

# **Software Requirements Specification**

**for**

## **Android Voting System**

**Prepared by**

**Arsh Ansari [20BCE0371]**

**Madhu Shree k [20BKT0083]**

**Aditya S Rawat [20BCT0349]**

**Jaganath T P [20BCE0840]**

**Shubhangi Tiwari [20BCE2688]**

**15.11.2022**

# Table of Contents

<b>Table of Contents .....</b>	<b>2</b>
<b>1. Introduction.....</b>	<b>4</b>
1.1 Purpose.....	4
1.2 Document Conventions .....	4
1.3 Intended Audience and Reading Suggestions .....	4
1.4 Product Scope .....	4
<b>2. Overall Description.....</b>	<b>5</b>
2.1 Product Perspective .....	5
2.2 Product Functions .....	5
2.3 User Classes and Characteristics .....	5
2.4 Operating Environment .....	6
2.5 Design and Implementation Constraints .....	6
2.6 User Documentation .....	7
2.7 Assumptions and Dependencies .....	10
<b>3. External Interface Requirements.....</b>	<b>11</b>
3.1 User Interfaces .....	11
3.2 Hardware Interfaces.....	11
3.3 Software Interfaces .....	11
3.4 Communications Interfaces .....	11
<b>4. System Features .....</b>	<b>11</b>
4.1 Introduction.....	11
4.2 Description and priority .....	11
4.3 Functional Requirements .....	11
4.4 Details of Software .....	13
4.5 System Attributes.....	13

---

<b>5. Other Nonfunctional Requirements.....</b>	<b>15</b>
5.1 Performance Requirements.....	15
5.2 Safety Requirements.....	15
5.3 Security Requirements.....	15

# Introduction:

## 1.1 Purpose:

- The purpose of this document is to describe the design and development of the software platform for voter registration, election voting, real-time election results collation and monitoring and mostly for voters' remote access to election such as elections at different levels (faculty, departments, school wide e.t.c) in the university.

## 1.2 Document Conventions:

- When writing this document, it was inherited that all requirements have the same priority.
- First an overall view of the system is presented and then all features and functions are analyzed in detail.

## 1.3 Intended audience and reading suggestions:

### 1.3.1 Development

- In order to be sure that our team is developing the right project that fulfils requirements provided in this document.
- In order to have an exact list of the features and functions that are required to respond according to requirements and provide diagrams.

### 1.3.2 Users

- In order to know exactly what they have to expect from the system, right inputs and outputs and response in error situations.

## 1.4 Product scope:

- One of the most important and hence core activity of voting system is to give the administration of school. Administration can save their time. Administration can get all information about voters. Hence a wide range of data is required to justify the content of application which should also be updated at regular intervals. Only authorized person(admin) can access the information. System offer multiple interfaces to enter different type of data.

## 2 Overall Description:

### 2.1 Product Perspective

This is a new, self-contained product. The product, an E-voting app aims to allow voters to vote for an election online with ease along with total votes calculation. The app can be used for any type of election and is not limited or linked to any specific larger system. Our objective is to Design and develop a software platform for voter registration, election voting, real-time election results collation and monitoring and mostly for voters' remote access to election. It will provide a means conduct a less stressful and fair elections at different levels (faculty, departments, school wide etc.) in the university.

### 2.2 Product Functions

- The app allows user to register onto the app successfully and assigns a unique id.
- The user can see the candidates standing for election on the app and vote for their desired candidate.
- User is allowed to vote only once. If they attempt to vote again, they will be brought back to results page.
- After voting is complete, the total votes are calculated
- Candidates and voters can both check the results page for the election.
- Admin monitors voting activities and removes wrong information.

### 2.3 User Classes and Characteristics

#### User Classes:

- **Voters:**  
This user class will utilize the app for voting in elections. They will login to the app, register and vote from their unique voter id. They will be able to see all the candidates up for election and be provided with information they need for a fair election. They can only vote once. Using this product, the voters can vote for all types of elections they are eligible present on the app. This is the user class that the product is majorly made for, it is made to provide ease the of voting process to the voters.
- **Candidates:**  
The candidate's user class will majorly use the app for checking the results of the election. As candidates they cannot vote but will want to check the election results. The needs of this particular user class may not be the most important to satisfy. They are simply using the app to check election results.
- **Application Admin:**  
The admin user class will use the system to create an election type, calculate total votes for a election and monitor voting activities to remove any wrong information. They will maintain the candidate and voter information database being used in the

app. This particular user class is from the developer side and is studying the needs of the other user classes.

- **Election departments:**

This is the user class that will be majorly serviced by our app. This user class includes any organization that wishes to use the app for conducting an election of any kind. It is an important user class and must be satisfied. This user class uses the services provided by the application to conduct elections at any level they require.

## 2.4 Operating Environment

**The software will operate only on Android smartphones hardware with:**

- 64-bit CPU with at least 4x Arm cores (6x or more Arm cores recommended)
- Maximum core speed 1.5 GHz or higher
- 1.5 GB of RAM or higher
- Internet access.

