

# **COMP3074 Report**

JESAL VADGAMA, The University of Nottingham, United Kingdom

## CONTENTS

Contents	1
1 Introduction	2
2 Chatbot Architecture	2
2.1 How Data is Stored	2
2.2 How Chatbot can Predict Intent	2
2.3 Identity Management	3
2.4 Transactional	3
2.5 Information Question and Answering	4
2.6 Small Talk	5
3 Conversational Design	5
3.1 Prompt Design	5
3.2 Discoverability	6
3.3 Error Handling	6
3.4 Personalisation	6
3.5 Confirmation	6
3.6 Context Tracking	7
4 Testing	7
4.1 Performance Testing	7
4.2 Usability Testing	9
5 Discussion of Responsible Research and Innovation Framework	9
References	9

## 1 INTRODUCTION

My chatbot is a NLP movie booking chatbot which helps users book, cancel and show you a list of movies available even via genre. It can provide descriptions for each movie and show you your bookings. It creates TF-IDF vectors and uses cosine similarity to predict intent and return accurate outputs. The chatbot functions are all defined with chatbot\_functions and some of them are called when intent is matched and a variable name is in response. If user inputs exit, the program breaks.

## 2 CHATBOT ARCHITECTURE

### 2.1 How Data is Stored

Data is stored in SQL databases such as movie\_database.db and bookings.db. Data used to train the chatbot is stored in dataset.csv. I created a function called create\_table which takes three parameters: cursor, table\_name, and fields which converts a list of fields into a single string separated by a comma for retrieving items in the database. A table is created if it does not already exist which stops errors when reading of the databases. The insert\_movie\_data function uses the cursor and movie parameters to insert movies into the movie\_database by iterating over a list of movies represented by a tuple. Duplicate movie insertions are checked so movies with the same name cannot be added to the movie\_database. The retrieve\_movies function uses cursor to SELECT all data from the movie table and uses fetchall() to retrieve all the rows from the query as tuples, if there is missing data, it returns an empty list.

Sqlite3 is used to create a connection with movie\_database.db and bookings.db. The movie\_database.db takes three parameters: cursor, movies, and a list of fields and uses cursor to execute commands within the database. If movie\_database.db and bookings.db does not exist, it creates it using the create\_table function. The fields I have used for my movie\_database is shown in Figure1. A list of tuples named movies is defined representing a movie and its details matching the fields in Figure1. The insert\_movie\_data function will iterate over the movie list, inserting each movie into the table and checks duplicate movies don't get added. The bookings.db takes three parameters: cursor\_bookings, bookings, and a list of fields and uses cursor\_bookings to execute commands within the database. The fields I have for my bookings.db is shown in Figure1 and this function has been created to store, update, insert, delete booking information. These functions have been created to store all movie details in retrievable and readable format. A csv.reader is used to parse dataset.csv, where each row from dataset.csv is iteratively added to a list called data. The dataset.csv has two columns Intent which stores phrases or words that a user might input and Response stores outputs or functions the chatbot can output (Figure1).

id	title	genre	release_year	director	show_time		Intent	Column
1	The Godfather	Crime	1972	Francis Ford Coppola	10:00,14:00,18:00		hi,Hello!	How can I assist you today?
2	The Shawshank Redemption	Drama	1994	Frank Darabont	11:00,15:00,19:00		hey,Hello!	How can I assist you today?
3	Titanic	Romance	1997	James Cameron	12:00,16:00,20:00		hello,Hello!	How can I assist you today?
4	The Dark Knight	Action	2008	Christopher Nolan	13:00,17:00,21:00		what can you do,what_can_you_do	
5	Avatar	Action	2009	James Cameron	10:30,14:30,18:30		can i book movie,book_movie	
6	Inception	Sci-Fi	2010	Christopher Nolan	11:30,15:30,19:30		can i book film,book_movie	
7	Interstellar	Sci-Fi	2014	Christopher Nolan	12:30,16:30,20:30		book movie,book_movie	
8	The Menu	Comedy	2022	Mark Mylod	13:30,17:30,21:30		book film,book_movie	
9	Spider-Man: Across the Spider-Verse	Animation	2023	Joaquim Dos Santos	10:00,14:00,18:00		how are you,I am great thank you	
10	Hunger Games: The Ballad of Songbirds & Snakes	Adventure	2023	Francis Lawrence	11:00,15:00,19:00		i am good,Thats glad to hear, How can i help you today	
11	Wonka	Musical	2023	Paul King	12:00,16:00,20:00		how it going,I am awesome thank you	
12	Napoleon	Biography	2023	Ridley Scott	13:00,17:00,21:00		show me all movies,show_movies	
13	wish	Animation	2023	Chris Buck	10:30,14:30,18:30		show me all films,show_movies	
14	Godzilla Minus One	Action	2023	Takashi Yamazaki	11:30,15:30,19:30			

