COGNICHAT

(An Intelligent Chatbot for Personalized Financial Services)

Minor Project Report

Submitted By:

Ishika Agarwal	(18103034)
Devanshi Garg	(18103045)
Riddhi Mahajan	(18103110)
Ishita Arora	(18103131)

Under the Supervision of:

Prof. Poonam Saini

Department of Computer Science and Engineering

Punjab Engineering College

DECLARATION

We hereby declare that the project work entitled "CogniChat" is an authentic record of our own work carried out at Punjab Engineering College (Deemed to be University), as a requirement of Minor Project for the award of degree of Bachelor of Engineering (Computer Science and Engineering), under the guidance of Prof. Poonam Saini (Department of Computer Science and Engineering) during August to December 2020.

We further declare that the information has been collected from genuine and authentic sources and we have not submitted this project report to this or any other university for award of diploma or degree of certificate examination.

- Ishika Agarwal

- Devanshi Garg

- Riddhi Mahajan

- Ishita Arora

CERTIFICATE

This is to certify that the project entitled CogniChat by Ishika Agarwal, Devanshi Garg, Riddhi Mahajan and Ishita Arora is an authentic record of our work carried out under the supervision of Prof. Poonam Saini, Computer Science and Engineering Department, Punjab Engineering College (Deemed to be University), Chandigarh in fulfilment of the requirements as a part of Minor Project for the award of 2 credits in semester 5 of the degree of Bachelor of Technology in Computer Science and Engineering.

Certified that the above statement made by the students is correct to the best of my knowledge and belief.

Prof. Poonam Saini
(Faculty Mentor)
Department of Computer Science and Engineering
Punjab Engineering College
(Deemed to be University)
Chandigarh

Ishika Agarwal
18103034

Devanshi Garg
18103045

Riddhi Mahajan
18103110

Ishita Arora
18103131

Dated: 17th December 2020

ACKNOWLEDGEMENT

The completion of any inter-disciplinary project depends upon cooperation, coordination and combined efforts of several sources of knowledge. It gives us immense pleasure to take this opportunity to thank Dr. Dheeraj Sanghi for giving us such a great opportunity to do our Minor project in their esteemed organization PEC University of Technology, Chandigarh.

We owe our profound gratitude to our project Mentor, Dr. Poonam Saini, who took keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

Finally, no word will be enough to express my deepest reverence to family and friends who have willingly helped us out with their abilities and without whose enthusiasm and support, We wouldn’t have been able to pursue our goals.

ABSTRACT

Even today, most individuals manage their finances all by themselves, which is prone to human error. While people have some compulsory scheduled expenses, impulsive expenditures often destroy the balance. CogniChat is an effective technology solution that provides the user with fast responses, 24/7 availability and personalized service thereby improving the user's financial well-being. It allows customers to manage requests in a faster and a more efficient way. CogniChat can assist clients in conducting a variety of financial transactions in a conversational and secure manner. From reviewing an account to making payments, to taking advice on a particular expenditure, the client can handle simple tasks on their own using this chatbot.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

In today’s digital world, managing finances is a challenge being faced by most individuals. While people have some compulsory scheduled expenses, impulsive expenditures often destroy the balance. For most people, the expenditure often exceeds the money present in their account and they are unable to keep a track of where they are spending their money. There is a need to keep a check on their financial status and provide them with the right assistance. Thus, our project is an attempt to address this issue. Our chatbot aims at providing the users with financial services which are personalised for them in the most interactive manner.

Nowadays, people interact with systems more and more through chatbots. The days of solely engaging with a service through a keyboard are over. Hence our project takes the form of a responsive web application where the user can post his/her query in the form of speech or text and get the appropriate responses. Artificial Intelligence techniques such as Natural Language Understanding have been used to facilitate training of the chatbot for generation of responses specific to the users’ queries.

Any user who has registered with CogniChat will be able to access and update his/her financial details such as account balance, expenditure among others. The user will be able to gain some valuable advice on whether or not he/she should make a particular transaction based on his current savings. Most importantly, the user will be able to analyze the expenditure that he/she has made.

Web Application

CogniChat takes the form of an interactive web application. The user can inform the chatbot about every activity where the money has been spent. The bot automatically infers the category of expenditure. These categorical expenses can then be accessed and analysed by the user in a graphical manner.

CogniChat uses HTML, CSS and JavaScript for the front-end and Django for the server-side scripting needs. The backend code is written in python.

CHAPTER 2

MOTIVATION AND PROBLEM FORMULATION

2.1 Motivation

- In today's world, managing finance is a challenge being faced by individuals. Most often, users are unable to keep a track of where they are spending their money and hence they need some assistance to keep a check on them.
- Our aim was to build a chatbot which can provide the user information about his financial status in an interactive manner.
- In addition to some basic financial assistance, CogniChat also intelligently categorizes the expenses thus keeping the user informed of his spending habits.
- It takes the form of a responsive web application, wherein the user can post his/her queries in form of text or as a speech input and get appropriate responses.
- It must also provide basic financial details of the user.
- It acts as a simple, intelligent bot that keeps a track of information regarding your private finances.
- It enables the user to analyze his/her spending habits and inspect where and how the money is being spent.
- Graphical representation of the various financial details must also be available for easy comprehension by the user.
- Since, no application combines all these features, we decided to go ahead and make our own chatbot, CogniChat.

2.2 Problem Formulation

We aim to build a simple, intelligent chatbot which will provide personalized financial services to a user based upon respective wallet details, account information and previous and scheduled expenditures. The chatbot will take the form of a responsive web application, wherein the user can post his/her queries and explore the best possible responses.

CogniChat is a solution for personalized financial assistance with an easy to use interface. It first takes a http request with a text from the user. Natural Language Understanding (NLU) is used to get the intent from the text. A trained model (an LSTM neural network implemented using Google's TensorFlow and Python) is used to predict the next action of the bot. The response is sent to the user and waits for the next sentence from the user.

CHAPTER 3

REQUIREMENTS AND DESIGN

3.1 Requirements

3.1.1 CogniChat Web Application

- **Knowledge of Web Development:** Knowledge of software development for the web is an essential requirement, since the registration process of a vehicle is implemented by a web-based application. The team has knowledge of Web Development, and has used HTML, CSS and JavaScript for the frontend and Ajax, SQLite and Django for the server-side scripting and logic implementation.
- **Knowledge of APIs:** The developer needs to have prior knowledge Application Programming Interfaces and how to create custom APIs. These APIs are necessary to send requests and queries to the backend via forms.
- **Knowledge of form handling:** Forms need to be handled since application is done by filling up a form. We have used Django Forms and Models to send information.
- **Knowledge of Database connectivity:** Since the application data needs to be stored, hence there is a requirement of knowledge related to databases and interacting with the said databases. The team has used SQLite as a database engine to store data and Django to interact with the database.

3.1.2 Natural Language Processing

- **Knowledge of NLP:** A basic understanding of NLP, RNN and LSTM is required to compare various models and their results. Knowledge of various GRUs is also helpful. Basic understanding of Deep Learning is a must too.
- **Knowledge of Python:** Python is the most widely used language for implementation of Deep Learning Libraries like TensorFlow and Keras. The Syntax and working of python must be known to comfortably work on this project.
- **Knowledge of TensorFlow:** TensorFlow and Keras are deep learning libraries in python which provide easy implementations of CNN, RNN and ANN layers. One must know the basics of tensors to work with Tensorflow.

