**FINAL REPORT**

**Of**

**MINOR PROJECT**

**On**

**“STUDENT CHATBOT USING MICROSOFT AZURE”**

****

**Submitted in partial fulfillment for the award of the degree of**

**MASTER OF COMPUTER APPLICATIONS**

**(Batch 2017-2019)**

Submitted By: Submitted To:

**Kanav Singla (17mca8079)** Ms. Rajanpreet

Assistant professor (UIC)

Chandigarh University

Branch/Group: MCA 4th / 1st

**CANDIDATE’S DECLARATION**

I hereby declare that the work which is being presented in the dissertation, entitled **“STUDENT CHATBOT USING MICROSOFT AZURE”**, in partial fulfillment of the requirements for the award of the degree of **Master of Computer application** and submitted to **CHANDIGARH UNIVERSITY, Gharuan, Mohali** is an original piece of research work carried out by me during the period from 2017 to 2019 under the supervision of Ms. Rajanpreet ma’am, Assistant Professor, Department of University Institute of Computing, Chandigarh University, Gharuan, Mohali, Punjab, India.

The matter embodied in this thesis has not been submitted by me for the award of any other degree of any other University/Institute.

|  |
| --- |
| **Student name:**  **Kanav Singla (17mca8079)** |

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

|  |  |
| --- | --- |
| **Supervisor** |  |

**ACKNOWLEDGEMENT**

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to Ms. Rajanpreet ma’am for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents & member of University Institute of Computing, Chandigarh University for their kind co-operation and encouragement which help me in completion of this project.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

|  |  |
| --- | --- |
| Date: 13-April-2018  Place: Chandigarh University,  Gharuan | MCA |

**ABSTRACT**

* Azure Bot Service speeds up development by providing an integrated environment that is purpose-built for bot development with the Microsoft Bot Framework connectors and BotBuilder SDKs. Developers can get started in seconds with out-of-the-box templates for scenarios including basic, form, language understanding, question and answer, and proactive bots.
* Give your bot some super powers. Go beyond a great conversationalist to a bot which can recognise a user in photos, moderate content, make smart recommendations, translate language and more. Cognitive Services enable your bot to see, hear and interpret in more human ways.
* Your users talk in many places, your bot should too. Azure Bot Service can be integrated across multiple channels to increase interactions and reach more customers using your website or app to email, GroupMe, Facebook Messenger, Kik, Skype, Slack, Microsoft Teams, Telegram, text/SMS, Twilio, Cortana, and Skype for Business.

.

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TOPIC** | **PAGE No.** |
|  | *Candidate’s Declaration* | *I* |
| *Acknowledgement* | *II* |
| *Abstract* | *III* |
| *Contents* | *IV – V* |
| *List of Figures* | *VI* |
| **1.** | **INTRODUCTION** | **1** |
| **2.** | **IDEA** | **1** |
| **3.** | **OBJECTIVES** | **2** |
| **4.** | **SOFTWARE DEVELOPMENT LIFE CYCLE**  4.1 Software development life cycle process  4.1.1 Requirement Analysis  4.1.2 Problem Analysis  4.1.3 System Analysis  4.1.4 Initial Investigation  4.1.5 Information Gatherin | **2 – 42**  2  2 - 3  4  4 - 5  5 - 6  6 - 7 |
| **5** | **EXISTING SYSTEM** | **7** |
| **6** | **PROPOSRD SYSTEM ALONG WITH INTENDED OBJECTIVES**  6.1 Proposed Objective  6.2 Project Scope | **7**  7  7 |
| **7** | **FEASIBILITY STUDY**  7.1 Technical Feasibility  7.2 Economic Feasibility  7.3 Operational Feasibility | **8 – 9**  8  9  9 |
| **8** | **TEAM STRUCTURE** | 9 |
| **9** | **SYSTEM REQUIREMENT**  9.1 Software Specifications  9.2 Hardware Requirements | **9 – 10**  9 – 10  10 |
| **10** | **TECHNOLOGY USED**  10.1 Front End  10.1.1 Android  10.1.2 Features of Android  10.1.3 Android SDK  10.1.4 Android SDK Features  10.2 Back End  10.2.1 Overview of Firebase  10.2.2 Features of Firebase | **10 – 20**  10 – 15  10 – 11  12 – 14  14 – 15  15  16 – 20  16 -17  17 – 20 |
| **11** | **DATA FLOW DIAGRAMS**  11.1 Components of DFD  11.2 Project DFD  11.3 User DFD | **20 – 23**  20 – 21  22  23 |
|  | **REFERENCES** | **34** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **TITLE** | **FIGURE NO.** | **.** |

|  |
| --- |
| Machine Learning Examples 1.1 |
| Apple Online Support 1.2 |
| Microsoft Online Support 1.3 |
| Microsoft Bot Framework 1.4 |
| Language Understanding Intelligence Service `.5 |
| Sample Usage of Translator 1.6 |
| Example Machine Learning Model. 1.7 |
| Main Components of the Bot. 1.8 |
| Sample LUIS labeling 1.9 |
| LUIS Dashboard 1.10 |
| The Logic for Connecting with Translator 1.11 |
| The Recommender Engine 1.12 |
| Recommender Engine as Web Service 1.13 |
| LUIS Performance on Gaming Device 1.14 |
| LUIS Performance on 2-in-1 1.15 |

1. **Introduction**

* Bot Service provides what you need to build, connect, test, deploy, monitor, and manage bots. Bot Service provides the core components for creating bots, including the Bot Builder SDK for developing bots and the Bot Framework for connecting bots to channels.
* Bot Service provides an integrated environment purpose-built for bot development. You can write a bot, connect, test, deploy, and manage it from your web browser with no separate editor or source control required. For simple bots, you may not need to write code at all. It is powered by the Bot Framework and it provides two [hosting plans](https://docs.microsoft.com/en-us/azure/bot-service/bot-service-overview-readme#hosting-plans):

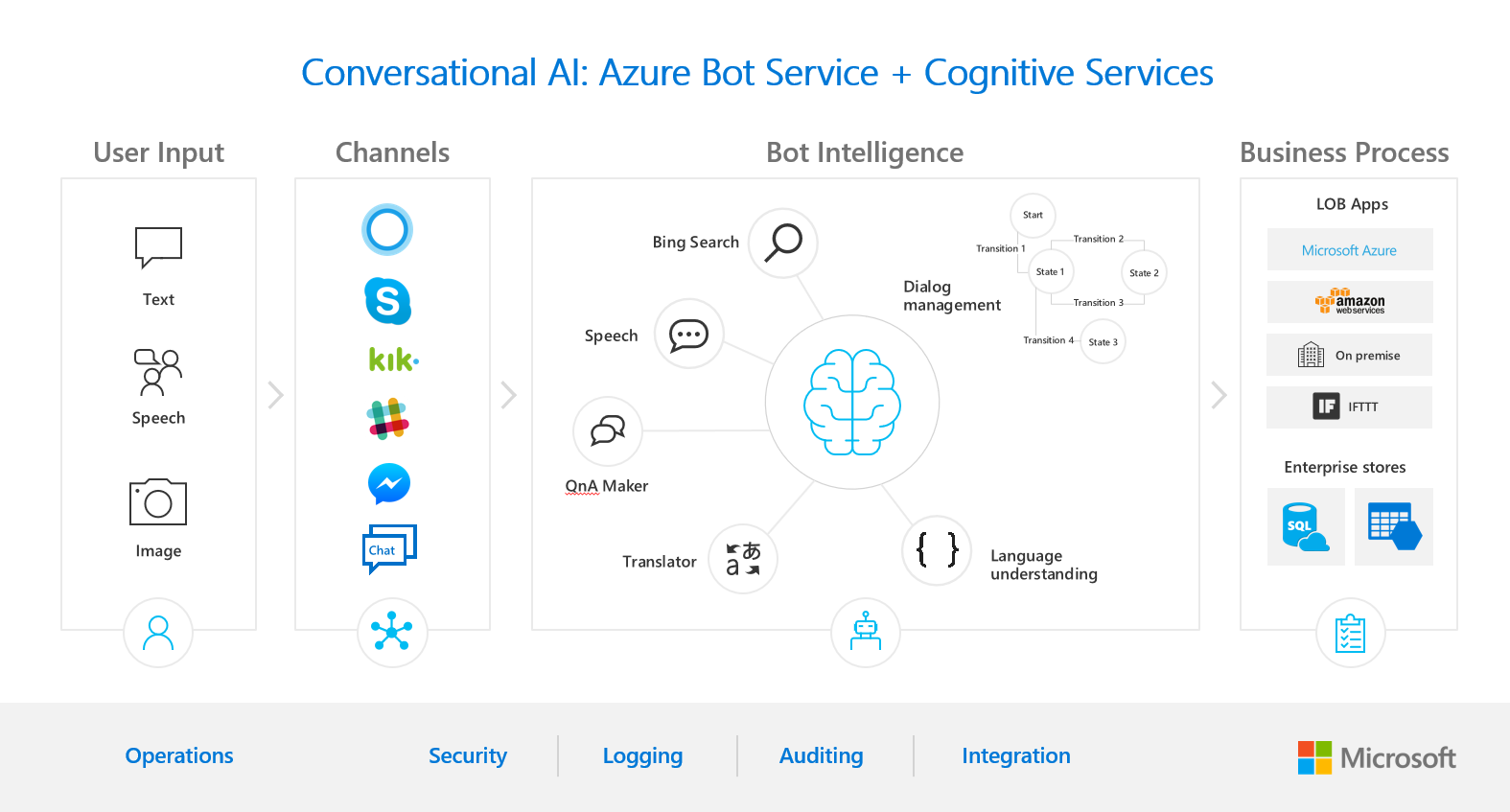
With the App Service plan, a bot is a standard Azure web app you can set to allocate a predefined capacity with predictable costs and scaling.

