**Contents**

Contents……………………………………………………………...1

1. Introduction…………………………………………………… 3
   1. Purpose………………………………………………………...3

Motivation……………………………………………………………3

* 1. Literature Survey……………………………………………...3
  2. Existing System………………………………………………..4
  3. Proposed System………………………………………………5

1. Scope…………………………………………………………..6

2.1. Project Scope…………………………………………………..6

2.2. Out of Scope………………………………………………….7

3. Analysis/ Requirement Specifications…………………………8

3.1. User Interface………………………………………………….8

3.2. Software Interface…………………………………………….8

3.3. Hardware Interface……………………………………………8

3.4. Functional Requirements…………………………………….9

3.5. Non-Functional Requirements……………………………….10

4. Design……….……………………………………………….12

4.1. High level conceptual solution Architecture…………..…….12

4.2. UML Diagrams………………………………………………14

5. Implementation………………………………………………18

5.1. Module Description………………………………………….18

5.2. Programs…………………………………………………….19

6. Testing……………………………………………………….38

6.1. Types of Testing……………………………………………..38

6.2. Test cases…………………………………………………….41

7. Results……………………………………………………….43

8. Conclusion and Future Enhancements……………………….51

9. Appendix……………………………………………………..52

10. Bibilography…………………………………………………53

**1. INTRODUCTION**

**1.1 Purpose of the system:**

The project aims to fulfill the 3 basic rights of every citizen,

1. Right to vote

2. Right for freedom of speech

3. Right to know.

The right to vote is fulfilled by the e-voting system that is a part of the application, which saves a lot of voheter time and provides ease of access. You can cast your vote at your fingertips.

Right to freedom of speech is fulfilled by the Public opinion portal which is a key module of the project. It provides users a stable platform for expressing their views. Right to know, is provided where the user (Guest/Registered) can know the current MLA/MP of his/her constituency and the Bio-data of the public representative.

**1.2 Background/Literature Survey of the project:**

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.

Online voting is an electronic way of choosing leaders via a web driven application. The advantage of online voting over the common “queue method” is that the voters have the choice of voting at their own free time and there is reduced congestion. It also minimizes on errors of vote counting. The individual votes are submitted in a database which can be queried to find out who of the aspirants for a given post has the highest number of votes.

This system is geared towards increasing the voting percentage in Kenya since it has been noted that with the old voting method {the Queue System}, the voter turnout has been a wanting case. With system in place also, if high security is applied, cases of false votes shall be reduced.

With the “ONLINE VOTING SYSTEM”, a voter can use his\her voting right online without any difficulty. He\She has to register as a voter first before being authorized to vote. The registration should be done prior to the voting date to enable data update in the database.

However, not just anybody can vote. For one to participate in the elections, he/she must have the requirements. For instance, he/she must be a registered citizen i.e. must be 18 and above years old. As already stated, the project ‘Online Voting' provides means for fast and convenient voting and access to this system is limited only to registered voters.

People are getting more used to work with computers to do all sorts of things, namely sensitive operations such as shopping and home banking and they allow people to vote far from where they usually live, helping to reduce absenteeism rate.

**1.3 Existing System:**

At present there is no application that would provide the above mentioned services in a user friendly manner. The existing voting system is a very lengthy process, people need to wait for hours in the queue to cast their vote. Also, there is no proper platform where people can express their views and get acknowledged.

**Disadvantages:**

1. People are very least aware of the things going on in the politics in the current busy world.
2. Voting at polling stations is very length process and requires a lot of patience.
3. Voters are not known about the candidates that are contesting in the elections.

**1.4 Proposed System:**

We propose the following research question: is there an application which would act as a one-stop-shop for all that you need to know about politics or current issues, the solution is No. We would like to establish one such system that would serve the purpose of every Citizen of this country.

**Advantages:**

The m-governance application would be a one-stop shop for all the important services. User can use it from anywhere in India. Also, online voting (e-voting) would be more convenient, relatively secure and utilize fewer resources. To be able to access e-voting system from a handheld device like a smartphone may be more convenient for many people needing to vote. This could potentially be a solution for the low voter turnout at the polls.

**2. SCOPE**

**2.1. Project Scope**

The main scope of this application is to be able to show the usability of online services to cast vote. Additionally, there will be maintenance of other useful features like

The e-voting system to be discussed makes up a relatively small part of the whole  
election process. From a technical viewpoint the elections are made up of the following components:

* calling of elections,
* registration of candidates,
* preparation of polling list,
* voting (a subset of which is e-voting),
* Counting of votes.
* Opinion poll
* My Neta
* Local &global news.

Other components such as auditing, reviewing of complaints and other supporting activities could be mentioned.

The e-voting app, that will be developed assumes: Voter lists have been prepared and are available in a database. Users are assigned “User” type and Administrators are assigned “Admin” type. The candidate lists have been prepared and are available in a suitable format and lastly, e-votes are counted separately and are later added to the rest of the votes.

**2.2. Out of Scope**

In this aspect we use android studio as a tool to build the app.we use cloud storage to store voters,votes and candidate information.cloud storage named firebase provide the security requirements like authentication, secure database access and updates itself without refreshing.

**3. ANALYSIS/REQUIREMENT SPECIFICATIONS**

**3.1 User Interfaces**

* Splash Screen: It is the first screen which is shown to the user. It loads other modules of the project.
* Main Form: It provides various features to user about the mode of use of software. User can choose text, image options and can work on it.It also provides various other features like user management background verification, picture password, database updations using cloud storage.
* Error Messages: Proper error messages will appear when any error is encountered.

**3.2 Software Interfaces**

* Operating system : Android Ice Cream Sandwich (v.4.0) or latest
* Coding Language : Java & XML
* Data Base : Cloud Database (Parse)
* Documentation : MS Office
* IDE : Android Studio 1.1.0

# 3.3 Hardware Interfaces

* System : 1.3 GHz or higher mobile processor.
* Disk Space : 500MB.
* Display : 4 Inch or more Capacitive Touchscreen
* Ram : 1 GB or more

**3.4 Functional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **FIELD DESCRIPTION** | **FIELD TYPE** | **BUSINESS**  **RULES** |
| Adhar number | Unique 12 digit number given by country to its citizen | Includes numbers. | Size should be 12 digits and should match with DB |
| Voter id | A unique 10 digit number given to citizens greater than 18 yrs to cast vote. | Includes letters and numbers | Size should be 10 digits and number should match with DB. |
| Mobile num | 10 digit mobile number of voter | Includes numbers | Size should be 10 digits and number should match with DB. |
| Mail id | Valid mail id of voter | Includes characters, numbers | Should not be empty and .should match with DB during sign in. |
| Password | Secret key to account. | Includes characters and numbers | It should be minimum of 6 letters. |
| Security questions | Questions asked to voter at the time of sigh up and to cast vote | Includes questions | It should not be empty |
| Picture password | Image with grids each with unique id. | Includes image | Points are properly selected. |
| Vote | It cast vote to respected candidate. | Includes button. | It updates cloud about the vote. |
| Speak now | Opinion of people | Includes radio buttons and submit button | It updates people opinion on a question posted. |
| Elections | Contesting Candidate details are entered | Includes text boxes, dropdown box, button | Candidate details are made available in users app |
| Opinions | Questions are posted to know peoples opinion | Include text box, button | Questions are made available to user app |
| Complaints | Complaints about the user app are displayed | Include text boxes ,button | Complaints should be seen. |
| News | Local and national news are viewed | Includes text | Just view purpose |

**Fig3.4:** FUNCTIONAL REQUIREMENTS

**3.5** **Non-Functional Requirements:**

**Performance Requirements:**

It must be fast to process the request. Especially when the application is processing the data, there might be some instances where a lot of data needs to be transferred. The design needs to be done by making sure that application will operate fast System maintenance must be easy. System will be used be using some additional components. Each of these components might need an update or changes. Applying these changes during the maintenance should be relatively easy.

**Safety Requirements:**

The contents in the cloud are to be safe, because there is a possibility of hacking the credentials of the voter and can cast vote on behalf of voter.. To prevent data loss in case of system failure, the result of votes that are polled till then have to be saved in database.

**Security Requirements:**

The data transaction between client and server must be secure All the passwords that are generated or accepted must be stored in database in secured way. It should prevent unauthorized access.

**Usability:**

It is expected that the user should be able to vote easily online. Administration of the page also should be user friendly. Provide step by step guide for both admin and users. User should complete voting in a few minutes. Provide an online help. Provide also a quick guide for users.

**Scalability:**

It should provide access to one user at a time. Multiple users can cast vote through single app.

**Reliability:**

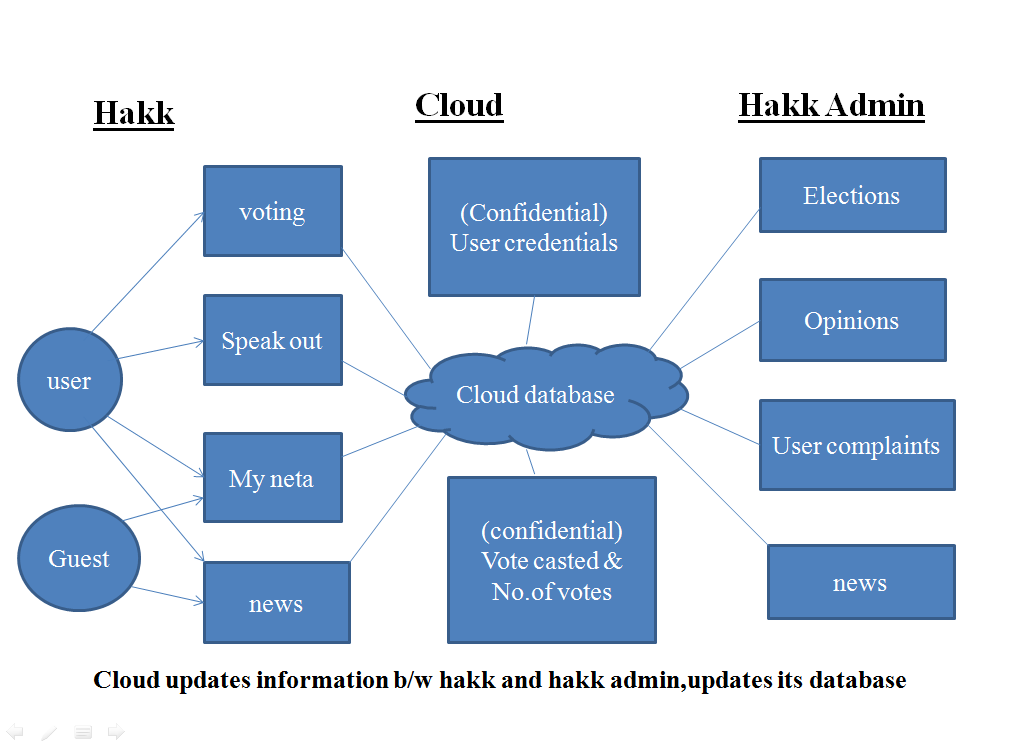
The system should be reliable. Security is a major concern for an e-voting system. Process used in this system should be secure enough to be able to meet the requirements mentioned for e-voting. It requires database connections and network connections. Changes can be done in the databases to store the votes. All changes needs to be confirmed and if the transfer is complete the confirmation should be displayed. The changes should be monitored.