id	booking_id	user_name	movie	date	time	tickets	
1	b60edf285c9749b429f37180dd58065	Jesal	Wish		-	-	
2	booking_e66d9fc15654b02085af1a73475f19	Jesal	Wonka	2027-12-12	18:30	7	
3	b3c0d74dc644de9e90c1f12200e053d	Jesal	Wonka	2027-12-12	12:00	10	
4	booking_1c01e2a798744bc79cf64397204c5730	Jesal	Wish	2029-12-12	14:30	9	
5	booking_1854445edefab4362897474fab65c48dc	Jesal	Hunger Games: The Ballad of Songbirds & Snakes	2023-12-17	11:00	6	

Figure 1: Shows how Data is Stored

### 2.2 How Chatbot can Predict Intent

The TfidfVectorizer is instantiated as movie\_tfidf\_vectorizer and intent\_tfidf\_vectorizer to create TF-IDF vectors for movie and intent matching. Title and director are retrieved from movie\_database.db using retrieve\_movies(cursor).

The intent\_texts list is created by using intent from each row in data from dataset.csv. The fit\_transform method has been used for movie\_texts and intent\_texts to create TF-IDF vectors stored as movie\_vectors and intent\_vectors. These vectors can be used in a cosine similarity to predict the movie and user intent based on their input to output correct responses

The preprocess function cleans user\_input by converting it to lowercase, removing extra space and non-alphanumeric characters. It then tokenizes the user input, performs POS tagging, and removes stopwords, except for ‘what, show, your etc’ which I keep. The tokens are then lemmatized and joined back into a string. This returned in a clean version of the users input returning processed\_input.

The predict\_intent function checks if the cosine\_similarities is > 0.5, if so, it outputs a response for the best\_match\_index from data, passing it to chatbot\_functions along with user\_name. If < 0.5, it checks if the processed\_input exactly matches intent\_texts. If so, it returns a matching response from data and if there is no matching response it returns, I don’t understand your request. This function is created to predict accurate intents of the user input.

The find\_movie function calculates cosine similarities between the user\_vector and the movie\_vectors. It preprocesses user\_input and transforms it into user\_vector so the user can book a movie without needing to type the exact movie. If the cosine\_similarities is > 0.5, if so, it returns the matching movie(movie\_match) from the movie\_database and if < 0.5, it returns not\_found.

The get\_response function takes the user\_input and user\_name as arguments. The user\_input is preprocessed and transformed into a TF-IDF vector using intent\_tfidf\_vectorizer. The function uses cosine similarities for the user\_vector and the intent\_vector. The index of intent\_vector that has the greatest cosine similarity with the user\_vector is called best\_match\_index. The function calls process\_intent with the cosine\_similarities, best\_match\_index, processed\_input, intent\_texts, and user’s name as arguments and returns a response based on user input.

### 2.3 Identity Management

Identity management is used at the beginning of the chatbot via username and is used in multiple functions, asking the user to enter a name and auto capitalises every word within a while loop. If name is correct, the chatbot continues and If name is incorrect the user to enter a name again till correct. The change\_name function can update user\_name by asking the user to enter a new name, If it’s correct new\_name = user\_name breaking the loop. If incorrect it will continue to ask for a name till it gets it correct. If invalid input is entered, it will ask the user to enter only yes or no. The user can ask what is my name which outputs your name is user\_name which works when user\_name is updated.

