ASHESI UNIVERSITY COLLEGE

THE DESIGN AND IMPLEMENTATION OF AN ELECTRONIC VOTING SYSTEM ON SMARTPHONES

Undergraduate Applied Project

B.Sc. Computer Science

Frances Efua Antwiwah Antwi-Donkor

2018

ASHESI UNIVERSITY COLLEGE

The Design and Implementation of an Electronic Voting System on Smartphones

Undergraduate Applied Project

Applied Project submitted to the Department of Computer Science, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science

Frances Efua Antwiwah Antwi-Donkor

April 2018

DECLARATION

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

…………………………………………………………………………………………

Candidate's Name:

…………………………………………………………………………………………

Date: ……………………...…………………………………………………………

I hereby declare that preparation and presentation of this Applied Project were supervised in accordance with the guidelines on supervision of Mr. David Sampah laid down by Ashesi University College.

Supervisor's Signature:

…………………………………………………………………………………………

Supervisor's Name:

…………………………………………………………………………………………

Date:

…………………………………………………………………………………………

Acknowledgement

I am thankful to my supervisor and family for the support I received throughout the journey of working on this project. I am also grateful that through this application I have been able to gain skills beyond computer science, that will impact my life.

Abstract

Voting is key in maintaining the structure of society. Voting aids in making a choice in situations where there are multiple options pertaining to a decision being made. The most popular form of voting is the paper-ballot system where voters cast their votes by imprinting their fingerprints on special ballot papers. However, with the rise in technology which seeks to make life better, there have been various attempts to automate the voting process to reduce time spent on casting votes and counting results to determine the winner. This project seeks to build a voting application on smartphones using Ashesi University College as a case study. This application makes use of certain functionalities such as registering and verifying users. The application also makes use of security features that prevent users from voting more than once. This application will cater for general and special elections that go on during the school year where the manual process falls short.

Table of Contents

Contents

Table of Figures

Chapter 1: Introduction

1.0 Background

Ashesi University College is located in the Berekuso Township, Eastern Region of Ghana. This is one of the many private universities in the country and can boast of its unique way of education, which is the Liberal Arts system. The university comprises of many nationalities and people made up of unique ideas which contributes to the ecosystem of the university.

General elections are held every year, where the outgoing office gives way for new executives. The elections comprise of students vying to head the Ashesi Students Council and the Judicial and Electoral Council. These are separate seats of government but work together, to push the interests of students forward. These executives are elected by the student population (Ashesi Student Council, 2016).

Elections in Ashesi are not just limited to the Student Council and Judicial and Electoral executives. In events where the opinion of the student body is required, elections are held. An event that requires an election includes; the Ubora Excellence Awards program. Here, students nominate other students and voting is undertaken for awards pertaining to categories such as Entrepreneurial Spirit, a student with an entrepreneurial attitude and Golden Mouse, a student that is dedicated to impacting other students with technological knowledge (Ashesi Student Council, 2016).

With respect to the evolving use of technology today, everything done or in this situation, applications that are built are at great risk of being manipulated. With this in mind, necessary steps must be taken to cater for the security aspect of any and every application.

With the advancement of technology there are always prying eyes that will want to access data. With the implementation of an electronic voting application, security is its backbone and it must be guarded at all cost. This functionality is the most important feature, as the system has to protect user information. Security protocols can be used to help prevent illegal access to data being collected and stored.

1.1 Proposed System

An electronic voting system also known as Online Voting System (OVS) is a term used to describe the method of voting not via the traditional paper-based system. It allows voters to vote through computers or with respect to this project smartphones and view the results in the web browser (Salkar, Gupta, & Dsouza, 2016). This includes but not limited to voting via punched cards, specialized voting kiosk and optical voting systems.

This system of voting has been around and used in various countries. Countries such as Brazil, Austria and France use this method of voting.

This proposed system seeks to take electronic voting to the next level, where users will now have the ability to vote on their smartphones. This method of voting is appealing as it reduces manual efforts and bulk of information can easily be handled and utilized (Salkar, Gupta, & Dsouza, 2016).

1.2 Problems and Benefits

1.2.1 Problems of Previous System

As of today, elections are done via the traditional way of using ballot papers. This way of voting makes use of the manual way of selecting or choosing a student president and their respective vice president. After this exercise has been undertaken, the ballot papers are then counted and recorded and this may lead to errors and confusion (Ichinoa & Schündelnb, 2012). Despite Schündelnb and Ichinoa's article looking at voting irregularities such as inflating votes on a national level, influence of observers and miscalculation of votes, this can be related to the Ashesi University Community.

An advantage of paper-based voting is its reliability. This is because, the paper-based voting system allows the user to have evidence of the vote that they place and this in essence leaves a verifiable audit trail of their vote caste (Bannet, Price, Singer, & Wallach, 2004). This can be further explained that, the system can never go down as compared to electronic voting. On the other hand, paper-based voting is slow, expensive and in terms of it been printed out, the print-out may be illegible. Another disadvantage may include changes which cannot easily be rectified if the need arises.

1.2.2 Benefit of Proposed System

With the use of an electronic voting system, the electoral committee will not have to look for common free times during class to be able to hold voting. Furthermore, this will help eliminate the long winding queues and increase the number of students that cast their vote as they can vote at their own convenience. Time spent in counting votes will reduce and this means will help eliminate potential voting irregularities and human error.

1.3 Related Work

The implementation of this project relies heavily on security as this is the most sensitive functionality of the application. In Centinkaya's paper he highlights that there is a trade-off between receipt-freeness and individual verification. The paper proposes a solution to help reduce this problem. The scheme introduced is known as PreFote scheme which provides direct individual verifiability without sacrificing receipt-freeness.

The paper goes on further to highlight the security requirements necessary for an efficient implementation of this application. These requirements include voter privacy, eligibility and fairness (Centinkaya, 2008).

Furthermore, the smartphone usage in Ghana was highlighted in a research conducted by Pew Research Center. The paper states that 25% of the adult population used internet occasionally or reported owning a smartphone whilst 21% of adults reported owning a smartphone. However, between 2014 and 2015 there has been an increase in the number of times adults access the internet and make use of certain functionalities such as the camera, fingerprint scanner and GPS (Pew Research Center, 2016).

Bringing this survey to the Ashesi campus, with respect to research conducted it was realized that approximately 90% of the student population owned or used smartphones. This was an important discovery that was necessary for this project as this is the main and most essential tool needed for the completion of the project.

In the research paper *A Taxonomy Refining the Security Requirements for Electronic Voting: Analyzing Helios as a Proof of Concept,* certain security requirements and levels of verifiability were laid out in trying to establish a secure voting application. However, the paper goes highlights that these security requirements change with respect to the level of secrecy one is trying to achieve. Drawing from this paper and juxtaposing it to Ashesi University College voting requirements, the level of secrecy to be achieved includes voters not being linked to their votes and the anonymity of voters. With respect to the level of verifiability the paper explains that this level is closely related to the security requirement of accuracy and that integrity must be established. This integrity comes in two forms; Universal integrity and Individual integrity.

To explain further, Universal integrity deals with the integrity of the collection of votes during the tallying phase, whilst individual integrity dwells more on the individual vote that is being placed (Langer, Schmidt, Buchmann, & Volkamer, 2010).

1.4 Problem Statement

Building a fully functional online system will not only go a long way to rectifying some of the issues Ashesi University College students face during their election period. Some of these problems include trying to vote only during their free periods, joining long queues to cast their vote which inevitably wastes time and also losing their right to vote if they forget their school identification card. Furthermore, this system will not only help reduce some of the problems mentioned above but will go a long way to increasing the percentage of students that turn up to vote and in addition drastically reduce the potential errors made during collating results.

1.5 Project Objectives

This project seeks to build an electronic voting system that will help to reduce the amount of time spent at casting votes and include a majority of the student population during elections. The objectives of the project are:

- Design and implement a system that is secure enough to take up the voting functionality and reduce voting irregularities
- Implement a system that has all information regarding upcoming elections such as candidate manifestoes
- Design and implement a system that will reduce the amount of time spent on voting

- Design and implement a system that produces voting results in the shortest possible time.

