalli (Voice Assistant) messages to the left side.

Finally, message history is preserved and saved in the Google Firebase Database. The message history is retrieved, and the messages are visualized on user login.



7.4 USER PROFILE

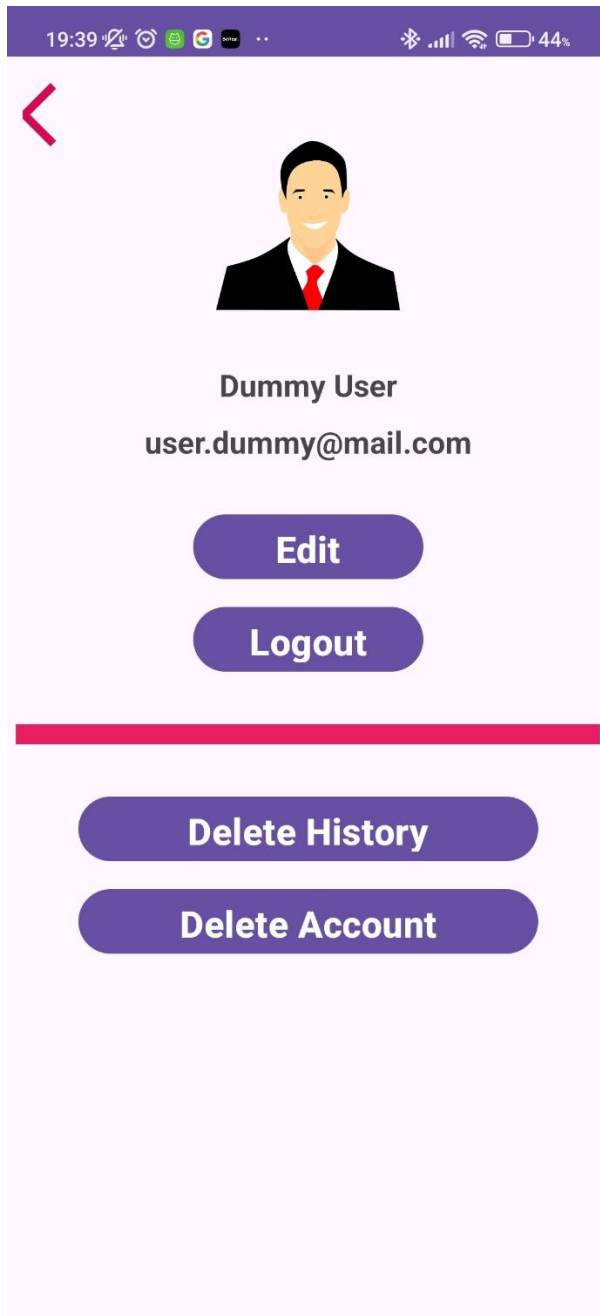


Figure 11: User Profile Options

User Profile Activity is accessible by pressing on the user profile icon on top right in the Chat Activity.

Through this menu, user can see basic information about his profile and perform different action. These actions include the ability to edit his account, log out, delete the message history, or delete his entire account.

When logging out the user is then redirected to the Login Activity and must re-enter his credentials to access again the assistant.

When deleting his account, the user no longer has access to his account and his message history is deleted from the database. To gain access again to the Voice Assistant he must create a new account.