**Operating systems requirements:**

- Android Version 6.0 or higher

## 2.5 Design and Implementation Constraint

**Development workstation should meet or exceed these hardware requirements and constraints:**

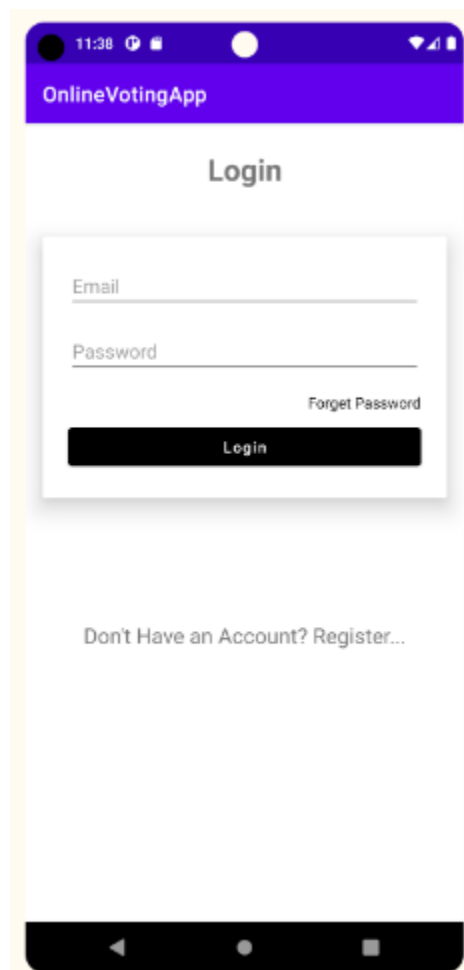
The following are the system requirements for Android Studio on Windows.

- 64-bit Microsoft® Windows® 8/10/11
- x86\_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor
- 8 GB RAM or more
- 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
- 1280 x 800 minimum screen resolution
- A 64-bit development environment is required for Android 6.0, including the master branch. You can compile older versions on 32-bit systems.
- Need to use Android 6.0.

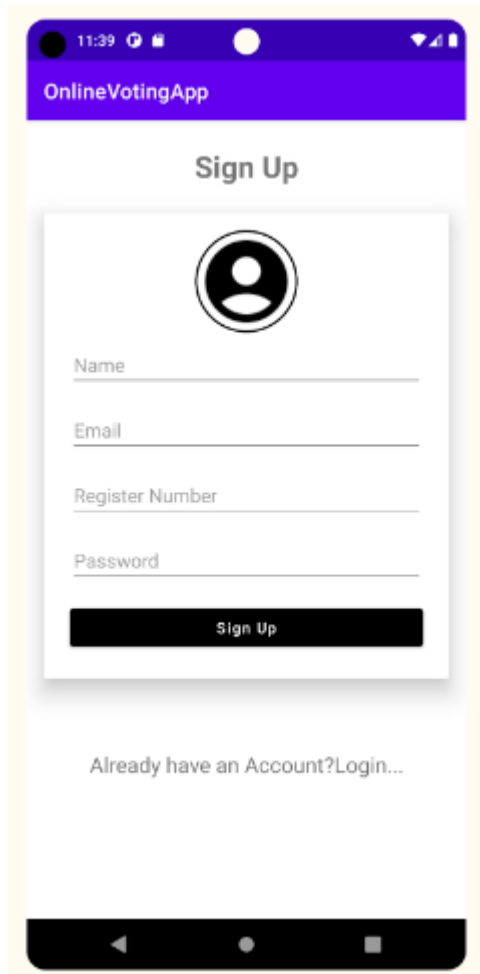
## 2.6 User Documentation

- Our app is relatively very simple to use hence the documentation for user is as simple as follow the buttons and ability to read English as it was built for easy accessibility.
- The admin support will handle all queries related to vote registering through admin channels which will be built ones its launched.

