

INTELLIGENT VOICE ASSISTANT

Bachelor Thesis

Spring 2012

*School of Health and Society
Department Computer Science
Computer Software Development*

Intelligent Voice Assistant

Writer

Shen Hui

Song Qunying

Instructor

Andreas Nilsson

Examiner

Christian Andersson

INTELLIGENT VOICE ASSISTANT

School of Health and Society
Department Computer Science
Kristianstad University
SE-291 88 Kristianstad
Sweden

Author, Program and Year:

Song Qunying, Bachelor in Computer Software Development 2012

Shen Hui, Bachelor in Computer Software Development 2012

Instructor:

Andreas Nilsson, School of Health and Society, HKr

Examination:

This graduation work on 15 higher education credits is a part of the requirements for a
Degree of Bachelor in Computer Software Development (as specified in the English translation)

Title:

Intelligent Voice Assistant

Abstract:

This project includes an implementation of an intelligent voice recognition assistant for Android where functionality on current existing applications on other platforms is compared. Until this day, there has not been any good alternative for Android, so this project aims to implement a voice assistant for the Android platform while describing the difficulties and challenges that lies in this task.

Language:

English

Approved by:

Christian Andersson

Date

Examiner

Table of Contents	Page
Document page	I
Abstract	I
Table of Contents	II
1 Introduction	1
1.1 Context	1
1.2 Aim and Purpose	2
1.3 Method and Resources	3
1.4 Project Work Organization	7
1.5 Acknowledgements	8
2 Analysis	9
2.1 Information Retrieval	9
2.2 Theory Model	11
2.3 Alternative Models/solution	15
2.4 Environmental Consequences	20
3 Realization	21
3.1 Choice of Solution	21
3.2 Equipment/ Choice of Materials	30

3.3 Problems and Solutions	31
4 Results	34
4.1 Design	34
4.2 Functioning	36
4.3 Operation and Maintenance	39
5 Conclusions	45
6 Recommendations for Further Work	47
6.1 Design Improvements	47
6.2 Additional Functions	47
6.3 Database Capacity	47
6.4 Humanized Voice Recognition	48
6.5 Improved Interface	48
7 References	49
8 Appendix A Figure	50
9 Appendix B Code	52

1 Introduction

1.1 Context

This project is based on Android application development and provide personal assistant using voice recognition or text mode operation. This program includes the functions and services of: calling services, text message transformation, mail exchange, alarm, event handler, location services, music player service, checking weather, Google searching engine, Wikipedia searching engine, robot chat, camera, Bing translator, Bluetooth headset support, help menu and Windows azure cloud computing.

As it integrates most of the mobile phone services for daily use, it could be useful for getting a more convenient life and it will be helpful for those people who have disabilities for manual operations. This is also part of the reason why it has been chosen as the degree project.

This project is originated from a popular application from Apple called “Siri” [1]. This application was released on the date when the iPhone4S was published. This application is very interesting, easy going and convenient, with wide real world usage and large developing potential. This application is not limited by different generations and occupations, and can be applied to many industries that we have in the real world. For instance, the voice assistance is very useful for personal assistants, direction guides or driving, helps among the disabled community, and so on.

This is a short description about “Siri” from Wikipedia to illustrate the voice product: “Siri” an intelligent personal assistant and knowledge navigator which works as an application for Apple's iOS. The application uses a natural language user interface to answer questions, make recommendations, and perform actions by delegating requests to a set of web services. Apple claims that the software adapts to the user's individual preferences over time and personalizes results, and performing tasks such as finding recommendations for nearby restaurants or getting directions.

1.2 Aim and Purpose

According to the overall description in the context, the purpose of the project is to develop an Android application that provides an intelligent voice assistant with the functionalities as calling services, message transformation, mail exchange, alarm, event handler, location services, music play service, checking weather, searching engine (Google, Wikipedia), camera, Bing translator, Bluetooth headset support, help menu and Windows azure cloud computing.

Many years ago, software programs were developed and run on the computer. Nowadays, smart phones are widely used by all people. About 35 percent of the Americans have some sort of Smartphone. This shows that the market is increasing fast and there are also more capabilities for Smartphone because of this wide use. [2]

Therefore, the software development on the Smartphone is very promising. The operation modes on the Smartphone are by working with gestures and through the keyboard. It is not a convenient way for users with completely manually input. The common way of communication used by people in daily life is through the speech. If the mobile phone can listen to the user for the request or handle the daily affairs, then give the right response, it will be much easier for users to communicate with their phone, and the mobile phone will be much more “Smart” as a human assistant.

This project is focusing on the Android development over the voice control (recognition, generate and analyze corresponding commands, intelligent responses automatically), Google products and relevant APIs (Google map, Google weather, Google search and etc), Wikipedia API and mobile device references ranging from Speech-To-Text, Text-To-Speech technology, Bluetooth headset support and camera; advanced techniques of Cloud computing, Multi-threading, Adobe Photoshop image editing skills. As all those functionalities and services for the project have been explained, the main structure and construction of the project has been basically illustrated with its goals.

1.3 Method and Resources

This project mainly concerns the work on Android application development; request calling between different Android applications, human-mobile phone interaction, database creation and management, the program will reference a lot of APIs from Google, Wikipedia, and Android development skills.

Apart from the project itself, there is also some investigation works on the existed products in this area and the tendency of voice product, personal assistant developing. Two products were mainly investigated that are popular and representative, the English product of “Siri” and the Chinese product of “iFly” [Chinese name: 讯飞语点 [3]]. The investigation focus on how those ideas originated; what functionalities and services they have; how they provide these services to the customers; test the product and related functions to get the architect, structure, logical algorithms of those products; how they spread and promote the product in marketing; and how they refine and upgrade the products from different versions. Table-1 shows the comparison about some basic functions between “Siri” and “iFly”.

Function	Siri	iFly
Call Service	Yes	Yes
SMS Message Service	Yes	Yes
Open Application	No	Yes
Web Search Service	Google Search Engine	Baidu Search Engine
Reminder	24h	Unlimited
Music Play	Local Library	Local + Remote Library
Command Text Modify	Yes	No
Language	English & French & German & Japanese	Chinese

Table-1

In addition, it has been investigated that the developing tendency in this area based on the internet information and online video of conference from Apple, Android and some other Chinese products.

INTELLIGENT VOICE ASSISTANT

To learn how they are going to develop the products in this area from all possible aspects and the potential developing factors.

For a better and efficient development, the project is carried out over the XP (Extreme Programming) model. Extreme programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles (timeboxing), which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted. [4]

The developments will on the small cycle model repeatedly, every cycle will have analysis, design, implementation and test. Figure-1 somehow shows how to follow the XP develop model.

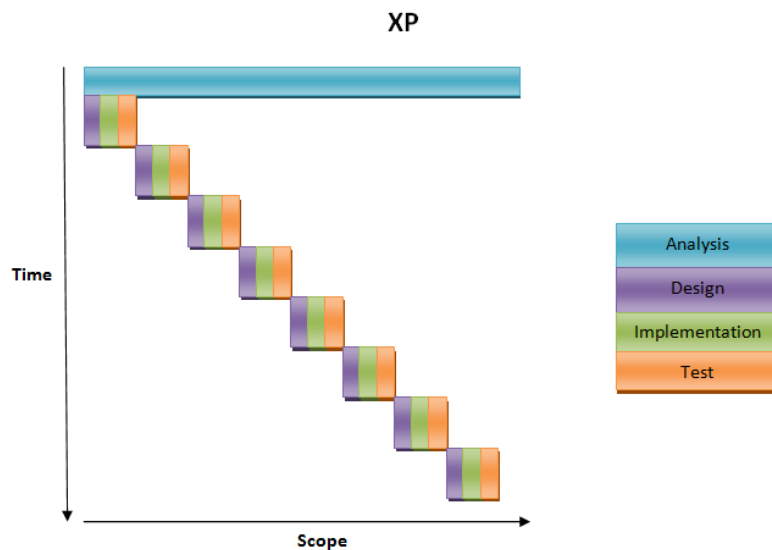


Figure-1

The total work have been defined in one hundred percentages, the list show how many percentages developers finished in each week; totally it has been worked for eight weeks to complete the project. In addition, the chart also shows how much that has been completed for the different part of the development from the requirement to the test. Figure-2 figures out the process and the progress that has been finished in each phase to complete the project.

INTELLIGENT VOICE ASSISTANT

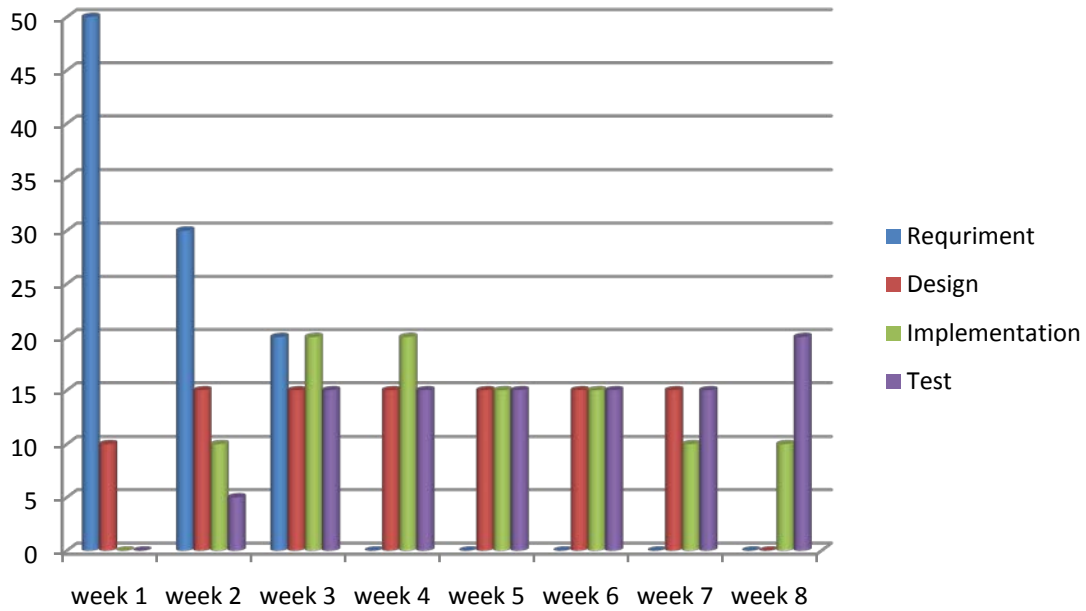


Figure-2

Figure-3 shows the process of the completion percentages with the timeline for each perspective includes requirement, design, implementation and test. Figure-3 presents the efficiency and completion of the project from all aspects.

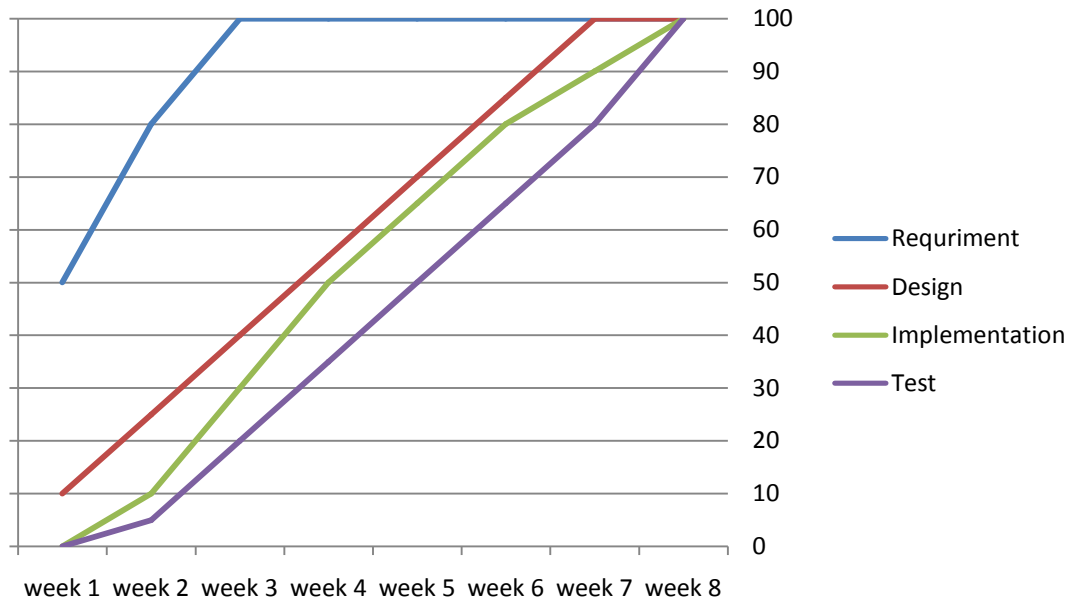


Figure-3

INTELLIGENT VOICE ASSISTANT

Figure-3 also indicates the tendency and expected working process of the project work. In addition, the efficiency and evaluation speed of the project can be seen from it. And most important is the diagram points out how the project will be completed in time.

