Galway West 2016

Application Design Document

v1.0

Designed and Developed by:

Hugh Moran

G00216157

1 Introduction

1.1 Purpose

The purpose of this design document is to give a brief outline the application design and its general purpose and usage. It was assumed throughout that the application would only be available for particular users to go and poll the general public, rather that it being a publically available application.

1.2 Scope

The scope of the project was to create an Android Application which would provide the functionality that will allow the user to take polls of voters for the 2016 General Election in Galway West. The application is required to ask the relevant questions of an individual and has that information stored in an SQLite database. The information that is to be stored in the database should then be queryable through parameters that could be chosen by the user and the required data returned to be viewed. It is also required for the application that the results of the poll are displayed, this is to show the candidates and other criteria's which were polled that got the highest percentage of the vote.

1.3 Use of Document

The use of this document will outline the various design decisions that were made during the development of the application and any other assumptions that were made along the way. Inside the document, you will find various screen shots that will highlight each area use and any assumptions that were taken at that stage. Also, any difficulties that arose in the development of the application will be illustrated also.

2 Application Design

2.1 Home Page

The home page *figure 1.* of the application shows a simple design with just two buttons. The first button add new poll allows for a new poll to be taken from a member of the general public. The second button will allow the user of the application to view the required stats at any point throughout the day while out taking the polls of the general public.

The logo at the top of the page was taken from the election logo its self which was used for the 2016 General Elections and adapted to fit the page for the application. The logo allows the application to look more user-friendly and not just a plain background that you'd have for a more organisation orientated application.

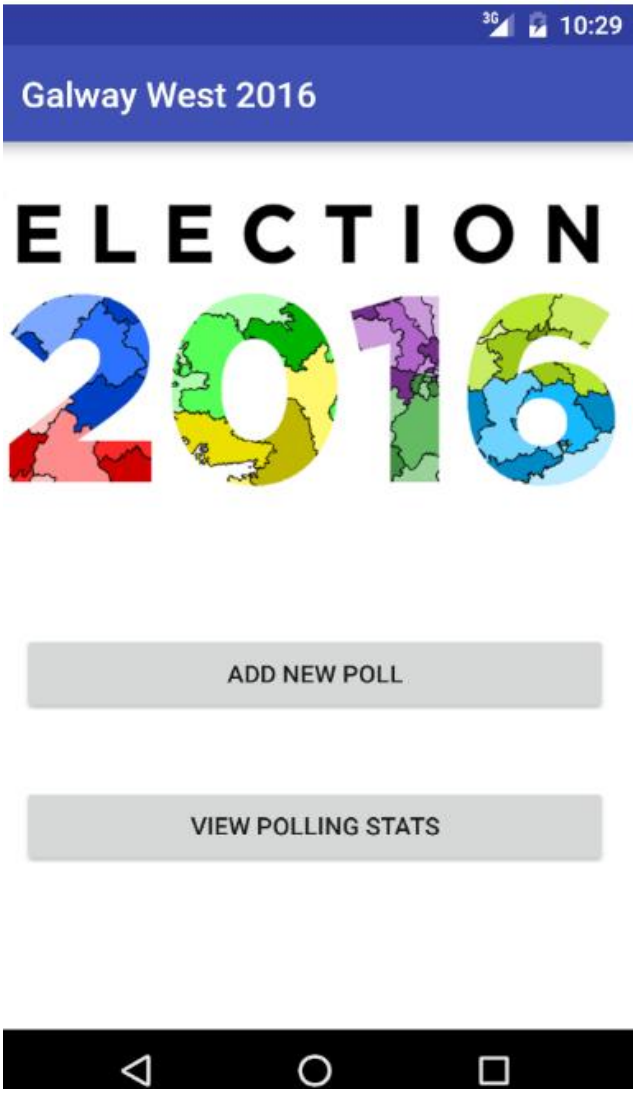

Figure: 1 Home Page

2.2 Disclaimer Page

The disclaimer page *figure 2* allows the user to read a brief description of the application and its usage. If the user does not agree with the terms they simply hit the back button which will take you back to the homepage. But if they accept the conditions by checking the check box the continue button will appear and allow them to continue with the poll.

