1. 在澳洲，有許多對身障人士有利的措施，Hearing Loop便是其中之一。Hearing Loop利用電磁場等科學原理，為佩戴著助聽器的聽障人士提供友善的生活體驗。在公車站、學校等等都可以見到hearing loop的身影。他們將對大眾的廣播透過電磁效應使得這些廣播能直接在助聽器內撥放以隔絕外界聲音的干擾。我們的專案計畫將開一個app，將這些透過hearing loop傳送到聽障者耳裡的音訊轉為文字顯示在他們的手機上，並額外增加震動功能已讓他們能獲取更全面的廣播體驗。另外，我們將以此為基底，允許使用者直接監聽環境音。讓使用者不限於只能從hearing loop獲取音源，可以直接接收環境音並將其轉為文字訊息。同樣支援震動以及顯示在手機上的功能，方便他們反覆查看。

本專案中，我將主要負責後端的部分，這包括正確部屬通知功能、app資料庫的建立。另外，我也將負責管理小組專案的github，並在隊員使用github時提供適當幫助。

1. 以下是我們專案app中的工作流程：首先，app在hearing loop或是公眾廣播中獲取音源，將這些音源錄入到app內部並透過speech to text轉為文字。轉為文字後並以適當的layout傳送通知並震動告知使用者可以查看轉為文字的訊息。同時這些文字會被儲存到app的資料庫中，以提供使用者可以從app中反覆查看這些訊息，甚至可以管理這些訊息(標記、刪除、以特定方式排序他們等等)。由此可知，傳送通知以及管理資料庫在此app生命流程中扮演著重要的角色。可以說是除了UI之外唯一會跟使用者進行交互的部分。

至於github的管理，因小組成員們在前期就已經告知對於github不太熟悉，我便主動攬下管理github的任務。我建立git repo，並且在專案下的README檔案簡單教導成員們如何從0開始的使用github。從git的安裝一直到正確push / pull本地專案至git repo中。在開發期間，若成員們在使用github上有問題(絕大多是都是與branch等議題相關)我也會提供協助。

1. 我們成員之中的團隊合作整體來說相當和諧。我們使用discord作為即時通訊軟體，在開發專案中如果有任何疑慮都可以直接在discord中發布，溝通毫無障礙。此外，我們維持著良好的開會頻率，就算是各自開發的期間，保持良好的開會頻率被認為也是有好處的，我們不僅能報告自己的進度給其他成員，同時也能知道當下成員進行的開發過程。我們在statement of work中編寫了timeline，基本上成員們都會根據timeline來開發軟體，並不會有成員不知道現在該做甚麼的問題出現。根據課堂上ECP各個interim assessments，我們也能夠知道開發速度使否符合教學團隊們的預期。基本上我們都能按時提交interim project並得到不錯的回饋。至少這證明著我們的開發流程並沒有太大的delay產生，這也讓我覺得，這是一個不錯且有效的團體軟體開發流程。最重要的就是保持聯繫，多溝通、多見面。
2. 以下是小組中各成員對專案所負責的部分：Ryan, Tom, Jose負責前端的開發。就我所知，前端的開發是一體成形的，並沒有明確地聲明各個部件的開發由誰負責。尤其是Ryan以及Tom，他們倆個是很好的朋友，幾乎形影不離，我相信他們通常來說都是待在一起開發專案前端的，包括從Prototype的設計到實際編寫代碼。Jose則是在每周五的上課時間由Ryan跟Tom與他進行溝通確認彼此的開發進度。與後端不同的是，前端的部分一直都是一體成形的，並不像後端(我、Kenny、Alhan一樣明確分離工作並最終merge到一起)。我、Kenny、Alhan則是負責後端的開發。先前提到，我的責任是正確部屬app的通知功能以及綜觀整個app的資料庫管理。Kenny則是負責有關speech to text的一切，而Alhan則是負責音源流在app內的傳送(意即，音源流被接收後傳至speech to text也就是Kenny的部分)。

Ryan, Tom, Kenny可以說是小組的核心，不僅僅是因為他們三個彼此認識，更重要者是他們三人兩人前端一人後端，將小組的開發以及組內氣氛變得相當和諧有步調。他們三人可以一方面與教學團隊溝通，另一方面也可以促進我、Alhan、Jose三人的溝通。這讓開發過程所有組員都能感到非常舒適，也讓我們成員之間的交流變得像是如與朋友一起做一個挑戰一般自在，很大程度地增加團隊的意識以及默契。

