E-voting For Smart City

By

Urooj Wahid

Supervised By

Memoona Afsheen

Department of Computer Sciences

Quaid-i-Azam University

Islamabad

2015-2019

ACKNOWLEDGMNET

In the name of Allah, the most Beneficent and the most Merciful

First, I would like to thanks to Allah Almighty. Through his mercy, I was able to complete my project. Countless salutations upon the Holy Prophet Hazrat Muhammad (S.A.W), source of knowledge for enlightening with essence of faith in Allah and guiding the humanity, the true path of life.

In my journey towards this degree, I have found a teacher, a friend, an inspiration, a role model and a pillar of support in my Guide, Ma'am Memoona Afsheen Malik. Without her able guidance, this project would not have been possible and I shall eternally be grateful to her for her assistance. Furthermore, I would like to give thanks to all faculty members of computer science department for their help, giving proper attention and time.

I would also like to express my gratitude to my parents and all my family members for their support and encouragement. It would be inappropriate if I omit to mention the names of my dear friends Muhammad Sumair Muneer, Muhammad Ahmad, Ifza Gul, Sadia Manzoor, Ali Yar Khan and Ali Azeem Khan who have in their own ways, kept me going on my path to success, assisting me as per their abilities, in whatever manner possible and for ensuring that good times keep flowing.

Urooj Wahid
2015-2019

Abstract

Fair electoral process played a vital role in any democratic country for its progress and considering the voice of common people in country's daily business. In most of countries electoral process is manual but recently few countries adopted e-voting. Currently, E-voting application is managed by centralized authorities which can be: (i) a point of failure, (ii) source of tampering of votes for illegal benefits, and (iii) used to accept the influence of non-democratic forces. In this project, E-voting system for smart city is designed and implemented. It is a decentralized application which is developed by using the blockchain tools, smart contracts, RESTAPI and client side web technologies. The REST API facilitates to verify users, create electoral areas, register candidates (those are participating in electoral process), provide list of candidates to the users. Since this application has certain limitations therefore we divided functionality into two main groups (also known as roles): (i) Admin, it creates areas, register elected candidates and verify citizen's information to authorize them to cast vote in their constituency, (ii) Citizen, the Citizens register themselves and manage their personnel information which is stored in above mentioned REST API based web services. Based on the designated areas, each citizen can select the candidate from the given list and then poll the vote. The vote poling process is implemented by using Blockchain technology in which this process is governed by the smart contract and exploits the concept of crypto-token in which the token cannot be "double spend". This concept avoids the citizens to poll double votes, which is the basic concern of the most of exiting voting process. In addition to that the voting process can be verifiable and immutable which helps to develop trust on the system. . After completion of the voting process, counting process will start and then will display the results.

Table Of Contents

Chapter 1: Introduction 1
1.1 Problem Definition 2
1.2 Existing System 2
1.3 Proposed solution 3
1.4 Scope 3
1.5 Objective 4
1.6 Constraints 4
1.7 Project Organization 4
1.8 Project Deliverables 5
1.9 Risks and Contingencies 5
1.10 Project management plan 6
Chapter 2: E-Voting and Blockchain 8
2.1 E-Voting 9
2.2 Blockchain: 9
2.3 Role of Blockchain in E-Voting: 11
Chapter 3: Requirement Gathering and Analysis 12
3.1 Product Overview 13
3.2 Major Functions 13
3.3 Major Inputs and outputs 13
3.4 Definitions, Acronyms and Abbreviations 14
3.5 Overview 14
3.6 User Characteristics 14
3.7 Constraint 15
3.8 Assumption and Dependencies. 15
3.9 Specific Requirements 15
3.10 Software Interfaces 16
3.11 Communication Protocol 16
3.12 Non-Functional Requirements 16
3.13 Use Cases Description 18
3.14 System Sequence Diagram 27
3.15 Domain Model 35
3.16 Database Requirements: 35
Chapter 4: Software Design Description 36
4.1 Introduction 37

4.2 System Architecture Design 38
4.3 User Interface Design 39
4.4 Sequence Diagram 42
4.5 Class Diagram 45
Chapter 5: Software Testing 46
5.1 Introduction 47
5.2 Test Plan Environment 47
Chapter 6: Software Implementation 50
6.1 Language Selection 51
6.2 Tools Selection 51
7.1 Summary 53
7.2 Conclusion 53
7.3 Future Enhancements 53
References 54

List of Tables

Table 1: Definitions, Acronyms and Abbreviations 14
Table 2: Register User 18
Table 3: verify user against double registration 18
Table 4: Authentication of user 19
Table 5: Create Area 19
Table 6: Assign Candidates to Area 20
Table 7: View Area 20
Table 8: Delete Area 21
Table 9: Add Election 21
Table 10: View Election 22
Table 11: Update Election 22
Table 12: Delete Election 23
Table 13: Add Party Candidate 23
Table 14: View Party Candidate 24
Table 15: Update Election 24
Table 16: Delete Party candidate 25
Table 17: Show Candidate List 25
Table 18: Poll Vote 26
Table 19: Count Vote 26
Table 21: Requirement Traceability matrix 37
Table 22: Register user test case 47
Table 23: Create area test case 48
Table 24: Assign area to candidate test case 49
Table 25: Poll vote test case 49

List of Figures

Figure 1.1: Time Table 1 6
Figure 1.2 : Time Table 2 6
Figure 1.3: Gantt chart 1 7
Figure 1.4: Gantt chart 7
Figure 3.1: Use Case Diagram 17
Figure 3.2: Register user System Sequence Diagram 27
Figure 3.3: Authenticate user System Sequence Diagram 28
Figure 3.4: Create Area System Sequence Diagram 28
Figure 3.5: Assign candidates to area System Sequence Diagram 29
Figure 3.6: Add candidate System Sequence Diagram 29
Figure 3.7: Add Election System Sequence Diagram 30
Figure 3.8: Update Candidate System Sequence Diagram 30
Figure 3.9: Update election System Sequence Diagram 31
Figure 3.10: View Candidate System Sequence Diagram 31
Figure 3.11: View area System Sequence Diagram 32
Figure 3.12: View Election System Sequence Diagram 32
Figure 3.13: Poll vote System Sequence Diagram 33
Figure 3.14: Delete candidate System Sequence Diagram 33
Figure 3.15: Delete area System Sequence Diagram 34
Figure 3.16: Delete election System Sequence Diagram 34
Figure 3.17: Count votes System Sequence Diagram 34
Figure 3.18: Domain Model 35
Figure 3.19: ERD 35
Figure 4.1: System Architecture 38
Figure 4.2: Admin login Interface 39
Figure 4.3: Admin dashboard Interface 39
Figure 4.4: Announce Election Interface 40
Figure 4.5: Electoral Area Interface 40
Figure 4.6: Register Candidate Interface 41
Figure 4.7: Main Website Interface 41
Figure 4.8: User Login Interface 42
Figure 4.9: Registration Form interface 42
Figure 4.10: Sequence Diagram Register user 43

Figure 4.11: Sequence Diagram Authenticate User 43
Figure 4.12: Sequence Diagram Assign candidates to Area 44
Figure 4.13: Sequence Diagram Poll Vote 44
Figure 4.14: Class diagram 45

Chapter 1: Introduction

Introduction

This chapter first introduces the concept of E-voting for smart city and then highlights the problems in existing systems. It also describes the project planning and project organization. This section includes scope and objective of the system

1.1 Problem Definition

The title of the project is “E-voting for Smart City”. In this project, the objective is to design and develop a blockchain based voting system. Currently in our country, the electoral process is based on paper voting which has many flaws. For example, invalid votes, double voting issue, integrity of votes, misplace of votes, vote counting problems and fraud in final vote counting. Such problems motivated us to consider the modern tools and technologies for developing fast and fair voting process.

1.2 Existing System

The existing system of electoral process in our country is based on paper voting which has certain flaws. For example,

- Invalid votes: check whether vote is valid or not valid (vote verification).
- Integrity of votes: if a voter cast a vote, it won’t be changed or modified.
- Misplace of votes: sometimes vote may misplace.
- Problems in vote counting: counting of votes take a lot of time.
- Fraud in final vote counting: sometimes polling agents add more votes during counting.

Our Voting system comprises on two main phases:

1.2.1 Identification and Validation of Voters

In this process the polling agents identified the voters and then validate them through available paper-based voting lists. In the identification process, the polling agent requested voters to present his National Identity Card (NIC) which is used to verify his NIC number, his name and then matches his photo with the presenter. Upon successfully matching of the identification credentials, the voter prints his thumb on the voter list. After that the polling agent issues him a ballet paper on which various candidates along with allotted symbols are displayed. This ballet list is known as actual *Vote*. Otherwise, if anyone of the credentials does not match then the polling agent does not allow them to cast their vote.

1.2.2 Voting and counting process

In order to maintain the transparency in voting process, the voter walks to an isolation place where he stamped on the ballet paper. After that the voter puts vote in to the ballot box. After balloting time expiry, the polling agents and their colleagues start counting and then forward results to the central election office. During the counting, if the vote is not properly stamped on the designated area then the vote is rejected.

