

TheALLDictionary:

Software Design Document

Spring 2021 CSE 416 Final project

Team Members:

Young-jae Moon → Product owner	
Sub-team A	Sub-team B
Young-jae Moon → Project manager youngjae.moon@stonybrook.edu Yoo-ra Kim → Lead programmer yoora.kim@stonybrook.edu	Seung-jae Kang → Designer seungjae.kang@stonybrook.edu Seo-young Ko → Designer seoyoung.ko@stonybrook.edu

1. Introduction

1.1 Product description

When people do not know an English word, they have to search its definition from a dictionary. Although they may be satisfied with the meanings and the example sentences given from the dictionary, they may also be dissatisfied when they cannot fully grasp the word after reading the dictionary's explanations. In this case, they then have to search its meaning through other dictionaries, encyclopedias or search relevant photos or videos from Google or YouTube for complete understanding.

“TheALLDictionary” is a web-based application that helps people to search the meanings of the term from Merriam-Webster Learner's dictionary, Merriam-Webster dictionary, Oxford English Dictionary, Urban dictionary, Wikipedia, Google News, Google Images, and YouTube at once. Here Merriam-Webster's Learner's is the most straightforward dictionary in which the terms are explained in most simple and plain English. By contrast, the Merriam-Webster dictionary explains the terms at the intermediate level, and the Oxford English Dictionary defines them at more advanced levels. Thus, the Urban dictionary is especially useful for finding the definitions of slang words. Hence the user can see the explanations of the term at various levels from each dictionary. Thus, if a dictionary is not sufficient to understand the meaning of the word, the user can refer to Wikipedia, Google News, Google Images and YouTube videos to understand more thoroughly.

The target audience of TheALLDictionary is someone who wants to search for the definition of an English word. Nonetheless, it is expected that students who learn English at school will use this application most frequently. Therefore, TheALLDictionary will include

additional features that help people to memorize the vocabularies that they have searched for. These features include creating and shuffling note cards for testing their learnings, sharing note cards, generating and taking sample exams, and participating in the word challenge that the user has made to its friends.

1.2 Scope

TheALLDictionary primarily serves the people who want to search the meanings and the exemplar sentences of an English term from the following dictionaries: Merriam-Webster's Learner's dictionary, Merriam-Webster dictionary, Oxford English Dictionary, and Urban dictionary. Thus, it primarily serves the people who want to search more about the concept of the term in Wikipedia, relevant news in Google News, relevant photos in Google Images, and related YouTube videos. Notice that English is the only supported language from these dictionaries, but not from Wikipedia, Google News, Google Images, and YouTube. For Wikipedia, Google News, Google Images, and YouTube, all the languages supported by these sources will be supported.

Another feature of TheALLDictionary is providing various study assistance features for the premium users. For example, the premium users will be able to create, edit, save and share its note cards. Here notice that the application will save the user's search history under the user's consent, and thereby they can select the words from their search histories to skilfully create note cards. Thus, the premium users will be able to generate and take sample tests based upon the note cards created.

The users can access TheALLDictionary by opening the following domain from their web browsers: www.thealldictionary.com.

1.3 Users

The primary users of TheALLDictionary are students and professionals who study English vocabularies and seek appropriate English words for their works. These users will likely know how to browse websites and use search tools.

Even though these are the primary users of TheALLDictionary, TheALLDictionary will be available to the general public.

1.4 User feedback

We will get our first feedback from our three potential users - Young-ho Kim, Sudara Ranasinghe, and Hae-in Park after completing the web views of all the pages. We will check whether they do not find any difficulties with the user interface (UI) and receive suggestions about improving the design of our UI.

Then, we will get our second feedback from these users when we release our beta version of the TheALLDictionary for project milestone 3. We will incorporate the feedback into our final project.

We will receive feedback from these users through having a zoom meeting for each of them and asking questions related to the project during the meeting for three main reasons. First, oral feedback usually contains more information than written feedback. Second, Sudara is currently living in Sri Lanka and hence physical meeting is not possible. Third, we would like to have virtual meetings for preventing the infection of coronavirus disease of 2019 (COVID-19).

We expect that the first feedback will be most useful, as it becomes harder to modify the direction of the project after this stage.

1.5 Existing alternatives

As aforementioned, there is an existing alternative mobile application called “All English Dictionaries” for searching a term from various dictionaries and other sources at once. Here notice that while “All English Dictionaries” is only available on iOS and Android devices, TheALLDictionary will be available on Windows and Macintosh devices as well. In this way, the user of TheALLDictionary will be able to search words on larger screens than the user of “All English Dictionaries.”

Besides, while “All English Dictionaries” outputs the meaning of terms from various English-Korean, Korean-English, and English-English dictionaries; TheALLDictionary outputs the meaning of terms only from English-English dictionaries. In this way, we will encourage people to grasp English vocabularies in English to foster English learning.

Furthermore, “All English Dictionaries” has a tab of Google Images to show users images of the term that they searched. For further extension, TheALLDictionary would include a YouTube tab, indicating videos related to the term.

The user interface (UI) of TheALLDictionary will be similar to the user interface of “All English Dictionaries” and www.quizlet.com. We found that the UI of “All English Dictionaries” is convenient to search words and check definitions from multiple sources. In this instance, though, unlike where “All English Dictionaries” do not tell explicitly about the difficulty of each dictionary, TheALLDictionary will show the difficulty of each dictionary to the user. Thus, we found that the UI of www.quizlet.com is straightforward and comfortable for creating and sharing note cards and test concepts.

1.6 Definitions

Term	Definition
Admin	Administrator
App	Application
API	Application Programming Interface
COVID-19	Coronavirus disease of 2019
Customer	Someone who pays money for subscribing to the premium features of the TheALLDictionary
DBMS	Database Management System
FK	Foreign Key
HTTP	Hypertext Transfer Protocol
ID	Identification number
MVC	Model-View-Controller
PK	Primary Key
RDBMS	Relational Database Management System
REST API	Representational state transfer
SAT	Scholastic Aptitude Test
SQL	Structured Query Language
TLS	Transport Layer Security
UI	User interface
User	Someone who interacts with the TheALLDictionary website
URL	Uniform resource locator

1.7 References

1. All English Dictionaries. (2021). Retrieved from <https://play.google.com/store/apps/details?id=com.copyharuki.englishenglishdictionaries&hl=en&gl=US>
2. Quizlet. (2021). Retrieved from <https://quizlet.com/en-gb>

2. Requirements

2.1 Functional Requirements

All the users shall:

1. be able to look up for words.
2. be able to view the search results corresponding to each source.
3. make a payment to be a premium user and thereby access all of the features.
4. be able to see the difficulty of each dictionary.

The signed-in users shall:

1. be able to change its nickname.
2. be able to change its password.
3. be able to change its phone number.
4. be able to change its email address.
5. be able to add, edit and delete profile pictures.
6. be able to delete its account.
7. be able to see the previous search history.
8. be able to view the note cards that the premium customers have created and shared to the public.
9. be able to create, edit and save the note cards.
10. be able to pay for a subscription.

The premium customers shall:

1. be able to change the order of each dictionary and the other sources to be displayed after search.
2. be able to make friends with other premium customers.
3. be able to make, edit and delete note cards.
 - a. can add an image URL for a description of a term.
 - b. can add a video URL for a description of a term.
4. be able to share its note cards with...
 - a. to specified friends.
 - b. to all of their friends
 - c. to all of their friends except specified friends
 - d. to all users
 - e. to all users except specified friends
5. be able to share its note cards to the public.
6. be able to take online sample quizzes generated by the system based on its or other premium customers' lists of note cards.
 - a. can set the total number of questions and the total score of the quiz.
7. be able to see its previous quiz results.
8. be able to cancel its subscription.
9. be able to take words challenges generated by the system.
10. be able to see the ranking of the customers for each word challenge based upon the score and the time taken to complete the word challenge.
11. be able to set its profile picture.

The premium customers should:

1. be able to share its note cards with...
 - a. their friends of friends
2. be able to chat with their friends.
3. be able to use a live photo for their profile pictures.

The administrator shall:

1. be able to change its nickname.
2. be able to change its password.
3. be able to change its phone number.
4. be able to change its email address.
5. be able to add, edit and delete profile pictures.
6. be able to delete its account.
7. be able to create word challenges and set
 - a. the number of questions
 - b. the total score
 - c. the time limit
 - d. the deadline

The system shall:

1. display the search bar on the top when the mouse cursor goes to the top of the screen.
2. display a tab bar on the bottom for changing sources when the mouse cursor goes to the bottom of the screen.

3. be able to generate problems for the quiz based upon the note cards created by the premium customers.
4. be able to grade the test submitted by the premium customers.
5. be able to show the result of the test with exact answers after the premium customer has gone through the test.
6. be able to generate problems for the word challenge based on the Scholastic Aptitude Test (SAT) vocabularies.
7. determine the ranking of the premium customers for each challenge based on the score and the time taken to complete the word challenge.
 - a. If the scores of two or more customers are the same, then the ranking will be determined by the time taken to complete the word challenge.
 - b. If all the scores are different, then the ranking will be purely determined by the score.
8. only allow the premium customers to access the premium features.

The system should:

1. provide a payment system created by KG Inicis.
2. verify the user's email address through sending the verification code to the user.
3. verify the user's phone number through sending the verification code to the user.
4. assist the premium customers in creating notecards based upon their search histories.

2.2 Use cases

2.2.1 Use Case: Look up a word

Primary Actor:	Users
Priority:	Essential
The goal in context:	The user wants to look up a word.
Preconditions:	The user has opened TheALLDictionary website.
Trigger:	The user decided to look up a word.
Scenario:	<ol style="list-style-type: none"> 1. The user types the word that wants to be looked up in the text box. 2. The user clicks the search button. 3. The system displays the searched results from Merriam Webster Learner's dictionary, Merriam Webster dictionary, Oxford English dictionary, Urban dictionary in its unique format. Thus, the system displays links for the search results obtained from Wikipedia, Google News, Google Images, and YouTube.
Extensions:	<p>2a. The user clicks a back button in the browser.</p> <p>2a.1 The searching process is canceled, and the system returns to the Home Page.</p>

2.2.2 Use Case: Change the order of the sources on the bottom tab bar

Primary Actor:	Users
Priority:	Essential
The goal in context:	The user wants to change the order of the sources being displayed while looking up for a word.
Preconditions:	The user has opened TheALLDictionary website and has signed in.
Trigger:	The user decided to change the order of the sources being displayed while looking up for a word.
Scenario:	<ol style="list-style-type: none"> 1. The user clicks the Settings button on the bottom tab bar. 2. The user changes the order of each source to be displayed. 3. The user clicks the “Save” button. 4. The system saves the new order and returns it to the Home Page.
Extensions:	<p>2a. The user clicks the “Reset” button.</p> <p style="padding-left: 40px;">2a.1 The system automatically changes the order to its original format.</p> <p>2b. The user clicks a back button in the browser.</p> <p style="padding-left: 40px;">2b.1 The editing process is canceled, and the system returns to the Home Page</p>

2.2.3 Use Case: *Pay for a subscription*

Primary Actor:	Customers
Priority:	Essential
The goal in context:	The customer wants to pay for a subscription to access all premium features.
Preconditions:	The customer has opened TheALLDictionary website and has signed in.
Trigger:	The customer who already has an account tries to use premium features for the first time.
Scenario:	<ol style="list-style-type: none"> 1. The customer clicks “My Page” and chooses the “Premium” tab. 2. The system will display a list of premium features and a payment information window. 3. The customer may read all the features and put its information such as name, card number, and expiration day on the corresponding text boxes in the payment window. 4. The customer clicks the “Pay” button. 5. If all the payment information is correct and processes succeed, the system will display the premium page.

	<p>6. The system will change the status of the customer to “Subscribing Premium.”</p> <p>7. The customer can check its new status in “My Account” in “My Page.”</p>
Extensions:	<p>2a. The customer clicks a back button in the browser.</p> <p>2a.1 The system will exit the premium page and go to the previous page.</p> <p>3a. After the customer puts all of its payment information and clicks a back button in the browser.</p> <p>3a.1 The system will exit the premium page and go to the previous page.</p> <p>4a. The payment information of the customer is wrong.</p> <p>4a.1 The system will display an error message that the payment information is wrong.</p> <p>4a.2 If the customer clicks the “OK” button, then all the text boxes become empty.</p>

2.2.4 Use Case: Cancel the subscription

Primary Actor:	Customers
Priority:	Essential
The goal in context:	The customer cancels its subscription.

Preconditions:	The customer has opened TheALLDictionary website and has signed in.
Trigger:	The customer wants to cancel the premium subscription and stop paying for it.
Scenario:	<ol style="list-style-type: none"> 1. The customer clicks “My Page” and chooses the “My Account” tab. 2. At the “My Account” tab, the customer can see his or her profile, such as name, phone number, and email address, and subscription status. 3. The customer clicks the “I want to stop my subscription” button. 4. The system will display an announcement pop-up window that asks, “Are you sure to stop the premium subscription? If you do so, you cannot use premium features anymore.” 5. The customer clicks the “Yes” button on the announcement pop-up window. 6. The system pops up another announcement window which is written, “Your subscription is successfully canceled.” 7. The customer’s subscription is canceled, and his or her payment information is deleted in the system.
Extensions:	5a. The customer clicks the “No” button on the announcement pop-up

	<p>window.</p> <p>5a.1 The announcement pop-up window disappears, and it goes back to the “My Account” window.</p>
--	--

2.2.5 Use Case: Take a test based on note cards

Primary Actor:	Customers
Priority:	Essential
The goal in context:	The customer takes an online sample test and gets the grade of the test.
Preconditions:	The customer has made note cards.
Trigger:	The customer wants to take an online sample test generated by the system based on the premium customer’s note cards.
Scenario:	<ol style="list-style-type: none"> 1. The customer clicks the “Quiz” button. 2. The customer chooses the bundle of note cards for testing. 3. The customer clicks the “Start” button. 4. The system generates questions based on the selected bundle of note cards. 5. The system displays the question and four option buttons. 6. The customer clicks one button as the answer. 7. The system changes the color of the selected button to blue. 8. The customer clicks the “Next” button.

	<p>9. The system displays the next question.</p> <p>10. The customer clicks the “Submit” button to submit the test.</p> <p>11. The system grades the test.</p> <p>12. The system displays the result of the test.</p>
Extensions:	<p>3a. The customer clicks the “History” button.</p> <p>3a.1 The system displays previous test results.</p> <p>7a. The customer clicks the “Prev” button.</p> <p>7a.1 The system displays the previous question.</p> <p>7b. The system disables the “Prev” button when the question is the first question.</p> <p>7c. The system disables the “Next” button when the question is the last question.</p>

2.2.6 Use Case: Take the words challenge

Primary Actor:	Customers
Priority:	Essential
The goal in context:	The customer takes the words challenge test generated by the system based on the random vocabulary.
Preconditions:	The system has generated questions of the words challenge test.
Trigger:	The customer wants to take the words challenge test.