1.4 Project Work Organization

The project work is organized based on the actual task for the designing, implementation, test and optimization. As it has been primary planned, each of the developers worked 5 days a week; 3 days for implementation, and 2 days for testing and summarizing the work, totally it is 8 weeks' work. Apart from the designing, implementation and testing, developers also defined the work plan every time before the implementation and improve the project after the accomplishment of each individual section.

Developers communicate though the MSN, Facebook and Skype for sharing the ideas and discuss the project. Data statistics and relative materials is collected and shared through Dropbox. Mostly the work was done by pairing programming, that is, every time developers made a meeting and set together for designing, figure out a valid solution and doing the implementation together.

The high-level designing and the framework was done together, and the individual implementation of functions was assigned to different developers, but the developer was not only caring his own part, but also considering the whole program.

1.5 Acknowledgements

As it requires an Android phone testing and running the program, the Android phone is quite necessary. The school provided a Sony Ericsson phone with the Android operating system, but the phone was in a 2.0 version which is too low to implement the project. Thanks to WANG LINLIN for lending the phone and it can be frequently used for the project development.

2 Analysis

2.1 Information Retrieval

As this program includes the functions and services of: calling services, text message transformation, mail exchange, alarm, event handler, location services, music player service, checking weather, Google searching engine, Wikipedia searching engine, robot chat, camera, Bing translator, Bluetooth headset support and help menu. The list below indicates the information and the requirements of each individual function.

The program has two modes to well fetch the services and functions. The program will start with voice mode as its primary mode to provide the voice assistant, but the user can select switching to the text mode if he or she is not well working with the voice mode or the surrounds don't support the voice recognition well.

- Calling service, the application should allow the users to give a call to the person in the contacts. By giving a correct command with the calling request to a stored person, the Android phone should successfully direct to the number of the person requested.
- Text Message transformation, customers are able to send the SMS to a specific person in the contacts. By giving a correct command contains the messaging request keyword together with the destination person, the message should be sent to the destination immediately.
- Mail exchange, customers are able to send the mail to the person with mail address in the contacts. By giving a correct command contains the mail request keyword together with the destination person; the mail should be received by the recipient after it has been sent.
- Alarm, as a basic function on the mobile phone, it is frequently that users need to set the alarm to a specific time. The user could set the alarm through the request with the given time.
- Event handler, the application should allow the user to set as many events as they want. Customers with the event content should be stored and available for the user to check, modify and delete.
- Location services, location services provide the functions for the user to check the current location or find the direction to a destination. The user should get an easy to understand map with the locations or routes depending on the category of the request.

INTELLIGENT VOICE ASSISTANT

- Music player service, the music player offers the services to the user to play a named or a randomly picked song in the pre-stored song list on the mobile phone. And it could be stopped when the user wants to terminate it.
- Checking weather, the user could check the weather in any place. In addition, the weather is returned with the temperature and humidity; the user could also check the weather for current day, tomorrow or in next four days.
- Google searching engine, the search engine enable the user to search anything on Google. The search engine will give result list back and displayed on the browser.
- Wikipedia searching engine, the search engine enable the user to search anything on Wikipedia. The result is given back on the web browser with the searched content on Wikipedia.
- Robot chat, this is the robot chat which provides fun to the user. After enter the chat mode, a text response will given by the mobile phone whenever the user speaks to it.
- Camera, the camera function will call the camera on the mobile phone to take a picture of the current view, the picture will be stored in the Gallery for later viewing and operation.
- Bing translator, the translator will translate the original text in the object language the user wants. There have been 25 object languages stored and the original text should be English.
- Bluetooth headset support, since it is not possible to do the voice recognition while the music player is playing or the surroundings are noisy; the Bluetooth headset support makes it possible to speak to the headset rather than the mobile phone if the user enables it.
- Help menu, the user can choose the help menu if the user doesn't know how to work with the functions. The help menu gives the list of functions with the examples and explanation of how to work with different functions as well.

2.2 Theory Model

The project is based on the theories related to various aspects of software engineering principles and software development model; Java programming skills and Android tutorials, Database management and network communication technologies.

The database and the web service in this project are put on the windows azure cloud; developers will never be required to run the web service and database locally. The cloud platform will handle the execution and maintenance. Hence, cloud computing is an important concept and theory guide the development.

- Cloud computing: **Cloud computing** refers to the delivery of computing and storage capacity as a service to a heterogeneous community of end-recipients. The name comes from the use of clouds as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts services with a user's data, software and computation over a network. It has considerable overlap with software as a service (SaaS). [5]
- Software engineering principles

Extreme programming will direct the development process of the project, it focus on the development cycle of defining the requirement, corresponding design and test, integration and simplicity; during the development, there should always be working in pair programming, as well as doing the revision control, calculate the velocity and efficiency.

Extreme programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles (timeboxing), which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted. [6]

- Java programming: java API and reference, which is helpful in guide programming in eclipse and construction of the framework, and the completion of the functions.

Java is a programming language originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write

INTELLIGENT VOICE ASSISTANT

once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java is currently one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users. [7] [8]

- Android: this project is mainly focus on the Android development to enable most of the Android functions for daily use ranging from check the weather to check location, and weather services and etc, Android reference will be the theory promote the development of the project and related applications, [9] [10] [11]
- Database management: The program will always work with different databases like Microsoft SQL Server and MySQL Server. Cloud database to handle the data storing, updating, and retrieving. The following chapters indicate the usage and information of each database. Through this information, it can be obtained of the advantages and disadvantages of each database.

The data stored in this project is not so much and complicated as in a corporation; therefore, each of the databases mentioned can well meet the requirement of the data storage, updating or be dropped as well. However, the choice of the databases still depends upon the convenience while considering the advantage and disadvantage. As the commands are received by the program, the command should be analyzed with the database, MS SQL Server has been the best choice since it provides the method to search in the content which is convenient to identify the keyword, keyword category and keyword content, this advantage is contributed by the method CHARINDEX.

Microsoft SQL Server 2012 is a cloud-ready information platform that will help organizations unlock breakthrough insights across the organization and quickly build solutions to extend data across on-premises and public cloud, backed by mission critical confidence [12]

The MySQL database has become the world's most popular open source database because of its high performance, high reliability and ease of use. It is also the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python.) Many of the world's largest and fastest-growing organizations including Facebook, Google, Adobe, Alcatel Lucent and Zappos rely on MySQL to save time and money powering their high-volume Web sites, business-critical systems and packaged software.

MySQL runs on more than 20 platforms including Linux, Windows, Mac OS, Solaris, IBM AIX, giving you the kind of flexibility that puts you in control. Whether you're new to database technology or an experienced developer or DBA, MySQL offers a comprehensive range of database tools, support, training and consulting services to make you successful. [13]

INTELLIGENT VOICE ASSISTANT

SQL Azure is a highly available and scalable cloud database service built on SQL Server technologies. With SQL Azure, developers do not have to install setup or manage any database. High availability and fault tolerance is built-in and no physical administration is required. SQL Azure is a managed service that is operated by Microsoft and has a 99.9% monthly SLA. [14]

- Network communication technologies

The communication in this program is based on the predefined protocol, the communication within the program is implemented in following the pre-defined protocol, the other main part of the communication is between the Android program in eclipse and the cloud platform, this will be done by working with URL, WSDL file. Figure-4 shows some knowledge of cloud platform, URL and WSDL. [15]

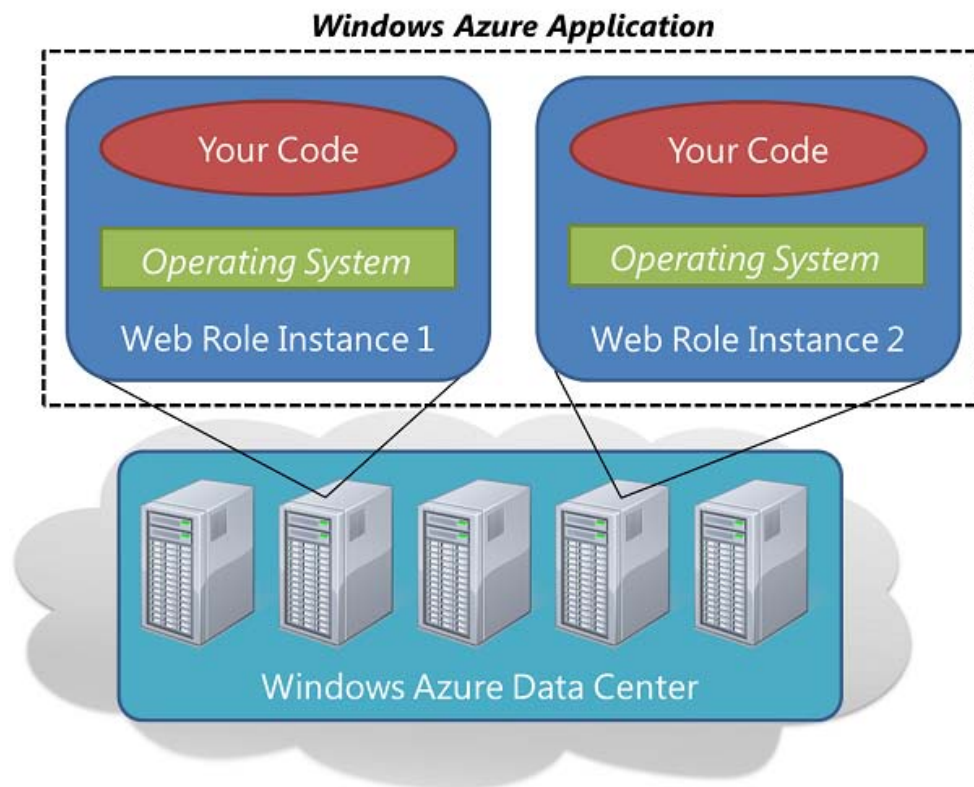


Figure 4

The WSDL describes services as collections of network endpoints, or ports. The WSDL specification provides an XML format for documents for this purpose. The abstract definitions of ports and messages are separated from their concrete use or instance, allowing the reuse of these definitions. A port is defined by associating a network address with a reusable binding, and a collection of ports defines a service. Messages are abstract descriptions of the data being exchanged, and port types are abstract collections of supported operations. The concrete protocol and data format specifications for a

INTELLIGENT VOICE ASSISTANT

particular port type constitutes a reusable binding, where the operations and messages are then bound to a concrete network protocol and message format. In this way, WSDL describes the public interface to the Web service. [16]

2.3 Alternative Models/solution

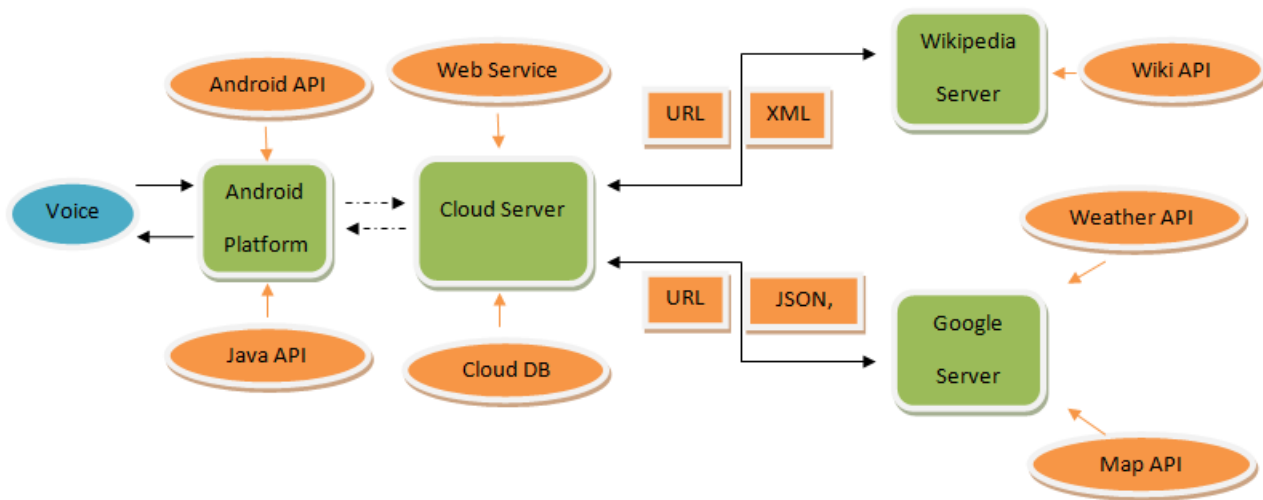


Figure-5

The architecture (see figure-5) is depending on the developing simulation. The architecture diagram is not only directed the development of the project, but also figure out the main fields and technique references related to implementing the project with expected functions.

