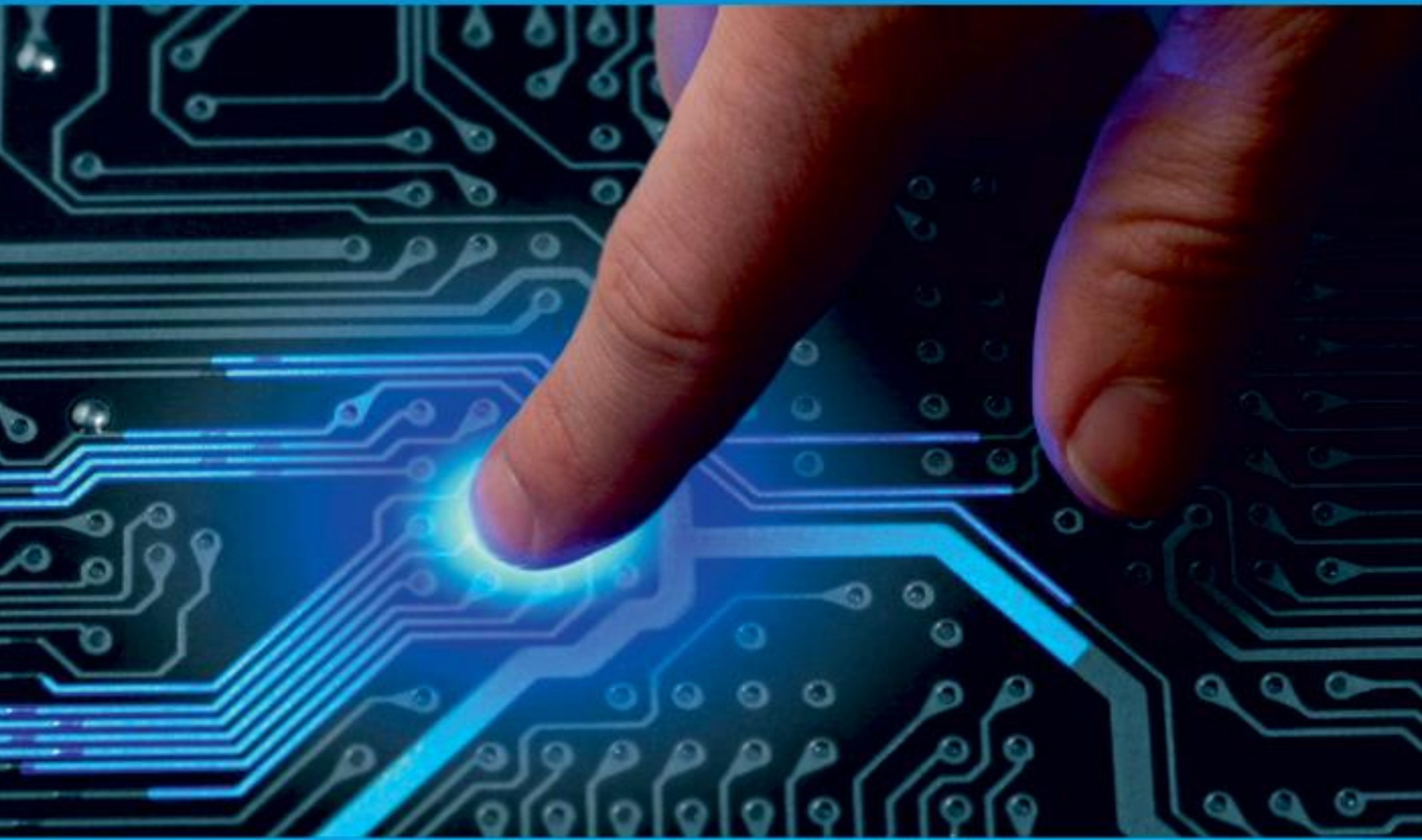
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 3, March 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

EV24 Chat Bot Using Python

AYUSH MANPURKAR

Department Computer Engineering, MM Polytechnic, Pimpri-Chinchwad, Maharashtra, India

ABSTRACT: In recent years, the use of electric vehicles (EVs) has been on the rise. However, the adoption of EVs has been slow due to various factors, one of which is the lack of infrastructure such as charging points and service centers. To address this issue, an EV chatbot has been developed using Python programming language. The chatbot has been designed to assist EV drivers by providing them with useful information on their journeys.

The EV chatbot has several features that make it a valuable tool for EV drivers. One of these features is the ability to show nearby charging points. The chatbot uses location data to identify the nearest charging points and provides the driver with directions on how to get there. This feature helps to reduce the anxiety that EV drivers may feel about running out of battery power.

Another useful feature of the EV chatbot is the ability to show nearby service centers. The chatbot provides drivers with information on the closest service centers and their contact details. This feature helps drivers to quickly and easily find assistance in case of any issues with their EVs.

