

1 - Write the name of the group and the list of your group members

Team Name: Search Party

Greg Pogue – 4583993

Joel Jacob – 6603245

Madeline Janecek – 6436620

Sam Langdon – 6180137

Brendan Park – 6541288

Kylee Schram – 6131726

GitHub: <https://github.com/janecek/COSC4P02Project2022>

2 - Include a brief list/description of the features (subsystems) you planned to implement in each sprint and whether or not you accomplished them

Sprint 3:

Preprocessing and populating database with course and offering info

Create a docker container for the application to assist in server deployment

NLP to Database querying:

- Research into SQLAlchemy querying methods
- Route keywords to query template

NLP:

- Evaluation of NER for detection of locations, people
- Formatting of extracted keywords for use by database/querying
- Creation of spelling correction system

Develop splash page with options for Brock University and Canada Games

Testing and deployment research

Sprint 4:

Database querying using NLP keywords

- Filter keywords to match database formatting

Abstract SQLAlchemy data models from server class

Spell correction preprocessing layer added to NLP pipeline

NLP:

- Integration of spell checker into the NLP pipeline
- Addition of link responses for general questions and difficult questions where we are unable to get an answer from the database for the given query
- Debugging/improvement of keyword extraction patterns
- Implementation/integration of NER for detecting peoples' names
- Generation of text responses using data returned from the database for a selection of queries

Produced unit test cases for different functions within the application

Research testing automation

Deployment pipeline to Heroku from GitHub main branch

Sprint 5 (currently ongoing):

Course table updates

- Adding additional information (title, restriction, and format)

Offering functionality

- Complete query options for Offering table
- Generate responses to Offering queries

Exam functionality

- Preprocessing exam data
- Add queries for Exam table
- Generate responses to Exam queries

NLP

- Expanding response generation capabilities
- Increasing keyword extraction scope
- Enhancing spell check

Automate testing

Automate deployment

Incomplete:

- Database querying using NLP keywords (From Sprint 3 moved to Sprint 4)
- Response generation (From Sprint 3 moved to Sprint 4)
- User story: Wifi only option for mobile devices (Rejected)
- User story: Chatbot features to be raised above the keyboard on mobile devices (Rejected)
- User story: Frame rate limiter to manage power consumption (Rejected)

3 - Include a brief list of features (subsystems) that you plan to implement in the following sprints

Brock University Chatbot:

- Important dates
- Building codes
- Program information

Canada Games Chatbot:

- Separate database
- Additional data collection
- Create new NLP matching patterns

Testing:

- Develop an automated testing framework using PyTest

NLP:

