Requirements Specification

SmartTalk

November 21, 2020

Sponsor: Dr. Okim Kang

Mentor: Fabio Santos

Version: 1.0

Team members: Joseph Vargovich (Lead), Andrew Munoz, Kehan Cao, Christian Bito-on, Malik Jones

Accepted as baseline requirements for the project:

Client Signature: ______________________________________ Date: __________

Team Signature:____Date:___________

Table of Contents

1.0 Introduction

Mobile language learning apps such as Memrise, Duolingo, and many more try to help users learn a second language. Most of all these language learning apps lack the focus on pronunciation practice. This causes many language learners to be unable to communicate with others in their target language. Also, some of these programs do not provide significant feedback on supra-segmental features of pronunciation such as word stressing on the correct syllable. Another limitation is the gamification to motivate the practice and acquisition of pronunciation. By applying gamified elements into the language learning elements correctly will boost engagement, provides motivation through goal tracking, improves knowledge retention, promotes team building, and offers valuable feedback. Implementing gamification elements into mobile applications they should be able to enjoy the activity and be able to focus more on pronunciation rather than memorizing how to say a phrase in that specific language. If your users don't focus on the pronunciation to obtain a secondary language and try to speak to someone in that foreign language that native will not be able to comprehend.

Our sponsor is Dr.Okim Kang who is a Director of the Applied Linguistics Speech Lab and a Professor at Northern Arizona. The Speech lab focuses on CAPT (Computer-Assisted Pronunciation Training). Her overall main research goal is to analyze the nature of accented speech of non-native speakers in English:

- ➔ How accent is perceived by listeners
- ➔ How accented speech is characterized linguistically,
- ➔ How the assessment of accented speech is validated through automatic systems,
- ➔ How speakers with accents can better communicate with others.

This project will benefit our sponsor by developing a tool for her graduate students to use for their teaching purposes to help speakers with accents that can better communicate with others by sharpening their pronunciation skills. SmartTalk is thrilled to be working with Dr.Kang on developing a mobile app to help improve students' pronunciation and perception of a target language in the most engaging and entertaining way possible.

2.0 Problem Statement

By looking into the project details of what the dilemma is, our sponsor states that she noticed the lack of mobile software for foreign language learners that both focus on pronunciation and encourages learning through gamification. The current software applications for language learning have many drawbacks such as:

- ➔ Being both costly to the learner and prohibitive for researchers to use.
- ➔ They do not allow course designers to create relevant tasks for learners.
- ➔ A lack of relevant feedback on key features of pronunciation.
- ➔ They do not implement gamification to motivate the practice of pronunciation.
- ➔ Most modern ASR technology is difficult for linguistics researchers to use.

While developing this project with Dr. Kang we hope to have a basic functional workflow for her and a speech lab to analyze non-native speakers in English. As well as being able to pass down our work to the next capstone group. After all the issues that we have discussed with Dr. Kang, we have narrowed down the core issues and will be able to solve most of the issues by the end of the Spring 2021 semester. By having a clear understanding of the problem and being able to focus on the main needs of our client's team, we have now come up with a solution to effectively accommodate the issues present within the workflow of our client and their research goals.

This project will provide Dr. Kang's team with easy access to ASR analysis and pronunciation data gathered from the learner users of the mobile application. This will allow for a convenient way to analyze language learning patterns in an efficient manner. This project will improve the overall workflow of Dr. Kang's team by providing an alternative to expensive, slow, and difficult to use CAPT software systems the team currently relies on for pronunciation training and analysis. Through gamification, learners will be encouraged to practice their pronunciation skills and therefore provide linguistics researchers with valuable information to further their studies of English language learning and linguistics as a whole.

3.0 Solution Vision

For this app, we have a huge dependence on the data collected from users. This data will be retrieved through our gamified aspects as well as ASR technologies and will include email addresses and passwords for specific users, which we will use to assign users to specific teachers and custom lesson plans, as well as pronunciation and speech data to track progression and award users with milestones marks.

Deriving this data from ASR technology will allow our system to generate a series of scores and rankings that the users may use to track their progress in speaking a certain language. Our system will also provide our sponsor with an easy-to-use application that can compute user data, including how users are learning, as well as speech patterns from users who are just beginning to learn a language.

Collecting extensive data will also allow us to analyze the role that gamified elements play in learning a new language, as compared to language learning apps such as Duolingo which are focused on repetition and a simple interface. While we considered building the app to be more focused on gamified elements rather than ASR technology, we ultimately decided to incorporate more of the ASR technology in order to make the app more beneficial to users in terms of learning outcomes. This allows for constructive feedback as well as boosts the information we get from users themselves.

From a long-term view, this software could lead to a new way of learning languages by creating a way to learn that is both fun and simple for users, as well as incorporating more than simple repetition as the main way to learn. This can also lead to children to have a new way to learn a language where they can learn so in an environment meant to keep their attention and teach them at the same time.

