University of Ottawa
Department of Electrical and Computer Engineering

GNG 5125/ Data Science Applications

Professor: Arya Rahgozar

Project Report on Chatbot Implementation

Submitted by

Group: DSA_202101_ 7

Yusri Al-Sanaani	300216450
Hetvi Soni	300200976
Tavleen Kour	300213090
Immanuella Iyawe	300150838

Due Date: Monday, 22st March 2021

1. Introduction

The assignment requires us to implement a basic chatbot using Google's dialogflow tool. We took the opportunity and created a chatbot which we think would be very helpful to its target audience.

Topic: Frequent questions asked by the first-year newly admitted students at a university.

We took into consideration all the challenges we faced once we started studying in the university so we came up with a topic which would help students with a similar situation as ours when they start studying themselves.

Our approach included the following steps:

Step 1: Come up with the competency questions.
Step 2: Create an ontology of our thoughts using Protege.
Step 3: Think about all the sub-sections involved with our competency questions.
Step 4: Set up Google Dialog Flow account.
Step 5: Create Intents for all the competency questions and sub questions.
Step 6: Come up with different test cases which involved multiple training questions and their answers.
Step 7: Implement the chatbot and test it thoroughly to find the edge cases.
Step 8: Figure out the weak spots of our implementation and areas of improvement.
Step 9: Add code for the chatbot on the web page to validate the final product.

2. Ontology

Ontology defines the concepts and relationships used to describe and represent an area of knowledge. In this assignment, we used Protégé 5.5.0 for our topic.
Classes are the focus of most ontologies. Classes describe concepts in the domain. For example, a class of students represents all students. Specific students are instances of this class. The new admitted student is an instance of the class of students. A class can have subclasses that represent concepts that are more specific than the superclass. For example, we can divide the class of students into new students and old students.
Slots describe properties of classes and instances:
Registration Office can issue the uOttawa ID for new students by online application, so the ID card has slots such as body, way of application, the issuer, and receiver.
In our situation, all instances of the class **Inquiries** and its subclasses have slot produces that from **Campus Services** class to **Students** class.
In practical terms, developing an ontology includes:

- Defining classes in the ontology.
- Arranging the classes in a taxonomic (subclass–superclass) hierarchy.
- Defining slots and describing allowed values for these slots.
- Filling in the values for slots for instances.

We can then create a knowledge base by defining individual instances of these classes filling in specific slot value information and additional slot restrictions.

Competency Questions

Q1: How can I get to know my university?
Q2: Where can I find solutions for academic issues?

Q3: How to get information related to jobs and job searches?
Q4: How to get information regarding scholarships and fees?
Q5: How do I secure my rights and safety at the university?
Q6: How does the university's health insurance scheme work?

A. Determine the domain and scope of the ontology

Based on our competency questions, we plan to use this ontology for first year newly admitted students at a university to provide them answers for their most frequent questions.

B. Define the classes and the class hierarchy

A top-down development process has been used to define the most general concepts in the domain and subsequent specialization of the concepts. Figure 1, 2, and 3 show the classes hierarchy.

C. Define the properties of classes-slots

The classes alone will not provide enough information to answer the competency questions. Once we have defined some of the classes, we must describe the internal structure of concepts.

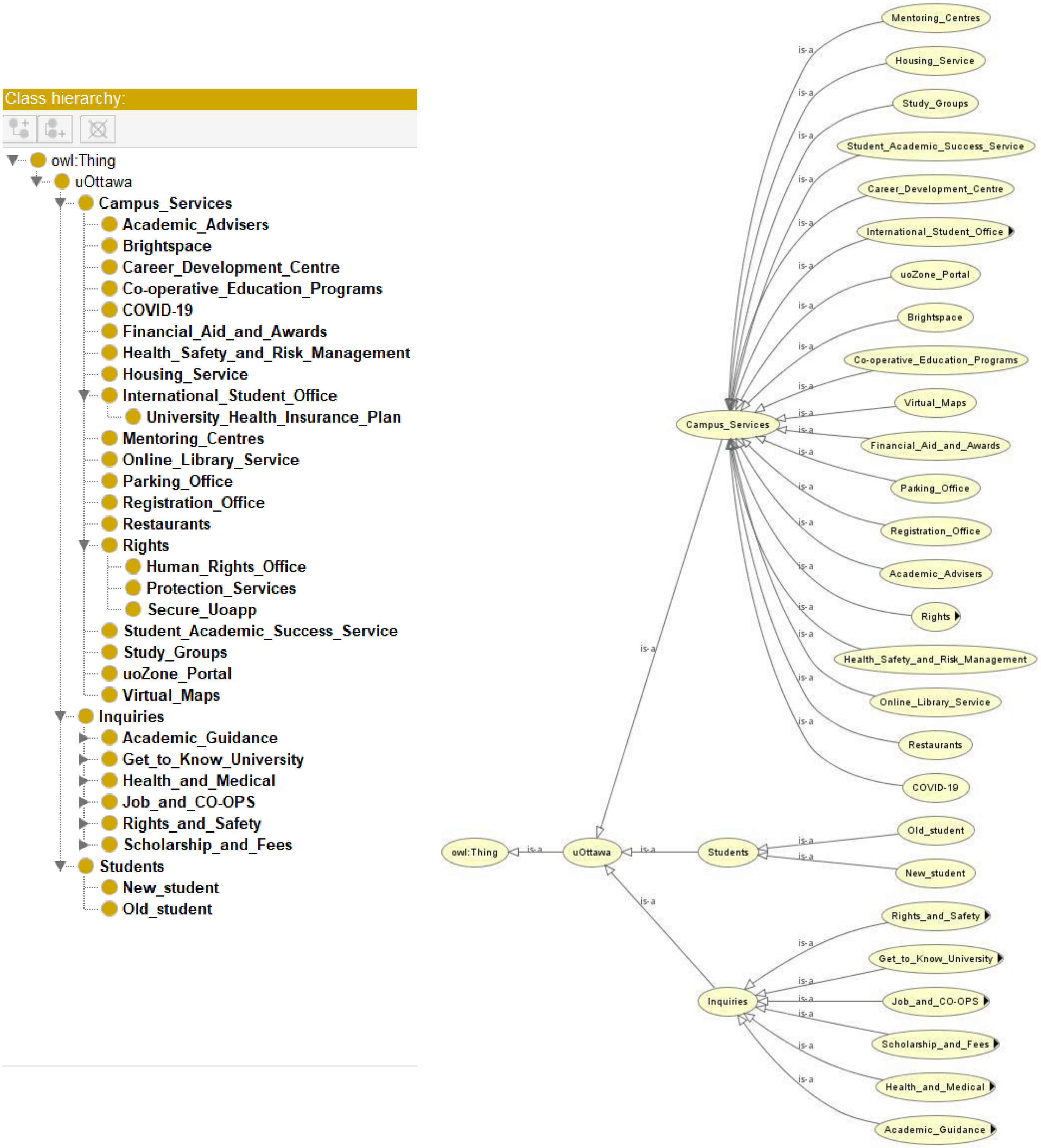
We have already selected classes from the list of terms. Most of the remaining terms are likely to be properties of these classes. For each property in the list, we must determine which class it describes. These properties become slots attached to classes. Figure 4 shows some created properties of classes-slots.

D. Domain and range of a slot

Allowed classes for slots of type Instance are often called a range of a slot. The classes to which a slot is attached or a classes which property a slot describes, are called the domain of the slot. In the systems where we attach slots to classes, the classes to which the slot is attached usually constitute the domain of that slot. Figure 4 shows some created properties of classes-slots, showing the domain and range of a slot as well.

E. Create instances

The last step is creating individual instances of classes in the hierarchy. Defining an individual instance of a class requires (1) choosing a class, (2) creating an individual instance of that class, and (3) filling in the slot values. Figure 5 and 6 show some examples of created instances.



Class hierarchy:

- owl:Thing
  - uOttawa
    - Campus_Services
      - Academic_Advisers
      - Brightspace
      - Career_Development_Centre
      - Co-operative_Education_Programs
      - COVID-19
      - Financial_Aid_and_Awards
      - Health_Safety_and_Risk_Management
      - Housing_Service
      - International_Student_Office
        - University_Health_Insurance_Plan
      - Mentoring_Centres
      - Online_Library_Service
      - Parking_Office
      - Registration_Office
      - Restaurants
      - Rights
        - Human_Rights_Office
        - Protection_Services
        - Secure_Uoapp
      - Student_Academic_Success_Service
      - Study_Groups
      - uoZone_Portal
      - Virtual_Maps
    - Inquiries
      - Academic_Guidance
      - Get_to_Know_University
      - Health_and_Medical
      - Job_and_CO-OPS
      - Rights_and_Safety
      - Scholarship_and_Fees
    - Students
      - New_student
      - Old_student

Figure 1:Classes hierarchy for uOttawa, showing the campus services, students, and inquiries.

