Electronic Voting System

Software Design Document

Hania Arif (15-SE-03)

Department of Software
Engineering

University of Engineering
and Technology, Taxila

Date: August 21, 2017

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

This Software Design Specification (SDS) provides an overview of the proposed Online E-Voting system project design. It will encompass in detail the basic outline of our project and represent a basis for the development process. This will also allow critical analysis of the logical and functional aspects of the design before any commitment is made to actual code.

This Software Design Document (henceforth referred to as SDD) provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

The overall scope of the components in this SDD concerns the back-end infrastructure for the Electronic Voting System. The front-end components are being designed and implemented separately; however, we will specify the format and nature of the interfaces between the front end and the back end in this SDD.

The design document will show the interactions between the web services, between different forms used by both voters and administrators who are the main actors in the design.

SDS will be used by the project manager and the development team.

The main objective of this product is to simply make voting - during the elections - easy, secure and quick. This document describes the design and architecture of the system. The main aim of the system is to provide a set of protocols that allow voters to cast ballots while their votes are being saved and the results are being calculated at the same time and at the end the administrator of the system gets the finalized results and all the voters can also see the results.

1.3 Overview

This document identifies the system overview, design considerations and system architecture with components and user interface designs.

The guiding principle behind the design of this system is that the election system should be publicly verifiable: individual voters should be able to check that their votes have not been tampered with, and anyone should be able to verify the final election tally. Every step of the process should be verifiable, and should not require trust in any individual person or machine.

The rest of this document will specifically cover what would be needed to design an Electronic Voting System considering standards such as integrity, security, usability, anonymity, scalability, speed, audibility and accuracy.

1.4 Definitions and Acronyms

Ballot	A ballot is used by a voter to cast their vote. A ballot consists of an ordered series of candidates against which the voter may vote.
Admin	An administrator
Candidate	A person who seeks or who is nominated for an office in an election
Election	The selection of a person or persons for office by vote, or a public vote on a proposed submittal.
Voter	The one who votes, a member of the electorate, or a citizen with the right to vote
ECP	Election Commission of Pakistan – The organization that holds and conducts elections in Pakistan.
E-Voting System	Electronic Voting System
E-Voting Machine	Electronic Voting Machine
E-Voting Site	Electronic Voting System’s Sever Site
E-Voting Database	Electronic Voting System’s Database
HTTP	An application protocol for distributed, collaborative and hypermedia information systems
JVM	A Java virtual machine (JVM) is an abstract computing machine that enables a computer to run a Java program.

2. SYSTEM OVERVIEW

Give a general description of the functionality, context and design of your project. Provide any background information if necessary.

In Pakistan, usually manual voting takes place which is hectic and time consuming. This manual process of voting is also liable to errors. Electronic Voting System makes the voting process faster, easier and secure. The voter can cast his vote with confidentiality and his vote will be casted with least delay, disturbance or error. His vote would also be counted and saved automatically. By the use of this system the user can easily check online:

- ✓ His name in the voter list
- ✓ The candidate list of his area
- ✓ The status of his vote
- ✓ Results of the elections

Electronic voting refers to the use of computers or computerized voting equipment to cast ballots in an election. Sometimes, this term is used more specifically to refer to voting that takes place over the Internet. Electronic systems can be used to register voters, tally ballots, and record votes.

2.1 In / Out List

Features	**In**	**Out**
Votes and voting details will be stored by this system.	✓	
Election results will be calculated by this system.	✓	
Voter profiles will be stored in the database of this system.	✓	
This system will also have a backup server and a backup database so as to avoid crash.	✓	
Voters can vote easily using this system.	✓	
The election center authorizer can check the voting status from each voting machine in that center.	✓	
The user can check the voter list and the candidate list on the E-Voting Website.	✓	
User vote will be kept secure by this system.	✓	

Different system interfaces can be integrated at different places.	✓	
System is made more responsive by adding the backup server and the backup database.	✓	
The "see voter status" feature used by the authorizer makes the election process more transparent.	✓	
The voter list and the candidate list will be provided by ECP.		✓
The voters will get registered and verified by NADRA.		✓
The user cannot download the election results. He can only, view and check election results.		✓
Till now support for disabled people is not integrated in this system.		✓

2.2 Actor-Goal List

Actor	Goal
Administrator	Control the election process and get election results
Authorizer	Monitor the voting status of a voter while he is voting
Voter	Cast vote to candidate and see election results
Candidate	See election results
Database	Store election data
Server	To process election data and calculate election results

2.3 Use-Case Briefs

1. Vote

In elections, the voters vote the candidate of their choice. As a result of which in the overall results a single person gets selected by the majority's choice.

1a. Login

Before voting a candidate, the voter has to login to his voter account. For this purpose, the voter has to enter and verify his voter credentials and thus the following use-cases are used respectively.

i. Enter credentials
ii. Verify credentials

1b. Logout

After casting his vote, the voter has to logout of his account so that no one else can access his data, as his data in his voter account is very sensitive and confidential.

1c. Select language

For the ease of user, and usability of the system, the user is given the option to select a of his choice in which he might feel convenient to proceed the voting process.

1d. Select election

The user has to select an election for which he wants to cast his vote as there are two types of elections going on at the same time for which the user has to cast his vote. i.e. National and Provincial assembly elections. In this part, it is also ensured that a voter doesn't vote more than once for an election.

1e. View ballot paper

In order to vote the candidate, the voter has to request the system to show the ballot paper on the E-Voting Machine's screen so that he can cast his vote to the candidate of his choice.

1f. Select candidate

After viewing the ballot paper, the voter selects a candidate of his choice. In it, it is also endured that the voter doesn't select more than one candidate at a time as to meet the functional and the security requirements of the system.

1g. Submit vote

After selecting the candidate of his choice, the voter submits his vote so that his vote might be casted and counted.

1h. Get **voting** receipt

After submitting the vote, the voter gets a printed vote receipt, so that it could act as a proof that this person has casted his vote.

2. Monitor voting status

While the voter is casting his vote the authorizer i.e. the person present there- a member of the election committee, should be able to see the status of every polling booth so that it can be ensured that every vote is being casted properly and which of the polling booths are free so that they could be used by the other voters.

3. Proceed complaint

If the voter has got some problem during and before elections, he can submit his complaint to the Authorizer, who will then proceed the voter's complaint (if applicable or s per rules).

4. **Register complaint**

   If the voter has got some problem during and before elections, he can submit his complaint to the Authorizer, this complaint must be registered and stored so that there might be a record of it.

5. **See voter list**

   Before every election, the voter must check his name in the voter list for which this use case is used.

6. **See candidate list**

   Before every election, the voter must check his name in the voter list for which this use case is used.

7. **Process vote**

   When the user has casted his vote the server has to process that vote i.e. in a sense that it has to add the vote to the proper candidate’s vote graph and save the vote in the database.

8. **Calculate election results**

   When the time of voting has ended, the server calculates the overall election result and the election results of every sector and sends them to the administrator and the authorizer so that they can announce the election results publicly.

9. **See time left to vote**

   As a reminder, every voter should be able to see that how much time is left, so that he can cast his vote before that voting time ends.

10. **Send notification**

    When some voter submits a complaint, a new election is created or the voting timer starts, a notification is sent to every user as a reminder.

11. **Make complaint**

    If the voter has got some problem during and before elections, he can submit his complaint to the Authorizer which will be proceeded according to the rules.

3. FUNCTIONAL REQUIREMENTS

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 the sense that it allows a variety of ballot question formats including open-ended questions (e.g. Write-in candidates and survey questions).
6. **Support for Disabled Voters:**
   The system shall cater to the needs of physically challenged voters (e.g. blind voters).
7. **Accuracy:**
   The system shall record and count all the votes and shall do so correctly.
8. **Eligibility:**
   Only authorized voters, who are registered, should be able to vote.
9. **Uniqueness:**
   No voter should be able to vote more than once.
10. **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).
11. **Voter Confirmation:**
    The voter shall be able to confirm clearly how his vote is being cast, and shall be given a chance to modify his vote before he commits it.
12. **To issue Receipt or not?**
    The system may issue a receipt to the voter if and only if it can be ensured that vote-coercion and vote-selling are prevented, so that he may verify his vote at any time and also contend, if necessary.
13. **No Over-voting:**

The voter shall be prevented from choosing more than one candidate / answer.

14. **Under-voting:**
    The voter may receive a warning of not voting, but the system must not prevent under-voting.
15. **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.
16. **Documentation and Assurance:**
    The design, implementation, and testing procedures must be well-documented so that the voter-confidence in the election process is ensured.
17. **Cost-effectiveness:**
    Election systems should be affordable and efficient.

