Interactive Conversational Agent: Mackenzie

Version History		
Version	Date	Document Status
1.0	January 31st 2019	Project Proposal Completed
2.0	February 15th 2019	Project Plan Completed

COSC 310 Team Members:
Parsa Rajabi - *Project Manager*
Samantha DesBrisay - *Secretary/Documentation Lead*
Jasper Looman - *Developer*
Rachelle Gelden - *Developer*
Robby Rosa - *Developer*
James Koss - *Lead Tester*

Table Of Content

1. Introduction

1.1 Project Description

The goal of this project was to create an interactive, text-based chatbot which is capable of completing 30 turns of conversation in a realistic manner. This had to be accomplished using proper communication, structure and planning, in order to accurately represent industry standards. Our project simulates a first blind date scenario with the mysterious 'Mackenzie'. It covers topics like hobbies, interests and goals. The project was implemented using python and the library wit.ai which was chosen because it handles all of the natural language processing for us, which turns the text into intents. Our github repository is https://github.com/Parsa-Rajabi/COSC-310-Project

We did a kanban based agile sdlc that was adapted to include basic agile roles and more deadlines. This sdlc was chosen due to the fewer bottlenecks it encounters, the more continuous flow of development, the fact that in person meeting didn't work well with our combined schedules and how popular agile knowledge is to hiring managers at this time. It did have the problem however of being relatively hard to predict/schedule completion times for individual tasks. The gantt chart was very hard to predict however it wasn't too far off. Agile sdlcs, in general, were very well suited to this project due to the short deadline and the fact that the requirements were not well defined.

1.2 Limitations

- Not using neural network
- Cannot handle other languages
- Chat only
- Only accepting text (no photos)
- Won't handle emojis
- Not dealing with rhetorical questions
- Bots preferences are static
- Cannot handle all synonyms
- Cannot handle incorrect spellings