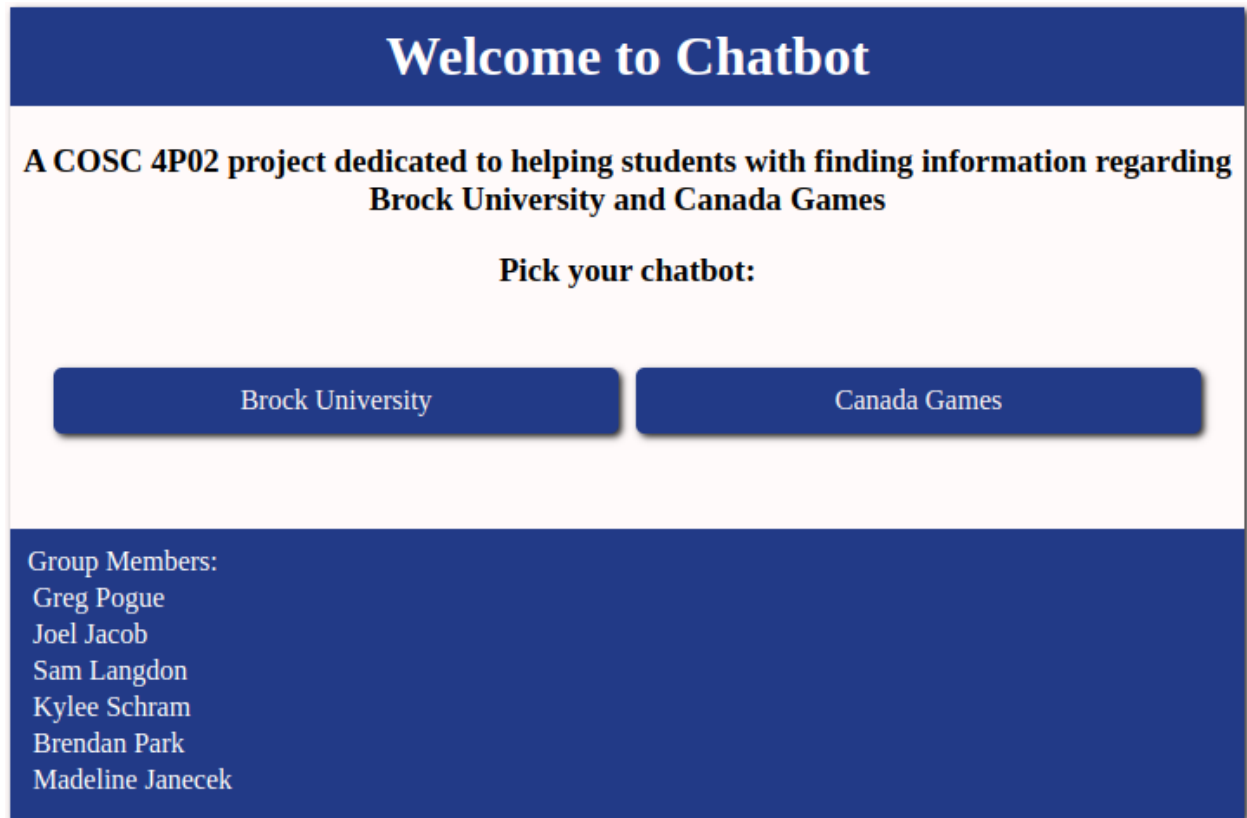
- Autocorrection of course codes and similar terms not in the default dictionary of the current autocorrection system
- Expanding support for a wider variety of user questions

Front-end:

- Refining the front-end user interface

4 - Include screenshots if you have a working version of the system

Splash Page:



Brock University Chatbot:



BrockChatBOT



Hello! Welcome to the Brock chat bot! What can I help you with today?