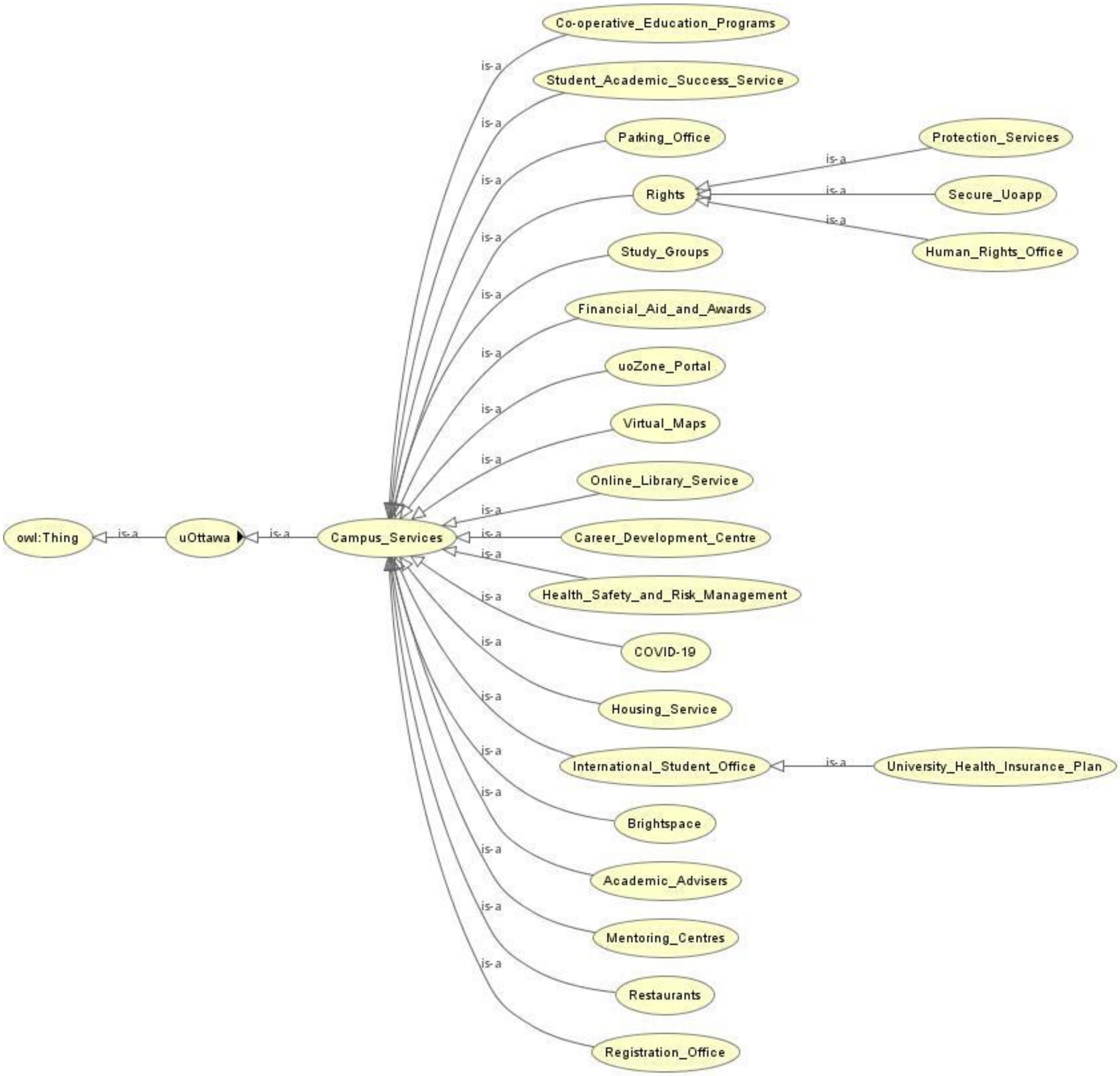


Figure 2:Classes hierarchy for campus services.

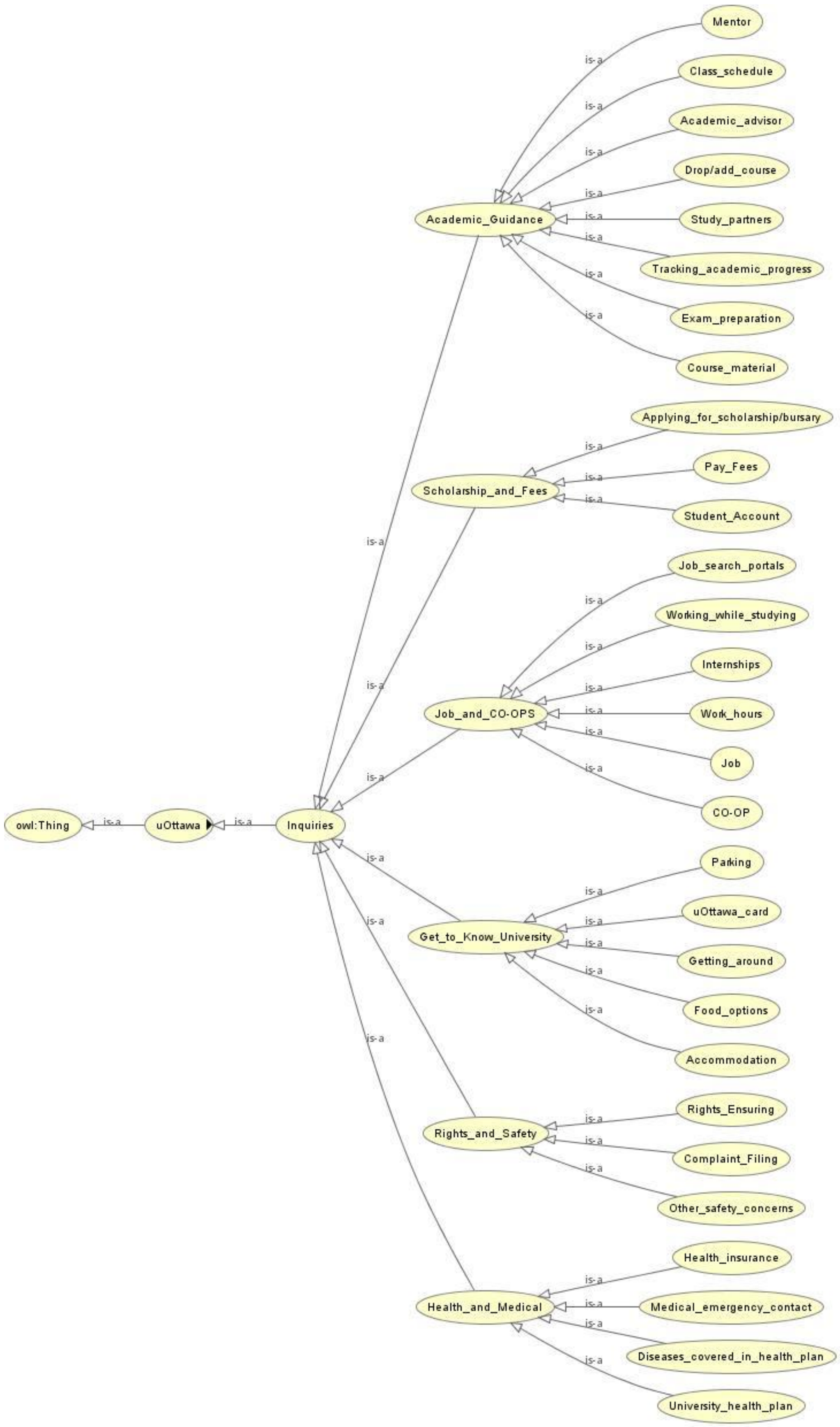
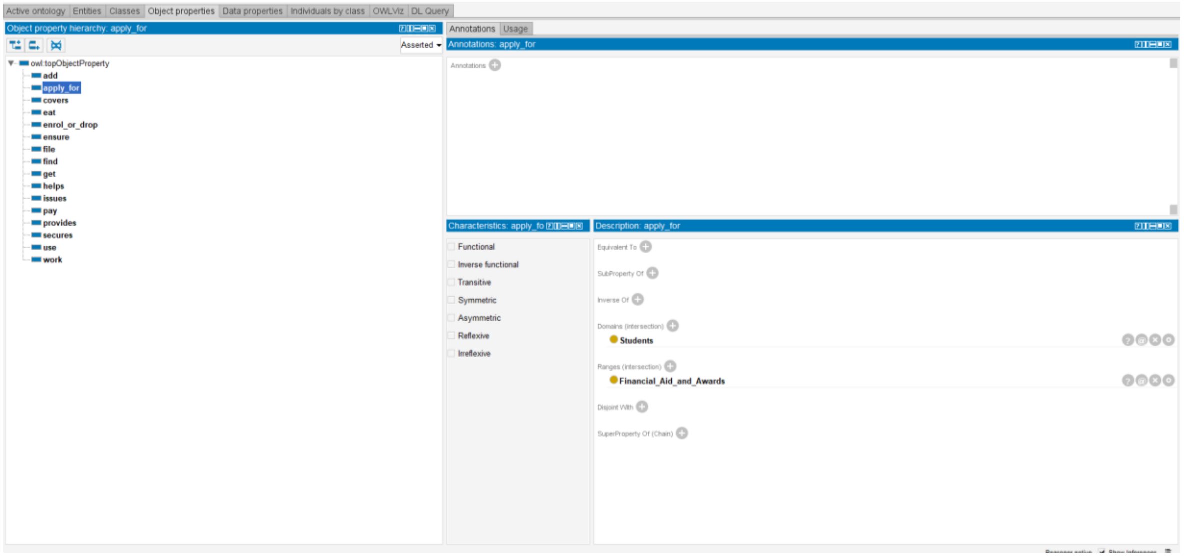


Figure 3:Classes hierarchy for inquiries, showing the main topics for competency questions and their possible branches.



