Voice based weather Chatbot

Sharwari Joshi
(112950662)
Stony Brook University
New York, USA
sharwari.joshi@stonybrook.edu

Kondeti Rahul Reddy
(112961400)
Stony Brook University
New York, USA.
kondeti.rahulreddy@stonybrook.edu

Manvitha Vajrala
(112961002)
Stony Brook University
New York, USA
manvitha.vajrala@stonybrook.edu

***Abstract*—Every now and then, we have uncertainty with respect to weather. To address this problem, an android app is created, which interacts with the cloud and returns weather information. This app interacts with users in voice mode, has in-built Natural Language Processing(NLP) techniques from Google Dialog flow for an interactive session and returns with voice output to user's questions.This app saves time and energy from frequent visits to the browser.**

***Index Terms*—weather chat-bot, Google Dialogflow, mobile cloud computing, open weather API.**

I. Introduction

Recently, there is a significant increase in mobile subscriptions due to the rapid advance in mobile technology. On the other parallel, cloud technology is booming along with wireless networking. There are many significant problems which can be addressed with the combination of these technologies. Weather information is one such significant application which has many potential users who use it on a daily note. An Android app with easy GUI, accurate results which can ease the process of gathering information is the target of this project. The android app developed has the following unique advantages -

A. Compute efficiency:

By offloading, demanding the workloads and large data to the cloud infrastructure, the mobile device at the user's end can limit the resources. It can conserve the amount of processing power and data storage which is required.

B. Energy efficiency:

Since our app has a cloud at the back-end, we have the potential to create this application in an energy efficient way.

C. Thin Mobile clients:

As our app offloads significant work to the cloud, we can build this application targeting less powerful mobile devices. This gives us the ability to shrink the mobile handsets to the extent that they only process user interaction and offload all application processing work to the cloud.

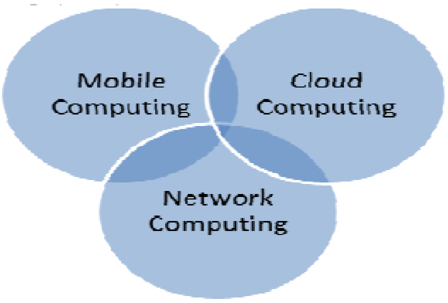


Fig. 1: The scope of Mobile Cloud Computing

II. Mobile Cloud Computing

A. What is mobile cloud computing ?

There are many definitions available for MCC. For example, MCC defined as "a rich mobile computing technology that leverages unified elastic resources of varied clouds and network technologies toward unrestricted functionality, storage, and mobility. It serves a multitude of mobile devices anywhere anytime through the channel of Ethernet or Internet regardless of heterogeneous environments and platforms based on the pay-as-you-earn-principle[1].

MCC is often a technology which involves the combination of cloud computing, mobile computing, and networking The given are the below primary features of MCC:

- Auto resource provision and de-provision.
- Scalability.
- Mobility,flexibility, and accessibility.
- Mobile cloud service connectivity.
- Virtualization.
- Multi-tenancy.
- Mobile cyber-security and privacy.
- Mobile utility billing and energy efficient.

B. Why Mobile Cloud Computing?

There are many motivation factors to choose Mobile Cloud Computing (MCC). Addressing the needs with the increasing processing power and battery life time of user devices, Cope with increasing services and application needs of most mobile users with low-end mobile devices, maximizing the resource sharing, re-usability of existing resources in computing are the few factors to choose Mobile Cloud Computing. MCC offers various advantages to both business entities and mobile users. MCC brings the following benefits to the business entities :

- Broader Reaching - Reaches all the devices through a browser or some other infrastructure and has the potential to reach the last user through the cloud computing.
- Expanding - It is projected to keep increasing with the ubiquitous computing. The business is not expected to see a plummeting scenario in the near future.
- Increasing Revenue - With the users increasing, resource sharing, new technology revenue is expected to increase.
- Reducing costs - with resource shared , using energy efficient solutions in mobile cloud platforms both energy and costs can be optimized.

Similarly , the MCC adds advantages to the mobile users as well. Few of them are -

- Increasing battery life- with complex computation offloaded to the cloud resources, and users handsets mostly responsible for exchanging data with cloud servers, the battery life performs better.
- Eliminating regional limitations with cloud connectivity.
- Enabling diverse virtualization of wireless networks, mobile devices, and connectivity to already existing cloud facilities.

