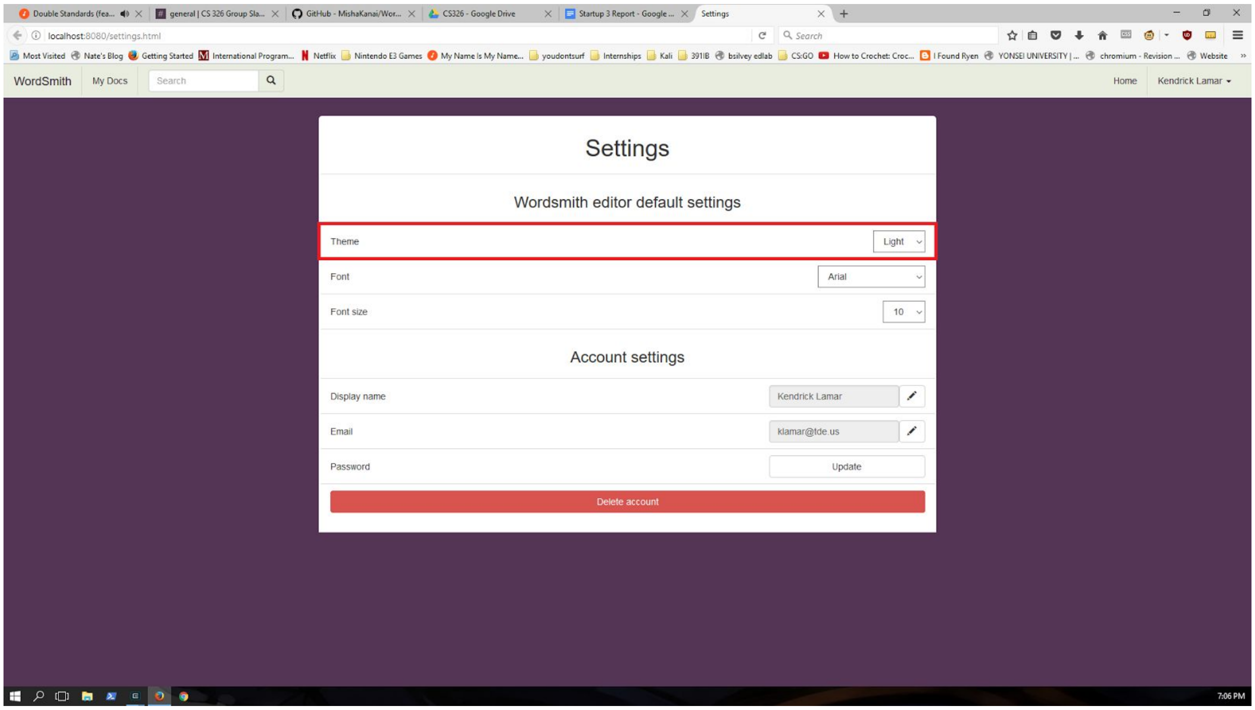
Startup Report #3
Team: Date
2017-03-26
CS326 Tim Richards

Hiram Silvey:
I was responsible for converting the static UI settings page to static React. I created a Settings.js file in app/components/.

Bugs:
- On the current settings page, the white box surrounding the actual settings is supposed to stretch to the bottom of the screen. However, after React conversion, it seems to stop right after the settings. I am currently working on a fix for this.

Components:
- Element: The basic list element on the settings page, as highlighted below. It repeats 6 times on the current page, with text on the left side and varying other components on the right.