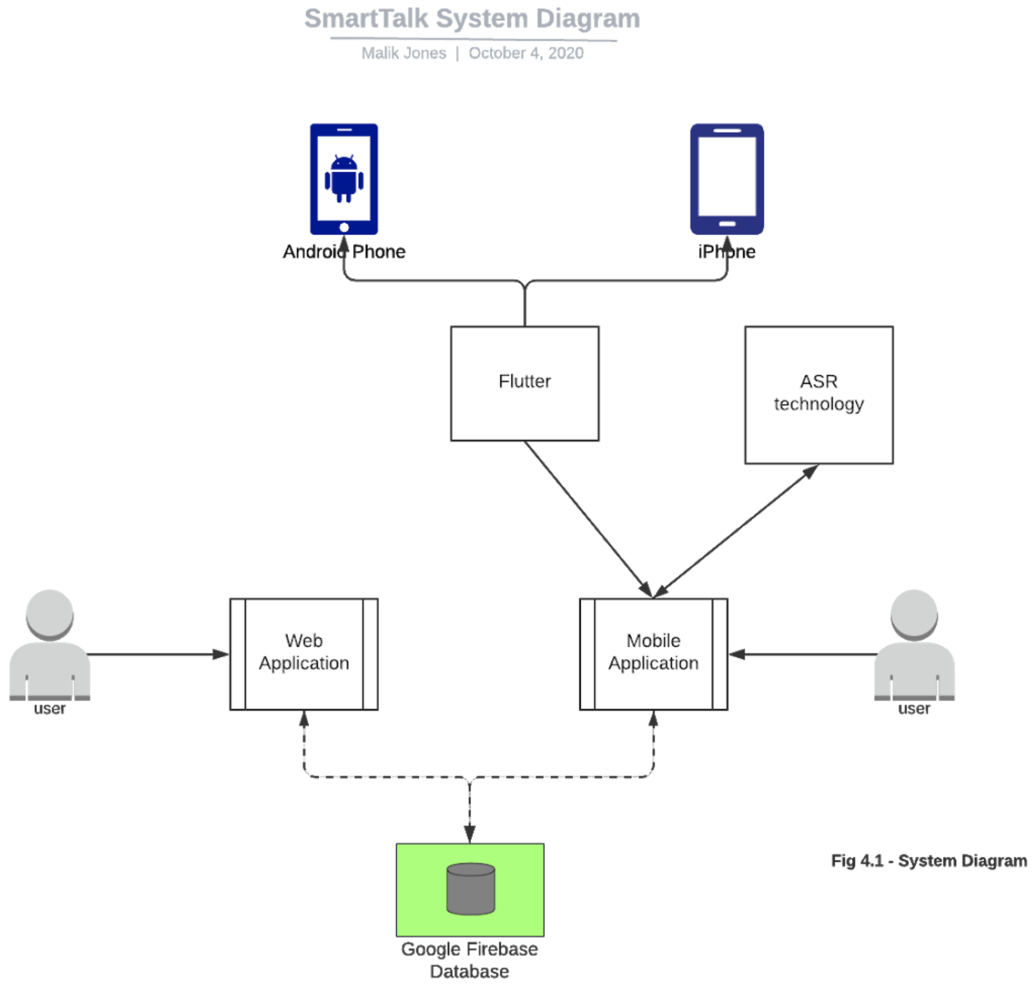


Figure 1 - Diagram of the overall architecture of the system as well as the integration of each technology.

4.0 Project Requirements

There are a variety of Functional and Performance requirements for this project. These functional requirements will serve as a metric for measuring project progress and the performance of the team.

4.1 Functional Requirements:

The functional requirements for this project can be divided into several high-level functions. These functions represent the core tasks of the mobile and web applications and the chief functionality desired by the client and her team. These core functions are as follows:

- ➢ Creation of Courses, Modules, Lessons, and Tasks through the website dashboard
- ➢ Creation of three main types of tasks: Production, Perception, and Instruction Tasks
- ➢ Gamification to incentivize and motivate user learning and practice
- ➢ ASR usage and feedback to be used in linguistics research and analysis of language learning

Below we detail and explore these four high-level functions and their associated requirements.

4.1.1 Creation of Courses, Modules, Lessons, and Learner Tasks (Course Structure)

Each course designer can create a Course, which contains Modules, which contain Lessons, which finally contain Tasks for learners. Figure 2 outlines the hierarchy of Courses, Modules, Lessons, and Tasks.

Courses will have a variable amount of modules assigned to them by the course designer. The number of modules per course is completely up to the designer. Learner users will be assigned to each course and will be able to see its modules.

Modules will have a variable amount of lessons assigned to them by the course designer. The number of lessons per module is completely up to the designer. Learner users will be able to access each module when it is revealed to them by the course designer.

Lessons will have a variable amount of learner tasks assigned to them by the course designer. The number of tasks and task types in each lesson is completely up to the designer. Learner users will be able to access each lesson when it is revealed to them by the course designer.