4. NON-FUNCTIONAL REQUIREMENTS

4.1 Security Requirements

1. **Voter Authenticity:**
   Ensure that the voter must identify himself (with respect to the registration database) to be entitled to vote. If voting other than at his home precinct, the voter may be asked to show some legal identification document.
2. **Registration:**
   The voter registration shall be done in person only. However, the computerized registration database shall be made available to polling-booths all around the nation.
3. **Voter Anonymity:**
   Ensure that votes must not be associated with voter identity.
4. **System Integrity:**
   Ensure that the system cannot be re-configured during operation.
5. **Data Integrity:**
   Ensure that each vote is recorded as intended and cannot be tampered with in any manner, once recorded (i.e., votes should not be modified, forged or deleted without detection).
6. **Secrecy / Privacy:**
   No one should be able to determine how any individual voted.
7. **Non- coercibility and no Vote-selling:**
   Voters should not be able to prove to others how they voted (which would facilitate vote selling or coercion).
8. **Reliability:**

Election systems should work robustly, without loss of any votes, even in the face of numerous failures, including failures of voting machines and total loss of network communication. The system shall be developed in a manner that ensures there is no malicious code or bugs.

9. **Availability:**
   Ensure that system is protected against accidental and malicious denial of service attacks. Also, setup redundant communication paths so that availability is ensured.
10. **System Disclosure:**
    The core of the system, especially the vote-casting equipment, shall be opensource, so that it can allow external inspection and auditing.
11. **Simplicity:**
    The system shall be designed to be extremely simple, as complexity is the enemy of security.
12. **Testing and Certification:**
    The system should be tested by experts with respect to all of the security considerations, so that election officials have the confidence that the system meets the necessary criteria.
13. **System Accountability:**
    Ensure that system operations are logged and audited.
14. **Personnel Integrity:**
    Those developing and operating the voting system should have unquestionable records of behavior.
15. **Operator Authentication and Control:**
    Ensure that those operating and administering the system are authenticated and have strictly controlled functional access on the system.
16. **Distribution of Authority:**
    The administrative authority shall not rest with a single entity. The authority shall be distributed among multiple administrators, who are known not to collude among themselves (e.g., different political parties).

5. SYSTEM USE-CASE VIEW

Referred to the Use Case Diagrams, Sequence Diagrams, Activity Diagrams and State Diagrams made according to the system – See Appendix A.

5.1.1 Vote

Use Case ID:	**1**

Use Case Name:	**Vote**

Primary Actors:	User
Secondary Actors:	Database, Server
Description:	This use case describes how the User votes a candidate.
Trigger:	The voting time has started and the User is at the polling booth to vote a candidate.
Preconditions:	Actor Network is idle and active.
Postconditions:	User has voted to the candidate of his choice.
Normal Flow:	User selects the language in which he wants to proceed. User enters his User ID and password. User submits confirmation to proceed. User selects the election he wants to vote for. User proceeds to see the ballot paper. User checks the candidate list and selects the candidate he wants to vote. User submits his vote. User gets a vote-receipt. User Logs Out of his User account.
Alternative Flows:	User enters invalid credentials. The system cancels proceeding. System gives an error message to the User. User re-enters his correct credentials. The log in ID was invalid. The User enters correct Log in ID. The log in password was invalid. The User enters correct password. User continues from step 3.
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	Log In Enter Credentials Verify Credentials Log Out

	Select Language Select Election View Ballot Paper Select Candidate Get Voting Receipt Submit Vote
Priority:	High
Frequency of Use:	High
Special Requirements:	Any user who has a User ID and is eligible to vote can assess the log in page and vote for the candidate of his choice.
Technology and Data Variants:	The electronic voting machine used might be: Direct Recording Electronic (DRE) System, Electronic Ballot Printers (EBPs) or etc. and different servers may be connected to different servers.
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.1.1.a Select Language

Use Case ID:	**1.c**
Use Case Name:	**Select language**

Primary Actors:	User
Secondary Actors:	Database, Server
Description:	This use case describes how the User selects a language of his choice.
Trigger:	The User has logged into his account and he wants to proceed in a language which is convenient to use for him.
Preconditions:	Actor Network is idle and active.
Postconditions:	User has selected the language of his choice.
Normal Flow:	User selects the language in which he wants to proceed.

Alternative Flows:	None
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	None
Technology and Data Variants:	The electronic voting machine used might be: Direct Recording Electronic (DRE) System, Electronic Ballot Printers (EBPs) or etc. and different servers may be connected to different servers.
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.1.1.b Log In

Use Case ID:	**1.b**
Use Case Name:	**Log In**

Primary Actors:	User
Secondary Actors:	Database, Server
Description:	This use case describes how the user logs in into his account.
Trigger:	The User or candidate wants to vote or the authorizer or administrator want to log in to their account to perform some activity.
Preconditions:	Actor Network is idle and active.
Postconditions:	The user has logged in to his account.

Normal Flow:	Voter selects the language in which he wants to proceed. Voter enters his voter ID and password. Voter submits confirmation to proceed.
Alternative Flows:	User enters invalid credentials. The system cancels proceeding. System gives an error message to the User. User re-enters his correct credentials. The log in ID was invalid. The User enters correct Log in ID. The log in password was invalid. The User enters correct password. User continues from step 3.
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	Enter Credentials Verify Credentials
Priority:	High
Frequency of Use:	High
Special Requirements:	Any user who has a user ID can log in to his user account.
Technology and Data Variants:	The electronic voting machine used might be: Direct Recording Electronic (DRE) System, Electronic Ballot Printers (EBPs) or etc. and different servers may be connected to different servers.
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.1.1.b.i Enter Credentials

Use Case ID:	**1.b.i**
Use Case Name:	**Enter Credentials**

Primary Actors:	User
Secondary Actors:	Database, Server
Description:	This use case describes how the user enters his credentials to log into his User account.
Trigger:	The User or candidate wants to vote or the authorizer or administrator want to log in to their account to perform some activity.
Preconditions:	Actor Network is idle and active.
Postconditions:	The user has entered his credentials.
Normal Flow:	Voter selects the language in which he wants to proceed. Voter enters his voter ID and password.
Alternative Flows:	None
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	Any user who has a user ID can log in to his user account.
Technology and Data Variants:	The electronic voting machine used might be: Direct Recording Electronic (DRE) System, Electronic Ballot Printers (EBPs) or etc. and different servers may be connected to different servers.
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.1.1.b.ii Verify credentials

Use Case ID:	**1.b.ii**
Use Case Name:	**Verify Credentials**

Primary Actors:	User
Secondary Actors:	Database, Server
Description:	This use case describes how the user verifies his credentials.
Trigger:	The User or candidate wants to vote or the authorizer or administrator want to log in to their account to perform some activity.
Preconditions:	Actor Network is idle and active.
Postconditions:	The user has logged in to his account.
Normal Flow:	Voter selects the language in which he wants to proceed. Voter enters his voter ID and password. Voter submits confirmation to proceed.
Alternative Flows:	User enters invalid credentials. The system cancels proceeding. System gives an error message to the User. User re-enters his correct credentials. The log in ID was invalid. The User enters correct Log in ID. The log in password was invalid. The User enters correct password. User continues from step 3.
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	Any user who has a user ID can log in to his user account.
Technology and Data Variants:	The electronic voting machine used might be: Direct Recording Electronic (DRE) System, Electronic Ballot Printers (EBPs) or etc. and different servers may be connected to different servers.
Open Issues	None

Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.1.1.c Select Election

Use Case ID:	**1.c**
Use Case Name:	**Select Election**

Primary Actors:	User
Secondary Actors:	Database, Server
Description:	This use case describes how the User selects an election type.
Trigger:	The User wants to cast his vote for an election.
Preconditions:	Actor Network is idle and active.
Postconditions:	User has selected an election for which he wants to cast his vote.
Normal Flow:	User selects the language in which he wants to proceed. User enters his User ID and password. User submits confirmation to proceed. User selects the election he wants to vote for.
Alternative Flows:	User enters invalid credentials. The system cancels proceeding. System gives an error message to the User. User re-enters his correct credentials. The log in ID was invalid. The User enters correct Log in ID. The log in password was invalid. The User enters correct password. User continues from step 3.
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High

Frequency of Use:	High
Special Requirements:	Any user who has a User ID and is eligible to vote can assess the log in page and select an election to cast his vote.
Technology and Data Variants:	The electronic voting machine used might be: Direct Recording Electronic (DRE) System, Electronic Ballot Printers (EBPs) or etc. and different servers may be connected to different servers.
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.1.1.d View Ballot Paper

Use Case ID:	**1.d**
Use Case Name:	**View Ballot Paper**

Primary Actors:	User
Secondary Actors:	Database, Server
Description:	This use case describes how the User sees the ballot paper of a specific election type.
Trigger:	The User has selected the election for which he wants to cast his vote now he wants to see the ballot paper so that he can cast his vote for a candidate of his choice.
Preconditions:	Actor Network is idle and active.
Postconditions:	User has viewed the ballot paper of the specific election type.
Normal Flow:	User selects the language in which he wants to proceed. User enters his User ID and password. User submits confirmation to proceed. User selects the election he wants to vote for. User proceeds to see the ballot paper.
Alternative Flows:	User enters invalid credentials.