1.6 Motivation

In most countries, voting is one of the most important civic duties of many citizens. This responsibility gives the people a voice and helps them elect into power someone who will be their beacon of hope and help push their ideas forward. In developing countries, voting for a president is a major task given to the citizen and this is by no means just any another responsibility. However, this solemn and sacred task has been tarnished by voting irregularities (Ichinoa & Schündelnb, 2012) and thus taints this process. Bringing this issue down to college level, it has been observed that not many students as will be preferred, actively engage in elections due to time constraints. Furthermore, counting of ballot papers go into the night. Having identified these issues, this project seeks to find a solution to make voting in schools seamless as possible whilst engaging the entire student body.

1.7 Overview of Chapters

This paper is made of six chapters. With each chapter a brief introduction of what the chapter talks about is given to help the reader understand before hand what the chapter is trying to convey.

Chapter 1, discusses the background and scope of the project, related works, the problem statement, the project objectives and the motivation for undertaking this project. Chapter 2, gives an understanding into the system requirements of this project. Here, the audience and sample use cases are given to help illustrate the development of the project better. The third chapter gives an exhaustive explanation of the design and architecture of the system being proposed.

A comprehensive analysis of the implementation and design process of the application is outlined in chapter 4. This gives an introduction of the tools that were used in building this application. Chapter 5, highlights the various tests that were conducted and the results generated. In the last chapter, which is chapter 6, the paper gives an overview of the project done, checks if all the functionalities stated were met, the limitations and future works to be conducted.

Chapter 2: System Requirements

This chapter seeks to provide an in-depth understanding of the functionalities the electronic voting system will be implementing. It further describes use cases in which the application will be used with respect to its target users.

2.1 Project Overview

The electronic voting application will be made up of two major parts. The first part of this application is the administration side where the electoral committee will be able to access the data collected during voting. Here, the electoral committee will get real time information on the statistics of the voting process and the final outcome, registered voters and the number of students who have placed their votes.

The second component of this application is the client-side. This aspect is mostly reserved for students who will be casting their votes. Students will be able to register, verify their account and check the statistics of ongoing elections, reading manifestos of vetted candidates.

2.2 Target Users and Characteristics

This application is mainly built to aid students cast their vote easily. The application can be accessed by smartphone users as this type of phone will help in implementing the intended functionalities. The application caters for the needs of both the electoral committee of the school and the student population.

2.2.1 Use Case Scenarios

2.2.1.1 Electoral committee/ Administrator

Currently after every election, the electoral committee sits down to count the votes before publishing. However, the proposed system will automate the publishing of results. Therefore, the committee's responsibility is to verify that each student voted once and also make sure that the system was not tempered with. The committee can then publish the results after all checks have been made.

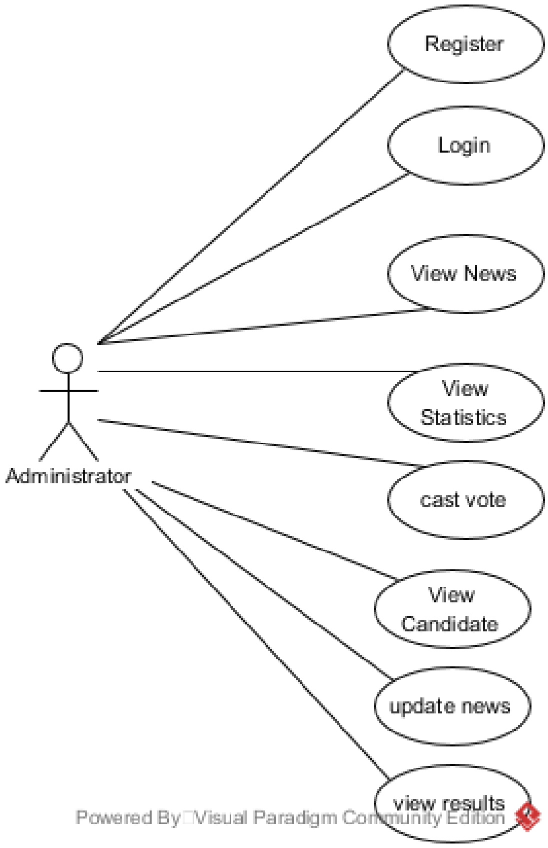


Figure 1: Use case diagram of the Administrator

2.2.1.2 Voters

Ashesi students take part in a yearly Ashesi Student Council (ASC) elections to elect the next ASC executives and Judicial and Electoral Council (JEC) executives. Some students are reluctant to go and vote as they reside off-campus and it is very far from the school. However, with the adoption of the proposed system, students will be able to cast their votes electronically. Students will get real-time update on the ongoing voting and general statistics of information surrounding the voting process. Thus, this student will not have to walk all the way to school and stand in a queue to cast a vote.

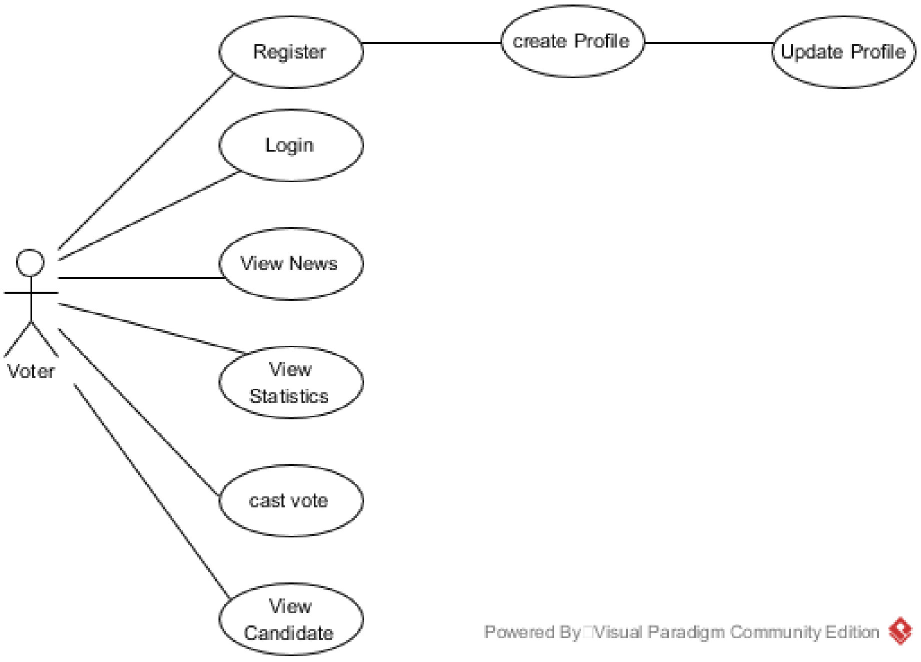


Figure 2: Use case diagram of Voter

2.3 Product Functions

The subsequent list gives a brief description of the core features and functions of the electronic voting system. The core system features provide the comprehensive functionalities of the application and must be implemented as effectively as possible.

2.3.1 Core Features

2.3.1.1 Registration and Login

- Users can register and verify their legibility as a voter
- Users should be able to log into their accounts with no problems.
- Candidates can register for their various positions

2.3.1.2 Data processing and storage

- After user information is entered, the system stores' this data into the database
- The system should be able to display all registered voters to the administrator/ electoral committee

2.3.1.3 Uploading and viewing of manifestos

- Candidates running for election can upload manifestoes
- Candidates can view information pertaining to the other candidates

2.3.1.4 Security

- Users are verified based on their Ashesi email

- Cryptographic and security protocols are put in place to reduce data manipulation and the possibility of users hacking into the system.

2.3.1.5 Committee

- The committee should set up candidates
- Audit manifestos and all related documents before upload

2.4 Non-Functional Requirements

The non-functional requirements of this application are usability, availability and reliability.

2.4.1 Usability

Usability refers to how easily people relate to an application. The application should be easy to understand, to use and should be implemented in a way that will limit user errors which may result from unintentional manipulation of the data. The application should provide information on the basic functionalities and what the application is about. Furthermore, the application should be familiar to the user and must not require a lot of computer expertise to use it

2.4.2 Availability

The user must be able to control the application to the best of their ability. Given the time voting is open and when it closes, the user can choose to vote within that set time without having wait for their free period like the earlier system encourages.

