**BOT COUNSELLER**

SDS (SOFTWARE DESIGN SPECIFICATION)

Prepared by

MUHAMMAD HASSAN (1612195)

MUHAMMAD DANIYAL(1612238)

**1. Introduction**

**1.1 Purpose of this document**

This software design document describes the architecture and system design of Bot Counselor, a question answering system for education industry. It is intended to outline the system structure for the project manager and stakeholder, and provide technical guidance to development team.

**1.2 Scope of the development project**

This will be similar to what was written in the SRS. Bot Counselor is an AI chatbot that receives questions from the user, tries to understand the question and provide appropriate answer. It does by converting and English sentence into machine friendly query, looking up in trained model for necessary information to answer the question and finally returning the answer in natural language.

The main objective is to develop a bot that provides such service that will help the user in selection of the undergraduate field.

The goal is to provide the students of intermediate and A-level to provide counselling about, which field they should enroll in.

**1.3 Definitions, acronyms, and abbreviations**

* **Chatbot:** An interface, usually text based, specializing in mimicry of natural language conversation.
* **HTML:** Hyper Text Markup Language, a standardized system for tagging text file to achieve font, color, graphic and hyperlink effects on webpage.
* **JSON:** JavaScript Object Notation, a data interchange format that is commonly used in exchanging the data over the internet.
* **Flask:** it is a python framework for the development of web application.
* **Neural network:** it is an Artificial Intelligence concept for the training of machine
* **MVC:** Model View Controller. It is now considered as standard concept/framework for the development of web application
* **Pymongo:** it is a library of python for connection of data base and transection of data from database

**1.4 References**

* **For content related:** <http://www.cci.drexel.edu/SeniorDesign/2016_2017/DrexelChatbot/DrexelChatbotDD.pdf>

**1.5 Overview of document**

* **Introduction:** provides an overview of application, explains objective and goal of project and describe the structure of document
* **System over view:** gives general description of functionality of context and design of bot counselor.
* **System architecture:** break the project down into various subsystems, defines how those subsystems interact.
* **Data design:** describes the organization of data in mongodb database implemented for bot counselor.

**2. System architecture description**

* 1. **Section Overview**

Most of the search engines today, like google, use a system to rank web pages. When user enter a query, the query is interpreted as keyword and system returns list of highest ranked web pages which may have answer to query. Them user must go through web pages to find the answer they are looking for. Bot counselor, however, will try to understand the query and provide a definitive answer.

* 1. **General Constraints**

as the application is developed in python, the only limitation of end result will be the speed of the bot. to overcome this issue, it is advised to the user to have a high-speed internet connection and a good processor, this will be beneficial for the user.

* 1. **Data Design**
     1. **Database design**

|  |  |
| --- | --- |
| USER NAME | Varchar |
| EMAIL | Varchar (unique) |
| IMAGE-FILE | Varchar |
| PASSWORD | Int(unique) |
| ID | Int (unique) |

|  |  |
| --- | --- |
| ID | Int |
| MESSAGE | Varchar |
| SENDER | Varchar |
| DATE-TIME | TIME |
| USER ID | INT |

* + 1. **Dataset questions**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Questions** | **Option A** | **Option B** | **Option C** | **Option D** | **Option E** |
| **About Youth** | Help others | Explore world | Make money | Build stuff |  |
| **Well done job?** | Assets | Neat environment | Creating something | Solving problem |  |
| **Ultimate goal** | Understand humans | Invent something | Be president | Make better world |  |
| **Interested in** | Technology | Learning computer | Intellectual ideas | Knowing nature |  |
| **You like to do** | Singing | Vlogging | Computing | Surfing internet | Thinking ideas |
| **Favorite class** | Performing | Human behavior | Deep thinking | experimenting |  |
| **Like to help others?** | Yes | No |  |  |  |
| **Know how machine working** | Yes | No |  |  |  |
| **You are?** | Ambitious | Go with flow |  |  |  |
| **Socializing?** | Yes | Sitting at home |  |  |  |
| **Concerned about state of environment** | Yes | No |  |  |  |
| **Interested in body heal** | Yes | No |  |  |  |
| **Like working with hand and outdoor** | Yes | No |  |  |  |
| **Graphic\web design** | Yes | No |  |  |  |
| **Good at numbers** | Yes | No |  |  |  |
| **Work with people** | Yes | No |  |  |  |
| **Friends know you for** | Organizer | Go with gut feelings | Showing initiative | Job done |  |
| **People consider you** | Risk taker | Logical | Generous | Cool headed |  |
| **After 20 years** | Entrepreneur | Doctor | Actor | Engineer | Computer scientist |
| **Free time activities** |  |  |  |  |  |

.

* 1. **Program Structure**



Flask is being used to develop the frontend and web application. Flask is the framework of python that is used to develop the web app. The nature of flask is minimalistic. This nature gives us a lot of freedom in how to structure our project. By getting advantage from this flexibility of the framework, the project will be based on Model View Controller concepts.