	The system cancels proceeding. System gives an error message to the User. User re-enters his correct credentials. The log in ID was invalid. The User enters correct Log in ID. The log in password was invalid. The User enters correct password. User continues from step 3.
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	Any user who has logged in to his User account and has selected the election type, can see the ballot paper of that election.
Technology and Data Variants:	The electronic voting machine used might be: Direct Recording Electronic (DRE) System, Electronic Ballot Printers (EBPs) or etc. and different servers may be connected to different servers.
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.1.1.e Select Candidate

Use Case ID:	**1.e**
Use Case Name:	**Select Candidate**

Primary Actors:	User
Secondary Actors:	Database, Server
Description:	This use case describes how the User selects a candidate of his choice.

Trigger:	The User has seen the ballot paper now he has to select a candidate of his choice for whom he wants to vote for.
Preconditions:	Actor Network is idle and active.
Postconditions:	User has selected the candidate of his choice for whom he wants to vote for.
Normal Flow:	User selects the language in which he wants to proceed. User enters his User ID and password. User submits confirmation to proceed. User selects the election he wants to vote for. User proceeds to see the ballot paper. User checks the candidate list and selects the candidate he wants to vote.
Alternative Flows:	User enters invalid credentials. The system cancels proceeding. System gives an error message to the User. User re-enters his correct credentials. The log in ID was invalid. The User enters correct Log in ID. The log in password was invalid. The User enters correct password. User continues from step 3.
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	Any user who has a User ID and has the logged in to his User account can select the candidate of his choice after selecting the election type.
Technology and Data Variants:	The electronic voting machine used might be: Direct Recording Electronic (DRE) System, Electronic Ballot Printers (EBPs) or etc. and different servers may be connected to different servers.
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.

Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.1.1.f Submit Vote

Use Case ID:	**1.f**
Use Case Name:	**Submit Vote**

Primary Actors:	User
Secondary Actors:	Database, Server
Description:	This use case describes how the User submits his vote.
Trigger:	The User has selected the candidate of his choice and now he wants to submit his vote.
Preconditions:	Actor Network is idle and active.
Postconditions:	User has submitted his vote.
Normal Flow:	User selects the language in which he wants to proceed. User enters his User ID and password. User submits confirmation to proceed. User selects the election he wants to vote for. User proceeds to see the ballot paper. User checks the candidate list and selects the candidate he wants to vote. User submits his vote.
Alternative Flows:	User enters invalid credentials. The system cancels proceeding. System gives an error message to the User. User re-enters his correct credentials. The log in ID was invalid. The User enters correct Log in ID. The log in password was invalid. The User enters correct password. User continues from step 3.

Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	Any user who has a User ID, has logged in to his User account, has selected the election type and a candidate for whom he wants to cast his vote can submit his vote.
Technology and Data Variants:	The electronic voting machine used might be: Direct Recording Electronic (DRE) System, Electronic Ballot Printers (EBPs) or etc. and different servers may be connected to different servers.
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.1.1.g Get Voting Receipt

Use Case ID:	**1.g**
Use Case Name:	**Getting Voting Receipt**

Primary Actors:	User
Secondary Actors:	Database, Server
Description:	This use case describes how the User gets the voting receipt.
Trigger:	The User has casted his vote now he wants to get the voting receipt as a record that he has casted his vote.
Preconditions:	Actor Network is idle and active.
Postconditions:	User has got the printed voting receipt.
Normal Flow:	User selects the language in which he wants to proceed. User enters his User ID and password. User submits confirmation to proceed.

	User selects the election he wants to vote for. User proceeds to see the ballot paper. User checks the candidate list and selects the candidate he wants to vote. User submits his vote. User gets a vote-receipt.
Alternative Flows:	User enters invalid credentials. The system cancels proceeding. System gives an error message to the User. User re-enters his correct credentials. The log in ID was invalid. The User enters correct Log in ID. The log in password was invalid. The User enters correct password. User continues from step 3.
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	Any user who has a User who has casted his vote gets the voting receipt.
Technology and Data Variants:	The electronic voting machine used might be: Direct Recording Electronic (DRE) System, Electronic Ballot Printers (EBPs) or etc. and different servers may be connected to different servers.
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.1.1.h Log Out

Use Case ID:	**1.h**

Use Case Name:	**Log Out**

Primary Actors:	User
Secondary Actors:	Database, Server
Description:	This use case describes how the user logs out of his User account.
Trigger:	The user has performed the activity he wanted to perform and now he wants to log out of his account so that no one else can use his account data as it is confidential.
Preconditions:	Actor Network is idle and active.
Postconditions:	User has logged out of his account.
Normal Flow:	User selects the language in which he wants to proceed. User enters his User ID and password. User submits confirmation to proceed. User has done the activity he wanted to do. User Logs Out of his User account.
Alternative Flows:	None
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	Any user who has a User ID and is eligible to vote can assess the log in page and vote for the candidate of his choice.
Technology and Data Variants:	The electronic voting machine used might be: Direct Recording Electronic (DRE) System, Electronic Ballot Printers (EBPs) or etc. and different servers may be connected to different servers.
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.2 Monitor Voting Status

Use Case ID:	**2**
Use Case Name:	**Monitor Voting Status**

Primary Actors:	Authorizer
Secondary Actors:	E-Voting Machine, Server
Description:	This use case describes how the Authorizer monitors the voting status of every E-Voting Machine.
Trigger:	The voting time has started, the User is at the polling booth to vote a candidate and the Authorizer wants to monitor its voting status.
Preconditions:	Actor Network is idle and active.
Postconditions:	The Authorizer is able to monitor the voting status of every polling booth in that voting station.
Normal Flow:	Voter starts to vote. Server collects the status of voting on that E-Voting machine. The authorizer has logged in to his account. The servers complies the result of each status of each E-Voting Machine. The server sends the status to the Authorizer's device. The Authorizer monitors the voting status of each E-Voting Machine. The Server keeps updating the E-Voting Machine status and sending it to the Authorizer's device continuously.
Alternative Flows:	None
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	Only the authorizer can view the status of all the E-Voting Machines of that Voting Center on his synchronized device.

Technology and Data Variants:	The electronic voting machine used might be: Direct Recording Electronic (DRE) System, Electronic Ballot Printers (EBPs) or etc. and different servers may be connected to different servers.
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.3 Proceed Complaint

Use Case ID:	**3**
Use Case Name:	**Proceed Complaint**

Primary Actors:	Authorizer
Secondary Actors:	Administrator, Server, Database
Description:	This use case describes how the Authorizer proceed the Voter complaint.
Trigger:	The voter has registered a complaint.
Preconditions:	Actor Network is idle and active.
Postconditions:	The authorizer has proceeded the complaint.
Normal Flow:	Voter registers a complaint. Authorizer writes the complaint. Voter verifies the complaint written by the Authorizer. Authorizer submits the complaint. Voter gets a printout of his complaint. Server saves the complaint in database. Server forwards the complaint to Administrator. Server notifies the Authorizer, Voter and the Administrator.
Alternative Flows:	Voter finds that something wrong in the written complaint Voter tells it to the Authorizer Authorizer corrects the mistake. Process is continues from step 3.

Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	Only Authorizer can make the complaint proceed.
Technology and Data Variants:	None
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.4 Register Complaint

Use Case ID:	**4**
Use Case Name:	**Register Complaint**

Primary Actors:	Authorizer
Secondary Actors:	Administrator, Server, Database
Description:	This use case describes how the Authorizer registers the Voter complaint.
Trigger:	The voter has to register a complaint.
Preconditions:	Actor Network is idle and active.
Postconditions:	The authorizer has proceeded the complaint.
Normal Flow:	Voter comes to the Voting Center Authorizer Voter tells the authorizer about his complaint. Authorizer sees the complaint if to be resolved on spot, otherwise Authorizer writes the complaint in his device. Voter validates the complaint written by the Authorizer.