2.4.2 Reliability

The application should be reliable. The data entered by a user must be promptly and correctly stored in the database. In the event of any error while the data was being uploaded an error message or feedback should be given to the user.

Chapter 3: Architecture and Design

Every application has a system architecture which can be defined as its core schematic to building a successful application. This skeletal structure gives a detailed understanding of how the system works, the components that make up the system and how these components come together to generate a fully functional system. The architecture of this system should be able to adequately manage the functional requirements of the system that were highlighted, not forgetting the non-functional requirements (Sommerville, 2011). The system makes use of the Client-Server computing architecture and the Model View Controller Architecture.

3.1 Client-Service Architecture

With respect to this form of architecture, the user of the software interacts with a program which runs on their local computer. This program may be web or phone based and in turn runs on a remote computer, known as the server (Sommerville, 2011).

This application runs on Google Server and the database in use is Firebase. Firebase is a no-SQL database that runs differently from relational database.

For the users view, the device is solely a mobile application that runs strictly on smartphones. This application will run on the Windows, Android and iOS smartphone platform. Information gathered from the users will be stored on Firebase cloud services and not on the mobile device. Therefore, in the case of the users' phone being hacked data cannot be retrieved. This provides some security to the data gathered and protects the choices that the user makes.

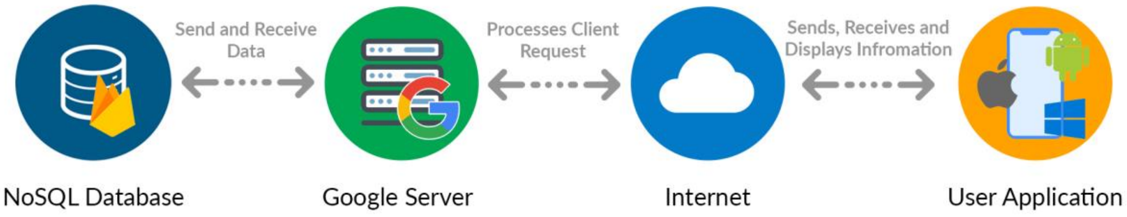


Figure 3: The General view of the system

Figure 3 shows the system architecture of the application. As explained earlier, the system makes use of the noSql firebase Database, this is hosted on Google Servers and relies heavily on the internet. From the diagram, users send information via their smartphones over the internet and this information is stored in Firebase database.

3.2 System in Detail

The system architecture makes use of separation of data. Here, data can be separated into four distinct layers that however interact with each other and come together to form a complete system. The layers include; the presentation layer, the data management layer, the application layer and the database layer (Sommerville, 2011).

3.2.1 Presentation Layer

The presentation layer is concerned with presenting information to the user and managing all user interactions (Sommerville, 2011). This can simply be explained as the view that enables the user to interact with the application. The application being proposed uses the ionic framework and the mobile phone device to help with the user views. Here, it is created in such a way that, users will be able to interact with the system easily regardless of their level of computer literacy, as the application will run on the three major mobile phone platforms (Android, iOS, Windows).

3.2.2 Data Management Layer

This layer manages the data that is passed to and from the client. The layer implements checks on the data and manages and generates pages (Sommerville, 2011). In this application, specific codes or program functions are put in place to make checks and to send data smoothly from the user to the server.

The connection to the Firebase Database via the use of AngularFire2 helps in the smooth running of the application. Also, within this application with every input a user makes, there is a reliability check performed and if any error is generated, feedback is given to the user.

3.2.3 Application Processing Layer

This is concerned with implementing the logic of the application and providing the required functionalities to the end user (Sommerville, 2011). The proposed system seeks to provide the needed functionalities of the system to the user to ensure a complete and usable program. Some functionalities the application provides includes the registration functionality. With respect to this, the user registers with details such as their email address, which is strictly for authentication, their major, gender, and date of graduation. This type of information is gathered for the sole purpose of statistics and analysing the trend of voting in the school.

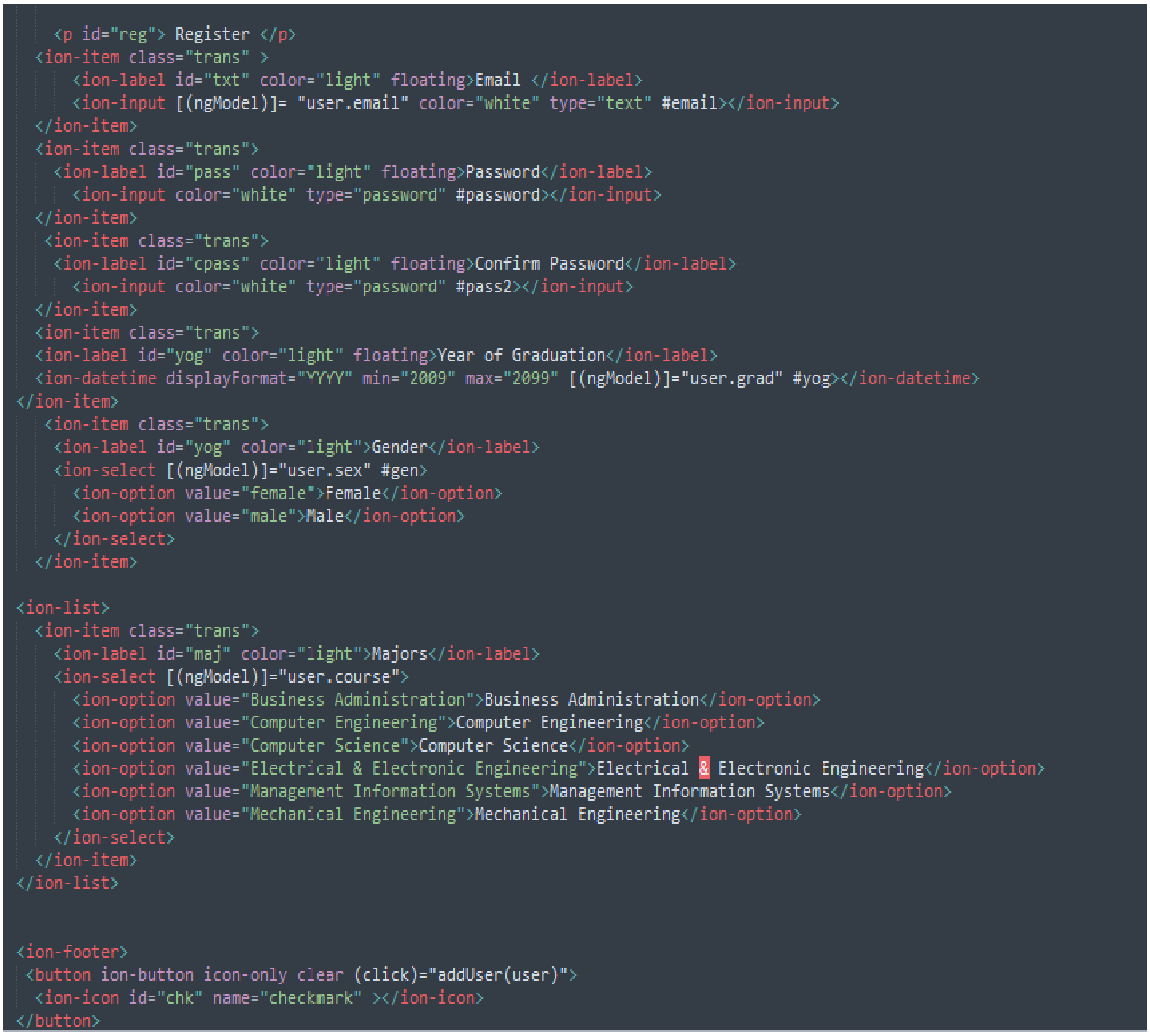
Once a user is registered, they are required to login in again as another step in the security feature. From this point the user now has the liberty to view the types of voting going on such as ASC elections or Ubora elections, they are allowed to read general news pertaining to the elections, view and read the manifestos of candidates and finally cast their votes. Users will be given up to date statistics of the voting process, however, care will be taken not to reveal emerging winners of the elections.