Scenario:	<ol style="list-style-type: none"> 1. The customer puts the mouse cursor on the “Quiz” button. 2. The system will then pop up the drop-down menu. 3. The customer clicks the “Words Challenge” button. 4. The system goes to the words challenge page. 5. The customer clicks the “Start” button. 6. The system displays the question and four option buttons. 7. The customer clicks one button as the answer. 8. The system changes the color of the selected button to blue. 9. The system displays the next question after 3 seconds. 10. The system grades the challenge test after the customer solves the whole question. 11. The system goes to the result page.
Extensions:	<ol style="list-style-type: none"> 2a. The customer clicks the “Quiz” button. <ol style="list-style-type: none"> 2a.1 The system goes to the quiz page. 4a. The customer clicks the “View Rank” button. <ol style="list-style-type: none"> 4a.1 The system goes to the challenge ranking page. 6a. The customer does not click any button as the answer in 3 seconds. <ol style="list-style-type: none"> 6a.1 The system displays the next question.

2.2.7 Use Case: Make the card

Primary Actor:	Customers
----------------	-----------

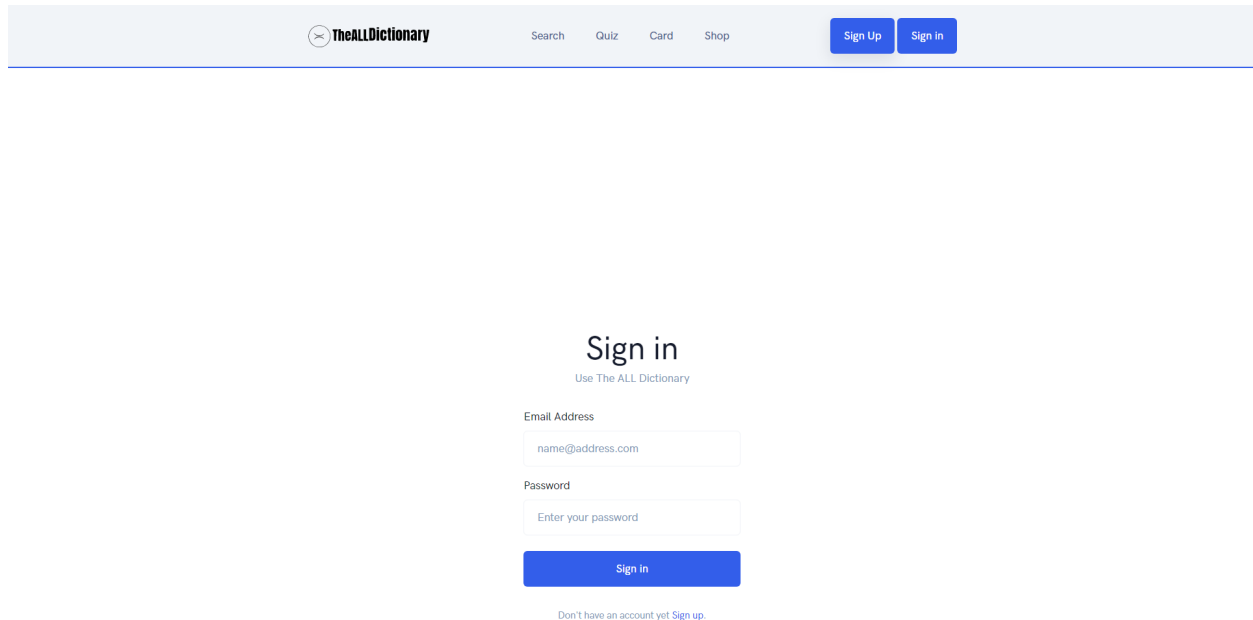
Priority:	Essential
The goal in context:	The customer can create note cards.
Preconditions:	The customer has opened TheALLDictionary website and has signed in.
Trigger:	The customer wants to make its own vocabulary cards.
Scenario:	<ol style="list-style-type: none"> 1. The customer clicks the “Card” button and reviews the search history. 2. The customer clicks a desired word from the search history. 3. The customer writes its meaning and additional memo on the box at the right. 4. The customer clicks the “Save” button.
Extensions:	<p>3a. The customer inserts an image by clicking the “Image” button.</p> <p>4a. The customer cancels and discards the current working card by clicking the “Cancel” button.</p>

2.2.8 Use Case: Review the card

Primary Actor:	Customers
Priority:	Essential
The goal in context:	The customer can access and change created cards.

Preconditions:	The customer has made vocabulary cards.
Trigger:	The customer wants to see created cards.
Scenario:	<ol style="list-style-type: none">1. The customer clicks one from the different lists.2. The customer can now view the cards created.
Extensions:	<ol style="list-style-type: none">1a. The customer moves to other lists by clicking the drop-down button at the right.2a. The customer shuffles the order of cards by clicking the shuffle shape button.2b. The customer changes the memo of the card by clicking the pencil shape button.2c. The customer deletes the selected card by clicking the trash bin shape button.

2.3 User Interfaces



The screenshot shows the top navigation bar of TheALLDictionary website. It includes the logo, a search bar, and links for Search, Quiz, Card, and Shop. There are also buttons for Sign Up and Sign in. Below the navigation bar, the main content area displays the 'Sign in' heading, a subheading 'Use The ALL Dictionary', and two input fields: 'Email Address' (with placeholder 'name@address.com') and 'Password' (with placeholder 'Enter your password'). A blue 'Sign in' button is positioned below the password field. At the bottom, there is a link for users who don't have an account yet to 'Sign up'.

Figure 1. Sign-in page

This is the sign-in page. If the user clicks the “Sign in” button at the top of the screen, it will redirect to the sign-in page. Once the user types its id and password and then clicks the “Sign in” button, the system will check whether the user has entered them correctly. If the user has entered them correctly, the system will allow the user to sign in and then redirect to the home page, as shown in Figure 3.

If the user does not have an account yet, the user can also create an account by clicking the “Sign up” button. This will direct the user to the sign-up page, as shown in Figure 2.

Note that even if the user does not sign in, the user can use the word searching feature. The user can open the search page by clicking the “TheALLDictionary” or “Search” on the navigation bar.

The screenshot shows the 'Sign up' page for 'TheALLDictionary'. The header includes the logo, navigation links (Search, Quiz, Card, Shop), and buttons for 'Sign Up' and 'Sign in'. The main content area is titled 'Sign up' with the subtitle 'Use The ALL Dictionary'. Below this is a form with the following fields:

- User ID: Enter your id
- Password: Enter your password
- Name: Enter your name
- Nickname: Enter your nickname
- Email Address: Enter your email address
- Phone Number: Enter your phone number

A blue 'Sign up' button is located at the bottom of the form. Below the button, there is a link: 'Already have an account? [Log in.](#)'

Figure 2. Sign-up page

This page lets the user register a new account. The user can create a new account once the user fills out the tabular form correctly and then click the “Create” button. Then, the system will create an account based upon the user’s inputs and then redirect the user to the Login page.

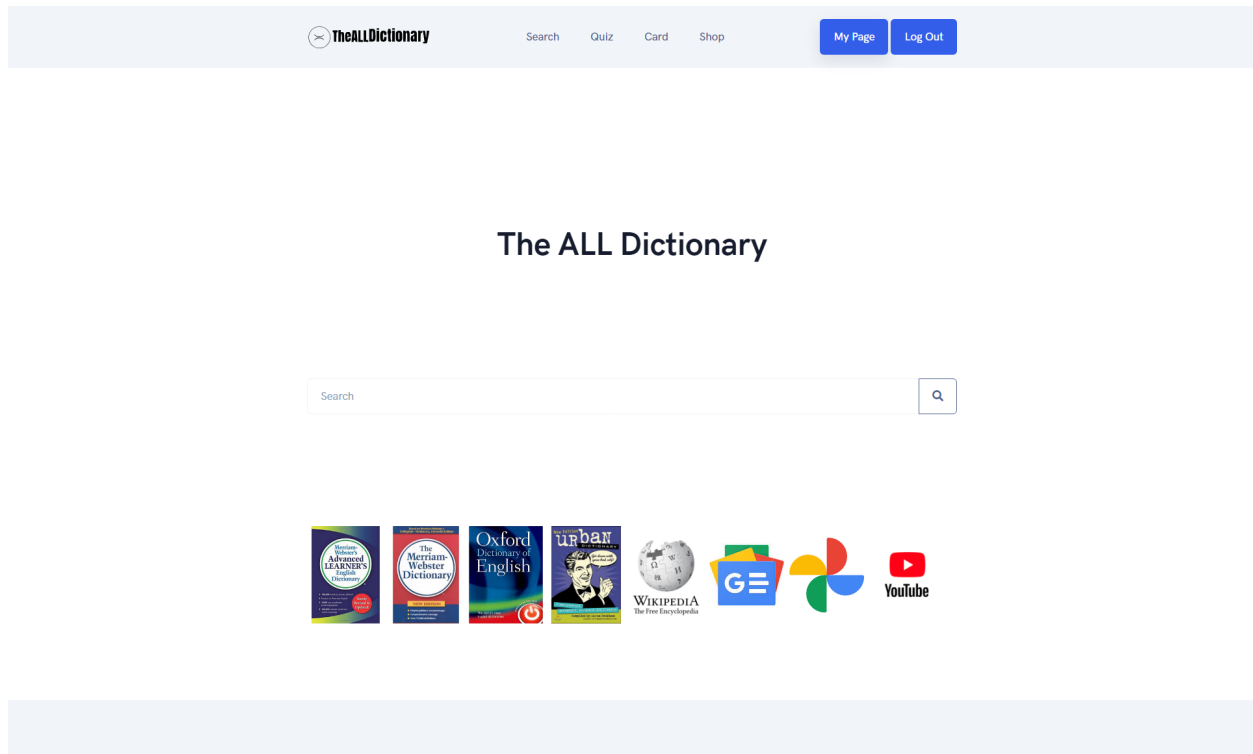




Figure 3. Search page

This is the first screen for the user who got into TheALLDictionary website. The user can open the search page by clicking the “Search” at the top tab bar.

Once the user has signed in, the “My Page” button and the “Sign out” button will appear in the upper right corner. If the user clicks the “My Page” button, the system will open the My page, as shown in Figure 6. If the user clicks the “Sign out” button, the system will sign out the user and then redirect it to the search page.

Note that the user does not have to sign in with its account to access the search page.

 [Search](#) [Quiz](#) [Card](#) [Shop](#) [My Page](#) [Log Out](#)



Merriam Webster's Learner
Definition1
definition2
definition3

Merriam Webster
Definition1
definition2
definition3

Oxford Dictionary
Definition1
definition2
definition3

Urban Dictionary
Definition1
definition2
definition3

Google News
Definition1
definition2
definition3

Google Photo
photo1
photo2
photo3


Youtube
video1
video2
video3

Change Dictionary Order

Copyright © CSE 416 The ALL Dictionary

Figure 3-1. Dictionary search result

Once the user clicks the search button, then redirect to the result page.

 Search Quiz Card Shop [My Page](#) [Log Out](#)

Change Dictionary Order

1. First Order

Open this select menu

2. Second Order

Open this select menu

3. Third Order

Open this select menu

4.

Open this select menu

5.

Open this select menu

6.

Open this select menu

7.

Open this select menu

8.

Open this select menu

Change

Copyright © CSE 416 The ALL Dictionary

Figure 3-2. Dictionary order change page

If a user clicks the button “Dictionary Order Change” at the bottom of 3-1, the user goes to the dictionary order change page. On this page, the user can change the order of dictionaries by combo box.

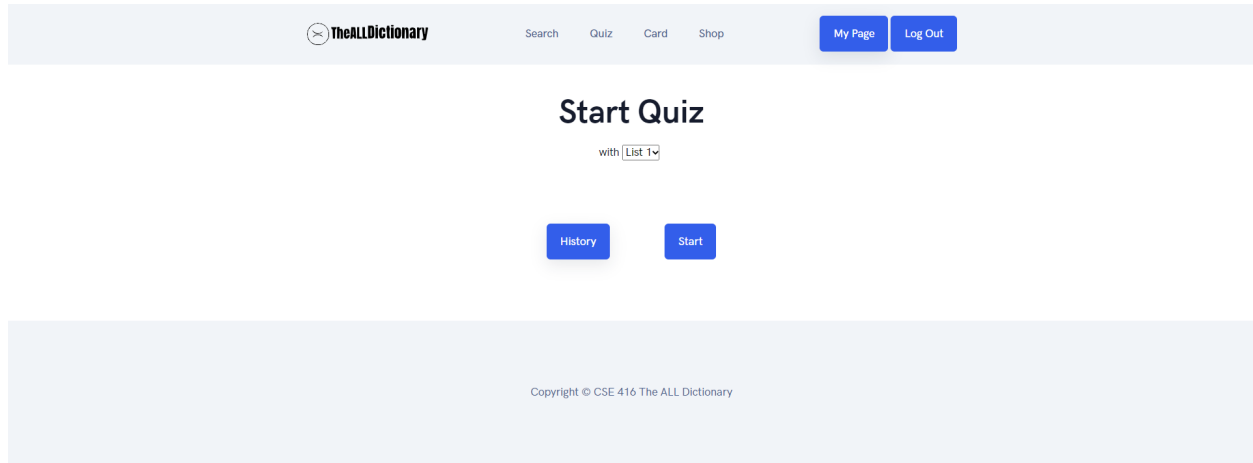


Figure 4. Start quiz page

The Start quiz page can be reached by clicking the “Quiz” button at the top tab bar. After the premium customer has made a list of note cards, he or she can select the list to let the system generate a quiz only based upon the note cards from the selected list. Then, the customer can click the “Start” button to start the quiz, as shown in Figure 4-2.

The screenshot shows a web interface for a quiz. At the top, there is a navigation bar with the logo 'TheALLDictionary' and links for 'Search', 'Quiz', 'Card', 'Shop', 'My Page', and 'Log Out'. The main heading is 'Question 10 of 10'. Below this, a box contains two multiple-choice options: '1. The process of writing and testing computer programs.' and '2. The planning of which television or radio programs to broadcast.' Underneath the options are four buttons: 'Programming', 'React', 'Active', and 'Database'. Below these are three buttons: 'Previous', 'Next', and 'Submit'. At the bottom, a footer contains the text 'Copyright © CSE 416 The ALL Dictionary'.

Figure 4-2. Sample multiple-choice question in the quiz

Here notice that the system has generated a sample multiple-choice question. Therefore, the customer has to choose a word from each of the four choices and then click the “Previous” and “Next” button to move to the next question. The “Submit” button will only show up when the customer has entered the last question. The left arrow shape button lets the customer go back to the previous question to review its answer.