	Authorizer submits the complaint.
Alternative Flows:	Voter finds that something wrong in the written complaint Voter tells it to the Authorizer Authorizer corrects the mistake. Process is continues from step 4.
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	Only Authorizer can register the complaint.
Technology and Data Variants:	None
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.5 See Voter List or See Candidate List

Use Case ID:	**5**
Use Case Name:	**See Voter List**

Primary Actors:	User
Secondary Actors:	Server, Database
Description:	This use case describes how the User sees a Voter list.
Trigger:	The voter has to check his name in the voter list.
Preconditions:	Actor Network is idle and active.
Postconditions:	User has checked the voter list.

Normal Flow:	User opens the E-Voting website. User selects the see list option. User selects which list he wants to see. Server process the user request. Server opens the list demanded by the user. User checks the list he demanded.
Alternative Flows:	User selects a wrong option User presses the back button. User continues from step 2.
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	None
Technology and Data Variants:	None
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.6 Process Vote

Use Case ID:	**6**
Use Case Name:	**Process Vote**

Primary Actors:	Server
Secondary Actors:	Database
Description:	This use case describes how the Server processes the voter’s vote.

Trigger:	The voter has casted his vote.
Preconditions:	Actor Network is idle and active.
Postconditions:	The server has stored the vote in the database.
Normal Flow:	Sever verifies the vote. Server adds one to the votes of the candidate, for whom the vote has been casted. Server sends the vote details to the database for storing. Database stores the vote details.
Alternative Flows:	None
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	None
Technology and Data Variants:	None
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.7 Calculate Election Results

Use Case ID:	**7**
Use Case Name:	**Calculate Election Results**

Primary Actors:	Server
Secondary Actors:	Database

Description:	This use case describes how the Server calculates the election results.
Trigger:	Voting time has ended and the Administrator has requested to see election results.
Preconditions:	Actor Network is idle and active.
Postconditions:	The server has calculated the election results
Normal Flow:	Sever gets the vote details from the database. Server verifies the data it has, with the processed votes' record. Server calculates the election results of each sector. Server calculates the compiled results. Server makes graphs according to the calculated results. Server forms the compiled results in user readable form.
Alternative Flows:	None
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	None
Technology and Data Variants:	None
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.8 See Election Results

Use Case ID:	**8**
Use Case Name:	**See Election Results**

Primary Actors:	User
Secondary Actors:	Database, Server
Description:	This use case describes how the User sees the election results.
Trigger:	The voting time has ended and the server has calculated the election results.
Preconditions:	Actor Network is idle and active.
Postconditions:	User has checked the election results.
Normal Flow:	User requests to see the election results. The server checks the election result file. Server makes the E-Voting website show the election results file. User checks the election results.
Alternative Flows:	None
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	None
Technology and Data Variants:	None
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.9 See Time Left to Vote

Use Case ID:	**9**
Use Case Name:	**See Time-Left to Vote**

Primary Actors:	User
Secondary Actors:	Server
Description:	This use case describes how the user sees the time left to vote.
Trigger:	The user wants to see the time left to vote.
Preconditions:	Actor Network is idle and active.
Postconditions:	The server has stored the vote in the database.
Normal Flow:	The user requests to see the time left to vote Server calculates the time left to vote Server makes the E-Voting site to display the time left to vote. User sees the time left to vote.
Alternative Flows:	None
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	None
Technology and Data Variants:	None
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.10 Send Notification

Use Case ID:	**10**
Use Case Name:	**Send Notification**

Primary Actors:	Server
Secondary Actors:	None
Description:	This use case describes how the User sees a Voter list.
Trigger:	The Authorizer has proceeded a complaint.
Preconditions:	Actor Network is idle and active.
Postconditions:	The server has sent notification to the Administrator, Voter (who made the complaint) and the Authorizer.
Normal Flow:	Voter registers a complaint. Authorizer submits the complaint. Voter gets a printout of his complaint. Server saves the complaint in database. Server forwards the complaint to Administrator. Server notifies the Authorizer, Voter and the Administrator.
Alternative Flows:	None
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	None
Technology and Data Variants:	None
Open Issues	None
Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

5.11 Make Complaint

Use Case ID:	**11**
Use Case Name:	**Make Complaint**

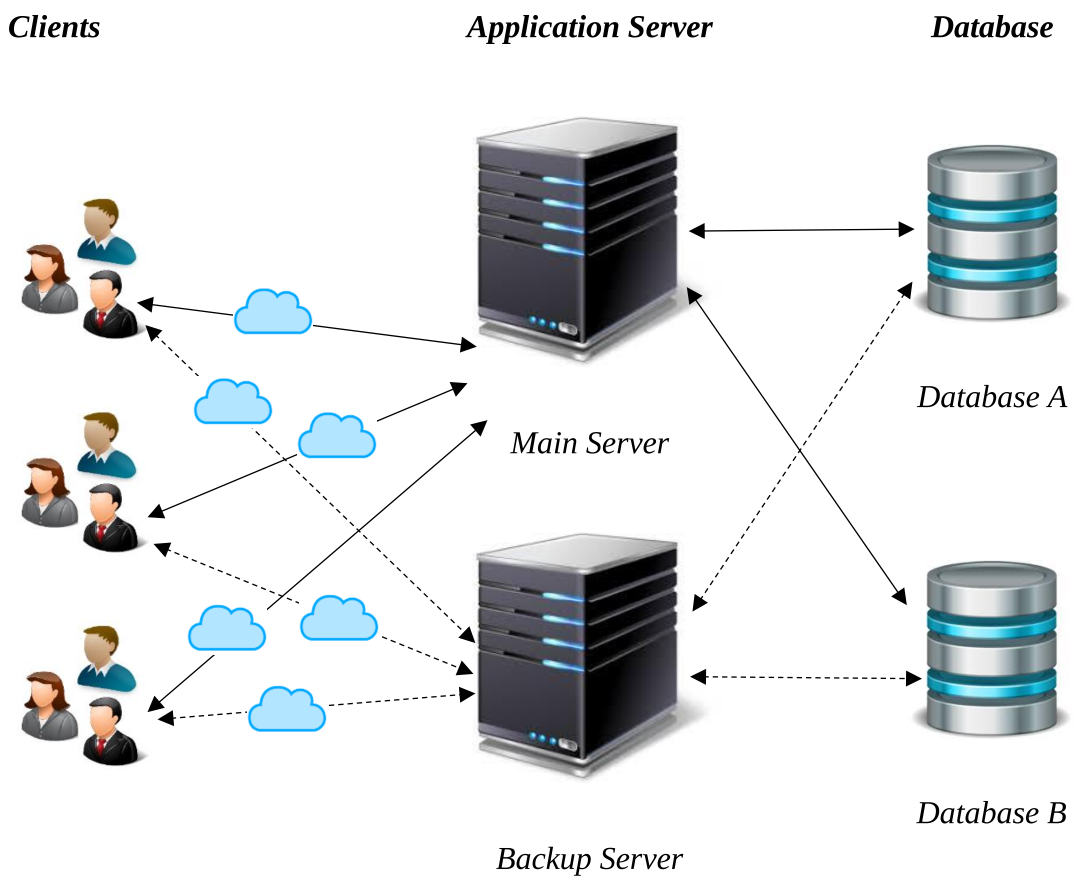
Primary Actors:	Voter
Secondary Actors:	Authorizer, Server, Database
Description:	This use case describes how the Voter makes a complaint.
Trigger:	The voter has to register a complaint.
Preconditions:	Actor Network is idle and active.
Postconditions:	The authorizer has proceeded the complaint.
Normal Flow:	Voter comes to the Voting Center Authorizer Voter tells the authorizer about his complaint. Authorizer sees the complaint if to be resolved on spot, otherwise Authorizer writes the complaint in his device. Voter validates the complaint written by the Authorizer. Authorizer submits the complaint.
Alternative Flows:	Voter finds that something wrong in the written complaint Voter tells it to the Authorizer Authorizer corrects the mistake. Process is continues from step 4.
Exceptions:	When there is a database connectivity problem report the problem via Error Messaging Technique.
Includes:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	Only Authorizer can register the complaint.
Technology and Data Variants:	None
Open Issues	None

Assumptions:	Server using e-voting has a network and internet connection.
Notes and Issues:	Any internet connection or network connection issue will cause access problems.

