Flash Carma

By Adrian West, Brandon Sitz,

Joseph Ramos-Garcia, and Josh Moore

# Requirements

1. Navigation bar

1.1. Search bar

1.1.1. Allow the user to search for a study deck in the existing decks.

1.2. The following page links will be displayed:

1.2.1. Homepage

1.2.2. View Study Decks page

1.2.3. Study Session page

1.2.4. Settings page

1.2.5. Profile

2. Login

2.1. The user will enter their email address and password in a form.

2.1.1. If the information is incorrect, the page will be reloaded, and a message indicating an invalid username or password will be displayed.

2.1.2. If the login information is correct, the user will be redirected to the Homepage.

2.2. The user can also click a link labeled “Sign Up” that directs them to the Sign-Up page.

3. Sign Up

3.1. Multiple prompts will be presented- each with its respective text box- for the user to enter a username, an email address, and a password. There will also be another text box presented with a prompt to re-enter the password. Once the user is finished, they will click “Sign Up”. The two text boxes will be compared to make sure they match.

3.1.1. If the passwords in the text boxes do not match, the user will be prompted to try entering the password again.

3.2. After clicking “Sign Up” and successfully creating a profile, the user is directed to the Login page where they can sign into their profile.

4. Homepage

4.1. Displays user’s current study score.

4.2. Displays four most recently used study decks.

5. Settings page

5.1. Profile settings

5.1.1. Edit username

5.1.1.1. When clicked, the user’s current username will be presented in a text box for the user to modify.

5.1.1.2. When the user is finished, they will click “Done”.

5.1.1.3. If the username is currently taken, a dialog will appear with a message stating that the name is taken. Otherwise, the user’s new username will appear in-place of the old one.

5.1.2. Edit password

5.1.2.1. When clicked, a dialog appears requesting the user’s current password before allowing any changes.

5.1.2.2. If the password provided is correct, a text box will be presented with a prompt for the user to enter their new password. There will also be another text box presented with a prompt to re-enter the password. Once the user is finished, they will click “Done”. The two text boxes will be compared to make sure they match.

5.1.2.2.1. If the contents of the text boxes do not match, the user will be prompted to try entering the new password again.

5.1.2.3. If the password provided is incorrect, a dialog will appear with a message stating the password provided is incorrect. Until a correct password is provided, the user cannot proceed with any changes.

5.2. Other settings

5.2.1. Switch between light and dark mode.

5.2.2. Switch between grid and list layouts.

6. View Study Decks page

6.1. Each study deck will be displayed in individual containers.

6.1.1 If the user clicks on one of these decks, they will be shown a drop-down menu with the options to edit, delete, or study a deck.

6.1.1.1. If the user clicks “Edit” they will be sent to a page where they can edit the selected study deck.

6.1.1.2. If the user clicks “Study”, they will be sent to the Study Session page.

6.1.1.3. If the user clicks “Delete”, they will be given a prompt asking if they want to delete the deck.

6.2. Button to create a new study deck.

6.2.1. The user will be prompted with a dialogue box asking what to name the deck.

6.3. Edit existing study deck

6.3.1. Add flashcard

6.3.1.1. The user will be prompted to edit the flashcard or cancel. If they fill out the required information, a new card will be added with that information.

6.3.2. Delete flashcard

6.3.2.1. The user will be prompted with a dialog box that will ask them if they want to delete a flash card to the deck or to cancel.

6.3.3. Edit flashcard

6.3.3.1. The user will be prompted with a dialogue box asking them for the information on the front of the card and information on the back.

6.4. Study session page

6.4.1 Flashcard display

6.4.1.1 The first flashcard from the selected study deck will be displayed to the user.

6.4.1.2. The user will have the option to click the flashcard to display the opposite side.

6.4.1.3. Once the flashcard is clicked, an animation will be triggered to display the opposite side.

6.4.2. Correct choice

6.4.2.1. A button will be displayed to indicate a correct guess by the user.

6.4.2.2. If the user guesses correctly, they will click the button and the next flashcard in the study deck will appear.

6.4.3. Incorrect choice

6.4.3.1. A button will be displayed to indicate an incorrect guess by the user.

6.4.3.2. If the user guesses incorrectly, they will click the button and the next flashcard in the study deck will appear.

6.4.4. Study session score

6.4.4.1. The study session score will be displayed to the user.

6.4.4.2. If the user selects the correct button, the score will be increased.

6.4.4.3. If the user selects the incorrect button, the score will not change.

6.4.5. Correct answer streak

6.4.5.1. If the user answers correctly twice or more consecutively, a correct answer streak will be displayed to the user indicating the number of consecutive correct answers in the current study session.

