COSC 4P02 - Software Engineering 2
Progress Report 1

Roadmen Bugsy
Team leader: Abdul Rahman Abdullah, 6320493
Aws Al Jumaily, 6459861
Anwar Hashmi, 6584957
Ifti Ahmed, 6559348
Ray Keating, 6510200
Nilaanjann Misra, 6366298
Amots Avigad, 6556591
Austin Goodman, 6314363

****

1- Write the name of the group and the list of your group members.
2- Include a brief list/description of the features (subsystems) you planned to implement in each sprint and whether or not you accomplished them.
3- Include a brief list of features (subsystems) that you plan to implement in the following sprints.
4- Include screenshots if you have a working version of the system.
5- Mention any issues you encountered.
6- Describe the contributions and achievements of each member of the group.

7- And anything else you would like to add.

****

Features we planned to implement in Sprint 1 and 2

Features planned for sprint 1

Feature

A basic chat app that sends and receives messages.

≡ 📖 2

As a user I would like to be able to ✎ type out my question to the chatbot and then wait for its response so that I can communicate my queries to the chatbot in simple English without any technical jargon.

📖 1   ☑ 2/2

As a user I would like to be able to ✎ type in my messages to the chatbot and see all the texts that I have sent to and received from the chat bot during the entirety of our session so that I can read the messages at my pace and reference them later on.

☑ 4/4

As a user I would like the chatbot responses to my query be intelligent, coherent in simple English so that I can understand the response and get a feeling that I am talking to a human.

Features planned for sprint 2

Feature

Web scraping for courses and programs

≡ 📎 1 📖 5

Scraping/DB

As a student I would like to for the chatbot to have up-to-date data from brock's website about course Ids and faculty info so that I can trust that the answers I am getting are accurate and non-conflicting.

☑ 4/4

Frontend

As a user, I would like a dedicated submit button which will allow me to submit a message I have typed in the text box.

☑ 2/2

Frontend

As a user, I would like to see an error on screen or a message from the bot when the input field is empty and enter/submit is hit, so I am aware of the error made.

☑ 3/3

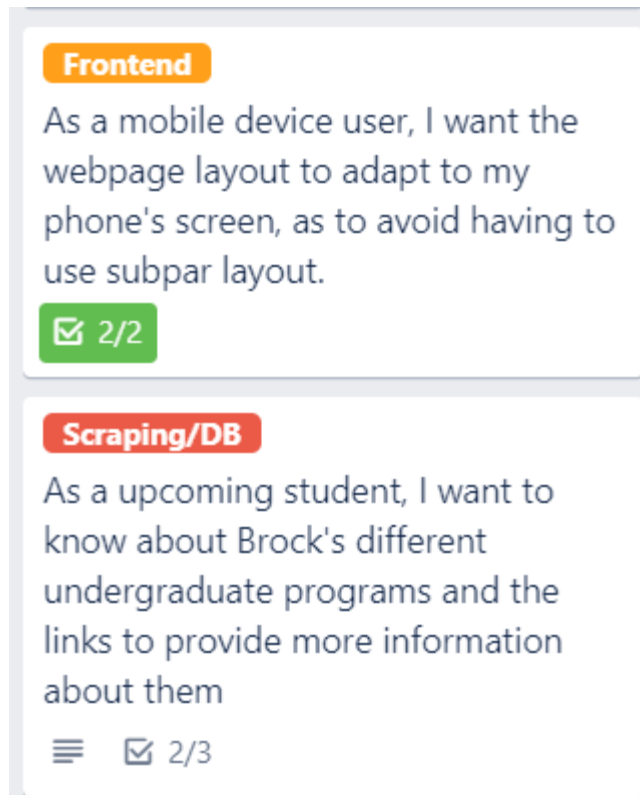**Frontend**

As a user, I would like a visual indication that the bot is currently working on an answer.

☑ 2/2

Frontend

As a user, I would like the text box to remain focused upon submitting a message.