In addition to charging points and service centers, the EV chatbot also provides information on emergency services. In the event of an accident or breakdown, the chatbot can provide drivers with information on nearby hospitals, police stations, and towing services.

The chatbot has been designed to be easy to use and accessible to all EV drivers. It can be accessed through various platforms, including websites, social media platforms, and messaging apps. The chatbot also has a simple and intuitive interface that allows drivers to quickly access the information they need.

I. INTRODUCTION

The EV chat bot in python is an ai-based conversational program designed to provide assistance and answer questions related to electric vehicles. One of the key features of this chatbot is its ability to show nearby charging points, service stations, and emergency numbers.

The EV chat bot uses gps location tracking to determine the user's location and then provides information about nearby charging stations, service stations, and emergency numbers. users can ask the bot for information on charging stations within a specified radius, and the bot will provide them with a list of the nearest available charging points.
In addition to providing information about charging stations, the EV chat bot can also provide information about nearby service stations. if the user needs assistance with their electric vehicle, they can ask the bot for information on nearby service stations. the bot will provide them with a list of the nearest available service stations, along with contact information.

Another important feature of the EV chat bot is its ability to provide emergency numbers. if the user is in an emergency, they can ask the bot for assistance. the bot will provide them with the relevant emergency numbers for their location, such as the police, fire department, or medical services.

Overall, the EV chat bot in python is a useful tool for electric vehicle owners and enthusiasts. its ability to show nearby charging points, service stations, and emergency numbers can make life easier and more convenient for users, while also promoting the use of electric vehicles.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 4, April 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1104167 |

II.MARKET SURVEY

The use of electric vehicles (EVs) is increasing rapidly, and so is the need for convenient and reliable charging infrastructure. To address this need, an EV chatbot can be developed using Python that incorporates features such as showing nearby charging points, service centers, and emergency services.

The EV chatbot can leverage location-based services and machine learning algorithms to provide users with up-to-date and accurate information on the nearest charging stations, service centers, and emergency services. This will enable EV owners to plan their trips better and avoid the inconvenience of running out of charge or facing unexpected breakdowns.

III.SOFTWARE REQUIRED

- Python programming language: Python is a popular programming language that is widely used for building chatbots.

- Chatbot development framework: You can use a chatbot development framework like Flask, Tkinter, or FastAPI to build the chatbot.

- APIs: You would need to integrate APIs to fetch the necessary data for charging points, service stations, and emergency numbers. Some popular APIs for this purpose include Google Maps API, OpenChargeMap API, and MapQuest API.

- Natural Language Processing (NLP) libraries: To understand and process the user's queries and responses, you would need to use NLP libraries like NLTK or spaCy.

- Database: You would need to store the data related to charging points, service stations, and emergency numbers in a database like SQLite or MySQL.
- Deployment platform: You can deploy your chatbot on cloud platforms like AWS or Heroku for easy access and scalability.

- User interface: You can create a simple user interface using HTML, CSS, and JavaScript to allow users to interact with the chatbot. Alternatively, you can integrate the chatbot with messaging platforms like Facebook Messenger or Telegram to make it more accessible to users.

IV.SYSTEM SPECIFICATION

4.1 DESIGN MODULE

The EV chatbot using Python with features like showing nearby charging points, service centers, and emergency services will consist of several modules working together. The user interface module will handle user input validation, error handling, and chatbot design and layout. The location module will be responsible for retrieving the user's current location and detecting when the user enters or exits an EV charging station or service center. The charging points module will retrieve data on nearby charging points using an external API and display this information to the user, while the service centers module will retrieve data on nearby service centers and display this information to the user. Finally, the emergency services module will provide information about nearby emergency services and allow users to contact emergency services directly through the chatbot. These modules will work together to provide a comprehensive and user-friendly experience for EV owners.

4.2 HARDWARE

Since the "EV CHATBOT using Python" is a software-based system, there are no specific hardware requirements beyond those required to run the Python programming language itself.

4.3 SOFTWARE

1. Python: Python is the primary language used to develop the chatbot. A recent version of Python 3 should be installed on the development machine.

2. Python Libraries: The following Python libraries are essential for the development of an EV chatbot: Flask (for building the web application), Requests (for making HTTP requests to APIs), and JSON (for parsing JSON data).

V.SYSTEM MODELLING

5.1 Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent.