6.4.6. End study session

6.4.6.1. A button will be displayed to end the current study session.

6.4.6.2. If clicked the current study session will end and the results of the study session will be displayed.

6.5. Results of study session

6.5.1. A results screen will display the following:

6.5.2. Study session score

6.5.3. Correct answer streak

6.5.4. Name of the study deck

7. Firebase

7.1. Database

7.1.1. User

7.1.1.1. The following user information will be stored:

7.1.1.1.1. User ID

7.1.1.1.2. Username

7.1.1.1.3. User password

7.1.1.1.4. User score

7.1.2. Study deck

7.1.2.1 The following study deck information will be stored:

7.1.2.1.1. Study deck name

7.1.2.1.2. Number of flash cards in study deck

7.1.2.1.3. Associated flash cards

7.1.3. Flashcards

7.1.3.1 The following flashcard information will be stored:

7.1.3.1.1. Flashcard question

7.1.3.1.2. Flashcard answer

7.3. User authentication

7.3.1. Provided via Firebase username and password authentication function call.

7.4. Website hosting

7.4.1. Set up project using the Firebase CLI (Command Line Interface).

7.4.2. Connect directory to Firebase project with “Firebase init” in the Firebase CLI.

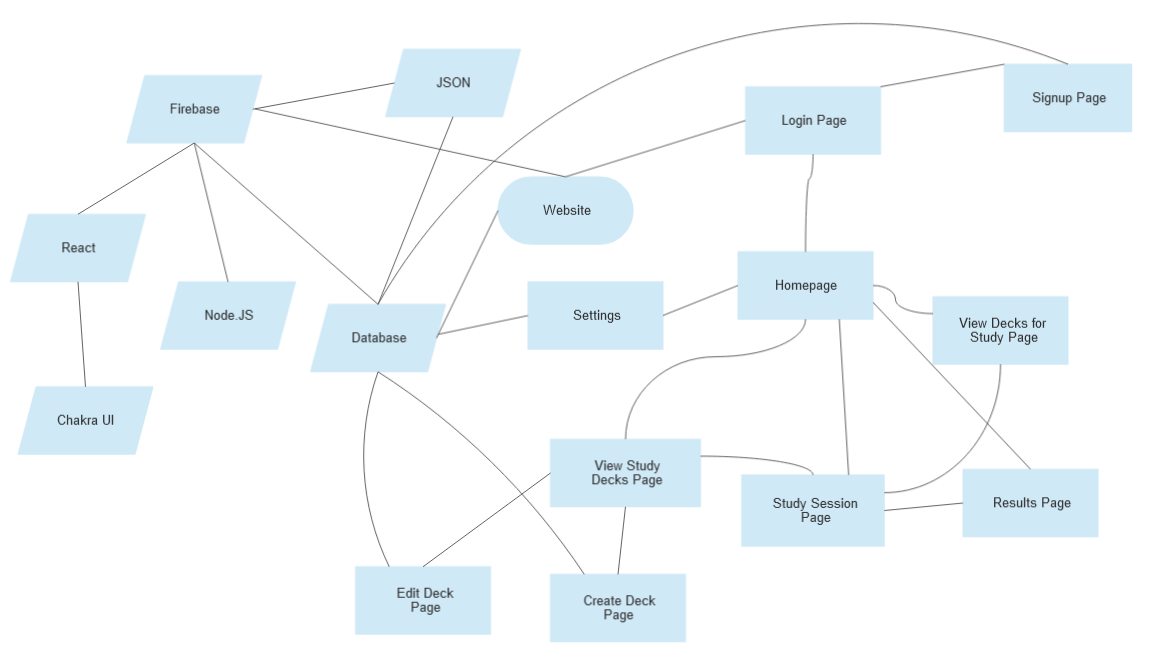
7.4.3. Deploy the web application onto the server with “Firebase deploy” in the Firebase CLI.

# Design Description

Flash Carma is a web-based application that utilizes digital flash cards and rewards the user for answering questions correctly. Upon entering the URL for the application, the user is met with the Login screen. The Login screen prompts the user for an email address and password and provides respective text boxes to enter the information. Once the user’s information has been entered, they can click the “Log In” button and an authentication request will be sent to firebase to verify they are a registered user. The user will be directed to the Homepage -provided that the information entered corresponds with an already existing profile. If the information entered does not correspond to an existing profile, the user will be notified that either their email or password is incorrect. If the user is new and does not yet have a profile, they can click the “Sign Up” link and will be taken to the Sign-Up page. The Sign-Up page prompts the user to enter a username, email address, and password, which will all be used to create the user’s new profile. Each of these prompts is accompanied by its own textbox for the user to enter the information. The user is also prompted to enter their password a second time in an additional textbox, which will be compared to the first entry of their password to ensure the characters making up the password are what the user intended. Once the information has been entered, the user can click the “Sign Up” button to create their profile. The user’s information will be saved to the database according to *Figure 11* in the message documentation. After the user’s profile has been created, they will be directed to the Login page, where they can now log in to their existing profile. Once they reach the Homepage, which is depicted in *Figure 5*, the user gains access to the navigation bar. The navigation bar provides links to various pages within the application. The first link is represented by the Flash Carma logo and will direct the user to the Homepage when clicked. The second link, labeled “View Study Decks”, will direct the user to the View Study Decks page, where they can view all of their existing study decks. The third link is labeled “Study Session” and will direct the user to a screen that presents the user- in a grid layout that is similar to the View Study Decks page- with all of their existing study decks, allowing them to choose which one they wish to study. The next thing within the navigation bar is less of a link to a specific page and more of a feature for the application. To the right of the “Study Session” link is a text box followed by a button labeled “Search”. With this feature, the user can enter a word(s) and click “Search”. Once “Search” is clicked, a request will be made to the database to search for the requested name in the user’s current study decks and the user will be taken to a screen where all study decks that include the entered word(s) in their title will be displayed in a grid layout. The next link in the navigation bar is labeled “Settings” and will direct the user to the Settings page when clicked. From the settings page, the user can personalize certain things within the app, such as selecting whether their study decks are presented in a grid layout- which is the default setting- or a list. The user can also change their username, email address, and password from this page. The final link within the navigation bar, labeled “Profile”, directs the user to the Profile page, which displays their current username and the email address associated with their profile. Should the user wish to change their username, email address, or password, the Profile page provides buttons that allow the user to make those changes. The navigation bar is available to the user at any time and is present on all pages with the exception of the Login and Sign-Up pages. As the user completes study sessions in Flash Carma, they are awarded points depending on the number of questions they answer correctly. On the Homepage, the user can see their cumulative score from all their past study sessions. Additionally, the Homepage displays the four most recently studied study decks. The View Study Decks page presents all the user’s existing study decks in a grid layout. The user’s study decks- organized in a way where the most recently edited deck is first and the least recently edited deck is last- will begin on the second element in the grid and continue to the final element in the grid. The first element in the grid is a square button- shaped like the study decks in the grid- with a “+” symbol on it. When that button is clicked, the user will be taken to the Create Study Deck page, where they can create a new study deck. If the user clicks on one of the study decks, a drop-down menu will display. The drop-down menu will give the user the option to “Edit”, “Delete”, or “Study” the selected deck. If the user selects the option to delete the study deck, they will be given a prompt asking if they want to delete the chosen study deck. If the user selects the option to edit the study deck, they will be redirected to the create study decks page. The create study decks page will allow the user to edit, add, and delete flashcards from the selected study deck. If the user clicks on the option to delete a flashcard, a prompt will be displayed to verify they want to delete the flashcard. If the user selects the option to add a flashcard, two text boxes will be displayed, one for the front and the back of the flashcard. Once a flashcard has been created, it will be added to the current study deck. If the user would like to edit a flash card, they can change the text on the front or the back of the flashcard. Once the user navigates to the Study Session page, the selected study deck will be requested from the database according to *Figure 12* in the message documentation.The study session page, which is depicted in *Figures 7-8*, will display the first flashcard in the study deck to the user in the center of the page. If the user clicks on the flashcard, an animation will be triggered to flip the flashcard to either the front or the back. The user’s current score will be displayed directly above the flashcard at any given time during the study session. A correct choice button and an incorrect choice button will be displayed below the flashcard. If the user guesses correctly, they will select the correct choice button. If the user selects the correct choice button, their score will be incremented, and the next flashcard in the study deck will be displayed. If the user selects the incorrect choice button, the score will not change, and the next flashcard in the study deck will be displayed. Additionally, the current flashcard out of the total number of flashcards will be displayed above the current flashcard and the user’s score. Once the user moves to the next flashcard in the study deck, the number indicating the current flashcard will be incremented by one. The user will also have the option to end the current study session at any time via a circular button containing an “x” symbol at the top right of the page. If the user selects the button to end the study session, the study session will end, and the user will be redirected to the results of the study session page. The results of the study session page, which is depicted in *Figure 9*, will display the results associated with the completed study session. A title with the text “Results” will be displayed at the top of the page. Below the “Results” text, the number of flashcards completed out of the total number of flashcards in the study deck will be displayed to the user. Additionally, the number of points accumulated in the study session will be displayed underneath the number of flashcards completed. The user will have the option to restart the previous study session via a “Replay” button at the bottom of the screen. The user will also have the option to return to the home page via a “Home” button located to the right of the “Replay” button. If the user elects to restart the study session, they will be redirected to the study session page. In addition, the study session score will be added to the user’s overall score. The study session score will be reset to zero, and the new study session will begin with the first flashcard in the current study deck.

