SOFTWARE REQUIREMENTS SPECIFICATION

for

\<The Red Book\>

Version 1.1

Prepared by \<Hanna Jung,
Joon Han,
Sean (Sang-Uk) Lee\>

\<CSE308\>

September 25, 2019

Contents

Revision History

Version	Date	Changes
1.0	9/23/2019	Initial Document Creation
1.1	9/24/2019	Minor change of features and scope of the project

1 Introduction

1.1 Product Description

Using both the left brain and the right brain is the best method for humans to learn a new language. Young children rely on using the right brain to learn new things quickly and easily. Studies have show that it is more effective for children to learn by using their senses such seeing the images and hearing the sound of words than just by looking at the words itself. The Red Book is a web application that utilizes the visual and auditory advantages of learning to allow students to play games to learn and review English words in a fun and engaging way.

However, we know each student learns at a different pace and have different interests. To make the Red Book be suitable for all students, the Red Book provides teachers to create customized learning sets (images and corresponding their words) for each student. Teachers will be able to create these learning sets to allow each student to learn fun English words at their own pace. The Red Book is also a tool for teachers to observe the learning progress of their students and identify their strengths and weaknesses to tailor their lessons for the future. Using data visualization and keeping track of every student actions, teachers will be able to conveniently view each student's progress unlike in a regular classroom setting which can promote a much better learning environment.

The Red Book is a great educational web application that can be used as a complementary tool for teachers to use in their classrooms. Its fun interactive games with customized study sets will keep students en ganged and learn English at their own pace while allowing teachers to observe the students' progress.

1.2 Existing Alternatives

Two existing alternatives that the Red Book is similar to are: **Duolingo** which is a crowd-sourced language learning web application, and **Rosetta Stone** which is a visual and audio based learning solution. Duolingo provides their users to learn their desired language through a form of quiz games and automated pronunciation or accent check. Rosetta Stone provides users to learn their desired language through visual instructions and audio, and helps them acquire an ability to have fluent daily conversations after using their *participated learning.*

While both of these applications are great for the students to learn a language, they may not be the best options for an elementary student that require an engaging and customized learning experience. The Red Book provides a flexible system teachers to create study-sets for their lessons and provides them the ability to customize the experience for each individual student.

1.3 Scope

The Red Book is web-based application that will allow students to play games to learn English and for teachers to observe and analyze student activity. The Red Book will be accessible on any device that supports a web browser.

1.4 Documentation & Support

For Students: In order for students to become familiar with this application, a first-run walkthrough will guide the students to demo each interface and games to familiarize them with the application. **For**

Teachers: In order for teachers to become familiar with this application, an online user-manual will be provided to familiarize them with the application. Additionally, a community board will also be available for discussions between teachers and the developers for suggestions to improve the application and request for features.

2 Features & Requirements

2.1 Major Features

2.1.1 Customized Visual Learning

The Red Book provides system-provided and teacher-created study sets to visually simulate students to associate English words with the objects in the images. The Red Book is available with variety of categories and difficulty levels for different types of students. However, teachers will be able to create and customize the datasets for the games to create appropriate learning activities for each class and individual student. This flexibility allows the Red Book to be tailored to each individual student.

2.1.2 Auditory Learning

The Red Book provides audio samples of the displayed words that appear in the games to simulate students ears and to associate English words with the sounds from the audio samples.

2.1.3 Data Analysis

Teachers will be able to visually observe the learning trends of each class and student, and analyze particular difficulties each student encounters.

2.1.4 Game Modes

Different game modes will allow students to engage with the problems differently, and learn the English words in multiple different ways.

2.1.5 Stretch Goals

Community for Co-operative Development

Unless the developer continually updates and creates content for the Red Book, it cannot continually be used as a learning application once all the datasets have been memorized. Therefore creating a community website where teachers may create, share, and download each other's datasets will be important for the continued life cycle of the Red Book.

Additional Game Modes:

The initial game modes that the Red Book will provide may not give students enough variety to keep their interest. We plan to explore different game methods for education, and implement additional games that use images and words as a dataset.

Reward System

Studies have demonstrated that rewarding users in games with badges or medals motivate them to participate more. By implementing a reward system, such as badges or promotions to celebrate the students usage the Red Book, we believe it'll intice them to play more.

