Greg Shepherd – CDA2520C – Web Framework – Module 01 – Course Project – 07/10/2020

Design Document for Note-taking Functionality

Overview

This single-pace application (SPA) aims to replicate the note taking functionality of Google Keep. This design document is an initial draft and will likely encounter broad change. The purpose of this note taking functionality is to allow users to create, edit, and delete notes stored online. The goal will be to implement this functionality and allow users to create their notes and interact with them as desired.

Scenario

To help develop the design I will go through a sample scenario of a user interacting with this SPA.

**Scenario:** John who is a student wants to create notes for his psychology course for an upcoming test. John wants to be able to review his notes on his mobile phone during his commute home after class. John utilizes our SPA to create his note of applicable terms to study for his test. On his commute home he is able to view the note from his phone and use this time to study. He notices a spelling error during this session and edits the note accordingly. When he arrives home, he is able to study again using his desktop platform. After succeeding at his test, he is able to delete this note and continue using the SPA to create more note for future courses.

Non-Goals

It’s important to bear in mind that while we are replicating the note taking functionality online that some features commonly associated with Google Keep will not be present in our SPA. These are as follows:

* Voice recognition
* Reminders
* Export to Google Docs
* Location services
* Social features
* Image functionality

Data Structures

The data structure being a method of organizing data to be later used. A way to accomplish this could be through the use of NodeList. This is an array type object storing an ordered list of nodes. This is especially relevant as Google Keep does utilize a default first in first out (FIFO) order. These NodeList objects are actually queries being run against the DOM structure, this allows automatic change of the NodeList. In this sense each note becomes a sibling of other notes while the note nodes may be children of the body element.

User Interface

The user interface is quite simple. Once on the SPA the functionality to create a note requires data entry and selection of a close button. Each note has a title section which becomes a label. When a note is selected all fields become available for editing with a close button. Other buttons are a possibility, however most have been identified as non-goals of our SPA. Some of these additional buttons include change color, delete note, add label, etc. The navigational components include the notes main page, edit labels, archive, and trash.

Data Interaction

When a user creates a note, it becomes a Node within the NodeList. The nodes being actual queries being run against the DOM allows changes to be reflected in NodeList objects automatically. If a note was to be archived it could be transferred to the archive NodeList. If deleted to the respective trash NodeList.

Flowcharts

\*Located on next page.