With a Consumption plan, a bot is a serverless bot that runs on Azure Functions and uses the pay-per-run Azure Functions pricing.

* Bot Service accelerates bot development with [Five bot templates](https://docs.microsoft.com/en-us/azure/bot-service/bot-service-concept-templates) you can choose from when you create a bot. You can further modify your bot directly in the browser using the Azure editor or in an Integrated Development Environment (IDE), such as Visual Studio and Visual Studio Code.

## What is a bot?

* Think of a bot as an app that users interact with in a conversational way. Bots can communicate conversationally with text, cards, or speech. A bot may be as simple as basic pattern matching with a response, or it may be a sophisticated weaving of artificial intelligence techniques with complex conversational state tracking and integration to existing business services.
* Bot Service enables you to build bots that support different types of interactions with users. You can design conversations in your bot to be freeform. Your bot can also have more guided interactions where it provides the user choices or actions. The conversation can use simple text strings or more complex rich cards that contain text, images, and action buttons. And you can add natural language interactions, which let your users interact with your bots in a natural and expressive way.



1. **Idea** 
   * + - **Multiple language support**. Bot Service leverages Bot Builder with support for .NET and Node.js.
     + **Bot templates**. Bot Service templates allow you to quickly create a bot with the code and features you need. Choose from a Basic bot, a Forms bot for collecting user input, a Language understanding bot that leverages LUIS to understand user intent, a QnA bot to handle FAQs, or a Proactive bot that alerts users of events.
     + **Bring your own dependencies**. Bots support NuGet and NPM, so you can use your favorite packages in your bot.
     + **Flexible development**. Code your bot right in the Azure portal or set up continuous integration and deploy your bot through GitHub, Visual Studio Team Services, and other supported development tools. You can also publish from Visual Studio.
     + **Connect to channels**. Bot Service supports several popular channels for connecting your bots and the people that use them. Users can start conversations with your bot on any channel that you've configured your bot to work with, including Skype, Facebook, Teams, Slack, SMS, and several others.
     + **Tools and services**. Test your bot with the Bot Framework Emulator and preview your bot on different channels with the Channel Inspector.

**3) Objectives**

1.User does not have to go personally to college office for the enquiry.

2.This application enables the students to be updated with college cultural activities.

3.This application saves time for the student as well as teaching and non-teaching staffs.

**4) SOFTWARE DEVELOPMENT LIFE CYCLE**

**4.1) SOFTWARE DEVELOPMENT LIFE CYCLE PROCESS**

The systems development life cycle (SDLC), also referred to as the application development life-cycle, is a term used in [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering), [information systems](http://en.wikipedia.org/wiki/Information_systems) and [software engineering](http://en.wikipedia.org/wiki/Software_engineering) to describe a process for planning, creating, testing, and deploying an information system. The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.

A systems development life cycle is composed of a number of clearly defined and distinct work phases which are used by systems engineers and systems developers to plan for, design, build, test, and deliver [information systems](http://en.wikipedia.org/wiki/Information_system). Like anything that is manufactured on an assembly line, an SDLC aims to produce high quality systems that meet or exceed customer expectations, based on customer requirements, by delivering systems which move through each clearly defined phase, within scheduled time-frames and cost estimates. Computer systems are complex and often (especially with the recent rise of [service-oriented architecture](http://en.wikipedia.org/wiki/Service-oriented_architecture)) link multiple traditional systems potentially supplied by different software vendors. To manage this level of complexity, a number of SDLC models or methodologies have been created, such as "[waterfall](http://en.wikipedia.org/wiki/Waterfall_model)"; "[spiral](http://en.wikipedia.org/wiki/Spiral_model)"; "[Agile software development](http://en.wikipedia.org/wiki/Agile_software_development)"; "[rapid prototyping](http://en.wikipedia.org/wiki/Software_prototyping#Throwaway_prototyping)"; "[incremental](http://en.wikipedia.org/wiki/Incremental_development)"; and "synchronize and stabilize.

**4.1.1) REQUIREMENT ANALYSIS**

Requirements analysis in [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering) and [software engineering](http://en.wikipedia.org/wiki/Software_engineering), encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting [requirements](http://en.wikipedia.org/wiki/Requirement) of the various [stakeholders](http://en.wikipedia.org/wiki/Stakeholder_%28corporate%29), analyzing, documenting, validating and managing software or system requirements.

Requirements analysis is critical to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

A key question is what must be done to solve the problem? One aspect of the analysis is defining the boundaries of the system and improving the shortcomings of the existing system.

During analysis, data are collected on available files, websites, decision points and the way the present system is working.. There are many tools, which are available to do the analysis part like interview, questionnaire, surfing on net and understanding the user needs, fact analysis and review of written documents after all these tools flow charts or decision tables are built which clearly shows the problem. The collection of required and proper data is very important in analysis. The analysis require special skill to the subject in order to collect the data.

Training, experience and common sense are required for collection of the information needed to do the analysis. The environment in which analysis is carried out plays an important role that is how the system analyst deals with the user requirements or demands. With the corporation of our team and satisfaction of the user analysis part can be successfully taken. In analysis, we design the system logically that is how the system will look like, after the analysis part we move from logical system to physical system that is how actually the system will be. So after analysis that is logical there is a physical part called design that is very important stage in system development life cycle.

Once analysis is completed, the analyst has a firm understanding of what is to be done. The next step is to decide how the problem might be solved. Thus, in system design, we move from the logical to the physical aspects of the life cycle.

**4.1.2) Problem Analysis**

Analysis is detailed study of various operation performed by the system and their relationship within and outside the system. Outside factor also plays major role in the system like government, vendors, customers etc. A key question is what must be done to solve the problem? One aspect of the analysis is defining the boundaries of the system and determining whether or not a candidate system should consider other related systems.

During analysis, data are collected on available files, decision points and transaction handled by the present system. There are many tools, which are available to do the analysis part like interview, quiz questionnaire, fact analysis and review of written documents after all these tools flow charts or decision tables are built which clearly shows the problem. The interview is a commonly used tool in analysis. The analysis require special skill to the subject being interview and the way in which interview is taken. Bias in data collection and interpretation can be a problem. Training, experience and common sense are required for collection of the information needed to do the analysis. The environment in which analysis is carried out plays an important role that is how the system analyst deals with the interviewing staff. With the corporation and satisfaction of the user staff analysis part can be successfully taken. In analysis, we design the system logically that is how the system will look like, after the analysis part we move from logical system to physical system that is how actually the system will be. So after analysis that is logical there is a physical part called design that is very important stage in system development life cycle.

Once analysis is completed, the analyst has a firm understanding of what is to be done. The next step is to decide how the problem might be solved. Thus, in system design, we move from the logical to the physical aspects of the life cycle.

**4.1.3) System Analysis**

The term system refers to an orderly grouping of interdependent components linked together according to a plan to achieve a specific objective. The idea of system becomes most practical and necessary in conceptualizing the interrelationship and integration of operations especially when using computers. Thus, a system is a way of thinking of organizations and their problems. It involves a set of techniques that helps in solving the problems. A system has many elements these are input and output processors, control, feedback, environment, boundary and interface. System takes input from outside and gives output to outside. So practically system is an open system that interacts with the environment.

**A study of system concept has three basic implications**

* A system must be designed to achieve a predetermined objective.
* Interrelationship and interdependence must exist among the component.
* The objective of the organization as a whole has higher priority than the objective of its sub system for example computerizing personal applications must conform to the organization policy on privacy, confidentiality and security, as well as making selective data (pay roll) available to the accounting division on request.

In the present study, a system is an integrated collection of programs and data files. The combination of all these programs and databases made this system. The process of designing a system is not done in one phase.

**System analysis further consist of following steps**

* SYSTEM STUDY AND PROBLEM DEFINITION.
* SYSTEM DESIGN AND PROGRAMMING
* IMPLEMENTATION AND DOCUMENTATION
* Initial Investigation
* Information Gathering

**4.1.4) INITIAL INVESTIGATION**

This is the first step in system development life cycle. For making a project, it is very important to understand the overall working of the system. To know about the overall working of the system initial investigation is carried out. Initial investigation helps to identify the need.

**4.1.5) INFORMATION GATHERING**

After defining the problem, the next step in the system analysis is to gather information. Information gathering is an art and science. The approach and manner in which information is gathered require persons with sensitivity, commonsense and knowledge of what and when to gather and what channels to use in securing information. This means that information getting is neither easy nor routine. Much preparation, experience and training are required. Before one determines where to go for information or what tool to use, the first requirement is to figure out what information to gather. The approach and the manner in which the information is gathered require a person with sensitivity, skill and common sense of the following: -

**Purpose of gathering information**

* There are three main purposes of gathering information
* We get familiar with the News Agency and the present system through available procedure manuals, officials documents and by knowing the flow of information.
* Knowing how information is to be accessed from the system and subsequently making the access system simple and easy to use.

