Software Design Document

JabberJack

Team Member:

Sara Harris, Tyler Zimmerman, Jiasheng Yang, Gabriel Proctor

Team Mentor: Felicity H. Escarzaga

Client: Dr. Andy Wang

Northern Arizona University

Version 2.0

---

07 February, 2022

Contents

1.0 Introduction

JabberJack

Chatbots and other helpful bots are becoming increasingly more common throughout both the world of business and within academia [1]. Websites use chatbots to answer simple questions that visitors may have about products or services. There are even chatbots that are present on some Northern Arizona University (NAU) websites such as the one present on the Campus Health Services page [2]. Chatbots are not as complex as many people believe them to be; chatbots are not able to hold a conversation or answer extremely complex questions[1]. Even though chatbots cannot interact with humans in a human-like manner, there are a great deal of research teams working to improve the interactivity between chatbots and humans[1]. Chatbots can either be simple such as rule based chatbots which operate based on a number of rules, or they can be complex with artificial intelligence (AI) such as chatbots similar to Amazon's Alexa and Apple's Siri which learn from interactions to improve future interactions. The world runs on data, and the more data that a chatbot has available to it, the more helpful they become [4].

Chatbots are popular for customer consumption too, meaning that a great deal of chatbots can be found within regular people's homes. Amazon's Alexa is by far the most popular and the company holds about 62% of the market share within this industry [5]. In addition to Alexa, Google and Apple both have a comparable product. All of these chatbots are built utilizing complex computing called artificial intelligence. While artificial intelligence (AI) is the most popular type of chatbot, the more common chatbots are the ones that people can find located on websites. These chatbots are usually basic and only hold a small amount of information with little to no learning ability. They can answer basic questions about the website or about the company's service or product, so the company does not need to employ a real person for that role.

Our client Dr. Andy Wang is the dean of NAU's college of engineering, informatics, and applied sciences (CEIAS). Dr. Wang maintains and runs the programs contained within CEIAS and facilitates growth for both the college and programs in CEIAS. Dr. Wang is in the business of computing, engineering, and most importantly helping people. Dr. Wang deals with lots of people on a daily basis and is interested in helping them find answers to their questions efficiently.

Dr. Andy Wang manages the entire college of engineering, informatics, and applied sciences; he wants the building to operate as efficiently as possible. The engineering building at Northern Arizona University has faculty that do all sorts of tasks from teaching classes to assisting students in the creation of their schedules. There are also frequent tours that come through the building that are given to prospective students of CEIAS programs.

The operations of the building have to function every day in order to keep the college running smoothly.

Visitors and students of the college alike have questions; sometimes these questions cannot even be answered by an internet search. Unfortunately, users still face lots of problematic business functions:

- Information is scattered and can be contradicting at times; A basic search, unless extremely specific will not usually give a straight answer.
- Questioning faculty takes time away from the job that they were hired for and can cause them to trail behind in their work.
- The existing chabot system at NAU cannot improve users' satisfaction because of the ugly User Interface and no support of text-to-speech or speech-to-text.

To solve this problem Team JabberJack is creating an "intelligent" chatbot so that visitors, students, and even faculty can have quick access to information about Northern Arizona University. Here are core features that the ChatterJack will equip:

- Reasonable, quick, and intelligent responses to users' questions about NAU via text and audio.
- A friendly User Interface, a smile face, will make users feel like they are talking with a robot, not just using a software application.
- A centralized updatable database containing at minimum 500 question/answer (QA) pairs.
- Upload the database of QA securely with User Authentication.

Overall, ChatterJack will be a smart chatbot to meet all the client's business requirements so that it is capable of reducing faculty members' burden, attracting more students, and making the whole engineering building run efficiently. This project will set the foundation for future projects involving the chatbot and can one day even be integrated into a real free standing robot.

2.0 Implementation Overview

JabberJack

The ChatterJack will be a secure Web2.0 application which will be deployed on the NAU website to meet all core features mentioned in the last section. This section will discuss the core techniques that JabberJack has chosen for generating an intelligent chatbot.

According to the different purposes, the ChatterJack will be separated into two different parts: local software and web application. The local software will be capable of connecting to an online database to download new information and to update existing information located within a localized database. The online portion of the project will be a web application where designated users will be able to interact with the online database to add or edit the information that the chatbot will have access to.

The chatbot itself will be written in python which will be used to create both the user interface and functionality of the chatterjack software. The software will use the spaCy library to allow the chatbot to be able to correctly analyze and in a sense understand questions being asked by users. The spaCy library utilizes natural language processing which allows the chatbot to read and understand the questions that are being asked by users.