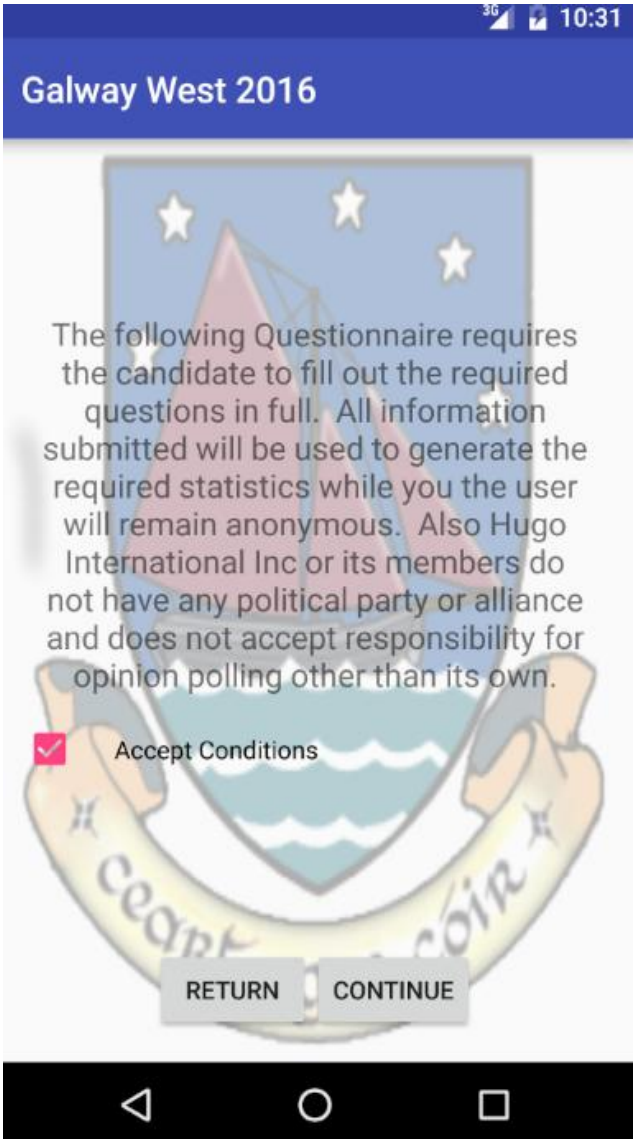

Figure: 2 Disclaimer Page

2.3 Polling Pages

Each of the polling pages are very similar as can be seen from *figure 3* each page contains a selection of Radio Groups and Spinners. Also to poll the users age a Text Edit was used which keeps 'Enter Age Here' as a hint and I'll disappear upon typing the age through the number pad. All information was added to a string then sent through from page to page in an intent. The bundle was used to recall them on each page to allow the information to be passed to the last page. Each page also has a transparent Galway crest to make the writing easier to read and so that it is more pleasing to the eye for the person taking the poll.