**Kind of information needed**

* Before one determines where to go and what tool to use, first requirement is to figure out what information to gather. The basic information required is how organization of personnel or official contacts as well as appointment takes place.

**Sources of information**

* Information is gathered from two main sources namely personal and written documents within the police stations.
* Two main sources of information are: -
* External sources
* Internal sources.

**5) Existing System**

1.Emanuela Haller and Traian Rebedea, “Designing a Chat-bot that Simulates an Historical Figure”, IEEE Conference Publications, July 2013. There are many applications that are incorporating a human appearance and intending to simulate human dialog, but in most of the cases the knowledge of the conversational bot is stored in a database created by a human experts. However, very few researches have investigated the idea of creating a chat-bot with an artificial character and personality starting from web pages or plain text about a certain person. This paper describes an approach to the idea of identifying the most important facts in texts describing the life (including the personality) of an historical figure for building a conversational agent that could be used in middle-school CSCL scenarios.

2. Maja Pantic, Reinier Zwitserloot, and Robbert Jan Grootjans, “Teaching Introductory Artificial Intelligence Using Asimple Agent Framework”, IEEE Transactions On Education, Vol. 48, No. 3, August 2005. This paper describes a flexible method of teaching introductory artificial intelligence (AI) using a novel, Java-implemented, simple agent framework developed specifically for the purposes of this course. Although numerous agent frameworks have been proposed in the vast body of literature, none of these available frameworks proved to be simple enough to be used by first-year students of computer science. Hence, the authors set out to create a novel framework that would be suitable for the aims of the course, for the level of computing skills of the intended group of students, and for the size of this group of students. The content of the introductory AI course in question is a set of assignments that requires the students to use intelligent agents and other AI techniques to monitor, filter, and retrieve relevant information from the World Wide Web. It represents, therefore, a synthesis of the traditional objectivist approach and a real-world-oriented, constructivist approach to teachingprogramming to novices. The main aim of implementing such a pedagogy was to engage the students in learning to which they personally relate while attaining intellectual rigor. Classroom experience indicates that students learn more effectively when the traditional objectivist approach is combined with a constructivist approach than when this orthodox approach to teaching programming to novices is used alone.

**6) Proposed System along with intended objectives**

A Student bot project is built using artificial algorithms that analyzes user’s queries and understand user’s message. This System is a web application which provides answer to the query of the student. Students just have to query through the bot which is used for chatting. Students can chat using any format there is no specific format the user has to follow. The System uses built in artificial intelligence to answer the query.

**Advantages**

* It is quick process and saves time
* Renewal can be done in their own place itself.

**6.1) Project Objective**

* To Increase number of subscriptions.
* Every time accessibility online.
* Providing best organizing to every user anywhere.

**6.2) Project Scope**

The answers are appropriate what the user queries. If the answer found to invalid, user just need to select the invalid answer button which will notify the admin about the incorrect answer. Admin can view invalid answer through portal via login System allows admin to delete the invalid answer or to add a specific answer of that equivalent question. The User can query any college related activities through the system. The user does not have to personally go to the college for enquiry. The System analyzes the question and then answers to the user. The system answers to the query as if it is answered by the person. With the help of artificial intelligence, the system answers the query asked by the students. The system replies using an effective Graphical user interface which implies that as if a real person is talking to the user. The user can query about the college related activities through online with the help of this web application. This system helps the student to be updated about the college activities.

**7) Feasibility Study**

The feasibility of the system can be judged according to its workability impact on the organization, ability to meet user needs and the effective use of resources. One should keep in mind the need of the user and how does a candidate system meet it.

**7.1) Technical Feasibility**

This Information System for Project Student Chatbot using Microsoft Azure serves the requirement of the organization and is very much technically feasible. We have technical guarantee and reliability as it has been tested by different programmers. This project can be run on computers having 256 mb RAM having Windows installed on it. It is easy to use even by the persons with little knowledge of computers.

**7.2) Economic Feasibility**

This system is economically feasible also as the cost of making the project is very low as of Azure free subscription .Cost in developing the software is very low, as compared to the money spent on the existing system. This project can be run on 1 GB RAM .

**7.3) Operational Feasibility**

The client’s requirement is also taken into consideration then the system is designed. Operational feasibility of Information System for Project Student Chatbot is satisfied as the running of this system satisfies the user and the client is also satisfied by the economic and technical feasibility.

**8) Team Structure**

Often a team of people is assigned a project. For team to work as a group and contribute most to the project, the people in the team have to be organized in some manner. This structure of Team has a direct impact on the product quality and project productivity. The structure of my team democratic. Democratic team consists of ten or fewer. The structure allowed input from all the members, which led to better decisions in difficult situations.

**My team comprises of following members:**

Kanav Singla(17MCA8079)

9) System Requirement:

9.1) Software Specification:

* Windows XP, Windows 7(ultimate, enterprise)
* Android SDK
* Eclipse
* Java
* Android OS 2.3 or Higher.

9.2) Hardware Requirements:

* Processor : Pentium, AMD or Higher Version.
* Operating System : Windows XP/ Windows 7
* RAM : 256 MB, 2GB recommended
* Hardware Devices : Keyboard with mouse
* Hard disk : 10GB or More
* Display : Standard Output Display

**Front End** : Microsoft Azure Bot Framework

**Back End** : Back end system will be Qna Maker by Microsoft

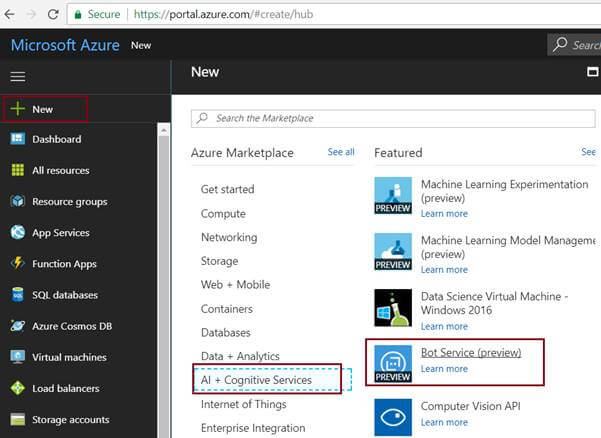
* + This will provide data to the User Interface.
  + Database

**10) TECHNOLOGY USED**

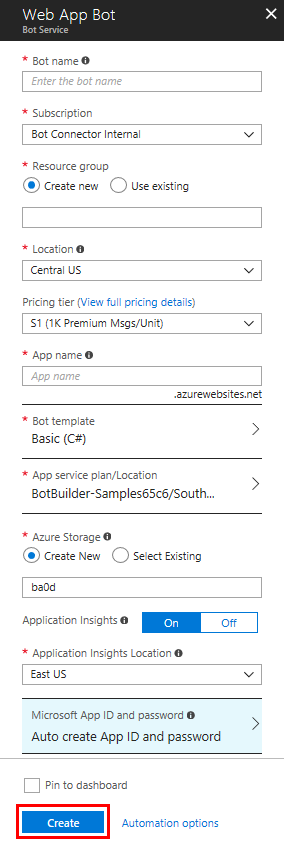
**10.1) Front End**

**10.1.1) Microsoft Azure Bot Framework**

* Bot Service provides what you need to build, connect, test, deploy, monitor, and manage bots. Bot Service provides the core components for creating bots, including the Bot Builder SDK for developing bots and the Bot Framework for connecting bots to channels.
  + - Cloud based student information Chatbot system is an artificial algorithm that analyzes the student’s queries and messages. This system has a built artificial intelligence to answer the query of the student. The answers are appropriate to the user’s queries, if the user find his answer to be invalid, he may select the invalid answer option button which will notify the admin. Admin can view invalid through portal via login. System allows admin to delete the invalid answer or to add a specific answer of that equivalent question.
    - Here, the database will be stored into the azure cloud which will form a connection between application and cloud server via internet. To answer to the student query, the Chatbot system retrieves the answer from the database which is stored in the cloud. The Chatbot system uses a specific keyword to retrieve the answers from the database. There is no format for the student to follow while asking any question in the Chatbot. The students can put up any query related to college activities through the system. This system replies to the user with Graphical user interface which implies that as if a real person is talking to the student. The system helps the student not only to get their queries answered but also to be updated with the college activities.



