

Project Title:

Auditory Training App

Team Members:

Capstone 2: Danny Pascual, Samuel Perez, and Norberto Merani

Product Owner(s): Alliete Alfano

Mentor(s): Masoud Sadjadi

Abstract

The project documentation will introduce users to all features, architecture, system, development, and future plans pertaining to the Auditory Training App. The Auditory Training App is a web application that hosts a variety of exercises for cochlear implant patients. These hearing loss patients who undergo cochlear implant surgery must spend months in aural rehabilitation in order to strengthen their lingual comprehension and adapt to the new sound signals. The purpose of the Auditory Training App is primarily to aid users in their rehabilitation process by providing hearing drills, games, and activities that scale in complexity and difficulty as they strengthen their listening skills. It also seeks to provide a platform for individuals to connect with others in the community by sharing their progress and hearloss journey as they continue to work through the various activities.

Table of Contents

Abstract.....	2
TableofContents.....	2
Introduction.....	4
PurposeofSystem.....	6
UserStories.....	7
Implemented User Stories	7
Pending user stories	9
Project Plan	10
ProjectGoals.....	10
Project Personnel Organization.....	10
Hardware and Software Resources	11
Sprint Plan	12
Sprint 1.....	12
Sprint 2.....	12
Sprint 3.....	13
Sprint 4.....	13
Sprint 5.....	14
Sprint 6.....	12
Sprint 7.....	12
System Design	15
Architectural Patterns	15
UML Diagram - Class Diagram.....	16
Validation.....	17
Persistent Data Management.....	17
Appendix.....	19
Appendix A – UML Diagrams	19
Appendix B – User Interface.....	20
Appendix C – Sprint Retrospective Reports.....	24
Sprint 1.....	24
Sprint 2.....	25
Sprint 3.....	26
Sprint 4.....	26
Sprint 5.....	27
Sprint 6.....	22
Sprint 7.....	22
Appendix D – User Manuals, Installation/Maintenance Document,Wishlist.....	28
User Manual.....	28
README.....	23
Installation Guide	36

Wishlist	37
----------------	----

Introduction

The Auditory Training App, is a web application designed to provide a centralized support and language practice hub for patients who have undergone cochlear implant surgery to treat hearing loss. While hearing aids work by amplifying sound, cochlear implants convert sounds to electronic signals, which initially cannot be properly processed by the patients' brains. To be utilized effectively, patients must undergo aural rehabilitation to adapt to the new auditory implants, which bypass damaged parts of the ear to send signals to the brain. Though there are currently multiple listening games and exercises designed to meet this need, they all fail to effectively train patients from beginning to end. This is due to a lack of engaging gameplay and meaningful progression. The Auditory Training App seeks to create an engaging and effective application to train cochlear implant patients through fun games and an integrated social network. Led by Dr. Alliete Alfano, the Auditory Training App is currently in its fifth semester of development.

The Auditory Training App, features a variety of currently built-out and in-development features to provide an engaging system for practicing linguistic skills. This includes drill and game activities designed for each linguistic category (comprehension, detection, identification, and discrimination), a public and modifiable user profile screen, a forum page to share progress and hearing loss stories, a settings page to modify account settings, a reports page to view user activity, and a support page to request assistance.

Purpose of System

The goal behind the Auditory Training App is to provide a platform for patients that have undergone cochlear implant surgery to help them practice receiving and understanding the electrical signals outputted by the cochlear implant through various hearing activities. Each activity should increase with difficulty as the user progresses. Different activity types (detection, discrimination, identification, and comprehension) allow patients to learn step by step and practice as many times as they would like before moving onto more complex activities. The platform also facilitates the development of a hearing loss support network via the forum, where users can chat with one another to share their experiences, tips they've discovered, and their progress.

User Stories

Implemented User Stories

Sprint	User Story
1	Get familiar with existing code and environments
1	Create Preliminary Mock-Ups to update UI design for application
1	After approval from PO, begin updating UI Icons and colors
2	Create Mock-Ups for Forum Pages
2	Create Mock-Ups for Navigation Bar
2	Change Level Indicator to number value
2	Create Mock-Up Icons for lingActivity Game cards
2	Edit colors for Detection Activity Icons
2	Add description and titles to Detection Activity Cards
2	Fix Invalid API Key Error—3/5 members cannot access webapp
3	Center Align Icons and Text on Activities Pages
3	Update icons within lingActivity game cards
3	Add GamePageDetection, GamePageDistrimination, GamePageIdentification

Sprint	User Story
3	Begin to update forum page UI
3	Add directions to lingActivity games
3	Fix sound issue in detection & discrimination games
3	Add inBetweenPage to choose between drill and game activities
4	Update navigation bar UI and functionality
4	Modify position of directions for each game
4	Updated forum page UI
4	Create Settings page mock-up
4	Create Support page mock-up
4	Create Reports page mock-up
4	Make Export button in Reports page functional
5	Fix any outstanding UI alignment issues
5	Debug and implement Recent Activities functionality
5	Modify Reports page UI
5	Add Pagination to Reports page

Sprint	User Story
5	Make titles consistent throughout app
5	Add Support Page
5	Finalize Forum page Update
5	Finalize Settings page update
5	Make all textboxes consistent with settings page mock-up textboxes

Pending user stories

Sprint	User Story
5	Fix level updater for activities
5	Finish implementing forum threads
5	Finish implementing messaging system on forum
5	Finish implementing support page 'submit' button
5	Update profile photo throughout app
5	Add comprehension lingActivity

Project Plan

Project Goals

This semester, our goal was to add more drill activities, add a voice selection feature, and add games to the application. To meet our first and last goal, all of the frontend components were updated to give the app more pages and functionality. Before, the application lacked pages that were needed for the new goals. Additionally, there were no distinguishing icons for the different activity types or indicative visuals or instructions within the lingActivity and lingGame pages. To address the second goal, a settings icon was added to the ling activities. Both of these goals worked together in tandem to create a complete product. Since all of these changes have been made, future semesters can focus on adding more features such as deployment and a community feature.

Project Personnel Organization

Each Capstone 2 team member was assigned certain features to work on this semester. All team members enhanced their front and back end development skills throughout the semester. Team members were encouraged to work alongside and support each other, as consistency in both the code and the UI was key to accomplish our project goals.

Capstone 2 Member	Assigned Feature	Main Items in Feature
Danny Pascual	Game Options Page Matching Game Levels Page Four Matching Game Levels Word Search Game Levels Page Two Word Search Game Levels	Update UI for Game Options pages Add Game Level Pages for both Matching and Word Search Update game page & internal game cards Add icons for new games
Samuel Perez	Settings Page Game Icon Mock-Ups	Update UI for Settings Page Create Icon Mock-ups for lingActivity cards

Norberto Merani	Support Page Mock ups	Create UI and backend to develop Support page Begin working on making submit button functional
-----------------	------------------------------	---

Hardware and Software Resources

Developers will require a computer that runs non-mobile versions of web browsers as there is no mobile application in development at this point. Programs used in development include:

1. IDE of choice (Visual Studio Code is recommended)
2. Git and Github
3. JavaScript using the React framework
4. the Node.js package framework
5. Google Firebase serving as our database and backend.