# Appendix

## Block Diagram

***Figure 1****: Block Diagram for the Flash Carma web application.*

The website will use Firebase to host the website. It will use Node JS as a framework, and React as a means to make a well-done and good-looking UI. The Chakra UI library will be used to assist with that. Firebase will also provide us with a firestore database, which will have data stored in JSON. This data will include storing all the user authentication data and user information, which would include data such as their decks and cards stored within.

The Website itself will ask the database at various pages for different types of information. The figure above also conveys the general layout of the pages as well as points where the database will be updated:

Login Page: requests user information from the database. If it is provided, proceed to the homepage. Otherwise, the user can proceed to the signup page to register.

Sign-Up Page: New user is added to the database; links to the login page.

Homepage: Names of recently studied decks requested. Links to view study deck, view decks for study, study session (through recent decks), and settings pages.

Settings Page: Option to modify the user information (username/password). Links to homepage.

View Study Decks Page: requests the names of all the user’s decks. Links to the create study deck, edit deck, and study session pages.

Create Study Deck Page: Adds a new deck to the list of decks that a user has created. Links back to view study decks page.

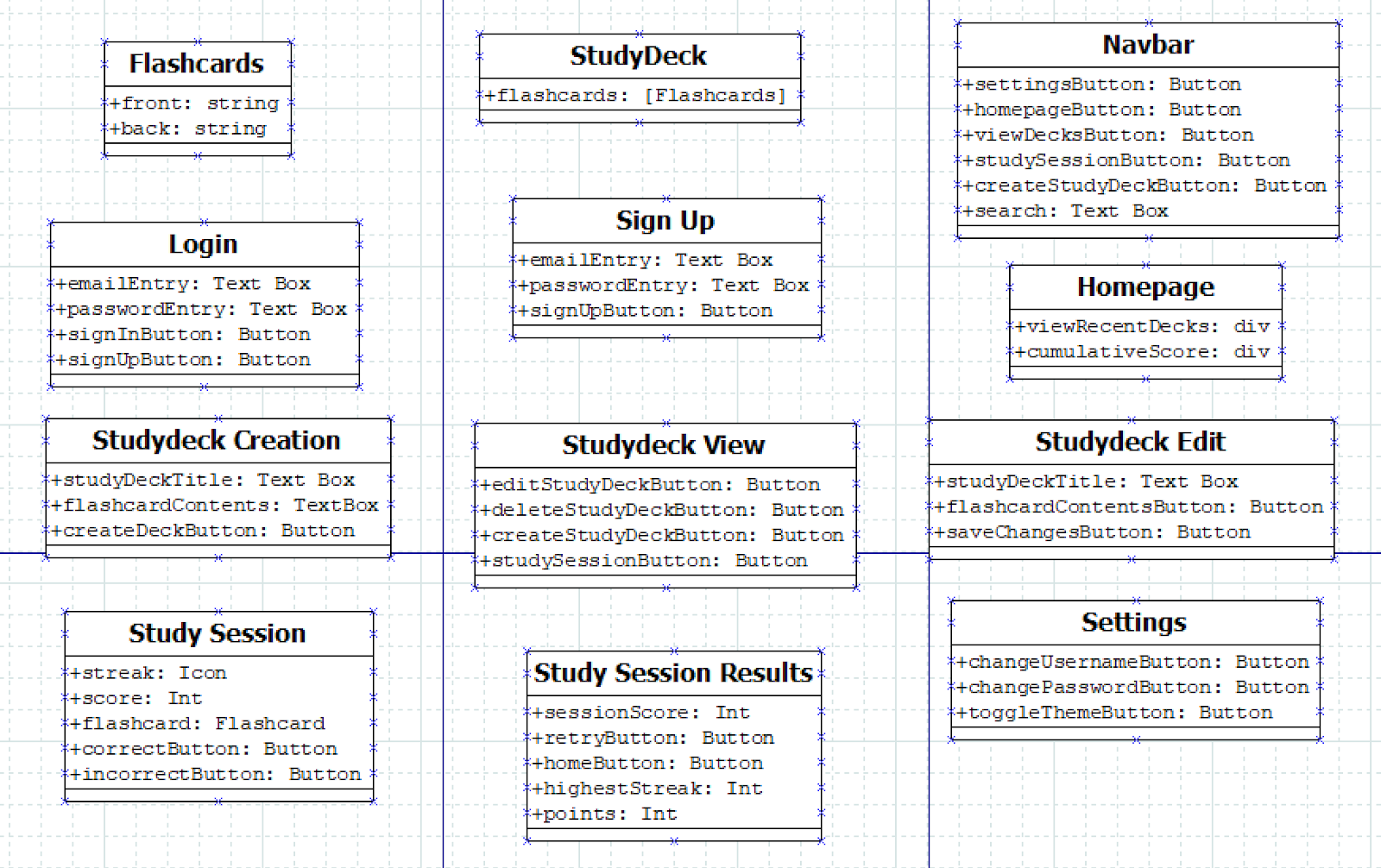
Edit Deck Page: Adds new cards under a specific deck and edit the name of the selected deck. Links back to view study decks page.

View Decks for Study Page: Requests the names of the study decks. Links to study session page and homepages.