3.2 Design – UML Diagrams

3.2.1 Use-Case Diagram

A UML use case diagram is the primary form of system/software requirements for a new software program. It does not show the details of the use cases. It only summarizes some of the relationships between use cases, actors, and systems. It does not show the order in which steps are performed to achieve the goals of each use case.

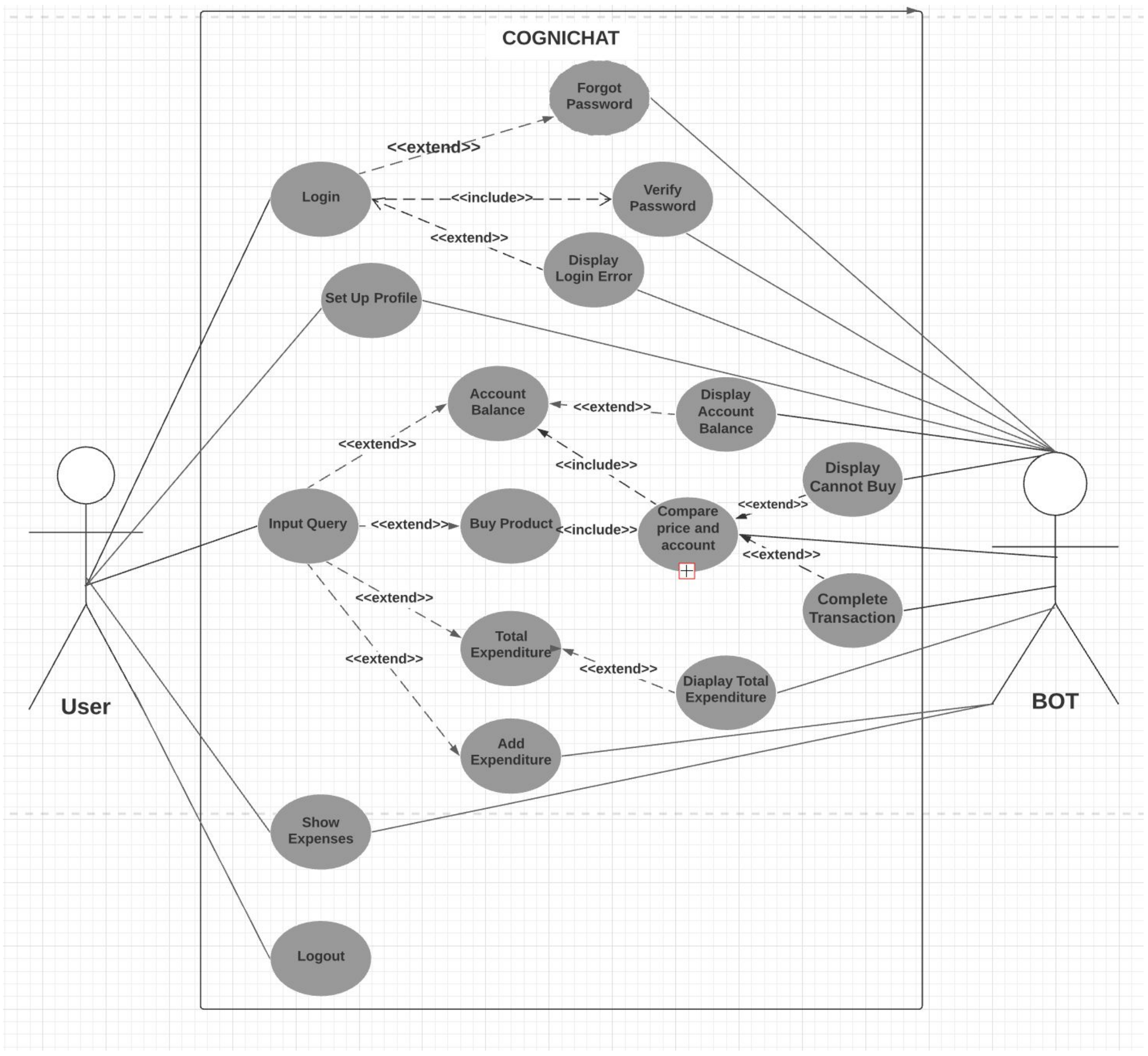


Illustration 1: Use-Case Diagram

3.2.2 Class Diagram

The UML Class diagram is a graphical notation used to construct and visualize object oriented systems. A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Purpose of Class Diagram:

- Shows static structure of classifiers in a system
- Diagram provides a basic notation for other structure diagrams prescribed by UML
- Helpful for developers and other team members too
- Business Analysts can use class diagrams to model systems from a business perspective

UML diagrams precisely convey how code should be implemented from diagrams, hence it is imperative to model them correctly.