**Availability:**

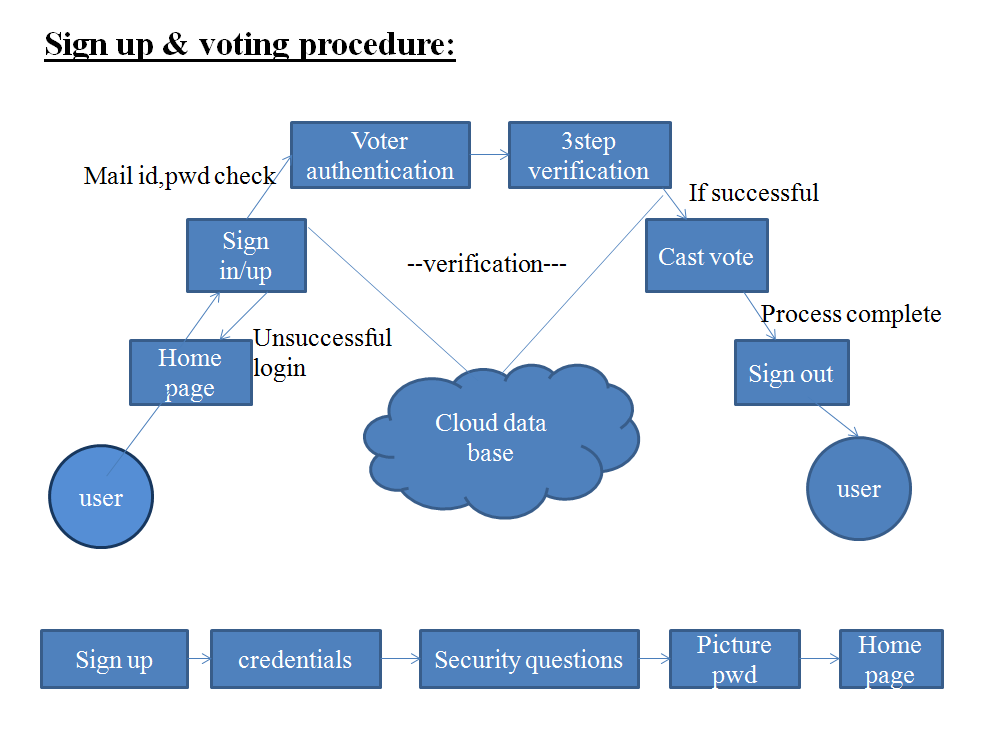
The application should be made available in minimum down-time.the application should provider user with appropriate error messages and should handle run-time exceptions in a controlled manner in order to avoid abnormal termination.

**4. DESIGN**

**4.1 High level conceptual solution Architecture:**

****

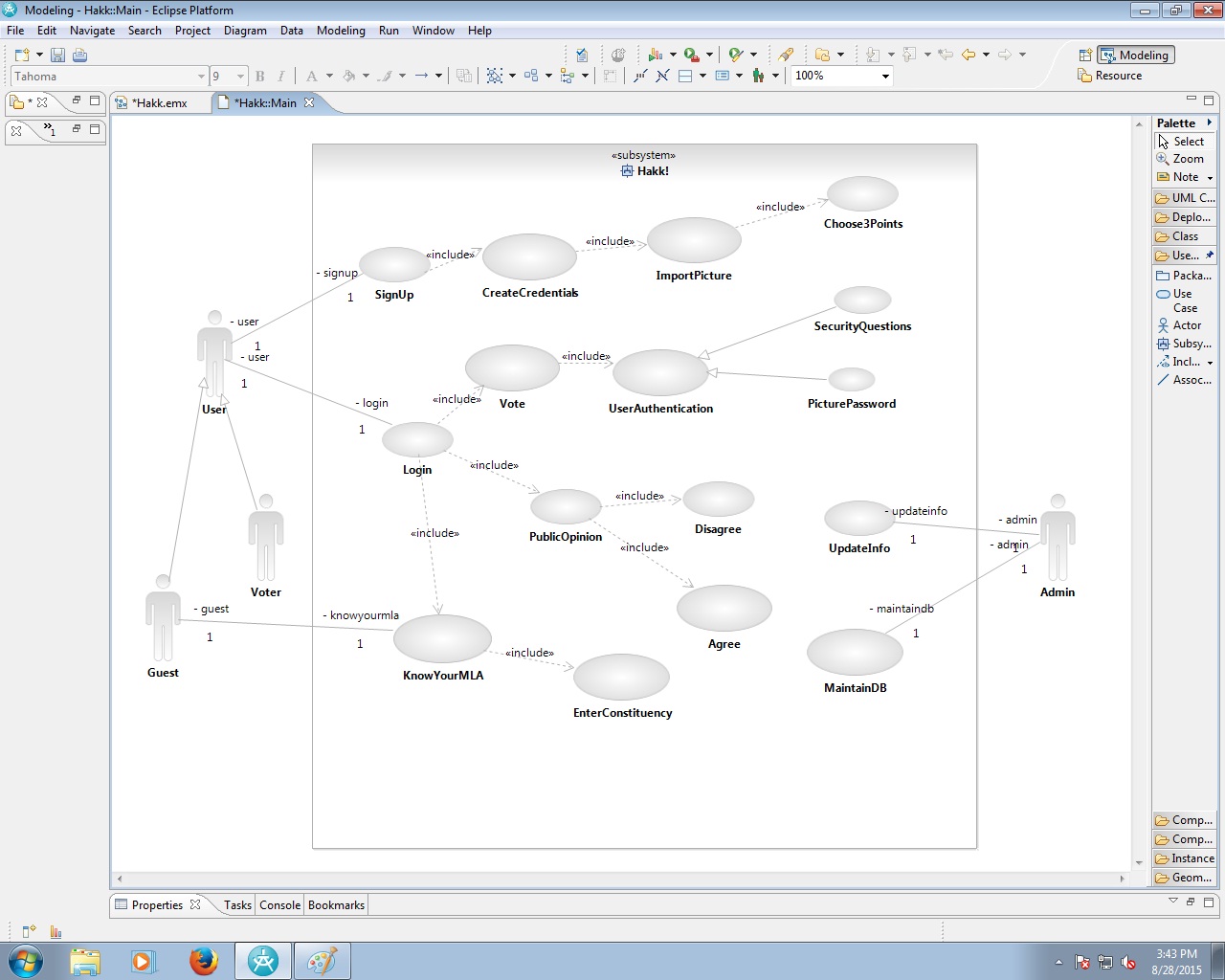
**Fig4.1.1:** ARCHITECTURE

****

**Fig4.1.2:** SIGNUP AND VOTING

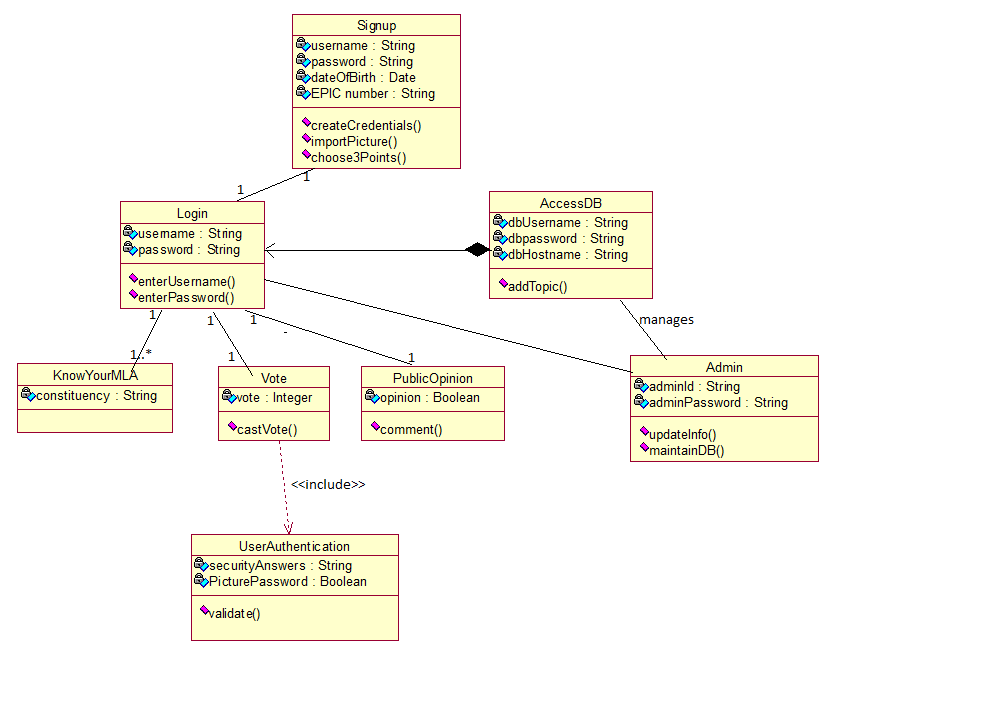
**4.2 UML Diagrams**

**4.2.1. Use Case Diagram:**



**Fig4.2.1:** USE CASE DIAGRAM

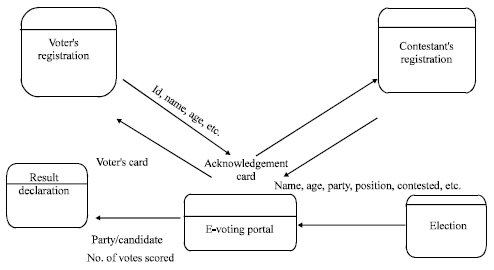
The above diagram is Use Case diagram of our system. It shows the set of actions performed by various users. In our system we have 3 types of users. They are 1) Administrator 2) Voter and 3) Guest. As described earlier, the content in the Ovals are actions performed in the system and those actors are like symbols represent users in system. Those dashed lines from user to action means users are performing those actions respectively.

**4.2.2. Class Diagram:** 

**Fig4.2.2**: CLASS DIAGRAM

The above diagram represents class diagram of our system i.e., it shows various classes used in our system and the relationship with one class to other in the system. Each rectangle box represents a class and the upper portion of it represents class name and middle portion represents attributes of the class and the lower represents the functions performed by that class.