* 1. **Alternatives Considered**N/A.

1. **Detailed description of components**  
     
   1. **Section Overview**  
      The end product (chatbot) will be the mixture of many components. Each component will be playing a huge roll in the development of the bot. every feature of the bot will go through all the component of the bot. The components are Flask, Neural Network, MongoDB while features are chatting, sign in, sign up.
   2. **Component n Detail (include a sub-section for each component)**
      1. **Flask:**

|  |  |
| --- | --- |
| Identification | Flask |
| Type | Initial File |
| Purpose | To start application and medium for user interface and server requests |
| Function | ‘app’ is the base attribute for the flask. Other functions are based upon this attribute. Every web page of the application has a function associated. |
| Subordinates | ‘Model View and Controller’ concept is implemented. |
| Dependencies | None |
| Interfaces | In views, all the pages are available. All of them are developed on bootstrap studio. So, all the interfaces are integrated in view page. |
| Resources | Flask itself is a resource for the application. So, it doesn’t require any resource to operate for further operation. A browser is needed to showcase the results of flask. |
| Processing | Flask, a framework works the way you want it to work. the is no specific structure for flask to work on. But bot counsellor will be developed using MVC frame work  Flask is initiated by creating app I,e the object of flask  app = Flask(\_\_name\_\_)  and route ‘/’ oi followed  @app.route(‘/’)  def <functionName>():  return <some web page> |
| Data | Flask does not have concern with data. It doesn’t require any data to process. It just works to send request to the server and send response to the user. Data from the form is travelled by request of flask to the server. |

* + 1. **Login:**

|  |  |
| --- | --- |
| Identification | LOGIN |
| Type | Class |
| Purpose | Process by which an individual gains access to his / her profile by identifying and authenticating themselves. |
| Function | Login are used to gain access to and control of Users to get excess of his/her Chatroom. It requires Username and Password. This function will log in a user based being matched in MONGO-DB |
| Subordinates | The screen contains link to Registration Screen |
| Dependencies | It should be connected with the server, which is connected with the database for validation. |
| Interfaces | Designed using HTML, CSS, BOOTSTRAP and JAVASCRIPT using simple colors |
| Resources | System must be connected to MONGO-DB in order to fetch login details |
| Processing | Data which will be provided by the user is to validate it’s information |
| Data | Data is provided in login in the form of INPUT , |

* + 1. **Register**

|  |  |
| --- | --- |
| Identification | SIGNUP |
| Type | Class |
| Purpose | It will allow unregistered candidates to become a member of the system. |
| Function | It will register the new User. It requires the form which has to be filled by the user. |
| Subordinates | The screen contains link to the login page. |
| Dependencies | It should be connected with the server, which is connected with database for update. |
| Interfaces | Designed using HTML, CSS BOOTSTRAP |
| Resources | System must be connected to MONGO-DB in order to register user |
| Processing | Data which will be provided by the user will be validated. |
| Data | Data will be provided in the form of INPUT |

* + 1. **Sqlalchemy:**

|  |  |
| --- | --- |
| Identification | Sql alchemy |
| Type | DATABASE connection |
| Purpose | It connects the database |
| Function | It performs crud operation using from database |
| Subordinates | NONE |
| Dependencies | NONE |
| Interfaces | NONE |
| Resources | Sql alchemy should be installed |
| Processing | It makes connection using code and performs crud operation using code |
| Data | Data is saved in schema format |

1. **User Interface Design**
   1. **Section Overview**

Just like every chat container, bot counselor will be a simple design. And a couple of common ingredients will be there and that are login and registration form. This will help us to maintain the chat that is done with the bot. Further details are followed by headings below.

* 1. **Interface Design Rules**

N/A

* 1. **GUI Components**  
     GUI components are the web pages that will be shown to the user. Following are the ones that will be shown

**USECASE**

**1)**

|  |  |
| --- | --- |
| **ID:** | **1** |
| **Title:** | LOGIN |
| **Description:** | THROUGH THIS FEATURE USER WILL BE ABLE TO ACCESS HIS/HER PROFILE |
| **Primary Actor:** | THE USER HIMSELF/HERSELF |
| **Preconditions:** | THERE WILL BE A SIMPLE WEB PAGE THAT WILL ASK THE USER TO SIGN IN TO ACCESS THE PROFILE |
| **Postconditions:** | AFTER LOGGING IN, USER WILL BE ABLE TO ACCESS THE PROFLE AND CONTINUE CHATTING WITH THE BOT |
| **Main  Success Scenario:** | AT THE BIGENNIHG, USER WILL BE SHOWN A WEB PAGE IN WHERE S(HE) WILL BE ASKED TO ENTER EMAIL AND PASSWORD. ON ENTERING THE ASKED DETAILS, S(HE) WILL BE REDIRECTED TO THE CHATROOM |
| **Extensions:** | THERE IS A POSSIBILITY THAT USER WILL ENTER WRONG PASSORD OR MAY NOT HAVE ACCOUNT AND JUST FILLING THE FIELD. THIS WILL GIVE EXCPETION THAT LIKE ‘USER NOT FOUND’ OR ‘WRONG EMAIL OR PASSWORD’ |
| **Frequency of Use:** | EVERY TIME WHEN USER VISITS WEB PAGE |
| **Status:** | DESIGN IS COMPLETED AND IMPLEMENTATION IS LEFT |
| **Owner:** | BOTH THE TEAM MEMBERS |
| **Priority:** | HIGH. CAUSE IT WILL ACCESS THE PROFILE |