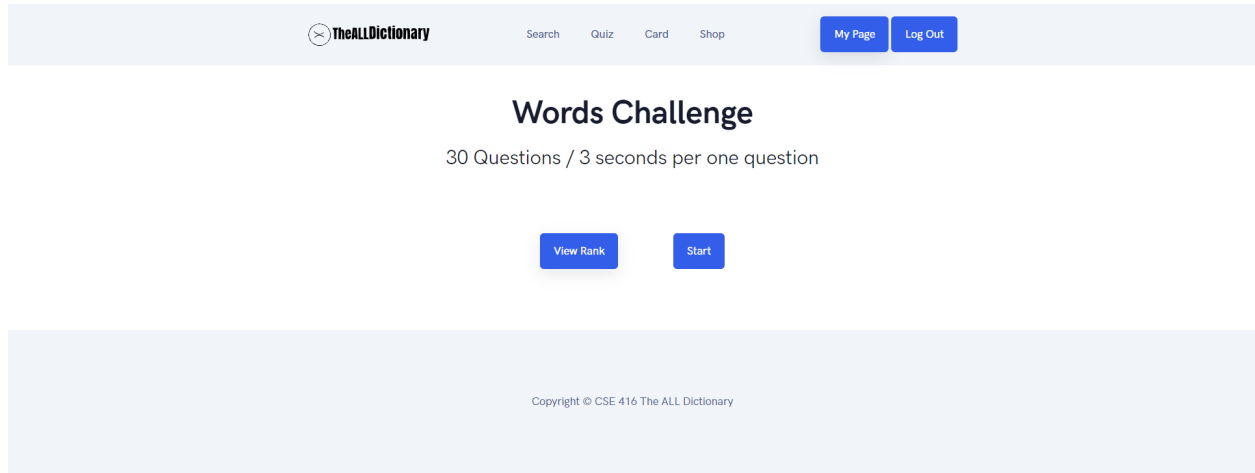


Figure 4-3. Start words challenge page

When the customer puts the mouse cursor on the “Quiz” button at the top of the screen, the drop-down menu will be popped up. From the drop-down menu, the customer can click the “Words Challenge” button to open the page, as shown in Figure 4-3.

Here clicking the “Start” button will open the page, as shown in Figure 4-4. Moreover, clicking the “View Rank” button will open the page, as shown in Figure 4-5.

The screenshot shows a web interface for a word challenge. At the top, there is a navigation bar with the logo 'TheALLDictionary' and links for 'Search', 'Quiz', 'Card', 'Shop', 'My Page', and 'Log Out'. The main heading is 'Question 15 of 30'. Below this, a timer indicates '3 seconds remaining'. A large rectangular box contains two numbered options: '1. The process of writing and testing computer programs.' and '2. The planning of which television or radio programs to broadcast.' Below the options are four buttons: 'Programming', 'React', 'Active', and 'Database'. The 'Programming' button is highlighted with a blue border. Below these buttons is a 'next' button. At the bottom of the page, a copyright notice reads 'Copyright © CSE 416 The ALL Dictionary'.

Figure 4-4. Sample multiple-choice question in a word challenge

Once the customer clicks the “Start” button on the page, as shown in Figure 4-3, he or she can begin the word challenge. Each question has a 3 seconds time limit to prevent premium customers from cheating. The customer needs to choose one word from the four options and click the “Next” button to submit its answer to the current question and move on to the next question. Note that if the customer goes over the time limit, then the clicked option will be automatically submitted as the answer to the system. In the case above, the word “Programming” will be automatically submitted as the answer. If there is not any option clicked, it means that the customer has not solved the problem on time. Therefore, the system will not give any points to this question.

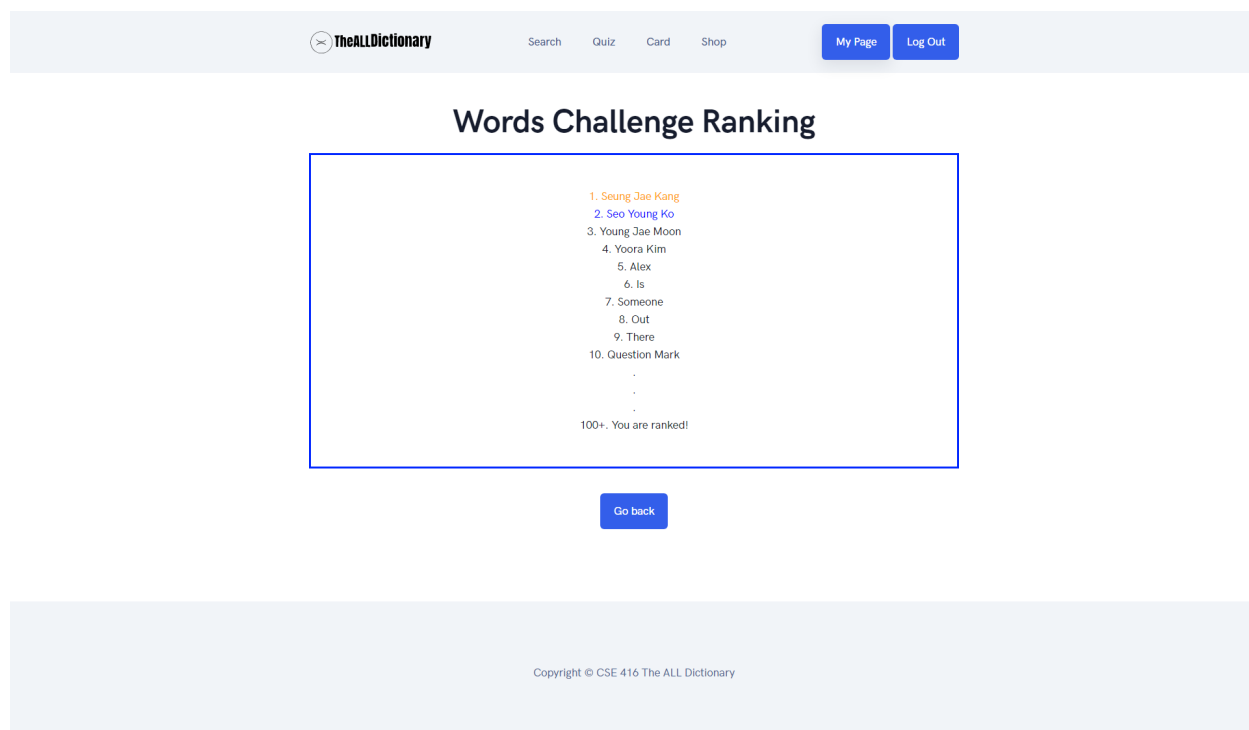


Figure 4-5. Rank page

The rank page can be opened by clicking the “View Rank” button on Figure 4-3. The ranking is based upon the scores of “Words Challenge” from each premium customer.

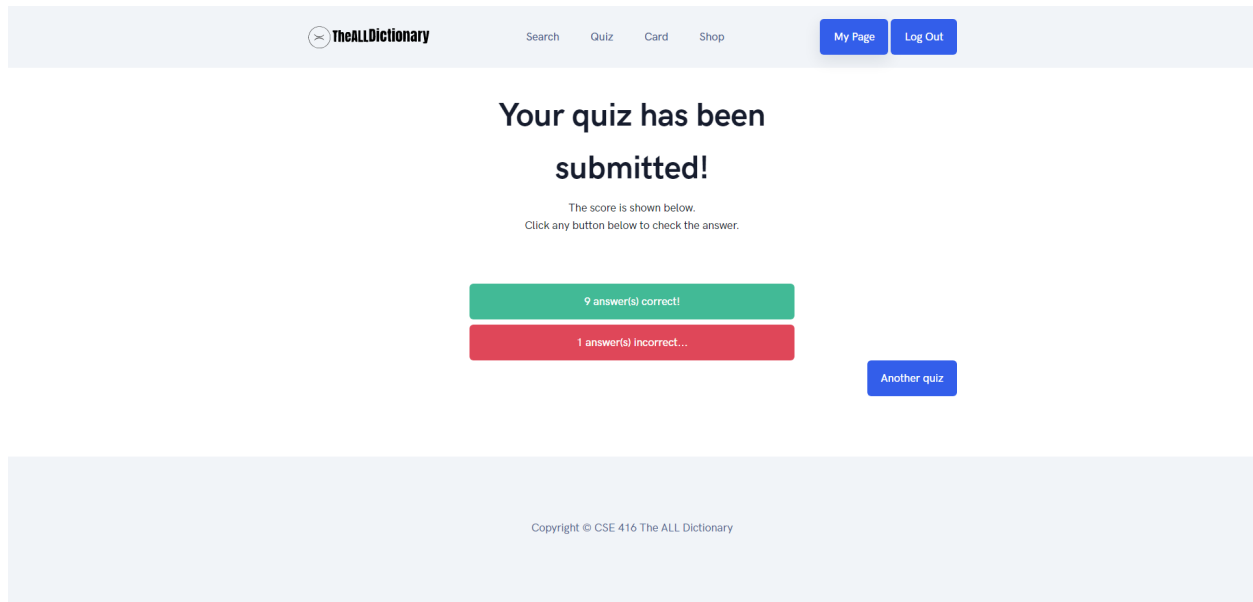


Figure 4-6. Quiz result page

Once the premium customer has finished and submitted the quiz on time or the time has gone over, he or she can check its results on the quiz result page immediately. Clicking on the red or green button will lead the user to the result page which looks like figure 4-8.

Note that the customer can take a new quiz by clicking the “Another Quiz” button. This will, in turn, open a page, as shown in Figure 4.

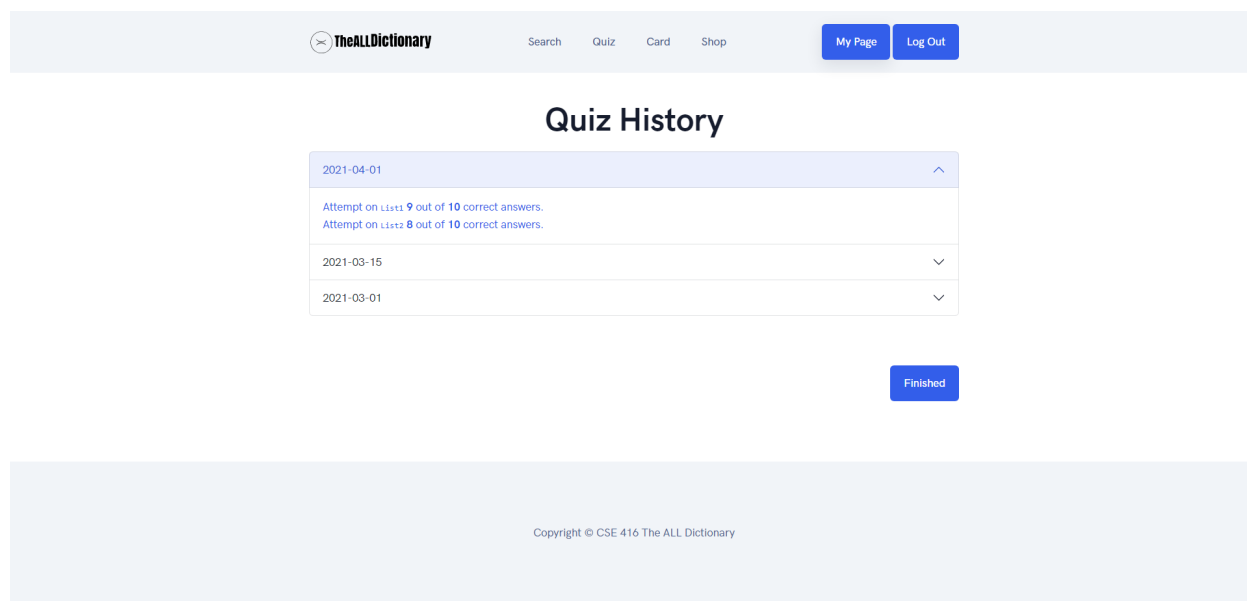


Figure 4-7. Quiz History page

The quiz history page can be opened by clicking the “History” button in Figure 4. On the quiz history page, the user can check his or her quiz attempts and the simple result it. If the user clicks the quiz result, the user can go to the quiz review page.

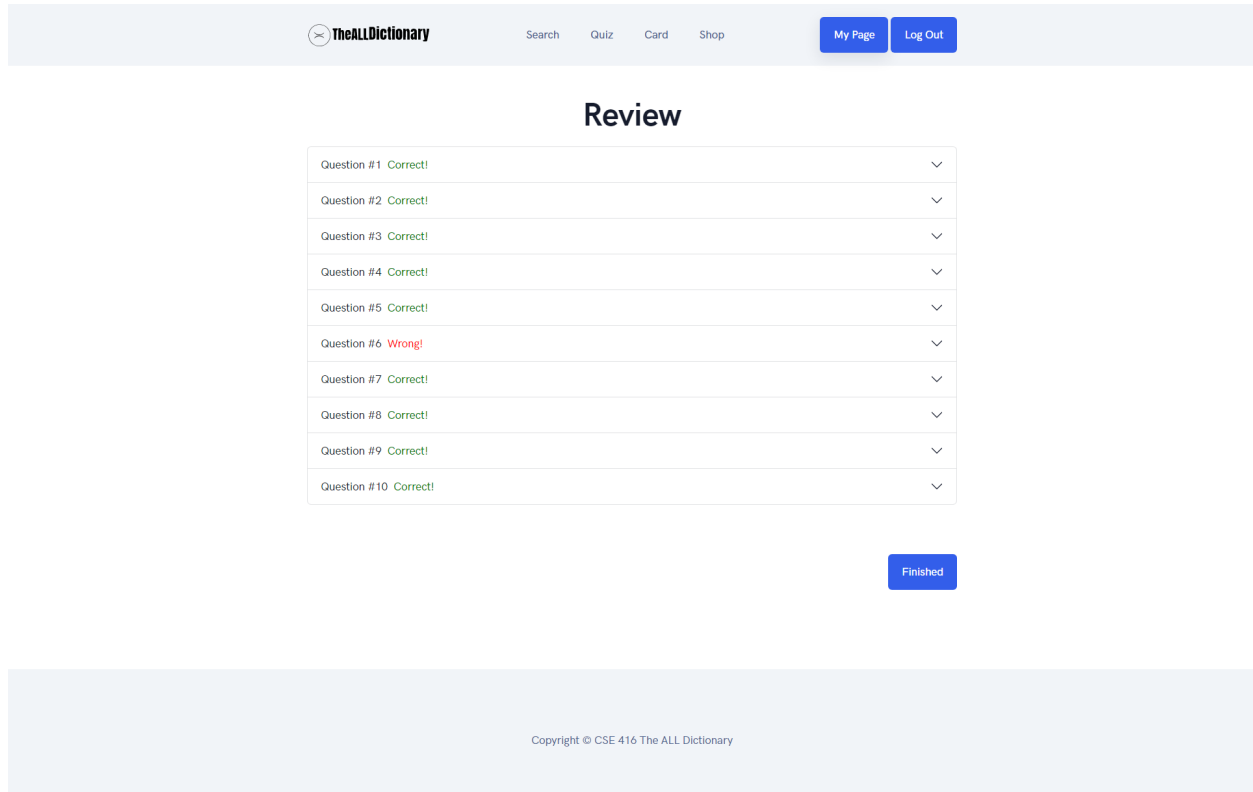


Figure 4-8. Quiz review page

From the quiz review page, the premium customer can check the correct answer for all questions. After checking all of the questions, the customer can click the “Finished” button at the top right corner to redirect to the quiz start page.

