

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Implement Candidates Database](#)

[Task 4: Implement Candidates Service](#)

[Task 5: Implement Authentication Service](#)

**GitHub Username:** `chevelle88`

## Candidates Corner

### Description

Candidates Corner showcases candidates for the upcoming chapter election. It highlights the candidates' profiles for each chapter office and allows members to rate the candidates.

Candidates Corner removes the annoyance of checking your email for the latest information from the chapter president.

### Intended User

This app is for members of the Annapolis Alumnae Chapter of Delta Sigma Theta Sorority, Inc.

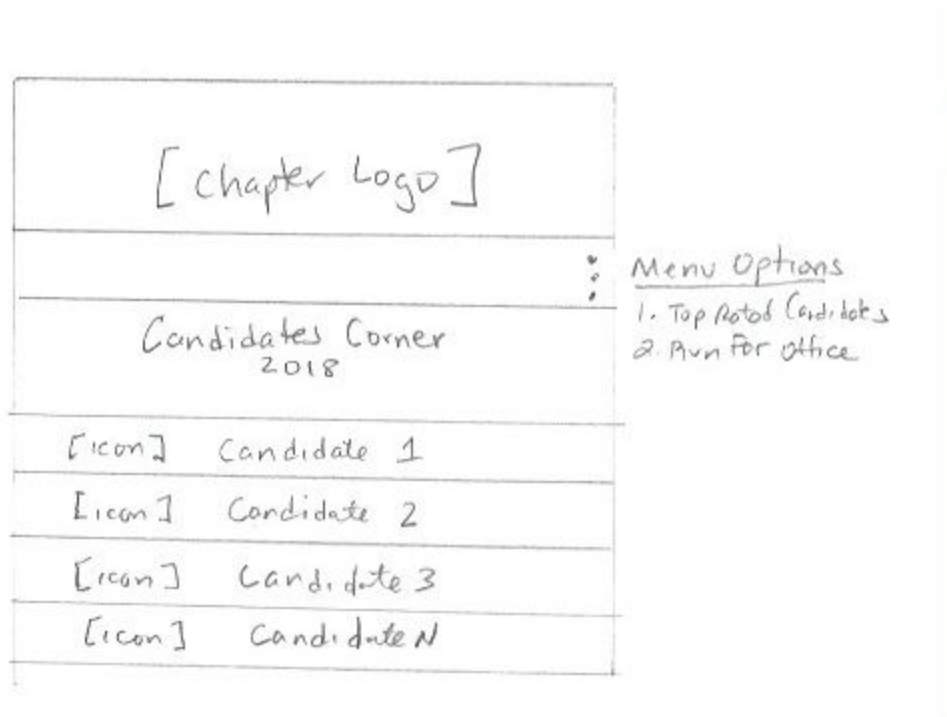
## Features

- Retrieves the latest candidates information from an external data storage
- Displays the selected candidate's information
- Saves candidates' ratings
- Shows the top rated candidates

## User Interface Mocks

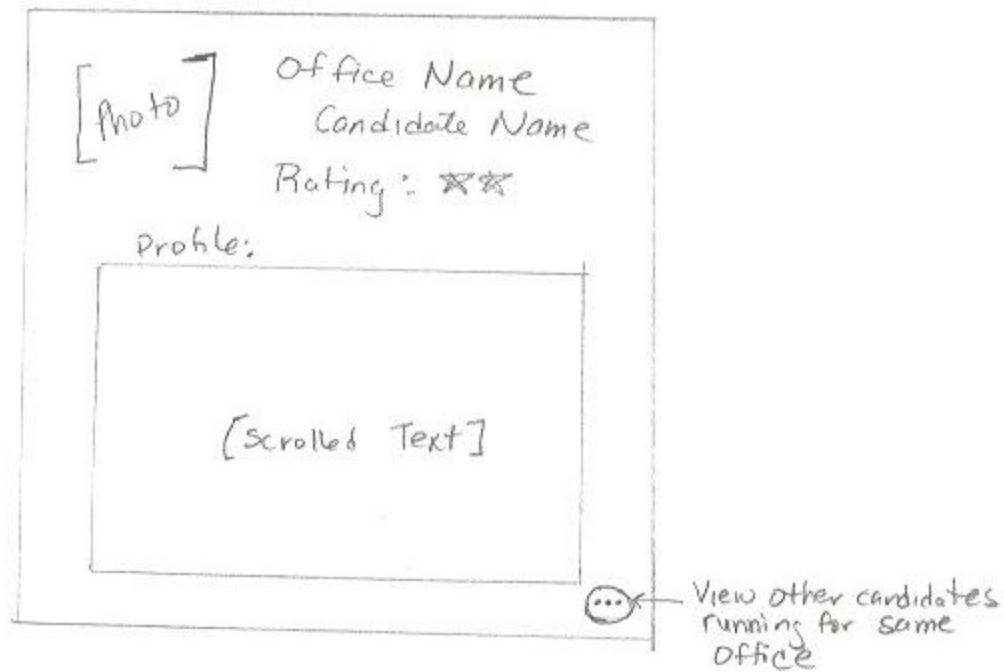
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

