

QuickCanvass:

Improving Student Campaign Canvassing

Team Information:

- Project Leader: Jessica Ji (jnj5@princeton.edu)
- Luisa Goytia (lgoytia@princeton.edu)
- Sam Russell (samueljr@princeton.edu)
- Grace Turner (gracekt@princeton.edu)

I. OVERVIEW

We want to make it easier for Princeton students to run for student government or organize a campaign for a referendum. We will do this using a website-based platform and student data taken from TigerBook. It will have features such as basic login, the ability to collect and download survey results and voter data, and the ability to log individual reactions to particular voters. We will also have a customized function that takes an address and gives you nearby voters who have not been canvassed yet. Campaign Managers can create a more advanced account in order to coordinate their volunteers and customize their surveys. In this way, the platform will serve to coordinate canvassing campaigns and have a better understanding of voters population.

II. REQUIREMENTS AND TARGET AUDIENCE

All political campaigns need to keep track of two things - the people they've talked to and what those people believe. Campaigns for student government or ballot referendums at Princeton are no exception. The standard way that these campaigns do this is by canvassing - going door to door to talk to particular voters - but there is no platform for keeping track of the data campaigns collect from this. Currently, campaign managers at Princeton have very little idea of which doors they need to target and must manually manipulate an excel sheet if they wish to store any of this information.

With our solution, campaign managers can build up data they about constituents and get unique profiles of each voter. QuickCanvass also allows campaign managers to work together with their volunteers to contact particular groups of voters.

III. FUNCTIONALITY

Scenario 1:

Alex is a freshman running for student government in the fall. Alex lives in Wilson but is unfamiliar with the other residential colleges. He also only wants to promote his campaign among freshmen, but doesn't even know which rooms have freshmen in them. Alex starts a campaign on QuickCanvass as a Campaign Manager. Through QuickCanvass, he can see a list of all freshmen and their room numbers, and he has the ability to sort them by residential college. He also has an easy way to keep track of which freshmen he has spoken with and which he has not and which students are supporters of his campaign for later get-out-the-vote efforts.

Scenario 2:

Kate is president of a student group interested in passing a USG referendum to promote energy conservation. She wants to find students who would be interested in helping out on her energy conservation campaign, while identifying who supports her referendum but is not willing to volunteer. Kate's goal is to talk to a broad range of students from different class years, majors, and hometowns. However, she also wants to minimize the time that her members will need to spend canvassing. Kate starts a campaign on QuickCanvass as a Campaign Manager and adds five members of her student group as campaigners. Using QuickCanvass, they can coordinate their canvassing across different areas of campus.

David is a member of Kate's student group who signed up to canvass but is generally busy with other tasks. While studying in a friend's room one night, he finds himself with twenty minutes to spare and enters his current location into QuickCanvass. QuickCanvass provides him with information about the occupants of the rooms closest to him that have not been canvassed yet. David can spend then his twenty minutes conducting targeted canvassing instead of randomly knocking on doors or wasting time searching for information online.

IV. DESIGN

The QuickCanvass website and database will be hosted and deployed on Google Cloud Platform. All code will be hosted in a private Git repository.

FRONTEND

The QuickCanvass website will be built using the Django Python framework, HTML/CSS Django templates, and CSS Bootstrap templates. Student data is largely static and will only have to be updated once per school year, so JS/jQuery will be used to answer user queries.

BACKEND

QuickCanvass's student data is scraped from TigerBook in JSON format and stored in a Google Cloud SQL MySQL database. The database will be generated and queried in Python via Google Cloud Platform's Python API and connected to the frontend using GCP-and-Django-compatible Python.

The student database contains the following fields:

- City (hometown)
- Major
- Last name
- First name
- Email (netID)
- Class year
- Residential college
- Dorm room

V. TIMELINE

Week of March 13-19:

Design Document: March 19th

Week of March 20-26:

Have UI designed based on our finalized functionality design for the project

Create Project Status Website

Login / Logout

CAS login

Ability to remember people who created an account

Link database to frontend

Week of March 27-April 2:

Ability to create a campaign

Create dashboard for Directors and Volunteers

Ability for volunteers to sign up for a campaign with a code

Status website needs to be online: March 27.

Week of April 3-9:

Ability to provide a list of non-canvassed students near given location

Ability to upload data you just collected about those formerly non-canvassed students - hook up to database; implement UI for this.

Week of 10-16:

Prototype due: April 14.

Week of 17-23:

First-round user testing

Week of 24-30:

Alpha test: April 28.

First round of debugging and clean up in response to first-round user testing

Week of May 1-7:

Second-round user testing

Week of May 8-14:

Second round of debugging and clean up in response to second-round user testing

Optimizations

Demo Days

Final Document due May 14th.

VI. RISKS AND OUTCOMES

SECURITY

We need to make sure that the data collected through our application is secure and only accessible to the respective Campaign Managers. We will be using CAS authentication for each user to achieve this.

BACKUP

Having only one copy of the data collected can be a risk in the case where data gets corrupted for some reason. To address this, we might have a backup copy that can be used to restore data in case it is corrupted or deleted.

CALCULATING DISTANCE, MAPS AND FRONTEND VISUALIZATIONS

We are not using Google maps therefore we have to create our own algorithm that will return the closest non-canvassed rooms to the given location.

DATA DEPENDENCE

QuickCanvass's data is currently scraped entirely from TigerBook, whose longevity and continued maintenance is not guaranteed.

ISSUES RELATED TO GOOGLE CLOUD PLATFORM

We anticipate potential delays related to connecting different Google Cloud Platform services, such as linking the Google Cloud SQL database with the frontend and deploying the final product via Google Cloud Platform. Our worst-case deployment alternative is Heroku, which we should be able to link our Google Cloud SQL database(s) to via the Cloud SQL Python Client API.