THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

CREATE CHANGE

School of Electrical Engineering and Computer Science

DECO3801/DECO7381 Individual Assessment Declaration Cover Sheet

Course Code:

Coursework Item:

Name: Wei–Ting Hong **Student Number:** 47523483

Please confirm each statement by ticking the boxes before signing the bottom.

I declare that:

1. The information reported in this work, except where otherwise indicated, is my own original work. □
2. This work has not been submitted for any degree or examination at any other university and/or for any other course at UQ. □
3. This work does not contain other persons' and/or generative Artificial Intelligence (AI) data, pictures, graphs, or other information, unless specifically acknowledged as being sourced from other persons and/or generative AI. □
4. This work does not contain other persons' and/or generative AI writing, unless specifically acknowledged as being sourced from other sources. Where other written sources have been quoted, then:
    a. Their words have been re-written, but the general information attributed to them has been referenced.
    b. Where their exact words have been used, their writing has been placed inside quotation marks, and referenced. □
    c. Where generative AI has been used to aid writing, all prompts used are fully disclosed in the appendix.
5. This work does not contain text, graphics or tables copied and pasted from the Internet and/or generative AI, unless specifically acknowledged, and the source being detailed in the bibliography and appendix sections. □
6. I have checked the work to ensure that there are no instances of academic misconduct, including plagiarism herein. □

The University defines Academic Misconduct as involving 'a range of unethical behaviours that are designed to give a student an unfair and unearned advantage over their peers'. The University takes Academic Misconduct very seriously and any suspected cases will be investigated through the University's standard policy.

I hereby declare that I understand the meaning of academic misconduct and recognise that submission of work for assessment that includes instances of academic misconduct is an offence that may result in disciplinary action being taken against me.

Wei–Ting Hong
Signature

2024/4/10
Date

Hearing Loop is one of the many tools in Australia that benefit people with disabilities, which utilising scientific principles such as electromagnetic to provide a friendly living experience for deaf people that wearing hearing aids. Hearing loops can be found at public facilities such as bus stops, schools, and other similar locations. It will use electromagnetic to broadcast to the deaf people, allowing these broadcasts to be played directly in hearing aids, with also isolating external noise interference. Our project will develop an app that will convert the audio that is delivered to the deaf people into text on their smart phone, with the addition of a vibration function that provide them to have a fuller inform experience. In addition, we will use this function as a base to extend app that allow users to directly monitor environments sounds as input audio stream, users will be able to receive ambient audio and convert it into text messages. Likewise, it also supports vibration and display on the phone so that they can view it over and over again.
In this project, I take responsible for the back-end, which includes the notification to the app users, and the creation and management of the app's database. In addition, I will also be responsible for managing the github for the group project, and providing assistance to the team members when they use github.

Here's the workflow in our project app: First, the app gets audio from hearing loop or public announcement, transmit them into app and converts them to text via speech to text api. After conversion to text, a notification with appropriate layout and vibration is sent to inform the user that the converted message can be viewed. At the same time, the text is stored in the app's database, allowing the user to view these messages in future, or even manage them (tagging, deleting, sorting them in a particular way, etc.). As described, sending notifications and managing the database plays an important role in the lifetime of the app. It can be said that it is the only way the app to interacts with the user besides front-end UI.
As for the github management, I took the responsibility to managing the github since the team members told me that they were not familiar with github. I set up a git repo, and in a text file under the project, I briefly taught the team how to use github from the beginning, from installing git to properly pushing/pulling their local projects into the git repo. During the development period, if members have questions about using github (most of them are related to branch issues), I will also provide assistance.

The teamwork among our team is generally quite harmonious. We use Discord as our instant messaging software, if we have any issues during a development project, we can post them directly on Discord and discuss without any barrier. In addition, we maintain a good frequency of meetings, even during the period of individual development, it is still considered beneficial to maintain a good frequency of meeting, not only we can report our own progress to other members, but also know the current members of their development process. We have written a timeline in the statement of work, basically members will follow the timeline to develop the software, to avoid situation that members don't know what to do at any time. Based on the ECP interim assessments, we were able to see if the development speed met the expectations from the teaching team. We submitted our half-done projects on time and get good feedback generally, which at least proves that we don't have too many delays in our development process. This makes me think this is a good and effective process for group software development.