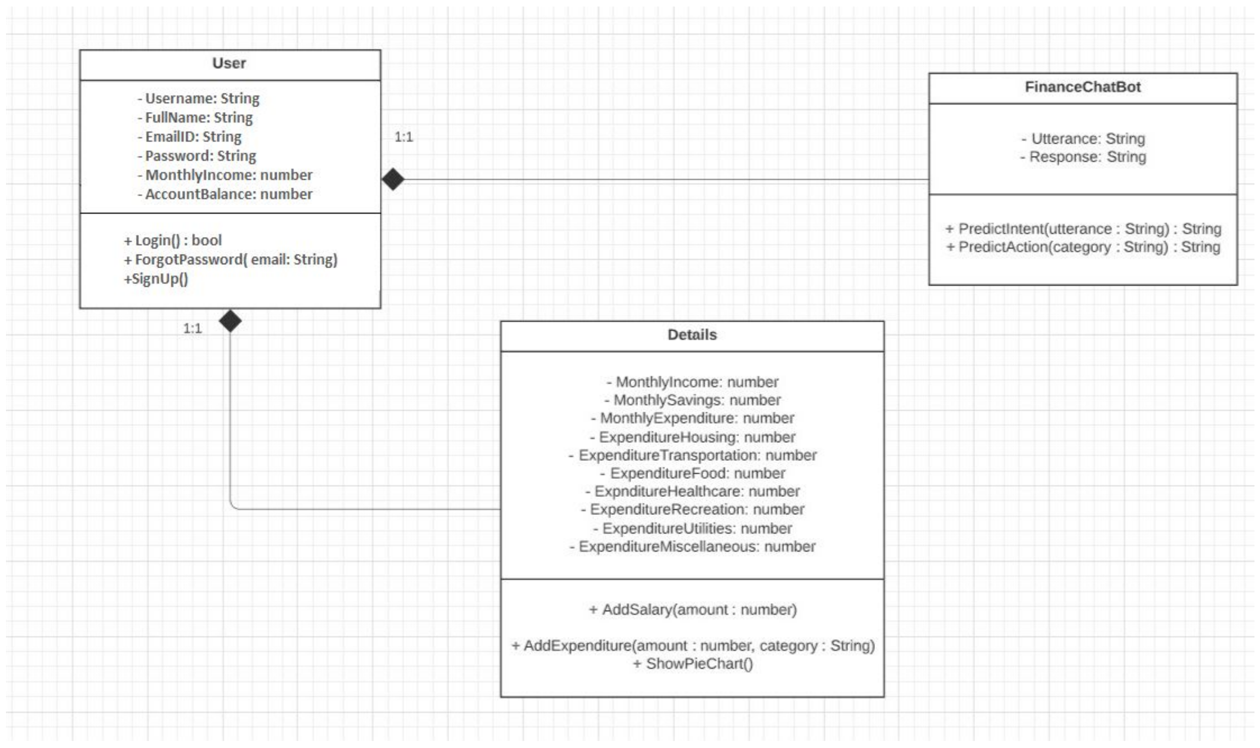

User
- Username: String
- FullName: String
- EmailID: String
- Password: String
- MonthlyIncome: number
- AccountBalance: number
+ Login() : bool
+ ForgotPassword(email: String)
+SignUp()
1:1
FinanceChatBot
- Utterance: String
- Response: String
+ PredictIntent(utterance : String) : String
+ PredictAction(category : String) : String
1:1
Details
- MonthlyIncome: number
- MonthlySavings: number
- MonthlyExpenditure: number
- ExpenditureHousing: number
- ExpenditureTransportation: number
- ExpenditureFood: number
- ExpndiitureHealthcare: number
- ExpenditureRecreation: number
- ExpenditureUtilities: number
- ExpenditureMiscellaneous: number
+ AddSalary(amount : number)
+ AddExpenditure(amount : number, category : String)
+ ShowPieChart()

Illustration 2: Class Diagram

3.2.3 State Chart Diagram

Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model the lifetime of an object from creation to termination. Statechart diagrams are also used for forward and reverse engineering of a system.

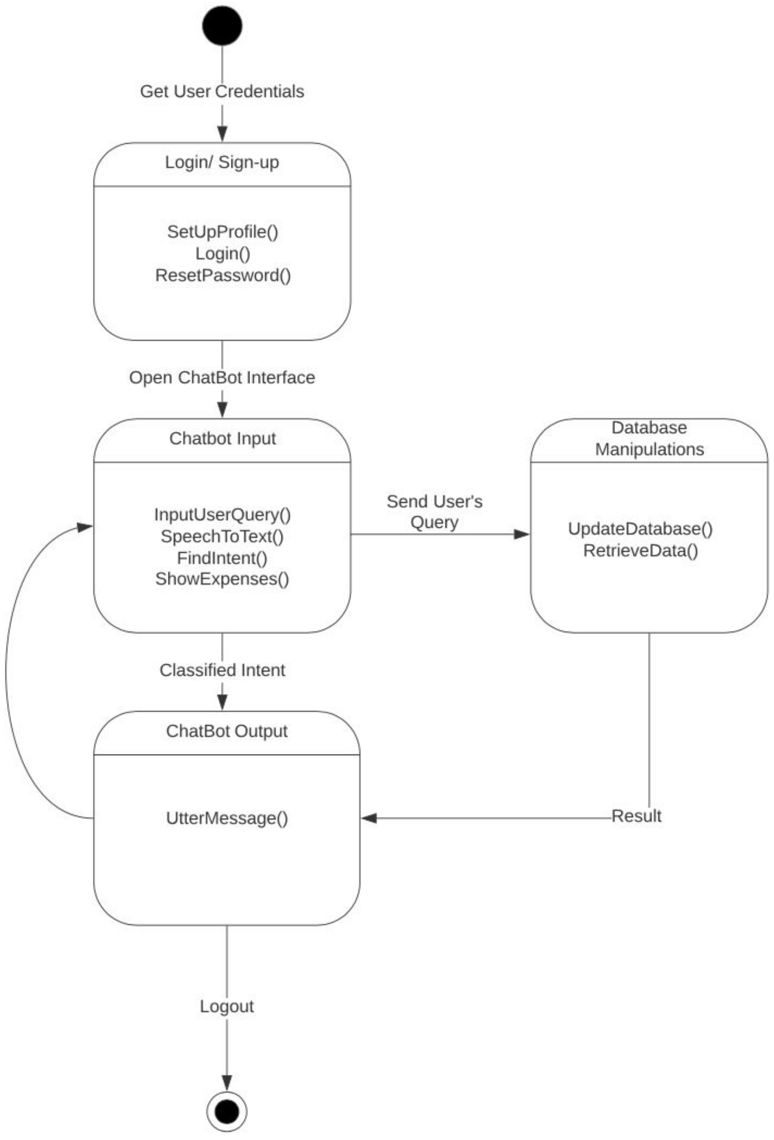


Illustration 3: State Chart Diagram

3.2.4 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

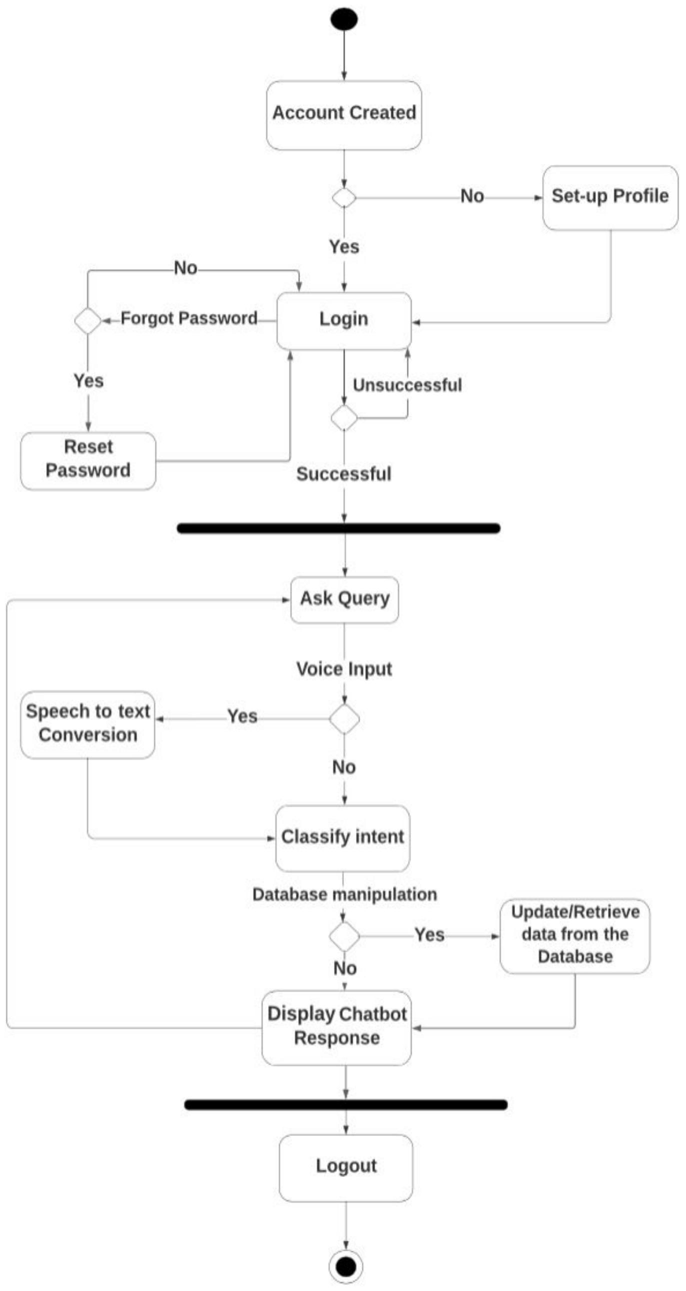


Illustration 4: Activity Diagram

3.2.5 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Uses of Sequence Diagrams:

- Used to model and visualise the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.