III. Voice Based Weather Chatbot

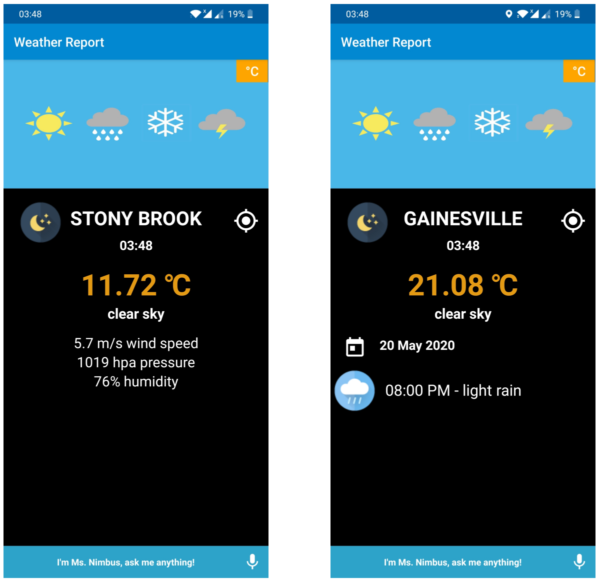
A. Introduction

In this project, we have developed a chatbot called Ms.Nimbus which gives us the weather of a particular area based on voice input from the user. It responds to various types of input phrases such as ‘weather’, ‘weather in New York’ etc., where the keyword being weather and the city name. Dialogflow is used to process different input types using Natural Language Processing. Other features include Celsius to Fahrenheit conversion, clock which shows the time of the location that is queried, an icon representing the weather condition for cloudy, sunny, etc., The weather data is collected from OpenWeatherMap API.

B. Working of the app

The user gives the voice input using the button ’Click here to Speak’ at the bottom of the application. Speech to text processing is done on the user input. There are many possible ways in which user can interact. There are direct questions such as ’Weather today’, ’What is the weather tomorrow’, ’weather in New York’ etc., which can be identified by the keywords like weather or city name easily. But for complicated questions such as ’will it rain tomorrow’, ’whats the probability of rain’, ’should I carry an umbrella’ is where Google Dialogflow really helps in processing the question using machine learning.

The request is then forwarded to the Open-Weather Map API to obtain the weather details. Then the response is converted to voice again and the output is received. It gives current weather details for the input related to general weather condition and if there is a future rain prediction the output says the predicted timings for the rain for the day.


03:48
19%
Weather Report
°C
I'm Ms. Nimbus, ask me anything!
03:48
19%
Weather Report
°C
I'm Ms. Nimbus, ask me anything!

STONY BROOK
03:48
11.72 °C
clear sky
5.7 m/s wind speed
1019 hpa pressure
76% humidity

GAINESVILLE
03:48
21.08 °C
clear sky
20 May 2020
08:00 PM - light rain

(a) Output 1

(b) Output 2

Fig. 2: Screenshots of the app

For the voice input ’weather in my location’ the app takes the device’s current location using the GPS and gives the output as shown in 2a) Output 1. For the input ’will it rain in Gainesville tomorrow’ the app responds saying ’yes it will rain at 2.00 PM tomorrow. The app also responds to various casual inputs such as common greetings. The button beside the location takes the user back to the current location from any state.

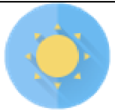
While the voice assistant says the temperature only, other information such as wind speed, pressure and humidity levels are also displayed on the screen. In addition to that the icon on the left changes according to the weather conditions such as sun for sunny weather, clouds for cloudy weather etc., Another feature of the app is that it can convert the scale from Celsius to Fahrenheit by tapping the button on the top right corner.

C. Features of the application

The app offers many advantages. These include -

- *1. Voice mode communication* - It is known that Voice is the most natural mode of communication and this application features with voice input as the default input.
- *2. Smoother experience* - When it comes to traditional chat bot, voice is less tedious and easy to use. Enabling a single button facilitates the user to communicate with the application. The end result is reflected on the screen and the same is also communicated with voice output, making it easy for the users to understand the results in low visibility modes.
- *3.On APP buttons to migrate to switch across Centigrade and Fahrenheit-* Across the world, Centigrade and Fahrenheit are the modes of measuring temperature and it varies with user preferences. For this purpose, the app provides an on-screen button to switch between these modes.

TABLE I: Weather Icons used in our Application

Serial No.	Weather Icons		
	Day Icon	*Night Icon*	*Description*
1			clear sky
2			few clouds
3			scattered clouds
4			broken clouds
5			shower rain
6			rain
7			thunderstorm
8			snow
9			mist

- *4.Text-Speech and speech-text conversion* - Uses Google text-to-speech converter. Google is a proven technology and most stable when it comes to conversion of the text-speech and speech-text modes.
- *5.Intelligent* - Google Dialog flow is integrated in the application. Google has come up with Dialogflow for natural language problems. This helps to address user specific questions .It provides a casual way of interacting with the bot.
- *6. Icons used* - We use the following list of icons to enhance the visual experience and improve the user experience of the app. We use the icons shown in Table 1 to represent the different weather conditions of the location.
- *7. Default Location* - The app by default picks up the current location as the default location.

D. Target Users/Application scenarios

Almost everyone. This application has a very wide range of audience. Weather has become a predominant factor in deciding the calendar of a person. Every individual who has smart phone and refers to internet for the weather information is target user for the application. It is useful for people having partial blindness or are visually challenged as well, as the app is voice dependant.