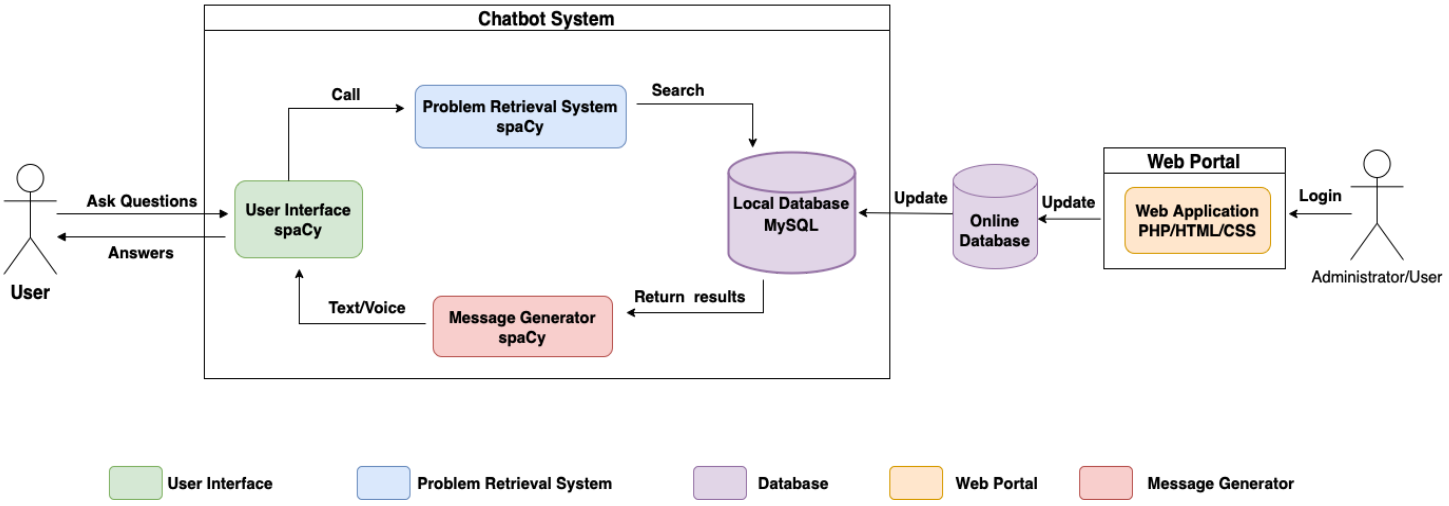
The database will be created using MySQL both within the localized software as well as the connected online database. MySQL is a database management system that uses tables to relate data to each other. In this database there will be question and answer pairs that the chatbot will have access to. These entries will be able to be updated through the online portion of the project, the web portal.

The online portion of the project, the web portal, will be written in PHP which will handle the security as well as serving all of the HTML/CSS pages that the web application will utilize. User information will be contained within the database but will not be accessible to normal users and will be protected using PHP and MySQL together. PHP has a built in hashing function that will be utilized for authenticating user passwords.

3.0 Architectural Overview

JabberJack

The chatterjack can be broken down into four key components: the question and answer database, the Problem Retrieval System and within that the Message Generator, the web portal, and the User Interface.



The question and answer database is arguably the most important part of the project as everything that the chatbot knows will be contained within this database. The database will be managed by MySQL which stores data in relational tables. The online database will have the ability to be modified by users such as NAU staff and Faculty. The local database will have the ability to connect to the online database to grab new information and to update it's contents. The local database will only be able to grab new information from the online database; it won't send anything to the server.

The Problem Retrieval System and Message Generator are how the chatbot will connect with the database; both of them are designed and developed based on the Python library named spaCy to meet Natural Language Processing (NLP). When a user is finished asking their question the problem retrieval system will segment the text to word to understand the users' main purpose and match up an answer within the database. The main responsibility of the problem retrieval system is traversing the database in order to find an answer to the message generator. It will need to be able to "read" and "understand" the user input and then be able to go through the database to find the most relevant information to

respond with. If nothing is found it will need to respond with a blanket “I don’t know” statement. The problem retrieval system will have access to the local database and will be able to access everything within it. It will also have the ability to interact directly with the user interface to grab the user’s input. Once the problem retrieval system finds the correct answer it’ll utilize a message generator to create a coherent sentence to deliver back to the user.

The web portal will primarily be a place to host an online database but with an added feature of user/database interaction. Users such as NAU faculty and staff will be able to access the web portal through a user authentication system and from there they are able to add/edit/delete information within the online database. These changes will then be reflected within the local chatbot’s database whenever the chatbot connects to the internet. Information can be entered into the database through the web portal. Then once the local database of the chatbot connects to the internet it’ll update by accessing the online database and making changes.

The user interface is important as it is the part where the users will be able to interact with the chatbot as well as the web portal. The user interface will be simple and clean so that users can easily interact with the software both on the chatbot side as well as the web portal side. The user interface for the web portal will be a mask over the inner workings of the website. Therefore, it does need to communicate with other modules. The user interface of the chatbot will be more than just a face; it will have to be able to take and scrub input to pass along to the problem retrieval system. The chatbot interface will utilize natural language processing to send user input to the problem retrieval system. The message generator will return the text or voice response to the user interface.

4.0 Module and Interface Description

JabberJack

The following section will describe the modules in detail. This design will help the team develop the software's architecture as well as allow the team to visualize the structure of the system.

4.1 Problem Retrieval System

The Problem Retrieval System is used for searching the saved database of the Question and Answer pairs to find the same or similar answer. When the user asks a question, the chatbot is capable of providing an almost instantaneous and accurate answer to the Message Generator. This is more like the Google Search Engine, matching the related problem from the Google database.

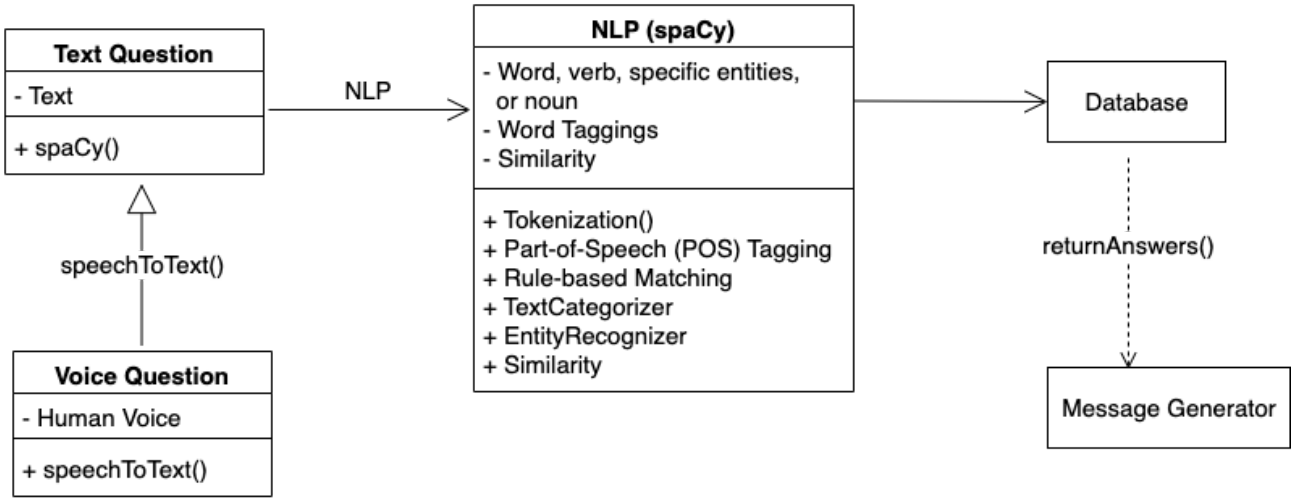


Figure 4.1.1 Problem Retrieval System

JabberJack will develop the Problem Retrieval System based on the Python library named spaCy to achieve grabbing the answer from the saved database. The general idea for the system is segmenting text into words and matching the key information from the database. There are six main features of spaCy used to build the system: Tokenization, Part-of-speech (POS) Tagging, Rule-based Matching, TextCategorizer, EntityRecognizer, and Similarity. Tokenization is used for segmenting text into words, punctuations marks; Part-of-speech (POS) Tagging and TextCategorizer are used for labeling the words, verb or noun, special entities like professors' name, so that the chatbot can utilize the key information to search and answer the questions; it will also utilize Rule-based Matching to match the information from the database. In **figure 4.1.1**: Problem Retrieval System will separate the text into words and mark the different information. Then it will use the

matching functions to search for the corresponding information. Last, it will return the key answer to the message generator. If the user inputs the question by speaking, the ChatterJack will utilize the Speech-To-Text engine to convert voice to text at first, then execute the same operation.

4.2 Message Generator

The message generator is used for combining key information that is retrieved from the database with other words so that the response is not only accurate, but also polite and intelligent. In other words, it is a form of interaction vaguely indistinguishable from human to human interactions. For example, when the user asks: “Hello, where is Dr. Doerry’s office?”, the chatbot will answer: “Hello! Dr. Doerry’s office is at Room 217 in the SICCS Building. Thanks.” There are three key components in this response: the location as the key information, the greetings, and linking words as needed to form a full response. These responses will give the ChatterJack an edge in simulating human conversations.

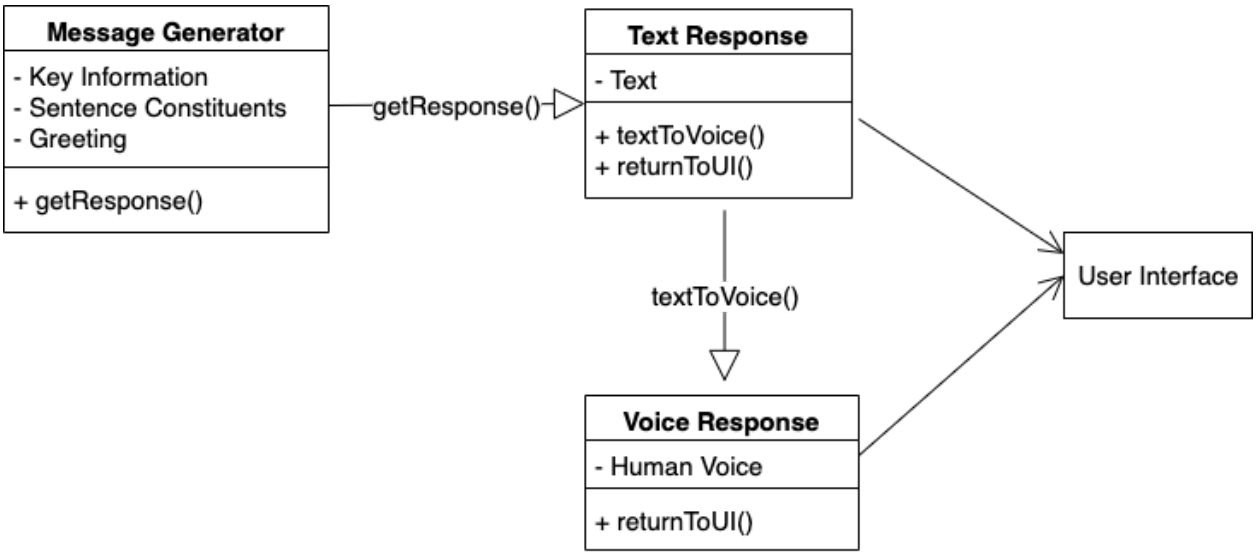


Figure 4.2.1 Message Generator

The overarching components of the Message Generator are two parts: Text Response and Voice Response, where the Voice Response is based on the generated text. JabberJack will utilize the Python library called spaCy to achieve reasonable responses. SpaCy is a python library that allows programs to break down a natural language. Then if the user wants a voice response, the ChatterJack will utilize Text-To-Speech (TTS) engine to convert text into speech. Both responses will return the answer to the User Interface. In **figure 4.2.1** the message generator generates and gets the response and creates text for it. The text response is passed to the user interface or to the voice response module, if desired, and then gets passed to the user interface. All of these methods are public to the user interface as it needs all of them to adequately display responses to the user.

4.3 User Interface

The User Interface (UI) component will consist of two parts: the server side user interface and the local chatbot user interface. The server side UI will only be used by admins and accounts granted access by the admin, these accounts will have the ability to view and edit the database. The chatbot UI will be used by the general public who will insert a question related to Northern Arizona University into a text box or via speech to text, then submit their question to the chatbot and receive a clear and concise answer from the database within a reasonable amount of time.

