

## AI Fo 2023 Graded Miniproject

This document specifies the miniproject. It will be graded and it counts 40% of your AI Fo grade (the other 60% come from the written exam).

### 1. Important Dates:

**Start:** Thursday, October 5<sup>th</sup> 2023

**Deadline:** **Sunday, November 5<sup>th</sup> 2023 (upload your report before 23:59)**

### 2. Teamwork!

- Work in teams of two people. Larger teams are not allowed. Working alone is possible if you send me an email indicating your reason!
- Collaborate effectively! You learn more and you are faster when you discuss your ideas and problems.
- Both team members will get the same grade. We reserve the right to interview (randomly selected) team members in order to make sure no one gets a free ride! Note that fraud is not accepted at OST and may have severe consequences.

### 3. Specification

The goal of this miniproject is to implement a service (of your choice) that integrates DialogFlow. Your client provides (at least) a command-line interface and calls the DialogFlow API.

Grading will be based on your report (PDF- File to be uploaded on Moodle), the source-code is not graded.

#### 3.1. Minimal requirements

To obtain a **grade 4.0**, the following **minimal requirements** need to be implemented:

- **DialogFlow Agent:**  
Specify chat-bot in DialogFlow. You are free to start from a predefined agent but you have to define at least one additional intent and entity.
- **Client:**  
Implement (at least) a simple command line interface. The client reads text input, sends it to DialogFlow (using the DialogFlow API) and writes the agent's response to the command line.
- **Technology stack:**
  - Language of your choice (but we provide a code example only for Python)
  - Any type of client (CLI, Web-Client, GUI). But note that you are not allowed to pick one of the "no-code" integrations provided by DialogFlow (WhatsApp etc)
  - You can use any library/API/Service you want.
- **Report (PDF File):**  
Describe the architecture, the concepts you used, the features you implemented, and whatever you consider interesting. Make a few screenshots that clarify the usage of your client. Document both, the DialogFlow backend and the client.
  - write in German or English
  - The length of the report should be between 4 and 6 pages (not including a title page and an optional appendix).
  - Have a title page with all relevant information (team members, title, ...)
  - Add screenshots of your client application to show how it's performing.
  - Use an appropriate diagram to visualize the architecture of your application.

- Do not copy large code-fragments into the main report. But show at least one (interesting/relevant) code fragment and explain what it does.
- Structure your report. Write at least four sections:
  - **Introduction** where you give some context and explain the goal of the project.
  - Documentation of the **Client**: What your kind of service your client is providing. Screenshots of the working client
  - Some details about the **DialogFlow-backend**. E.g. which intents etc. did you specify.
  - **Discussion**: what you learned, what are the limitations of the chat-bot, etc.
- If you have additional material which doesn't fit into the main report, but you don't want to delete, then add an appendix (a chapter at the end of the report). Typically, in the main text you would write something like "for technical details see appendix xy". Note that the appendix is optional. Assume that we do **not** read it. Grading is based on the main text.
- Focus on relevant content. Be concise. Remove clutter:  
<https://www.acrolinx.com/blog/how-to-tidy-up-your-writing/>

### 3.2. How to get extra points

We do not specify a detailed, exhaustive list of criteria to get extra points. Anything that goes beyond the minimal requirements is appreciated. Here are some ideas:

You can add more intents and make the dialog more complex. Or you can browse the DialogFlow documentation and use some of the countless options. You could implement a more advanced process using fulfilments and make use of the GCP services (cloud functions, persistence, ...). In general, be creative! Implement an interesting chat-bot! Also, make sure your report explains how you structured your software project. Use appropriate diagrams to document the architecture. For a project that shows some original implementations (something that goes beyond the examples given in the exercise session), and which is reasonably well documented, you can expect a grade around 4.5 to 5.5.

The grade 6 will be given for exceptional projects which exhibit some additional technical creativity.

### 4. How to submit your report

There will be an assignment called "Graded Project Report Submission" on Moodle (in the week 7 section). If you work in a team, **both team members will have to upload the (same) PDF File**. Do not upload the source code.

### 5. Manage your resources!

It is easy to get lost in the sea of possibilities and spend way too much time on this project. This project is worth 40% of your final AI Fo-grade. Manage your resources accordingly.

### 6. Respect the deadline

If you do not hand-in a report by the deadline, we have to give you a grade 1.0. If you are unable to meet the deadline for important reason (illness, military service, etc.) the deadline can be discussed. Contact us as early as possible!