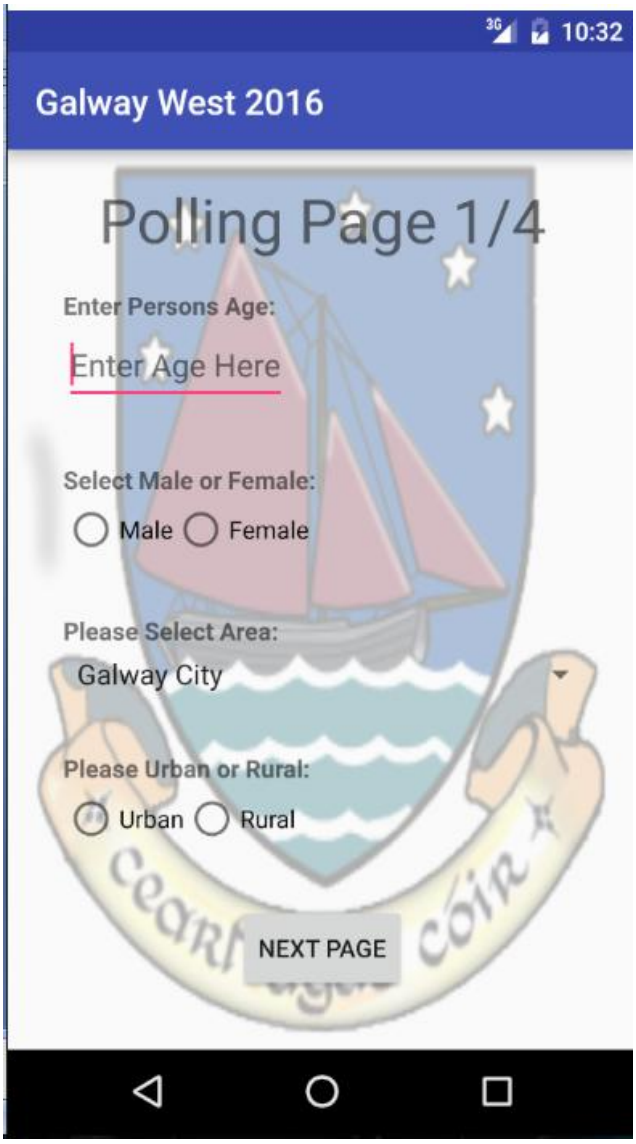

Figure: 3 Page 1 of Poll

2.4 Custom List #1

The custom list view is an easy user-friendly way for the user to choose who their first preference candidate is *figure 4*. When the user chooses their candidate that is stored in a string and it is from this page then that the information is sent into the database. Upon submitting the data the application immediately sends the user back to the homepage where they again have the choice of entering another poll or looking at the results.

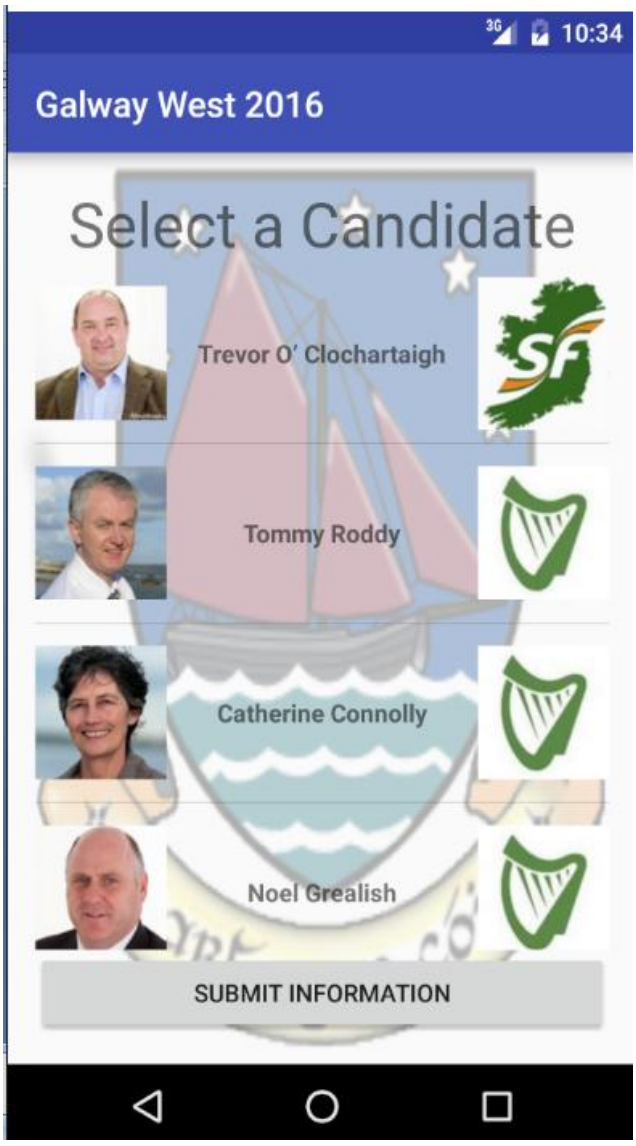

Figure: 4 Select Candidate

2.5 Stats Option Page

As this page figure: 5 is more for the owner of the app the crest has been taken away as there is no real need for it to be as appealing to the user. These pages are only here to serve the function of allowing the owner to view all the required stats. The first button leads to a query page where the user can query through 3 parameters. The second button allows the user to view the results of the poll for the candidates, party and party leaders. The final button will allow the user to view the full database without having to pass it any parameters.