Object property hierarchy: apply_for

- owl:topObjectProperty
  - add
  - apply_for
  - covers
  - eat
  - enrol_or_drop
  - ensure
  - file
  - find
  - get
  - helps
  - issues
  - pay
  - provides
  - secures
  - use
  - work

Annotations: apply_for

Annotations

Characteristics: apply_fo

□ Functional
□ Inverse functional
□ Transitive
□ Symmetric
□ Asymmetric
□ Reflexive
□ Irreflexive

Description: apply_for

Equivalent To

SubProperty Of

Inverse Of

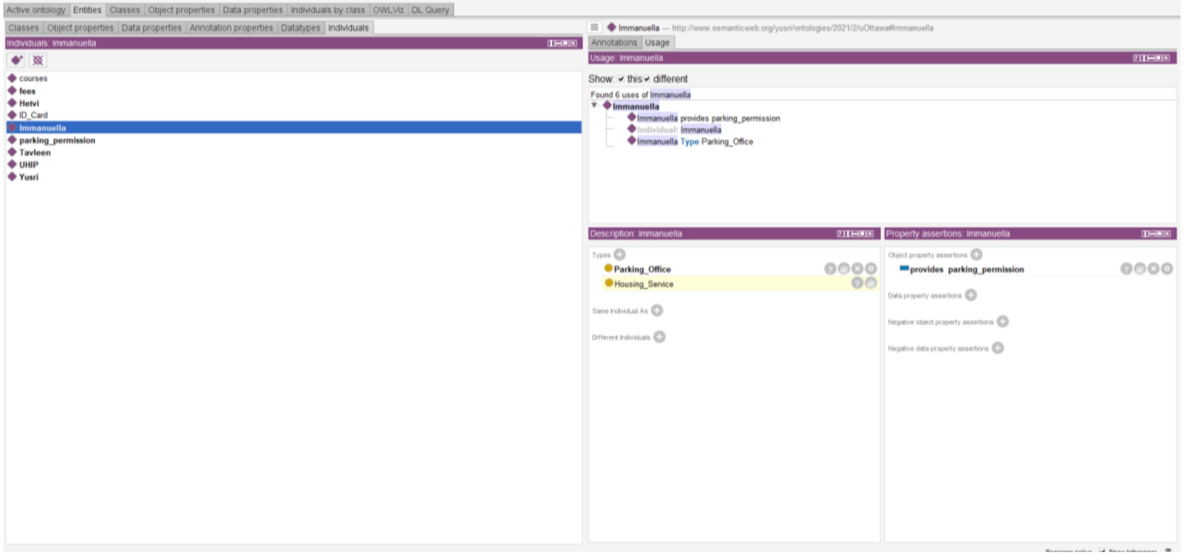
Domains (intersection)
Students

Ranges (intersection)
Financial_Aid_and_Awards

Disjoint With

SuperProperty Of (Chain)

Figure 4: Define the properties of classes-slots, showing the domain and range of a slot.



Individuals: Immanuella

- courses
- fees
- Hetvi
- ID_Card
- Immanuella
- parking_permission
- Tavleen
- UHIP
- Yusri

Immanuella — http://www.semanticweb.org/yusri/ontologies/2021/2/uOttawa#Immanuella

Usage: Immanuella

Show: ✓ this ✓ different

Found 6 uses of Immanuella

- Immanuella
  - Immanuella provides parking_permission
  - Individual: Immanuella
  - Immanuella Type Parking_Office

Description: Immanuella

Types
Parking_Office
Housing_Service

Same Individual As

Different Individuals

Property assertions: Immanuella

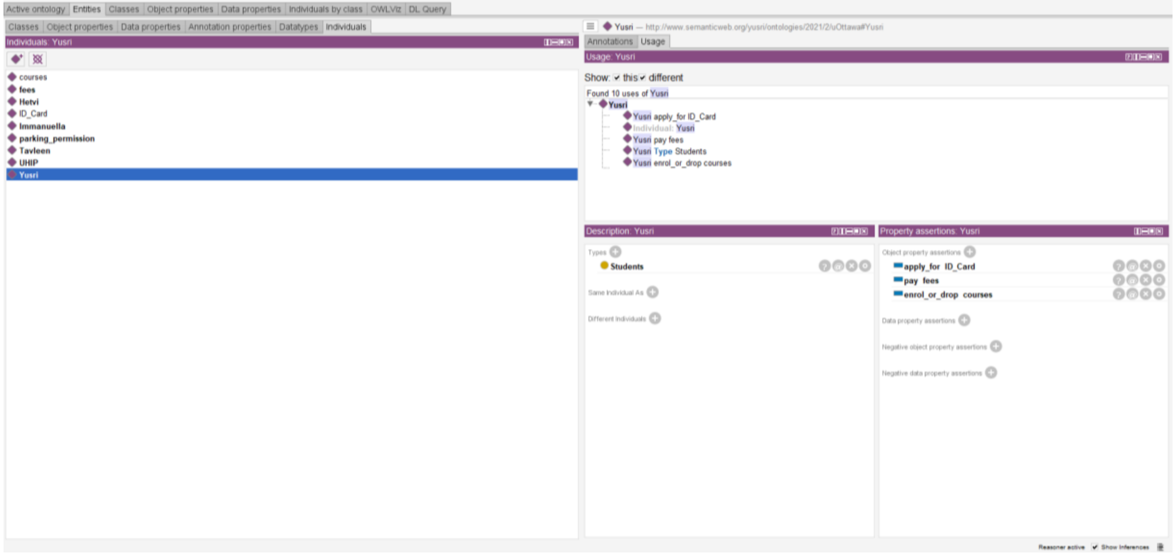
Object property assertions
provides parking_permission

Data property assertions

Negative object property assertions

Negative data property assertions

Figure 5: Create instances



Individuals: Yusri

- courses
- fees
- Hetvi
- ID_Card
- Immanuella
- parking_permission
- Tavleen
- UHIP
- Yusri

Yusri — http://www.semanticweb.org/yusri/ontologies/2021/2/uOttawa#Yusri

Usage: Yusri

Show: ✓ this ✓ different

Found 10 uses of Yusri

- Yusri
  - Yusri apply_for ID_Card
  - Individual: Yusri
  - Yusri pay fees
  - Yusri Type Students
  - Yusri enrol_or_drop courses

Description: Yusri

Types
Students

Same Individual As

Different Individuals

Property assertions: Yusri

Object property assertions
apply_for ID_Card
pay fees
enrol_or_drop courses

Data property assertions

Negative object property assertions

Negative data property assertions

Figure 6: Create instances

After building the ontology structure and its classes hierarchy, we tried to find priorities and relations for the classes and subclasses as shown below. So, we got a general knowledge about how to create sub questions for chatbot.

Knowledge structure

uOttawa

- **Students**
    - New student
    - Old student
- **Campus Services**
    - Housing Service
    - Parking Office
    - Registration Office
    - Restaurants and Coffee
    - Virtual Maps
    - Brightspace
    - Student Academic Success Service
    - Study Groups
    - uoZone Portal
    - Academic Advisers
    - Mentoring Centers
    - Online Library Service
    - Financial Aid and Awards
    - Career Development Centre
    - Co-operative Education Programs
    - International Office
    - Health and dental insurance
    - University Health Insurance Plan (UHIP)
    - Coronavirus (COVID-19)
    - Health, Safety and Risk Management
    - Protection Services
    - Human Rights Office
    - Secure Uoapp
- **Inquiries**
    - **Get to Know the University**
        - Getting around
        - Parking
        - Accommodation
        - Food options
        - uOttawa card

Verbs	Eat, drink, park, add, deposit, add, live, rent, get, purchase, provide, issue, order, apply.
Nouns	Restaurants, coffee, money, apartment, accommodation, campus, registration office, ID card, Parking office, Parking permit, Monthly/daily/hourly permit prices, card, online, Virtual Maps, Housing Service.
Relations examples	The student can use Virtual Maps to get around the university The Parking Office provides permission to students The students can use Virtual Maps to locate the parking sites. Registration Office issues the uOttawa ID for new students by online application.

	Housing Service at uOttawa provides accommodation to students. Students eat at Restaurants on-campus

- **Academic Guidance**
  - Mentor
  - Exam preparation
  - Class schedule
  - Course material
  - Study partners
  - Drop/add course
  - Tracking academic progress
  - Academic advisor

Verbs	Access, prepare, find, share, keep, track, enrol, drop, get.
Nouns	Material, courses, Brightspace, exams, study, partners, class, schedule, time, academic plan/matters, mentor, read, books, Student Academic Success Service, Study Groups, uoZone portal, academic advisers, Mentoring Centres, online library service.

- **Scholarship and Fees**
  - Applying for scholarship and bursary
  - How to pay fees
  - Track of student account

Verbs	Get, offer, apply for, pay, keep track
Nouns	Service, Study Groups, uoZone portal, academic advisers, Mentoring Centres, online library service, Financial aid and awards, Scholarships and bursaries, uoZone portal, fees, scholarships, bursary, student account, information.