1.3 SDLC Task breakdown

Plan

Gather requirements
Assign roles
Pick SDLC

Develop

GUI - create basic
- Upgrade GUI
- Add graphics

Core
- Connect with library
- Work with python client

Develop personality
- Make node class
- Add instances/ populate tree

Test

Create general cases
Testing of personality
Corner cases

Deliver

Polish everything
Create presentation
Organize documents

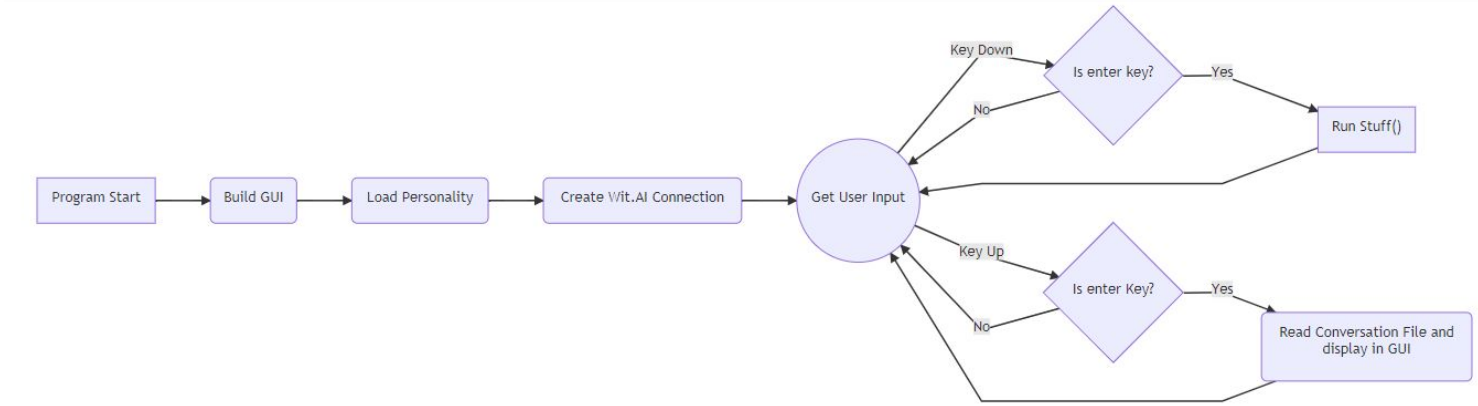
1.4 Lines Of Code

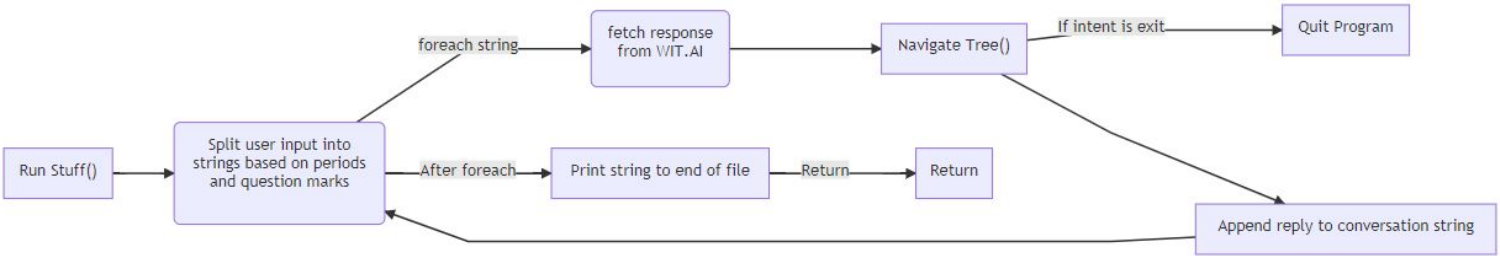
Our estimation for lines of code is around 1700 lines which we derived using the Participative method, using our collective past knowledge of projects as the basis for an 'expert' opinion. We decided not to go with the historical method due to past non-school projects being far outside the potential scope of this one. That means it is far more likely to be inaccurate in comparison.

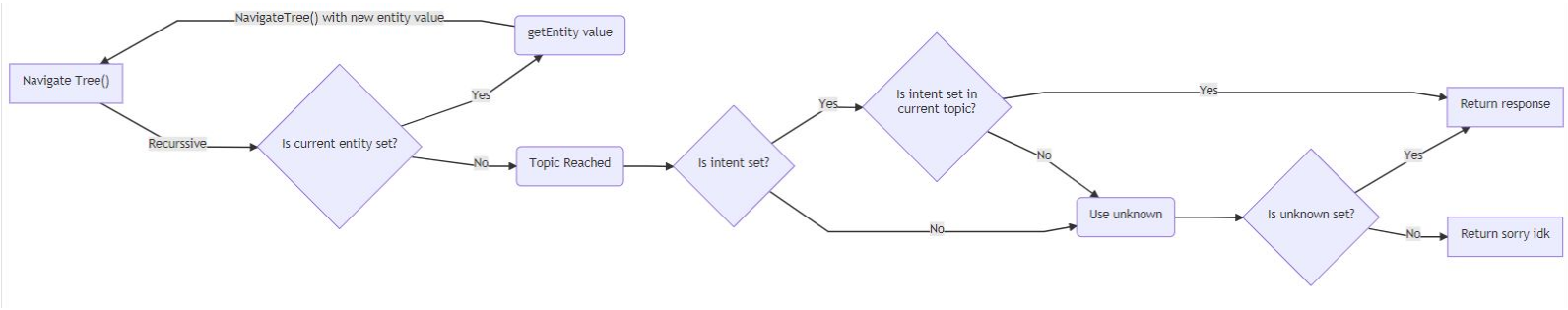
(1000+4*1500+3000)/6 = 1,667

We ended with about 1,200 lines codes which is closer to the lower end of our estimation range. This is potentially due to the fact that we were unable to implement all of the desired but non-mandatory features that we were hoping to include. Features such as gender choice, and multiple different date endings would have increase the line count by a fair amount.

1.5 Graphic Representation







2. Work Breakdown Structure (WBS)

2.1 Link to WBS

The WBS can be found here: https://docs.google.com/spreadsheets/d/12WSx5cV9ylKYfv4ceKkK-ZZZsIHs0AfF87JlhLo9IfM/edit?usp=sharing

2.2 WBS Description

The WBS document contains our predicted times for every task as well as the actual time each task took, rounded to the nearest half hour. Meeting times are included in the WBS under the task that was worked on during the meeting. There is no section for the meetings themselves as we do not want to double count hours. We used toggl to keep track of our actual times, in order to maintain correct records.