☑ 1/1

Features we planned to Implement in the Following Sprints

Features planned for sprint 3

- More advanced chatbot features and better responses
- Larger database with additional relevant information about Brock
- More sophisticated front-end
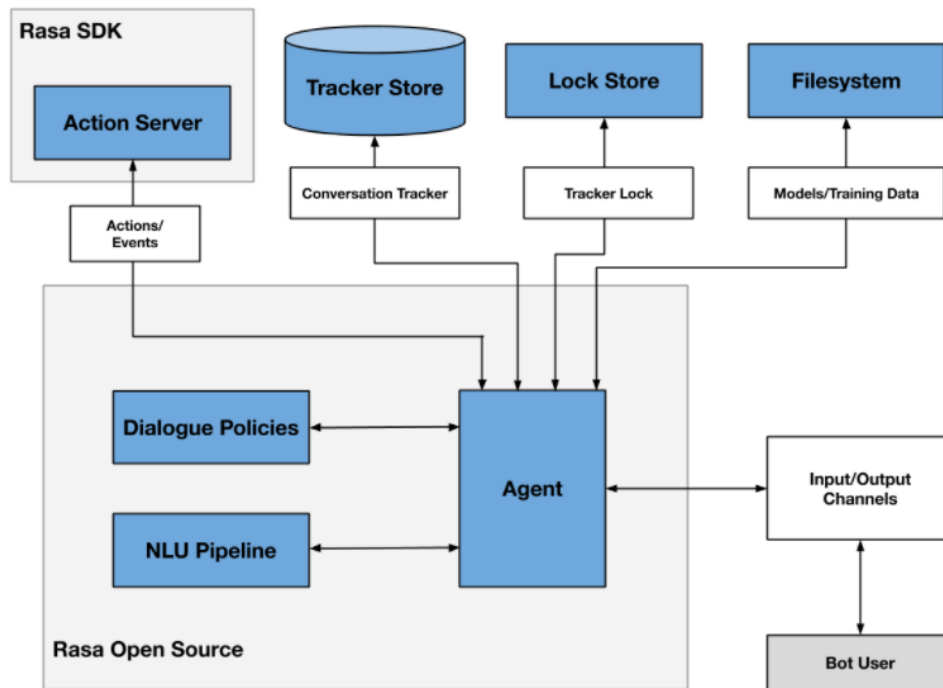
Features planned for sprint 4

- Not sure yet. Likely we will work on carryover from Sprint 3 and begin to think about testing and potentially building the Canada Games version of the bot.
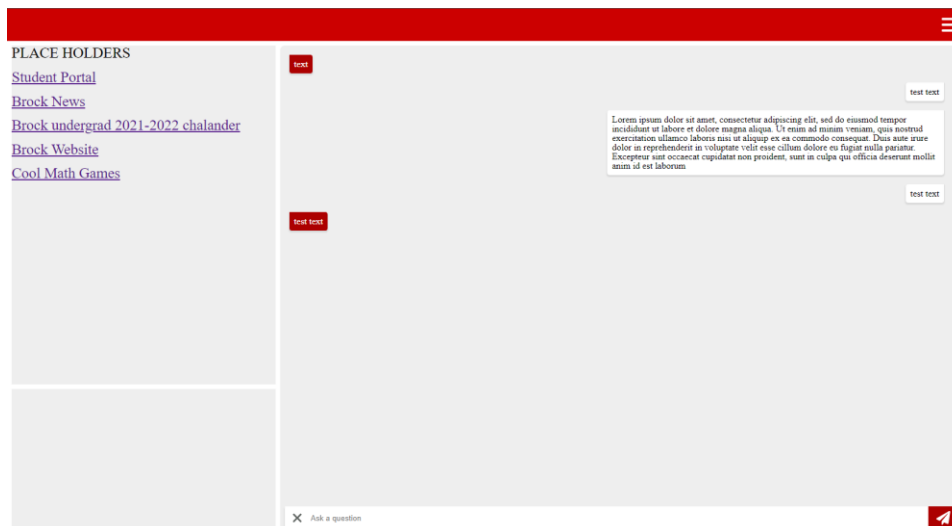
Screenshots of our System