Settings

Wordsmith editor default settings

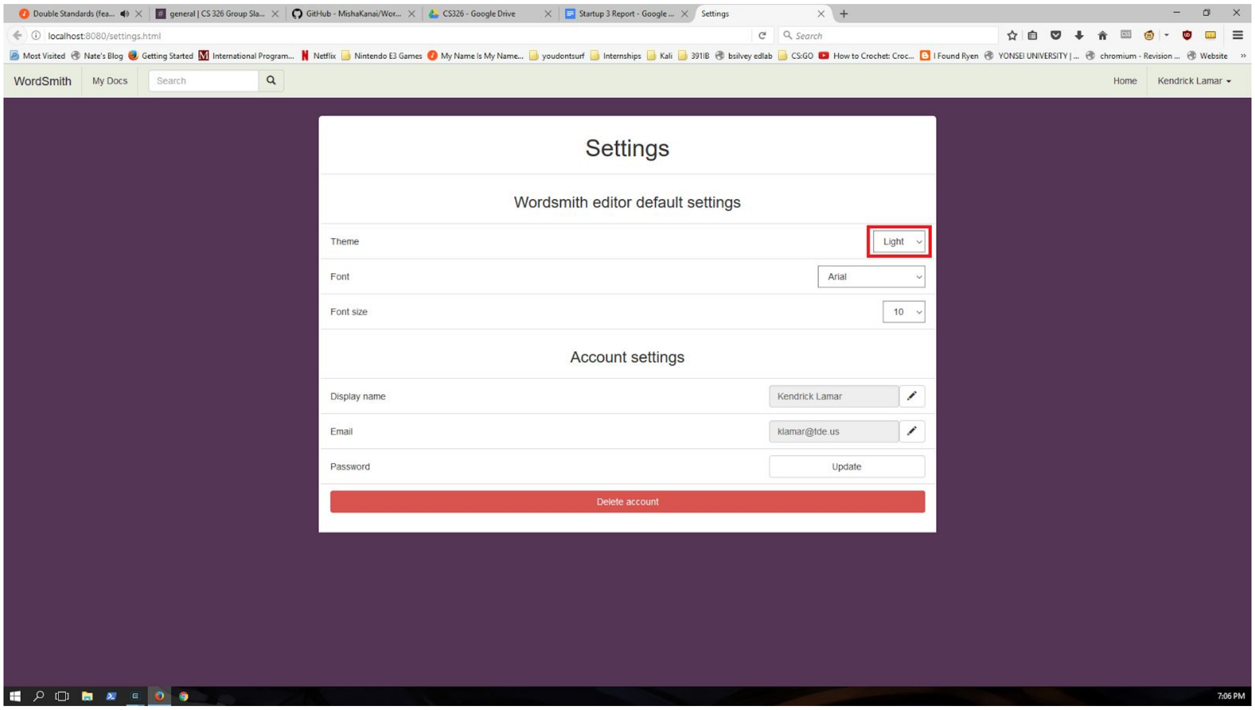
Theme Light
Font Arial
Font size 10

Account settings

Display name Kendrick Lamar
Email klamar@tde.us
Password Update

Delete account

- ListElements: The right part of Element that contains a list of values. The top 3 Elements contain a ListElements on the right.