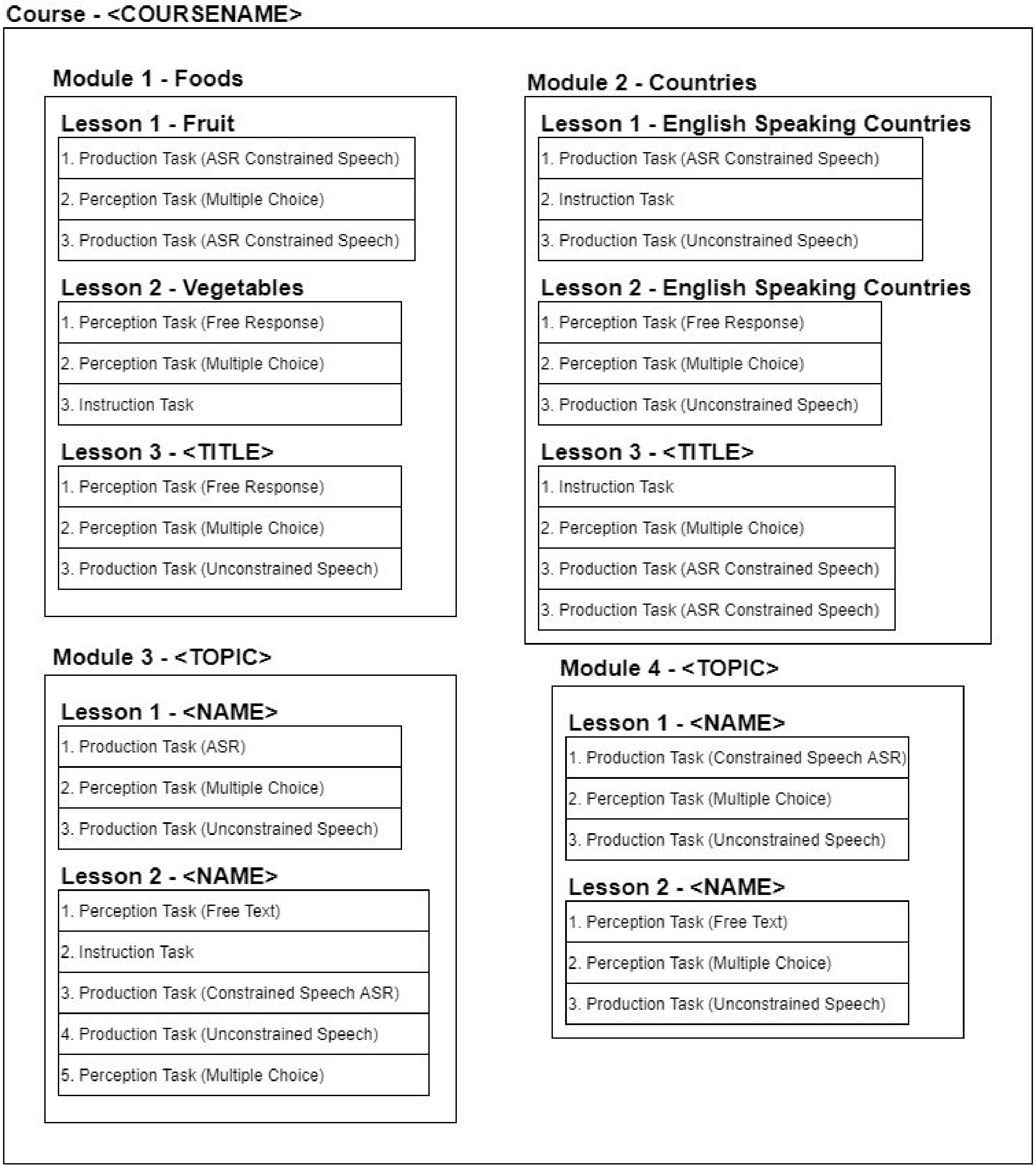

Course - <COURSENAME>
Module 1 - Foods
Lesson 1 - Fruit
1. Production Task (ASR Constrained Speech)
2. Perception Task (Multiple Choice)
3. Production Task (ASR Constrained Speech)
Lesson 2 - Vegetables
1. Perception Task (Free Response)
2. Perception Task (Multiple Choice)
3. Instruction Task
Lesson 3 - <TITLE>
1. Perception Task (Free Response)
2. Perception Task (Multiple Choice)
3. Production Task (Unconstrained Speech)
Module 2 - Countries
Lesson 1 - English Speaking Countries
1. Production Task (ASR Constrained Speech)
2. Instruction Task
3. Production Task (Unconstrained Speech)
Lesson 2 - English Speaking Countries
1. Perception Task (Free Response)
2. Perception Task (Multiple Choice)
3. Production Task (Unconstrained Speech)
Lesson 3 - <TITLE>
1. Instruction Task
2. Perception Task (Multiple Choice)
3. Production Task (ASR Constrained Speech)
3. Production Task (ASR Constrained Speech)
Module 3 - <TOPIC>
Lesson 1 - <NAME>
1. Production Task (ASR)
2. Perception Task (Multiple Choice)
3. Production Task (Unconstrained Speech)
Lesson 2 - <NAME>
1. Perception Task (Free Text)
2. Instruction Task
3. Production Task (Constrained Speech ASR)
4. Production Task (Unconstrained Speech)
5. Perception Task (Multiple Choice)
Module 4 - <TOPIC>
Lesson 1 - <NAME>
1. Production Task (Constrained Speech ASR)
2. Perception Task (Multiple Choice)
3. Production Task (Unconstrained Speech)
Lesson 2 - <NAME>
1. Perception Task (Free Text)
2. Perception Task (Multiple Choice)
3. Production Task (Unconstrained Speech)

Figure 2 - Overview of course structure and hierarchy, of Course, -> Module -> Lesson -> Task. Topics in <> symbols are editable, with some fields already populated with sample titles.

A **minimum viable product** will fulfill the following course-related requirements:

1. User accounts can be created both on the website dashboard and mobile application. Registered users can be assigned to either the learner of designer roles depending on the platform they are using. Learners can be assigned to a course, and a course can have multiple designers in charge of creating and managing it.
    a. User creates an account by signing up on the login screen.
    b. User signs into this account through the login screen with email and password
    c. User account keeps track of assigned courses, modules, lessons, and tasks.
    d. User account keeps track of assigned modules, lessons, and tasks.
    e. Users can sign out and sign in again and have their progress saved.
    f. Users can change their password using their provided email.

2. Designers can create and populate courses through the website dashboard and assign registered learners to the course. Courses and their content can be edited and added to after creation.
    a. Designer signs into the website dashboard and clicks the add course icon
    b. Designer enters a name for the course: ENG 400, then returns to the course list and sees that ENG 400 has been added to their course page. This name can be changed by clicking the edit module button on the created course.
    c. Designer returns to the course list page and clicks the “add learner” button
        i. Designer then assigns a user (via their email address) to the course.
        ii. The specified User is then added to the list of Learners for the created course. The Designer can click the list of learners button to see who is enrolled in the course.
    d. Designer clicks on the view button for the course and then is navigated to the modules page for the course.
    e. Designers can then click the add module icon on the module page for the created course.

f. Designer will be navigated to the create module form, which will ask for a name of the module (Module 1: Foods). This name can be edited by clicking on the edit module button on the modules page.

g. Designer will return to the module page and see Module 1: Foods listed.

   i. Designers can open the module to the entire class, learners registered in ENG 400 can now see Module 1.

h. Designer can view the module by clicking it

i. Here the Designer will see a page that allows them to add lessons

j. Once a lesson is added, designers can assign specific learner tasks to it, described below in section 4.1.2

A **fully usable product** will fulfill the following course-related requirements:

1. Courses will be set to expire within a given range of dates and/or time periods.
2. The visibility of Courses, Modules, and Lessons will be toggled and/or assigned to the desired time period by the course designer.
   a. Designer will click the atomic visibility switch to show to all users or hide from all users
   b. Designers will enter a range of dates that the course, module, or lesson will be open for.
3. Additional lessons can be assigned to specific users as course designers see fit.
   a. The Designer user assigns a specific lesson to the learner user that is not visible to other learners.
   b. The Designer will click
4. Designers can specify the behavior of a lesson by requiring learners to retry a lesson if they perform under a threshold they specify.
5. Designers can send direct messages to specific Learner users.

Some **stretch goals** related to course design and management include:

1. The addition of live lessons that incorporate multiple learners simultaneously.
2. Learner collaboration and feedback on each other's performance on a given lesson or task.

4.1.2 Types of Learner Tasks

Within any given lesson, there will be three given types of tasks that the learner will complete. These tasks can appear in any order as constructed by the course designers. Furthermore, there can be a number of each type of task in any given lesson and are all designed by the instructor.

Instruction tasks teach the learner the pronunciation of a sound or word and are accompanied by a sound recording example. In addition to the recording, course designers should also be able to place instruction text to help guide learners.

Perception tasks involve the learner by requesting they select answers based on an example. These tasks can involve selecting the most correct answer, ranking answers based on correctness, or typing in the answer. For this type of task, if a user does not correctly answer, the course designer can decide if the learner should: simply move one, get recommended to try again but allow them to move on, or be required to try again until the task is correctly completed.

Production tasks require the user to speak into the device a specific word or phrase related to the lesson. This type of task will incorporate ASR technology to produce feedback. There is also an unconstrained version of the production that users will record their speech and send it to instructors for manual feedback.

A **minimum viable product** that these three types of tasks entail are:

1. There will be three different types of tasks:
    a. Instruction tasks to teach the user the lesson sounds
        i. These tasks will provide a sound recording to the user to guide them in pronouncing the sounds
    b. Perception tasks will have the user identify a lesson sound
        i. These tasks will have a question prompt asking to identify the sound
        ii. These tasks will provide a sound recording as an example
        iii. These tasks will be answered in a multiple choice format
    c. Production tasks will have the user recreate the lesson sound
        i. These tasks will prompt the user to reproduce an example sound
        ii. These tasks will have accompanying recordings as an example

        iii. Constrained versions of this task, where the outcome is identified, will use ASR technology
        iv. Unconstrained versions of this task, where the outcome is dependent on the user, will instead by manually reviewed by the course instructor

A f**ully usable product** related to the requirements will incorporate:

1. Instruction tasks will include more information about the lesson
    a. These tasks will include instruction text provided by the course instructor detailing pronunciation as well as keynotes about the specific sound.
2. Perception tasks include more methods of testing/answering questions
    a. More types of answering methods:
        i. Type the answer in
        ii. Rank the answers
        iii. Multiple answer
    b. Perception tasks implement a retrial system depending on the settings laid out by the instructor:
        i. Move on regardless of the answer.
        ii. If incorrect, **recommend** trying again.
        iii. If incorrect, **must** try again.
3. Instructors can provide more aid to the learner in the form of text editing tools:
    a. Being able to bold and underline text
    b. Add in specific instructions
    c. Add in pictures

Some **stretch goals** in regards to these three types of tasks include:

1. Perception tasks now implement more complex question types:
    a. Listening to a scenario and responding to it
        i. Multiple choice or type in the answer
        ii. Unconstrained recording that will be reviewed by the instructor
        iii. Video Playback for these scenarios
2. Instruction tasks have video playback for further clarity.

    a. Shows speaker pronouncing the sound

4.1.3 ASR Feedback and Analysis

Within constrained production tasks, where users are given a sentence or a phrase to repeat, ASR technology will be used to gather information from the user and automatically calculates the results, mainly in the form of recognized words. These results will be utilized by both the user and the instructor.

On the user side, the results will be compared to the correct answer and are either deemed as correct or incorrect. Users should also be able to view the results and see which words they have mispronounced, regardless of whether their complete answer is determined to be correct or not.

On the instructor side, the ASR feedback will be sent to them formatted alongside a recording of the speaker in the default ASR file format that will accompany this file, that instructors are able to view manually through the website. Through the manual review, instructors can override automatic feedback if need be.

Minimum viable product requirements that ASR feedback will have to meet are:

1. The user should be able to speak to the device and have it be analyzed by the ASR technology.
    a. Should recognize individual words and full sentences
2. Calculated basic results in the form of binary feedback (correct/incorrect) should be given.
    a. For full sentences, if ASR recognizes the word the instructor is looking for, then the answer is deemed correct
    b. For individual words, if ASR recognizes the sound the instructor is looking for, then the answer is deemed correct
3. Feedback should be shown to the user so they can review their work.
    a. For words, a message saying that the word ASR is looking for was not found
    b. For sound, a message saying that the sound ASR is looking for wais not found

A f**ully usable product** related to the requirements will incorporate:

1. Formatted ASR feedback for instructors to view.
    a. The feedback given to the user should be included for reference
    b. For words, a list of recognized words
    c. For sounds, a list of recognized sounds
2. A recording of the user's speech alongside the ASR feedback file should be sent to the database for review
    a. The recording format will be in a .WAV file format
    b. The recording should be able to be played from the website
    c. Feedback file will be in the form of a text file
3. Instructors should be able to override automatic results.
    a. For automatically graded production tasks, the instructor can change the correctness of the student's response.

Some **stretch goals** regarding ASR feedback are:

1. Pitch and intensity graphs created from feedback
    a. These graphs will be shown as a picture
2. Scalar feedback in the form of a percentage of correction, i.e. 80% of the user phrase is correct.
    a. For words, a percentage of words out of a sentence were correctly identified
    b. For sounds, a percentage of sounds in a word that were both correctly identified and are in the correct order
3. Adjustable ASR thresholds and settings for more detailed feedback and tracking of relevant statistics from a user response.
    a. ASR thresholds include deeming sounds similar to the lesson sounds as correct or partially correct.
    b. Settings to include the recognition process of the ASR program in the text file
4. Further formatting of the ASR information into a file format such as JavaScript Object Notation (JSON), Excel Spreadsheet (xlxs), or Comma-Separated Values (CSV).
    a. For sentences, group each individual lesson word with the word ASR recognized

b. For words, group each individual lesson sound with the sound ASR recognized

4.1.4 Gamification

In order to implement gamification into the mobile application, there will be badges, achievements, and several other game mechanics that will allow the user to enjoy the course. Having a well-balanced learning objective alongside these game mechanics will ideally have the user fully engaged within the activity, and have them spending more time practicing the material. The system must show a badge when the user reaches a threshold defined by the client.

A **minimum viable product** to implement game mechanics to the language learning app will include:

1. Badges are given to the user upon completion of an activity, with different badges depending on the specific percentage they receive on the perception or production task.
   a. Learner completes a lesson and receives a badge
      i. Gold Badge 90-100%
      ii. Silver 80-89%
      iii. Bronze Badge 70-79%
2. Achievements given to the user upon completing the entire module demonstrating a full understanding of the material.
   a. Achievement is given to the learner when completing the entire module of that course.
   b. Badge is given to the learner when completing a lesson associated with a module.

A f**ully usable product** related to the requirements will incorporate:

1. Users will be able to examine a progress bar or percentages of completion in order to show them how far they are in each task.
2. Users will be able to obtain 10 unique badges and 5 unique achievements by completing materials in the modules and lessons for a specific course.

3. Daily practice streaks will allow the user to repeat exercises and gain new rewards, such as achievements.
4. Pop-up notifications with encouraging messages to the user based on their percentage in the course.
    a. The Designer can select a Learner to send a pop-up message to or send it to the entire class.

Some **stretch goals** regarding game mechanics to implement into the language learning app could include:

1. Peers can compete and quiz each other through the app within a specific course.
2. Classwide live quizzes and events.
3. Preset games that are simple for the user to interact with, and which administrators will be able to place into the lesson plan for their course.
4. A leaderboard displaying users with the most achievements
5. Send prizes and/or points for users to purchase simple prizes with.

4.1.5 Core Use Cases and Users

The two core users of our system are:

Learners

Learners are the users of the mobile application. Students create an account and sign-in through the mobile application and access courses they have been assigned to by the course **designer** user. Learners are able to complete the modules, lessons, and tasks assigned to them by the course **designer.** Learners may be asked to repeat tasks and/or lessons based on the preference of the course designer. Learners are freely able to set a profile picture and a color scheme preference from some preset options and avatars. More avatars can be unlocked through a point system in the future.

Designers

Designers are the users of the website dashboard. Designers may include teachers, researchers, or students themselves. The primary goal of this user is to create a course that contains modules, lessons, and tasks for the **learner** user to complete on the mobile application. The designer is able to view results of ASR analysis, view user scores and completion of lessons, and suggest additional tasks for learners to complete based on how well they are performing on each type of task that has been assigned to them. O Designers determine the criteria for learners to move on in each module and/or lesson.

Designers can set three conditions at the task level in response to an incorrect answer:

1. Learners are prompted to move on with the lesson without a retry.
2. Learners are recommended to retry the task but are not required to.
3. Learners are required to retry the task, and cannot move to the next task without doing so.

Once learners have completed a lesson, the designer can view the ASR feedback from production tasks, as well as the overall performance statistics of the learner's performance.

Figure 3 outlines the connections between Learners and Designers via a use case diagram. These two users are the primary users of mobile and web applications respectively.