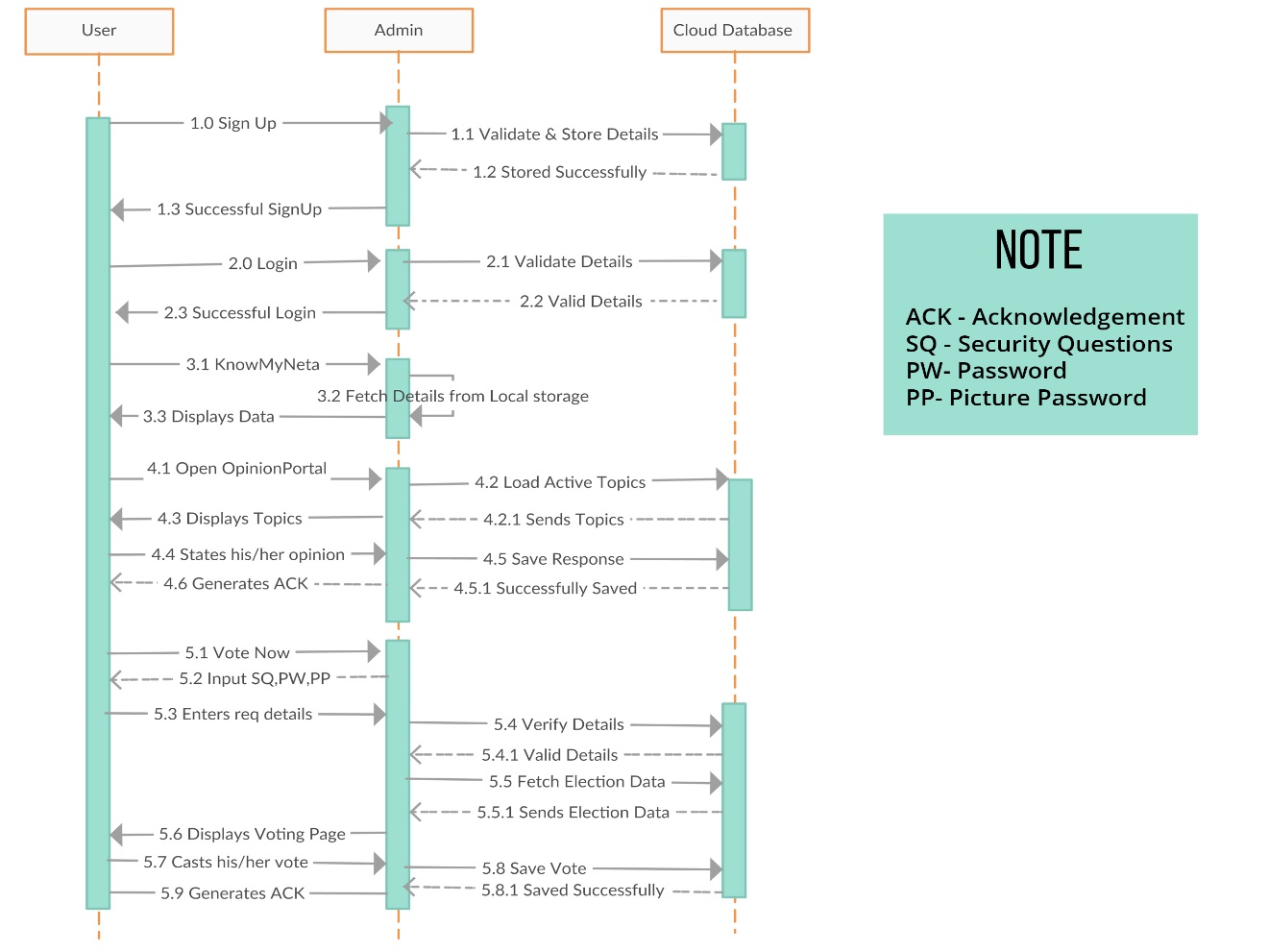
**4.2.3.Dataflow Diagram**



**Fig4.2.3:** DATAFLOW DIAGRAM

The above diagram represents dataflow diagram of the system i.e., it represents the flow of data in our project for Mobile. On seeing it we can understand the flow activities that has to be gone from start to end.

**4.2.4.Sequence Diagram**

****

**Fig4.2.4:** SEQUENCE DIAGRAM

**5. IMPLEMENTATION**

**5.1 Module Description**

1. Three level authentication system.
2. An online opinion portal (Speak Out!)
3. e-Voting system working model
4. Information about public representatives

**Three Level Authentication system:**

In E-Voting, to prevent voter frauds we use three levels of authentication. A Voter ID (EPIC number), UID (Aadhar Number) and password are used as the first level of security. The data entered by the user is verified with the contents of the database, if the data is correct then the user need to enter the answers for the high security questions (entered at the time of registration) in the final level the user need to unlock (decode) a picture password, which the user has set at the time of registration. If the data entered by the user in all the three levels is found valid, he gets access to the voting system. The System validates the user and gives the permission to cast his vote and records the UID and EPIC number of the user to prevent casting of multiple votes.

**Online Opinion Portal (Speak Out):**

In public opinion system (Speak Out!) , the registered users has the right to express their opinions on the schemes and the policies implemented by the government and the cumulative data would be displayed depicting the voice of the people and giving a clear picture whether the government has the acceptance from the people or not.

**E-Voting System:**

In e-voting system, people can cast their vote online using the mobile application installed in their smartphone. In order to prevent voter frauds we use three levels of security. A user ID (EPIC number), UID (Aadhar Number) and password are used as the first level of security. The data entered by the user is verified with the contents of the database, if the data is correct then the user need to enter the answers for the high security questions (entered at the time of registration) in the final level the user need to unlock (decode) a picture password, which the user has set at the time of registration. If the data entered by the user in all the three levels is found valid, he gets access to the voting system. The System validates the user and gives the permission to cast his vote and records the UID and EPIC number of the user to prevent casting of multiple votes.

**Information about public representatives:**

Apart from these services the user (Guest/Registered) can also know the current MLA/MP of his/her constituency and the Bio-data of the public representative. The details also include mobile number and email of the representative.

Apart from these services the user (Guest/Registered) can also know the current MLA/MP of his/her constituency and the Bio-data of the public representative. The details also include mobile number and email of the

**5.2 Programs**

**Code showing the verification of Successful LOGIN**

frg.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i2 = new Intent(MainActivity.this,Forgot.class);

startActivity(i2);

}

});

gstj.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent gs = new Intent(MainActivity.this,KnowYourMLA.class);

startActivity(gs);

}

});

sgi.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

boolean validationError = false;

StringBuilder validationErrorMessage =

new StringBuilder("Please enter your ");

if (isEmpty(use)) {

validationError = true;

validationErrorMessage.append("username");

}

if (isEmpty(paw)) {

if (validationError) {

validationErrorMessage.append(" and ");

}

validationError = true;

validationErrorMessage.append("password");

}

validationErrorMessage.append(".");

// If there is a validation error, display the error

if (validationError) {

Toast.makeText(MainActivity.this, validationErrorMessage.toString(), Toast.LENGTH\_LONG)

.show();

return;

}

ConnectivityManager cm = (ConnectivityManager) getSystemService(Context.CONNECTIVITY\_SERVICE);

NetworkInfo ni = cm.getActiveNetworkInfo();

if (ni == null) {

// There are no active networks.

Toast.makeText(MainActivity.this, "Network not found. Please check your Internet Connection", Toast.LENGTH\_LONG).show();

//return false;

} else {

//return true;

// Set up a progress dialog

final ProgressDialog dlg = new ProgressDialog(MainActivity.this);

dlg.setTitle("Please wait.");

dlg.setMessage("Logging in. Please wait.");

dlg.show();

// Call the Parse login method

ParseUser.logInInBackground(use.getText().toString(), paw.getText().toString(), new LogInCallback() {

@Override

public void done(ParseUser parseUser, com.parse.ParseException e) {

dlg.dismiss();

if (e != null) {

// Show the error message

Toast.makeText(MainActivity.this, "Invalid Login Credentials", Toast.LENGTH\_LONG).show();

warj.setVisibility(View.VISIBLE);

invj.setText("Invalid Login Credentials"); }

else {

// Start an intent for the dispatch activity

warj.setVisibility(View.INVISIBLE);

invj.setText("");

Intent intent = new Intent(MainActivity.this, HomeActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

}

}

});

}

}

});

sgu.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent iv = new Intent (MainActivity.this,SignUpActivity.class);

startActivity(iv);

}

});

}

private boolean isEmpty(EditText etText) {

if (etText.getText().toString().trim().length() > 0) {

return false;

} else {

return true;

}

}

}

**Code showing the Layout design using XML**

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools" android:layout\_width="match\_parent"

android:layout\_height="match\_parent" android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:background="#0AA4A2"

tools:context="com.photuva.splashhakk.QuestionActivity">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<ImageView

android:layout\_width="210dp"

android:layout\_height="105dp"

android:src="@mipmap/spk"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="30dp"

/>

</RelativeLayout>

<Button

android:layout\_width="match\_parent"

android:layout\_height="120dp"

android:text="Question"

android:id="@+id/q1"

android:textSize="17dp"

android:textAllCaps="false"

android:background="@drawable/mybutton2"

android:textColor="#ffffff"

android:layout\_marginTop="30dp" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:gravity="center"

android:layout\_marginLeft="20dp"

android:layout\_marginRight="20dp"

android:layout\_marginTop="10dp"

>

<RadioGroup

android:id="@+id/qg1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<RadioButton

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:text="YES"

android:width="40dp"

android:height="40dp"

android:textSize="30dp"

android:textColor="#ffffff"

android:layout\_marginLeft="20dp"

android:buttonTint="#ffffff"

android:id="@+id/ag1"

android:layout\_gravity="center\_horizontal"

android:gravity="left|center\_vertical" />

<RadioButton

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:text="NO"

android:width="40dp"

android:height="40dp"

android:textSize="30dp"

android:textColor="#ffffff"

android:buttonTint="#ffffff"

android:id="@+id/dag1"

android:layout\_gravity="center\_horizontal"

android:gravity="left|center\_vertical" />

</RadioGroup>

</LinearLayout>

<Button

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:text="Submit!"

android:id="@+id/sbmt1x"

android:textAllCaps="false"

android:textSize="24dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="10dp"

android:background="@drawable/mybutton"

android:textColor="#3AA0B3" />

<Button

android:layout\_width="match\_parent"

android:layout\_height="120dp"

android:text="Question"

android:id="@+id/q2"

android:textSize="17dp"

android:textAllCaps="false"

android:background="@drawable/mybutton2"

android:textColor="#ffffff"

android:layout\_marginTop="50dp" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:gravity="center"

android:layout\_marginLeft="20dp"

android:layout\_marginRight="20dp"

android:layout\_marginTop="10dp"

>

<RadioGroup

android:id="@+id/qg2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<RadioButton

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:text="YES"

android:width="40dp"

android:height="40dp"

android:textSize="30dp"

android:textColor="#ffffff"

android:layout\_marginLeft="20dp"

android:buttonTint="#ffffff"

android:id="@+id/ag2"

android:layout\_gravity="center\_horizontal"

android:gravity="left|center\_vertical" />

<RadioButton

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:text="NO"

android:width="40dp"

android:height="40dp"

android:textSize="30dp"

android:textColor="#ffffff"

android:buttonTint="#ffffff"

android:id="@+id/dag2"

android:layout\_gravity="center\_horizontal"

android:gravity="left|center\_vertical" />

</RadioGroup>

</LinearLayout>

<Button

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:text="Submit!"

