Phonetic Evolution

9/29/2023

SOFTWARE DESIGN

Kenzie Norris
Krystian Bednarz
Sam Asher
Preston Lee

Version 1.1

Sponsor: Dr. Okim Kang
Mentor: Vahid F.
Organizer: Michael Leverington

Table of Contents

1. Introduction

The United States is the highest-ranked country for students looking to study abroad. Academics travel from around the globe to seek out its institutions and universities in hopes of furthering their education and experiencing new cultures. Although the United States has no official language at the federal level, the most commonly spoken language is English. Anyone pursuing an education in the country will necessarily have to obtain a strong understanding of the language. English is notoriously difficult to learn if you are not a native speaker. Research has shown that it is easiest to learn a language before the age of 10, as children's brains are much more adaptable and flexible, and therefore better suited to picking up crucial language learning patterns at that stage in their lives. While learning a new language later in life is a much more challenging task, it is a highly rewarding endeavor that millions around the world undertake every day. Hearing a language in its natural context provides some very distinct advantages, such as improved comprehension, enhanced pronunciation abilities, better accuracy at identifying accents, and increased confidence in actually speaking the language.

Our project will be a website application that consists of various learning tools and games that aim to teach contextual awareness, as well as a mastery of pronunciation with words in the English language as a whole. To facilitate this learning process, our website will have multiple gaming elements specifically geared towards these specific language learning areas. An expansive language library database with a context search filter will aid users in understanding how words are used in specific circumstances, and expose them to the intricacies of a language as it is spoken in day-to-day conversation. Open-world gaming elements are present in our project scope as well, where interactable objects can be referenced and stored in a user's own personal library for future reference.

Our clients, Okim Kang and Kevin Hirshi, are both a part of the NAU Department of English. For years they have been working with different teams of developers to develop new tools and applications to help aid transfer students adjust and master the English language. They want to help students transferring to the US learn the language faster, with the use of context, a key element that is missing from modern-day language learning applications. On top of that, they want to help transfer students master the proper pronunciation, another thing missing from those same language learning sites. Our goal is to take the years of research Kevin and Okim have compiled, and use it to create a gamified program that focuses on more than just vocabulary and grammar learning; teaching students context and helping them master their pronunciation.

Through a lengthy requirements specification process, we have narrowed down the key features and functionalities that our website needs in order to best achieve our goal. These requirements include the library search, user and administrator management functions, user interface visuals, and the game and practice pages themselves. Beyond functional aspects of the project, we have determined testable means to gauge how well the site runs as a whole through our performance requirements. How long the library searching algorithm takes to output a result, how many details are provided within each user's account information, how quickly backend database changes are propagated to the frontend site, how fast visual page changes occur, and how long it takes to load the games on the site are all aspects that can be measured and optimized. Ensuring the site works at a reasonable pace is necessary to minimize wait times and maximize learning, ensuring our site is as efficient as possible.

There are certain unchangeable environmental restrictions that can impede site performance that too must be accounted for. Such environmental requirements include Gmail authentication being problematic for students who reside in China, and a different means of authentication must be used as a result. Our backend code used for database management must also be housed within the Amazon AWS instance we are provided. Other environmental facets include what programming languages the site must run on, and what supporting languages we can use alongside it.

Now that the project requirements are set, and what the website needs to accomplish the goals of the project are identified, we must determine what structure each facet of the project needs to adhere to. How modules interact with each other, what data must be sent, where that data needs to go, and how each component of the project is designed must be identified to ensure smooth project completion.

2. Implementation Overview

Our project is a gamified language data explorer that will help students with pronunciation through analysis of data, or PANDA for short. It is a browser-based web application that is connected to a large audio transcription database provided by our sponsors. The database contains segments of speech from native speakers of a language, taken from a variety of sources, as well as an accompanying text transcript of the audio.

Our application will contain a robust and efficient search feature capable of accurately retrieving relevant results from the database based on the user's entry and displaying them in a user-friendly and organized fashion. This is a key feature of our application and will provide a lot of utility to all users. This system will be primarily used for research purposes and will provide a useful tool for allowing our clients to further test their language learning theories. Such theories involve learning word definitions through the context of where those words are present in a sentence. This will be a strong foundation for an application that they can add to and improve upon in the future, as they come up with new features that they feel would benefit users.

The project will consist of three main components and each relies upon the others to allow the software to function properly. Our main objective will be to tie our backend database with a newly implemented frontend website, allowing the end user to manipulate the database to fit their learning needs. More specifically, students who use the site can change certain areas of information in the database through the website, such as by creating a user account or posting performance statistics in our games on the site. The backend component consists of an EC2 Amazon AWS instance that hosts a database coded in MySQL. Amazon AWS is a database hosting platform that holds all the data gathered from the website. MySQL is a programming language for a database that our team has determined to be best suited for data collection and data acquisition. The website will run on HTML, the standard programming language for websites, with supporting files being coded in CSS, which are HTML based files that aid in more visual aspects of the site (font color, shape creation, menu look, etc.) Communication with the database is facilitated through PHP, a branch of HTML with additional connectivity features necessary for data transfer with the database and the website. The game applications within the website will be coded in Javascript, a programming language better suited for the creation of gamified elements, such as in browser games.

3. Architectural Overview

Prior investigation of what requirements are necessary for the completion of this project aligns with the overall structure of the project itself. **Figure 1** better portrays this topic shown below, where each component is listed, and more specific attributes of each component are displayed.

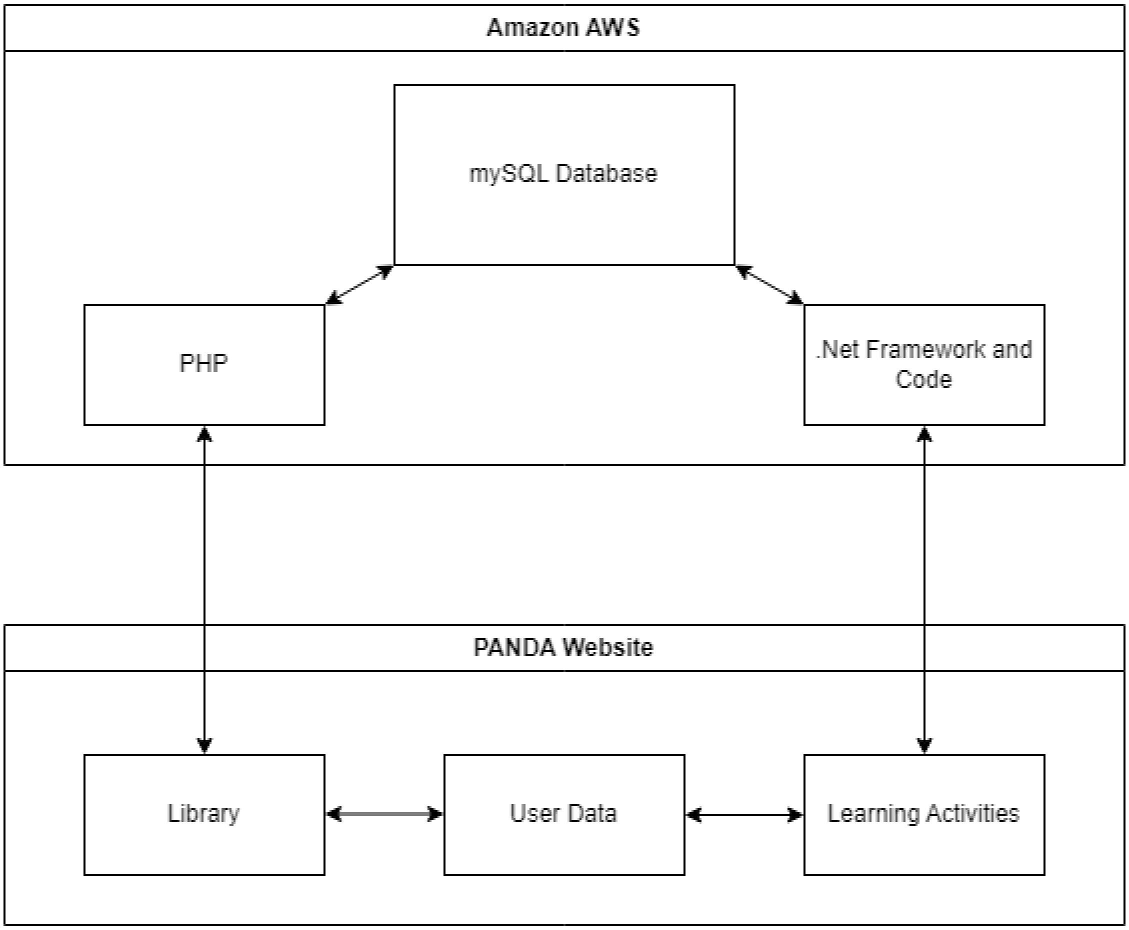


Figure 1: The Architectural Functionality of the Panda Website

Our library search functionality allows users to look up words, manipulate a context value within the search, and listen to chosen words received from the words found after searching. This works by examining the given word, and comparing that term to all words present within our database while referencing the input context value. The search is present within the site and communicates with the database system to acquire words for display to the user. Each word is connected to an audio prompt that users can click on to hear the result. This method of word comparison to data present in the database is standard, differing where the presence of audio prompts alongside the words is present within the search functionality.

The user account system on our site is used to hold user information pertaining to progression in the learning course, daily missions, login credentials, and any information relevant for each user per user. Users are able to create their own accounts on the website, where completed user accounts are then transferred to the database for user data storage. This database is the same system that hosts audio prompt information for the library search system. Administrative functionality works similarly to the user account system, differing where site admins have the ability to actively manipulate data present within the database system. Site admins can add, remove, and edit data present within the database such as user groupings, game additions, and visual site changes. The style of the account system is unique to our project, as the data we choose to save is specific to what the site needs to function the way we desire. More common practices of password encryption and the overall concept of user data storage are in reference to user storage systems already present on other sites.

The games on the site are housed within the practice page. The games on the practice page include the pronunciation, fill in the blank, and context games. The main function of the games is to aid in user language learning and is the major component that teaches the user overall. Performance on games on the site is stored in user data, specific for each user and stored within the database. The games themselves are run on the site, where the performance data the user made through playing the game is transferred to the database. Similar examples of gamified elements were referenced from current language learning software, such as Duolingo. The major difference in architecture pertains to how learning occurs, where the presence of contextual mechanics and game elements are more present within our project than in other examples we have observed.

4. Module and Interface Descriptions

The two major components required for our project to function are the website and the database systems. The website is the major component for user interaction, as the site is the delivery system for our language learning process. The database serves as a data housing system for the site to access when prompted by the user. How the website and database function to provide users with their language learning journey is outlined here.

4.1: Website

The website serves as the direct language learning component in our project. Users first create their own accounts on the site, where the information they provide is stored in the database for later use (user login credentials, for example.) Additional user information will continue to populate the database throughout the usage of the site, such as game performance statistics. Administrative functionality allows for specific users to manipulate the content present on the site, as well as assign activities to grouped non-admin users. Gamified elements on the site consist of daily activities and games available on the practice page. Daily activities and missions are assigned through administrators on the site, and can be changed to better suit teaching preferences for each user group's associated administrator. The practice page holds the pronunciation game, the fill in the blank game, and the context game. User performance from participating in these games are first recorded into the database system and are then sent for display when accessing the user account page.

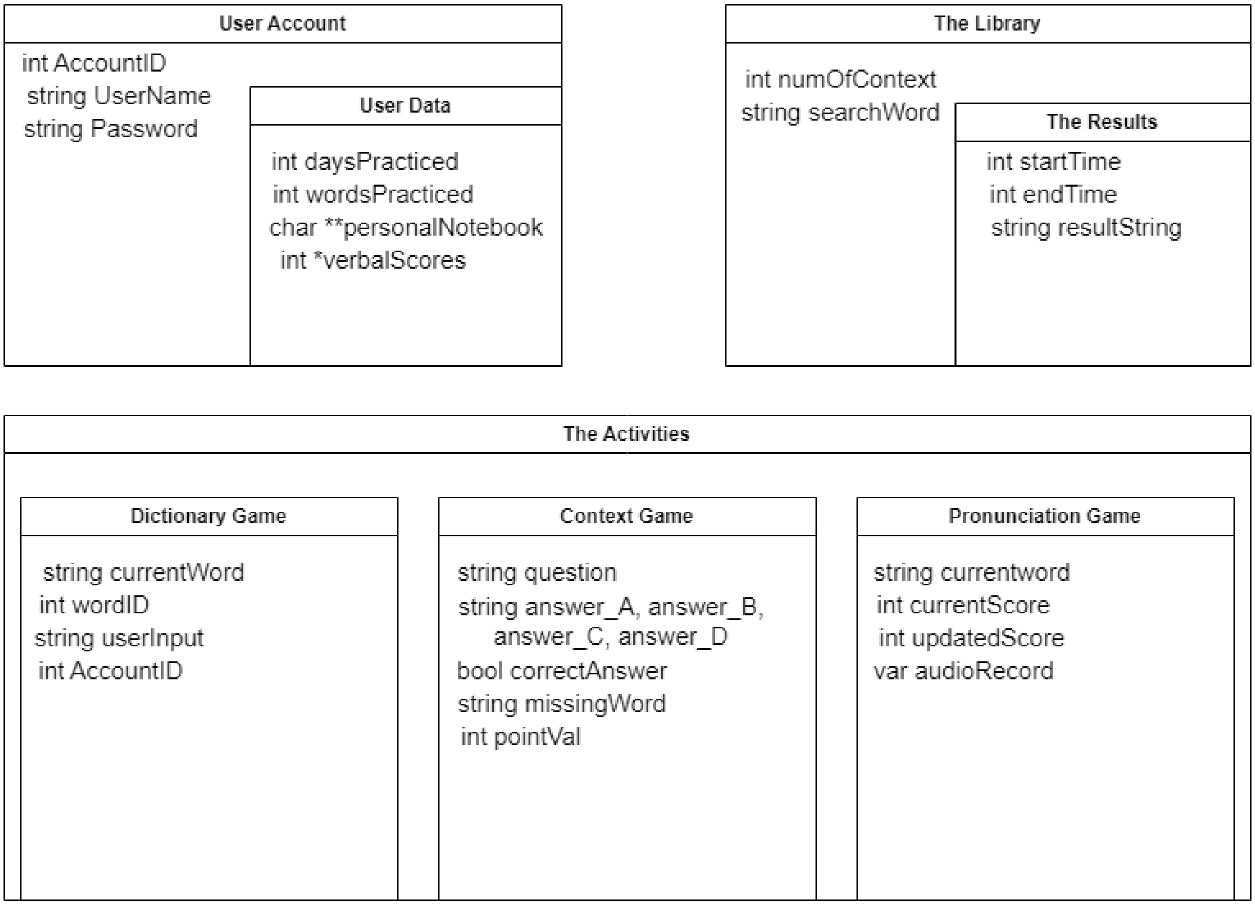


Figure 2: Website architecture diagram

4.1.1: User Account System

The user account module is responsible for maintaining data related to the users of the website. We will need to store account information for anyone who makes an account. The data that is stored includes their username, password, and the user's identification number. Due to environmental restraints we are unable to use existing sign-in authentication tools, so we will need to encrypt passwords to ensure user safety. User accounts are also linked to each user's unique data, which keeps a record of important metrics that are tracked while using the site. This data is stored in the backend database and will be covered more in-depth in the database section. We want to allow the users to view the number of days and words they have practiced, as well as their scores for the games they play. This information will be available to view on their user page. Also, there will be a personal notebook on their user page on their user page, which contains references to sentence and word IDs within the database, and allows the user to access words that they have starred while utilizing the library search function. User-created definitions are also added to the personal notebook.

4.1.2: Library System

The library page is the main tool of our website and will give the user the power to explore the database. This process involves the user navigating to the library page on the site, and prompting the search system with a word. All words housed within the database align with a predetermined identification value (or ID number.) The search function works by pulling up all word IDs that match the query word using the MySQL search function. This search function is already implemented within the MySQL system we are using, so this process is as simple as using the feature within the system using the term the user provides as input. From those search results, we add and subtract the context value from the current index to get all of the surrounding words, as well as audio times to link up with the text. Manipulating the context value directly changes the search, so the terms that are output align with more or less specific words given the user's context value as an input. Audio prompts are then displayed alongside the returned words, providing the results for the user to select upon completion of the search function.

4.1.3: Activities

The practice page on the website houses the three games users can choose to play. User performance within these games is sent to the database to access in the user account system. The daily activity system works by selecting one of these three games at random, or are selected directly via administrative controls.

A. **Dictionary Game**
   Users are prompted with a word and are tasked with defining that word. The description the user provides is sent to the user's notebook for later review. Personal notebook entries are stored in the database and are associated with a user ID. Users can continue to define words to populate the notebook for each word they describe.

B. **Context Game (Fill in the Blank)**
   Users are given a word and are tasked to determine in which sentence the word is best used in a specific context. Four multiple-choice options are provided for the user to select from, where each sentence contains an empty slot for the prompted word to fill in.

C. **Pronunciation Game**
   The user is prompted with a word in some language and is instructed to speak that word into their microphone. Storage of the user's audio is unnecessary here, as user audio will be discarded immediately after, but users will still have the opportunity to compare their audio with the audio for the term they were prompted to speak. Users will grade themselves based on a scale of 0 to 100 percent.

4.2: Database

The database houses both the language audio and transcript files, as well as user data. User data held in the database consists of account information and performance statistics.

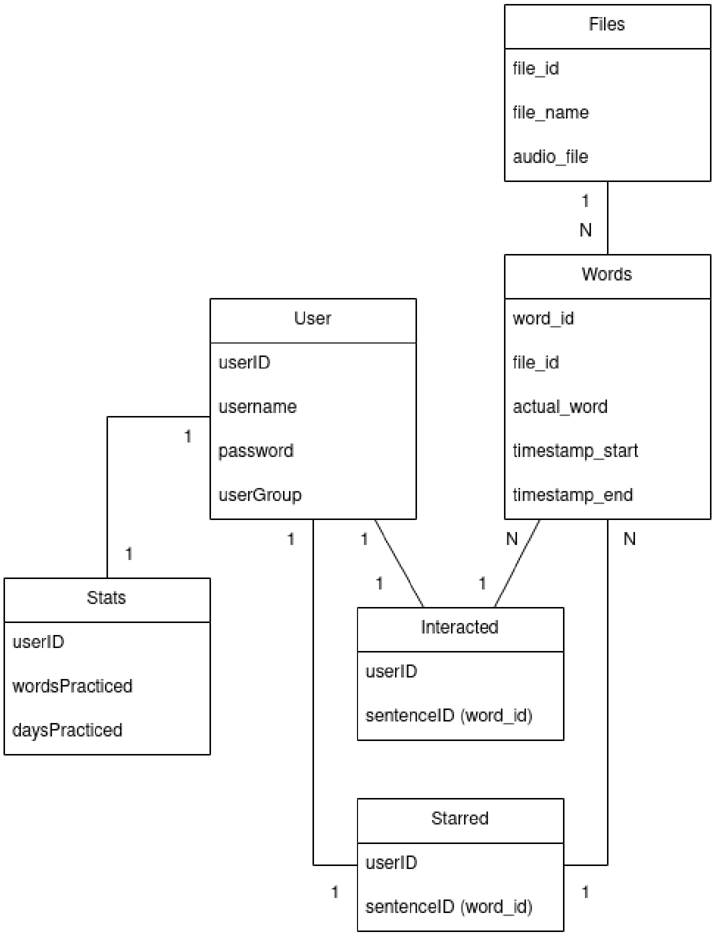


Figure 3: Database architecture diagram

The database organizes both the users and words contained in all the audio files, as well as their relations to each other. Words are referenced from their respective files based on file ID primary keys. Each word has attributes that describe its beginning and ending timestamp within the audio recording of its associated file. This allows for contextual searching on the library page and for accurate playback. Users associate themselves with words through their interaction with them and can add stars to terms within the library search. This way each user is able to save words that they want to come back to later in a personal notebook.

Interacted and starred words are collected in the personal notebook connected with each user account. Also associated with every user entry are personal stats for the games that they complete and their usage metrics. Initially, we plan to only store the number of words practiced and days practiced, but this could easily be expanded to record other useful statistics. This information is then available to the user on their user page, and in the future, we would like to create visualizations to accompany this data, to give the user a more in-depth look at their progress over time. Storage and retrieval within the database can be conducted using PHP from the website and will serve as our main link to the database.

5. Implementation Plan

Figure 4 below displays our current planned implementation schedule for the months of September through late November. It shows the priority order of each component of the project as well as who is taking the lead role in completing that section. We plan to have our largest development milestones completed by the end of October, in order to have a robust alpha demo that we can get more accurate feedback on and have a better idea of where to focus our attention for further improvement and fixes.

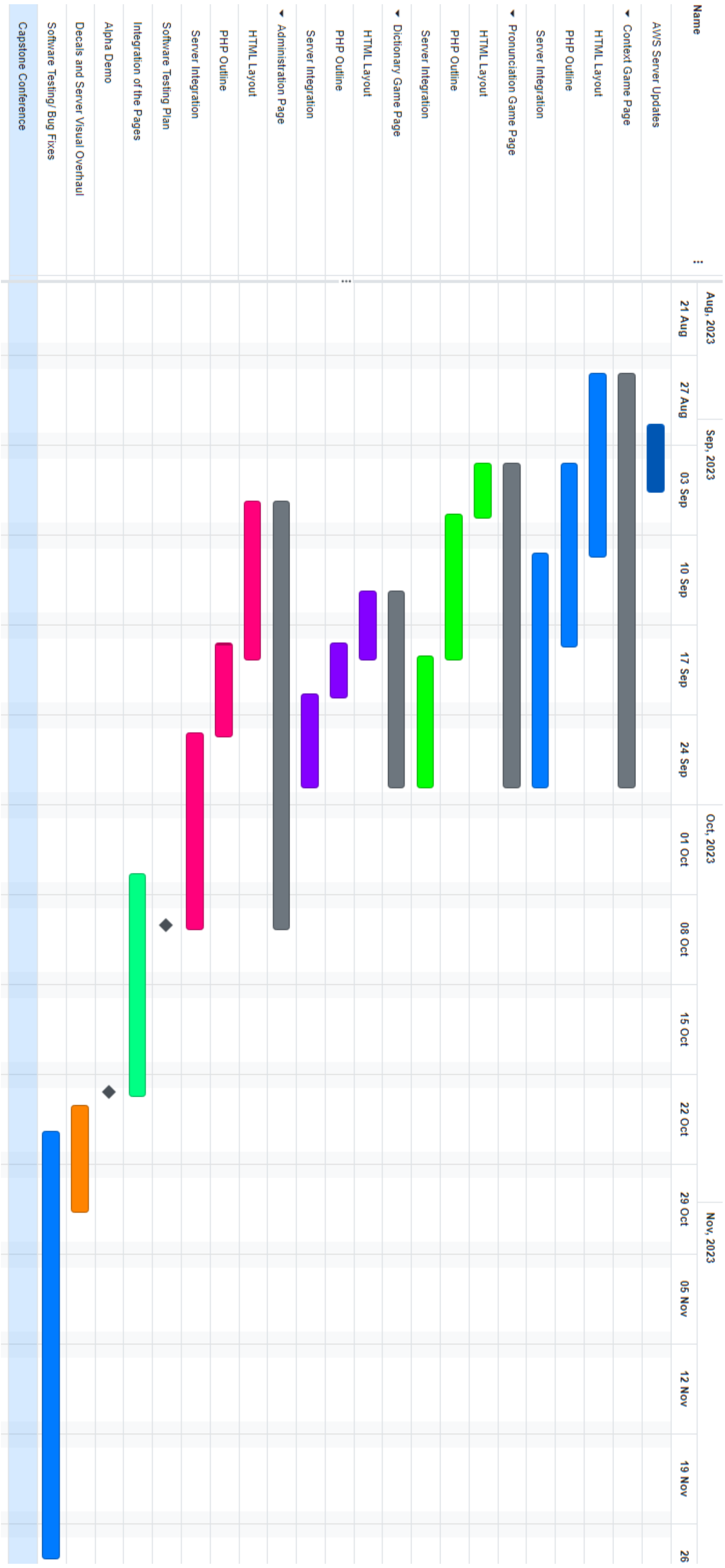
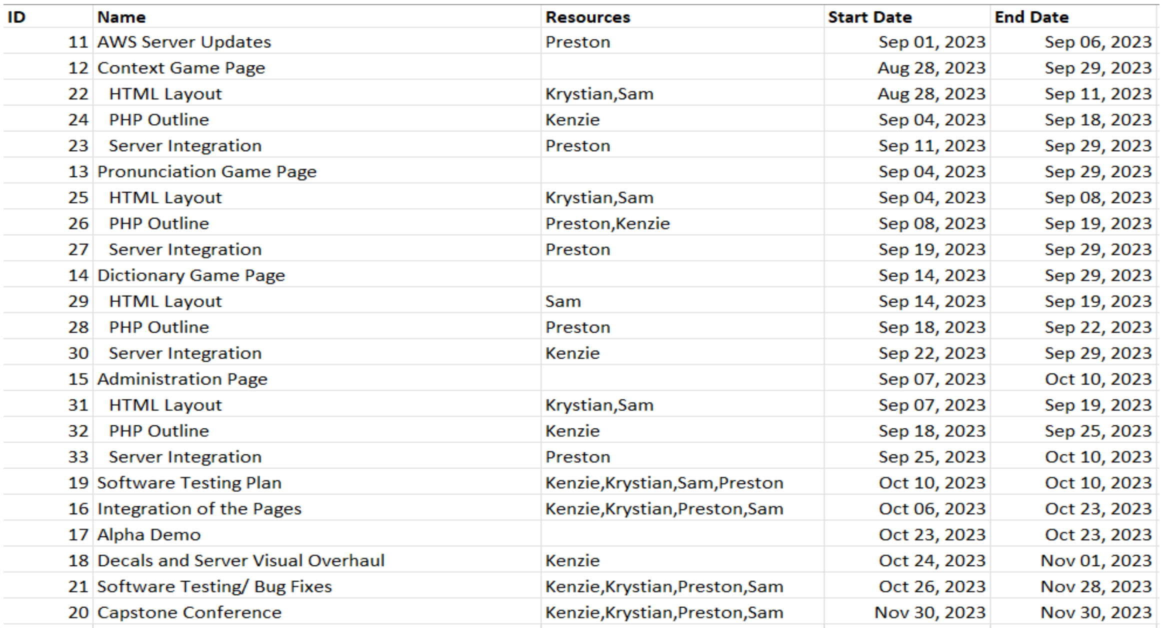