### 2.4 Transactional

The chatbot\_state is set to false at the beginning of every function. If no is entered in any of these functions, chatbot\_state = True, stopping the current booking process and break the loop. These functions return chatbot\_state: movie, date, time and tickets. choose\_movie initialises movie as an empty string and the user can input a movie name within a while loop. If a movie is inputted, find\_movie searches for the closest movie match. If a movie is found, it confirms the selection, stores the movie, and breaks the loop. choose\_date function initialises date as an empty string and allows the user to enter a date in the YYYY-MM-DD format within a while loop. If a valid date is entered, it confirms the selection, stores the date, and breaks the loop. choose\_time function takes movie as an argument, initialises time as an empty string and show\_times is retrieved and selected for a movie. Available show times is returned by splits show\_time in the movie\_database which is stored as a comma separated string into individual showtimes. The user to enters a time within a while loop, and If the user selects a valid time, it confirms the selection, stores the time, and breaks the loop. choose\_tickets initialise tickets as an empty string,

getting the user to enter the total number of tickets within a while loop. If the user enters a number  $\leq 10$ , It confirms the selection, stores the number of tickets, and breaks the loop.

The book\_movie function all the above functions together and chosen\_movie = movie. If the chosen movie is not selected with a valid movie, it will display the current movies and ask the user to choose a movie. If a valid movie is chosen, it skips choose\_movie and continues with asking the user to enter a date using choose\_date, time using choose\_time and number of tickets using choose\_tickets. Once a valid movie, date, time, and tickets are returned from each function, get\_confirmation will ask the user to confirm booking, if not the booking process will be stopped. If confirmed check booking\_id will be created using uuid, then it's shown to the user and will return the booking made by the user with the booking details which is inserted bookings.db.

The cancel\_booking function guides the user through the cancellation process and If the user doesn't know their booking\_id, they're asked to get it first. The function then asks the user to enter the matching booking\_id and user\_name. It looks for a matching booking and if a match is found, the user is asked to confirm the cancellation. The booking is deleted and the user is told this. If the user says no, the cancellation process is stopped and If no match is found, the user is asked to start the process again.

The show\_booking function uses a while loop asking the user if they know their booking\_id. If no is entered, the find\_booking\_id function will be returned to help find their booking\_id and If cancel is entered, the loop breaks. If yes is entered, chatbot will ask for a booking\_id and the matching name, where SQL commands will be used to find the booking. Booking details will be outputted if booking is found and if its not found, it will return find\_booking\_id. If there is an invalid input the loop will not break, till either yes, no or cancel is entered.

The find\_your\_booking\_id function asks the user to enter their user\_name used to book the movie which is stored in user\_name. If no is entered the chatbot\_state will return true, and a custom error message. The user is asked to enter the movie booked whilst making the booking and the choose\_movie is called. It then asks the user to enter the day you booked the movie and the choose\_date is called. The user is asked what time did you enter whilst making your booking and choose\_time(movie) is called. Then bookings.db is search with a booking that matches the user\_name, movie, date and time values returned by the user. If booking is found it will return booking details, if not, it will ask the user to try again.

The edit\_booking function will ask the user to choose from three options: change\_date, change\_time, change\_tickets. These functions will ask the user to choose a new date, time ticket, then it will call choose\_date, choose\_time, choose\_tickets and return date, time tickets and chatbot\_state. These will then be passed to the update\_booking and the field in bookings.db will be updated with the new date, time and tickets and return confirmation of the update.

```
Please enter your name: Jesal
Is Jesal correct? (yes/no): yes
Name: my booking
Bot: What time did you enter whilst making your booking? Jesal
Bot: What movie did you enter whilst making your book (Type no to cancel)?
You: wonka
Bot: Jesal You have selected the movie Wonka
Bot: Please enter a day you booked the movie (in YYYY-MM-DD format)?
You: 2027-12-12
Bot: What time did you choose to book a movie on 2027-12-12
Bot: What time did you enter whilst making your booking?
Bot: The available show times for Wonka (2023) are 12:00, 16:00, 20:00.
You: 12:00
Bot: Jesal you have selected 12:00
Bot: This is your Booking ID: ('booking_ec6de8fc5684b93985e1fa32475f7f9')
Bot: Booking Details: Wonka for 12:00 on 2027-12-12 with a ticket total of 10

Please enter your name: jesal vadgama
Is Jesal vadgama correct? (yes/no): yes
You: cancel my booking
Bot: If you do not know booking Id, stop this process and ask find booking Id
You: cancel my booking
Bot: Please enter a booking id: booking_ac4c19a9340d8808d62693810df49
Bot: Please enter your name: Jesal Vadgama
Bot: Booking Details: The Dark Knight for 21:00 on 2023-12-15 with a ticket total of 2
Bot: Jesal Vadgama, do you want to cancel this booking? (yes/no): yes
Bot: Jesal Vadgama, you have successfully cancelled your booking for The Dark Knight for 21:00 on 2023-12-15 with a ticket total of 2
You: []

Please enter your name: jesal vadgama
Is Jesal Vadgama correct? (yes/no): yes
You: show my booking
Bot: Do you know your Booking ID? (yes/no/cancel): yes
Bot: Please enter your booking id: booking_6c0161bc088a4299a5831d5f8f8995a6ad
Bot: Please: Enter your name: Jesal Vadgama
Bot: Booking Details: The Dark Knight for 21:00 on 2023-12-15 with a ticket total of 1
You: []
```