- The voice input will be firstly recorded by the Android phone.
- The voice will be recognized by the Android applications by using the Android API and Java API. A recorded text will be generated and send to the cloud server or Android applications depending on the command.
- The cloud server will decode the received text with the Java API, references, and predefined database, then decide the following procedures that should be executed.
- A command will be generated into a URL by the cloud server and sent to the specific server (Google server, Wikipedia server).
- The server which receives the request will using the specific API of Wikipedia API, Google API to generate the response in XML or JSON format.
- Cloud server will obtain the XML/ JSON response file and transform to a specific response which will be led to the Android application.
- The Android application will generate the audio output to the customers with the mobile speaker.

INTELLIGENT VOICE ASSISTANT

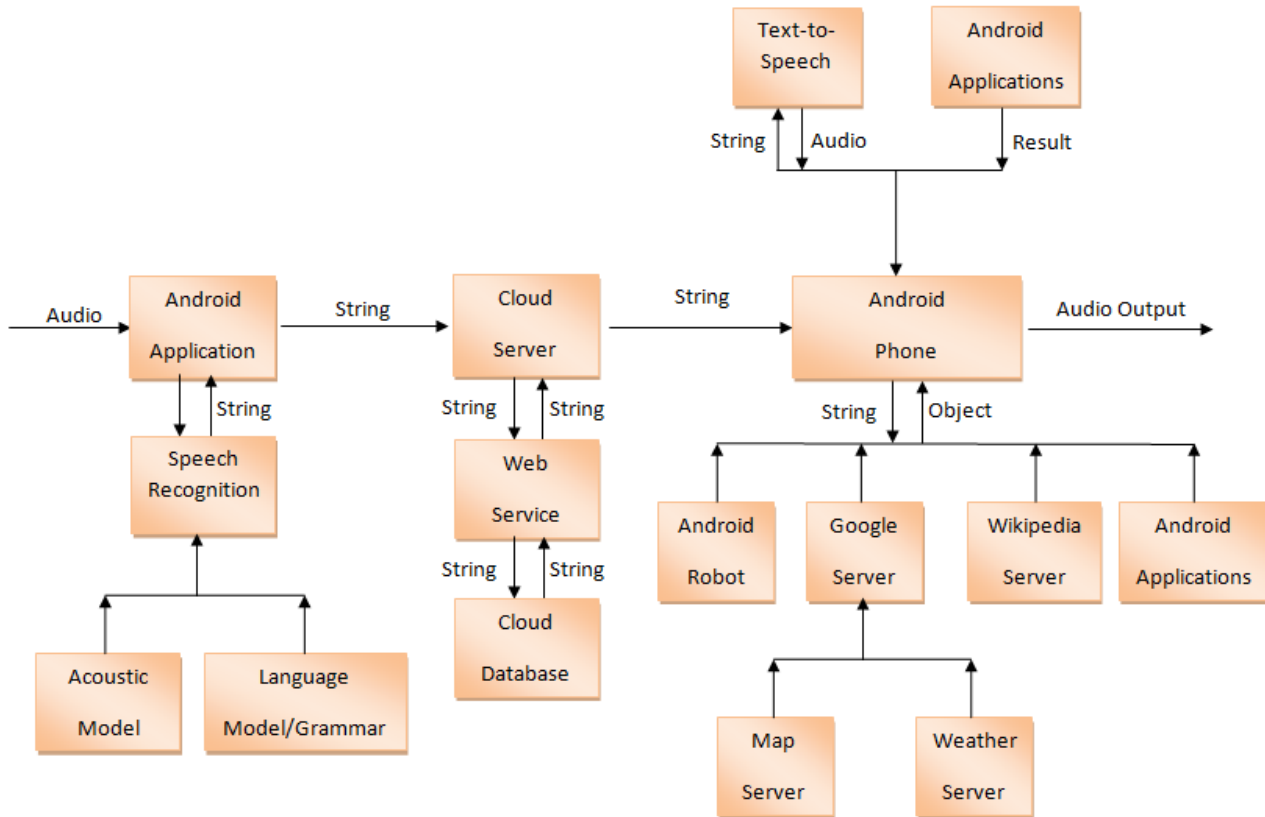


Figure-6

The configuration diagram (see figure-6) explains all the develop methods, the real strategies and development process with the core techniques that are used in the project work. Most of the applications and useful mechanisms are included.

- When the Android application received the audio input, the speech recognition will record the voice with acoustic model and language model/Grammar, a string reflects the audio input will be delivered to the java server.
- Whenever the string input is received by the cloud server, it will be passed down to the web service, further decoded with the cloud database which includes all the possible commands.
- While the meaning of the string has been detected, the corresponding command is transmitted to the specific application/ server program depending on the command.
- The application/server program generates the result and response back with an object to the Android phone according to the command and relevant data.

INTELLIGENT VOICE ASSISTANT

- The Android phone generates the response into the audio output that delivered to the customer, or the operations that should be carried out to complete the expected result.

List below indicates the solutions to each of the functions in this program.

- Calling service, when the application receive the command for making phone calls to someone, with the name it will first check the contacts and find the phone number of the person that contain the given name, then make the phone call by directing to that number. By checking through the contact list and find the phone number, the calling can be dialled out by calling the system call action intent in Android.
- Text Message, when the application receives the command for sending a SMS message to someone, it will first check the contacts and find the phone number of the person contain the given name, and then send the message successfully. By checking through the contact list and find the phone number, the message can be sent out by calling the system message send action intent in Android. There are two alternative solutions to complete it:
 - By capturing the content and name, directly send the message content to the number of the person's name.
 - By capturing the content and name, switch to the message sending function interface on the mobile phone with the person and content, the user can decide if send it or not depend on the captured content is correct or not.
- Mail exchange, when the application receives the command for sending email to someone, it will first check the contacts and find the email address of the person which contains the given name, and then send the email successfully to the destination. By checking through the contact list and find the email address, the email can be sent out by calling the system email send intent in Android. There are two alternative solutions to complete it:
 - By capturing the content and name, directly send the email content to the address of the person's name.
 - By capturing the content and name, switch to the email interface on the mobile phone with the person and content, the user can decide if send it or not depend on the captured content is correct or not.
- Alarm, when the application receives the request of setting the alarm to a valid time, the program will get the time with dedicated hour and minute, the set the alarm to that specific time. The alarm can be set by the system alarm manager with the time. There are two forms to get alarm work:

INTELLIGENT VOICE ASSISTANT

- Set the alarm to the given time and alarm will be activated when the time comes up.
 - Set the alarm to the given time and alter appears when the time comes up, the user decide to stop the alarm or not.
-
- Event handler, when the application receives the command of setting an event to a valid time, the event will be stored and can be viewed later. The user can check one/all events and choose to modify, delete the event, this function can be achieved by starting the sub event application in the program with the title, content, and time from the user's input.
 - Location service, the location service can be categorized in two forms depend on the request, the service will either return the current location or the route from the current location to the destination depending on what is required by the user.
 - If the user wants to check the current location, the program will get and display the location through mobile phone GPS module in a map.
 - If the user wants to get the route from the current location to other city, the program will first check the GEO information for the destination, then send current and destination position to the Google map server and get the route info then display it in a map with highlighted route.
 - Music player service, when the program receive a command of play a song, it will firstly check whether the command contains the song's name or not, if the name appeared in the command, it will get the path of the song and play the song, otherwise the program will randomly pick a song from the media library and play it. Every time when the program is loaded, it will called the system to collection all the media files on the mobile then save then into the library, if the user wants to play the song, it will start the music play service and play the music in background.
 - Checking weather, when the program receives the request of checking weather, it will firstly check whether the command contains the location information or not, if the command contain to a location name, then it will check if the command have a date value included or not. The Google weather service is used to accomplish the weather checking.
 - If today is detected, the weather for today in the specified location will be presented with the temperature, condition, humidity, wind direction.
 - If no today is detected, the weather for next four days in the specified location will be presented with the highest and lowest temperature, condition.

INTELLIGENT VOICE ASSISTANT

- Google searching engine, when the command is detected with the action to use the Google search engine. The program will generate the URL of the Google search link with the given search content and then start the system's internet browser with this link, finally gives the result back on the web browser.
- Wikipedia searching engine, when the command is detected with the action to use the Wikipedia engine. The program will generate the URL of the Wikipedia link with the given search content and then start the system's internet browser with this link, finally gives the result back on the web browser.
- Robot chat, if the command contains the words can be understood to enable chat, the program will enter the chat mode and give the user a predefined response to each sentence give by the user. The chat mode continues work until the chat is finished by the correct command.
- Camera, when the application receives the command to start the camera, the program will start the intent to enable the camera preview, after the photo has been captured, it will be saved in to the SD card memory and notify to the gallery for updating.
- Bing translator, the program will firstly detect the command with the destination language code and the content; generate a URL that contains the original language, the destination language and content. Then the URL will be opened and receive the result from the Bing translator for presenting to the user.
- Bluetooth headset support, the Bluetooth mode can be automatically activated by the user plug in the Bluetooth headset to the mobile phone. The program will enable the Bluetooth button when it receives a broadcast from the system. When the connection is enabled between the mobile phone and the Bluetooth headset, the audio manager of the mobile phone can be set to the Bluetooth headset mode and use the Microphone and speaker on the headset.
- Help menu, the user can open the help menu by selecting on the main option menu or given the help command, the help menu is designed by a main help menu and a list of sub menus, each sub menu is corresponded to one function with the explanation and examples to show how it works, each menu is designed in an individual activity.

2.4 Environmental Consequences

This program is green to the environment and no pollution will be generated by the software or hardware. During the development, the process will not do any harm to the surrounding environment since it is software development on the computer. The following list contains all the software, hardware, develop platform, developing process we use in this project. Hence it can prove that no pollution is created by these rephrases.

Develop platform: Microsoft Windows 7, Windows Azure Platform.

Develop tools and environment: Java [™], JDK, Eclipse IDE, Android SDK, ADT Plug-in, ADV, and Plug-in for Eclipse, MySQL query browser, DB-Designer, SQL Server Management Studio, and Microsoft Visual Web Developer 2010 Express.

API and reference: Java API, Android API, Google API (Google Map, Google Weather), Wikipedia API, SQL tutorial, UML reference, JSON, XML, Cloud computing, multi-threading techniques, .net framework 4.0.

Software application on Android phone: Android Internet explorer, Google voice recognize, TTS Service Extended, Alarm, Mobile phone calling services, text message services.

Support application: Adobe Photoshop CS5, Meitu (Chinese).

Hardware support: Android phone [HTC/Samsung], PC.