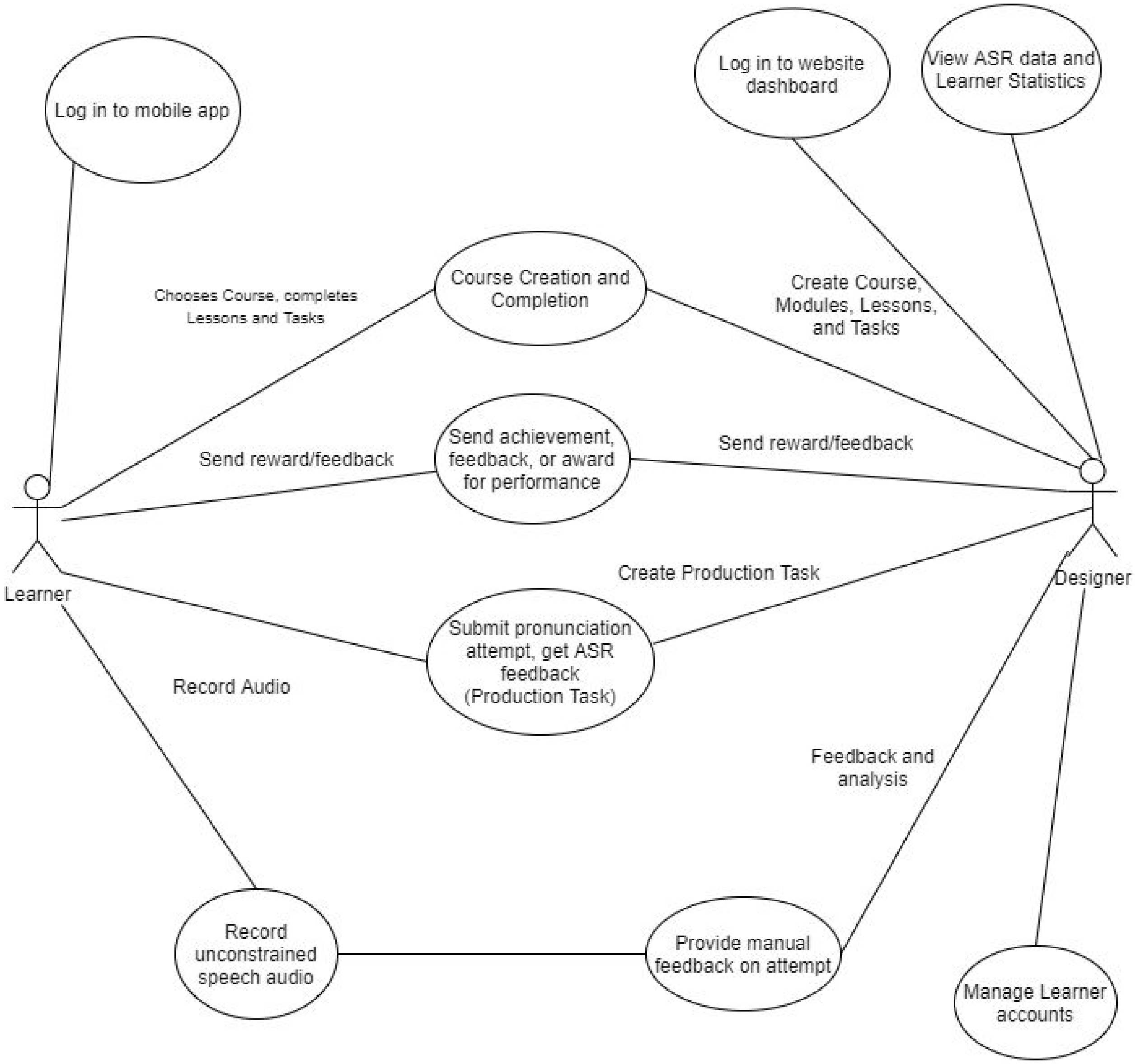

Log in to mobile app
Log in to website dashboard
View ASR data and Learner Statistics
Chooses Course, completes Lessons and Tasks
Course Creation and Completion
Create Course, Modules, Lessons, and Tasks
Send reward/feedback
Send achievement, feedback, or award for performance
Send reward/feedback
Learner
Designer
Create Production Task
Submit pronunciation attempt, get ASR feedback (Production Task)
Record Audio
Feedback and analysis
Record unconstrained speech audio
Provide manual feedback on attempt
Manage Learner accounts

Figure 3 - Use case diagram showing the relationship between Learners and Designers and their use cases for the proposed system.

4.2 Performance requirements (Non-functional requirements)

There are three key metrics that determine the performance requirements of our system. These three requirements are usability, speed of ASR feedback, and battery life.

4.2.1 Usability

Usability is defined as the amount of time it takes users to complete a task using mobile and web applications. Usability testing will be conducted via user testing of the software product. This will be quantified by recording the amount of time it takes user testers to perform key use cases of the system. For the mobile application, we will ask users to perform several tasks:

- ➢ Please create an account through the mobile application. Time expected: 2 minutes.
- ➢ Please log in to the mobile application. Time expected: 1 minute.
- ➢ Please navigate to the course module you have been assigned to by a course designer. Time expected: 1 minute.
- ➢ Please click on the first module listed in the course: Time expected: 30 seconds.
- ➢ Please navigate to the first lesson of the module: Time expected: 10 seconds.
- ➢ Please start the lesson and complete the first production task: Time expected: 2 minutes.
- ➢ Please move on to the next task, which is a perception task. Complete it and submit your answer. Time expected: 2 minutes.
- ➢ Please continue and finish the remaining 8 questions in the lesson. Time expected: 5 minutes.

These tasks outline the primary use cases of the mobile application. We will accept the state of each use case if 80% of users can perform the task listed in the expected time. If less than 80% of users can complete the task, we will need to either adjust the timing goal or improve the user interface and tutorials to make sure users understand how to perform each task efficiently. These timing windows will likely be refined and discussed after early user testing.

For the website dashboard, a similar process will be used. This time, there will be different tasks that are specific to course designers and their use cases. The desired tasks are as follows:

- ➢ Please create an account through the website dashboard. Time expected: 2 minutes.

- Please log into the website dashboard. Time expected: 1 minute.
- Please create a course titled ENG 400. Time expected: 2 minutes.
- Please navigate to your new course ENG 400. Time expected: 30 seconds
- Please create a new blank module within your course. Time expected: 2 minutes.
- Please add a new blank lesson to the module. Time expected: 1 minute.
- Please add a blank production task to the lesson. Time expected: 1 minute
- Please add a question to the production task. Highlight and underline two words of the question. Time expected: 1 minute.
- Please add an expected answer to the production task. An audio recording of the word. Time expected: 2 minutes.
- Please add a new perception task. Time expected: 1 minute
- Please add an expected answer to the perception task. This could be a multiple-choice selection, free text answer, or short answer to be matched. Time expected: 1-2 minutes.
- Please add a new instruction task, detailing some information that the user should learn about. Time expected: 1 minute.
- Please confirm your lesson with the three tasks. Time expected: 1 minute.
- Please toggle the visibility of the lesson to the registered users of the course. Time expected: 1 minute.
- Please access the associated ASR feedback data from the production task. Time expected: 1 minute.