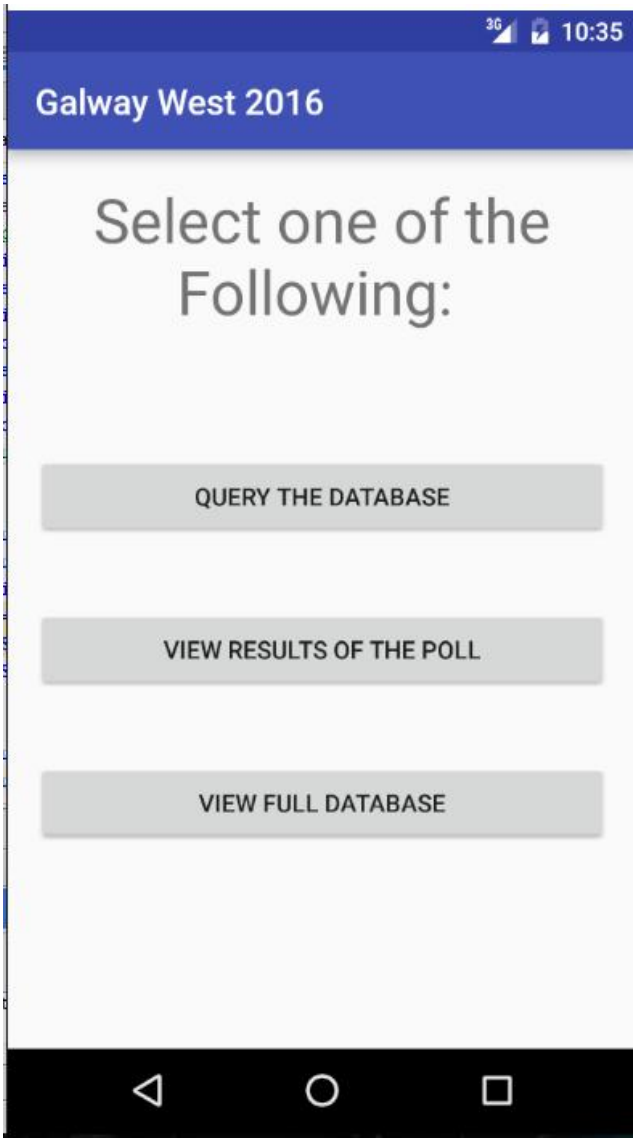

Figure: 5 Results Option

2.6 Query Page

The Query *page figure: 6* to design one of the easier pages. With three spinners and two buttons to select your parameters and to submit them to the database to be queried. Each parameter is divided by using the <View> command in the XML allows the queries to be separated into sections on the screen. The two buttons at the bottom of the page allow the user to return home to the home screen or view the required parameters that are bundled through an intent to the next page.

