|  |
| --- |
|  |
| Credit Bot |
| Conversational AI and Virtual Assistants |

|  |
| --- |
|  |

|  |  |
| --- | --- |
| Author: | Dominic Schori  Emanuel Graf  Kevin Bollier |
| Lecturer: | Martin, Andreas  Pande, Charuta |
| Place, Date: | Basel, 15.12.2019 |

Table of Contents

[Introduction 2](#_Toc27330319)

[Basic concept: 2](#_Toc27330320)

[Use cases 2](#_Toc27330321)

[User Persona 1 2](#_Toc27330322)

[User Persona 2 2](#_Toc27330323)

[Approach 2](#_Toc27330324)

[V1.0 2](#_Toc27330325)

[V2.0 2](#_Toc27330326)

[Final features 3](#_Toc27330327)

# Introduction

For the course conversational AI, we created a basic chatbot that can recognize inputs and answer to user questions.

## Basic concept:

The basic concept of our chatbot is very simple. A student can ask the bot about possible modules to reach the required credits to finish his bachelor. To get this information, the user gives the bot information about the semester he is in, the amount of credits he needs and if he wants elective modules or core modules. Further he can specify if he wants modules that only have a pass/fail grading or a 1 to 6 grading scale.

## Use cases

The start of the project was made with two very simple user personas, which we used to determine which questions should be asked to specify what information the bot needs to give an answer.

### User Persona 1

Andrew is a BIT student in his fifth semester. He wants to use the AI to find out which options he has to get his last 5 credits to finish his degree. He needs these credits in the business administration section.

### User Persona 2

Sean is a BIT student in his fourth semester and needs 5 free credits until the end of his studies. He wants to see what his options are.

### Key Use Cases

From the two user personas we can derive several key use cases.

* The chatbot should give an answer.
* The user should be able to state the amount of credits he needs.
* The user should be able to state the semester he is in.
* The user should be able to specify if he is interested in elective modules or core modules.
* The user should be able to specify if he wants a specific grading system.
  + If yes, the user should be able to specify the grading system.
* The user should be able to specify a specific module group.

## Approach

We created a simple walkthrough with Botsociety and exported it then to Dialogflow where we trained the bot to recognize more sentences and implemented a basic fallback as well as an interface with integromat, which was needed so the bot could answer with an e-mail.

This base in botsociety was reworked a couple times, with two main versions.

### V1.0

The first version of the chatbot contained three possible paths after the user was asked if he would like proposed modules from a certain type. The three answers possible were elective, main modules or both. Together with various yes or no questions, this led to problems, as the path for the bot got confusing, and some errors occurred that we could not fix. We also worked with quick inputs, which lead to further confusion when we exported the bot to Dialogflow.

### V2.0

Version two is a bit simplified in the idea that it does not hold any yes/no questions to keep the path as clear as possible. Also, the option to select main modules AND electives was deleted, as well as the quick answers the user could click. This version works through the whole process without trouble and sends an e-mail after completing the process.

## Final features

The final version of the bot lets the user work through a set of questions in which the bot registers the semester the user is in, the amount of credits needed, if the modules should be graded in a certain way, if it should be elective modules or core modules, and if the modules need to be part of a certain group, such as foundation or information technology.

After the user works through the whole process he receives an e-mail. For the example this mail is set to a fix address, but it could be implemented that the user enters his address after a request from the bot.

The bot also has now a fallback. If the user enters an unknown term, the bot sends a basic reply, stating that he does not recognize the user statement.