



## Voting System (offline) ACP Internship

Overview:-

This is app to vote. In this app there will be two Logion access: Admin and User.

Admin will check count for each party.

User can vote to any party present in the list.

According to the login credentials, system will determine if the user is Administrator or the normal User and open appropriate activity.

To store data regarding user and vote count we can use **SQLite DB or file storage**.

In case, if you use **SQLite DB**, then voting DB has to designed by the student himself.

When using **File storage, code to** create 2 files with names **credentials.txt** and **voteCount.txt**.

**credentials.txt** → This file contains loginid, adhar no and status of user in following format

a,111111111111,0, b,222222222222,0, c,333333333333,0

**voteCount.txt** → contains vote count for each party. Initially count will be 0 for each party and will be incremented gradually as various users vote.



This file contains data in following format assuming there are 4 parties.  
**0,0,0,0**

The minimum list of .java and .xml files required are:

1. **MainActivity.java => (main.xml)**

This activity would have the code for creating 2 files (credentials.txt & voteCount.txt ) and writing some data (as explained above) **only once**.

This activity will accept username and password, has a two buttons – **OK and Admin.**

When MainActivity.java is loaded, first of all **credentials.txt** is read. After that data from Text file is obtained into String which is separated by using Split function into String Array. When user clicks on **OK button** then username and password entered in edit text are compared with that of in **String Array**. If they are valid then further check status of that user from **String array**. If status=0 then **MainActivity.java** calls **Vote.java** If login credentials are not valid, show toast as “Not Valid User” and if status=1 then show toast as “You have already voted.”



**{Above Interfaces are very raw, Students are expected to creative in making better UI with validations}**



When Admin button gets clicked then call goes to **Admin.java**



**Vote.java =>vote.xml**

When Vote Activity is loaded, it reads **voteCount.txt**, **credentials.txt**.

Data from 2 text files is obtained into 2 Strings. Each String is separated by using Split function into separate String Array. When user select the candidate and click on save button then vote count of selected party is incremented by 1 in array and updated array is written in **voteCount.txt** file in same format mentioned above. After that **End.java** is called.

In Vote activity user can see the party Name, and give vote to the candidate by clicking on save button.

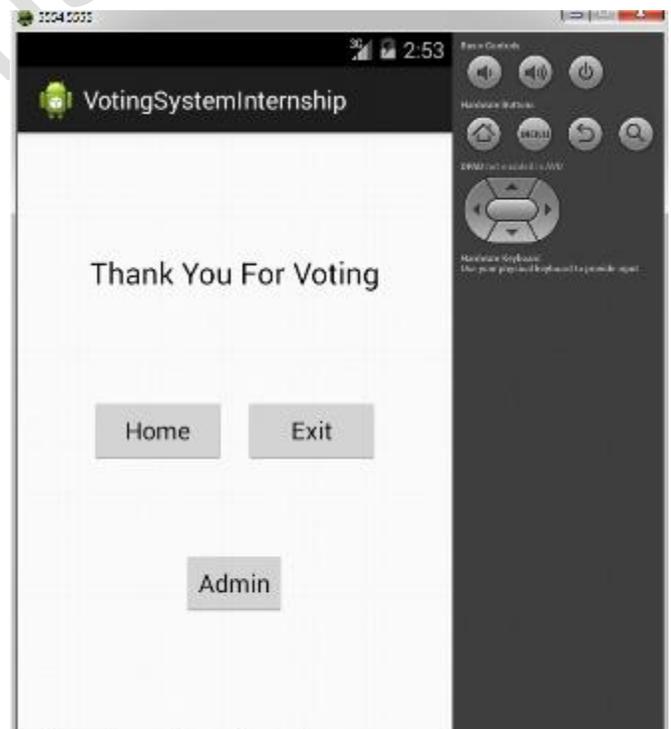


**End.java => end.xml**

When user clicks on **Home Button** then **MainActivity.java** is called.