Settings

Wordsmith editor default settings

Theme: Light

Font: Arial

Font size: 10

Account settings

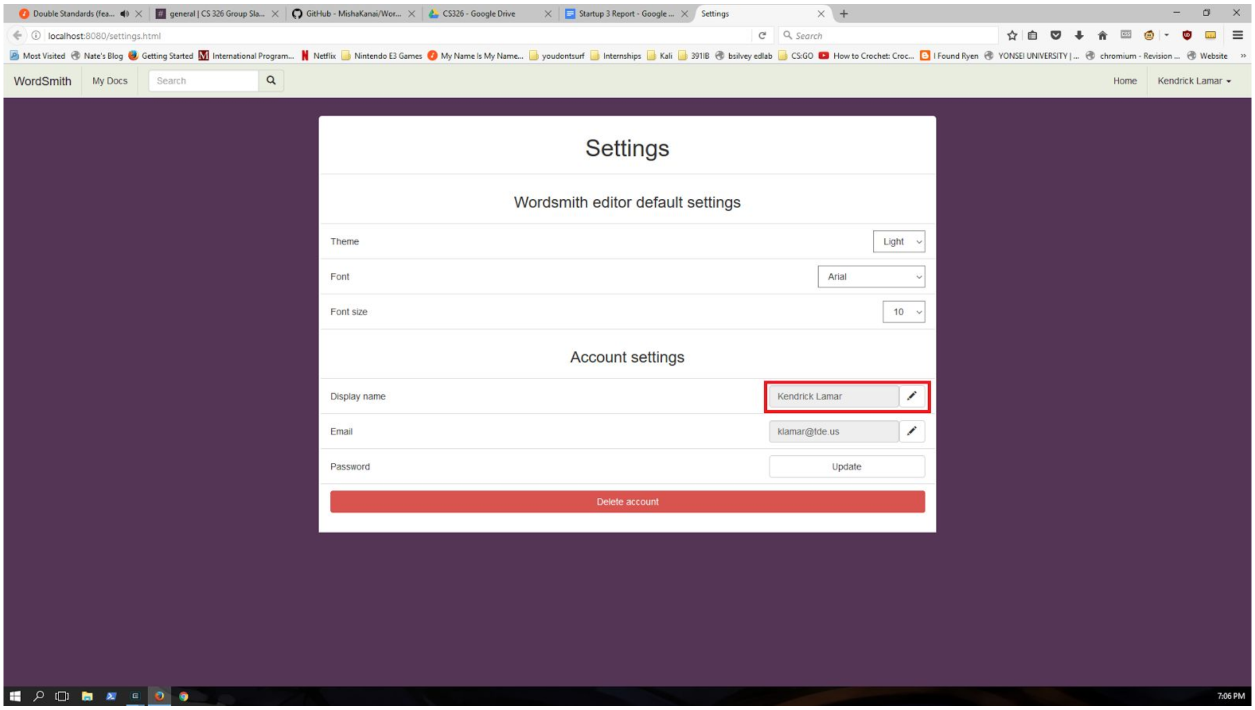
Display name: Kendrick Lamar

Email: klamar@tde.us

Password: Update

Delete account

- EditElement: The right part of an Element that contains an editable field. 2 Elements have this component on their right side.