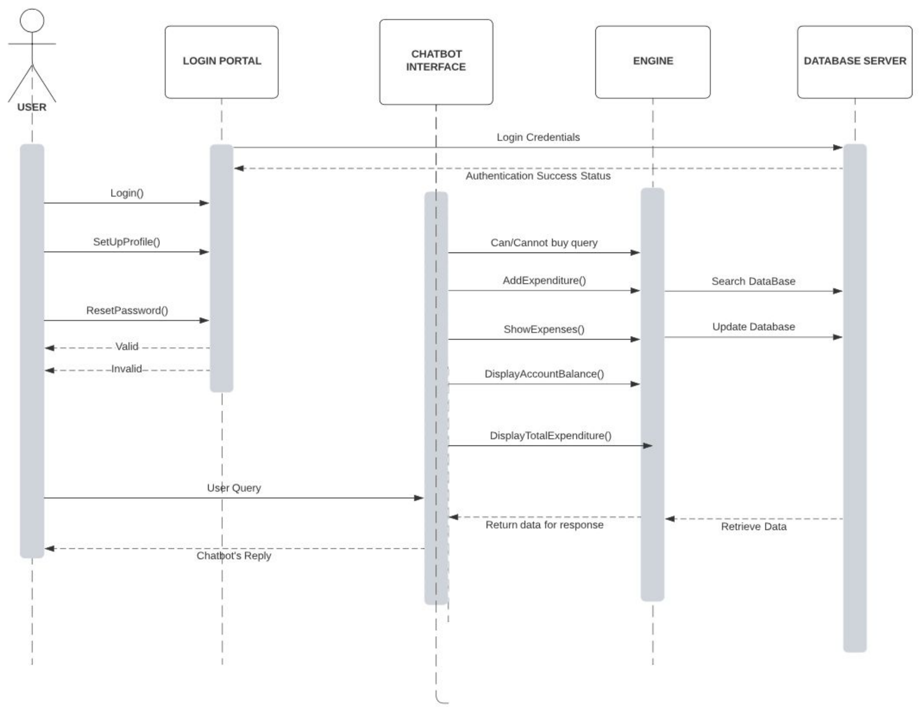


Illustration 5: Sequence Diagram

3.3 Database Design

The user details are stored in two databases:

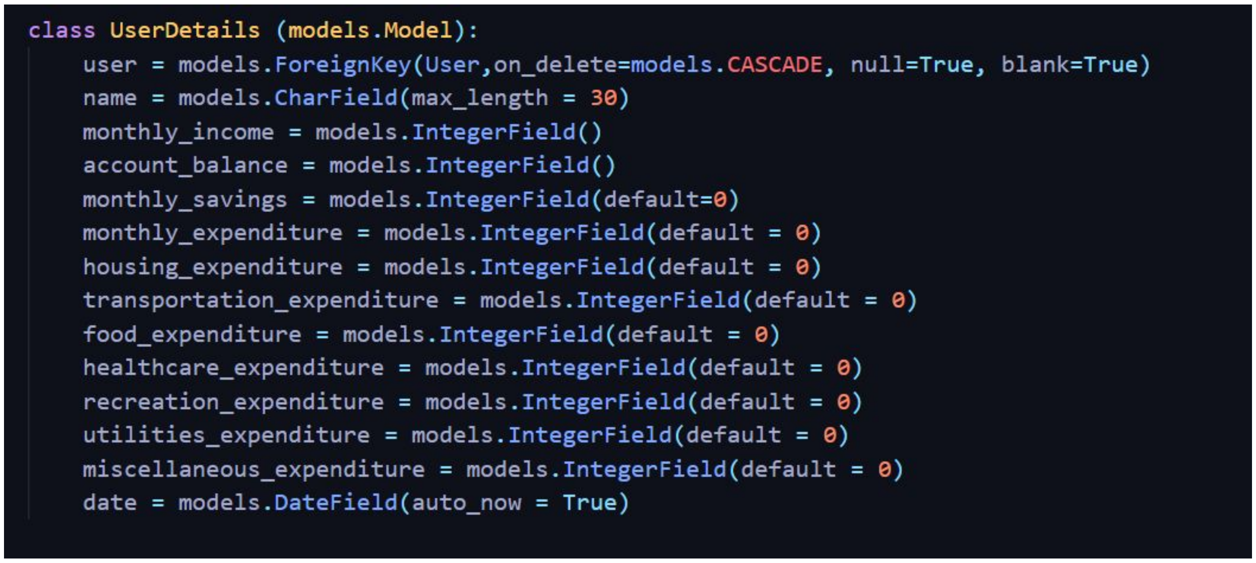
3.3.1 User Database

This is the database used for user authentication. It includes the fields user_id, username and password (which is stored in encrypted form for security reasons). user_id is required to create different user details databases for each of them. user_id acts as foreign key in the user details database.

3.3.2 User Details Database

We have stored other user details such as the user's account balance, total expenditure and expenses in various categories such as food, transportation etc in this database. These values are either updated or retrieved based on the user's query. We have used user_id as the foreign key to connect this database to the user database.

Here we have a screenshot of the exact model that was created.



```python
class UserDetails (models.Model):
    user = models.ForeignKey(User,on_delete=models.CASCADE, null=True, blank=True)
    name = models.CharField(max_length = 30)
    monthly_income = models.IntegerField()
    account_balance = models.IntegerField()
    monthly_savings = models.IntegerField(default=0)
    monthly_expenditure = models.IntegerField(default = 0)
    housing_expenditure = models.IntegerField(default = 0)
    transportation_expenditure = models.IntegerField(default = 0)
    food_expenditure = models.IntegerField(default = 0)
    healthcare_expenditure = models.IntegerField(default = 0)
    recreation_expenditure = models.IntegerField(default = 0)
    utilities_expenditure = models.IntegerField(default = 0)
    miscellaneous_expenditure = models.IntegerField(default = 0)
    date = models.DateField(auto_now = True)
```

Illustration 6: UserDetails Model

CHAPTER 4

IMPLEMENTATION

4.1 Web Application

4.1.1 Registration

The user can register on this page in case he doesn't have a pre-existing account.

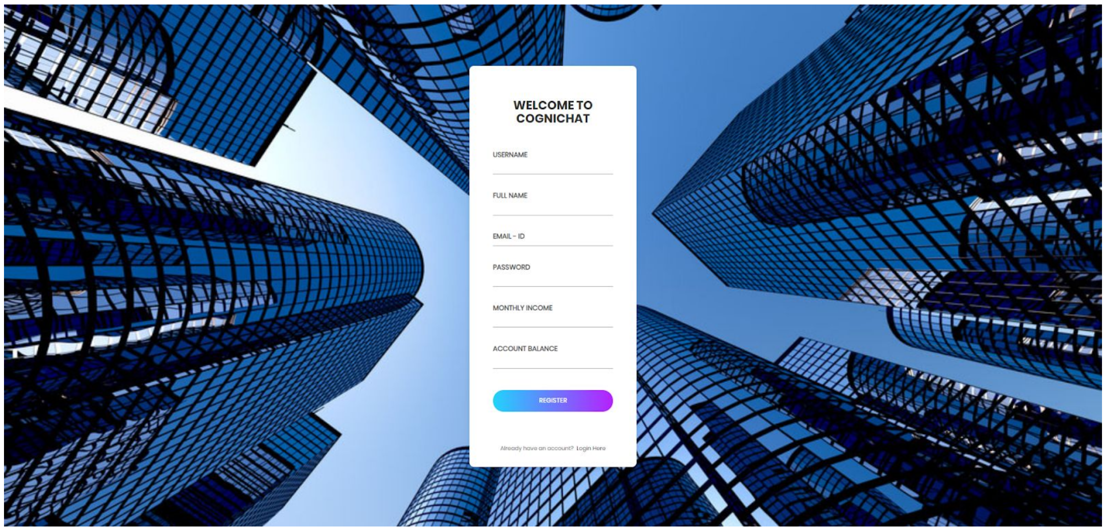


Illustration 7: Registration Page

4.1.2 Login

The user logs in with his Username and Password. This opens up the ChatBot interface, where he can log his queries and get appropriate responses.

