

# Johnson Controls Polling Application

Caitlin Fanning, Teddy Hung, Mia Kim, Nicholas Loyd, and Ruiyou Ma

Faculty Mentor: Eric Jiang, PhD  
Department of Computer Science



## Motivation

- Many companies such as Johnson Controls - a multinational provider of commercial building services - are often tasked with making a large range of decisions, from who will present at the next meeting to which product or path would be most suitable for any given project.
- Detailing a specific question with valid choices can be difficult in a purely oral form, and chain emails between different departments can easily be lost in a busy employee's inbox.
- Our solution for simplifying the decision-making process is a web-based polling application.

## Overview

The screenshot displays the application's user interface. On the left, there is a 'Register' form with input fields for 'First Name' (containing 'Diego'), 'Last Name' (containing 'Torero'), 'Username' (containing 'dtorero'), and 'Password'. Below the form are 'Register' and 'Cancel' buttons. On the right, a 'Welcome, Diego!' message is shown, followed by a prompt to 'Begin by creating a new poll or reviewing an active poll.' Below this, there are two sections: 'Active Polls:' and 'All registered users:'. The 'All registered users:' section lists several users with their names and a 'Delete' button next to each name.

Registration and Basic Home Page Layout

## Polling Application

- This project aims to develop an application that will allow employees to create polls and vote on topics with a multitude of features and options aimed at streamlining the decision-making process.
- Through this application, polling administrators will be able to collect topic-specific information from groups of employees efficiently and successfully on a wide array of topics, and succinctly redistribute the results.
- Our developed application will also give administrators the ability to keep track of decisions made by separate groups, and give them an overview of the various teams they have created.

## Front-End

The web application's front-end uses Angular, which is a framework built in TypeScript. The Angular framework is based on components, which allows our app to be developed with consistency and modularity.

As team members work independently on building out components, the base begins the same way. Although the end result of each component may vary, overall the structure of each remains similar.

The screenshot shows the 'Create New Poll' form. It has a title 'Create New Poll' and a navigation bar with 'Home', 'New Poll', and 'Logout'. The form contains several input fields: 'Poll Title:', 'Options:', 'Expiration Date:', 'Add User or Group:', and 'Visibility Limit:'. There are also buttons for '+ Add Option', '+ Add User', and 'Submit'.

```
new-poll.component.spec.ts - Untitled (Workspace)
1 import { async, ComponentFixture, TestBed } from '@angular/core/testing';
2 import { NewPollComponent } from './new-poll.component';
3
4 describe('NewPollComponent', () => {
5   let component: NewPollComponent;
6   let fixture: ComponentFixture<NewPollComponent>;
7   beforeEach(() => {
8     TestBed.configureTestingModule({
9       declarations: [ NewPollComponent ]
10      });
11     fixture = TestBed.createComponent(NewPollComponent);
12     component = fixture.componentInstance;
13     fixture.detectChanges();
14   });
15   it('should create', () => {
16     expect(component).toBeTruthy();
17   });
18 });
```

Example of the new poll component set-up and corresponding view: Angular allows very different components of our project (e.g. login, registration, new poll) to be built in a similar manner.