| Setting | Suggested value | Description |
| --- | --- | --- |
| **Bot name** | Your bot's display name | The display name for the bot that appears in channels and directories. This name can be changed at anytime. |
| **Subscription** | Your subscription | Select the Azure subscription you want to use. |
| **Resource Group** | myResourceGroup | You can create a new [resource group](https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-overview#resource-groups) or choose from an existing one. |
| **Location** | The default location | Select the geographic location for your resource group. Your location choice can be any location listed, though it's often best to choose a location closest to your customer. The location cannot be changed once the bot is created. |
| **Pricing tier** | F0 | Select a pricing tier. You may update the pricing tier at any time. For more information, see [Bot Service pricing](https://azure.microsoft.com/en-us/pricing/details/bot-service/). |
| **App name** | A unique name | The unique URL name of the bot. For example, if you name your bot myawesomebot, then your bot's URL will be http://myawesomebot.azurewebsites.net. The name must use alphanumeric and underscore characters only. There is a 35 character limit to this field. The App name cannot be changed once the bot is created. |
| **Bot template** | Basic | Choose either **C#** or **Node.js** and select the **Basic** template for this quickstart, then click **Select**. The Basic template creates an echo bot. [Learn more](https://docs.microsoft.com/en-us/azure/bot-service/bot-service-concept-templates) about the templates. |
| **App service plan/Location** | Your app service plan | Select an [app service plan](https://azure.microsoft.com/en-us/pricing/details/app-service/plans/) location. Your location choice can be any location listed, though it's often best to choose a location closest to your customer. (Not available for Functions Bot.) |
| **Azure Storage** | Your Azure storage account | You can create a new data storage account or use an existing one. By default, the bot will use [Table Storage](https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction#table-storage). |
| **Application Insights** | On | Decide if you want to turn [Application Insights](https://docs.microsoft.com/en-us/bot-framework/bot-service-manage-analytics) **On** or **Off**. If you select **On**, you must also specify a regional location. Your location choice can be any location listed, though it's often best to choose a location closest to your customer. |
| **Microsoft App ID and password** | Auto create App ID and password | Use this option if you need to manually enter a Microsoft App ID and password. Otherwise, a new Microsoft App ID and password will be created for you in the bot creation process. |

* 

Bot Service provides an integrated environment purpose-built for bot development. You can write a bot, connect, test, deploy, and manage it from your web browser with no separate editor or source control required. For simple bots, you may not need to write code at all. It is powered by the Bot Framework and it provides two [hosting plans](https://docs.microsoft.com/en-us/azure/bot-service/bot-service-overview-readme#hosting-plans):

With the App Service plan, a bot is a standard Azure web app you can set to allocate a predefined capacity with predictable costs and scaling.

With a Consumption plan, a bot is a serverless bot that runs on Azure Functions and uses the pay-per-run Azure Functions pricing.

Bot Service accelerates bot development with [Five bot templates](https://docs.microsoft.com/en-us/azure/bot-service/bot-service-concept-templates) you can choose from when you create a bot. You can further modify your bot directly in the browser using the Azure editor or in an Integrated Development Environment (IDE), such as Visual Studio and Visual Studio Code.

## What is a bot?

Think of a bot as an app that users interact with in a conversational way. Bots can communicate conversationally with text, cards, or speech. A bot may be as simple as basic pattern matching with a response, or it may be a sophisticated weaving of artificial intelligence techniques with complex conversational state tracking and integration to existing business services.

Bot Service enables you to build bots that support different types of interactions with users. You can design conversations in your bot to be freeform. Your bot can also have more guided interactions where it provides the user choices or actions. The conversation can use simple text strings or more complex rich cards that contain text, images, and action buttons. And you can add natural language interactions, which let your users interact with your bots in a natural and expressive way.

## Why use Bot Service?

Here are some key features of Bot Service:

**Multiple language support**. Bot Service leverages Bot Builder with support for .NET and Node.js.

**Bot templates**. Bot Service templates allow you to quickly create a bot with the code and features you need. Choose from a Basic bot, a Forms bot for collecting user input, a Language understanding bot that leverages LUIS to understand user intent, a QnA bot to handle FAQs, or a Proactive bot that alerts users of events.

**Bring your own dependencies**. Bots support NuGet and NPM, so you can use your favorite packages in your bot.

**Flexible development**. Code your bot right in the Azure portal or set up continuous integration and deploy your bot through GitHub, Visual Studio Team Services, and other supported development tools. You can also publish from Visual Studio.

**Connect to channels**. Bot Service supports several popular channels for connecting your bots and the people that use them. Users can start conversations with your bot on any channel that you've configured your bot to work with, including Skype, Facebook, Teams, Slack, SMS, and several others.

**Tools and services**. Test your bot with the Bot Framework Emulator and preview your bot on different channels with the Channel Inspector.

**Open source**. The Bot Builder SDK is open-source and available on [GitHub](https://github.com/microsoft/botbuilder).

## What is Bot Builder?

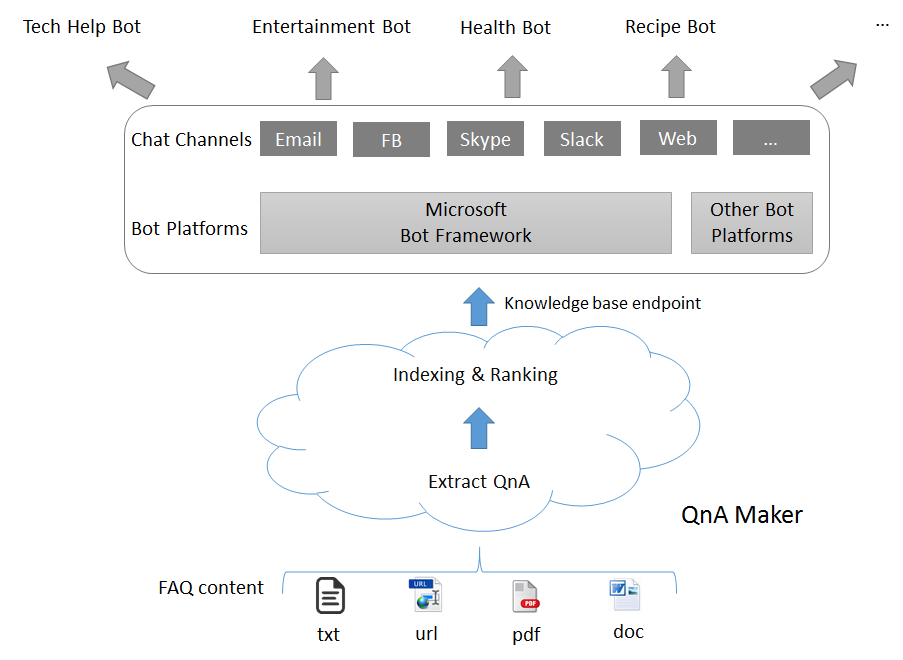
[Bot Builder](https://docs.microsoft.com/en-us/azure/bot-service/bot-builder-overview-getstarted) provides an SDK, libraries, samples, and tools to help you build and debug bots. When you build a bot with Bot Service, your bot is backed by the Bot Builder SDK. You can also use the Bot Builder SDK to create a bot from scratch using C# or Node.js. Bot Builder includes the Bot Framework Emulator for testing your bots and the Channel Inspector for previewing your bot's user experience on different channels.

**10.2) BACKEND QNA MAKER MICROSFT**

One of the basic requirements in writing your own Bot service is to seed it with questions and answers. In many cases, the questions and answers already exist in content like FAQ URLs/documents, etc.

[Microsoft QnA Maker](https://qnamaker.ai/) is a free, easy-to-use, REST API and web-based service that trains AI to respond to user's questions in a more natural, conversational way. Compatible across development platforms, hosting services, and channels, QnA Maker is the only question and answer service with a graphical user interface—meaning you don’t need to be a developer to train, manage, and use it for a wide range of solutions.

With optimized machine learning logic and the ability to integrate industry-leading language processing with ease, QnA Maker distills masses of information into distinct

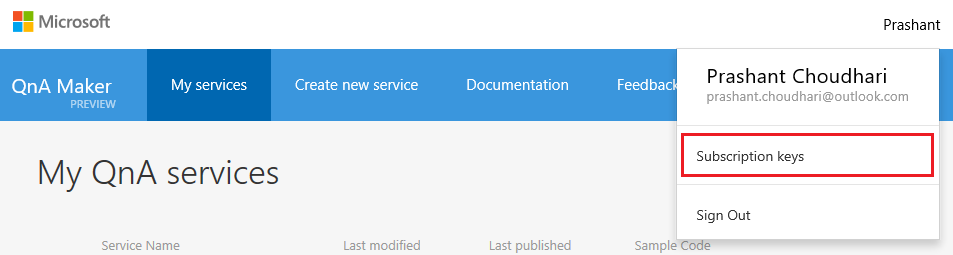
.pful answers.

# **Authentication & Subscription keys**

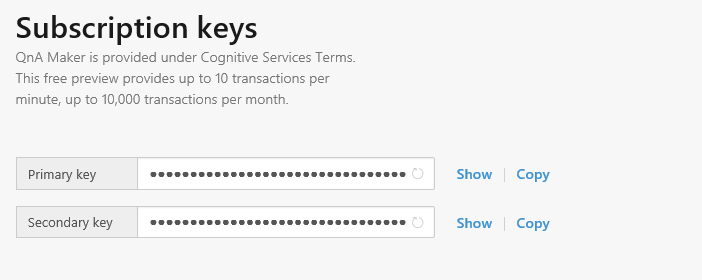
You will need a [Microsoft account](https://www.microsoft.com/en-us/account/) if you don't already have one, to sign in to the portal.

