**GitHub Username**: JimNtexas

# Electoral College Calculator v1

## Introduction

Way back in 1995 I was driving to work and heard a person on the radio discussing the 1996 presidential election between Bill Clinton and Bob Dole say 'Dole can't win because he can't carry California'.

At that time I was working on something called a 'web browser', a tool for using this new http protocol. I figured that at least one of the 23,500 web sites then in existence would include a calculator of some sort for gaming the Electoral College System.

There wasn't, so I had to invent one. Our company was experimenting with a new language called 'Java', that was then in Beta. I got permission to use some of my time to develop a Java Applet for this purpose. I worked on this app for several years, it supported all U.S. Presidential elections from 1789 through 2008. Users could load a historic election and change the results to explore different possible outcomes. The applet is still available at http://grayraven.com/ec2/ , but no current web browsers support this old Java applet.

The purpose of my Capstone is to create an Android application that will allow users to review historical elections, and to create their own projections for the 2016 Presidential election, and to save and share their projections with other users.

## Description

The ‘EC Calculator’ version one will allow the user to select and display historical electoral college election results. It will allow users to ‘save as’ historical election results into a user created election file, which the user can then modify and save in a cloud database. Users will be able to share their election files with other users.

Version one of the app will support at least the historical election results from 1996, 2000, 2004, 2008, and 2012, and will support user projected results for the 2016 elections.

Version one will support tabular display of results for two political parties. Future versions will support all past elections, multiple parties, and a clickable svg based map of the United States.

## Intended User

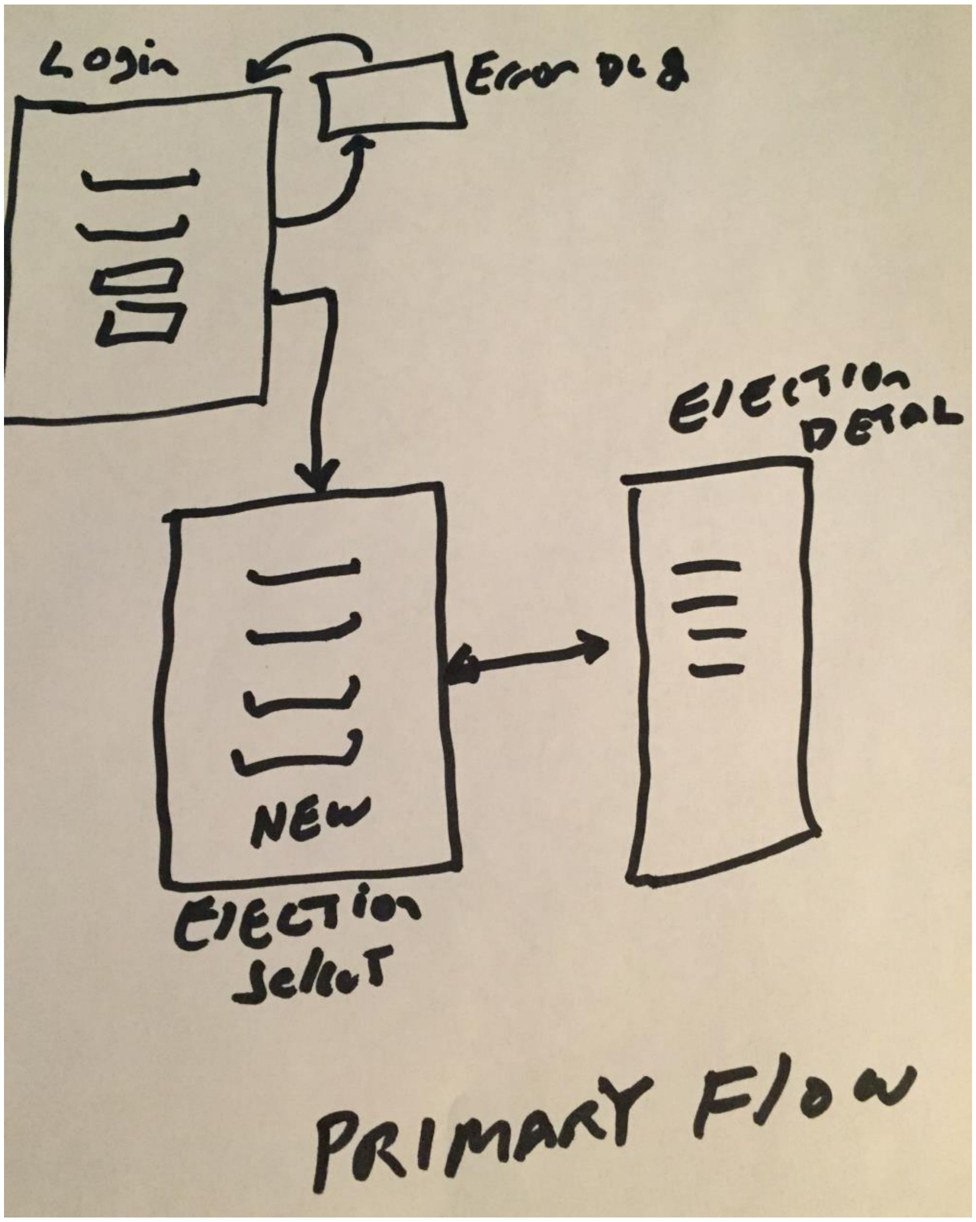
This application is intended for persons interested in reviewing past elections and projecting possible outcomes of the 2016 election.

