**1. INTRODUCTION**

Personal assistants have become an integral part of any smart phone users these days. Academic Advisor is a virtual assistant which aims at providing a user interface which is interactive and can keep the user engaged while providing the best possible facilities of a smart virtual assistant. Academic Advisor can read out news, give answers to general questions and open applications for you.

All of this functionality can be divided into 3 main domains. The user interface, app integration and the smart answering capability. We shall discuss each one in detail. The user interface of Academic Advisor is simple yet solid. It will take no time for the user to get used to it because the interface is highly inspired by the very popular messaging at chat applications (threaded messages). The query sent in by the user will aligned to the right and the response generated will be aligned to the left. All this will be displayed on the view in form of a list of messages which will become scrollable once the list is big enough. This domain also includes the the ways in which the user can interact with the application.

The first method being the text input through the keyboard and the next being the voice input which can be activated using the on screen voice button present on the interface. The output as well will have two methods to be conveyed, it will be displayed to the user as well as read out.

The second domain is app integration, which suggests that Academic Advisor must be integrated with some applications. Academic Advisor can open various applications for you. We will start of by integrating a few apps and then keep adding applications to that list. If you want to open some application all you need to do is to tell Academic Advisor to open it i.e. Open \*Application name\*. If the application is not installed on your phone Academic Advisor would prompt you to install that application first.

The next domain and the most important one is the smart answering capability. The output received will be in json format which will then be parsed using the jsoup library and then displayed or read out as mentioned earlier.

**1.1 OVERVIEW**

Project is aimed at making a virtual assistant which is artificially intelligent and can help the user in more than one way. Smart assistants typically these days have become really capable but our aim is to design as simple, interactive and unique application which can perform most of functionalities of a smart assistant, such as providing courses , reviews,placements of all colleges and also launching other applications on demand.

**1.2 SCOPE OF THE PROJECT**

**Requirements:** To be able deliver a smooth user experience with a chat like (threaded messaging) view to answer questions asked by the user.

* To be able to open up various applications when on users demand.
* To be able to read out the responses for better user experience.

**Limitations:**

* Will require internet connection as there is no offline database.
* Will not be able to keep up with context while having conversations.

**1.3 ORGANISATION OF THESIS**

Chapter 2 deals with literature survey where we see the comparisons of the respective project with the different smart assistants. It contains all the information studied and collected for developing this project .It covers all the areas and study necessary to understand the project completely.

Chapter 3 contains the problems of the existing system of our project and also contains proposed system of project. It evens contains the feasibility study (technical, economical, operational) and time and cost estimation of the project .Software requirement specification contains all the hardware and software requirements as per the IEEE standard.

Chapter 4 includes system design which contains architecture of the following project and uml diagrams. Chapter 5 contains the implementation part which generally contains the code which includes the necessary classes, interfaces, functions. Chapter 6 contains the testing the project where we test the project. Chapter 7 contains the output screens of the entire project. It shows all the features embedded in this project.

**2. LITERATURE SURVEY**

Personal assistants have become very common these days, but they come at the cost of high memory usage, we device a method to not just create a a.i based assistant but also keep it small in terms of memory usage. The application will not have a database of its own for the purpose of extracting the answers, thus making it very memory effective.

**2.1 TECHNOLOGIES:**

**IBM WATSON:**

With the IBM Watson™ Conversation service, you can build a solution that understands natural-language input and uses machine learning to respond to customers in a way that simulates a conversation between humans.

* The **application** sends the user input to the Conversation service.
  + The application connects to a *workspace*, which is a container for your dialog flow and training data.
  + The service interprets the user input, directs the flow of the conversation and gathers information that it needs.
  + You can connect additional Watson services to analyze user input, such as Tone Analyzer or Speech to Text.
* The application can interact with your **back-end systems** based on the user's intent and additional information. For example, answer question, open tickets, update account information, or place orders. There is no limit to what you can do.

**Implementation**

Here's how you will implement your conversation:

* **Configure a workspace.** With the easy-to-use graphical environment, set up the training data and dialog for your conversation.

The training data consists of the following artifacts:

* + **Intents**: Goals that you anticipate your users will have when they interact with the service. Define one intent for each goal that can be identified in a user's input. For example, you might define an intent named *store\_hours* that answers questions about store hours. For each intent, you add sample utterances that reflect the input customers might use to ask for the information they need, such as, "What time do you open?"
  + **Entities**: An entity represents a term or object that provides context for an intent. For example, an entity might be a city name that helps your dialog to distinguish which store the user wants to know store hours for.

As you add training data, a natural language classifier is automatically added to the workspace, and is trained to understand the types of requests that you have indicated the service should listen for and respond to.

Use the dialog tool to build a dialog flow that incorporates your intents and entities. The dialog flow is represented graphically in the tool as a tree. You can add a branch to process each of the intents that you want the service to handle. You can then add branch nodes that handle the many possible permutations of a request based on other factors, such as the entities found in the user input or information that is passed to the service from your application or another external service.

**3. SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

Google, apple and Microsoft are leading the trend of virtual assistants in the market. Apple's siri, Google's Google assistant and Microsoft's cortana are very powerful and intelligent. These assistants can do a lot of things and are very capable. Google has also introduced context based conversation which makes it the best in the market right now.

**3.2 DISADVANTAGES OF EXISTING SYSTEM**

* Traditional virtual assistants provide a very good response and are very efficient, although they take humongous space on the mobile devices.
* Effects the average user.
* Memory may not be a problem for flagship devices but for low end smart phones and mid rangers where expandable memory is not an option, it is really a big problem.
* All the aforementioned virtual assistants are limited in terms of the operating system, such as google assistant is available with the latest android nougaht(7.0) update only likewise siri and cortana also have the flaw of not being able to work on the older versions of their respective operating system.

**3.3 PROPOSED SYSTEM**

Although Google and apple are masters in the game of virtual assistance we aim at building a system which can be used by the masses. Our product is aimed for the people using mid range and low end android devices as it will work with android version 5.0(api21 and above), which makes up the majority of the market share for android.

Additionally our second most important aspect for improvement would be the memory, the application would be build around native android coding (using android studio) and a few external libraries hence making it very efficient in terms of memory.

**3.4 ADVANTAGES OF PROPOSED SYSTEM**

### Simple, efficient and visual appealing.

### Takes much less memory on device.

### Self-learning Ai, improves with time as more people interact with it.

### Works seamlessly on majority of android devices.

* Response time is very less.