The following are the parts of the project that each member of the team is responsible for: Ryan, Tom, and Jose are responsible for front-end development. As far as I know, the front-end development is developed monolithically, there is no clear statement of who is responsible for which component. Ryan and Tom are very good friends, almost inseparable, I believe they usually stay together to develop the front-end of the project, from prototype design to actual coding, while Jose has Ryan and Tom communicate with him every Friday during class time to confirm each other's development progress. On the other hand, the back-end which Kenny, Alhan, and I worked in is separately develop and eventually merge together. As I mentioned earlier, I was responsible for the notification functionality of the app and the overall database management of the app, Kenny was responsible for everything related to speech to text, and Alhan was responsible for the audio streaming within the app (i.e., the audio stream was received and then sent to speech to text, which was Kenny's portion of the project).
Ryan, Tom, and Kenny are the core of the group, not only because they know each other before the class, but more importantly, with two of them in the front-end and one in the back-end, they keep the development and the vibe of the group in a harmonious pace. They can communicate with the teaching team on the one hand, and facilitate the communication between me, Alhan and Jose on the other. This made the development process very comfortable for all the team members, and also made the relationship between us as if we are doing a challenge together with our friends, which greatly increased the team's awareness and chemistry within the team.

In the project, we expect the app to be able to receive the audio stream from hearing loop and convert it into text. For this purpose, we rented a receiver from university library, which can convert the audio from the microphone in the classroom into audio through hearing loop and output it. The app will receive this audio input and convert it into text. However, this was the biggest bottleneck we encountered in entire development process. For some reason, the receiver could not receive the microphone's announcements correctly through hearing loop. The reason may be the connection between microphone and heraring loop, may be the heraring loop itself, or may be the receiver itself. We spent a lot of time trying to solve this problem, includes consulting the teaching team, contacting the school library, looking up information on the Internet, etc. However, the problem was still not solved and we started to consider whether or not to abandon this feature. After discussion, we've decided to abandon the hearing loop. As compensation, we're going to develop the remaining functionality, which is to listen directly to the environment to get audio. The first thing that came to our mind to plenty the functionality was CRUD principle. In addition to the basic CR (for receiving audio streams, converting them to text and storing them in the database for display in the "history section" of the design), we can also add the UD principle. We will provide a "add to favorites" feature, which allows the user to add a specific piece of text to favorites for future viewing. Originally, the title and "favorite" attributes of all data are set to null by default, by updating the title, favorite of that data will allow the user to give a special title and whether or not it is added to the favorites. Principle D: We will provide the user with the ability to delete one or multiple records. Although basically users would only use delete function to remove unused records, providing this feature to users will still improve their user experience.

In addition, Kenny has mentioned a problem, there is an issue when if the ambient sound is too noisy so that the api can't listen to the broadcast correctly (e.g., the broadcast on the bus says, "Please make sure you have your luggage when you get off the bus, thank you," but according to the ambient sound is too noisy, the api can only listen to the -> Please XXXXXXXX when you get off the bus, thank you), then how to deal with that? Our potential solution is to create a new database for storing common "keywords" for public announcements. If the api receives and outputs text that matches, then the notification is changed to a more generalised notification from the database and sent to the phone.

