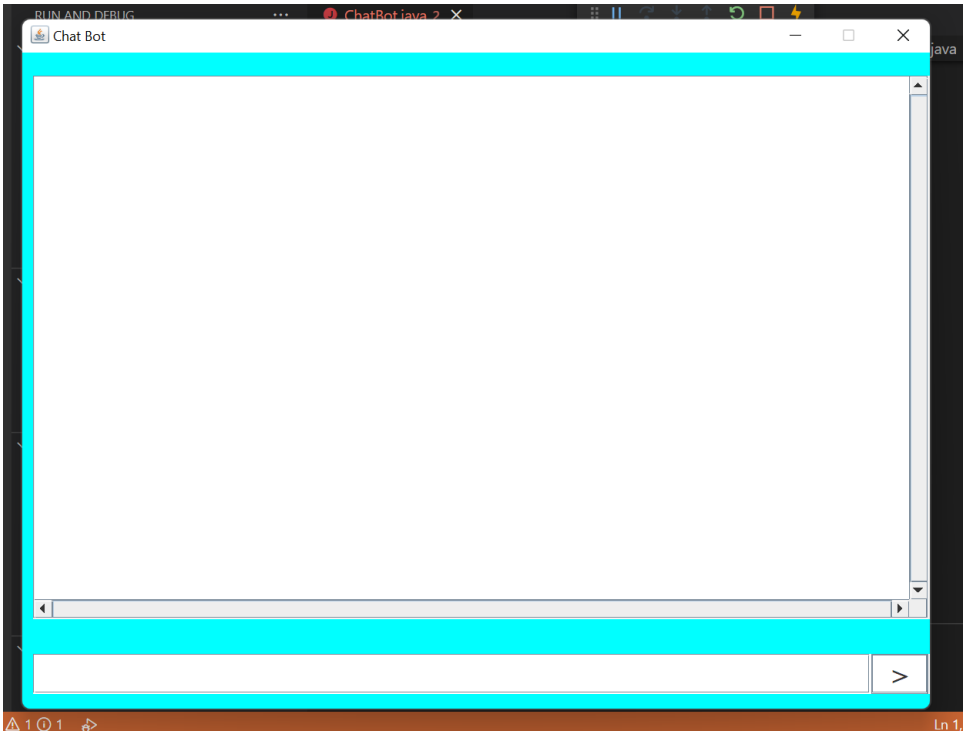
Assignment 3

Coding

1 point: Make a simple GUI so that the user is typing into a nicer interface and can view a recent history of the conversation



0.5 point: Add an extra topic to your agent's repertoire. Ensure this topic has similarities with the original topic. For example, if your original topic is volleyball, you may want to add basketball as a second topic.

Answer) Our two topics are related to going out with a friend , One is going for a movie and other is going out for a dinner.

0.5 point: Add a feature that enables your agent to give at least 5 different reasonable responses when the user enters something outside the two topics.

Answer) Whenever the response is outside the topic our following code produces desired results :

```
259                 }
260
261             } catch (Exception eee) {
262                 int num = random.nextInt(5);
263                 if (num == 0) {
264                     bot("Sorry ,I can't understand you !");
265                 } else if (num == 1) {
266                     bot("That is off topic ");
267                 } else if (num == 2) {
268                     bot("My apologies...I don't know what you are saying ");
269                 }else if (num == 3) {
270                     bot("My apologies...I don't understand ");
271                 }else if (num == 4) {
272                     bot("Come again please ");
273                 }
274
275             }
276         }
277
```

Documentation

• 0.5 points: README file in your repository describing what you've done. You are likely to just add onto the README from A2. If you've changed a lot since A2, you will have to rewrite the README so it reflects your current submission.

Readme file updated in GitHub

• 1 point: At the end of your README file, include: o a list of each feature you programmed for this assignment o for each item on that list, explain briefly how you used that feature to improve your agent's conversation or your overall system o for each explanation, give a snippet of a conversation that demonstrates your feature

Updated in the readme file

• 0.5 points: Provide a Level 0 Data Flow Diagram (DFD) for your system with description.

Available in the GitHub

• 1 point: Provide a Level 1 DFD for your system with description.