Figure 4: PANDA Software Development Gantt chart

The library search page has been mostly completed over the summer, and we will continue to make edits as we add the other pages, as it will need to be integrated. Our initial goal will be to complete each of the game pages individually, and then near the end of October, we will integrate each one so that they all fit together into one package. We believe this will give us a good chance to safely test and modify each page individually before adding them to the website. We will be working with parallelism to accomplish all of this, with the starts and ends of many of the tasks overlapping with one another. This works by rotating the responsibilities of team members, so that once the initial tasks of a component have been completed, they can be handed off to the next person, and work can begin on the upcoming batch. Sam and Krystian have been delegated the majority of the HTML work and getting the web pages set up, while Kenzie and Preston will be working mainly on the backend, and getting the server and PHP functionality working. Once we've completed development of the individual games and pages we can then begin to add visuals to make the site look better, as well as do software testing and bug fixes related to the site as a whole. **Figure 5** below gives a more detailed breakdown of the assignments of each team member and our prospective start and end dates.



ID	Name	Resources	Start Date	End Date
11	AWS Server Updates	Preston	Sep 01, 2023	Sep 06, 2023
12	Context Game Page		Aug 28, 2023	Sep 29, 2023
22	HTML Layout	Krystian,Sam	Aug 28, 2023	Sep 11, 2023
24	PHP Outline	Kenzie	Sep 04, 2023	Sep 18, 2023
23	Server Integration	Preston	Sep 11, 2023	Sep 29, 2023
13	Pronunciation Game Page		Sep 04, 2023	Sep 29, 2023
25	HTML Layout	Krystian,Sam	Sep 04, 2023	Sep 08, 2023
26	PHP Outline	Preston,Kenzie	Sep 08, 2023	Sep 19, 2023
27	Server Integration	Preston	Sep 19, 2023	Sep 29, 2023
14	Dictionary Game Page		Sep 14, 2023	Sep 29, 2023
29	HTML Layout	Sam	Sep 14, 2023	Sep 19, 2023
28	PHP Outline	Preston	Sep 18, 2023	Sep 22, 2023
30	Server Integration	Kenzie	Sep 22, 2023	Sep 29, 2023
15	Administration Page		Sep 07, 2023	Oct 10, 2023
31	HTML Layout	Krystian,Sam	Sep 07, 2023	Sep 19, 2023
32	PHP Outline	Kenzie	Sep 18, 2023	Sep 25, 2023
33	Server Integration	Preston	Sep 25, 2023	Oct 10, 2023
19	Software Testing Plan	Kenzie,Krystian,Sam,Preston	Oct 10, 2023	Oct 10, 2023
16	Integration of the Pages	Kenzie,Krystian,Preston,Sam	Oct 06, 2023	Oct 23, 2023
17	Alpha Demo		Oct 23, 2023	Oct 23, 2023
18	Decals and Server Visual Overhaul	Kenzie	Oct 24, 2023	Nov 01, 2023
21	Software Testing/ Bug Fixes	Kenzie,Krystian,Preston,Sam	Oct 26, 2023	Nov 28, 2023
20	Capstone Conference	Kenzie,Krystian,Preston,Sam	Nov 30, 2023	Nov 30, 2023