Settings

Wordsmith editor default settings

Theme: Light

Font: Arial

Font size: 10

Account settings

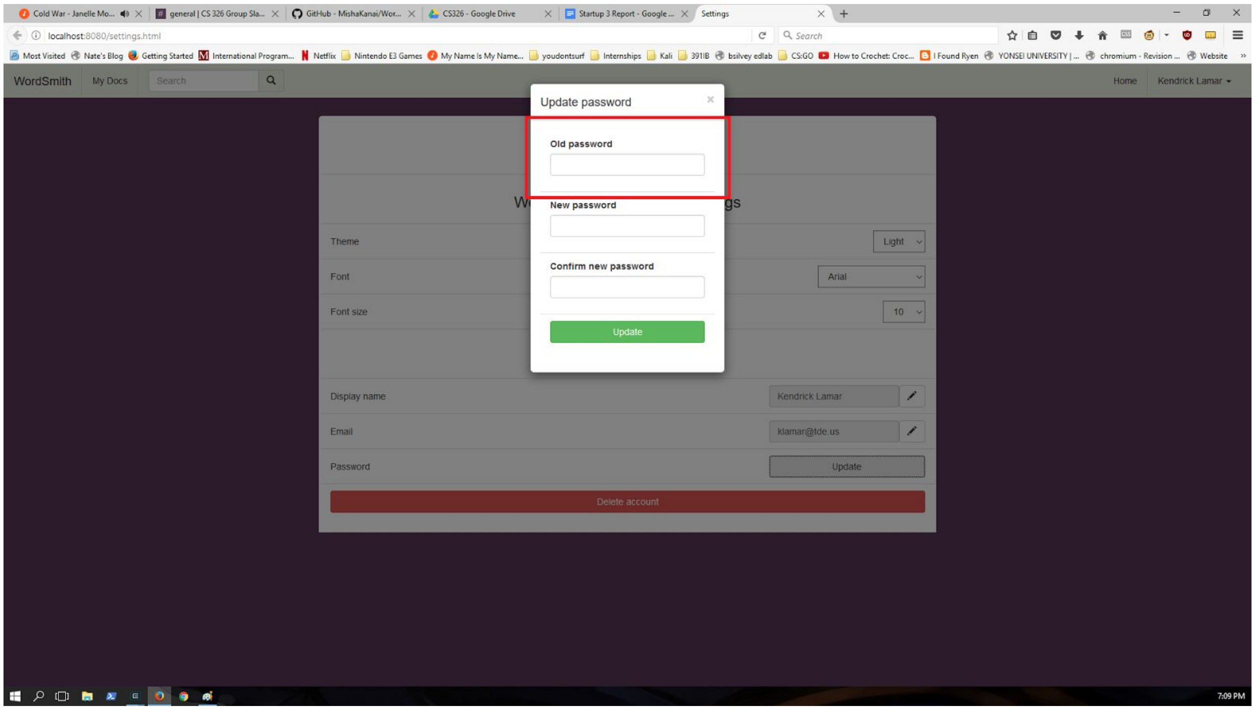
Display name: Kendrick Lamar

Email: klamar@tde.us

Password: Update

Delete account

- ModalElement: The repeating modal elements shown below. Each contains text and a child subcomponent, which will be explained next.