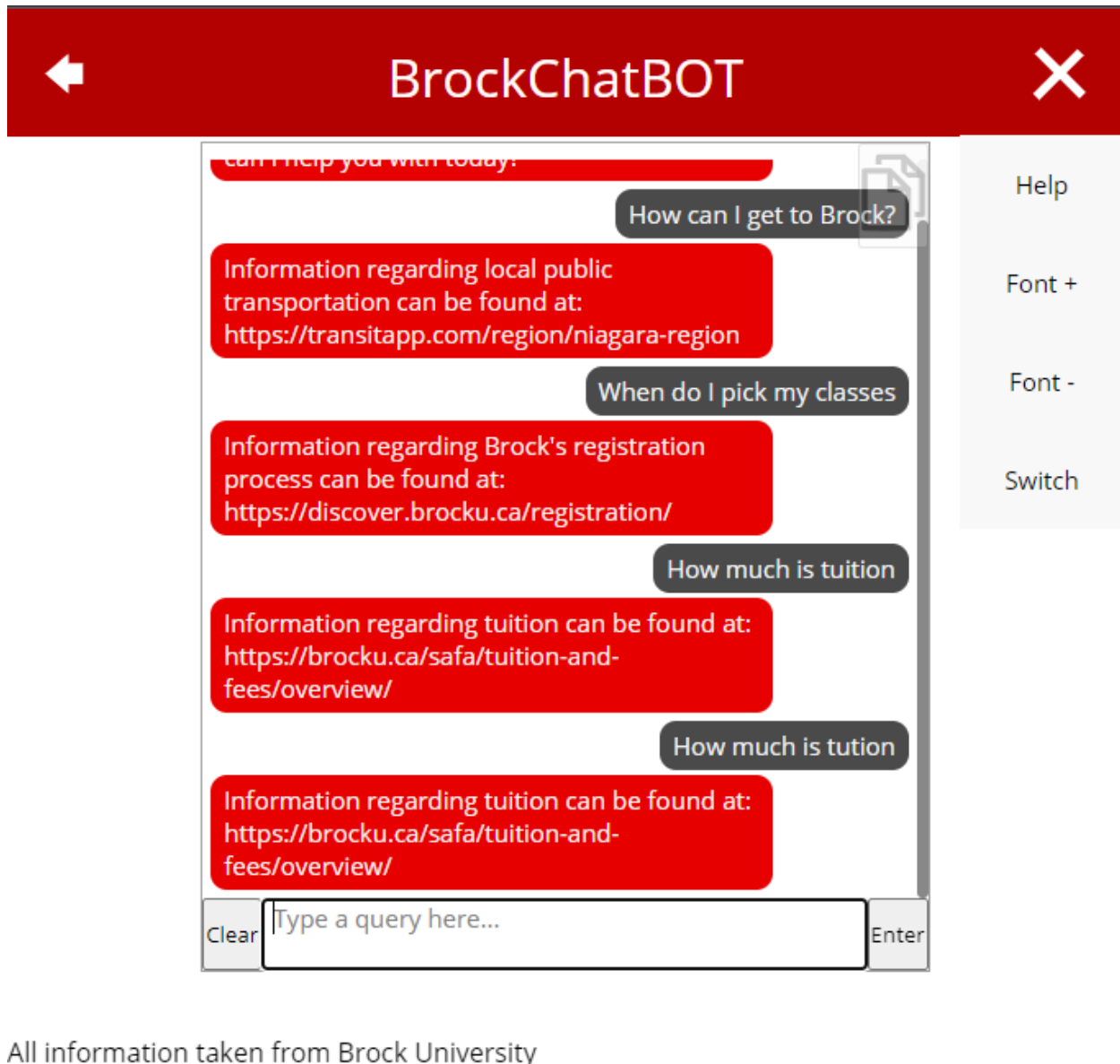


Clear

Type a query here...

Enter

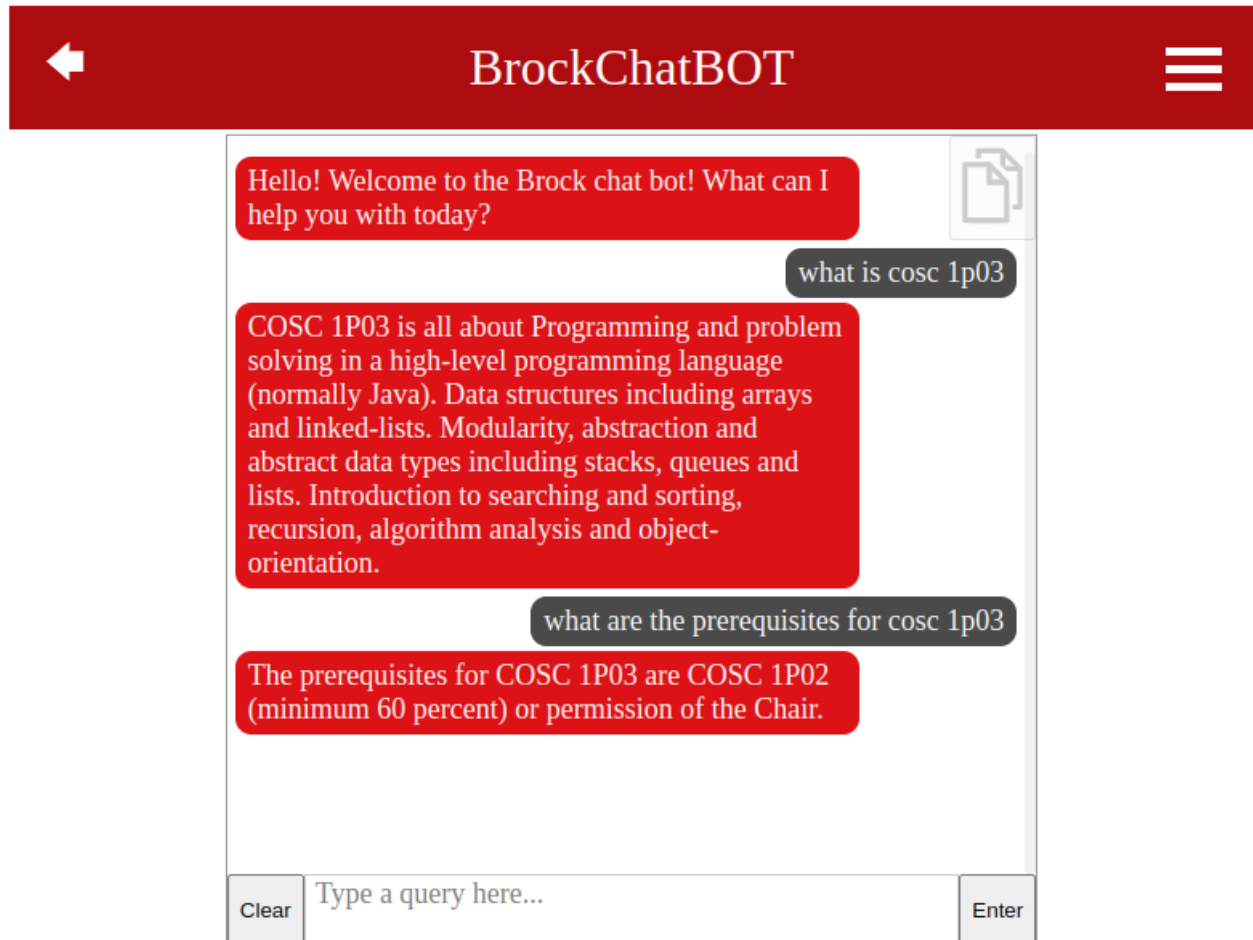
All information taken from Brock University