RASA

{'intent': {'name': 'program_inquiry', 'confidence': 0.9999630451202393}
'entities': [{'entity': 'program_name', 'start': 14, 'end': 24, 'confidence_entity': 0.9184684157371521, 'value': 'Biophysics
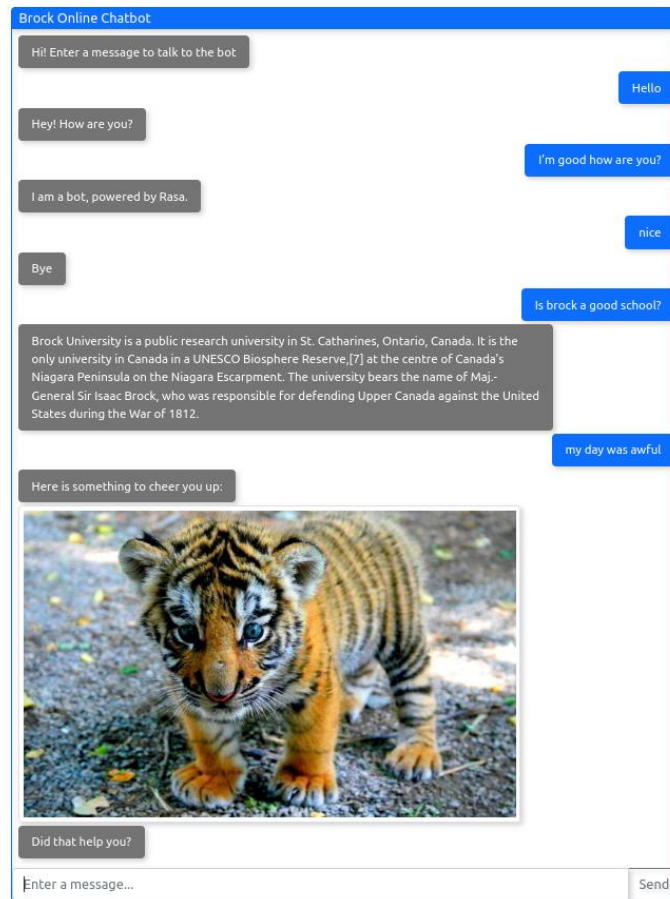
(Retrieved from: https://rasa.com/docs/rasa/next/arch-overview/)

Front-end

Our React Frontend

Our Prototype Frontend

Brock Online Chatbot

Hi! Enter a message to talk to the bot

Hello

Hey! How are you?

I'm good how are you?

I am a bot, powered by Rasa.

nice

Bye

Is brock a good school?

Brock University is a public research university in St. Catharines, Ontario, Canada. It is the only university in Canada in a UNESCO Biosphere Reserve,[7] at the centre of Canada's Niagara Peninsula on the Niagara Escarpment. The university bears the name of Maj.-General Sir Isaac Brock, who was responsible for defending Upper Canada against the United States during the War of 1812.

my day was awful

Here is something to cheer you up:

Did that help you?

Enter a message...   Send

Screenshots of MongoDB Atlas

Our Test Database. No data in it yet except for 5 test entries.

Cluster0

Overview   Real Time   Metrics   **Collections**   Search   Profiler   Performance /

