Design Document Team 6 CS246

# Team Info:

|  |  |  |
| --- | --- | --- |
| Adrian Lane | 909-900-9935 | lan16009@byui.edu |
| Kenyon Bunker | 480-789-1665 | bun17008@byui.edu |
| Graham Walker | 8179448515 | Walkerg5@byu.edu |
| Ruben Rodriquez | 5491123881047 | Rod17015@byui.edu |

# UML Design:

# [https://www.lucidchart.com/documents/view/e3820a0c-4e6d-40d3-a7ed-72b5d430ee52#](https://www.lucidchart.com/documents/view/e3820a0c-4e6d-40d3-a7ed-72b5d430ee52)

# Project Overview:

Create a spelling app for young kids. At a minimum, it should have the ability to take a picture of a spelling list from child’s teacher and add those words to a spelling game.

# Core Requirements:

1. OCR image to words
2. Text box to add words
3. Text to speech from the list of words
4. Single game mode: spell the spoken word
5. Local storage for the word list
6. A simple intuitive UI
7. Simple reward system for each correctly spelled word

# Stretch Goals:

1. 1) Master word bank
2. Progress system with rewards/badges
3. Random spelling list from word bank
4. SUPER STRETCH - Advanced reporting: overall progress, trouble words, accuracy trending, difficulty by word length

# Overview of the design / approach:

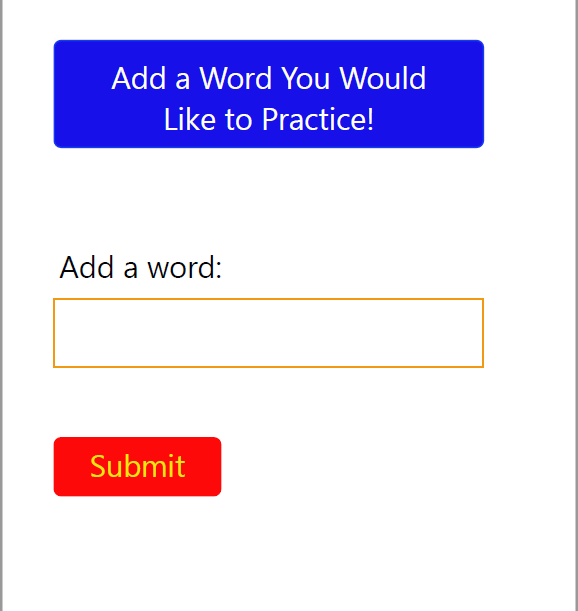
1. Main Menu – Activity - Kenyon
   1. Take image of spelling list
   2. Enter Game Mode
   3. Parent can see advance reporting (Stretch)
   4. View Existing images
2. Take Image – Activity - Kenyon
   1. The user will be able to take a photo of the spelling list
   2. Use built in camera app
   3. Saves photo to file, only visible to app. **Do not** include in device main phot album.
   4. User inputs a name or week # or some other way of identifying the spelling list
   5. Auto deletes photos after a week, if they have already been consumed by the OCR service
3. Manually add spelling words – Activity - Graham
   1. User should be able to add words manually to an existing list
4. View Images – Activity – Kenyon
   1. User will be able to see past images
   2. This will be reliant on blob storage as local images are deleted after a week
5. View Images – Class – Kenyon
   1. Helper class for View Images – Activity
6. OCR API – Class(es) - Adrian
   1. Send image(s) to Azure Computer Vision API
   2. Process image(s) – user should be able to see status
   3. Return text list to activity for user/parent to review and approve
   4. User confirms the name of the list chosen when submitting the image(s)
   5. List is added to the game mode
   6. List is added to the master list (Stretch)
   7. User is returned to main menu
7. Game Mode – Spell the Spoken Word – Activity – Adrian
   1. Since this is going to be big part of the app. Adrian will be doing the UML design for this section. But for all the other development parts of the project we may want to work on this together and divide and conquer as nessiary.
   2. Make it simple but inviting.
   3. Make it fun for a child (K-6)
      1. Reward system with children appropriate badges (Stretch)
      2. Positive verbal and visual feedback for mis-spelled word
      3. Verbal and visual reward for correctly spelled word
   4. Selection for user to choose spelling list by name
   5. Text to Speech integration – App will say the word out loud
   6. User will use keyboard to spell the word.
   7. Button for word to be repeated
   8. Adds progress to advanced reporting (Stretch)
8. Text-to-Speech – Class(es) - Graham
   1. Azure cognitive services or Android Studio API
   2. User should hear the word(s) from the word list in game mode
   3. In Game Mode the Text-to-Speech should “read” the words one at a time
9. Cloud Storage - Ruben
   1. Azure Blob storage for stored images
   2. Azure Database for all other data persistence