When user clicks on **Exit Button** user exit from app and home screen is brought to front.

When user clicks on **Admin Button** then **Admin.java** is called.





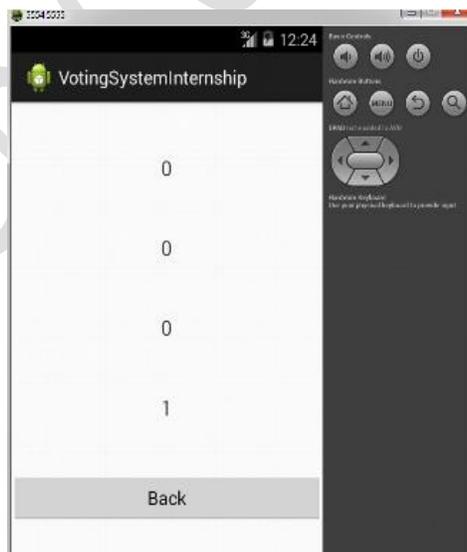
## Admin.java => admin.xml

Admin logs in by his login Id="admin" and password="1234567890" and clicks on Ok button. When OK is clicked call goes to Count.java

## Count.java=> (count.xml)

When Count Activity is loaded, first voteCount.txt is read. After that data from Text file is obtained into String which is separated by using Split function into String Array.

**Count.java** will fetch the count for each party from String Array, and will show in this activity as show in below.





## Sample code snippet for reading a file and split:

```
// ***** CODE TO READ FILE CONTENT (i.e. Login-Id, Adhar, Vote_Status) AS A
STRING *****
    try
    {
        FileInputStream fIn =openFileInput("login.txt");
        InputStreamReader isr = new InputStreamReader(fIn);
        char[] inputBuffer = new char[READ_BLOCK_SIZE];
        int charRead;

        while ((charRead = isr.read(inputBuffer))>0)
        {
            //---convert the chars to a String---
            String readString =String.valueOf(inputBuffer, 0,charRead);
            s += readString;
            inputBuffer = new char[READ_BLOCK_SIZE];
            System.out.print("s is "+s);
        }
        if(s!=null)
        {
            entryArray=s.split(","); //<---- ***** CODE TO PUT READ STRING INTO
ARRAY AFTER SPLITTING BY COMMA *****
        }
        else
        {
            Toast.makeText(getApplicationContext(),"File not loaded
successfully!",Toast.LENGTH_LONG).show();
        }
    }
    catch (IOException ioe)
    {
        ioe.printStackTrace();
    }

    admin.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            b.putString("category","Main");
            Intent i=new Intent(MainActivity.this,Admin.class);
            i.putExtras(b);
            startActivity(i);
            finish();
        }
    });
});
```



## Code snippet for verifying Login credentials

```
ok.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        login=loginEditText.getText().toString();
        adhar=adharEditText.getText().toString();

        if(login.equals("") || adhar.equals(""))
        {
            Toast.makeText(getApplicationContext(), "Enter Login-ID and Adhar No.",
Toast.LENGTH_LONG).show();
        }
        else
        {
            for(int i=0;i<entryArray.length;i=i+3)
            {
                if(adhar.equals(entryArray[i+1]) && login.equals( entryArray[i]))
                {
                    if(entryArray[i+2].equals("0")) {
                        b.putString("login", login);
                        b.putString("adhar", adhar);
                        Intent in=new Intent(MainActivity.this,Vote.class);
                        in.putExtras(b);
                        startActivity(in);
                        finish();
                        break;
                    }else
                    {
                        Toast.makeText(getApplicationContext(), "You have already VOTED",
Toast.LENGTH_LONG).show();
                        loginEditText.setText("");
                        adharEditText.setText("");
                        break;
                    }
                }
            }
            else
            {
                if((i==entryArray.length-3))
                {
                    Toast.makeText(getApplicationContext(), "You are not a valid User",
Toast.LENGTH_LONG).show();
                }
            }
        }
    }
});
```