6. SYSTEM ARCHITECTURE

6.1 Architectural Design

Basically, our system is a 3-tier Client/Server architecture which comprises of two databases, two Application servers and PC stations. The additional application server presented in dotted lines in the diagram above acts as a backup to the working main server. Therefore, during critical operations, in case of failure the reserve server comes into operation. The two databases work together during critical operations of polling votes. However, the backup server is responsible for storing critical information like votes and results of election. In the front phase of the system architecture lies the clients. The client represent the Voting centers formed throughout the country during election periods. It also represents any Electronic Voting Machine that can connect to our server during normal working days for regular applications like viewing election results, editing profiles and so on. The middle phase of the architecture comprises of Application servers. It should be noted that the servers consist of back-end applications to handle different tasks. The end phase is comprised of storage subsystems, mainly the databases. These phases communicate in a formal protocol. That is, application server communicates directly with the clients and the storage devices. However, clients-databases communication is not direct. The application server – through a database connector handles all database requests from the clients-side to the database, as well as the responses are controlled by the server. *(A general view of this system is given in the diagram)*



General View of System Architecture

The major components in the system can be represented in form of modules. Therefore, we have three unique major modules Clients, Application server, and Database server modules. A client issues a communication or data request with the server. The server (in many functions of the systems) checks the validity and eligibility of the client to the system by contacting the data storage server. Upon the response from the database server; the Application server responds to the client request with positive or negative acknowledgement. *It should be remembered that there is no direct communication between the clients and the database server.*

6.2 System Package Distribution

We can decompose the main classes / objects this system into different packages on the basis of their responsibilities and the work they had to perform into different packages. These packages are: (*Referred to the Architecture Diagrams made according to the system – See Appendix B).*

6.2.1 Client Layer:

- **Clients**
  It includes the users of the system.
- **Voter Complaint Proceeding**
  It includes all the classes that are necessary to proceed the voters' complaints.
- **Voter-Vote Interaction**
  It includes all the classes that are necessary for casting a vote and for the voter-vote interaction.
- **User Log In portal**
  It contains the classes that are necessary for making the user log in to the system with his specific credentials.
- **Pre- and Post-Election Voter-System Interaction**
  This package contains all the classes through which the voter can get information about the election – the time before and after elections – to see voter or candidate list or to see the election results.

6.2.2 Database Layer:

- **Server**
  The server performs all the calculations and the logical work required by every user request.
- **E-Voting Server Functions**
  This package defines the classes that are important for performing different functions, by the server.
- **Calculated Results**
  This package contains the election results that are to be seen by the E-Voting System user.

6.2.3 Database Layer:

- **Database**
  It contains the database class- the database stores all the data regarding the E-Voting System.
- **Credentials**
  It contains the credentials of all the users.
- **Lists**
  This package contains all the lists, that are used and seen before and during elections.
- **Profiles**
  This package contains the profile data of all the E-Voting System users.

6.3 System Deployment

As the 3-tier architecture has been selected for this system, one of the reason that why this architecture style was chosen for this system is its deployment. As the Electronic Voting System is a vast system and its interface components has to be deployed at many different places in the whole country. For calculating the results and processing all the data the Server and the database should be deployed at some other places in the country such as in NADRA or PEC offices. So that when we divide the system into three layers according to the 3-tier architecture, each of the layer i.e. the client layer, the server layer and the database layer can be deployed at different places as desired by the election authorities.

During the deployment, this system will use the fingerprint scanners so as to scan the voters' fingerprints so that he could be verified. This system also uses many artifacts, devices and many different execution environments so that it can process correctly and fulfill the user requirements.

The deployment diagram can be seen in the Appendix B.

6.4 Design Rationale

The 3-tier Client/Server Architecture is chosen for this system as it is the most suitable architecture for this system. It could also be the layered architecture but it wouldn't be suitable as in the layered architecture there is only one-way communication while in this system we need two-way communication. That's why layered architecture was rejected for this system.

There is also the 2-tier client/server architecture present but according to the requirements of this system. Where there is a lot of user communication with the server, which processes the requests and performs a lot of work and stores a lot of data in the database. It was better to choose the 3-tier architecture. As, this is a voting system so that all of these layers had to be integrated at different places in the country. That's why it was better to choose the 3-tier architecture.

Other architecture designs simply don't serve the purpose which is required for this system. Hence, 3-tier client server architecture is chosen to be the most suitable architecture for this system.

7. HUMAN INTERFACE DESIGN

7.1 Overview of User Interface

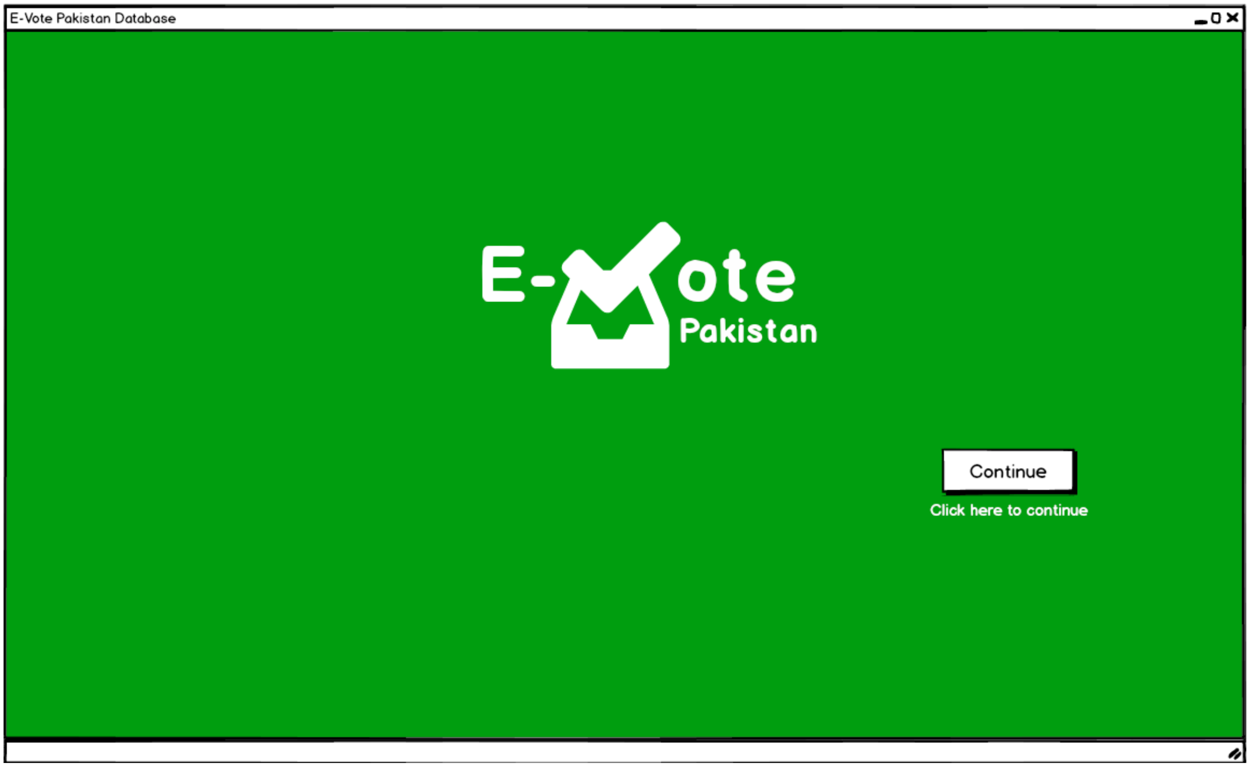
The interface of this system will be made in such a way that it would be very easy to use. User-interfaces in this system will be different in those two modes. In normal interactive mode, there will be common home page interface for all system users and they will use this page for login operation.