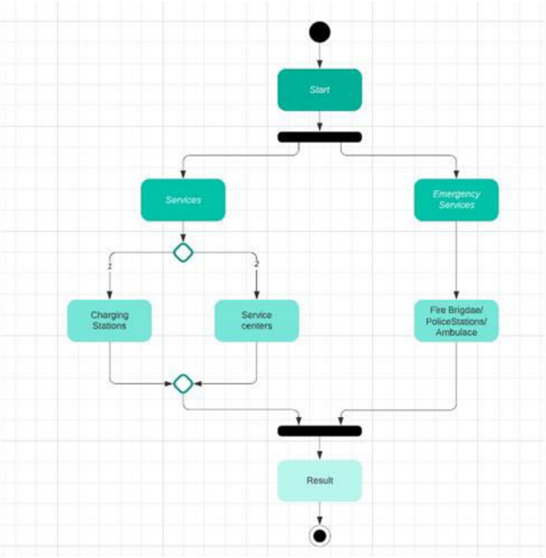


Fig. 4.1 EV CHAT BOT Activity diagram

1. APIs: External APIs will be needed to retrieve information about nearby charging points, service centers, and emergency services. The APIs used should be reliable, up-to-date, and provide the required information in a structured format.

2. Web server: A web server is required to host the chatbot application. Popular web servers for hosting Flask applications include Apache and Nginx.

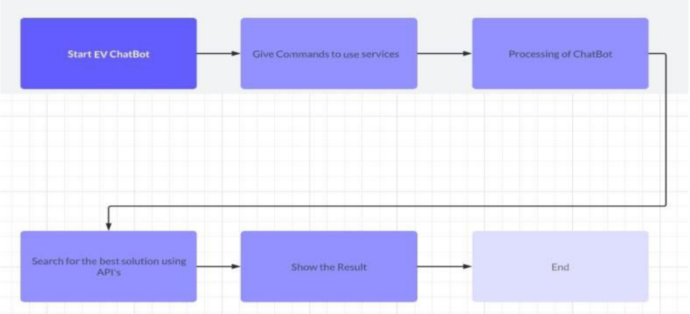
3. Database: A database may be needed to store user data, such as user preferences for charging stations or service centers. SQLite is a lightweight database that can be easily integrated into a Python application.

4. Development Environment: An Integrated Development Environment (IDE) is recommended to facilitate the development process. Popular IDEs for Python development include PyCharm and Visual Studio Code.

5. Version Control System: A version control system is recommended to manage the codebase and track changes. Git is a popular version control system used by many development teams.

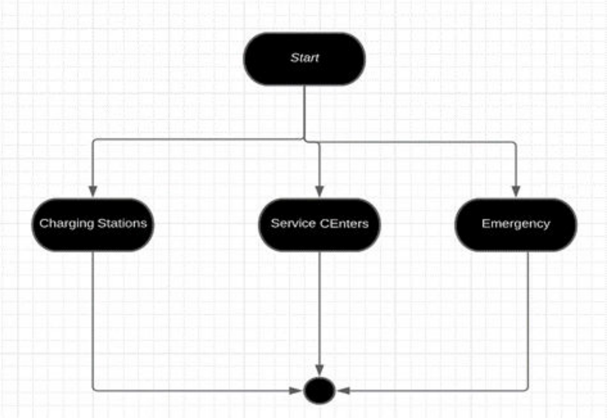
5.2 Sequence Diagram

UMLSequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when


Start EV ChatBot
Give Commands to use services
Processing of ChatBot
Search for the best solution using API's
Show the Result
End

5.3 State Diagram

Specifically, a state diagram describes the behavior of a single object in response to a series of events in a system. Sometimes it's also known as a Harrell state chart or a state machine diagram. This UML diagram models the dynamic flow of control from state to state of a particular object within a system.



6. APIs: External APIs will be needed to retrieve information about nearby charging points, service centers, and emergency services. The APIs used should be reliable, up-to-date, and provide the required information in a structured format.

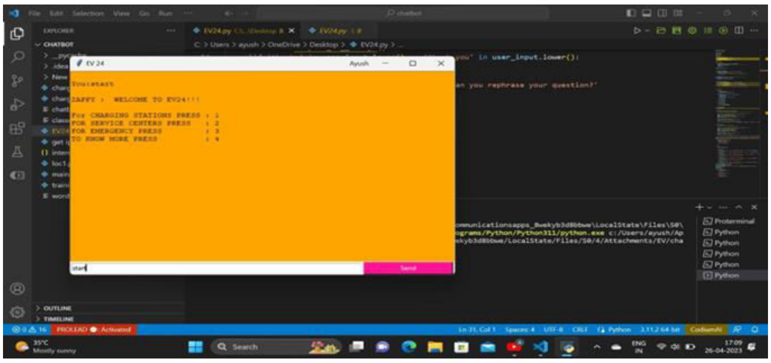
7. Web server: A web server is required to host the chatbot application. Popular web servers for hosting Flask applications include Apache and Nginx.

8. Database: A database may be needed to store user data, such as user preferences for charging stations or service centers. SQLite is a lightweight database that can be easily integrated into a Python application.