1.3 Proposed solution

To solve the problems as discussed in existing system, various protocols will be designed which will anonymously identify, validate and then will help citizens to cast their votes. In this project various tools, technologies and protocols can be used but conceptually the overall system will base on Blockchain (Distributed Ledger Technology - DLT) technology due to its promising features like immutability, verifiability and avoiding double spending problem. All these features are the core requirements of the voting system. The proposed system will provide solutions to manage digital identities (information about citizens like standard (Identity Management System) IDMS), secure credential handling and management of voting process (in this scenario it is known as transaction). In the whole system the identities will play a vital role because this information is to develop consensus on an issue or selection of a candidate in democratic process. E-voting for smart city has very unique requirements e.g. anonymous identities, verifiability, and immutability, etc.

1.3.1 Blockchain based Identity Framework

In this framework, we will investigate, design and implement a solution that will help to manage anonymous, verifiable and trusted identities of citizens. The framework will be based on the concept of design by ownership in which it will help to register citizens, manages their personal information in blockchain and then generate verifiable anonymous identities for each citizen. In addition to that, the double registration problem will be solved to create a unique and global identity for each citizen. This framework can be considered as an extension of our local NARDA identity system.

In current applications, whenever a third-party application will use our identity framework then it will be used anonymously without revealing citizen's personal information since most of the third-party system (applications) only requires validation of the identity and trust on these identities instead of using their personal information. In order to develop trust on the identities, our designed solution will serve as a logical engine to verify the validity of the identity at any point during execution or may be after completion of any transaction. In our case E-voting is a use case of the identity framework.

1.3.2 E-Voting Application

On top of the above framework, we will design an application for Blockchain based e-voting where Blockchain will be used to:

- Prevent the voting process from double voting,
- Provide auditability of the voting process, and

1.4 Scope

The designed system will provide the major functionalities related to e-voting process such as registration of the voters, registration of the candidates for elections, initiating voting process from voter, verification and validation of identities, vote casting process and then final counting of the votes. In this whole scenario, the actual voting process needs carefully designing of the e-voting protocol. For such purpose, the system will generate a key for the voter which will serve to identify and validate a voter. After that, may be, the same key can be used to cast the vote according to the process explained in the proposed solution section. The validation and avoiding double voting process, we will use the concepts of smart contract therefore whenever this process completes a transaction it must first execute the smart contract to validate the voter and then check whether its vote is already polled or not.

1.5 Objective

The primary objective of this system is to create decentralized online e-voting system (web based) using smart contracts. In this we will register user, manage its personnel information, and provide features to organize electoral process. In addition, this system will generate a public key for the user that help the user to cast a vote. This type of system will increase security because it will be based on public key cryptography which will secure all transactions. To design and implement a voting process which should provide anonymity, validation and immutability of voting record

1.6 Constraints

The major constraints in this project are:

- Voter’s anonymity: keep the voter’s information anonymous.
- Unique key: for every voter, unique key will be generated.
- Traceability: The ability to trace all the stages that led to particular point in a process that consist of a chain of interrelated events.

1.7 Project Organization

In project organization, we will explain about which process model will be followed, roles and responsibilities and which tools and techniques are used for the development of the system.

1.7.1 Process Model

For this framework, agile methodology will be pursued is Spiral model. The purpose for picking this model is that the technology used in this framework is new and I have no involvement on this innovative technology, further the prerequisites can change with the progression of time therefore we will divide our project into small sprints (a use case) and each single sprint will be focused. Before dividing the project in to sprint, various protocols such as registration, voting, validation, etc. will be carefully designed to avoid any integration problem at the end of the project. The major disadvantage of this model is that for the successful completion of the project is very much dependent on Risk Analysis. So, divide-and-conquer (project into smaller sprints) rule will help us to minimize the Risk involve in this project.

1.7.2 Roles and Responsibilities

The “E-voting for Smart City” is a single student project. To gather requirements, communication with the supervisor is important. With the regular interaction with supervisor, refinement of the requirements and testing of the system will be done according to the requirements. Meet with the stakeholder for gathering requirements. . The following are the roles and responsibilities of the student formulation are; project plan, requirements specification, analysis, architecture specifications, component or object specification, source code, test plan and final deliverable.

1.7.3 Tools and Techniques

Argo UML tool and Microsoft Visio for UML diagrams such as use case diagram, class diagram, activity diagram, domain model and Entity relationship diagram and for writing documentation Microsoft word is used. For designing a plan of the system, project libre is used.

Html along with Ajax will be used to create front end of the application. CSS will be used as cascading style sheets to style the contents. Bootstrap and JavaScript will also be used for creating interactive web modules.

In order to develop Registration services, a REST API based framework will be used. In this system, the plan is to consider drop wizard framework for developing REST API for registration services. This framework is available for java-based backend development with rich functions and features.

For integrating Registration services and automating voting process, a blockchain (DLT) based solution will be used. In this regard, our intension is to first assess Ethereum or Corda tools and then will decide most appropriate tools for our project.

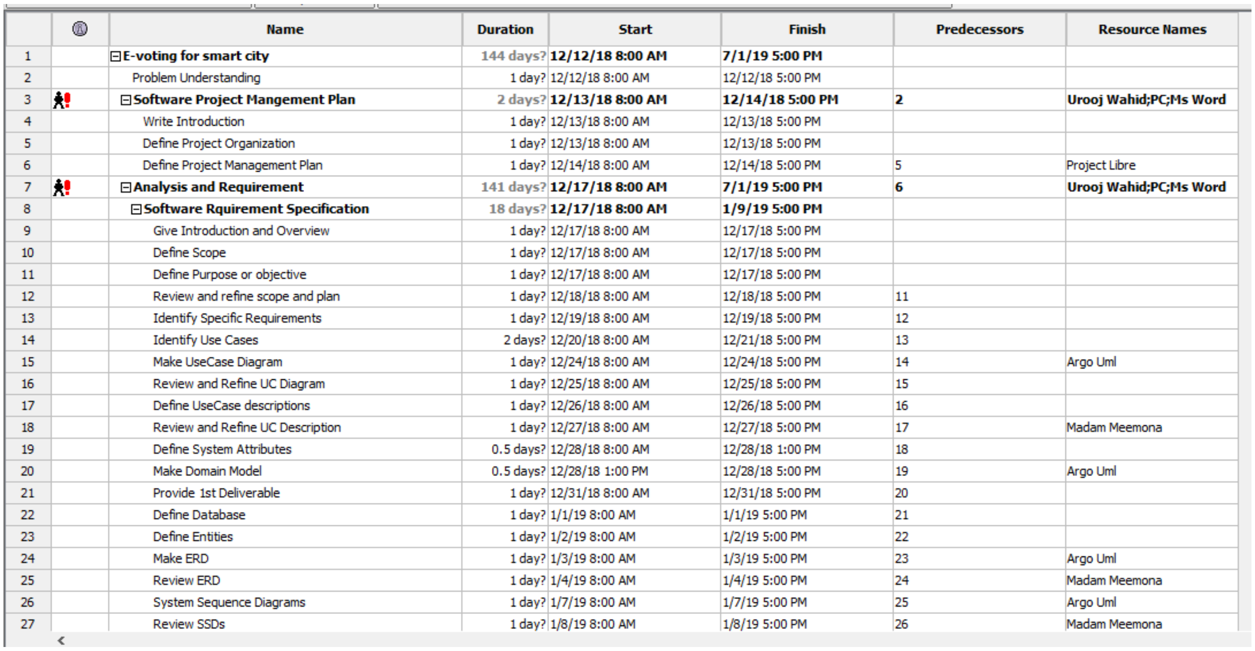
1.8 Project Deliverables

- Software Process Management Plane
- Software Requirements Specification
- Software Design Description
- Software Test Documentation

1.9 Risks and Contingencies

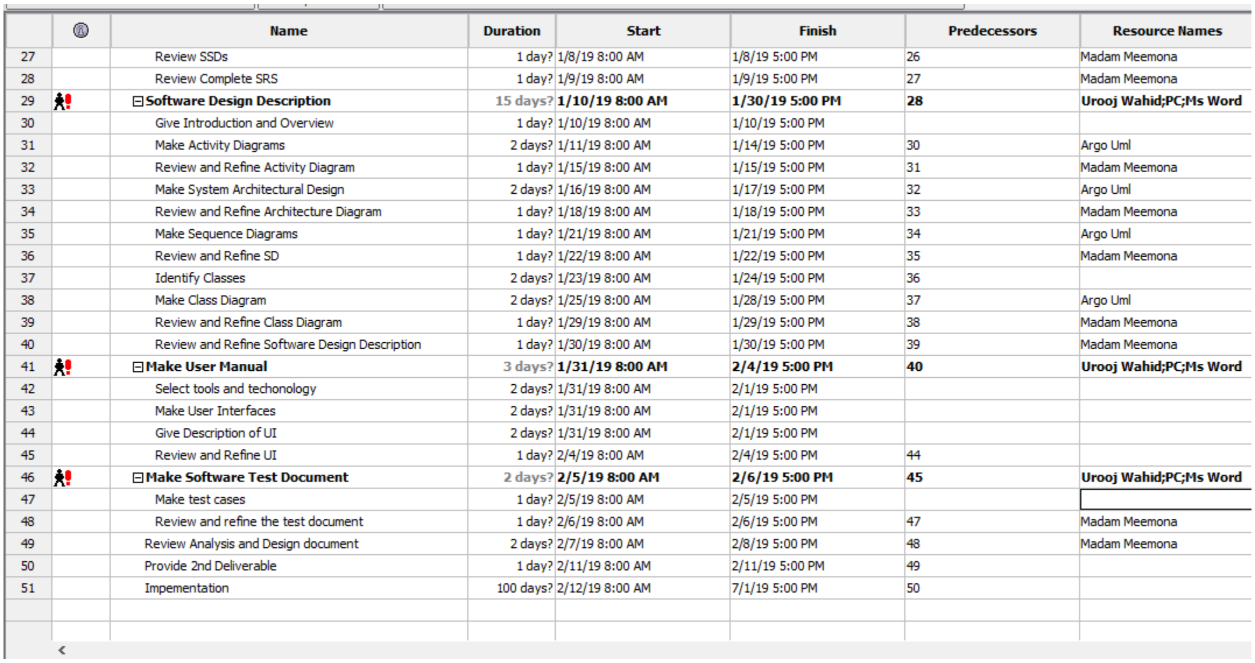
Smart Contracts technology is still evolving and its coding techniques are changing rapidly.