- **Job and CO-OPS**
  - Working while studying
  - CO-OP
  - Job
  - Work hours
  - Internships
  - Job search portals

Verbs	Obtain, earn, apply
Nouns	Career Development Centre, job, co-op, internship, Work off/on campus as an international student, Co-operative Education Programs

- **Health and Medical**
  - Health insurance
  - Medical emergency contact
  - University health plan
  - Diseases covered in health plan

Verbs	work , offer, cover, opt out, include, get, contact
Nouns	Student Accounts, Health and dental insurance, International Office, University Health Insurance Plan (UHIP), Coronavirus (COVID-19), Health, Safety and Risk Management, Protection Services, health insurance, university, medical insurance, treatment, UHIP card, emergency

- Rights and Safety
    - Rights Ensuring
    - Other safety concerns
    - Complaint Filing

Verbs	Secure, ensure, manage, complaint, file
Nouns	Protection Services, Human Rights Office, rights, safety, university, secure Uoapp, playstore, complaint, harassment, discrimination

3. Competency Questions

We implemented six different competency questions which were further divided into multiple sub-questions based on knowledge structure obtained from ontology. All of the details related to these questions can be found below:

Question 1: **How can I get to know my university?**

- ➔ Where can I park my bike or car on campus?
- ➔ Where can I eat on campus?
- ➔ How can I avoid the lines to get my uOttawa Card?
- ➔ How can I add money to uOttawa card?
- ➔ How do I get around campus?
- ➔ How can I find accommodation on campus?

Question 2: **Where can I find solutions for academic issues?**

- ➔ How can I access the study material?
- ➔ How to prepare for the exams?
- ➔ How can I find study partners?
- ➔ How to access the class schedule and timings?
- ➔ With whom and how can I share my academic plans?
- ➔ How can I keep track of my academic matters? (grades)
- ➔ How can I enrol or drop my courses?
- ➔ How can I get a mentor?
- ➔ Where can I find material to read and books for the course?

Question 3: **How to get information related to jobs and job searches?**

- ➔ Can I earn money while studying?
- ➔ How to apply for jobs, internships?
- ➔ How to apply for co-op positions?

Question 4: **How to get information regarding scholarships and fees?**

- ➔ Where can I apply for scholarships?
- ➔ How to pay my fees and keep track of my student account?
- ➔ What is the scheme regarding bursary?

Question 5: **How do I secure my rights and safety at the university?**

- ➔ How can I ensure my rights are protected?
- ➔ What if I have other safety concerns?

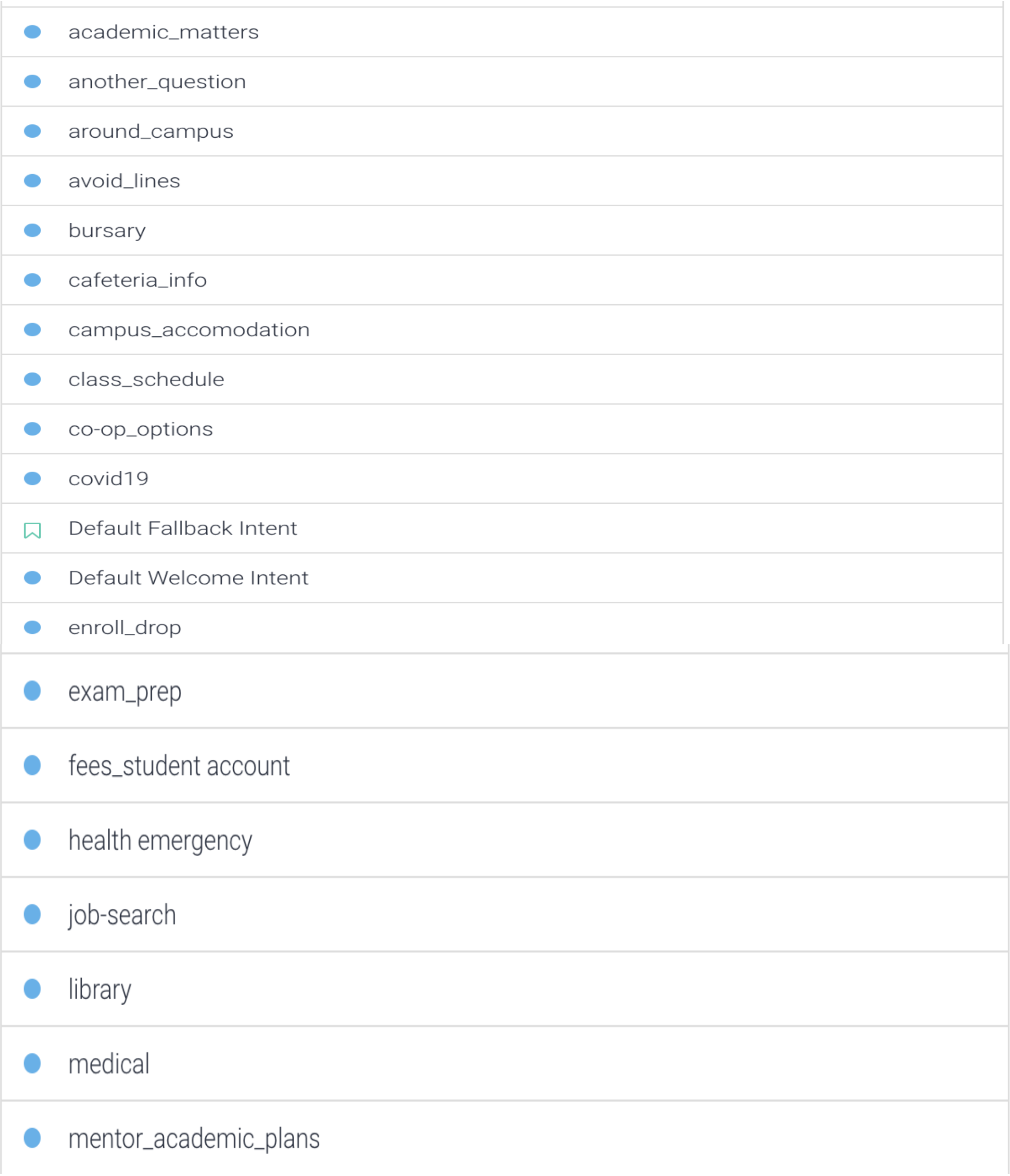
- ➔ How can I file a complaint against someone?

<u>Question 6</u>: How does the university's health insurance scheme work?

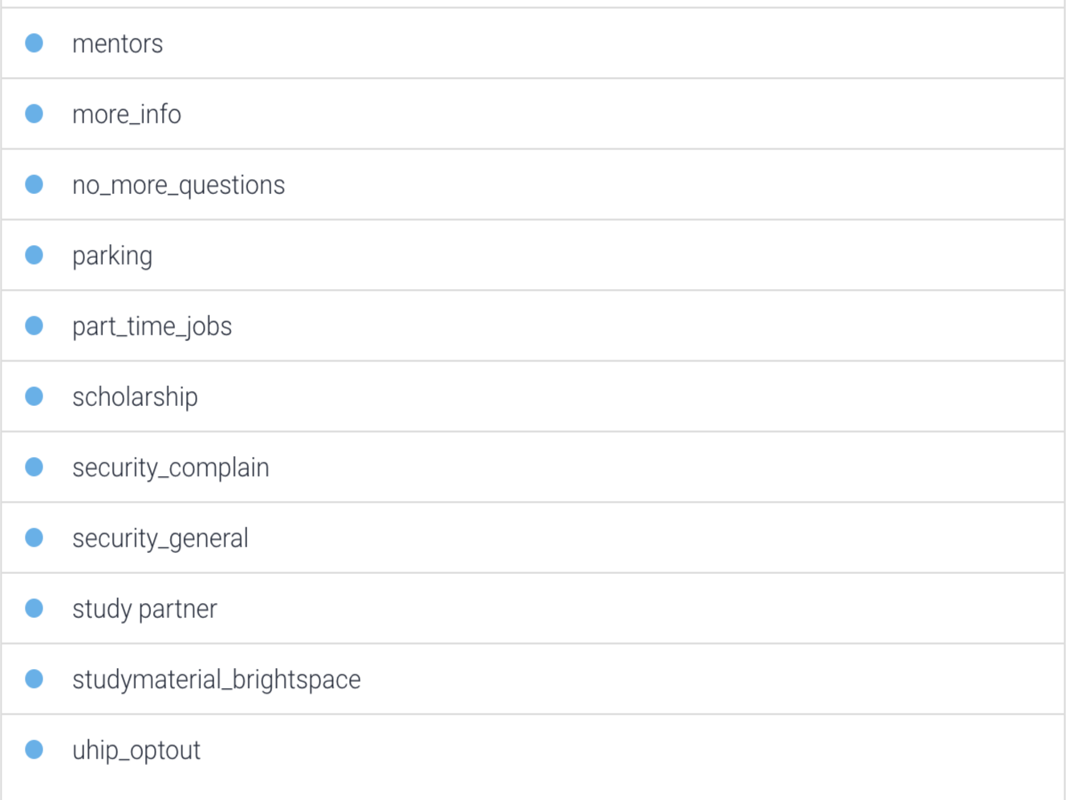
- ➔ What is the medical insurance offered by the university?
- ➔ Can we opt out of medical insurance?
- ➔ Is COVID-19 treatment/vaccination included in the insurance plan?
- ➔ From where can I get the UHIP card?
- ➔ Whom to contact in case of a health emergency?

4. Intent Development

We developed multiple intents as per the sub-questions for our competency questions as shown in figure 7. To develop an intent, login on the dialog flow tool using your google account. Create a new agent and for each question add a new intent. So, the number of intents is equal to the total number of questions in our case.