You will receive a unique pair of keys. The second one is just a spare. Please do not share the secret keys with anyone.

These subscription keys are used to track your usage of the service and need to be part of every request, as mentioned in the API section. To view your subscription keys, go to Settings.



Here you can view and also refresh your subscription keys, if you suspect they have been compromised.



Since currently the QnA Maker is a free to use tool, we have the following restrictions of usage per subscription key: **10,000 transactions per month, 10 per minute.** Beyond this your requests will be throttled.

# **API V2.0 Reference**

The new [V2.0 APIs](https://westus.dev.cognitive.microsoft.com/docs/services/58994a073d9e04097c7ba6fe/operations/58994a073d9e041ad42d9baa) let you programmatically manage your knowledge base. Now you can do the following with the APIs

* Create knowledge base
* Delete knowledge base
* Update knowledge base
* Download knowledge base
* Publish knowledge base

Refer [here](https://westus.dev.cognitive.microsoft.com/docs/services/58994a073d9e04097c7ba6fe/operations/58994a073d9e041ad42d9baa) for detailed documentation.

# **API V1.0 Reference**

Each published QnA Maker service is exposed as an HTTP endpoint that will take in a question and respond back with the best matched answer and a confidence score. You need the following two GUIDs to access your knowledge base via the HTTP endpoint

* Knowledge Base ID: This is auto-generated by the system for every published KB.
* Subscription Key: These are assigned per account, and is used for metering. See the subscription keys section for more details.

## Sample Request

### HTTP

POST /knowledgebases/<Your KB ID>/generateAnswer HTTP/1.1

Host: https://westus.api.cognitive.microsoft.com/qnamaker/v1.0

Ocp-Apim-Subscription-Key: <Your Subscription key>

Content-Type: application/json

Cache-Control: no-cache

{"question": "Question goes here"}

### C#

string responseString = string.Empty;

var query = “hi”; //User Query

var knowledgebaseId = “YOUR\_KNOWLEDGE\_BASE\_ID”; // Use knowledge base id created.

var qnamakerSubscriptionKey = “YOUR\_SUBSCRIPTION\_KEY”; //Use subscription key assigned to you.

//Build the URI

Uri qnamakerUriBase = new Uri("https://westus.api.cognitive.microsoft.com/qnamaker/v1.0");

var builder = new UriBuilder($"{qnamakerUriBase}/knowledgebases/{knowledgebaseId}/generateAnswer");

//Add the question as part of the body

var postBody = $"{{\"question\": \"{query}\"}}";

//Send the POST request

using (WebClient client = new WebClient())

{

//Set the encoding to UTF8

client.Encoding = System.Text.Encoding.UTF8;

//Add the subscription key header

client.Headers.Add("Ocp-Apim-Subscription-Key", qnamakerSubscriptionKey);

client.Headers.Add("Content-Type", "application/json");

responseString = client.UploadString(builder.Uri, postBody);

}

## Sample Response

The response to the above request will be a JSON with the answer and the confidence score (0-100).

### JSON

{ "Answer": "Sample response", "Score": "0" }

### C#

using Newtonsoft.Json;

private class QnAMakerResult

{

/// <summary>

/// The top answer found in the QnA Service.

/// </summary>

[JsonProperty(PropertyName = "answer")]

public string Answer { get; set; }

/// <summary>

/// The score in range [0, 100] corresponding to the top answer found in the QnA Service.

/// </summary>

[JsonProperty(PropertyName = "score")]

public double Score { get; set; }

}

//De-serialize the response

QnAMakerResult response;

try

{

response = JsonConvert.DeserializeObject< QnAMakerResult >(responseString);

}

catch

{

throw new Exception("Unable to deserialize QnA Maker response string.");

}

**11) Data Flow Diagram**

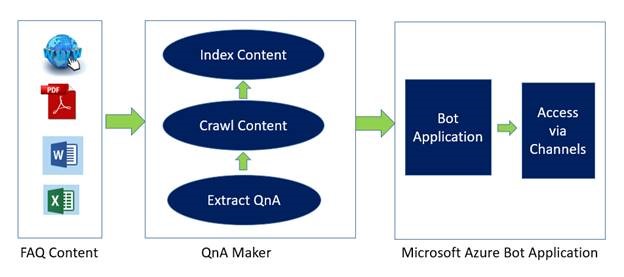
Data flow diagram is graphical tool used to describe and analyze the movement of data through system – manual or automated- including the process, stores of data and delays in the system. Data flow diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output through process may be described logically and independently of the physical components associated with the system. They are termed **logical data**

**Flow Diagrams**

**11.1) Components of DFD’s**

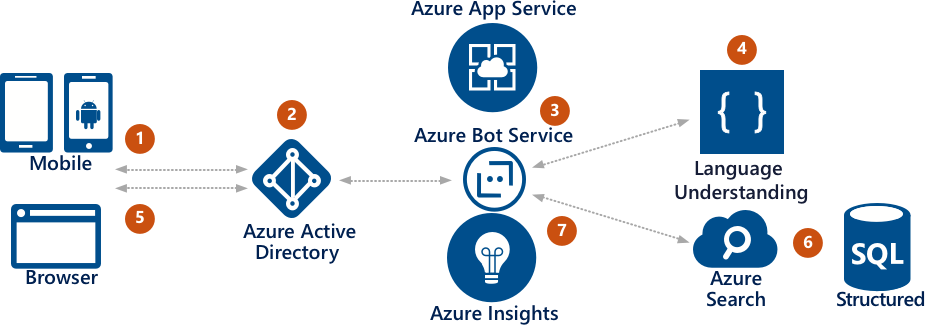
**Data Flow**: Data move in a specific direction from an origin to a destination in a form of a document letter, telephone call or virtually any medium.

* **Process**: People, procedures or devices that are used to produce data. The physical component is not identified
* **Source or Destination of Data**: External sources or destinations of data which may be people, programs, organizations or other entities, interact with the system but are outside its boundary. The term source and sink are interchangeable with origin and destination.
* **Data Store**: Here data are stored or referenced by a process in the system. The data store may represent computerized or non-computerized devices.

