

Intelligent Companion APP

Prof. Manoj Deshpande, Gourab Chakraborty, Tejal Bhagat, Gayatri Padile, Ritesh Yadav

Department of Computer Engineering

A C Patil College of Engineering, Kharghar, Navi Mumbai, Maharashtra, India

Abstract: *Virtual assistants are a boon for everyone in this new era of the 21st century. It has paved the way for a new technology where we can ask questions to machines and can interact with IVAs as people do with humans. This new technology attracted almost the whole world in many ways like smart phones, laptops, computers etc. Some of the significant VPs are like Siri, Google Assistant, Cortana, and Alexa. Voice recognition, contextual understanding and human interaction are the issues which are not solved yet in these IVAs. According to those results many services were covered by these assistants but still there are some improvements required in voice recognition, contextual understanding and hand free interaction. After addressing these improvements in IVAs will definitely increase its use is the main goal for this research paper.*

Keywords: Virtual assistant, Contextual understanding

I. INTRODUCTION

One of the best applications of artificial intelligence is the virtual personal assistant (VPA), which offers a new way for people to delegate tasks to machines. It is designed to offer maximum convenience to the users. Intelligent Companion App – our Android-based project is designed and developed using Dialogue flow to help comprehend natural language and can assist with tasks like calling a specific contact, adjusting volumes, and providing weather forecasts. This voice assistant companion app helps in establishing a new environment between the user and the system. It seeks to create a better method of engagement that gives the user the impression that they are speaking with another individual.

1.1 Scope and Objectives

Many people enjoy using voice assistants because they enable us to perform a range of things without using our hands, especially on mobile devices. It makes use of voice recognition technology and artificial intelligence to provide the user's desired outcome precisely and quickly. While asking a computer to set a timer might seem straightforward, the technology is fascinating. A digital companion instinctively knows when, how, and if to interact with each of us. A digital companion adapts and implements its interactive experience in accordance with our specific needs and preferences rather than waiting for our command.

1.2 Modules and its Description

The system comprises 2 major modules with their submodules as follows: USER:

1. Basic Voice Assistant

- The user can have a seamless conversation with the system as it is made using Dialogue flow.
- The system will comprise various intents to carry out the user's commands.
- The user can verbally ask to open WhatsApp and other apps.
- The user can ask about weather updates.
- The user can ask to call a specific contact.
- The user can verbally adjust volume controls and can also change profiles.

2. Manage Schedule

- The user will need to manually add his schedule for the day.
- The system will start talking whenever it analyses the user's free time.
- The system will automatically shut down if there is no response from the user within a minute.
- The conversation by the Bot will be based on emotional analysis by the audio of the user.

1.3 Existing System

1. Problem with the current scenario Many techniques and procedures proposed traditionally have not produced accurate results. Furthermore, these approaches did not perform well in challenging settings.
2. Drawbacks of the existing system In the existing system, it has become difficult to detect the mood or emotions of the users for a better experience. Sometimes disturbance in speech recognition was giving out faulty results.

1.4 Proposed System

Intelligent Companion App is an Android-based project designed by using Google Dialogue flow as the NLU platform. All the models used for this system are structured in a way to work as efficiently as possible. Dialogue flow's key advantage is its integration with Google. The Machine Learning Algorithms incorporated into the platform assist agents in interpreting the natural language that is the user's expression. Each Agent includes Intents that match the user's expression, and action is taken in response to the query. It not only delivers answers to clients but also allows agents to have brief conversations with them. The user can assign tasks to the system verbally, like opening WhatsApp or some other app, weather updates, calling a specific contact, controlling volume and changing profile. Google APIs will be used to assist with such actions. The user can also manually update their schedule. The system will detect the free time in between the schedules and start free time talking. The conversation by the Bot will be based on the user's audio emotional analysis.

The system will stop free time talking if it doesn't detect any response from the user within a minute. For this project, XML and Android – Java is used on the front end and SQLite is used on the backend. The IDE used is Android Studio.