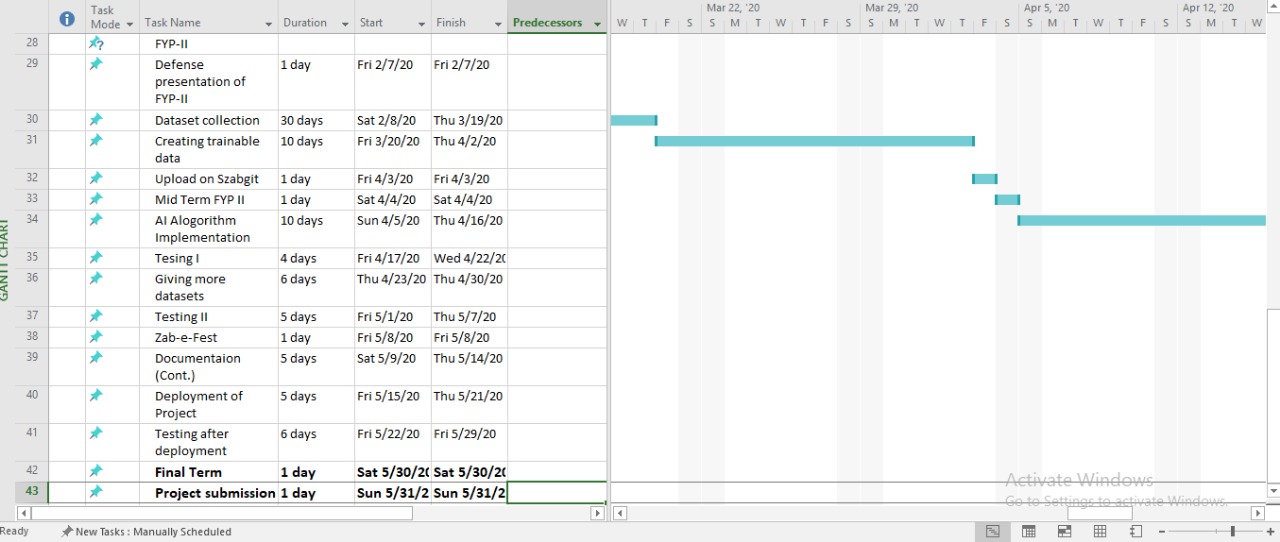
**2)**

|  |  |
| --- | --- |
| **ID:** | **2** |
| **Title:** | REGISTER |
| **Description:** | USER WILL CREATE HIS PROFILE TO GET ACCESS TO THE CHATROOM |
| **Primary Actor:** | USER HIMSELF/HERSELF |
| **Preconditions:** | THERE WILL BE A SIGNUP OPTION AVAILABLE FOR NEW USER |
| **Postconditions:** | HE WILL BE ABLE TO ACCESS THE CHATROOM ON HIS OWN PROFIILE |
| **Main  Success Scenario:** | TO MAKE USER A PERTICIPENT OF THE PROTAL SO THAT HE WILL BE ABLE TO AVAIL THE BOT COUNSELLING |
| **Extensions:** | WHEN USER GIVE WRONG DETAILE OR NOT ACCORDING TO THE GIVEN INSTRUCTIONS |
| **Frequency of Use:** | FIRST TIME ONLY WHEN USER IS NEW |
| **Status:** | DESIGN IS COMPLETED AND IMPLIMENTATION IS LEFT |
| **Owner:** | BOTH OF TEAM MEMBERS |
| **Priority:** | HIGH, AS CREATING PROFILE IS IMPORTANT |