1.10 Project management plan



		Name	Duration	Start	Finish	Predecessors	Resource Names
1		⊟E-voting for smart city	144 days?	12/12/18 8:00 AM	7/1/19 5:00 PM		
2		Problem Understanding	1 day?	12/12/18 8:00 AM	12/12/18 5:00 PM		
3		⊟Software Project Mangement Plan	2 days?	12/13/18 8:00 AM	12/14/18 5:00 PM	2	Urooj Wahid;PC;Ms Word
4		Write Introduction	1 day?	12/13/18 8:00 AM	12/13/18 5:00 PM		
5		Define Project Organization	1 day?	12/13/18 8:00 AM	12/13/18 5:00 PM		
6		Define Project Management Plan	1 day?	12/14/18 8:00 AM	12/14/18 5:00 PM	5	Project Libre
7		⊟Analysis and Requirement	141 days?	12/17/18 8:00 AM	7/1/19 5:00 PM	6	Urooj Wahid;PC;Ms Word
8		⊟Software Rquirement Specification	18 days?	12/17/18 8:00 AM	1/9/19 5:00 PM		
9		Give Introduction and Overview	1 day?	12/17/18 8:00 AM	12/17/18 5:00 PM		
10		Define Scope	1 day?	12/17/18 8:00 AM	12/17/18 5:00 PM		
11		Define Purpose or objective	1 day?	12/17/18 8:00 AM	12/17/18 5:00 PM		
12		Review and refine scope and plan	1 day?	12/18/18 8:00 AM	12/18/18 5:00 PM	11	
13		Identify Specific Requirements	1 day?	12/19/18 8:00 AM	12/19/18 5:00 PM	12	
14		Identify Use Cases	2 days?	12/20/18 8:00 AM	12/21/18 5:00 PM	13	
15		Make UseCase Diagram	1 day?	12/24/18 8:00 AM	12/24/18 5:00 PM	14	Argo Uml
16		Review and Refine UC Diagram	1 day?	12/25/18 8:00 AM	12/25/18 5:00 PM	15	
17		Define UseCase descriptions	1 day?	12/26/18 8:00 AM	12/26/18 5:00 PM	16	
18		Review and Refine UC Description	1 day?	12/27/18 8:00 AM	12/27/18 5:00 PM	17	Madam Meemona
19		Define System Attributes	0.5 days?	12/28/18 8:00 AM	12/28/18 1:00 PM	18	
20		Make Domain Model	0.5 days?	12/28/18 1:00 PM	12/28/18 5:00 PM	19	Argo Uml
21		Provide 1st Deliverable	1 day?	12/31/18 8:00 AM	12/31/18 5:00 PM	20	
22		Define Database	1 day?	1/1/19 8:00 AM	1/1/19 5:00 PM	21	
23		Define Entities	1 day?	1/2/19 8:00 AM	1/2/19 5:00 PM	22	
24		Make ERD	1 day?	1/3/19 8:00 AM	1/3/19 5:00 PM	23	Argo Uml
25		Review ERD	1 day?	1/4/19 8:00 AM	1/4/19 5:00 PM	24	Madam Meemona
26		System Sequence Diagrams	1 day?	1/7/19 8:00 AM	1/7/19 5:00 PM	25	Argo Uml
27		Review SSDs	1 day?	1/8/19 8:00 AM	1/8/19 5:00 PM	26	Madam Meemona

Figure 1.1: Time Table 1



		Name	Duration	Start	Finish	Predecessors	Resource Names
27		Review SSDs	1 day?	1/8/19 8:00 AM	1/8/19 5:00 PM	26	Madam Meemona
28		Review Complete SRS	1 day?	1/9/19 8:00 AM	1/9/19 5:00 PM	27	Madam Meemona
29		⊟Software Design Description	15 days?	1/10/19 8:00 AM	1/30/19 5:00 PM	28	Urooj Wahid;PC;Ms Word
30		Give Introduction and Overview	1 day?	1/10/19 8:00 AM	1/10/19 5:00 PM		
31		Make Activity Diagrams	2 days?	1/11/19 8:00 AM	1/14/19 5:00 PM	30	Argo Uml
32		Review and Refine Activity Diagram	1 day?	1/15/19 8:00 AM	1/15/19 5:00 PM	31	Madam Meemona
33		Make System Architectural Design	2 days?	1/16/19 8:00 AM	1/17/19 5:00 PM	32	Argo Uml
34		Review and Refine Architecture Diagram	1 day?	1/18/19 8:00 AM	1/18/19 5:00 PM	33	Madam Meemona
35		Make Sequence Diagrams	1 day?	1/21/19 8:00 AM	1/21/19 5:00 PM	34	Argo Uml
36		Review and Refine SD	1 day?	1/22/19 8:00 AM	1/22/19 5:00 PM	35	Madam Meemona
37		Identify Classes	2 days?	1/23/19 8:00 AM	1/24/19 5:00 PM	36	
38		Make Class Diagram	2 days?	1/25/19 8:00 AM	1/28/19 5:00 PM	37	Argo Uml
39		Review and Refine Class Diagram	1 day?	1/29/19 8:00 AM	1/29/19 5:00 PM	38	Madam Meemona
40		Review and Refine Software Design Description	1 day?	1/30/19 8:00 AM	1/30/19 5:00 PM	39	Madam Meemona
41		⊟Make User Manual	3 days?	1/31/19 8:00 AM	2/4/19 5:00 PM	40	Urooj Wahid;PC;Ms Word
42		Select tools and techonology	2 days?	1/31/19 8:00 AM	2/1/19 5:00 PM		
43		Make User Interfaces	2 days?	1/31/19 8:00 AM	2/1/19 5:00 PM		
44		Give Description of UI	2 days?	1/31/19 8:00 AM	2/1/19 5:00 PM		
45		Review and Refine UI	1 day?	2/4/19 8:00 AM	2/4/19 5:00 PM	44	
46		⊟Make Software Test Document	2 days?	2/5/19 8:00 AM	2/6/19 5:00 PM	45	Urooj Wahid;PC;Ms Word
47		Make test cases	1 day?	2/5/19 8:00 AM	2/5/19 5:00 PM		
48		Review and refine the test document	1 day?	2/6/19 8:00 AM	2/6/19 5:00 PM	47	Madam Meemona
49		Review Analysis and Design document	2 days?	2/7/19 8:00 AM	2/8/19 5:00 PM	48	Madam Meemona
50		Provide 2nd Deliverable	1 day?	2/11/19 8:00 AM	2/11/19 5:00 PM	49	
51		Impementation	100 days?	2/12/19 8:00 AM	7/1/19 5:00 PM	50	