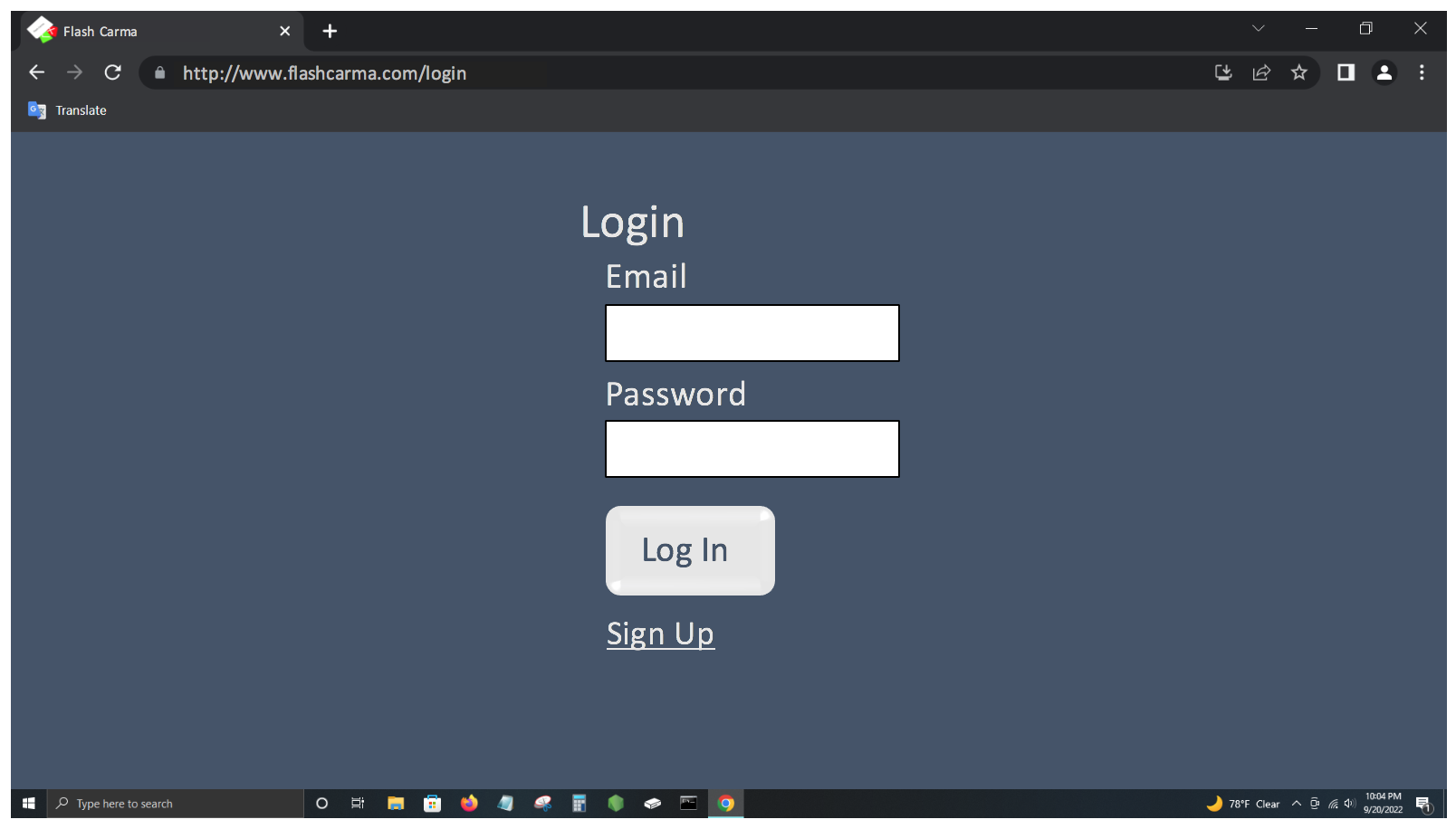
Study Session Page: Requests the title of the deck and the information on each card in a specified deck. Links to results page.

Results Page: links to the homepage or the study session page for the current deck.

