**Overview:**

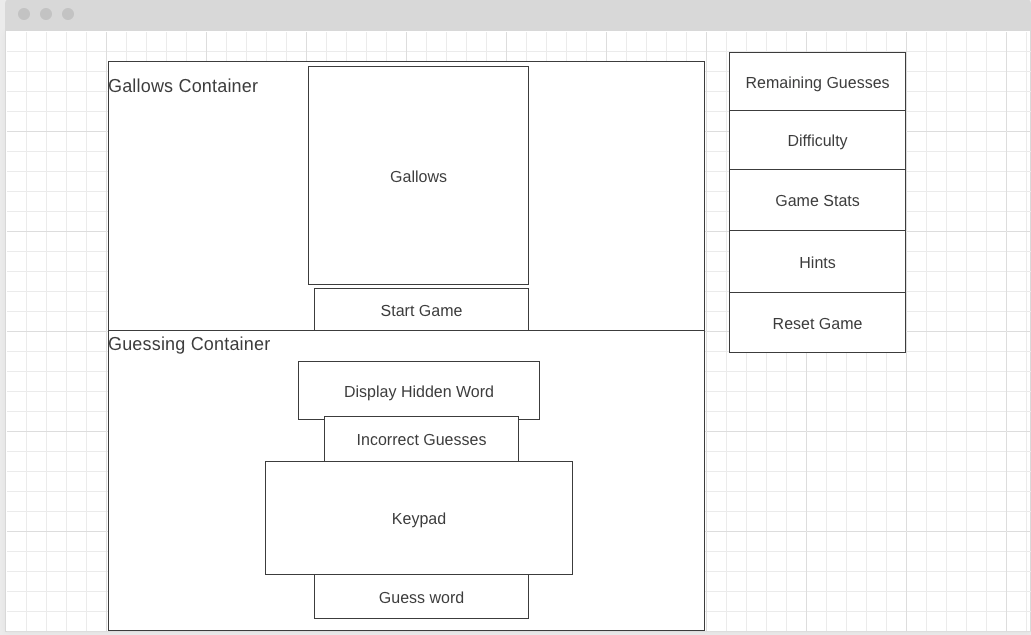
The purpose of application is a game where the computer, **secret-keeper**, randomly selects a word from a word bank. The user, **guesser**, then has to guess the word. The guesser can choose to guess the word one letter at a time or the whole word. The guesser will be allowed six incorrect guesses before the game is over.

The **secret-keeper** will reveal one letter at a time as the guesser correctly guesses. At the end of the game, if the **guesser** has not guessed all the correct words then the **secret-keeper** will reveal the remaining letters.

In short, the game needs the following:

* Get a random word.
* User guesses one letter at a time or the entire word.
* The user will have six wrong guesses. Each wrong guess results in one less guess.
* If the user runs out of guesses, then game will reveal the entire word.

**User Interface:**



* Gallows Container
  + Gallows – Shows the hangman images as the user gets wrong guesses.
  + Start Game
    - Allow user to select difficulty then start game. Difficulty scale from 1-10. Will affect request to word bank.
    - Transitions to a spinner while the app makes an API request to the word bank. Then disappears from UI. Word list retrieved will be saved to state as to not trigger additional unnecessary API calls.
    - At end of game, reappears allowing player to start another game.
* Guessing Container – Appears after user clicks start game.
  + Display Hidden Word – Represent hidden word as underscores. Each correct guess reveals one character. If user correctly guesses entire word then the entire word is shown. Entire word is revealed at the end.
  + Incorrect Guesses – Display incorrect letters and word guesses.
  + Keypad
    - Lists all 26 alphabetical letters.
    - Allow user to click on letters to make their guess.
    - Each clicked letter is disabled to avoid repetitive clicking.
    - Letter is also disabled if the user guesses just one letter in the Guess Word section. Entire keypad is disabled at the end of game.
  + Guess Word
    - Form input field where user can guess the entire word.
    - Single letters will also work and is respectively disabled in the keypad section.
    - Automatically convert input to uppercase.
    - Input validation – Cannot submit duplicate guesses, numbers, special characters, etc. Alphabetical characters only.
    - Disable at end of game.
* Sidebar – Contains game information
  + Remaining Guesses – Begins at 6 and decrements by 1 for each wrong guess. Resets to 6 for each new game.
  + Difficulty – Shows difficulty level selected at beginning of game.
  + Game Stats – Shows # of wins, # of losses, and current winning streak during the game session.
  + Hints
    - Only appears after the user selects a difficulty level and select game.
    - Has button inside to toggle whether to show hints or not. Has to toggle “show hints” for each new game.
    - After toggling “show hints”, display number of unique vowels and unique letters. This is because some vowels and letters are repeated.
  + Reset Game
    - Only appears after the user selects a difficulty level and select game.
    - This resets the game to the very beginning where the user has to select a difficulty level and start game.
    - Erases everything including game stats and any data related to the current game session.

**Technologies:**

* React.js – To build out the user interface into a series of components.
* Redux.js – Global state management. Allows various React components to update and receive updates as necessary depending on the current state. Redux will be used to contain the current word list, word, guessed letters, wrong letters, game stats. Word list will be maintained in state as to not require additional API calls that could slow down user interaction.
* Bootstrap 4 – CSS Framework to help build out UI. Use customized CSS as necessary.