Theme: Light

Font: Arial

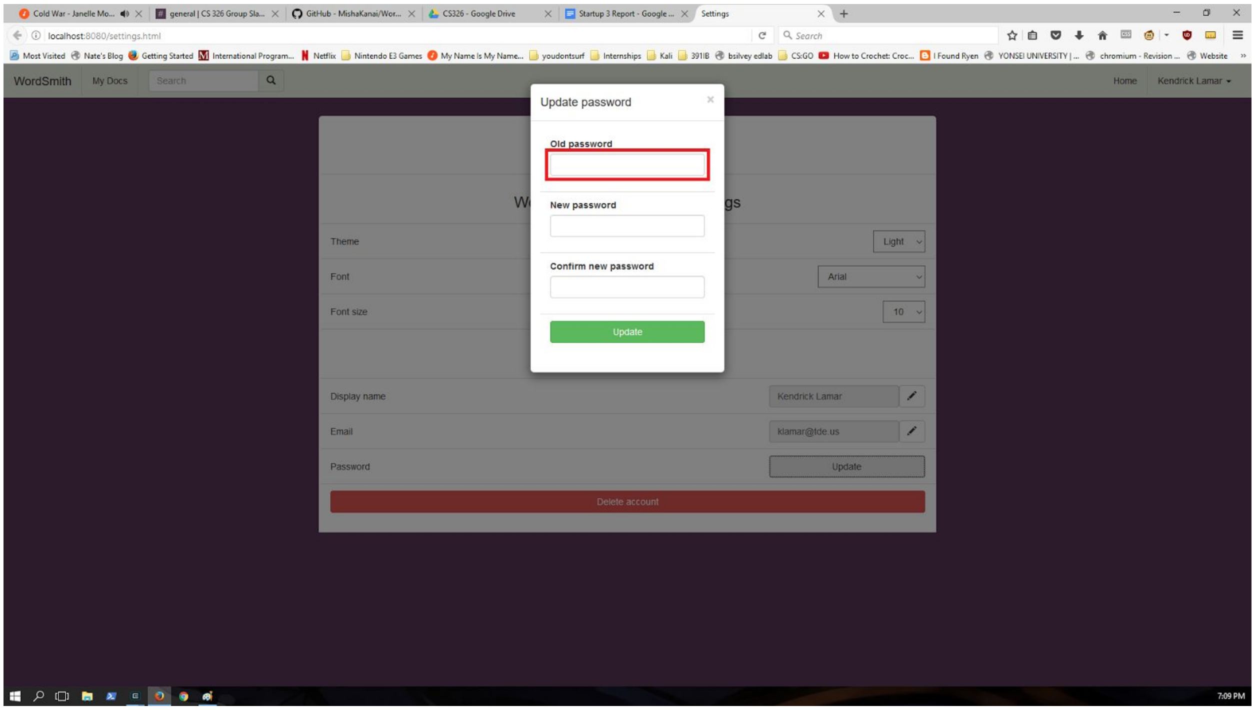
Font size: 10

Display name: Kendrick Lamar

Email: klamar@tde.us

Password: Update

- PasswordBox: The password field used inside of ModalElement here. Each has a unique ID specified.



Theme: Light

Font: Arial

Font size: 10

Display name: Kendrick Lamar

Email: klamar@tde.us

Password: Update

Misha Kanai:

I was responsible for creating the Workspace widget (sans the suggestions widget, which was done by Alan), TextEditor react component, and associated popover . This can be seen in the Workspace.js and TextEditor.js files.

Bugs:

Currently, we are not rendering the results of our API hit to the rhyming dictionary api. Also, on window width < 402px the TextEditor and SuggestionsBar overlap. Although this is a desktop oriented app, it would be preferable to have a flexible design, to accommodate users with multiple windows open. We could solve this by using media queries in our CSS, to increase the padding at the 402px breakpoint.

Furthermore, text-selection only works on single words, even when multiple words are selected manually. The design choice to select single words no longer applies to the direction of our app since we intend to incorporate the UrbanDictionary api, requiring the use of multi-word phrases. This will be fixed by modifying word-extraction code in app/js/applyClassesToWord.js when internal whitespace is encountered.

Components:

- Workspace (Workspace.js) renders the TextEditor and SuggestionsBar side by side.
- TextEditor (TextEditor.js) renders the react Quill editor, as well as the popover when words are right-clicked

Matthieu Oliveira:
I was responsible for converting the list of documents in the saved document page and turning them into React Components. The file responsible for this is SavedDocuments.js. I also wrote a small python script that given a word, scrapes urban dictionary and finds all the meanings of the given word. The purpose of this script is that it will integrate with the ‘urban dictionary’ pill in the suggestion bar pulling data for a user needs.

Bugs:

- When I now adjust the page size, the column alignment at a certain position is off. I am planning to fix this within the next couple days.

Components:
For the Saved Document page, the main component is SavedDocument.js which displays the documents a user has. I added a little functionality to the SavedDocuement.js file such as, for the amount of files a user has, the amount of files to output are based on that size. Currently this size is fixed and will be implemented properly once a user’s files are sitting on a server.

Alan Van Dijk:
I was responsible for converting the suggestions widget on the text editor page into React components. The files I wrote to do this are SuggestionsBar.js and Suggestion.js

Bugs:

- Currently, no data is being sent into each suggestion component, since their contents will be determined by a call to the rhyme API, and the user's selection of the category of suggestion via the popover when words are right-clicked. As it is right now, the suggestions bar is passed an arbitrary number of placeholder words.

Components:

- The main component of the suggestions widget is found in SuggestionsBar.js. This is passed the word to be displayed, the category of suggestion, and the suggestions themselves. The HTML in SuggestionsBar contains the pills labelled "Rhyme, Mime, Define, Slang". To determine which of these should be highlighted, a prop has been added to SuggestionsBar, because the user will have the option to determine which category of suggestions will be displayed via the popover in the text editor, and thus data will have to be passed down from the Workspace component to determine which will be highlighted as a result of this selection. Since the highlight is determined by CSS, each pill's className is determined by ternary operators on the prop. Another prop of SuggestionsBar is an array of all of the suggestions that will be displayed. In SuggestionsBar, each element of the array is mapped into a Suggestion component, passing the element as a prop into the Suggestion.