- academic_matters
- another_question
- around_campus
- avoid_lines
- bursary
- cafeteria_info
- campus_accomodation
- class_schedule
- co-op_options
- covid19
- Default Fallback Intent
- Default Welcome Intent
- enroll_drop
- exam_prep
- fees_student account
- health emergency
- job-search
- library
- medical
- mentor_academic_plans



- mentors
- more_info
- no_more_questions
- parking
- part_time_jobs
- scholarship
- security_complain
- security_general
- study partner
- studymaterial_brightspace
- uhip_optout

Figure 7: Intent Development

5. Test Cases

Next step is to add multiple test cases. These test cases involve multiple sets of training questions and their responses.

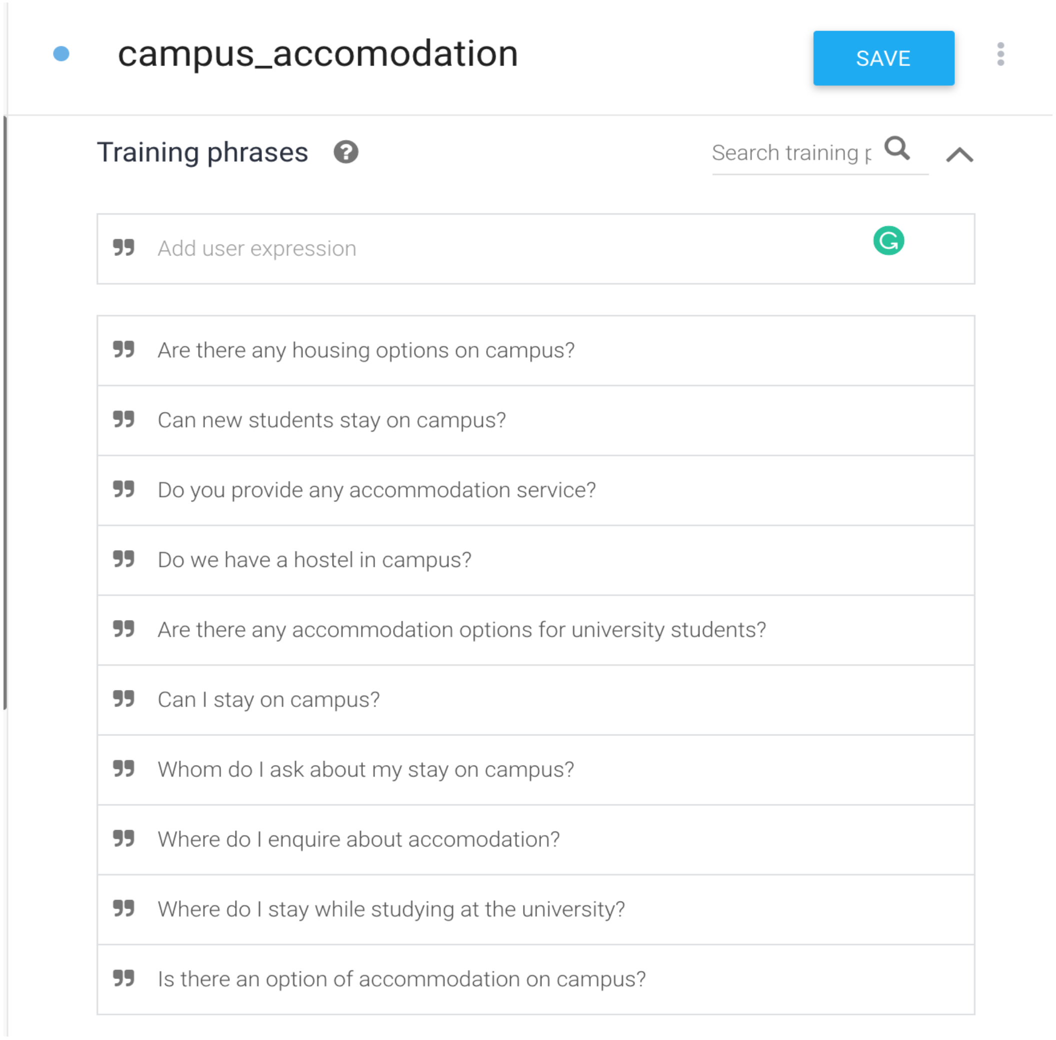
We added at least five different training questions for each of our intent. Our test cases required us to add hyperlinks, clickable-items, etc in the implementation so we used Custom Payload in most of our responses instead of a simple text field option.

We can also add multiple plain text statements in one response which means only one out of all would be picked during implementation, OR, we can add different responses which means each of these different sets of responses would be called during implementation.

Intent 1

Campus Accommodation Intent:

All the training phrases for campus_accomodation intent can be found in the below screenshot. These phrases train the chatbot to recognise when a question related to the intent topic is asked by the end user.

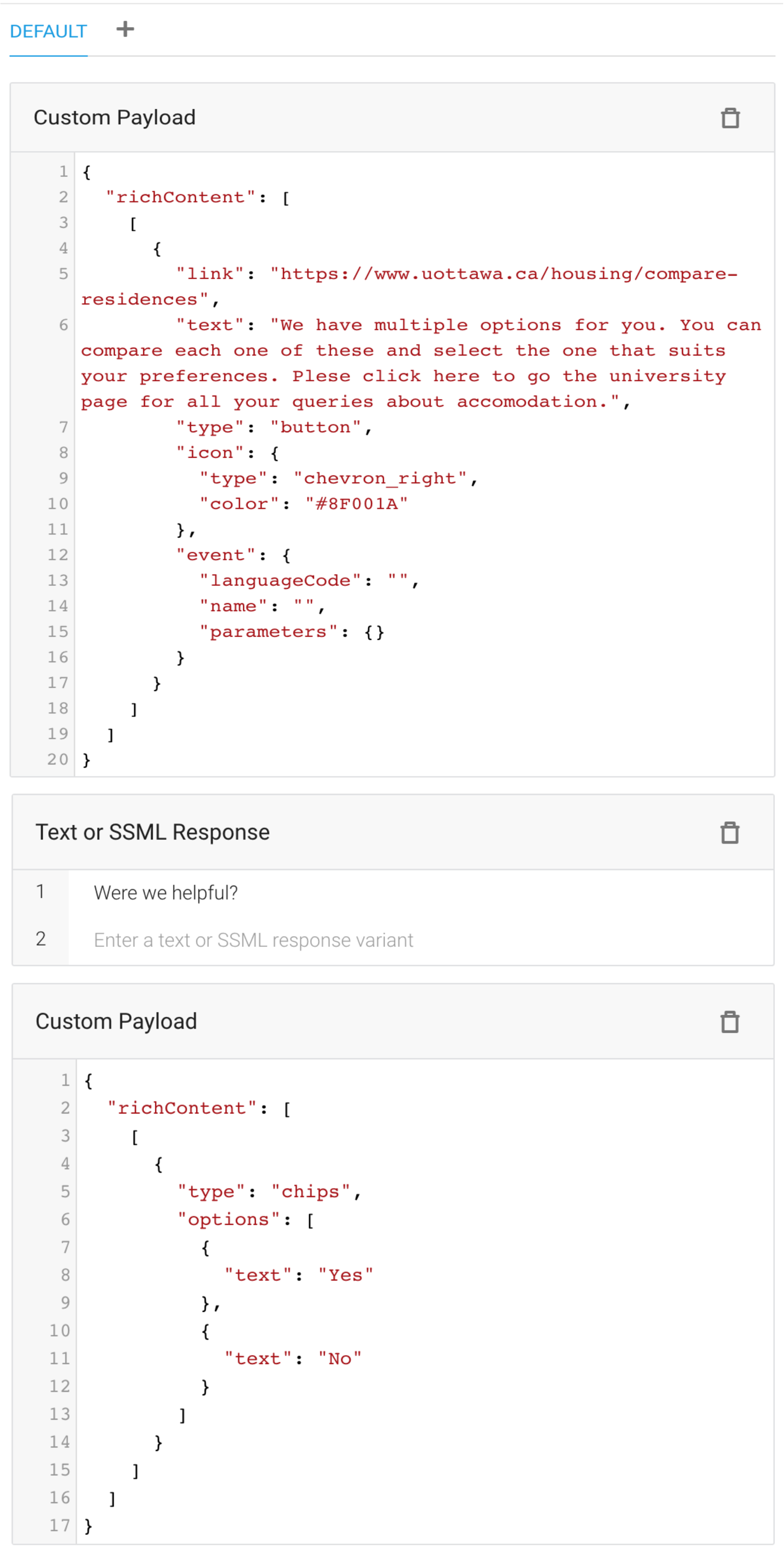


campus_accomodation

- Are there any housing options on campus?
- Can new students stay on campus?
- Do you provide any accommodation service?
- Do we have a hostel in campus?
- Are there any accommodation options for university students?
- Can I stay on campus?
- Whom do I ask about my stay on campus?
- Where do I enquire about accomodation?
- Where do I stay while studying at the university?
- Is there an option of accommodation on campus?

Figure 8: Campus Accommodation Intent.

