WordBattles

CSC309H1F
November 2016

Proposal By:
Alexander Chernous (1000494128)
Justin Chew (999618855)
Pak Ki Lam (1000084507)
Asier Ugarteche (999466028)

1- Overview

WordBattles is a web application that allows users to challenge each other on a classical game called hangman. The target audience is any web-users from all ages.

Users will be able to create and log into the application. The application is linked to a database that keeps track of user play records and win rates.
If users rank as one of the top players, they will be placed on a leaderboard.
Users will be able to challenge another online user into a word battle wherein the challenger will be able to submit a dictionary valid word for the challenged player to solve in a hangman-like game where the challenged player will be given limited chances to pick letters in order to guess the word.
A challenged player will be able to accept or decline any given incoming challenge.
For each game, there will be one winning and one losing player.
If the challenged player guesses the word, then that player is declared the winner and the other player will be declared the loser. Vice versa if the hidden word is not solved.
The results of the finished game will then be logged in the database.
Outside of the game, players will be able to access their profile and view/edit their name and bio and also see their gameplay statistics.

2-Structure

The basic structure of the web application is a front-end interface working with a backend-interface which stores and fetches data in a database. The web application will be hosted on a live hosting server like Heroku.

Front-end: We will be implementing the user interface by employing external libraries such as REACT for dynamic interactions and Bootstrap for a responsive design.

Back-end: The server side of the web application will be built with Node.js and using various node modules like express, express-validator, body-parser, cookie-parser, cookie-session, etc.

Database: The storage of data will be utilizing a NoSQL database, in particular, MongoDB.

The client side and server side will be communicating via HTTP requests and passing data in JSON format. Cookies and sessions will be used to identify the users and grant privileges after authentication has been done. The server interacts with the database by simply utilizing the mongodb module or the corresponding driver for node js.

3- Views & Specifications

mongoDB:

The database stores user ID's, passwords, number of wins, and bios.

Admin ID's and passwords are also stored in the database.

WELCOME PAGE (This is where admins login, and users login or signup):

-The only thing needed to sign up is a user ID that is not already in the mongoDB and a password longer than 5 characters.

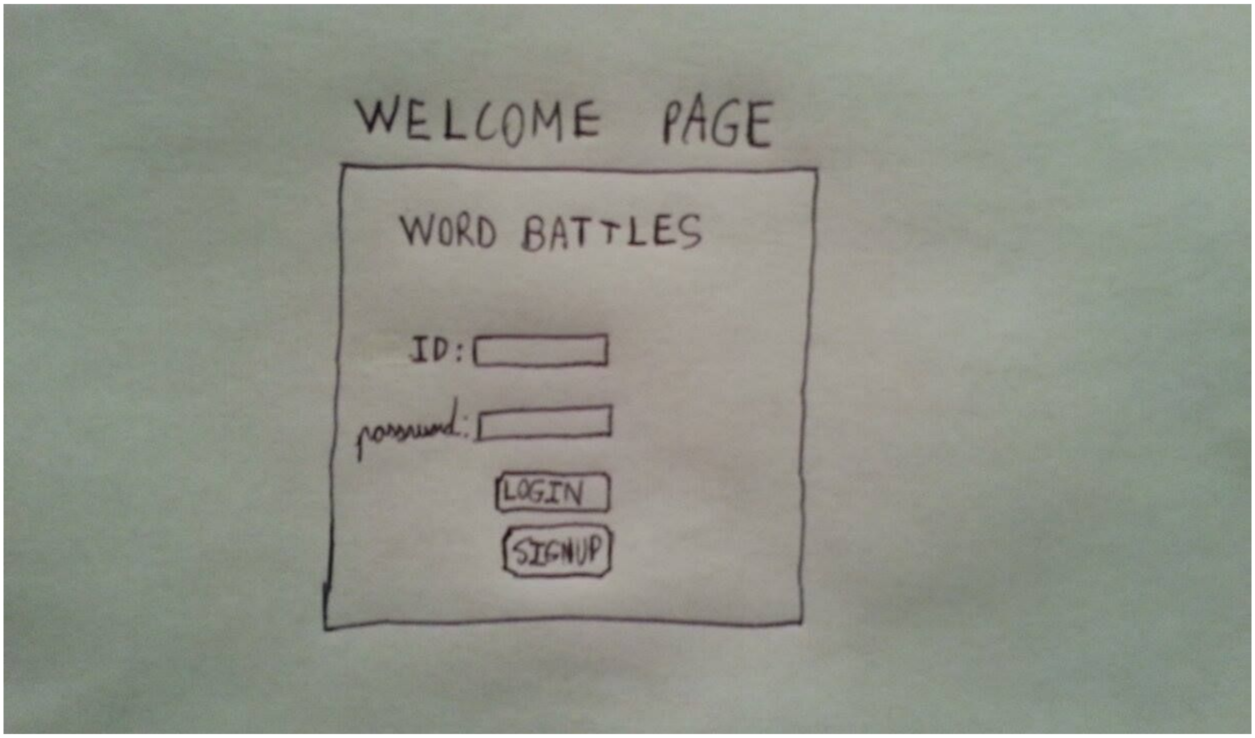
-Sign up data gets passed to server for validation and added to the database if valid.