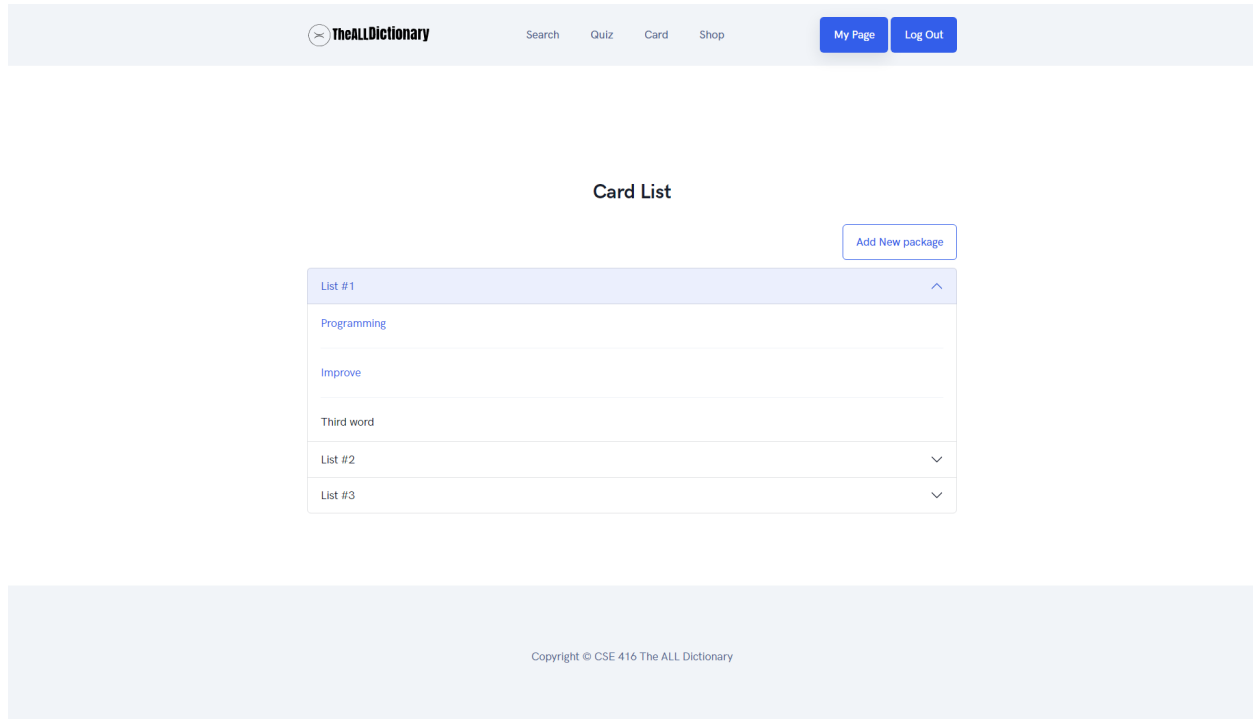


Figure 5. Card home page

The premium user can open the card page by clicking the “Card” button at the top tab bar. On this page, users can view their own card packages. The user can see note cards of the package by using a drop-down list. Also, the user can make a new card package by clicking the “Add New Package” button on the right-top side. If the user clicks the “Add New Package” button, a modal pops up to help users add a new package.

The screenshot shows the 'Make Your Own Cards!' interface. At the top, there's a navigation bar with 'TheALLDictionary' logo, 'Search', 'Quiz', 'Card', 'Shop', 'My Page', and 'Log Out' buttons. Below this, the 'Search History' section lists five items: 'Programming', 'Improve', 'Third checkbox', 'Fourth checkbox', and 'Fifth checkbox', each with a trash icon. An 'Add New Package' button is at the bottom of this list. The main 'Make Your Own Cards!' section has a 'Choose Image' button and a 'List #1' dropdown menu. The card form itself has fields for 'Write the Word' (containing 'Programming'), 'Part of Speech' (containing 'Noun'), and 'Write the Definition' (containing '1. The process of writing and testing computer programs.'). 'Cancel' and 'Save' buttons are at the bottom of the card form. The footer contains the text 'Copyright © CSE 416 The ALL Dictionary'.

Figure 5-1. Make card page

On this page, the user can make the notecard with his or her own preferences. For the first, the user has to set the list by clicking the list drop button at the right-top side of the notecard box. After the user decides the list, he or she can add an image to the notecard by clicking the “Choose Image” button at the left-top of the notecard box. Also, the user can make a new package of notecard by clicking the “Add New Package” button below the history box.

Then the user can type the word, part of speech, and definition of the word in the corresponding input boxes inside of the notecard box. After finishing the typing, the user can click the “Save” button to save the notecard to the list of note cards. Or the user can click the “Cancel” button to discard the notecard being created. When the user clicks the “Save” button, the system automatically displays the notecard that the user just created.

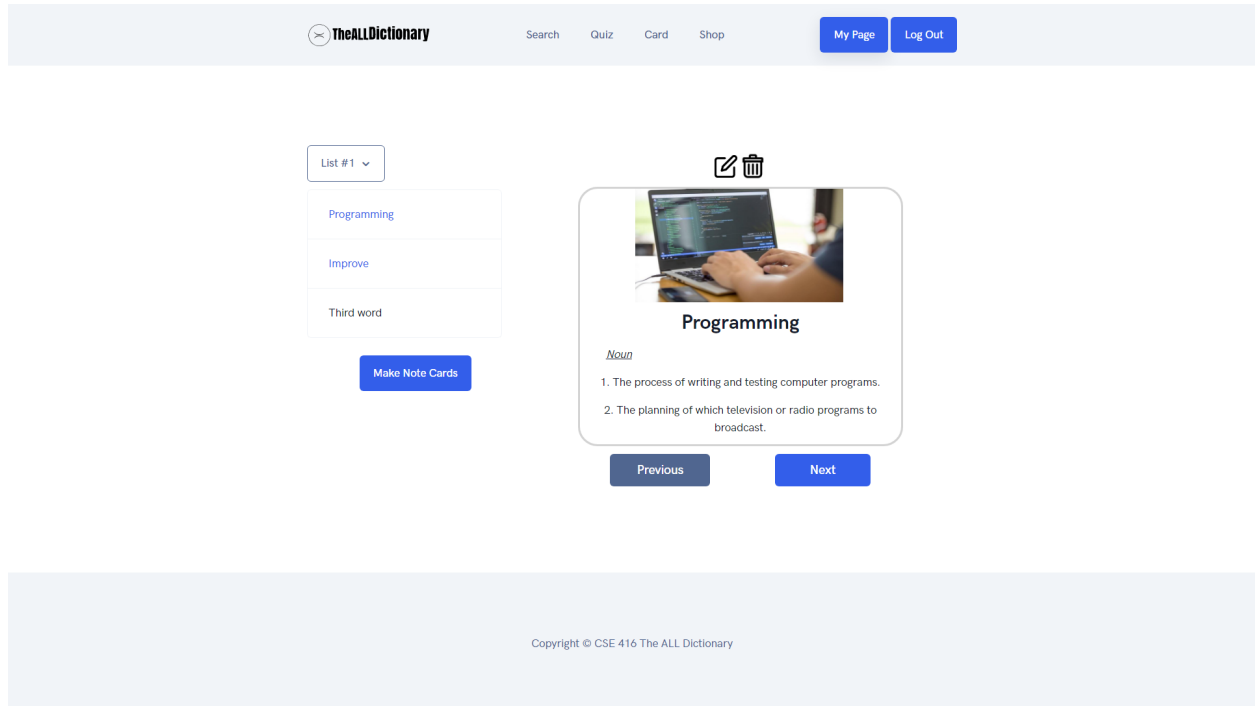


Figure 5-2. Sample of a saved note card

If the user clicks the name of the card package in Figure 5, the user can enter this page. The user can edit the notecard being selected by clicking the pencil-shaped button on the top. The user can also delete the notecard by clicking the garbage-shaped button. If the user wants to make another notecard, the user can go to the make card page by clicking “Make Note Card”. The user can click the “Previous” button to view the previous notecard from the list. Thus, the user can click the “Next” buttons to view the next notecard from the list. Also, the user can see another notecard package by using the drop-down button on the left-top side.

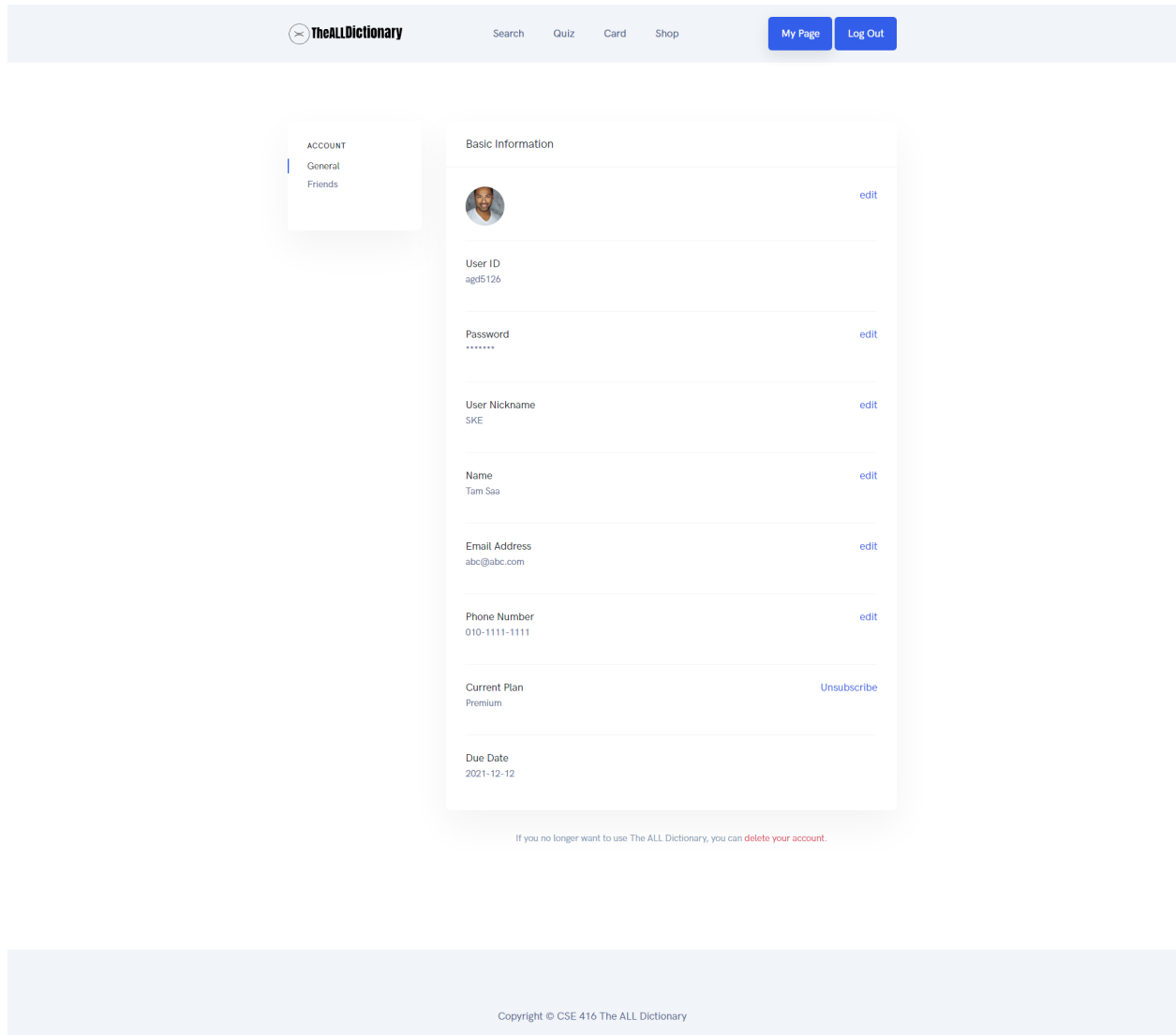


Figure 6. My Page

The signed-in user can open my page by clicking the “My Page” button at the top tab bar. The user can edit his or her personal information by clicking the “edit” button at the right of each menu. If the user is a premium user, the user can change its subscription status by clicking the “Unsubscribe” text. This will, in turn, open the subscription page. The user can delete its account by clicking “delete your account” at the bottom of my page.

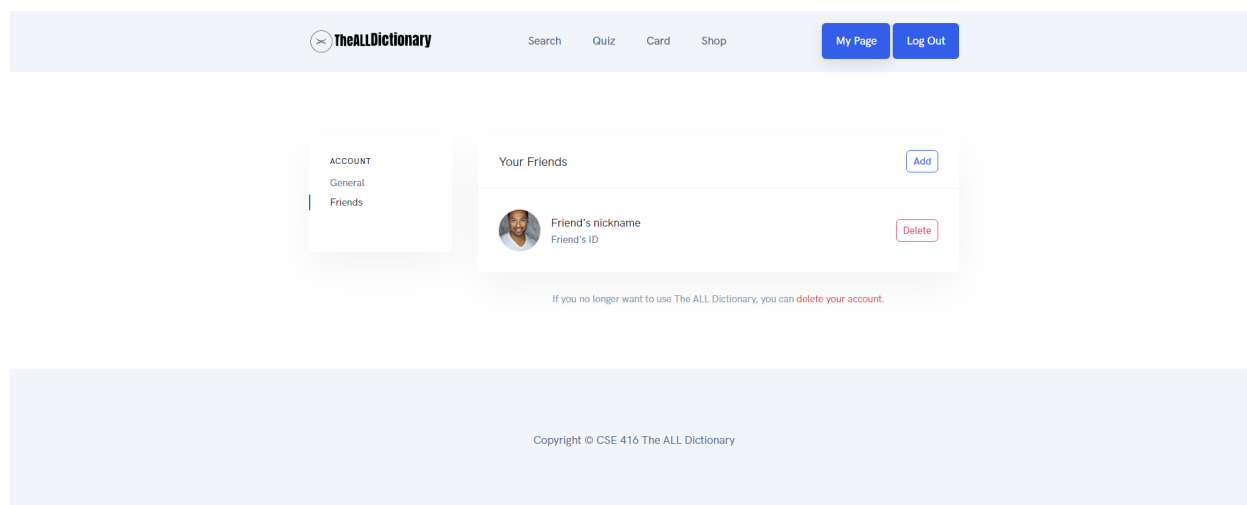


Figure 6-1. Friends page

On this page, the user can manage his or her friend list. The user can send friend requests by the “Add” button. Also, users can delete his or her friend from the friend list by clicking the “Delete” button.