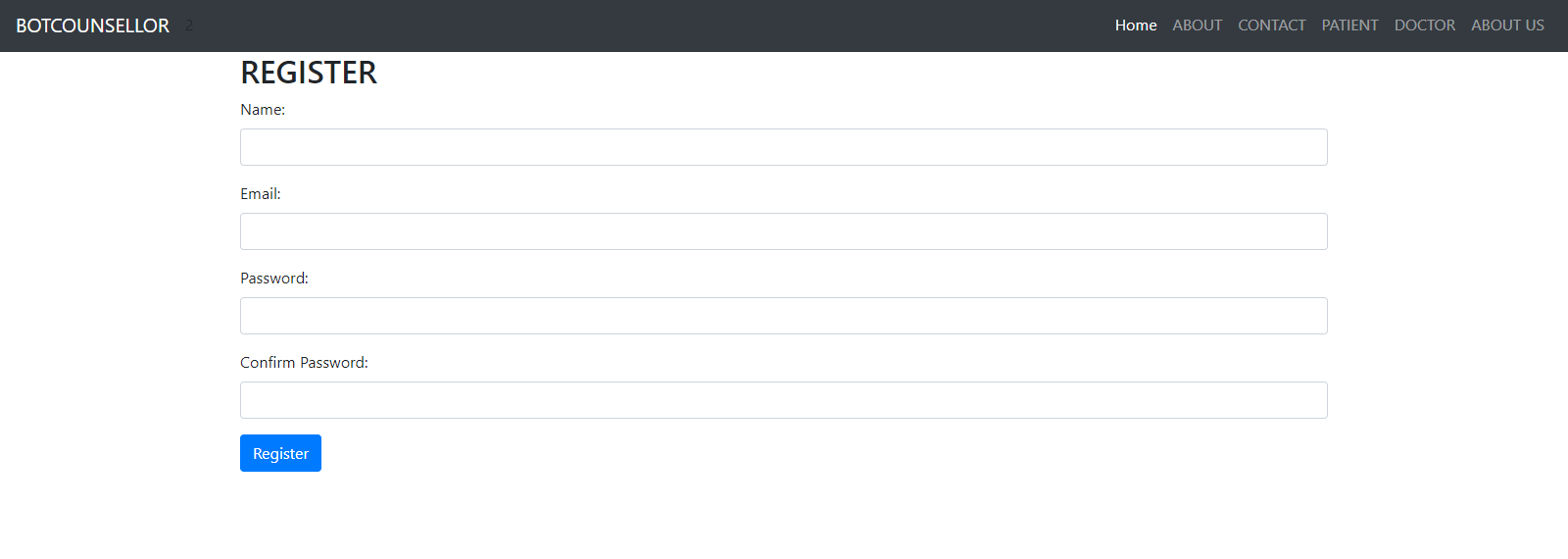
**3)**

|  |  |
| --- | --- |
| **ID:** | **3** |
| **Title:** | CHAT |
| **Description:** | USER WILL CHAT WITH THE COUNSELLOR |
| **Primary Actor:** | USER HIMSELF/HERSELF |
| **Preconditions:** | - |
| **Postconditions:** | HE WILL BE ABLE TO GET THE ANSWER |
| **Main  Success Scenario:** | THE USER WILL BE ASKED SOME QUESTION ABOUT BY COUNSELLOR |
| **Extensions:** | \_ |
| **Frequency of Use:** | CHAT |
| **Status:** | IMPLEMENTATION AND DESIGN IS COMPLETED |
| **Owner:** | USER / COUNSELLOR |
| **Priority:** | - |