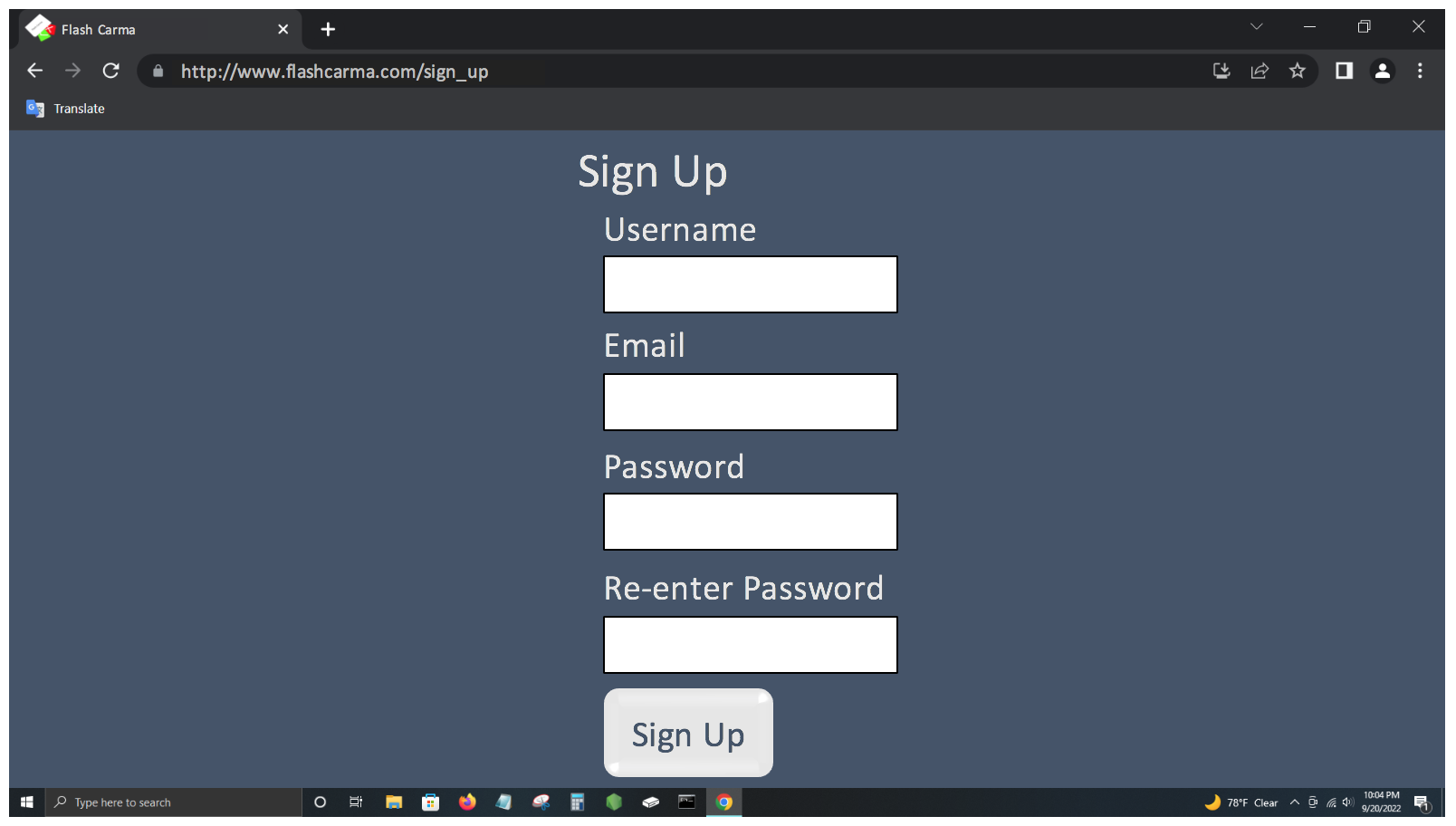
## Component Diagram

***Figure 2****: Component Diagram for the Flash Carma web application.*

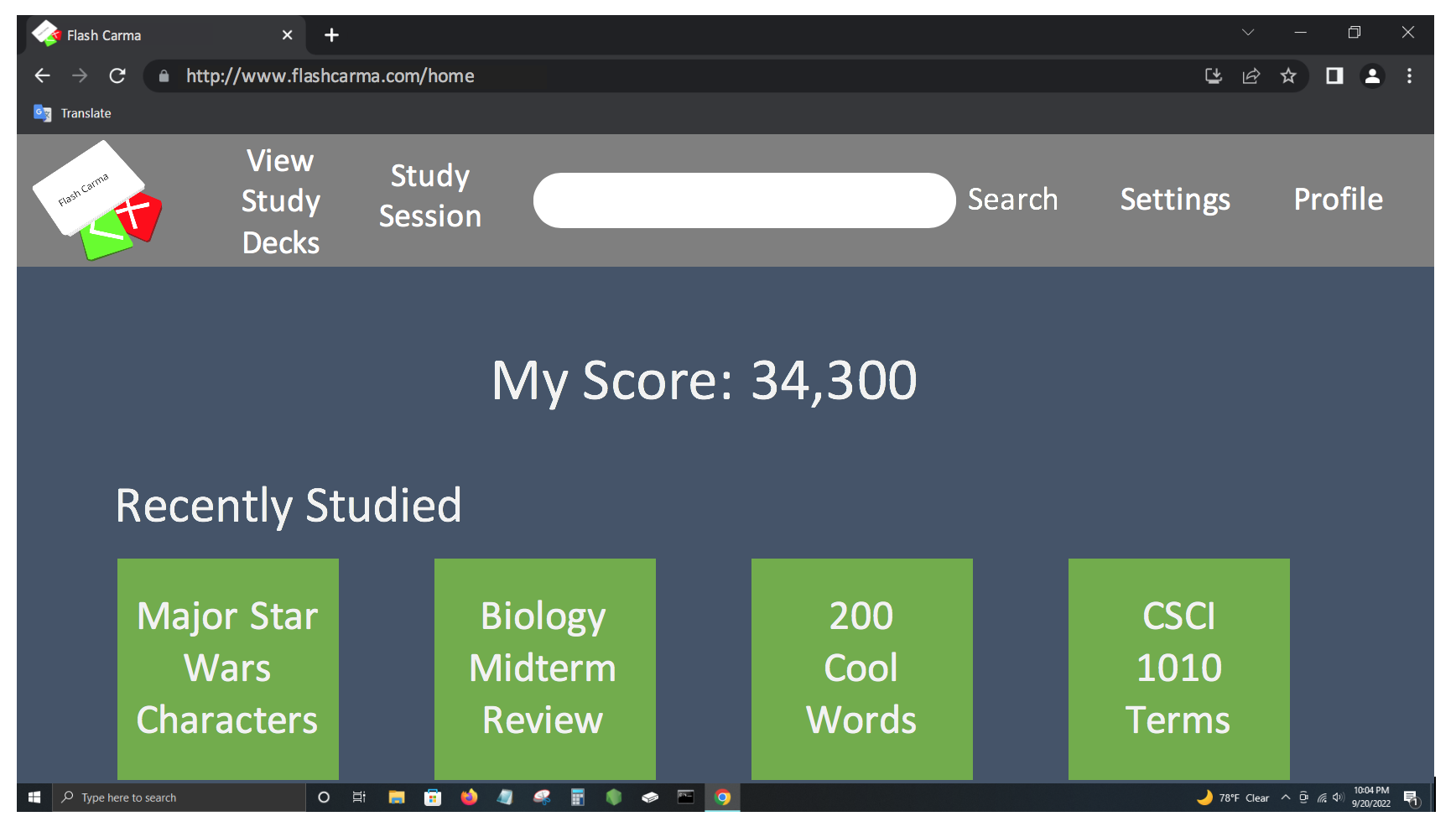