Developing model: XP (Extreme programming)

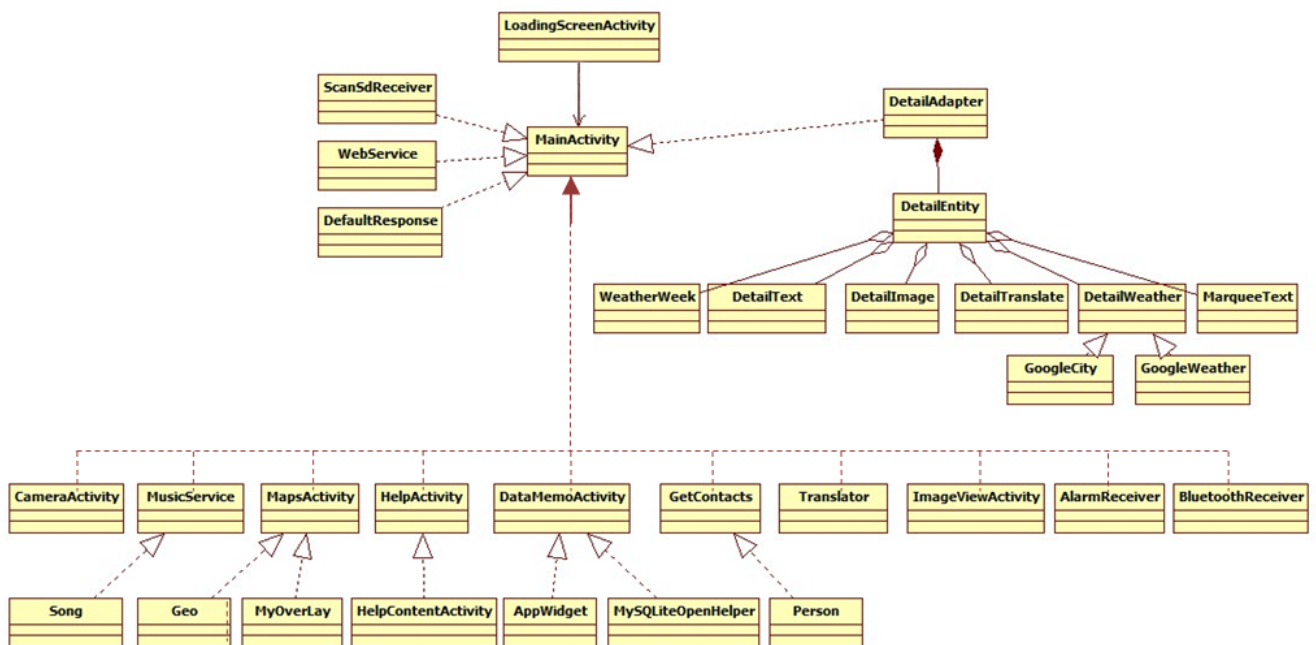
3 Realization

3.1 Choice of Solution

This chapter explains the actual solution to construct of the whole program. The functions include: Calling services, message transformation, mail exchange, alarm, event handler, location services, music play service, checking weather, searching engine (Google, Wikipedia), camera, Bing translator, Bluetooth headset support, help menu and Windows azure cloud computing.

As it has been illustrated in 2.3, the whole construction of the program mainly cover Android application development, the database design, web service and cloud computing.

The Android application, which implements and presents all the functions, is constructed in Eclipse with Android development references. The program implements voice recognition to capture the incoming requests. Creating the main activity and building each of the functions, implementing the logic to construct the whole program. Further by fetching the web service on the Windows Azure Cloud, the command can be analyzed with the storage on the database; corresponding responses will be directed to specific function in the program. Figure-7 shows the overall design of the program through UML.



INTELLIGENT VOICE ASSISTANT

Figure-7

The database is designed with MS SQL server. By creating different tables to store the data in different category, the data can be well stored, retrieved, updated or deleted. To well support the data process in web service, the database is uploaded on the Windows Azure Cloud.

Web service, the web service is implemented in C# since it is placed on the Windows Cloud. The web service takes the incoming request as the parameter; analyze it by check the keyword contained in the request, and give correct response to the program. The same with the database, the web service is uploaded on the Windows Azure Cloud.

Cloud computing, Windows Azure has been chosen as the cloud platform since it provide a three months' free use with a registered account. By establishing the database and creating the web services for intended use, the database and the web service are uploaded on the Cloud and, the data processing are going as cloud computing.

The following indicate the design for each individual function in this program.

- The programs start with the voice recognition, by implementing the RecognitionListener, it will capture the text every time the speaker speaks to it, then the generated text and send to the cloud (see Figure-8).

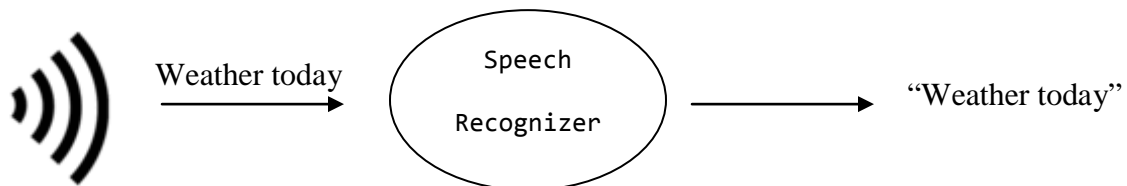


Figure-8

- The azure cloud which is an open cloud platform, where the software, database, web service can be placed there for future use. In this program, the web service and database are uploaded on the azure cloud for executing and maintenance (see Figure-9).

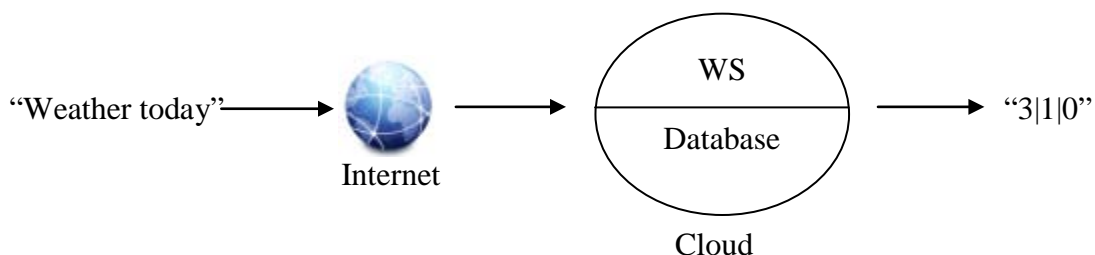


Figure-9

- The web service is written in C# and connects to the cloud database, the captured text will firstly be sent to the cloud as a parameter to call the analysis method and the method will check the keywords from the database keyword library. When the keyword is identified, it will implement different operations depending on the keyword category and give corresponding response that follows the protocol. (See Table-2 from Appendix A)
- The database was created in MS SQL server and uploaded on windows azure cloud through windows azure database manager, it defines the different keyword categories depending on the functions, the keywords for each category and response for different keywords category. (see Figure-10)

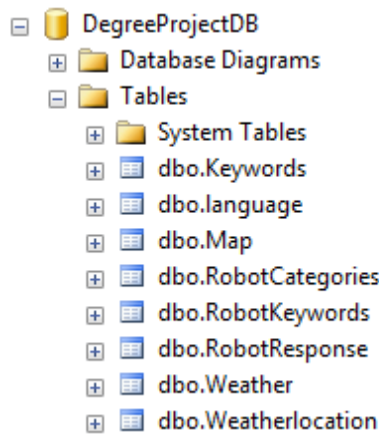


Figure-10

The database has been designed into eight tables, each table contain different information for each category, the “Keywords”, “Language”, “Map”, “Weather”, “Weatherlocation” tables is used for the application to identify the different command, “RobotCategories”, “RobotKeywords”, “RobotResponse” table are used for the robot chat. The following chapters describe each of the table and what is intended usage.

“Keywords” table: (see Figure-11)

		<table border="1"> <thead> <tr> <th></th> <th>KeywordsID</th> <th>KeywordsContent</th> <th>KeywordsCategory</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>weather</td> <td>3</td> </tr> <tr> <td>2</td> <td>2</td> <td>temperature</td> <td>3</td> </tr> <tr> <td>3</td> <td>3</td> <td>climate</td> <td>3</td> </tr> <tr> <td>4</td> <td>4</td> <td>where</td> <td>2</td> </tr> </tbody> </table>		KeywordsID	KeywordsContent	KeywordsCategory	1	1	weather	3	2	2	temperature	3	3	3	climate	3	4	4	where	2
	KeywordsID	KeywordsContent	KeywordsCategory																			
1	1	weather	3																			
2	2	temperature	3																			
3	3	climate	3																			
4	4	where	2																			

Figure-11

INTELLIGENT VOICE ASSISTANT

The “keywords” table contains three columns to present the data, the “KeywordsID” column is used to specify the different keyword in its unique ID, “KeywordsContent” column is used to save the keyword info and the “KeywordsCategory” classify the content into different category.

“Language” table: (see Figure-12)



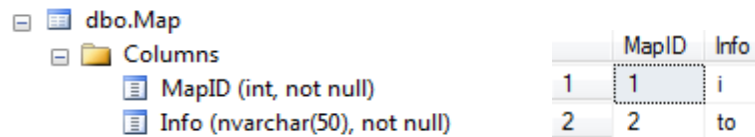
The screenshot shows the 'dbo.language' table structure in SQL Server Enterprise Manager. The table has three columns: 'lanaguageID' (Primary Key, int, not null), 'lanaguageDescription' (nvarchar(50), not null), and 'languageCode' (nvarchar(50), not null). To the right, a data table shows the following rows:

	lanaguageID	lanaguageDescription	languageCode
1	1	traditional chinese	zh-cn
2	2	simplified chinese	zh-cn
3	3	japanese	ja
4	4	italian	it

Figure-12

The “Language” table is used to discern the language and translate it to objective language code, the “languageID” column is used to specify the different language in its unique ID, “languageDescription” is used to describe the language and the “languageCode” is used to change the text-based language in to language code.

“Map” table: (see Figure-13)



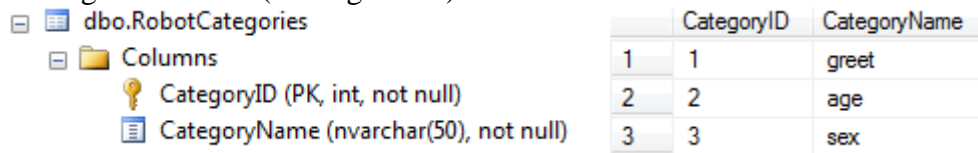
The screenshot shows the 'dbo.Map' table structure in SQL Server Enterprise Manager. The table has two columns: 'MapID' (int, not null) and 'Info' (nvarchar(50), not null). To the right, a data table shows the following rows:

	MapID	Info
1	1	i
2	2	to

Figure-13

The “Map” table is used to discern the user’s navigation proposes. The “MapID” is used to specify the different info in its unique ID and the “Info” table is used to specify the content.

“RobotCategories” table: (see Figure-14)



The screenshot shows the 'dbo.RobotCategories' table structure in SQL Server Enterprise Manager. The table has two columns: 'CategoryID' (Primary Key, int, not null) and 'CategoryName' (nvarchar(50), not null). To the right, a data table shows the following rows:

	CategoryID	CategoryName
1	1	greet
2	2	age
3	3	sex

Figure-14

The “RobotCategories” table is used to discern the robot response category. The “CategoryID” is used to specify the different category name in its unique ID and the “CategoryName” column is used to specify the content in different case.

“RobotKeywords” table: (see Figure-15)

	KeywordID	KeywordContent	CategoryID
1	1	hello	1
2	2	greeting	1
3	3	meet you	1
4	4	nice see you	1

Figure-15

The “RobotKeywords” table is used to discern the robot response category. The “KeywordID” is used to specify the different keyword in its unique ID, the “KeywordContent” column is used to give the response to the user in different case and the “CategoryID” column is used to specify the content in different case based on the “RobotCategories” table.

“RobotResponse” table: (see Figure-16)

	ResponseID	CategoryID	ResponseContent
1	1	1	Hello, my friends!
2	2	1	Hello!
3	3	1	Hi.
4	4	1	Greetings.

Figure-16

The “RobotResponse” table is used to given the robot response depending on the request category. The “ResponseID” is used to specify the different response in its unique ID, the “CategoryID” column is used to give the response to the user into different case and the “Response content” column is used to give the response content.

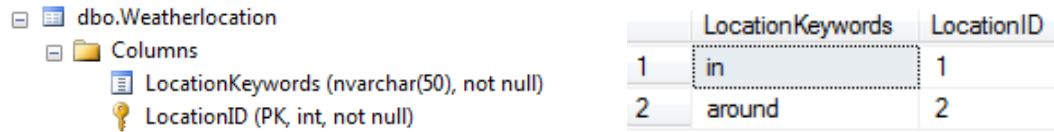
“Weather” table: (see Figure-17)

	KeywordCategory	TimeID	Time
1	1	1	today
2	1	2	tomorrow
3	1	3	yesterday
4	1	4	day after tomorrow

Figure-17

The “Weather” table is used to discern the robot response category. The “KeywordCategory” is used to define the category of this content, the “TimeID” column is used to give the unique number to each “Time” content and the “Time” column is used to specify the time content.