### 2.6.1 Users Guide to Online Voting System:

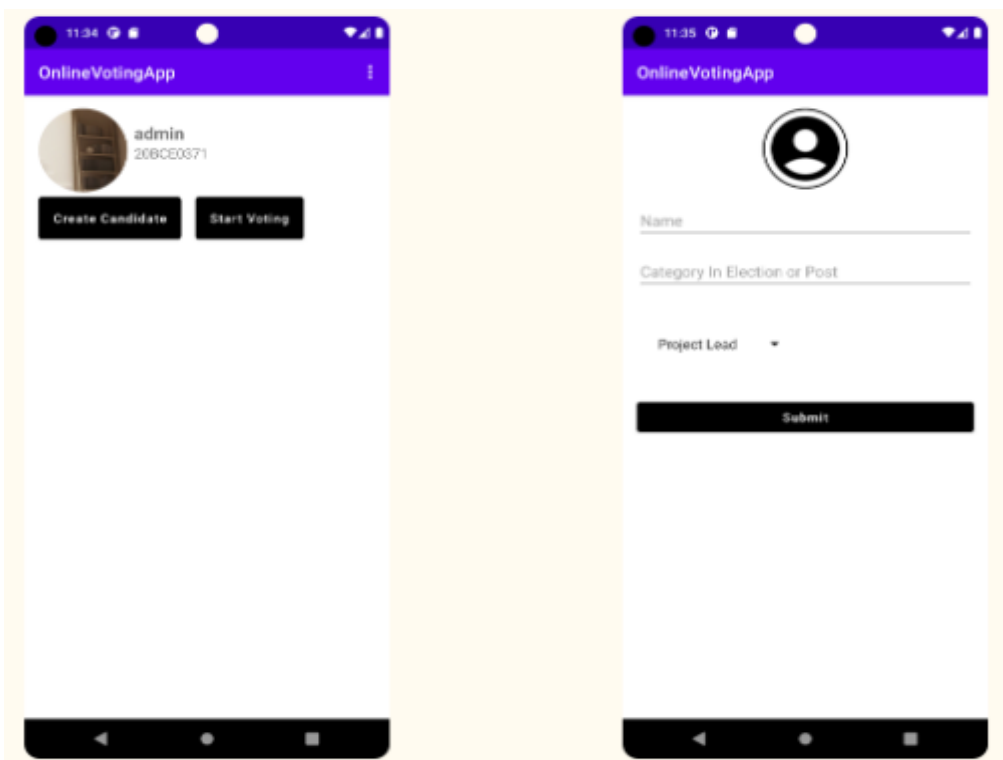


**Login Page** were the user logs in using their respective credentials.

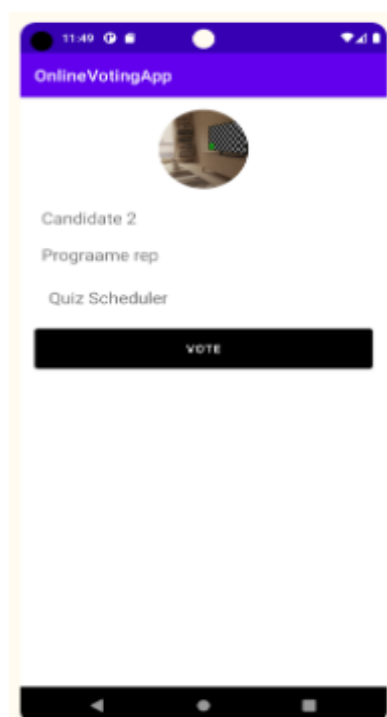
The image shows a mobile application interface for signing up. At the top, there is a purple header bar with the text "OnlineVotingApp". Below the header, the title "Sign Up" is centered. The main content area is a white card with a black border. Inside the card, there is a circular profile icon placeholder. Below the icon, there are four text input fields labeled "Name", "Email", "Register Number", and "Password". At the bottom of the card is a black button with the text "Sign Up". Below the card, there is a link that says "Already have an Account?Login...". The entire interface is set against a light yellow background.**Signup page for Registering users**

- User can add text by clicking on text fields and can add pictures clicking on profile icon.
- Do verify your email using the verification mail sent to the email id being registered for.

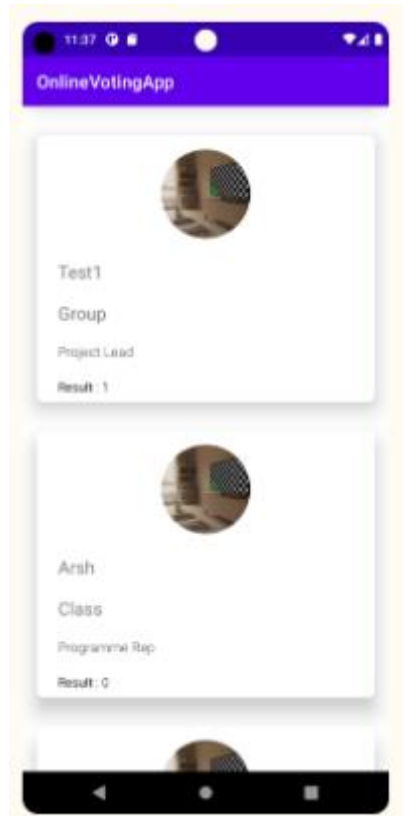




**Admin Registers the candidate and starts the election for the respective posts.**



**Voting page where various elections show up and voters vote for specific elections.**



**Results page where the results of votes for tallied and posted Realtime.**

## **2.7 Assumptions and Dependencies**

Assumptions are as follows:

- User has a compatible device.
- User has uninterrupted internet while voting.
- User knows that vote cannot be forged and trying is useless time-waste.

## **3. External Interface Requirements**

### **3.1 User Interface**

The system must provide a user interface for all types of users (ECA, ESS, EC, and Voter) that is available through android portal.

### **3.2 Hardware Interface**

There are no hardware interfaces to this software system. The only interfaces are through a android smartphone.

### **3.3 Software Interface**

The “Electronic Voting System” software will interface with a google Firebase database for writing and verifying vote-records.

### **3.4 Communication Interface**

Using Google Firebase, a service that offers such a real-time database server in addition to a variety of other features, relatively create communication-based application.

## **4. SYSTEM FEATURES**

### **4.1 INTRODUCTION**

A software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate. This section demonstrates Online voting system’s most prominent features and explain how they can be used and the results they will give back to user.

### **4.2 Description and priority**

The Online voting system (OVS) also known as E-Voting is a term encompassing several different types of voting embracing both electronic means of counting votes. The highest priority of online voting system is creating user friendly voting Ballot where user can vote their candidate.

### **4.3 Functional Requirements**

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behaviour, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describing all the cases where the system uses the functional requirements are captured in use cases.

The various methods used in this project are as follows: -

- Emulator: To perform and display the functionality of the project.
- Android studio: To create, design, test, debug and run the android project.
- Mouse: To navigate through the emulator.