Sprint Plan

Sprint 1

	Story Title	Story Points	Assigned To
--	-------------	--------------	-------------

1	Get familiar with existing code and environments	5	Everyone
2	Create Preliminary Mock-Ups to update UI design for application	5	Everyone
3	After approval from PO, begin updating UI pages and icons	5	Everyone

Sprint 2

	Story Title	Story Points	Assigned To
1	Create Mock-Ups for Game Pages	5	Danny Pascual
2	Create Mock-Ups for Drill activities	5	Samuel Perez
3	Create Mock-Ups for Community Page	5	Norberto Merani
4	Add Icons for lingActivity and lingGames	3	Everyone
4	Add description and titles to drill activity Cards	4	Samuel Perez

12

5	Edit colors for Drill Activity Icons	3	Norberto Merani
6	Fix .map error with CardDB	8	Everyone

Sprint 3

	Story Title	Story Points	Assigned To
1	Center Align Icons and Text on Game Pages	5	Danny Pascual
2	Update icons within lingGame game cards	6	Danny Pascual
3	Add GamePageMatching and GamePageWordSearch	7	Danny Pascual
4	Begin to update forum page UI	8	Norberto Merani
5	Add directions to drill activities	2	Samuel Perez
6	Fix sound issue in drill activity	8	Samuel Perez Norberto Merani
6	Add GameOptionPage to choose between word search and matching	6	Danny Pascual

Sprint 4

	Story Title	Story Points	Assigned To
1	Create setting icon UI and functionality	10	Everyone
2	Modify position of directions for each game	6	Danny Pascual

3	Created setting sidebar UI	9	Samuel Perez
4	Create Settings sidebar mock-up	5	Samuel Perez
5	Create Matching page mock-up	5	Danny Pascual
6	Create Word Search page mock-up	6	Danny Pascual
7	Fixed bugs in forum page	10	Norberto Merani

Sprint 5

	Story Title	Story Points	Assigned To
1	Create more drill activities	5	Samuel Perez
2	Create levels for matching game	10	Danny Pascual
3	Modify Home page UI	8	Norberto Merani
4	Record sounds and upload to firebase	10	Everyone

Sprint 6

	Story Title	Story Points	Assigned To
1	Import voices to matching and word search games	8	Danny Pascual
2	Import voices to drill activities	8	Samuel Perez
3	Record and upload more voices to firebase	5	Norberto Merani
4	Debug application	10	Everyone

Sprint 7

	Story Title	Story Points	Assigned To
1	Debug and finalize all changes in application	10	Everyone

System Design

The purpose of this document section is to outline the key design principles driving the application. Creating a successful product necessitates a well-defined plan. Consequently,

comprehending the fundamental principles governing the code structure and the reasons behind the team's decision to adopt this design is of utmost importance.

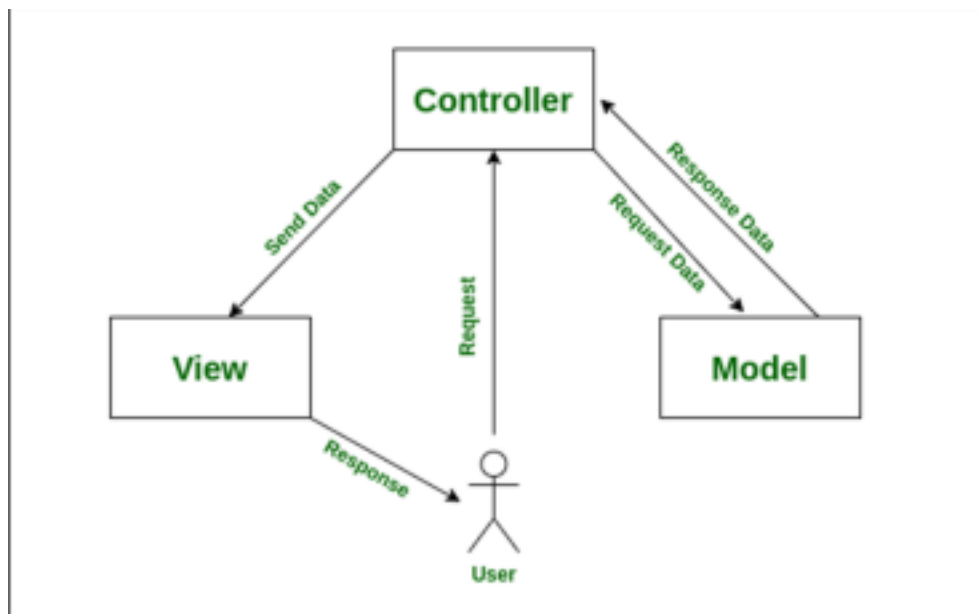
Architectural Patterns

In order to maintain consistency within the application, we chose to follow the model-view controller (MVC) structure for test-driven development. The effectiveness of MVC becomes evident when creating a modular application. Given our collaborative development approach, prioritizing a design that is modular-friendly was crucial to ensure a smooth workflow. Hence, opting for an architecture that supports modular applications and preserves their modular nature seemed like a sensible and logical choice.