Firstly is the app's notification permission, it involves both ethical concerns and entire operation of the app. This issue can simply be solved with android studio. When asking the users if they allow the app to send notifications, the pop-up message will also mention that 'the app requires your microphone' and ask for the user's permission. Next comes to confidentiality. Recording audio and converting it to text may involve private or sensitive information. It is important to ensure that this data is protected from unauthorised access or disclosure. To do this we build an into app's database(that is, SQLite) and its security will be protected by app itself. Next is user control, which we have already considered at the beginning of the project, that is, the user should be able to have full control over the recording function of the app, specifically freely control when to turn recording on or off, and having the ability to delete past recordings. This gives the user a sense of autonomy and reduces privacy risks. Finally, and most difficult one, the ambient sound that an application monitor may inadvertently capture someone else's conversation and cause the application to be used abusively. Therefore, in the best case scenario, we have to set a clear scope for listening to audio stream sources. (This is not yet possible).

This course was my first attempt at coding a mobile app, I had no prior experience at all in this field. Although we were writing in java, I realised that it was very different from the computer-based java projects I had previously learned. A complete java mobile app contains many files that I am not familiar with (e.g. gurdle, main activity, application file, etc.), and I have to understand what role they play in a mobile app and edit them appropriately to make sure that the plugins, dependencies, etc. work correctly. The next step is to use firebase, which is a plugin that can be used to send notifications, analyse user behaviour, etc. As mentioned, I took responsibility of database, at first I wanted to use AWS, GCP for external data storage, but when I was searching relevant information, I found out that there is a common database software SQLite built in every mobile phone in default, and after I got more information with it, I even found that SQLite contracted more than 90% of the database work of the mobile app. I didn't know these things at all before I learnt about SQLite. The knowledge that I learnt on SQLite is very beneficial to me, and having this knowledge will make me more competitive when I am looking for a job in the future. As for the software development process itself, it’s proved a few time that software development processes are iterative, very often in the early stages of a project, people can always write their perfect expectations, but in the actual development of the project there is a high probability that they could not achieved all of them. Lastly, I am the only international student in the group, and I used to be afraid that my English and my learning ability would not be able to follow with team members, therefore I became quiet and almost didn't express my opinions, but that didn't mean I didn't have any opinion on the project. Ryan and Tom were there to encourage me to express my views. In fact, throughout the development period, it is frequent that my opinions actually resolved the problem. This reinforced my belief that sometimes just be brave to express my opinion, teamwork is an important part of a team, only more communication could make members know what you are thinking. Having good teammates and the experience of developing software with them is quite invaluable to me.

Assuming that we have enough time to completely develop this software, the first thing we would do is to solve the receiver problem, because in fact this is our original idea, which is to let the deaf people have other ways to receive information from outside world other than only listening to it via hearing loop. The next thing is trying to complete the CRUD principle for these recorded texts, to increase the user experience, and even to extend the app like a calendar or a memo application. Of course, all these are from my point of view, other members must have their expectations they want to expand, if time permits, I hope we can make a good work that meets everyone’s expectations.

I have learnt a lot from this project that I may need to keep in mind when developing other software in future. If I am going to develop same project with my team again, I believe I would try to express my views, try to relax and avoid putting too much pressure on myself. Trying to be as active as possible to help the team members, rather than being conservative and speak when the they are already in problem for a long time. I realised that sometimes, as an international student, I was the one who draw the line between myself and the local students (native English speakers), rather than they drawing the line to me.
This experience has taught me the importance of bridging that gap and being more open to collaboration, regardless of language or background.

When I first came to Australia to start my studies, I had used Otter, which is also a speech-to-text application. I somehow knew about Otter, therefore I took it into account and tried to improve our project based on Otter throughout the developing period. I can't say that our project could be better than Otter (it's impossible), but we did try to improve on some of Otter's disadvantages: Otter doesn't have an ‘add to my favourites’ feature, it simply monitors the entire audio stream of up to a few tens of minutes and converts it to text immediately. But the fact is that not every sentence is important, or students (probably) don't need Otter to translate every sentence from the teacher line by line. Therefore, our app will provide an on/off button, so that from the moment you turn it on, the phone will record the audio stream until you turn it off. It will then start processing the audio stream for that period and convert it into text. This gives the user quite a bit of freedom to choose which audio stream they want to convert into text.