## User Interface Storyboard



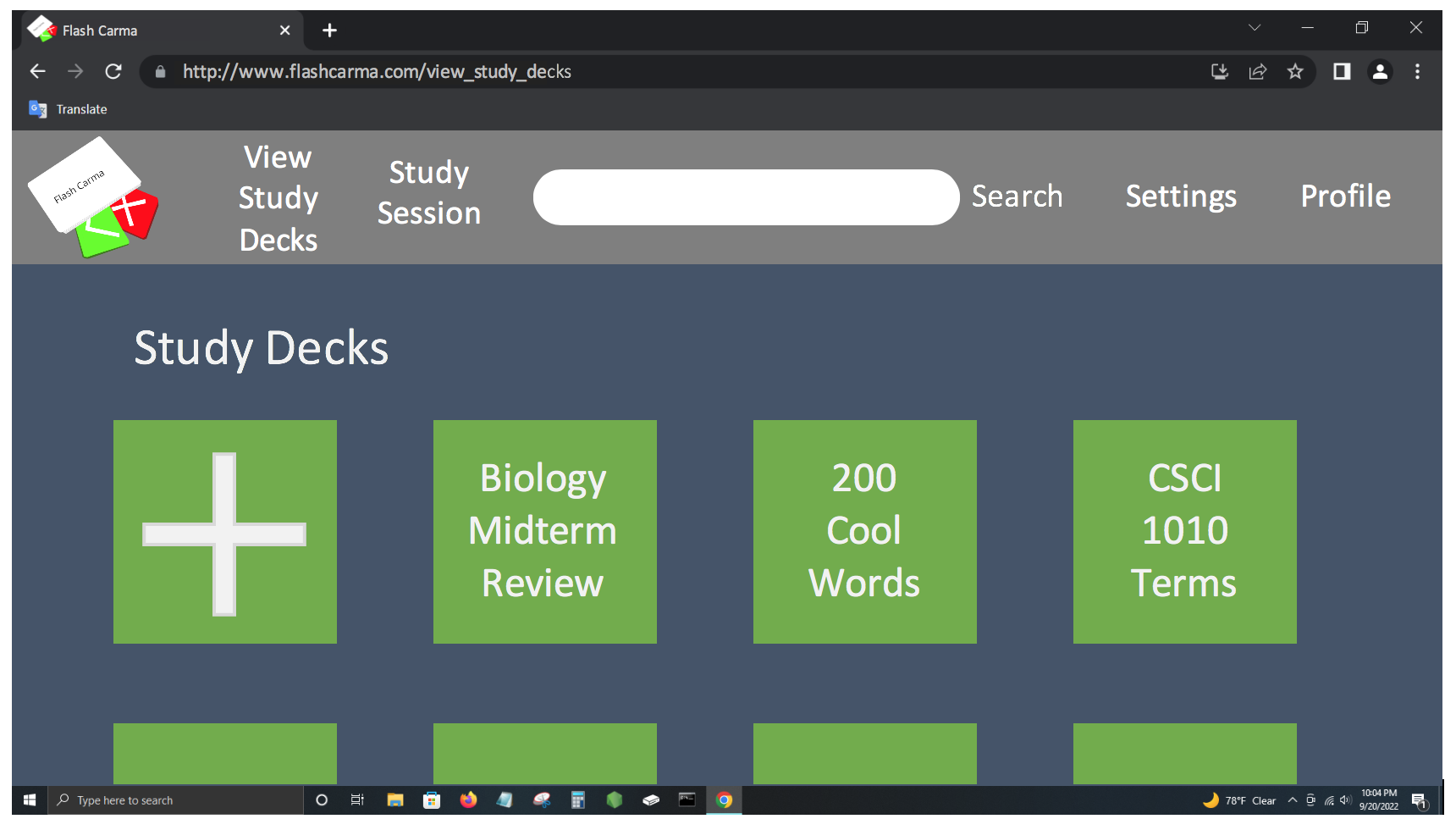
***Figure 3****: Sample of the Login page.*



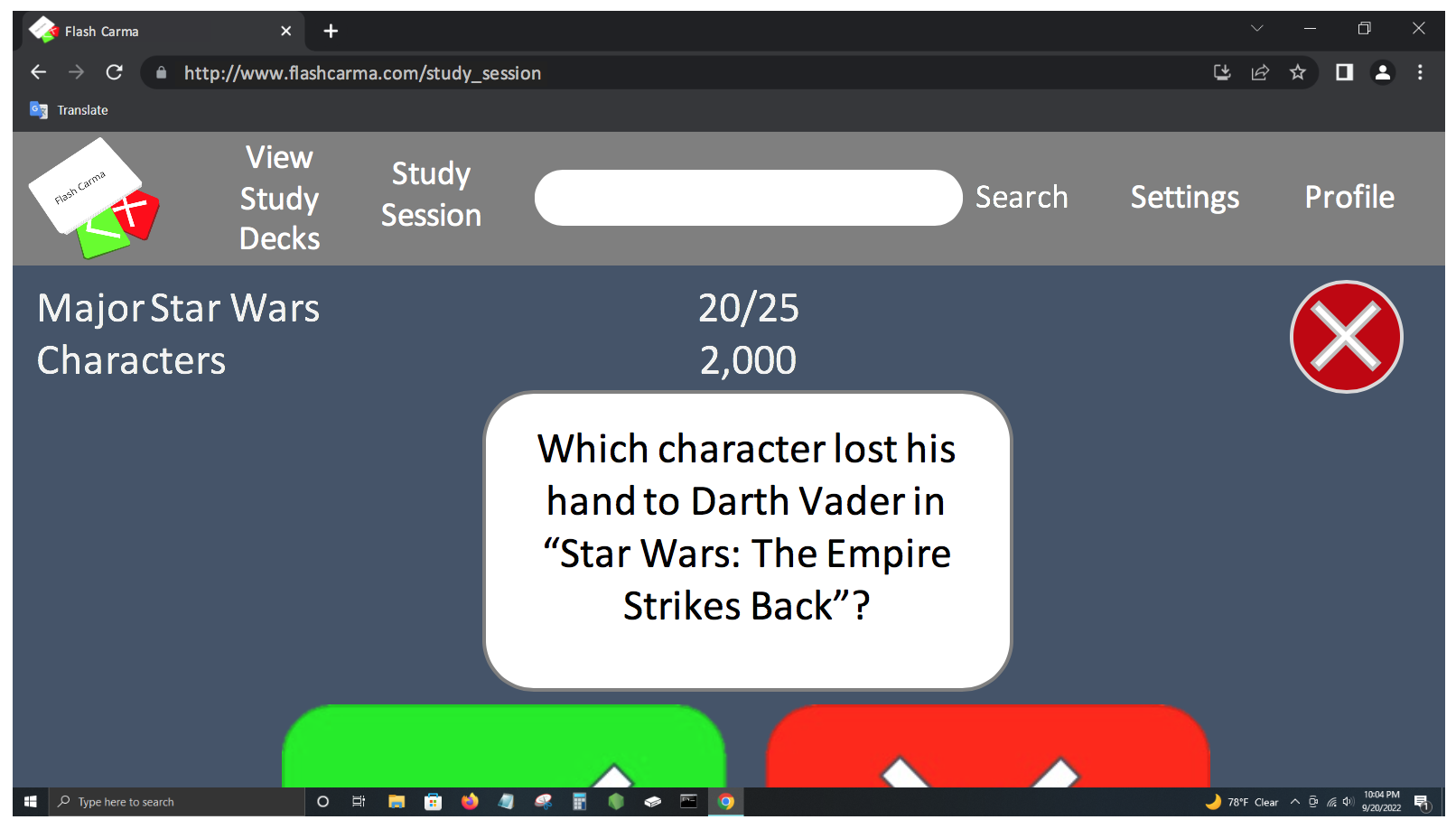
***Figure 4****: Sample of the Sign-Up page.*