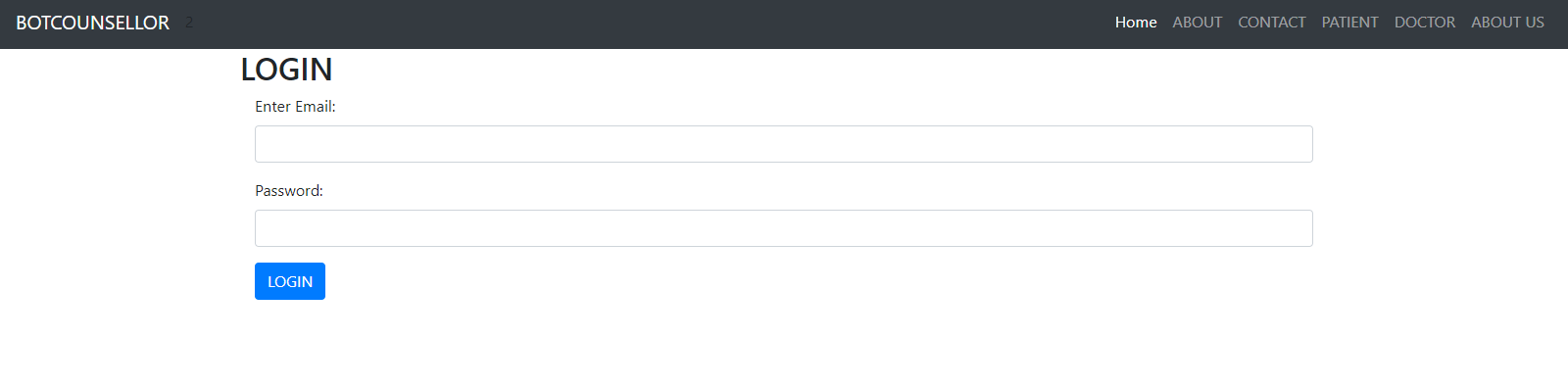
**GANNT CHART**



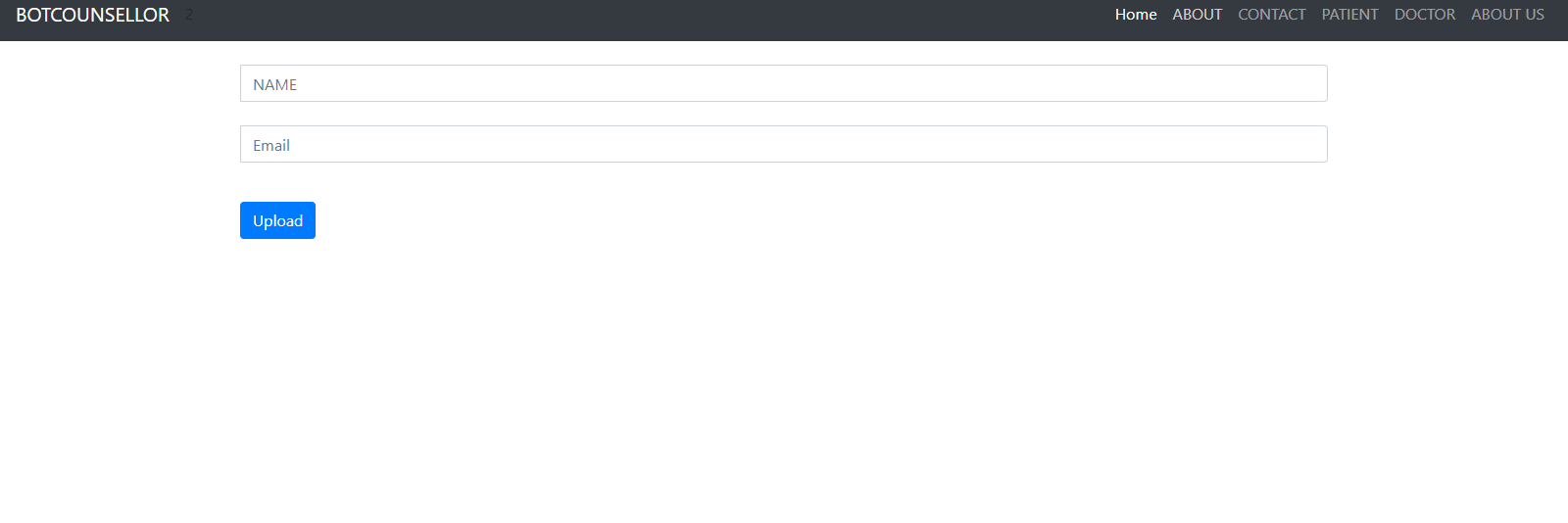
**REGISTER**



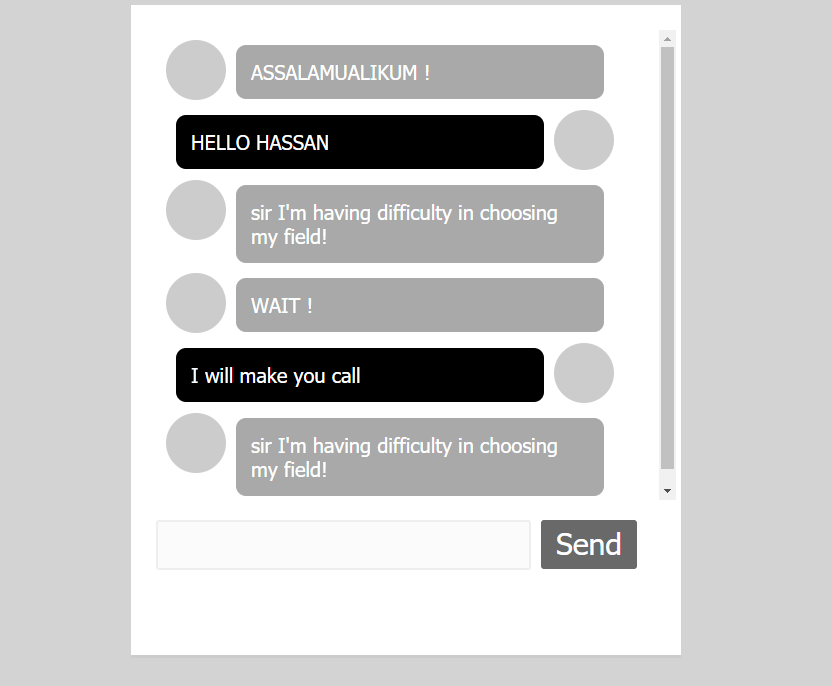
**LOGIN**

****

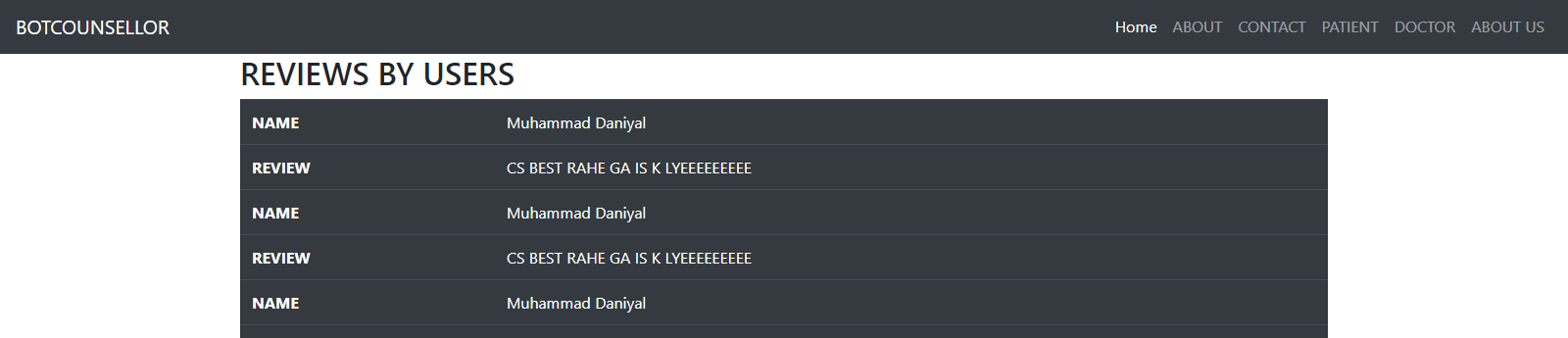
**ACCOUNT**

****

**CHAT**

****

**ACCOUNT**

****

* 1. **Detailed Description**

|  |  |
| --- | --- |
| **Public login()** |  |
| **Input** | Email, Password |
| **Output** | Page Render, error |
| **Description** | This function will take username and password as input in form and validate them. And output will result accordingly. Will be rendered to chatroom on right input or get error on bed input. |

|  |  |
| --- | --- |
| **Public register()** |  |
| **Input** | Username, Email, Password, Confirm Password |
| **Output** | Page Render, Error |
| **Description** | This function will take the mentioned fields from the from and validate whether the given information is in correct format and then generate the result accordingly. |

|  |  |
| --- | --- |
| **Public send()** |  |
| **Input** | Void |
| **Output** | Void |
| **Description** | This function will send the user entered message from designated text area and send that message to the server. Then server will respond according to the entered message. |

1. **Reuse and relationships to other products**

Just like every other software, the reusability of classes and functions will always be looked after, in order to use them in future for further modification. For the bot counselor, the class that involves training will be constant in every modification, unless any new method is found. Webpages are made using bootstrap and bootstrap’s source file will remain as it is for further development and modification. And code can be integrated in html format files. Model classes are also a part of reusable classes. And those can be modified with the advancement in features and user data requirement. In a nut shell, each and every component can be used in future for further modification and advancement in the end project. The only limitation is that the project is centralized so updating will stop the services of bot.