# SnapNSpell UI Mockups

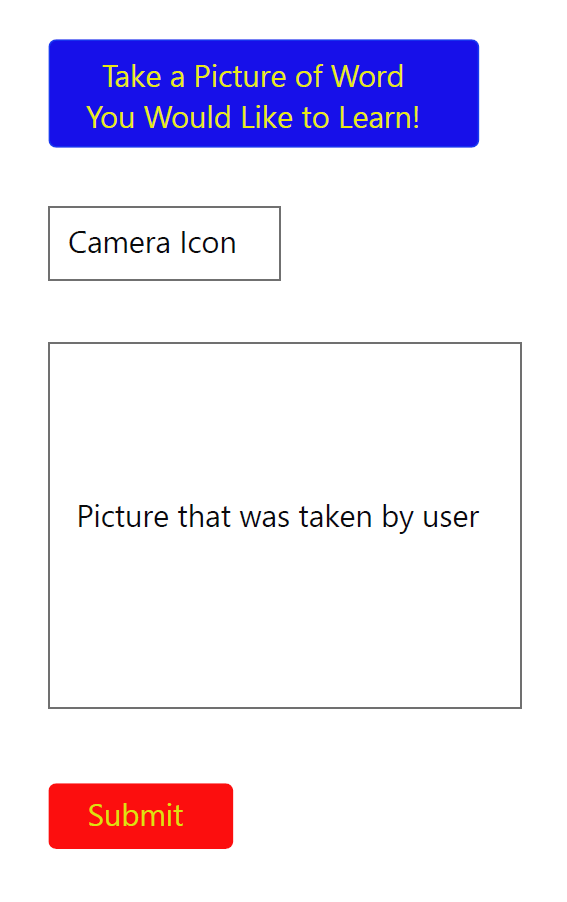
**HOME PAGE**



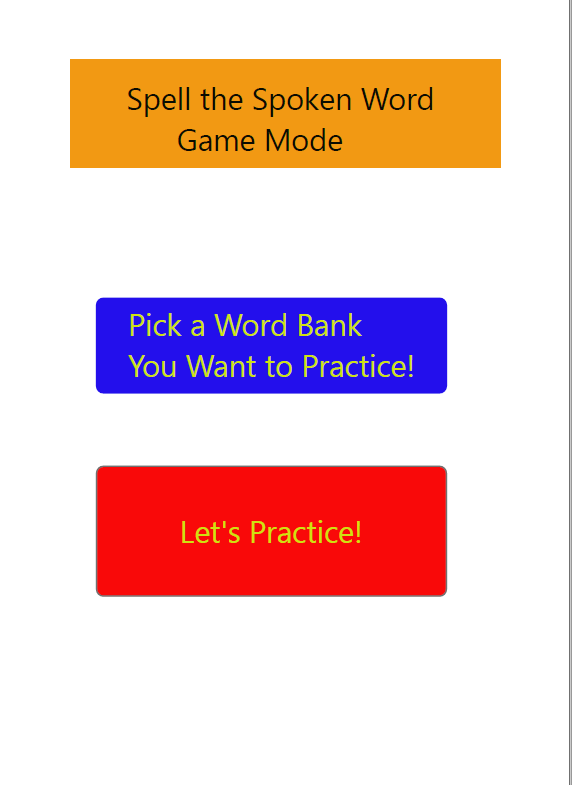
**MANUALLY ADD WORDS**



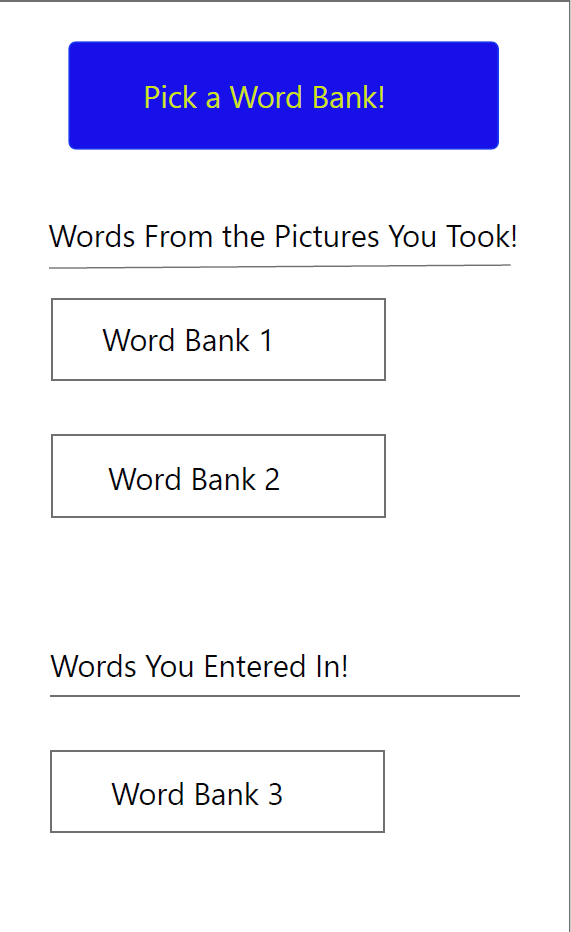
