After my team officially started working on the project, we have mostly maintained good progress and communication, almost everyone is working on their own tasks pretty well. Ryan, Tom, Jose are focusing on the front-end, and according to them they will be able to finish the front-end and start helping with the back-end development by next week. Kenny, Alhan, and I are responsible for the back-end. Kenny takes responsibility for the configuration of the speech to text api, and Alhan is responsible for the audio catching. And I'm currently in charge of notification sending, database in the app, and some github environment setup. I realised that the members might not know much about github, so I took over the task of managing github. I've created a github repository and a README that lists how to use github to manage our shared project files in a concise manner, and to help team members if they get confused.

Back to the project, I am responsible for the notification sending and external database deployment. For the notification, I use the Firebase system. Firebase is connected to the project to send notifications to the user when needed (i.e. when there is an external accountment or when the environment sound is received by the app and converted to text). However, there is a problem when it was finished. Firebase could be confirmed that had been connected to the notification java class (Android Assistant - firebase is ticked). It was also confirmed that the cannel ID were the same on both ports. But for some reason, it just doesn't work. After the team learnt about it, they were happy to help, I am very grateful to them and hope the problem can be solved as expected. My second task was to set up an external storage database. Tom's app design mentioned that it should allow users to sort messages by date, emergency level, and transport as features. I think that perhaps the most commonly used feature is the timestamp, and since announcement are generally more real-time, the other two features may not be as important. I'll talk to the group about this, it will make the database easier to set up and more efficient to query. As for the external database itself. I have experience with AWS (GCP for this semester), but I prefer to use local databases (SQLite) for the project, it mainly because of in this project we won't have heavy workload for database, and also it costs for us to use AWS. Of course, I will discuss my pre-decision with my teammates.

In fact, there was a risk within the team is that Alhan did not really participate much in the group's activities, it is common for him to be late or just absent. His absence also leads to a lack of communication with him, so we don't know the progress of his progress at all. We have raised this issue with tutor, and tutor has deal with it already. Alhan's part is audio stream catching, which is critical to our project, we need the app to be able to capture the audio stream from an external source (surrounding sounds) before we can start following works (speech to text) on it. Without Alhan's part of the app, that is to say, the whole app would not be usable.

When I first came to Australia to start my studies, I had used Otter, which is also a speech to

text application. I somehow know about Otter, therefore I took it into account and tried to improve our project base 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 a '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 freedom to choose which audio stream he wants to convert into text.