-Unsuccessful sign ups result in an appropriate error message.

-A previously signed up user can log in with his/her user ID and password.

-When users log in, they are taken to the MAIN PAGE

-Admins also log in using this page. When admins log in, it takes them to the ADMIN PAGE



MAIN PAGE:

-The rules for the hangman game are explained in a text box.

-On the right side, there’s a small ONLINE USERS box that displays icons with all the users currently online (and whether they are just online or they are currently battling).

-Below that box, there’s a TOP20 USERS box that shows a ranking of the top 20 users with most wins. If users are tied with wins, order is alphabetical.

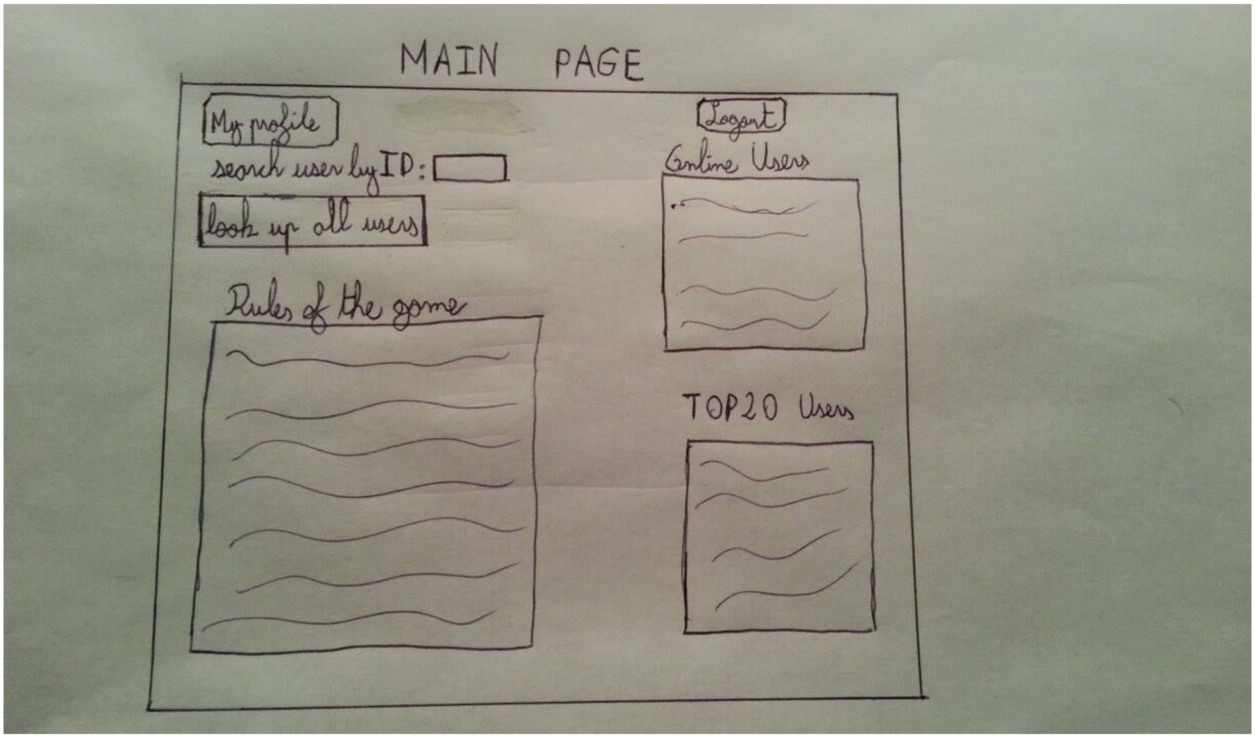
-”My Profile” button: Users can view their own profile. This takes users to the PROFILE PAGE

-Search bar for users: Users can search other users by ID. This takes users to the PROFILE PAGE.

-”Look-up all users” button: Users can view all the registered users. This also takes users to the PROFILE PAGE

-”Logout” button: Users can logout of the application. This takes users to the WELCOME PAGE

-In the ONLINE USERS box. A user X can click on an online user Y icon, this will alert user Y that user X wants to battle. If user Y accepts the challenge, the battle begins and both users are taken to the BATTLE PAGE.