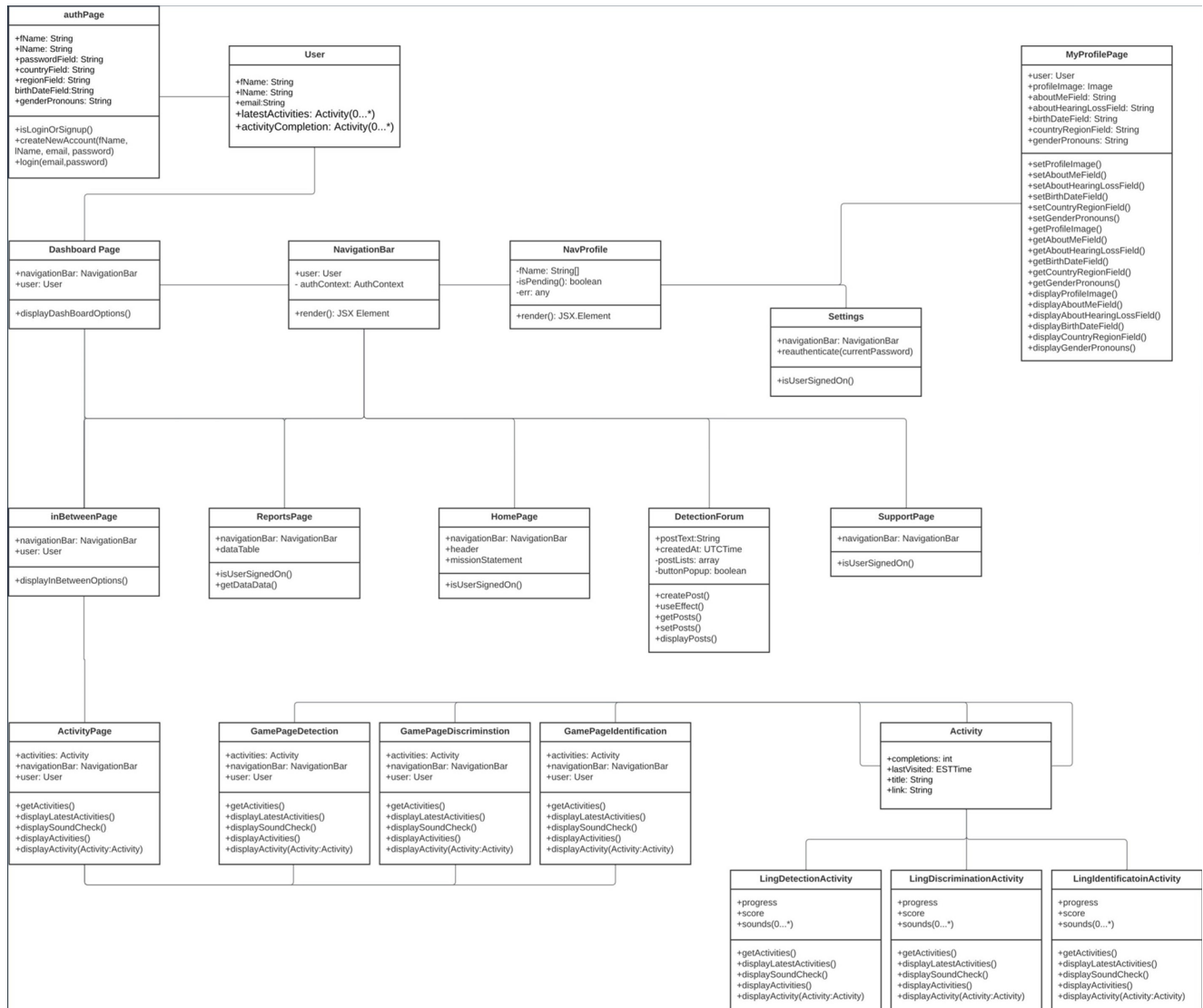
The specifics of the MVC architecture pattern are as follows:

1. Model --> Code logic used to extract information from the database to the controller module.
2. Controller --> Code logic that presents the information in the view module to the viewer, and takes the user's input to request data from the model module.
3. View --> Code logic composed of the tools the user can use to request and receive data from the controller.

Below is a visual representation of the MVC structure:



UML Diagram - Class Diagram



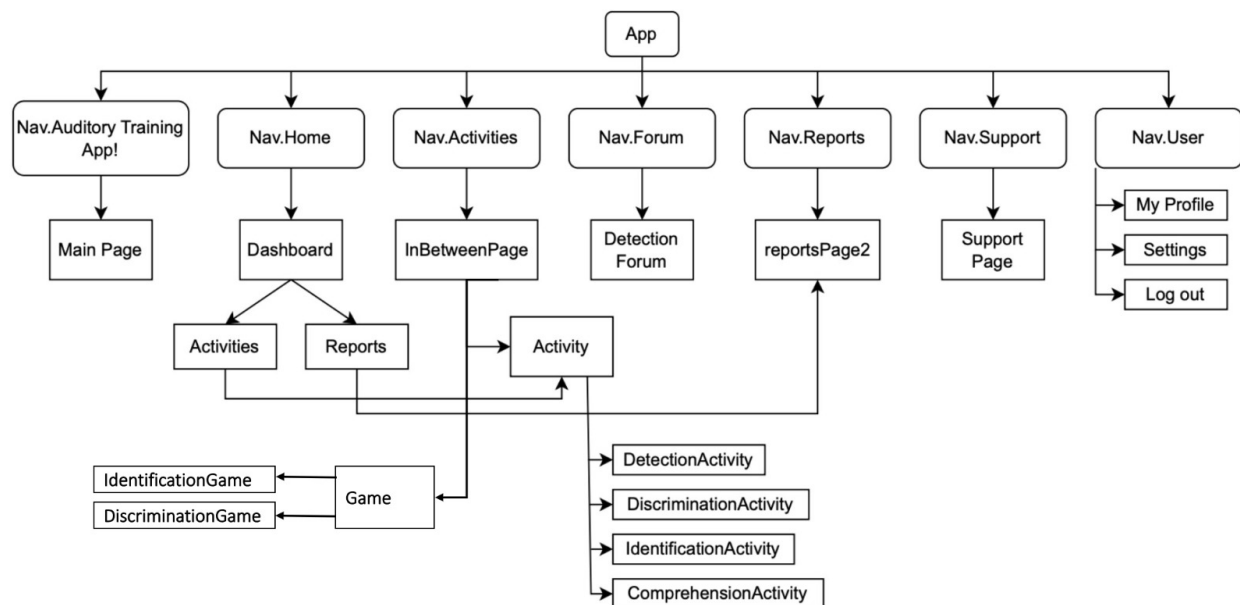
Validation

Validation was done manually by all team members at the end of each sprint to ensure no bugs were introduced and no functionality was modified unintentionally. At the end of the last sprint, all team members tested the entire application, ensuring all games were working as intended, all popups were appearing properly, and all functions worked as intended,