****

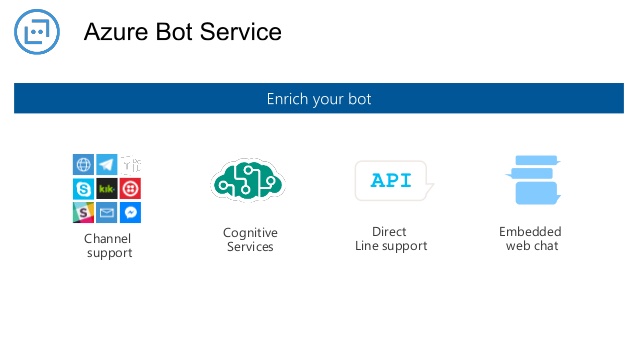
# **11.2) Project DFD**

# 

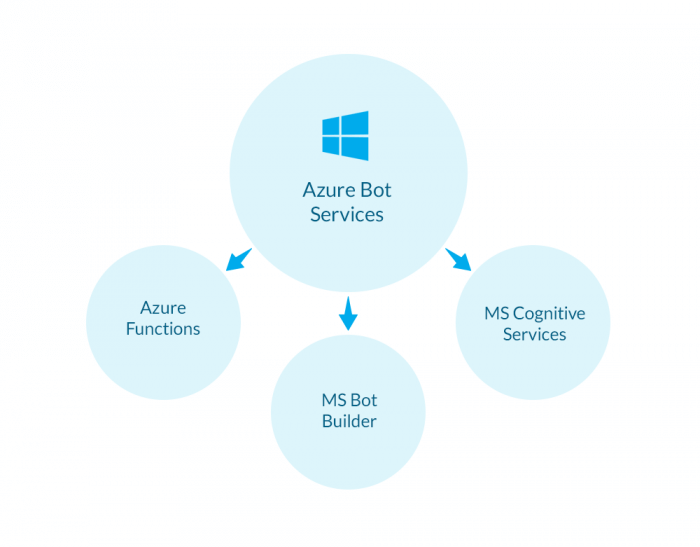


**Fig 1.3** System Flow Chart.

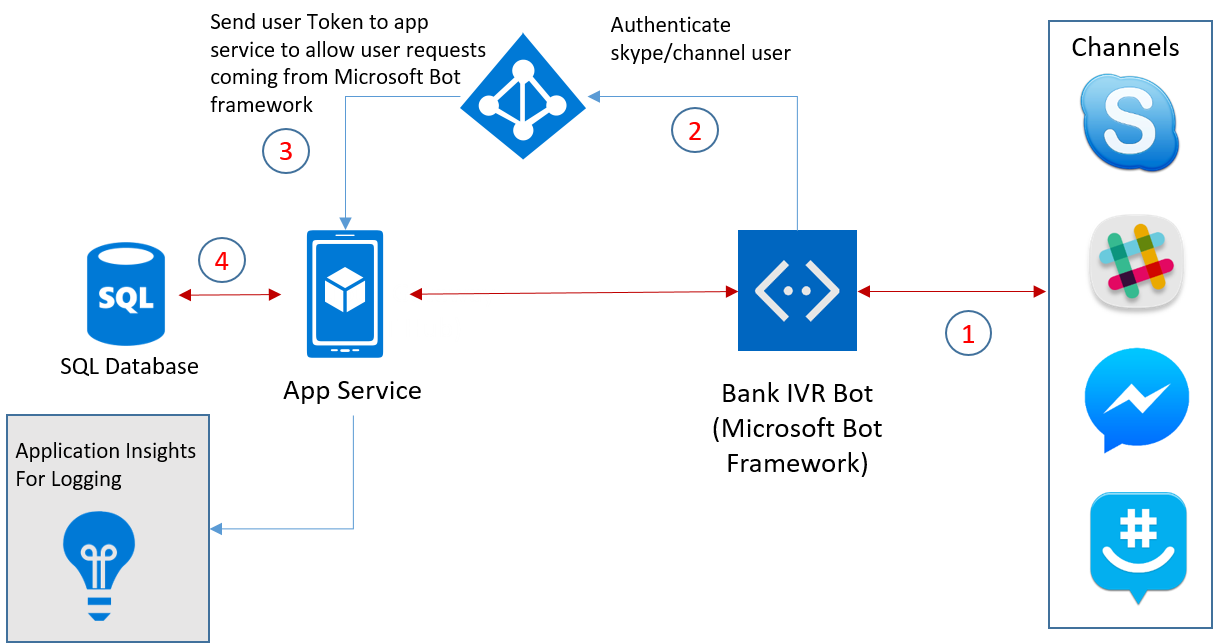
**11.3) USER DFD**

****

**Fig 1.4** Use Case Diagram for User

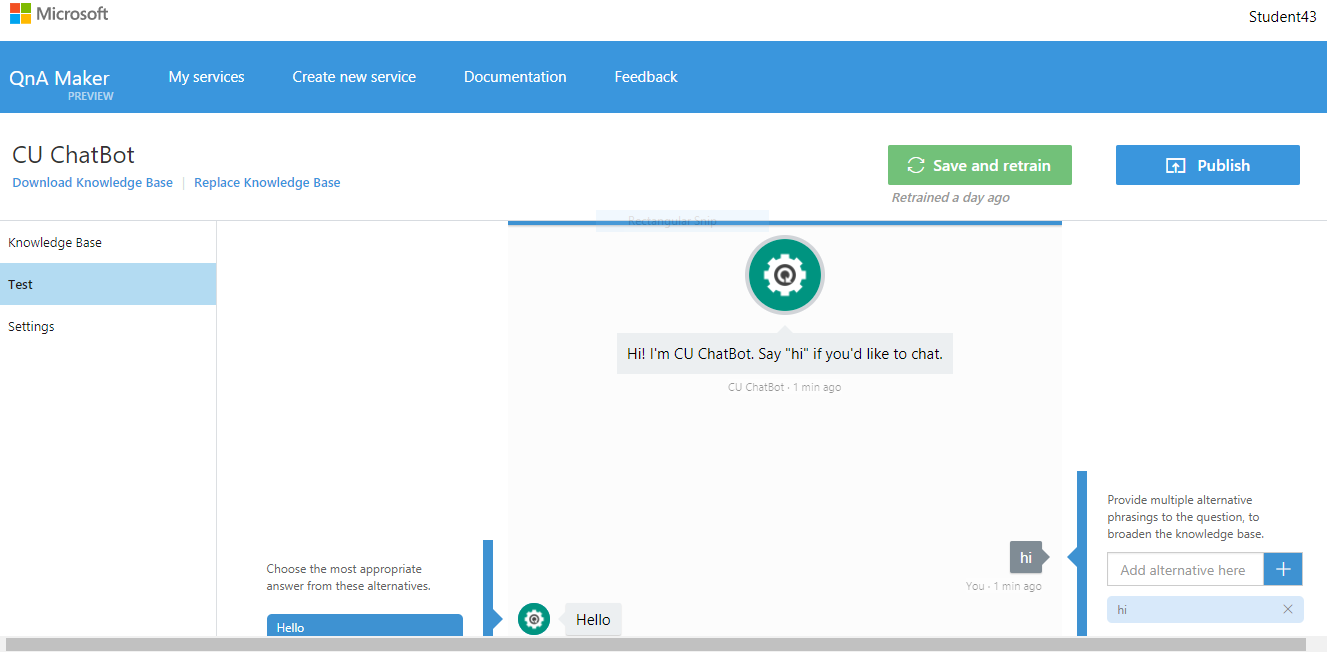


Here is some of the work that we have finished till now, it is not sufficient but we are doing our best to make this project enough interactive and user friendly. So, that we can make it worth selling in market.

****

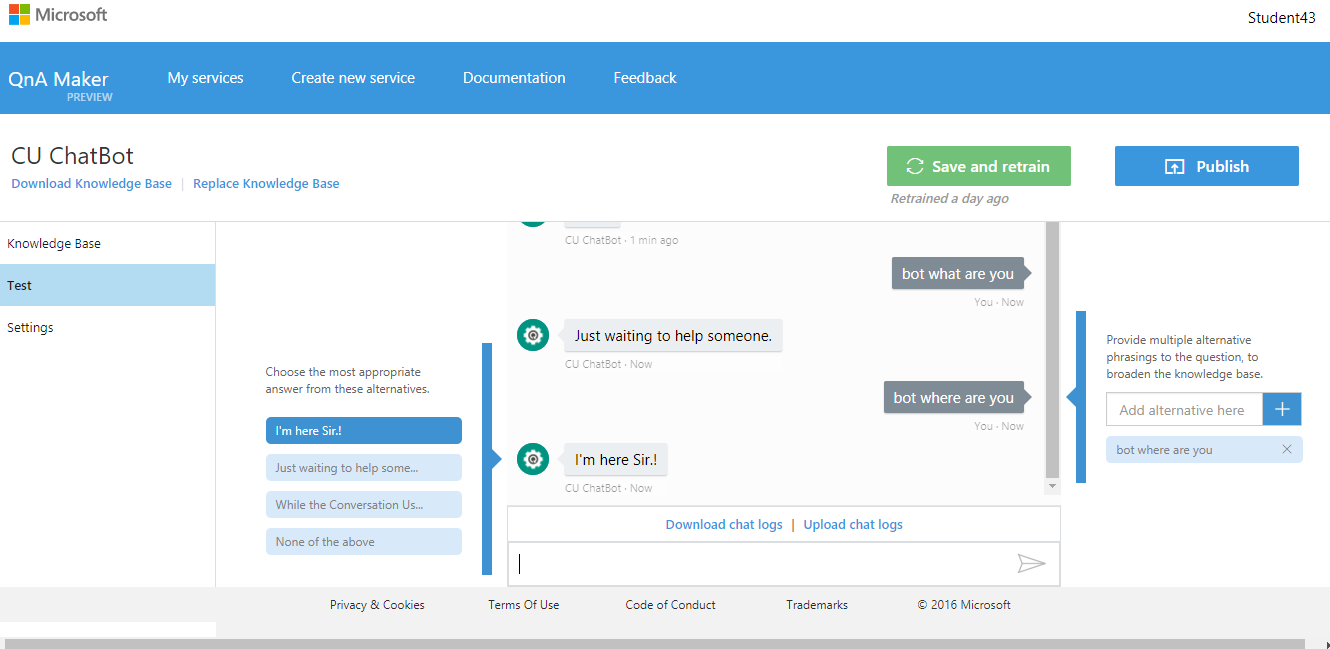
**STUDENT CHATBOT WORKING**

**Welcome Screen**

****

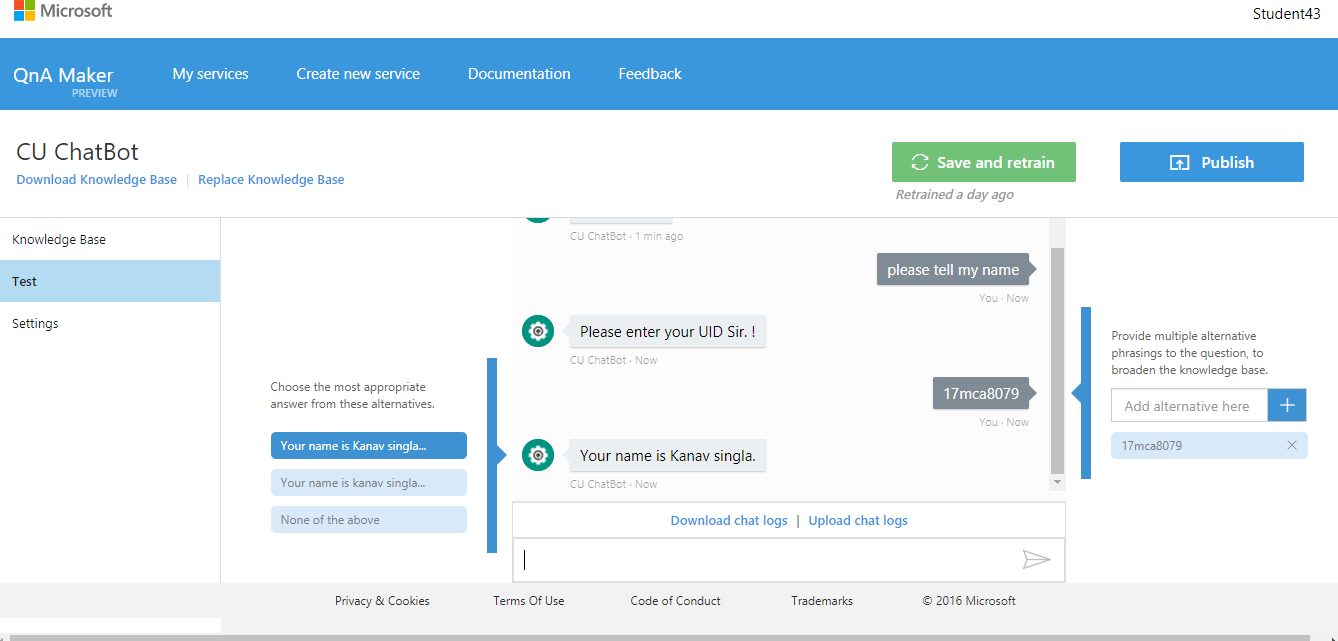
**Fig 1.5.1** Welcome Screen for Application

**Bot Interaction**

****

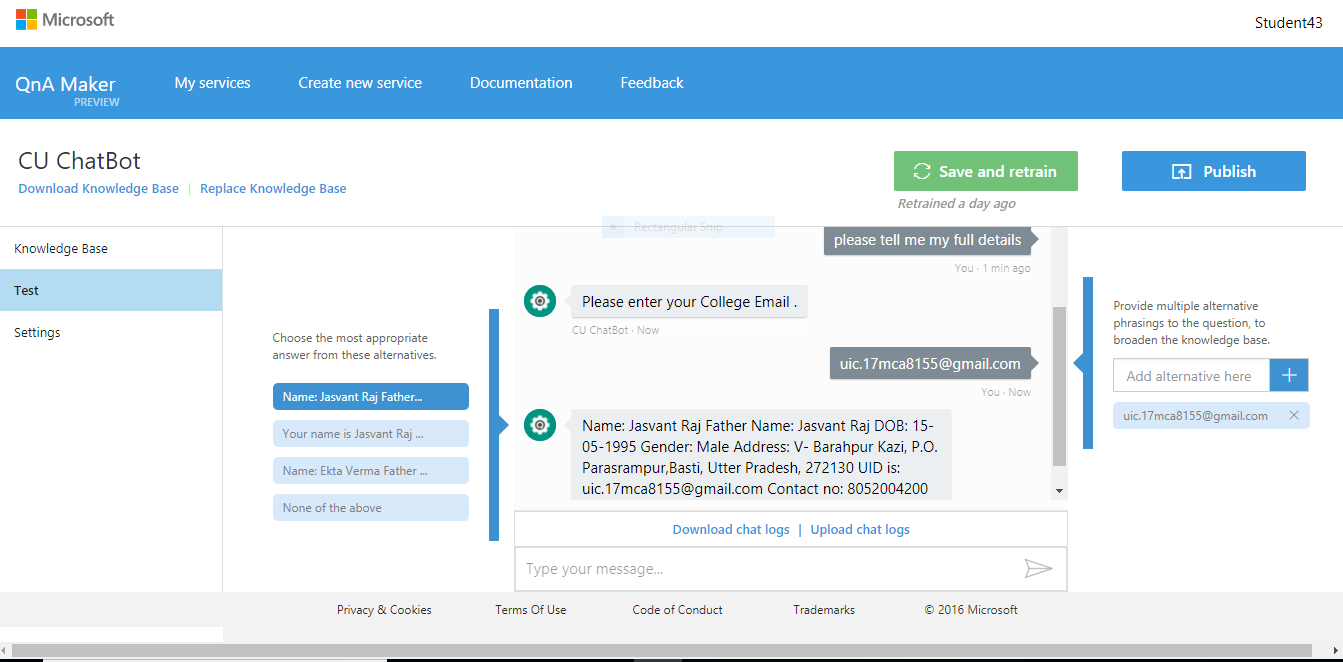
**Fig 1.5.2 Application Screen**

* **Student chatbot**

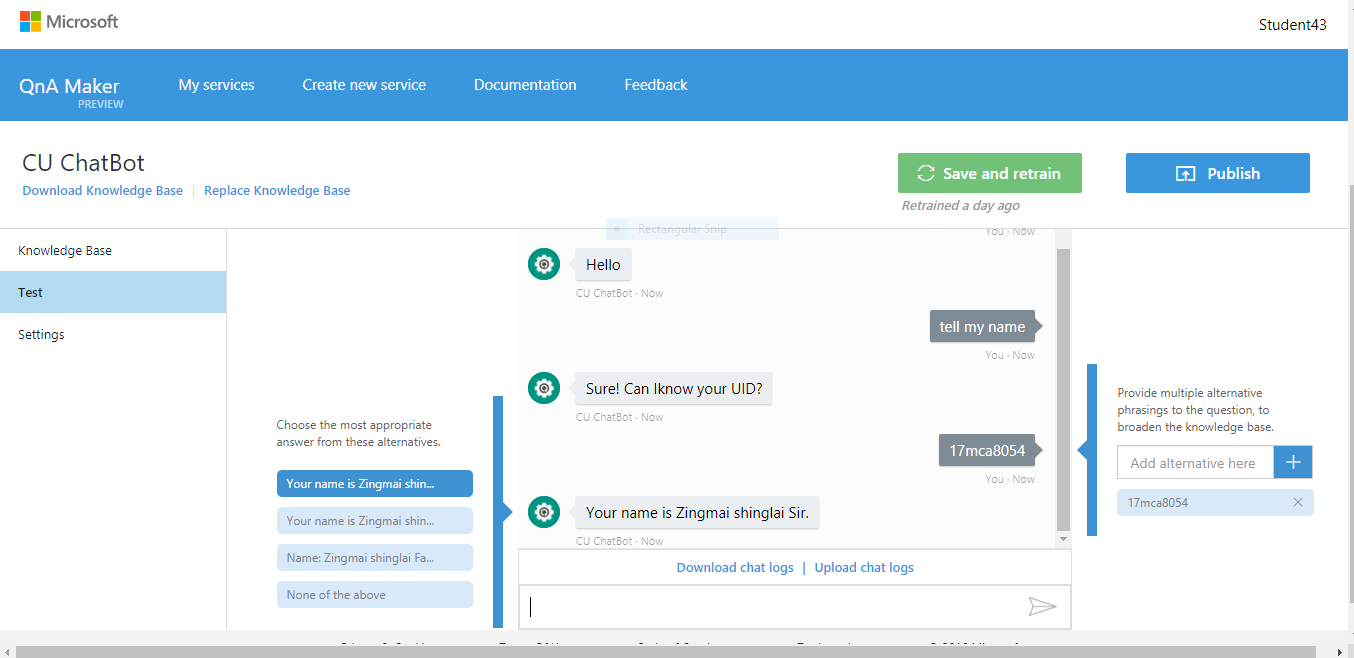
****

**Fig 1.5.3** Bot Iteraction for Application

* **Student Chatbot**

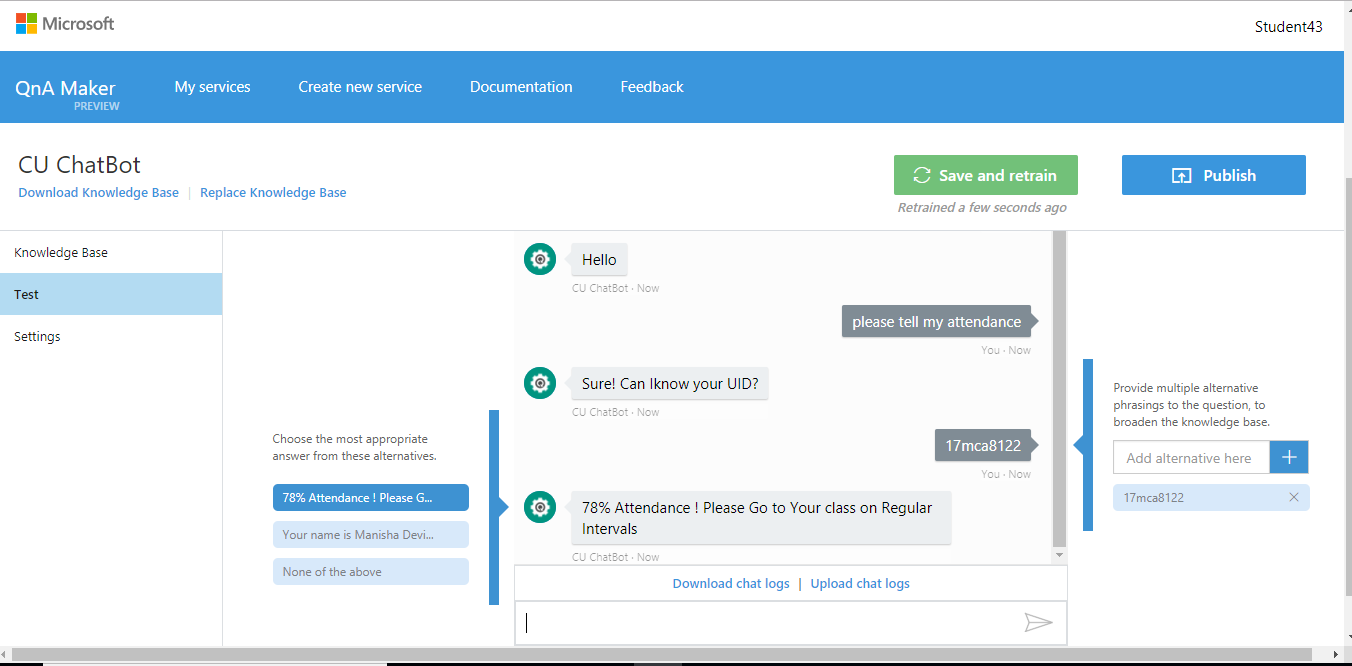
****

**Fig 1.5.4**

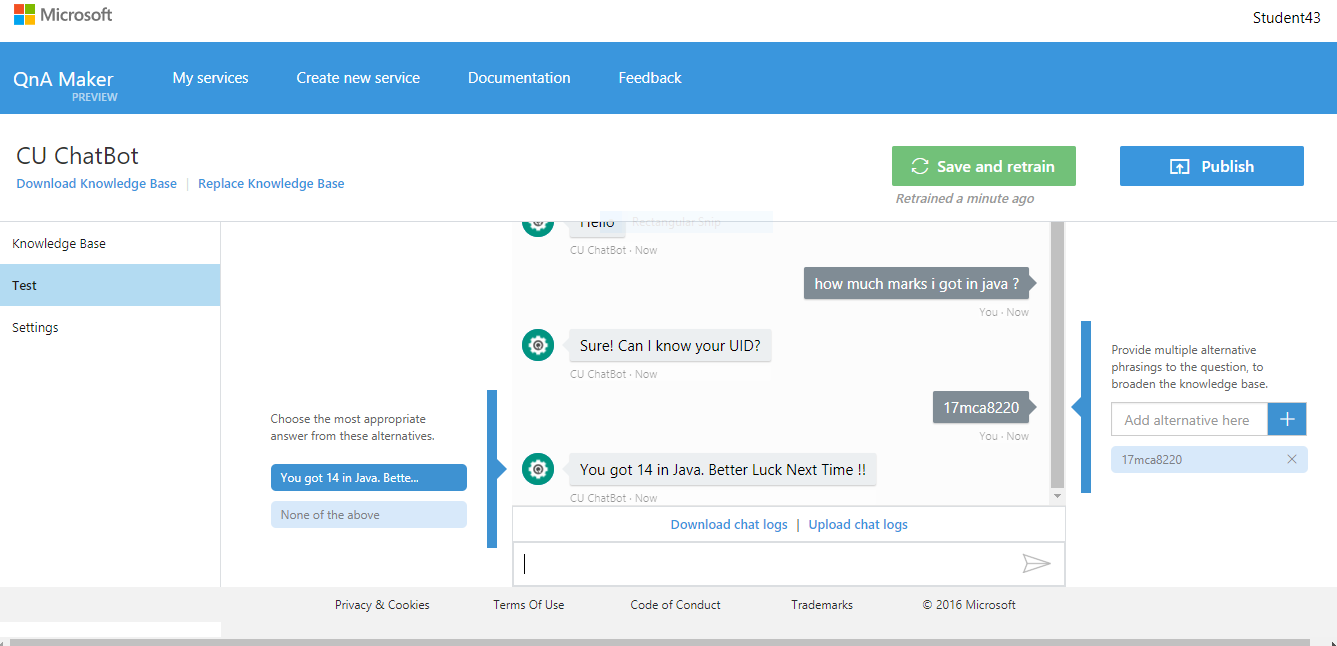


**Fig 1.5**

* **Student Attendance Details :**

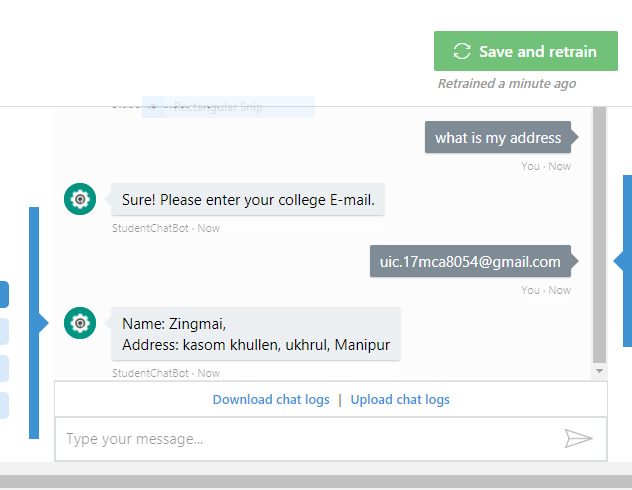
****

* **Chatbot Marks Update :**

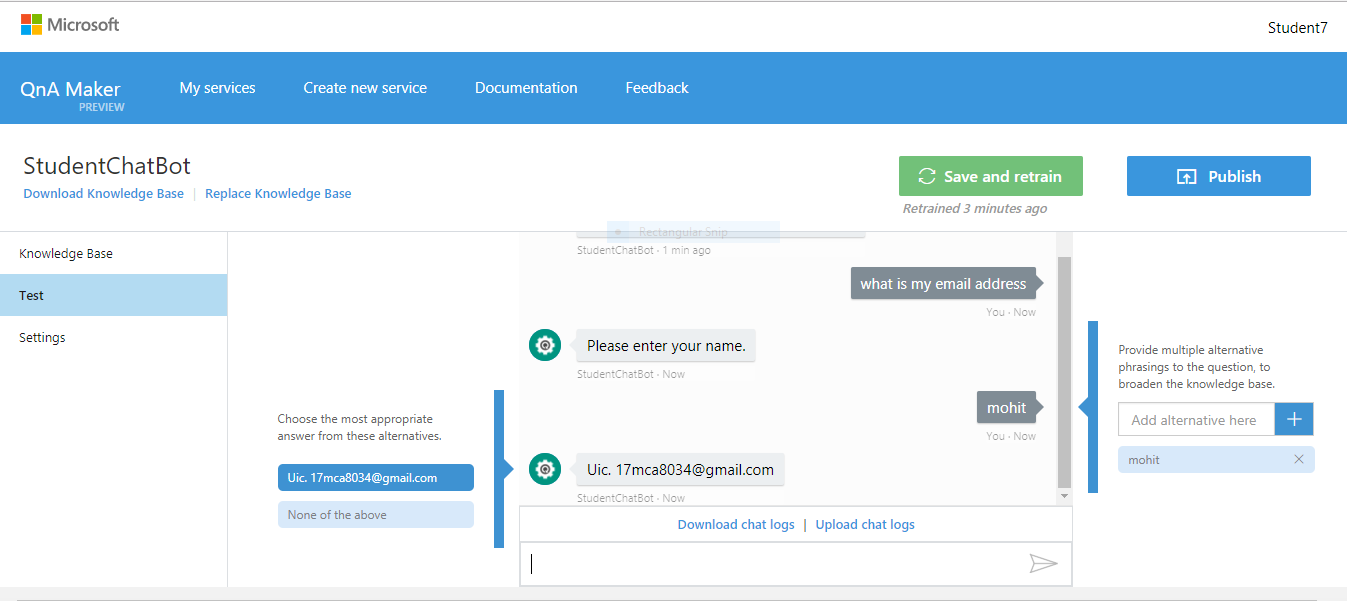