“Weatherlocation” table: (see Figure-18)



	LocationKeywords	LocationID
1	in	1
2	around	2

Figure-18

- Detailed solutions and implementation for each function depend on the request categories.
 0. Chat Mode: The program will get the captured text and send it to the cloud web service, the cloud will loop over the robot chat keywords and identify the keyword category; the response will be randomly accessed through the response pool according to the keyword category, finally the program init the [TextToSpeech](#) engine from the Android system and generate the audio output with the response. [Code-0-1]
 1. Chat Mode Switcher: the program will have a Boolean variable initiated to false. If the chat mode is enabled, the variable will be assigned as true and anything captured will be in the chat mode until the chat mode is finished. While the chat mode is exited, it gets back to the normal mode and analyzes the requested commands. [Code-1-1]
 2. Location Service: The program will firstly distinguish the command in two different ways; one is to find the current location, another one is find the routes between the current location and the destination location. To find the current location, the program will check the location information from the device GPS Module and get the current Longitude and Latitude values, then start the MapActivity by assign the pair values and the mode “current”, present the maps for the user. To find the route to a specific destination, the program will also check the current location and get the [GEO](#) values, generate the target location name to an URL, read the [GEO](#) Information from the link [Code-2-1], with the [GEO](#) info for both the origination and destination, the program will start the MapActivity by assign the current location geo value and the remote location geo value with mode “Remote”. The map activity will generate that information to an URL and send to the Google map server, then get the route XML. And draw the route on the map.
 3. Weather: the program will firstly check the command whether it has the specific city name, if the city name is obtained in the command, the program will send the city name to the [Google map server](#) and get the corresponding geo information with the longitude and latitude and set as a location to get the weather condition;

INTELLIGENT VOICE ASSISTANT

otherwise the location will be the current location information from the mobile `GPS Module`, if no city name is given, the program will generate an URL by the location's geo info, and get the corresponding weather condition XML from the Google weather Server. The program will also check the data info from the cloud response, if the user requires the weather for today, the program will present the first weather condition from the XML, otherwise, it will get the next four days conditions.[Code-3-1]

4. Wikipedia search: the program will replace the space in the search content to “+” and formalize the searching URL, and then switch to the search activity by calling `ACTION_VIEW` and give back the result as navigate to the previous obtained URL. [Code-4-1]
5. Calling service: the program will extract the name section from the response accessed from the cloud web service, then check through the contact list and get all the stored contacts [Code-5-1], further fetch all the details of the person with name, email, phone number [Code-5-2]. Identifying the person and get the first phone number, and the system will make the phone call by calling the system `ACTION_CALL` intent and start the calling activity. [Code-5-3]
6. SMS: the program will extract the name section and the message content from the response accessed from the cloud web service, then check through the contact list and get all the stored contacts [Code-5-1], further fetch all the details of the person with name, email, phone number [Code-5-2]. Identifying the person and get the first phone number, and the system will send the message by calling the system `ACTION_SENDTO` intent and start the sending message activity. [Code-6-1]
7. Email: the program will extract the name section and the email content from the response obtained from the cloud web service, then check through the contact list and get all the stored contacts [Code-51], further fetch all the details of the person with name, email, phone number [Code-52]. Identifying the person and get the first email address, and the system will send the email by calling the system `ACTION_SEND` intent and start the sending email activity. [Code-7-1]
8. Google Search: the program will replace the space in the search content to “+” and formalize the searching URL, and then switch to the search activity by calling `ACTION_VIEW` and give back the result as navigate to the previously obtained URL. [Code-8-1]
9. Alarm: the program will extract the Hour and Minute parts from the response obtained from the cloud web service, set a calendar with the requested time of hour, minute and second. Then start the Alarm manager by calling the system

INTELLIGENT VOICE ASSISTANT

`ALARM_SERVICE` with the settled calendar and broadcast. In addition, the broadcast is a trigger to activated an alert and the alarm music will by played when the alarm is activated by system action `RTC_WAKEUP`. [Code-9-1]

10. Music Player: When the program is loaded and initialized, it will call the system `ACTION_MEDIA_SCANNER_FINISHED` to scan all the media files on the SD card memory and save the file's path, id, title, and put all these attributes into a list[Code-10-1], the program will first extract the action command from the response obtained from the cloud web service, if the command requires to playing music, it will further check whether the response contain with the song's name or not, if the request does not have a specified name of the song, the program will randomly pick a song from the list and start the music play service by given the path of the requested song, otherwise, the song's path will be obtained from the list by the song's name and start the music with start command[Code-10-2]. If the response contains the pause command, the program will set the music service at a pause state. As it is the same with pause, the stop command also will be sent in this way and the music player will stop playing the music. [Code-10-3].
11. Event handler, the program will firstly extract the command part to decide if the user wants to add or view or delete events. The event program will navigate to the event activity with the requested command. The layout of the event activity is designed through the XML file and different operations "Add/View/Delete" are set on the interface. By extending `SQLiteOpenHelper` and `SQLiteDatabase`, the events can be stored, and updated or deleted.
12. Camera: when the program receives the start camera command, it will start the Camera activity, then init the Speech Recognizer on that activity. After the user take photo by recognize the "Cheese" command and save the image into the SD card memory, a broadcast will be triggered to notify the system's gallery to refresh the photos. After the photo has been taken and stored, the camera activity is finished and give the image path back to main activity, and the main activity will present the image to the user based on the image path from the given path, the user also can touch on the preview image to view the image detail by start the `ImageViewActivity`.
13. Help: the program will navigate from the current activity to the help activity while the help menu is activated from the main option menu or by the detected command. The help activity contains a list of items correspond to each different function; they share the same outline with an icon, text explanation [Figure 13]. If any image button is clicked, it will switch to the help content activity with the corresponding name of the function. By getting the name of the function, the content activity will fill its content with the icon, title, and the examples to tell

INTELLIGENT VOICE ASSISTANT

how to work with the function. The layout of the activity mainly been constructed with the TextView, ImageView, and ListView. [Code-13]

14. Translate: the program will get the target language code and the content text, then generate the original language code, target language code and the content text to a URL; start the URL and get the translate result from Bing, finally present the result with the original text and the translated text for user.[Code-14]
15. Bluetooth headset support: when the user plug-in the Bluetooth headset the system will send a broadcast to the program, the program will use a Bluetooth receiver to receive this broadcast then enable the button for user to select if use the Bluetooth or not.

3.2 Equipment/ Choice of Materials

This chapter indicates all the equipments of the hardware, software and developing platforms. Apart from the equipments, the materials that used in developing the program are also showed in API and reference.

Develop tools and environment: Java TM, JDK, Eclipse IDE, Android SDK, ADT Plugin, ADV, and Plug-in for Eclipse, MySQL query browser, DB-Designer, Microsoft Visual Web Developer 2010 Express and Windows Azure Cloud Platform.

API and reference: Java API, Android API, Google API (Google Map, Google Weather), Wikipedia API, SQL tutorial, UML reference, JSON, XML, WSDL, Cloud computing, multi-threading techniques.

Software application on Android phone: Android Internet explorer, Google voice recognize, TTS Service Extended, Alarm, Mobile phone calling services, text message services.

Support application: Adobe Photoshop CS5, StarUML, Meitu (Chinese).

Hardware support: Android phone [HTC/Samsung], PC, Bluetooth Headset.

Developing model: XP (Extreme programming) (specify model)

- Requirement Card (Requirement analysis and identification)
- Design Card (Implementation and construction of modules)
- Test Card (Black & White Box test on modules)
- Pair-programming (Code modification, optimization and Communication)
- Integration and Simplicity (Integrated modules)
- High-Level Test (Black & White Box test on system)
- System debug (Potential errors and possible bugs)
- Build Product & Revision control (Evaluation and developing history)
- Calculate velocity and efficiency

3.3 Problems and Solutions

During development (see Figure-6) we have encountered many problems while implementing those functions. Selected core problems with their solutions are listed by the following section:

- **Chat Mode VS Command Mode:** When the user wants to chat with the robot, the program will not distinguish the keyword in the statements, because the chat is random and every sentence have higher possibility to contain a keyword that Mapping to a command, that will cause the program confuse about the words and may give a false response.

Solution: The program has been designed in two modes: Chat Mode and Command Mode. Both modes has different databases(explain), if the user want to chat with the robot, he or she can say "Chat mode enable" or "Let us chat", that will lead the program enter the chat mode. After entering the chat mode, every statement will be a chat request and a response will be given until the user says "finish chat" or "end chat". During the chat mode, the program will give chat response for the command statements like "weather today" or "where am I" instead of giving response to the weather/location functions, that will be much easier for the program to distinguish the keywords.

- **Location:** there have been problems in getting the GEO info according to the given city name when implementing the location service. Except getting the current location where the user is, there should also be allowed to get the location by a city name. The direction must be precisely given from the current location to the destination according to the given name.

Google Map Service is the solution to get the GEO info based on the city name. By implementing the Google Map Service which is a free API, the GEO info and the route trace from the current location to the destination can be accessed and clearly presented on the map.

- **Weather data retrieving:** When trying to design the keyword functions about the weather data part, it was discovered that in the sentences "tomorrow" and "the day after tomorrow" it was hard for the program to distinguish the actual data info in the statement. Since the statement "the day after tomorrow" also contains the word "tomorrow", the program may only capture the word "tomorrow" and skip "the day after". To solve this problem, the weather condition will display the next 4 days' weather in to an entity. When the program captures the word "today", it will only show the current weather condition, otherwise, the program will show the forecast for next 4 days in an entity for all the other cases.
- **Calling service:** There has been a very fundamental problem when implementing the calling service. The program cannot run properly with the expected function after finishing the implementation of the coding. And it was always the same runtime problem when it was tested and it was modified lots of times without any solutions.

INTELLIGENT VOICE ASSISTANT

The solution is found after the CONCAT explorer is opened, and developers can access each entity of the running message and identify the problem. There has been found no calling permission is allowed in this program and that is reason why the program gets crash while trying to revoke the calling service. Access the manifest.xml file of the program and add calling permission, then the program works as expected and calling service can be successfully made.

- Alarm: The alarm was firstly implemented with a broadcast which will be triggered when the time comes up, but after a carefully concern on the user-friendly design, the broadcast should also have a alert as well to stop the alarm music which is implemented on other class [main activity – since the system music player must be implemented in the activity which the broadcast is not an activity] rather than the broadcast. Therefore, the problem was how to trigger an event in another class.

Different solution has been tried as define the music player a static/ final static object which can be directly fetched from the other classes, define methods as to get and set the different variables between classes, and etc; all those solutions failed because the mismatch between two classes, hence, the object might be null while they were sent to another class which generated NULLPOINTEXCEPTION. The final solution which solved this problem was using message handler. Send the message handler info to stop the alarm while the time is up and the message handler will trigger the alert and actually stop the music.

- Music player: there have been problems of how to get the get, load and update the list of music when doing the music player. Since the user might update the list of the music any time as he/ she wants, the program should load the song list with updated info.

The solution is to implement a broadcast, while loading the list of the songs every time the program is started, the broadcast will inform the broadcast receiver to scan and filter the mobile phone SD card, then access all the music available and store with the info of each song into a list for later use.

- Camera: The Android mobile may have the two cameras: front camera and the back camera. The front camera is always used for self-shooting or video chat, this camera does not have the autofocus function and that it is a low-definition device, the back camera always use for shooting the landscapes, or Portraits. The program needs to be designed to have a function for the use choose to switch the cameras. For this function required, the program has to use the API in Android library but the front camera method only implement since API 10 (Android 2.3), and the program has build on API 8 (Android 2.2), in that API, the switch camera method cannot be implement since we use API 8, so we decide to update the whole program on API 10 to solve this problem.

INTELLIGENT VOICE ASSISTANT

Another part of the camera is about the voice record. At the beginning, this function was designed to take a long range self-shooting, during the camera listening mode, it can automatically record the speech about every 3-5 seconds and then distinguish the statement whether contain the keywords, after the word has been captured, the camera will automatically capture the photo or do another listening to the user. But during the testing, the voice recognizer was not be able to enable the microphone to start a new listening after the first recognition, so a button has been put on the screen to let the user to start a new listening by pushing the button on the screen manually instead of automatically start a new listening after each time.

After the photo has been taken, the photo will be saved into the SD-card memory, but the Android's system will not automatically update the photos into the gallery. The system's gallery only refreshes its source when the system starts. So there have to design a method to broadcast a message to notify the system gallery to refresh its library on the SD-card when the photo has been captured.

- Owing no equipment: it has been a long time problem for the development since having no Android phone. Even the program can be write in Eclipse and test with the emulator, the physical phone is needed for real-time test on the real phone; having no mobile phone, the voice recognition cannot be test and there can only be text input manually if the program need to be test. In addition, the school provided a Sony Ericsson phone with Android operating system, but that phone was too old with a 2.0 Version which cannot implement this program.