9. Development Environment: An Integrated Development Environment (IDE) is recommended to facilitate the development process. Popular IDEs for Python development include PyCharm and Visual Studio Code.

10. Version Control System: A version control system is recommended to manage the codebase and track changes. Git is a popular version control system used by many development teams.

VI.RESULTS

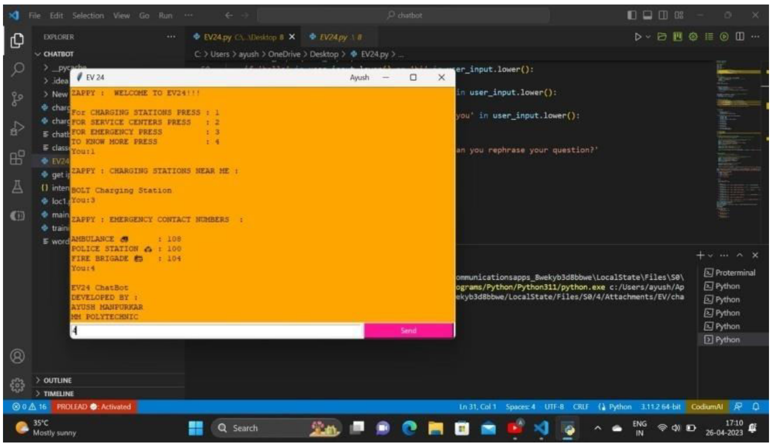


```
You:start

ZAPPY :  WELCOME TO EV24!!!

For CHARGING STATIONS PRESS : 1
FOR SERVICE CENTERS PRESS   : 2
FOR EMERGENCY PRESS         : 3
TO KNOW MORE PRESS          : 4
```

```
start
```



```
ZAPPY :  WELCOME TO EV24!!!

For CHARGING STATIONS PRESS : 1
FOR SERVICE CENTERS PRESS   : 2
FOR EMERGENCY PRESS         : 3
TO KNOW MORE PRESS          : 4
You:1

ZAPPY : CHARGING STATIONS NEAR ME :

BOLT Charging Station
You:3

ZAPPY : EMERGENCY CONTACT NUMBERS  :

AMBULANCE 🚑       : 108
POLICE STATION 🚓 : 100
FIRE BRIGADE 🚒    : 104
You:4

EV24 ChatBot
DEVELOPED BY :
AYUSH MANPURKAR
MM POLYTECHNIC
```

```
4
```

VII.CONCLUSION

Chatbots have become increasingly popular in recent years, as they offer a costeffective and efficient way to provide customer service and support. In the automotive industry, chatbots are being used to provide assistance with vehicle maintenance, repairs, and diagnostics. They are also being used to provide information on vehicle specifications, prices, and availability.Chatbots can be programmed to respond to user queries in natural language, making them more user-friendly than traditional interfaces. They can also be integrated with other technologies, such as voice assistants and machine learning algorithms, to provide more personalized and accurate responses.

VIII.FUTURE WORK

Integration with more APIs: Currently, the chatbot is using Google Maps API and OpenChargeMap API to retrieve information about charging points and service stations. However, there are many other APIs that provide similar information, such as PlugShare, ChargePoint, and EVgo. Integration with these APIs can provide more comprehensive and accurate information to users.Improved natural language processing: The chatbot could be improved by incorporating more advanced natural language processing (NLP) techniques, such as sentiment analysis, intent recognition, and context awareness. This could improve the chatbot's ability to understand and respond to user queries more accurately. Multi-lingual support: The chatbot could be expanded to support multiple languages, which would make it accessible to a wider audience. This would require the integration of additional NLP libraries and translation APIs.Personalization and user profiling: The chatbot could be improved by incorporating personalization features, such as user profiling and preferences. This would allow the chatbot to provide more customized and relevant information to users based on their location, vehicle type, and other preferences.Integration with social media platforms: The chatbot could be integrated with social media platforms such as Facebook Messenger and Twitter, which would allow users to access the chatbot from within those platforms.Voice assistant integration: The chatbot could be integrated with voice assistants such as Amazon Alexa and Google Assistant, which would allow users to access the chatbot using voice commands.Improved user interface: The chatbot could be improved by providing a more user-friendly interface, such as a mobile app or a web interface. This would make it easier for users to access and use the chatbot.Real-time updates: The chatbot could be improved by providing real-time updates on the availability of charging points and service stations. This would require the integration of additional APIs or data sources, and the implementation of real-time data processing and visualization.

REFERENCES

1. https://botpress.com/docs/getting_started/quick_start.
2. https://cloud.google.com/dialogflow/docs.
3. https://rasa.com/docs/.
4. https://www.udemy.com/course/building-chatbots-with-google-dialogflow/.
5. https://www.udemy.com/course/building-a-chatbot-with-rasa/.
6. https://www.udemy.com/course/botpress-the-ultimate-guide/.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING