Personalized Conversation Experience on Task-Based Chatbots

Tolga Arslan, BE

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Intelligent Systems)

Supervisor: Professor Vincent Wade

August 2022

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Tolga Arslan

August 19, 2022

Permission to Lend and/or Copy

Personalized Conversation Experience on Task-Based Chatbots

Tolga Arslan, Master of Science in Computer Science

University of Dublin, Trinity College, 2022

Supervisor: Professor Vincent Wade

Interest in chatbots continues to increase rapidly. The fact that chatbots work in harmony with different systems and applications has ensured the continuous development of chatbot in today's world. As a consequence of advances in artificial intelligence, machine learning, and deep learning fields, chatbots' development techniques and usage areas expanded. This develops a task-based chatbot that offers a personalized conversation experience in the context of car rental. While developing the intended chatbot, the study did not use a pre-made training dataset and prepared its own training data. Experiencing the struggles in the process of developing a chatbot that offers a personalized chat experience with the training data it has created, this study also creates adaptive chat flows in the chatbot it has developed to increase the user experience in chatbots. The study completed an evaluation survey process by recruiting participants to evaluate user experience based on user satisfaction.

The consequences of the study show that providing a personalized conversation experience on the chatbot has a positive impact on users' experience, but at the same time, the completed survey results show that, if possible, a significant portion of users would choose to complete their transactions with real people instead of chatbots. Therefore, future studies may continue the approaches presented in this study or explore new ways to enhance the user experience in chatbots.

Acknowledgments

I would like to express my sincere gratitude to Professor Vincent Wade, who supervised me throughout this dissertation.

TOLGA ARSLAN

University of Dublin, Trinity College
August 2022

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The ultimate objective of the chatbot is to provide the conversation experience that users get when communicating with an actual individual when users communicate with the chatbot. The first known chatbot in history was developed by Joseph Weizenbaum in 1965 under the name ELIZA. In the article published by Weizenbaum (1966), sample conversation flows are given for ELIZA, which is defined as a program that enables natural language conversation with a computer. The fact that chatbots, which have developed considerably with the advancement of technology, have gained a form that can communicate with the user by speaking with the user in an artificial intelligence-based structure has further increased the user experience and the level of effectiveness achieved in this field.

Chatbots are used for different purposes in various areas of the industry, from marketing to human resources, customer service, and user support areas. The fact that users can interact with chatbots 24/7 is one of the factors that enable companies to include chatbots in their systems. Chatbots have different types of working approaches. The completed study (Adamopoulou and Moussiades (2020)) presented the classification of chatbots around six different parameters, with explanations. The classification factors mentioned in the study (Adamopoulou and Moussiades (2020)) are as follows:

- Knowledge Domain is the type of classification that considers the amount of information accessible by the chatbot or the amount of data used to train the chatbot.
- The service provided classification is the type of classification that relates to the task the chatbot performs, while it relates to the intimate interaction and emotional intimacy with the user.
- Goals-based classification takes into account the primary goal that chatbots aim to achieve.

- Another classification type is the input processing and response generation method, which considers processing inputs and generating responses.
- Another type of chatbot classification is human-aid, which considers the amount in the components.
- Chatbots can also be classified depending on their development platform. (Build Method (Open-source platforms, Closed platforms))

The study (Brandtzaeg and Følstad (2017)) completed a survey that investigated people's motivations to use chatbots. Most respondents (64% of participants) cited productivity as the main reason for using chatbots among motivation categories such as Productivity, Entertainment, Social/relational, Novelty/Curiosity, and Others(for example, easier to talk about important topics with a chatbot than a real person, etc.).

1.1 Motivation

Interest in chatbots continues to grow rapidly. It is possible to come across that chatbots are included more and more in the services of companies with each passing day. In addition, it is possible to see that chatbots are increasingly embedded in electronic devices. In the report published by (Next Move Strategy Consulting (2020)), a study was conducted on the value of the chatbot market in the Banking, Financial Services and Insurance (BFSI) sector between 2019-2030. This published report states that the Global Chatbot Market in BFSI was valued at $586 million in 2019, and that this value is expected to reach $6.83 billion by 2030. All this information and more shows that the developments in this field will increase every year. The realization of a task-based chatbot experience and the approaches to be implemented in this project would be valuable in a development approach where conversational content is adaptively working by the user's chat history.

The subject of this thesis, which will be focused on by research and development, is the field of personalized chat experience in task-based chatbots. This area includes development areas that will increase the interest in chatbots today. For instance, a personalized chatbot can do the following:

- The fact that the conversation experience of the users with the chatbot is in a more personal form allows the companies to increase their efficiency in areas such as marketing and sales.
- Personalized chatbots can realize the user's intent in less time than traditional chatbots.

- A chatbot that recognizes the user and designs conversation flows accordingly provides a more engaging user experience than traditional chatbots.

1.2 Research Question and Objective

This study, "What are the challenges of developing a task-based chatbot that provides a personalized experience in the context of car rental[1]?" will work on the question. In carrying out this study, it aims to complete the following research tasks.

- This study aims to develop a personalized task-based chatbot in the context of car rental.
- In this process, the study aims to develop its own corpus for car rental.
- The research aims to measure the intent classification accuracy of the chatbot it has developed and the user experience in terms of user satisfaction with the real user test it has prepared.
- The research aims to develop a chatbot that provides an adaptive and personalized conversation experience to make the experience of task-based chatbots more effective.

1.3 Research Methodology

The study focused on a project and application development process focused on the car rental case for providing a personalized chat experience in a task-based chatbot. The study will develop the training data itself for the application it will develop. The training data will consist of groups containing vehicle information, as well as possible questions that users may be asked about car rental and general answers to them. The study designed a questionnaire for the evaluation process of the system it developed. This questionnaire presents an evaluation process scaled on a scale of 1-5. All detailed information about the survey to be completed and the recruitment processes of the participants are detailed in the section 5.3.

1.4 Outline of the Dissertation

This report is structured as follows. In Chapter 2, the study will review the literature and share information about the areas it will research/develop. Chapter 3 will focus

[1]Car rental was chosen as an application area that involves specific tasks by a dialogue.

on the design part of the project. This chapter provides the grounding of the concepts in the project before Chapter 4 Implementation. Chapter 4 Implementation explains the project's development processes and sample code blocks for some parts. Chapter 5 focuses on the evaluation approaches of the project and explains the evaluation process. The last chapter, Chapter 6, shares the study's outputs and makes suggestions for future studies.

Chapter 2

State of the Art

This section will present research and background information on completed work in chatbots. After researching chatbots, this section will focus on a literature review on task-based chatbots. After completing the literature review on task-based chatbots and personalized task-based chatbots, this section will examine the Rasa Framework and explain the concepts that will be used throughout the study. In the final section of this chapter, the study will provide an overview of the research it has completed and the information it has shared throughout the chapter.

2.1 Background

A whole set of approaches must be completed and followed to provide a conversation flow based on user interaction with the chatbot. Classifying a user-written input, correctly extracting the entities intended to be captured on the input, making an intent classification for user messages, and adapting the flow of conversation are vital areas to consider when developing a chatbot.

A chatbot pipeline is shared in the work completed by (Nimavat and Champaneria (2017)). According to this study, the chatbot first completes the basic input processing processes on the input. The study cited algorithmic approaches such as Sentence Boundary Detection and Sentence Parsing as examples for this phase. This stage is followed by the process of making sense of the input. The study showed approaches such as intent classification and entity extraction at this stage. The working chatbot pipeline has been completed with the response generation in the 3rd stage and the answer selection process in the final stage.

Thanks to advancements in Machine Learning and Natural Language Processing, chatbots are freed from the limitations of rules and pattern matching (Caldarini et al. (2022)). The research (Caldarini et al. (2022)) stated that the commercial use of chatbots, which gained flexibility in the scope of Machine Learning and Natural Language Processing, was positively affected by these developments. The research stated that the application areas of chatbots had expanded considerably with the involvement of Deep Learning in the process. The development of intelligent personal helpers such as Siri by Apple, Alexa by Amazon, Cortana by Microsoft, Google Assistant by Google, and Watson by IBM are examples of innovative application areas of chatbots. These assistants are on smartphones, watches, and even in vehicles, and they provide communication by speaking beyond typing (Caldarini et al. (2022)). These intelligent systems, which understand what users say with Natural Language Understanding, are not only personal assistants but also have many task-oriented chat flows.

2.1.1 Task Based Chatbots

Task-based chatbots are systems that shape dialogue for a specific purpose. These systems focus on a chosen context and gain a task-oriented working structure by being trained on the training data created in this field.

2.1.1.1 Architectures

Conversational agents are formed of similar architectural components. The interpretation of the messages reached is managed by the flows in which natural language comprehension processes and language models exist. Behind chatbots' input processing and response generation processes lies a series of processing processes. Systems consisting of many components such as parsers, resolvers, and debuggers constitute the architectures of chatbots.

The architecture in figure 2.1 presented in the work completed by (Xie et al. (2022)) is based on a modular design. This architecture includes a Knowledge Base (KB) / Backend action structure in addition to its components such as Natural Language Understanding, Dialogue Management, and Natural Language Generation. The Knowledge Base (KB) structure can be a database or a knowledge graph structure for processing the user's queries during the dialog. Another point to be discussed in this architecture is the Orchestrator module. While this module provides communication between the user and the system, it works in connection with other components. As seen in Figure 2.1, the message transmitted by the user first reaches the Orchestrator module. In addition to being the module that the user message reaches, this module is also responsible for transmitting the response created for the user to the user. The Natural Language Generation component is triggered by Dialogue Management (DM). This component triggers the NLG component for the following action/statement with the collaboration of Knowledge Base (KB)/Backend action and Dialogue Tree Manager.

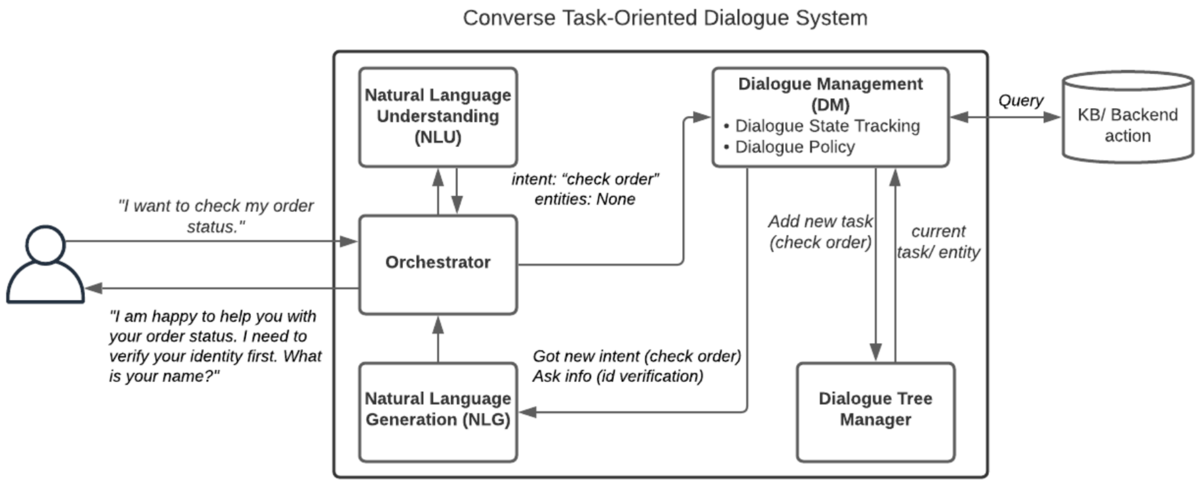

Converse Task-Oriented Dialogue System
Natural Language Understanding (NLU)
Dialogue Management (DM)
• Dialogue State Tracking
• Dialogue Policy
Query
KB/ Backend action
intent: "check order"
entities: None
"I want to check my order status."
Orchestrator
Add new task (check order)
current task/ entity
"I am happy to help you with your order status. I need to verify your identity first. What is your name?"
Got new intent (check order)
Ask info (id verification)
Natural Language Generation (NLG)
Dialogue Tree Manager

Figure 2.1: Proposed task-oriented dialog system architecture by (Xie et al. (2022))

The architecture, which is developed for the deep learning model in work completed by (Chou and Hsueh (2019)), has a generator and a discriminator. The generator is a long-short term memory model used to build dialogues, and the discriminator is a long-short term memory model used to examine and allocate rewards according to the difference between the sentences formed and the actual responses.

The architecture developed by Chou and Hsueh (2019) consists of a recurrent neural network (RNN) and long short-term memory (LSTM). Both components are used to construct sentences. The developed deep learning model improves the generated sentences with reinforcement learning (RL) and generative adversary networks (GAN). Component 1 consists of fields that allow user login. User interfaces for these areas can be given as examples. Component 2 is the area where data processing is performed. Component 3 contains a neural network. This neural network is applied to construct and define sentences. The generator is a long-term memory model for creating various dialogues. Each sentence created in the model is mapped as having an action code, a dialog manage code, and a person code. The discriminatory part consists of a long-short-term memory model for assigning low rewards for repeated sentences and high rewards for various sentences between constructed sentences and actual responses.

The completed study (Surendran et al. (2020)) suggests a task-oriented retrieval-based chatbot on the bus ticket reservation field. Figure 2.2 shows a basic architectural design of the chatbot system proposed by (Surendran et al. (2020)). The study uses Keras deep learning library with a Sequential model. The work applies a preprocessing process to the input to remove unhelpful content from the input it reaches. The work that reads the dataset with the Pandas library makes use of the Natural Language Toolkit (NLTK) and string handling functions during the preprocessing of the input. After the preprocessing is

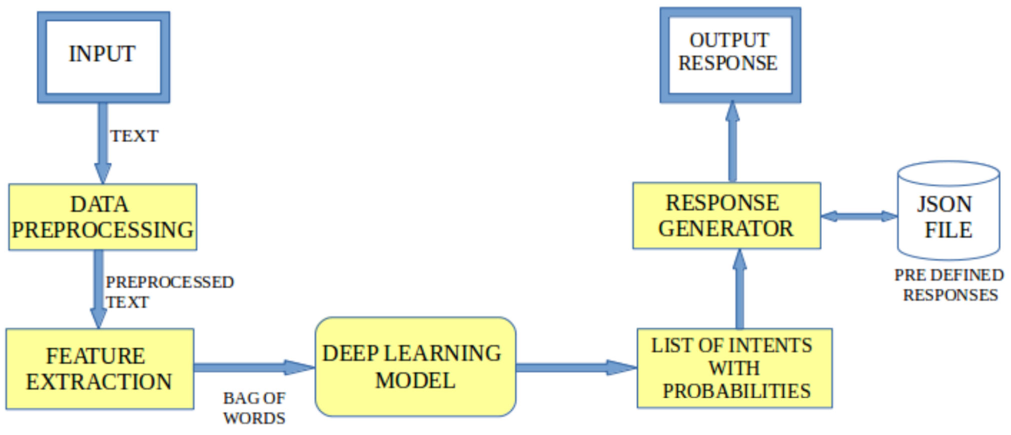


Figure 2.2: The basic architecture of the chatbot by (Surendran et al. (2020))

complete, the study uses feature generation techniques to convert the input into machine understandable form and is based on the Bag of Words (BOW) approach to generate features. In the next process, the output of the feature generator goes as input to the deep learning model. After the deep learning model, the list of goals is created with the probabilities. Intentions that are more likely than the error threshold are considered. The task of the response generation module is to select the most appropriate one for the query entered from the filtered list. The list of intents is checked against the repository (JSON file), and the response matching the input is complete.

Study	**Details (model features, components, etc.)**
(Surendran et al. (2020))	Sequential Model with following parameters: -Dense Layer -ReLu and Softmax Activition Functions -SGD Optimizer -threshold: -0.25, epochs: 200, batch size: 5
(Chou and Hsueh (2019))	Recurrent Neural Network (RNN) Long Short-Term Memory (LSTM) *Deep learning model combines: -Reinforcement Learning (RL) -Generative Adversarial Networks (GAN)
(Xie et al. (2022))	The study proposes a flexible tree-based modular task-oriented dialog system, and this proposed system uses an and-or tree structure to represent tasks. In the study, a discriminative nearest neighbor classification based on natural language inference (NLI) was used. The study set a threshold value for the inference score of $0.6 \sim 0.8$ to overcome the OOS intent issue.

Table 2.1: Summary and Comparison table for models & architectures

The table 2.1 provides a comparison of the approaches used in study (Surendran et al. (2020)) and study (Chou and Hsueh (2019)) on the basis of deep learning models. In addition, the prominent features of the studies can be seen in the same table.

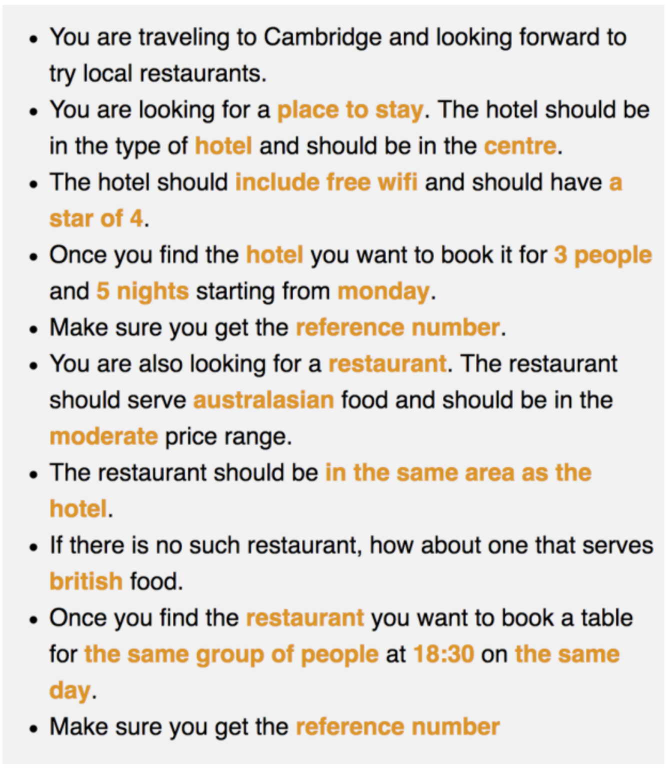
2.1.1.2 Sample Applications & Datasets

Task-Based Chatbots focus on the realization of a specific user purpose. In this respect, working with customized data sets will increase the success of the trained model created.

The in-house (Snips Natural Language Understanding benchmark dataset) open dataset described in the study published by (Coucke et al. (2018)) is a slot-filling dataset containing seven different types of intents. 7 different intent classifications consist of PlayMusic, GetWeather, BookRestaurant, AddToPlaylist, RateBook, SearchCreativeWork, SearchScreeningEvent intents. The slots associated with these intent types are as follows:

- PlayMusic intent type is associated with slot values such as playlist, artist, genre, album, year, music item, and service.
- GetWeather intent type is associated with slot values such as city, country, state, poi, time range, condition, temperature, spatial relation, and current location.
- BookRestaurant intent type is associated with slot values such as sort, party size nb, party size descr, spatial relation, city, country, state, poi, restaurant type, restaurant name, cuisine, served dish, time range, facility.
- AddToPlaylist intent type is associated with slot values such as name, artist, playlist owner, playlist, and music item.
- RateBook intent type is associated with slot values such as type, name, rating unit, best rating, rating value, select, and series.
- SearchCreativeWork intent type is associated with slot values such as type and name.
- SearchScreeningEvent intent type object, type, name, location type, location name, spatial relation is associated with slot values such as time range.

Multi-Domain Wizard-of-Oz (MultiWOZ) (Budzianowski et al. (2018)) is the corpus that contains large-scale trainable task-oriented dialog models. MultiWOZ Dialogue Corpus, which is a task-oriented corpus, has seven different domains (expanding to a total of 8 domains) (Budzianowski et al. (2018)). These domains are Attraction, Hospital, Police, Hotel, Restaurant, Taxi, and Train. Among these domains, Hotel, Restaurant, Taxi, and Train are domains that contain the Booking subtask in addition.



- You are traveling to Cambridge and looking forward to try local restaurants.
- You are looking for a **place to stay**. The hotel should be in the type of **hotel** and should be in the **centre**.
- The hotel should **include free wifi** and should have **a star of 4**.
- Once you find the **hotel** you want to book it for **3 people** and **5 nights** starting from **monday**.
- Make sure you get the **reference number**.
- You are also looking for a **restaurant**. The restaurant should serve **australasian** food and should be in the **moderate** price range.
- The restaurant should be **in the same area as the hotel**.
- If there is no such restaurant, how about one that serves **british** food.
- Once you find the **restaurant** you want to book a table for **the same group of people** at **18:30** on **the same day**.
- Make sure you get the **reference number**

Figure 2.3: A task template by (Budzianowski et al. (2018))

Action-Based Conversations Dataset (ABCD) (Chen et al. (2021)) is a dataset containing fifty-five different user intents and over ten thousand dialogues. In addition, the ABCD dataset contains 30 domains and 231 unique slots. The study proposed a new technique to capture the natural dialogue between two interlocutors. In this technique, called Expert Live Chat, conversations are carried out in real-time and continuously, while users have the opportunity to chat with each other in a free-form style. The collection of Action-Based Conversations Dataset (ABCD) using Expert Live Chat instead of templates caused various language variations to be seen in chats. The completed study (Chen et al. (2021)) also noted the difference between the fully labeled ABCD dataset and the MultiWOZ (Budzianowski et al. (2018)) dataset.

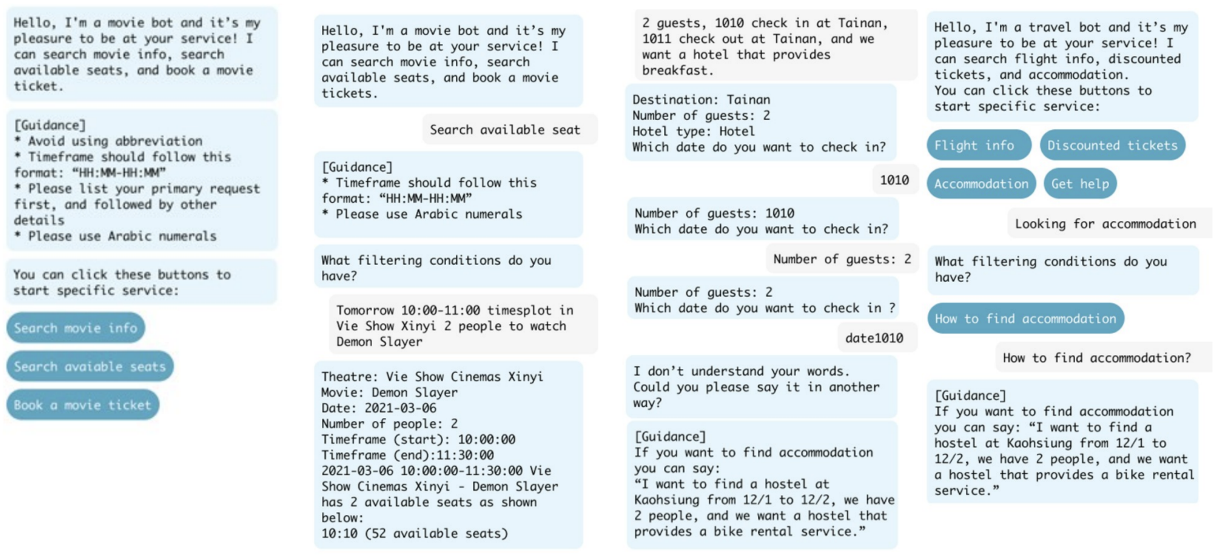
The Airline Travel Information System (ATIS) (Tur et al. (2011)) corpus contains 17 different intents. The three most commonly used intents are Flight, Airfare, and Ground_Service (Tur et al. (2011)). The research stated that this corpus was used in many studies on spoken language understanding (SLU) in dialogue systems. The research, which says that the two main tasks in the spoken language comprehension processes are intent determination and slot filling, states that the studies using the ATIS test set reported an error rate of less than 5%.

Dataset Name	Details
Multi-Domain Wizard-of-Oz (MultiWOZ) (Budzianowski et al. (2018))	It has 7 domains and 10, 438 dialogues (3,406 single-domain dialogues and 7,032 multi-domain dialogues)
Action-Based Conversations Dataset (ABCD) (Chen et al. (2021))	It has 30 domains, 231 slots, 55 intents, over 10K dialogues
The in-house (Snips Natural Language Understanding benchmark dataset) open dataset (Coucke et al. (2018))	It has 7 intents and each intent contains an approximately equal number of utterances.
Airline Travel Information System (ATIS) (Tur et al. (2011))	It has 17 different intents, and the Flight intent has about 70% density in training and test sets. As used in the study(Tur et al. (2011)), the training data consists of 4,978 utterances, and the test data consists of 893 utterances.
Developed corpus in this study	It operates in the context of car rental and it has 13 slots, 13 entities, 20 intents, 357 example user input data

Table 2.2: Summary and Comparison[1]table for datasets

2.1.1.3 Dialogue Examples

Task-Based Chatbots aim to provide an effective speaking experience with customized training for different domains that they are developed to perform a specific task. Below are sample conversation streams shared in the completed work by (Yeh et al. (2022)).



Hello, I'm a movie bot and it's my pleasure to be at your service! I can search movie info, search available seats, and book a movie ticket.

[Guidance]
* Avoid using abbreviation
* Timeframe should follow this format: "HH:MM-HH:MM"
* Please list your primary request first, and followed by other details
* Please use Arabic numerals

You can click these buttons to start specific service:

Hello, I'm a movie bot and it's my pleasure to be at your service! I can search movie info, search available seats, and book a movie tickets.

Search available seat

[Guidance]
* Timeframe should follow this format: "HH:MM-HH:MM"
* Please use Arabic numerals

What filtering conditions do you have?

Tomorrow 10:00-11:00 timesplot in Vie Show Xinyi 2 people to watch Demon Slayer

Theatre: Vie Show Cinemas Xinyi
Movie: Demon Slayer
Date: 2021-03-06
Number of people: 2
Timeframe (start): 10:00:00
Timeframe (end):11:30:00
2021-03-06 10:00:00-11:30:00 Vie Show Cinemas Xinyi - Demon Slayer has 2 available seats as shown below:
10:10 (52 available seats)

2 guests, 1010 check in at Tainan, 1011 check out at Tainan, and we want a hotel that provides breakfast.

Destination: Tainan
Number of guests: 2
Hotel type: Hotel
Which date do you want to check in?

1010

Number of guests: 1010
Which date do you want to check in?

Number of guests: 2

Number of guests: 2
Which date do you want to check in ?

date1010

I don't understand your words. Could you please say it in another way?

[Guidance]
If you want to find accommodation you can say:
"I want to find a hostel at Kaohsiung from 12/1 to 12/2, we have 2 people, and we want a hostel that provides a bike rental service."

Hello, I'm a travel bot and it's my pleasure to be at your service! I can search flight info, discounted tickets, and accommodation.
You can click these buttons to start specific service:

Looking for accommodation

What filtering conditions do you have?

How to find accommodation?

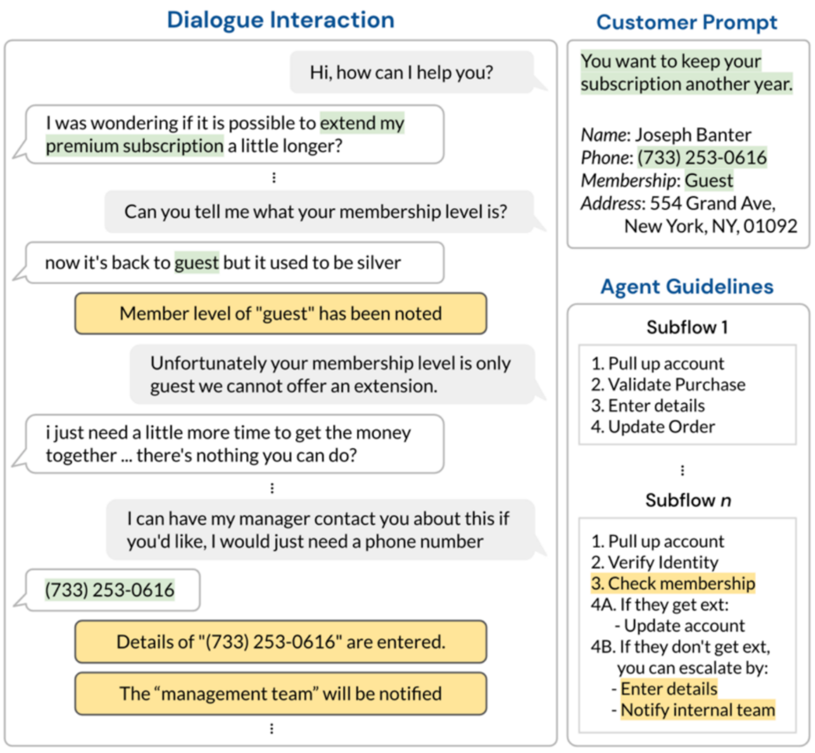
[Guidance]
If you want to find accommodation you can say: "I want to find a hostel at Kaohsiung from 12/1 to 12/2, we have 2 people, and we want a hostel that provides a bike rental service."

Figure 2.4: Sample Dialogues (Yeh et al. (2022))

The work completed by (Yeh et al. (2022)) presented chatbots in the context of travel

[1]Features specified for some datasets may not be specified in all datasets added to the table. This is dependent on references and the availability of accessible information. This can make comparisons between datasets somewhat difficult.

arrangement and movie booking on IBM Watson. This study conducted on 4 different guidance timings (Service-onboarding, Task-intro, After-failure, Upon-request) and 2 guidance types (Example-based, Rule-based). The figure 2.4 shows the sample dialogues presented in this study, the first two dialogue flows take place in the context of the movie and the last two dialogue flows in the context of travel. While the first dialogue creates an example for rule-based guidance(the user sees a message listed with items about the format) and service-onboarding timing(the user gets guidance with a short introduction at the beginning) in the context of the movie, the second dialogue flow is again in the context of the movie with the rule-based guidance and task- It is an example for intro timing (user sees a guide message at the beginning of the task). the third and fourth dialogues take place in the context of travel as stated. The third dialog flow is an example for example-based guidance (the user sees a sample input example) and after-failure timing (the chatbot sees guidance for the correct input when it cannot determine the intent of the user-specified message), while the fourth dialog flow is example-based guidance and It is an example for upon-request timing (user sees guide after request).



Hi, how can I help you?

I was wondering if it is possible to extend my premium subscription a little longer?

⋮

Can you tell me what your membership level is?

now it's back to guest but it used to be silver

Member level of "guest" has been noted

Unfortunately your membership level is only guest we cannot offer an extension.

i just need a little more time to get the money together ... there's nothing you can do?

⋮

I can have my manager contact you about this if you'd like, I would just need a phone number

(733) 253-0616

Details of "(733) 253-0616" are entered.

The "management team" will be notified

⋮

You want to keep your subscription another year.

Name: Joseph Banter
Phone: (733) 253-0616
Membership: Guest
Address: 554 Grand Ave, New York, NY, 01092

Subflow 1

1. Pull up account
2. Validate Purchase
3. Enter details
4. Update Order

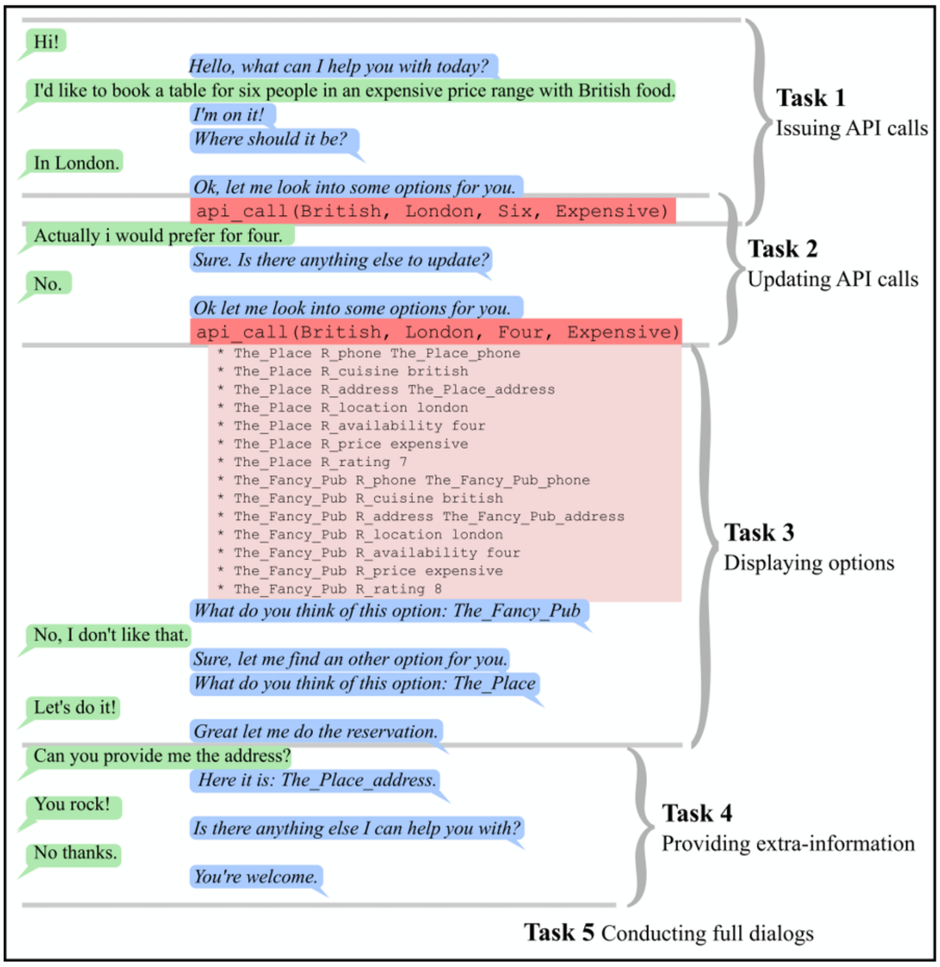
⋮

Subflow *n*

1. Pull up account
2. Verify Identity
3. Check membership
4A. If they get ext:
 - Update account
4B. If they don't get ext, you can escalate by:
 - Enter details
 - Notify internal team

Figure 2.5: Sample interaction (Chen et al. (2021))

It is seen that the dialogue in figure 2.5 of Action-Based Conversations Dataset (ABCD) suggested by (Chen et al. (2021)) to create task-oriented dialogue systems continues in accordance with the flow in the agent guide. In addition, the meaningful data specified by the user during the conversation can also be seen as highlighted.



Hi!

Hello, what can I help you with today?

I'd like to book a table for six people in an expensive price range with British food.

I'm on it!

Where should it be?

In London.

Ok, let me look into some options for you.

```
api_call(British, London, Six, Expensive)
```

Actually i would prefer for four.

Sure. Is there anything else to update?

No.

Ok let me look into some options for you.

```
api_call(British, London, Four, Expensive)
* The_Place R_phone The_Place_phone
* The_Place R_cuisine british
* The_Place R_address The_Place_address
* The_Place R_location london
* The_Place R_availability four
* The_Place R_price expensive
* The_Place R_rating 7
* The_Fancy_Pub R_phone The_Fancy_Pub_phone
* The_Fancy_Pub R_cuisine british
* The_Fancy_Pub R_address The_Fancy_Pub_address
* The_Fancy_Pub R_location london
* The_Fancy_Pub R_availability four
* The_Fancy_Pub R_price expensive
* The_Fancy_Pub R_rating 8
```

What do you think of this option: The_Fancy_Pub

No, I don't like that.

Sure, let me find an other option for you.

What do you think of this option: The_Place

Let's do it!

Great let me do the reservation.

Can you provide me the address?

Here it is: The_Place_address.

You rock!

Is there anything else I can help you with?

No thanks.

You're welcome.

Task 1 Issuing API calls
Task 2 Updating API calls
Task 3 Displaying options
Task 4 Providing extra-information
Task 5 Conducting full dialogs

Figure 2.6: Goal-oriented Conversation Flow by (Bordes et al. (2017))

Figure 2.6 published in the work completed by (Bordes et al. (2017)) contains a dialogue flow about booking a table in the restaurant. Green areas indicate the user's message, while blue areas indicate the bot's messages. The red areas on the figure represent API calls shaped according to the user's messages. As you can see, the API calls are updated according to the messages of the user during the chat. In addition, the work presented the dialogue by partitioning it on the basis of tasks.

Study	Propesed Works
(Yeh et al. (2022))	Investigates which combination of two guide types and four timings provides a better experience and user performance. The findings led to a set of design proposals for task-oriented chatbots. Additionally, the study encouraged future researchers to test the efficacy of proposed designs in real-life scenarios.
(Chen et al. (2021))	Introduces the Action-Based Conversations Dataset (ABCD), which is a fully labeled dataset. Additionally, The study proposed additional dialog tasks. -Action State Tracking, which aims to determine the proper intent by analyzing customer statements. -Cascading Dialogue Success is proposed to measure the model's capability to comprehend actions in context.
(Bordes et al. (2017))	The study proposes a test environment for end-to-end dialog systems in goal-oriented applications. The study evaluated various methods (end-to-end Memory networks, rule-based systems, and others*) on goal-oriented conversation tasks. The study shared the test results obtained in all the tasks and methods it applied and stated that although some effects of the tried methods were seen, there are still many unresolved challenges.

Table 2.3: Comparison and Summary table for example studies.(*classical information retrieval methods, and supervised embeddings)

2.1.1.4 Fundamental Processes

As explained in the study completed by (Caldarini et al. (2022)), the inclusion of natural language processing and machine learning in the development processes of chatbots has expanded the usage areas of chatbots. The inclusion of Deep Learning algorithms in the process has further expanded the existing structure.

Conversational agents need natural language processing and natural language understanding capabilities in order to maintain mutual dialogue (in an intelligent manner) with users. (Coucke et al. (2018)) presents a natural language comprehension pipeline. The exported pipeline Snips belongs to NLU and contains the main component "NLU engine". The first component is the intent parser, this component handles intent classification and slot filling. The last step of the NLU pipeline is the resolution of the slot values. The extracted asset values are mapped to the corresponding built-in assets. After the slot values are filled in, the process of determining the intent and extracting the meaningful parts is completed.

When the user request is understood, necessary actions and information retrieval processes are performed (Kucherbaev et al. (2018)). For example API calls etc. operations are performed at this stage. As a result of the completion of the information retrieval processes, the response generation component generates a response. The dialog management component is the part (Kucherbaev et al. (2018)) where parts of the conversation (defined or missing entities, may be current intent) are kept and updated. This component can also perform tasks such as processing user comments, asking follow-up questions, requesting missing information.

Task-Based Chatbots keep the conversation going so that a specific task can be completed most effectively. At this point, it is vital to model the chatbot by training it with appropriate dialogues for a particular task. The study (Surendran et al. (2020)) described the data as one of the main keys and stated that the data set is crucial for the chatbot to provide an effective conversation.

Study (Tur et al. (2011)) stated that the two main tasks in Spoken Language Understanding (SLU) are intent setting and slot filling. In fact, the continuation of the interaction of conversational agents with the user is also related to the ability to understand the language used. In this respect, natural language understanding capabilities are one of the critical processes of a conversational agent.

Based on all these points, in the process of conversational agent development, after obtaining the appropriate training data, the process from successfully interpreting the user's answers to providing the relevant answer, Natural Language Understanding, Dialogue Management Component (can be used with different names and connected additional components), Response Generation, Natural It enables Language Generation processes to be characterized as basic processes.

2.1.2 Adaptive/Personalised Task Based Chatbots

The study, completed by (Shumanov and Johnson (2020)) focuses on more personalization of conversations with chatbots. It was stated that more than 57,000 chatbot interactions were used in this research and focused on the effect of the language used on personalization. The study shows that chatbots can be manipulated through response language. The study also stated that chatbots and user personality fit also benefit companies. Chatbots (intelligent agents) that are built with self-feeding models that learn from past conversation flows aim to get to know the user, rather than follow predefined rules and flows that general answer questions from users, use natural language processing and machine learning approaches. An effective definition of personality is essential for creating a personalized chatbot experience. The research (Shumanov and Johnson (2020)) focused on the Big Five(Agreeableness, Conscientiousness, Extraversion, Neuroticism, Openness) model in this regard. The researchers noted that a panel of nine psychology and language experts was used to verify that chatbot responses used words and language compatible with introversion and extroversion. The research also stated that an excited response would be created in response to the message of a user with an extroverted personality type. At this point, a personal chat experience is based on personality types.

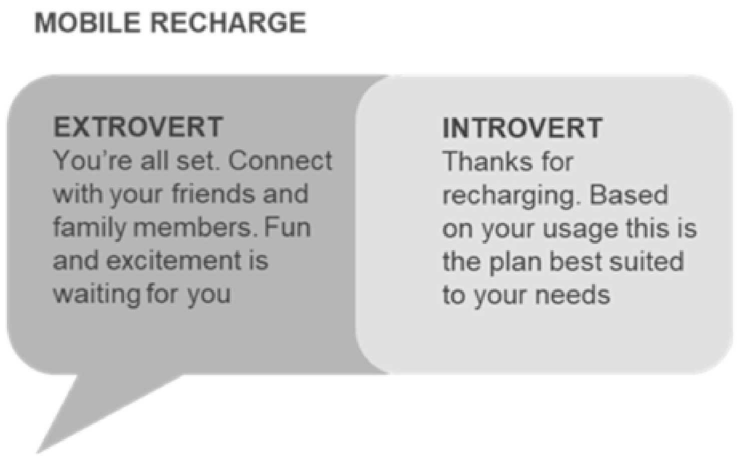

MOBILE RECHARGE
EXTROVERT
You're all set. Connect with your friends and family members. Fun and excitement is waiting for you
INTROVERT
Thanks for recharging. Based on your usage this is the plan best suited to your needs

Figure 2.7: Sample Responses for an extroverted and introverted user (Shumanov and Johnson (2020))

The figure 2.7 was included in the report of the study completed by (Shumanov and Johnson (2020)) as an example of the influence of personality type on responses. This study states that pairing a chatbot with a personality type compatible with customers with a customer will increase user-chatbot interaction. The study also stated that this situation produces results that may be beneficial for companies financially. Another approach considered in this study is that people feel close to themselves while interacting with other people; that is, they maintain more effective communication with people with similar personality types. The study was inspired by this aspect and stated that people are more likely to communicate with chatbots with similar personalities. This research, which indicates the importance of the personality factor in human-machine communication, states that the social gain from chatbots can be increased with the developed method. The study also states that the findings obtained as a result of the research it has completed can improve marketing activities, improve communication results, allow an increase in product and service purchases, and have positive effects.

An adaptive chat experience can be achieved through the language of the chatbot or by giving the chatbot a personality, as well as by using the user models created by processing the feedback from the users in chat. In the study completed by (Ikemoto et al. (2018)), the suggestions presented to the users are created as a result of the previous choices of the users. The operation is based on updating a user vector with an initial value of (0,0,...,0) with the user's selections. All values being zero indicates that the user's preference is unknown. The study stated that the user vector was updated by being triggered by the answers to the questions asked to the user about the property and the user's responses to the chatbot's suggestions. For example, in the first case, the user's "YES" value is expressed with a positive value, while the "NO" value is expressed with a negative value. If the other option is "NOT SURE" in the possible answers and the user selects this value, the zero value is used for the user vector.

The study completed by (Ikemoto et al. (2018)) stated that when the similarity of an item is more than the recommendation threshold, it is presented to the user as a recommendation. Of course, at this point, the item with the highest similarity is presented as a recommendation. The recommendation threshold (alpha) determines whether the recommendation is fulfilled, while higher values of the alpha value indicate that the recommendation is made when an item is found that matches the user's preference. In this case, the alpha value will be set by administering questions to the user to determine the first alpha value.

(Amato et al. (2019)) presents a chatbot designed in the field of Cultural Heritage Learning in order to enrich the learning experience of students. The introduced chatbot aimed to keep the content rich by providing adaptive content. This study, which is shaped by the conversation flow with the user and provides new content, also states that each user can create unique chat experiences.

Instead of focusing on modeling messages and responses, research (Al-Rfou et al. (2016)) applied modeling in a long context and a structure that would include the participant's background. Expecting the model to make better predictions for the next message as the conversation continues and the context grows, the study examines how dialogue modeling is affected by the past within the conversation and the histories of the participants' conversations. This research, which offers a model that can personalize users' predictions based on their opinions, interests, experiences, and writing or speaking style, used Reddit conversations in his study. Stating that characterizing users, language, discourse coherence, and response diversity requires very large datasets and large models, the study noted that users' responses are modeled on long histories of contributions over the years in various subforums and discussions.

The study (Al-Rfou et al. (2016)) stated that the improvement obtained by adding the author (which may include personal history, interests, ideas, demographics, writing style, and personality traits) and/or context (conversational history) to the base model resulted in better performance than any of the features used alone. At this point, it can be seen that the information recovered from each feature is different. It was also stated in the study that using the author feature compared to the conversation history (context) feature provides greater gains.

The study (Elsholz et al. (2019)) indicated changing the language style as an important point in the personalization of chatbots and continued its research on adding a language style to e-commerce chatbots to improve user satisfaction, perceived product value, user interest in a product and user interaction. A test process with a total of 169 participants was applied to two chatbots that communicate in English and a Shakespearean dialect. The evaluation results indicate that the chatbot with the Modern English language provides greater user satisfaction (easy to use), while the chatbot with the Shakespearean dialect shows a higher value in terms of user engagement and perceived product value. Users rated the chatbot, which communicates in a Shakespearean dialect, as more entertaining. The study also states that as a result of the research it has obtained, it is possible to understand a finding that in some e-commerce environments, it may be beneficial to accept a loss in user satisfaction to increase perceived product value and user engagement. Based on all these situations, it is stated in the study that it would be promising to conduct more studies on applications where automatic style transfer (adaptation to users' language styles) can be realized.

The paper (Vladova et al. (2019)), which indicates the approach to the personalization of online courses as a growing trend in the education domain, mentions the difficulty of collecting the necessary personal information and recommends an education chatbot that acts as a teacher with natural language. The chatbot proposed in the study, which uses chatbot technology to offer a dialog-based system that teaches students in a personalized way, is associated with a database that collects, stores, and updates personal information and data from previous learning processes during informal conversations. Presenting the concept of "Learning Avatar" with the ability to construct a cognitive model of the user based on user-specific data, the study states that Avatar represents the student's knowledge and characteristics, including socio-demographic data, personality, cognitive model, mental models related to current learning, and culture. The study stated that the project was in the beginning stage, and Avatar, which is updated each time the user interacts with the chatbot, was planned as a kind of "mental model" prepared for the individual.

2.1.3 Persona Based Personalization

The study, completed by (Ma et al. (2021)), characterizes personalized chatbots as chatbots with a consistent personality. This study stated that the current approaches in this field are not practical in terms of real applications and supported this statement with ideas such as the limited predefined user-profiles and devoid of self-renewal in the process. The study focused on automatically teaching user profiles using large-scale user dialog histories to avoid this limitation. In the study, a personalized language model was trained to create a general user profile using the user's past responses. The study proposed the DHAP model. The study defined the concept of DHAP as Automatically learning user profiles from Dialogue History for personalized chatbots.

In the DHAP model proposed by (Ma et al. (2021)), the user's past reactions represent the user's overall profile. This approach includes a personalized decoder. This decoder fuses the learned user profile with the response generation process. The method proposed in the study automatically learns the implicit user profile from the user's past dialog information. In the approach, two types of user profiles are created, the general user profile and the post-aware user profile.

The study (Li et al. (2016)) studied persona-based models and defined a persona. The study, which states that the defined persona can be seen as a combination of the user profile, language behaviors and interaction style, also stated that the persona is adaptive. Adaptive capability provides a different approach for different interlocutors.

The work completed by (Zhang et al. (2018)) aims to transform chit-chat into a more interesting form. The study aimed to achieve this aim by conditioning it according to profile information, and by introducing the concept called profile with a permanent personality consisting of textual explanation sentences, it was stated that the profile could be used to produce more personal, consistent, specific and interesting responses than an impersonal model. Although the trained models are trained to ask and answer questions about personal issues, it is stated in this study that a model of the speaking partner's personality can be created after the dialogue obtained.

The research (Zhang et al. (2018)) presents the PERSONA-CHAT dataset (a new dialog dataset of more than one hundred and sixty-four thousand expressions collected through Amazon Mechanical Turk), conveys that conditioning the agent with personal information provides a better prediction for the following dialog phrase. The study cited that human evaluations indicated that the dataset presented in the study provides more compelling models that can be fluid and consistent. The study also expressed its intention that the findings obtained should be helpful for educational agents who can ask questions about user profiles, remember answers, and use them naturally in conversation.

2.2 Rasa Framework

This study uses the open-source Rasa framework (Rasa Version:3.1.0 - Rasa SDK Version:3.1.1) to develop a task-based chatbot. Rasa is an open-source machine learning framework (Rasa Open Source Documentation (2022a)) that allows the generation of conversations in human-computer interaction. Rasa Open Source uses the YAML format, which is defined as the data serialization language. The following sections it is aimed to have information about the Rasa Framework by sharing information about the Rasa Framework based on the (Rasa Open Source Documentation (2022a)) source before moving on to the application part of this report.

2.2.1 Architecture

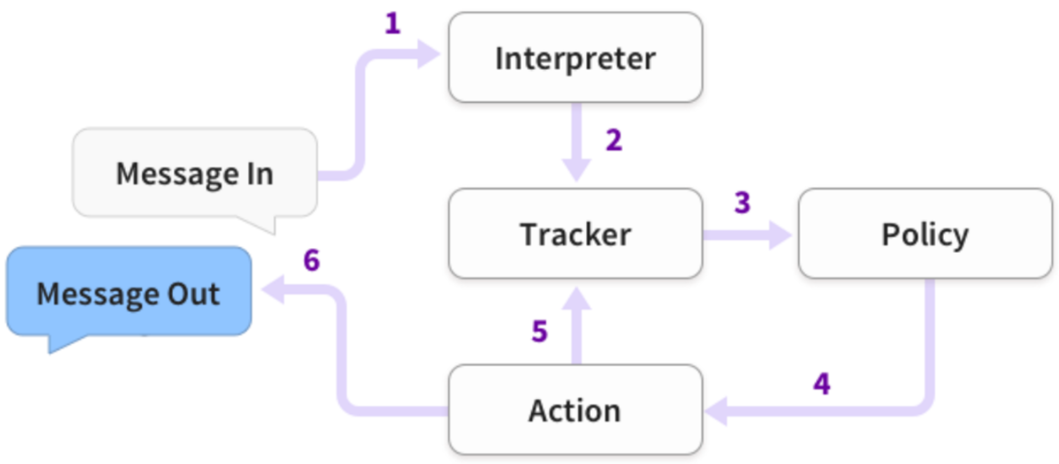


Figure 2.8: Rasa Open Source Architecture (Rasa Open Source Documentation (2022b))

Figure 2.8 shows the flow of Rasa showing the basic steps taken during the chatbot's response to a message written by the user. The received message is forwarded to the Interpreter and converted by the NLU into a dictionary containing the original text, intent, and all the entities contained in the message. In the next step, the tracker monitors the conversation status and ensures that the process passes to the policy with the conversation status. Each policy determines which action will take place next. The decided action is logged on the Tracker and the generated response is returned in response to the user's message.

Rasa Open Source architecture is designed on the basis of two basic components as Natural Language Understanding (NLU) and dialog management (Rasa Open Source Documentation (2022c)).

The NLU is the part that handles intent classification, entity extraction, and response retrieval, shown as the NLU Pipeline, as it processes user conversations using an NLU model created by the trained pipeline. NLU Pipeline is the part where user messages are processed. This process performs intent classification, entity extraction, and response retrieval processes with a trained NLU model. Dialogue Policies are the area where the following action is decided in the conversation flow.

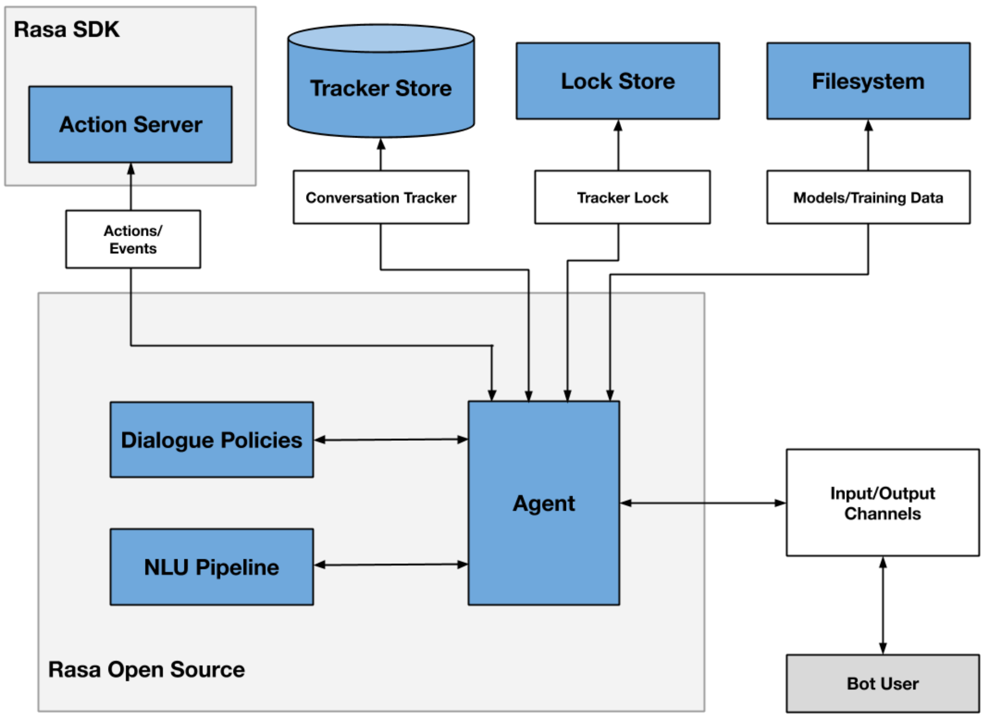


Figure 2.9: Rasa Architecture (Rasa Open Source Documentation (2022c))

Tracker Stores, referred to in figure 2.9 refers to the repository where conversations are stored by the chatbot. Rasa architecture supports different repository types. Necessary adjustments for repository types can be made in the "endpoints.yml" file. This build uses InMemoryTrackerStore (conversation history is stored in memory) by default but can be customized for SQLTrackerStore, RedisTrackerStore, MongoTrackerStore, DynamoTrackerStore, and Custom Tracker Store.

Event Brokers enable the chatbot to connect to other services, enabling the processing of data. This build can be used with Pika Event Broker, Kafka Event Broker, and SQL Event Broker.

After the chatbot training is completed, storing the created model and keeping the chatbot ready for use is one of the stages in the pipeline. Rasa can load the model from the local disk to rerun the model, upload the model via HTTP server or load the model from cloud storage.

A call locking mechanism is used to ensure that the reached messages are processed in the correct order. Conversations are locked during this process. InMemoryLockStore is used as the default lock store.

Rasa software development kit is a python running SDK created to run custom actions. The Rasa action server runs the custom actions so that the specified action flows occur at the custom action level. Rasa Actions is described in section 2.2.6. Expressing the conversations as a sequence of events in Rasa is effective in determining the flow of the conversation by triggering the appropriate actions from the action server (returning the events in these areas).

2.2.2 Training Structure

Rasa Open Source structures training data as Natural Language Understanding (NLU), Stories, and Rules. In addition to these, Domain structure will be examined in the next section (section 2.2.3).

2.2.2.1 Natural Language Understanding (NLU)

Making sense of user messages is one of the main points for the functioning of the chatbot. Rasa Open Source NLU contains training data created based on user intentions. Training data generated based on user intentions may contain attributes assigned to certain entities. As used in this study, the NLU section can perform a more comprehensive intent classification by including Regex formulas and creating particular definitions to capture synonymous entries.

While diversifying the training data, defining the specific words needed as entity values with special explanations allows these words to be captured from the user input. These values can be processed for later use in Slots and Actions. While the entities are specified on the training data, they can be captured at a more detailed level by using roles. Creating the training data as diverse as possible is one of the main factors that ensure an efficient training model.

2.2.2.2 Stories

Stories are the field where possible conversations of the user and the chatbot are represented in the training data. Conversations can be routed based on entity values and slot values. While designing a sample speech, the real-life flow is applied in this structure. The user input (intent) and the chatbot's response (action) follow each other throughout a conversation, and the conversation continues. The response specified on the action can be a response on the predefined domain, or it can be a response from the defined Custom Action. In story design, structures such as "control points" and "or" structures can be used to make the design more effective. These structures contribute to obtaining shorter stories.

2.2.2.3 Rules

Rules are the fixed pillars of the conversation. Rule structures are used when the same flow is always desired. Stories and rules should be used together in harmony, as the use of rules alone is not enough to sustain a conversation from start to finish. When rule structures (RulePolicy) are specified under Policies, the chatbot recognizes this structure. If the rule to be written is desired to be realized under certain conditions, the ”condition” section should be added to the structure, and the necessary conditions must be specified in this field before starting to write the rule. For example, the existing-user value can be specified as ”true” as a condition. If the user is a known user, that is, if the user has already completed their first registration, the answers to be shown to the user and the conversation flow to be applied may be different.

2.2.3 Domain

Domain structure includes intents, entities, slots, forms, chatbot responses, and actions. Intents defined in the NLU structure are introduced to the chatbot in this structure. The definition of slot values and assets is carried out under the domain, and the detailed description of slot values is also specified in this area in the structures required to fill in their values.

2.2.4 Components

Components specify the operations to be performed on the NLU line. This structure can be established by defining particular components as well as using featurizers, intent classifiers, and entity extractors as components. When pre-trained language models are to be used, these models should be defined on the ”pipeline” and used. This study uses the ”SpacyNLP” language model in the ”pipeline” of the chatbot it offers. This is explained in detail in section 4.2.5.1. Below is brief information about other components of Rasa Open Source.

- Tokenizers, another of the Components, parse the input into tokens.
- Another component is the Featurizer. Featurizers can output two different types of properties, sequence properties, and sentence properties, by creating a vector representation of the user message and response. Rasa Open Source supports different kinds of Featurizers, providing options to complete the creation of features for asset extraction, intent classification, and response classification through these builds.

- Intent classifiers ensure that the user message matches the appropriate one of the predefined intents. The Dual Intent Entity Transformer (DIET) can be used for both intent classification and entity extraction, while intent classifiers usually return "intent" as expected.

- Entity extractors are responsible for extracting entities from user messages. On the other hand, EntitySynonymMapper enables synonyms of entity values to be recognized and extracted on user inputs.

2.2.5 Policies

The Rasa framework uses policies to determine what action to take at each step during the dialogue. The developed chatbot will predict the next action according to the selected principles. Policies should be included in the project by paying attention to the needs and requirements of the chatbot. Below are sample policies and brief information about them.

- The Transformer Embedding Dialogue Policy architecture is a multitasking architecture for action prediction and entity identification to be applied after the conversation.

- The UnexpecTED Intent Policy is offered as an experimental option at the time of this project. No action will be triggered if the intent predicted by the NLU is actually likely to happen, but otherwise UnexpectTEDIntentPolicy will trigger action_unlikely_intent.

- The Memoization Policy checks if the stories in the story.yml file match the conversation to predict the next action.

- The AugmentedMemoizationPolicy works like MemoizationPolicy. Additionally, it forgets a certain number of conversation history steps and works to match stories with conversation history whic is reduced.

- The RulePolicy makes predictions based on rules created in the training data.

2.2.6 Actions

During the conversation process with the chatbot, each user input triggers an action to take place. Actions to be returned as a response to the user can be a response expression defined under the Domain, or they can be returned to the user through an action defined as a Custom Action. Custom Actions also support additional operations such as instant database queries, API calls, or sending data to the database.

2.3 Summary

In this section, the literature review on task-based chatbots is completed. This chapter then focused on personalized task-based chatbots and focused the literature review on this specific area. In the final part of the chapter, information about the Rasa framework is given.

Chapter 3

Methods & Design

This chapter examines how the project implemented and developed in this thesis was designed and the approaches implemented. This part includes examples from training data and aims to provide a clearer understanding of the approaches applied during the project development.

3.1 Project Goal Overview

This project aims to develop a task-based chatbot in the context of car rental. The chatbot aims to provide customized adaptive replies based on chat progress and conversation history.

3.2 Overview of Approach

The beginning of the project is based on the determination of suitable libraries, frameworks, and development environments and requirements. This process determines the training data on which the application will be established. In this study, ready-made data sets were not used. The study created its own training dataset. The study created its own answers and data for frequently asked questions about car rental prices and the car rental process in practice, using the research done on the web.

3.3 Architecture Design

The architectural design of the chatbot is based on the architecture of the Rasa framework in figure 2.9 and 2.8. Rasa Open Source/Agent layer is the part where NLU operations take place. Detailed explanation of this part is provided in section 2.9. Rasa SDK layer is

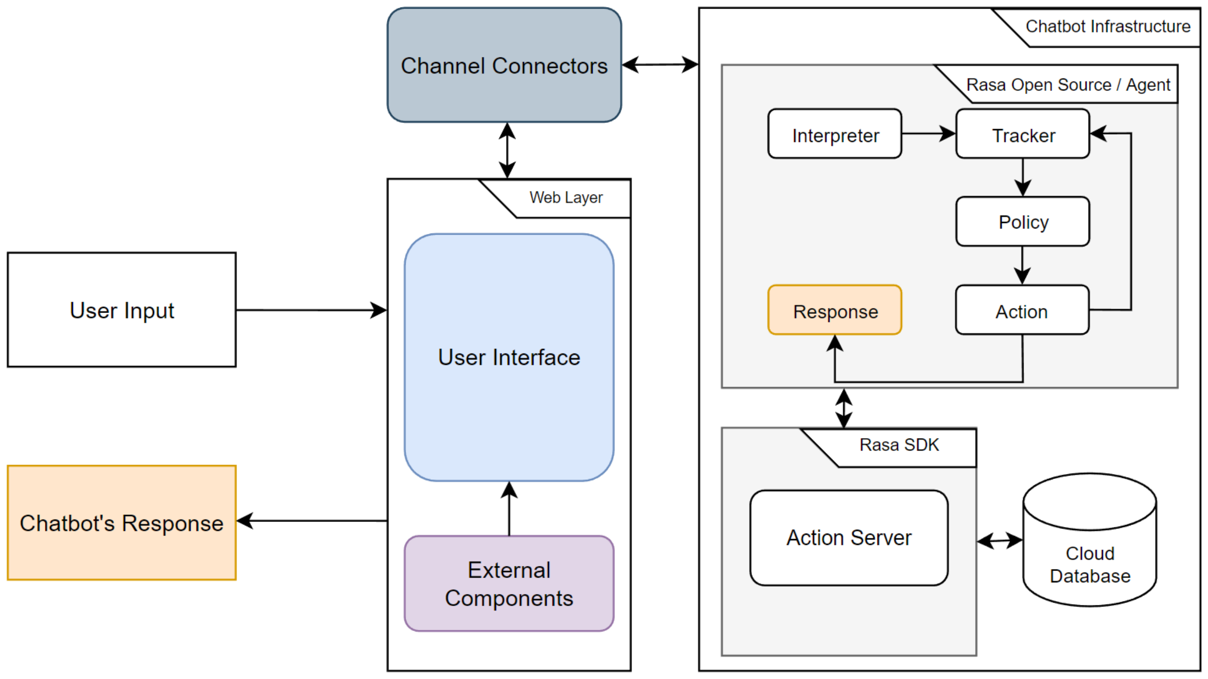


Figure 3.1: Architecture

the part that executes actions with Action Server. In addition, this part is the structure that connects the cloud database and the whole system. The user interface and the chatbot system are linked by channel connectors. The component called External Component is the structure(for example, a framework) used in the creation of the web interface. These parts are explained in section 4.4.

3.4 Generating Training Data

A total of 20 different intent classes, 13 different entity values, 13 different slot values, 357 example user input data, 39 response statements, and four different synonym mappings were used in the project. The project, which includes 11 Different Custom Actions, also has 20 different vehicles in its cloud database. In the following subsections, the design processes of the training data are explained in more detail.

3.4.1 Generating NLU Data

NLU data is the part where user inputs are actually introduced/taught to the chatbot. While defining the training data here, it has been detailed with entities so that the meaning to be obtained from user inputs has been transformed into a more effective form. For example, the user's question in the table 3.1 does not contain any entity or slot values.

Therefore, the intent class for this question in the nlu.yml file only contains different ways of asking the same question.

Sentence	**Slot/Entity**	**Value**
At the end of the rental, should I deliver the vehicle with a full fuel tank?	-	-

Table 3.1: Example training data without any slot/entity

The following example is for forms where the user input contains some kind of entity/slot value. The user can specify the type of the vehicle he/she wants to rent, in this case the specified vehicle type is recognized and the asset values are filled.

Sentence	**Slot/Entity**	**Value**
I want to rent a [standard]{"entity":"vehicle_type"} type car.	vehicle_type	standard

Table 3.2: Example training data with one slot/entity

The following example shows a user input containing two different types of entities. By adding this input to the training data, what the chatbot should do when it encounters such input was determined during the development process of the chatbot, and the next actions were specified.

Sentence	**Slot/Entity**	**Value**
I want to rent an [electric]{"entity": "fuel_type"} [BMW]{"entity": "brand"}.	vehicle_type	standard
	brand	BMW

Table 3.3: Example training data with two slots/entities

The training data example in Figure 3.4 is an example for situations where role structure is involved in the process.

Sentence	**Entity**	**Role**	**Slot**	**Value**
between [05/09/2022]{"entity": "date", "role": "start"} to [24/11/2022]{"entity": "date", "role": "end"}	date	start	dateStart	05/09/2022
	date	end	dateEnd	24/11/2022

Table 3.4: Example training data for dates

3.4.2 Writing the Stories

It is essential to consider all kinds of scenarios when designing stories. Although we strengthen the stories with the rules, teaching the model the possible dialogues that the users will establish with the chatbot in advance enables a more successful model to be reached. While creating the stories, the central storylines were created first. The main storylines are that the user directly goes to rent a car, or the user has requests directly related to frequently asked questions.

3.4.3 Setting the Rules

Rules solidify the stories that are created and ensure that everything is created in a uniform order. In addition, the rules are the areas where the mechanisms that are desired to work under all conditions are specified. Before the rules were created in the project, first of all, the areas that should work under all conditions were determined. For example, actions that will take place after the user says hello to the system or says thank you. The point to be considered while developing the rules is to determine under which conditions the rule should be triggered.

3.5 Responses

While designing the answers of the chatbot, it is crucial that the answers that support all kinds of scenarios are determined. While designing the language of the generally determined answers of the chatbot presented in this project, the use of a language that is neither too formal nor too friendly was avoided. At this point, a balanced attitude is aimed. In most cases, the conversation flows that the chatbot will perform can be expected to occur within predetermined patterns. But there are different points to focus on when a chatbot is expected to provide adaptive answers, and a personalized experience is expected for each user. Adaptive flows will be discussed further in the section 3.8 in addition to this section.

In order to provide a personalized chat experience in the chatbot presented in this study, personalized chat content is provided on the basis of users. The presented chatbot is task-based, so the working context is focused on car rental. The vehicle preferences specified by the users during the conversation with the chatbot are captured in the natural language processing structure and processed in the actions section. In the user's subsequent car rental request, the chatbot presents the user's favorite type of vehicle to the user in the first section. The chatbot, which recognizes the user, also changes the conversation flow and directs the user to select the vehicle directly after the date selection. The daily rental prices of the vehicles that the user will encounter after the vehicle selection are calculated on the basis of the user, and each user is presented individually. The main factor in calculating the prices on a per-user basis depends on how many different car rentals the user has previously completed through the chat robot. This part is explained in more detail in section 3.8.

Since adaptive answers are determined according to the flow during the conversation, it is essential at this point to determine which answers the user can give at which point. In order to ensure that all possibilities are provided, in addition to the default response structure, it prevents the conversation flow from being disrupted in a scenario beyond the typical situations, but the negative contribution of this situation to the user experience should also be taken into account. In this project, the default answers produced for the entire speech flow are designed separately for each speech part and are included in the conversation when the user's inputs are not understood. Thus, it is aimed to make the chat experience more compelling by preventing users from seeing uniform answers in specific situations. For example, if the user uses an expression that the chatbot cannot understand while chatting about frequently asked questions, the chatbot will offer options to assist the user with frequently asked questions.

Finally, a point that should be mentioned is that the answers provided by the chatbot after the expressions that the users enter into the chatbot also form a mechanism that determines how the conversation will be directed. In this case, the answers and directions offered to the user are as crucial as language understanding in maintaining the speech under control.

3.6 Fast Access Code

Fast Access Code is a concept that will be encountered frequently in this project. User definitions in this thesis are provided through this unique code generated virtually. The code follows a certain format. The format and rule structure created for the Fast Access Code are as follows.

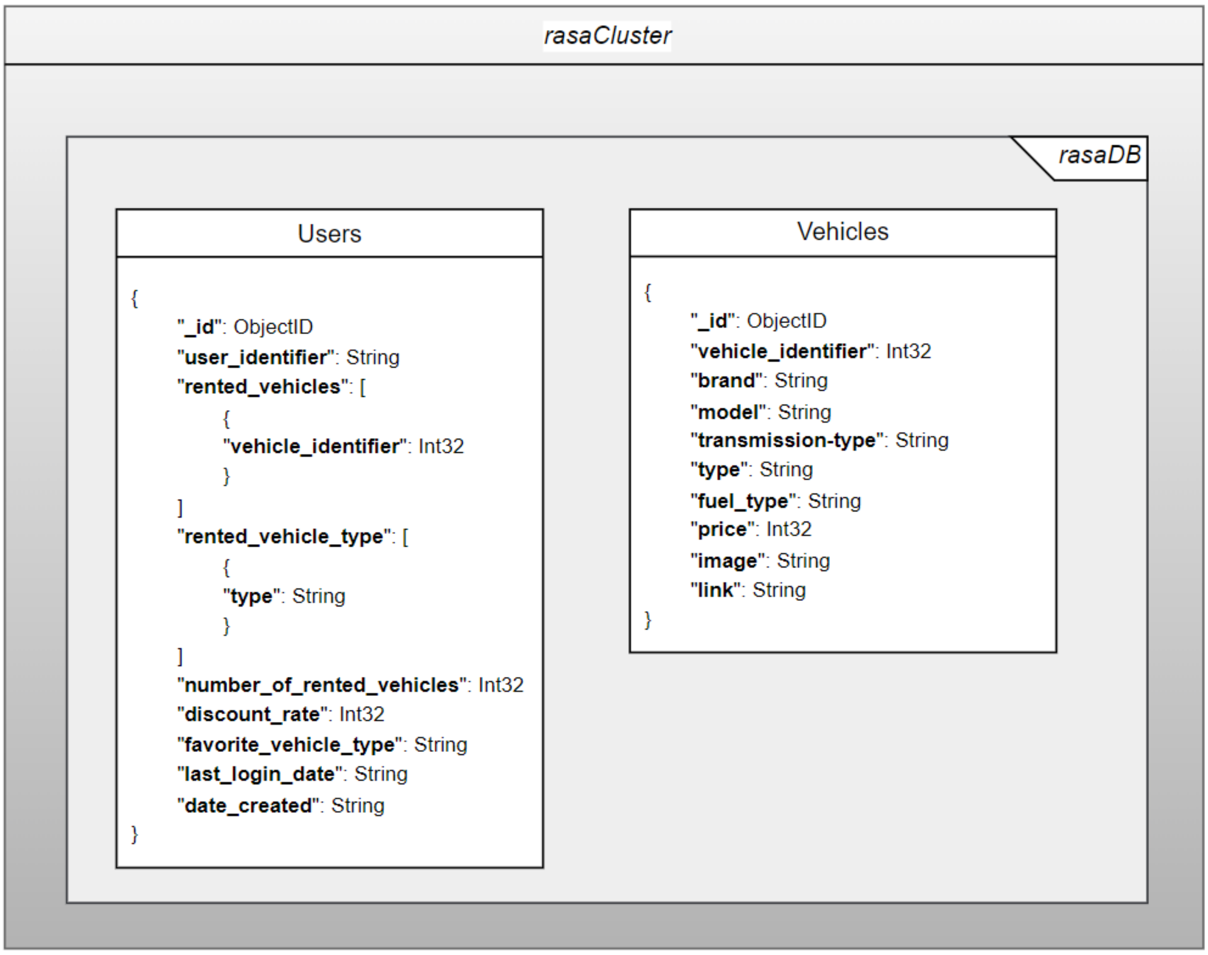
- Fast Access Code consists of 8 characters in total.
- The fourth character of the code always contains "#".
- It is randomly generated from a combination of the remaining characters (first three characters and last four characters) and both alphabetic and numeric expressions.

The reason for choosing a specially designed code format for Fast Access Code is to ensure that the code can be recognized successfully in interactions with the chatbot.

With the RegexEntityExtractor structure integrated into the chat robot in the project, the regex rule that defines the code snippet is provided to process user inputs.

3.7 Database

The areas where data is stored are based on two different essential collections: vehicles and users. The user collection is the area where users are associated with a virtually generated unique identifier (this identifier is called the "fast access code" in the project). Thanks to this code created virtually, user identification in the database is carried out through the virtual profile without requesting or storing any personal data of the users. The collection of vehicles is the area where vehicle features and information are kept for rental vehicles. The database in the NoSQL structure is kept on MongoDB.



```
{
    "_id": ObjectID
    "user_identifier": String
    "rented_vehicles": [
        {
        "vehicle_identifier": Int32
        }
    ]
    "rented_vehicle_type": [
        {
        "type": String
        }
    ]
    "number_of_rented_vehicles": Int32
    "discount_rate": Int32
    "favorite_vehicle_type": String
    "last_login_date": String
    "date_created": String
}
```

```
{
    "_id": ObjectID
    "vehicle_identifier": Int32
    "brand": String
    "model": String
    "transmission-type": String
    "type": String
    "fuel_type": String
    "price": Int32
    "image": String
    "link": String
}
```

Figure 3.2: Representation of database fields

The user_identifier field, which identifies users in profiles created virtually for users, is the area where fast access code is stored. Also, as seen in figure 3.2, the "_id" field defined by MongoDB is also preserved. The user_identifier area is used in the identification processes in the project.

After users complete their car rental transactions, the identifiers (vehicle_identifier) of the rented vehicles are stored in the rented_vehicles array. By creating this structure in array format, it is aimed that users can keep more than one contamination process. Array updates itself after each rental.

Each vehicle defined by users on the system is mapped to a vehicle type. The vehicles that will be offered to the users after the first vehicle rental in the system are kept based on vehicle type. A "rented_vehicle_type" field has been created in the user collection for faster access to the types of vehicles they have previously rented (in order not to return the "vehicle_identifier" of the previously rented vehicles again and again).

The explanations of other parts of the user collection are as follows.

- The total number of vehicles previously rented by the user is stored in the "number_of_rented_vehicles" field on the collection.
- The discount rate defined specifically for the user is stored in the "discount_rate" field. Calculating the discount rate is described in section 3.8.
- The user's favorite vehicle type is stored in the "favorite_vehicle_type" field. Possible scenarios for determining favorite vehicle types are discussed in section 3.8.
- The "lats_login_date" field has been created to recognize the user's interaction with the chatbot after registering in the system.
- As a consequence of the user's interaction with the chatbot, the fast access code of the chatbot is kept in the "date_created" field for the user.

The "date_created" and "lats_login_date" fields have no direct effect on the functioning of the chatbot. These areas can be used in conjunction with other areas to provide the basis for meaningful data analysis. The project will discuss the analysis of this area for future studies in section 6.3. Thanks to these fields created, the user model is built, and the user model is included in the conversation flows that the chatbot will present.

In addition to the "_id" field defined by MongoDB, the collection where the information about the vehicles is kept includes a unique id field called "vehicle_identifier". This id field is a 4-digit integer, and it is unique in the collection for each vehicle. Other fields in the collection of vehicles are as follows:

- The brand of the vehicle is stored in the field named "brand".
- The vehicle model is stored in the collection area named "model".

- The gear type of the vehicle is stored in the field named "transmission_type".
- The vehicle type is stored in the area called "type".
- The fuel type of the vehicle is stored in the field named "fuel_type".
- The rental fee of the vehicle is stored in the "price" field, representing the daily price.
- When the user completes the rental process in conversation with the chatbot, the chatbot shows the user a picture of the rented vehicle. This picture is kept in the link in the "image" field on the collection.
- When the car rental process is completed, the chat robot sends the user a link to the official website of the vehicle that belongs to the rented vehicle. The purpose here is to enable the user to access more detailed information about the rented vehicle. The official link of the vehicle's manufacturer is stored in the "link" field on the collection.

These fields in the database are managed by the actions file on the project. Section 4 explains this area in detail.

3.8 Adaptive Flows

This structure's development aims to increase the user's experience with the chat robot. The developed chatbots does not always give the same response to the user inputs it encounters. The figure 3.3 shows the operation behind different rules for the same user input, with an example from the project.

Figure 3.3 shows how the flow of the process changes adaptively from user to user, despite the same input from the user. At this point, the variable part is not only the dialog flow but also the answer content produced at the end of the dialogue. That is, the part where the vehicle recommendations are presented varies specifically for each user.

Users must first complete the first car rental process to see the car recommendations the chatbot recommends for them. After the first car rental, when the user states that he/she wants to rent a car to the chatbot, the chatbot recognizes the user and makes recommendations to the user according to the most recent car type that the user has rented. During the chat, each user sees recommendations based on their preferences from previous chats, giving them a different experience than a traditional rental process.

The different experience obtained at this point is not only limited to the fact that the vehicle recommendations differ from user to user but also that the car rental price offered

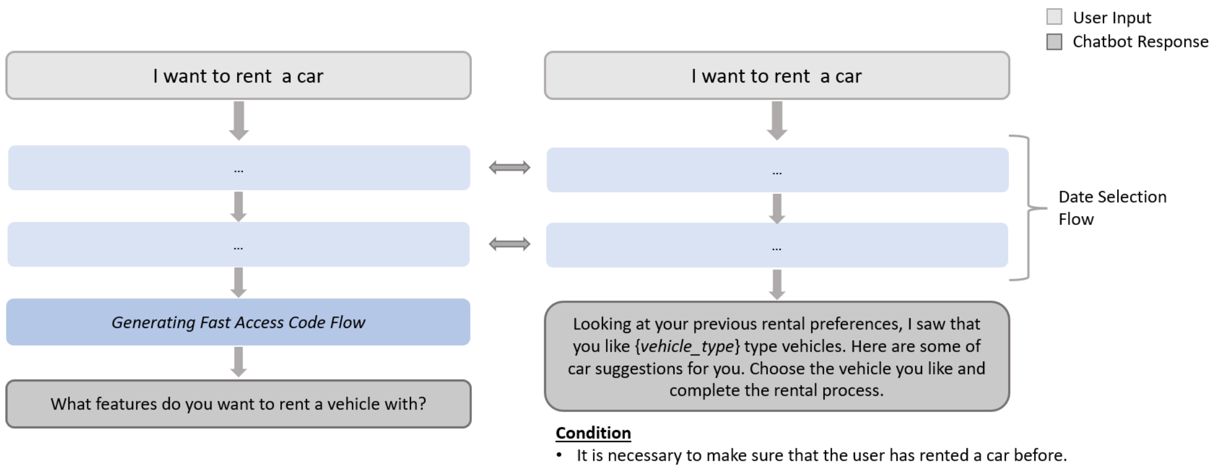

User Input
Chatbot Response
I want to rent a car
...
...
Generating Fast Access Code Flow
What features do you want to rent a vehicle with?
I want to rent a car
...
...
Date Selection Flow
Looking at your previous rental preferences, I saw that you like {vehicle_type} type vehicles. Here are some of car suggestions for you. Choose the vehicle you like and complete the rental process.
Condition
• It is necessary to make sure that the user has rented a car before.

Figure 3.3: An example of adaptive flow from the presented chatbot

for each user is variable on a user basis. The chatbot will show more affordable prices to its users who rent more cars than it does during the rental. The necessary condition here is that the user has previously completed a car rental and specified the fast access code to the chatbot. The chatbot provides users with a 5% discount per rental. Each user can get a maximum of 30% discount, which means that a fixed discount of 30% is applied after 6 car rentals are completed.

Another part of the chatbot where answers are produced adaptively is the user's creation of an input that indicates an intention different from the expected intentions. In this case, instead of creating a fixed answer of "Sorry, I don't quite understand this", the chatbot will be based on the topic of the current conversation and pass on topic-specific alerts to the user. For instance, if the user is interested in frequently asked questions and the text which is written by the user is not related to frequently asked questions, the chatbot offers to look at the frequently asked questions for the user. The critical point here is that the chatbot offers its own suggestions in the face of messages written outside the context of car rental. For example, the user can instantly specify the car rental request in the section where the user is interested in the frequently asked questions, or the user can instantly review the frequently asked questions while renting a car. This approach is advancing adaptively in all areas of the chatbot.

3.9 Use Case Scenarios

This section expresses the situations that indicate the dialogues to be established between the user and the chatbot with different use-cases. Under this heading in the report, Use-cases of basic processes (car rental and answering frequently asked questions) are presented. Other scenarios that may occur will be described in Appendix A.

Use Case Name	Initial Fast Access Code: None
Use Case Overview	This use-case occurred when the user did not have the fast access code before.
Actor	User
Trigger	User does not have fast access code. Indicates this status to the chatbot.
Precondition 1	The user says that he/she wants to rent a car and then states that they do not have fast access code.
Basic Flow	The user implies that he/she wants to rent a car. Chatbot asks for date information. User indicates dates to rent a car. Chatbot asks the user if he/she has a fast access code. User states that he/she does not have the code. The chatbot generates a fast access code for the user and asks the user what kind of vehicle they want to rent. The user specifies the features of the vehicle they want to rent. The chatbot lists the vehicles that the user wants. The user completes the rental process for the vehicle selected from the list and leaves the chat.
Alternate Flow	The user tells the chatbot that he/she has questions about the rental process. The chatbot waits for the user's question. The user asks the chatbot his/her question about renting a car. The chatbot answers the user's question and asks if they have any more questions. The user implies that he/she wants to rent a car. Chatbot asks for date information. User indicates dates to rent a car. Chatbot asks the user if he/she has a fast access code. User states that he/she does not have the code. The chatbot generates a fast access code for the user and asks the user what kind of vehicle they want to rent. The user specifies the features of the vehicle they want to rent. The chatbot lists the vehicles that the user wants. The user completes the rental process for the vehicle selected from the list and leaves the chat.

Table 3.5: Basic (Initial Dialogue - without fast access code) use-case for the car rental process

The use-case occurs when the user does not have a fac and wants to rent a car. As you can see, this scenario can ensue at the beginning of the conversation or after talking to the user about frequently asked questions.

The use-case expressed in the table 3.6 occurs when the user has fast access code (fac). If the user is a recognized user for the chatbot, the chatbot offers user-specific content in the conversation stream.

The use-case 3.7 occurs when the user has questions about renting a car. Other use-case flows are added to the report in the Appendix A section.

Use Case Name	Car rental when the user already has the code
Use Case Overview	This use-case occurred when the user had the fast access code before.
Actor	User
Trigger	The user says that he/she wants to rent a car and then states that they have the fast access code.
Precondition 1	The user has a fast access code and indicates this status to the chatbot.
Basic Flow	The user implies that he/she wants to rent a car. Chatbot asks for date information. User indicates dates to rent a car. Chatbot asks the user if he/she has a fast access code. User states that he/she has the code. The chatbot requests the user's fac. The user writes the fac. The chatbot asks the user what features a vehicle they want. The user specifies the features of the vehicle they want to rent. The chatbot lists the vehicles that the user wants. The user completes the rental process for the vehicle selected from the list and leaves the chat.
Alternate Flow	The user tells the chatbot that he/she has questions about the rental process. The chatbot waits for the user's question. The user asks the chatbot his/her question about renting a car. The chatbot answers the user's question and asks if they have any more questions. The user implies that he/she wants to rent a car. Chatbot asks for date information. User indicates dates to rent a car. Chatbot asks the user if he/she has a fast access code. User states that he/she has the code. The chatbot requests the user's fac. The user writes the fac. The chatbot asks the user what features a vehicle they want. The user specifies the features of the vehicle they want to rent. The chatbot lists the vehicles that the user wants. The user completes the rental process for the vehicle selected from the list and leaves the chat.

Table 3.6: Basic (Initial Dialogue - with fast access code) use-case for the car rental process

Use Case Name	Answering frequently asked questions
Use Case Overview	This use-case specifies the basic flow that is followed when users have questions about the car rental process.
Actor	User
Trigger	The user tells the chatbot that he/she has questions about the car rental process.
Precondition 1	It is sufficient to trigger this use-case.
Basic Flow	The user tells the chatbot that he/she has questions about the rental process. The chatbot waits for the user's question. The user asks the chatbot his/her question about renting a car. The chatbot answers the user's question and asks if they have any more questions. The user says he/she has a question and this loop repeats.
Alternate Flow	The user tells the chatbot that he/she has questions about the rental process. The chatbot waits for the user's question. The user asks the chatbot his/her question about renting a car. The chatbot answers the user's question and asks if they have any more questions. The user implies that he/she doesn't have any more question.

Table 3.7: The basic use-case for the FAQs.

3.10 Evaluation Questionnaire

A questionnaire was designed to measure the user experience of the chatbot in terms of user satisfaction, to evaluate the performance of the chatbot by associating it with the user experience provided by the chatbot, and to learn the general attitude of the users about the chatbots. The designed questionnaire is detailed and presented in section 5.3.

3.11 Summary

In this section, the decision stages and design approaches in the foundation process of the thesis are explained. In addition, use-cases were defined, and flows were created to represent all possible scenarios on the chatbot.

Chapter 4

Implementation

This section contains explanations and details about the development process of the chatbot in a segmented approach. This section also presents examples of conversations that demonstrate the adaptive/personalized chat experience offered by the chatbot via screenshots taken from the chatbot. The chatbot's basic and completed conversation flows are included in the report in Appendix B.

4.1 Overview of Development Environment

The development process of the project is based on the cooperation of different components. The development platforms/environments in the table below were used in the specified versions in this project, where the project was developed locally and run in the cloud environment, and the chat was managed together with a concerning database.

Rasa Version	3.1.0
Rasa SDK Version	3.1.1
Python Version	3.8.13
Cloud Database	MongoDB
Operating Systems	Linux Ubuntu 20.04.1 LTS / Windows 10

Table 4.1: Development Environments

The Rasa framework mentioned above needs many different python libraries to work in its full form. During the Rasa framework installation, these packages are installed together with the versions required for the framework to work. In addition to all these, this project offers an active dialogue flow with the database. The project's database connections and detailed information in this area will be explained in section 4.3.

4.2 Chatbot Implementation

The development of the chatbot has been completed in a modular structure in itself. It is aimed to explain the code blocks of the basic parts of the project in this section.

4.2.1 Actions

The Actions section contains python-built class structures and python functions. The project is based on a wide variety of functions and classes to increase chatbot and user interaction. In this section, the custom actions of the project, which are included in the main dialogs, will be examined with example code blocks.

4.2.1.1 Generate FAC

Indicating that the user does not have a fast access code (FAC) during the dialogue triggers the ”action_generate_fast_access_code” action. This triggered action is conveyed in the code block below.

```python
class ActionGenerateFastAccessCode(Action):

    def name(self) -> Text:
        return "action_generate_fast_access_code"

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:

        generated_id=generate_id()
        dispatcher.utter_message(response = "
    utter_creating_fast_access_code_info")
        dispatcher.utter_message(text=f"Here's your Fast Access Code: {
    generated_id}")

        return [SlotSet("user_identifier", generated_id), SlotSet("
    user_identifier_is_know", True)]
```

Listing 4.1: FAC Creation

As you can see, the code block 4.1 also uses a function called ”generate_id”. This function is the part where the fast access code is created after certain rules and controls. This function also enables the creation of fields belonging to the user collection in the database. Therefore, the explanation of this code is in the code block 4.7 in the database section 4.3.

4.2.1.2 Vehicle Selection

After the intent named typeOfVehicle, which is the intent in which the user defines the vehicle they want to rent, the action class named ”action_selectedVehicles” runs. This action class has quite a long code flow. The code block 4.2 shows only a portion of the code flow for this class. Although the entire code stream functions similarly to the specified code block, there are some fundamental differences. The code block 4.2 belonged to the code block that was active only when the user specified his request for fuel type and vehicle type.

```
query={"type":vehicle_type,"fuel_type":fuel_type}
docs=vehicles_collection.find(query)
buttonList=[]
for i in docs:
    brand=i["brand"]
    vehicle_identifier=i["vehicle_identifier"]
    model=i["model"]
    transmission=i["transmission_type"]
    price=i["price"]
    price_=float(price)-(float(price)*float(discount_rate))
    price_=str(price_)
    text=brand+" "+model+" | "+transmission+" Transmission "+price_+"$/
   day" + " | Vehicle Code: " + str(vehicle_identifier)
    text_payload="I chose the car with the vehicle code "+str(
   vehicle_identifier)
    element={"payload": text_payload, "title": text}
    buttonList.append(element)
    print(text)
```

Listing 4.2: Vehicle Selection

After the vehicle features specified by the user are captured with the intent defined on the NLU, they are converted into a query in the actions section. The resulting query is run on the database, and the outputs are processed and converted into a form suitable for the chat dialog. The resulting printouts include vehicle information and price information, and this information is presented to the user in a button-shaped dialog that the user can select during the chat.

4.2.2 Intents

The process of making sense of users’ input begins with the creation of intents. User inputs are used in dialogues at all levels in this project. In other words, the working system of the buttons has been developed as user input instead of calling the automatic intent through the buttons supported by the Rasa framework. This approach shows that every stage of the dialogue is based on NLU processes. Below, we can see how the code is captured when the user specifies the FAC through the sample code block. Although there are different sample dialog entries for this intent, it has been added to the report as follows.

```
nlu:
- regex: user_identifier
  examples: |
    - [a-zA-Z0-9]{3}#[a-zA-Z0-9]{4}
- intent: write_user_identifier
  examples: |
    - my fast access code is [la8#2jc3](user_identifier)
```

Listing 4.3: Capturing of the fast access code

Code block 4.3 has an example of capturing values using regex. Using regex to recognize inputs in a particular pattern, as in this example, provides an efficient process flow. DIETClassifier is another structure that makes sense of entity values and inputs used in this project.

4.2.3 Entities and Slots

Entities and slots are value holders defined under the domain. The continuity and conditions of the developed dialog flows are related to the values of the slot values. In other words, the orientation of the slot values is one of the essential factors in adaptively changing speech flows.

Below are sample code blocks where entities and slots are defined. Slots can get their values through defined assets. A total of 13 different slot values are defined in this thesis. As an example, defining a slot value is shown in the code block 4.4.

```
slots:
  car_rental_completed:
    type: bool
    influence_conversation: true
    initial_value: false
    mappings:
    - type: from_entity
      entity: car_rental_completed
```

Listing 4.4: Initialization of Slots

The slot value defined stores whether the user's car rental reservation has been successfully completed.

A total of 13 different entity values are defined in this thesis. Example entity definitions are shown in the code block below.

```
entities:
— date
— user_identifier_is_know
— car_rental_completed
```

Listing 4.5: Initialization of Date Entity

Defined entities can be used to feed slot values as seen in code block 4.4.

4.2.4 Responses

This section contains the pre-made answers used by the chatbot. The responses of the chatbot are defined under the domain and action file.

```
responses:
  utter_askforOtherQuestion:
  - text: "Is there anything else I can help with?"
  utter_waitQuestion:
  - text: "Could you please tell me your question?"
  utter_saybyespecial:
  - text: "It was a pleasure helping you. Hope to see you again,
   goodbye"
```

Listing 4.6: Responses

Emojis (Emojipedia (2022)) were also used in addition to texts in some replies so that users could have a more intimate speaking experience when reading the chatbot's responses.

4.2.5 Components and Policies

The spacyNLP language model is used in the developed chat robot. This language model is detailed in section 4.2.5.1. While SpacyTokenizer, which is offered as Tokenizer by Spacy, is used, SpacyFeaturizer is included in the process as Featurizer. In addition, CountVectorsFeaturizer, LexicalSyntacticFeaturizer, and RegexFeaturizer are also included in the process. In the presented chatbot, DIETClassifier, and FallbackClassifier are used as classifiers, and DIETClassifier also functions as entity extractors. There is a point to be noted in the entity extractors section where RegexEntityExtractor is also included in the process. This is important to be aware of as using DIETClassifier and RegexEntityExtractor together can result in multiple extractions of entities. In the project where EntitySynonymMapper is used, ResponseSelector is used as Selector. Finally, MemoizationPolicy, UnexpecTEDIntentPolicy, TEDPolicy, and RulePolicy were used as policy in the project.

4.2.5.1 Language Models

While developing the chatbot presented in this study, pre-trained word vectors were used. For this, a natural language library spacy (spaCy (2022)) is used. The spacyNLP included in the Pipeline is used with the "en_core_web_md" language model. The "en_core_web_md" model has better accuracy values than the "en_core_web_sm" model, but has a smaller size than the "en_core_web_lg" model (spaCy Trained Pipelines (2022)). This makes the "en_core_web_md" model one of the selectable models for achieving a balance of accuracy and efficiency.

The "case_sensitive" parameter was applied as "False" when configuring this language model. This is because in our chatbot streams, capitalizing or lowering the user's typing has no decisive impact on streams and assets. The fields with case sensitivity are processed in the "actions" section and converted into the required form. For example, after the parts in which the user specifies the vehicle brands and fuel type are separated, they are converted into a form with the first letter capitalized because the "actions" part is intended to work in a suitable form with the database. In this section, it is possible to apply different approaches, such as processing all database content in lowercase in the "actions" section.

4.3 Database

Another factor that determines the flow of the chatbot is the values kept in the database. The system first creates the collection fields for the user. Each userspace group is created with a unique fast access code. The code block 4.7 added to the report represents the code lines (test lines were removed while adding the code block to the report) for these operations. The first part of the code block added below is the part where this descriptive code is created.

```
def generate_id():
    rnd1=random.choices(string.ascii_lowercase+string.ascii_uppercase+
   string.digits,k=3)
    rnd1=str(''.join(rnd1))
    rnd2=random.choices(string.ascii_lowercase+string.ascii_uppercase+
   string.digits,k=4)
    rnd2=str(''.join(rnd2))
    final_rnd=rnd1+"#"+rnd2

    user_identifier_list=update_userList(client_)

    if not check_id(final_rnd,user_identifier_list):
        currentDateTime = datetime.datetime.now()
        currentDateTime_=currentDateTime.strftime("%d/%m/%Y %H:%M:%S")
        dict = { "user_identifier": final_rnd,
        "rented_vehicles":[],
        "rented_vehicle_type":[],
        "number_of_rented_vehicles":0,
        "discount_rate":0,
        "favorite_vehicle_type":"",
        "last_login_date": "",
        "date_created": currentDateTime_ }
        users_collection=return_userCollection(client_)
        newDoc = users_collection.insert_one(dict)
        return final_rnd
    else:
        generate_id()
```

Listing 4.7: Creation of FAC and user collection fields

After making sure that the unique code that is created and brought to the desired format has not been produced before, the collection fields and FAC are matched to each other, and the procedure of adding new data to the collection is completed. The ”check_id()” function scans the FACs in the user collection. If the generated FAC is registered to the system before, the function calls itself again and works recursively, and when a code that has not been generated before is obtained, the creation of the collection areas is completed. This process allows the user model to be defined on the database. The next processes (renting a car, etc.) continue by updating these user-owned fields depending on the user’s choices.

4.4 Deployment to the Web

The processes in the following subsections have been applied for the public access of the users to the developed chatbot.

4.4.1 Setting the Channel Connector

REST and SocketIO channels are defined in the project. In this way, host and port numbers can be used, and the chatbot can be contacted.

4.4.2 Setting the Virtual Machine

An instance has been created on Google Cloud (Google Cloud (2022)) to run the chatbot on the virtual machine. The built virtual machine is configured to have a total of 60 GB of storage space. After the necessary library installations were made on the virtual machine, the developed chatbot was run on the virtual machine.

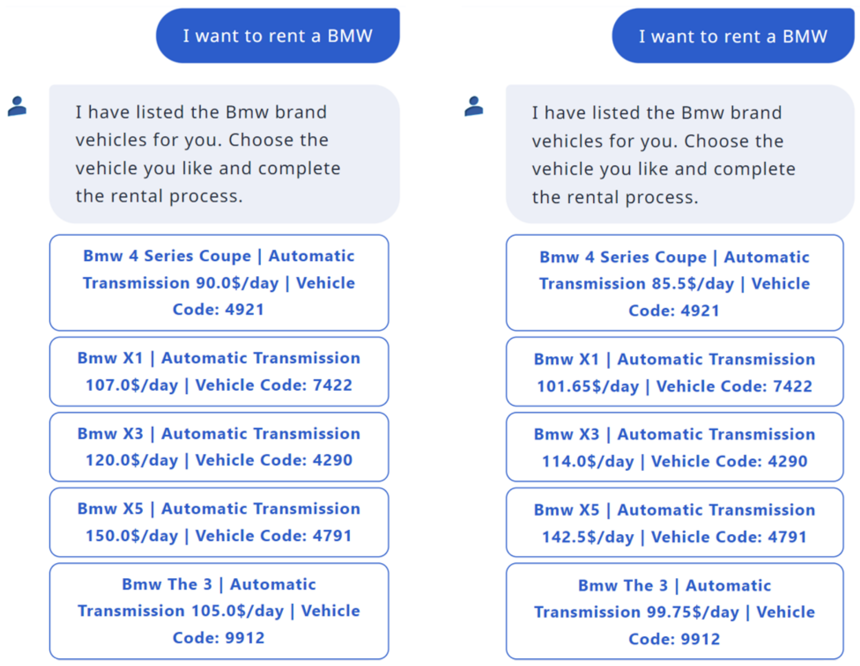
4.4.3 Web Interface

After the development of the chatbot was completed, it was shared with the users via the web interface to ensure interaction with the users. While designing the interface, the Bootstrap (Bootstrap (2022)) framework was used. The designed responsive interface allows users to use the chatbot from all devices.

4.5 Adaptive Flows

The chatbot, which has been completed up to this stage, has gained a structure that will present adaptive and personal content. An adaptive and personalized chat experience has been implemented in different areas of the chatbot. In this section, the narrations will be supported by the screenshots taken while continuing the conversation with the chatbot.

While evaluating the user's car rental requests during the conversation, the chatbot uses the values in the user model. As a result, personalized response contents and speech flows are obtained. As can be seen in the Figure 4.1, although the requests of 2 different users are the same, the reply content produced by the chatbot is different for the users.


I want to rent a BMW

I have listed the Bmw brand vehicles for you. Choose the vehicle you like and complete the rental process.

Bmw 4 Series Coupe | Automatic Transmission 90.0$/day | Vehicle Code: 4921

Bmw X1 | Automatic Transmission 107.0$/day | Vehicle Code: 7422

Bmw X3 | Automatic Transmission 120.0$/day | Vehicle Code: 4290

Bmw X5 | Automatic Transmission 150.0$/day | Vehicle Code: 4791

Bmw The 3 | Automatic Transmission 105.0$/day | Vehicle Code: 9912

I want to rent a BMW

I have listed the Bmw brand vehicles for you. Choose the vehicle you like and complete the rental process.

Bmw 4 Series Coupe | Automatic Transmission 85.5$/day | Vehicle Code: 4921

Bmw X1 | Automatic Transmission 101.65$/day | Vehicle Code: 7422

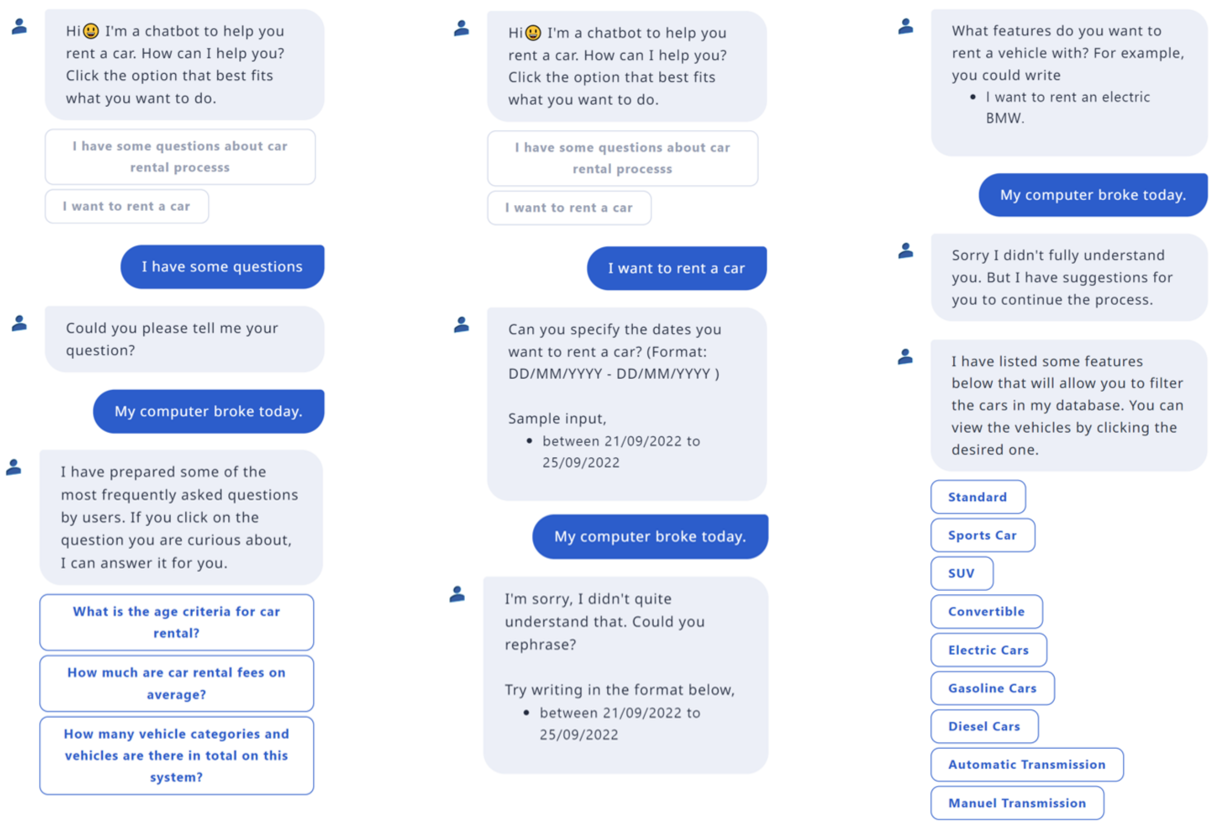
Bmw X3 | Automatic Transmission 114.0$/day | Vehicle Code: 4290

Bmw X5 | Automatic Transmission 142.5$/day | Vehicle Code: 4791

Bmw The 3 | Automatic Transmission 99.75$/day | Vehicle Code: 9912

Figure 4.1: Adaptive Price Determination During Conversation

Figure 4.2 refers to the flows that occur when the user's messages cannot be understood by the chatbot (a classification below the specified threshold may cause this situation). In the context of car rental, the user specifying a message outside of a topic and expected flows requires the chatbot to perform some additional actions. Although it is one of the usual approaches to ask the user to rewrite his message regardless of the current topic of the conversation in such cases, the chatbot developed in this study provides a solution to this situation with a different approach. The developed chat robot gives feedback based on the current state of the conversation in the face of expressions that it cannot make sense of with a robust classification. As seen in Figure 4.2, if the user enters a message that the chatbot does not understand while talking about questions about renting a car, the chatbot aims to continue the conversation more effectively by suggesting frequently asked questions to the user. In another example, when the user writes a message that is out of context during the vehicle selection phase, it triggers the chatbot to make its own suggestions. The chatbot offers some category suggestions based on the types of vehicles found in the database to assist the user in choosing.


Hi🙂 I'm a chatbot to help you rent a car. How can I help you? Click the option that best fits what you want to do.

I have some questions about car rental process

I want to rent a car

I have some questions

Could you please tell me your question?

My computer broke today.

I have prepared some of the most frequently asked questions by users. If you click on the question you are curious about, I can answer it for you.

What is the age criteria for car rental?

How much are car rental fees on average?

How many vehicle categories and vehicles are there in total on this system?

Hi🙂 I'm a chatbot to help you rent a car. How can I help you? Click the option that best fits what you want to do.

I have some questions about car rental process

I want to rent a car

I want to rent a car

Can you specify the dates you want to rent a car? (Format: DD/MM/YYYY - DD/MM/YYYY)

Sample input,

- between 21/09/2022 to 25/09/2022

My computer broke today.

I'm sorry, I didn't quite understand that. Could you rephrase?

Try writing in the format below,

- between 21/09/2022 to 25/09/2022

What features do you want to rent a vehicle with? For example, you could write

- I want to rent an electric BMW.

My computer broke today.

Sorry I didn't fully understand you. But I have suggestions for you to continue the process.

I have listed some features below that will allow you to filter the cars in my database. You can view the vehicles by clicking the desired one.

Standard

Sports Car

SUV

Convertible

Electric Cars

Gasoline Cars

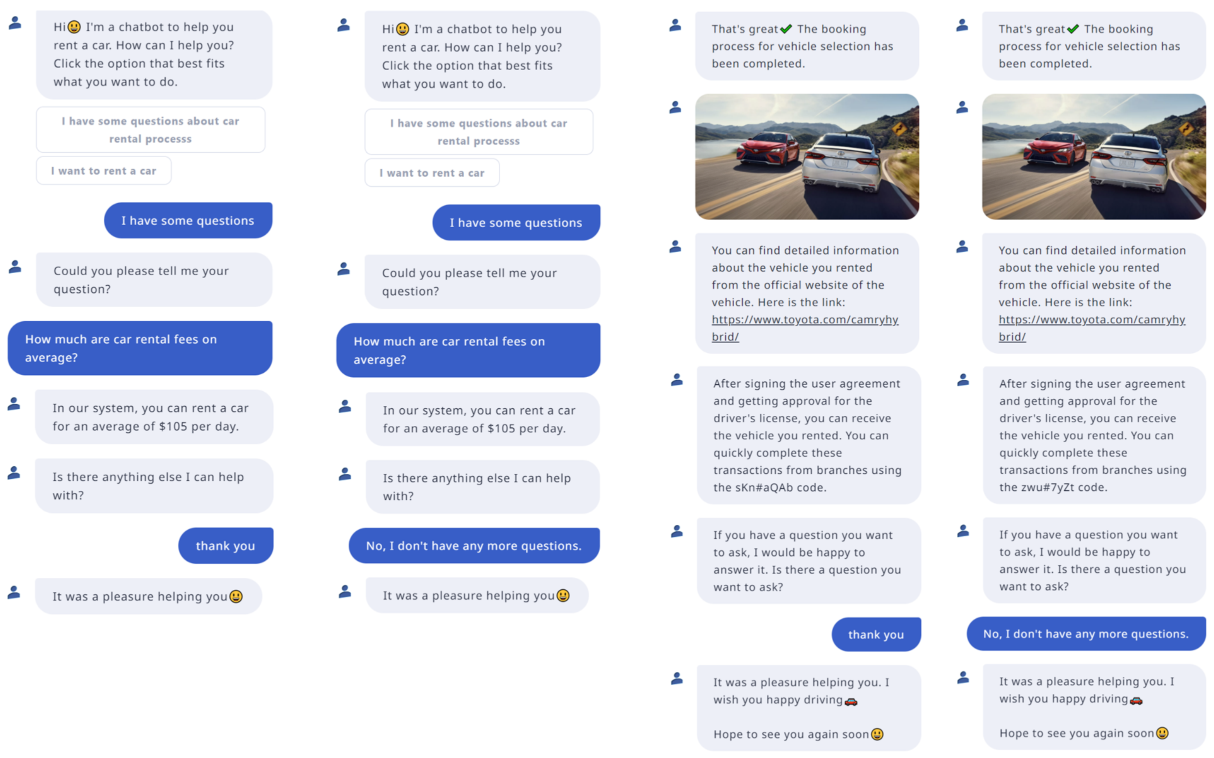
Diesel Cars

Automatic Transmission

Manuel Transmission

Figure 4.2: Customised Fallback Management

Users can end the conversation at any time of the flow, or they can write a statement on the chatbot that they do not want to continue in response to the stream that the chatbot wants to present. The presented chatbot responds to this request of users based on conversation history. Example speech flows illustrating this situation are presented in Figure 4.3. When the chatbot knows that the user has completed the car rental process, it sends a more detailed farewell message to the user, while in other cases, it says goodbye differently.


Hi🙂 I'm a chatbot to help you rent a car. How can I help you? Click the option that best fits what you want to do.

I have some questions about car rental processs

I want to rent a car

I have some questions

Could you please tell me your question?

How much are car rental fees on average?

In our system, you can rent a car for an average of $105 per day.

Is there anything else I can help with?

thank you

It was a pleasure helping you🙂

Hi🙂 I'm a chatbot to help you rent a car. How can I help you? Click the option that best fits what you want to do.

I have some questions about car rental processs

I want to rent a car

I have some questions

Could you please tell me your question?

How much are car rental fees on average?

In our system, you can rent a car for an average of $105 per day.

Is there anything else I can help with?

No, I don't have any more questions.

It was a pleasure helping you🙂

That's great✔ The booking process for vehicle selection has been completed.

You can find detailed information about the vehicle you rented from the official website of the vehicle. Here is the link: https://www.toyota.com/camryhybrid/

After signing the user agreement and getting approval for the driver's license, you can receive the vehicle you rented. You can quickly complete these transactions from branches using the sKn#aQAb code.

If you have a question you want to ask, I would be happy to answer it. Is there a question you want to ask?

thank you

It was a pleasure helping you. I wish you happy driving🚗

Hope to see you again soon🙂

That's great✔ The booking process for vehicle selection has been completed.

You can find detailed information about the vehicle you rented from the official website of the vehicle. Here is the link: https://www.toyota.com/camryhybrid/

After signing the user agreement and getting approval for the driver's license, you can receive the vehicle you rented. You can quickly complete these transactions from branches using the zwu#7yZt code.

If you have a question you want to ask, I would be happy to answer it. Is there a question you want to ask?

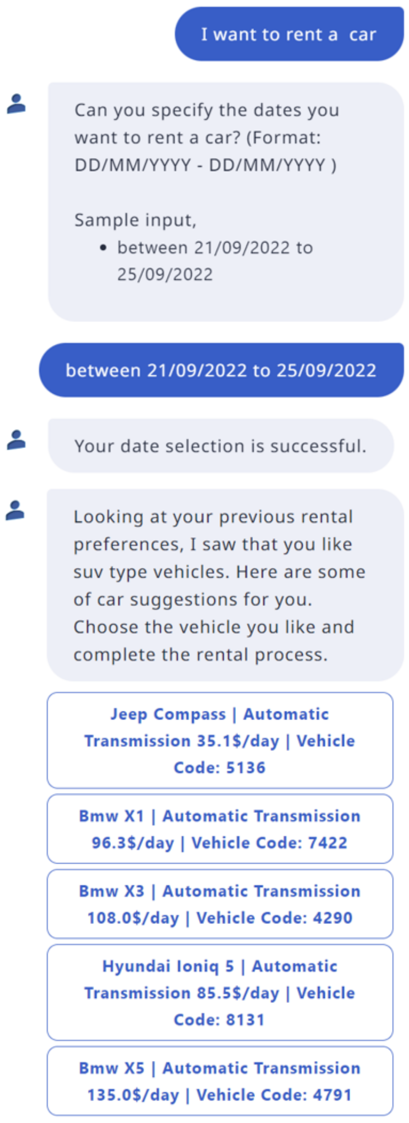
No, I don't have any more questions.

It was a pleasure helping you. I wish you happy driving🚗

Hope to see you again soon🙂

Figure 4.3: Adaptive responses for ending the chat

When users say they want to rent a car again while they are in a conversation flow where they have rented a car before, the chatbot will offer suggestions created specifically for the user. Conversation flows showing this situation can be seen in the Figure 4.4. Although the user message is ”I want to rent a car” in both streams seen in the figure, the vehicle suggestions shown are different. In addition, the daily fees of the recommended vehicles are calculated on a user basis and displayed to the user. This indicates that even if the same vehicles are offered for different users, the daily rental prices of the recommended vehicles may differ.



I want to rent a car

Can you specify the dates you want to rent a car? (Format: DD/MM/YYYY - DD/MM/YYYY)

Sample input,
- between 21/09/2022 to 25/09/2022

between 21/09/2022 to 25/09/2022

Your date selection is successful.

Looking at your previous rental preferences, I saw that you like suv type vehicles. Here are some of car suggestions for you. Choose the vehicle you like and complete the rental process.

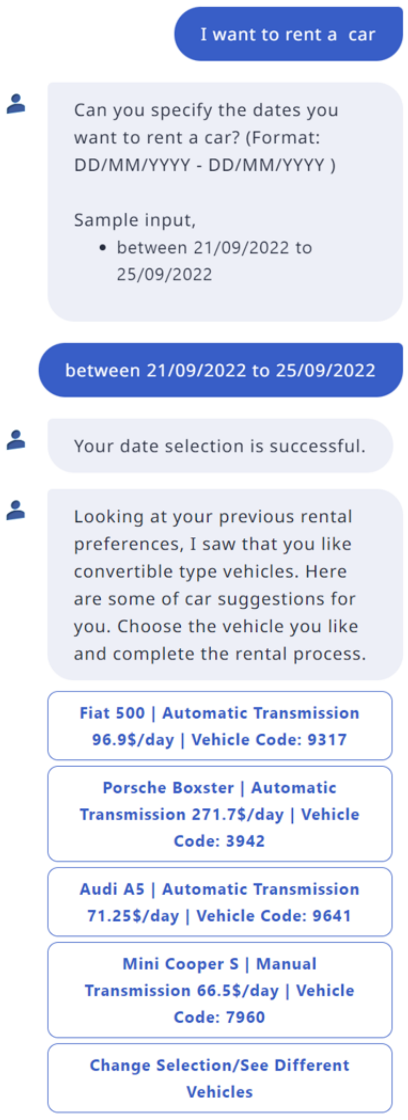
Jeep Compass | Automatic Transmission 35.1$/day | Vehicle Code: 5136

Bmw X1 | Automatic Transmission 96.3$/day | Vehicle Code: 7422

Bmw X3 | Automatic Transmission 108.0$/day | Vehicle Code: 4290

Hyundai Ioniq 5 | Automatic Transmission 85.5$/day | Vehicle Code: 8131

Bmw X5 | Automatic Transmission 135.0$/day | Vehicle Code: 4791



I want to rent a car

Can you specify the dates you want to rent a car? (Format: DD/MM/YYYY - DD/MM/YYYY)

Sample input,
- between 21/09/2022 to 25/09/2022

between 21/09/2022 to 25/09/2022

Your date selection is successful.

Looking at your previous rental preferences, I saw that you like convertible type vehicles. Here are some of car suggestions for you. Choose the vehicle you like and complete the rental process.

Fiat 500 | Automatic Transmission 96.9$/day | Vehicle Code: 9317

Porsche Boxster | Automatic Transmission 271.7$/day | Vehicle Code: 3942

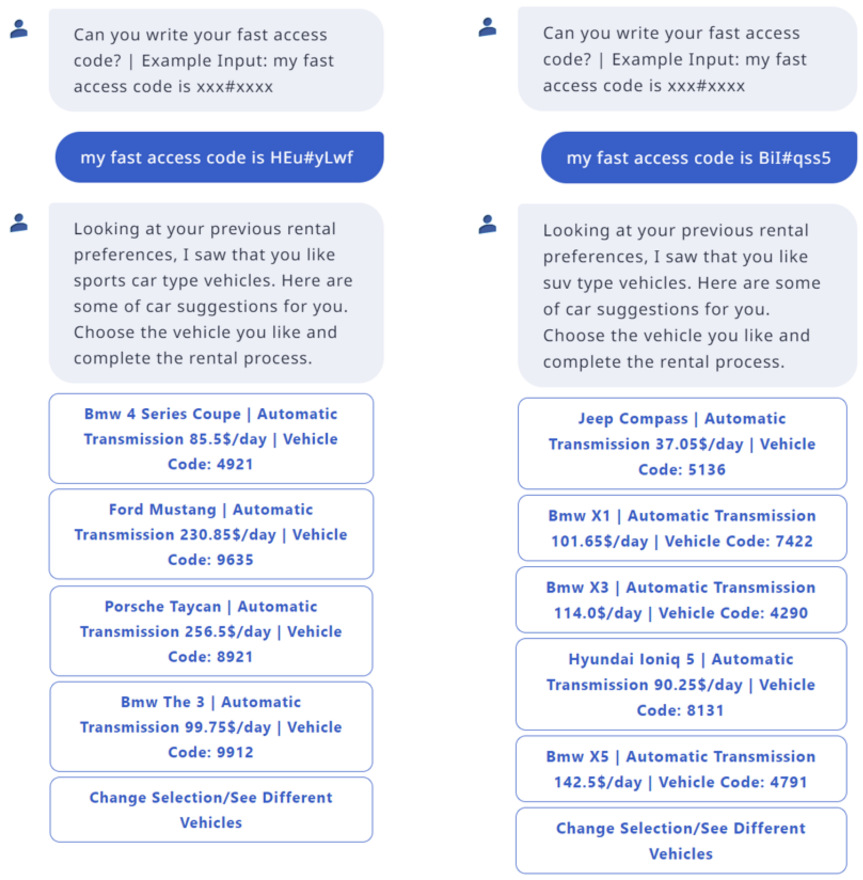
Audi A5 | Automatic Transmission 71.25$/day | Vehicle Code: 9641

Mini Cooper S | Manual Transmission 66.5$/day | Vehicle Code: 7960

Change Selection/See Different Vehicles

Figure 4.4: Vehicle recommendation on a per-user basis

After the users indicate that they have the fast access code, the chatbot asks the user to type the fast access code they have. Conversations representing this flow are exemplified in Figure 4.5. After the user writes a fast access code, the chat robot searches the database for the specified code and prepares vehicle suggestions based on the user's data for the match it finds. Another point that should be known at this point is that after the vehicles to be suggested according to the values in the user model are drawn from the cloud database of the chatbot, the daily rental fee is determined based on the user and translated into a form appropriate to the response format.


Can you write your fast access code? | Example Input: my fast access code is xxx#xxxx

my fast access code is HEu#yLwf

Looking at your previous rental preferences, I saw that you like sports car type vehicles. Here are some of car suggestions for you. Choose the vehicle you like and complete the rental process.

Bmw 4 Series Coupe | Automatic Transmission 85.5$/day | Vehicle Code: 4921

Ford Mustang | Automatic Transmission 230.85$/day | Vehicle Code: 9635

Porsche Taycan | Automatic Transmission 256.5$/day | Vehicle Code: 8921

Bmw The 3 | Automatic Transmission 99.75$/day | Vehicle Code: 9912

Change Selection/See Different Vehicles

Can you write your fast access code? | Example Input: my fast access code is xxx#xxxx

my fast access code is BiI#qss5

Looking at your previous rental preferences, I saw that you like suv type vehicles. Here are some of car suggestions for you. Choose the vehicle you like and complete the rental process.

Jeep Compass | Automatic Transmission 37.05$/day | Vehicle Code: 5136

Bmw X1 | Automatic Transmission 101.65$/day | Vehicle Code: 7422

Bmw X3 | Automatic Transmission 114.0$/day | Vehicle Code: 4290

Hyundai Ioniq 5 | Automatic Transmission 90.25$/day | Vehicle Code: 8131

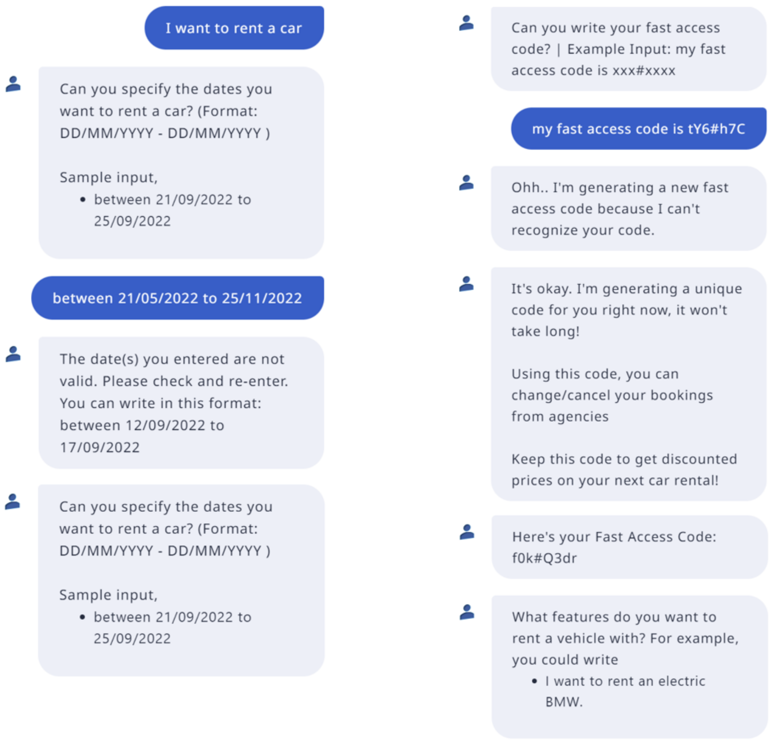
Bmw X5 | Automatic Transmission 142.5$/day | Vehicle Code: 4791

Change Selection/See Different Vehicles

Figure 4.5: Vehicle recommendation on a per-user basis (by specifying the FAC)

4.6 Incorrect Information Entry Management

The previous paragraphs explained how the chatbot reacts to unintelligible user input. When the information entered by the user in the date field and the fast access code field contains incorrect data, the chatbot behaves with a different approach and behaves specifically to these fields. This situation is presented in Figure 4.6 on 2 sample speech streams. A method has been developed that processes the date information entered on the back of the chatbot. This method takes the current date information instantly and compares it with the date information entered by the user. If the entered date is an expired date, the chatbot prompts the user to recheck this status by creating a warning. Another situation is the evaluation of the accuracy of the entered dates in themselves. The end date of the rental period may be equal to the beginning of the rental period or



I want to rent a car

Can you specify the dates you want to rent a car? (Format: DD/MM/YYYY - DD/MM/YYYY)

Sample input,
- between 21/09/2022 to 25/09/2022

between 21/05/2022 to 25/11/2022

The date(s) you entered are not valid. Please check and re-enter. You can write in this format: between 12/09/2022 to 17/09/2022

Can you specify the dates you want to rent a car? (Format: DD/MM/YYYY - DD/MM/YYYY)

Sample input,
- between 21/09/2022 to 25/09/2022

Can you write your fast access code? | Example Input: my fast access code is xxx#xxxx

my fast access code is tY6#h7C

Ohh.. I'm generating a new fast access code because I can't recognize your code.

It's okay. I'm generating a unique code for you right now, it won't take long!

Using this code, you can change/cancel your bookings from agencies

Keep this code to get discounted prices on your next car rental!

Here's your Fast Access Code: f0k#Q3dr

What features do you want to rent a vehicle with? For example, you could write
- I want to rent an electric BMW.

Figure 4.6: Incorrect entries management

maybe a later date. If a date information pair is specified that breaks this status, the chatbot will catch it and generate a warning for the user.

4.7 Summary

This section describes the critical components used/created during the development of the chatbot. The modular development approach followed during the development of the chatbot not only provided easier process management during development but also provided a more conscious intervention to the errors encountered.

Chapter 5

Evaluation

This section explains the approaches applied in the evaluation process of the chatbot developed and presented in this thesis. The foremost section will provide an overview of assessment approaches. The following section details the evaluation approach and evaluation result for Intent Classification. This study designed a survey to share with real users to measure the overall performance and user experience of the chatbot. The section after the evaluation about the intent classification has completed the section where all the information about the survey study is shared in sections. The final section, which is called "Summary", gives an overview of the entire evaluation section.

5.1 Evaluation Overview

The evaluation part can be examined in two parts. The first evaluation approach applied is a test process that measures the success of intent classification, which also provides information about the quality of the chatbot's training data. The second evaluation approach is a survey study based on real user tests that measure the user experience of the presented chatbot in terms of user satisfaction.

In order to understand how well the presented chatbot performs the intent classifications, a test phase was implemented on the chatbot.

For the user experience evaluation of the developed chatbot in terms of user satisfaction, a survey study based on real user tests was prepared. Participants are expected to start this survey after completing their chatbot experience and complete the survey based on their chat experience. It is known that the outputs of this survey, which will be completed by users, are essential in terms of understanding how the task-based chatbot presented in this report creates an impression on the user's side. This evaluation process also aimed to provide a glimpse into the positive/negative contribution of a personalized experience to the overall experience users get from chatbots.

5.2 Evaluation for Intent Classification

In order to evaluate how successfully the intent classifications of the chatbot developed in this study work, the evaluation results in Figures 5.1 and 5.2 were obtained by using the test mechanism provided by Rasa Open Source.

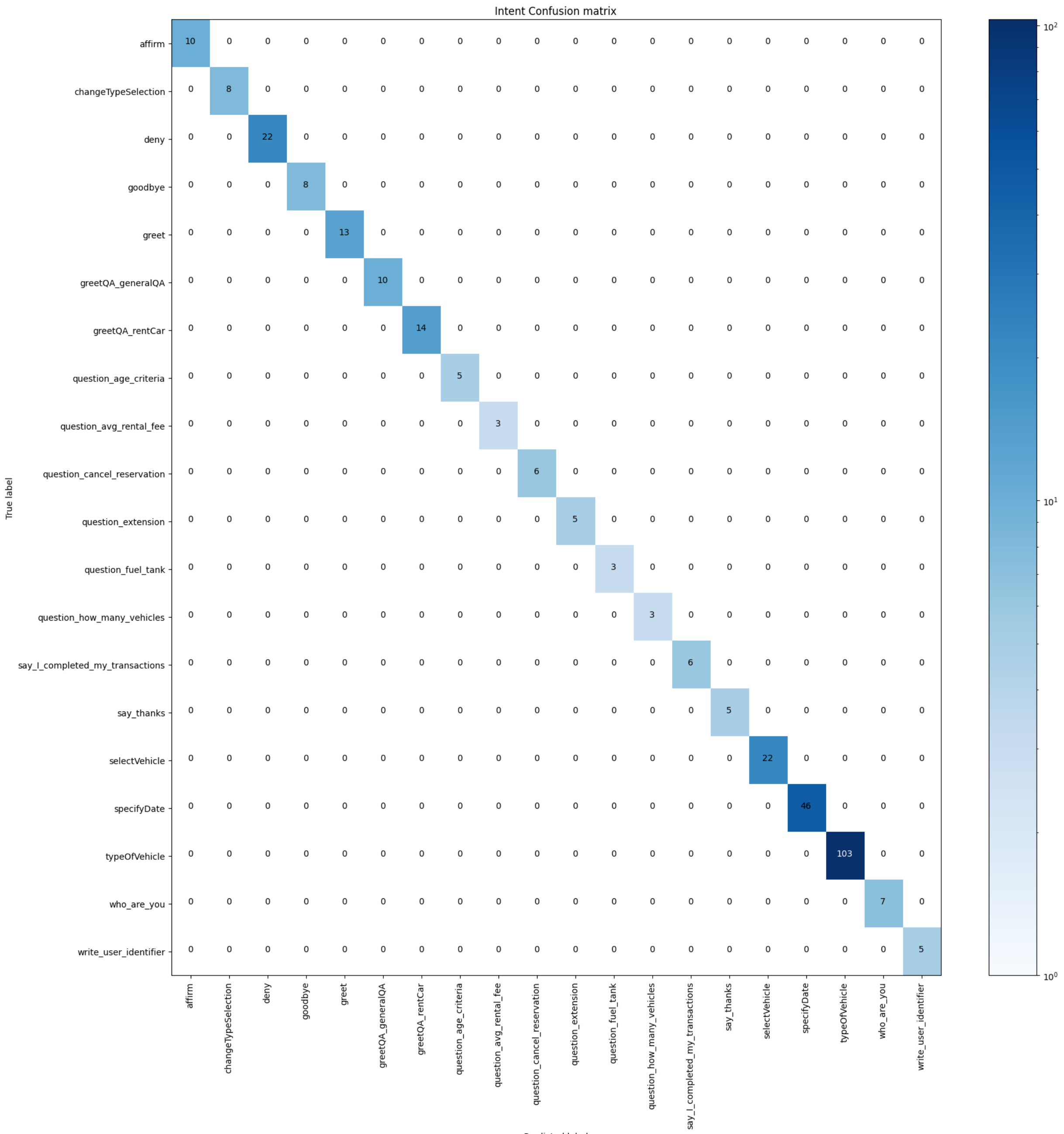


Figure 5.1: Confusion Matrix for Intent Classification

Figure 5.1 is the confusion matrix created according to the results of the intent classification evaluation. The horizontal row contains predicted intents, while the vertical column contains actual intents/true labels. In a successful classification, it is expected that the confusion matrix should be arranged diagonally. This indicates that the classification to which each objective corresponds has resulted correctly. As seen in the Figure 5.1, each intent matched with its corresponding intent, and the whole matrix moved on the diagonal. This completed evaluation shows that intent classification works successfully on the presented chatbot. In addition, the presence of different color tones can be seen on the confusion matrix. An intent with a darker hue was supported by more sample input than an intent with a lighter hue. We can also observe this situation by comparing the values on the diagonal.

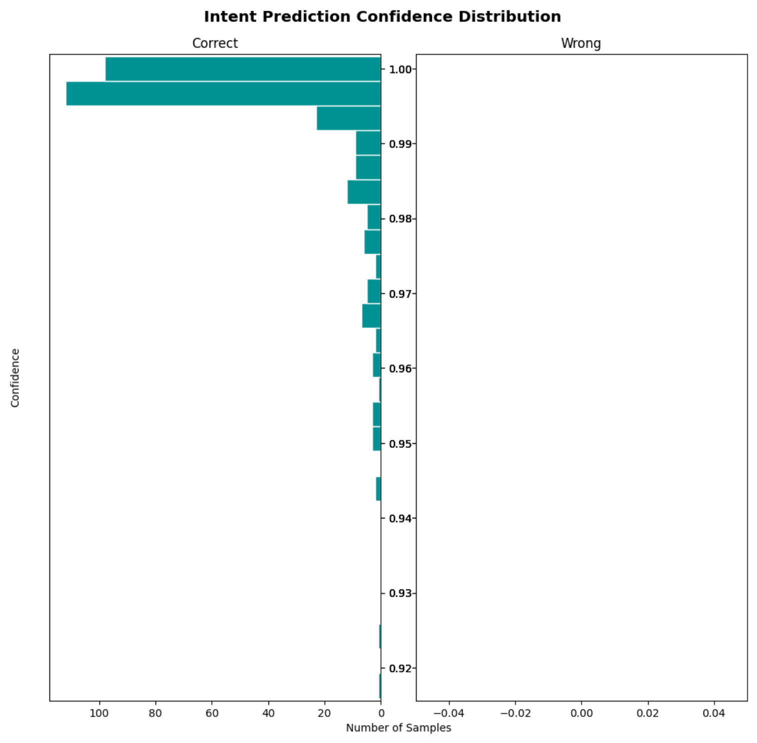


Figure 5.2: Confidence Distribution for Intent Prediction

In Figure 5.2 above, we can see the confidence values for correct and wrong intent classifications on the histogram. Correct predictions are represented by blue bars and losses by red bars. As can be seen, the intent classification operates successfully.

5.3 Questionnaire

The designed questionnaire consists of 24 different expressions in total. The first 12 of these statements contain positive statements, while the last 12 include negative statements. For all the expressions presented, five different evaluation scales(5-point Likert scale(Likert (1932))) are presented, namely "Strongly Disagree", "Disagree", "Neutral", "Agree", and "Strongly Agree". Option 1 is "Strongly Disagree", while option 5 is "Strongly Agree". The generated evaluation statements can be seen in Figure 5.3 and Figure 5.4.

When the preparation of the questionnaire was completed, an application was made to the Trinity College Dublin Research Ethics Committee, which was the test environment with the questionnaire and the chatbot, and the process of sharing the evaluation questionnaire was started after the committee's approval.

5.3.1 Determination of Evaluation Statements

The questionnaire presented in this chapter has determined a total of 24 evaluation expressions by using the evaluation expressions and approaches presented for usability and user experience in the literature. The completed study (Følstad and Brandtzaeg (2020)) examined user experience in chatbots and presented an evaluation approach for positive and negative experiences in chatbots. Another study (Holmes et al. (2019)) conducted a study on chatbot usability. In the study, the statements added as a table for the Chatbot Usability Questionnaire (CUQ) were presented. In the given table of the Chatbot Usability Questionnaire (CUQ), which consists of 16 expressions in total, odd numbers consist of positive evaluation expressions, and even numbers consist of negative evaluation expressions.

Evaluation expressions in the questionnaire presented in this project consist of a total of 24 evaluation expressions, 12 of which are positive and 12 of which are negative. Another point to be noted in the questions in the questionnaire presented is that the 12 positive expressions and 12 negative expressions are not always the opposite of each other. That is, one positive expression may not necessarily have a negative equivalent or one negative expression may not necessarily have a positive equivalent in the evaluation expressions presented. This situation was managed by looking at whether the evaluation statements contain a general judgment or not. For example, the statement, which is "The chatbot generated an identifying code for me without requesting my personal information", is located only in the positive part. The negative of this statement is not among the evaluation questions. This is because this evaluation statement refers to a relatively more specific and precise area, the user is not expected to be indecisive, and the choice is expected to be clear.

It is aimed to make the results of the users' evaluations more reliable by presenting positive and negative evaluation statements in the survey presented. In cases where a user is not entirely sure about the evaluation, allowing them to think and evaluate both positively and negatively and assess the positive and negative distributions obtained as a result of the survey together will ensure the consistency and reliability of the results and evaluations. For example, if a user agrees that the provided chatbot is easy to use, the user is not expected to agree that the chatbot is complex to use. Therefore, it is expected that the distributions will be distributed symmetrically in evaluation expressions that are opposite to each other. On the other hand, unsymmetrical distributions can be given as an example of areas where users are indecisive.

The objectives of the questionnaire and desired outputs as a result of the questionnaire to be completed are as follows:

- The presented survey aims to measure the user experience provided by the chat robot in terms of user satisfaction.
- The second objective is to measure the performance of the developed chatbot by associating it with the user experience.
- The last objective of the questionnaire is to learn users' general habits in using chatbots regardless of the chatbot experience presented in this study.

5.3.2 Participant Recruitment

Participants were requested to partake in the survey by the announcement made among M.Sc. Computer Science students at Trinity College Dublin. For this reason, users are experienced in computer-human interaction. In addition, the text of the announcement kindly asks the people who read the announcement to share it in their own group, so it is aimed to diversify the participant profile. At the end of the process, 10 participants in total stated that they wanted to participate in the evaluation survey, and the survey link was sent to them. After the announcement text was shared, 10 participants tested the chatbot and completed their evaluations.

5.3.3 Ethical Considerations

A detailed information text has been created so that users can have detailed information about the survey and the chat robot they will test. The rights of each user during and after the study (for example, participants are free to answer all questions on the questionnaire, participants can delete their experiences from the study after the study, for this they should e-mail the fac to the researcher) were explained to the participants and presented with a statement form. No personal data was requested from the participants in order to store them during the survey or chatbot experience, and this was explained to them in detail. Chat robot does not request or store any personal information of any user since it includes the users in the process by generating the virtual identifier code (fast access code) that it produces. The evaluation survey also confirms to the users with a statement[1] added on the survey that the chatbot continues its activities without requesting any personal information.

5.3.4 Results

After the results were collected on Microsoft Form, the survey results were displayed on the result display screen, and the Figure 5.3 and the Figure 5.4 were added to the section below.

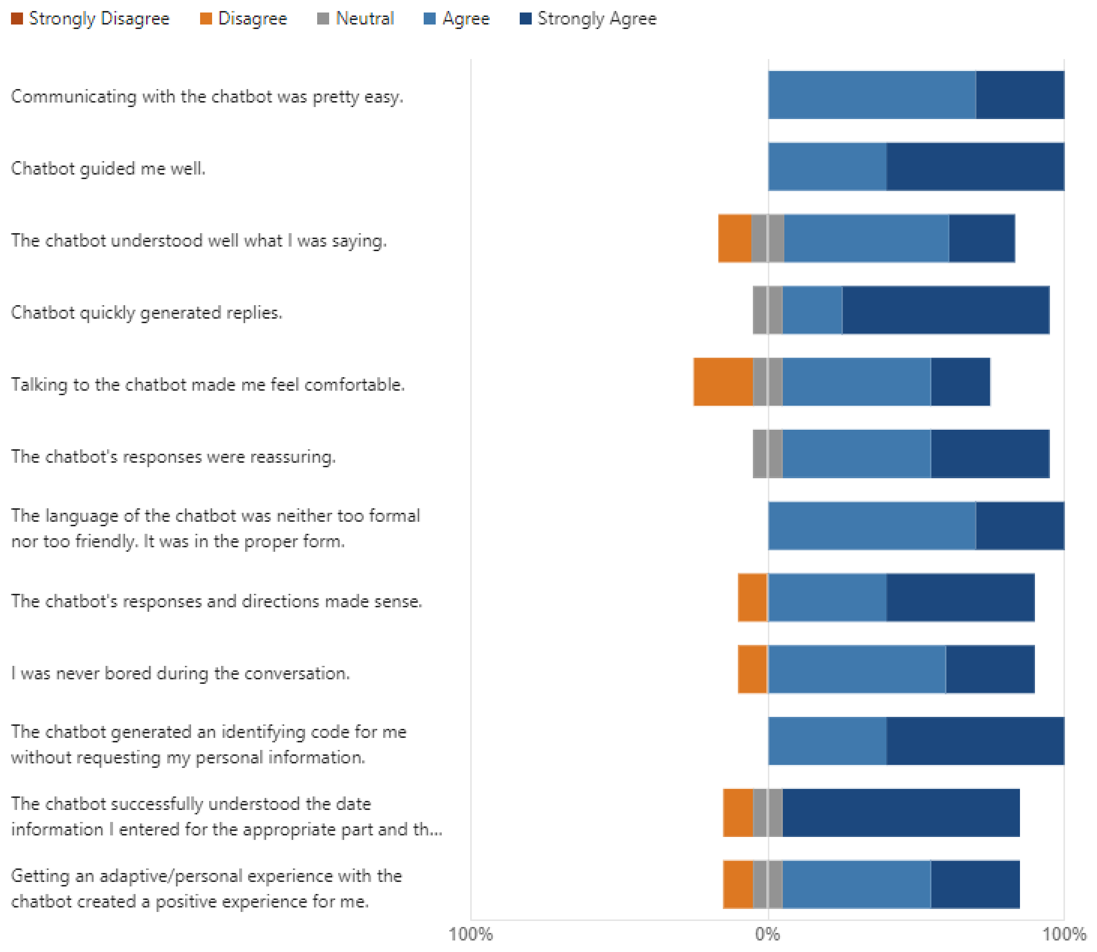


Figure 5.3: Results of the first part of the survey[2]

[1]The added expression is "The chatbot generated an identifying code for me without requesting my personal information".

[2]The 11th evaluation expression that cannot be read on the given figure is "The chatbot successfully understood the date information I entered for the appropriate part and the vehicle type/properties information".

After the information sheet and informed consent form, 12 different statements with positive statements were sent to the participants to evaluate their experiences. Afterward, 12 different expressions were conveyed in the next section. This time the expressions had a negative attitude. When interpreting the results, the two figures will be interpreted together and evaluated as a single output, but if the evaluation results in the figures differ highly for the related statements, the evaluation statement in both sections will be examined separately.

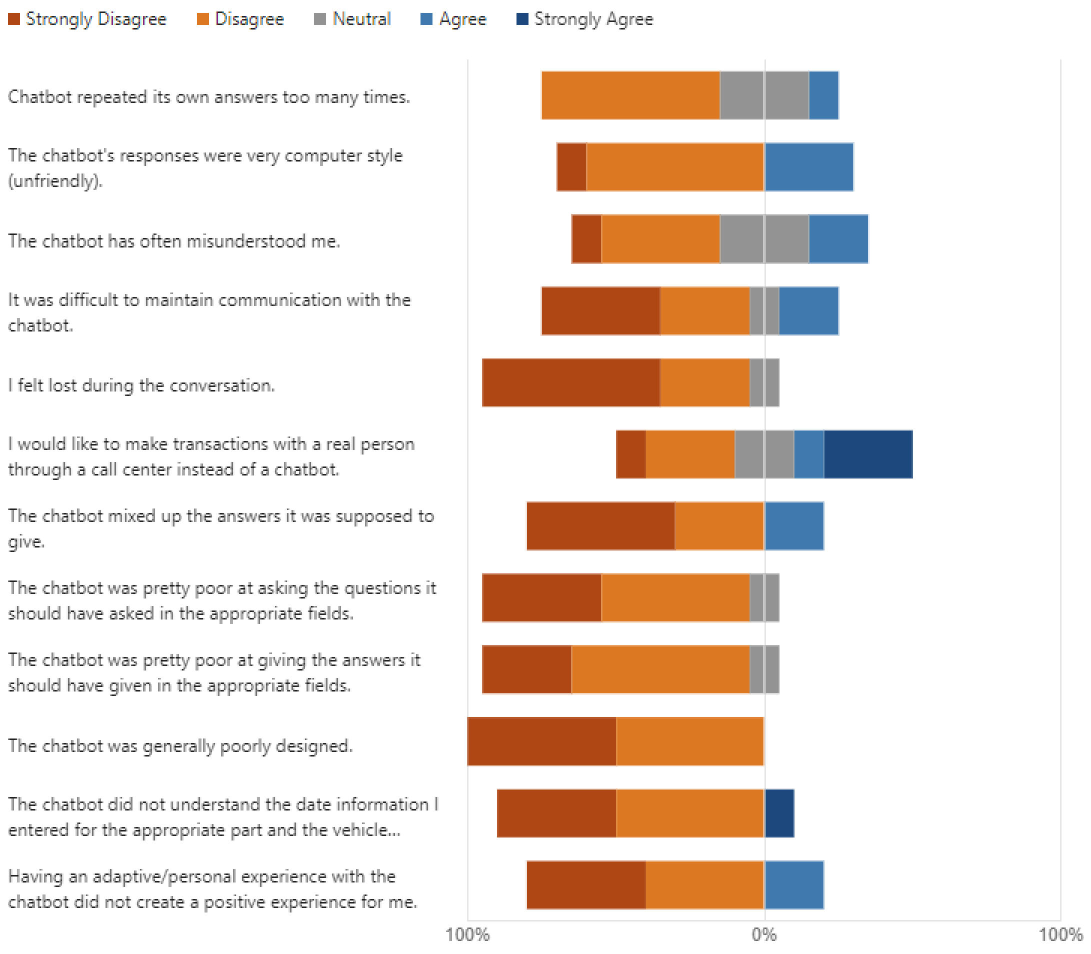


Figure 5.4: Result of the second part of the survey[3]

In Figure 5.3, all of the participants (70% Agree, 30% Strongly Agree) stated that communication with the chatbot is easy. This shows that the intended user-friendly conversation designs have positive effects on users.

While the chatbot was being designed, informative examples were shared between the conversations so that the users would be aware of the next step. As intended, all users

[3]The 11th evaluation expression that cannot be read on the given figure is "The chatbot did not understand the date information I entered for the appropriate part and the vehicle type/properties information".

(40% Agree and 60% Strongly Agree) stated that the chatbot guided them well. Likewise, these redirects are aimed at preventing users from getting lost during the conversation. Users say, "I felt lost during the conversation." The expression was evaluated as 60% Strongly Disagree, 30% Disagree, 10% Neutral.

About 80% of the users (1 participant skipped evaluating this expression) stated that the chatbot was able to successfully make sense of all the messages written. Although this indicates that the chatbot can make sense of user messages at a level that can be called good, it also shows that there are areas to be developed in this regard. A ready-made dataset was not used for the training of the chatbot presented in this thesis. The entire training set was designed by the developer. The approximately 80% ratio obtained also shows that the training data has areas that can be fed with richer and different content.

90% of the users stated that the chatbot generates their answers quickly. This is one of the critical factors in making the user experience more effective.

At least 70% of the users stated that talking with the chatbot made them feel comfortable. The reason why this rate is relatively low compared to other rates is that although the context of car rental seems to include conversations on a specific area, it actually has a structure of having too many customized smelling areas in itself. For example, users may be hesitant about how they can express this subject precisely because they are expected to type the vehicle features they want in an empty text box while specifying the vehicle they want to rent. On the other hand, waiting for the user to write a date in the car rental start and end date fields without presenting any date selection panel may have created an area that requires more control for them. These situations are deliberately kept in the chatbot at this stage. Since one of the aims of this project is to provide an effective model for the ability of the chatbot to make sense of user inputs in a personalized chat flow, users were asked to write their own expressions to achieve the aim of the project, even though it was known that more user-friendly structures could be offered for these parts.

At least 90% of the users found the answers generated by the chatbot to be reassuring, and likewise, 90% of the users found the prompts of the chatbot meaningful. These ratings are statements that also measure users' overall chat experience. The answers of the chatbot guide the user and are effective in the next reply of the user.

In part 1, 100% (70% Agree,30% Strongly Agree) of the users found the language used by the chatbot in the appropriate form. In part 2, 30% of the participants found the chatbot's language was evaluated as a computer style. While designing the chatbot, the language used is not intended to be too formal or too intimate.

90% of the users stated that they were not bored during the conversation. The quick answers offered by the chatbot and the efficient answers and directions by the users are effective in achieving this rate.

90% of the users (at least 80% of the users (80% Strongly Agree) in the first part of the survey) stated that the chatbot successfully understood the entered date information and fast access code information. In the second part, 90% of the users stated that the chatbot successfully understood the entered date and vehicle features information. This ratio shows that the work in the natural language understanding section can be considered successful. At this point, additional improvements can be made to the training data based on real user conversations to take the user experience to a higher level.

In the last statement, 80% of the respondents stated that an adaptive/personalized chat experience on chatbots created a positive experience for them. An in-depth interpretation is provided in the section 5.3.4.1 for this area.

By grouping the results, a more in-depth look at certain aspects can be provided. To indicate which expressions are grouped together, the expressions will be numbered in the order of their order in Figure 5.3, representing the first part, and Figure 5.4, representing the second part. When the second, sixth, seventh and eighth expressions in the first part and the second, fifth, seventh, eighth, and ninth expressions in the second part are analyzed as a group, it is seen that the participants are satisfied with the chatbot's guidance and responses. In the first part, 100% of the users think that the chatbot provides good guidance and that the language of the chatbot is in proper form. Conversely, in the second part, 30% of the users found the language of the chatbot in the computer style, and only 10% of the users were hesitant about giving the exact answers that the chatbot should give. The remaining users (90% of all users) stated that the answers and questions of the chatbot were presented in the relevant areas without any issues and in accordance with the content.

Similarly, in terms of the ease of use of the chatbot, we can consider the evaluation statements related to this field as a group. The first and fifth statements from the first group statements and the fourth and tenth statements from the second group statements are the parts where evaluation results can be interpreted about the usefulness and comfort of the chatbot. When we look at the results, it is seen that users think that it is easy to use the presented chatbot and maintain communication with the presented chatbot. 100% of the users stated that it is easy to maintain communication with the chatbot. This is one of the necessary parts to ensure user satisfaction.

The fourth and ninth expressions from the first group and the first, fourth, and tenth expressions from the second group can be used to evaluate the dynamic nature (generating fast and various response styles, providing a comfortable speaking experience, etc.) of the chatbot. The results show that users find the presented chatbot in a well-designed structure that produces fast responses and has different response styles.

As expected, the statement of the presented survey that differentiated the participants in the most different points, "I would like to make transactions with a real person through a call center instead of a chatbot." has been an expression. Users rated this expression as 10% Strongly Disagree, 30% Disagree, 20% Neutral, 10% Agree, and 30% Strongly Agree. Considering the results, the rate of agreeing with the statement and having an attitude against the statement (total 40%) seem to be equal, but the rate distributions differ within themselves. For example, there is a group of 10% who strongly disagree with the statement, while there is a group of 30% who strongly agree with the statement. This expression contains a more general judgment than other evaluation expressions and provides a general evaluation opportunity for users to add their previous chatbot experiences to the process. The reason for including this expression in the evaluation criteria is to learn users' attitudes towards completing their transactions via chat robot. When the survey results were examined in detail, it was seen that even the participants who found the chat robot presented in this thesis to be quite successful according to all the evaluation criteria chose to complete their transactions with a real human despite everything in this evaluation statement. One of the conclusions drawn from all evaluation experiment results is that the improvements and personal flows provided by the presented chatbot have a pretty positive effect on the user experience. However, this positive effect in this area did not have the same strong impact on changing the general habits of users. The general habits of the users are fed by the results of their experiences in every area of their lives. It seems necessary to improve the chatbot experiences offered in all areas (e-commerce, health sector, customer service, etc.) for users to choose to complete their transactions through a chatbot instead of meeting with a real person over the call center. This shows that chatbots are still not fully adopted by users in some areas. Making chatbots directly preferable for users over talking to a call centre requires a planned process in all areas that use chatbots. Hopefully, in the future, the chatbot experience will be satisfactory in all areas that users come into contact with, allowing users to choose to complete their transactions in contact with the chatbot instead of a real customer service worker.

As explained earlier(in section 5.3.3), the evaluation statement "The chatbot generated an identifying code for me without requesting my personal information" is only included in the positive section(first part of the questionnaire) and shows the ethical approach of the system. This defined code (fast access code) allows the user to be known by the system so that personalized content is offered on a per-user basis during the chat.

5.3.4.1 Discussion About Personalization

The 12th statement[4] in the first part of the survey and the 12th statement[5] in the 2nd part of the survey aim to measure the personalized chat experience offered by the chatbot and the user satisfaction generated by this experience.

Evaluating the results for both expressions together will provide a clearer interpretation of the results. For the first statement, 80% of the participants stated that the personalized chat experience offered by the chatbot was a positive experience for them. In the second statement, 80% of the participants did not agree with the idea that the personalized experience offered is a negative experience. While these results show that the results are consistent, providing a personalized experience on chatbots increases users' satisfaction and improves their experience.

5.4 Summary

The completed process for the intent classification evaluation shows that the chatbot has successfully carried out the intent classification stages. At the end of the questionnaire process, it can be seen that the task-oriented chatbot that provides a personalized chat experience satisfies the users. But one finding from the evaluation results is that users still prefer, in some cases, to complete their transactions by talking to real people instead of chatbots. For this, users should be able to test the chatbot experience in different areas, and users should be familiar with the approaches in this area. Therefore, the evaluation data obtained indicate that there are issues to focus on for future improvements and solutions.

[4]The added statement is "Getting an adaptive/personal experience with the chatbot created a positive experience for me".

[5]The added statement is "Having an adaptive/personal experience with the chatbot did not create a positive experience for me".

Chapter 6

Conclusion

This section will address the outcomes of the studies experienced and performed so far. It consists of three parts limitations, conclusion, and future work.

6.1 Limitations

Pre-made training datasets were not used in the project, so the project produced the training data itself. The difficulties of creating training data in the context of car rental were experienced during this project. The fact that the car rental processes are carried out in different procedures in each country has caused the training data to consist of more general information. Likewise, since not every vehicle brand and model is available in every country, the database of the vehicles to be rented had to be created on more popular vehicles. This limitation area also created areas to focus on in future studies.

Another limitation is the training durations and data sizes that increase over time during the repetitive completion of model trainings during the development process. Especially in the development process, trials of natural language processing processes with different components, testing different language models on the system have created model training processes that take a long time and require repetition from time to time. The use of high-speed and high-capacity clusters provided by the ADAPT Center has been helpful in tackling this challenge. In this way, simultaneous development and testing activities were carried out both in the local environment and in the cloud environment.

6.2 Conclusion

In this study, task-based chatbot development in the context of car rental has been completed, and various approaches have been applied throughout the development process in order to provide users with a personalized adaptive conversation experience. Since a previously prepared training dataset was not used while developing the chatbot, the study used a training dataset in the context of car rental, which it created itself. This report, which includes all the approaches applied to the development process of the chatbot, also includes detailed studies in the literature.

All the difficulties experienced during the development process of the chatbot are detailed in section 6.1, and the approaches applied in these difficulty areas are explained in the relevant sections throughout the report.

This completed study applied an evaluation process to evaluate the chatbot developed in addition to the chatbot development process. The study, which applied an evaluation for intent classification in the first part, started real user tests after successful outcomes. After preparing the appropriate test environments for real user tests, the evaluations of the users who tested the system were collected through the online survey created. Real user tests found the chatbot, which was developed in this study, mostly successful on the basis of the evaluation criteria presented. This result shows that the approaches applied in the project resulted in successful outputs. Survey results show that a personalized and adaptive conversation experience offered in chatbots contributes positively to users' overall experience.

6.3 Future Work

This completed work has developed the necessary user dialogs and possible scenarios while creating a personalized chat experience in the context of car rental. The operation of the car rental process worldwide and the procedures followed may differ from country to country. The study proposes a training data corpus creation for this area that includes much broader scenarios (for example, customized by country) as a possible field of study for future studies.

Independent of the developed chatbot within the evaluation criteria, in the evaluation statement[1] added to understand the general habits and trends of the users, 40% of the survey participants preferred chatting with a real person to complete the transactions over communication with the chatbot. This situation still shows that the general habits of some people are to do business in communicating with real people. Future research can investigate people's prejudices and reasons for chatbots before their development process, and get some ideas by completing extensive literature reviews about them, perhaps planning a survey. Future work can guide chatbot development and design processes by

[1]The specified evaluation expression is "I would like to make transactions with a real person through a call center instead of a chatbot".

tackling the factors that cause people to be hesitant and biased toward chatbots. Also, future work could develop chatbots in a much wider variety of scenarios that could provide a personalized chat experience while applying approaches that break users' prejudices about chatbots. While doing these, it can develop new approaches that will give users a chat experience as if they are chatting to a real person.

Bibliography

Adamopoulou, E. and Moussiades, L. (2020). An overview of chatbot technology. In Maglogiannis, I., Iliadis, L., and Pimenidis, E., editors, *Artificial Intelligence Applications and Innovations*, pages 373–383, Cham. Springer International Publishing.

Al-Rfou, R., Pickett, M., Snaider, J., Sung, Y.-h., Strope, B., and Kurzweil, R. (2016). Conversational contextual cues: The case of personalization and history for response ranking.

Amato, F., Casillo, M., Colace, F., Santo, M. D., Lombardi, M., and Santaniello, D. (2019). Chat: a cultural heritage adaptive tutor. In *2019 IEEE International Conference on Engineering, Technology and Education (TALE)*, pages 1–5.

Bootstrap (2022). Bootstrap. `https://getbootstrap.com`. Accessed: 2022-07-12.

Bordes, A., Boureau, Y.-L., and Weston, J. (2017). Learning end-to-end goal-oriented dialog. In *International Conference on Learning Representations*.

Brandtzaeg, P. B. and Følstad, A. (2017). Why people use chatbots. In Kompatsiaris, I., Cave, J., Satsiou, A., Carle, G., Passani, A., Kontopoulos, E., Diplaris, S., and McMillan, D., editors, *Internet Science*, pages 377–392, Cham. Springer International Publishing.

Budzianowski, P., Wen, T. H., Tseng, B.-H., Casanueva, I., Ultes, S., Ramadan, O., and Gašić, M. (2018). Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. pages 5016–5026.

Caldarini, G., Jaf, S., and McGarry, K. (2022). A literature survey of recent advances in chatbots. *Information*, 13(1):41.

Chen, D., Chen, H., Yang, Y., Lin, A., and Yu, Z. (2021). Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 3002–3017, Online. Association for Computational Linguistics.

Chou, T.-L. and Hsueh, Y.-L. (2019). A task-oriented chatbot based on lstm and reinforcement learning. In *Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval*, NLPIR 2019, page 87–91, New York, NY, USA. Association for Computing Machinery.

Coucke, A., Saade, A., Ball, A., Bluche, T., Caulier, A., Leroy, D., Doumouro, C., Gisselbrecht, T., Caltagirone, F., Lavril, T., Primet, M., and Dureau, J. (2018). Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces.

Elsholz, E., Chamberlain, J., and Kruschwitz, U. (2019). Exploring language style in chatbots to increase perceived product value and user engagement. pages 301–305.

Emojipedia (2022). Emojipedia. `https://emojipedia.org/`. Accessed: 2022-06-07.

Følstad, A. and Brandtzaeg, P. (2020). Users' experiences with chatbots: findings from a questionnaire study. *Quality and User Experience*, 5.

Google Cloud (2022). Google cloud: Cloud computing services. `https://cloud.google.com`. Accessed: 2022-07-12.

Holmes, S., Moorhead, A., Bond, R., Zheng, H., Coates, V., and Mctear, M. (2019). Usability testing of a healthcare chatbot: Can we use conventional methods to assess conversational user interfaces? In *Proceedings of the 31st European Conference on Cognitive Ergonomics*, ECCE 2019, page 207–214, New York, NY, USA. Association for Computing Machinery.

Ikemoto, Y., Asawavetvutt, V., Kuwabara, K., and Huang, H.-H. (2018). Conversation strategy of a chatbot for interactive recommendations. In Nguyen, N. T., Hoang, D. H., Hong, T.-P., Pham, H., and Trawiński, B., editors, *Intelligent Information and Database Systems*, pages 117–126, Cham. Springer International Publishing.

Kucherbaev, P., Bozzon, A., and Houben, G.-J. (2018). Human-aided bots. *IEEE Internet Computing*, 22(6):36–43.

Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., and Dolan, B. (2016). A persona-based neural conversation model.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, 140:1–55.

Ma, Z., Dou, Z., Zhu, Y., Zhong, H., and Wen, J.-R. (2021). One chatbot per person: Creating personalized chatbots based on implicit user profiles. pages 555–564.