Persistent Data Management

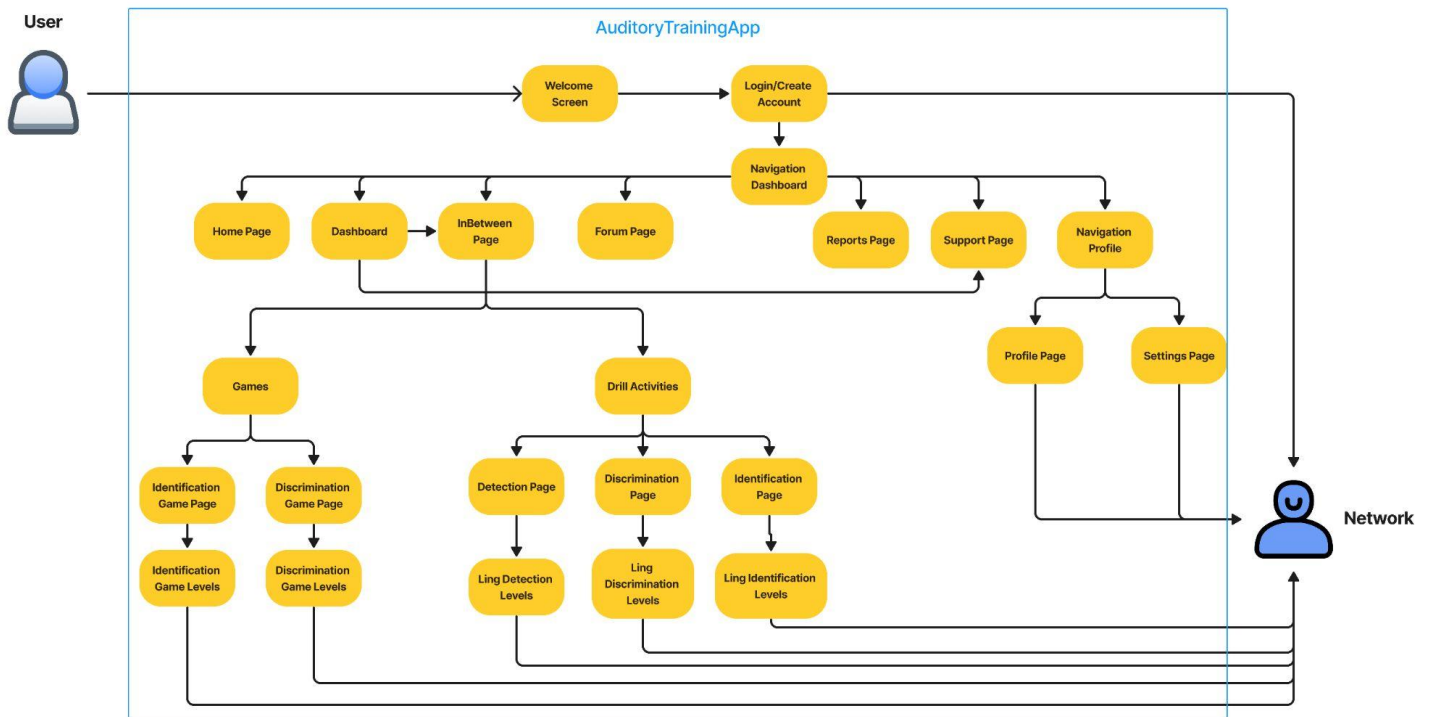
Persistent data management for the application is done through the Google Firebase Firestore. This allows the user to save their information such as profile settings, activity history, forum posts, and retrieve it upon their return to the application. The Firestore also allows us to utilize static data like the sounds being used in games and exercises.

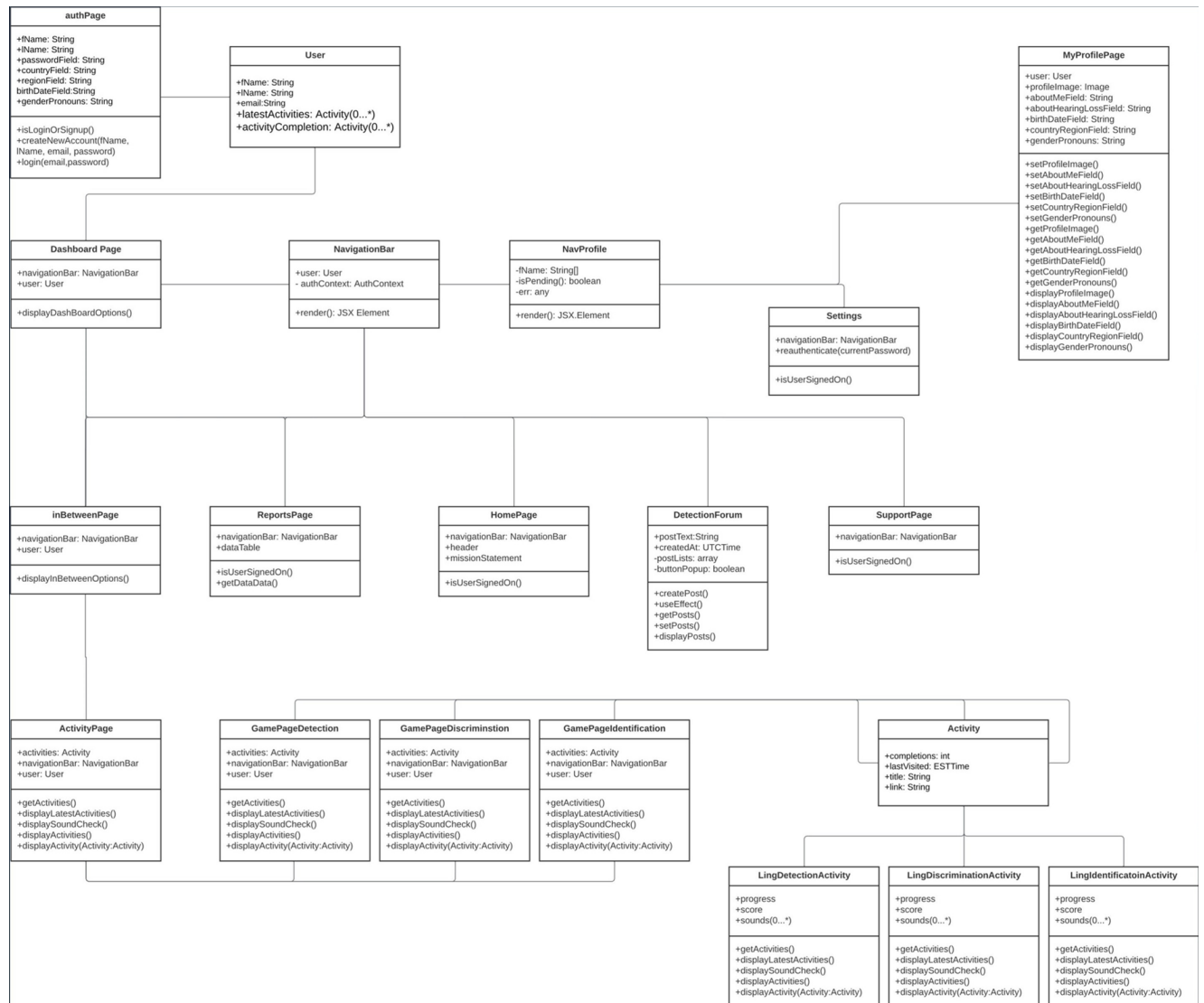
The following flowchart provides an overview of the major pages and modules within the application:



Appendix

Appendix A – UML Diagrams





Appendix B – User Interface

Mission: Audition!

HomeActivitiesForumReportsSupport

Samuel

Detection Games

Please select your activity!