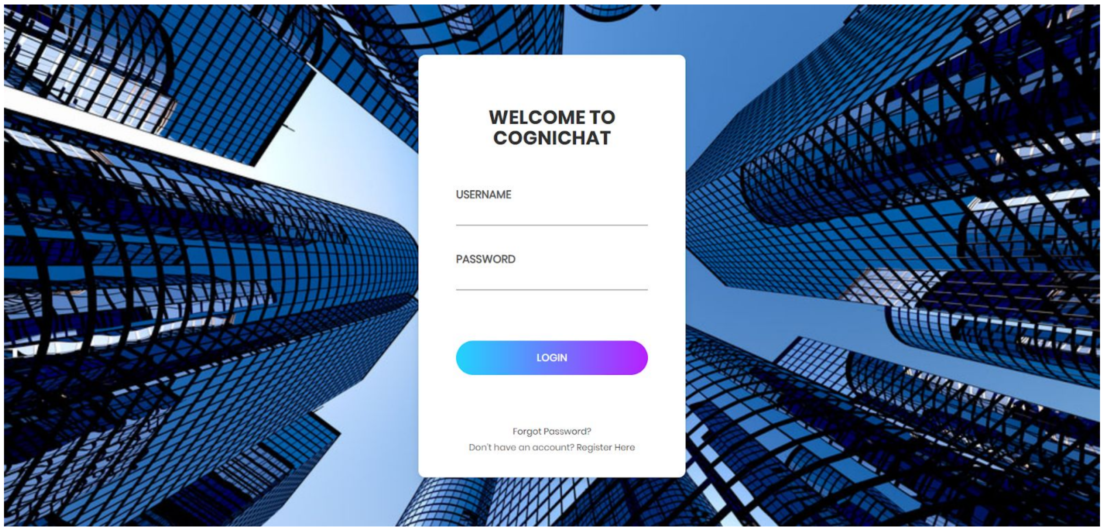


Illustration 8: Login Page

4.1.3 Forgot Password

The user can reset his password if he forgets his original one. This is done using the Email-ID. Inbuilt Django password reset has been used for the same.

As the website is locally hosted for now, the email is displayed on the command line and not actually sent via the mail protocol.

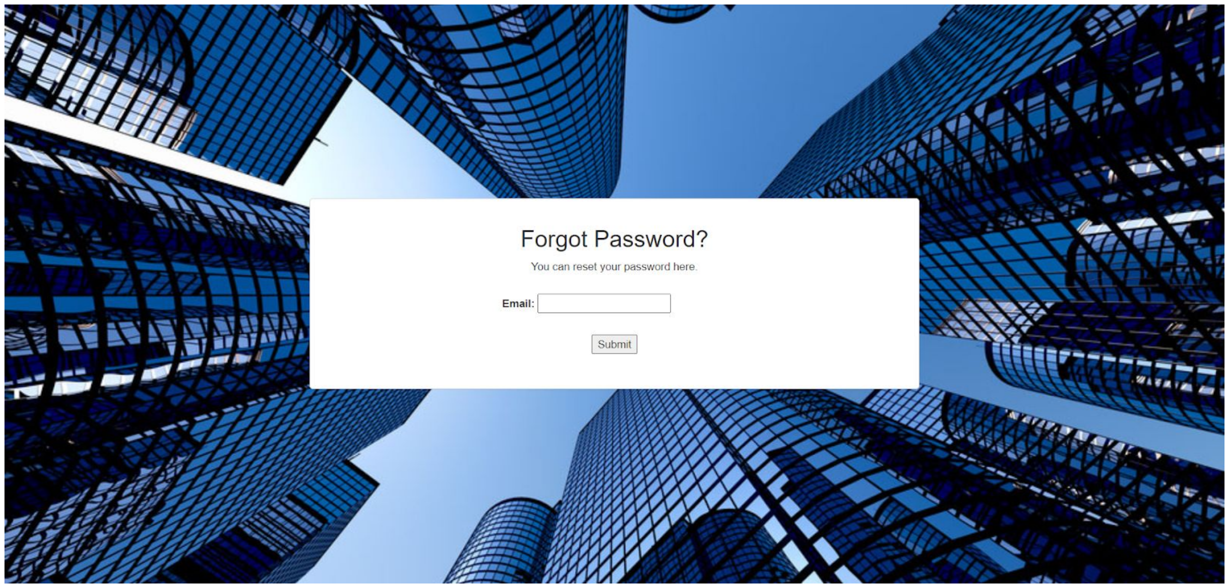
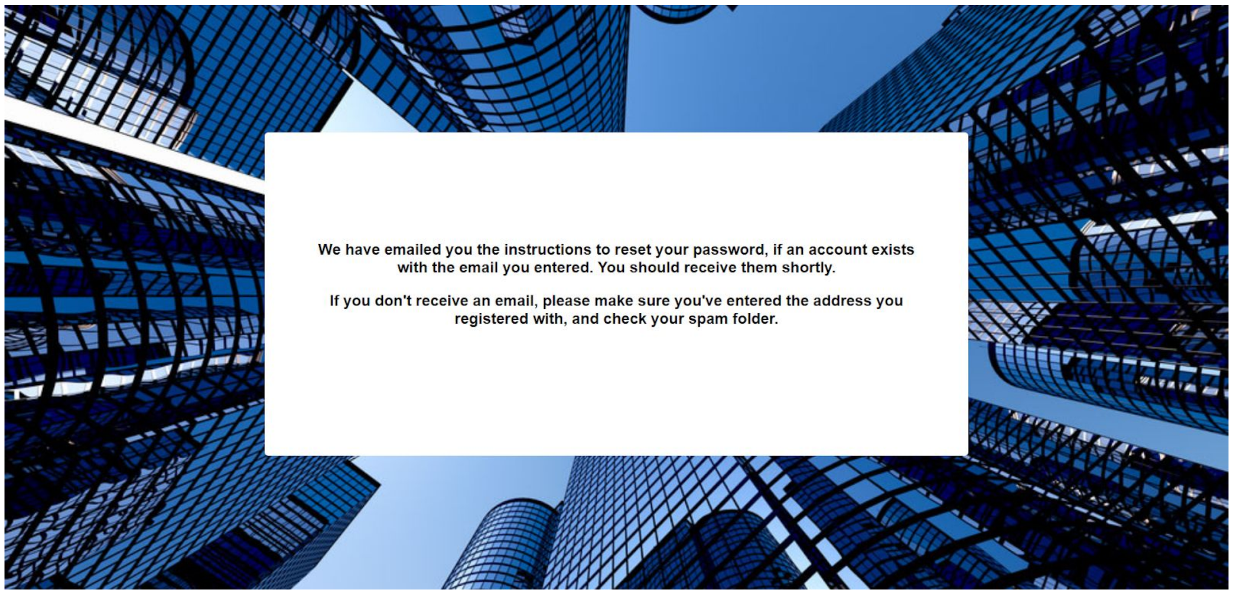


Illustration 9: Forgot Password



We have emailed you the instructions to reset your password, if an account exists with the email you entered. You should receive them shortly.