android:id="@+id/sbmt2x"

android:textAllCaps="false"

android:textSize="24dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="10dp"

android:background="@drawable/mybutton"

android:textColor="#3AA0B3" />

<Button

android:layout\_width="match\_parent"

android:layout\_height="120dp"

android:text="Question"

android:id="@+id/q3"

android:textSize="17dp"

android:textAllCaps="false"

android:background="@drawable/mybutton2"

android:textColor="#ffffff"

android:layout\_marginTop="50dp" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:gravity="center"

android:layout\_marginLeft="20dp"

android:layout\_marginRight="20dp"

android:layout\_marginTop="10dp"

>

<RadioGroup

android:id="@+id/qg3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<RadioButton

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:text="YES"

android:width="40dp"

android:height="40dp"

android:textSize="30dp"

android:textColor="#ffffff"

android:layout\_marginLeft="20dp"

android:buttonTint="#ffffff"

android:id="@+id/ag3"

android:layout\_gravity="center\_horizontal"

android:gravity="left|center\_vertical" />

<RadioButton

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:text="NO"

android:width="40dp"

android:height="40dp"

android:textSize="30dp"

android:textColor="#ffffff"

android:buttonTint="#ffffff"

android:id="@+id/dag3"

android:layout\_gravity="center\_horizontal"

android:gravity="left|center\_vertical" />

</RadioGroup>

</LinearLayout>

<Button

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:text="Submit!"

android:id="@+id/sbmt3x"

android:textAllCaps="false"

android:textSize="24dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="10dp"

android:background="@drawable/mybutton"

android:textColor="#3AA0B3" />

<Button

android:layout\_width="match\_parent"

android:layout\_height="120dp"

android:text="Question"

android:id="@+id/q4"

android:textSize="17dp"

android:textAllCaps="false"

android:background="@drawable/mybutton2"

android:textColor="#ffffff"

android:layout\_marginTop="50dp" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:gravity="center"

android:layout\_marginLeft="20dp"

android:layout\_marginRight="20dp"

android:layout\_marginTop="10dp"

>

<RadioGroup

android:id="@+id/qg4"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<RadioButton

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:text="YES"

android:width="40dp"

android:height="40dp"

android:textSize="30dp"

android:textColor="#ffffff"

android:layout\_marginLeft="20dp"

android:buttonTint="#ffffff"

android:id="@+id/ag4"

android:layout\_gravity="center\_horizontal"

android:gravity="left|center\_vertical" />

<RadioButton

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:text="NO"

android:width="40dp"

android:height="40dp"

android:textSize="30dp"

android:textColor="#ffffff"

android:buttonTint="#ffffff"

android:id="@+id/dag4"

android:layout\_gravity="center\_horizontal"

android:gravity="left|center\_vertical" />

</RadioGroup>

</LinearLayout>

<Button

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:text="Submit!"

android:id="@+id/sbmt4x"

android:textAllCaps="false"

android:textSize="24dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="10dp"

android:background="@drawable/mybutton"

android:textColor="#3AA0B3" />

<Button

android:layout\_width="match\_parent"

android:layout\_height="120dp"

android:text="Question"

android:id="@+id/q5"

android:textSize="17dp"

android:textAllCaps="false"

android:background="@drawable/mybutton2"

android:textColor="#ffffff"

android:layout\_marginTop="50dp" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:gravity="center"

android:layout\_marginLeft="20dp"

android:layout\_marginRight="20dp"

android:layout\_marginTop="10dp"

>

<RadioGroup

android:id="@+id/qg5"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<RadioButton

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:text="YES"

android:width="40dp"

android:height="40dp"

android:textSize="30dp"

android:textColor="#ffffff"

android:layout\_marginLeft="20dp"

android:buttonTint="#ffffff"

android:id="@+id/ag5"

android:layout\_gravity="center\_horizontal"

android:gravity="left|center\_vertical" />

<RadioButton

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:text="NO"

android:width="40dp"

android:height="40dp"

android:textSize="30dp"

android:textColor="#ffffff"

android:buttonTint="#ffffff"

android:id="@+id/dag5"

android:layout\_gravity="center\_horizontal"

android:gravity="left|center\_vertical" />

</RadioGroup>

</LinearLayout>

<Button

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:text="Submit!"

android:id="@+id/sbmt5x"

android:textAllCaps="false"

android:textSize="24dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="20dp"

android:background="@drawable/mybutton"

android:textColor="#3AA0B3" />

</LinearLayout>

</ScrollView>

**Code showing the prevention of Multiple Votes**

case R.id.vt12: {

AlertDialog.Builder alertDialogBuilder12 = new AlertDialog.Builder(VoteAcrivity.this);

// set title

alertDialogBuilder12.setTitle("Confirm your vote");

// set dialog message

alertDialogBuilder12

.setMessage("Click YES to confirm")

.setCancelable(false)

.setPositiveButton("YES", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog12, int id) {

final ParseUser use = ParseUser.getCurrentUser();

final int ans = use.getInt("vote");

ParseQuery<ParseObject> query = ParseQuery.getQuery("WarangalByPoll");

query.getInBackground("PU6zGs3zuP", new GetCallback<ParseObject>() {

public void done(ParseObject pobj, ParseException e) {

if (e == null && ans==0) {

pobj.increment("votes");

pobj.saveInBackground();

use.put("vote",1);

String cand = pobj.getString("candidate");

use.saveInBackground();

Intent f2 = new Intent(VoteAcrivity.this,VoteSuccActivity.class);

f2.putExtra("CANDIT", cand);

startActivity(f2);

// Toast.makeText(VoteAcrivity.this, "Your response has been submitted. Cheers!", Toast.LENGTH\_LONG).show();

}

else {

Toast.makeText(VoteAcrivity.this, "You have already casted your vote. Please go back to Homepage", Toast.LENGTH\_LONG).show();

}

}

});

dialog12.cancel();

}

})

.setNegativeButton("NO", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog12, int id) {

// if this button is clicked, just close

// the dialog box and do nothing

dialog12.cancel();

}

});

// create alert dialog

AlertDialog alertDialog12 = alertDialogBuilder12.create();

// show it

alertDialog12.show();

break;

}

**6. TESTING**

**6.1 Types of Testing**

**1. Unit testing:**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**2**. **Integration testing:**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**3**. **Functional testing:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

* Valid Input : identified classes of valid input must be accepted.
* Invalid Input : identified classes of invalid input must be rejected.
* Functions : identified functions must be exercised.
* Output : identified classes of application outputs must be exercised.
* Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**4. System Testing:**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**5. White Box Testing:**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**6. Black Box Testing:**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**7. Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach Field testing will be performed manually and functional tests will be written in detail.

Test objectives

• All field entries must work properly.

• Pages must be activated from the identified link.

• Verify that the entries are of the correct format

• No duplicate entries should be allowed

• All links should take the user to the correct page.

**8. Integration Testing:**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

**9. Acceptance Testing:**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. Test Results: All the test cases mentioned above passed successfully. No defects encountered.

**6.2 Test Cases**

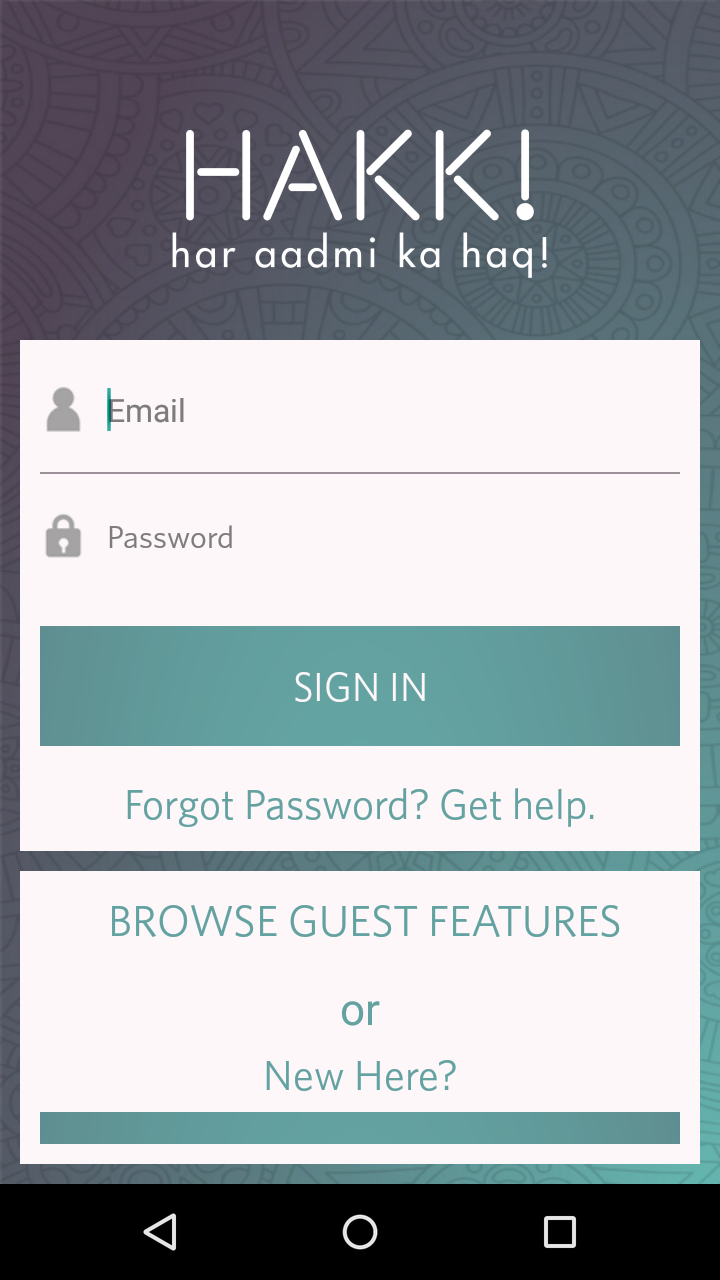
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step #** | **Step details** | **Expected Results** | **Actual Results** | **Pass/Fail**  **Notexecuted/**  **Suspended** |
| 1. | No network connection | Warning message showing no network detected | Warning message showing no network detected | Pass |
| 2. | Database Updation | Immediate update of information should be observed | Immediate update of information is observed | Pass |
| 3. | Empty textboxes and invalid entry of Credentials | Warning message to enter required/valid details | Warning message to enter required/valid details | Pass |
| 4. | Duplication of Credentials | Message showing that details are already used | Message showing that details are already used | Pass |
| 5. | Empty security questions and answers | Message showing that please enter the required details | Message showing that please enter the required details | Pass |
| 6. | No points are selected for Picture Password | Warning message asking to select the points | Warning message asking to select the points | Pass |
| 7. | Empty/Invalid sign in details | Warning message to enter required/valid details | Warning message to enter required/valid details | Pass |
| 8. | Duplication of Votes | Warning message showing you have already casted your vote | Warning message showing you have already casted your vote | Pass |
| 9. | Opinion response updation | Message showing successfully updated in the database | Message showing successfully updated in the database | Pass |
| 10. | Multiple time response for questions | Message showing you have already this question | Message showing you have already this question | Pass |
| 11. | My neta selection priority MLA/MP | Constituencies should be selected accordingly in the dropdown list | Constituencies should be selected accordingly in the dropdown list | Pass |
| 12. | Change Password | Message showing updated successfully | Message showing updated successfully | Pass |
| 13. | Administration sign in credentials | Message showing enter the required/valid details | Message showing enter the required/valid details | Pass |
| 14. | Posting questions in speak out | Message showing your question has been posted successfully | Message showing your question has been posted successfully | Pass |
| 15. | Adding candidates to the list | Message showing candidate added successfully | Message showing candidate added successfully | Pass |