All information taken from Brock University

The features shown in the above screen shot include:

- Menu bar that opens when clicked - includes help button, font resizing buttons, and button to toggle between two chat bot modes (Canada Games and Brock)
- Clear and Enter buttons for user input
- Automatic resizing text field for user input
- Message display that automatically scrolls down when size of conversation exceeds the allotted area
- Copy button - copies the entire conversation to the user's clipboard
- Spell check - The chat bot is capable of identifying small typos (i.e. 'tution' is recognized as a misspelled tuition)



The above screenshot shows response generation for queries returning course description and prerequisites, both found in the Course table.

5 - Mention any issues you encountered

Database:

- Early on in sprint 2 and 3, data was preprocessed to be compatible with the database. The consistency of data over tables was overlooked, as the preprocessing was performed by two people and the data was received from Brock in multiple different formats. This issue became apparent when we started querying the database in sprint 4, as this was the first time we were required to write functional code using the queried data. This led to additional manpower to further clean the data in the following sprints.
- When building the SQLAlchemy connection to the database, the object relational models are stored in the main server.py file. As we added more columns and tables we realized that these should be abstracted from this file to prevent it from ballooning in size. This solution for a modular design would also positively impact the addition of a second database. In doing so we ran into a circular import error in Python. To solve this we replaced the one instance where there was a global import that could be refactored into

a local import. This solved our issue and led to the now modular models.py which holds the object relational models of our database.

NLP:

- Finding effective autocorrection technique. (Lots of ineffective methods were looked at before we found the one we settled on)
- Finding ways around blocking tasks such as certain data yet to be posted in the database being needed for certain NLP queries

Testing:

- Determining what functions needed to be modularized out for unit testing - ie, some NLP functions were closely integrated with database functions, but if we could unit test the interior functions before the larger one combining them that would be sufficient

Front-end

- Copy functionality does not work in some environments

6 - Describe the contributions and achievements of each member of the group

Greg Pogue

- Contributes to: Database, Hosting, Containerization
- Achievements:
 - Populating and preprocessing data for database and SQLAlchemy Course model
 - Solved database querying and routing keywords from NLP
 - Modularized back-end code to assist with Canada Games Expansion
 - Assisted with Docker containerization and deployment to Heroku

Joel Jacob

- Contributes to: Hosting, Containerization, Testing
- Achievements:
 - Developed the pipeline in Heroku to continuously deploy code
 - Worked on Containerization of app using Docker to easily package and deploy web app
 - Researched into Pytest modules for Python, and developed unit test cases for the application.