Figure 2: Shows Transactional Features

## 2.5 Information Question and Answering

The show\_genre function gets the user to enter a genre within a while loop, if no is entered the process is stopped and the loop breaks. If a valid genre is entered, it will use an SQL command to find the movies based on the chosen genre and return those movies. This function has been created so the user can filter out what genre of movie they would like to book.

The show\_name function will return the current user\_name entered the chatbot. This function is called when the user inputs what is my name or has a similar intent.

The show\_movie function gets a list of movies from the movie\_database using retrieve\_movies(cursor), formats each movie, year and director into a string and returns the list. This is so the user can see all movies available so that the user can try book a movie in the chatbot.

```

Please enter your name: jesal vadgama
Is Jesal Vadgama correct? (yes/no): yes
You: show me a list of available movies
Bot: The Godfather (1972) - Francis Ford Coppola
The Shawshank Redemption (1994) - Frank Darabont
Titanic (1997) - James Cameron
The Dark Knight (2008) - Christopher Nolan
Avatar (2009) - James Cameron
Inception (2010) - Christopher Nolan
Interstellar (2014) - Christopher Nolan
The Menu (2022) - Mark Mylod
Spider-Man: Across the Spider-Verse (2023) - Joaquin Dos Santos
Maze Runner: The Death Cure (2018) - Wes Ball
Songbirds & Snakes (2023) - Francis Lawrence
Wooka (2023) - Paul King
Napoleon (2023) - Ridley Scott
Wish (2023) - Chris Buck
Godzilla Minus One (2023) - Takashi Yamazaki
Maze Runner: The Death Cure (2018) - Wes Ball
Songbirds & Snakes (2023) - Francis Lawrence
You: []

Please enter your name: jesal vadgama
Is Jesal Vadgama correct? (yes/no): yes
You: show me the genres for the movies
Bot: The Godfather (1972) - Crime, Drama, Romance, Action, Sci-Fi, Comedy, Animation, Adventure, Musical, Biography
The Dark Knight (2008) - Christopher Nolan
Avatar (2009) - James Cameron
Godzilla Minus One (2023) - Takashi Yamazaki
Bot: These are movies of your chosen genre: action, Jesal Vadgama
You: []

```

Figure 3: Shows Information and question answering

## 2.6 Small Talk

These is small talk features in my chatbot and can take in greetings such as hi, hello etc from dataset.csv. It asks about the well being of the chatbot and the user can also enter how they are feeling and it understands what the user has entered and outputs a response. It can tell a joke, and the user can ask about the weather but it can't provide real time weather updates. The chatbot even has a favourite colour which is blue and can't recommend films but prompts the user to try ask the chatbot to show all films available. The small talk is designed to engage the user and create a more human like conversational experience.

```

Please enter your name: jesal vadgama
Is Jesal Vadgama correct? (yes/no): yes
You: hi
Bot: Hello! How can I assist you today?
You: how are you
Bot: I am great thank you
You: i am good too
Bot: Thats glad to hear. How can i help you today
You: tell a joke
Bot: Okk! Why don't scientists trust atoms? Because they make up everything!
You: nice
Bot: thanks
You: what is the weather like today
Bot: I can't give real-time weather updates. Try a reliable weather forecasting website or app.
You: favourite colour
Bot: Its blue
You: recommend a film
Bot: I can't give movie recommendations movies but i can show all films available
You: []