Detection Level 1
Detect whether a sound
played or not: Ling 6 sounds



Detection Level 2
Detect whether a sound
played or not: Ling 10 sounds

Voice Selector



Click the button above to play the sound

If you heard a sound, click the button below

If you did not hear a sound, click the button below

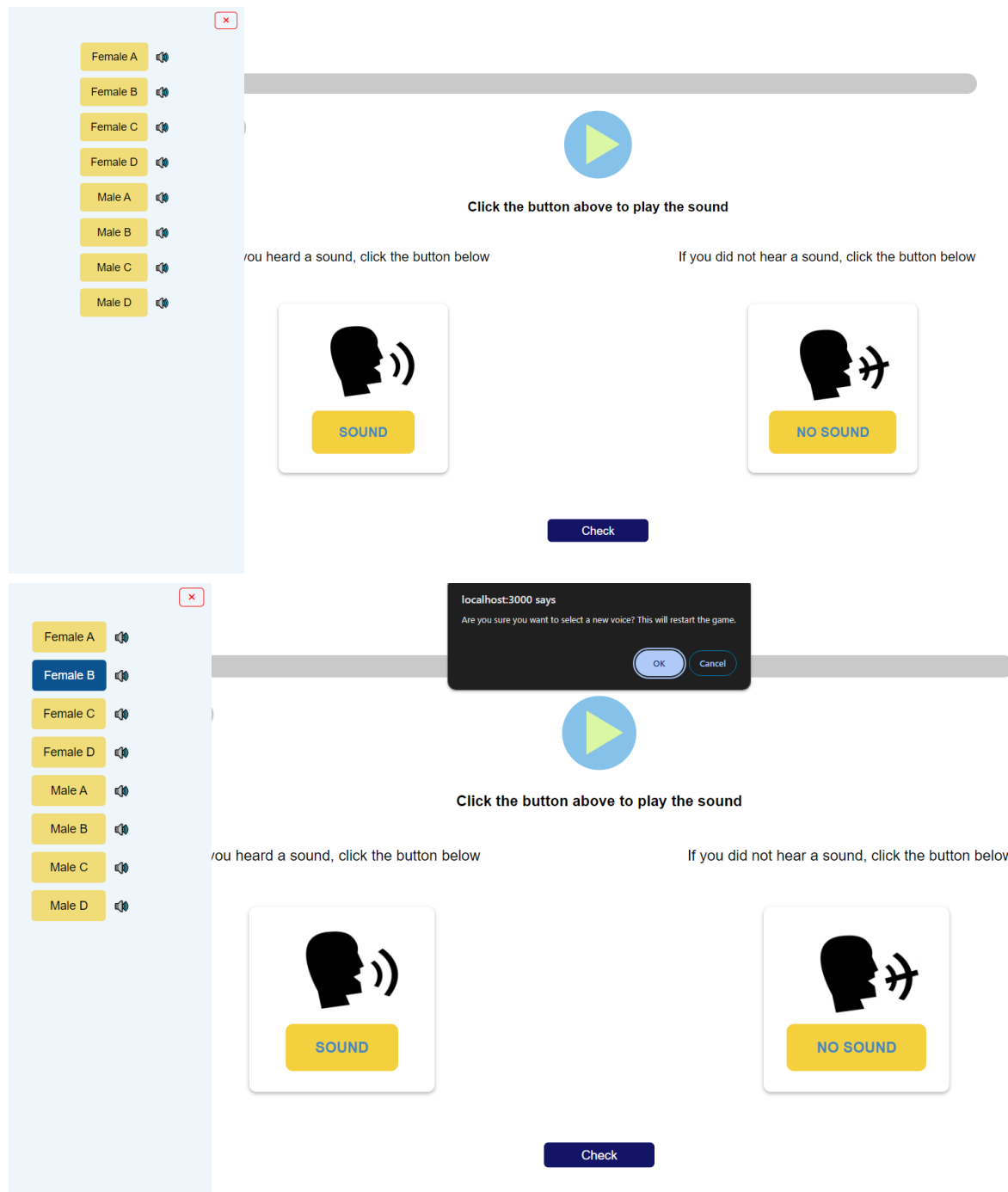


SOUND



NO SOUND

Check



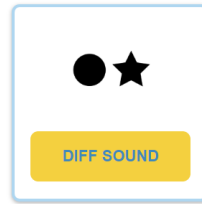
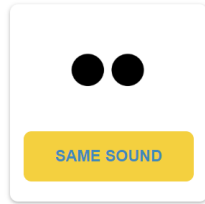
Voice Selector



Click each button above to play the sounds

If both sounds were the same, click the button below

If both sounds were different, click the button below



Check

×

Female A

Female B

Female C

Female D

Male A

Male B

Male C

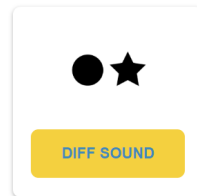
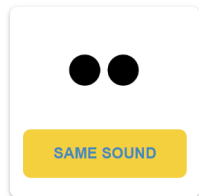
Male D