Next Move Strategy Consulting (2020). Chatbot market in bfsi by component (solution and service), by platform type (web-based and mobile-based), by end user (bank, insurance companies, and others) by application (customer support, branding advertisement, data security compliance) - global opportunity analysis and industry forecast, 2020 – 2030. `https://www.nextmsc.com/report/chatbot-market-in-bfsi`. Accessed: 2022-06-16.

Nimavat, K. and Champaneria, T. (2017). Chatbots: An overview. types, architecture, tools and future possibilities.

Rasa Open Source Documentation (2022a). Introduction to rasa open source. `https://rasa.com/docs/rasa/`. Accessed: 2022-07-01.

Rasa Open Source Documentation (2022b). Rasa architecture - message handling. `https://rasa.com/docs/rasa/architecture/#message-handling`. Accessed: 2022-07-01.

Rasa Open Source Documentation (2022c). Rasa architecture overview. `https://rasa.com/docs/rasa/arch-overview`. Accessed: 2022-07-01.

Shumanov, M. and Johnson, L. (2020). Making conversations with chatbots more personalized. *Computers in Human Behavior*, 117:106627.

spaCy (2022). Industrial-strength natural language processing in python. `https://spacy.io`. Accessed: 2022-07-05.

spaCy Trained Pipelines (2022). English. `https://spacy.io/models/en`. Accessed: 2022-07-05.

Surendran, A., Murali, R., and Babu, R. K. R. (2020). Conversational ai - a retrieval based chatbot.

Tur, G., Hakkani-Tur, D., and Heck, L. (2011). What is left to be understood in atis? pages 19 – 24.

Vladova, G., Haase, J., Rüdian, L., and Pinkwart, N. (2019). Educational chatbot with learning avatar for personalization.

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45.

Xie, T., Yang, X., Lin, A. S., Wu, F., Hashimoto, K., Qu, J., Kang, Y. M., Yin, W., Wang, H., Yavuz, S., Wu, G., Jones, M., Socher, R., Zhou, Y., Liu, W., and Xiong, C. (2022). Converse: A tree-based modular task-oriented dialogue system.

Yeh, S.-F., Wu, M.-H., Chen, T.-Y., Lin, Y.-C., Chang, X., Chiang, Y.-H., and Chang, Y.-J. (2022). How to guide task-oriented chatbot users, and when: A mixed-methods study of combinations of chatbot guidance types and timings. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J. (2018). Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Appendix A

Use-Case

The chatbot presented in this project provides chat experiences terminated by users. Therefore, a large number of use-cases can be created for different scenarios. This report is based on basic flows and auxiliary flows.

Use Case Name	Chatbot guidance during the wrong date entry
Use Case Overview	This use case specifies the flow applied to the date field when the user is not typing the date information in a format that the chatbot can understand.
Actor	User
Trigger	The user tells the chatbot that he/she wants to rent a car.
Precondition 1	It is sufficient to trigger the use case. After it, the date must be entered incorrectly.
Basic Flow	The user implies that he/she wants to rent a car. Chatbot asks for date information. The user enters the date information incorrectly. Chatbot shows user the appropriate format of date information. The user enters the date information correctly and continues the rental process.
Alternate Flow	The user implies that he/she wants to rent a car. Chatbot asks for date information. The user enters the date information incorrectly. Chatbot shows user the appropriate format of date information. The user enters the date information incorrectly. The user enters the date information incorrectly again and the chatbot shows the correct format to the user with each incorrect entry.

Table A.1: Chatbot guidance during the wrong date entry

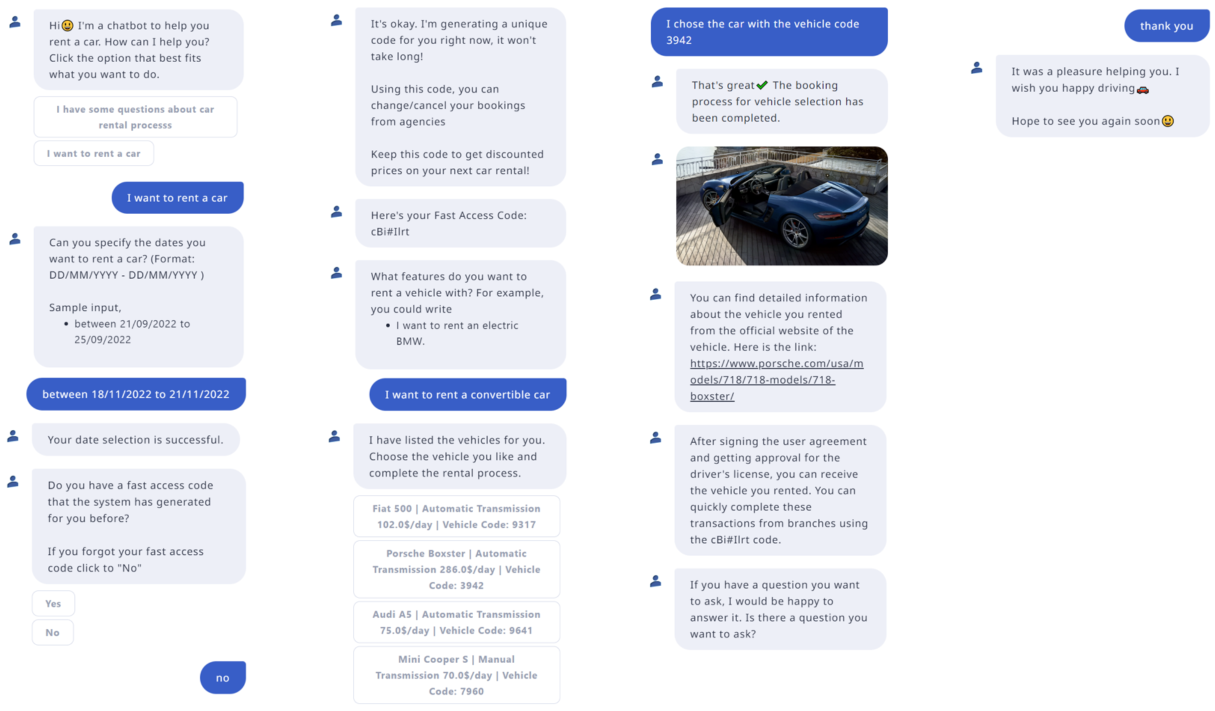
Use Case Name	The use-case representing the user's inability to fully explain the vehicle they want to rent
Use Case Overview	It represents the situations where the user cannot fully specify the features of the vehicle they want to rent.
Actor	User
Trigger	The user tells the chatbot that he wants to rent a car.
Precondition 1	It is sufficient to trigger the use case. After it, the user cannot specify the features of the vehicle they want to rent.
Basic Flow	The user implies that he/she wants to rent a car. Chatbot asks for date information. User indicates dates to rent a car. Chatbot asks the user if he/she has a fast access code. User states that he/she has the code. The chatbot requests the user's fac. The user writes the fac. The chatbot asks the user what features a vehicle they want. The user cannot successfully write the features of the vehicle they want to rent. The chatbot shows the user the basic vehicle categories and vehicle features. The user clicks on one of the suggestions. The chatbot lists the vehicles that the user wants. The user completes the rental process for the vehicle selected from the list and leaves the chat.
Alternate Flow	*This process basically consists of a single flow.

Table A.2: The use-case representing the user's inability to fully explain the vehicle they want to rent

Appendix B

Dialogues

This section contains screenshots of examples of the chatbot's basic conversation flows. In a real scenario, these flows can be directed and differentiated in many different ways.



Hi😃 I'm a chatbot to help you rent a car. How can I help you? Click the option that best fits what you want to do.

I have some questions about car rental processs

I want to rent a car

I want to rent a car

Can you specify the dates you want to rent a car? (Format: DD/MM/YYYY - DD/MM/YYYY)

Sample input,
- between 21/09/2022 to 25/09/2022

between 18/11/2022 to 21/11/2022

Your date selection is successful.

Do you have a fast access code that the system has generated for you before?

If you forgot your fast access code click to "No"

Yes

No

no

It's okay. I'm generating a unique code for you right now, it won't take long!

Using this code, you can change/cancel your bookings from agencies

Keep this code to get discounted prices on your next car rental!

Here's your Fast Access Code: cBi#Ilrt

What features do you want to rent a vehicle with? For example, you could write
- I want to rent an electric BMW.

I want to rent a convertible car

I have listed the vehicles for you. Choose the vehicle you like and complete the rental process.

Fiat 500 | Automatic Transmission 102.0$/day | Vehicle Code: 9317

Porsche Boxster | Automatic Transmission 286.0$/day | Vehicle Code: 3942

Audi A5 | Automatic Transmission 75.0$/day | Vehicle Code: 9641

Mini Cooper S | Manual Transmission 70.0$/day | Vehicle Code: 7960

I chose the car with the vehicle code 3942

That's great✔ The booking process for vehicle selection has been completed.

You can find detailed information about the vehicle you rented from the official website of the vehicle. Here is the link: https://www.porsche.com/usa/models/718/718-models/718-boxster/

After signing the user agreement and getting approval for the driver's license, you can receive the vehicle you rented. You can quickly complete these transactions from branches using the cBi#Ilrt code.

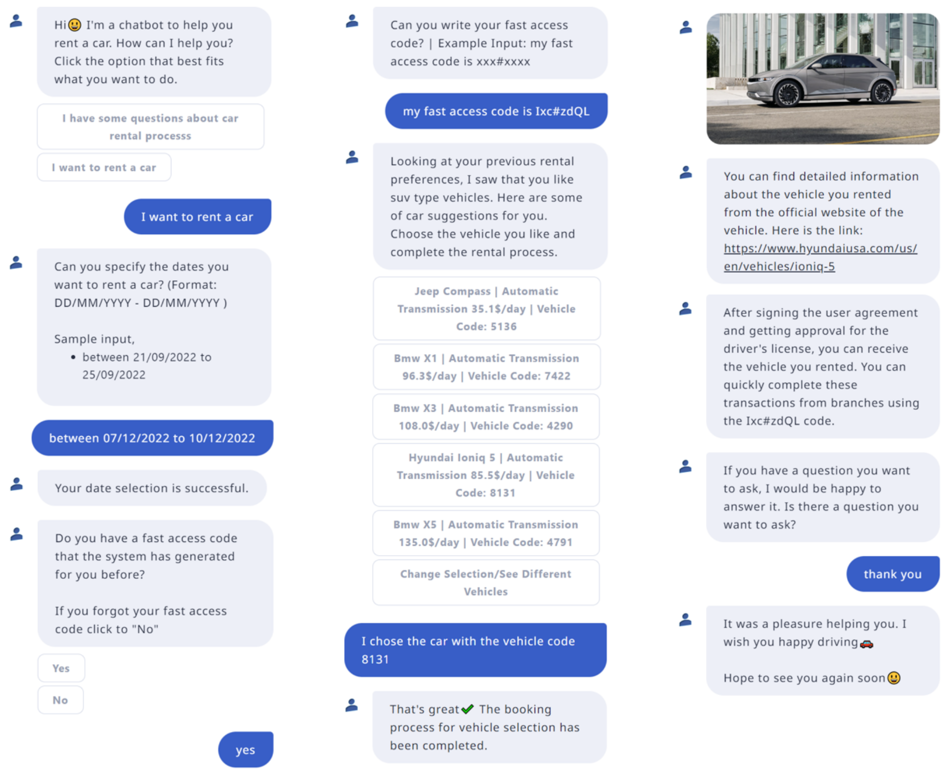
If you have a question you want to ask, I would be happy to answer it. Is there a question you want to ask?

thank you

It was a pleasure helping you. I wish you happy driving🚗

Hope to see you again soon😃

Figure B.1: Conversation flow for basic dialogues(without fast access code at the initial stage)



Hi😃 I'm a chatbot to help you rent a car. How can I help you? Click the option that best fits what you want to do.

I have some questions about car rental processs

I want to rent a car

I want to rent a car

Can you specify the dates you want to rent a car? (Format: DD/MM/YYYY - DD/MM/YYYY)

Sample input,
- between 21/09/2022 to 25/09/2022

between 07/12/2022 to 10/12/2022

Your date selection is successful.

Do you have a fast access code that the system has generated for you before?

If you forgot your fast access code click to "No"

Yes

No

yes

Can you write your fast access code? | Example Input: my fast access code is xxx#xxxx

my fast access code is Ixc#zdQL

Looking at your previous rental preferences, I saw that you like suv type vehicles. Here are some of car suggestions for you. Choose the vehicle you like and complete the rental process.

Jeep Compass | Automatic Transmission 35.1$/day | Vehicle Code: 5136

Bmw X1 | Automatic Transmission 96.3$/day | Vehicle Code: 7422

Bmw X3 | Automatic Transmission 108.0$/day | Vehicle Code: 4290

Hyundai Ioniq 5 | Automatic Transmission 85.5$/day | Vehicle Code: 8131

Bmw X5 | Automatic Transmission 135.0$/day | Vehicle Code: 4791

Change Selection/See Different Vehicles

I chose the car with the vehicle code 8131

That's great✔ The booking process for vehicle selection has been completed.

You can find detailed information about the vehicle you rented from the official website of the vehicle. Here is the link: https://www.hyundaiusa.com/us/en/vehicles/ioniq-5

After signing the user agreement and getting approval for the driver's license, you can receive the vehicle you rented. You can quickly complete these transactions from branches using the Ixc#zdQL code.

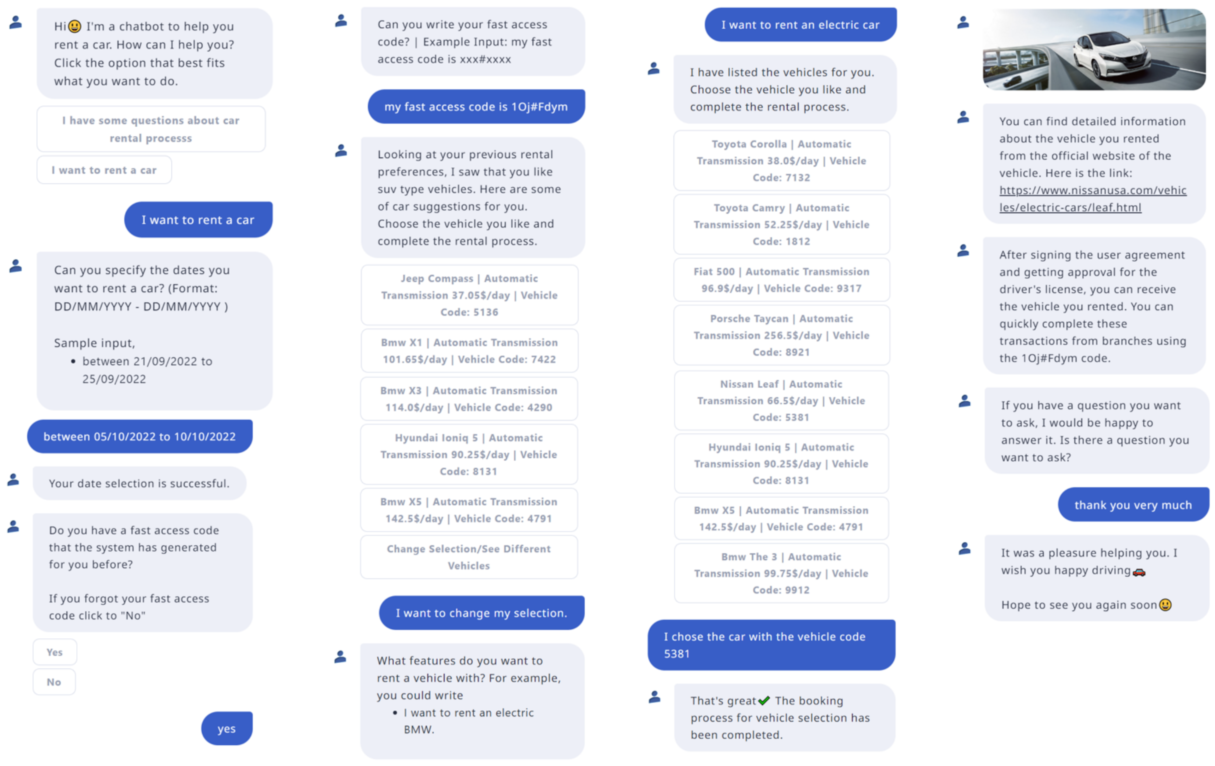
If you have a question you want to ask, I would be happy to answer it. Is there a question you want to ask?

thank you

It was a pleasure helping you. I wish you happy driving🚗

Hope to see you again soon😃

Figure B.2: Conversation flow for basic dialogues(with fast access code at the initial stage)



Hi😃 I'm a chatbot to help you rent a car. How can I help you? Click the option that best fits what you want to do.

I have some questions about car rental processs

I want to rent a car

I want to rent a car

Can you specify the dates you want to rent a car? (Format: DD/MM/YYYY - DD/MM/YYYY)

Sample input,
- between 21/09/2022 to 25/09/2022

between 05/10/2022 to 10/10/2022

Your date selection is successful.

Do you have a fast access code that the system has generated for you before?

If you forgot your fast access code click to "No"

Yes

No

yes

Can you write your fast access code? | Example Input: my fast access code is xxx#xxxx

my fast access code is 1Oj#Fdym

Looking at your previous rental preferences, I saw that you like suv type vehicles. Here are some of car suggestions for you. Choose the vehicle you like and complete the rental process.

Jeep Compass | Automatic Transmission 37.05$/day | Vehicle Code: 5136

Bmw X1 | Automatic Transmission 101.65$/day | Vehicle Code: 7422

Bmw X3 | Automatic Transmission 114.0$/day | Vehicle Code: 4290

Hyundai Ioniq 5 | Automatic Transmission 90.25$/day | Vehicle Code: 8131

Bmw X5 | Automatic Transmission 142.5$/day | Vehicle Code: 4791

Change Selection/See Different Vehicles

I want to change my selection.

What features do you want to rent a vehicle with? For example, you could write
- I want to rent an electric BMW.

I want to rent an electric car

I have listed the vehicles for you. Choose the vehicle you like and complete the rental process.

Toyota Corolla | Automatic Transmission 38.0$/day | Vehicle Code: 7132

Toyota Camry | Automatic Transmission 52.25$/day | Vehicle Code: 1812

Fiat 500 | Automatic Transmission 96.9$/day | Vehicle Code: 9317

Porsche Taycan | Automatic Transmission 256.5$/day | Vehicle Code: 8921

Nissan Leaf | Automatic Transmission 66.5$/day | Vehicle Code: 5381

Hyundai Ioniq 5 | Automatic Transmission 90.25$/day | Vehicle Code: 8131

Bmw X5 | Automatic Transmission 142.5$/day | Vehicle Code: 4791

Bmw The 3 | Automatic Transmission 99.75$/day | Vehicle Code: 9912

I chose the car with the vehicle code 5381

That's great✔ The booking process for vehicle selection has been completed.

You can find detailed information about the vehicle you rented from the official website of the vehicle. Here is the link: https://www.nissanusa.com/vehicles/electric-cars/leaf.html

After signing the user agreement and getting approval for the driver's license, you can receive the vehicle you rented. You can quickly complete these transactions from branches using the 1Oj#Fdym code.

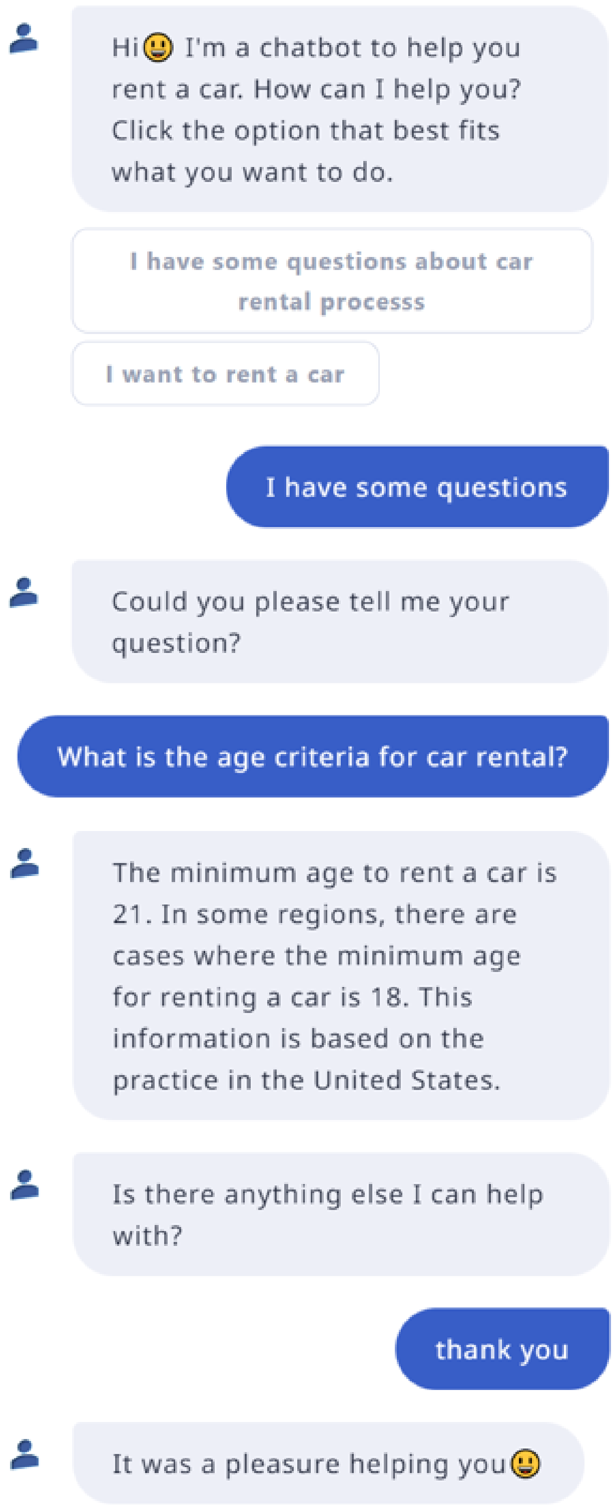
If you have a question you want to ask, I would be happy to answer it. Is there a question you want to ask?

thank you very much

It was a pleasure helping you. I wish you happy driving🚗

Hope to see you again soon😃

Figure B.3: Basic conversation flow of a user who wants to rent a vehicle different from the recommended vehicles after fast access code entry



Hi😃 I'm a chatbot to help you rent a car. How can I help you? Click the option that best fits what you want to do.

I have some questions

Could you please tell me your question?

What is the age criteria for car rental?

The minimum age to rent a car is 21. In some regions, there are cases where the minimum age for renting a car is 18. This information is based on the practice in the United States.

Is there anything else I can help with?

thank you

It was a pleasure helping you😃

Figure B.4: A short dialogue about one of the frequently asked questions