1. 在專案計畫中，我們原定app的功能能夠接收hearing loop的音源流並將其轉化為文字。為此我們從學校圖書館租借了一個receiver，該receiver可以透過hearing loop將教室內由麥克風傳出的音源轉化為音頻並輸出。而app將接收此音源輸入並將其轉化為文字。然而，這便是我們在本課程中遇到的最大瓶頸。不知為何，receiver總是無法透過hearing loop正確接收麥克風的廣播。這可能是麥克風與hearing loop連動的問題、也可能是hearing loop本身的問題、或可能是receiver的問題。為此我們花費了不少的時間嘗試解決，諮詢教學團隊、連絡學校圖書館、上網查資料等等。在經過兩個禮拜的努力後，問題仍然沒有解決，為此我們毫無頭緒，並思考著是否應該要放棄此功能。軟體開發是一個跌代的過程。在開發的過程中可能會遇到很多的困難，實際上並不需要完成所有當初在設計軟體時所提到的所有內容。另一堂課的sprint processes, MVP等等概念在此時獲得了實踐。因此我們打算先將receiver接收hearing loop這條道路先行擱置。既然我們已經決定捨棄hearing loop的功能，我們將會對剩餘的功能，也就是直接監聽環境以獲取音源的功能進行更加完整的開發。我們第一點想到的便是CRUD原則。除了基本的CR(用於在接收到音源流並將其轉化為文字後存儲於資料庫中以供顯示在設計中的”history部分”)之外。我們還可以添加UD原則。我們將會提供”加入到最愛”功能，這個功能允許使用者可以把特定的某一段文字加入到最愛以供後續查閱。這個就必須要使用到u的功能，原本所有的資料其title以及”favourite”屬性都被預設為null。利用update來更改該筆資料的title, favourite就可以提供使用者賦予某段文字一個特別的標題以及是否被加入到最愛。D原則：我們將提供使用者可以刪除某段或是多段文字的功能。雖然他基本上都是讓使用者刪除過多的資訊，但提供此功能給使用者將提高他們的用戶體驗。

另外，kenny有提到一個問題，若是我們目前只確定開發環境音源流，有一個問題是當如果環境聲音過於吵雜以至於api無法完整閱聽廣播(例如，車上廣播提及，"下車時請確認好行李再下車，謝謝"但應環境聲音過於吵雜api只能閱聽到->下車時XXXXXX，謝謝)時該如何應對？ 我們的解決方案是新建立一個資料庫用於存取常用的車上廣播的"關鍵詞"。若api接收並輸出的文字與之匹配，則將通知改為資料庫內的較為一般化的通知送出到手機。