Madeline Janecek

- Contributes to: Front-end, NLP, Hosting
- Achievements:
 - Design and develop splash page
 - Assisted with setting up Heroku to deploy code
 - Added to existing match patterns to accommodate a larger range of questions
 - Response generation for questions where the database is not involved or when information can not be retrieved from the database

Sam Langdon

- Contributes to: NLP

- Achievements:
 - Integration of NER for detection of person names into the NLP pipeline
 - Debugging/improvements of existing match patterns
 - Response generation using information retrieved from database
 - Added building code keyword extraction to NLP

Brendan Park

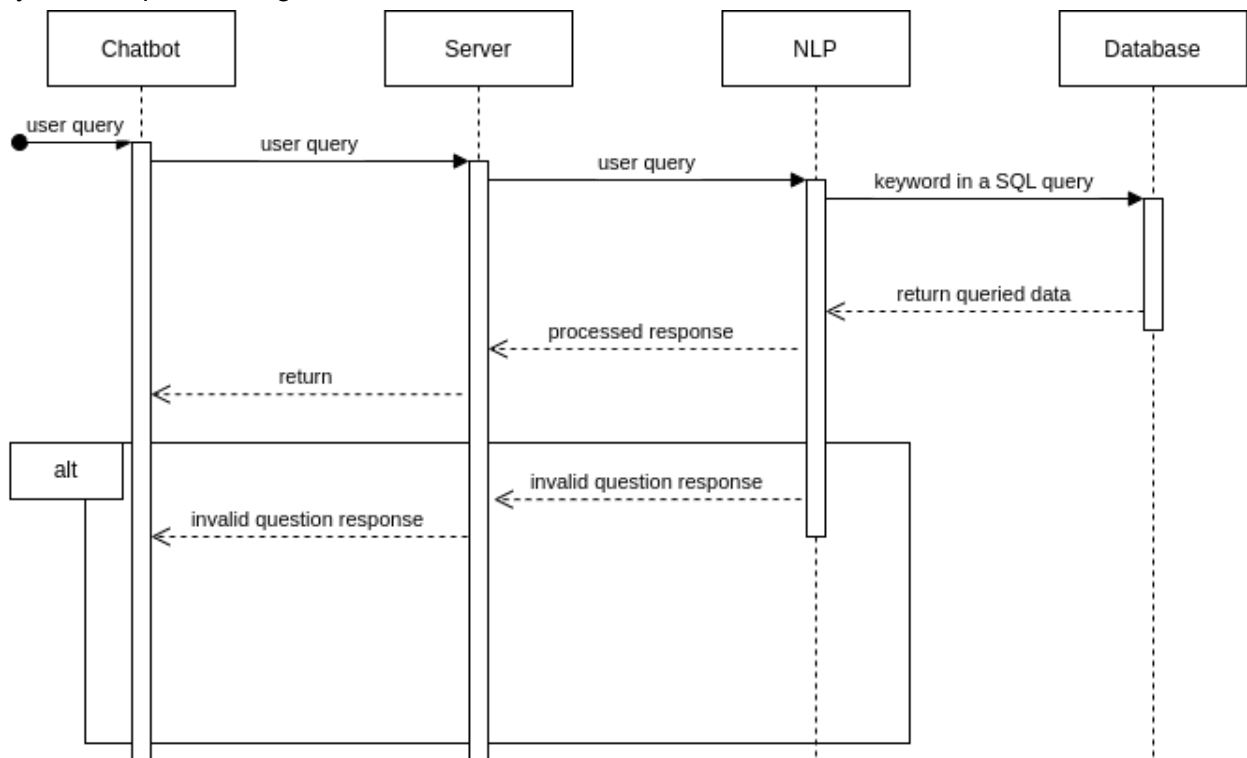
- Contributes to: NLP
- Achievements:
 - Integration of effective spell correction as preprocessing step to NLP pipeline