### Screen 1



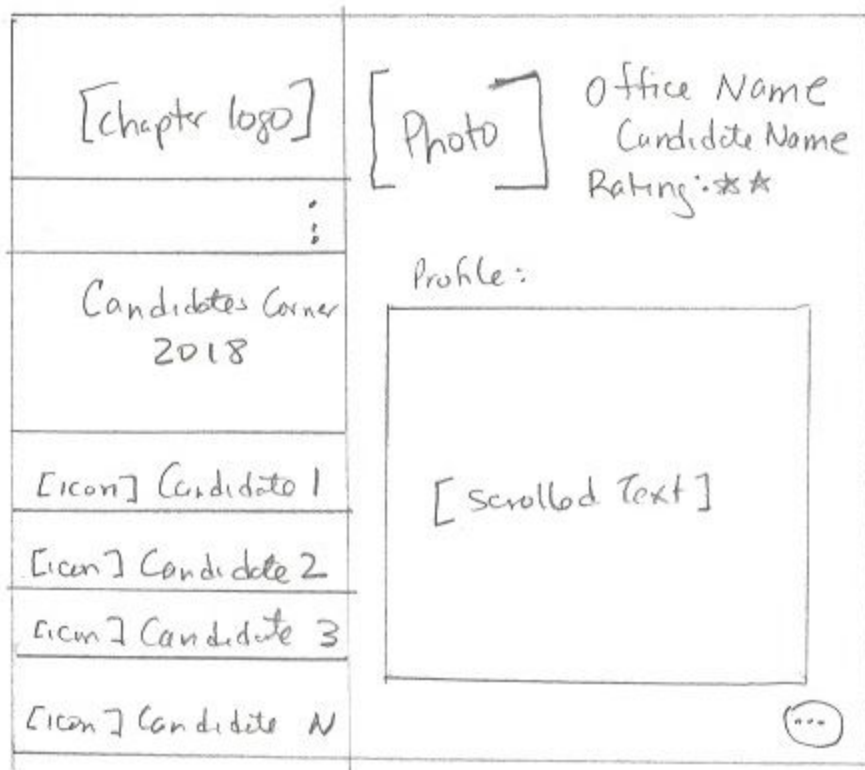
The home screen lists candidates for all chapter offices. On this screen, the user is allowed to select a candidate to view her campaign information. Also, there is a toolbar menu that provides options to views the top-rated candidates and send an email to become a candidate.

## Screen 2



The detail view shows the candidate's campaign information. This view also allows the user to rate the candidate and view other candidates running for the same office.

### Screen 3



Shows a landscape view of the app.

### Key Considerations

How will your app handle data persistence?

Candidates Corner will persist the candidates' ratings to shared preferences and will not persist the candidates information. It will use a Content Provider to retrieve the candidates data from the external data storage to retrieve the list of all candidates for the main view and specific candidate's information for the details view.

Describe any corner cases in the UX.

On the details view, the more button will only be shown in there are two or more candidates running for the same office.

**Describe any libraries you'll be using and share your reasoning for including them.**

Candidates Corner will be using the following libraries.

1. Picasso to handle image loading of candidate's photo
2. Firebase to authenticate chapter members
3. RestSql to handle database calls to an external MySQL database

**Describe how you will implement Google Play Services.**

Candidates Corner will use FirebaseUI to sign-in users for email and password authentication. Then forward that information to Firebase Authentication service to validate financial members of the Annapolis Alumnae Chapter. Once verified, the user information will be populated in the Run For Office dialog to send an email to the chapter's Nominating Chair if a member wants to run for a specific office within the chapter.

## Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

### Task 1: Project Setup

- Update Android Studio with the latest upgrade.
- Import project, Capstone-Project.git, from github into Android Studio.
- Modify gradle script to include library dependencies.
- Use Android Asset to download and move icons into the project.
- Create different sizes of the chapter logo to copy into the project.
- Add app to Firebase account.
- Setup a guest account to be used for reviewer.

### Task 2: Implement UI for Each Activity and Fragment

- Create a UI layout for MainActivity
- Create a menu layout
- Create a UI layout for DetailActivity
- Create a fragment layout to view the app in landscape

- Create a UI layout for the RunForOffice dialog
- Build the UI classes to support the app's activities, fragments and dialog

### **Task 3: Implement Candidates Database**

- Design database schema
- Use MySQL to create the database
- Load the candidates' data

### **Task 4: Implement Candidate Service**

- Create the candidate service to make RESTful calls to the external data storage
- Create a content provider to work with the candidate service and UI

### **Task 5: Integrate FirebaseUI for Authentication**

- Modify build script to include the dependencies for FirebaseUI
- Create a custom scheme for FirebaseUI for consistent look-and-feel
- Add code to execute the authentication activity