Thanks to WANG LINLIN who lends her HTC to the developers and the program was well finished and test on the real phone. The mobile phone will be available to use until the program is fully finished.

4 Results

4.1 Design

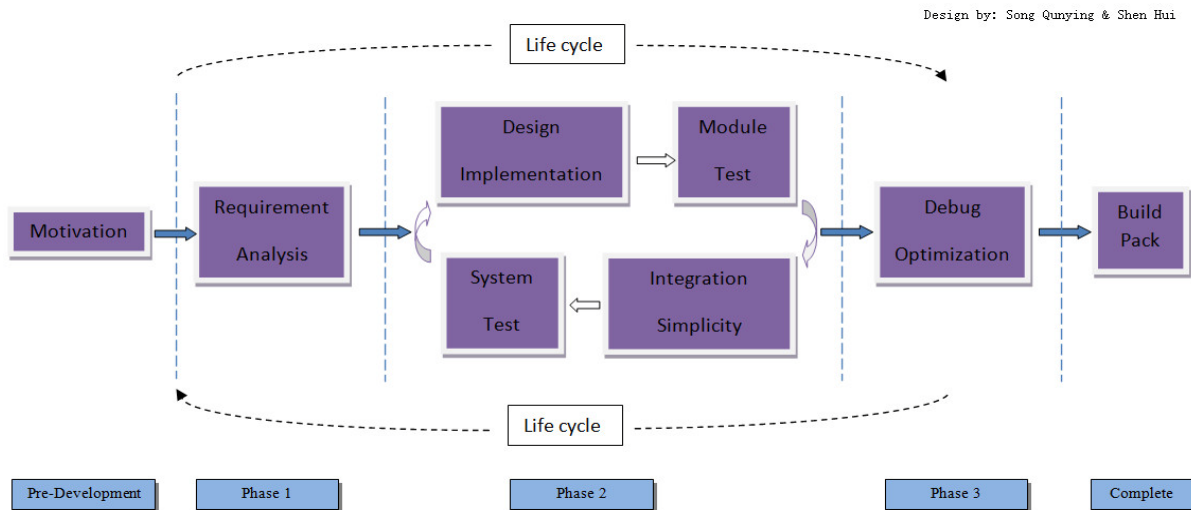


Figure-19

The Model and Flow Chart (see Figure-19) describes the develop process that include all the phases in the software development life cycle. This chart is well illustrating how the project is carried out and how the development was managed. The project started with the motivation and brain storm, repeatedly implement in the developing life cycle until the system has been fully constructed.

- Brain storm, the project start with the ideas from the brain storm. Here the basic ideas and design the primary concepts, prototype of the program have been obtained.
- While the ideas has been obtained, it has been analyzed which of them can be accomplished and make sure the structure of the project.
- According to the requirements that had been identified, collected all the resources and useful references from any channel, together with the programming skills and experiences, the design items were pointed out.
- Implement each individual design item based on the planning, structure and references.
- Test each single module that has been implemented and fix the possible bugs appear in the code implementation and make sure the functions are well constructed.
- Integrate all the individual sections to contribute to a complete system.
- Try the black and white box testing strategies to test the system, both the functional and non-functional logic and implementation should be verified.

INTELLIGENT VOICE ASSISTANT

- Debug the system and optimize the project from the possible aspects.
- Build the product and pack all the stuffs as a whole.

4.2 Functioning

The program should firstly be started on the Android phone; the initial mode of the program is Voice mode since this program aims at making a voice assistant program. However, if there are users who prefer to operate in text mode by inputting the text manually, the text mode is also available.

After the program has been started, the user should have correct voice input “command/request” to make those functions work properly. And this program includes the functions and services of: calling services, text message transformation, mail exchange, alarm, event handler, location services, music player service, checking weather, Google searching engine, Wikipedia searching engine, robot chat, camera, Bing translator, Bluetooth headset support, help menu. The details below explain how those functions work and different possibilities while facing different commands.

- Calling service, the calling function allows the users to give a call to the person in the contacts. By giving a correct command with the calling request to a stored person, the Android phone will check the contact list and get the phone number of the person, then successfully direct to the phone number found in the contacts.
- Text Message transformation, the text message transformation enable customers able to send the SMS to the person in the contacts. By giving a correct command contains the request keyword to send SMS together with the destination person; the program will navigate to the sending message function on the mobile phone with the phone number, message content. The message will be sent to the destination immediately if the user selects to send it with the correct content.
- Mail exchange, customers are able to send the mail to the person with mail address in the contacts. By giving a correct command contains the mail request keyword together with the destination person; the program will switch to the sending mail function on the mobile phone with the mail address and mail content. If the content is correctly detected, the mail will be received by the recipient after the user selects to send the mail, otherwise the user can modify the mail content if the voice recognition is not well detect the mail content.
- Alarm, as a basic function on the mobile phone, the user could simply set the alarm through the command with the setting alarm keyword and a specific valid time. When the alarm request and time are detected, the program will set the alarm to the given time with dedicated hour, minute and second; when the time comes up, the alarm will be trigged with a alarm bell and an alert notification which the user can choose to stop the alarm, otherwise the alarm will keep working and the song will always be playing.
- Event handler, the application allows the user to set as many events as they want. Customers set the events with the content and title, the program switch to the event handler interface with the content and the title, and the event will be stored immediately if the user ensure the event. With

INTELLIGENT VOICE ASSISTANT

the stored events, the event handler makes the events available for the user to check all events, check one event, modify the selected event and delete all events.

- Location services, location services works in two categories depending on the request.

If it has been required to present the current location of the user, the location services check the GEO info by using the Google Map Service and give back the result as a map with the current location.

If it has been required to provide the route trace from the current position to a specific city, the location service check the GEO info of both the origination and the destination, and provides the direction on the map with a route indicating how to get to the destination from the origination.

- Music player service, the music player offers the services to the user to play a named or random song in the pre-stored song list depending on the request.

The music player service will play the specific song according to the name given by the user, the music player check the music list and identify the song, then presenting to the user.

The music player service will play a randomly picked song through the list if the user doesn't provide the song that he or she wants. The music player traces through the music list and get one from it for playing to the user.

The music player could be also be stopped or paused while it is playing a song. By giving the correct commands, the working music player will be paused or stop playing.

- Checking weather, weather service provides the user the weather condition in different city on different dates. This service works in the same logic and gives back different result depending on the requested date and city.

The weather service return the current date weather condition of the current location with the humidity, wind speed, temperature scope and display in a formalized entity which can be easily read by the user if the local weather for current date weather is required.

The weather service return the next four days' weather condition of the current location with the date, wind speed, temperature scope and display in a formalized entity which can be easily read by the user if local weather for other dates except today's weather is required.

The weather service return the current date weather condition of the given city with the humidity, wind speed, temperature scope and display in a formalized entity which can be easily read by the user if weather for current date weather for the given city is required.

The weather service return the next four days' weather condition of the given city with the date, wind speed, temperature scope and display in a formalized entity which can be easily read by the user if weather for next for days of the given city is required.

INTELLIGENT VOICE ASSISTANT

- Google searching engine, the search engine enable the use to search anything on Google. By detecting the search keyword and search request, the Google search engine will returns the result list displayed on the browser on the mobile phone.
- Wikipedia searching engine, the search engine enable the use to search anything on Wikipedia. By detecting the search keyword and search request, the Wikipedia search engine will returns the Wikipedia result displayed on the browser on the mobile phone.
- Robot chat, the robot chat enables the user to chat with the Android phone to have fun. The chat mode is initially closed and will be required to activate it with the corresponding command. After entering the chat mode, a text response will given by the mobile phone whenever the user speaks to it; the response, however, were predefined and stored in the database. For each request, the program will define the request category and randomly pick a response from the response pool depending on the request category.
- Camera, the camera function enables the user to capture the current view with the camera on the mobile phone. When the camera is activated by the user, the user can selects to use the front or back camera on the mobile phone manually, and the picture will be taken by the camera if the user selects to photograph the current view, an instant picture for previously taken will be displayed in the program for viewing as a entity, and the picture will be stored in the Gallery for later checking.
- Bing translator, the translator will provide the user both the original text and the translated text depending on the objective language the user given. The user gives the original text and the object language the he wants; then the translator will give the result back of a translated text based on the original text and required language. Meanwhile, there have been 25 object languages stored in the database which the user can enjoy and the original text should be in English to use the translate function.
- Bluetooth headset support, the Bluetooth headset support makes the program well work especially the phone is playing music or the surrounding is noise which affect the voice recognition. Since it is not possible to do the voice recognition while the music player is playing, the Bluetooth will be loaded and available to the user, the user can select to turn on or turn off the Bluetooth function, and the Bluetooth headset support makes it possible to speak to the headset rather than the mobile phone if the Bluetooth is enabled.
- Help menu, the help menu provides the user a help list to each function in this program. The user can choose the help menu manually or over the voice if the user doesn't know how to work with the functions. While the help menu is opened, the help menu gives the examples and explanation of how to work with different functions, the examples clearly show how to work with the function and the user can simply imitate the example to work with different functions.

4.3 Operation and Maintenance

Operation

- **Calling Service:** If the user wants to consume the calling service, he or she must have a command contains a valid name the calling keyword like “call”, ”make a phone”, then the call will be made if the person is found in the contacts. There are different ways to make a phone call, the list below shows the correct command to use the calling service.

“**Call Tom**”, make a phone call to tom. The program will first capture the key words “call”, and then the program will continue to capture the person’s name “Tom” after the word “call”, then get all the contacts on the mobile and compare them one by one, if “Tom” is equal to the name that the user is give in the command, the phone call will be made to “Tom”.

“**I want to give a call to Lucy**”, make a phone call to Lucy. The program will capture the command keyword “call” and the name “Lucy” and make a phone call to Lucy.

- **Text Message Transfer:** If the user wants to use the application to send the text message, he or she must have a command with the SMS message keyword and a valid name, then the message will be send if the person is found in the contacts. They are different forms to send the message; the list below shows the correct command can do the message sending.

“**Send a message to LiLei Let's dinner together**”, send a message to LiLei with the content “Let’s dinner together”, the program will capture the keyword “message” and the content “let’s dinner together”, then the program will check the mobile contacts and get the first phone number corresponding to “LiLei” and send the message to LiLei.

“**SMS Hui Nihao**”, send a message to Hui with the content “Nihao”.

- **Mail exchange:** The user can send an email to the person in his contacts and with person’s email address. He or she must have a command with the email keyword like “Mail”, “Post” and a valid name; the email will be send if the person is found in the contacts. They are different forms to send the message; the list below shows the correct command can do the email sending.

“**Mail Bellis it will rain today**”, send an email to Bellis the content “it will rain today”, the program will capture the keyword “Mail” and the content “it will rain today”, and then the program will check the mobile contacts and get the email address corresponding to “Bellis” and send the message to “Bellis”.

“**Post Mimy a boy is waiting for you**” send an email to “Mimy” with the content “a boy is waiting for you”

INTELLIGENT VOICE ASSISTANT

- Alarm: The user can use the set alarm command to set an alarm at the corresponding time. When the time is up, the alarm will be activated and play the sound; meanwhile, an alert will be presented for the user to stop the alarm.

“**Set alarm to 10**” the alarm will be set at 10 o'clock. The program will capture the setting command “Set alarm” and get the time command “10” and then the alarm will be active at 10AM.

“**Make time to 11:50**” the alarm will be set at 11:50. The program will capture the command “Make time” and the time “11:50”, the alarm will be wake up at 11:50.

- Event handler: The application can allow the user set many events. He or she can set many events and be saved into the application's database by using the add event command, also he or she can view the event or delete the event by corresponding keywords like “Set up”, “make up”, “View one/all event(s)”, “Delete”. The list below shows the correct command can do the operation.

Add Event:

“**Set up a meeting at 10**”, the Program will first capture the keyword “set up” then the title of the event “a meeting” and the content “a meeting at 10”, then the event activity will be start with the add event dialog, automatically fill with the title “a meeting” and content “a meeting at 10”. Then the user should to choose the date time and add the event.

View Event(s)

“**View all/one event(s)**” / “**Find event**”, the Program will start the event activity and present the event(s) based on the user choose to show the all events or one event. If the user chooses to show one event, the data picker will be shows up and the user can choose the event that he or she wants to present based on the date, otherwise, the application will show up all the events if the user wants to check all.