```

Figure 4: Shows small talk

## 3 CONVERSATIONAL DESIGN

### 3.1 Prompt Design

Conversational markers such as the chatbot will output what the user has selected when making a movie booking, for example in figure 8, you can see the functions outputting the user\_name heavily in book\_movie. It uses conversational markers when showing the movies by outputting user\_name these are the available movies. The edit\_booking function also outputs what you have selected so the user is informed of the changes made. These markers are concise and clear so that the user knows what there doing when using the chatbot.

### 3.2 Discoverability

If the user enters ‘what can you do’, It will give a list of tasks the user can do when using the chatbot(figure5) Contextual help is provided when a user doesn’t know their booking\_id and is inside a function such as cancel\_booking and show\_booking where a booking id is required, it prompts the user to exit and find\_booking\_id. When the user enters a movie name into the chatbot, it stores the movie the user has chosen in movie and calls the book\_movie function. It will skip the first step of book movie which is choose\_movie as movie has been found and it will continue to book the movie. If thank you has been entered, it will output ‘would you like to make another book’, if yes, it will continue to run, if no, it will restart from the beginning. Another discoverability feature is in context tracking(figure7)

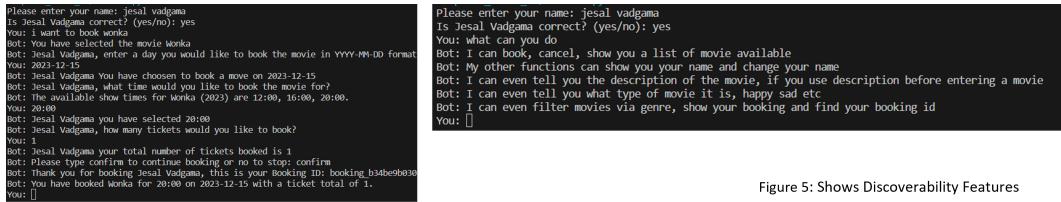


Figure 5: Shows Discoverability Features

### 3.3 Error Handling

Input validation techniques are used to handle errors and invalid user inputs. When the movie is not found in choose\_movie, the while loop repeats until a valid movie is found. The choose\_date can only accept a future date and uses datetime to compare the user\_input with the current date so an invalid input is not entered. It repeats and provides error messages until the user enters a valid date. The choose\_time function can only accept a time which exactly matches the format off the available showtimes, if not, the function will repeat till a valid response is entered. The choose\_tickets function ensures a ticket number >10 can’t be entered and does not enter an invalid input. If there is an input >10, the user is given an error message, and this loops till the user enters a valid input. The get\_confirmation function asks for a booking confirmation and only accepts confirm or no otherwise it loops till a valid input is entered. If genre is not found, a valid genre must be entered to break the loop or no must be entered. In cancel\_booking if the user\_input is invalid it outputs please enter yes and no again. These error handling strategies are used effectively so valid inputs are returned. It uses while loops in each function to validate inputs and can be stopped at any time using no. Functions in book\_movie stops invalid data being entered and output error messages to guide the user towards correct inputs.

### 3.4 Personalisation

The user\_name function helps personalise the chatbot so the user feel for comfortable with using A.I. The user can choose a movie based on genre preferences. The chatbot allows for the user to edit movie bookings and user\_name is called multiple times when booking a movie so that the user understand the processes of booking a movie. The chatbot even allows for the user to change its name making the interaction more personal.

### 3.5 Confirmation

Confirmation will be asked at the end of book\_movie so that if the user is satisfied with the choose booking, entering confirm will book it. When the chatbot starts the user will be asked for its name and the confirm if the name is correct, if not, the user can enter the name again. Confirm is needed when the matching booking has been found for the user and it is ready to delete this from bookings.db. This is done so the user doesn’t accidentally make and rush decisions and checks if the booking meets the users’ requirements.