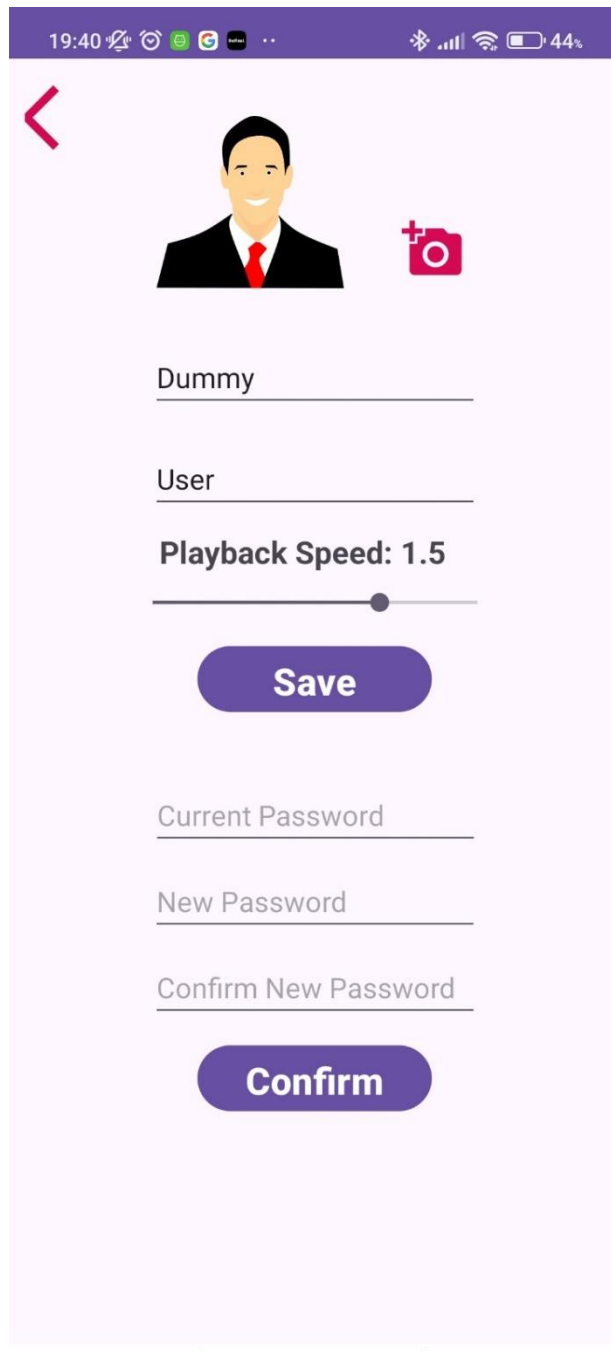


Figure 12: User Edit Profile

User Profile Edit Activity offers the user the ability to alter his profile and change a variety of settings.

He can change his first and last name or his profile picture by selecting a new one from his gallery and pressing the save button.

Also, he is given the ability to change the playback speed of the assistant. This is a crucial and useful feature from all users enhancing the accessibility capabilities of the Voice Assistant. Users with hearing problems can use a slower playback speed in order to be able to understand the responses of the assistant.

Finally, the user can use this menu to change his password. To do so, he must enter his current password and then enter and confirm his new password.

8 FUTURE IMPROVEMENTS AND THOUGHTS

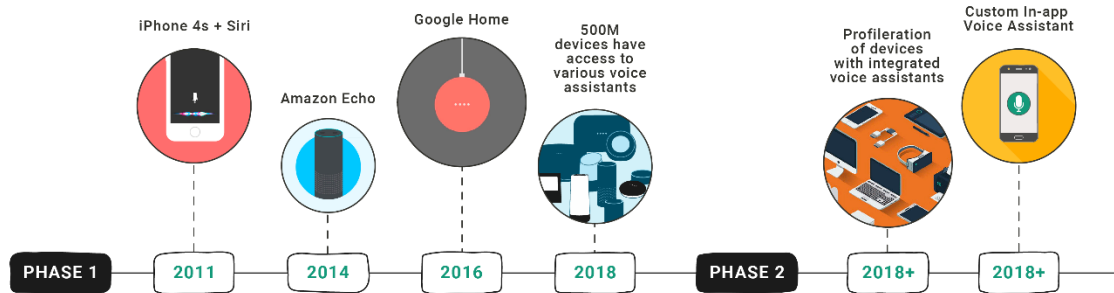


Figure 13: Voice Assistants Evolution Timeline

8.1 ORIGINS

Voice assistants have evolved significantly since their inception. While the concept of voice-controlled systems origins back to the 1950s, it wasn't until the early 2000s when the technology began to gain traction. In 2011, Apple introduced Siri as the first mainstream voice assistant, which was quickly followed by Google's Google Assistant, Amazon's Alexa, and Microsoft's Cortana.