Delete Events

“**Delete all events**”, the Program will delete all the events that in the application's database.

- Location services: The user can use this service to locate the user's position or get the routes to the destination by giving the city name. There are different ways to locate the position or navigate to a specific city. The use must use the keywords “where” and “I” or “my location” to let the application to know he or she wants to locate the current position. And the keywords “go to” and the name of the place to get the route to the destination.

Locate position

“Where am I” / “Show my current location”, the program will present the current location of the user on the map.

Navigation

“How can I go to Lund” / “Navigation to Lund”, the program will present the routes to “Lund” on the map with the highlighted route from the current location to Lund.

- Music player service, the user can use this application to play songs, his or her command must contain keyword “play”. If the user wants to play the specific song, he or she should also say the name of the song after “play”, and the song should be exist in the SD-card memory. Or if the user wants to play a random song, he or she just needs to say “a song” instead of the song’s name. During the playing, the user can pause or stop the song by giving the command “pause” or “stop”.

Play

“Play Canon”, the program will play the song “Canon”

“Play a song for me”, the program will randomly pick a song from the library and play it.

Pause

“Pause playing music”, the song will be paused immediately.

Stop

“Stop music player”, the song will be stopped immediately.

- Checking weather: the user can use the application to check the weather for recent days in local place or specific location. He or she should say the keyword “weather”, then the user should notify the date that should be presented as “today/tomorrow/the day after tomorrow” if he or she wants to get the information about the other days otherwise the application will default set the date as today, and the user can also can choose to tell about the place name “in Malmo”, the application will check the weather belong to that place, otherwise the place will be set as locally.

Weather check today:

“What's the weather for today”, the current weather condition for local place will be show.

“What's the weather in Malmo”, the current weather condition for Malmo will be show.

Weather check other days:

“What's the weather next few days”, the forecast in next 4 days will be show.

INTELLIGENT VOICE ASSISTANT

“[What's the weather next few days in Malmo](#)”, the forecast for Malmo in next 4 days will be show.

- Google searching engine, the Google search engine is activated by the user commands which contain ‘Google’ or ‘Search’. By detecting the search keyword and search request, the Google search engine will returns the search result displayed on the browser on the mobile phone.

“[Google China](#)”, the keyword ‘Google’ is detected and the result will be presented on the web browser by searching ‘China’ on Google.

“[Try to Google Java API](#)”, the user can have the keyword Google in the middle of a request and the result of searching ‘Java API’ on Google will be displayed on the web browser.

“[Search for apple](#)”, the user can also use the keyword ‘search’ to do the Google search, this command will have the result of searching ‘apple’ on Google.

- Wikipedia searching engine, whenever the user wants to search any content in Wikipedia, it is possible to do in this program by having a command contain the keyword ‘define’. If ‘define’ is detected by the program, the program will automatically give the result by search the content after ‘define’ in Wikipedia.

“[Define Android](#)”, the keyword ‘define’ is detected, and the program will return the result by searching ‘Android’ on Wikipedia.

“[Define true love](#)”, the keyword ‘define’ is detected, and the program will return the result by search the content after ‘define’, which is ‘true love’ on Wikipedia.

- Robot chat, the robot chat will work only after the chat mode is enabled which can be done with a command that contains keyword ‘chat’. After the chat mode is enabled, a response will be given every time when the user gives a request. The chat can be finished by the user commands contain the keywords of ‘finish/ disable/ end/ complete chat’.

“[Enable chat](#)”, the keyword ‘chat’ will be detected and the chat mode will be enabled. Now the user can enjoy the chat by inputting any text he /she wants.

“[Let's chat](#)”, the keyword ‘chat’ will be detected and the chat mode will be enabled. Now the user can enjoy the chat by inputting any text he /she wants.

“[Finish chat](#)”, the keyword ‘finish chat’ is detected and the chat mode will be disabled. When the user exits the chat mode, the program gets back the normal mode to receive and analyze the commands, and give correct response.

INTELLIGENT VOICE ASSISTANT

- Camera, the camera is started while the keyword 'camera' is detected. Therefore, the user who wants to operate with the camera will have to give a command with camera inside. After camera is started by the correct command, the camera itself will guide the user how to take photograph.
“[Open the camera](#)”, as the keyword 'camera' is detected, the camera is started. And the user can work with the camera by clicking the different selection on the mobile phone.
“[Start the camera](#)”, as the keyword 'camera' is detected, the camera is started. And the user can work with the camera by clicking the different selection on the mobile phone.
“[I want to use the camera](#)”, as the keyword 'camera' is detected, the camera is started. And the user can work with the camera by clicking the different selection on the mobile phone.
- Bing translator, the user should have the keyword 'translate' / 'how to say' as the keywords to define this is a translate request, and 'in' as keyword to indicate the objective language. As the user have the command contains these keywords, the translator will return the result with the text in the objective language.
“[Translate I love you in Chinese](#)”, as 'translate' and 'in' are detected by the program, the program will call the translator with 'I love you' as the original text and Chinese as the objective language, the result will be the Chinese words of 'I love you'.
“[How to say hello in Swedish](#)”, as 'how to say' and 'in' are detected by the program, the program will activate the translator with 'hello' as the original text and Swedish as the objective language, the result will be the Swedish text of 'hello'.
- Bluetooth headset support, the Bluetooth headset support will be enabled when the program is loaded. The user should firstly turn on the Bluetooth in the setting of the mobile phone, and the Bluetooth icon will be valid in the program after executing the program. The user will be required to plug in the Bluetooth headset and turn on /off it manually by clicking on the Bluetooth icon.
- Help menu, the help menu can be activated by manually select on the option menu or through the command. The commands should have 'help' as the keyword contained, then the help menu will be activated and the help menu provides the list of all functions with their explanation and examples to use it.
“[I want to check the help menu](#)”, if the users have the keyword 'help' contained in the command, it will be detected as a keyword and the help menu will be returned with a list of the functions, the functions are presented in two pages and user can scroll the pages by slipping the touch pad of the mobile phone; by selecting on each of the functions, the user can enjoy the details of the explanation and the examples of each function.

Maintenance

After the program is completed, the program still needs long term maintenance to make it available and stable to execute. The program will be test after a certain period of time and debug each of the function and possible bugs, whenever a potential bug is detected; the program needs to be refined to a better design. Meanwhile, there will update and add more data to the database to increase the database capacity. Depending on the new keywords, responses, relevant data found that could be applied in this program; the database will always be improved and can handle more and more cases.

5 Conclusions

- Project development and implementation

As it has been previously stated, the program is mainly concerned with the techniques of Android development, Java programming, Database management, Cloud computing, different APIs for Google products, Bing translate and etc. The program is developed by two developers and follows the extreme programming model. During the eight weeks development, the developers did the same cycle in each phase of analyze requirements, construct design, implement the solutions in pair programming mode and test the result. The development is carried out as its primary planning which guides the work process of how to work with the program, how much time should each of the developers spend in every week, the resources needed for developing and how to handle the problems while it came up. The project was efficiently completed under the development model and the resources we found in early time were really useful when implementing the program.

- Project usage & prospect, potential

The project is very useful and owns a large potential use in different industries. Although the program primarily concerns more about how to do the personal assistant on Android phone using the voice, the concept of voice recognition can be applied in different industries as in many situations it will be more convenient, save a lot of time and helpful especially for those who have difficulty in working with manual operations. Thus, the concept is only for programming the Android application.

For the program itself, it is a collection of 15 functions that are frequently used on a mobile phone. The user can enjoy different services within this platform. Therefore, it is easy to use with simple operation compared with the traditional working strategies which the user should well know how to work with the mobile phone.

In addition, the program which works using the voice is helpful for those who prefer voice operation and those who have difficulty /disability with the manual operations. The primary objective of the program is to provide services using the voice, and it enables more people who can enjoy this program.

The prospect of the program can be more applications or products developed using the voice control, and it could in some sense change the working forms that is totally different from the traditional form. As people can easily operate and have a lot of fun from it, it owns an enlightened prospect as SIRI succeed in attracting people in the market.

INTELLIGENT VOICE ASSISTANT

INTELLIGENT VOICE ASSISTANT

- Project experience & teamwork

Apart from the program, we as the developers have improved a lot from the degree project. It is quite different from what we previously experienced in the working model, volume of tasks, and the problems we have encountered. In conclusion, we have been improved a lot from the project development, and gained development experience as well as programming skills; the most important is work as a team for a long term, challenge development.

6 Recommendations for Further Work

6.1 Design Improvements

No program has a perfect design without any flaws; it is the same here in this program. Even though the program is completed with all the primary functions implemented and work properly, there are still many things that can be done with this program. As the future improvement, the potential work that can be implemented ranging from adding more functions to offering the user a more comprehensive, convenient program, refining the logic to make the program more humanized and easy to use, increase the database capacity and add more possible keywords, responses and data in this program, interface optimization and etc.

6.2 Additional Functions

Add more functions: although there have been 15 normal functions that are used really often with the mobile phone, there can be more functions which simplify our daily life and make it convenient to use. Functions as playing movies, checking stocks, exchange rate, downloading and uploading, installing APPs and etc, these can be the potential functions that make the program more comprehensive and people can enjoy more services in this program.

6.3 Database Capacity

Add database capacity and more humanized logical design; the program has a predefined logic to make it work with the corresponding commands. Thus, the user need to follow the structure of the commands, contain the dedicated keywords and well formalize the commands to work with each of the functions. In other words, the program is limited by the database capacity and no solution will be found if the user gives commands that are not readable by the program. Even if two commands have the same meaning and should get exactly same result set, the result might be that of one is working and the other one fails. Hence, the program is to some extent limited by the vocabulary and can be further optimized.

6.4 Humanized Voice Recognition

The more humanized the program is, more easier the user can use it. People should accept that even if developers constantly try to add more predefined commands, more responses to it, analyze and respond to the command more intelligently, the program will never be completely comprehensive and contain all the possible circumstances that the users meets. Nevertheless, the program will certainly be improved and be more user-friendly if there can be more readable commands, more humanized structure and more intelligent response.

6.5 Improved Interface

Interface optimization, the interface can be further improved to make it nice to the users. Currently the interface design meets the basic requirement to present everything for this program, and the users are able to interact with the program through this interface, but the interface can always be optimized and more suitable constructed.

7 References

7.1 List of References

- [1] [http://en.wikipedia.org/wiki/Siri_\(software\)](http://en.wikipedia.org/wiki/Siri_(software))
- [2] <http://en.wikipedia.org/wiki/Smartphone>
- [3] <http://yudian.voicecloud.cn/>
- [4] http://en.wikipedia.org/wiki/Extreme_programming
- [5] http://en.wikipedia.org/wiki/Cloud_computing
- [6] http://en.wikipedia.org/wiki/Extreme_programming
- [7] http://en.wikipedia.org/wiki/Java_programming
- [8] <http://docs.oracle.com/javase/6/docs/api/>
- [9] <http://developer.Android.com/index.html>
- [10] <http://developer.Android.com/reference/packages.html>
- [11] <http://developer.Android.com/guide/index.html>
- [12] <http://www.microsoft.com/sqlserver/en/us/product-info/overview-capabilities.aspx>
- [13] <http://www.mysql.com/why-mysql/>
- [14] <http://www.windowsazure.com/en-us/home/features/sql-azure/>
- [15] <https://www.windowsazure.com/en-us/develop/net/fundamentals/intro-to-windows-azure/#cloud>
- [16] http://en.wikipedia.org/wiki/Web_Services_Description_Language

8 Appendix A Figure

Request	Request category	Response	Response code
Chat	0	Chat	“0 1 Content”
		Disable	“0 0”
	1	Enable	“1 1”
		Disable	“1 0”
Location Service	2	Location	“2 1”
		Direction	“2 2 Destination City Name”
Weather	3	Local & today	“3 1 0”
		Local & other day	“3 2 0”
		Remote & today	“3 1 1 City Name”
		Remote & other day	“3 2 1 City Name”
Wikipedia	4	Definition	“4 Content”
Calling Service	5	Make phone call	“5 Receiver’s name”
SMS	6	Send message	“6 Receiver’s name Content”
Email	7	Send email	“7 Receiver’s name Content”
Google	8	Search engine	“8 Content”
Alarm	9	Set alarm	“9 Hour Minute”
Music player	10	Start & Random Song	“10 1 000”
		Start & Given Song	“10 1 Song name”
		Pause	“10 2”
		Stop	“10 3”