1. **Mobility:** The voter should not be restricted to cast his ballot at a single poll-site at his home precinct.
  - Realistic: He shall be able to vote from any poll-site within the nation.
  - Unrealistic/Expensive: He shall be able to vote from any county-controlled kiosk (situated at public places such as banks, shopping malls, etc.) within the nation. (Unrealistic because of logistical and cost issues).
  - Infeasible: He shall be able to vote from virtually anywhere using an Internet connection. (Infeasible both for technical security issues as well as social science issues).
2. **Convenience:** The system shall allow the voters to cast their votes quickly, in one session, and should not require many special skills or intimidate the voter (to ensure Equality of Access to Voters).
3. **User-Interface:** The system shall provide an easy-to-use user-interface. Also, it shall not disadvantage any candidate while displaying the choices (e.g., by requiring the user to scroll down to see the last few choices).
4. **Transparency:** Voters should be able to possess a general knowledge and understanding of the voting process.
5. **Flexibility:** The system shall be flexible in that it allows a variety of ballot question formats including open-ended questions (e.g. Write-in candidates and survey questions).
6. **Accuracy:** The system shall record and count all the votes and shall do so correctly.
7. **Eligibility:** Only authorized voters, who are registered, should be able to vote.
8. **Uniqueness:** No voter should be able to vote more than once.
9. **Auditability:** It should be possible to verify that all votes have been correctly accounted for in the final election tally, and there should be reliable and demonstrably authentic election records, in terms of physical, permanent audit trail (which should not reveal the user's identity in any manner).
10. **Voter Confirmation:** The voter shall be able to confirm clearly how his vote is being cast.
11. **Provisional Ballots:** The voter shall be able to vote with a provisional (electronic) ballot if he has some registration problems, which could be counted if verified by the authorities later.
12. **Documentation and Assurance:** The design, implementation, and testing procedures must be well documented so that the voter-confidence in the election process is ensured.
13. **Cost-effectiveness:** Election systems should be affordable and efficient.
14. **No Over-voting:** The voter shall be prevented from choosing more than one candidate / answer.
15. **Under-voting:** The voter may receive a warning of not voting, but the system must not prevent under voting.
16. **Cloud Firestore:** The voting system uses firestore database which is good for getting started with the mobile and web client libraries, but allows anyone to read and overwrite

your data in test mode. After testing, make sure to review the secure the voting data section.

17. **Authentication:** The voting system uses Email and password Authentication. This will authenticate users with their email addresses and passwords. The Firebase Authentication SDK provides methods to create and manage users that use their email addresses and passwords to sign in. Firebase Authentication also handles sending password reset emails.

## 4.4 Details of Software:

Here, the coding, creating, designing, testing, debugging and running of our project is done in Android Studio.

### 4.4.1 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android Operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription based since in 2022. It is a replacement for the Eclipse Android Development Tools (EADT) as the primary IDE for native Android application development. Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features. While IntelliJ states that Android Studio supports all released Java versions, and Java 12. Some new language features up to Java 12 are usable in Android.

## 4.5 System Attributes

### 4.5.1 Communications Security

All communications between the Voting Client and the Elections Server must be encrypted to ensure the privacy of votes and voter information. Encryption of communications will also ensure that anyone packet sniffing over the network will be unable to extract any usable information from the data that is sent between the Client and Server. The server will time out after three minutes if no message is received from the Client.

Voters must be sure that nobody else has access to their e-mail addresses, as anyone reading their e-mail would have access to their voter identification information. The Elections Server will limit the number of logins attempts to prevent automated attacks to gain or prevent access, however the voters must ensure that their password is secure as the system would not be able to detect misrepresentation of the voter. This system ensures user to validate their email id for their login Authentication which helps to prevent further tampering of data in login.

No encryption is fool proof, although many are highly reliable. It must be understood that developments of new algorithms could potentially break any encryption, as could sustained efforts with high-powered computers. Elections should thus be run over limited periods of time, such as a single day, in order to minimize the chances of security breaches.

### **4.5.2 Storage Security**

All voters must be able to record their votes anonymously without anybody being able to determine how they voted or change their votes. If voters are going to be able to change their own votes, the system must store their identification information along with their votes. This means that the file storing their votes must be encrypted so that nobody can read it directly at any stage and should be deleted soon after the election ends, but not so soon as to make a recount impossible. It must therefore be as close to impossible as possible for anyone to break into the database for at least the length of the election plus the amount of time the election data is stored afterwards.

### **4.5.3 Maintainability**

The software will be well documented and it will be designed to be modular. The use of object-oriented programming (java) and structured programming (xml), it will also help to increase maintainability. This will make it easier for future developers to make changes and updates to the software with a minimal amount of effort.

### **4.5.4 Scalability**

Both the jdk software and this document are meant to be easily scalable to increase the scope and size of elections. All efforts will thus be made to use a software design that does not have built in size limitations.

### **4.5.5 Reliability**

All efforts will be made to write software that is entirely reliable. However, the viability of electronic voting rests, in part, on the ability of systems administrators and elections officials to incorporate redundancy into any deployed voting system and to develop contingency plans for possible failures.