BATTLE PAGE:

-For now, we are aiming to implement a generic hangman game (player who challenged is the one who selects the word for the other user to guess) with a generic hangman UI component. We might change the game rules and the component look during the implementation process.

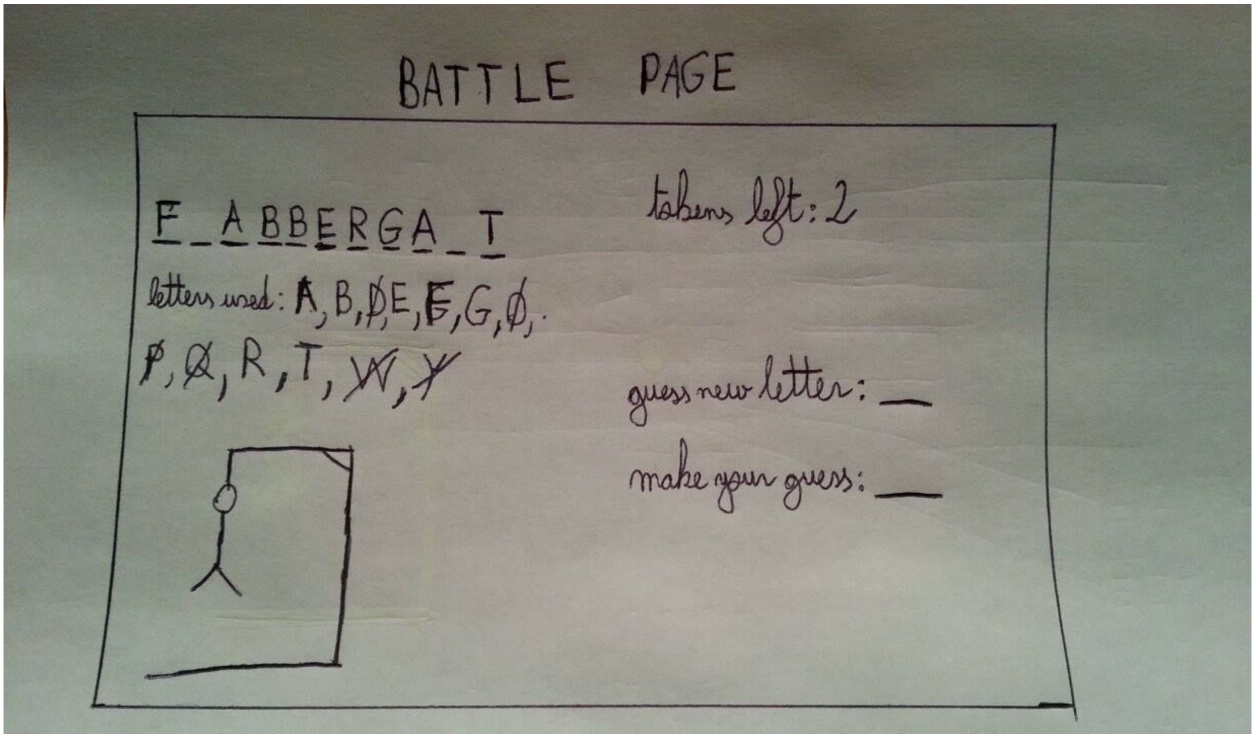
-Both users on the Battle Page see the same hangman component (which will be implemented using REACT).

-The user making the guesses has a limited amount of tokens for how many letters he can guess.

-The user making the guesses can guess the whole word at any time. If he succeeds, he wins. If he fails, he loses.

-There is a 2 minutes timeout. If nobody has won within 2 minutes, the user who chose the word automatically wins.

-Upon battle ending, there is a winner and a loser, the mongoDB gets updated accordingly. Users are redirected to the MAIN PAGE.



PROFILE PAGE:

-The view of this page is different depending on which action triggered it.

-If the “My Profile” button on the MAIN PAGE triggered it, then the PROFILE PAGE shows the personal user ID, the number of wins, and the personal BIO. User’s can change their user ID’s here (as long as it doesn’t conflict with the data inside the database). They can also change their personal bio.

-If the Search bar for users triggered it, then the PROFILE PAGE shows that user’s ID, its number of wins and its personal BIO.

-if the “Look up all users” button triggered it, then the PROFILE PAGE shows a table with all the user ID’s and their number of wins . The table entries are ordered alphabetically by user ID.

(Since the PROFILE PAGE changes its view based on the event triggering it, we are not providing a sketch for it)

ADMIN PAGE:

-Displays a table with all the User Id’s, their number of wins, and their passwords.