**Fig6.2:** TESTCASES

**7. RESULTS**

**HAKK APPLICATION:**

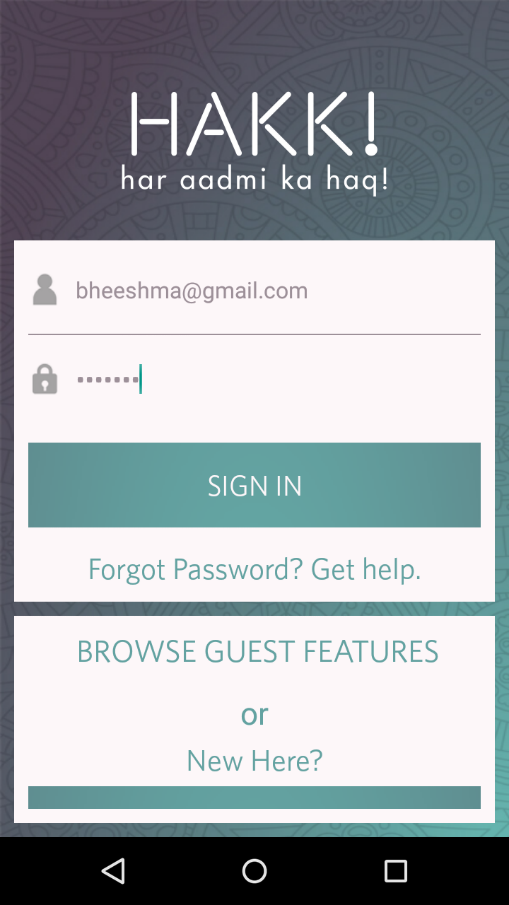
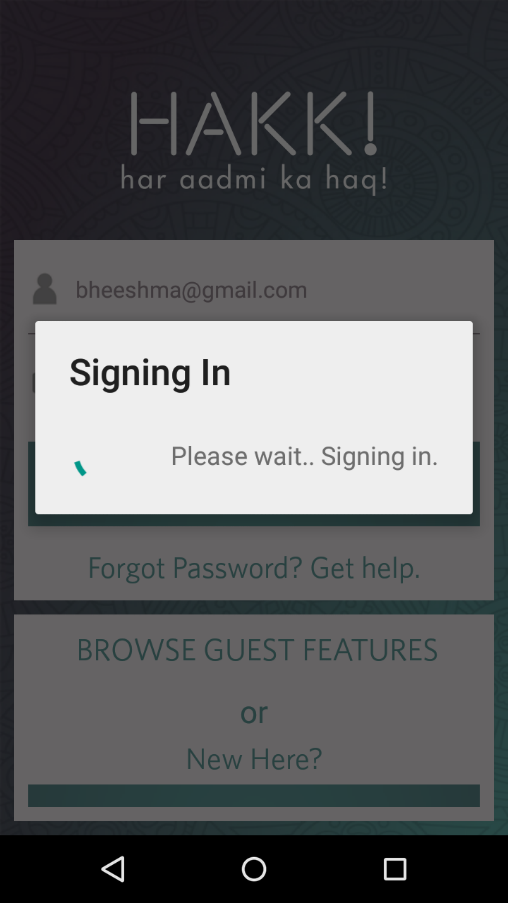
**Fig7.1: HOME PAGE:**

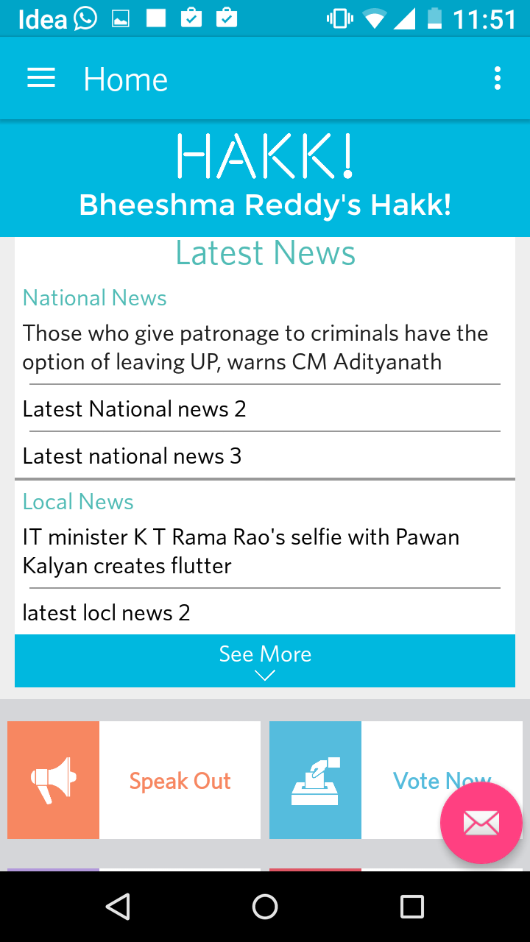
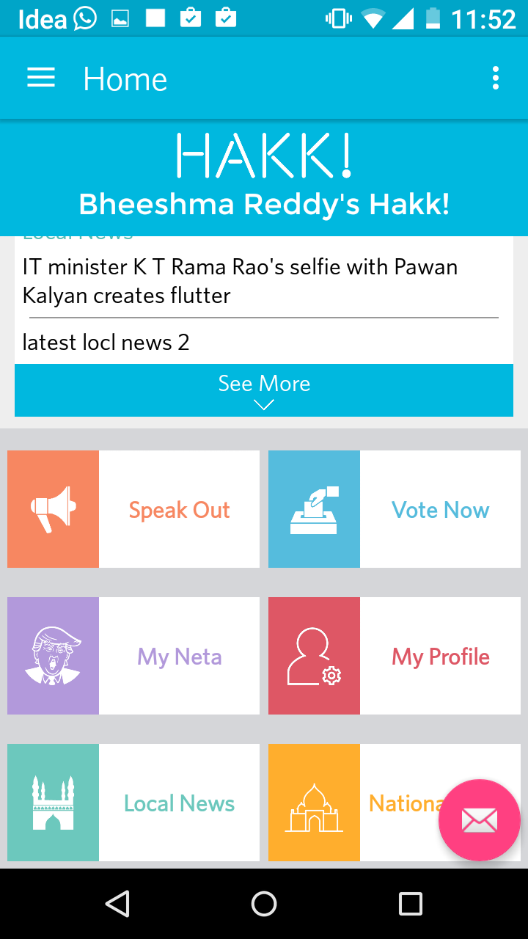
****