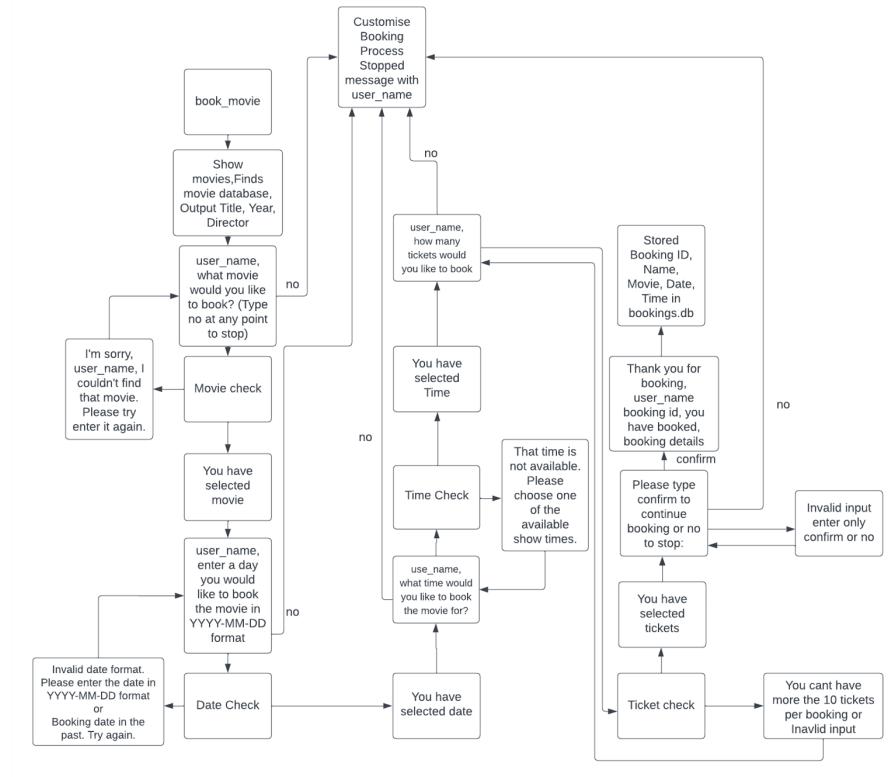


Figure 6: Book a movie flowchart

Fig. 1. Book a movie flowchart

### 3.6 Context Tracking

Context tracking is used in the get response function where the function `last_movie` keeps track of the last movie entered in the conversation. If the user input contains the word `description` with a movie name, the chatbot would assume the user wants a description of movie and outputs the movie data for the chosen movie. If the user input contains the word `movie` or `film` and any of the list keywords, the function checks the genre of the `last_movie` and returns a response on what feeling associates with the genre of the chosen movie(figure7). Otherwise, it would then assume the user is trying to book a movie and updates the `last_movie` function which starts the booking process. Figure7 shows the user has entered two movies and keeps track of the context effectively using `last_movie` function.

## 4 TESTING

### 4.1 Performance Testing

Figure8 shows the performance tests where `no` is entered into the chatbot at any point of `book_movie`, if so, it returned a specific error message. Test 5 shows how the chatbot informed the user of the errors wrong with the

```
Please enter your name: jesal vadgama
Is Jesal Vadgama correct? (yes/no): yes
You: wonka description
Bot: Title: Wonka
Genre: Musical
Release Year: 2023
Director: Paul King
You: is it a happy movie
Bot: Yes, it is a happy movie with lots of songs.
You: description dark knight
Bot: Title: The Dark Knight
Genre: Action
Release Year: 2008
Director: Christopher Nolan
You: is it a happy movie
Bot: It's a great movie that you might enjoy.
You: is it a exciting movie
Bot: Yes, it is an action-packed movie full of excitement.
You: []
```

Figure 7: Shows Context Tracking

user\_input, trying to prompt the user to enter a valid user input. It was able to handle all invalid input including spelling movie names wrong and gave the correct movie output.