2.2 Use Cases

2.2.1 Sign Up

Use-Case:	Sign Up
Primary Actor:	User
Goal in Context:	The user can create an account for the application
Preconditions	The application has been started and the user has pressed the **Sign Up** button from home screen
Trigger:	The user clicks on the **Sign Up** button
Priority:	Essential: Must be implemented

Scenario:	1. User starts the application, which loads the home screen
	2. User views the home screen, and clicks on the **Sign Up** button
	3. Application loads the create account page
	4. User fills in the account name, password, name field and upload a profile picture
	    4.1. User may fill in the special teacher's code given to them by their institution to create a teacher account
	5. User presses the **Create An Account** button
	    5.1. If the account name exists, notify the user with a pop-up to choose new account name
	    5.2. If the teacher code is invalid, notify the user with a pop-up to re-enter teacher code
	    5.3. If all information is valid, the account is created and directed to their respective main(teacher—student) page

Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.2 Create Classroom

Use-Case:	Create Classroom
Primary Actor:	Teacher
Goal in Context:	The teacher can create a classroom
Preconditions	The account is identified as a teacher account and teacher has pressed **Create Classroom** button from teacher page
Trigger:	The teacher clicks on the **Create Classroom** button
Priority:	Essential: Must be implemented

Scenario:	1. Teacher views the teacher page, and clicks on the **Create Classroom** button
	2. Application loads the create classroom page
	3. Teacher fills in the classroom name and classroom code
	4. Teacher presses the **Complete** button
	    4.1. If the classroom code exists, notify the user with a pop-up to choose new classroom code
	    4.2. If the classroom name and code is unfilled, notify the teacher with a pop-up to re-enter classroom name code
	    4.3. If all information is valid, the classroom is created and directed to the classroom page

Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.3 Add Student

Use-Case:	Add Student
Primary Actor:	Teacher
Goal in Context:	The teacher adds a student to a classroom
Preconditions	The teacher is in a classroom page
Trigger:	The teacher clicks on the **Add Student** button
Priority:	Essential: Must be implemented

Scenario:	1. Teacher views the teacher page, and clicks on the **Add Student** button
	2. Application loads the add student page
	3. Teacher fills in the student id
	4. Teacher presses the **Search** button
	   4.1. If the student id is unfilled or does not exist, notify the teacher with a pop-up to re-enter student id
	   4.2. If all information is valid, the student a pop-up with the student id, student name, and the classroom the student will be added to is displayed with an **Add** button
	      4.2.1 Teacher presses the **Add** button, the student is added to the classroom and the teacher is directed to the classroom page

Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.4 View Classrooms

Use-Case:	View Classrooms
Primary Actor:	Teacher
Goal in Context:	The teacher views all of the created classrooms
Preconditions	The teacher is in the teacher page
Trigger:	The teacher clicks on the **View Classrooms** button
Priority:	Essential: Must be implemented
Scenario:	1. Teacher views the teacher page, and clicks on the **View Classrooms** button 2. Application loads the view classrooms page 3. A list of created classrooms is displayed, each with its name and a **View** button
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.5 View Classroom

Use-Case:	View Classroom
Primary Actor:	Teacher
Goal in Context:	The teacher views a classroom
Preconditions	The teacher is in the view classrooms page
Trigger:	The teacher clicks on the **View** button
Priority:	Essential: Must be implemented
Scenario:	1. Teacher views the classrooms page, and clicks on the **View** button 2. Application loads the classroom page 3. A list of students enrolled in the classroom is displayed, each with its name and a **View** button
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.6 View Student

Use-Case:	View Student
Primary Actor:	Teacher
Goal in Context:	The teacher views a student profile
Preconditions	The teacher is in the classroom page
Trigger:	The teacher clicks on the **View** button
Priority:	Essential: Must be implemented
Scenario:	1. Teacher views the classroom page, and clicks on the **View** button
	2. Application loads the student profile page
	3.Student name, profile picture, and the list of games the student has participated in is displayed. Each listing of a game contains a **View Game** button
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.7 View Game

Use-Case:	View Gameplay Record
Primary Actor:	User
Goal in Context:	The users views a student gameplay record
Preconditions	The user is in the student profile page
Trigger:	The user clicks on the **View Game** button
Priority:	Essential: Must be implemented
Scenario:	1. User views the student profile page, and clicks on the **View Game** button
	2. Application loads the gameplay record page
	3. A list of record with the date of gameplay, game score, with a **View Score** button is displayed
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.8 View Game Score

Use-Case:	View Game Score
Primary Actor:	User
Goal in Context:	The user views score of a game
Preconditions	The user is in the gameplay record page
Trigger:	The user clicks on the **View Score** button
Priority:	Essential: Must be implemented
Scenario:	1. User views the gameplay record page, and clicks on the **View Score** button
	2. Application loads the game score page
	3. A list of the correct answers along with the student input is displayed is displayed
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.9 Create Word List

Use-Case:	Word List creation
Primary Actor:	Teacher
Goal in Context:	The teacher can create a word list
Preconditions	The account is identified as a teacher account and teacher has pressed **Create Word List** button from teacher page
Trigger:	The teacher clicks on the **Create Word List** button
Priority:	Essential: Must be implemented