3.2.3.1 Ionic Framework and Programming Languages

For users to be able to fully utilize the functions of this application, they interact with the system using the ionic framework. The Ionic framework is an HTML mobile app development framework which helps in the building of hybrid mobile applications (Drifty). The Ionic Framework helps to easily develop mobile and web applications by providing user interface components. This framework makes use of HTML-like functionalities, Cascading Style Sheets (CSS) and the backend programming languages, Angular JS and TypeScript.

This framework provides functionalities such as storing your project to the cloud and easy deployment. Once a project is created, the relevant JSON files such as *ionic.config.json* file, which contains all ionic configurations, the source file and other files which is needed for the project to run is created. After this has been done, the project is now ready to be programmed. To be able to view what one is working on, the command --*ionic lab*-- is run. This runs the program in mobile interface and --*ionic serve*-- for web-based applications.

The ionic framework in conjunction with the programming languages allows the program being written to be accessible across all the major platforms i.e. Windows, Android and iOS.



```
    <p id="reg"> Register </p>
  <ion-item class="trans" >
      <ion-label id="txt" color="light" floating>Email </ion-label>
      <ion-input [(ngModel)]= "user.email" color="white" type="text" #email></ion-input>
  </ion-item>
  <ion-item class="trans">
    <ion-label id="pass" color="light" floating>Password</ion-label>
      <ion-input color="white" type="password" #password></ion-input>
  </ion-item>
   <ion-item class="trans">
    <ion-label id="cpass" color="light" floating>Confirm Password</ion-label>
      <ion-input color="white" type="password" #pass2></ion-input>
  </ion-item>
  <ion-item class="trans">
  <ion-label id="yog" color="light" floating>Year of Graduation</ion-label>
  <ion-datetime displayFormat="YYYY" min="2009" max="2099" [(ngModel)]="user.grad" #yog></ion-datetime>
</ion-item>
   <ion-item class="trans">
    <ion-label id="yog" color="light">Gender</ion-label>
    <ion-select [(ngModel)]="user.sex" #gen>
      <ion-option value="female">Female</ion-option>
      <ion-option value="male">Male</ion-option>
    </ion-select>
  </ion-item>

<ion-list>
  <ion-item class="trans">
    <ion-label id="maj" color="light">Majors</ion-label>
    <ion-select [(ngModel)]="user.course">
      <ion-option value="Business Administration">Business Administration</ion-option>
      <ion-option value="Computer Engineering">Computer Engineering</ion-option>
      <ion-option value="Computer Science">Computer Science</ion-option>
      <ion-option value="Electrical & Electronic Engineering">Electrical & Electronic Engineering</ion-option>
      <ion-option value="Management Information Systems">Management Information Systems</ion-option>
      <ion-option value="Mechanical Engineering">Mechanical Engineering</ion-option>
    </ion-select>
  </ion-item>
</ion-list>


<ion-footer>
 <button ion-button icon-only clear (click)="addUser(user)">
  <ion-icon id="chk" name="checkmark" ></ion-icon>
</button>
```

Figure 4: The structure of the ionic framework

Figure 4 illustrates the structure of the ionic framework. This is very similar to the HTML structure. Here the programming language used is Angularjs. Components such as *ngModel* can be seen. To explain briefly, this component helps in the retrieval of user inputs and this will be passed unto the controller page and finally stored in the database.

3.2.4 Database Layer

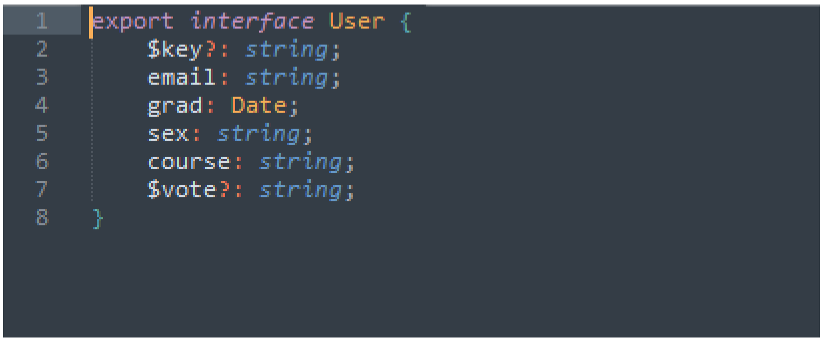
This layer stores the data and helps with easy retrieval for the perusal of the user. This layer is the blueprint of the system and without it, the system will not function as intended. The database being used is a non-relational Firebase database. Firebase is a structure-free database, that does not require the developer to know the number of tables needed or the types of data needed before creating the application. This type of database is beneficial as it is able to store large amounts of data. The database of the system, will store the information pertaining to each user, the statistics of the vote and the profile of candidates.

3.3 Model View Controller Architecture

The Model View Controller Architecture, can be explained as a way of separating or splitting the application into three main parts. These parts are; the Model, the View and the Controller. These parts have separate tasks they perform but come together to make the application work.

3.3.1 The Model

The model of the application can be explained as the part of the application where data is manipulated and stored. This part of the application works hand in hand with the database of the system.



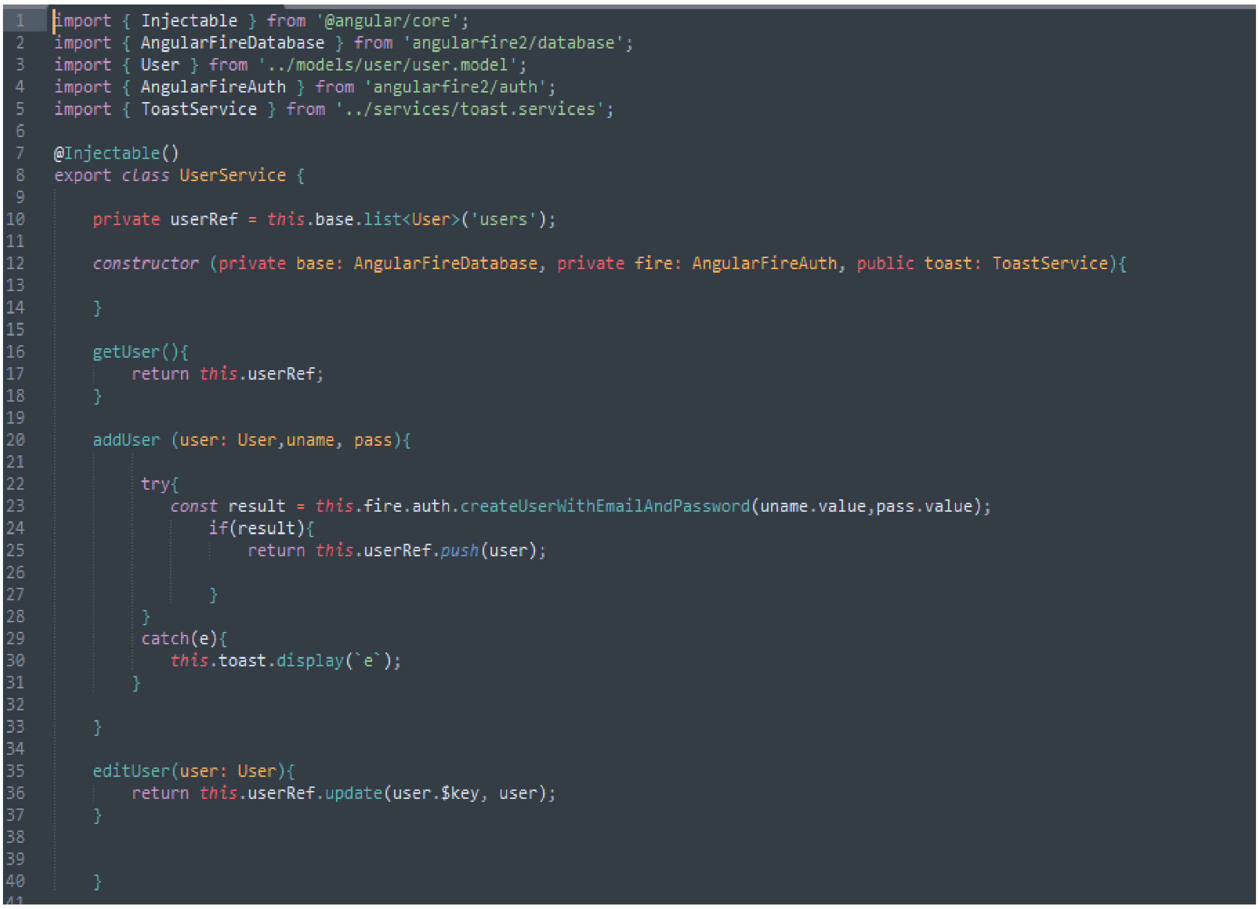
```
export interface User {
    $key?: string;
    email: string;
    grad: Date;
    sex: string;
    course: string;
    $vote?: string;
}
```