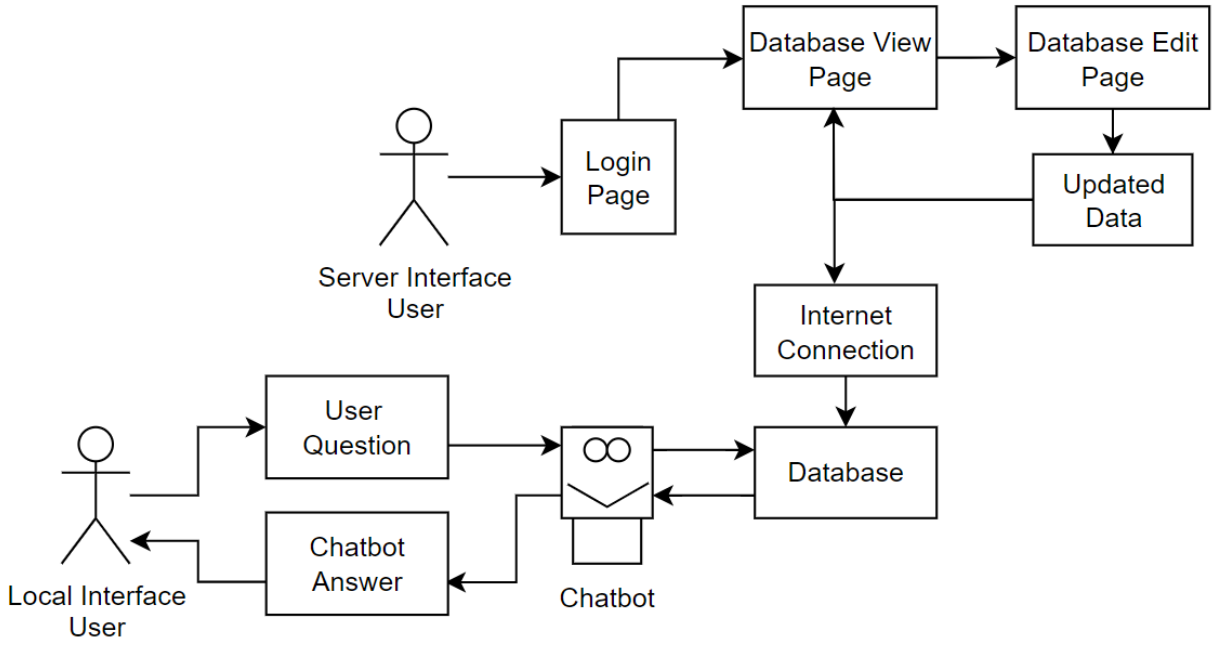


Figure 4.3.1 User Interface Diagram

4.3.1 Server Interface

The server side will consist of three main pages as seen in **Figure 4.3.1**. The first being the login page for admins, and accounts graduated access by the admins to interact with the database. Second is a page to view the current entries into the database, and third is a page for adding/removing or editing entries in the database. These three pages will be crucial to help maintain a secure database and easily manipulate the database by authorized users. Page two and three are separated in order to more clearly define when an entry is to be changed or viewed. Once a database entry has been added, removed, or edited in any way the new information will be updated on the local database as well once a stable internet connection is made as seen in **Figure 4.3.1**.

4.3.2 Local Interface

The local interface will be used almost exclusively by students and parents to ask questions about the NAU campus. There will only be one page where a friendly robot face will greet any user as seen in figure 4.3.1, this is important in order to have an approachable

feeling with the chatbot. There will be very few visible UI elements on the screen, the main source of interaction will be speaking to the chatbot. The chatbot will then display what was heard and ask a confirmation if what was heard was correct with a yes or no. If yes then it will begin searching the database, if not then it will try asking once again to repeat the question and if still left misunderstood then a text box will pop up and give the user the ability to directly type the question.

After the initial process of asking the question is finished it will send the question to the backend of the program and receive the answer in under ten seconds and use text to voice to reply to the user with the answer as seen in **Figure 4.3.1.** The answer will also be displayed in closed captions in case the chatbot is not easy to understand for a user.

4.4 Web Portal

The web portal will be where users can interact with the online database containing question and answer pairs. Users will need to have login credentials given to them so that they can access the platform as not just anybody should have access to the data. Once the user is logged in they are able to add new information to the database. Users will be able to edit information that they entered into the database. Administrators are able to edit all data entered into the database and will have absolute authority over all question/answer pairs within the database. The web portal or more specifically the server is the anchor point of the project as it will serve an area where users can update information, add information, and interact directly with the database that the local chatbot will use to answer questions. The server will as a whole contain both the online database and the portal needed to interact with it. The portal can access the database to update it and the database will be used to store question/answer pairs as well as user credentials. User credentials will not be accessible through relations in the database.