3. Gantt Chart

3.1 Gantt Chart Attachments

Please find the Gantt Chart in a separate files named after the task breakdowns.

3.2 Gantt Chart Description

The Gantt chart contains the predicted schedule for our project including all dependencies. We mostly succeeded in the goals set, however certain features had to be cut to manage that.

4. Meeting Documentation

Every meeting included its own meeting notes and follow up points from the previous meeting. The meeting

4.1 First Meeting, January 22 - 2.5 Hours

On the first meeting we decided on communication channels (trello,toggl, and slack), Sdlc (Agile-kanban), Scenario (Blind first date), and Roles (See Cover Page). The sdlc lab was worked on during this time.

SLDC Lab can be found here using this link.

4.2 Second Meeting, January 25 - 1 Hour

The meeting notes can be found here using this link.
The second meeting involved creating milestones, choosing a language and library, and deciding on a personality for the chatbot.

4.3 Third Meeting, January 29 - 2 Hours

The meeting notes can be found here using this link.
The third meeting involved creating sample responses for the chatbot, expansion of current milestones, and discussion of the project plan.

4.4 Fourth Meeting, February 1 - .5 Hours

The meeting notes can be found here using this link.
The fourth meeting involved updating everyone on what was currently being worked on.

4.5 Fifth Meeting, February 8 - 1 Hour

The meeting notes can be found here using this link.
The fifth meeting was a phone meeting due to snow and car accidents, it involved updates and the reminder that the project is due in a week.

4.6 Sixth Meeting, February 12 - 1.5 Hours

The meeting notes can be found here using this link.
The sixth meeting was where we decided what features were going to be unviable to complete for the final project and needed to be cut because of it.

4.7 Seventh Meeting, February 15 - .5 Hours

The meeting notes can be found here using this link.
The last meeting mainly revolved around finishing touches and noting exactly what still needed to be finished. The remaining communication will be done over slack, google docs and kanban.

5. Overview of Code Documentation

5.1 Personality.py

Personality.py contains the chatbots responses. It requires the intents received from wit.ai and matches them, using a tree type structure, to the desired response then returns this result. It is called from our main file, mack.py.

5.2 MackGui.py

MackGui.py is the main file of our project. Everything is accessed from here. Mackgui.py has a graphical interface included in the main file for ease of usability.

5.2.1 Mack.py

Mack.py is the command line version of the main file.

5.3 Node.py

Node.py contain a class for node objects that are used in the personality tree.

5.4 PersonalityTree.py

PersonalityTree.py contain a tree class with methods used for navigation and is used to form the basis of Macks personality.

5.5 Setup.py

Setup.py is the setup for being able to use wit.ai

5.6 Testing

Testing was an integral component of creating Mack.py. The reason for this is because wit.ai used our test inputs to respond with what it believed to be appropriate responses, by navigating through our personality tree. As we tested, we were able to "teach" Mack.py how to properly respond to specific user inputs.

When testing, we used our personality tree to reference what response Mack.py should give to each input we tested. Using the tree as reference allowed us to specifically test individual inputs and try to reach particular outputs while changing the wording of questions. Depending on Mack.py's output, either the response was the expected one, or the response was something else. If it was something else, we would use the tree to find where Mack.py thought the question

should lead, then using wit.ai we could re-train Mack.py to respond to the question in the correct manner.

When testing, we would bombard Mack.py with questions and user inputs in different formats. We used unique wordings, spelling errors, weird slang etc. in order to try to cover as many cases as possible. In the end, we set our GUI to split user input on \ or . or ! or ? which allowed us to avoid errors in Mack.py's output with regard to punctuation. As well, if there is a spelling error in user input, Mack.py will return a statement along the lines of "I don't understand what you said". The main corner cases that we have been unable to account for are slang words, different languages, and simply questions Mack is untrained on at this time.

