# Specifications

## Overview

Researchers are increasingly using large amounts of Online Social Network (OSN) data to study how users interact with technologies, communities, and society at large. However, current approaches to dataset generation are often manual, time-consuming, and difficult to reproduce. To address these issues, our prior work introduced SMIDGen: a hybrid manual+computational approach for enhancing the replicability and scalability of data collection from OSNs to support mixed-methods research and analysis. The SMIDGen approach successfully integrates information retrieval and machine learning techniques with human expertise to aid in the development of search terms and data acquisition while incorporating predictive algorithms to accelerate manual coding and analysis processes. A prior case study focused on the collection and annotation of YouTube videos demonstrating that this approach can surface data that manual curation might otherwise miss, increasing the overall proportion of relevant data collected by subsequent queries. In this project, we focus on constructing a web service that scaffolds this process and makes advances in natural language processing, machine learning, and statistical analysis available through a combination of point-and-click, WYSIWYG, and interactive visual interfaces. If successful, this service will be available to support a growing and diverse community of researchers and analysts studying OSN data.

## Reading

Going through the first part of this paper would probably be very helpful for onboarding:

<https://www.eecis.udel.edu/~mlm/docs/2018-Mauriello-HCIL-SMIDGen-TechReport.pdf>

## GitHub

Code should be stored in the follow GitHub repository:  
<https://github.com/Sensify-Lab/SMIDGen>

> Pages could be enabled as an initial space to test html, css, and client-side JavaScript while back-end code would need to be run locally until hosting can be set up.

## Hosting

The platform will be initially be hosted locally and then transferred to the ECE/CIS Computing Lab servers under a URL similar to “smidgen.cis.udel.edu”. The main goal of the project is to scaffold out the SMIDGen dataset data generation process as described in this tech report

[smidgen.cis.udel.edu](http://smidgen.cis.udel.edu)

## 

## Google Share

Team files can be shared here:

<https://drive.google.com/drive/folders/1anFhpKSj_NSH3yPR9Q39ilvnRgz-OTBe?usp=sharing>

## Early Designs

Early designs for the application are available here:

1. <https://docs.google.com/presentation/d/10FBWX2HuSIA25ZqrlZK-rGnFUD_noZ3t/edit?usp=sharing&ouid=117544310133681741155&rtpof=true&sd=true>
2. <https://docs.google.com/presentation/d/1Yed6M-GLq41YJWaZAxb0kQHMmA2vvYsX/edit?usp=sharing&ouid=117544310133681741155&rtpof=true&sd=true>

## Core Features

* Basic Pages
  + (Public) Landing Page & About Page
  + (Public) Account Creation/Register (Username and Password, Link to Login, Captcha)
  + (Public) Login Page (Username, Password, Forgot Password Process, Link to Register)
* Projects Dashboard
  + Create a project
  + List of projects you own or have permission to access
    - Delete a project
    - Open a project
    - Rename a project
    - Add users to project (if owner)
    - Manager user permissions (view, edit, admin, transfer ownership)
  + Access help information
* Project (Collecting Data)
  + Setup: What information do we need to integrate with the Twitter API? What rate limits apply? Are there alternatives? Where do we save and configure this information on our platform?
  + Users should be able to specify a “seed list” of search terms
  + Users should be able to specify how many results to pull in for each term (e.g., top 200)
  + Users should be able to specify the type of search {singular, pairwise, or both}.
  + When the user searches, data should be pulled into a CSV-like data structure
  + Once results are pulled in, there should be some way for users to easily browse the tweet
    - e.g. could we create a browsing iframe?
    - Could we also surface some “relevance score”?
  + Users should be able to explicitly save their progress whenever they like
  + Users should be able to delete search terms from their “seed list”
    - Deleting a term should delete any data associated with the term in dynamic memory
  + After pulling in a set of data/tweets and agreeing that this is “good data”, the user should be able to run a word co-occurrence analysis and surface a new list of words.
    - Some piece of data should inform the user about the quality of the co-occuring words
    - Users should be able to add/remove words to a new expanded list of keywords
    - Users should be able to run new searches - through the system should be smart enough not to re-run prior searches unncessarily
  + User should have the ability to use filters provided by the Twitter API
  + Seed lists, expanded search term lists, data (tweets) should all be saved with appropriate metadata including owner, project, and timestamp.
    - Data should be saved with the search terms used to find the tweet

* Project (Visualization)
  + When the user has a set of data saved, they should be able to create a word cloud with various options for settings
* Labeling Tweets
  + Will want to fire up this tool (<https://github.com/cbuntain/collabortweet> ) and figure out how to integrate it into our application as the labeling mechanism; it makes heavy use of Node.js and third-party packages, like Express, Sqlite, etc.
* Topic Modeling
  + TBD
* Relevance Filtering
  + TBD
* Post-Relevance, Multi-Class Filtering
  + TBD
* Exporting & Import
  + Users should be able to download a full archive of their project workspace as a Zip file.
    - Search terms
    - Expanded Search Terms
    - CSV of Data
    - Project Meta Information
  + Users should be able to import this archive to restore their project even if it has been deleted

* Account Management
  + User can put in basic information about themselves (Name, Organization, Phone number)
  + Update email address associated with account
  + Update password
* Other Integration
  + Google Analytics
  + Logout when logged in.
  + Form field should have some validation to help users enter information correctly (e.g., Account Management)
* Documentation
  + README
  + Code comments
  + In-product help pages, tooltips, etc.