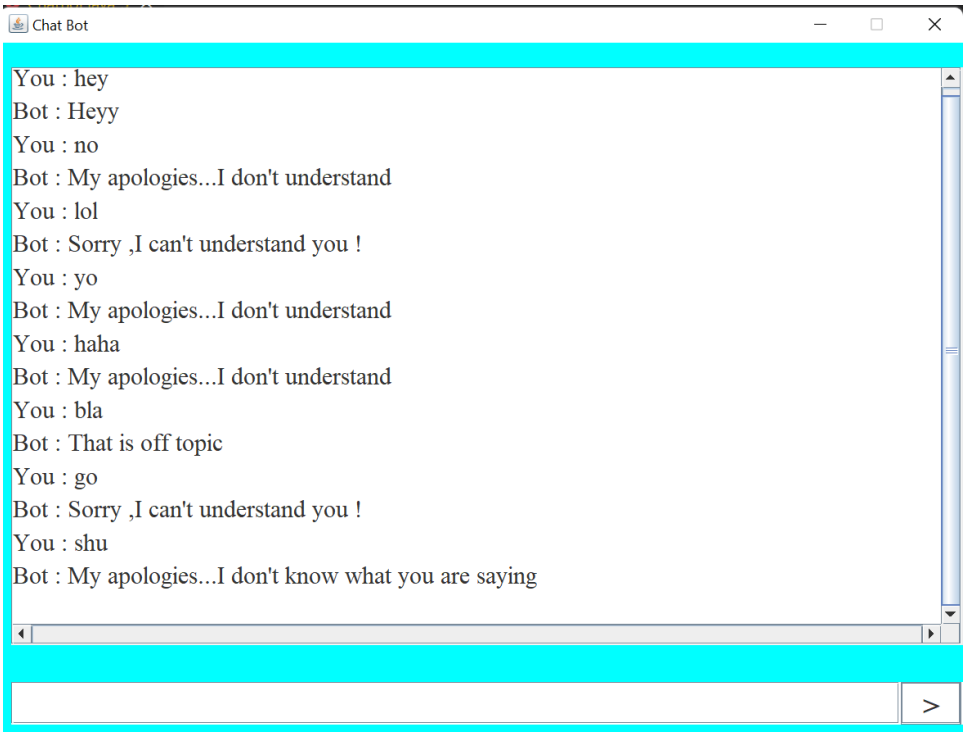
Available in the GitHub

• 1 point: Submission of your GitHub repository. Graph showing different features developed on a separate branch and the commits made in the repository.

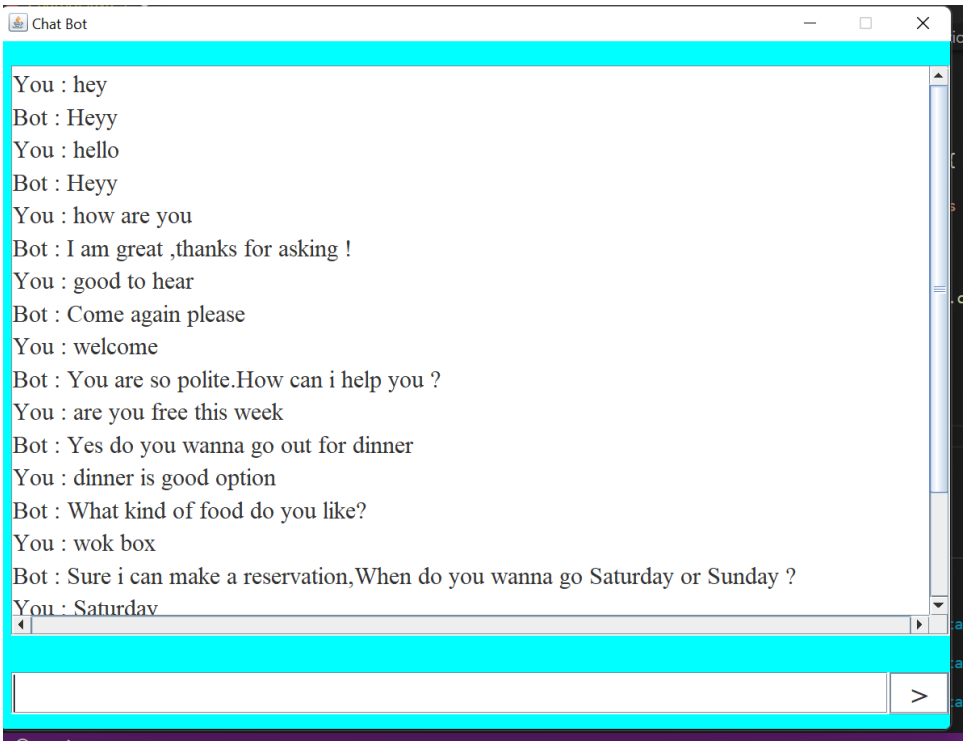
https://github.com/aaliazoya/COSC-310-Assignment2.git