Initially, these voice assistants had limited capabilities and often struggled with understanding user commands. However, advancements in natural language processing (NLP) and machine learning have led to rapid improvements in their accuracy and comprehension. Today, voice assistants can understand context, recognizing accents, and engaging in meaningful conversations with users. This way they have become a must-have tool for every user during their daily routine.

8.2 FUTURE OF VOICE ASSISTANTS

The future of voice assistants appears promising, driven by ongoing advancements in AI and natural language processing (NLP). Several trends are poised to shape their evolution:

- ❖ **Multimodal Interaction:** As the development of voice assistants continues to advance, they are poised to integrate seamlessly with other interfaces such as augmented reality (AR) and virtual reality (VR), revolutionizing user experiences. This integration will offer users a more immersive and intuitive interaction with their digital environments. By blending the capabilities of voice assistants with AR and VR technologies, users will be able to engage with digital content and services in innovative ways, from navigating virtual worlds to accessing information and performing tasks with natural language commands. This convergence of voice, AR, and VR will unlock new possibilities for enhanced communication, productivity, and entertainment, marking a significant step forward in the evolution of interactive technology.
- ❖ **Personalization:** In the coming years, voice assistants are anticipated to provide increasingly personalized interactions, enhancing user experiences by tailoring responses and actions to individual preferences, habits, and even emotional cues. This level of customization will enable voice assistants to build deeper connections with users, understanding their unique needs and preferences over time. By leveraging advanced AI algorithms, voice assistants will adapt their interactions in real-time, creating more meaningful and relevant experiences for each user. Whether it's recommending personalized content, adjusting settings to match individual preferences, or providing empathetic responses, personalized interactions will elevate the role of voice assistants in our daily lives, making them indispensable companions in various contexts. Additionally, as LLM advances, users will have the opportunity to actively train voice assistants on their specific data and needs, further enhancing personalization. Through feedback mechanisms and interactive training sessions, users can provide voice assistants with valuable insights into their preferences, habits, and even nuances of speech. This user-driven training approach will enable voice assistants to continuously refine their understanding of individual users, ensuring that they adapt and evolve alongside changing preferences and behaviors. By empowering users to actively shape their voice assistant's capabilities, this collaborative training process will foster a deeper sense of ownership and customization, ultimately leading to more seamless and intuitive interactions tailored to each user's unique requirements.
- ❖ **Integration with IoT:** As the Internet of Things (IoT) ecosystem continues to expand, voice assistants are poised to assume a pivotal role in connecting and orchestrating smart devices. By leveraging their natural language processing capabilities, voice assistants will streamline the management of interconnected home environments, offering users unprecedented convenience and control. Whether it's adjusting thermostat settings, controlling lighting, or even managing security systems, voice commands will become the primary interface for interacting with IoT devices. This seamless integration of voice assistants with the IoT landscape promises to enhance the efficiency and functionality of modern homes, ushering in an era of interconnected living where everyday tasks are effortlessly automated and managed.
- ❖ **Voice Commerce Expansion:** The surge in voice shopping is poised to revolutionize the e-commerce realm, offering consumers unparalleled convenience by enabling them to make purchases effortlessly through voice commands. This transformative trend is set to redefine the way people shop online, as voice assistants seamlessly facilitate transactions based on user

preferences and needs. As voice shopping becomes increasingly prevalent, it will further embed voice assistants into the fabric of daily life, transforming them from mere assistants to indispensable companions in the consumer journey. With the ability to browse, select, and purchase items simply by speaking, voice shopping promises to streamline the shopping experience and elevate convenience to new heights, reshaping the e-commerce landscape in the process.

Imagine this scenario: you open your refrigerator and realize that you're out of milk. Instead of jotting it down on a shopping list or reaching for your phone, you simply speak aloud, "Order milk." Instantly, your voice assistant, seamlessly integrated into your smart home ecosystem, springs into action. It swiftly processes your request, ensuring that a fresh carton of milk is on its way to your doorstep without you ever lifting a finger. This effortless interaction exemplifies the transformative power of voice shopping, where everyday tasks are simplified through the intuitive use of voice commands, revolutionizing the e-commerce landscape one purchase at a time.