### **4.5.6 Interface**

All aspects of the online voting system will have a simple point and click interface using menus, text fields, buttons and all of the other components of systems with graphical user interfaces. This interface will be designed to be consistent.

## **5. Other Non-functional Requirements**

### **5.1 Performance Requirements**

- The system should respond in a reasonable amount of time, according to expectations.
- The elector ought to be able to. Once logged in, he/she should receive a response to his requests in 2–3 seconds.

### **5.2 Safety Requirements**

- The outcome of votes cast up until that point must be saved in a database to prevent data loss in the event of a system failure.
- To protect already polled votes, the ECA should be able to immediately shut down the system and stop all connections to the server in the event that a security issue is discovered in the system.

### **5.3 Security Requirements**

- All generated or accepted passwords must be kept in database in an encrypted format.
- ADMIN has the only right to check and count all votes and announce the final result.
- The data transaction between client and server must be encrypted using Firebase default encryption.

# GROUP 3

## Online Voting App

---

Arsh Ansari [20BCE0371]

Madhu Shree k [20BKT0083]

Aditya S Rawat [20BCT0349]

Jaganath T P [20BCE0840]

Shubhangi Tiwari [20BCE2688]



# Problem Statement

- Manual voting system has been deployed for many years in colleges, universities and schools. However, in many colleges people cannot attend the voting because of several reasons.
  - To illustrate, sometimes people may not be in their own registration region and due to this fact they cannot fulfill their voting duties. In order to solve these problems, there is a need of online election voting system in addition to manual voting system.
-

# Scope

## Aim:

The aim of this project is that to design and implement a low cost automated real-time voting system.

## Objective:

Design and develop a software platforms for voter registration, election voting, real-time election results collation and monitoring and mostly for voters remote access to election.

## Applicability:

It will provides a means conduct a more less stressful and fair elections at different levels(faculty, departments, school wide e.t.c) in the university.

# Process Model

The process model used in online voting system is Rapid application development model (RAD). RAD model is used since the product development is short-termed. The model is based on prototyping and iterative cycle with less specific planning. RAD model used in voting system because the requirements are well known and risk involved is limited and can be easily rectified. Voting system has low risk like count error, voting twice etc. This model increases reusability of the features.

# Stakeholders

The term Software Project Stakeholder refers to, “a person, group or company that is directly or indirectly involved in the project and who may affect or get affected by the outcome of the project”.

1. Voters
2. Candidates
3. Admin

# Requirements wrt Stakeholders

**Voters:** The voters should be able to login to the app. They should be able to view all the candidates present for election to get all the information they need for a fair election. They should be able to vote for a candidate successfully. Must have a unique voter id assigned to them.

**Candidates:** Candidates should be able to access the system and must be able to check the status of the results.

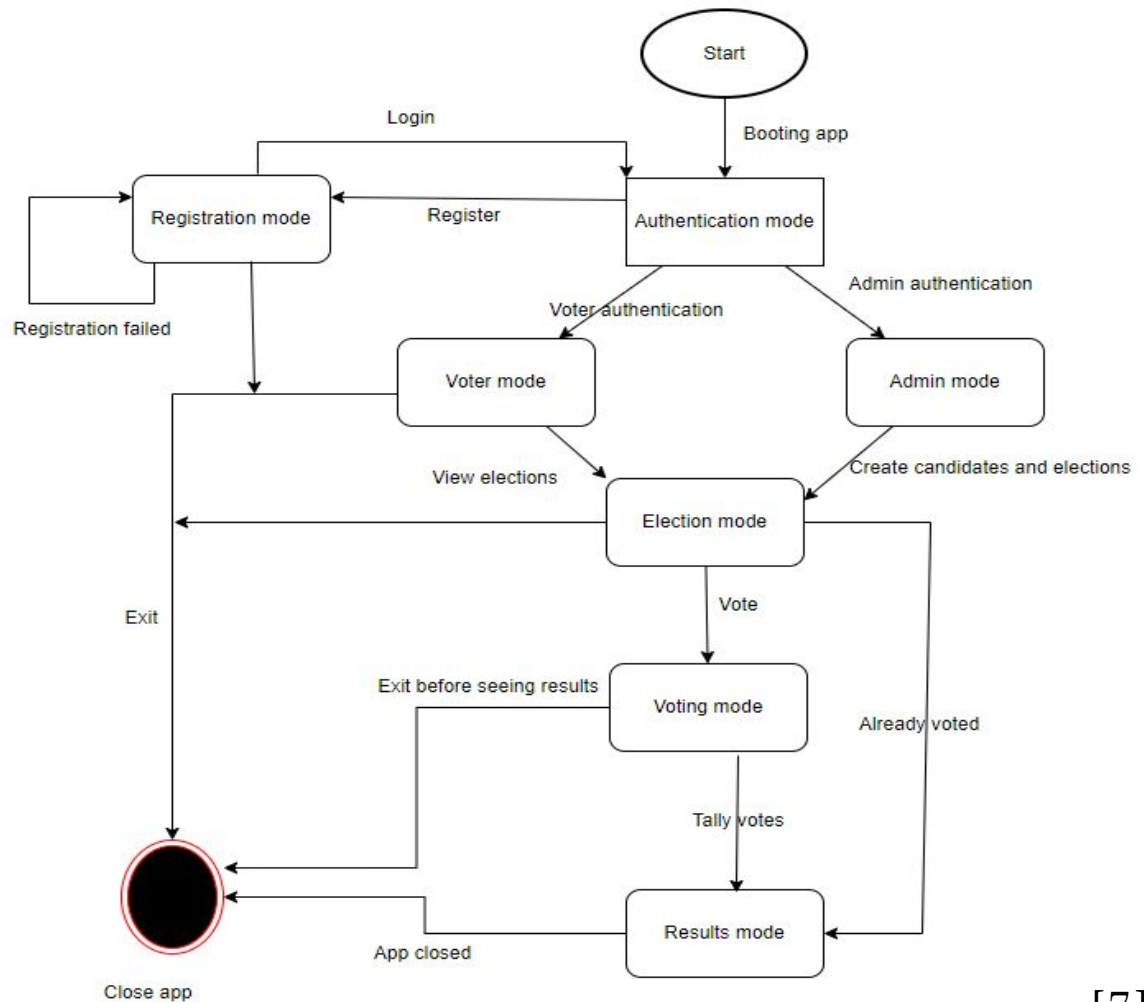
**Admin:** They should have a database of information on all the candidates standing for the elections, Voter ids and voter information. They should be able to create an election type, calculate total votes and send it to election commission. Must be able to control list of candidates and voters registered. Must be able to make changes and monitor the voting activities of user and remove wrong information.