Scenario:	1. Teacher views the teacher page, and clicks on the **Create Word List** button
	2. Application loads the create word list page
	3. The teacher types in the new name for the word list
	4. Teacher types in the word in the word search text box
	4 Teacher presses the **Search** button
	   5.1. If no image exists in the system for the specified word, the system will generate an image in the image box
	   5.2. If the image exists in the system, the system will show the image in the image box
	6. The teacher may press the **Add Word** button to add the word to the list
	7. The teacher may press the **Finish** button to complete creating the word list

Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.10 Data Analysis

Use-Case:	Data Analysis
Primary Actor:	Teacher
Goal in Context:	The teacher can analyze the data of its students
Preconditions	The account is identified as a teacher account and teacher has pressed **Data Analysis** button from teacher page
Trigger:	The teacher clicks on the **Data Analysis** button
Priority:	Essential: Must be implemented

Scenario:	1. Teacher views the teacher page, and clicks on the **Data Analysis** button
	2. Application loads the data analysis page
	3. Teacher selects in the filters to receive data for
	4. Teacher presses the **Search** button
	   4.1. If no data exists for the specified filters, notify the teacher with a pop-up to choose new filters
	   4.2. If all information is valid, visualization of data is presented along with list of difficult and easy words of the filtered students

Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.11 Find Classroom

Use-Case:	Find Classroom
Primary Actor:	Student
Goal in Context:	The Student wants to final a classroom
Preconditions	The account is identified as a student account and the student has pressed **Find Classroom** button from student page
Trigger:	The teacher clicks on the **Find Classroom** button
Priority:	Essential: Must be implemented

Scenario:	1. Student views the student page, and clicks on the **Find Classroom** button
	2. Application loads the find classroom page
	3. Student fills in the classroom code
	4. Student presses the **Search** button
	   4.1. If no classroom code exists, notify the student with a pop-up to re-enter the class code
	   4.2. If information is valid, display the classroom code and classroom name along with the **Register** button

Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.12 Register into Classroom

Use-Case:	Register into Classroom
Primary Actor:	Student
Goal in Context:	The Student wants to register into a classroom
Preconditions	The account is identified as a student account and the student has pressed **Register** button from the find classroom page
Trigger:	The student clicks on the **Register** button
Priority:	Essential: Must be implemented
Scenario:	1. Student views the find classroom page, and clicks on the **Register** button 2. Student is registered into the classroom
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.13 Show Games

Use-Case:	Show Games
Primary Actor:	Student
Goal in Context:	The Student wants to see the games available to play
Preconditions	The account is identified as a student account and the student has pressed **Play Games** button from the student page
Trigger:	The student clicks on the **Play Games** button
Priority:	Essential: Must be implemented
Scenario:	1. Student views the find student page, and clicks on the **Play Games** button 2. Student is shown the list of games available to play
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.14 Start Spelling Bee

Use-Case:	Start Spelling Bee Game
Primary Actor:	Student
Goal in Context:	The Student wants to play the spelling bee game
Preconditions	The account is identified as a student account and the student has pressed **Spelling Bee** button from the play games page
Trigger:	The student clicks on the **Spelling Bee** button
Priority:	Essential: Must be implemented
Scenario:	1. Student views the find play games page, and clicks on the **Spelling Bee** button 2. Student is shown the list of topics available to play
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.15 Choosing Category for Spelling Bee

Use-Case:	Start Spelling Bee Game
Primary Actor:	Student
Goal in Context:	The Student wants to play the spelling bee game
Preconditions	The account is identified as a student account and the student has pressed **Spelling Bee** button from the play games page
Trigger:	The student clicks on the **Spelling Bee** button
Priority:	Essential: Must be implemented
Scenario:	1. Student views the find play games page, and clicks on the **Spelling Bee** button 2. Student is shown the list of topics available to play
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.16 Play Spelling Bee

Use-Case:	Play Spelling Bee Game
Primary Actor:	Student
Goal in Context:	The plays the spelling bee game
Preconditions	The account is identified as a student account and the student has chosen the topic from choose categories page
Trigger:	The student clicks on the **Spelling Bee** button
Priority:	Essential: Must be implemented
Scenario:	1. Student is shown an image alone with the corresponding box for each letter of the image 2. Student chooses presses the underlined text box. 3. A pop-up box with choices of letters is displayed for the student. 4. Student clicks on one of the letter. 5. The letter appears on the text box. 6.1 Student may press **Submit** button after completion. 6.2 Student may press the **Sound** button to hear the audio sample of the word. 6.3 Student may press the **Hint** button. 6.3.1. One of the text box is filled with the correct corresponding letter.
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.17 Start Matching Game