Figure 5: The Model of the Registration page

Figure 5 illustrates what the model of the registration page is. Here the parameters that are key to registering are placed here. With respect to the key and the vote parameter, these are optional and are not required by the user that is why they bear the ‘?’ symbol. The data type of each parameter is also given. With every input the user gives with respect to these parameters, it is sent directly to the database.

3.3.2 The Controller

The controller acts as an intermediary between the model and the view. This works in the background and coordinates information that is passed from the user view to the model. In this application, the controllers were labelled services.



```
import { Injectable } from '@angular/core';
import { AngularFireDatabase } from 'angularfire2/database';
import { User } from '../models/user/user.model';
import { AngularFireAuth } from 'angularfire2/auth';
import { ToastService } from '../services/toast.services';

@Injectable()
export class UserService {

    private userRef = this.base.list<User>('users');

    constructor (private base: AngularFireDatabase, private fire: AngularFireAuth, public toast: ToastService){

    }

    getUser(){
        return this.userRef;
    }

    addUser (user: User,uname, pass){

         try{
            const result = this.fire.auth.createUserWithEmailAndPassword(uname.value,pass.value);
                if(result){
                    return this.userRef.push(user);

                }
         }
         catch(e){
            this.toast.display(`e`);
        }

    }

    editUser(user: User){
        return this.userRef.update(user.$key, user);
    }


    }
```

Figure 6: The Controller of the Registration Page

Figure 6 shows the controller page for the registration page. In this page, functions such as *getUser()*, *addUser()* and *editUser()* have been defined. To explain, the *addUser()* functionality takes in 3 parameters, this is an instance of the user, the email and the password. A user profile is created in the Firebase Database using the inbuilt function *createUserWithEmailAndPassword()* and this information is stored in the users folder in Firebase.

3.3.3 View

The view of the application can be explained as the user interface and what the user sees when he or she uses the application. Here, the user can input data easily without having to know what is happening at the backend of the application. With respect to this application, Angularjs and the Ionic Framework help to make the user interface possible and appealing to the user.

Chapter 4: Implementation

This chapter seeks to outline and describe how the electronic voting application was implemented. The application, as said earlier will run on different mobile operating system platforms, to cater for all users. The software of this application will be examined on how data is collected, collated and then published for the users. It goes on further to illustrate how the data is stored in the database.

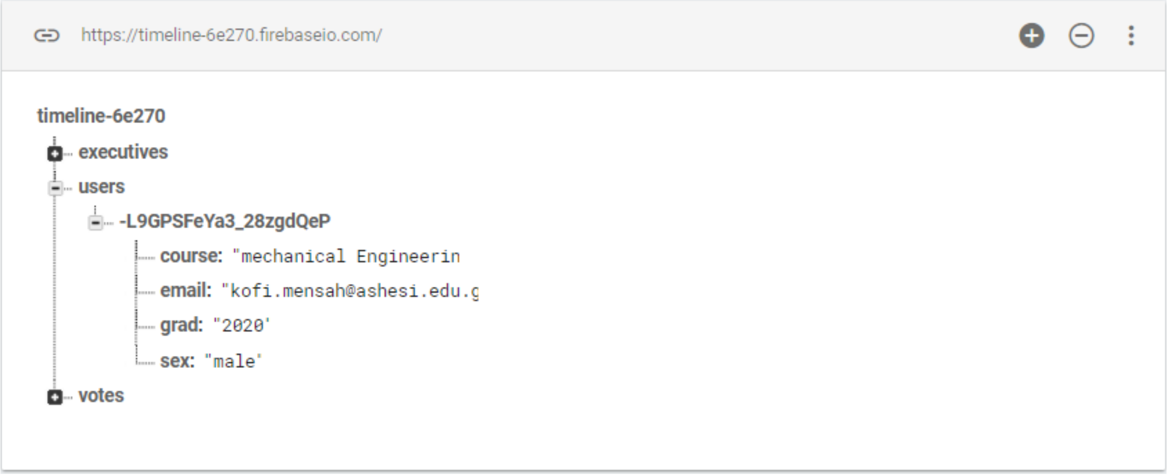
4.1 Tools, Programming Languages and Frameworks Used

The subsequent section gives an in-depth analysis of the tools used in the development of the system. These tools include the Firebase database, Ionic Framework and the programming languages; that is Typescript and Angular JavaScript also known as Angularjs.

4.1.1 The Database

Firebase is a no-SQL database that helps in developing applications. A no-SQL database can be explained as a database structure that does not require a predefined schema before an application is built on it. Here, the developer can easily add, delete, update and ultimately alter the data in the database as they work. This form of structure is flexible and data can easily be manipulated to suit the changing demands of the database.

Firebase is a real time database that allows the JSON, (this is a programming syntax for storing and exchanging data), information being passed to be synced easily with other devices (Google, n.d.). When data is stored in real time, like Firebase allows, this said data can be easily manipulated across all devices connected to the particular database and to the internet.

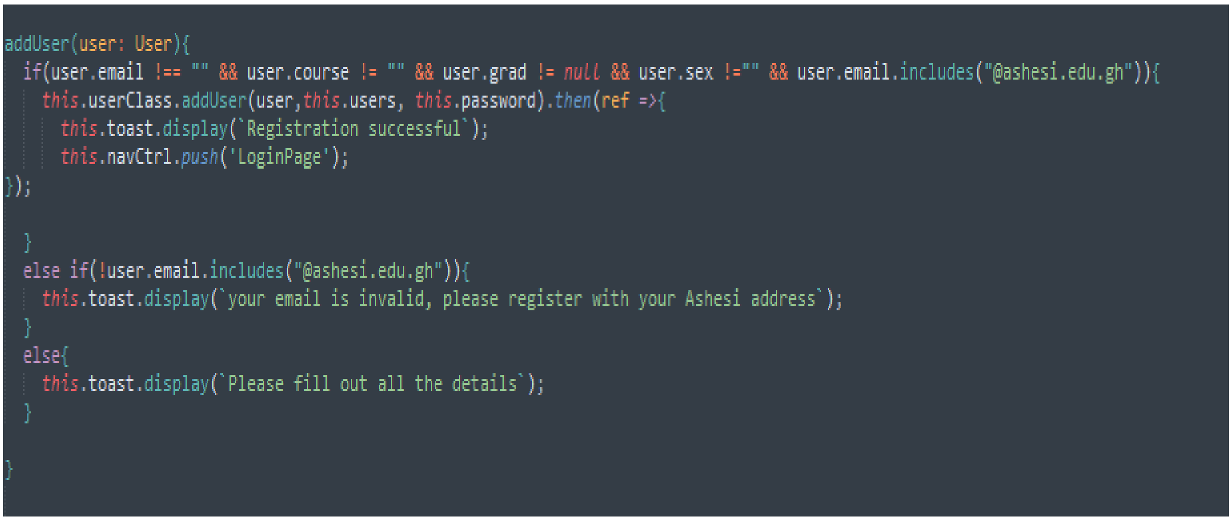


https://timeline-6e270.firebaseio.com/

```
timeline-6e270
  executives
  users
    -L9GPSFeYa3_28zgdQeP
      course: "mechanical Engineerin
      email: "kofi.mensah@ashesi.edu.g
      grad: "2020"
      sex: "male"
  votes
```

Figure 7: Structure of the Firebase Database which illustrates registered students

The figure above shows the structure of the database. This is in JSON format and here the user Kofi Mensah has registered with his course, gender, graduation year and email.



```
addUser(user: User){
  if(user.email !== "" && user.course != "" && user.grad != null && user.sex !="" && user.email.includes("@ashesi.edu.gh")){
    this.userClass.addUser(user,this.users, this.password).then(ref =>{
      this.toast.display(`Registration successful`);
      this.navCtrl.push('LoginPage');
});

  }
  else if(!user.email.includes("@ashesi.edu.gh")){
    this.toast.display(`your email is invalid, please register with your Ashesi address`);
  }
  else{
    this.toast.display(`Please fill out all the details`);
  }

}
```

Figure 8: Code Snippet for Registration and Authentication

Figure 8 shows how the project makes use of the AngularFire2 Authentication, which allows the users to authenticate with their Ashesi University email address.