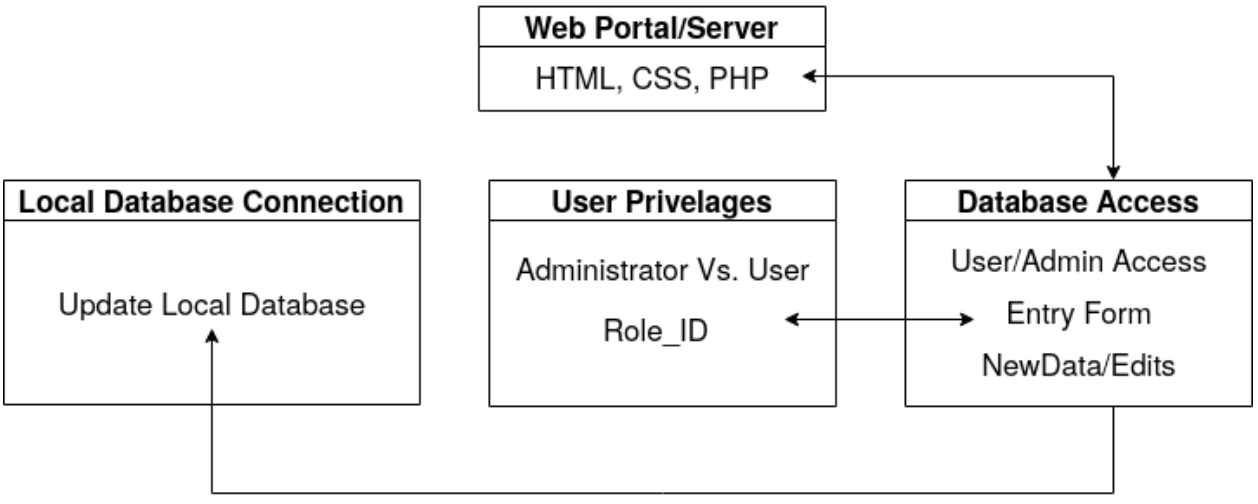


Figure 4.4.1 Web Portal Diagram

The overarching component of the web portal is to serve as a place where users can input new information directly into the database as well as to have a place to update old information. The server will also be a place for the database to live and for that to work the online database will need to be able to communicate with the local database to edit/add information when the local software is connected to the internet. Users will be able to

update/add new information into the database through an entry form that will interact with the database. There will also be a way to differentiate between whether a user is an administrator or if they are a “normal” user; this will be done through a role ID so that the portal will know what tools to display to what users. The web portal will be hosted on an AWS server where PHP, HTML, and CSS will be used to provide the user with a clean and intuitive interface.

The web portal will have full access to the database as it’s needed in order to make changes to it. The database will also contain all of the needed user credentials. The database will be able to access the web portals user interface in order to display database information to users. The web portal will not have any access to the local database in the sense that it will not display or directly change data contained within it. In **Figure 4.4.1** the local to online database communication is only one way; the local database can grab information from the online database. The communication between the database and the portion labeled user privilege displays that the user privileges will be controlled through the database. **Figure 4.4.1** also shows that the database is accessible through the web portal using PHP, HTML, and CSS to create a user interface.

4.5 Database

The database will be responsible for storing database management accounts for both administrators and general users. It will also be responsible for storing question and answer pairs. The database is the connection between the problem retrieval system and message generator system. Connections will be made from the retrieval system, information will be then passed from the database to the system responsible for constructing the user response. On the back end of things, the database will be manipulated by all management accounts, with general users having the specific restriction of only being able to manipulate information that user added. For example an administrator might make several general accounts corresponding to multiple professors. This restriction allows Professor A to update information about himself but not update information about Professor B.