The responses related to the training phrases can be passed either as Custom payload or Plain text. The figure 9 below shows screenshots to display them both.



```
{
  "richContent": [
    [
      {
        "link": "https://www.uottawa.ca/housing/compare-residences",
        "text": "We have multiple options for you. You can compare each one of these and select the one that suits your preferences. Plese click here to go the university page for all your queries about accomodation.",
        "type": "button",
        "icon": {
          "type": "chevron_right",
          "color": "#8F001A"
        },
        "event": {
          "languageCode": "",
          "name": "",
          "parameters": {}
        }
      }
    ]
  ]
}
```

1 Were we helpful?

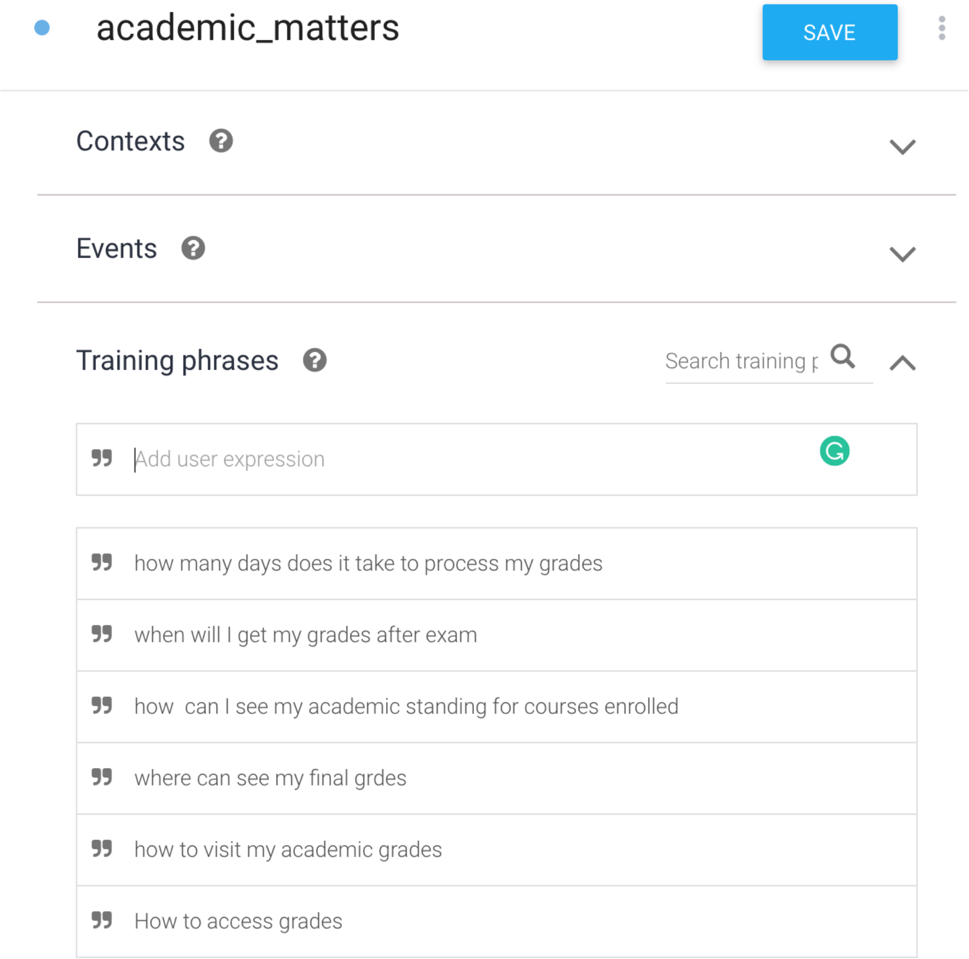
```
{
  "richContent": [
    [
      {
        "type": "chips",
        "options": [
          {
            "text": "Yes"
          },
          {
            "text": "No"
          }
        ]
      }
    ]
  ]
}
```

Figure 9: Responses related to the training phrases of intent 1.

<u>Intent 2</u>

Academic Matters Intent:

All the training phrases acadmic_matters intent can be found in the below screenshot, figure 10. These phrases train the chatbot to recognise when a question related to the intent topic is asked by the end user.



how many days does it take to process my grades

when will I get my grades after exam

how can I see my academic standing for courses enrolled

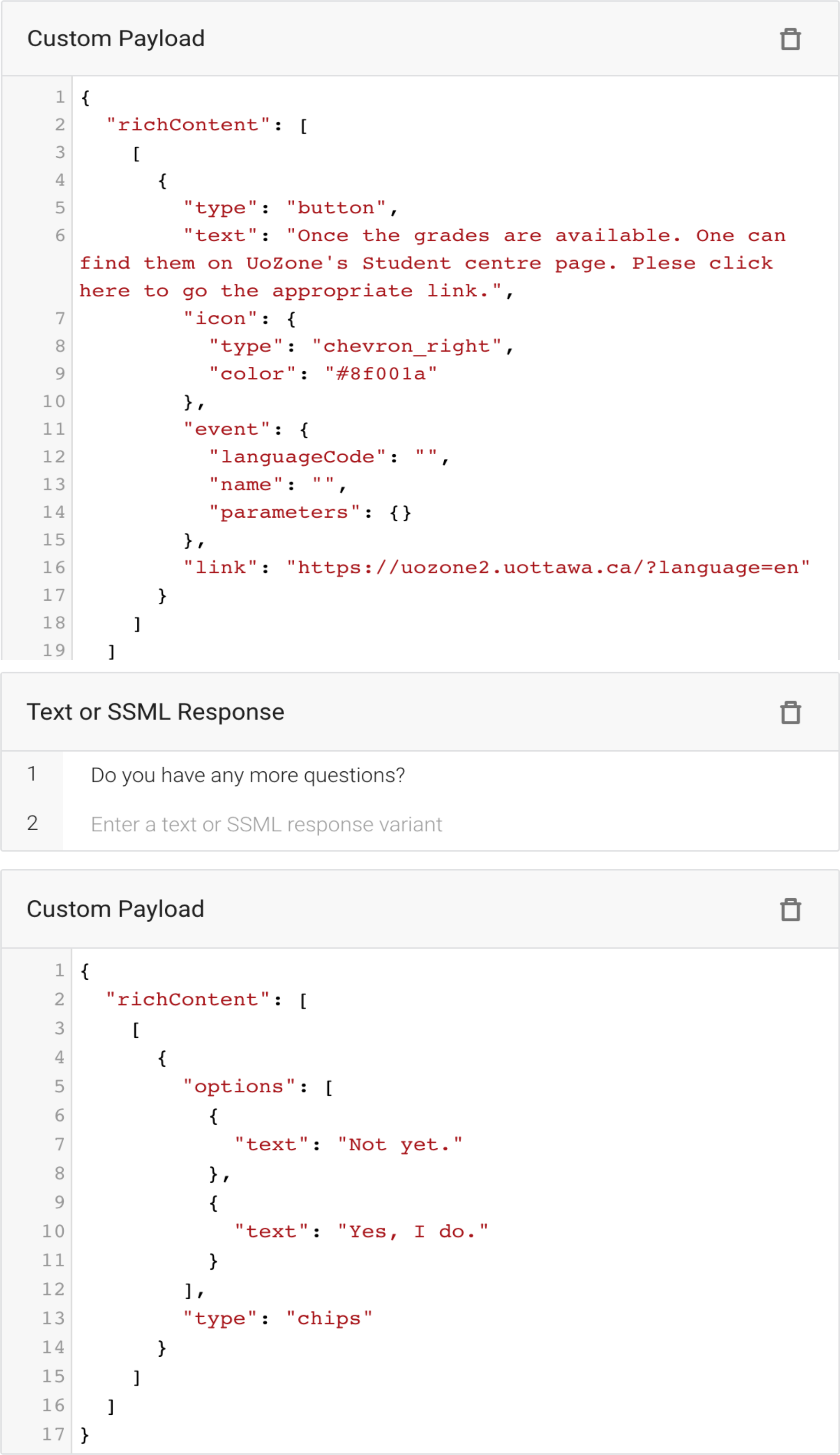
where can see my final grdes

how to visit my academic grades

How to access grades

Figure 10: Academic Matters Intent.

The responses related to the training phrases can be passed either as Custom payload or Plain text. The below screenshots display them both, figure 11.



```
{
  "richContent": [
    [
      {
        "type": "button",
        "text": "Once the grades are available. One can find them on UoZone's Student centre page. Plese click here to go the appropriate link.",
        "icon": {
          "type": "chevron_right",
          "color": "#8f001a"
        },
        "event": {
          "languageCode": "",
          "name": "",
          "parameters": {}
        },
        "link": "https://uozone2.uottawa.ca/?language=en"
      }
    ]
  ]
```

Do you have any more questions?

```
{
  "richContent": [
    [
      {
        "options": [
          {
            "text": "Not yet."
          },
          {
            "text": "Yes, I do."
          }
        ],
        "type": "chips"
      }
    ]
  ]
}
```

Figure 11: Responses related to the training phrases of intent 2.