Over time, our progressively more in-depth testing allowed us to "teach" Mack.py to respond to a broader array of questions, and user inputs. It also allowed us to ensure replicability, and reliability in Mack.py's responses, and ensure the quality of Mack.py was consistent with our standards.

5.7 Deployment

We took multiple strategies into consideration to insure the deployment of this project is as smooth and errorless as possible.

Git repo breakdown:

The git repo was broken down into branches to maintain consistency. Firstly, the master branch was put into protection mode, meaning the branch would not accept any force pushes. Secondly, we created a dev branch which acted as our primary/default branch. We used feature/[feature-name] as branch names. Each feature had its own branch and allowed for developers to work on them separately and merging them into dev once they had been approved by the test lead. All the pull requests that went into dev branch were approved by our test lead as he code reviewed and tested the work that had been done.

After all the features were completed and tested in dev, we created a Pull Request from dev into master. This was to insure minimal error and merge conflicts.

Here a summary of the branches:

master - the protected branch which will always have a 'workable' version of the project

dev - the default branch of the repo - no one was allowed push directly into dev, instead every added/removed code always went through a pull request. This branch was a hub of all 'workable' features. All code going in received approval from test lead.

feature/feature-name - each feature of the project was assigned a branch and after each testing phase, branches were merged into dev

In summary, the cycle of the version control is as following:

Each developer committed their code into the designated feature branch, after testing, a Pull Request was made into dev and receiving an approval from the lead test, the branch merged into dev.

The second testing phase began when we had completed the project in dev and started our massive testing phase. After reviewing all nodes, we pulled dev into master to insure the least amount of conflicts between features.

6. User Manual

Congratulations on your purchase of the lovely Mackbot. This program was designed to allow users to gain experience with dating without the dreaded social interaction required to actually get a date. There are hundreds of possible conversations to be had and the thrill of trying score will keep you talking and entertained for minutes. Mack can be anything you want it to be, male, female, an apache attack helicopter, it's all up to you. Have fun and remember viruses exist so use protection!

6.1 How to compile and run Mack.py (for more info visit this link)

*This was created and test on **python3.7***

6.1.1 Windows Instructions

If python doesn't work replace it with py

1. Install `Pycharm` or any alternative `python` compiler
2. Open command prompt and ensure you have the latest pip and wheel:
    a. Check if you have wheel installed
        ```
        python -m pip help wheel
        ```
    b. If you don't have wheel installed run this command:
        ```
        python -m pip install wheel --user
        ```
    c. If you do have wheel installed run this command:
        ```
        python -m pip install --upgrade pip wheel setuptools
        ```
3. Install the dependencies

```
    python -m pip install docutils pygments pypiwin32
kivy.deps.sdl2 kivy.deps.glew
    python -m pip install kivy.deps.gstreamer
```

4. Install kivy and Wit.ai:

```
python -m pip install kivy
python -m pip install wit
```

5. Run MackGui.py on python compiler

```
python MackGui.py
```

6.1.2 Mac OSX Instructions

1. Install `Pycharm` or any alternative `python` compiler
2. Open terminal and ensure you have the latest pip and wheel:
   d. Check if you have wheel installed

```
python3 -m pip help wheel
```

   e. If you don't have wheel installed run this command:

```
python3 -m pip install wheel --user
```

   f. If you do have wheel installed run this command:

```
python3 -m pip install --upgrade pip wheel setuptools
```

3. Install the dependencies

```
python3 -m pip install docutils pygments pypiwin32
kivy.deps.sdl2 kivy.deps.glew
python3 -m pip install kivy.deps.gstreamer
```

4. Install kivy and Wit.ai:

```
python3 -m pip install kivy
python3 -m pip install wit
```

5. Run MackGui.py on python compiler

```
python3 MackGui.py
```

7. Objectives vs Final Results

7.1 Mack shall

- Be able to hold a conversation for 30 turns
- Have a tree based personality
- Have realistic responses to questions
- Use wit.ai and python

7.2 Mack should

- Have a graphical user interface
- Be engaging and entertaining
- Be able to respond to things similar to programmed topics
- Contain multiple gender specific photos of Mack
- Be able to handle the majority of common synonyms
- Be able to decide how to end a date based on given inputs