Figure 1.2 : Time Table 2

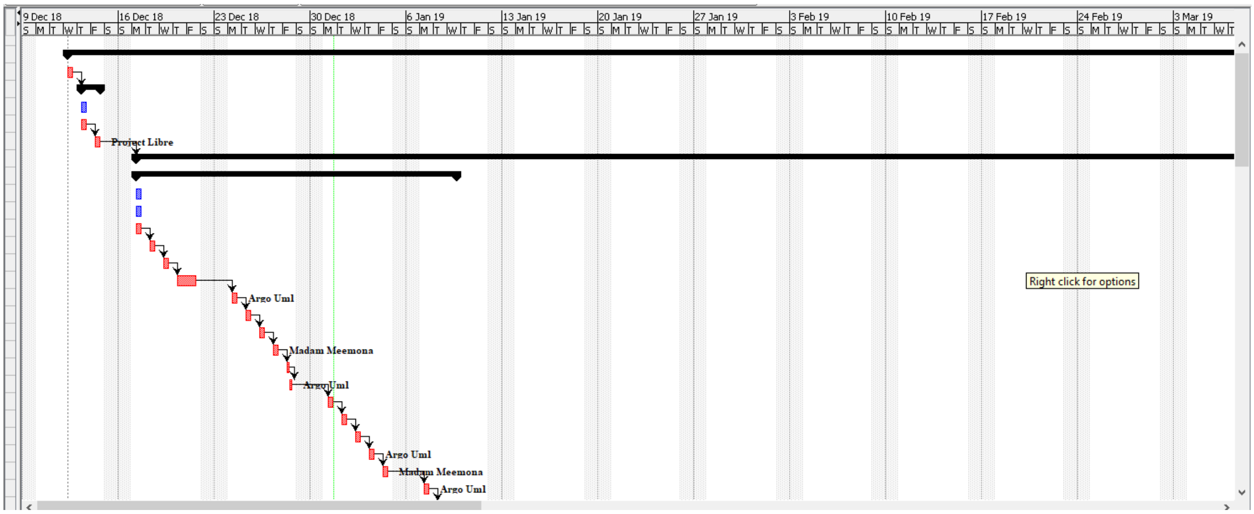


Figure 1.3: Gantt chart 1

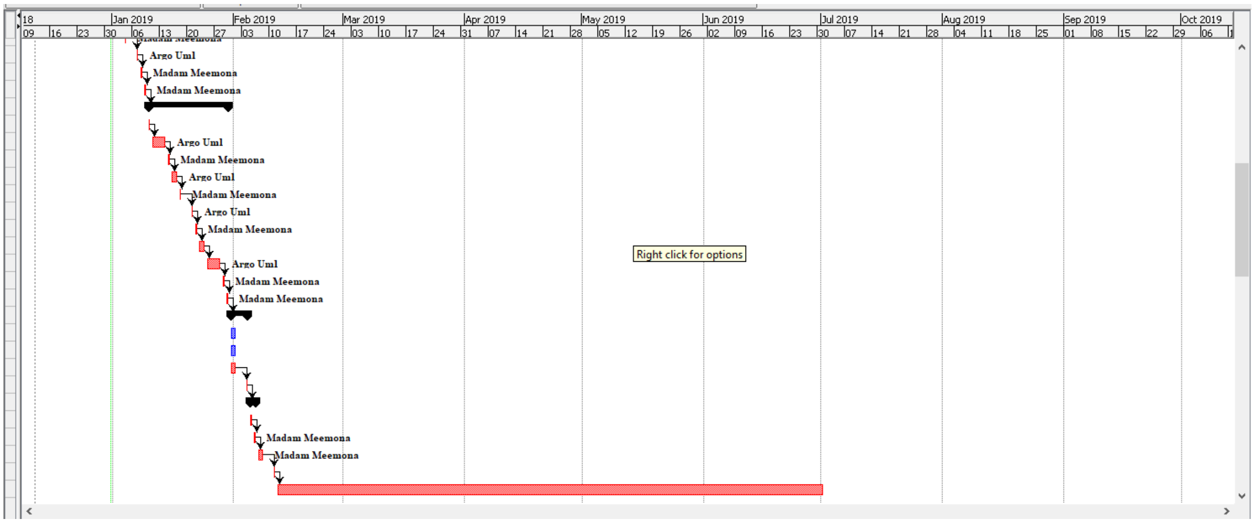


Figure 1.4: Gantt chart

Chapter 2: E-Voting and Blockchain

Introduction

The purpose of this chapter is to provide a short introduction to Blockchain and E-voting system. Voting is a general well-known process to elect a representative for specific designation. In this all the participant vote for potential representatives. The candidate which will secure more votes are declared winner. In most of cases this process is manual but due to the advancement in Information and Communication Technology (ICT), this process is also modernized, which is known as E-Voting.

2.1 E-Voting

Fair electoral process played a vital role in any democratic country for its progress and considering the voice of common people in country's daily business. In most of countries electoral process is manual but recently few countries adopted e-voting. E-voting, also called Electronic voting, utilizes electronic means to take care of polling of votes and counting of votes in an election. The introduction of e-voting raises some of the same challenges which are faced when applying electronic methods to any other subject such as security, accuracy, privacy, integrity etc. It should involve a secure transmission of ballots and votes via internet. Using voting process electronically has more advantages than paper based. It can save time in counting votes, results can be reported and published faster and expenses are expected to decrease. Voters save time and cost by being able to vote independently from their location. E-voting provides benefits to the people who are abroad, to cast a vote. For a country, electronic voting may improve the country's image and serve as promotion. In e-voting there is a voting machine which will be used to verify the identity of a voter and then helps to cast vote [1].

2.2 Blockchain:

Blockchain is a new and promising technology which provides certain prominent features required for distributed applications such as financial, business and public services in cities. [2]. These features are: (i) developing trust between two parties in decentralized environment, (ii) ensuring integrity of stored and exchanged information between stakeholders, and (iii) operating in an open environment by following certain rules and procedures defined in the smart contracts. Currently blockchain is widely used in the financial sector but by considering above features and strong cryptographic operations, it is a suitable candidate for smart city applications such as e-Voting, etc. Because such applications would need transparency in transactions, be trustworthy, open to all stakeholders in cities but maintains privacy by encapsulating identities, also provides controlled sharing of transactions/data with authorized stakeholders, and supports authentication and authorization features. After reading about the features of blockchain, we identified following features

1) ***Transparent:*** It is open in nature and all changes in the public blockchain are publicly viewable by all the stakeholders involved in the transaction which provides transparency. In modern democracy, it is required that all information related to public administration should be viewable by the citizens without violating citizens' privacy [3].

2) ***Immutability:*** All transactions in the ledger are immutable therefore it cannot be tampered and ensures the integrity of each transaction. If any part of transaction is changed or tempered, it can be easily detected by verifying its chain. Therefore, for the benefits of the society such features increase the confidence of the citizens on public administration services [4].
3) ***Ownership and Provenance:*** The data published in the ledger is cryptographically signed by the data producer. Therefore, it gives credit to its owner and ensures ownership of the data. Any change in the digital asset (e.g. business transaction, urban proposal from a citizen, etc) is recorded in the ledger block to provide fine-grained provenance [5].
4) ***Privacy****:* In today's digital age personal information (EU definition of personal info) from any application (e.g. health, public services, etc) should not be publically available e.g. through web apps or remotely accessed software systems. In case a particular information is required to be shared through open blockchain with selected recipients then it should be protected by using cryptographically encapsulated formats to protect the privacy of individuals [3].
5) ***Authorization:*** No doubt blockchain is open in nature but still using cryptographic techniques we can assure that only the authorized recipient or a group of recipients are able to view the shared information. This can be useful for GDPR's controlled sharing requirement by the data owner.
6) ***Trust:*** Since all information is immutable and verifiable therefore it will help to establish the trust of citizens on public administration services [3,4].
7) ***Identity Tracking:*** In most of the existing blockchain implementations, anonymous identity is used, which cannot be tracked back to its original owner. In case of smart cities and other business applications these identities can be linked with a citizen registration authority (e.g. city administration), so that these can be used to produce personal information for court and legal needs.
8) ***Consistency*** **and Accuracy:** Blockchain ledgers are replicated on different peer node which provides a consistent and accurate information across domains (or nodes) reducing the risk of fraud in data transaction [3].

As mentioned above that the blockchain is mostly used in crypto currencies because it protects the system from double spending therefore this technology is considered more suitable for e voting process.

2.3 Role of Blockchain in E-Voting:

In recent years, many organization and research communities are experimenting to use blockchain for serious business. E-voting is one of them. A well-known paper in this area is [6] in which Verify-Your-Vote (VYV) process is explained. It is an online electronic voting protocol that uses blockchain, Elliptic-Curve Cryptography (ECC), and Identity Based Encryption (IBE) for providing eligibility, fairness, robustness, data integrity; verifiability and vote-privacy. In this whole process, a Blockchain technology is used as a trusted web server, (peer to peer, decentralized) a public bulletin board to display all the public votes to ensure the verifiability. Integrity is achieved due to immutability property of blockchain. Four main entities Registration Server, Election Administrator, Eligible Voters, and Tallying Authority provides verification of vote's secrecy, voter's authentication and verification of vote's privacy. It has main four modules. (i) **Registration Server** registers eligible voters after authentication and provides parameters to the users for further use. (ii) **Election Administrator** does several tasks like defining time for each phase, authenticating voters and constructing ballots. (iii) **Eligible Voters** can vote many times until voting time ends but only last vote will be counted. (iv) **Tallying Authority** constructs ballots, decrypt votes, calculate final result and publish all of the values that are public to voters for the purpose of verifiability. The author of paper [7] also discussed the usage of blockchain in e-voting process. In this they proposed a solution which does not rely on centralized trusted authority and defined their own consensuses algorithms for voting. The process is same as mentioned in [6] but the algorithms are different.

Chapter 3: Requirement Gathering and Analysis

Introduction

The purpose of requirement gathering and analysis is to clear the requirements of the system and decide what the system should do and what the system should not do.

3.1 Product Overview

E-voting for smart city is an online voting system which is used to cast a vote. The whole system is designed on the concepts of web services and blockchain. The system will be implemented using blockchain software. The system can be used only for voting purpose. User will be able to cast a vote using a coin (token) which will be ether, if Ethereum is used otherwise relevant will be considered. The main voting system is decentralized so it will be on every system but the user\s registration and voting management process is centralized. User create their profiles. The account of the user contains user's data and public key. Private Key stays on user's system.

3.2 Major Functions

Major function of the system is to register citizen, voting process and counting of process. Mainly this project comprises three parts

1. **Pre-Process**
   - **Registration Process**
     As a prerequisite to voting process citizen must be registered,
   - **Assigned Areas.**
     Various geographically areas will be created and the candidates will be assigned to the areas.
   - **Announcement of Election**
     After that, election process will be announced.
2. **Main Process:** voting process in which citizen has to cast a vote which will be like a transaction from users account to selected candidate account but the user itself remain anonymous.

3. **Post Process**
   - **Counts Votes.**
     This function will count votes.
   - **Display results**
     This will display the results of the counting votes.