## Features

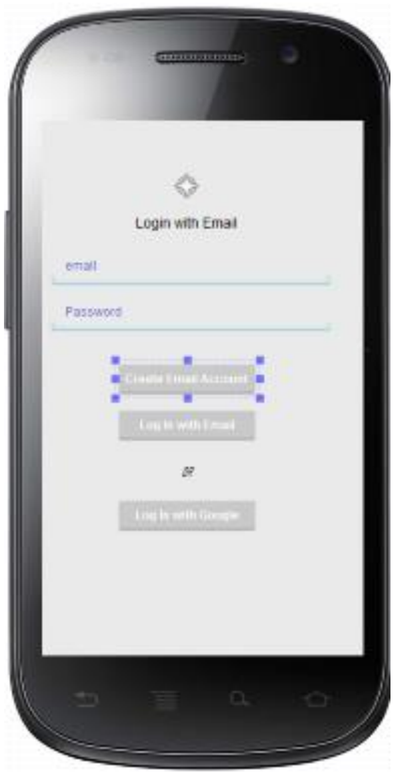
- Displays tabular results of U.S. Presidential elections held between 1996 and 2012.
- Uses correct electoral college vote distribution for each census decade.
- Allows users to login using an email/password pair.
- Allows users to login using Google OAUTH
- Allows users to save 2016 projected results
- Users can ‘save as’ historical election results as a basis for their projects.
- Allows users to share projected results.
- The app supports offline usage; cloud synchronization will be handled by Firebase.
- The app will display advertising

## User Interface Mocks

### General UI Flow



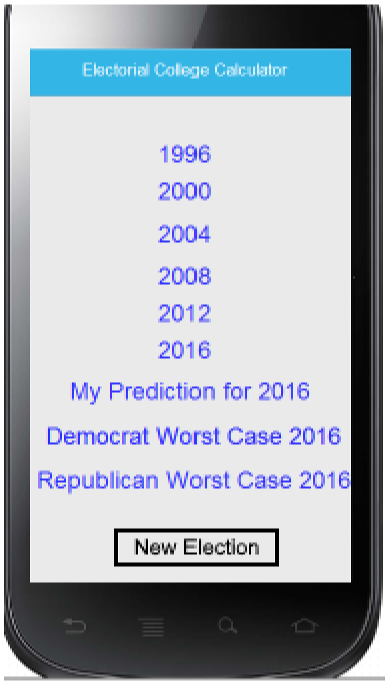
## Login Screen



Login screen implemented using Firebase for email/password and Google OAUTH.  If there is a problem during login an error dialog will be displayed.

After a successful login the election selection screen will be displayed.

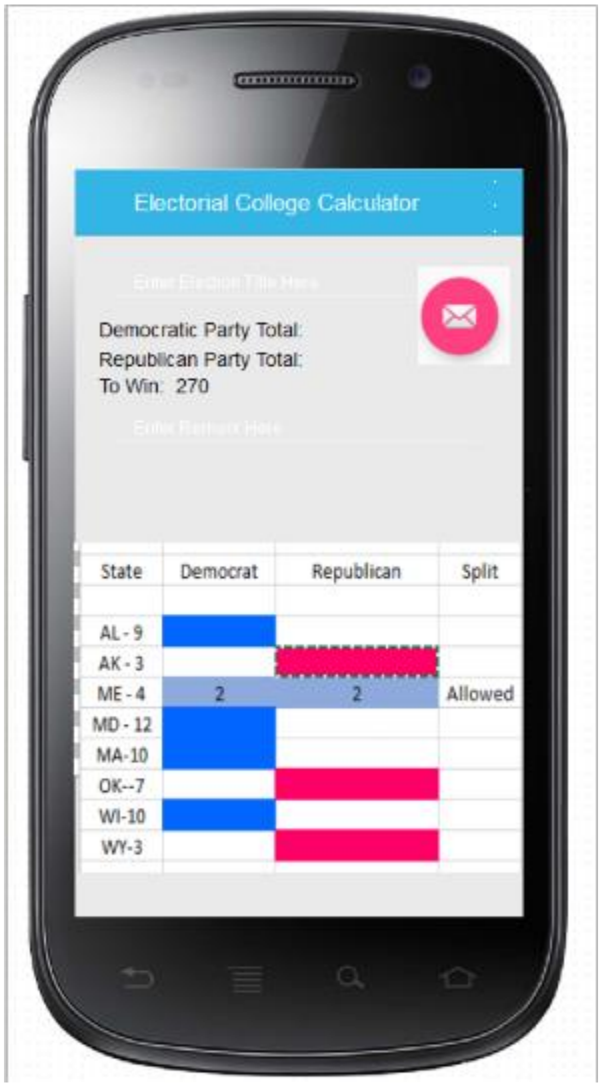
## Election Select Screen



The Election Selection Screen is a scrollable list of available elections. Historical elections are listed first, then user created election scenarios are displayed.