Use-Case:	Start Matching Game Game
Primary Actor:	Student
Goal in Context:	The Student wants to play the Matching Game game
Preconditions	The account is identified as a student account and the student has pressed **Matching Game** button from the play games page
Trigger:	The student clicks on the **Matching Game** button
Priority:	Essential: Must be implemented
Scenario:	1. Student views the find play games page, and clicks on the **Matching Game** button 2. Student is shown the list of topics available to play
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.18 Choosing Category for Matching Game

Use-Case:	Start Matching Game Game
Primary Actor:	Student
Goal in Context:	The Student wants to play the Matching Game game
Preconditions	The account is identified as a student account and the student has pressed **Matching Game** button from the play games page
Trigger:	The student clicks on the **Matching Game** button
Priority:	Essential: Must be implemented
Scenario:	1. Student views the find play games page, and clicks on the **Matching Game** button 2. Student is shown the list of topics available to play
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.19 Play Matching Game

Use-Case:	Play Matching Game Game
Primary Actor:	Student
Goal in Context:	The plays the Matching Game game
Preconditions	The account is identified as a student account and the student has chosen the topic from choose categories page
Trigger:	The student clicks on the **Matching Game** button
Priority:	Essential: Must be implemented
Scenario:	1. Student is shown an image alone with the corresponding choice of words for the image 2. Student clicks on one of the words. 3.1 Student may press **Submit** button after completion. 3.2 Student may press the **Sound** button to hear the audio sample of the word. 3.3 Student may press the **Hint** button. 3.3.1. One of the incorrect word box is eliminated.
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.2.20 View Score

Use-Case:	View Score
Primary Actor:	Student
Goal in Context:	The Student views its score
Preconditions	The teacher is in the student page
Trigger:	The student clicks on the **View Score** button
Priority:	Essential: Must be implemented
Scenario:	1. Student views the student page, and clicks on the **View Score** button 2. Application loads the student profile page 3.Student name, profile picture, and the list of games the student has participated in is displayed. Each listing of a game contains a **View Game** button
Open Issues:	Size, location, and style of button should be finalized by UI designer

2.3 User Interfaces
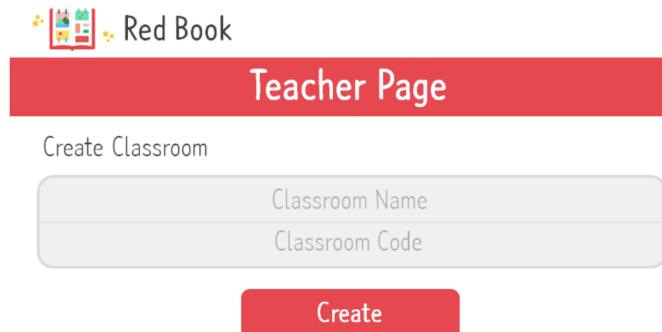
2.3.1 Home Screen
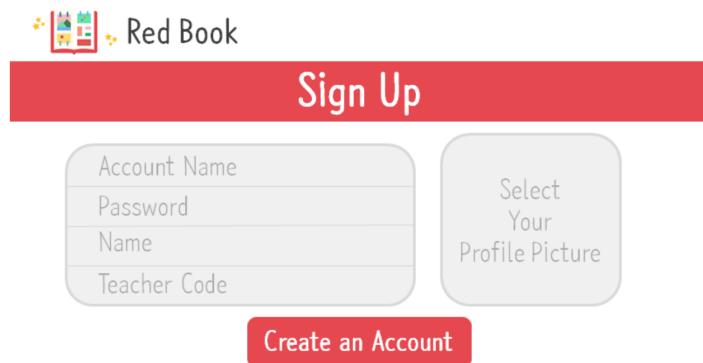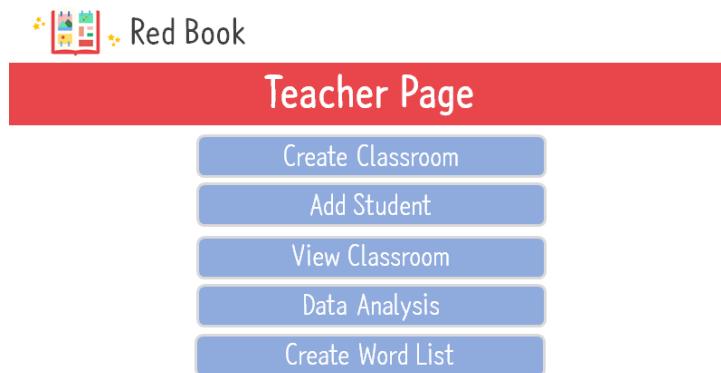
2.3.2 Sign Up Page

