# Polyforms

Flash: Max Zlotskiy, Michael Cheng, Andrew Wong, Gordon Lei
Period 9 Software Development Final Project Design Document

## Abstract

Our project is a form creator website, which allows you to create forms and gather responses. These responses will be stored in a database. You can then view the responses in a spreadsheet or create charts with them. Our goal, ultimately, is to make a lightweight survey service and integrate really cool data visualization capabilities.

## Features

- Registered users can make their own forms/surveys.
- Forms can include different types of questions
  - Radio buttons
  - Text responses
  - Number-only
  - Select all that apply
- the form creator can share the link to their form to the public. Respondents don't need an account to respond to the form (this can be changed, see below)
- The form's creator can view responses in a table or a saveable csv
- forms that want only one response per user will require login
- A form's creator can limit number of responses

## Component Map

diagram

## Component Description

Front End

- **stylesheet.css** ________
- **graphs.js** ________
- **d3.js** ________

Routes

- Move stuff from the table here

Python Files

- **__init__.py** contains request handling logic used by Flask

Database

- **Accounts**: stores user registration information
- **Forms:** stores metadata about every form that has been created
- **Questions:** stores data about questions (innertext, expected response type, order)
- **Options**: stores choices that will be displayed under radio-button-type questions
- **Responses**: stores all the responses to forms
- **Styles**: stores aesthetic changes made to a spreadsheet

# Site Map

DIAGRAM WILL BE MADE AFTER WE LIST ALL THE PAGES/ROUTES

| Route | Description |
|---|---|
| / | Home page. Has a toolbar with login button, forms/graphs search bar |
| /login | Login form. Redirects to root if already logged in |
| /results | Lists forms/graphs with the contained keyword |
| /my/forms | View a list of all your creations. Will also have a button to make a new form |
| /my/settings | Change account settings |
| /form/respond?id= | Fill out a response to the form |
| /form/edit?id= | The form's creator adds questions to a form on this editor page |
| /form/view?id= | Allows the form's creator to see the responses in a table. Graphs made using this data are also displayed on this page. |
| /home | Is the page you see after logging in. Has options to redirect you to making a form, viewing a form’s result, etc.  This could also redirect you to a random form to fill out. |
| /search | Route that handles the searching through databases for AJAX calls and other calculations |

# Database Schema

There will be a table containing a row for each form. Another table will store all the questions for all the forms. A third table will store the submitted responses to each question. Finally, there will be a simple user accounts table. We've organized them so that the first row has the column names, the second row has the data types, and the last row has example values.

**forms** Table: stores references to all the forms created as well as who created the forms. The “login_required” option will hold either 0 or 1 to represent True or False. if it is 1, then the person has to login into their account to answer the Form; else they can answer the Form without logging in. The “public_results” option will follow the same ”0 or 1 system” where if it equals 0 then only the creator of the form can view the data graphs. If it equals 1 then anyone can view the data graphs. The theme is the name of the html template.

| **form_id** | **owner_user_id** | **login_required** | **public_results** | **theme** |
|---|---|---|---|---|
| INTEGER PRIMARY KEY | INTEGER | INTEGER | INTEGER | TEXT |
| 1 | 1 | 0 | 1 | "purple.html" |

**responses** Table: stores responses to questions. Each record is one answer to one question on one form. When a user submits their responses to a form, their answer to each question is stored here. For a response, the answer to each question shares the same response_id. In the example below, the 25th response to form#1 (which has 2 questions) is shown.

| **form_id** | **question_id** | **user_id** | **response_id** | **response** | **timestamp** |
|---|---|---|---|---|---|
| INTEGER | INTEGER | INTEGER | INTEGER | BLOB | TEXT |
| 1 | 1 | 1 | 25 | “Not a pineapple” | 2008-12-14 03:00 |
| 1 | 2 | 1 | 25 | "Bob" | 2008-12-14 03:00 |

**questions** Table: stores all the questions and what kind of question they are. Although all forms share this table, you know which form each question belongs to based on its form id. The order of the questions is based on question_id (unique for each form, not for the whole table). The type column determines what kind of response this question expects. "Short" is a one-line text response, "long" is a textarea, "number" is a numerical response, and "choice" offers a set of responses (like radio buttons).

| question_id | question | type | form_id | min | max |
|---|---|---|---|---|---|
| INTEGER | TEXT | TEXT | INTEGER | INTEGER | INTEGER |
| The order of appearance of this question on the form | “...?” | "short", "long", "number", "choice" | the id of the form this question belongs to | how many options must be selected, characters typed, or the least value for number questions | how many options can be selected, characters typed, or the top value for number questions |

**Options** Table: stores all the options for “choose your answer” questions (ex. Questions with <select> or radio buttons)

| form_id | question_id | option_index | text_user_sees | value |
|---|---|---|---|---|
| INTEGER | INTEGER | INTEGER | TEXT | TEXT |
| 12 | 1 | 0 | “I like apples” | “apple” |

**Accounts** table: stores the registration information for each user

| user_id | username | password |
|---|---|---|
| INTEGER | TEXT | TEXT |
| 1 | “pineapple” | “s231dsa321sda321321” |

**charts** table: stores the configuration for each chart (basically tells you what type of question it is)

| question_id | type |
|---|---|
| INTEGER | TEXT |
| 1 | “select” |

**styles** table: stores aesthetic settings for the spreadsheet. While every cell has default properties (width, white background), the user can make some changes to them, and those changes are stored here. In the example below, row 1 has a height of 40px.

| **form_id** | **row** | **column** | **property** | **value** |
|---|---|---|---|---|
| INTEGER | INTEGER | INTEGER | TEXT | TEXT |
| 1 | 1 | -1 | height | 40px |