3.3 Major Inputs and outputs

The user gives some inputs to the system then the system will generate the response to those inputs respectively.

3.3.1 Major Inputs

Major input of the system is the client’s personal information and public key using which is digitally signed by using client’s private key. For actual voting process, client’s vote will be the second major input to the system.

3.3.2 Major Outputs

Major outputs of the system are the creation of digitally signed profiles of the citizens and result of the counting of votes.

3.4 Definitions, Acronyms and Abbreviations

Table 1: Definitions, Acronyms and Abbreviations

Smart Contracts	A smart contract, also known as a crypto contract, is a computer program that directly controls the transfer of digital currencies or assets between parties under certain conditions. A smart contract not only defines the rules and penalties related to an agreement in the same way that a traditional contract does, but it can also automatically enforce those obligations.
Blockchain	A digital ledger in which transactions are made for cryptocurrencies and these transactions are recorded chronologically and publicly.
Ethereum	Ethereum is an open source, distributed software platform and cryptocurrency built on the concept of blockchain technology. Blockchain is a distributed ledger technology (DLT) that keeps a permanent, immutable list of records.
Nonce	Based on the standard concept, nonce will be used with transaction to make it unique and fresh.
Solidity	Solidity is a contract-oriented programming language to write a smart contract.

3.5 Overview

This section contains topics that provide detailed information about the overall functionality of the system. The overall functionality comprises on functional and non-functional requirements, use cases and their description.

3.6 User Characteristics

As technology is evolving day-by-day and everyone is using it in their daily life. The purpose of making this system is to provide a platform to the citizen that will improve the quality of life of the citizens. Citizen will be able to cast a vote using this online e-voting system instead of going to polling station to cast a vote. For this, User has some basic knowledge of computer and laptop and also has a skill to operate web application. Moreover, citizen have some basic information about voting process which will help him to register themselves, create profiles and casting of a vote.

3.7 Constraint

E-voting for smart city will be a web-based system and the personal information of the user must be anonymous to avoid privacy threats. For achieving anonymity, we will use cryptographic techniques. Moreover to store selected candidate's information and citizen's personal information, we will deploy a database other than blockchain.

3.8 Assumption and Dependencies.

The web-based application depends upon the availability of the internet and having accounts in the system. It is assumed that user have computer or laptop connected to internet in order to access the system.

3.9 Specific Requirements

3.9.1 Functional Requirement

Functional requirements are the product abilities that must be available for the client in order to complete the services provided by the system and mention how the system should respond to the error conditions or invalid inputs. The system gives the proper message of any invalid entry. Main functional requirements of the system are:

1. Register User.
2. Verify User
3. Authenticate user.
4. Create Area.
5. View Area
6. Delete Area
7. Announce Election
8. Add Candidate
9. Update Candidate
10. View Candidate
11. Delete Candidate
12. Add Election
13. View Election
14. Update Election
15. Delete Election
16. Show candidates.
17. Poll vote.
18. Count Vote

3.9.2 External Interface Requirement

This section provides a description of all inputs into the system and outputs from the system. It also gives a description of the hardware, software and communication.

3.9.3 User Interfaces

User interfaces will help the user to interact with the system. E-voting will be Web-based application for the user. User can access this application through internet. User would have profiles to access this application and perform certain task.

3.10 Software Interfaces

E-voting for Smart City is decentralized web-based application and it will be implemented in REST API and Ethereum/Corda software, this system can run on any operating system. The internet is required to access the system. The system can be accessed through browsers like google chrome, Firefox.

3.11 Communication Protocol

Hypertext transfer protocol [http] is required by the system in order to communicate over the internet.

3.12 Non-Functional Requirements

3.12.1 Reliability

System should be reliable. There should be no occurrence of the failure. The system should be able to work properly all-time when the election process starts in country. The system should give the proper response to every query performed by user

3.12.2 Availability

System should be available to every user only when there is voting period in country to cast a vote, and for only getting information this system is available all the time. All the users should able to access the system at any time as it does not depend upon the centralized server.

3.12.3 Security

Since the system is decentralized, the majority of client's private information is on his PC. System should be based on public key cryptography. User should only be able to access the system through user own credentials and any other user should not be able to access the user private data.

3.12.4 Portability

Portability mainly deals with developing a web-based system. It can be improved by running on variety of platforms and connection speed. System should be lightweight so that it could be run on a machine with slow internet connection. To make the web application lightweight, simple libraries and tools should be used at developing phase

3.13 Use Case Diagram

With use case diagram, we are able to look more precisely that how user how user can interact with system to perform tasks.

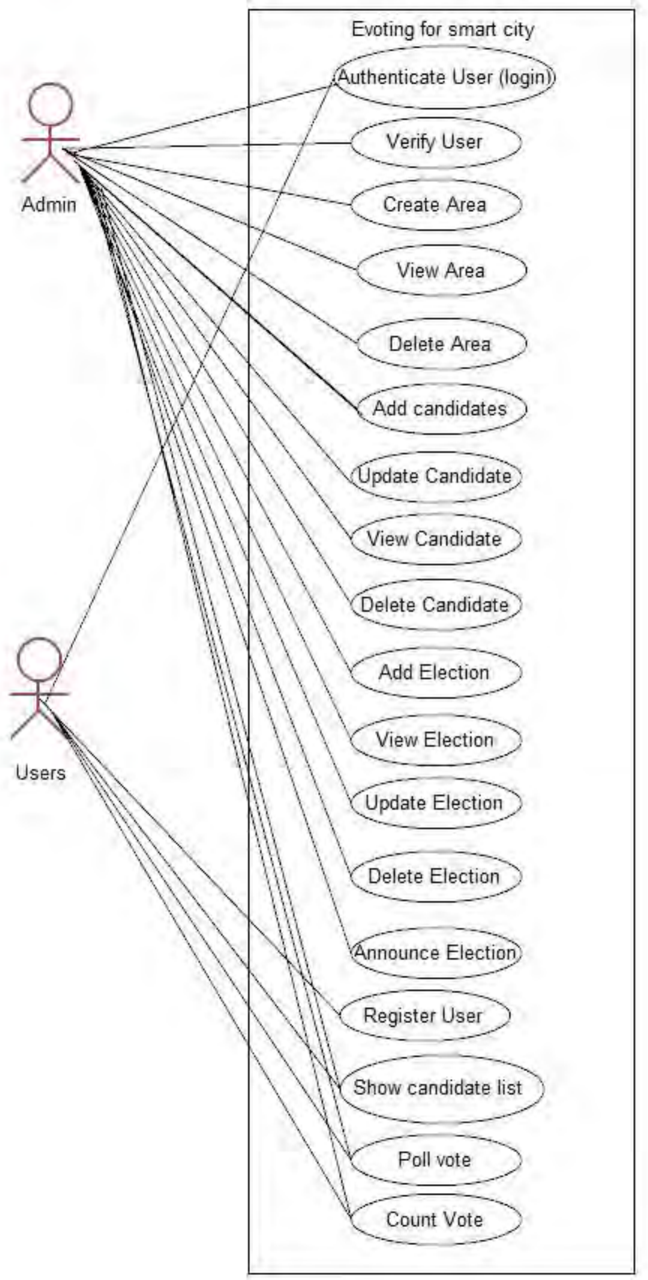


Figure 3.1: Use Case Diagram

3.13 Use Cases Description

3.13.1 Use case 1: Register User

Table 2: Register User

ID	**UC1**
Name	Register User.
Primary Actor	User.
Pre-Condition	User has to provide their personal information
Post-Condition	User data is stored in database and send to the admin for verification.
Main Success Scenario	1. User click sign up option. 2. System displays the sign-up form. 3. User enter required data in the fields 4. User click register button. 5. System send information to the admin for verification
Alternative flows or Extensions	Server and internet link down. 1. User waits until the internet and server recovered.
Frequency	Nearly Continuous till all users registered.

3.13.2 Use Case 2: Verify user against Double registration

Table 3: verify user against double registration

ID	**UC3**
Name	**Verify user against Double registration**
Primary Actor	Admin
Pre-Condition	Admin must be login to the account
Post-Condition	User is verified and has authorization to login to the account
Main Success Scenario	1. Admin click the view citizen option. 2. System displays list of users. 3. Admin click the verify button. 4. System displays message that the user is verified.
Alternative flows or Extensions	Server and internet link down. 1.Users' waits until the internet and server recovered 2.Admin enter incorrect credentials. 2a. admin enter correct username and password 3. No user with same CNIC is registered in system
Frequency	Nearly continues during elections.

3.13.3 Use Case 3: Authentication of users

Table 4: Authentication of *user*

ID	UC3
Name	Authentication of user
Primary Actor	Users, Admin
Pre-Condition	User open website.
Post-Condition	User is login to the system and automatically assigned address.
Main Success Scenario	1. User enter CNIC and password. 2. User click login button. 3. System display message "You are successfully login to your account "and assigned address.
Alternative flows or Extensions	Server and internet link down. 1.Users' waits until the internet and server recovered 2.Admin enter incorrect credentials. 2a. admin enter correct username and password
Frequency	Nearly continues during elections.

3.13.4 Use case 4: Create Area

Table 5: Create Area

ID	UC4
Name	Create Area
Primary Actor	Admin
Precondition	Admin must be login into the account.
Post Condition	Electoral Area is created.
Main Success Scenario	1. Admin click the area option. 2. System display area from 3. Admin enter the required information and join different postal code and create electoral areas. 4. System displays message "Successfully created Area" and display list created area to admin
Alternatives	Server and internet link down. 1. User waits until the internet and server recovered.
Frequency	Nearly continuous during election period.