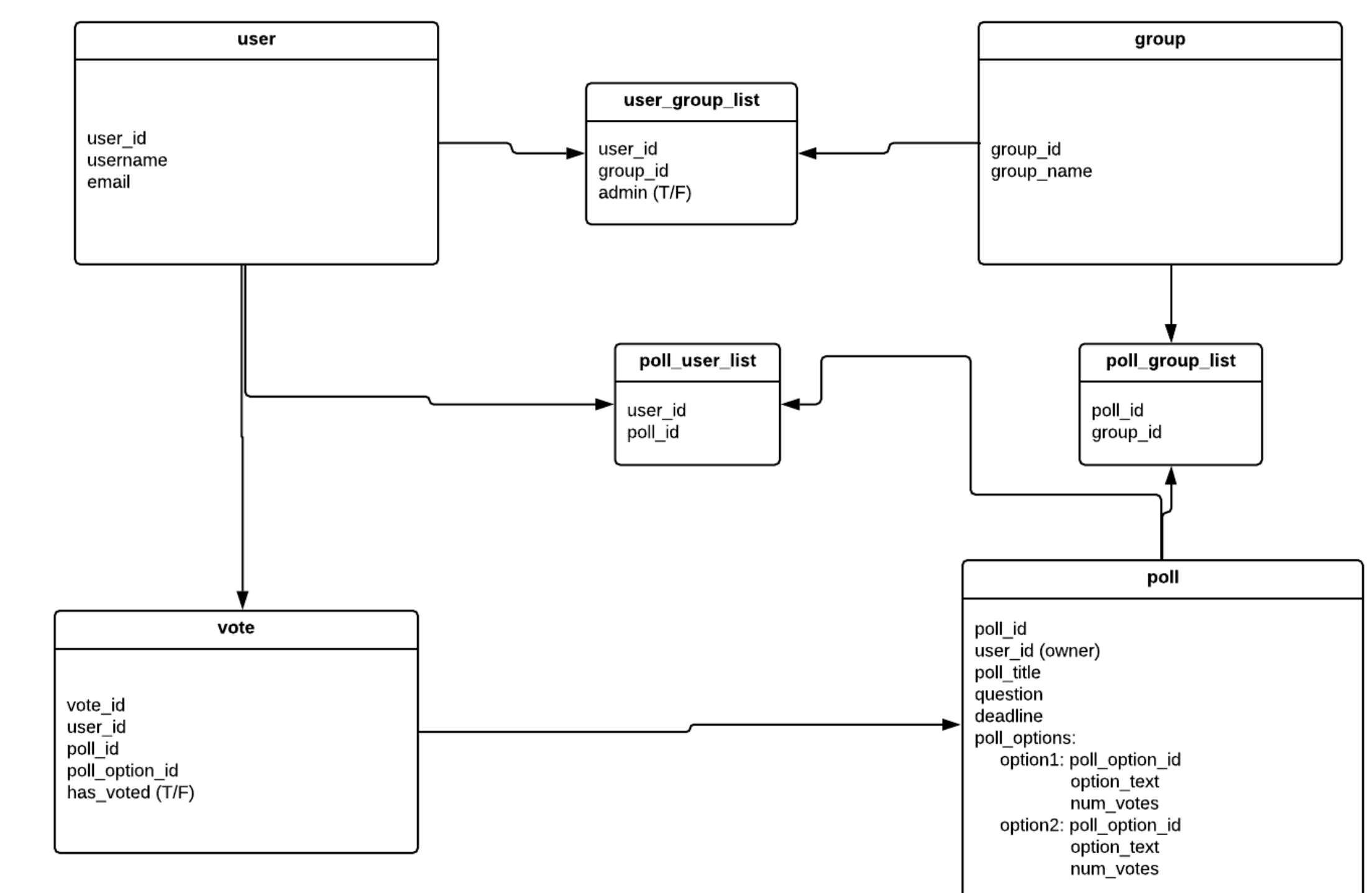
Angular also provides the project with modularity. Components and services are organized into modules, which can be thought of as containers. Each container or module can contain a feature or reusable block of code.

Modules are used to factor organization into the application, but also offer the team an opportunity to constantly redefine our understanding of successful use. As the project moves forward with an increasing number of components, a better comprehension of modularity is demanded with each additional component.

## Back-End

The web application's back-end is built in the Spring Boot framework, which partially automates the configuration of the application. The front-end communicates with the back-end through plaintext endpoints, and the back-end queries the database as needed before returning the requested information to the front-end.

The application uses a MongoDB database, a NoSQL database that stores data as JSON documents. Each data object in the backend (users, polls, etc.) is reflected as a collection within the database.



Database Structure: Basic hierarchy of the communication between multiple databases

## Future Work

In summary, this polling application aims to simplify the oftentimes complex process of workplace decision-making to increase productivity and significantly improve office workflow. We hope to complete the core functionalities of the application by the end of this semester, and future work on the project may include implementing profile-based security, so each active poll will be visible only to members of its group during the voting period. This is done to ensure most of the control of the polling process is given to fewer users, thereby simplifying the task, while giving administrators more responsibility.