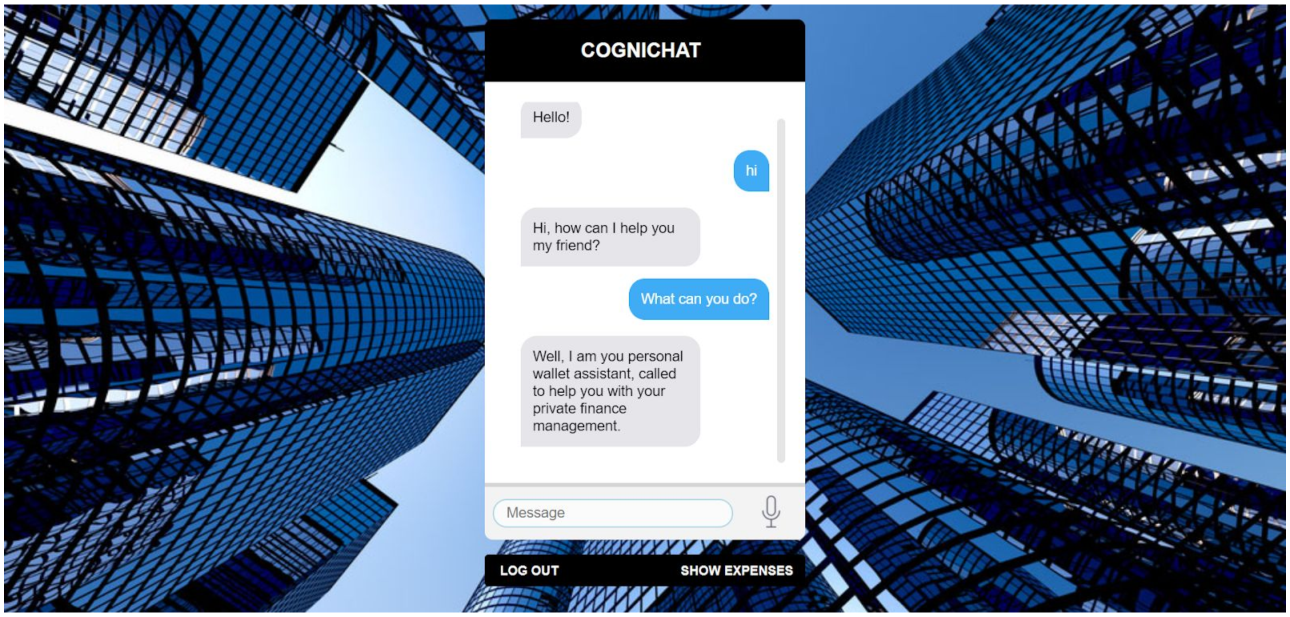
If you don't receive an email, please make sure you've entered the address you registered with, and check your spam folder.

Illustration 10: Reset Password

4.1.4 ChatBot Interface

The user login opens up the ChatBot Interface, which is the central point of use for the user. The user can ask queries and talk to the chatbot here. This process is facilitated by training of 2 models: Intents and Dialogue. Corresponding to the query entered by the user, the intent of that query is classified. Based on the intent, the chatbot identifies the utterance from the dialogue flow and provides the suited response corresponding to the utterance identified.

The user can enter his/her query in form of a text input or a speech input. In case a speech input is entered, the bot converts the speech to text using webKitSpeechRecognition functionality provided by JavaScript.


Hello!

hi

Hi, how can I help you my friend?

What can you do?

Well, I am you personal wallet assistant, called to help you with your private finance management.

Illustration 11: ChatBot Interface

Features of the ChatBot:

1. **Greetings:** The chatbot greets the user when the user initiates his conversation and says goodbye when the user ends it. This makes the chatbot more humanlike.
2. **Account Balance:** The chatbot can provide the current balance present in the user's account. It does so by retrieving information from the SQLite database which is specific for each user.

3. **Total Expenditure:** The chatbot provides the value of the total expenditure made by the user. It is altered with every new purchase made.
4. **Can/Cannot Buy:** The user may ask whether or not he/she is permitted to buy a particular entity or make a specific transaction. The user is required to enter the price involved. Based on this price and the money in the account balance, the chatbot advises the user in favour of or against that particular transaction.
5. **Categorical Expenditures:** When the user tells the chatbot about a particular expenditure that he/she has made, the chatbot automatically infers the category of expenditure, updates the value of that category's expenditure and informs the user.
6. Various categories of expenditure that our chatbot is able to recognize along with sample queries are:
   a. **Food:** I got burgers for 200 Rs
   b. **Housing:** Rent was 10,000 Rs
   c. **Recreation:** I went for a party and spent 5000 Rs there.
   d. **Transportation:** Flight tickets cost me 1,00,000 Rs
   e. **Healthcare:** I spent 2000 Rs on the eye checkup.
   f. **Utilities:** Electricity bill was 500 Rs
   g. **Miscellaneous:** I bought a pet for 20 Rs

   Corresponding to every category, examples have been given as to what kind of queries the user may input.

   For every category the user spends money in, that amount is added to said category as well as the total expenditure. Along with this, the above amount is subtracted from the account balance, which is in turn required for the can/cannot buy feature.

7. In case the user enters incomplete sentences and forgets to enter the price corresponding to every transaction, the chatbot prompts the user to complete the sentence and asks the user to enter the price.

   To enable this feature, the chatbot keeps a track of the second last intent as well, in order to ensure the user doesn't need to speak the whole sentence again.

4.1.5 Show Expenses

The chatbot provides for a pie-chart corresponding to the various categorical expenditures of the user. The user can analyze his/her spending habits via this pie chart representation.

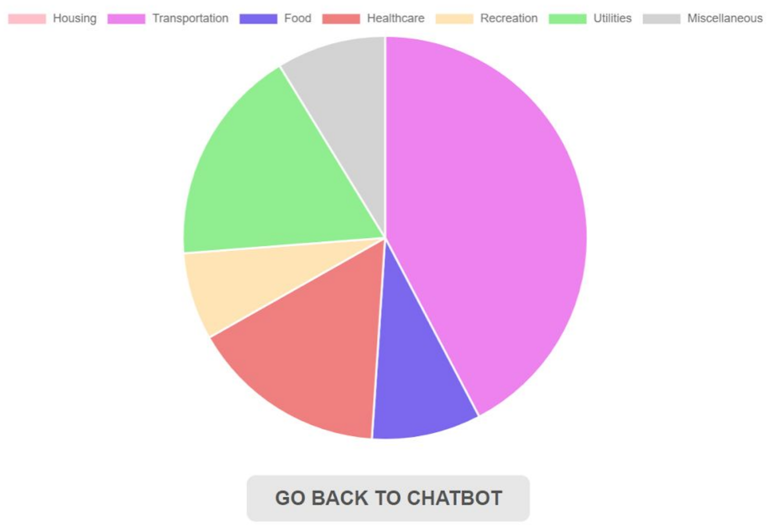


Illustration 12: Pie-Chart Representation

4.2 ChatBot Training

We have used keras which is an open-source software library that provides a Python interface to implement artificial neural networks. Natural Language processing is also done to understand the natural language input by the user and predict and utter the corresponding response. To make the chatbot interactive Natural Language Processing has been used. The model employs Recurrent Neural Networks using LSTM (Long Short Term Memory).