-Every entry in the table has an “EDIT USER” and “DELETE USER” button.

-At the top of the page, there’s an empty “Focus” div.

-When an admin clicks on an “EDIT USER” button, the “Focus” div is populated with the user ID, number of wins, password, and bio. Furthermore, a “SAVE USER” button populates the “Focus” div. Admins can change any of the parameters of the user and click on “SAVE USER” to push the changes into the mongoDB.

-When an admin clicks on a “DELETE USER” button, the user is deleted from the mongoDB.

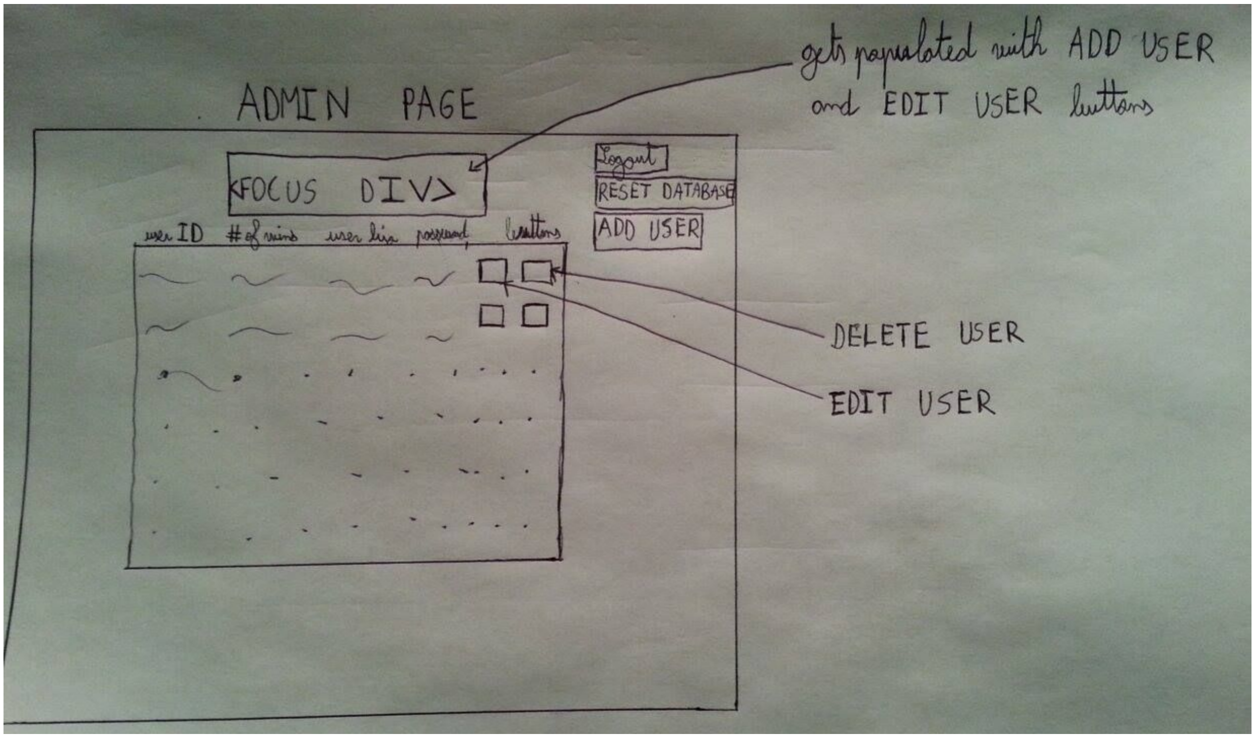
-There’s a “Logout” button at the top-right of the page. This takes admins to the WELCOME PAGE.

-There’s a “RESET DATABASE” button below the “Logout” button. This allows admin to delete all the users in the database.

-There’s a “ADD USER” button below the “RESET DATABASE” button.

-When the “ADD USER” button is clicked, the “Focus” div is populated with a user ID, number of wins, user bio, and user password input fields. Furthermore a “ADD” button populates the “Focus” div. Admins need to fill in all the fields and click on the “ADD” button to add the new user to the mongoDB.

-Everytime a “DELETE USER”, “SAVE USER” or “ADD” button is clicked, the page is reloaded so that the changes can be viewed in the page table.



4-ADDITIONAL FEATURES

These are features that we will implement if we have manage to implement all the features described previously and still have some extra time:

-In the MAIN PAGE, users can send friend requests to other users.
-In the MAIN PAGE, users can scroll through the TOP20 users and view the TOP20-TOP40 users, TOP40-TOP60 users, and so on…
-In the ADMIN PAGE, admins are allowed to create new admins (we will very much likely implement this functionality).