Clicking an election displays the Election Detail Screen, clicking ‘New Election’ displays a blank election detail page. Note: I may dispense with the ‘New Election Button’ in favor of a +FAB on the detail screen of an existing election or a menu selection for new elections.

## Election Detail Screen


Electorial College Calculator

Enter Election Title Here

Democratic Party Total:
Republican Party Total:
To Win: 270

Enter Remark Here

| State | Democrat | Republican | Split |
|---|---|---|---|
| AL - 9 | 🟦 | | |
| AK - 3 | | 🟥 | |
| ME - 4 | 2 | 2 | Allowed |
| MD - 12 | 🟦 | | |
| MA-10 | 🟦 | | |
| OK--7 | | 🟥 | |
| WI-10 | 🟦 | | |
| WY-3 | | 🟥 | |

The election detail screen displays an election summary at the top of the display, and a scrollable table listing each state and the total number of votes allocated to that state in the first column.

A scrollable table with a row for each state is located under the election details.

When the user clicks on the Democratic or Republican cell for that state's row the cell's background is changed to blue or red. The vote count in the detail section is updated.

When a party has 270 votes the detail screen visually indicates a winner with a green check mark or other visual indication.

In the case of a state that allows split electoral votes clicking on the ‘Split’ cell will allow the user to enter numbers in both party cells, which will then display with a purple background. The application must ensure that split voting state totals equal the total votes allocated to that state.

Clicking the envelope FAB will enable the user to share this scenario with other users.

This screen includes a menu with the selections ‘Load’, ‘Save’, and ‘Save As’. The ‘Save’ menu item is disabled for historic elections.

## Key Considerations

### How will your app handle data persistence?

User login and election data will be stored in a json object based Firebase cloud. Decadal vote distributions for each state will be held in an app local database, accessed through a content provider.

### Describe any corner cases in the UX.

If a user ‘saves as’ an election prior to the 2004 election then the app must ensure that vote distribution by state is correct for the current 2004 – 2020 decade if that is what the user intends.

Decide how to handle user who tries to login with Google, but doesn't have a Google account.

If the content provider is used in conjunction with an AsyncTask, before the app doesn’t crash if the screen orientation changes while the task is running. Perhaps prohibit rotation during the task.

The states of Maine and Nebraska do not have ‘winner take all’ Presidential elections. Instead they allocate one electoral vote per congressional district. The Election detail screen must allow users to split votes between parties for these states. Historically, there have been many elections in which ‘winner take all’ was not used by some states.

### Describe any libraries you’ll be using and share your reasoning for including them.

The application is based on Firebase legacy for data storage, user authentication. Google advertising will be supported. The app will use a publish and subscribe event notification library such as Eventbus. Images (if required) will be supported with Picasso.

I may use a library to generate tabular display of result data.

## Next Steps: Required Tasks

### Task 1: Database design and implementation

a. Using the data file I have from the Java applet, design and create a Firebase database modeling a Presidential election. Populate this data with historical data, which should be read only.
b. Implement POJO classes corresponding to the Firebase json data.
c. Design and implement a sqlLite database to hold electoral vote distribution by state for each supported election. Implement Content Provider/Content Resolver access for this data using Asnyc task or Intent Service.
d. Implement a Java class that combines election data with electoral vote distribution to represent a particular presidential election. This class will be the basis for election sharing and displays.
e. Design a file format for exporting and loading election scenarios.

f. Implement sharing election files via email.
g. Ensure the application works with local when offline, and that Firebase synchs application data when the network becomes a available.

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

Login screen:

Implement using the example in the Udacity 'Firebase Essentials for Android' as a guide. The example login screen is a bit awkward to use, I'll try to make a simpler login screen for this application.

Election Select Screen:

If possible, implement this list using a Firebase list adapter to scroll available elections.

Election Detail Screen

This screen will need through testing, as it implements the key application functionality. I hope to make the scrollable state results list as much like an excel spreadsheet as possible.

The States table should support sorting by state abbreviation or number of electoral votes.

## Task 3: Incorporate Google Advertising.

- Determine where to include advertising displays.  Right now I'm leaning towards Interstitial ads between the selection screen and the detail screen.

## Task 4: Implement election sharing.

- Implement loading and saving elections using the Election Detail menu or other GUI element.
- Investigate if sharing directly between Firebase user is possible within version one time constraints.