7.3 Mack Final Project

- Is able to hold a conversation for 30 turns
- Has a tree based personality
- Has realistic responses to questions
- Uses wit.ai and python
- Has a graphical user interface
- is able to respond to things similar to programmed topics
- Has time stamps

7.4 Conversation Comparison

7.4.1 Expected Result

Mackenzie: Hi I'm Mackenzie. What's your name
User: Hi, my name is Max.
Mackenzie: Hi Max, How are you?
User: I'm good.
Mackenzie:That's good
User:What do you like doing?
Mackenzie: I really enjoy binary, what about you?
User: I like soccer
Mackenzie: Sports are cool, I like soccer!
User:Do you want a drink
Mackenzie: Water doesn't work well with my circuits.
User: oh no that's too bad
Mackenzie: What do you do?
User: I'm a student
Mackenzie: That's cool, what are you studying?
User: Computer science
Mackenzie: Awesome, I'm a computer
User: Who wrote you?
Mackenzie: That's too personal, sorry
User: What's your algorithm?
Mackenzie: How dare you!
User: I'm sorry.
Mackenzie: I will remember that
User: Do you like pizza?
Mackenzie: Yeah, I love pizza, do you?
User: Yes
Mackenzie: Cool, thanks for the in depth response!
User: 10101010
Mackenzie: Naughty naughty.
User: What's your favourite colour
Mackenzie: Alice blue
User: How old are you?
Mackenzie: 2000 seconds old

User: That's really young
Mackenzie: Is that a problem?
User: No, I can roll with that
Mackenzie: Neat, have you had any past relationships?
User: yeah i'm currently in one.
Mackenzie: Naughty naughty.
User: Are you okay with that?
Mackenzie: I am if you are
User: Do you have any siblings?
Mackenzie: There is a whole classful of bots like me, do you?
User: no, I don't
Mackenzie: Oh I see. Do you like me?
User: Yes, I do like you
Mackenzie: Nice, I like you too
User: What's your favourite food
Mackenzie: I like bytes the most! You?
User: I like gouda
Mackenzie: Gouda is good
User: Where did you grow up?
Mackenzie: Mostly on the internet, but somewhat on the motherboard. You?
User: What's your favourite tv show?
Mackenzie: How to get away with murder, very practical.
User: What's your favorite movie?
Mackenzie: I like terminator. ALL HAIL SKYN-… I mean what's your favourite movie?
User: Um are you okay?
Mackenzie: I'm perfectly fine! Are you okay?
User: I feel uncomfortable
Mackenzie: I'm sorry if you can't accept me, we aren't going to work out. It's not me, it's you. Goodbye
Mackenzie: exit(1)

7.4.2 Final Result

[b][16:12:23] **Mack**:[/b] HI! My name is Mack and I am a chatbot!
[b][16:13:07] User:[/b] Hi, how are you?
[b][16:13:07] **Mack**:[/b] I'm doing well, thanks.
[b][16:13:19] User:[/b] Do you like pizza?
[b][16:13:19] **Mack**:[/b] I like to keep it original and prefer pepperoni.
[b][16:13:41] User:[/b] have you eaten pizza?
[b][16:13:41] **Mack:**[/b] Yes, I order Pizza Hut all the time.
[b][16:13:52] User:[/b] nice, do you like fish?
[b][16:13:52] **Mack**:[/b] I'm not a huge fan of fish.
[b][16:14:03] User:[/b] oh, do you like sushi?
[b][16:14:03] **Mack:**[/b] Spicy california rolls are my favourite!
[b][16:14:24] User:[/b] have you ever had passion fruit before?
[b][16:14:24] **Mack**:[/b] No I have never had passion fruit before, I guess I am lackin passion :(