DATABASES: 1   COLLECTIONS: 1

+ Create Database

Q NAMESPACES

▾ BrockInfo

| Courses

BrockInfo.Courses

STORAGE SIZE: 36KB   TOTAL DOCUMENTS: 5   INDEXES TOTAL SIZE: 36KB

Find   Indexes   Schema Anti-Patterns ⓪   Aggregation

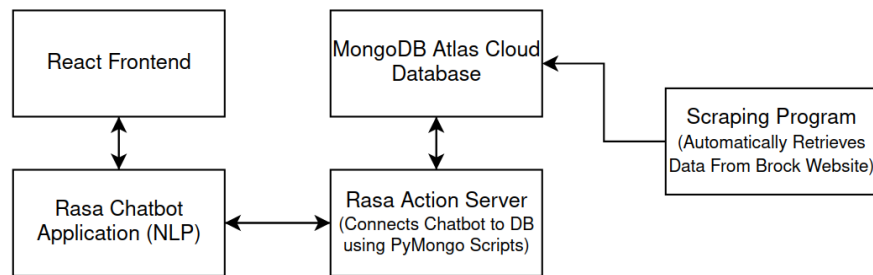FILTER { field: 'value' }

QUERY RESULTS 1-5 OF 5

```
_id: ObjectId("6205adc193155c8fbd9f02b8")
course_name: "Computer Architecture"
course_code: "COSC2P12"
instructor: "Dave Bockus"
```

```
_id: ObjectId("6205adc293155c8fbd9f02b9")
course_name: "Computer Networks"
course_code: "COSC4P14"
instructor: "Robson De Grande"
```

Screenshots of our Scraped Data (JSON Files)

```
EXPLORER                    ● actions.py    {} ProgramInfo.json U ✕    ! domain.yml    ! endpoints.yml    ! stories.yml
∨ SEARCH                    {} ProgramInfo.json > ...
 Search    Aa Abl .*     1  [
                         2   {"Name": "Accounting", "Link": "https://brocku.ca/programs/undergraduate/accounting/"},
 Replace   AB          3   {"Name": "Adult Education", "Link": "https://brocku.ca/programs/undergraduate/adult-education/"},
                         4   {"Name": "Applied Computing", "Link": "https://brocku.ca/programs/undergraduate/applied-computing/"},
                         5   {"Name": "Applied Linguistics", "Link": "https://brocku.ca/programs/undergraduate/applied-linguistics/"},
                         6   {"Name": "Biochemistry", "Link": "https://brocku.ca/programs/undergraduate/biochemistry/"},
                         7   {"Name": "Biological Sciences", "Link": "https://brocku.ca/programs/undergraduate/biological-sciences/"},
                         8   {"Name": "Biomedical Sciences", "Link": "https://brocku.ca/programs/undergraduate/biomedical-sciences/"},
                         9   {"Name": "Biophysics", "Link": "https://brocku.ca/programs/undergraduate/biophysics/"},
                        10   {"Name": "Biotechnology", "Link": "https://brocku.ca/programs/undergraduate/biotechnology/"},
∨ CHATBOT               11   {"Name": "Business Administration", "Link": "https://brocku.ca/programs/undergraduate/business-administr
 > frontend            12   {"Name": "Business Administration Co-op International Double Degree", "Link": "https://brocku.ca/program
 > JSON                13   {"Name": "Business Communication", "Link": "https://brocku.ca/programs/undergraduate/business-communicat
 > rasa                14   {"Name": "Business Economics", "Link": "https://brocku.ca/programs/undergraduate/business-economics/"},
 > webapp              15   {"Name": "Canadian Studies", "Link": "https://brocku.ca/programs/undergraduate/canadian-studies/"},
 .gitignore           16   {"Name": "Chemistry", "Link": "https://brocku.ca/programs/undergraduate/chemistry/"},
 {} AppliedHealthScince... 17   {"Name": "Child and Youth Studies", "Link": "https://brocku.ca/programs/undergraduate/child-and-youth-st
 {} CanadianStudiesCou...  18   {"Name": "Child Health", "Link": "https://brocku.ca/programs/undergraduate/child-health/"},
 {} ComputerScienceCo...   19   {"Name": "Classics and Archaeology", "Link": "https://brocku.ca/programs/undergraduate/classics/"},
 {} ComputerScienceFac...  20   {"Name": "Community Health", "Link": "https://brocku.ca/programs/undergraduate/community-health/"},
 courses.json         21   {"Name": "Computer Science", "Link": "https://brocku.ca/programs/undergraduate/computer-science/"},
 {} FacultyInfo.json    22   {"Name": "Computer Science and Mathematics", "Link": "https://brocku.ca/programs/undergraduate/computer-
 {} GoodManFacultyInfo... 23   {"Name": "Computing and Business", "Link": "https://brocku.ca/programs/undergraduate/computing-and-busin
 {} HealthScienceCours...  24   {"Name": "Computing and Network Communications", "Link": "https://brocku.ca/programs/undergraduate/compu
 {} HistoryFacultyInfo.json 25  {"Name": "Concurrent Teacher Education \u2013 Intermediate/Senior (Grades 7-12)", "Link": "https://brock
 {} MathematicsandScin... 26   {"Name": "Concurrent Teacher Education \u2013 Junior/Intermediate (Grades 4-10)", "Link": "https://brock
 {} MusicFacultyInfo.json 27   {"Name": "Concurrent Teacher Education \u2013 Primary/Junior (Grades K-6)", "Link": "https://brocku.ca/p
 📕 Product_and_Sprint...   28   {"Name": "Critical Criminology", "Link": "https://brocku.ca/programs/undergraduate/critical-criminology/
                        29   {"Name": "Dramatic Arts", "Link": "https://brocku.ca/programs/undergraduate/dramatic-arts/"},
                        30   {"Name": "Early Childhood Education", "Link": "https://brocku.ca/programs/undergraduate/early-childhood-
                        31   {"Name": "Earth Sciences", "Link": "https://brocku.ca/programs/undergraduate/earth-sciences/"},
                        32   {"Name": "Economics", "Link": "https://brocku.ca/programs/undergraduate/economics/"},
```