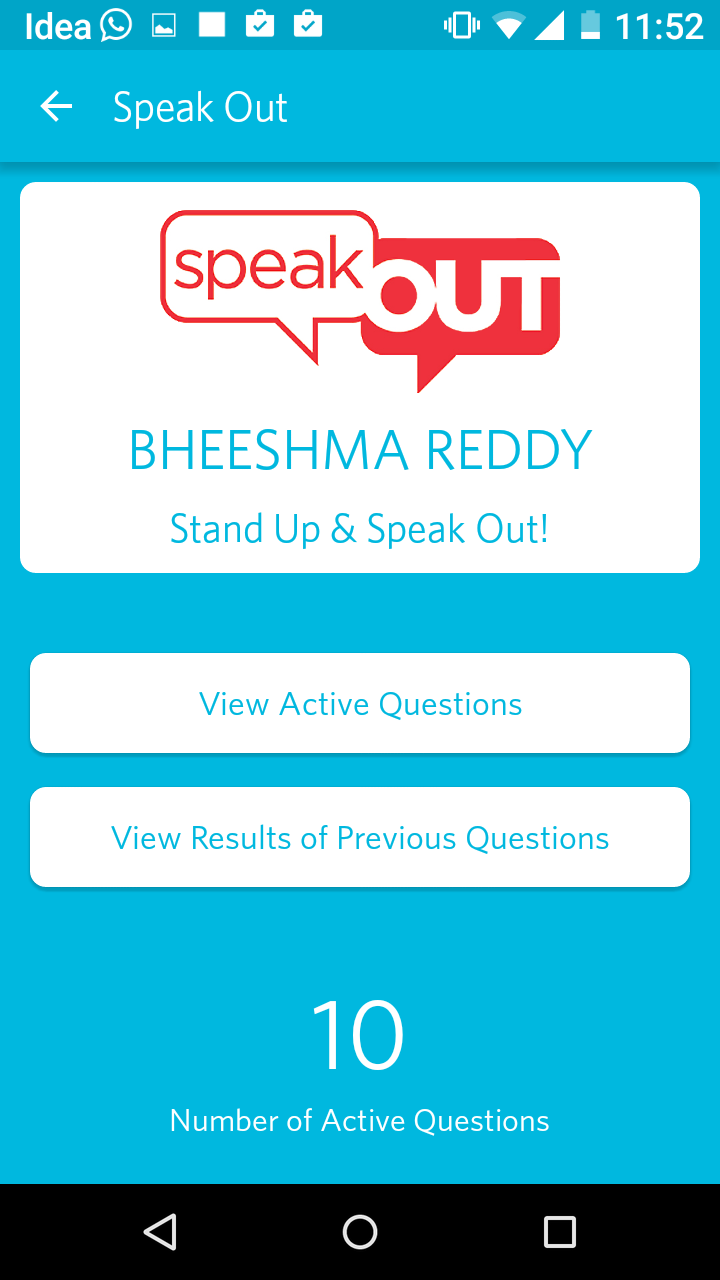
**Fig7.2: GUEST FEATURES:**

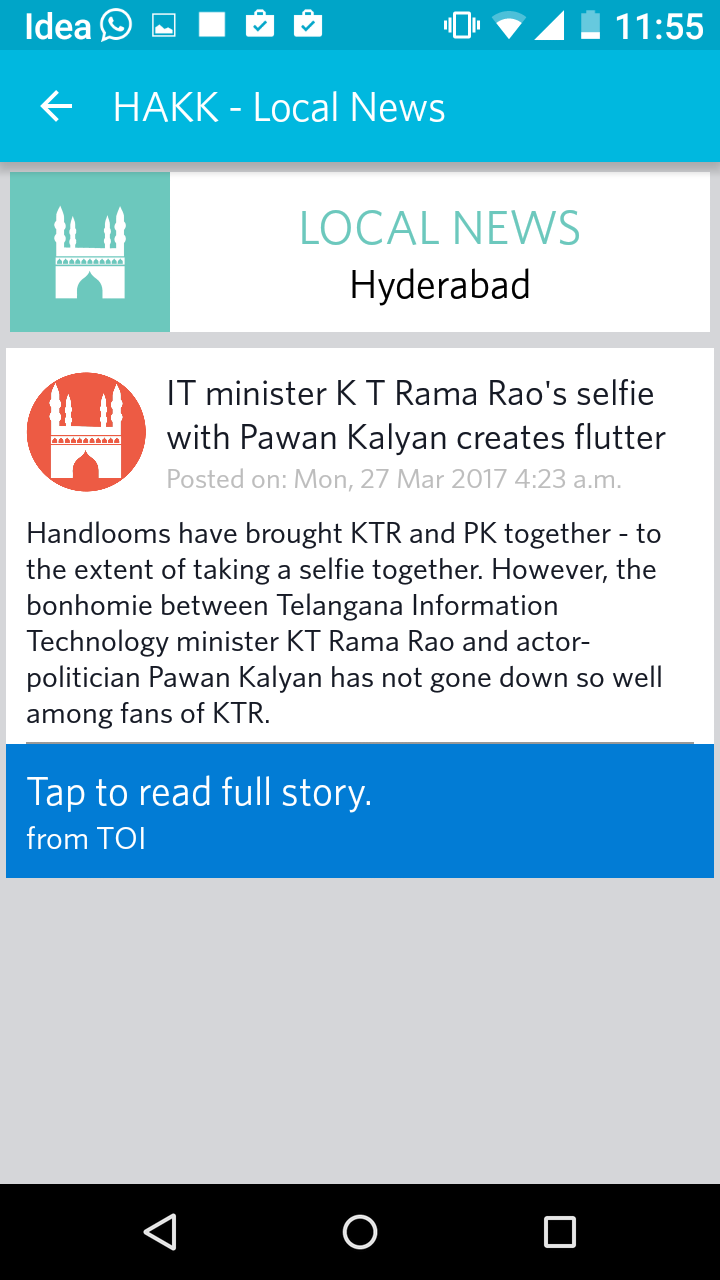
** **

**Fig7.3: SIGN IN & ACTIVITIES:**

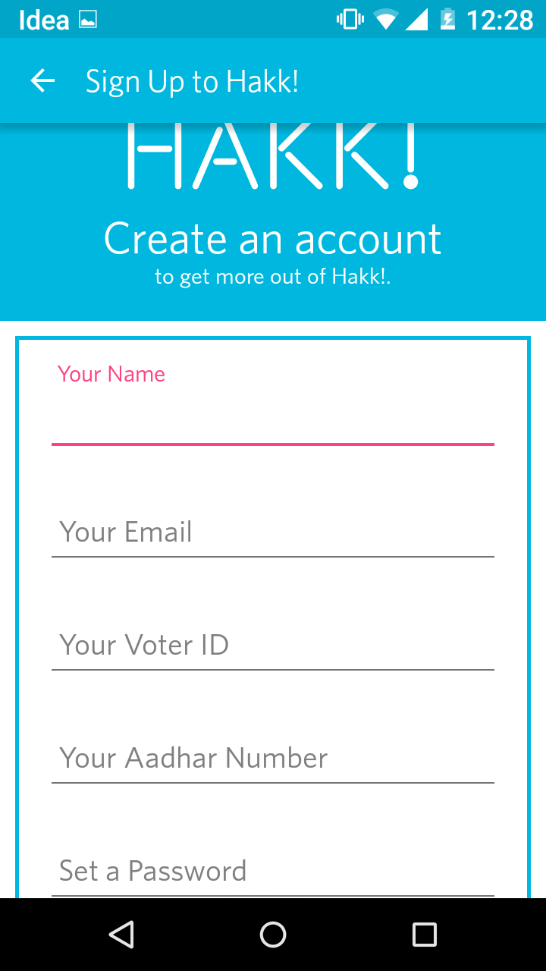
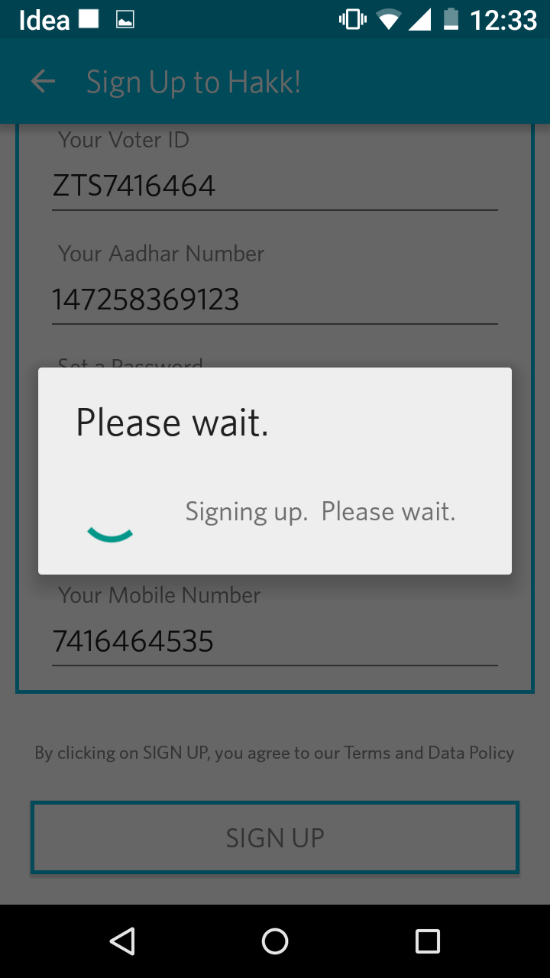
** **

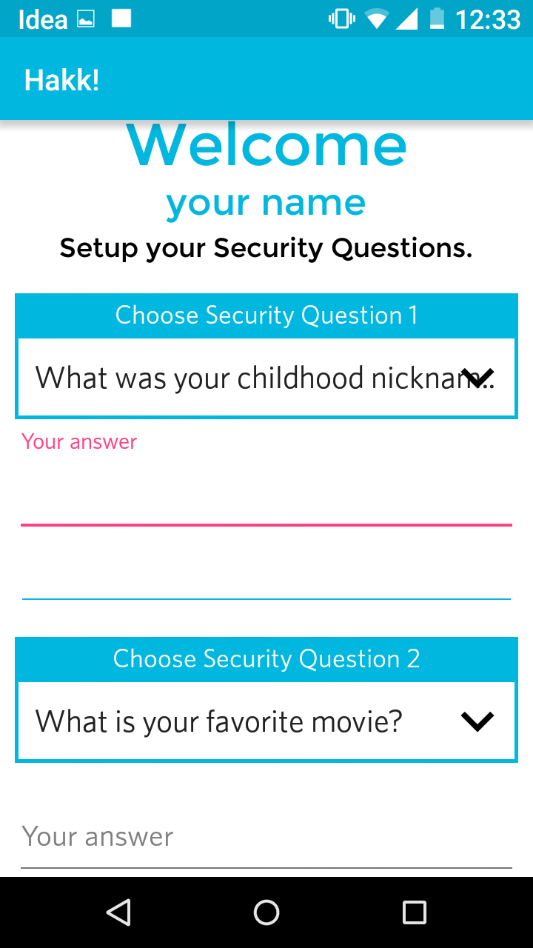
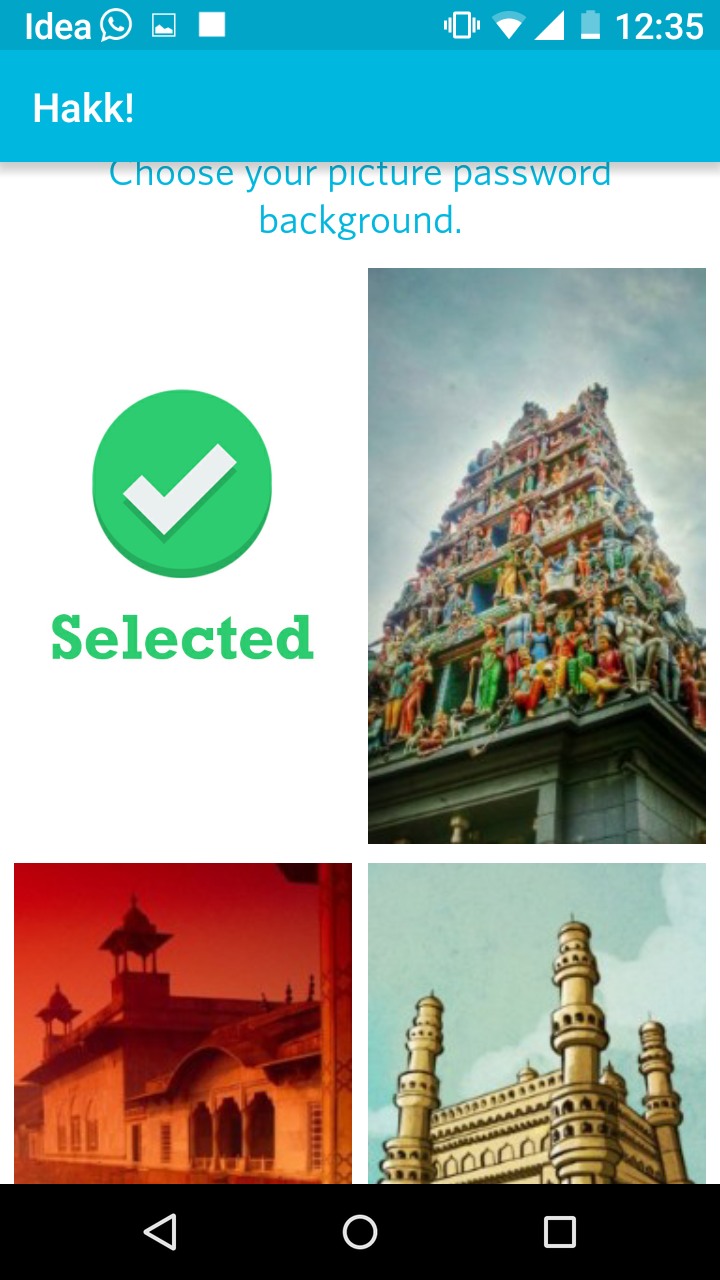
** **