- **Recurrent Neural Networks**

  It is a generalization of a feedforward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. LSTM is a special type of RNN, which has the added capability of remembering long-term dependencies.

Natural Language Processing

It is the technology used to aid computers to understand the human's natural language. Syntactic analysis and semantic analysis are the main techniques used to complete Natural Language Processing tasks.

1. **Syntactic analysis**

   It is used to assess how the natural language aligns with the grammatical rules. Some syntax techniques that can be used are:

   a. **Word segmentation:** It involves dividing a large piece of continuous text into distinct units.
   b. **Lemmatization:** It entails reducing the various inflected forms of a word into a single form for easy analysis.

2. **Semantics analysis**

   It involves applying computer algorithms to understand the meaning and interpretation of words and how sentences are structured. Technique used in semantic analysis is:

   Natural language generation: It involves using databases to derive semantic intentions and convert them into human language.

For our chatbot, a glove vector with 6 Billion tokens and 100 dimensional vectors was used for training the model. The optimizer used was Adam (Adaptive Moment Estimation), and the metric used was accuracy.

Two models were made: one for intent classification, and another for dialog flow prediction. They were trained for 400 and 500 epochs, and had final accuracies of 97.59% and 93.33% respectively.

Datasets for intent classification and dialog flow prediction were problem specific, and were made by us. We tried to make them as exhaustive as possible, however, there remains a solid scope for improvement. The glove vector used for initial training, however, is a universal dataset.

4.3 Technologies Used

4.3.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. HTML also provides a basic structure of the page.

We have 4 web pages: Login, Registration, ChatBot Interface and PieChart Representation.

4.3.2 CSS and Bootstrap

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. It has been used to define the front end styling of various extension components. Styling of all pages has been done with the help of CSS. The Styling of the webpages in the project has been done using CSS and Bootstrap.

4.3.3 JavaScript

JavaScript is a programming language that adds interactivity to the website. JavaScript is a full-fledged dynamic programming language that, when applied to an HTML document, can provide dynamic interactivity on websites. It is an interpreted programming language with object-oriented capabilities. JavaScript has been used for the following:

- **Speech Recognition:** We have used the Javascript framework webkitspeechrecognition to facilitate speech to text conversion in our chatbot.
- **Automatic Scrolling:** Facilitates automatic scroll down to the bottom of the page in case the length of the page increases.
- **Loading Symbol:** Makes the chatbots conversation more interactive by inserting a loading symbol during the buffer time taken by the chatbot to predict the output.
- **Dynamically-Sized Text Box:** JavaScript has been used to make our input text-box dynamic in size by automatic insertion of div tags.
- **Pie-Chart Representation:** Chart.js is an open source library for data visualization which we have used to represent our categorical expenses.

4.3.4 AJAX

Asynchronous JavaScript And XML is used to send and retrieve data from a server asynchronously without interfering with the display and behaviour of the existing page. AJAX is used for the following:

- **Send User Query:** Intents corresponding to the user queries are sent as a request to the server.
- **Retrieve Predictions:** Utterances which are predicted by the AI model are sent back as response from the server.

4.3.5 Django

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. CogniChat uses Django as the web framework. It uses the MTV architecture whose components include Models, Templates, and Views. Inbuilt Django libraries have been used for user authentication and to add "forgot password" functionality.

1. After the user sets up his/her profile, inbuilt Django authentication is used in order to allow the user to log in.
2. In case, the user forgets his/her password, an option to reset the password is also provided. The link to reset the password will be sent to the email account provided by the user. This functionality has been implemented using an in- built Django package.
3. While resetting the password, the user must satisfy essential password requirements such as its length, use of alphabets, numbers and special symbols. This strengthens the password and keeps it secure and safe from hacking.

4.3.6 SQLite

SQLite is a relational database management system (RDBMS) contained in a C library. It is the default database system provided by Django. We have SQLite to store information about the user and his/her various expenses.

4.3.7 Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. The power of python is it's libraries, that cover everything from Scientific computing to Artificial Intelligence and Image Processing. The whole backend code has been written in python.

4.3.8 TensorFlow and Keras

TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. It provides inbuilt implementations of CNN and RNN layers.

TensorFlow was used to make the LSTM Model used to train on the intents and utterances. The dataset used was a glove vector, consisting of 6 Billion tokens.

LSTM is a special type of Recurrent Neural Network which is used for Natural Language Processing. It has the added capability of remembering long-term dependencies. It does so via the use of structures called gates. There are input gates and output gates, which regulate the flow of information.

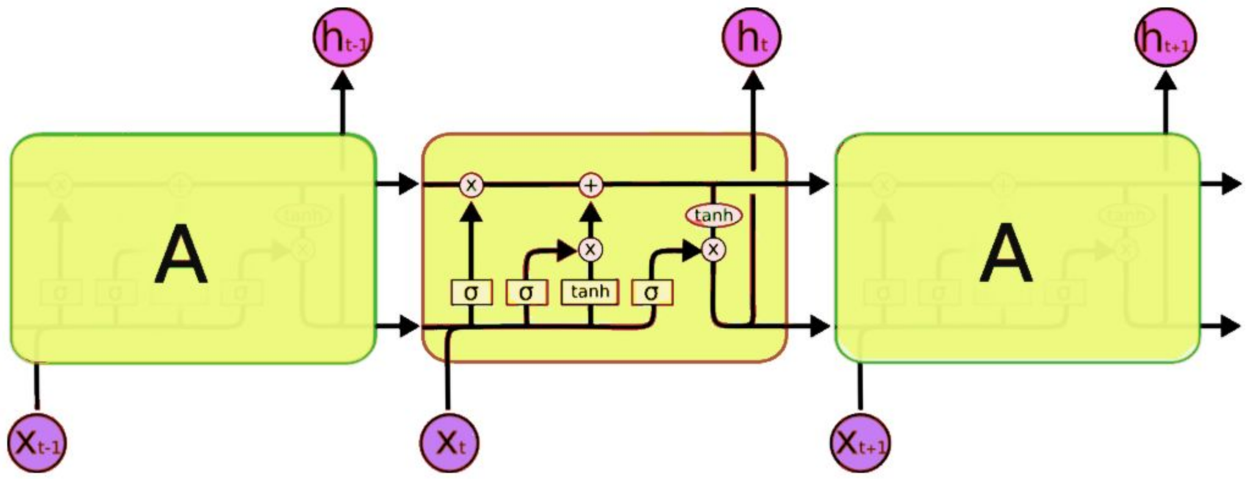


Illustration 13: LSTM Unit

<u>CHAPTER 5</u>

RESULTS AND ACCURACY

5.1 Results

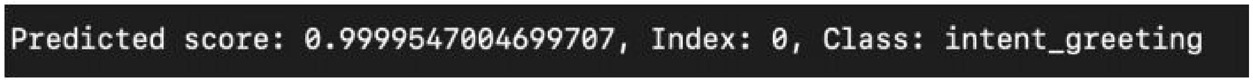
The results achieved so far are:

- The user can input their query either verbally or by typing in a message on the interface.
- The user can access their total account balance.
- The user can access their total expenditure.
- The user can verify whether or not they can buy an item based on the amount of money present in their wallet.
- The user can inform the chatbot about every activity where they have spent money. The bot automatically infers the category of expenditure and updates it in the database.
- The various categories our chatbot can recognize are: Food, Housing, Recreation, Transportation, Healthcare, Utilities and Miscellaneous.
- Categorical expenses can be accessed and analysed using a pie-chart.