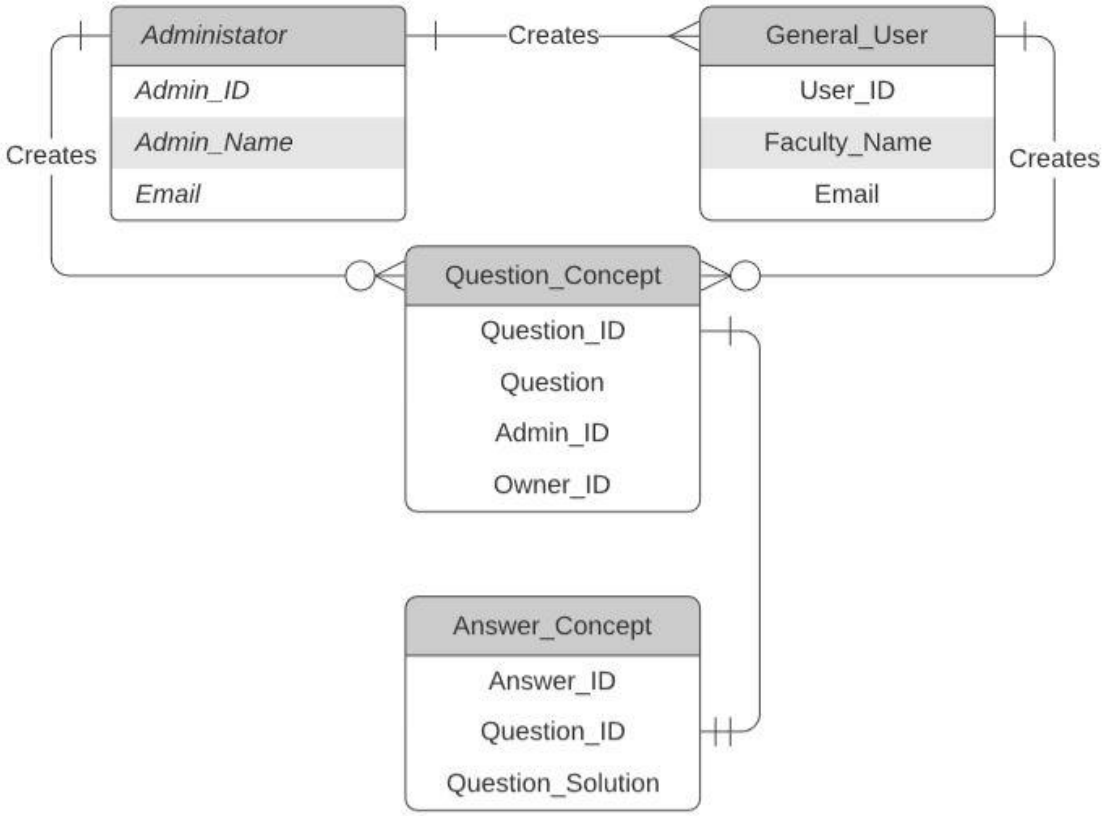


Figure 4.5.1 Database

Figure 4.5.1 outlines the database itself. Administrators have the ability to create general users as well as create question answer pairs, as shown in the figure these entities have a one to many relationship. Meaning one administrator can create multiple general users, but general users could only have one creator. General users will be able to create question pairs as well. The Question_Concept entity will contain both the administrator id as well as an owner id. These identities could be the same if the administrator was the one to create the question pair. Questions_Concept entities must have a one to one relationship with an Answer_Concept entity as shown in **figure 4.5.1**. The service that the database provides is the storage of all necessary information for the creation of the Chatter Jack chatbox. The database will contain a question and answer bank as well as the users responsible for maintaining and updating that bank of information.

5.0 Implementation Plan

JabberJack

Figure 5.1 focuses on the development plan this semester so that JabberJack can finish the Minimum Viable Product (MVP) before spring break. All implementation tasks will be divided into two different parts: design and development. The advanced level chatbot will meet the Text-To-Voice/Voice-To-Text functionality for the final product delivery, which is the foundation for the future project that is able to be an intelligent brain for "LumberJack" bot.

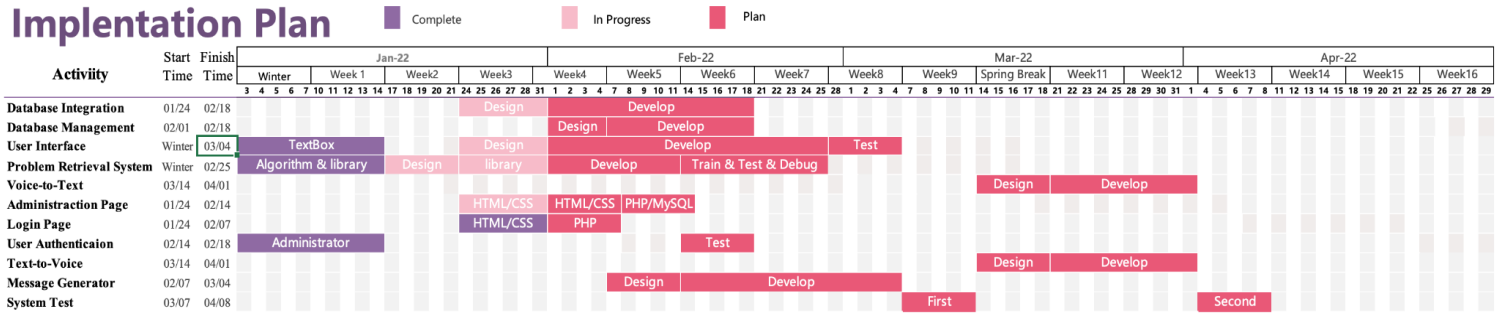


Figure 5.1 Implementation Plan

Here are the explanations for implementation plans:

- **Database Integration (01/24/2022 - 02/18/2022)**

  As the database is the first step to implementing the key requirements, JabberJack will begin to collect and integrate the data set of Question and Answer (QA) from the end of 2022 January. This will be finished by February 18th. JabberJack will utilize the QA pairs as the corpus to train the chatbot so that it could be the intelligent brain.

- **Database Management (02/01/2022 - 02/18/2022)**

  The database will be made up of two parts: local and online databases. The database management system will test with the administration system, which can update the Q/A through the administration system and online database.

- **Problem Retrieval System (01/17/2022 - 03/04/2022)**

  The Problem Retrieval System is the core of the chatbot, which affects the customer's satisfaction directly. It will utilize the spaCy and saved corpus to train the model. JabberJack will spend a lot of time to make sure that the system could return the instantaneous and accurate answer to the Message Generator.

- **Administration System (01/24/2022 - 02/18/2022)**

The overarching component of the administration system is the web application, which consists of four parts: login page, administration page, user authentication, and online database. This process will integrate testing with the database system management.

- **User Interface (01/24/2022 - 03/04/2022)**

  JabberJack will start designing the user interface (UI) from January 24th. The basic design should focus on the login window and chat window. The higher level should be a funny face shown on the screen. This process will integrate the Message Generator.

- **Message Generator (02/07/2022 - 03/04/2022)**

  The Message Generator will integrate with the Problem Retrieval System, User Interface, and Database. JabberJack will utilize it to debug and test the system so that its answers are always reasonable and polite.

- **System Test**

  There are two system tests, one for examining the MVP, the other for the final product delivery.

6.0 Conclusion

JabberJack

Dr. Andy Wang, the dean of the college of engineering, hopes that this chatbot will be able to help solve Northern Arizona University's problem of scattered information. The ChatterJack will become a chatbot that can interact with users and give users useful information about NAU. It will allow users to easily access information through an intuitive interface.

The chatbot is composed of five separate modules that will come together to create the system; there is the problem retrieval system and the message generator, the user interface, the web portal, and the database. The problem retrieval system and message generator will get the user's questions from the user interface and then find and form an answer to return back. The user interface is responsible for the view, the delivery of the answers and how users will interact with the chatbot as well as the interface for the web portal. The web portal is a server where users can access the database and edit/remove/add information to the database; this server is not open to everyone and users will need allowed access to the web portal. Finally there is the database which is the cornerstone of the project. The database will contain all of the information that the chatbot is able to utilize, it will also hold all user credentials for the web portal. The online database will be accessible through the web portal where users can update the information while the local database will pull information from the online database when it connects to the internet. With all of these modules working together the ChatterJack chatbot will be able to efficiently and correctly answer user questions.

Currently, JabberJack is developing five core modules at the same time, which will meet MVP criteria by early March. The web portal has been implemented and is slowly being built. The team will pay more attention to the database, user interface, problem retrieval system, and message generator in the remaining weeks of February. In March 2022, the team will focus on testing the centralized chatbot system and fully implementing all of the needed modules.

While the ChatterJack chatbot is not anywhere near the level of something like Amazon's Alexa or Apple's Siri, it is a start. This project will help NAU establish the foundations of a chatbot system that will continue to grow for years after.

7.0 Reference

JabberJack

[1] Miklosik, Andrej & Evans, Nina & Qureshi, Athar. (2021). The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review. IEEE Access. 9. 106530-106539. 10.1109/ACCESS.2021.3100885.

[2] NAU. 2021.(2021). Retrieved November 3, 2021 from https://in.nau.edu/campus-health-services/

[3] NAU. 2021. Facts and stats. (2021). Retrieved November 3, 2021 from https://nau.edu/about/facts-and-stats/

[4] Rooein, Donya. (2019). Data-Driven Edu Chatbots. 46-49. 10.1145/3308560.3314191.

[5] Smilijanic Stasha. 2021. Amazon Alexa Statistics, facts, and Trends. (February 2021). Retrieved November 16, 2021 from https://policyadvice.net/insurance/insights/amazon-alexa-statistics/