II. LITERATURE SURVEY

2.1 Literature Review

A computer primarily based approach for performing a command via a voice consumer interface on a subset of objects. The subset is selected from a fixed list of items, each having an object type at least one taggable field is associated with the object type and has a corresponding value. The set of objects is saved in the laptop memory. An Utterance is acquired from the person and consists of a command, an object type choice, a tag-gable field selection, and a price for the taggable discipline. Responsive to the utterance, at least one item is retrieved from the set of gadgets, the item of the sort selected through the user and having a price within the taggable area selection that matches the taggable field fee obtained from the user the command is done on the item. The object includes textual content that's converted to voice output. They envisioned that someday computers will recognize natural language and count on what we need, whilst and where we need it, and proactively whole responsibilities on our behalf. However, speech recognition and machine getting to know have persevered to be refined, and based records served through packages and content providers have emerged. We agree with that as computer systems turn out to be smaller and more ubiquitous. The recognizer is designed to change a verbal articulation from an individual into an alternate method of data (e.g., text). A hand held individual colleague including a voice-recognizer and characteristic dialect processor is disclosed. This snippet of data can be a plan for the day, data in the individual's logbook or data from the individual's address book, such as a telephone number. The most well known utilization of iPhone is "SIRI" which causes the end client to be versatile with voice and it additionally reacts to the voice charges of the client. 4 It is named as Personal Assistant with Voice Recognition Intelligence, which takes the client's contribution to type of voice or content and processes it and returns the yield in different structures like activity to be performed or the item is directed to the end client. Furthermore, this proposed framework can change the method for communications between the end client and the cell phones [4]. Open Data is currently gathering consideration for imaginative administration creation, predominantly in the zone of government, bio science, and shrewd venture. Be that as it may, to advance its application more for purchaser administrations, a web crawler for Open Data to realize what sort of information is there would be of assistance.

III. REQUIREMENT ANALYSIS

3.1 Hardware Requirements

- Windows 7 or higher
- 100 GB ROM or higher

- I3 processor system or higher
- 8 GB RAM or higher
- 102 Keyboard and Mouse
- Monitor
- Android Phone (6.0 and above)

3.2 Software Requirements

- Android studio

IV. IMPLEMENTATION

4.1 Introduction to Android

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

1. A flexible Gradle-based build system.
2. A fast and feature-rich emulator.
3. A unified environment where you can develop for all Android devices.
4. Instant Run to push changes to your running app without building a new APK.
5. Code templates and GitHub integration to help you build common app features and import sample code.
6. Extensive testing tools and frameworks.
7. Lint tools to catch performance, usability, version compatibility, and other problems.
8. C++ and NDK support
9. Built-in support for a. Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine. Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:
 - Android app modules
 - Library modules
 - Google App Engine modules By default, Android Studio displays your project files in the Android project view, as shown in figure 1. This view is organized by modules to provide quick access to your project's key source files. All the build files are visible at the top level under Gradle Scripts and each app module contains the following folders:
 - manifests: Contains the AndroidManifest.xml file.
 - java: Contains the Java source code files, including JUnit code.
 - res: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

4.2 Tool Windows

Instead of using preset perspectives, Android Studio follows your context and automatically brings up relevant tool windows as you work. By default, the most commonly used tool windows are pinned to the tool window bar at the edges of the application window. 1. To expand or collapse a tool window, click the tool's name in the tool window bar. You can also drag, pin, unpin, attach, and detach tool windows. 2. To return to the current default tool window layout, click Window $\bar{\iota}$ Restore Default Layout or customize your default layout by clicking Window $\bar{\iota}$ Store Current Layout as Default. 3. To show or hide the entire tool window bar, click the window icon in the bottom left-hand corner of the Android Studio window. 4. To locate a specific tool window, hover over the window icon and select the tool window from the menu.