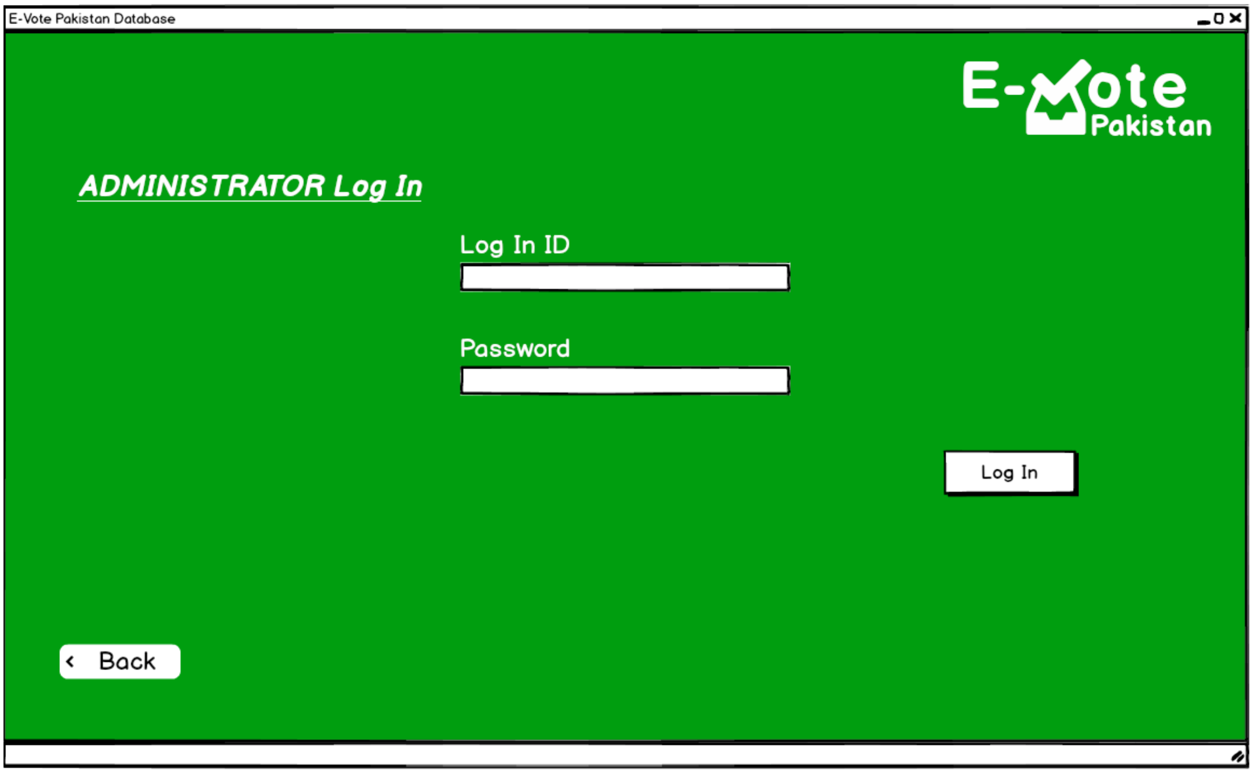
Administrator's interface will include his own profile and he will conduct the election tasks by using this interface. It interface will cover the functionalities related problem solving and election results.

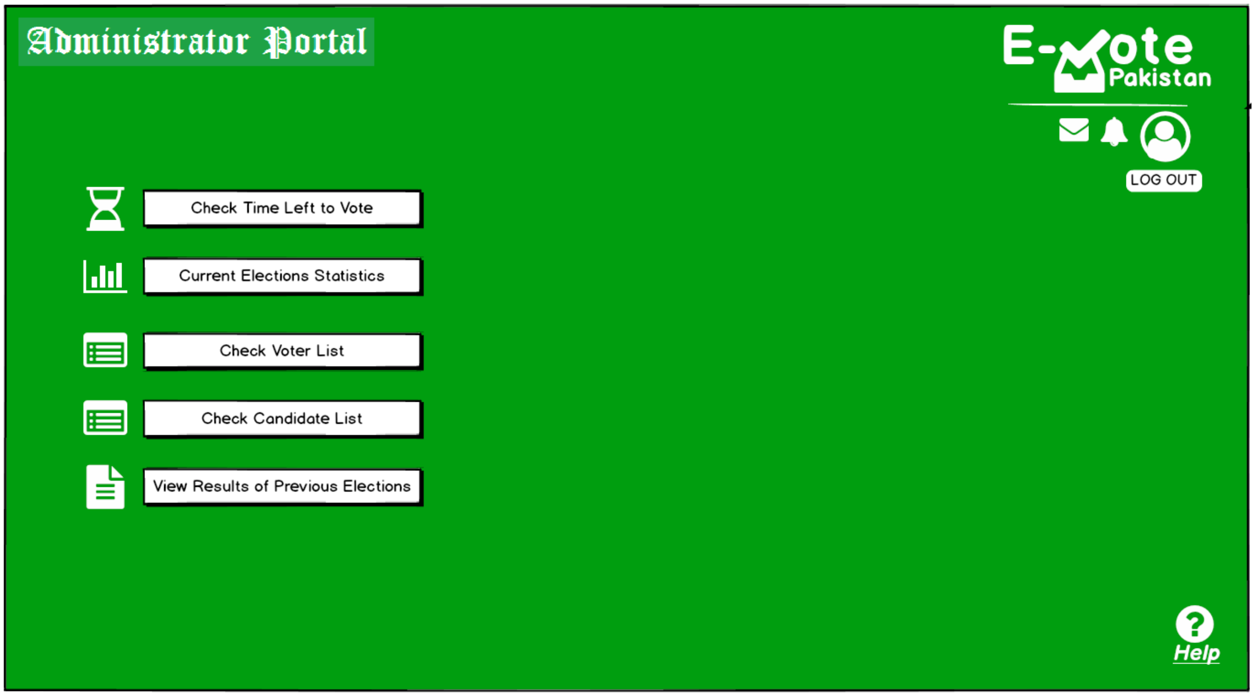
In election mode, there will be a major interface that the voting operation is executed. This interface will be used by the voters so they can cast their votes easily.

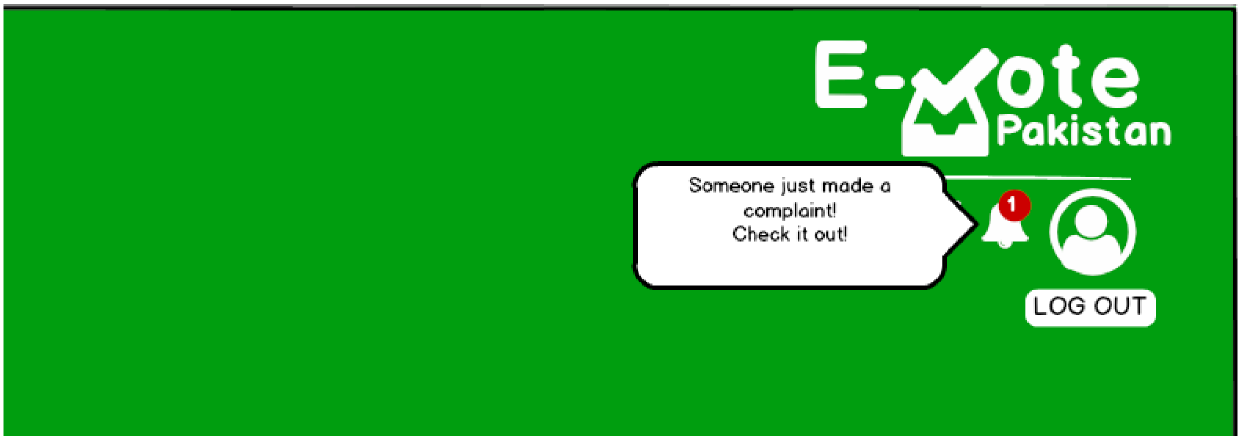
7.2 Screen Images

Following are the screenshots showing the interface from the user's perspective. *(This prototype has been made using the prototyping tool known as Balsamiq.)*