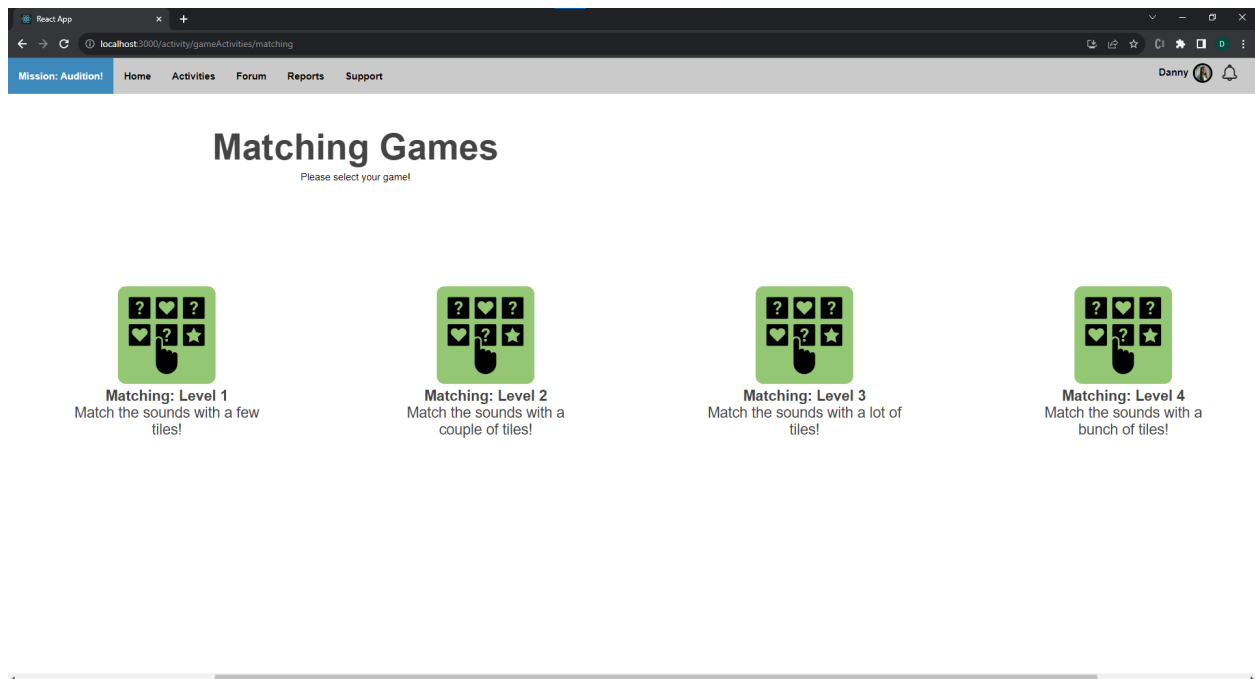
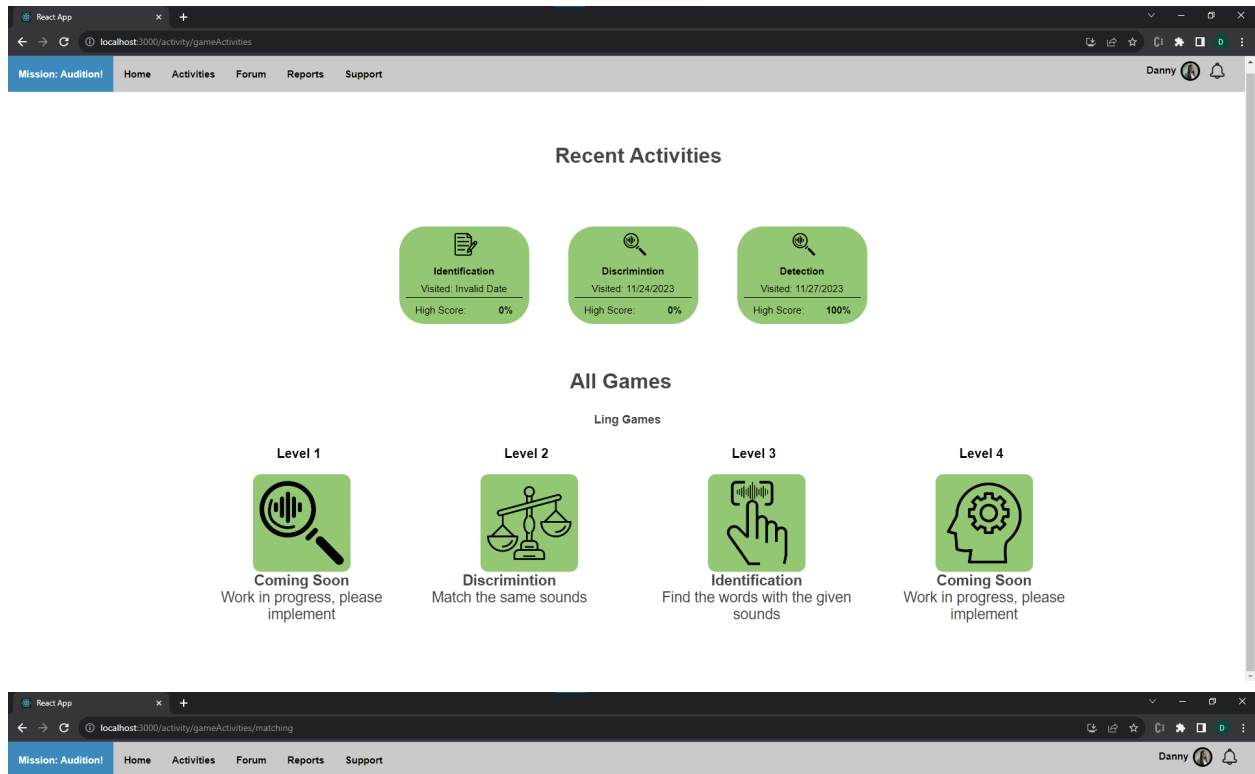
Click each button above to play the sounds

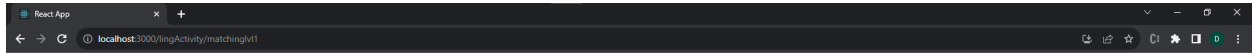
If both sounds were the same, click the button below

If both sounds were different, click the button below



Check

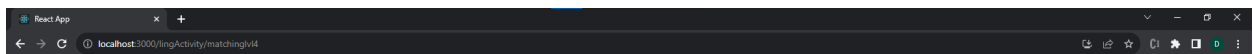




X

Matching Game: Level 1

1		2
	2	1



X

Matching Game: Level 4

		12		7	3
7	3	1	1		12

WordSearch Games

Please select your game!



WordSearch: Level 1
Find the words in the WordSearch puzzle using sounds!



WordSearch: Level 2
Find the words in the WordSearch puzzle using sounds!



WordSearch: Level 3
Coming Soon!



WordSearch: Level 4
Coming Soon!