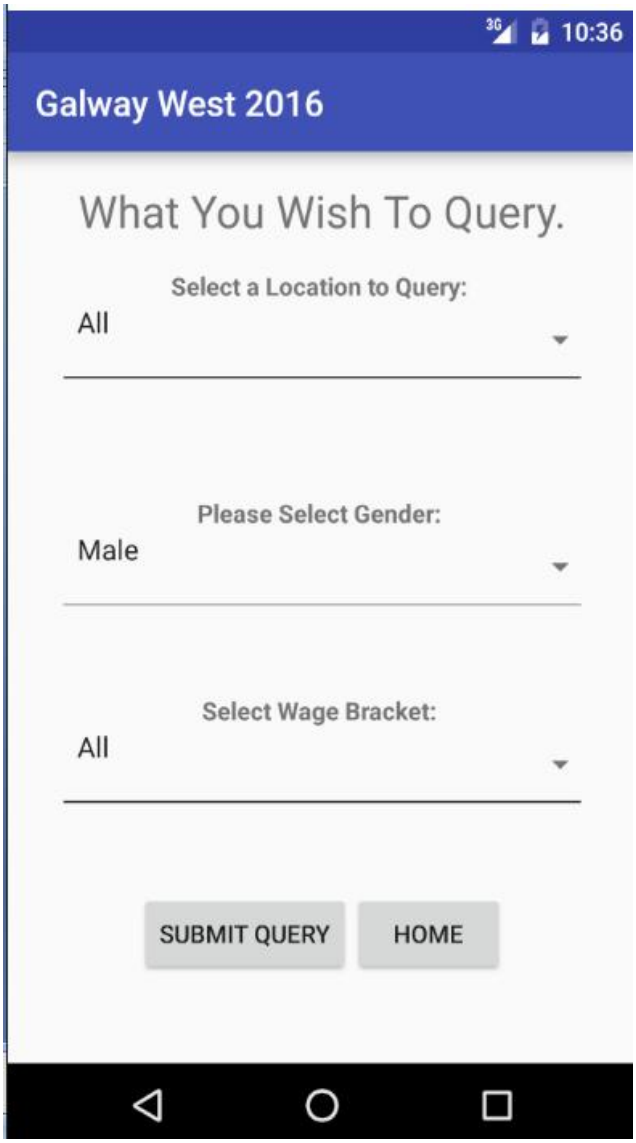

Figure: 6 Query Page

2.7 Queried Information

The queried information is bundled to this page and passed to the database to be queried and the required results returned. The view is also a scroll view so that the information can easily be seen if there was a lot of it. The biggest hurdle on this page was queering the database. For each scenario on the previous page i.e. if you want to return just males that earn a certain amount of money and from a certain location, a separate method had to be written in the java to allow the different parameters to be passed. There may have been an easier way but this is what I found worked for me and this application. The information that is returned for the queries just displays the choice of party leader, party, concern, and candidate. As I felt they would be the most relevant.