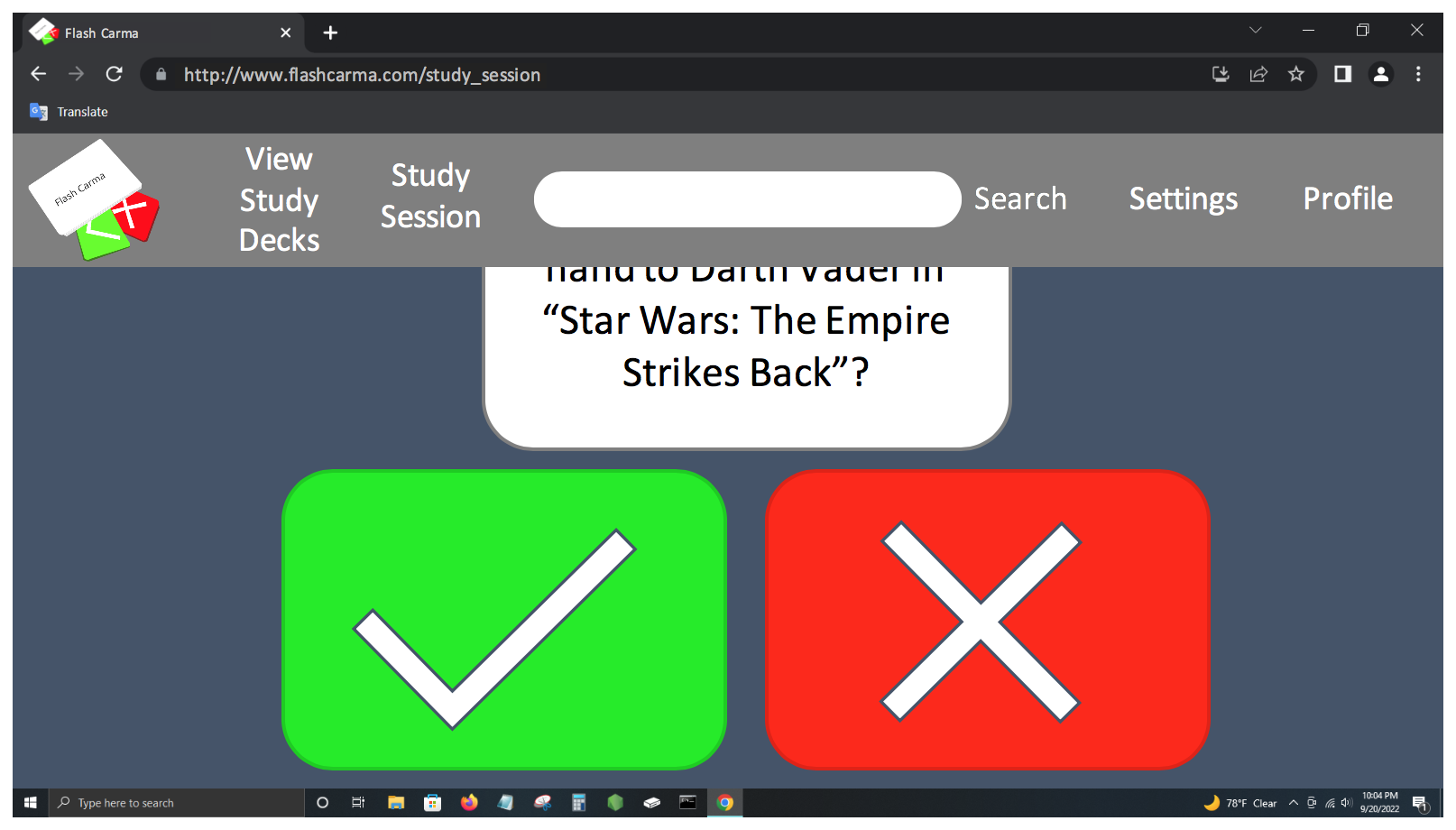
***Figure 5****: Sample of the Homepage.*



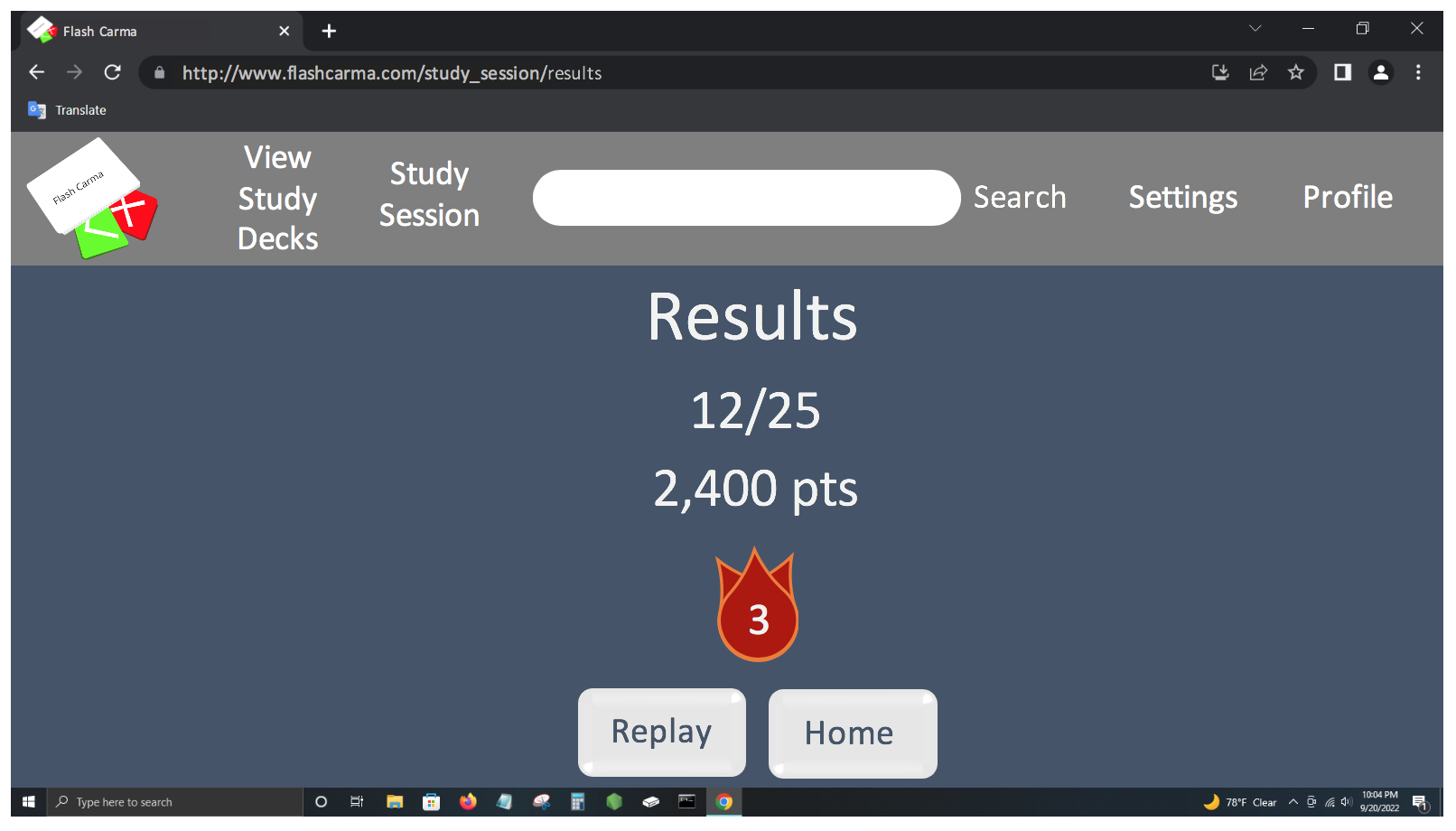
***Figure 6****: Sample of the View Study Decks page.*