Furthermore, the database provides cloud storage and database services which helps in the efficient storage of data on the cloud.

4.2 Software Functionalities

4.2.1 Registration and Log in

The system is a mobile application software that gives the user the basic functionalities of registering and also logging in to be able to access other functionalities of the application. The tool that enables the user to be authenticated is the “AngularFire2” Authentication code. Here, the user is signed in using their email address and their password. After the user is authenticated, the email address is cross referenced with the Ashesi University database of current students. This is to help distinguish between alums and current students to prevent alums from voting.

The data collected during registration are simple and short. This information gathered is relevant to analysis that will be made later on, such as when one wants to analyse the number of students who signed up to vote, the ratio of males to females who participated in the elections and the turn up rates for each class.

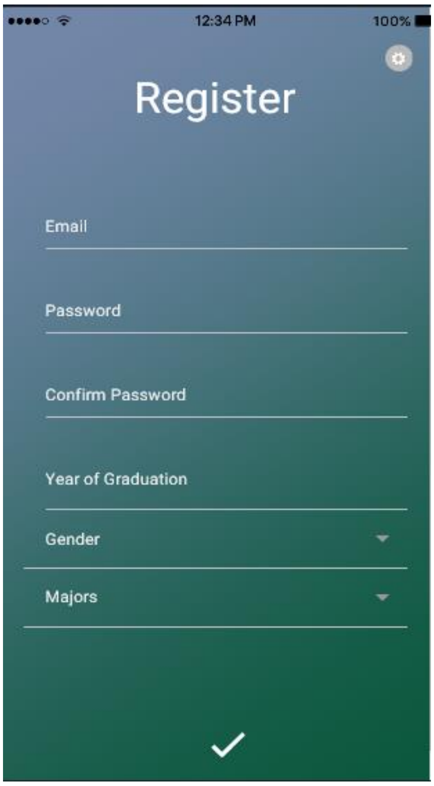


Figure 9: Registration Interface

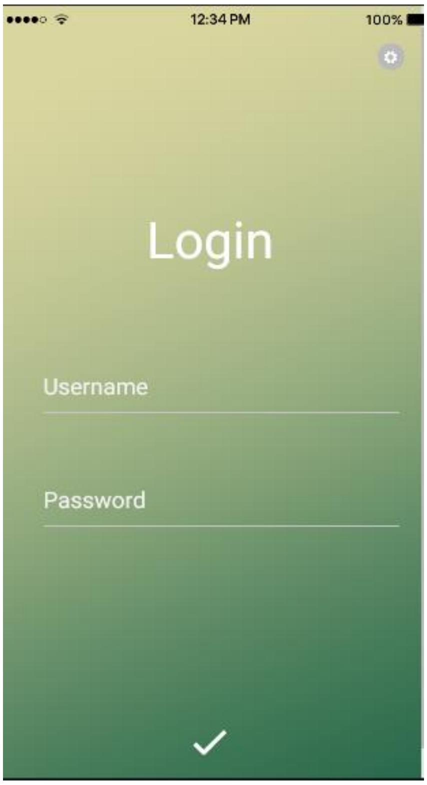
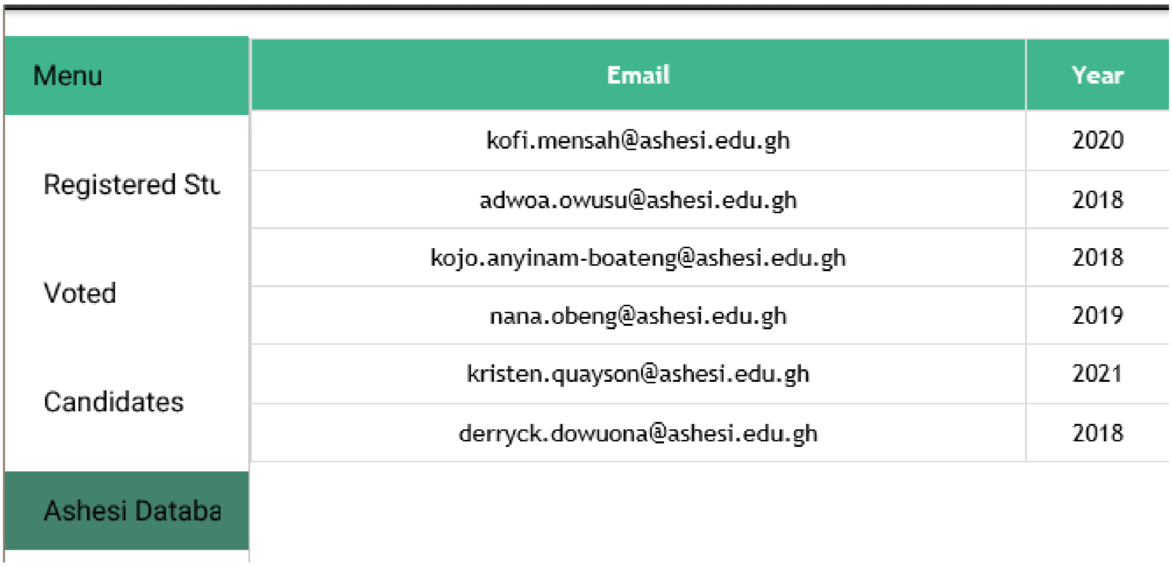


Figure 10: Login Interface



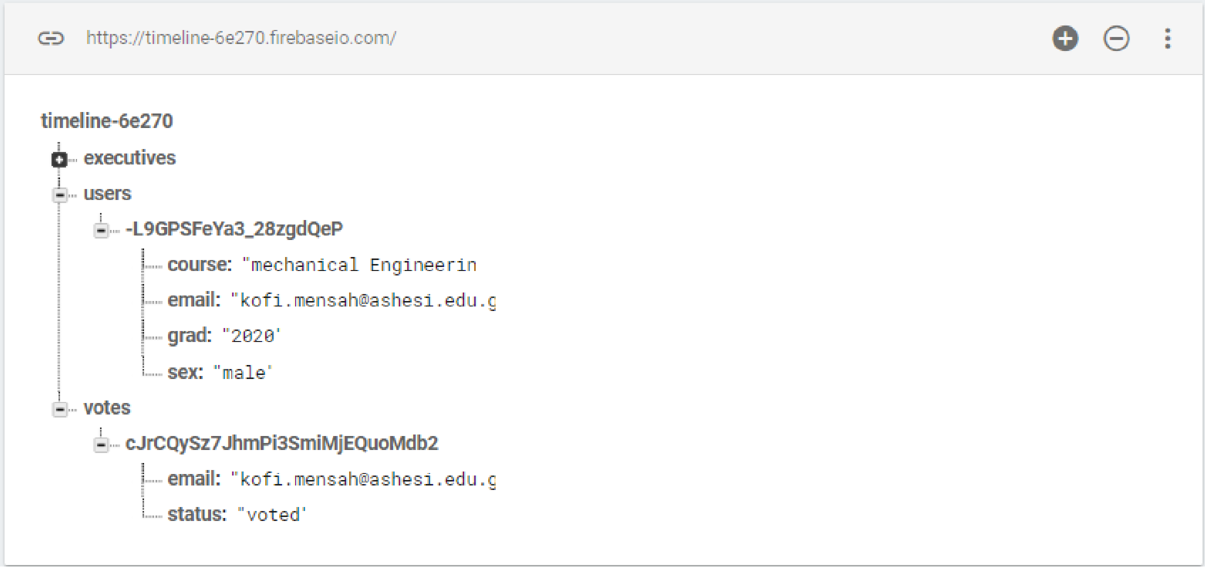
Email	Year
kofi.mensah@ashesi.edu.gh	2020
adwoa.owusu@ashesi.edu.gh	2018
kojo.anyinam-boateng@ashesi.edu.gh	2018
nana.obeng@ashesi.edu.gh	2019
kristen.quayson@ashesi.edu.gh	2021
derryck.dowuona@ashesi.edu.gh	2018

Figure 11: A dummy database of current Ashesi University College Students

4.2.2 Voting and Statistics

The main aim of this application is for users to cast their votes. Here, users are able to view the ongoing elections taking place. They can read the manifestos of each of the candidates and finally place their vote.