3.13.5 Use Case 5: Announce Election

Table 6: Assign Candidates to Area

ID	UC 5
Name	Announce Election
Primary Actor	Admin
Pre-Condition	Admin must be login to the account.
Post Condition	Address of the ballot paper is generated.
Main Success Scenario	1. Admin click the announce election Option. 2. Admin click the deploy option. 3. System displays a form. 4. Admin select the election name from the drop down. 5. Admin click ok button. 6. System display the address of the ballot box.
Alternative flows or Extensions	Server and internet link down. 1. User waits until the internet and server recovered. Due to some reason list is not received by admin. 1. Admin request for list to election commission
Frequency	Once in five years

3.13.6 Use Case 6: View Area

Table 7: View Area

ID	UC 6
Name	View Area
Primary Actor	Admin
Pre-Condition	Admin must be login to the system
Post Condition	List of Area is displayed.
Main Success Scenario	1. Admin click the view area option. 2. System display the list of area created by admin.
Alternative flows or Extensions	Server and internet link down. 1. User waits until the internet and server recovered. Due to some reason list is not received by admin. 1. Admin request for list to election commission
Frequency	Nearly Continuous

3.13.7 Use Case 7: Delete Area

Table 8: Delete Area

ID	**UC 7**
Name	Delete Area
Primary Actor	Admin
Pre-Condition	Admin must be login to the system
Post Condition	Selected area will be deleted from the database.
Main Success Scenario	1. Admin click the delete area option. 2. System display the form. 3. Admin select the area from the drop-down list. 4. Admin click the save button. 5. System displays the message "Successfully Deleted".
Alternative flows or Extensions	Server and internet link down. 1. User waits until the internet and server recovered.
Frequency	Nearly continuous

3.13.8 Use Case 8: Add Election

Table 9: Add Election

ID	**UC 8**
Name	Add Election
Primary Actor	Admin
Pre-Condition	Admin must be login to the system
Post Condition	Selected area will be deleted from the database.
Main Success Scenario	1. Admin click the add election option. 2. System display the form. 3. Admin enter the required data in the field. 4. Admin click save button 5. System displays the message "Successfully Added".
Alternative flows or Extensions	Server and internet link down. 6. User waits until the internet and server recovered. Due to some reason list is not received by admin. 1. Admin request for list to election commission
Frequency	Nearly continuous

3.13.9 Use Case 9: View Election

Table 10: View Election

ID	**UC 9**
Name	View Election
Primary Actor	Admin
Pre-Condition	Admin must be login to the system
Post Condition	List of Election is displayed.
Main Success Scenario	1. Admin click the view election option. 2. System display the list of Election created by admin.
Alternative flows or Extensions	Server and internet link down. 1. User waits until the internet and server recovered.
Frequency	Nearly Continuous

3.13.10 Use Case 10: Update Election

Table 11: Update Election

ID	**UC 10**
Name	Update Election
Primary Actor	Admin
Pre-Condition	Admin must be login to the system
Post Condition	Selected area will be deleted from the database.
Main Success Scenario	1. Admin click the update election option. 2. System display the form. 3. Admin select the election which he wants to update. 4. Admin click ok button 5. System display another form with data. 6. Admin change the value of the required fields. 7. Admin click the save button. 8. System displays the message “Successfully Updated”.
Alternative flows or Extensions	Server and internet link down. 1. User waits until the internet and server recovered.
Frequency	Nearly continuous

3.13.11 Use Case 11: Delete Election

Table 12: Delete Election

ID	UC 11
Name	Delete Election
Primary Actor	Admin
Pre-Condition	Admin must be login to the system
Post Condition	Selected area will be deleted from the database.
Main Success Scenario	1. Admin click the delete election option. 2. Admin select the area from the drop-down list. 3. System displays the message "Successfully Deleted".
Alternative flows or Extensions	Server and internet link down. 1. User waits until the internet and server recovered.
Frequency	Nearly continuous

3.13.12 Use Case 12: Add Party Candidate

Table 13: Add Party Candidate

ID	UC 12
Name	Add Party Candidate
Primary Actor	Admin
Pre-Condition	Admin must be login to the system
Post Condition	Selected area will be deleted from the database.
Main Success Scenario	1. Admin click add party candidate option. 2. System display the form. 3. Admin enter the required data in the field. 4. Admin click save button 5. System displays the message "Successfully Added".
Alternative flows or Extensions	Server and internet link down. 1. User waits until the internet and server recovered. Due to some reason list is not received by admin. 1. Admin request for list to election commission
Frequency	Nearly continuous

3.13.13 Use Case 13: View Party Candidate

Table 14: View Party Candidate

ID	**UC 13**
Name	View Election
Primary Actor	Admin
Pre-Condition	Admin must be login to the system
Post Condition	List of Election is displayed.
Main Success Scenario	1. Admin click the view election option. 2. System display the list of Election created by admin.
Alternative flows or Extensions	Server and internet link down. 1. User waits until the internet and server recovered.
Frequency	Nearly Continuous

3.13.14 Use Case 14: Update Election

Table 15: Update Election

ID	**UC 14**
Name	Update Party Candidate
Primary Actor	Admin
Pre-Condition	Admin must be login to the system
Post Condition	Selected area will be deleted from the database.
Main Success Scenario	1. Admin click the update party candidate option. 2. System display the form. 3. Admin select the name of the candidate which he wants to update. 4. Admin click ok button 5. System display another form with data. 6. Admin change the value of the required fields. 7. Admin click the save button. 9. System displays the message “Successfully Updated”.
Alternative flows or Extensions	Server and internet link down. 1. User waits until the internet and server recovered.
Frequency	Nearly continuous

3.13.15 Use Case 15: Delete Party candidate

Table 16: Delete Party candidate

ID	**UC 15**
Name	Delete Party Candidate
Primary Actor	Admin
Pre-Condition	Admin must be login to the system
Post Condition	Selected area will be deleted from the database.
Main Success Scenario	1. Admin click the delete candidate option. 2. System display the form. 3. Admin select the candidate from the drop-down list. 4. Admin click save button. 5. System displays the message "Successfully Deleted".
Alternative flows or Extensions	Server and internet link down. 6. User waits until the internet and server recovered.
Frequency	Nearly continuous

3.13.16 Use Case 16: Show Candidate list

Table 17: Show Candidate List

ID	**UC16**
Name	Show candidate list
Primary Actor	User
Pre-Condition	User must be authenticated
Post-Condition	Candidates list is displayed.
Main Success Scenario	1. User will authenticate their self. 2. System will check postal code of the user. 3. System will match user's postal code with area group and find area group id. 4. Against area group, all candidates list will be displayed.
Alternative flows or Extensions	1a. If user is not authenticated 1. System will display a message "You are not an authenticated user". 3a. If postal is not matched with area group 1. System will display a message "Your postal code is incorrect ".
Frequency	Nearly Continuous during election period.

3.13.17 Use Case 17: Poll Vote

Table 18: Poll Vote

ID	**UC17**
Name	Poll vote
Primary Actor	User
Pre-Condition	User has to login in to account.
Post-Condition	User casted a vote to a selected candidates.
Main Success Scenario	1. User login into their account. 2. User will select assembly. 3. System display candidate list against assembly. 4. User will select a candidate to whom user wants to cast a vote. 5. Vote generator (voteGen) will generate a coin which will represent actually a vote of a user and address of Ballot Box. 6. User digitally sign vote and selected candidate and then make a transaction to Ballot Box.
Alternative flows or Extensions	1a. if user is not successfully login into account. 1. System will display a message "unauthorized User". 2a. if user again request to voteGen for vote, 1. System will display a message "Sorry!! Already issued a vote".
Frequency	Nearly Continuous during election period.

3.13.18 Use case 18: Count Vote

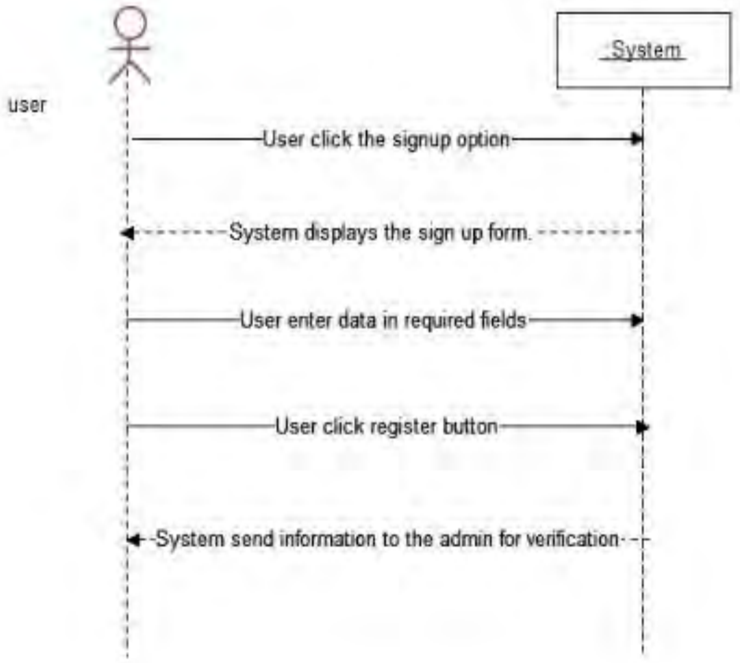
Table 19: Count Vote

ID	UC18
Name	Count Vote
Primary Actor	Admin
Pre-Condition	Admin must be login into account.
Post-Condition	Result will be displayed in forms of charts.
Main Success Scenario	1. Admin click the result option. 2. Admin click the count vote option. 3. System display two options bar chart and pie chart. 4. Admin select bar chart. 5. Result is displayed in form the form of charts.
Frequency	Nearly Continuous during election period.