How the Subsystems in Our Chatbot Interact

```
┌─────────────────┐        ┌─────────────────────┐
│                 │        │ MongoDB Atlas Cloud │
│  React Frontend │        │      Database       │
│                 │        │                     │        ┌─────────────────────┐
└────────┬────────┘        └──────────┬──────────┘        │  Scraping Program   │
         │                            │                   │(Automatically Retrieves│
         │                            │         ◄─────────│ Data From Brock Website)│
         ↕                            ↕                   └─────────────────────┘
┌─────────────────┐        ┌─────────────────────┐
│  Rasa Chatbot   │◄──────►│ Rasa Action Server  │
│Application (NLP) │        │(Connects Chatbot to DB│
│                 │        │ using PyMongo Scripts)│
└─────────────────┘        └─────────────────────┘
```

Issues encountered

Issues with Scrum

With most of us being new to practical development using Scrum methodology. It was an overwhelming experience to translate user stories effectively with modest estimates. Creating a new application from scratch it was difficult to design user stories and tasks that would help us get started with development in manageable phases.

To start with we had broad functionality written down as user stories that required a large amount of work to be completed. Our first issue was underestimating the work items we selected for sprint 1. At the end of Sprint 1, all but 3 user stories had to pushed to next sprints.

After our first review we received some useful feedback from our stakeholder Amirali Madani, that gave us a fresh perspective on our technique. We sat down, refined our backlog, and broke down large user stories into more manageable and valid smaller user stories. This time we were careful with our estimates and managed to demo all but 1 user story at our second sprints review.