***Figure 7****: Sample of the user’s screen mid-study session.*



***Figure 8****: Sample of the user’s screen mid-study session (continued).*



***Figure 9****: Sample of the user’s screen after completing a study session.*

## Message Documentation

* + ***Figure 10***: User authentication
    - The email and password entered by the user will be authenticated with the list of users in firebase.
  + ***Figure 11***: Request a user from the database
    - Users
      * List of current users
      * Type: List
    - Userid
      * User unique identifier
      * Type: Number
    - Email
      * The user’s email address
      * Type: String
    - Password
      * The user’s password
      * Type: String
    - Username
      * The user’s username
      * Type: String
    - Score
      * The user’s cumulative score
      * Type: Number
    - JSON message
      * {

"users": [

{

"userid": "123",

"email": "user@example.com",

"password": "password",

"username": "user",

"score": "100"

}

]

}

* + ***Figure 12***: Request study decks from the database
    - Study decks
      * List of study decks
      * Type: List
        + Study deck ID

Study deck unique identifier

Type: Number

Number of flashcards

Number of flashcards in the study deck

Type: Number

Timestamp

Date and time the study deck was created. Stored as seconds.

Type: Number

* + - Flashcards
      * A list of flashcards in the study deck
      * Type: List
        + Flashcard ID

Flashcard unique identifier

Type: Number

Flashcard question

The flashcard’s question

Type: String

Answer

The flashcard’s answer

Type: String

* + - JSON message
      * {

"studyDecks": [

{

"studyDeckid": "0123",

"numberFlaschcards": "2",

"timestamp": "0123456789",

"flashcards": [

{

"flashcardid": {

"question": "question",

"answer": "answer"

}

},

{

"flashcardid": {

"question": "question",

"answer": "answer"

}

}

]

}

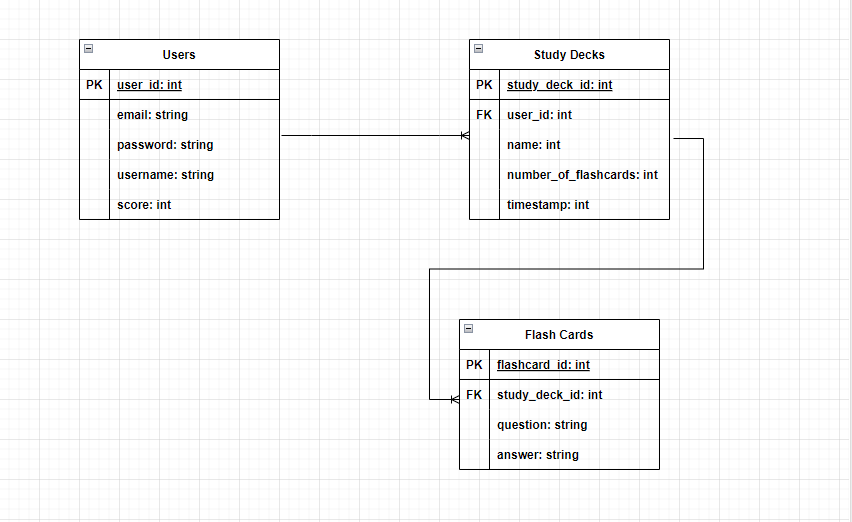
]

}

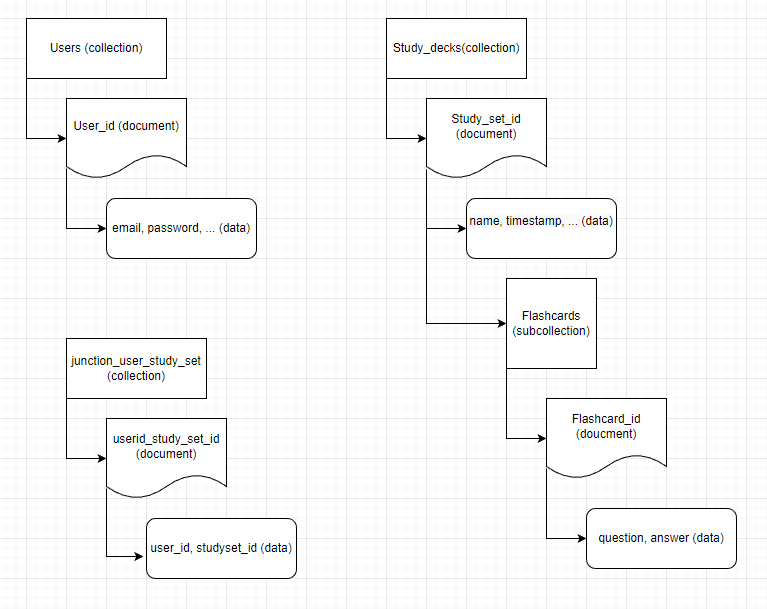
* + Each request to the database will be a subset of requesting a user and requesting study decks from the database.

## Storage Documentation

### Entity Relationship Diagram (ERD)

***Figure 13****: Entity Relationship Diagram.*

### Data model

***Figure 14****: Data Model Diagram.*