8.3 VOICE ASSISTANTS AS PART OF IOT

8.3.1 Historical Reference

In the 2013 movie, "Her," audiences were introduced to Theodore Twombly, an introverted man on the verge of divorce from his high school sweetheart. To cope with his loneliness, Theodore decides to purchase the new OS1, the world's first artificially intelligent operating system, advertised as "not just an operating system, but a consciousness" and names her Samantha. The operating system, designed to adapt and evolve like a human being, conducts all communication through voice commands and soon moves beyond being just a voice assistant to a love interest for Theodore.

This future may not seem too unrealistic with the rise of voice assistants like Alexa, Siri, and Bixby. They are an essential component of today's connected homes. Put simply, these devices are to the connected home what the brain is to the human body. They are the center of the system that connects all essential functions and the entity itself would be of little use without it.

Still, as an integral piece of the connected home, they are not being utilized to their full potential today. Most consumers who own devices with a voice assistant use the functionality at least once a day, according to PwC. But the use-cases still prove to be quite elementary. The PwC research outlines the most common uses of voice assistants as:

- ❖ Checking the weather or news.
- ❖ Playing music.
- ❖ Searching for something that they'd normally type into a search engine.
- ❖ Sending a text or email.
- ❖ Asking a quick question.

8.3.2 How Voice Assistants Enhance User's Experience

Voice assistants can play a significant role in making guests feel welcome and enhancing their overall experience in the hospitality industry. When integrated effectively, these AI-powered assistants can provide a personalized and convenient stay for guests. Some enhancements can be:

- ❖ **Personalized Greetings:** Greet users by name upon check-in for a warm welcome.
- ❖ **Room Control:** Control lights, thermostat, curtains, and TV with voice commands.
- ❖ **In-Room Dining:** Order food or get menu recommendations via voice.
- ❖ **Entertainment:** Stream content to the TV or get info on available channels.
- ❖ **Language Translation:** Assist users in daily routine with language translation.
- ❖ **Local Recommendations:** Receive personalized suggestions for nearby places.
- ❖ **Safety and Security:** Contact emergency services or request assistance in emergencies.
- ❖ **Accessibility:** Assist users with disabilities with voice-controlled features.
- ❖ **Privacy Controls:** Allow users to disable or clear voice command history.

8.3.3 Leverage Overall User's Routine

- ❖ **Keyless Entry:** IoT-enabled door locks can provide keyless entry through smartphone apps, reducing the need for physical keys or key cards and enhancing security.
- ❖ **Maintenance Alerts:** IoT sensors detect issues and incidents for proactive maintenance and response.
- ❖ **Asset Tracking:** Track location and status of assets using IoT sensors.