Kylee Schram

- Contributes to: Database, Testing
- Achievements:
 - Writing test case outlines for unit and integration tests
 - Researching pytest, python assert, and .contains() for testing
 - Completing Offering table for the database including the SQLAlchemy model and preprocessing
 - Assisted with database querying and population

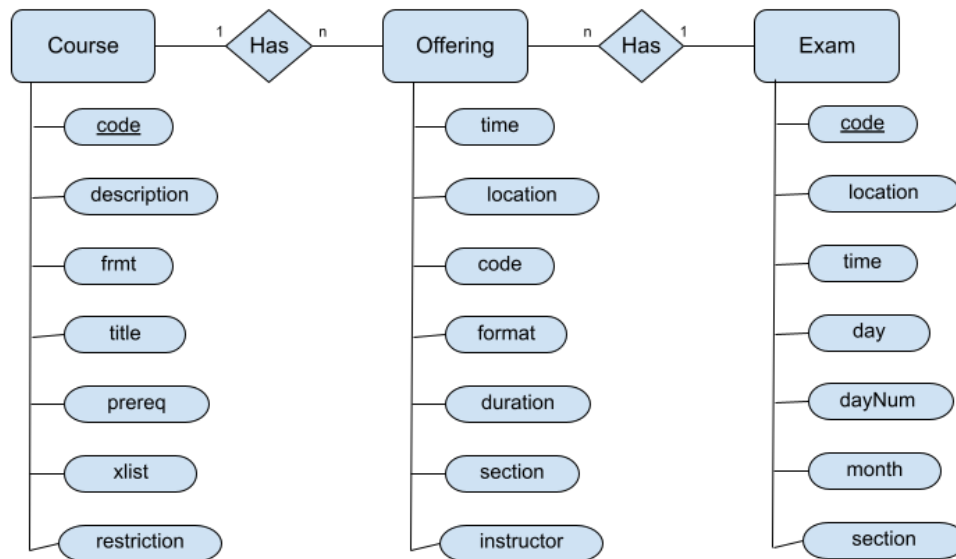
7 - Diagrams

System sequence diagram:

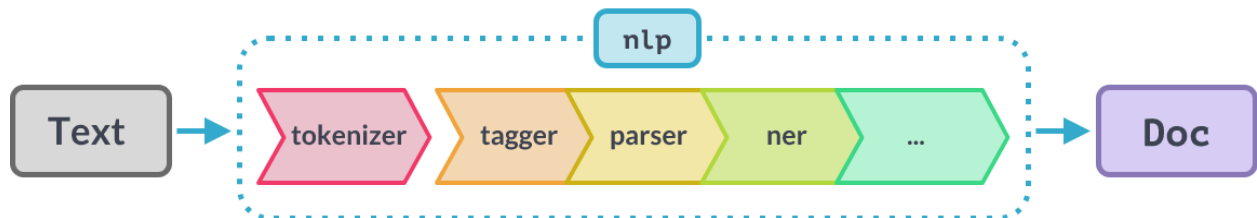


Entity-relationship diagram:

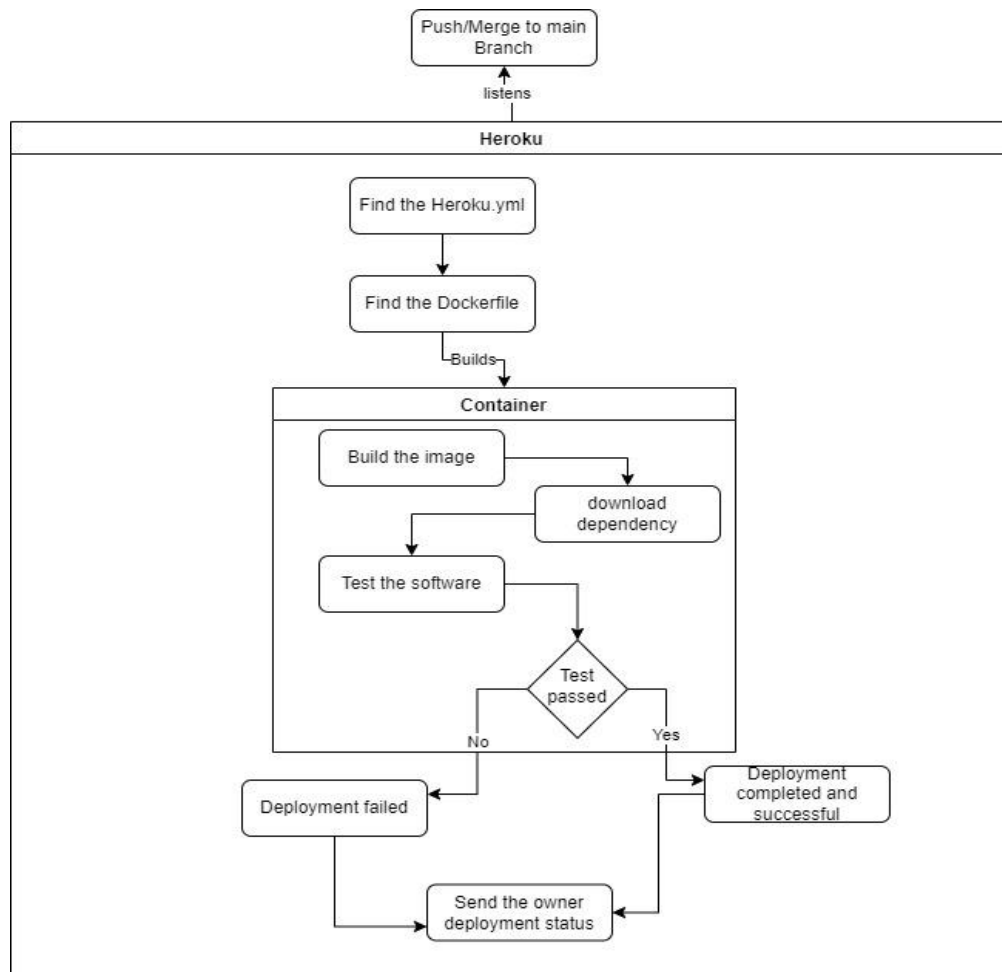
BU Chatbot



NLP Pipeline (ref: <https://spacy.io/usage/processing-pipelines>):



CI/CD model



Test Case Outlines

Function	Input	Expected Output
BotNLP.processQ	This is not a real question	"I am not quite sure what you're asking. Could you rephrase that?"
BotNLP.processQ	What is the prereqs for COSC 1P03	String containing "COSC 1p02"
BotNLP.processQ	What aer the Prereq for COSC 1p03	String containing "COSC 1p02"
BotNLP.processQ	Qitat is jet prereq for COSC 1p03	"I am not quite sure what you're asking. Could you rephrase that?"
BotNLP.processQ	Wat is prereqs for COSC 1P03	String containing "COSC 1P02"
BotNLP.getLink	["prereq"]#ask maddy	String containing "https://brocku.ca/webcal/undergrad/"
# test all links	# ask maddy	
BotNLP.formResponse	Database_answer{"code": COSC 1P03, "prereq": 1P02}, keys = null	String "The prerequisites for COSC 1P03 are COSC 1P02"
BotNLP.formResponse	Database_answer="im in danger", keys="prereq"	String containing "https://brocku.ca/webcal/undergrad/"
BotNLP.formResposne	database_answer="im in danger", keys = ""	String containing "https://brocku.ca/"
BotNLP.processKeywords	extractKeywords to work #question for sam and kylee	
BotNLP.extractKeywords	#joel and kylee and NLP team	
botNLP.processQ	HelloHello	String "Hello", exactly once
botNLP.processQ	What are the prereqs for *1_ 1P02 What are the prereqs for /**/ *1_	"I am not quite sure what you are asking. Could you rephrase that?" Don't want Info for all courses dumped
FRONTEND	<script>alert("AHHH");</script>	Have to do this one manually