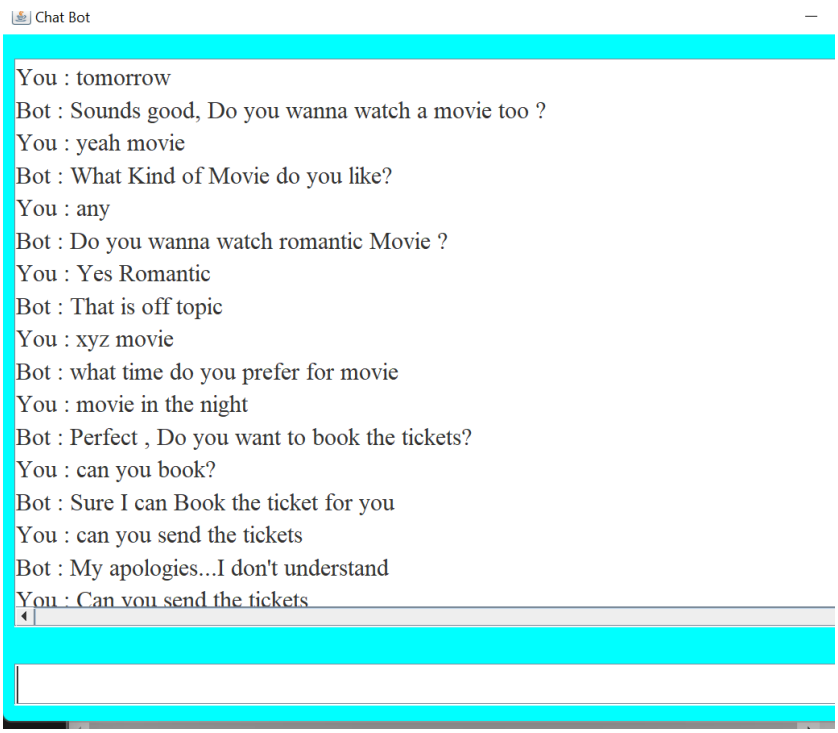
• 1 point: Include sample output in your project report. Have one dialogue (at least 30 turns) that show a good or feasible conversation -- ensure your new features are demonstrated! Document a list of limitations of your program, and have at least two short dialogues that show when your agent is not able to handle the conversation properly.

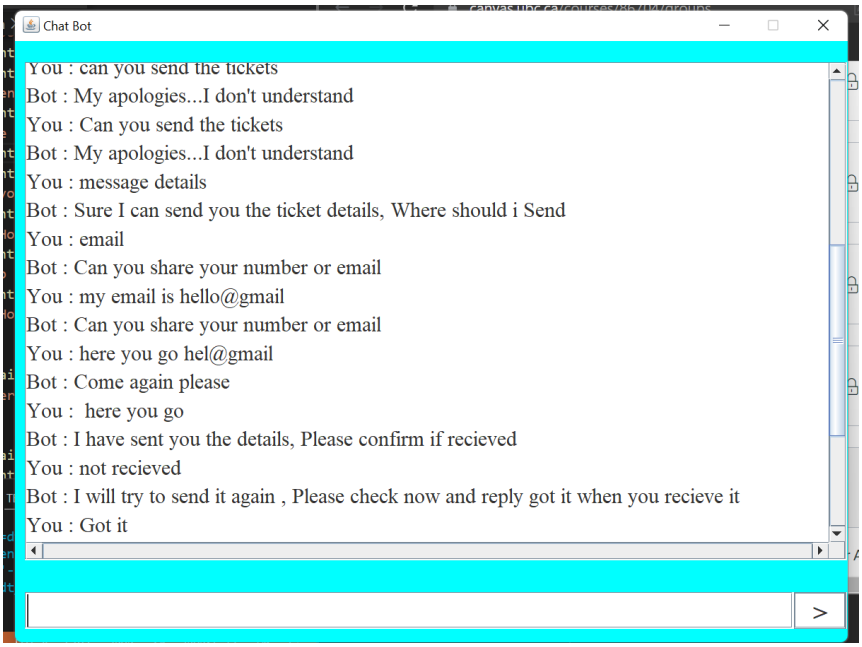
Agent cant handle



Proper Conversation with agent can't handle few lines







• 1 point: Based on your system, include a list of at least 5 features that you can extract from your code or design that can be shared with others as an API

Features of our code is as follow:

1) Added a better GUI for the user to feel like the individual is having a normal conversation with the bot.
2) Added another topic of discussion. This would allow the conversation to be more immersive.
3) Before there were 3 responses given if the user entered something out of topic, now there are five responses that the bot can provide the user with.
4) On entering “Clear screen” our Code clears the conversation and sets the screen empty
5) If the user input is not understood by the Code, our code opens google search to look up for solutions.

Presentation

Presentation (3 points) A 5-minute video of your assignment showing:

• A brief description of your program's conversational topic

Our project is a simple and friendly chat-bot with which the user can talk about their hobbies, interests etc. We are using the chat bot as a friend but in most cases these days these bots are used for customer service. Keeping in mind the recent pandemic and how it has changed everyone's social lives. A virtual "friend" is able to make normal conversations and allows the user to chat to this bot anytime and share their experiences when their friends are busy or for the people who have a hard time socializing.
We have used the Java GUI to create an interface for the chat bot and used Java as the programming language that is used to develop this chat bot.

• Each feature you've programmed and how you used it to improve either the conversation or the overall system (since A2)

1. Added a better GUI for the user to feel like the individual is having a normal conversation with the bot.
2. Added another topic of discussion regarding the topic, our bot was a friendly bot which would suggest some plans for the weekend, we added arcade to the list. This would allow the conversation to be more immersive.
3. Before there were 3 responses given if the user entered something out of topic, now there are five responses that the bot can provide the user with. The responses before were more repetitive and now as there are more responses that the bot could give, so the user feels like it is a more normal conversation rather than talking to a bot.
4. On entering "Clear screen" our Code clears the conversation and sets the screen empty
5. If the user input is not understood by the Code, our code opens google search to look up for solutions.

- A description of your DFDs