6. Implementation and Testing

After adding all the test cases, we save the changes and go to the integrations tab from the left side panel. Here we experimented with two different types of integrations:

Web Demo:

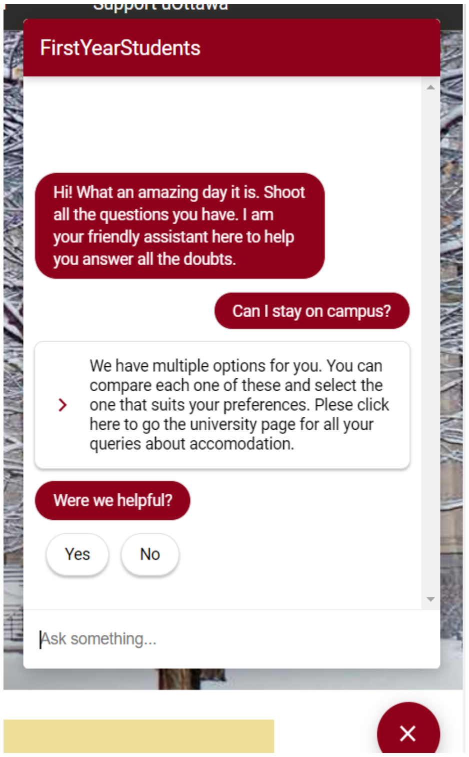
- ➔ Implementation: We can implement a web demo for basic implementation of our chatbot which requires only plain text in the response.
- ➔ Testing: To test the integration, we can either use the link provided when we click on this option or add the code snippet provided to our webpage.

Dialogflow Messenger:

- ➔ Implementation: We can implement a dialogflow messenger if we plan to implement so enhancement to the basic text. This can be done by adding rich content in the custom payload option for responses.
- ➔ Testing: To test the integration, we have to add the code snippet provided to our webpage. There is no option to copy the link for external usage.

Implementation 1
Campus Accomodation:

The below displays the screenshot of the chatbot which responses to the campus accommodation related questions successfully.
We notice that the chatbot responds back with a clickable link successfully and an input question as `Were we helpful?` to see if the user has any more queries.



Hi! What an amazing day it is. Shoot all the questions you have. I am your friendly assistant here to help you answer all the doubts.

Can I stay on campus?

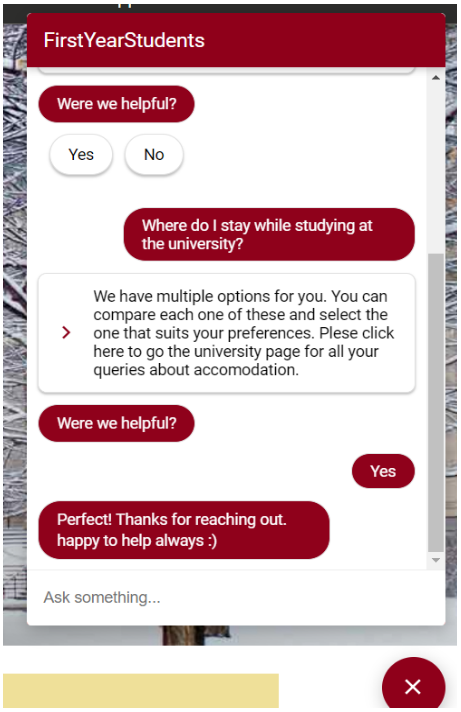
We have multiple options for you. You can compare each one of these and select the one that suits your preferences. Plese click here to go the university page for all your queries about accomodation.

Were we helpful?

Chat bot response to `Can I stay on campus?`

The chatbot then responds successfully as per the positive or negative inputs provided as Yes OR No. This can be configured either in small talk or separate intents.

Below is the example when the user responds as Yes.



Were we helpful?

Where do I stay while studying at the university?

We have multiple options for you. You can compare each one of these and select the one that suits your preferences. Plese click here to go the university page for all your queries about accomodation.

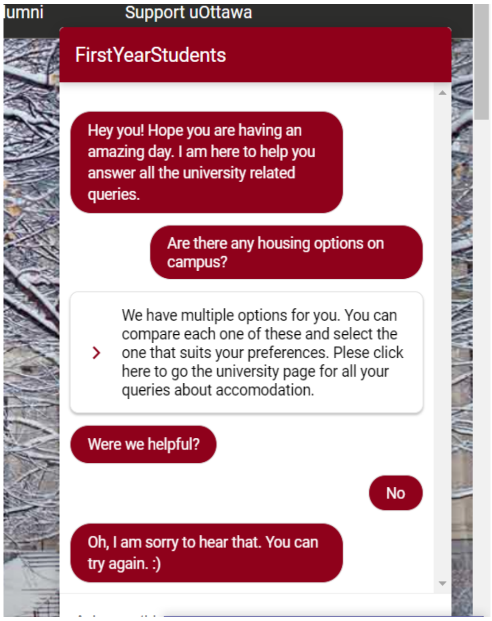
Were we helpful?

Yes

Perfect! Thanks for reaching out. happy to help always :)

Chatbot response to `Where do I can stay while studying at the university?`

Below is another example when a user responds as No if they are not satisfied by the information provided by the chatbot.



Hey you! Hope you are having an amazing day. I am here to help you answer all the university related queries.

Are there any housing options on campus?

We have multiple options for you. You can compare each one of these and select the one that suits your preferences. Plese click here to go the university page for all your queries about accomodation.

Were we helpful?

No

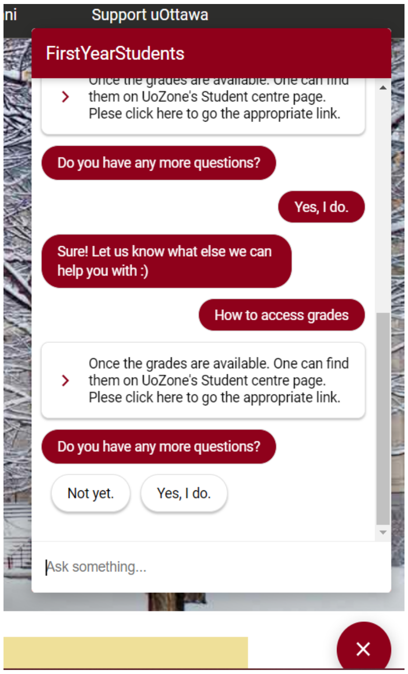
Oh, I am sorry to hear that. You can try again. :)

Chatbot response to `Are there any housing options on campus?`

Implementation 2

Academic Matters:

The below displays the screenshot of the chatbot which responds to the academic matters related questions successfully.
We notice that the chatbot responds back with a clickable link successfully and another input question as `Do you have any more questions?` to see if the user has any more queries.
The chatbot then responds successfully as per the positive or negative inputs provided as Yes OR No. This can be configured either in small talk or separate intents.



Once the grades are available. One can find them on UoZone's Student centre page. Plese click here to go the appropriate link.

Do you have any more questions?

Yes, I do.

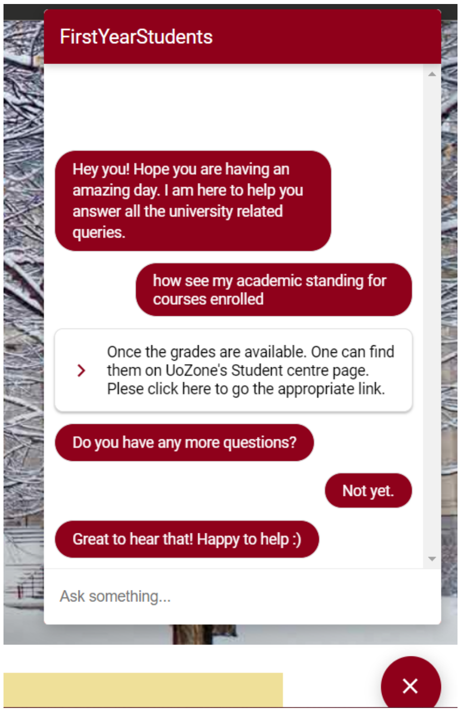
Sure! Let us know what else we can help you with :)

How to access grades

Once the grades are available. One can find them on UoZone's Student centre page. Plese click here to go the appropriate link.

Do you have any more questions?

Chatbot response to `How to access grades? `
Below is an example when a user responds as No if they do not have any more questions.



Hey you! Hope you are having an amazing day. I am here to help you answer all the university related queries.

how see my academic standing for courses enrolled

Once the grades are available. One can find them on UoZone's Student centre page. Plese click here to go the appropriate link.

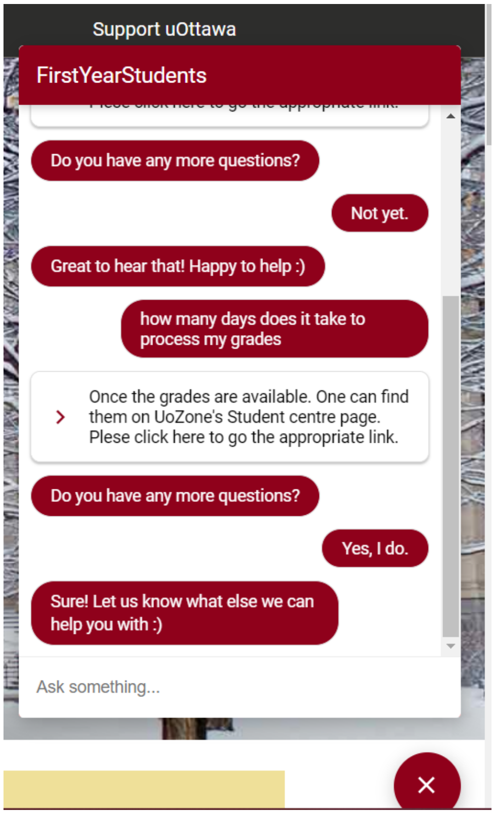
Do you have any more questions?

Not yet.