Test 1	Test 5
<pre>Please enter your name: jesal vadgama Is Jesal Vadgama correct? (yes/no): yes You: no Bot: I'm sorry you stopped booking and that i could not find your movie Bot: Unfortunately, Jesal Vadgama, the movie selection process has been stopped</pre>	<pre>Please enter your name: Jesal Is Jesal correct? (yes/no): nope Is Jesal correct? (yes/no): no Let's try again. Please enter your name: Jesal Vadgama Is Jesal Vadgama correct? (yes/no): yes You: could i book a movie Bot: These are the current movies available The Godfather (1972) - Francis Ford Coppola Raiders of the Lost Ark (1981) - Steven Spielberg Titanic (1997) - James Cameron The Dark Knight (2008) - Christopher Nolan Avatar (2009) - James Cameron Inception (2010) - Christopher Nolan Interstellar (2014) - Christopher Nolan The Shawshank Redemption (1994) - Frank Darabont Spider-Man: Across the Spider-Verse (2023) - Joaquin Dos Santos Hunger Games: The Ballad of Songbirds &amp; Snakes (2023) - Francis Lawrence Wanda (2023) - Paul King Napoleon (2023) - Ridley Scott Mad Max: Fury Road (2015) - George Miller Gorilla Minus One (2023) - Takashi Yamazaki Hunger-Games: Ballad of Songbirds-Snakes (2023) - Francis Lawrence</pre>
<pre>Bot: Jesal Vadgama, what movie would you like to book? (Type no at any point to stop) You: no Bot: I'm sorry you stopped booking and that i could not find your movie Bot: Unfortunately, Jesal Vadgama, the movie selection process has been stopped</pre>	<pre>You: dark night Bot: Jesal Vadgama You have selected the movie The Dark Knight Bot: Jesal Vadgama, enter a day you would like to book the movie in YYYY-MM-DD format You: no Bot: I'm sorry you stopped booking and that you changed your mind about booking a movie Bot: Unfortunately, Jesal Vadgama, the date selection process has been stopped</pre>
<pre>Test 3</pre>	<pre>Test 5</pre>
<pre>Bot: Jesal Vadgama, what movie would you like to book? (Type no at any point to stop) You: dark night Bot: Jesal Vadgama You have selected the movie The Dark Knight Bot: Jesal Vadgama, enter a day you would like to book the movie in YYYY-MM-DD format You: 2027-12-12 Bot: Jesal Vadgama You have chosen to book a move on 2027-12-12 Bot: Jesal Vadgama, what time would you like to book the movie for? Bot: The available show times for The Dark Knight (2008) are 13:00, 17:00, 21:00. You: no Bot: I'm sorry you stopped booking and that the time you picked wasn't available Bot: Unfortunately, Jesal Vadgama, the time selection process has been stopped</pre>	<pre>Bot: Jesal Vadgama You have selected the movie The Dark Knight You: 2022-12-12 Bot: Booking date in the past. Try again. You: 2024-12-12 Bot: Invalid date format. Please enter the date in YYYY-MM-DD format. You: hello Bot: Invalid date format. Please enter the date in YYYY-MM-DD format. You: 2024-12-12 Bot: 3 Jesal Vadgama You have chosen to book a move on 2024-12-12 Bot: Jesal Vadgama, what time would you like to book the movie for? Bot: The available show times for The Dark Knight (2008) are 13:00, 17:00, 21:00. You: no time Bot: That time is not available Jesal Vadgama please choose one of the available show times. You: 13:00 Bot: Jesal Vadgama you have selected 13:00 Bot: Jesal Vadgama, how many tickets would you like to book? You: none Bot: Invalid input. You: 123 Bot: Jesal Vadgama you can't book more than 10 tickets per booking You: 10 Bot: Jesal Vadgama your total number of tickets booked is 1 Bot: Please type confirm to continue booking or no to stop: i want to confirm Bot: I'm sorry, I didn't understand that. Type confirm or no. Bot: Please type confirm to continue booking or no to stop: confirm Bot: Thank you for booking Jesal Vadgama, this is your Booking ID: booking_b6e2c991ab9a4f8fb Bot: You have booked The Dark Knight for 13:00 on 2024-12-12 with a ticket total of 1.</pre>
<pre>Bot: Jesal Vadgama, what movie would you like to book? (Type no at any point to stop) You: dark night Bot: Jesal Vadgama You have selected the movie The Dark Knight Bot: Jesal Vadgama, enter a day you would like to book the movie in YYYY-MM-DD format You: 2024-12-12 Bot: Jesal Vadgama You have chosen to book a move on 2024-12-12 Bot: Jesal Vadgama, what time would you like to book the movie for? Bot: The available show times for The Dark Knight (2008) are 13:00, 17:00, 21:00. You: 17:00 Bot: Jesal Vadgama you have selected 17:00 Bot: Jesal Vadgama, how many tickets would you like to book? You: 1 Bot: Jesal Vadgama your total number of tickets booked is 1 Bot: Please type confirm to continue booking or no to stop: no Bot: I'm sorry you stopped booking Bot: Unfortunately, Jesal Vadgama, the booking process has been stopped</pre>	