1. **Design decisions and tradeoffs**

Use this section to motivate any decisions that will help the reader understand the design that your team is using. This section can also capture good ideas that were abandoned and the reasons for leaving them out of the design.

1. **Pseudocode for components**
   1. **Flask:**

App = flask(\_\_name\_\_)

@App.route(‘/’)

Def fucntionName():

Function definition

Return webpage

* 1. **Login:**

@app.route(‘/login’)

Def login():

User = input from form

Validate() // from database

Return webpage

* 1. **Register:**

@app.route(‘/register’):

Def register():

User = input from form

Db.insert(user) //into database

Return webpage

* 1. **Chat:**

Message = input from chatroom

Db.insert(message, sendby, time)

Respone = aiClass(message)

Return (web page, response)

* 1. **Sql**

Db = sqlAlchemy//create object

Db.sessionadd() //to insert into database

Db.query() //to read from database

1. **Appendices**

Other used algorithms and machine learning algorithms are:

* 1. **Neural network:**

there are two ways of creating chatbot. First, we write all the logic that would be able to take any question and give any appropriate response. This would be adding lots of rules manually and it will tale lots of time and effort. It will also not be reusable for any other domain. The second method is machine learning. Using neural network will allow us train the neural net with a set of questions and answers and if a new question is passed to neural net, the neural net will try to provide an appropriate answer.

* 1. **Python:**

We selected python as our main programing language because it has the highest number of libraries of artificial intelligence field, as well as each member on development team are reasonably familiar with python.

* 1. **Modelize:**

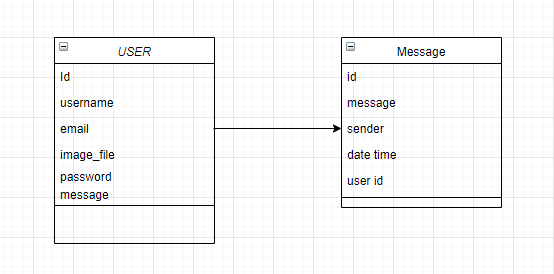
Since we are not sure how well some algorithms will perform, modulizing subsystems will allow us to retire poorly-functioning algorithms and introduce new ones without breaking entire system.

The following list presents the diagrams that should be included at appropriate places

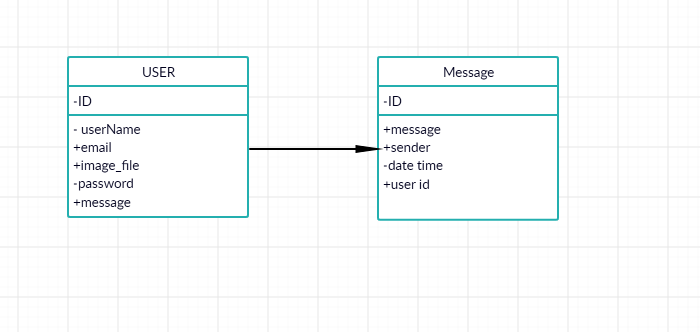
|  |  |
| --- | --- |
| Class Diagram | Describes the structure of a system |
| Object Diagram | Expresses possible object combinations of a specific Class Diagram |
| Statechart Diagram | Expresses possible states of a class (or a system) |
| Activity Diagram | Describes activities and actions taking place in a system |
| Sequence Diagram | Shows one or several sequences of messages sent among a set of objects |
| Collaboration Diagram | Describes a complete collaboration among a set of objects |
| Use-case Diagrams | Illustrates the relationships between use cases |
| Component Diagram | A special case of a Class Diagram used to describe components within a software system |
| Deployment Diagram | A special case of a Class Diagram used to describe hardware within the overall system architecture |
| System Block diagram | A diagram showing the major components of the system with its interconnections and external interfaces |