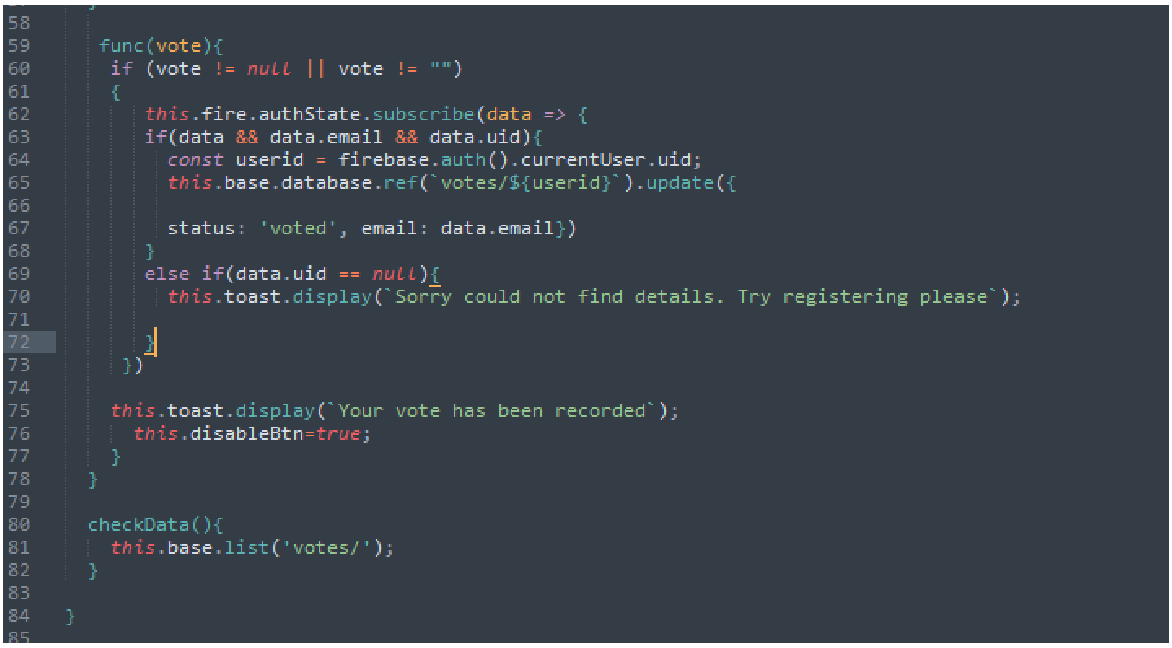
To prevent double voting, the session id of the current user is generated once they log into the application. If the current id matches the user id and a vote is placed, the users' email address along with their vote, masked as 'voted' is stored in the database. Furthermore, the button on the page is disabled and this prevents the user from voting again.



```
https://timeline-6e270.firebaseio.com/

timeline-6e270
  executives
  users
    -L9GPSFeYa3_28zgdQeP
        course: "mechanical Engineerin
        email: "kofi.mensah@ashesi.edu.g
        grad: "2020"
        sex: "male"
  votes
    cJrCQySz7JhmPi3SmiMjEQuoMdb2
        email: "kofi.mensah@ashesi.edu.g
        status: "voted"
```

Figure 12: Illustration showing user 'Kofi Mensah' has placed his vote



```
    func(vote){
     if (vote != null || vote != "")
     {
        this.fire.authState.subscribe(data => {
        if(data && data.email && data.uid){
          const userid = firebase.auth().currentUser.uid;
          this.base.database.ref(`votes/${userid}`).update({

          status: 'voted', email: data.email})
        }
        else if(data.uid == null){
          this.toast.display(`Sorry could not find details. Try registering please`);

        }
      })

     this.toast.display(`Your vote has been recorded`);
       this.disableBtn=true;
     }
   }

   checkData(){
     this.base.list('votes/');
   }

 }
```

Figure 13: Code snippet showing user authentication and verification of vote

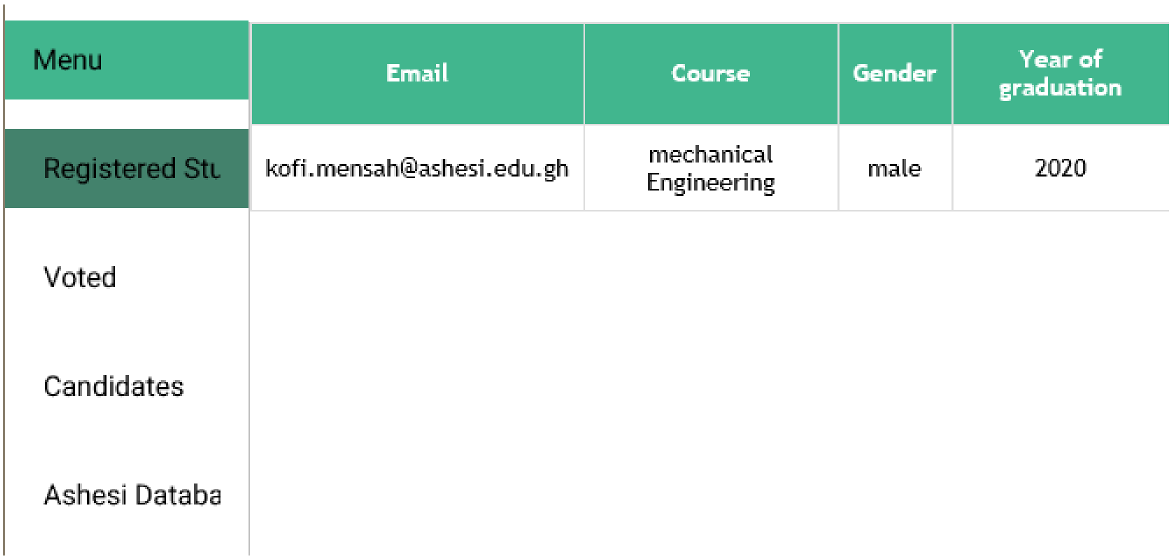
The above figure shows the firebase authenticating the user with respect to their id. Once the vote button is clicked the function automatically checks if the current id and the user id is similar if it is it updates the user profile and stores the status of the users vote as ‘voted’ it then disables the vote button preventing the user from voting again. In the event of an error, the application alerts the user that the id was not found.

The statistics page of the application, shows the charts of the ongoing vote. This page does not show the actual data recorded as this may influence voters indirectly. However, it shows statistics such as the ratio of men to women who have voted, the classes with the highest recorded votes. The reason for this functionality is to help make future decisions for both the voting application and for elections that will take place.

4.2.3 Administrator

The administrator side of the application manages the application. At this side of the application, the administrator has the privilege of executing certain functionalities on the platform. The administrator vets the candidates and when a candidate has passed the vetting stage, the administrator allows the vetted candidate access to upload their manifestoes and other documents relevant to the elections.

Furthermore, the administrator has the privilege of viewing the number of students that have registered with the application, the number of students who have placed their votes and finally candidates of various elections where, as said earlier, the administrator can grant permissions to.