Figure 8: Shows performance testing of user input

I prepared a test training set of user inputs and outputs, my classifiers are booking movie related intents, cancellation related intents, and queries relating to movies. I ran these inputs using the chatbot and compared the chatbots response to the expected output. I performed cross validation by dividing the test sets into k subsets, training the chatbot on the remaining k-1 and repeated it for each subset used for testing. The results show booking movie related intents had higher accuracy, cancellation had lower accuracy and queries related to movies had the worst being 0.75.

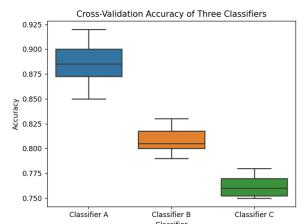


Figure 9: Results of classifiers

## 4.2 Usability Testing

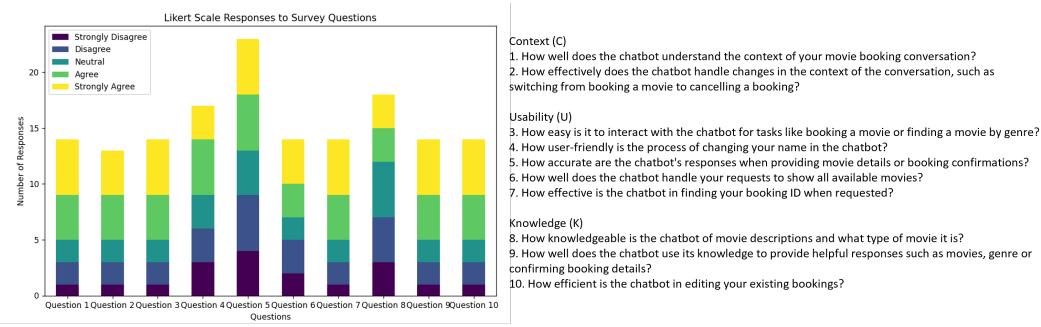


Figure 10: Results on Usability

Using Context, Usability, Knowledge, I created a survey of the chatbot, collected data on a 5-point scale (figure10). Chatbot understands the context of a movie booking conversation for Q1,Q2 and most selected strongly. Some users selected disagree but is expected when testing with a diverse range of users as some may not understand or be tech-savvy. The chatbot is user-friendly and easy to interact with Q3-Q7, the accuracy of chatbot's responses when providing movie details or booking confirmations could be improved, as some disagreed. For Q8-Q10 which tests knowledge had mostly agree and strongly agree showing the high of efficiency in handling some inputs. This shows how well the chatbot performs error handling and also shows cosine similarity numbers may need fine tuning.

## 5 DISCUSSION OF RESPONSIBLE RESEARCH AND INNOVATION FRAMEWORK

The dataset.csv is filled with intents and responses that are respectful, inclusive, and appropriate for all users. The chatbot treats all users equal regardless of belief and gender. There is no gender bias or bias and follows ethical guidelines by respecting privacy of the user as it only stores booking information with valid inputs. I created dataset.csv by asking user for inputs and testing the chatbot and with multiple users for diversity. The system is transparent and outputs everything written to the bookings.db and has functions to handles accidental inputs. The chatbot does not do automated tasks on its own and has been created to help users book movies. I would not know the effects of the chatbot if it were taken to different parts of the world because of the language, time, cultural differences. Unfortunately, the chat does not have screen reader to cater with those of vision difficulties.

## REFERENCES

- [1] Clos, J. (2023) *Lab 5: The Art of Evaluation*, COMP3074 Human-AI Interaction. Accessed: 15.12.2023
- [2] Fischer, J. (2023) *Lab 4: Conversational design*, COMP3074 Human-AI Interaction. Accessed: 15.12.2023