**ADD IMAGE OF WORD**

****

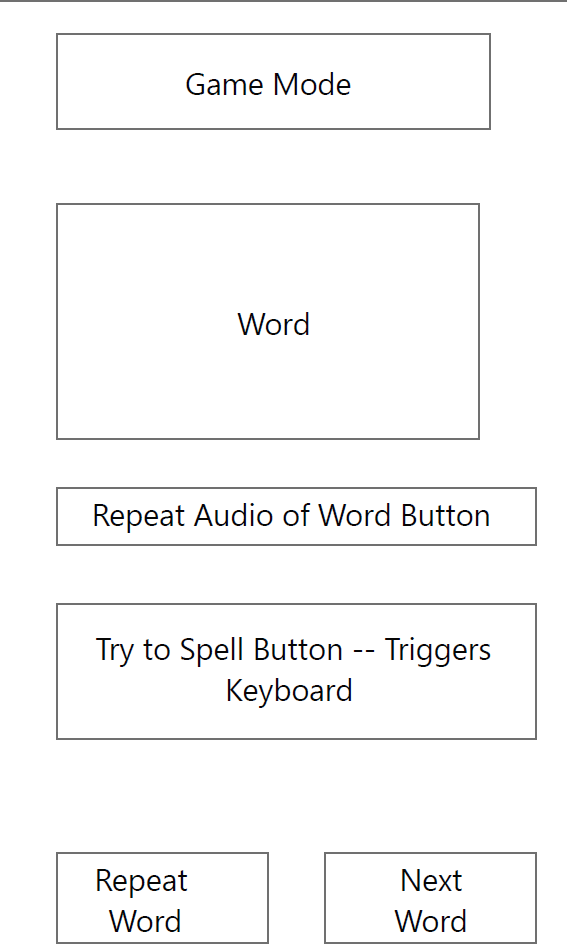
**SPELL THE SPOKEN WORD GAME MENU**

****

**PICK WORD BANK TO PRACTICE**

****

**SPELL THE SPOKEN WORD GAME MODE**

****