3.14 System Sequence Diagram

A system sequence diagram is fast and easily created artifact that illustrates input and output events related to the system. System behavior is a description of what the system does, without explaining how it does.

3.14.1 Register user



user
:System
User click the signup option
System displays the sign up form.
User enter data in required fields
User click register button
System send information to the admin for verification

Figure 3.2: Register user System Sequence Diagram

3.14.2 Authenticate user

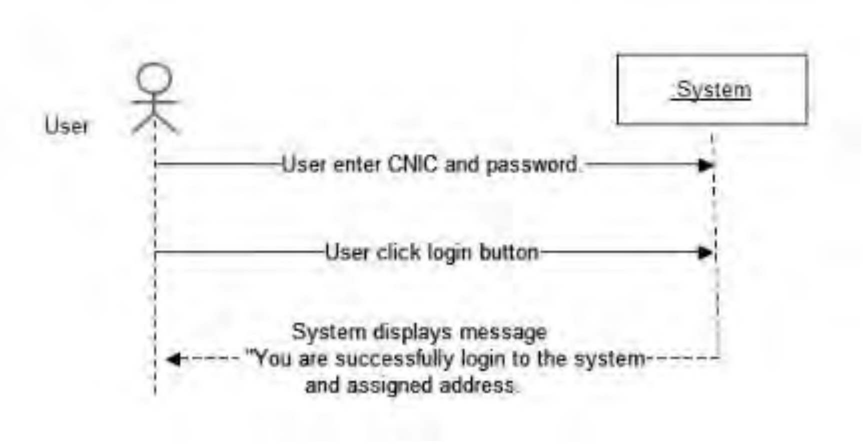


Figure 3.3: Authenticate user System Sequence Diagram

3.14.3 Create Area

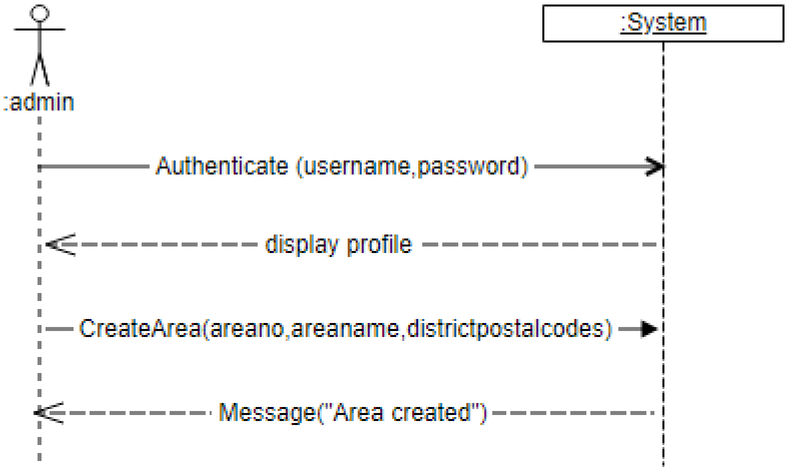

:System
:admin
Authenticate (username,password)
display profile
CreateArea(areano,areaname,districtpostalcodes)
Message("Area created")

Figure 3.4: Create Area System Sequence Diagram

3.14.4 Assign Candidates to Area

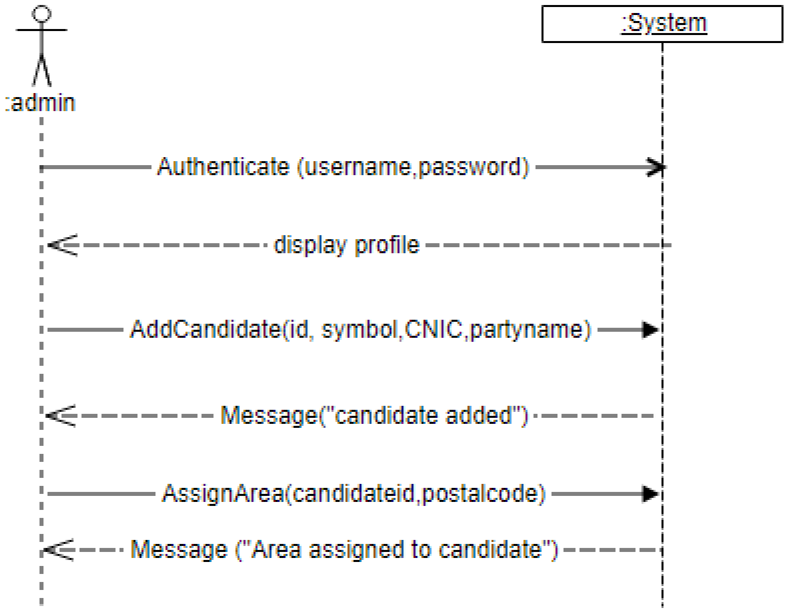

:admin
:System
Authenticate (username,password)
display profile
AddCandidate(id, symbol,CNIC,partyname)
Message("candidate added")
AssignArea(candidateid,postalcode)
Message ("Area assigned to candidate")

Figure 3.5: Assign candidates to area System Sequence Diagram

3.14.5 Add Candidate

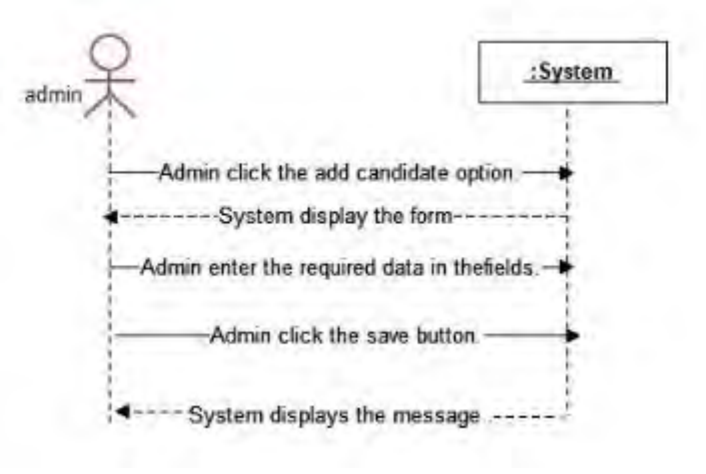

admin
:System
Admin click the add candidate option
System display the form
Admin enter the required data in thefields.
Admin click the save button
System displays the message

Figure 3.6: Add candidate System Sequence Diagram

3.14.6 Add Election

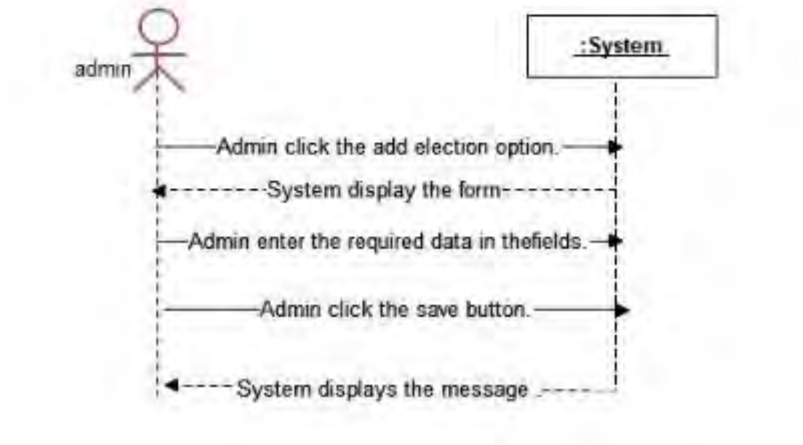

:System
admin
Admin click the add election option.
System display the form
Admin enter the required data in thefields.
Admin click the save button.
System displays the message

Figure 3.7: Add Election System Sequence Diagram

3.14.7 Update Candidate

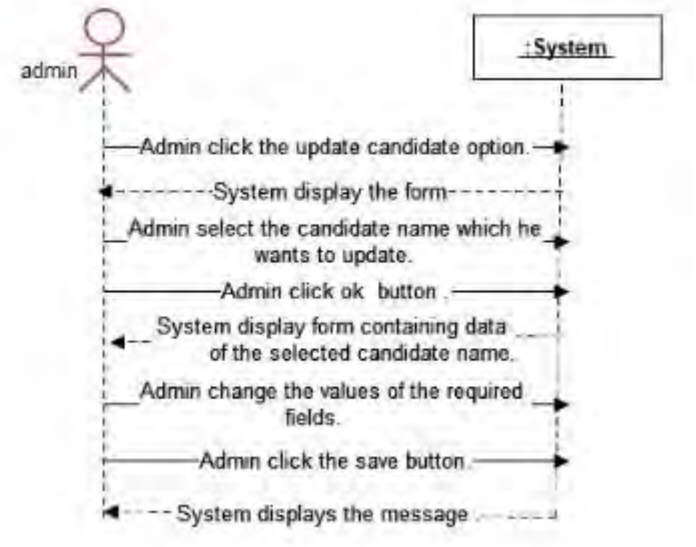

:System
admin
Admin click the update candidate option.
System display the form
Admin select the candidate name which he wants to update.
Admin click ok button
System display form containing data of the selected candidate name.
Admin change the values of the required fields.
Admin click the save button.
System displays the message