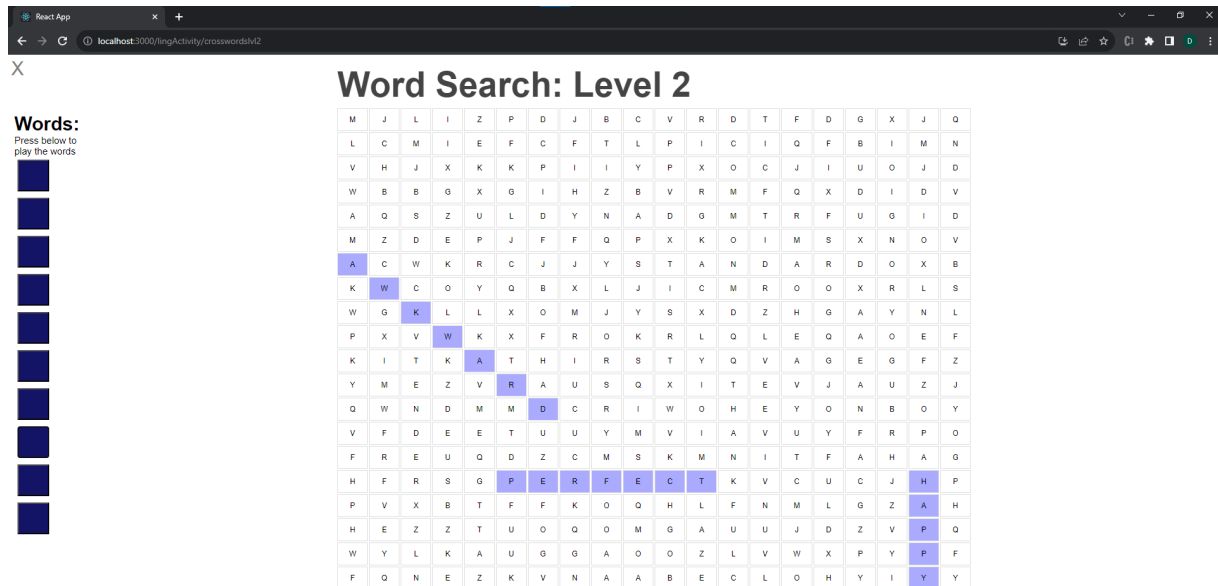
X

Words:
Press below to play the words



Word Search: Level 1

C	V	A	F	X	K	E	S	G	J	Z	G	B	C	C	F	N	M	U	J
T	E	V	J	K	X	K	A	D	I	R	B	W	K	C	Q	J	Y	B	B
S	G	Q	Y	F	G	V	G	X	P	M	R	O	D	S	N	J	O	G	N
J	F	Y	C	H	A	I	R	C	U	T	H	L	U	C	C	A	B	W	J
J	B	V	G	H	E	J	W	H	S	K	H	M	Y	Y	H	I	B	B	W
Q	E	L	V	A	E	G	M	V	X	W	Q	D	B	O	Z	R	O	H	K
F	R	W	J	I	R	E	B	E	N	M	B	B	C	Q	M	P	W	H	F
W	O	H	N	M	Q	M	S	I	R	Z	X	R	O	S	N	L	Q	D	N
E	R	O	S	P	R	A	C	E	R	Y	R	S	W	J	P	A	L	O	Q
S	C	R	T	F	Y	I	X	H	B	T	F	L	B	J	K	N	H	O	S
B	N	K	Y	P	E	W	L	G	A	A	H	Z	O	S	H	E	H	R	E
X	A	S	B	N	R	P	W	C	T	I	L	D	Y	M	T	H	D	M	G
G	E	S	Y	M	O	I	S	I	I	O	R	L	A	T	Y	M	B	A	S
L	B	V	E	Q	R	H	N	P	C	M	Q	B	M	Y	B	U	Q	T	P
P	G	J	H	B	U	F	L	T	I	H	P	S	R	T	E	H	C	Z	F
D	W	L	S	Z	A	B	E	O	C	U	P	C	A	K	E	T	A	D	Y
C	Z	Y	D	W	B	L	V	M	U	D	R	L	V	N	M	U	E	D	R
J	X	F	R	I	Z	Y	L	K	F	V	Y	S	S	Q	A	B	P	R	Y
Q	Y	X	Y	Z	M	K	Y	I	X	P	D	T	D	D	A	W	F	W	M
K	I	T	U	I	C	M	M	S	L	U	G	S	J	P	N	T	C	I	U



Appendix C – Sprint Retrospective Reports

Sprint Retrospective Meeting Minutes #1:

What went wrong?

- Did we do a good job estimating our team's velocity?
 - Yes, the velocity was correctly estimated.
- Did we do a good job estimating the points (time required) for each user story?
 - The time required for each user story was properly estimated.
- Did each team member work as scheduled?
 - Each team member worked as scheduled to have the project running.

What went right?

- The team members successfully ran the project.
- The meetings with the PO were useful to define the user stories.

How to address the issues in the next sprint?

- How to improve the process?
 - No improvements are needed as of now for the process.
- How to improve the product?
 - Work on the assigned user stories to improve the functionality of the product.

Sprint Retrospective Meeting Minutes #2:

What went wrong?

- Did we do a good job estimating our team's velocity?
 - Yes, the velocity was correctly estimated.

- Did we do a good job estimating the points (time required) for each user story?
 - The time required for each user story was properly estimated.
- Did each team member work as scheduled?
 - Each team member worked as scheduled to have the project running.

What went right?

- The team members successfully ran the project.
- Team members communicated efficiently and were very flexible.
- The meetings with the PO were useful to define the user stories.

How to address the issues in the next sprint?

- How to improve the process?
 - No improvements are needed as of now for the process.
- How to improve the product?
 - Work on the assigned user stories to improve the functionality of the product.

Sprint Retrospective Meeting Minutes #3:

What went wrong?

- Did we do a good job estimating our team's velocity?
 - Yes, the velocity was correctly estimated.
- Did we do a good job estimating the points (time required) for each user story?
 - The time required for each user story was properly estimated.
- Did each team member work as scheduled?
 - Each team member worked together on an issue with CardDB which caused the entire application to crash.

What went right?

- The team members successfully ran the project.
- Team members communicated efficiently and were very flexible.
- The meetings with the PO were useful to define the user stories.

How to address the issues in the next sprint?

- How to improve the process?
 - Collaborate together and accept each members ideas.
- How to improve the product?
 - Fix this issue and ensure steady progress.

Sprint Retrospective Meeting Minutes #4:

What went wrong?

- Did we do a good job estimating our team's velocity?
 - Yes, the velocity was correctly estimated.
- Did we do a good job estimating the points (time required) for each user story?
 - The time required for each user story was properly estimated.

- Did each team member work as scheduled?
 - Each team member worked as scheduled to have the project running.

What went right?

- The team members successfully ran the project.
- Team members communicated efficiently and were very flexible.
- The meetings with the PO were useful to define the user stories.

How to address the issues in the next sprint?

- How to improve the process?
 - No improvements are needed as of now for the process.
- How to improve the product?
 - Work on the assigned user stories to improve the functionality of the product.

Sprint Retrospective Meeting Minutes #5:

What went wrong?

- Did we do a good job estimating our team's velocity?
 - Yes, the velocity was correctly estimated.
- Did we do a good job estimating the points (time required) for each user story?
 - The time required for each user story was properly estimated.
- Did each team member work as scheduled?
 - Each team member worked as scheduled to have the project running.

What went right?

- The team members successfully ran the project.
- Team members communicated efficiently and were very flexible.
- The meetings with the PO were useful to define the user stories.

How to address the issues in the next sprint?

- How to improve the process?
 - No improvements are needed as of now for the process.
- How to improve the product?
 - Work on the assigned user stories to improve the functionality of the product.

Sprint Retrospective Meeting Minutes #6:

What went wrong?

- Did we do a good job estimating our team's velocity?
 - Yes, the velocity was correctly estimated.
- Did we do a good job estimating the points (time required) for each user story?
 - The time required for each user story was properly estimated.
- Did each team member work as scheduled?
 - Each team member worked as scheduled to have the project running.

What went right?

- The team members successfully ran the project.
- Team members communicated efficiently and were very flexible.
- The meetings with the PO were useful to define the user stories.

How to address the issues in the next sprint?

- How to improve the process?
 - No improvements are needed as of now for the process.
- How to improve the product?
 - Work on the assigned user stories to improve the functionality of the product.

Appendix D – User Manuals, Installation/Maintenance Document, Wishlist User Manual

Upon opening the application, the user is prompted to either sign up or log in if they have an existing account. New users are required to specify their name, email, password, location, preferred pronouns, and birth date. Once they have been authenticated, they will be taken to the landing page, which introduces the application, goals, and methodology behind the exercises and games. They are directed to click on the “Let’s Start” button.

Once a user is logged in, they can access the activity, games, and reports pages from the dashboard. Alternatively, they can use the navigation bar to access the home page, dashboard, activity page, forum page, reports page, and support page. From activities, they will enter the In Between Page and be prompted to choose between Drill and Game activities. The Game activities section is now functional as of this iteration. If the user clicks on Drill Activities, they are taken to the activity home page, where they can choose between Detection, Discrimination, Identification, and Comprehension activities (of these, comprehension is not yet functional). Clicking on any of these activity categories will prompt users to navigate to the Game Page for that activity. From each category’s game page, users can click on the game card to enter each game.

On the reports page, they can view their account activity history. This focuses on the date and score they received from the various available activities. They can scroll down and click the page numbers to view old report logs. Additionally, clicking the ‘Export’ button at the top will export the report data as a csv.

On the support page, users are prompted to enter their name, email, and message to request support with any of the activities or bugs encountered in the application. The ‘submit’ button is not yet functional.