Red Book

Sign Up

Account Name
Password
Name
Teacher Code

Select
Your
Profile Picture

Create an Account

2.3.3 Teacher Page

Red Book

Teacher Page

Create Classroom
Add Student
View Classroom
Data Analysis
Create Word List

2.3.4 Create Classroom Page

Red Book

Teacher Page

Create Classroom

Classroom Name
Classroom Code

Create

2.3.5 Add Student Page

Red Book

Teacher Page

Add Student

Student ID

Search
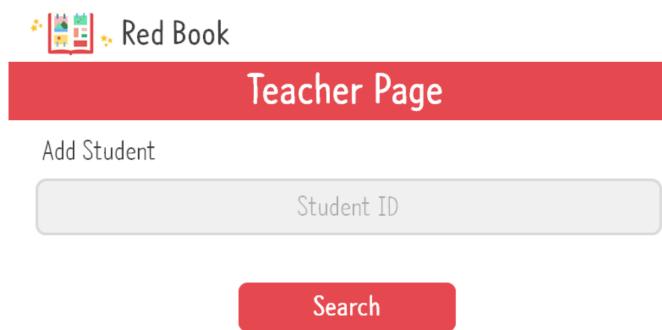
2.3.6 Add Student Page

Red Book

Teacher Page

Add Student

Student ID	111222333
Name	Alice Lee
Classroom	1st Grade

Add

2.3.7 View Classrooms Page

Red Book

Teacher Page

View Classroom

1st Grade Class	View
2nd Grade Class	View
3rd Grade Class	View
4th Grade Class	View

2.3.8 View Classroom
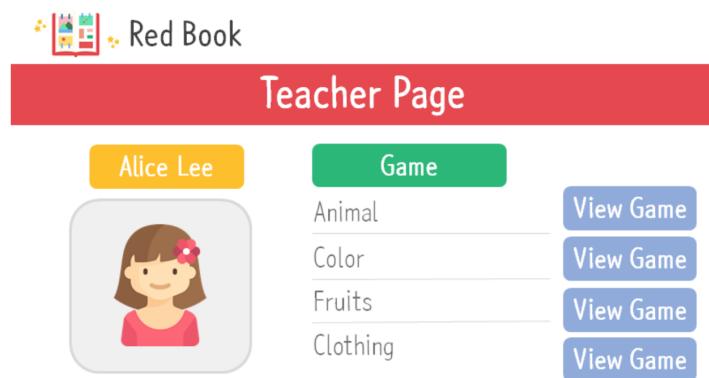
Red Book

Teacher Page

1st Grade Class

Hanna Jung	View
Joon Han	View
Sean Lee	View
Alexander Kuhn	View

2.3.9 View Student Profile Page

Red Book

Teacher Page

Alice Lee

Game	
Animal	View Game
Color	View Game
Fruits	View Game
Clothing	View Game

2.3.10 View Student Games Page

Red Book

Teacher Page

Alice Lee   Animal

Date	Score	Game	
09/24/19	08/15	Spelling Bee	View Score
09/25/19	10/15	Matching Game	View Score
09/29/19	13/15	Spelling Bee	View Score

2.3.11 View Student Scores Page

Red Book

Teacher Page

Alice Lee   Animal

Answer	My answer
cow	cow
dog	cat
fox	fox

2.3.12 Create Word List Page

2.3.13 Data Analysis Page
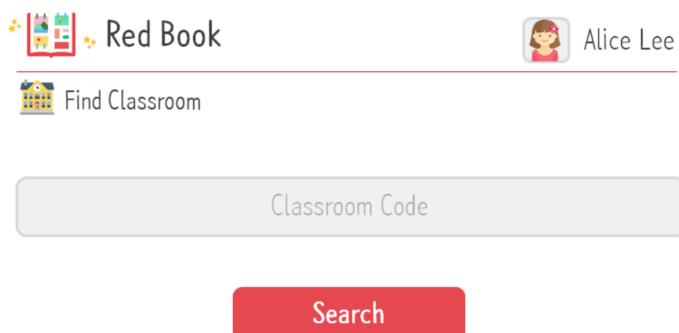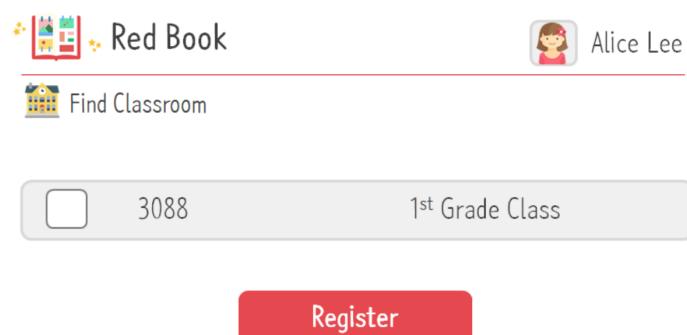
2.3.14 Student Page
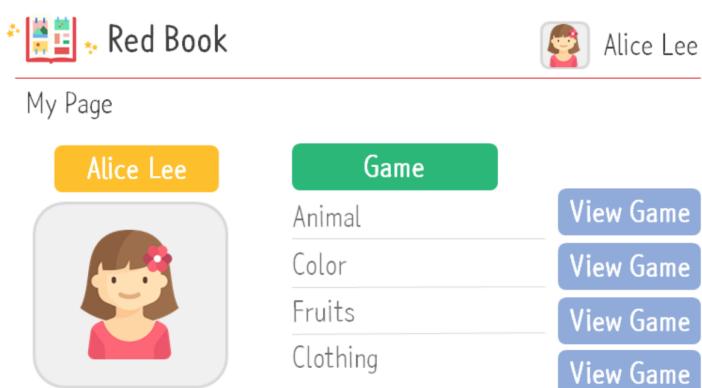
2.3.15 Find Classrooms Page