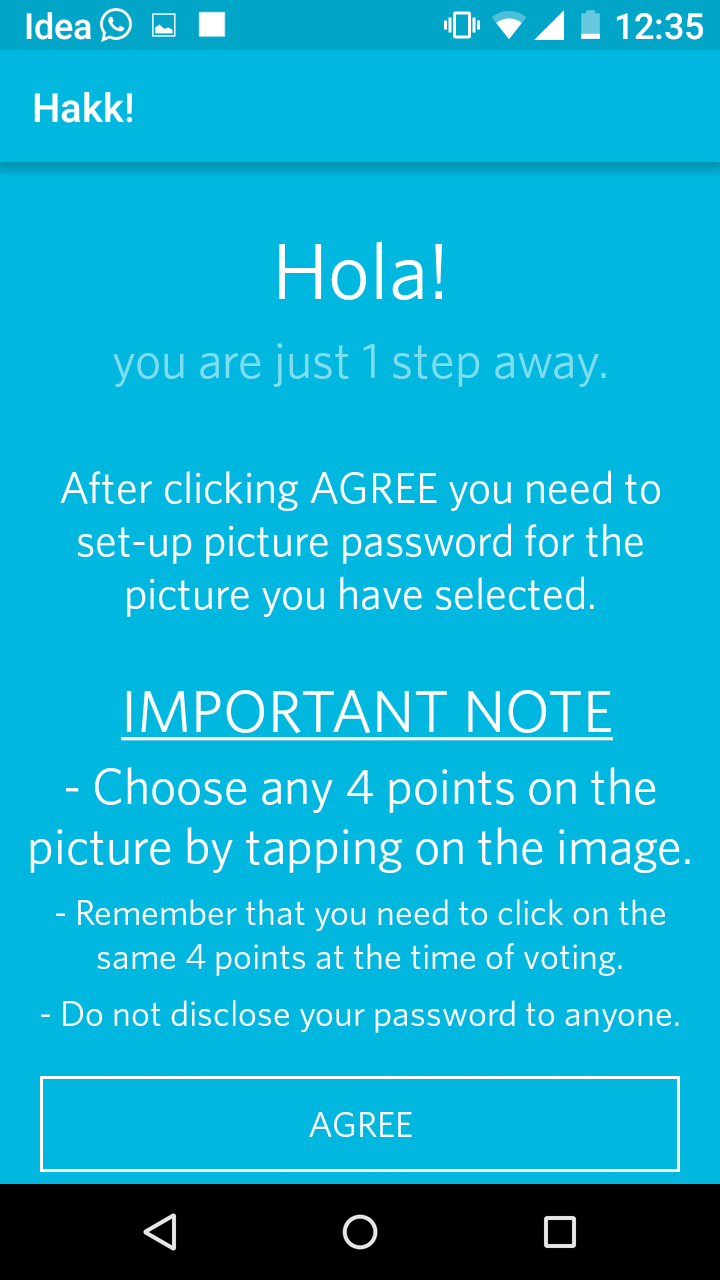
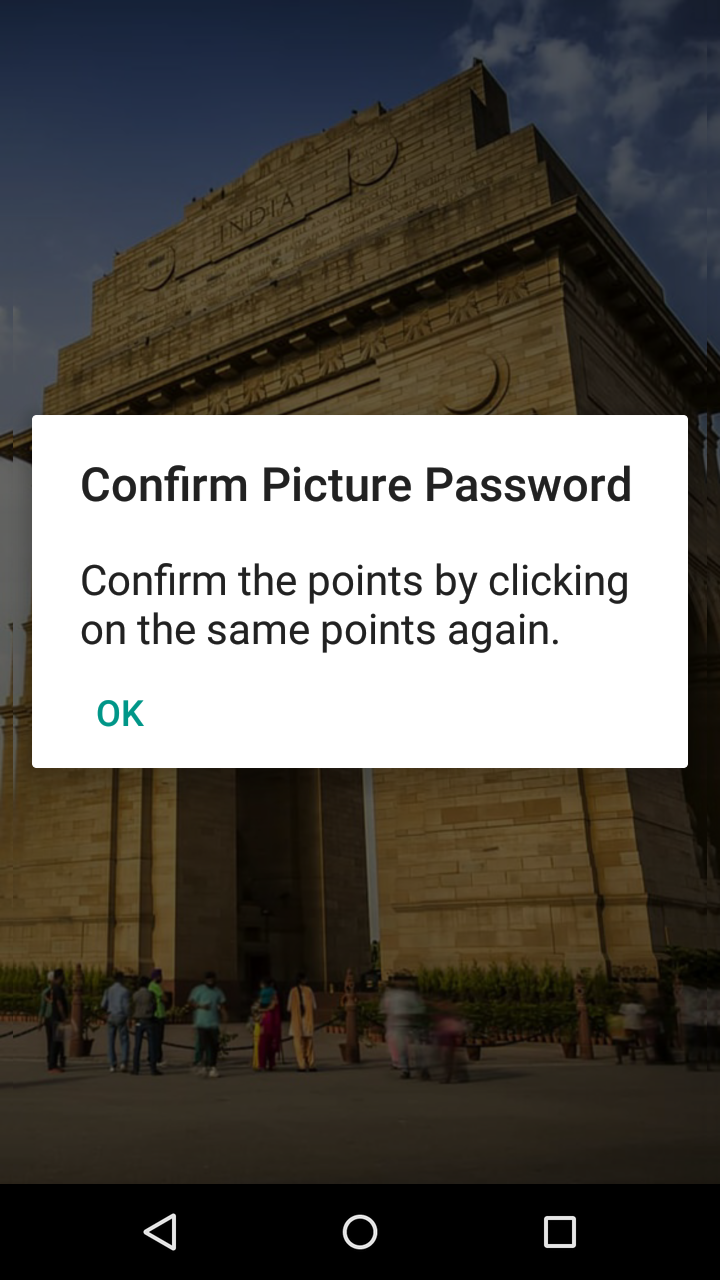
** **