Issues with MongoDB

The 1 user story that we could not manage to complete at the end of sprint 2 was due to another issue that we faced. This issue with our MongoDB Atlas server prevented us from establishing a connection to the database server. The user story in question is given below. As you can see, we managed to scrape the data for the user story, but we could not push it to the database in its table. The team currently is working hard to fix this issue and soon we should be on track to complete our next sprint.
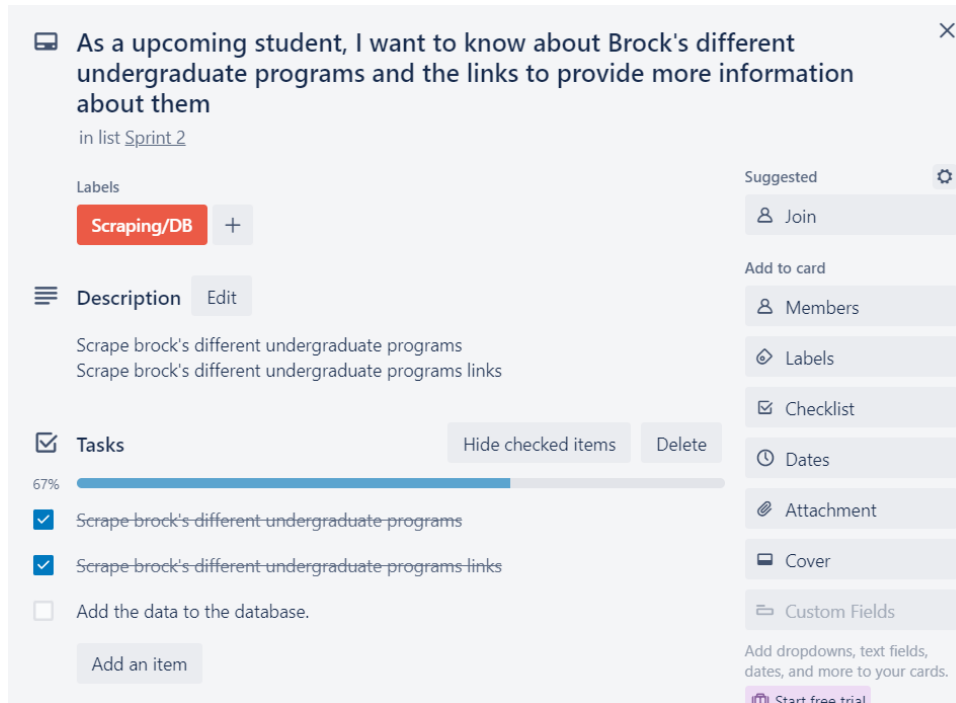
Issues with Rasa (Our Chatbot NLP System)

The NLP team has run into a few issues with the NLU system in Rasa. At first, we thought that recognizing "entities" within a question would be straightforward. Entities, in Rasa, are just words that fill up a slot in a particular intent, essentially variables within a sentence. For example, in our example data that is supposed to recognize when a user asks about an undergraduate program, we are trying to have the actual program name be recognized as an entity, so that we can extract it and query the database with that string. For some words, it works, but for others, it doesn't.

This issue with the entities is a bit technical and challenging to explain properly here, but it is an important one for us to fix. We just ran into it, so hopefully we can figure it out within the next few days. After that, we hope to be able to speed up the process of building the Rasa bot.

Issues with web scraping

Some of the challenges with web scraping we have faced were information being protected and therefore not visible when scraped. The scripts also needed to be modified as in multiple cases information was not present returning null values. To deal with this issue we are making changes to the script to sanitize the collected data.

Contributions of each Team Member

Abdul:

As the scrum master it was my duty to instill scrum practices and philosophy in my teammates and leverage it to deliver seamless and planned development experience. For the first sprint we encountered a couple of issues that we were able to address in the following sprint after receiving

feedback from one of the stakeholders in this project. These issues are described in detail in the 'Issues encountered' section of this report. Apart from managing the backlog, I helped facilitate our weekly scrum meetings where we discuss the progress we have made throughout the whole week. We update each other with new findings and any impediments faced by each other.

In my developmental role, I have taken up the task of designing the NoSQL database we need to be working with. My team focuses on scraping data from Brock university's website and my specific tasks are to create the relevant tables and adding the scraped data to the database. For now, what we have managed to do is create a dedicated scraper script that can be modified to scrap data across all webpages available in Brock University. Then I created a script to update the database with the scrapped JSON files.