Great to hear that! Happy to help :)

Chatbot response to `How can I see my academic standing for courses enrolled?`

Below is another example when the user responds as Yes.



Plese click here to go the appropriate link.

Do you have any more questions?

Not yet.

Great to hear that! Happy to help :)

how many days does it take to process my grades

Once the grades are available. One can find them on UoZone's Student centre page. Plese click here to go the appropriate link.

Do you have any more questions?

Yes, I do.

Sure! Let us know what else we can help you with :)

Chatbot response to `How many days does it take to process my grades?`

7. Weaknesses

Limitations we encountered about Google Dialogflow:

- ➔ There is no way to add a hyperlink in the text. The workaround we implemented is to add a button and assign it to a link.
- ➔ We found a small bug that we are not allowed to change the letter case of the intent names, because the text field is not case-sensitive and we see an error saying that the name already exists. The workaround we implemented is to change the name to a new name and then rename it later to the intended name again with the correct letter case.
- ➔ To test the integration using dialogflow messenger, we need to add the code snippet provided to our webpage. There is no option to copy the link of the implemented chatbot for external usage.

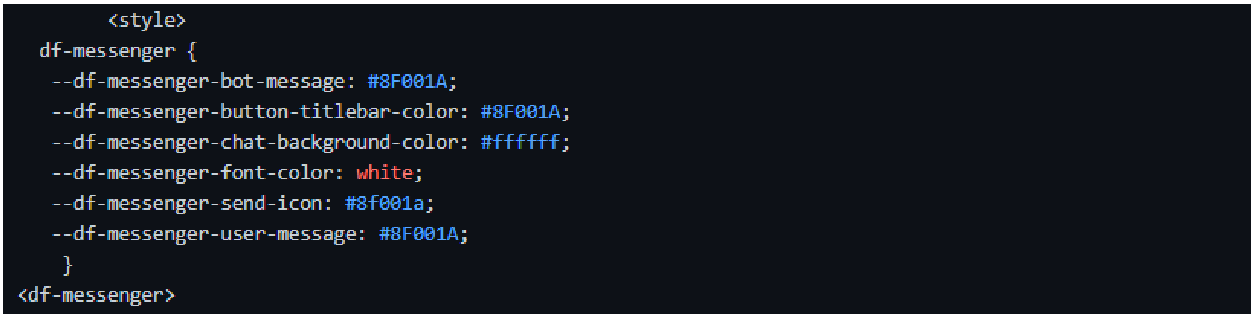
Weaknesses of the chatbot we implemented:

- ➔ Due to time limitation constraints we were unable to implement a case specific solution for the queries posted in the ChatBot.
- ➔ We didn't collect user information(store in database), which would help the user file a complaint and provide the user with a ticket and later respond to the user with the information they are looking for.

- ➔ We’re unable to add intents/questions regarding international student queries due to time constraint it requires multiple cases to be taken into consideration.

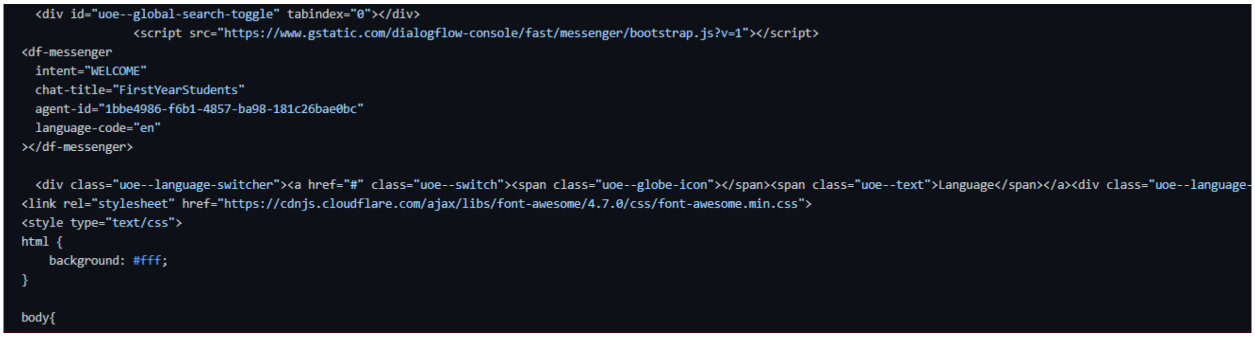
8. Integration of ChatBot using GitHub Web Pages

So as mentioned in the implementation part of the report we have used Dialogflow Messenger for our implementation of our ChatBot. So we decided to create a front end for the Chatbot where the testing of chatbot becomes easy and convenient.
Therefore we decided to copy the code snippet obtained from the dialog flow on a html webpage.



```
        <style>
  df-messenger {
   --df-messenger-bot-message: #8F001A;
   --df-messenger-button-titlebar-color: #8F001A;
   --df-messenger-chat-background-color: #ffffff;
   --df-messenger-font-color: white;
   --df-messenger-send-icon: #8f001a;
   --df-messenger-user-message: #8F001A;
    }
<df-messenger>
```

And What better webpage could it be than uOttawa Home Page. So, we created a clone of the uOttawa home page, using its html code. And after a few changes we integrated our chatbot code snippet with the home page code and created a new html document.

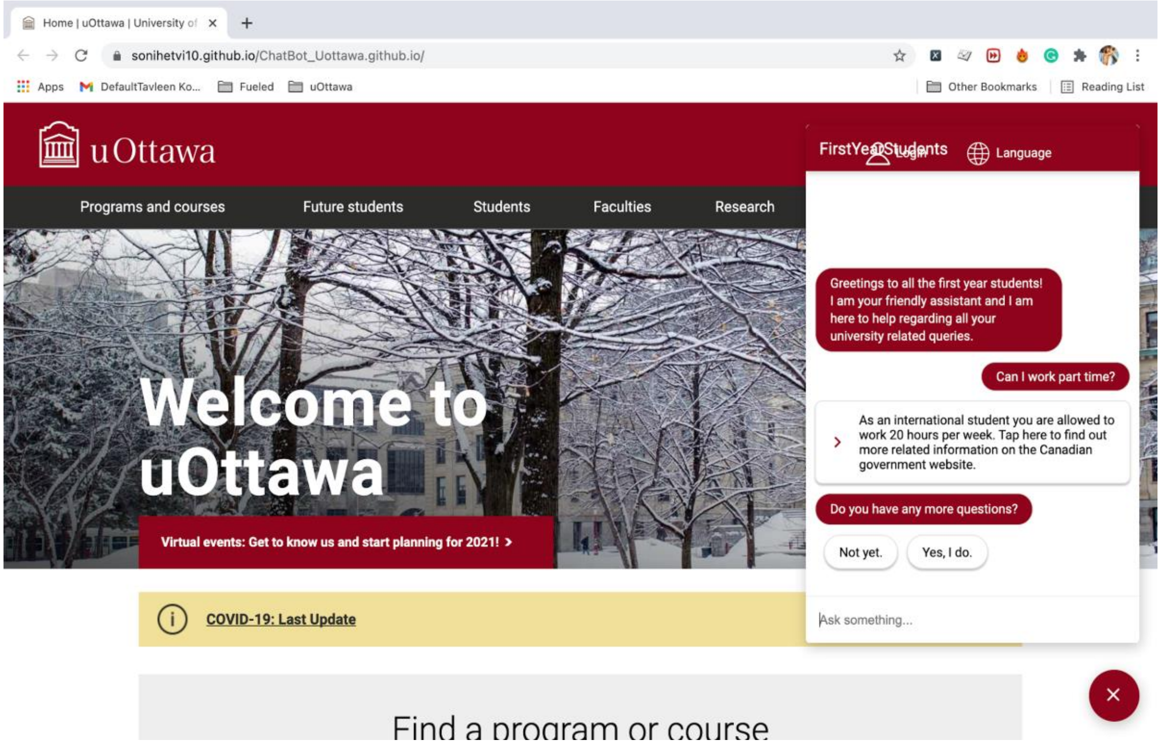


```
  <div id="uoe--global-search-toggle" tabindex="0"></div>
              <script src="https://www.gstatic.com/dialogflow-console/fast/messenger/bootstrap.js?v=1"></script>
<df-messenger
  intent="WELCOME"
  chat-title="FirstYearStudents"
  agent-id="1bbe4986-f6b1-4857-ba98-181c26bae0bc"
  language-code="en"
></df-messenger>

  <div class="uoe--language-switcher"><a href="#" class="uoe--switch"><span class="uoe--globe-icon"></span><span class="uoe--text">Language</span></a><div class="uoe--language-
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<style type="text/css">
html {
    background: #fff;
}

body{
```

Next step was to publish the new webpage created so for that we made use of github web pages. The final output can be seen in the screenshot below where our chatbot is successfully integrated with the webpage.



Greetings to all the first year students! I am your friendly assistant and I am here to help regarding all your university related queries.

Can I work part time?

As an international student you are allowed to work 20 hours per week. Tap here to find out more related information on the Canadian government website.

Do you have any more questions?

Not yet. Yes, I do.

9. Conclusion

In conclusion, the chatbot aims to solve issues first year students will have while trying to get accustomed to the university. We implemented this chatbot by providing uOttawa website as an external web page hence the different user interface. Costume payloader have been used to include clickable links and rich texts. Naturally dialogflow chatbot although allows for integration with popular channels, does not have a flow in a conversation hence there is limitation in error avoidance.