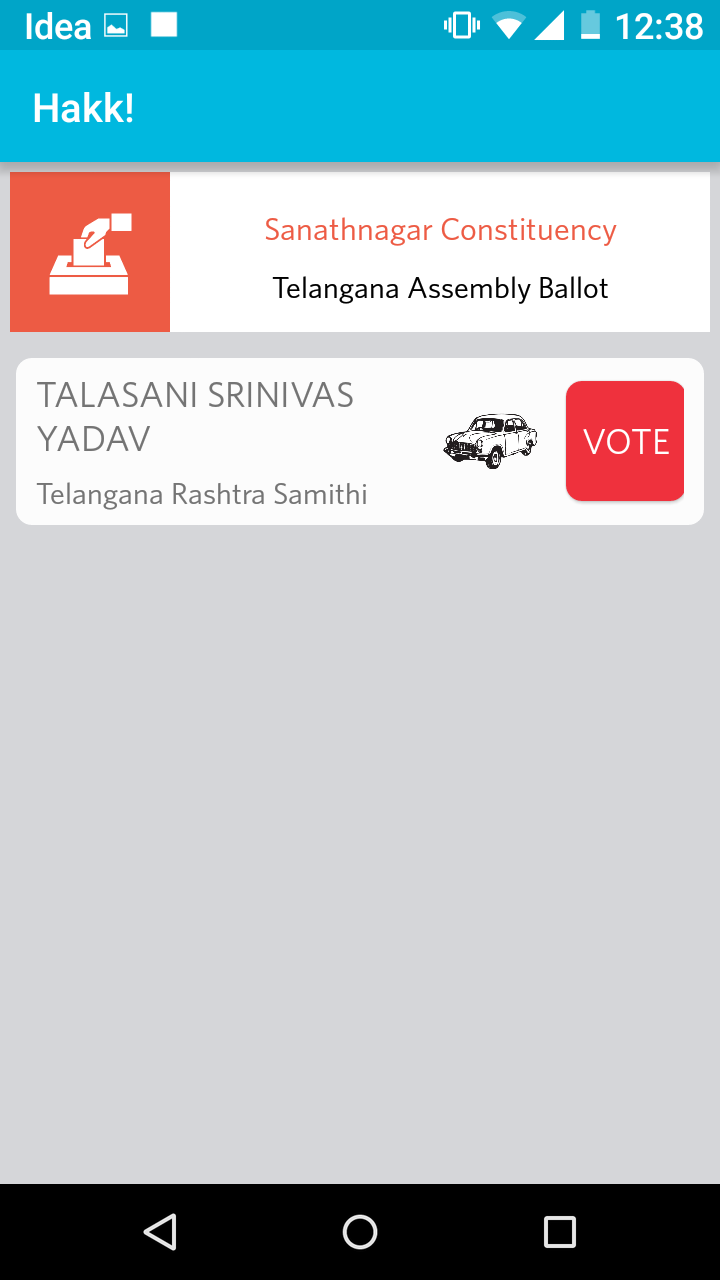
** **

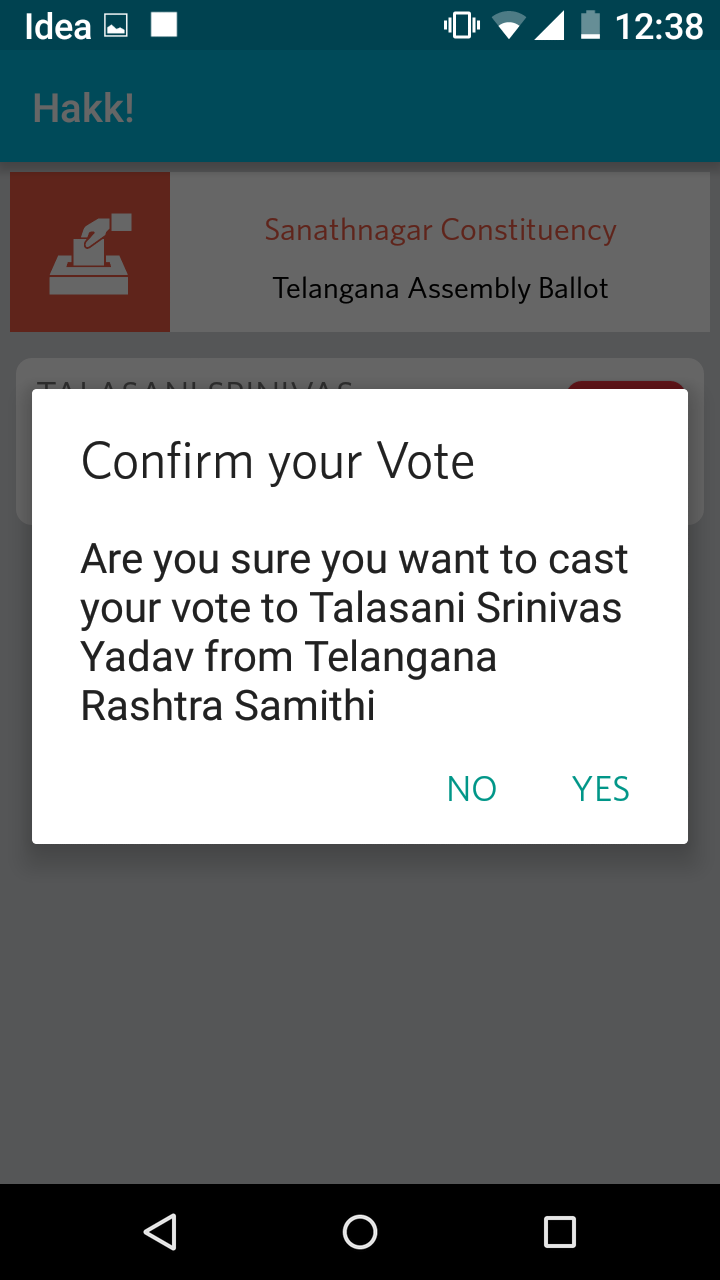
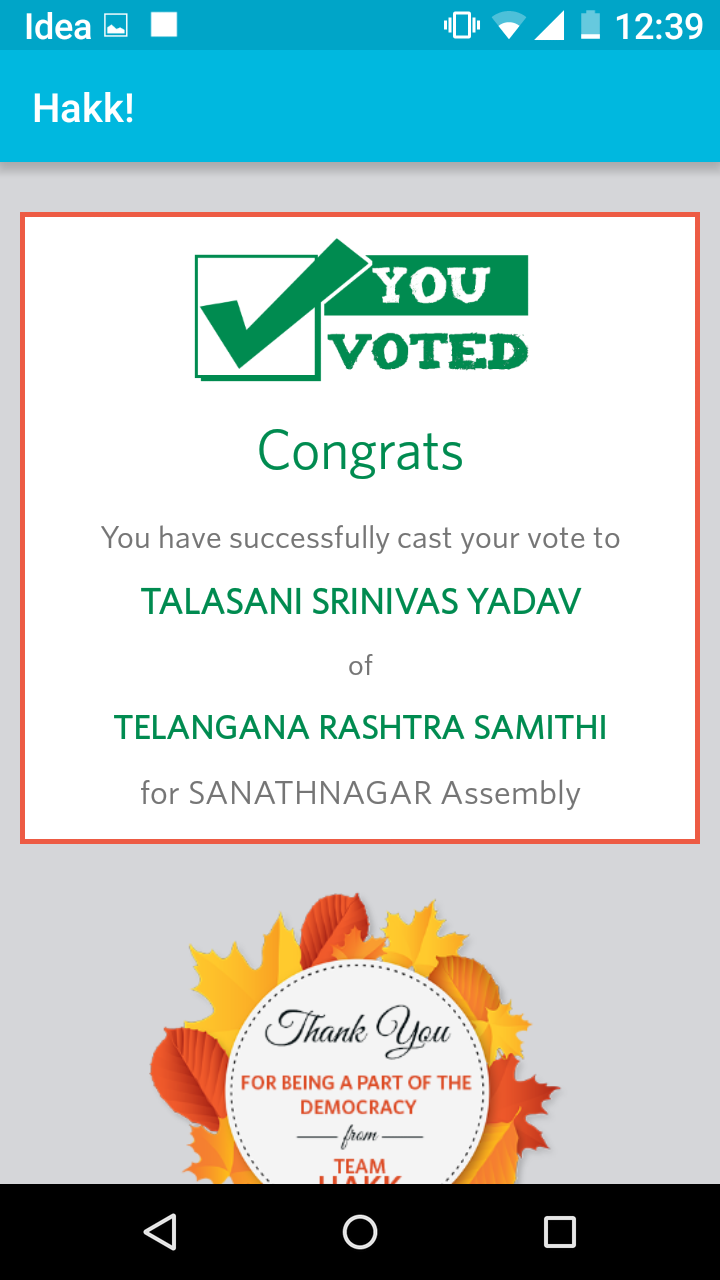
**Fig7.4: SIGN UP & ACTIVITIES:**

** **

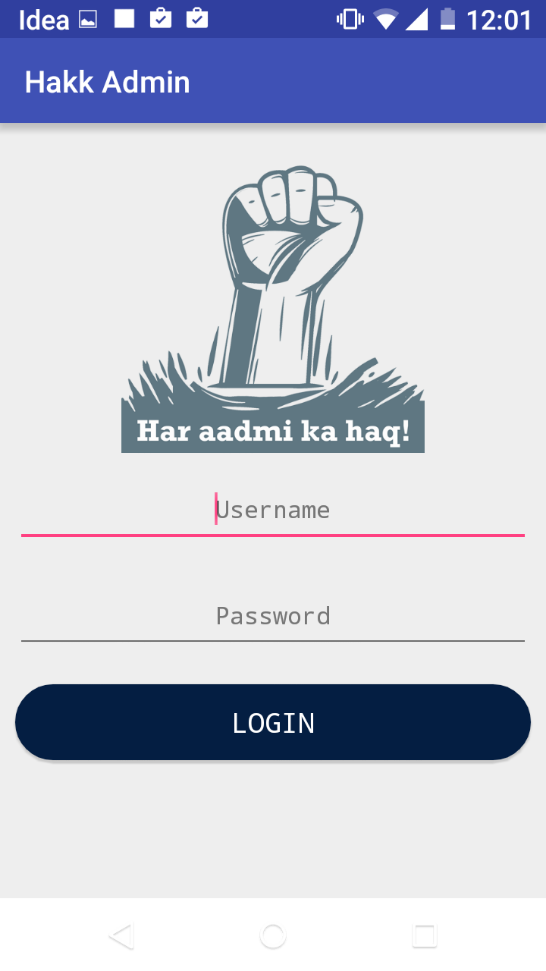
** **

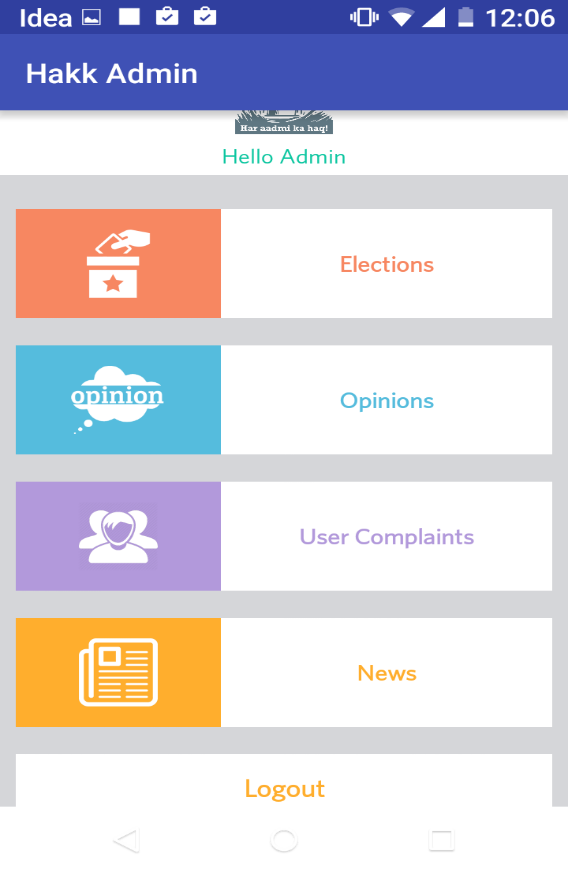
** **

** **

** **

**Fig7.5: HAKK ADMININSTRATION APP:**

****

****

**8. CONCLUSION & FUTURE ENHANCEMENTS**

In the current busy world the only way to increase the percentage of polling is to provide the facility for e-voting which would be very convenient for the user to cast his vote, but this system has to be very secures to avoid voter frauds. This application developed one such high security.

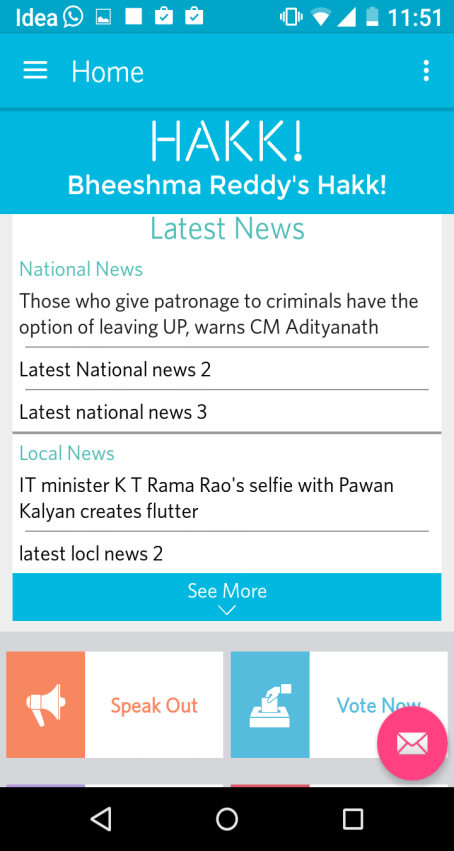
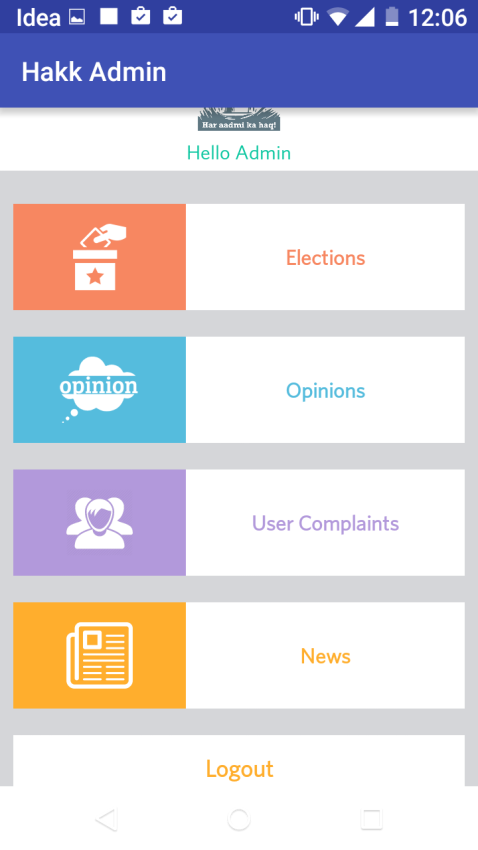
Also the people should be aware of the current issues and they should have the freedom to express their views on them, our project aims to fulfill that.

It is also the responsibility of the citizens to vote to the right candidate, they can judge the candidates only by knowing their bio-data's, which would be provided by Hakk!

Overall for any democracy people are the pillars, the democracy runs smooth freely only if its people have the privilege of enjoying their rights. Hakk! promises to provide them.

**9. APPENDIX**

**Home page: Admin home page:**

** **

**10. BIBLIOGRAPHY**

[1] Schneier, Bruce (September 2004), open Democracy What’s wrong with electronic voting machines?

[2] ‘Paper trail’ voting system used in Nevada, Associated PressSept. 7, 2004

[3] “How do we vote in India.” April 5th, 2009 , brainstorms.in/?p=309

[4] Padgett, Tim (2007-11-03)."Voting Out E-Voting Machines". TIME*.* Retrieved 2014-02-06.

[5] Grossman, Wendy M (30 April 2009)."Why machines are bad at counting votes". London: The Guardian*.* Retrieved 2009-07-14.