**DIAGRAMS**

* + 1. **Class diagram:**

****

* + 1. **Object diagram:**



* + 1. **Sequecne diagram:**
       1. **Chat:**



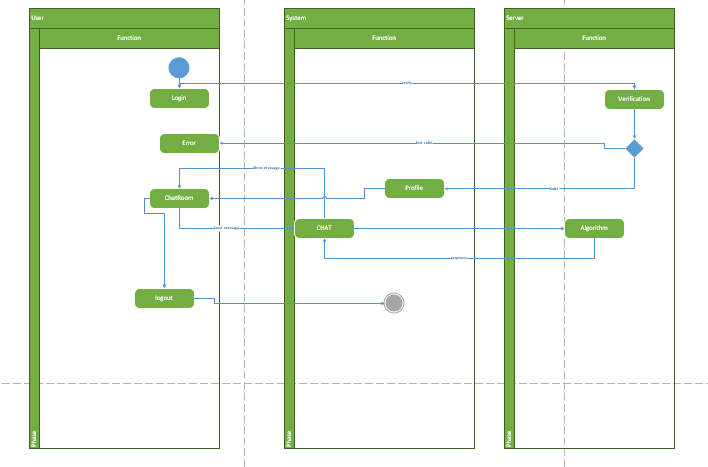
* + - 1. **Login:**



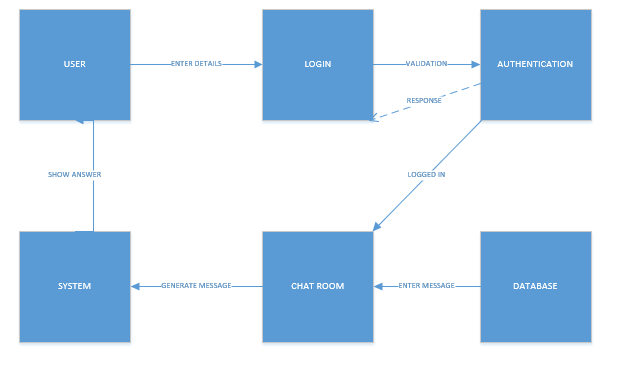
* + - 1. **Register:**



* + 1. **Activity Diagram:**



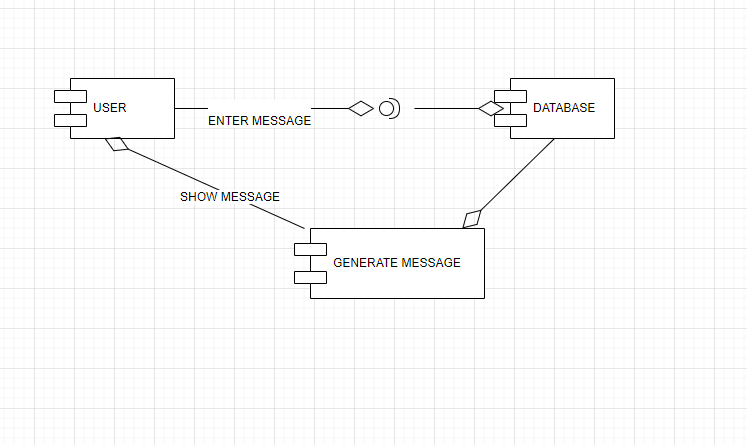
* + 1. **Collaboration diagram:**



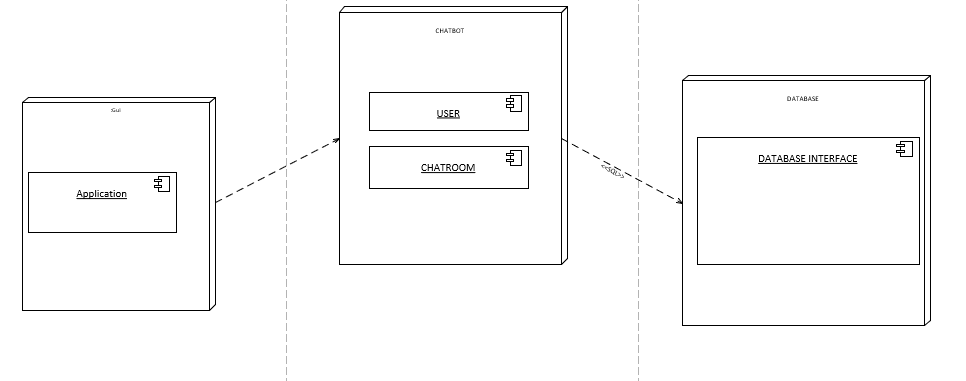
* + 1. **Use Case diagram:**



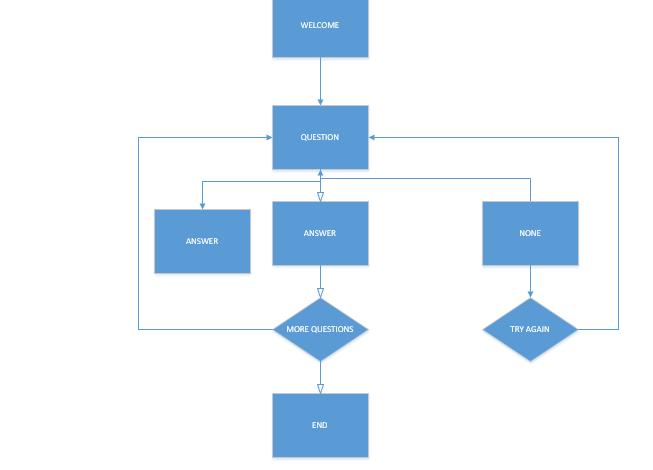
* + 1. **Component diagram:**



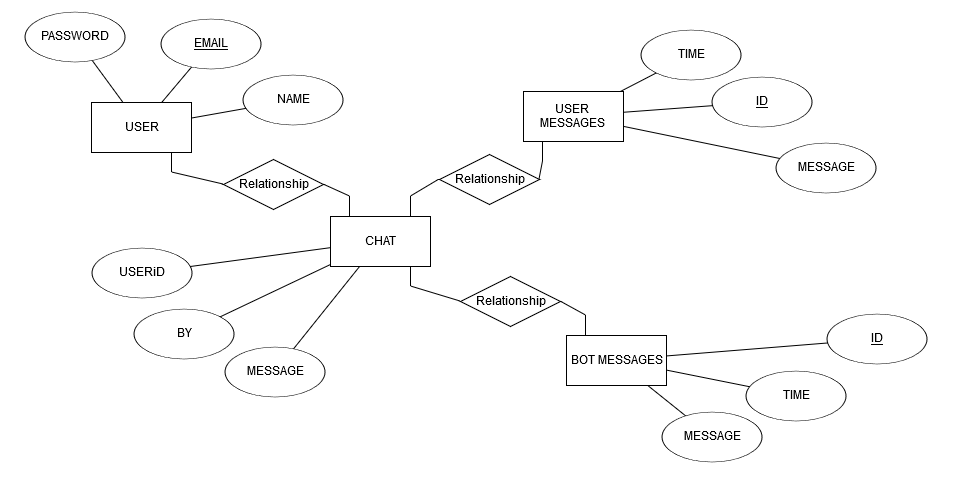
* + 1. **Deployment diagram:**



* + 1. **State Block diagram:**



* 1. **ERD:**



**SEQUENCE DIAGRAM**