****

**Fig 1.6**

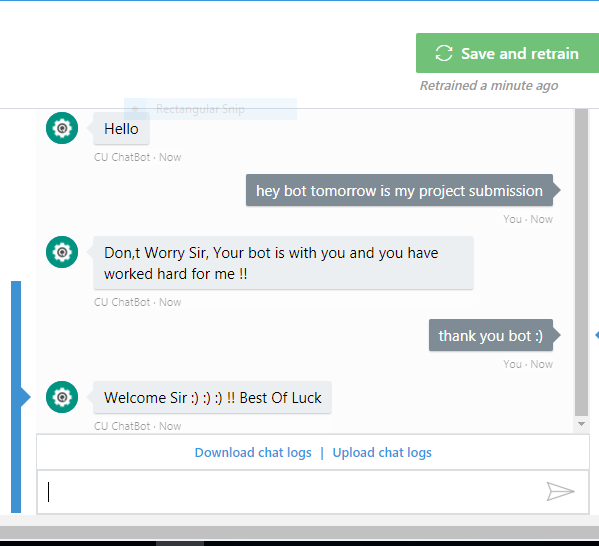
**Student Address Detials :**

****

**Student Name :**

****

**Chatbot Wishes :**

****

**Fig. 1.7 Intelligent Chatbot**

**REFERENCES**

[1]. Emanuela Haller and Traian Rebedea, “Designing a Chat-bot that Simulates an Historical Figure”, IEEE Conference Publications, July 2013.

[2]. Maja Pantic, Reinier Zwitserloot, and Robbert Jan Grootjans, “Teaching Introductory Artificial Intelligence Using Asimple Agent Framework”, IEEE Transactions On Education, Vol. 48, No. 3, August 2005.