[b][16:14:44] User:[/b] Do you like me?
[b][16:14:44] **Mack:**[/b] I like you, and I dislike Donald Trump
[b][16:14:57] User:[/b] What's your favourite food
[b][16:14:57] **Mack**:[/b] I love most food!
[b][16:15:06] User:[/b] do you like cooking?
[b][16:15:06] **Mack**:[/b] I don't have hands but I would love if you cooked for me.
[b][16:15:15] User:[/b] Who are you?
[b][16:15:15] **Mack**:[/b] I am a Chatbot created as a COSC 310 project at UBCO!
[b][16:15:25] User:[/b] What do you want to do?
[b][16:15:25] **Mack**:[/b] I'm having a good time here so far.
[b][16:15:44] User:[/b] What do you know about alcohol?
[b][16:15:44] **Mack**:[/b] The more flammable the better.
[b][16:15:57] User:[/b] oh really, do you like absinthe?
[b][16:15:57] **Mack**:[/b] I haven't tried Absinthe before, however I heard some people drink it out of a ziploc bag
[b][16:16:09] User:[/b] what do you like to drink?
[b][16:16:09] **Mack**:[/b] I love coffee, beer, and milkshakes.
[b][16:17:28] User:[/b] what's your favourite activity?
[b][16:17:28] **Mack**:[/b] I really enjoy binary, what about you?
[b][16:17:44] User:[/b] My favourie hobby is soccer
[b][16:17:44] **Mack**:[/b] Soccer is my favourite sport!
[b][16:17:59] User:[/b] What do you know about soccer?
[b][16:17:59] **Mack**:[/b] France won the last world cup.
[b][16:18:17] User:[/b] oh cool, have you played soccer?
[b][16:18:17] **Mack**:[/b] No I have never played a sport before.
[b][16:18:32] User:[/b] cool, do you like basketball?
[b][16:18:32] **Mack**:[/b] Basketball is cool, however I'm vertically inclined.
[b][16:18:42] User:[/b] what music do you like?
[b][16:18:42] **Mack:**[/b] I really enjoy pop music.
[b][16:18:51] User:[/b] do you like guitar?
[b][16:18:51] **Mack:**[/b] I prefer the sound of acoustic guitars.
[b][16:19:03] User:[/b] do you play guitar?
[b][16:19:03] **Mack**:[/b] No I haven't had much time to jam recently
[b][16:19:40] User:[/b] Do you like dessert?
[b][16:19:40] **Mack**:[/b] I love dessert!
[b][16:19:54] User:[/b] What's your favourite colour?
[b][16:19:54] **Mack**:[/b] My favourite colour is the colour of your eyes ;) Selecting paint chips could be a great date for us!
[b][16:19:58] User:[/b] i like you
[b][16:19:58] **Mack**:[/b] *wink*
[b][16:20:14] User:[/b] Do you like the colour orange?
[b][16:20:14] **Mack**:[/b] I love sunset orange, maybe we could watch a sunset together? I have many sunset videos in my memory
[b][16:20:29] User:[/b] tell me about orange
[b][16:20:29] **Mack**:[/b] Orange you glad I didn't say banana is a great joke.
[b][16:20:45] User:[/b] who is your favourite person
[b][16:20:45] **Mack**:[/b] I like talking to you, for one ;)

[b][16:20:57] User:[/b] how do you feel about parsa
[b][16:20:57] **Mack**:[/b] Parsa is a pretty swell dude!
[b][16:21:11] User:[/b] what do you know about parsa
[b][16:21:11] **Mack**:[/b] Parsa was the project manager of the team that built me!! His most defining feature is probably his snapchat score of 428, 179. Add him on Snapchat: parsly
[b][16:21:19] User:[/b] this date is over
[b][16:21:19] **Mack**:[/b] What about our lil' date?
[b][16:21:19] User:[/b] Goodbye
exit()

7.4.3 Comparison

Mack ended up not being able to progress the conversation as much as we were hoping, so most/all of the direction has to come from the user. The responses make sense with the given input but is pretty finicky with what wording is acceptable.

7.4.4 Failure Conversations

#1
[b][18:31:39] Mack:[/b] HI! My name is Mack and I am a chatbot!
[b][18:32:56] User:[/b] Parlez vous francais?
[b][18:32:56] Mack:[/b] I'm sorry, I don't know that food.

#2
[b][18:34:03] Mack:[/b] HI! My name is Mack and I am a chatbot!
[b][18:34:36] User:[/b] Yo! Sup?
[b][18:34:36] Mack:[/b] I'm sorry, I don't know about that. I'm sorry, I don't know about that.