To aid in enforcing a schema in a NoSQL database that does not have a schema or relationships. I created a high-level data model that lists all the tables, documents, and their fields that we need to aggregate in order to provide efficient answers to the user's query.
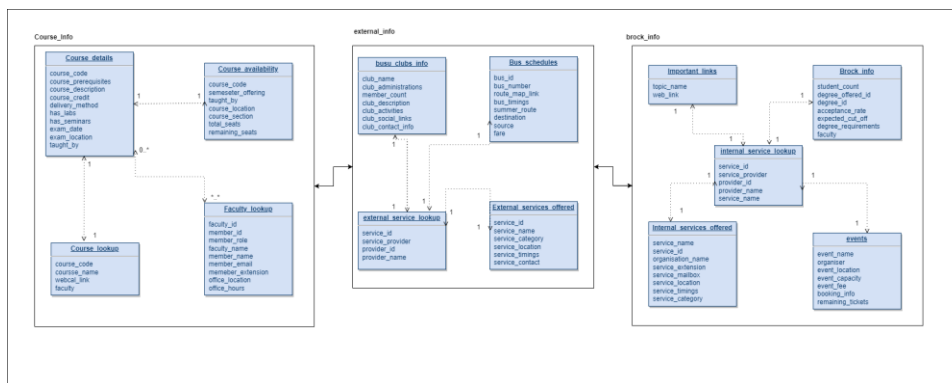
Aws:

In the first sprint, I took time to research the possible tools that could be used for the NLP part. I was going in the direction of writing a lot of original code, however, I quickly realized that this would be a lot of unnecessary work. Furthermore, this would not have been viable, as we do not have that much time to complete this project. Once it was decided that we would use RASA as the main bot component, I took time to learn how it works. It is a useful application; however, it can become complicated when the custom code must be written (which is our case).

In the second sprint, I began implementing some basic custom code to answer questions regarding Brock's programs. Due to the fact that we have problems with getting MongoDB up and running, I have stored the program data locally for retrieval. After training the model, the agent can achieve very high confidence levels when identifying an inquiry about a program at Brock. Most of the time, it is able to retrieve the correct program information from the data, however, sometimes it is unable to extract the program name from the user input, so it does not return anything. My plan is to straighten out this bug (possibly by using NLU pipelines) in the current sprint. Once we can get this working, then the framework can be used for all other inquiries.

Anwar:

For the first few sprints I will be working with the Scraping/DB team and upon completing those user stories, I will be assisting the NLP team going forward. Initially I worked with the NLP team in outlining the webpages to be modelled.

For the first sprint I familiarized myself with Python and web scraping and identified libraries to use. I also study the Brock website to find relevant pages which contain information that can be used to satisfy the user stories. In the second sprint I scraped the Brock club directory for information on clubs and their contact information and wrote a script to get the details of various programs offered at brock and their corresponding home sites. In both cases python scripts were used along with libraries like Beautiful Soup.

Ifti:

I worked on scraping data which will be transferred to our MongoDB database of our Chatbot project. I used scrapy tool along with PyCharm editor to scrap data to be stored in our database. Initially, I scraped all the course codes of different undergraduate programs available in our brock university. I used different Python scripts to scrap different course codes for each of the undergraduate programs in Brock University and then store those data in json format. Then, I used the same technique to grab all the data regarding the faculty information for each of the departments of Brock University. Then, I uploaded all those json files to our GitHub repository so that those json files can be transferred from there to our database system. As far as the user stories are concerned, I completed 1 user story requirement in sprint 1 and 1 user requirement in sprint 2. In addition to that, I am planning to scrap as much data as I can so that our database can be filled with as much information as possible.

Ray:

During the first sprint, I didn't focus much on the user stories, but instead tried to get a prototype version of a chatbot working. Most of my time was spent researching the various techniques and open-source libraries that people use to build chatbots and discussing with the team which one we liked best. We also spent a lot of time trying to figure out what everyone's role would be. Eventually I decided to focus on NLP, and try to help on front-end, wherever possible. So far most of the work I've done has been with NLP and the Rasa bot.