E. Technology/Packages used in the app

Few of the *dependencies* we have are:

- Multidex support is used so that we can use more than 64K methods in our project.We make use of it to avoid the 64k reference limit in the program. Compilers transform the source code into DEX files. This DEX file contains the compiled code used for running the app. The DEX file limits the total number of methods that can be referenced inside a single DEX file to 64K.
- Retrofit is a type-safe REST client for Android, Java and Kotlin developed by Square. The library provides a powerful framework for authenticating and interacting with APIs and sending network requests. This along with GSON library to serialize and deserialize JSON data makes downloading JSON or XML data from a web API very easy
- Butterknife support is used to inculcate Field and method binding for Android views. This package is useful for the following
  1) Eliminate findViewById calls by using @BindView on fields.
  2) Group multiple views in a list or array.
  3) Eliminate anonymous inner-classes for listeners by annotating methods with @OnClick and others.
  4) Eliminate resource look-ups by using resource annotations on fields.
- Algolia instant voice search tool is used to simplify the process of creating and integrating a complete search experience into the sites and/or applications with overlay which obtains user's voice permission and input as text.
- Apache Commons Codec package provides encoder and decoders for various formats also contains a collection of phonetic encoding utilities which we use in our application.

We have imported a few *packages* in our application, some of the major ones are explained:

- java.utils -Contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes (a string token-izer, a random-number generator, and a bit array).
- google.gson -This package provides the Gson class to convert Json to Java and vice-versa.
- java.io.IOException - Package for Exception Handling.
- ai.api - Package for linking Dialogflow
- com.google.android.material- Material Components for Android is available via Google's Maven Repository.
- com.google.android.gms- A collection of Google applications and APIs that help support functionality across devices. These apps work together seamlessly to ensure

that the device provides a great user experience right out of the box.

- android.annotation -AndroidX is the new extension libraries for backward compatibility support. A new feature backward compatibility support will be addressed in AndroidX in the future
- android.os.Looper-Android Looper is a Java class inside the Android UI which works along with the Handler class to process UI events such as button clicks, screen redraws and orientation switches. They are also used to upload content to an HTTP service, resize images and execute remote requests.

F. API- Open-weather Map API

Openweather is a small IT company, established in 2014 by a group of engineers and experts in Big Data, data processing, and satellite imagery processing. The headquarters is in the UK, and an office in the US, and the development team in Latvia (EU).

Advantages of the API-

- Simple, fast, and utterly convenient API for developers
- Current weather, historical weather and forecast
- Data availability for any Geo-location
- Data sources are NOAA (National Oceanic and Atmospheric administration) , Canadian Environment, and raw data from weather stations
- Open for integrating other data sources

G. Memory usage

Random-access memory (RAM) is a valuable resource in all software development environments, but it's more valuable on a mobile operating system where physical memory is often constrained. Our application is a light weight application and it uses 5.26 MB of internal memory.

H. Conclusion and Future Scope

We did not observe any significant problem with application. However, there is always scope for improvement. The application can be upgraded to support many different languages thus can be made available to non-English speaking population as well. The application can be made to alert the user if there are severe and/or sudden change in weather conditions. Since the app currently uses trial version of the API, this enables us to exchange only 1000 calls/day. If the users increase significantly, the trial version is to be replaced with the premium account to cater to a wider audience.

Acknowledgment

We would like to thank Professor Shan Lin for teaching us Mobile Cloud Computing course and helping us learn many things in the curriculum with a wide variety of labs, and assignments. Though the course has moved online due the COVID-19 pandemic, Professor Shan Lin has put many efforts in helping us cope with the newest mode of online education and made sure we were never short of resources in the completion of the project.Also, we would like to thank the TA Hongkai Chen for his time in helping us with labs and tutorials.

References

1) Zohreh Sanaei, Saeid Abolfazli, Abdullah Gani, Muhammad Shiraz, "SAMI:Service-Based Arbitrated Multi-Tier Infrastructure for Mobile Cloud Computing", IEEE MobiCC'12 conference, MobiCC 2012:IEEE Workshop on Mobile Cloud Computing, Beijing, China
2) Han Qi, Abdullah Gani, "Research on Mobile Cloud Computing:Review, Trend and Perspectives", https://arxiv.org/ftp/arxiv/papers/1206/1206.1118.pdf
3) Abdul-Kader, SA and Woods, JC (2015) 'Survey on Chatbot Design Techniques in Speech Conversation Systems.' International Journal of Advanced Computer Science and Applications, 6 (7). ISSN 2156-5570
4) https://en.wikipedia.org/wiki/Dialogflow
5) https://openweather.co.uk/about
6) Javadoc 8 -https://docs.oracle.com/javase/8/docs/api/java /util/package-summary.html
7) https://www.vogella.com/tutorials/AndroidButterknife/ article.html
8) https://cloud.google.com/dialogflow/docs
9) https://developer.android.com/studio/build/multidex
10) https://www.algolia.com/doc/guides/getting-started/ how-algolia-works/