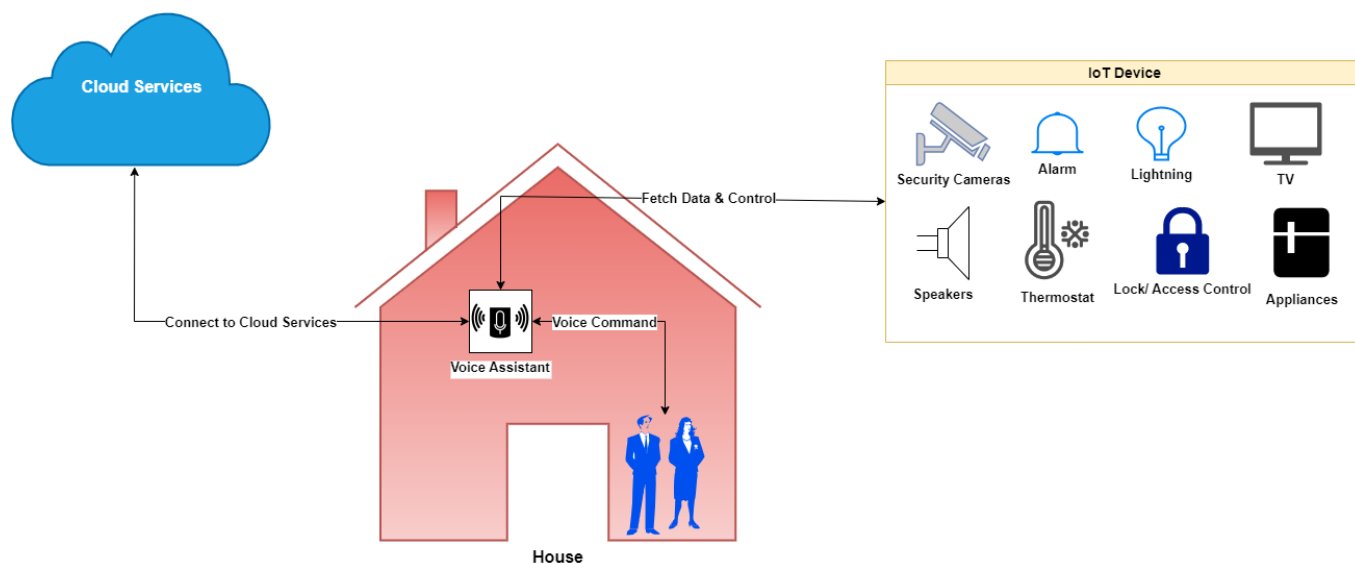


Figure 14: Voice Assistant with IoT

8.4 SECURITY CONCERNS

8.4.1 Problem

In 2018, PwC surveyed a sample of 1,000 Americans between 18-64 to learn more about the awareness of voice technology. The results were astounding: 90% of respondents said they were already familiar with voice assistant technology, with 57% of respondents already using their voice assistant on their smartphone and 20-30% using the technology on other devices like tablets, laptops, speakers, and TV remotes.

Back in 2013, Microsoft tried to pioneer voice assistants through the mandatory launch of Xbox Kinect with its new Xbox One. The launch should have gone off without a hitch. Its competing gaming console was far more expensive without being significantly more valuable. But Microsoft's sure win was not to be. The company bungled the Xbox One launch because it revealed that the Kinect (a mandatory accessory) was always listening and watching in the background.

From the consumer perspective, the issues associated with voice assistants are two-fold. First, the consumer is worried about "who is listening." In these cases, those who use voice assistants must be transparent about how the data gets used and who gets to hear it. The second issue is one of security. Voice assistants only "listen" in the background until called upon, but they still collect huge amounts of data, which will only be compounded by the number of available applications for voice assistants in the growth of users.

8.4.2 Common Attacks on Voice Assistants Systems

- ❖ **Acoustic Denial of Service (DoS) attack** targets the acoustic channel used for voice commands, disrupting the functionality of the Voice Assistant
- ❖ **Hidden Voice Commands:** malicious actors can embed commands within audio signals that are imperceptible to humans but can be interpreted by the Voice Assistants, leading to unauthorized actions.
- ❖ **Spoofing Attacks:** attackers attempt to deceive the system by presenting fake audio inputs.
- ❖ **Phishing and Social Engineering:** attackers may try tricking users into giving sensitive information or performing actions by mimicking the voice assistant's responses.
- ❖ **Context Manipulation:** attackers can try to exploit the context-aware nature of voice assistants to trigger unintended actions (e.g., manipulating smart home devices).
- ❖ **Data Leakage:** aiming in exploiting vulnerabilities to access sensitive user information stored on the device or transmitted through the voice assistant.
- ❖ **Eavesdropping:** unauthorized parties intercept voice commands or conversations using passive listening. Also, malicious apps or devices continuously listen to the environment without user concern using active listening.

8.4.3 Proposed solution

To address the security concerns surrounding voice assistants, particularly regarding data privacy, a proposed solution involves implementing on-premises functionality. As the Internet of Things (IoT) evolves and voice assistants become increasingly integrated into users' lives and smart homes, providing the option for on-premises deployment offers a promising approach. Under this model, each user would have their own voice assistant, initially pre-trained with general knowledge and capabilities. However, the crucial distinction lies in the assistant's ability to further personalize its responses and actions based on the user's specific needs and preferences.

