**Requirements Document**

**Project D.O.T**

**Name: Hieu Tran**

**Description**

Standalone application that functions as a typing environment featuring different typing exercises/simulations. Users type traditional tests which are short passages that are 20 to 100 words long. They may also try different tests focused on developing WPM. Accuracy is required; any typing errors in words have to be fixed before the test counts their correct keystrokes.

**Purpose**

The fact of the matter is that typing skill is an important thing to learn, at any age, to make you a more coveted employee. DOT’s purpose is to increase the user’s WPM and accuracy by building muscle memory.

**End Users**

* Students Typers/Keyboard Hobbyists
* Story Writers
* Programmers
* People that want to improve their typing speed and accuracy.

**Scope**

* Application that displays an assortment of words, at random or from a passage.
* Different test simulations featured: Traditional (Passages), Random, Conveyor belt, and Accelerated conveyor belt.
* User WPM and Accuracy stored with every test. Able to be viewed on profile.
* There will not be multilingual support.
* There will not be full dvorak ([keyboard layout](https://upload.wikimedia.org/wikipedia/commons/thumb/2/25/KB_United_States_Dvorak.svg/1200px-KB_United_States_Dvorak.svg.png)) support.

**Functionality**

Landing page (Hub) will have buttons to take the user to the Tests, their profile and close the program. There will be several tests to choose from the Tests menu and the user will be able to clear their stats in the profile menu.

Character with incorrect keystroke and all characters past that character (number depending on how many keystrokes the user made) are highlighted red.

Maps created from time spent to reach keystrokes will tell the user which keystrokes and words they have trouble reaching the most.

Typing tests will be recorded and information on WPM and accuracy will be saved.

User profile will be display WPM and accuracy for each typing test type or by average of all tests regardless of test type.

**Use Cases**

* Guest is not signed in and takes a traditional test. (Finishes it)
  + They will start from the hub page and select tests.
  + User is given a list of different tests they can take.
    - Different use cases for each test where the only difference for each test is the specific test itself and where the information about the test is saved.
  + The user types the entire passage correctly.
  + Nothing is saved because they are not logged in.
* Guest is not signed in and takes a traditional test. (Does not finish it)
  + They will start from the hub page and select tests.
  + User is given a list of different tests they can take.
  + The user does not finish the type test.
  + Nothing is saved because they are not logged in *as well* as never having finished the test..
* User wants to sign up for an account.
  + The user lands on the hub interface.
  + User clicks on the sign up button.
  + They enter all of their information and a new user profile is created
    - Else, the sign up page will tell the user that they are missing something.
* Becky logs in, takes a test but does not finish it.
  + She lands on the hub interface.
  + User clicks the sign in button
  + Dropdown block for username and password from the sign in button.
  + After successful login, she will be back on the hub but now signed in.
    - In the event of incorrect login, user will get a message about an incorrect login.
  + She takes a typing test but stops before the timer ends.
  + Test is terminated and data is not saved.
* Becky logs in, takes a test and finishes it
  + She lands on the hub interface.
  + User clicks the sign in button
  + Dropdown block for username and password from the sign in button.
  + After successful login, she will be back on the hub but now signed in.
    - In the event of incorrect login, user will get a message about an incorrect login.
  + She takes a typing test and inputs all correct keystrokes.
  + WPM and accuracy is calculated and saved into her user profile.
* Guest makes an incorrect keystroke and keeps typing.
  + User decides to take a traditional typing test
  + User makes incorrect keystroke on a character
  + System allows him to keep typing in the field but does not advance the string conveyor belt.
  + User types all available characters despite having wrong keystroke
  + Test is terminated because user is not logged in.
* Sam makes an incorrect keystroke and keeps typing.
  + She lands on the hub interface.
  + User clicks the sign in button
  + Dropdown block for username and password from the sign in button.
  + After successful login, she will be back on the hub but now signed in.
    - In the event of incorrect login, user will get a message about an incorrect login.
  + User decides to take a traditional typing test
  + User makes incorrect keystroke on a character
  + System allows him to keep typing in the field but does not advance the string conveyor belt.
  + User types all available characters despite having wrong keystroke
  + Test is saved. WPM and accuracy are calculated and saved into user profile.