4.3 Navigation

1. Switch between your recently accessed files using the Recent Files action. Press Control+E (Command+E on a Mac) to bring up the Recent Files action. By default, the last accessed file is selected. You can also access any tool window through the left column in this action.
2. adView the structure of the current file using the FileStructure action. Bring up the File Structure action by pressing Control+F12 (Command+F12 on a Mac). Using this action, you can quickly navigate to any part of your current file.
3. Search for and navigate to a specific class in your project using the Navigate to Class action. Bring up the action by pressing Control+N (Command+O on a Mac). Navigate to Class supports sophisticated expressions, including camel humps, paths, line navigator, middle name matching, and many more. If you call it twice in a row, it shows you the results of the project classes.
4. Navigate to a file or folder using the Navigate to File action. Bring up the Navigate to File action by pressing Control+Shift+N (Command+Shift+O on a Mac). To search for folders rather than files, add a / at the end of your expression.
5. Navigate to a method or field by name using the Navigateto Symbol action. Bring up the Navigate to Symbol action by pressing Control+Shift+Alt+N (Command+Shift+Alt+O on a Mac).
6. Find all the pieces of code referencing the class, method, field, parameter, or statement at the current cursor position by pressing Alt+F7.

4.4 Gradle Build System

Android Studio uses Gradle as the foundation of the build system, with more Android specific capabilities provided by the Android plugin for Gradle. This build system runs as an integrated tool from the Android Studio menu, and independently from the command line. By employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files. Android Studio build files are named build.gradle. They are plain text files that use Groovy syntax to configure the build with elements provided by the Android plugin for Gradle. Each project has one top-level build file for the entire project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

4.5 Multiple APK Support

Multiple APK support allows you to efficiently create multiple APKs based on screen density or ABI. For example, you can create separate APKs of an app for the hdpi and mdpi screen densities, while still considering them a single variant and allowing them to share test APK, javac, dx, and ProGuard settings. Debug and Profile Tools Android Studio assists you in debugging and improving the performance of your code, including inline debugging and performance analysis tools.

4.6 Performance Monitors

Android Studio provides performance monitors so you can more easily track your app's memory and CPU usage, find deallocated objects, locate memory leaks, optimize graphics performance, and analyze network requests. With your app running on a device or emulator, open the Android Monitor tool window, and then click the Monitors tab

- Allocation tracker: Android Studio allows you to track memory allocation as it monitors memory use. Tracking memory allocation allows you to monitor where objects are being allocated when you perform certain actions. Knowing these allocations enables you to optimize your app's performance and memory use by adjusting the method calls related to those actions.

4.7 Code inspection

Whenever you compile your program, Android Studio automatically runs configured Lint and other IDE inspections to help you easily identify and correct problems with the structural quality of your code. The Lint tool checks your Android project source files for potential bugs and optimization improvements for correctness, security, performance, usability, accessibility, and internationalization.

V. ADVANTAGES AND LIMITATIONS

5.1 Advantages

- It is user-friendly
- It is easy to use and free of cost.
- The usage of Dialogflow technology results in smooth conversation flow.
- The user can have a virtual companion anytime, anywhere.
- The user can carry out various tasks just by talking to the voice assistant.

5.2 Limitations

- Internet is mandatory. system will not work in the background or when the phone is locked.

5.3 Features

- Load Balancing: Since the system will be available only after the admin logs in the amount of load on the server will be limited to the period of admin access.
- Easy Accessibility: Records can be easily accessed and stored and other information respectively.
- User-Friendly: The website/application will be giving a very user-friendly approach for all users.
- Efficient and reliable: Maintaining the all-secured database on the server which will be accessible according to the user requirement without any maintenance cost will be very efficient as compared to storing all the customer data on the spreadsheet or physically in the record books. • Easy maintenance: Intelligent Companion App is designed as an easy way. So, maintenance is also easy.

VI. CONCLUSION

An Intelligent Companion App developed in Java programming language. The Development of this system takes a lot of effort from us. We think this system gave a lot of satisfaction to all of us. Though every task is never said to be perfect in this development field even more improvement may be possible in this application. We learned so many things and gained a lot of knowledge about the development field. We hope this will prove fruitful to u

BIBLIOGRAPHY

- [1]. Websites en.wikipedia.org
- [2]. <https://www.researchgate.net/publication/351723449AVoiceBasedAssistantUsingGoogleDialogflowandMachineLearning>
- [3]. <https://www.smartsheet.com/voice-assistants-artificialintelligence>
- [4]. <https://alan.app/blog/voiceassistant-2/>
- [5]. <https://blog.intuitionrobotics.com/shifting-from-voiceassistants-to-empathetic-digitalcompanions>