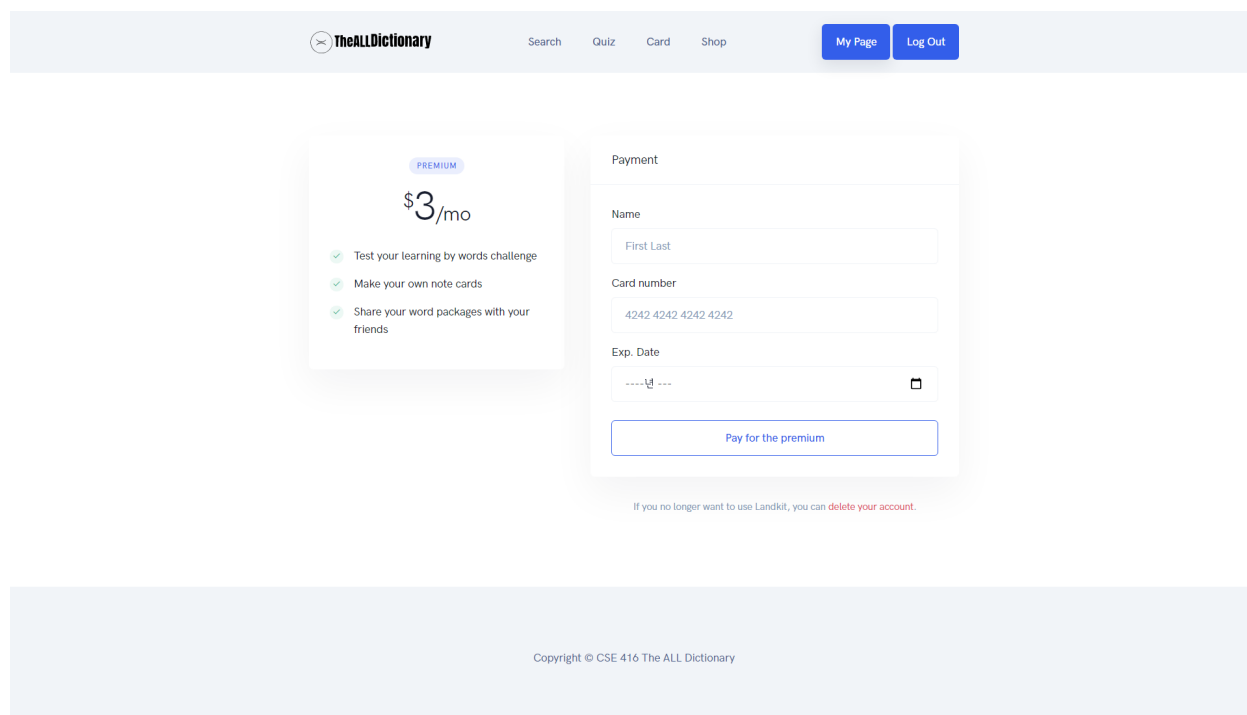


Figure 7. Subscription page

If the user clicks the “Shop” on the top bar, the subscription page will be opened. After the payment has been processed, the system will automatically redirect to My Page.

2.4 Non-functional requirements

2.4.1 Performance requirements

1. Creating, editing, saving, and sharing the notecards will take 1500 millisecond at maximum.
2. Updating the user's password will take 500 milliseconds at maximum after the new password has been validated and verified.
3. Updating the user's email address will take 500 milliseconds at maximum.
4. Updating the user's phone number will take 500 milliseconds at maximum.

2.4.2 Operating constraints

1. The system can be entirely operated only in the web browsers that YouTube, Google Images, Google News, and Wikipedia support. If the user's web browser does not support these websites, then the user will not effectively view contents from these websites.

2.4.3 Platform constraints

1. The system shall support Windows and Mac OS, Android OS, iOS, and iPad OS as they are widely used. Windows and Mac OS are the two most common operating systems for desktops and laptops. Thus, Android OS and iOS are the two most common operating systems for mobile devices, and iPad OS is one of the most common operating systems for tablet PCs.

2.4.4 Design constraints

Not applicable.

2.4.5 Reliability

1. The system will not make errors 99 percent of the time for each functionality unless there is an Internet disconnection.

2.4.6 Availability

1. The system shall be connected to the Internet to connect to the database and other third-party sources for operating all functionalities.
2. The search results from the Oxford English dictionary will be available 1000 times maximum per month, as the free version of the Oxford Dictionaries API only handles 1000 requests per month.
3. The search results from Google Image and Google News will be available 100 times per day, as the free version of Custom Search API only handles 100 requests per day.

2.4.7 Security

1. The password for each account shall contain at least nine letters, including at least one capital letter, one small letter, one number, and one special character.
2. The userID for each account shall be unique since the userID will be a primary key of a table in the database. Hence the system shall request the user or the admin to choose a different userID if the userID already exists. In this way, each user and admin will not be able to access other users' and admins' data.
3. If the user or the admin fails to sign in three consecutive times, the system shall disable the user or the admin to sign in for the next three minutes to prevent hacking.
4. The password of each account shall be saved in the encrypted format in the database by employing Werkzeug, a web application library developed by Flask. If a hacker maliciously gains illegal access to the database, the hacker needs to spend more time

figuring out the user's or the admin's actual password. As people tend to use the same password for multiple applications due to convenience, if the hacker then fails to decrypt the encrypted version of the password, the system thereby prevents additional hacking to the user's accounts for the other applications.

5. The system shall request the user or the admin to confirm its new password for verification if the user or the admin wants to change its password. If the new password is not confirmed correctly, the system shall not update the password in the database and alert the user to verify its new password again. Otherwise, the system shall update the user's or the admin's password in an encrypted format in the database.
6. The text entered in the password text-boxes shall always be invisible so that nobody can see the user's or the admin's password.

2.4.8 Modifiability

1. Changing the price for a subscription will require less than a minute since the only part of the code that needs to be changed is the price saved on the server.

2.4.9 Legal constraints

1. The system is limited to displaying YouTube videos that do not violate the government's laws to which the user currently belongs. Suppose the video is forbidden for political, ethical, or ideological reasons by the country that the user is currently accessing the system. In that case, the system cannot display the video on the search page.

2.4.10 Usability

1. The bottom tab bar is designed so that the user finds it easy to select for viewing the searched results from each source. At least nine out of ten users should not find any difficulty in choosing the source to view on the search page.

2.4.11 Portability

Not applicable.

3. System Architecture

3.1 Overview

3.1.1 Programming languages and DBMS used

Name of the Programming Language and DBMS	Version number	Reason
JavaScript	ECMAScript 2020	<ul style="list-style-type: none"> JavaScript improves the user experience by allowing the pages to be interactive. ECMAScript 2020 is the latest stable version of JavaScript.
MySQL	MySQL Community Edition 8.0.23	<ul style="list-style-type: none"> MySQL Community Edition is free to use. MySQL is the most popular RDBMS. Relational database allows normalization to reduce data redundancy and improve data integrity. All the members in our team learnt MySQL while taking CSE 305 in Spring 2020. v8.0.23 is the latest stable version of MySQL. Amazon RDS supports MySQL Community Edition version 8.0.
Python	3.9.4	<ul style="list-style-type: none"> v3.9.4 is the latest version of Python. According to Flask documentation, the latest version of Python is recommended. Reference: https://flask.palletsprojects.com/en/1.1.x/installation/#python-version

3.1.2 Libraries and frameworks used

Name of the library or framework	Version number	Reason
Bootstrap	v5.0.0-beta 3	<ul style="list-style-type: none"> • Bootstrap saves time to stylise the website. • jQuery is removed for Bootstrap 5 which, in turn, increased performance compared to when using Bootstrap 4. • Bootstrap 5 supports responsive font sizes. • Bootstrap Navbar has been optimised for Bootstrap 5. • We can integrate Bootstrap and Flask using the Flask-Bootstrap package.
Flask	1.1.2	<ul style="list-style-type: none"> • Flask is a microframework that enhances the performance of a web app. • v1.1.2 is the latest stable version of Flask.
WerkZeug	1.0.1	<ul style="list-style-type: none"> • WerkZeug, a web application library developed by Flask, will be used to encrypt the passwords of all accounts created. • v1.0.1 is the latest stable version of WerkZeug.
React*	17.0.2	<ul style="list-style-type: none"> • React will be employed only for front-end functionalities that cannot be implemented by Bootstrap. • v17.0.2 is the latest stable version of React.

3.1.3 APIs used

Name of the API	Version Number	Reason
Merriam-Webster's Learner's Dictionary with Audio	v3	<ul style="list-style-type: none"> • ‘Merriam-Webster's Learner's Dictionary with Audio’ will be used to display search results obtained from Merriam-Webster Learner’s Dictionary. • v3 is the latest stable version of ‘Merriam-Webster's Learner's Dictionary with Audio.’
Merriam-Webster's Collegiate® Dictionary with Audio	v3	<ul style="list-style-type: none"> • ‘Merriam-Webster's Collegiate® Dictionary with Audio’ will be used to display search results obtained from Merriam-Webster Dictionary. • v3 is the latest stable version of ‘Merriam-Webster's Collegiate® Dictionary with Audio.’
Oxford Dictionaries API	2.5	<ul style="list-style-type: none"> • Oxford Dictionaries API will be used to display search results obtained from Oxford English Dictionary. • Note that we will use the ‘Prototype’ version for Oxford Dictionaries API, since this is the free version. • v2.5 is the latest stable version of Oxford Dictionaries.
Custom Search API	N/A	<ul style="list-style-type: none"> • This is an API that needs to be enabled to use the ‘Google-Images-Search’ Python package.
YouTube Data API	v3	<ul style="list-style-type: none"> • YouTube Data API will be used to show relevant YouTube videos of a word that the user has searched. • v3 is the latest stable version of YouTube Data API.
INIAPI*	v1	<ul style="list-style-type: none"> • KG Inicis is one of the easiest payment systems in Korea to be implemented. • INIAPI supports TLS v1.2. • v1 is the latest stable version of INIAPI. • We will use this API if we have additional

		time to work on the stretch goals, as implementing a KG Inicis payment system is a stretch goal.
--	--	--

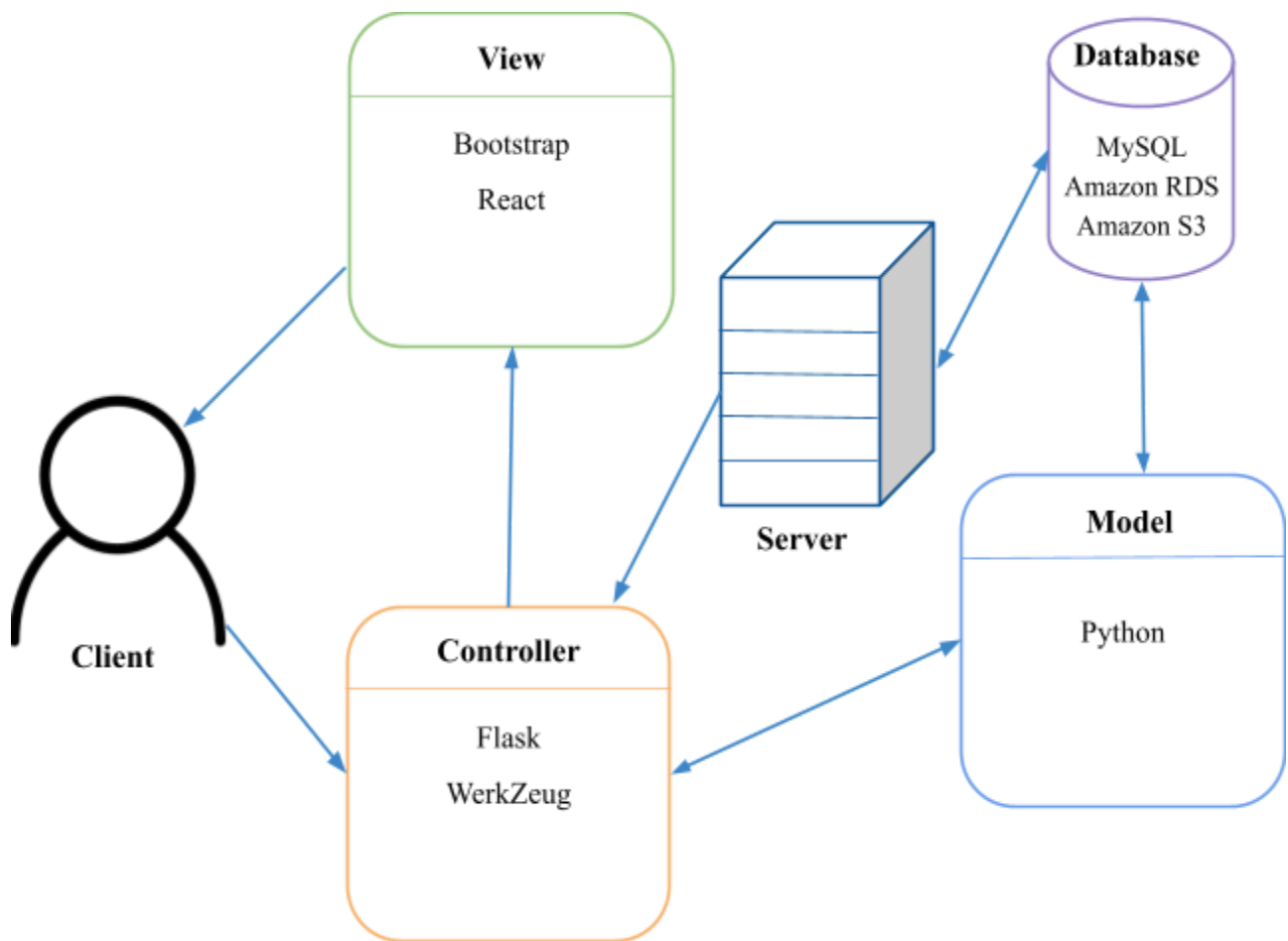
3.1.4 Third-party packages used

Name of the package	Version number	Reason
Flask-Bootstrap	3.3.7.1	<ul style="list-style-type: none"> Flask-Bootstrap allows the codes for Flask and the codes for Bootstrap to be connected. v3.3.7.1 is the latest stable version of Flask-Bootstrap.
MySQL Connector/Python	8.0	<ul style="list-style-type: none"> MySQL Connector/Python will be used to connect between Python code and MySQL server. v8.0 is the latest stable version of MySQL Connector/Python. This is the only version that supports Python 3.9.
Flask-SocketIO*	5.0.1	<ul style="list-style-type: none"> Flask-SocketIO can be used to implement chat functionality between premium customers. v5.0.1 is the latest stable version of Flask-SocketIO.
oxforddictionaries	0.1.101	<ul style="list-style-type: none"> This is a python package for showing results from Oxford English Dictionary. In other words, this is a python wrapper for the Oxford Dictionary API. Hence it is easy to integrate this package with Flask. v0.1.101 is the latest stable version of oxforddictionaries.
urbandict	0.6.1	<ul style="list-style-type: none"> This is a python package for showing results from Urban Dictionary. Hence it is easy to integrate this package with Flask. v0.6.1 is the latest version of urbandict.
Wikipedia-API	0.5.4	<ul style="list-style-type: none"> This is a python package for showing results from Wikipedia. Hence it is easy to integrate this package with Flask. v0.5.4 supports Python 3.4 or higher.