These tasks outline the primary use cases of the web application. We will accept the state of each use case if 80% of users can perform the task listed in the expected time. If less than 80% of users can complete the task, we will need to either adjust the timing goal or improve the user interface and tutorials to make sure users understand how to perform each task efficiently. These timing windows will likely be refined and discussed after early user testing.

4.2.2 Speed of ASR Feedback

The ASR results and analysis should take no more than 1 minute to complete. The user should be able to continue on to other exercises while the ASR analysis is running. Once the analysis completes, the user can submit or review the automatic feedback.

In order to test and verify this requirement, we will time the ASR analysis and ensure that it does not take over 1 minute to complete. If the analysis takes more than 1 minute, we would likely need to refine our ASR option or accept that the analysis will take a longer time than expected.

4.2.3 Battery Life and Phone Resources

As we are planning to run the ASR analysis on the user's mobile device, the amount of battery life and CPU usage of the phone must be considered. ASR analysis is a very performance heavy operation that must be monitored to ensure that the user's mobile device can keep up with the intense computational load. The main advantage of completing this task on the mobile device is the speed and relative ease of communicating ASR results to the main mobile application.

In order to test this, we would create a 10 question lesson that consisted only of ASR production tasks and see how much battery life and computational resources were used to complete all of the required tasks. We would like to keep the battery usage under 5% for the entirety of this 10 question lesson.

Environmental Requirements

There are not any significant environmental requirements placed on this project. We are free to use any open-source frameworks for mobile and web development. The client's team has used Kaldi for ASR in the past, but this is not a forced option for ASR feedback.

5.0 Potential Risks

For our system, there are a number of risks that affect both the learning capabilities of our users, as well as the research capabilities of our instructors, which chiefly are:

- Data Loss through Database Failure
- Ethics
- ASR Accuracy

While these issues are mainly out of our control, there are some potential solutions that we can implement to alleviate these risks.

5.1 Data Loss through Database Failure

As our system utilizes a database to manage information between both the mobile app and website, a potential database failure and resulting data loss heavily impacts both our users and researchers. This data loss can result in losing course information, user recordings, and user information. However, we can reduce this risk in two ways: First, since we are using a provider for our database, we offload most of the risk to them, as they are responsible for the upkeep and maintenance of the database. The other risk management plan we can implement is keeping backup versions of everything on the mobile phone. Since the website cannot store large chunks of information, backups of stored information such as course content and recording can be saved on the phone for reupload in case of database failure to recreate most of what will be lost. Currently, as the courses are designed by the instructor for a group of students for use, one of the students within that course can reupload the most recent version of the course back to the database. However, this requires that everything stored on the database is stored on at least one phone, so any information only in the database will still be lost. With the above, the risk is minimized through both offloading the risk to a third party, and adding in a method of reconstruction by having application users be able to reupload the downloaded course and recording information.

5.2 Ethics

One large issue pertaining to users and their learning experience is the ethics of how our applications will teach students. Other applications often advertise that users will be able to speak like a native. However, the issue with that claim is that speaking like "native" depends entirely on that language's region and dialect. This presents the risks of developing wrong language ethics. For example, teaching a user English based entirely around New York English and declaring that to be "native" when other equally-correct dialects like Texan English exist gives users incorrect information that might cause issues for them later on. To alleviate this issue, the implementation of complete course customization can help with eliminating this stigma, as course designers, which will mainly consist of language teachers, will have complete control over what each lesson entails as well as the course description, which then allows them to properly label each course preventing any miscommunication. For example, instructors can create two courses labeled "New York English" and "Texan English", which are nearly identical barring the dialect differences between the two, instead of creating a single "English" course. This splits the risks between the developers and instructors, as the developers will have to ensure that instructors have the tools to create these distinctions and the instructors will have to ensure that the courses they design do not create ethics issues. Another preventative measure we can implement is a disclaimer that users will have to agree to before using our application. This does not alleviate the actual issue, but we reduce our responsibility in fostering these issues.

5.3 ASR Accuracy

While the use of ASR technology in our application only makes up a portion of the mobile app, the feedback it provides is critical to the users' learning experience, and any inaccuracies on the part of ASR can hinder that experience. However, we have to largely accept this risk, as ASR technology is never perfect; it is always improving based on the capabilities of our machines. This means that no matter which technology we use, we will never reach perfect accuracy, however, we can at least indicate a level of acceptable inaccuracy to work with. As such, the ASR technology we use should meet this minimum requirement. Furthermore, we can also complement automatic feedback with manual review by instructors. While this does not fix any issues with ASR itself, we can alleviate the issues that inaccuracies cause by correcting any mistakes the system makes. In conclusion, we cannot directly deal with the innate inaccuracies of ASR, but the solutions above will cover through human review the mistakes it may make.