# Viewpoint oriented Analysis

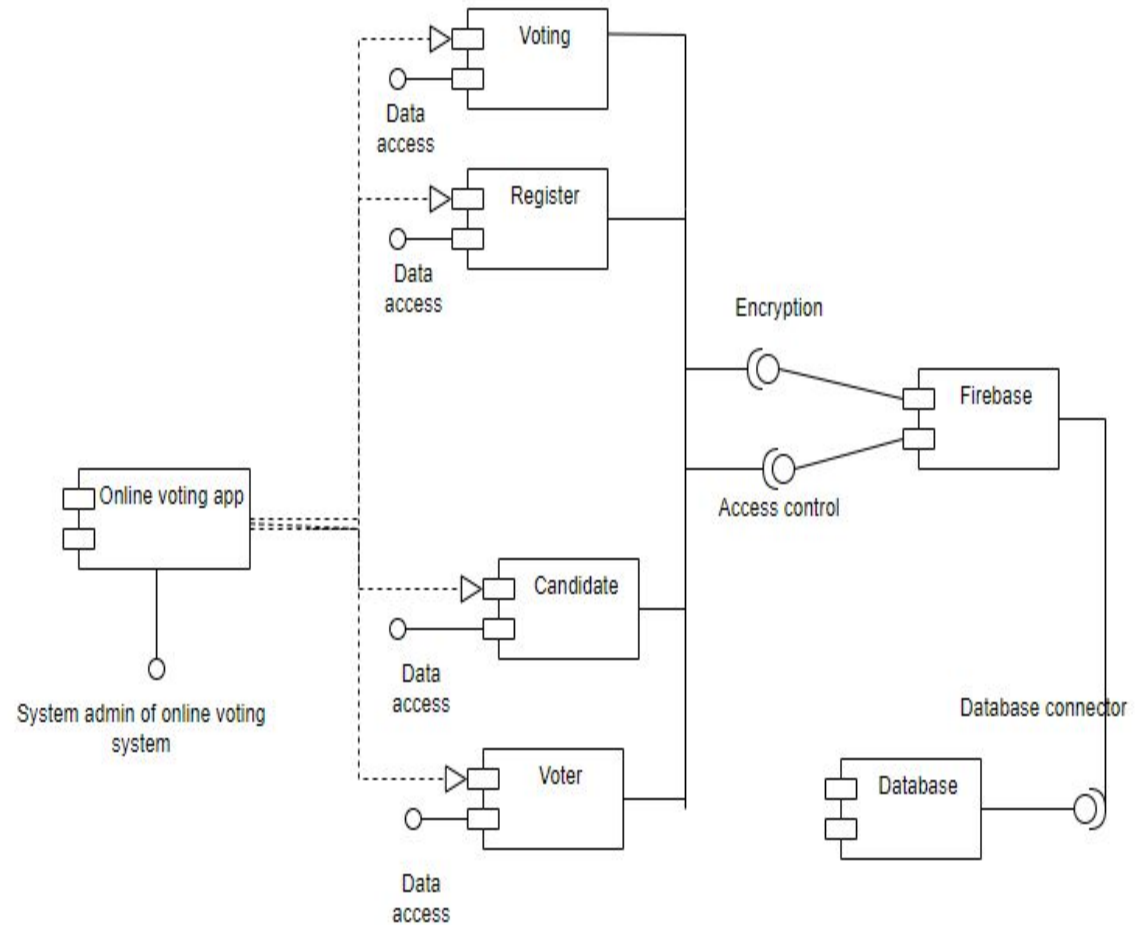
- For e-voting, we will be considering a **behavioural viewpoint**. This is apt for the system as our system deals with different users, and the app changes its functionality according to user behaviour. It will behave differently for the kind of user accessing it.
- Function Viewpoint cannot be considered as it is related to the process involved in the system. Our system deals with less process than data.
- Structural viewpoint can also not be considered because the app is not very information or data oriented.