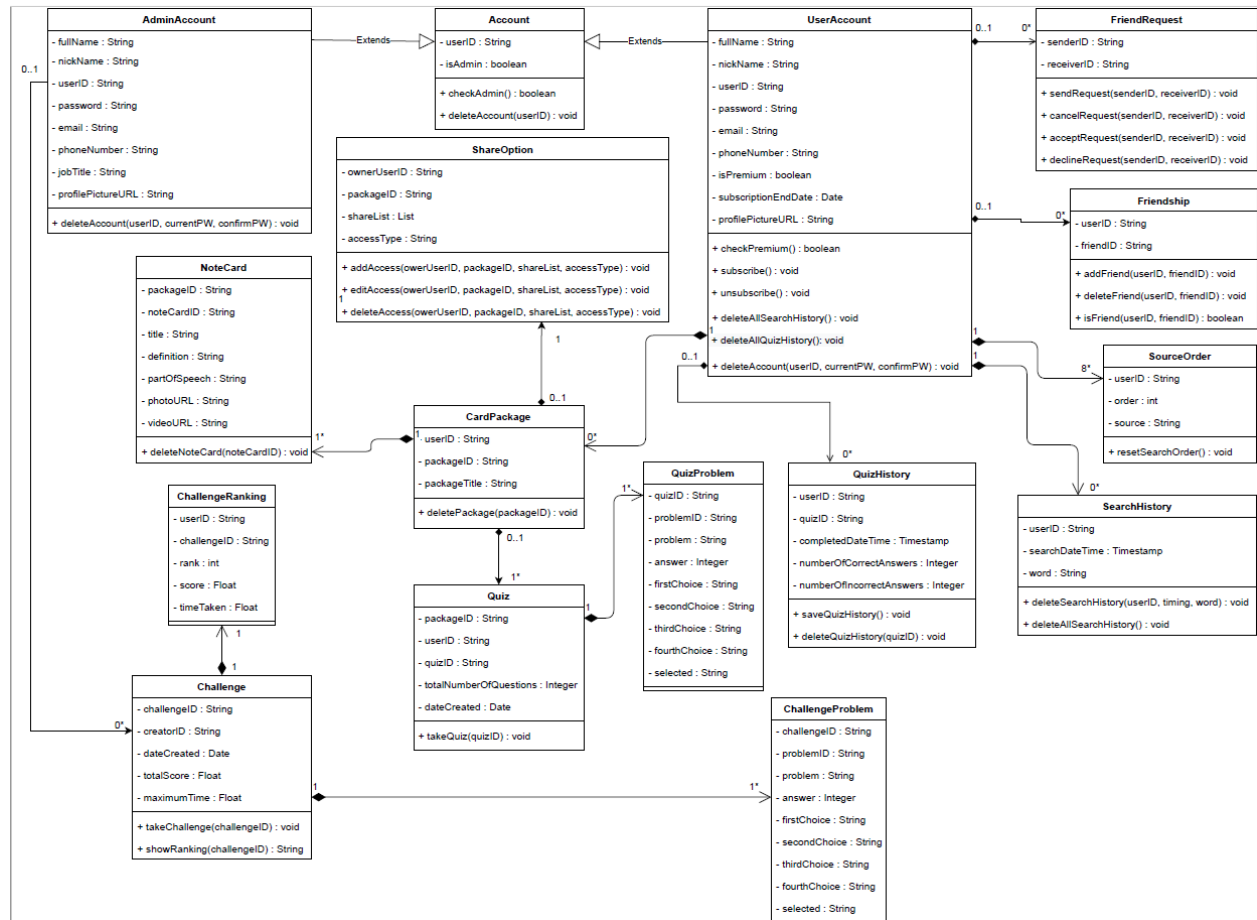
		<ul style="list-style-type: none"> • v0.5.4 is the latest stable version of Wikipedia-API
Google-Images-Search	1.3.8	<ul style="list-style-type: none"> • This is a python package for showing search results from Google Image. • Hence it is easy to integrate this package with Flask. • v1.3.8 is the latest stable version of Google-Images-Search
GoogleNews	1.5.7	<ul style="list-style-type: none"> • This is a python package for showing search results from Google News, • Hence it is easy to integrate this package with Flask. • v1.5.7 is the latest stable version of GoogleNews.
youtube-data-api	0.0.20	<ul style="list-style-type: none"> • This is a python package for showing search results from YouTube. • In other words, this is a python client for YouTube Data API (v3) • Hence it is easy to integrate this package with Flask. • v0.0.20 is the latest stable version of youtube-data-api.

3.1.5 Design patterns

We will follow the model-view-controller (MVC) design pattern and client-server pattern to implement our project. Here is the diagram that illustrates our design pattern:



3.2 UML Class Diagrams

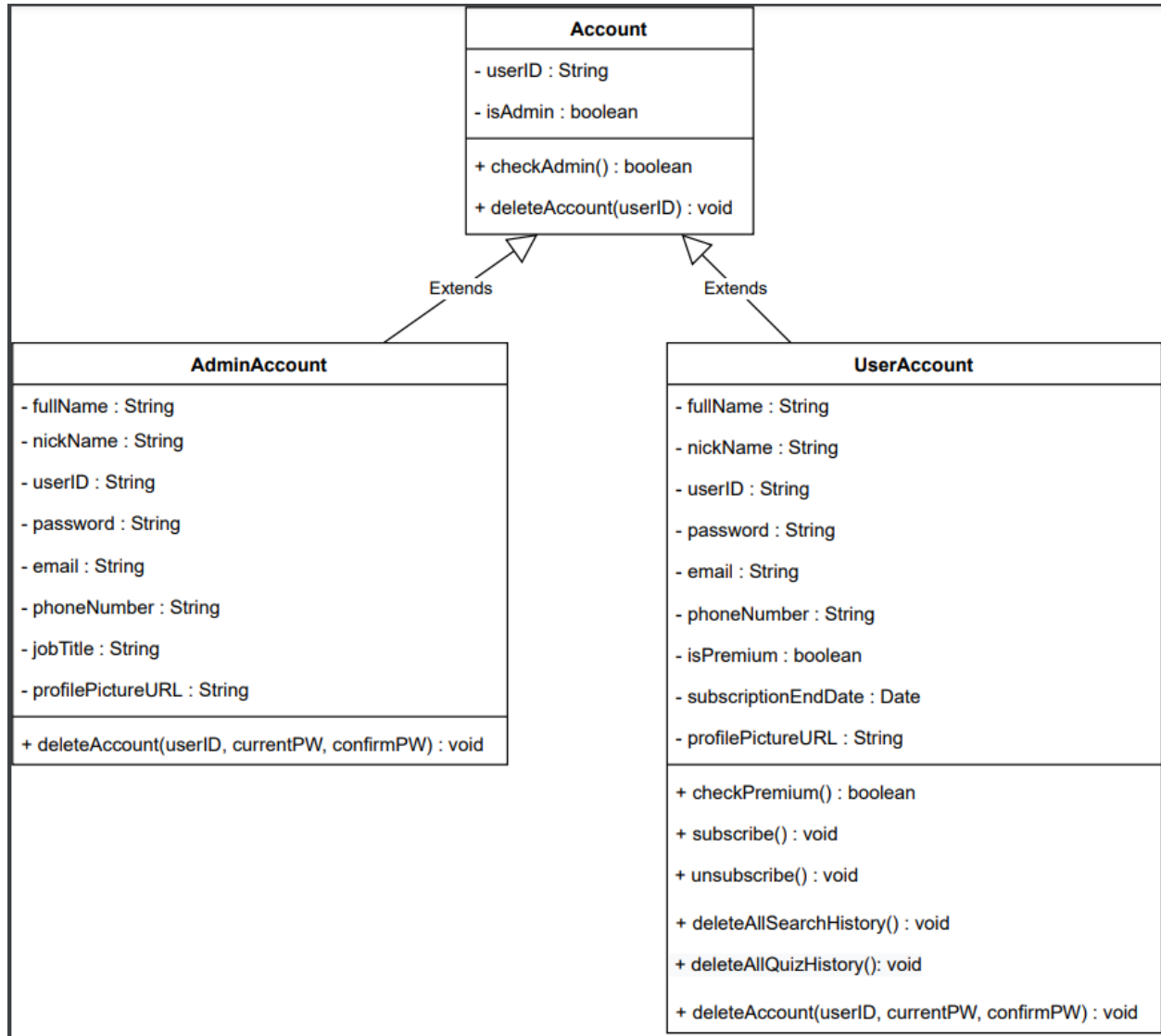


Here is the overall Unified Modeling Language (UML) Class Diagram of TheALLDictionary. Note that we have changed the name of the attributes from ‘date’ to ‘dateCreated’ for Challenge, CardPackage and Quiz classes. As ‘date’ is a type of data in MySQL, we decided to change the name of the attribute as dateCreated for MySQL code. Hence we decided also to change the name of the attribute for the UML Class Diagram as well.

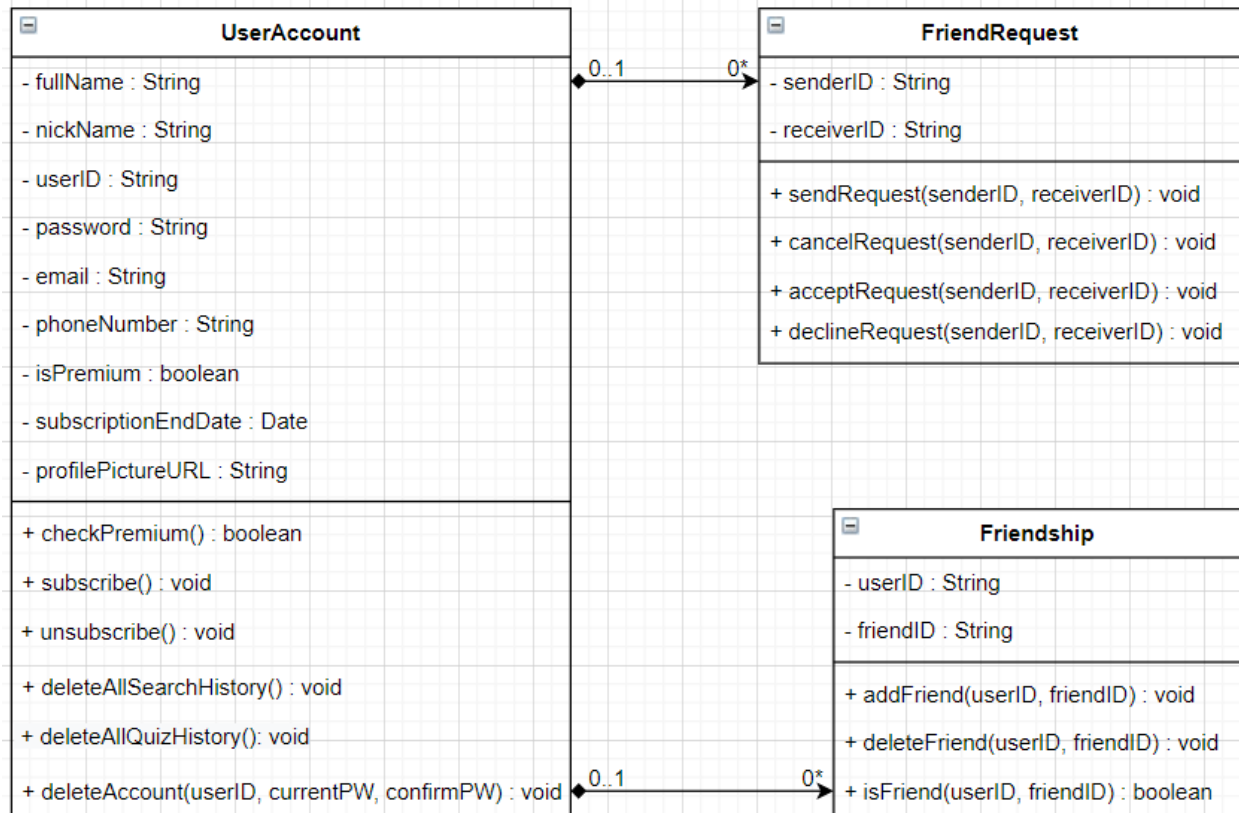
Moreover, we have removed the totalScore attribute for the Quiz table and replaced it with ‘totalNumberOfQuestions.’ Thus, we have removed the score attribute for HasQuizHistory table and added the following two attributes in this table: ‘numberOfCorrectAnswers’ and

`'numberOfIncorrectAnswers.'` This will reflect the user's performance for each quiz more precisely.