1. 首先便是app的通知權限，無論是出於道德問題或是app的運作。我們都應該要確保用戶在使用我們的app時知情此app可能會錄音並且傳送通知。這點透過android studio便可以解決。在詢問用戶是否允許此app傳送通知時便會提及”此app要求使用麥克風”等等有關的問題並尋求使用者同意。接下來是保密性，錄製聲音並將其轉換為文字，可能會涉及私人或敏感信息。必須確保這些數據被妥善保護，防止未經授權的訪問或洩露。為此我們會在app的資料庫中建立了該使用者在該app生命週期中所需的安全保障。同時，由android studio所建立的app也默認地會對整個app進行相當程度地保護。再來是用戶控制，我們已經在專案形成之初考量到這個問題，那就是用戶應該能夠完全控制應用程式的錄音功能，比如能選擇何時開啟或關閉錄音，並有能力刪除過去的紀錄。這樣可以讓用戶感到有自主權，並減少隱私風險。最後也是最難的一點，應用程式監聽的環境聲音可能無意中捕捉到他人的對話造成應用程式濫用。因此，在最好的情況之下，我們必須設定明確的音源監聽範圍。(但事實上還無法完成)。
2. 1
3. 本堂課程是我第一次嘗試撰寫手機app，在這之前我並無任何相關經驗。雖說我們是利用java來撰寫，但隨著開發前進，我認知到，這與我先前所學到的以電腦為主體的java專案有很大的不同。一個完整的java 手機app包含許多我不熟悉的文件(我最常編輯的是gurdle以及main activity, application等等)，我必須去了解他們在一個手機app中扮演著什麼樣的角色並適當編輯他們以確保插件、依賴等等能夠正確運行。接下來是firebase的使用，firebase是一個多功能的插件，它可以用來發送通知、分析使用者行為等等，具有強大的商業價值。我本身是自許成為一個資料庫工程師，因此我主動覽下專案的資料庫相關工作，一開始我想要使用AWS、GCP用於外部資料的存儲。但在實際編寫代碼時，我發現原來手機app內有內建簡易資料庫軟體SQLite，熟悉了解之後甚至發現SQLite承包了90%以上的手機app資料庫工作。這些事情是我在了解SQLite之前完全不知道的。這些從做中學的知識讓我獲利良多，擁有這些知識可以讓我在未來找工作時變得更有競爭力。至於與軟體開發過程的本身。我認知到，軟體的開發是一個迭代的過程，很多時候在專案的初期，人們總是可以把期望寫的天馬行空，但實際開發專案時很大機率可能不是如此。當我在我國家就讀學士期間(非IT專業)，教學團隊總是要求我們必須完全達到”你所提出的構想”。因此當我們遭遇困難時，我開始變得焦慮、恐慌，我害怕無法完成預期的任務(例如一開始的通知無法正確傳送、AWS、GCP可能不適用於手機app)，我害怕因為無法完成任務而給予其他隊友極大的麻煩。最後還是我鼓起勇氣向其他隊員表達我的堪憂。然而，我的隊友們(都是澳洲本地人)卻告訴我了他們的看法，也就是強調軟體開發是迭代的過程，並且在非常有限的教學期間之內，本就無法盡善盡美地完成所有預期的任務。他們提到，這就是MVP以及軟體開發周其中sprint的存在意義。我們只需要在最短的時間內，盡我們所能地去嘗試達到MVP。Ryan某次小組開會後跟我提到：難道一款遊戲在發布之初就不能有任何bug或是他應該一次性地發布所有遊戲內容嗎？這句話如同醍醐灌頂點醒了我。我就放鬆下來，嘗試不在給蓋多壓力到自己，整理好思緒後重新專注在我眼前應該做的事，而不菲遙遠之處的虛無飄渺。最後，如同前面所述，我是小組之內唯一的國際學生，我曾經也害怕我的英文能力、我的學習能力無法跟上他們，因而變得沉默寡言，幾乎不發表我的看法，但這並不代表我對專案沒有意見。同樣是Ryan(Ryan事實上為台裔澳洲人)以及Tom。他們給予我安慰，鼓勵我表達自己的看法。事實上，我所提出的意見很大程度都會被它們採用。這更加讓我確信，一個團隊之中，團隊精神是很重要的一環，儘管每個人都有屬於自己的優缺點，但作為一個團隊，我們就應該要團結在一起地去完成任務。擁有這些好隊友以及與他們一起開發軟體的這個經驗對我來說是難能可貴的。

假設我們擁有相當充足地時間可以去開發這個軟體，首當其衝便是解決receiver的問題，因為事實上這才是我們最原始的想法，就是讓聽障人士在使用hearing loop的時候除了用耳朵聽之外還有其他方法能夠接收到來自外界的聲音。接下來就是方才提到的，對於這些被記錄下來的音源，盡量地去完善所謂的CRUD原則，增加用戶體驗，甚至可以將app開發為類似notion的形式，擴展成行事曆、備忘錄等等的功能。當然，這些都是從我負責的角度出發，其他成員們肯定也有他們想要擴展的功能，若時間允許，我希望我們能做出一個符合我的所期待的好作品。

我在本次專案學習到了許多可能真正在開發一個軟體時所需要keep in mind的想法。若與此團隊的大家再次開發一個此專案，我相信我會盡量去表達我的看法，嘗試放鬆並避免給自己太大的壓力。盡全力地主動去幫助隊員，而非當隊員們已經一籌莫展的時候，才保守地講出自己的看法。我認知到，有的時候，是做為國際生的我先入為主地去劃清我與本地學生(英語母語者)的界線，並非他們對我劃清界限。在一個團隊中，多交流、勇於表達自己的看法、多討論、互相幫助才是重要的。

1. 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.