# System Model

## State Transition Diagram

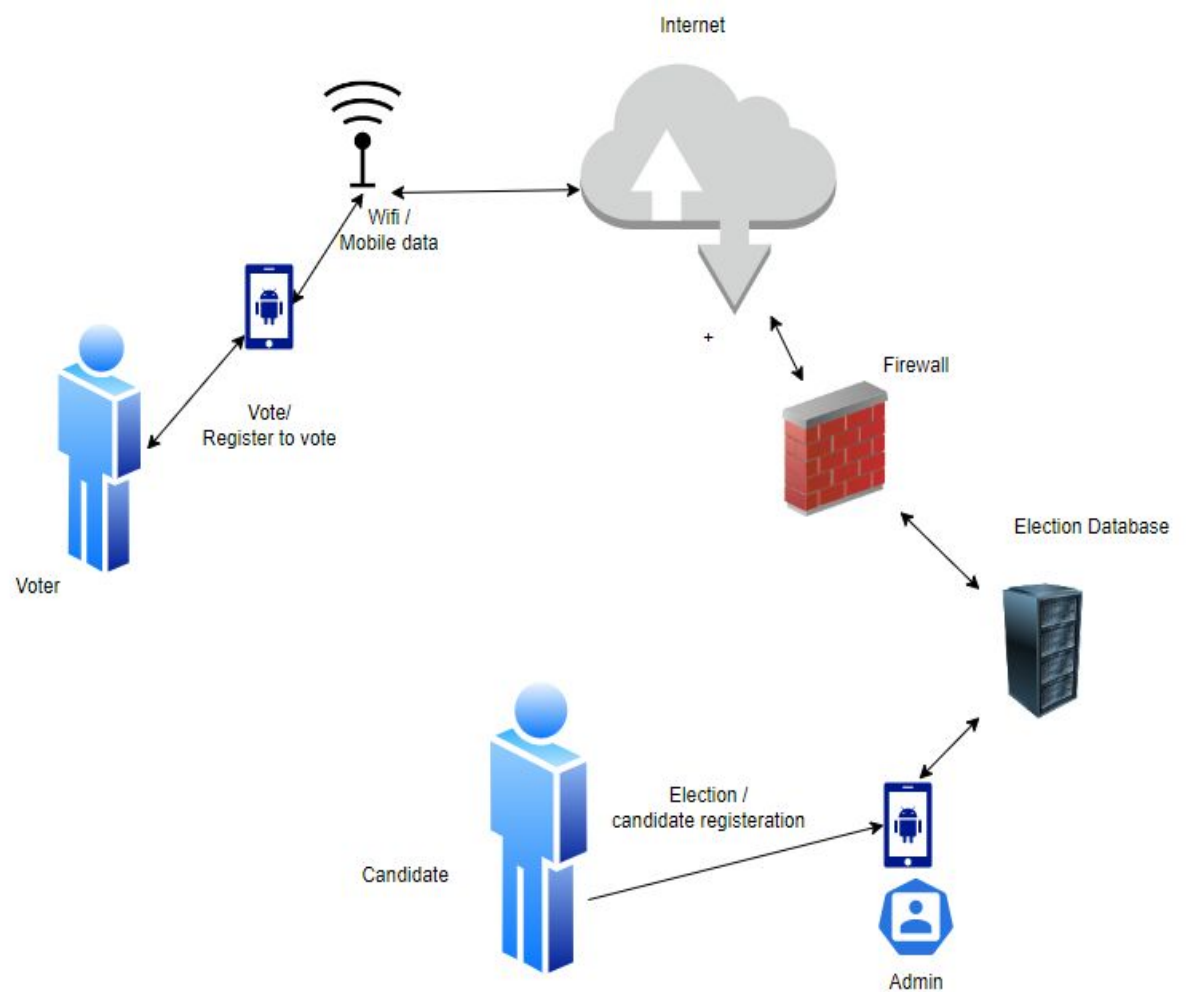


# Component Model





# Deployment Model

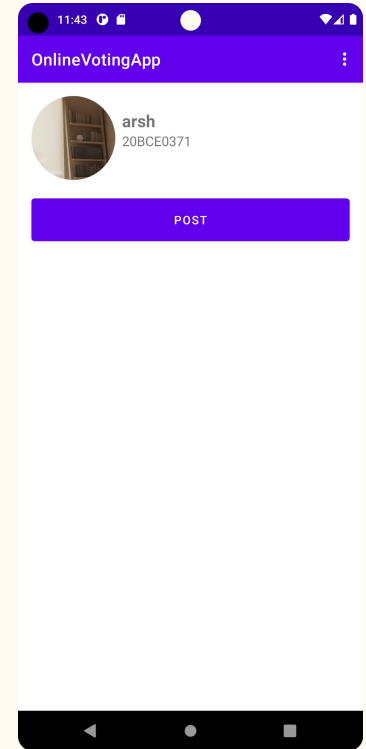
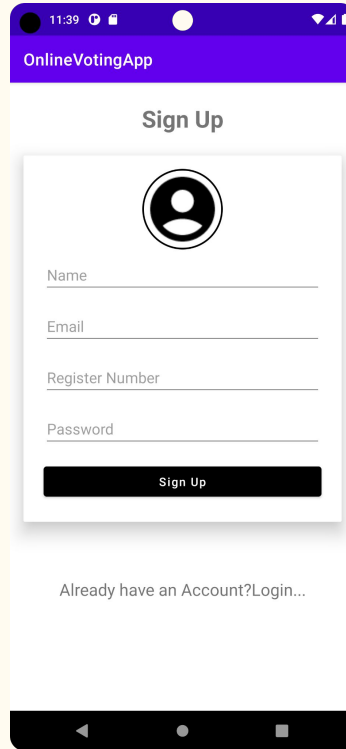
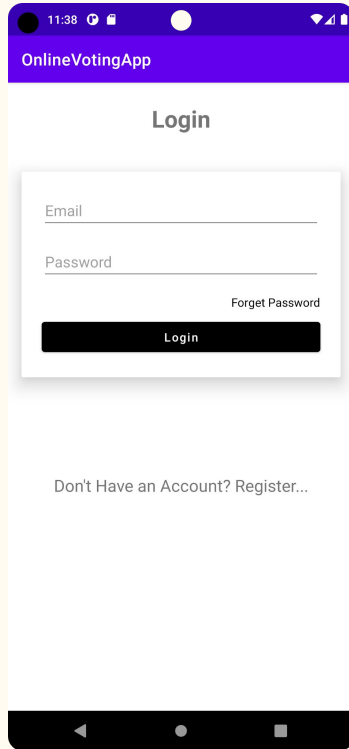


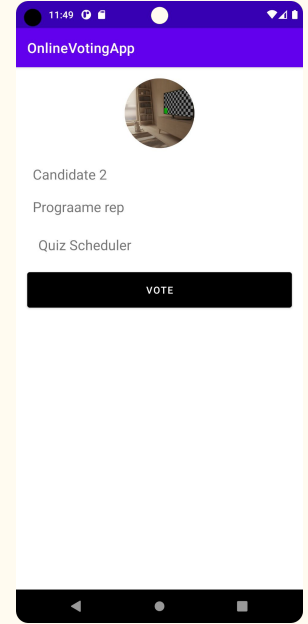
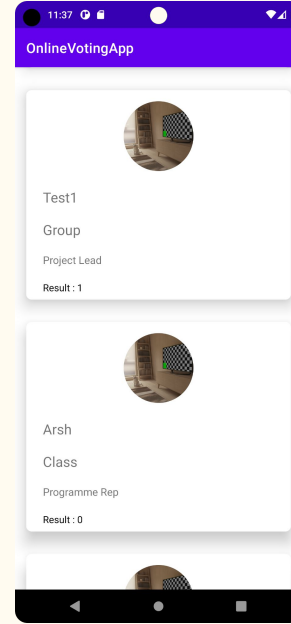
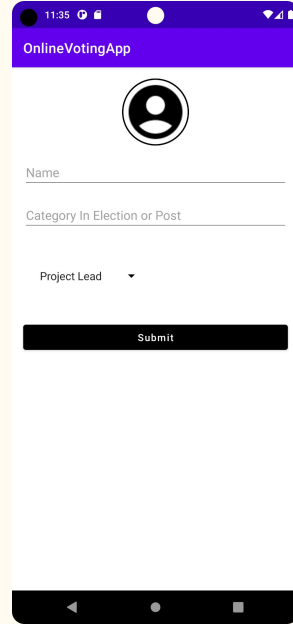
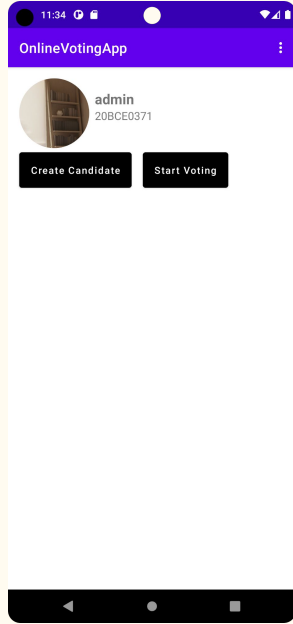
# Test Cases

Test Case Number	Test Case	If (condition = true)	else	Test Data	Actual Outcome	Test Case Status
1.	User Authentication	Login to respective flow	Stay in Same page	Email, password of a registered user	Login Successful if email verified	PASS
2.	Candidate Addition if User is admin	Show Create Candidate button	Show Vote Option	User name	If username is 'admin' create option is visible otherwise not	PASS
3.	No Voting if user has already voted	If 'voted' in then redirect to results page	Allow the user to vote	Device IP and voted status of user	If user has voted result page opens, otherwise he can vote	PASS
4.	Visibility of current status of votes shouldn't be visible unless voted	If not voted result Not shown	Show the results	A user with voted status	A screen if not voted ,if voted result shows	PASS

Test Case Number	Test Case	If (condition =true)	else	Test Data	Actual Outcome	Test Case Status
5.	Admin access should be only editable by database admins	If database admin uses firebase privileges change options enabled	User can't access firebase	--	Firebase backend only accessible to creator or whom permission is authorized	PASS
6.	User should not be able to revote unless database admin decides	If admin deletes the voted field user should be able to vote again	Normal flow	--	Admin is able to enable voting feature of user	PASS

# User Interface Diagrams





THANK YOU!!

---