Central to this solution is the concept of data localization. Rather than transmitting and storing user data on remote servers controlled by third-party companies, the voice assistant operates within the user's home environment. Data collected from interactions with the assistant remains securely stored on a local server or device, ensuring that sensitive information is not shared with external entities or subject to potential breaches. This localized approach grants the users with greater control over their data and alleviates concerns about unauthorized access or misuse.

Furthermore, extending the option for on-premises deployment to organizations and public sector entities offers additional benefits. Government agencies, military institutions, and other sensitive sectors can leverage voice assistant technology without compromising confidentiality or risking data leaks. By maintaining data sovereignty within their own infrastructure, these organizations can harness the productivity and efficiency gains associated with voice assistants while upholding stringent security protocols and regulatory compliance requirements.

Finally, embracing on-premises deployment for voice assistants represents a proactive step towards enhancing data privacy and security in an increasingly connected world. By empowering users with greater control over their personal information and offering secure alternatives for organizations, this solution paves the way for a more trusted and resilient ecosystem of voice-enabled technologies.

9 REFERENCES

1. **Firestore** (<https://firebase.google.com/>): Google platform that offers a suite of cloud-based tools and services for app development, including real-time databases, authentication, analytics, and more.
2. **Android Developers** (<https://developer.android.com/>): The official site for Android app development, providing documentation, guides, SDK tools, and resources for building Android applications.
3. **GeeksforGeeks - How to Create a Chatbot in Android with Brainshop API** (<https://www.geeksforgeeks.org/how-to-create-a-chatbot-in-android-with-brainshop-api/>): A tutorial on creating a chatbot for Android applications using the Brainshop API, with step-by-step instructions and code examples.
4. **Brainshop** (<https://brainshop.ai/>): A platform offering AI chatbot creation and management services, allowing users to build, train, and deploy chatbots for various applications.
5. **FavQs API - Quote of the Day** (<https://favqs.com/api/qotd>): An API providing a random quote of the day.
6. **Chuck Norris Jokes API** (<https://api.chucknorris.io/jokes/random>): An API that delivers random Chuck Norris jokes, used for entertainment purposes.
7. **Bored API** (<https://www.boredapi.com/api/activity>): An API that suggests random activities to do when bored, providing various ideas and activities.
8. **Slang Labs - Voice Assistant for Apps** (<https://www.slanglabs.in/think-voice/voice-assistant-for-apps>): A service that helps integrate voice assistant capabilities into applications, enhancing user interaction through voice commands.
9. **LinkedIn - Voice-Activated Future** (<https://www.linkedin.com/pulse/voice-activated-future-exploring-world-voice-assistants-probyto>): An article exploring the current state and future potential of voice assistants, discussing their applications and impact on technology.
10. **Privacy Policies - Voice Assistants Privacy Issues** (<https://www.privacypolicies.com/blog/voice-assistants-privacy-issues/>): A blog post addressing privacy concerns associated with voice assistants, including data collection, user consent, and best practices for maintaining privacy.
11. **Jabil - How Voice Assistants Will Transform the IoT Ecosystem** (<https://www.jabil.com/blog/how-voice-assistants-will-transform-the-iot-ecosystem.html>): An article discussing how voice assistants are influencing the Internet of Things (IoT) ecosystem, enhancing connectivity and user experience.
12. **Medium - The Synergy of Voice Assistants and the Internet of Things** (<https://medium.com/@phonesuites/the-synergy-of-voice-assistants-and-the-internet-of-things-de83315c1b21>): A post on Medium exploring the integration of voice assistants with IoT devices, highlighting the benefits and challenges of this synergy.
13. **Typeset - A Study on Smart Voice Assistants** (<https://typeset.io/papers/a-study-on-smart-voice-assistants-2wt11hfh>): An academic paper analyzing various aspects of smart voice assistants, including their design, functionality, and user's interaction.
14. **Typeset - Personal Voice Assistant Security and Privacy: A Survey** (<https://typeset.io/papers/personal-voice-assistant-security-and-privacy-a-survey-3nqkcrco>): A paper of security and privacy issues related to personal voice assistants, providing insights and recommendations for improving user security posture.