5.2 Accuracy

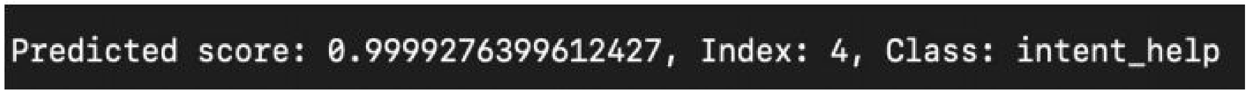
In order to check the accuracy of our project we gave 10 sample test cases and the system classified the intent of the statement given. The results are as follows:

Input Message 1: Hello



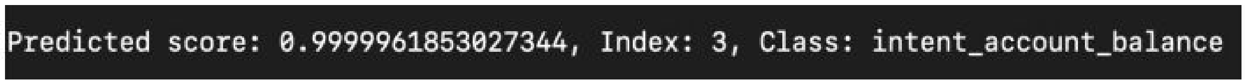
```
Predicted score: 0.9999547004699707, Index: 0, Class: intent_greeting
```

Input Message 2: What features do you have ?



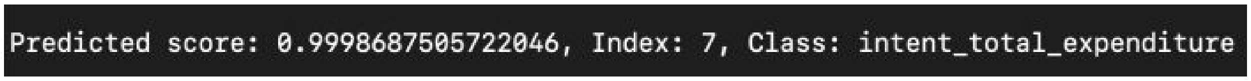
```
Predicted score: 0.9999276399612427, Index: 4, Class: intent_help
```

Input Message 3: How much money is there in my account?



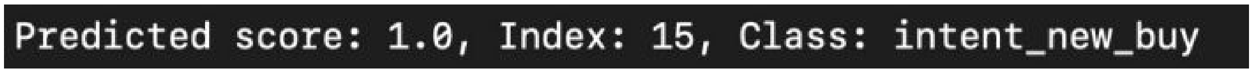
```
Predicted score: 0.9999961853027344, Index: 3, Class: intent_account_balance
```

Input Message 4: What is my total Expenditure?



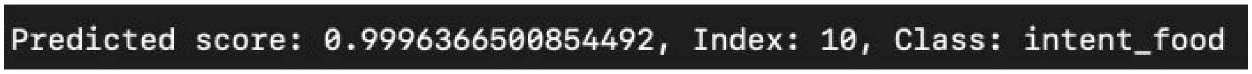
```
Predicted score: 0.9998687505722046, Index: 7, Class: intent_total_expenditure
```

Input Message 5: I want to buy a new phone.



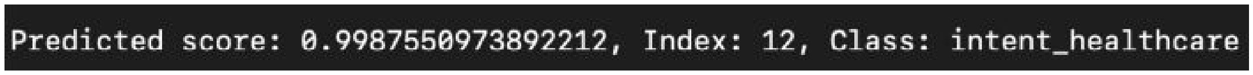
```
Predicted score: 1.0, Index: 15, Class: intent_new_buy
```

Input Message 6: Spent 200 on lunch.



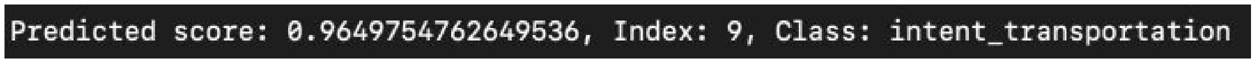
```
Predicted score: 0.9996366500854492, Index: 10, Class: intent_food
```

Input Message 7: Spent 500 for eye checkup.



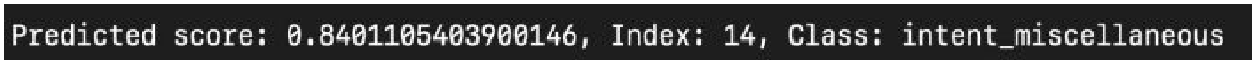
```
Predicted score: 0.9987550973892212, Index: 12, Class: intent_healthcare
```

Input Message 8: I spent 50 Rs on bus tickets.



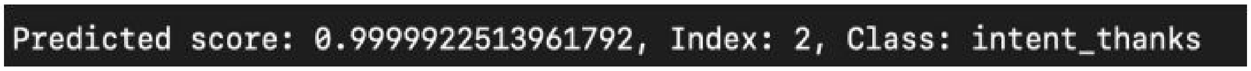
```
Predicted score: 0.9649754762649536, Index: 9, Class: intent_transportation
```

Input Message 9: I bought a dog.



```
Predicted score: 0.8401105403900146, Index: 14, Class: intent_miscellaneous
```

Input Message 10: Thank You very much



```
Predicted score: 0.9999922513961792, Index: 2, Class: intent_thanks
```

Hence, we can see that CogniChat is correctly able identify the intents corresponding to user queries. Further, we will be able to recognize the utterances with the help of our dialogue flow model and yield appropriate responses.

<u>CHAPTER 6</u>

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

CogniChat is a solution for personalised financial assistance with an easy to use interface. It takes the form of a responsive web application, in which the user can post his/her queries either as a typed message or verbally and get appropriate responses.
CogniChat acts as a simple, intelligent bot that keeps track of information regarding your private finances. It answers user queries like current account balance, total expenditure and also advises the user whether a transaction should be made or not. In addition to this, it intelligently categorizes the expenses thus keeping the user informed of his/her spending habits. This enables the user to analyze and inspect where and how the money is being spent.
Improving the efficiency of client service, minimizing human error and resolving client queries quicker is the aim of our chatbot. It allows the users to interact with the chatbot using natural language input. Generation of appropriate responses is facilitated by training the chatbot using tools that expose artificial intelligence methods.

6.2 FUTURE WORK

- Make a mobile application for the same, which could have access to the user's bank account and phone number which would allow for automatic updation of the dataset, based on the messages received.
- Improve conversational abilities of the chatbot. Along with NLU, we can also use NLG to enable the chatbot to generate responses on its own, instead of using pre-fed utterances.
- Add intents which would allow other sources of credit (Up till now only monthly income is used for adding money to the account).
- Train model longer and on more diverse dataset to improve its accuracy.

REFERENCES

https://github.com/meksikann/finbot-2

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

https://nlp.stanford.edu/projects/glove/

https://machinelearningmastery.com/natural-language-processing/

https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1

https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

https://developers.google.com/web/updates/2013/01/Voice-Driven-Web-Apps-Introduction-to-the-Web-Speech-API