During the second sprint, I worked with the front-end team a bit more to try and learn how they were building the front-end. Over the weekend Austin showed Neil and I how to use React to set up a front-end application, and I explained how to connect to the Rasa bot that the NLP team and I had begun to set up in the previous sprint. After some brief setup issues, we were able to connect the Rasa bot to the front-end application.

I also tried to continue working on the Rasa bot, to make it more sophisticated. While working on our first user story of the sprint, "As an upcoming student, I want to know about Brock's different undergrad programs", I tried to figure out how we would successfully recognize certain questions from the user and have a python script return a result. I eventually figured out that we needed to run the python scripts from an action server, a built-in feature of Rasa that allows you to run python scripts alongside

the Rasa bot and communicate with a database via the chatbot. I didn't make as much progress in this sprint as I wanted to, since, although we did figure out how to communicate with our database via the chatbot, we have a long way to go before it is sophisticated enough to get meaningful information out of it. In the next sprint I hope that we can iron out the issues we're having with our NLU so that we can finish all our user stories for that sprint.

Nilaanjann: During the first sprint, my primary focus was on getting the rough planning of the project fleshed out and focusing on areas of improvement, areas that would be required to be implemented in the project. As such, when it came to backend work, and our decision of using MongoDB, I decided to spend considerable time committing myself to learning the language and brushing up on it. Once done with that, as I was a member of the subgroup dealing with the NLP or natural language processing area of the project, we ran discussions on what the best way to tackle this situation would be and went through independent research in order to look for optimal NLP sources. After deciding on using Rasa, I deep dived into the documentation and set it up on my local machine in order to better understand it and test it out with sample inputs.

Once the second sprint came along, Ray, Aws and I set up several meetings in order to work through further implementations and ways that we could incorporate Rasa into our project while considering the database setup and the scraping team's progress. Eventually, progress was made on the scraping part, and we had real data to work with. We still lacked some knowledge regarding the frontend and as such, Austin came in and helped guide and show us the front-end setup. Towards the end of the sprint, Aws and I recognized some bugs with the sample data processing. Upon recognizing them, me and Aws decided to focus our attention on fixing the recognized bugs as it would be essential moving forward with the project.

Amots:

I spent the first week of the first sprint earning a base level understanding of the different programming languages and platforms we decided on using. near the end of the first I started work on the front-end of the chatbot by looking at chatting applications and designing potential presentations for the website. By the end of the first sprint, I finished the final design of the webpage layout with the team's input.

The second sprint was mostly used for familiarizing myself with React and the rest of the tools Austin used to create the front-end base implementation. To solve scrum problems from sprint 1, me and Austin created more concise user stories to be implemented; prioritizing requirements that must be implemented over the extra requirements listed. I completed the page reformatting for different devices user story as well as the clear input button assigned for sprint 3. The reformatting for the page takes account of page resizing and mobile devices' ability to switch from landscape to portrait view and back. Finally, I started switching some of the strict sizing of the page to work with other screen sizes, but this would ultimately be pushed to a later sprint to allow more components to be added first.

Austin:

I was responsible for configuring the frontend toolchain including react, webpack and other dependencies and development tools. I created the general structure and organization of the react application. I contributed to the creation, styling and functionality of UI elements according to the

team's design mockup. These include the header bar, info panels, chat panel, messages, chat bar, submit message button and a few other minor UI elements. I also helped the AI team with configuring the RASA bot to recognize entities specified in user inputs and save them to slots that may be accessed in action scripts.

In the second sprint I contributed the completion of tasks corresponding to four user stories. Firstly, a dedicated submit button which will allow the user to submit a message they have typed in the chat bar. This required the creation of the markup and styling for the button as well as the JavaScript necessary to submit the text input from the chat bar and send it to the RASA bot. Secondly, I created an error popup that informs the users when they have tried to submit an empty input from the chat bar. Thirdly, I implemented the necessary system to present a message from the bot to indicate when the bot has received a message and is working on an answer. Lastly, I ensured that the chat bar remains focused when a user submits an input.