

# MouthPiece

## COS301 Team Gamma: Converter Team Goals

March 2020

### **1 Introduction**

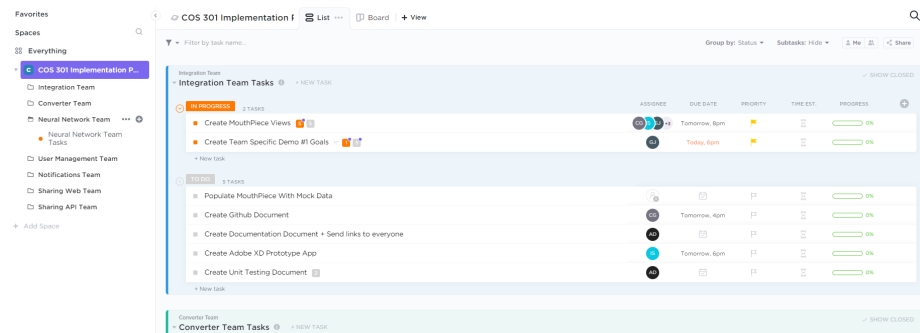
This document describes the responsibilities and all deliverables for the Converter Team.

Demo #1 Due Date: Thursday 12 March 20:00

## 2 Organisation

### 2.1 ClickUP

You need to elect a team leader if you have not done so already. Work together with this member to create tasks and sub-tasks for each member in your team on <http://clickup.com> (on the COS301 workspace you have been invited to).



Ensure that you indicate progress and mark tasks as complete. This page will be displayed at the demo.

### 2.2 Slack

Although we have created a WhatsApp group - we do still find it easier for in-group discussions to happen on the slack channels. Simply login to <http://cos-301.slack.com>

### 3 Overall Team Tasks

These are the tasks to be done by your team for the final product (not specifically Friday's demo).

- Code to be done in Flutter (Running on Android device)
- Retrieve real time audio from Android (microphone permissions and audio stream)
- Read selected algorithm preference (Formant, Volume) from Android
- Record training data (sound file, .mp3 or similar)
- Store training data on Android
- Send training data to Neural Network server
- Read selected Mouth Pack (image files) stored on Android device
- Implement data structure to switch between image files (array etc.)
- Implement the Volume Based algorithm (Decibels)
- Work with the Neural Network Team to run the Neural Network model on the Android device
- Send back the real time mouth image to display (this gets done at a high frequency, e.g. 10 images per second)

*The tasks are always subject to change, but we have tried our utmost best to sketch out the entire project ahead.*

## 4 Team Tasks for Friday

### 4.1 Documentation

All documentation to be created in Overleaf

#### 4.1.1 Class Diagrams

You are required to create formal class diagrams that explain the functions and where possible variables that are going to be used. Your class interface needs to get as close to the final product as possible. Try to use design patterns and good programming principals such as Object-oriented programming where possible. The class diagrams should encompass all functionality described in **Overall Team Tasks**.

#### 4.1.2 Frameworks

You are required to do some research on frameworks that you could implement to assist your **Overall Team Tasks**. Write a short report on only the frameworks that you are keen to implement. Show its benefits and how you can implement it with the existing technologies.

**Some examples**

- [https://pub.dev/packages/wave\\_generator](https://pub.dev/packages/wave_generator)
- [https://github.com/dooboolab/flutter\\_sound](https://github.com/dooboolab/flutter_sound)
- [https://pub.dev/packages/flutter\\_audio\\_recorder](https://pub.dev/packages/flutter_audio_recorder)

In the event that the framework is something web based like the Google Firebase API, do the documentation as above, but also setup an account and be ready to demo the dashboard and how you will interface with it.

### 4.2 Development Environment

You are required to setup the development environment (have it running on at least one member's device, ready to demo). Setup everything you think you require.

**At bear minimum**

- Android Studio
- Visual Studio Code
- Flutter

*This is required for your implementation in any case.*

## 4.3 Implementation

### 4.3.1 Voice Recorder Demo

You are required to build a small Flutter app (independent of everything else) that will:

- Display a mouth image (single image, horizontal as on Cadon's designs)
- Have a button to start recording (you can have this in a separate "config" view of the app as well, since it is demoing the functionality to record training data)
- Ask for microphone permission (see the frameworks linked above)
- Save the recording from the microphone to the device's internal storage
- You can play it back through the file manager on your device

### 4.3.2 Code Skeleton

You are required to create the skeleton functions for everything described in your class diagram. No need to implement the functionality - and don't worry too much if this changes in the future.

This code will be pushed to the Github to demo our branching setup as well.

### 4.3.3 Unit Testing

Describe how your team will implement unit testing. Better yet - implement it in the **Voice Recorder Demo** where possible.