Email	Course	Gender	Year of graduation
kofi.mensah@ashesi.edu.gh	mechanical Engineering	male	2020

Figure 14: A snapshot of the administrators' dashboard showing various functionalities

Chapter 5: Testing and Results

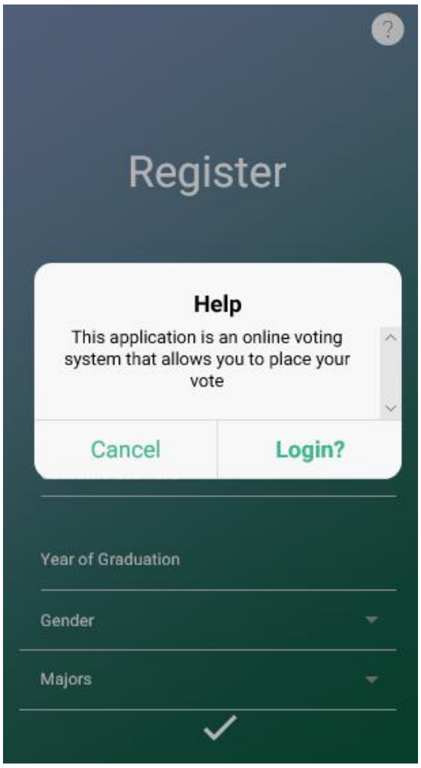
Before an application can be deployed, the developer must make sure that the application runs efficiently and executes all the functionalities and non-functional requirements it set out to explore. Testing helps to identify errors in the code and correct these errors. In this paper two types of error testing were conducted.

5.1 User Testing

The application was given to two different users to test the usability of the application. The application was deployed unto an iPhone. The users were given ample time to be able to navigate and use the application.

This test generated useful insights with respect to the functionality and interface of the application. Some users complained that the background of the application is too colourful and also some of the icons and buttons position was not intuitive enough making the application a bit cumbersome to use.

Upon analysis, it was realized that the login button was wrongly placed under the help button. Thus, if a user did not need help the user would find it difficult to access the login button and page from the registration page. The solution to this problem will be to create a separate icon button for the login instead of merging it with the help button.



This application is an online voting system that allows you to place your vote

Figure 15: The interface of the application

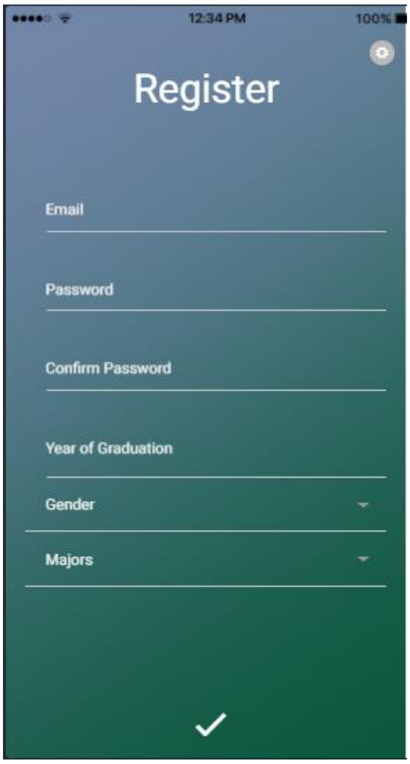


Figure 16: The new interface of the application

Figure 15 illustrates how the first registration user interface looked. After the user testing a settings button was introduced as this was more intuitive in prompting the user that there are more options. This new interface is illustrated in Figure 16.

5.2 System Testing

A system testing of a software can be described as how the various components of the system integrates with each other. Furthermore, this testing is to check if the system is in accordance with the specified requirements.

The requirements that were outlined to aid in this testing was to check if data was entering the database as it should; if users were able to cast their votes without problems and also to check if the administrator could view candidates and other statistics relevant to the application.

5.2.1 Database Test

With respect to the database requirements, data entered the database seamlessly, however when the user id had to be called on the log in page it generated an error. This problem was rectified by removing the function that checked if the user had an id as it was realized that at the time of login there will be no generated user id.

5.2.2 Casting of Votes Test

Users were able to cast their votes without any problems. Also, the button was disabled once this was done preventing users from double voting.

5.2.3 Administrator Test

With the use of the real-time database, the administrator was able to view data that was coming in in-real time.

Chapter 6: Conclusion and Recommendations

This chapter summaries and gives an overview of the project. The functional requirements and non-functional requirements of the project that was met.

6.1 Summary

The core aim of the project was to provide a secure and easy to use electronic voting application for students of Ashesi University. The first chapter spoke extensively on security, however, this was not implemented to its optimum, as a lot of the time was spent on other functionalities.

6.2 Functionalities

In chapter 2, core and non-functional requirements where listed out as the core module of this application. This section of chapter 6, highlights if these requirements were met.

6.2.1 Registration and Login

By using the Angular Firebase authentication, users were able to successfully login into the application. Furthermore, an added security functionality of verifying with the current Ashesi University College Database helped to provide additional authentication. Each data collected was stored separately in JSON format in the Firebase Database and stored under separate unique folders for easily identification by the administrator.

6.2.2 Uploading and Viewing of Manifestos

Prospective candidates were able to upload their manifestos and other relevant data unto the database and this was made readily available for students to read and help them in making a decision before casting their votes.

6.2.3 Security

As earlier stated, new students were able to register based on their Ashesi University College email address. However, the goal of employing cryptographic protocols to reduce data manipulation could not be met due to the given time frame in which the project was completed.

6.2.4 Usability

With respect to the user test done, the application was voted user friendly, easy to use and was intuitive. The application can be used by any kind of user regardless of their computer savviness.

6.3 Limitations and Challenges

One major challenge faced with the implementation of this application was the use of new tools. The integration of Ionic 3 and Firebase was new and thus, more time was spent reading documentation and watching videos to get abreast with the tools and frameworks.

Despite the successful deployment of the application, the Wi-Fi access is a limitation. This is mainly because, the application uses the Firebase Database and this database relies heavily on the internet. Here, despite some users registering successfully, it took a while before the data entered the database.

6.4 Future Works

The data stored in the database will be encrypted to prevent hackers from altering what is in the database. Furthermore, the application can be upgraded to handle large number of users. when the security features of this application is done properly and effectively the idea can be sold to benefit the elections of district assembly personnel and minor elections that may take place in other institutions.

6. Conclusion

This application marks the beginning of electronic voting in schools. Despite the application not being ready to be deployed on the market it has the potential of revolutionizing the voting process in Ashesi University to help cut down on the hours spent waiting in long queues to cast votes.

References

Ashesi Student Council. (2016). Constitution.

Bannet, J., Price, D. W., Singer, J. S., & Wallach, D. (2004). *Hack-a-Vote: Security Issues with Electronic Voting Systems.* IEEE Computer Society.

Centinkaya, O. (2008). Analysis of Security Requirements for Cryptographic Voting Protocols. *2008 Third International Conference on Availability, Reliability and Security,* (pp. 1451-1456).

Drifty. (n.d.). Welcome to Ionic- Ionic Framework. Retrieved from https://ionicframework.com/docs/v1/guide/preface.html

Google Developers. (n.d.). *Firebase Realtime Database*. Retrieved from Firebase: https://firebase.google.com/products/realtime-database/

Ichinoa, N., & Schündelnb, M. (2012). Deterring or Displacing Electoral Irregularities? Spillover Effects of Observers in a Randomized Field Experiment in Ghana. *The Journal of Politics*, *74*, 292-307.

Langer, L., Schmidt, A., Buchmann, J., & Volkamer, M. (2010). A Taxonomy Refining the Security Requirements for Electronic Voting:. *2010 International Conference on Availability, Reliability and Security*, 475-480. doi: 10.1109/ARES.2010.106

Pew Research Center. (2016). *Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies .*

Salkar, A., Gupta, V., & Dsouza, L. (2016). Online Voting System. *International Journal of Scientific and Technical Advancements*, *2*(2), 39-41.

Sommerville, I. (2011). *Software Engineering.* Boston: Pearson.