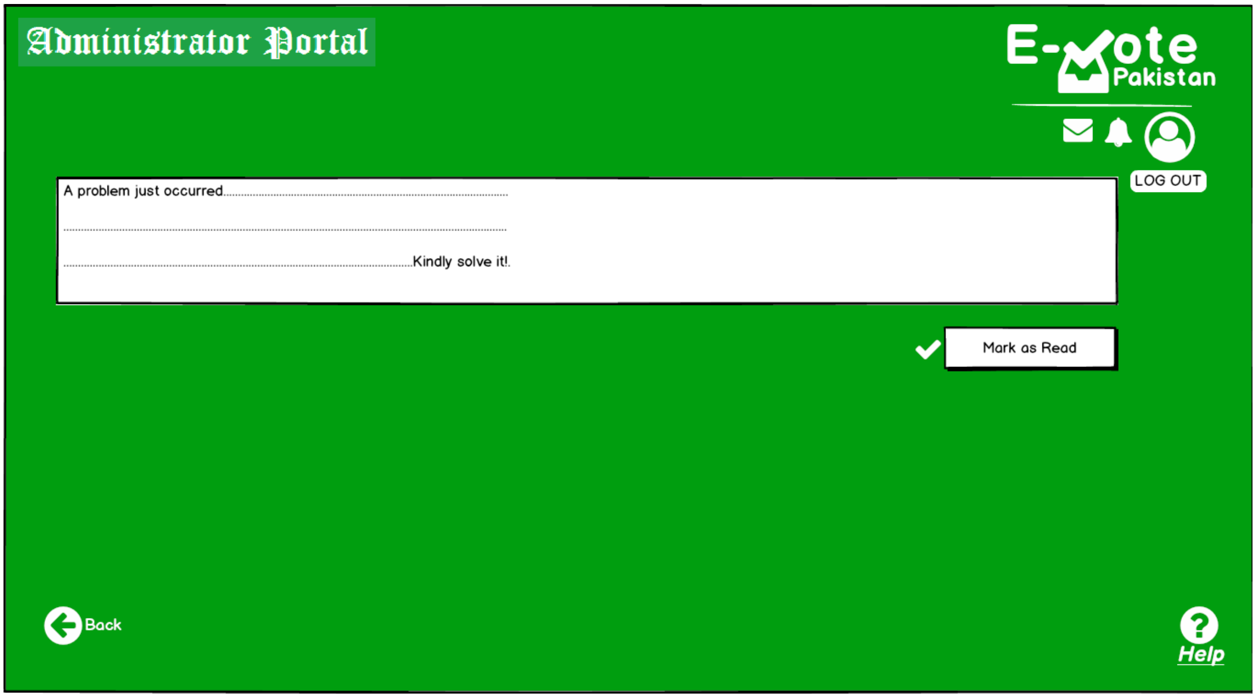

E-Vote Pakistan Database
E-Vote Pakistan
Continue
Click here to continue


E-Vote Pakistan Database
E-Vote Pakistan
ADMINISTRATOR Log In
Log In ID
Password
Log In
Back

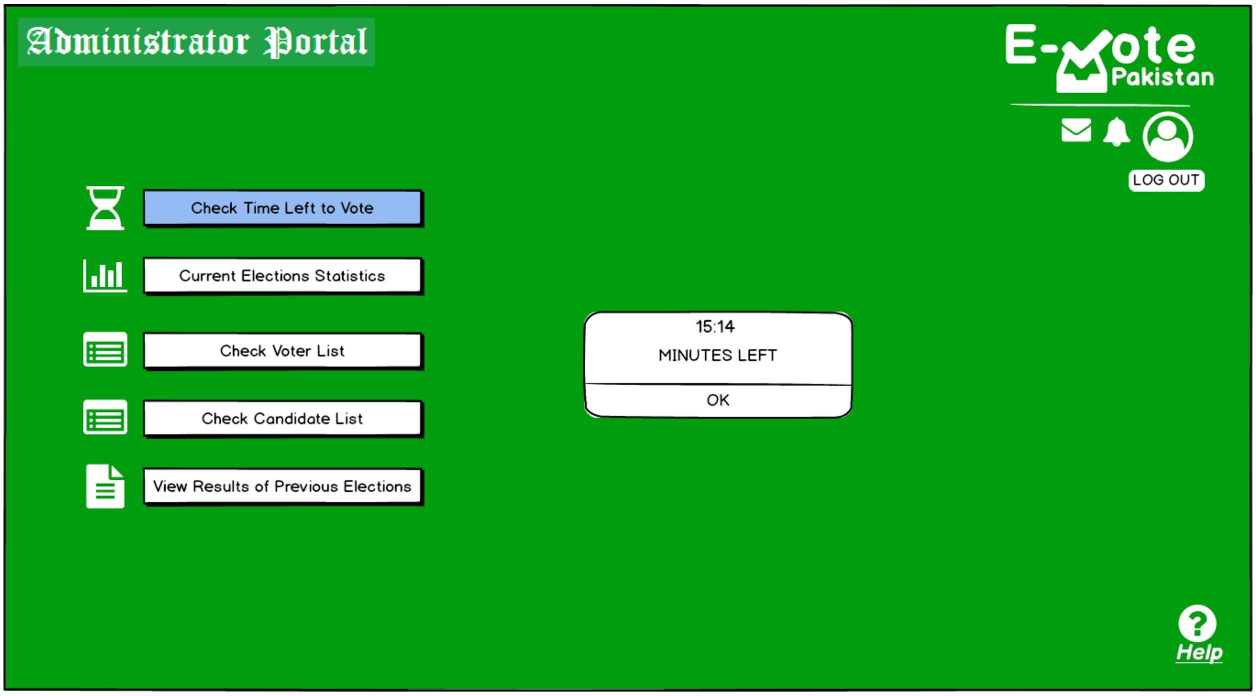




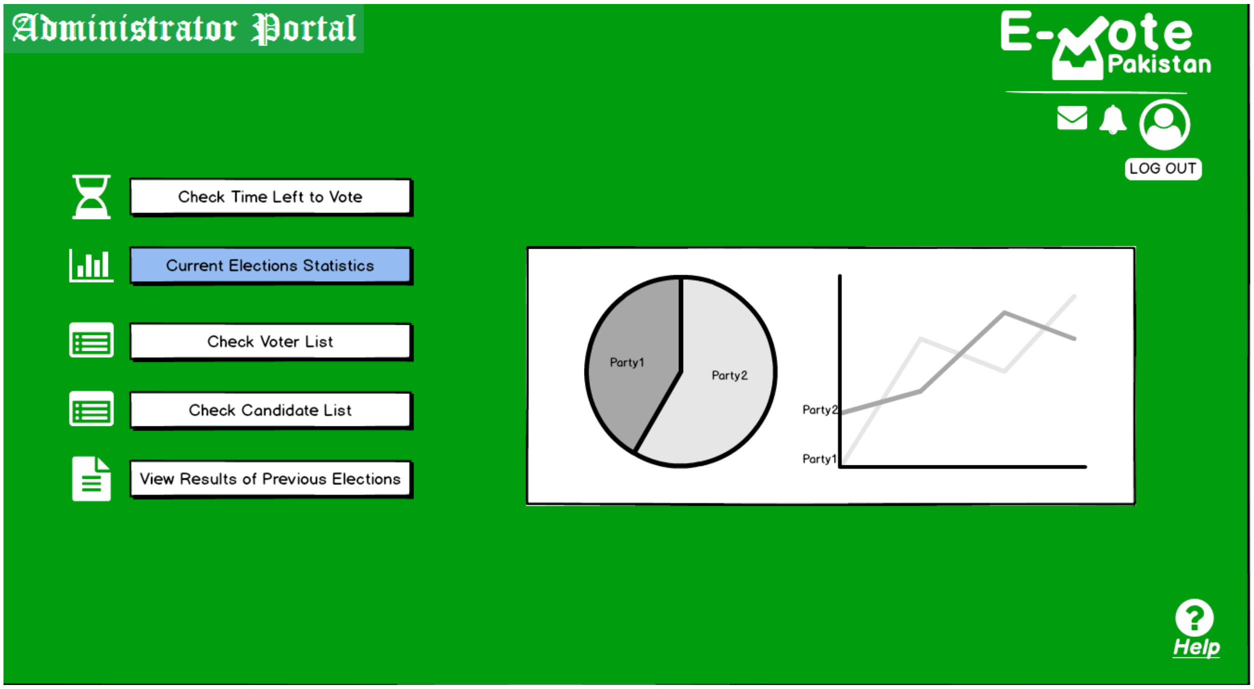
Someone just made a complaint!
Check it out!