6.0 Project Plan

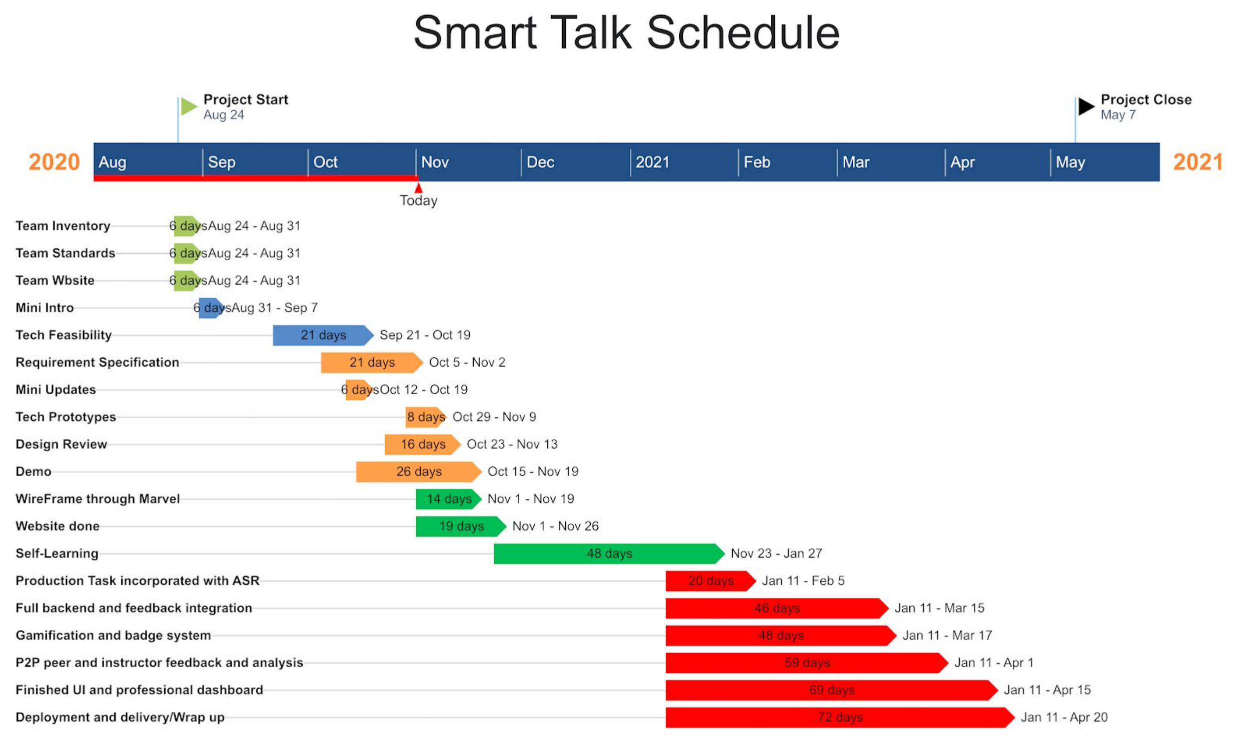


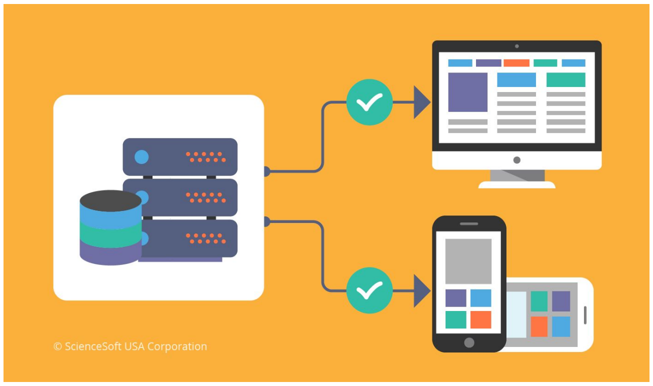
Figure 4 - Gantt chart of the project schedule and milestones.

In the past two months, we have had several meetings with our client, Dr. Kang, to learn more about our project and begin working on our technical demo. For now, we are working on setting up the Firebase and accessing it from the mobile app and web app. For the web, we are trying to make the web app have the function to allow users to create a course in the structure. We are building the mobile app interface and gathering audio samples with Flutter plugins.

By the end of this semester, we will be able to have a demo with some basic functions. During the winter break, we will learn more about Flutter and Flutter Web to make sure all team members can keep the same pace and solve problems together.

For next semester, we will focus on other objectives, which are as follows: Production Task incorporated with ASR(Jan 11 - Feb 5), Full backend and feedback integration(Jan11 - Mar15), Gamification and badge system(Jan 11 - Mar 17), Peer-to-Peer (P2P) learner and instructor feedback and analysis(Jan 11 - Ap r1), Finished UI and professional dashboard(Jan 11 - Apr 15), Deployment and delivery/Wrap up(Jan 11 - Apr 20). Finally, we will have a total test with our app and fix bugs as we conclude the spring 2021 semester.

7.0 Conclusion



With the ever-increasing popularity of language learning apps, we run into the problem of not having an outlet where users can stay engaged and enjoy the learning experience. Teaching methods such as repetition can only go so far in terms of learning a new language, where we feel as if creating a gamified mobile app can create an enjoyable experience that can be just as memorable as it is educational. While users spend tons of time on language apps struggling with the repetition of the lessons, they don't usually have a way in which they can make the time they are studying more enjoyable. We take a look at the language apps out there today and we see room to make something that captivates users. We aim to make a change by creating an app with web integration that allows users to receive lessons from instructors. This allows for greater interaction for language learners and provides a way for users to receive feedback on their current work.

Through this document, we have laid out an outline for which all the core parts of this project will be completed. We have laid out a tentative schedule for which we feel we can adhere to in order to produce a product that is both satisfactory for our client and to our own personal aspirations. We have also made note of all potential risks that can befall the project in the creation process. We also layout some potential ideas and functionalities of the app through the use of several graphs and sections of discussion. While at the moment we are still working on the details of the project we feel as if the constant client meetings and the amount of research done up to this moment have put us in a position to not only create this product but create this product well. We hope to capitalize on the progress we have made and use it to set a precedent for the tone of the rest of the project.