✦ 📕 **Red Book**   👧 Alice Lee

🏫 Find Classroom

Classroom Code

Search
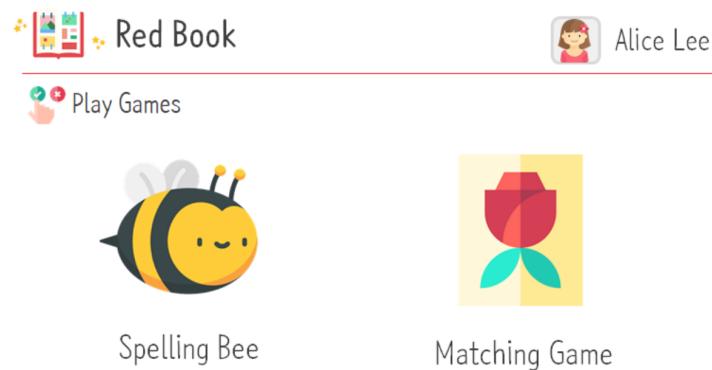
2.3.16 Join Classroom Page

✦ 📕 **Red Book**   👧 Alice Lee

🏫 Find Classroom

☐   3088   1st Grade Class

Register
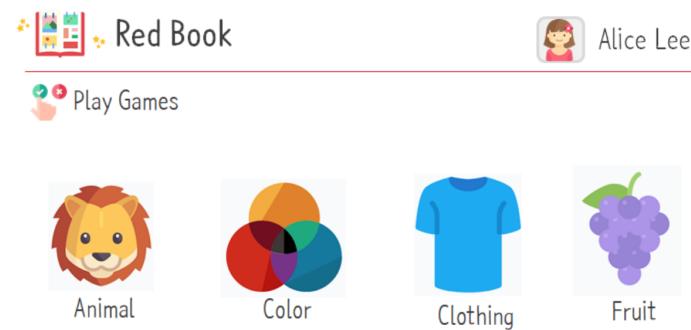
2.3.17 Student Profile Page

✦ 📕 **Red Book**   👧 Alice Lee

My Page

Alice Lee   **Game**

Animal   View Game

Color   View Game

Fruits   View Game

Clothing   View Game
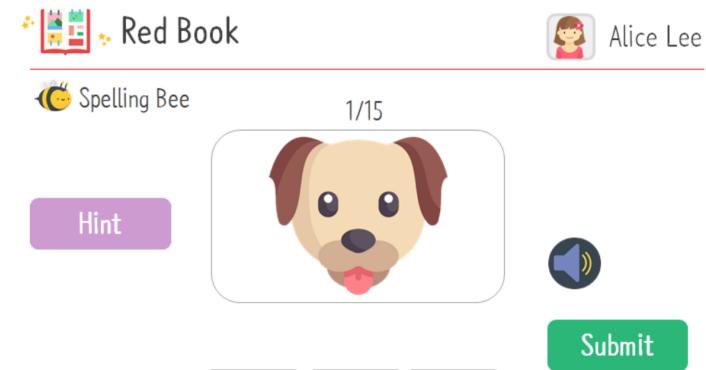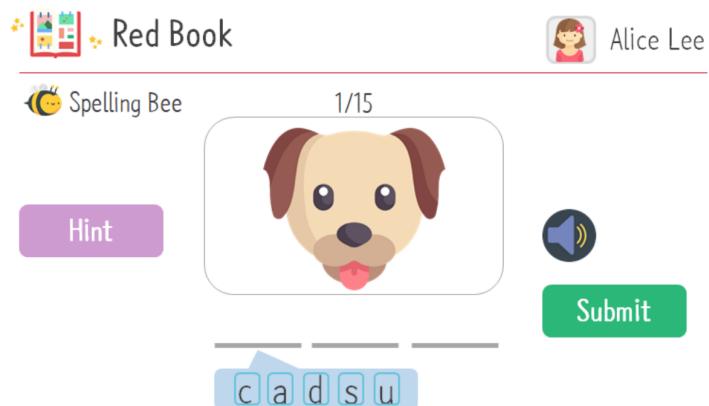
2.3.18 Play Games Page
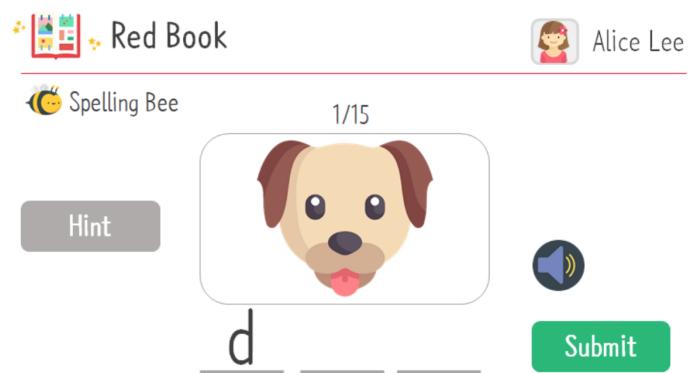