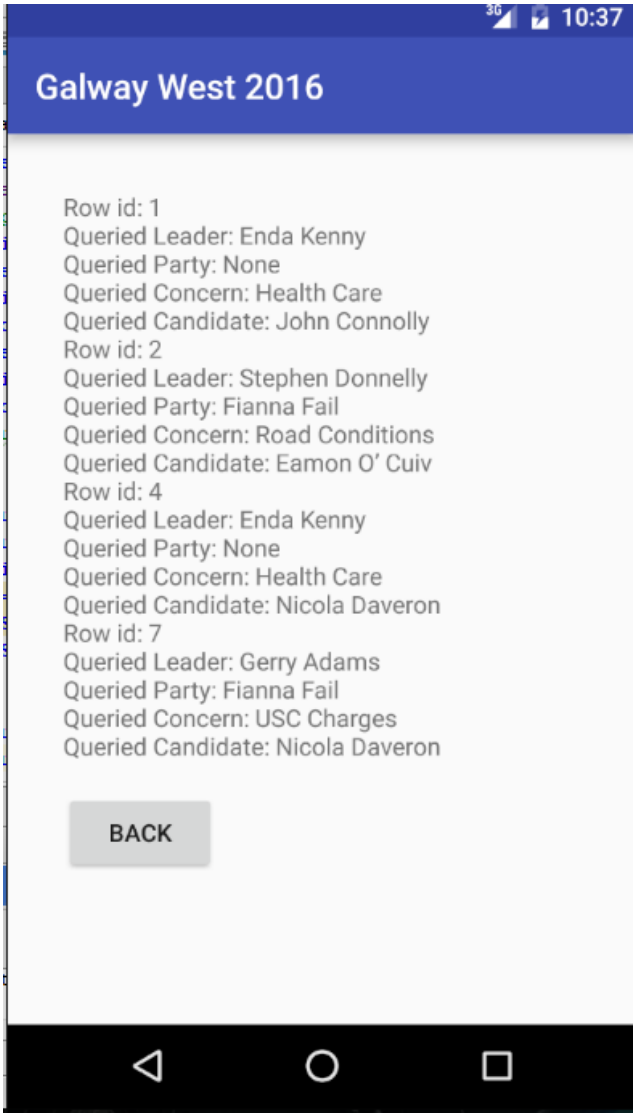

Figure: 7 Queried Data

2.8 Results Pages

To display the results a Custom list view is used to display the results with the appropriate pictures for each party, party leader, and candidate. For the first page, you have two list views this allows the user to scroll either list to view the required results. Each list view is divided using the <View> command from the XML page this eliminates any confusion as to what each section is *figure: 8*. The button at the bottom brings you to another list view that displays the candidate photo, name, and result. The percentage sign is missing from the result as I was unable to find a way of adding it to the index in the array along with the result of that particular item.

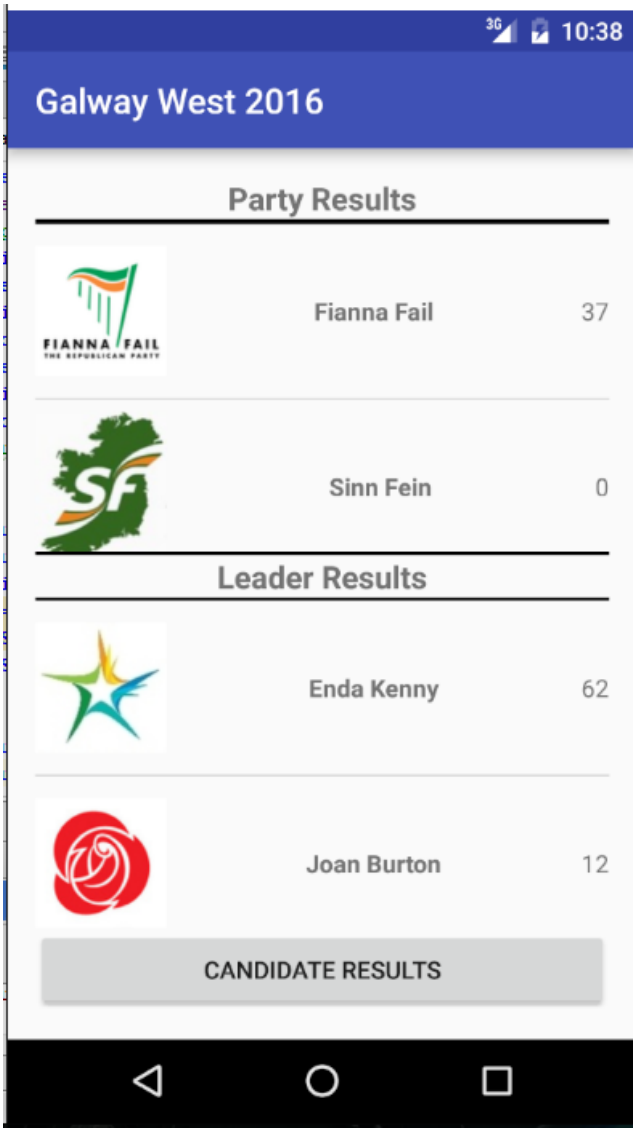

Figure: 8 Result Percentages

2.9 Full Database

This page is done in much the same way as the queries page except you are returning all the information that is available in the database. This is also done to a scrollable view which allows all the information to be viewed with ease regardless of the number of entries. This proved to be a useful way of testing whether the database was working correctly so it was kept in for the final submission. Also, it can be a useful tool if you wish to query the data quickly without passing any parameters.

Figure:9 Full Database

3 Operating the App

3.1 Testing

When testing the app throughout the project I used the built in Android Studio Emulator as I found that it worked best for me. It allowed me to use the debugger whenever I ran into difficulties this helped some of the times and a bit vague some other time. The application was then tested on the phone (HTC M8) to make sure that the screen was visible and that everything was fitting on the screen. The final result of the application works fine in all the required areas, can be a bit slow on the emulator possibly down to the picture on the home screen being a large size.

The application is fairly straight forward to use mainly through the use of buttons which was briefly explained in the design porting. The final page of the poll requires you to select a candidate from the list view and then submit your information to the database by pressing the button.