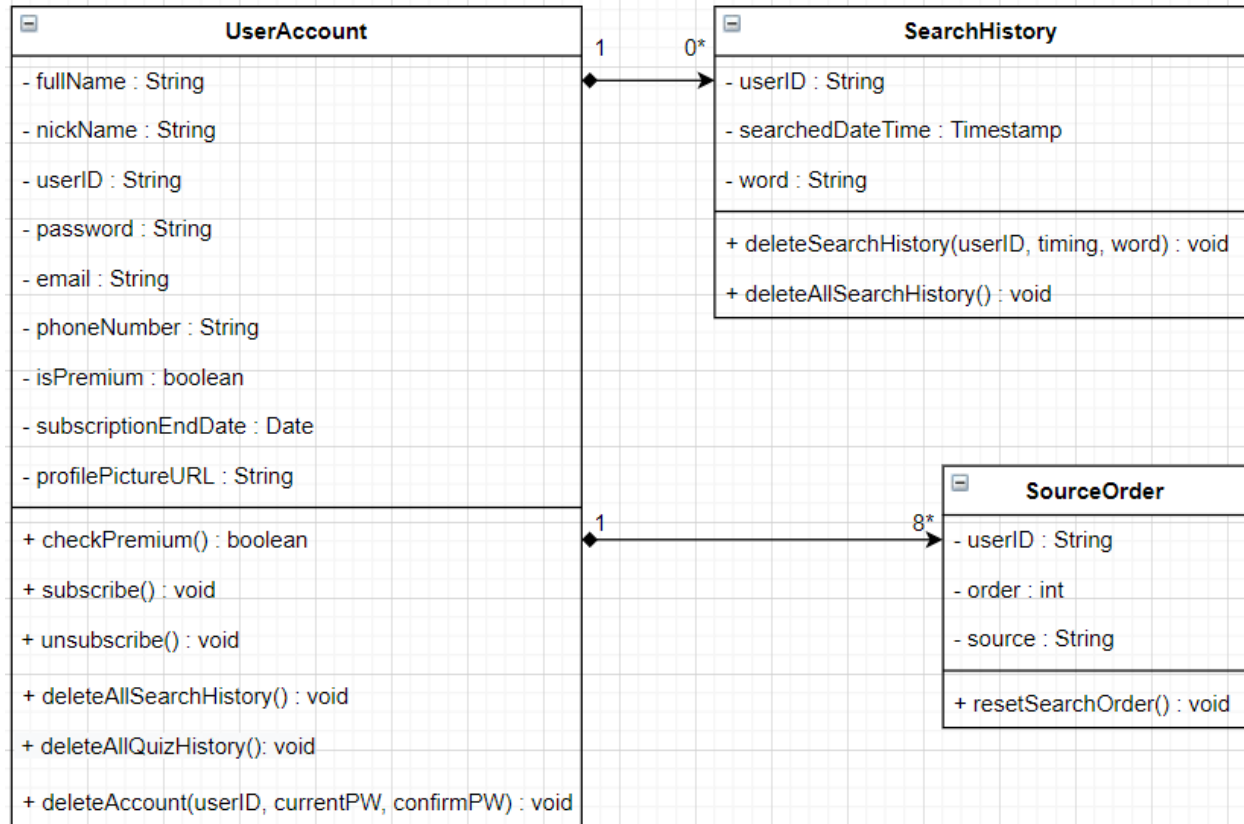
Since it is difficult to read the whole diagram, we decided to break this into multiple diagrams for easier viewing.



There are two main types of accounts in TheALLDictionary. One is the account for the administrator, and the other is the account for users. Hence we have created a superclass for the Account, and AdminAccount and UserAccount extend this superclass. Thus, since the users can be differentiated into free users and premium customers, we have added a boolean attribute, `isPremium`, to differentiate each user's type.

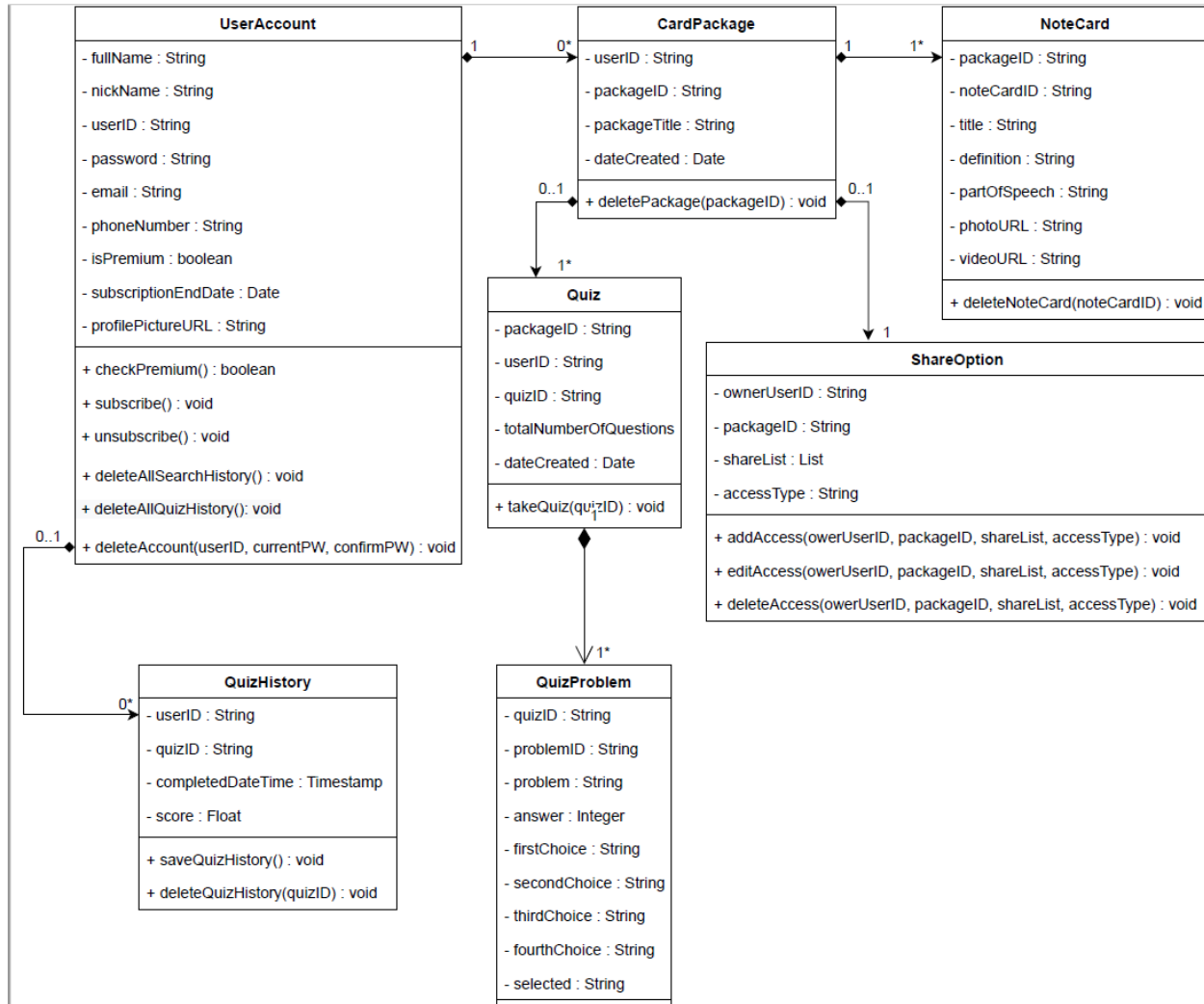


Adding a friend is a premium feature in TheALLDictionary. Hence not every user can send a friend request and thereby make a friend with other users. Thus, there is no limit with sending the friend request and making the friend. Hence there is a 0..1 to 0* association between UserAccount and FriendRequest, and UserAccount and Friendship.



If the user looks up a word, its search history is saved to the database. Hence there is a 1 to 0* association between UserAccount and SearchHistory.

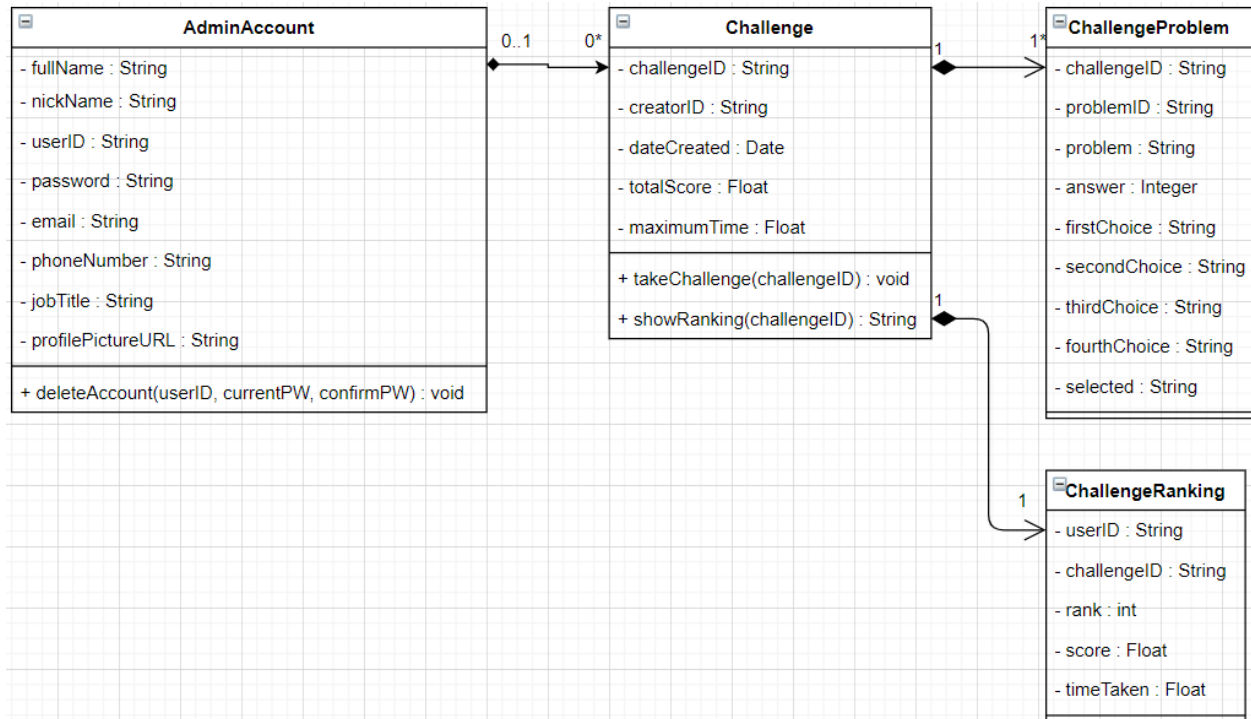
Thus, TheALLDictionary displays search results from eight different sources: Merriam-Webster's Learner's dictionary, Merriam-Webster dictionary, Oxford English Dictionary, Urban dictionary, Wikipedia, Google Images, Google News, and YouTube. And we may add more sources if we have extra time during the project. Hence there is a 1 to 8* association between UserAccount and SearchHistory.



All the users can create a package of note cards. Hence there is a 1 to 0* association between UserAccount and CardPackage and a 1 to 1* association between CardPackage and NoteCard.

Nonetheless, only premium users will be able to take quizzes that consists of one or more problems. Hence there is a 0..1 to 1* association between CardPackage and Quiz, and UserAccount and QuizHistory. Thus, there is a 1 to 1* association between Quiz and QuizProblem.

Moreover, only premium users are allowed to share their note cards. Hence there is a 0..1 to 1 association between CardPackage and ShareOption.



Only administrators can create word challenges. Hence there is a 0..1 to 0* association between AdminAccount and Challenge.

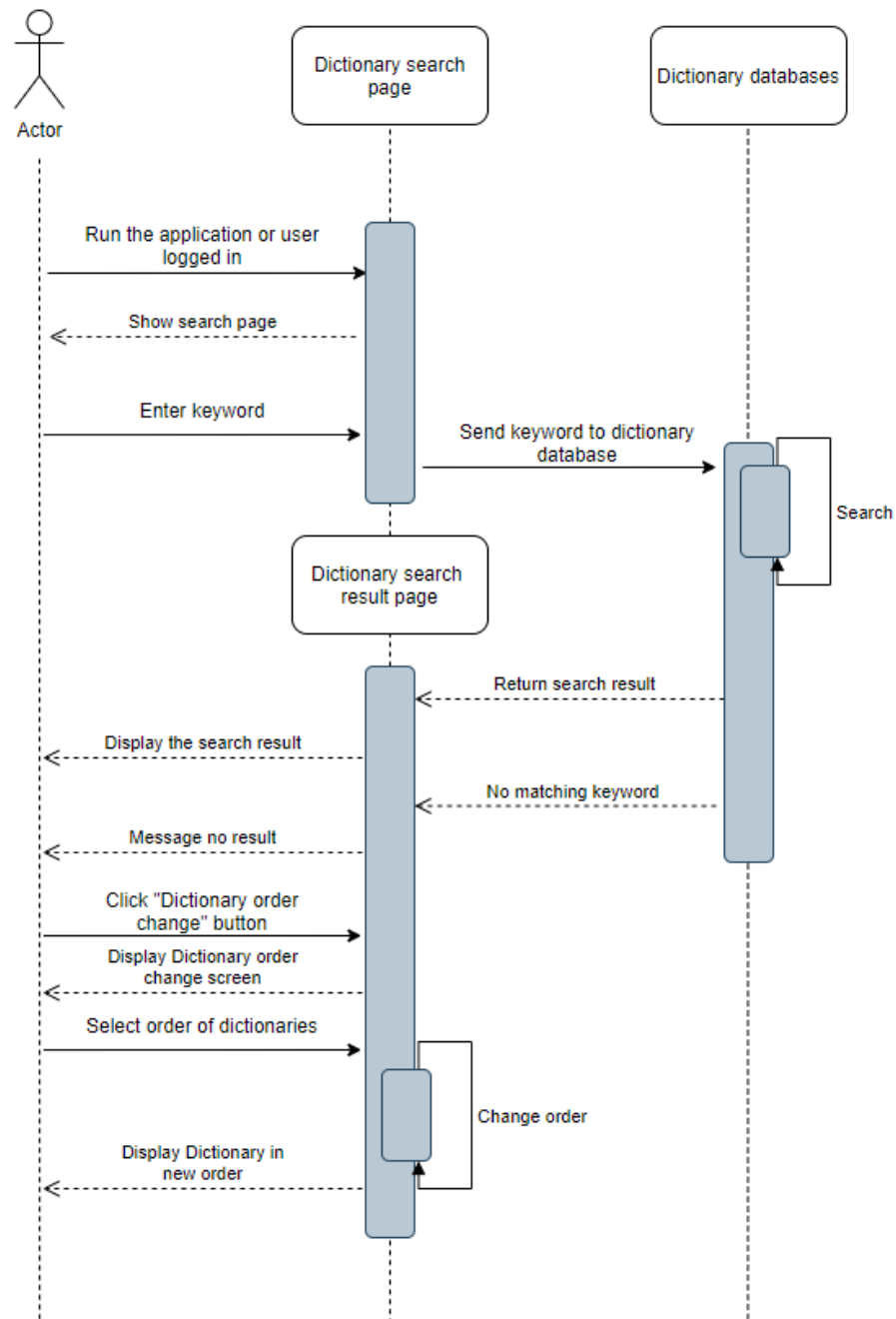
Moreover, these word challenges have one or more problems. Hence there is a 1 to 1* association between Challenge and ChallengeProblem.

The ranking is measured separately for each challenge. Hence there is a 1 to 1 association between Challenge and ChallengeRanking.