2.3.19 Choose Categories
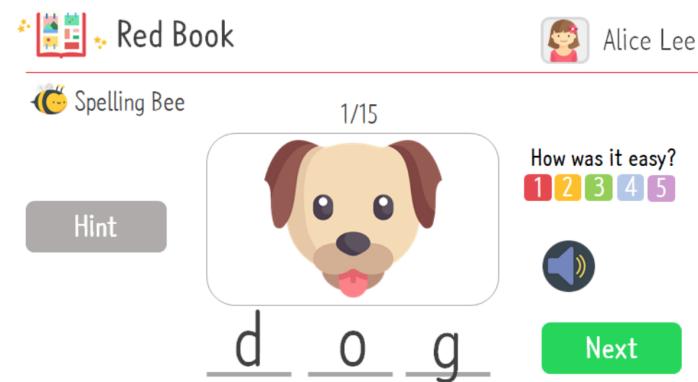

2.3.20 Spelling Bee Game Page
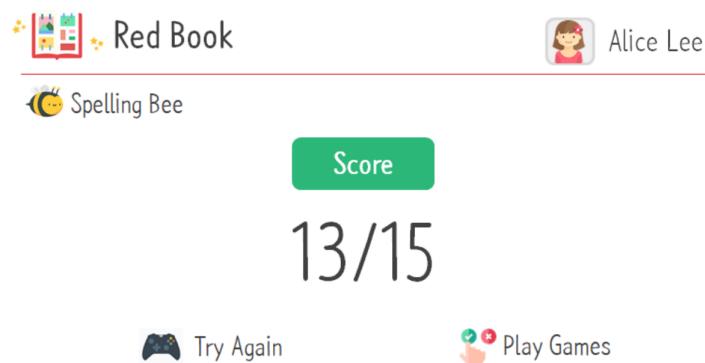
2.3.21 Spelling Bee Input

2.3.22 Spelling Bee Hint

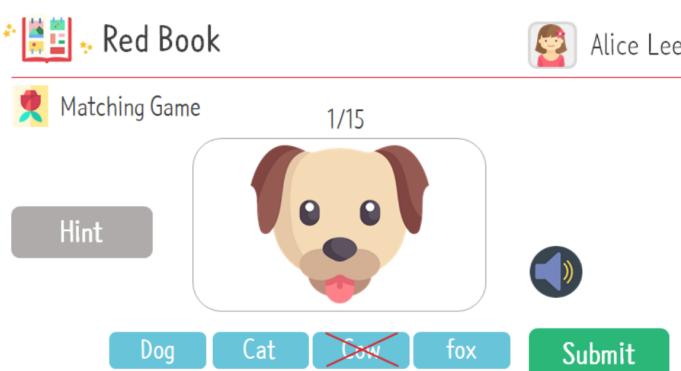### 2.3.23 Spelling Bee Answer
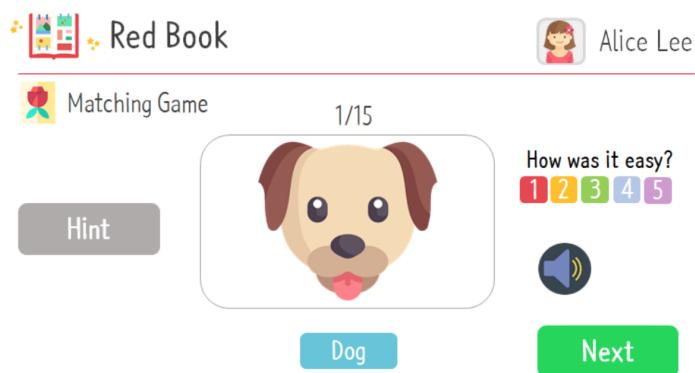
2.3.24 Matching Game Page

2.3.25 Matching Game Input

2.3.26 Matching Game Hint

2.3.27 Matching Game Answer

2.3.28 Student View My Games Page

Date	Score	Set	Game	
09/24/19	09/15	Animal	Spelling Bee	View Score
09/21/19	10/15	Color	Matching game	View Score
08/12/19	15/15	Clothing	Matching game	View Score

2.3.29 Student View Score Page

Answer	My answer
cow	cow
dog	cat
fox	fox

2.4 Non-functional Requirements

2.4.1 Safety

The Red Book should not allow inappropriate and graphic images be uploaded and be displayed as images for the games. Datasets must be approved by a set amount of teachers before it can be publicly shared and used in the application.