Users can access their user profile by hovering over their name on the top right corner and clicking on “My Profile”. This will take them to their public user profile displaying different information pertaining to them. Clicking on the “Edit Profile” button will prompt users to edit their profile details. Clicking on save after any changes will save these changes to the database, and their user object will be permanently altered.

Also navigated by hovering over their name is the “Settings” page. Here, users can choose to reset their password or email. They can also apply night mode to their application. Keep in mind

that functionality is still in development, so while the night mode is persistent, it does not apply to the entire application at this moment.

Users can access the forum page via the navigation bar. On the forum page, they can add new posts by clicking on the "Add new post" button in the top right corner. The messaging and thread functionalities are not yet finalized.

README

Auditory Training App

Authors:

Capstone 2: Nicole Cruz, Ashley Enalien, Rafael Mejia Galvan, Hubert Cabrera, Enrique

Portela Product Owner: Alliete Alfano

The Auditory Training App is currently in its fourth version. It is designed for adults with hearing loss who have received a hearing aid or cochlear implant and want to improve their hearing

via fun games and other listening activities.

Project Structure

The following tree structure will be useful to visualize how the project is structured:

...

mission-audition/

├ node_modules/

├ public/

| └ favicon.ico

| └ index.html

| └ logo192.png

- | └ logo512.png
- | └ manifest.json
- | └ robots.txt
- └ src/
 - | └ assets/
 - | | └ icons/
 - | | └ images/
 - | └ components/
 - | | └ activities/
 - | | └ activity/
 - | | └ auth/
 - | | └ backdrop/
 - | | └ cardDB/
 - | | └
 - cardsLatest/ | |
 - └ Completed/
 - | | └ custHooks/
 - | | └ forumComp/ |
 - | └ Game Option/ | |
 - └ Game Options/ | |
 - └ homepage-sects/ |
 - | └ lingDetection/ | |
 - └ lingDiscrimination/ |

- | └ lingIdentification/ |
- | └ nav/
- | | └ modal/
- | | └ Option/
- | | └ options/
- | | └ playButton/
- | | └ pop/
- | | └ progressBar/
- | | └ store/
- | └ helpers/
- | | └ allActivities.js
- | | └ allDashboardOptions.js
- | | └ allBetweenOptions.js
- | | └ allDiscriminationGames.js
- | | └ allIdentificationGames.js
- | | └ allDetectionGames.js
- | └ pages/
- | | └ activityPage/
- | | └ auth/
- | | └ dashboard/
- | | └ forum/
- | | └ gamePage/
- | | └ homePage/

```
| | └ inBetweenPage/
| | └ ling/
| | └ myProfile/
| | └ reportsPage/

| | └ reset/
| | └ supportPage/
| | └ settings/
| └ App.js
| └ firebase-config.js
| └ index.css
| └ index.js
└ .env
└ .gitattributes
└ .gitignore
└ package-lock.json
└ package.json
└ README.md
...
```

Code/Auditory Training App is the directory that contains all of the code required to make the app functional. If/When other directories are added, they will be used for documentation.

The Auditory Training App/src directory is where all the source code for the application lies. The main entry point of the application is through index.js, which is where the application is created, and immediately calls App.js where all other components that change the UI must be referenced.

Project Structure provides a visual imagery of how classes call each other and how the project is structured.

Installation & Running steps

The easiest way to get started with Auditory Training App is to install Node.js. You can do that by going to :

...

<https://nodejs.org/en/>.

...

This will automatically install the Node Package Manager (NPM).

It is also necessary to be able to run git commands in your terminal (<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>), or use the desktop app (<https://desktop.github.com/>).

After Node.js is installed, and you can run git commands/or the app, you will need to clone the GitHub repository to your local machine using the git clone command or do so internally within your IDE.

Using your terminal, You can now install all the necessary dependencies for the node_modules folder by navigating to the directory folder, and running :

...

```
npm install
```

```
npm install firebase
```

...

It is necessary to create a firebase-config.js file. This should be placed in the src folder. To find the firebase key that should be contained in that file, please refer to the following website: <https://support.google.com/firebase/answer/7015592#web&zippy=%2Cin-this-article>

Please be advised that you will have to include an export statement including db, auth, and storage.

Note: If you get an API invalid key error, please delete your local repository, re-clone from git, and run the package installation commands again.

The app can now be started on your local machine by running:

...

npm start

...

It is necessary to create a firebase-config.js file. This should be placed in the src folder. To find the firebase key that should be contained in that file, please refer to the following website:
<https://support.google.com/firebase/answer/7015592#web&zippy=%2Cin-this-article>

Please be advised that you will have to include an export statement including db, auth, and storage.

The app can now be started on your local machine by running:

...

npm start

...

Installation Guide

1. Installation:

- a. Install Node.js -> <https://nodejs.org/en/>
- b. Install git on your computer -> <https://git-scm.com/downloads>
- c. Download an IDE, we recommend Visual Studio Code ->
<https://code.visualstudio.com/download>

2. Obtaining the Project:

- a. Clone the project from the github repository found here :
<https://github.com/allietealfano/auditorytrainingappupdated>
 - i. This can be done internally through your IDE or through external tools such as Git Bash.
 - ii. Go to <https://docs.github.com/en/repositories/creating-and-managing>

repositories/cloning-a-repository for more information about cloning repositories.

- b. After cloning, navigate to where the repository is stored locally and open it using your preferred IDE.

3. Firebase Config --> this project relies upon a Firebase application. a. To connect Firebase to your local repository, you will need to have a configuration file called "firebase-config.js".

- b. This will be placed in the src folder within the project directory. It is not included in the remote repository due to security risks regarding the API key.

- c. Please refer to the Installation video or the README to see what should be in the firebase-config.js.

- i. It should contain the API key and firebase project configurations, but also export out the db, user, and storage functionalities.

4. Installing Dependencies:

- a. Navigate to the folder containing the project within your terminal, or within your IDE's terminal, after cloning.

- b. Install all dependencies by running "npm install" in your terminal. For first time installations, this will take some time.

- c. Run "npm install firebase" to finalize the firebase configuration.

- i. NOTE: If you get an API key error, you may have done something wrong while installing the dependencies. Please repeat steps 2-4.

5. Running the Application:

- a. Once all installation is complete, you may type npm start in the terminal to run the project.

- i. This will open the web application on your default browser at localhost:3000. ii.

When running, please make sure this port is not being used by another application.

Shortcomings / Wishlist

- Settings Page
 - Apply night theme to the rest of the application
- Game Activities
 - Add more activities for each drill activity category
 - Add difficulty slider by adding background noise
 - Implement level updater
- Forums
 - Add deleting post functionality
 - Finish implementing forum threads
 - Finish implementing forum messaging
 - Fetch profile picture from profile page to display on forum
- UI
 - Add Spinner/Progress bar animation to Reports UI to show that data is in progress of being fetched
 -

- Reports Page
 - Add filtering to filter by activities
 - Add sharing capability
 - Organize the data more meaningfully (EX: by activity type, by date played, etc.)
- In Between Page
 - Add game activities
- Support Page
 - Make submit button functional