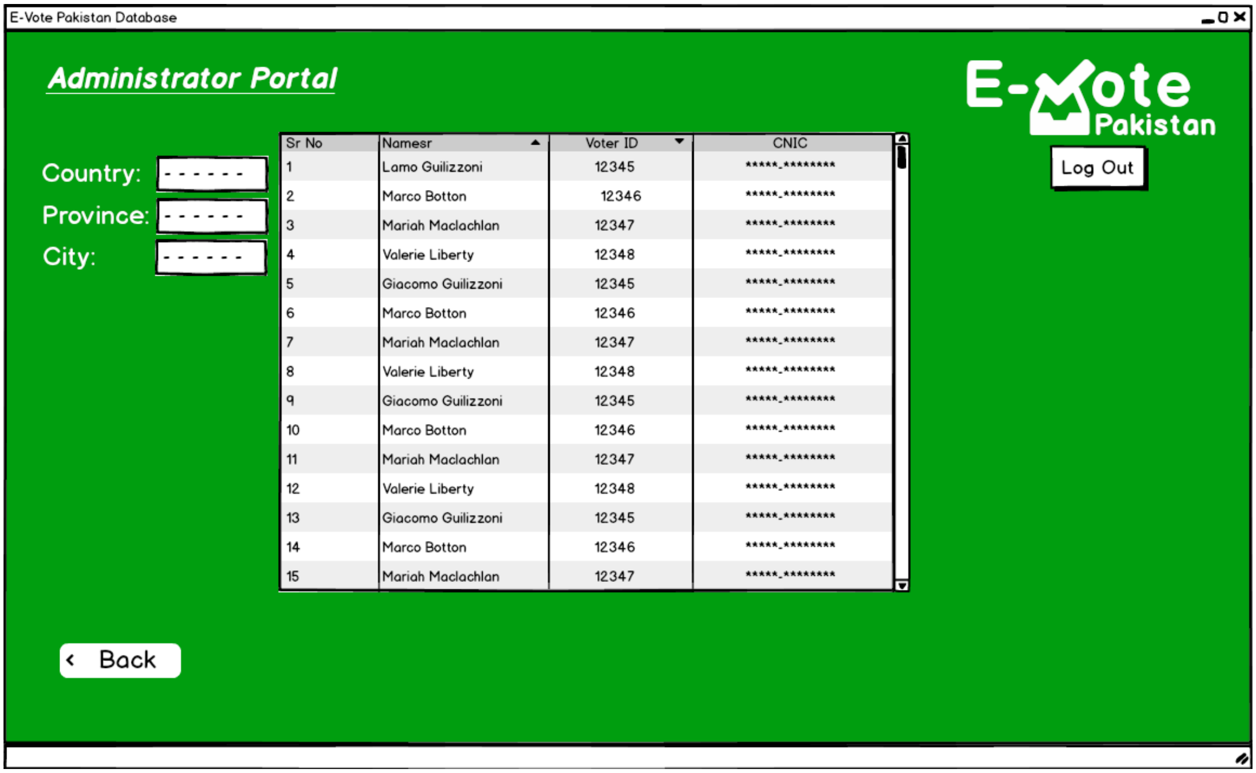


A problem just occurred..Kindly solve it!.

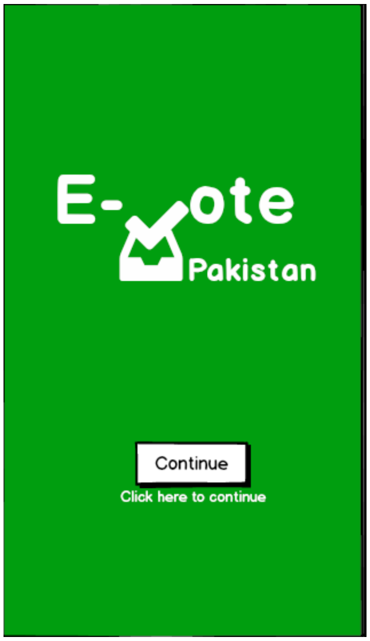


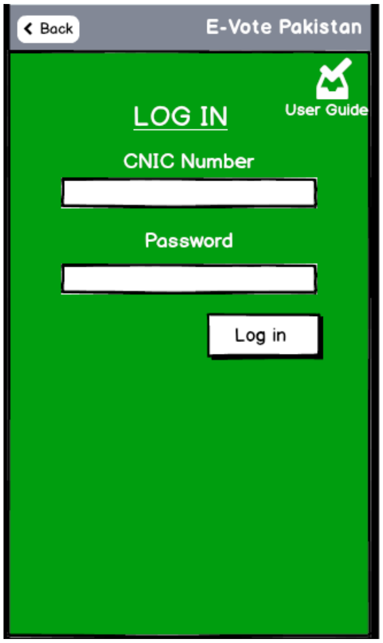
15:14
MINUTES LEFT

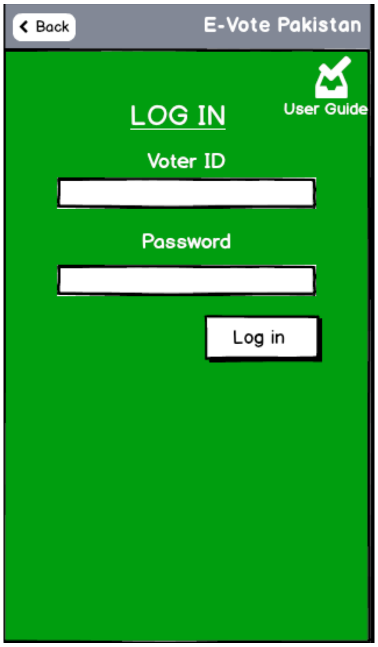


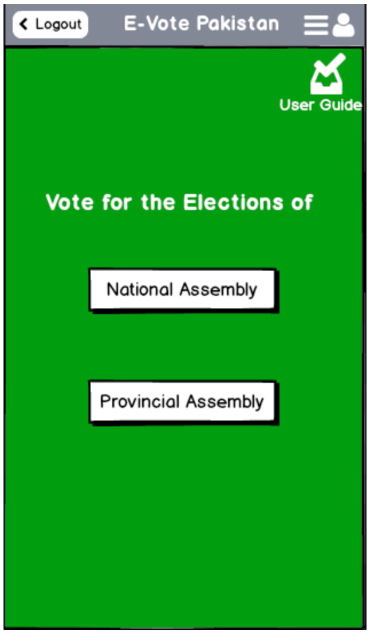


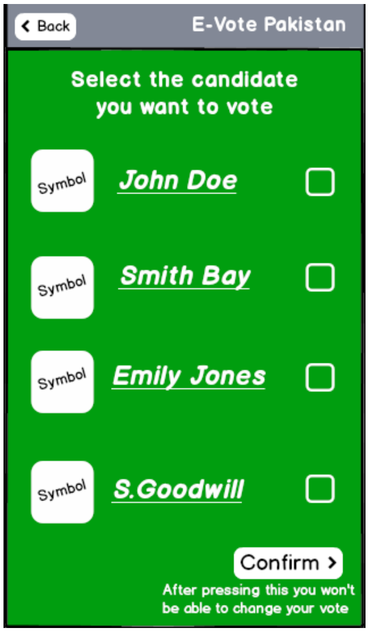
Sr No	Namesr	Voter ID	CNIC
1	Lamo Guilizzoni	12345	*****-********
2	Marco Botton	12346	*****-********
3	Mariah Maclachlan	12347	*****-********
4	Valerie Liberty	12348	*****-********
5	Giacomo Guilizzoni	12345	*****-********
6	Marco Botton	12346	*****-********
7	Mariah Maclachlan	12347	*****-********
8	Valerie Liberty	12348	*****-********
9	Giacomo Guilizzoni	12345	*****-********
10	Marco Botton	12346	*****-********
11	Mariah Maclachlan	12347	*****-********
12	Valerie Liberty	12348	*****-********
13	Giacomo Guilizzoni	12345	*****-********
14	Marco Botton	12346	*****-********
15	Mariah Maclachlan	12347	*****-********



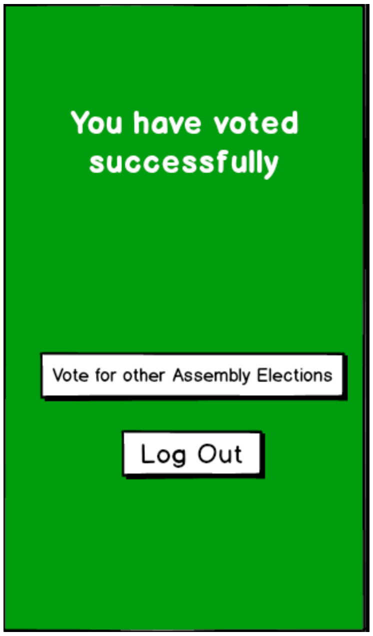




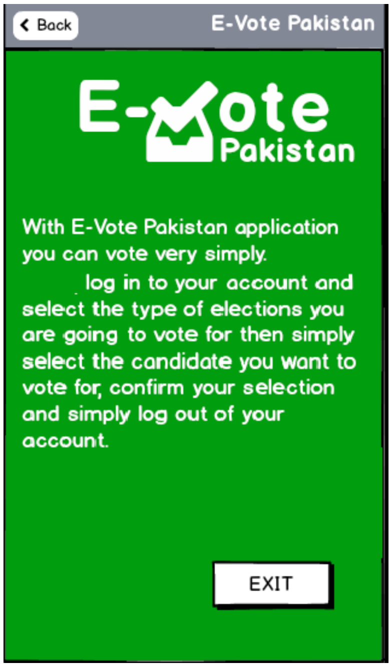




After pressing this you won't be able to change your vote



You have voted successfully



With E-Vote Pakistan application you can vote very simply.

log in to your account and select the type of elections you are going to vote for then simply select the candidate you want to vote for, confirm your selection and simply log out of your account.

8. APPENDIX A

(This page is intentionally left blank

9. APPENDIX B

(This page is intentionally left blank)