INTELLIGENT VOICE ASSISTANT

Event	11	Add an event	"11 1 title Content"
		View one event	"11 2"
		View all events	"11 3"
		Delete all events	"11 4"
Camera	12	Start camera	"12 1"
Help	13	Help Menu	"13 1"
Translate	14	Translate content	"14 target language code Content"

Table-2

9 Appendix B Code

Code-0-1

```

    case 0:

        int state = Integer.parseInt(getSecondInfo(response));
        switch (state) {
        case 1:
            chat = true;
            persendInfo("he", getThirdInfo(response));
            break;
        case 0:
            chat = false;
            persendInfo("he", DefaultResponse.getChatFinishSpeak());
            break;
        }
        break;

```

Code-1-1

```

    case 1:// robot
        switch (Integer.parseInt(getSecondInfo(response))) {
        case 1:
            chat = true;
            persendInfo("he", DefaultResponse.getChatSuccessSpeak());
            break;
        case 0:
            chat = false;
            persendInfo("he", DefaultResponse.getChatFinishSpeak());
            break;
        }
        break;

```

Code-2-1

```

public Geo(String location)
{
    String url = "http://maps.google.com/maps/geo?q="+location+"&output=csv";
    try
    {
        DefaultHttpClient client = new DefaultHttpClient();
        HttpRequest request = new HttpGet(url);
        HttpResponse Response = client.execute(request);
        HttpEntity Entity = Response.getEntity();
        InputStream stream = Entity.getContent();
        DataInputStream ds = new DataInputStream(stream);

        String toLine= ds.readLine();
        String geos[] = toLine.split(",");
        this._latitude = geos[2];
    }
}

```

```

        this._longtitude = geos[3];
        Log.v("Message", toLine);
    }
    catch (Exception ex)
    {
        System.out.println(ex.toString());
    }
}

```

Code-3-1

```

case 3:// weather
    if (getThirdInfo(response).equals("0")) { // local
        gw = new GoogleWeather(getLatitude(), getLongitude());
    }
    if (getThirdInfo(response).equals("1")) { // other place
        String cityName = getFourthInfo(response);
        geo = new Geo(cityName);
        gw = new GoogleWeather(Double.parseDouble(geo
            .get_latitude()), Double.parseDouble(geo
            .get_longitude()));
    }
    switch (Integer.parseInt(getSecondInfo(response))) {
        default: // local today weather
            persendInfo("he", "Here's the forecast for recent days:");
            ww = gw.getWeatherWeek();
            DetailEntity d2 = ww;
            list.add(d2);
            break;
        case 1: // local weather for recent days
            persendInfo("he", "Here's the weather for today:");
            dw=gw.getWeatherDetail(String.valueOf(getSecondInfo(response)));
            DetailEntity d3 = dw;
            list.add(d3);
            break;
    }
    break;

```

Code-4-1

```

case 4:// WikiPedia

String wikiWords = getSecondInfo(response);
persendInfo("he", "Searching for wikipedia");

Uri myBlogUri = Uri.parse("http://www.wikipedia.org/search-
redirect.php?search="+wikiWords+"&language=en&go=++& " + "++&go=Go");
Intent returnwikiIt = new Intent(Intent.ACTION_VIEW, myBlogUri);
startActivity(returnwikiIt);
break;

```

Code-5-1

```

public List<Person> getPerson(Context context)
{
    Uri uri=ContactsContract.Data.CONTENT_URI;
    Cursor cursor=context.getContentResolver().query(uri, null, null, null,
    "display_name");
    cursor.moveToFirst();
    List<Person> list=new ArrayList<Person>();
    int Index_CONTACT_ID = cursor.getColumnIndex(ContactsContract.Data.CONTACT_ID);
    int Index_DATA1 = cursor.getColumnIndex(ContactsContract.Data.DATA1);
    int Index_MIMETYPE = cursor.getColumnIndex(ContactsContract.Data.MIMETYPE);
    while(cursor.getCount()>cursor.getPosition())
    {
        Person person = null;
        String id=cursor.getString(Index_CONTACT_ID);
        String info=cursor.getString(Index_DATA1);
        String mimeType=cursor.getString(Index_MIMETYPE);

        for(int n = 0; n<list.size(); n++)
        {
            if(list.get(n).getID() != null)
            {
                if(list.get(n).getID().equals(id))
                {
                    person = list.get(n);
                    break;
                }
            }
        }

        if(person == null)
        {
            person=new Person();
            person.setID(id);
            list.add(person);
        }
        if(mimeType.equals("vnd.Android.cursor.item/email_v2"))
        {
            person.setEmail(info);
        }
        else if(mimeType.equals("vnd.Android.cursor.item/postal-address_v2"))
        {
            person.setAddress(info);
        }
        else if(mimeType.equals("vnd.Android.cursor.item/phone_v2"))
        {
            person.addPhone(info);
        }
        else if(mimeType.equals("vnd.Android.cursor.item/name"))
        {
            person.setName(info);
        }
        cursor.moveToNext();
    }
}

```

Code-5-2

```

        return list;
    }

```

Code-5-3

```

case 5:// call
    personList = getContacts.getPerson(MainActivity.this);
    String callName = getName(response);
    if (isNameFound(callName, personList)) {
        String tele = getTele(callName, personList);
        if (!tele.equals("0")) {
            persendInfo("he", "Calling to " + callName + ".");
            Intent myIntentDial = new Intent(Intent.ACTION_CALL,
                Uri.parse("tel:" + tele));

            startActivity(myIntentDial);
        } else {
            persendInfo("he", "Sorry, no telephone exist!");
        }
    } else {
        persendInfo("he",
            "Sorry, no person is found by the given name!");
    }
    break;

```

Code-6-1

```

case 6:// send SMS
    String smsName = getSecondInfo(response);
    String smsContent = getThirdInfo(response);
    if (smsName != null) {
        personList = getContacts.getPerson(MainActivity.this);
        if (isNameFound(smsName, personList)) {
            String tele = getTele(smsName, personList);
            if (!tele.equals("0")) {
                persendInfo("he", "Sending message to " + smsName);
                Log.v("tele", tele);
                Intent smsreturnIt = new Intent();
                smsreturnIt.setAction(Intent.ACTION_SENDTO);
                smsreturnIt.setData(Uri.parse("smsto:" + tele));
                smsreturnIt.putExtra("sms_body", smsContent);
                startActivity(smsreturnIt)
            } else {
                persendInfo("he", "Sorry, no telephone exist!");
            }
        } else {
            persendInfo("he", "Sorry, no person is found by the given name!");
        }
    }
    break;

```

Code-7-1

```

case 7:// send Email
    personList = getContacts.getPerson(MainActivity.this);
    String mailReceiver = getSecondInfo(response);
    String mailBody = getThirdInfo(response);
    if (isNameFound(mailReceiver, personList)) {
        String mailAddress = getMailAddress(mailReceiver,personList);
        if (!mailAddress.equals("0")) {
            sendInfo("he", "Sending email to " + mailReceiver
                + ".");
            Intent intent = new Intent(
                Android.content.Intent.ACTION_SEND);
            intent.setType("plain/text");
            intent.putExtra(Android.content.Intent.EXTRA_EMAIL,
                mailReceiver);
            intent.putExtra(Android.content.Intent.EXTRA_TEXT,
                mailBody);
            startActivity(Intent.createChooser(intent,
                "Choose Email Client"));
        } else {
            sendInfo("he", "Sorry, no mail box exist!");
        }
    } else {
        sendInfo("he", "Sorry, no person is found by the given name!");
    }
    break;

```

Code-8-1

```

case 8:// google search
    String googlWords = getSecondInfo(response);
    sendInfo("he", "Searching for google.");
    Uri searchUri = Uri
        .parse("http://www.google.com/search?hl=en&site=&source=hp&q="
            + googlWords + "&oq=" + googlWords);
    Intent returnIt = new Intent(Intent.ACTION_VIEW, searchUri);
    startActivity(returnIt);
    break;

```

Code-9-1

```

case 9:
    Calendar calendar = Calendar.getInstance();
    int hourOfDay = Integer.parseInt(getSecondInfo(response));
    int minute = Integer.parseInt(getThirdInfo(response));
    calendar.setTimeInMillis(System.currentTimeMillis());
    calendar.set(Calendar.HOUR_OF_DAY, hourOfDay);
    calendar.set(Calendar.MINUTE, minute);
    calendar.set(Calendar.SECOND, 0);
    calendar.set(Calendar.MILLISECOND, 0);
    Intent intentAlarm = new Intent(MainActivity.this,
        AlarmReceiver.class);
    PendingIntent pendingIntent = PendingIntent.getBroadcast(

```

INTELLIGENT VOICE ASSISTANT

```
        MainActivity.this, 0, intentAlarm, 0);
AlarmManager am;
am = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
am.set(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(),
        pendingIntent);
persendInfo("he", "Set up clock at " + hourOfDay + ":" + minute
        + " .");
break;
```

Code-10-1

```
}else if(Intent.ACTION_MEDIA_SCANNER_FINISHED.equals(action)){
    Cursor c2 = context.getContentResolver()
        .query(MediaStore.Audio.Media.EXTERNAL_CONTENT_URI,
            new String[]{MediaStore.Audio.Media.TITLE,
                MediaStore.Audio.Media.DURATION,
                MediaStore.Audio.Media.ARTIST,
                MediaStore.Audio.Media._ID,
                MediaStore.Audio.Media.DISPLAY_NAME },
            null, null, null);
    count2 = c2.getCount();
    count = count2-count1;

    c = context.getContentResolver()
        .query(MediaStore.Audio.Media.EXTERNAL_CONTENT_URI,
            new String[]{MediaStore.Audio.Media.TITLE,
                MediaStore.Audio.Media.DURATION,
                MediaStore.Audio.Media.ARTIST,
                MediaStore.Audio.Media._ID,
                MediaStore.Audio.Media.DISPLAY_NAME,
                MediaStore.Audio.Media.DATA},
            null, null, null);

    if (c==null || c.getCount()==0){
        System.out.println("No music in the library!");
    }
    c.moveToFirst();
    ids = new int[c.getCount()];
    titles = new String[c.getCount()];
    path = new String[c.getCount()];
    for(int i=0;i<c.getCount();i++){
        ids[i] = c.getInt(3);
        titles[i] = c.getString(0);
        path[i] = c.getString(5).substring(4);
        c.moveToNext();
    }
    isDone= true;
}
```

Code-10-2

```

case 1:// play song
    String songName = getThirdInfo(response);
    int songIds[] = scanSdReceiver.getIds();
    Intent intentMusic = new Intent();
    intentMusic.putExtra("_ids", songIds);
    int position = 0;
    if (!songName.equals("000")) {
        position = nameToId(songName,
            scanSdReceiver.getTitles(),
            scanSdReceiver.getIds());
    } else {
        Random r = new Random();
        position = r.nextInt(songIds.length);
        songName = idToName(position,
            scanSdReceiver.getTitles());
        persendInfo("he",
            DefaultResponse.playSongReponse(songName));
    }

    if (position != -1) {
        intentMusic.putExtra("position", position);
        intentMusic.putExtra("op", 1);
        intentMusic.setAction("com.Android.MusicService");
        startService(intentMusic);
        persendInfo("he",
            "Playing song " + songName + ".");
        Log.v("Music", "Play song :" + songName
            + ", position: " + position);
    } else {
        persendInfo("he",
            "Sorry, I can't find the song on your device.");
    }
    break;

```

Code-10-3

```

case 2:
    Intent intent1 = new Intent();
    intent1.setAction("com.Android.MusicService");
    intent1.putExtra("op", 2);
    startService(intent1);
    Log.v("Music", "pause song");
    break;
case 3:
    Intent intent2 = new Intent();
    intent2.setAction("com.Android.MusicService");
    intent2.putExtra("op", 3);
    startService(intent2);
    Log.v("Music", "Stop song");
    break;

```

Code-13

```
case 13:
    Intent in = new Intent();
    in.setClass(MainActivity.this, HelpActivity.class);
    startActivity(in);
    break;
```

Code-14

```
case 14:
    String languagecode = getSecondInfo(response);
    String text = getThirdInfo(response);
    persendInfo("he", DefaultResponse.getTranslateSuccessSpeak());
    Translator ts = new Translator(text,languagecode);
    DetailTranslate dt = new
    DetailTranslate(languagecode,ts.getTranslateResult(),
    text,R.layout.translate);
    list.add(dt);
    break;
```