Figure 5: Division of Responsibilities

Our development cycle can be subdivided into four main phases: page development, integration, software testing and bug fixes, and visuals. While many of these phases will be occurring at or near the same time, the bulk of the page development must be completed first, which will be followed by integration, and software testing.

Visuals are last, as it doesn't matter how good the website looks if it doesn't function correctly.

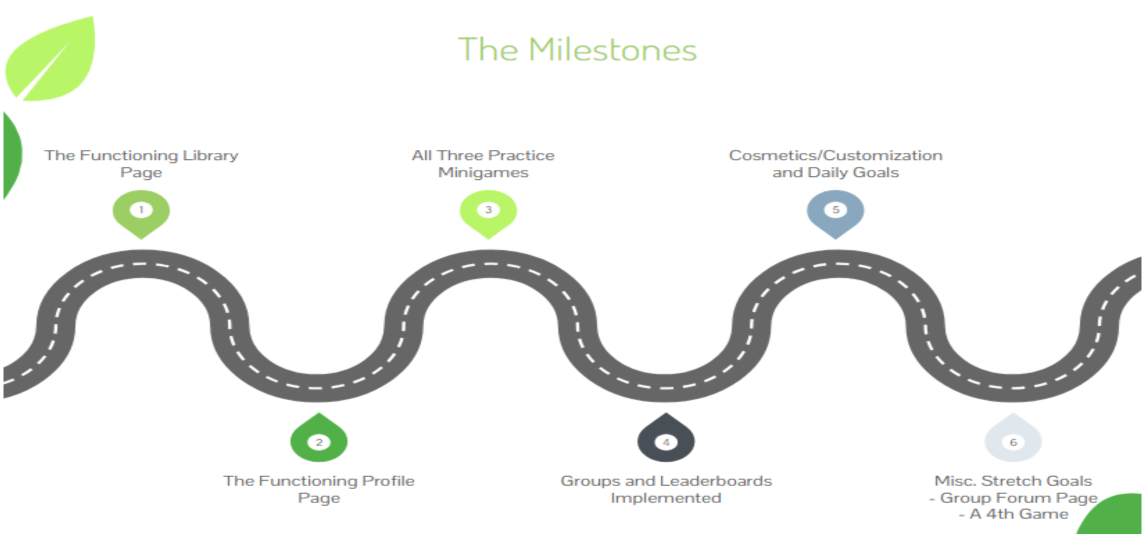

The Milestones
The Functioning Library Page
1
All Three Practice Minigames
3
Cosmetics/Customization and Daily Goals
5
2
The Functioning Profile Page
4
Groups and Leaderboards Implemented
6
Misc. Stretch Goals
- Group Forum Page
- A 4th Game

Figure 6: Milestones roadmap

Also included in the chart are important milestones, to give us a better reference of what we have to look forward to, and to help us track our overall progress. Above is a roadmap of what we deem to be the most important landmarks of our development journey and the rough order in which we would like to complete them. If we do have extra time at the end of our four phases then we will devote our time to implementing some features from our stretch goal wish list.

6. Conclusion

Our software aims to better the weaknesses of other existing language learning tools by focusing on elements such as pronunciation, and context. PANDA will be a tool that can help learners to better immerse themselves into a language, and give them the ability to practice speaking a language and listening to it in a more natural and intuitive way. Furthermore, we want to implement features that reward users for continued engagement and motivate them to continue their studies. The language learning process can be long and difficult, and we hope that our software will make the experience easier and more enjoyable for our users. Through the implementation of gamified elements, language library searching features, and audio prompting, we hope that our method of language learning will give students a different perspective on their language learning journey.

We can achieve our mission with the bipartite server and website model that is described in this document, and by following our carefully planned schedule to ensure that we meet our goals on time and stay on track. We hope that this document serves as valuable insight into our project for any future developers. We are making good progress toward our end goal, and we believe that we will have a great product to submit to our clients by the end of our development process.