2.4.2 Privacy

Student and teacher information such as name, id, profile picture, and game scores will be encrypted and stored in a private server. Only users who are verified as teachers may be able to search for students. In order to verify the teacher is legitimate, the teacher must request an one-time use unique code from the developer with their institutional e-mail to create a teacher account.

2.4.3 Availability

The application will continuously be online, and be available for use 24 hours, 7 days a week throughout the year.

2.4.4 Response Time

Responses to queries shall take no longer than 5 seconds to load onto the screen after the user submits the query.

3 Schedule and Risks

3.1 Schedule

Tentative Deadline	Objective	Led By (Respective Order)
Oct. 1st, 2019	Software Design Specification	Hanna Jung
Oct. 9th, 2019	Research Review on Learning Algorithms	Sean Lee, Joon Han
Oct. 10th, 2019	Design Presentation	Hanna Jung
Oct. 12th, 2019	Initial Environment Setup	Sean Lee, Joon Han
Oct. 15th, 2019	[Project Milestone 1] Alpha Version of The Games.	Sean Lee, Joon Han
Oct. 16th, 2019	Begin Conducting First User Study	The Team
Oct. 22nd, 2019	Research Review on Student Data Analysis and Data Visualization	(Sean Lee, Hanna Jung)
Oct. 29th, 2019	[Project Milestone 2] Application Features: Create Account, Login, Search/Add Student, View Student List, Enroll to Class, View Own Status	The Team
Oct. 31st, 2019	Verification and Test Plan	The Team
Nov. 12th, 2019	Beta Version Of The Games, Adjusted to the Feedbacks	Sean Lee, Joon Han
Oct. 16th, 2019	Begin Conducting Second User Study	The Team
Nov. 19th, 2019	[Project Milestone 3] Application Feature: Learning Algorithm, Data Analysis	The Team
Dec. 5th, 2019	Final Project Release	The Team
Dec. 9th, 2019	Final Presentation	The Team

3.2 Risk Analysis

What are the major risks involved in completing your project? List at least 3 items. Why are you most worried about these risks? Describe how you plan to gather more information or reduce these risks (such as running experiments using some software components early on). Describe what you will do if you are unable to overcome a problem, such as you cannot get a 3rd party component to work? This can include cutting features, but feature cutting should not be your solution to all of your major risks.

3.2.1 Games Are Neither Fun nor Educational

If the Red Book is neither fun nor educational, the application will be unusable as it does not achieve its main priority. To minimize this risk, the developers must test and receive constant feedback from students and teachers to verify that the application's goal is being met. Additionally, developers must examine and test existing educational applications to compare with the Red Book to verify that it's functionalities are similar.

3.2.2 Rate Limit or Deprecation of Text-to-Speech APIs

The Red Book uses third party component specifically, text-to-speech APIs to provide audio samples of the words to the students. The developers of these APIs may limit the number of requests it receives, temporary shut it down for maintenance, or may deprecate the API. In order to satisfy the stretch goal of allowing teachers to create their own game datasets, these text-to-speech APIs are crucial. The Red Book should use multiple services text-to-speech APIs in order to guarantee that at least one API is functioning to provide audio samples.

3.2.3 Database Malfunction

There could be an instance were the third party database hosting the Red Book's data may malfunction. Therefore, the application must hold multiple external back-up of the data in-case of aforementioned malfunction.

3.3 Acquiring Feedback

Acquiring frequent feedback during the development process of The Red Book is crucial to ensure that the application is able to achieve its goals of being a fun and educative game for students while also being a helpful tool for teachers to monitor and control the educational pace and content for each class and individual students. Feedback can be acquired from kindergarten or elementary schools near Incheon Global Campus. Additional feedback can be acquired from private tutoring centers in Songdo.

3.3.1 Student Feedback

First and foremost, the games on the application has to be engaging and educational for the students. We plan to measure students' enjoyment, length of activity with the application, and amount of interaction with each game.

3.3.2 Teacher Feedback

The Red Book is not just an English learning application, but also a tool for teachers to monitor classroom and student progress. We will acquire feedback regarding what kind of data visualisation is helpful, missing, and redundant for teachers in monitoring student learning progres.