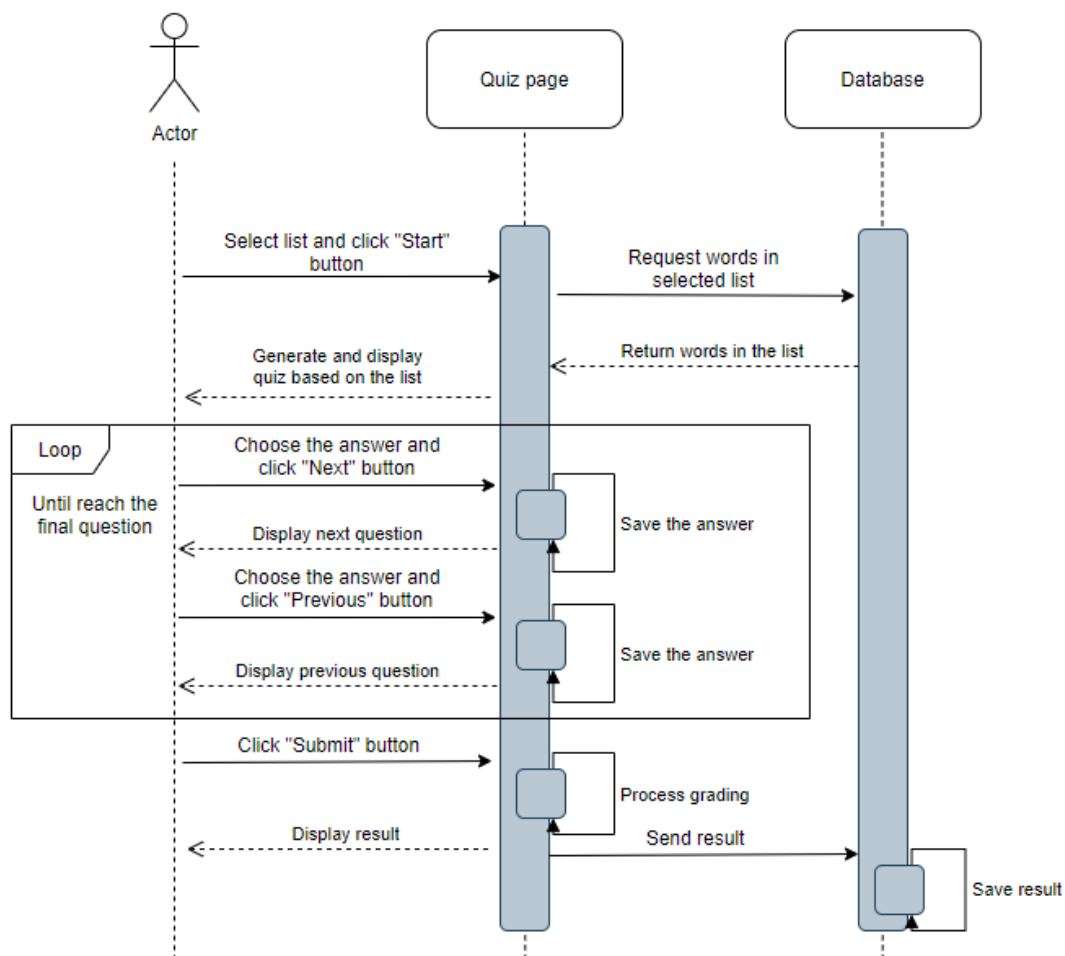
Note that each participant's ranking is measured in terms of score and time taken for each challenge.

3.3 UML Sequence Diagrams

3.3.1 Dictionary search page



3.3.2 Quiz page



3.4 API Design

Please refer to [TheALLDictionary-API Design](#) to view the list of every API our backend server will make available to our frontend client to use.

3.5 Deployment

We decided to use Heroku for deploying our application, as it is free to deploy a Flask application to Heroku. Thus, we will use Amazon RDS with MySQL for our relational database storage and Amazon S3 as simple object storage for static assets such as images and videos.

3.6 Code Conventions

3.6.1 Python Code Conventions

Since our back-end programming will be done using Flask, which is a microframework built-in Python, we will generally follow [PEP 8 -- Style Guide for Python Code](#) for our Python code convention.

Here note that we will use an upper camel case for classes (e.g. 'AdminAccount'). By contrast, we will use a lower camel case for attributes (e.g. 'fullName'). Besides, we will make classes singular. We will also make attributes singular unless they are a collection. Thus, we will use nouns for class and attribute names, but use verbs for method names. We will use application domain terms instead of programming terms for all names.

Moreover, we will mainly use block comments instead of inline comments for implementing our project. This is because block comments are more suited for explaining why the code does what it does, while inline comments are more suited for simply explaining what the code does.

Furthermore, we will use a tab as indentation instead of 4 spaces. This is because mixing tabs and spaces is not allowed for indentation in Python 3.

3.6.2 JavaScript Code Conventions

In the front-end programming part, we decided to use JavaScript. So, we will generally follow the [Google JavaScript Style Guide](#) as our JavaScript code convention.

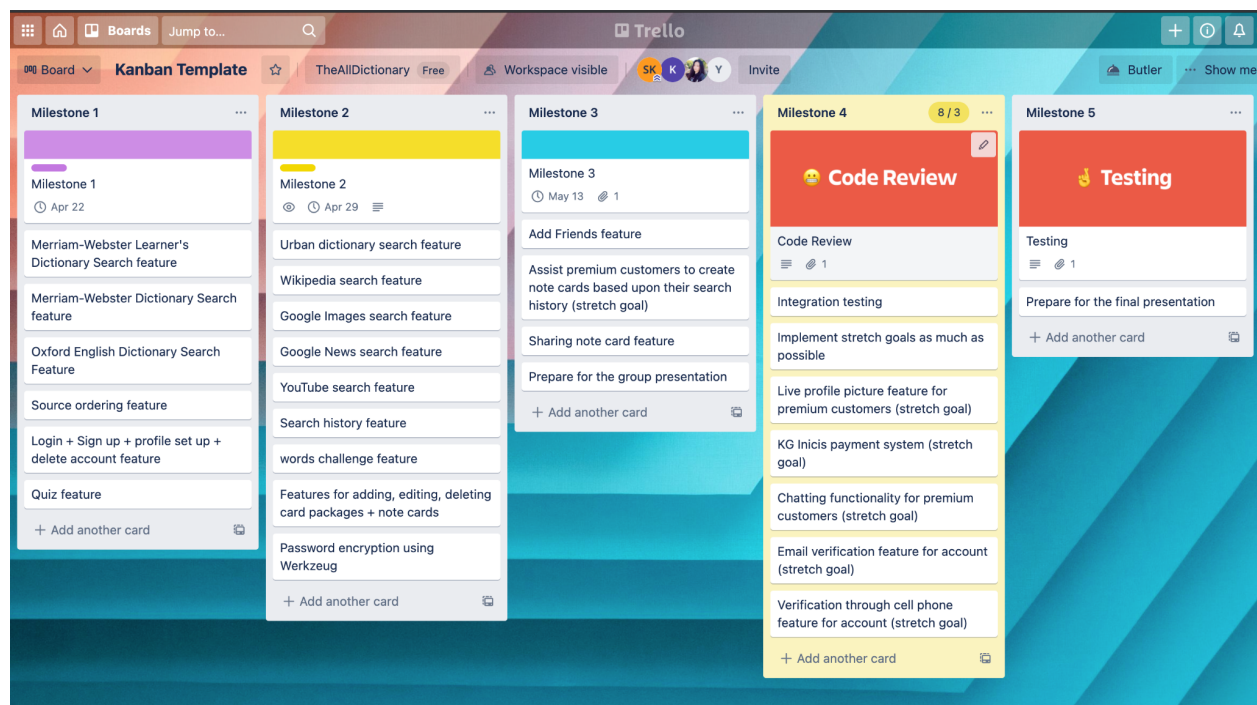
Here notice that we will use a lower camel case for module import names, package names, method names, non-constant field names, parameter names, and local variable names. By contrast, we will use an upper camel case for class names, enum names, and reference types.

In addition, the size of the tab is 2 spaces. Each time a new block or block-like construct is opened, the indent increases by two spaces.

4. Schedule

Our aim is to finish all of the functional requirements by milestone 3, which is the date for the beta release. Thus, we will start integration testing of these functionalities from milestone 4, and work on the stretch goals during milestone 4. In milestone 5, we will only work on testing each functionality and debugging the code. During this stage, no new functionality will be introduced and implemented.

For each milestone, each of us will do a code review which incorporates checking code conventions and formatting.



5. Contributions

Name	Contributions & Date Completed (DD/MM/YY)
Young-jae Moon	<ul style="list-style-type: none"> ● Formatted this document. ● Copied and pasted the materials from Chapter 1 and Chapter 2 of our SRS document except for the section for the UI mockups. (05/04/21) <ul style="list-style-type: none"> ○ Copied and pasted the UI mockups as Seoyoung told me that it is a good idea to update the UI mockups. (06/04/21) ● Mentioned word challenge feature in Section 1.1 (05/04/21) ● Revised Section 1.4 User Feedback. (05/04/21) <ul style="list-style-type: none"> ○ Mentioned at what points will feedback from people outside your team be most useful and how we will get that feedback. ● Added more terms and their definitions for Section 1.6 (10/04/21) ● Moved the survey results to the Appendix. (05/04/21) ● Modified the functional requirements for Section 2.1. (11/04/21) <ul style="list-style-type: none"> ○ Added functional requirements for administrator accounts. ○ Modified the requirements based upon the feedback given from the professor for improvement. ○ Edited the necessary + stretch goals. <ul style="list-style-type: none"> ■ Removed the stretch goal of implementing a payment system through PayPal. ■ Instead, added a stretch goal to support live photos to be used for profile pictures. ● Edited the wording for the precondition for section 2.2.7. (05/04/21) ● Edited the performance requirements for section 2.4.1. (05/04/21) <ul style="list-style-type: none"> ○ Specified “negligible amount of time” into milliseconds. ● Added some descriptions for Section 3.5 (05/04/21) ● Added UI design update section for the PPT (09/04/21) ● Completed Section 3.5 (09/04/21) ● Completed the code convention for the back-end programming for Section 3.6 (09/04/21) ● Updated the MySQL code based on the comments received from the Professor for the last assignment + added ‘IF NOT EXISTS’ condition for all table creations, and reuploaded the code into our GitHub repository. (11/04/21) ● Completed Section 3.2 based on the comments received from the Professor for the last assignment, and reuploaded to Section 3.2. (10/04/21) ● Completed Section 3.1.1 and 3.1.2 except the reason for using React. (10/04/21)

	<ul style="list-style-type: none"> ● Added 5 UML Class Diagrams to the PPT. (10/04/21) ● Completed creating the slides for System Architecture Overview. (11/04/21) ● Updated the security requirements in Section 2.4 as there are two types of an account: UserAccount, AdminAccount. (11/04/21) ● Created 'TheALLDictionary-API Design.docx' and made the REST API design tables for the following (11/04/21): <ul style="list-style-type: none"> ○ Creating accounts ○ Deleting accounts ○ Getting personal data ○ Updating personal data ○ Changing source order settings ● Revised and edited the REST API design tables that Seoyoung has made. (12/04/21) ● Removed totalScore attribute for Quiz table and replaced it with totalNumberOfQuestions. Removed score attribute for HasQuizHistory and added numberOfCorrectAnswers and numberOfIncorrectAnswers. Hence updated the MySQL code in GitHub and updated Section 3.2. (12/04/21) ● Completed REST API tables for sharing card packages. (12/04/21) ● Added overall description of our schedule in Section 4. (12/04/21) ● Added non-functional requirement about the maximum limit for searching images and news from Google in Section 2.4.6. (13/04/21) ● Added more packages and API used in Section 3.1 (13/04/21) ● Splitted Section 3.6 into two sections. (14/04/21)
Yoo-ra Kim	<ul style="list-style-type: none"> ● Updated the UI mockups for all the pages. (07/04/21) ● Edited 2.3 together. (10/04/21) ● Completed the code convention for the front-end programming for Section 3.6. (11/04/21) ● Edited cardHome.html, showCard.html, and makeCard.html (13/04/21) ● Updated the Figure 5, Figure 5-1, Figure 5-2 screenshot in Chapter 2.3. (13/04/21) ● Revised 2.3 User Interface Card part. (13/04/21)
Seo-young Ko	<ul style="list-style-type: none"> ● Edited 2.3 User Interface Card part (11/04/21) <ul style="list-style-type: none"> ○ Figure 5: Description updated ○ Figure5-1: Description updated ○ Figure5-2: Description updated ● Added REST API design tables for following (12/04/21): <ul style="list-style-type: none"> ○ Creating and deleting challenges ○ Creating and deleting challenge problem

	<ul style="list-style-type: none"> ○ Updating a challenge record ○ Creating and deleting Notecards ○ Updating quiz and quiz history ○ Creating quiz problem ○ Updating a quiz record ● Added 3.1.5 Design Patterns: MVC diagram (12/04/21) ● Added 4 Schedule (13/04/21) ● Software Design presentation (15/04/21)
Seung-jae Kang	<ul style="list-style-type: none"> ● Revised contents in 2.3 User Interface. (07/04/21) ● Made UML sequence diagram for Section 3.3. (12/04/21)

Note: The presenter for our SW Design Document has been changed from Young-jae Moon to Young-jae Moon + Seo-young Ko to balance the work from each member.

Appendix A - Survey results from potential users

Question: “What do you think about developing a software that searches definitions and example sentences of a term from multiple dictionaries such as Merriam Webster and Oxford English Dictionary?”

Answer from Young-ho Kim: “Sounds like a good idea! As a person who has used dictionaries since kindergarten to study words, it is interesting to see how we can integrate multiple dictionaries to study words. I think Merriam Webster and Oxford English Dictionary are great dictionaries to look up an English word especially to study for the Scholastic Aptitude Test (SAT) exam. I don’t think using a Naver English dictionary is a good idea especially for studying SAT vocabularies because they only provide definitions in Korean which is not good. It is a good idea to only use our English brain to study English and only use our Korean brain to study Korean. From this perspective, I think looking up a word from multiple dictionaries is good, but we should exclude dictionaries which provide a definition in another language.”

Answer from Sudara Ranasinghe: “I think that would be a really useful software because when I usually look for a word I am looking for the meaning of the word in the specific context in which I found the word. Having multiple sources would help clarify the usage of the word.”

Answer from Hae-in Park: “It sounds like a good idea! I like the fact that the project managers are trying to use different sources together. It would enhance the quality of the project even further.”

Question: “What kind of methods or sources have you used to memorize English vocabulary? If you have used any other third-party application to effectively memorize vocabularies, please could you comment on how effective they were for memorizing vocabularies?”

Answer from Young-ho Kim: “I have used Quizlet to study SAT vocabularies. I think creating flip cards either via paper card or online is helpful to memorize SAT vocabulary. Nonetheless, I found it annoying to manually type every word and its definition to create flip cards. It would be a good idea if you develop software that includes a feature to add a definition for each vocabulary either by a single click or a double click. That will make my life a lot easier to study English vocabularies.”

Answer from Sudara Ranasinghe: “I haven't used any software to remember words in English but I have used Duolingo to learn Korean vocabulary. The app helps learn vocab by providing simple quizzes.”

Answer from Hae-in Park: “I tried to study English and memorize the vocabularies via two methods. I used to tried to write the words and read out loud repeatedly, and I bought a software called 뇌새김 which helps the learners to memorize the vocabularies with related images”