Figure 3.8: Update Candidate System Sequence Diagram

3.14.8 Update Election

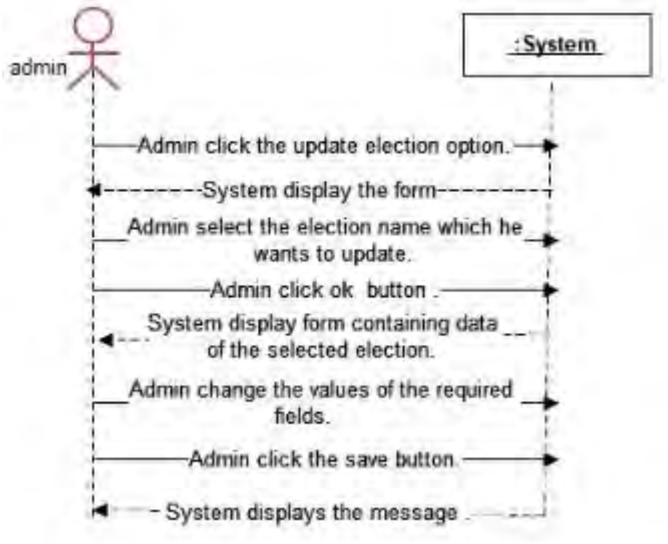

admin
:System
Admin click the update election option.
System display the form
Admin select the election name which he wants to update.
Admin click ok button .
System display form containing data of the selected election.
Admin change the values of the required fields.
Admin click the save button.
System displays the message .

Figure 3.9: Update election System Sequence Diagram

3.14.9 View Candidate

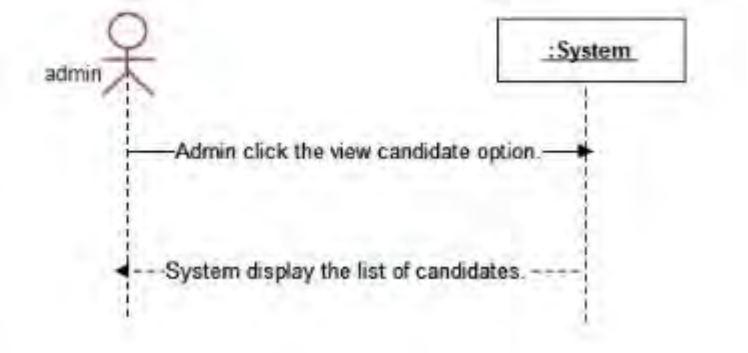

admin
:System
Admin click the view candidate option.
System display the list of candidates.

Figure 3.10: View Candidate System Sequence Diagram

3.14.10 View Area

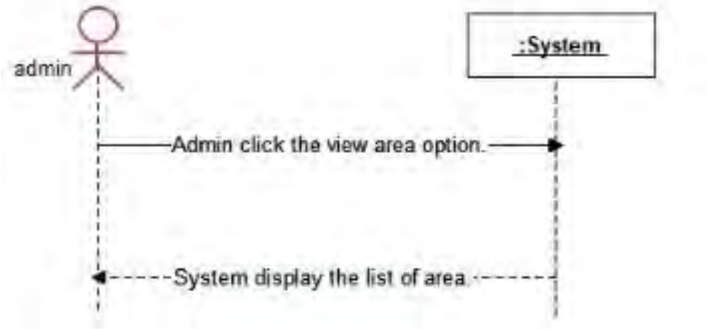

admin
:System
Admin click the view area option.
System display the list of area

Figure 3.11: View area System Sequence Diagram

3.14.11 View Election

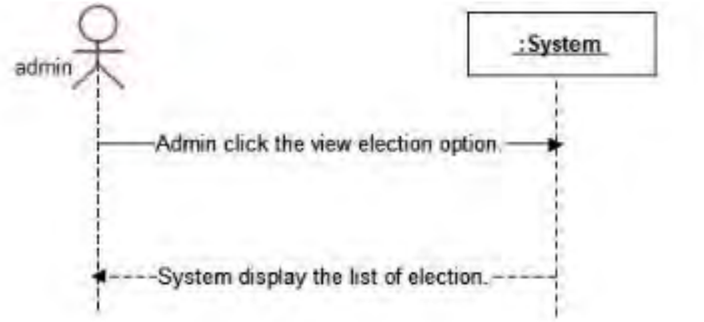

admin
:System
Admin click the view election option.
System display the list of election.

Figure 3.12: View Election System Sequence Diagram

3.14.12 Poll vote

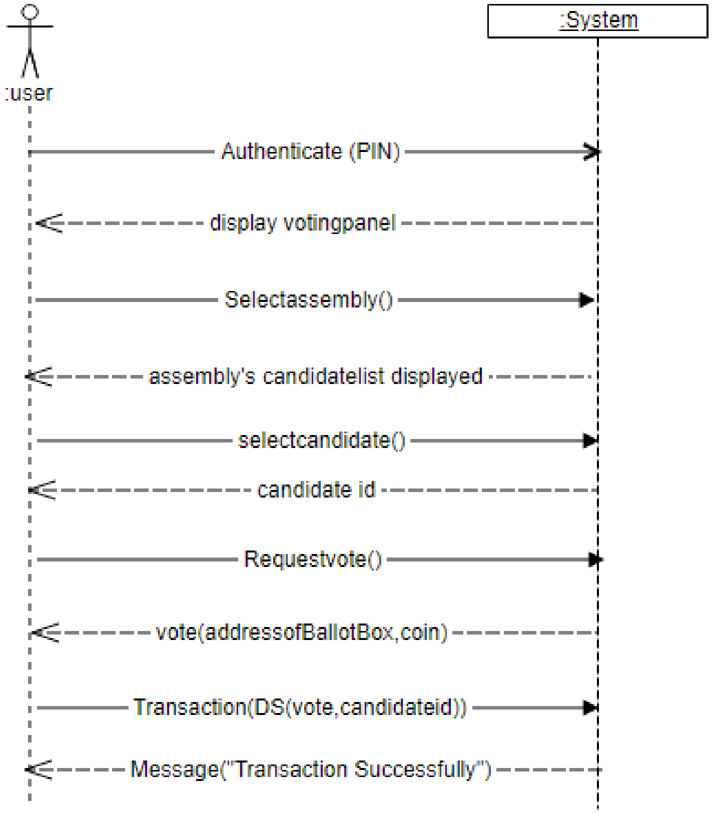

:user
:System
Authenticate (PIN)
display votingpanel
Selectassembly()
assembly's candidatelist displayed
selectcandidate()
candidate id
Requestvote()
vote(addressofBallotBox,coin)
Transaction(DS(vote,candidateid))
Message("Transaction Successfully")

Figure 3.13: Poll vote System Sequence Diagram

3.14.13 Delete Candidate

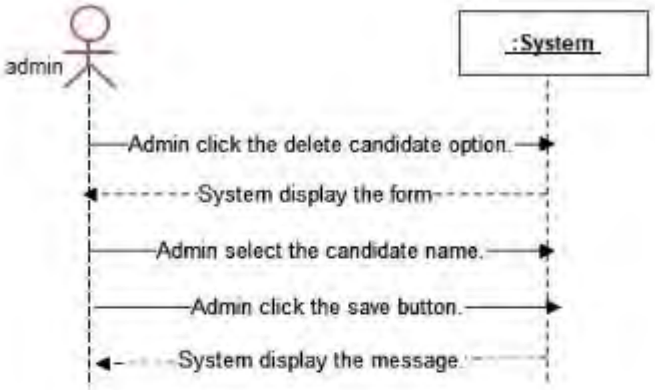

admin
:System
Admin click the delete candidate option.
System display the form
Admin select the candidate name.
Admin click the save button.
System display the message.

Figure 3.14: Delete candidate System Sequence Diagram

3.14.14 Delete Area

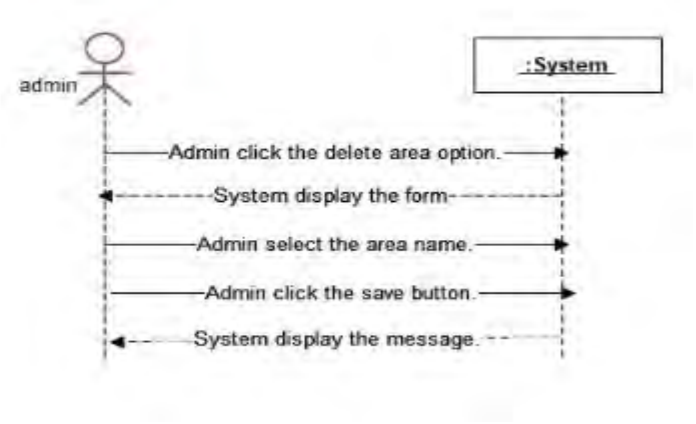

admin
:System
Admin click the delete area option.
System display the form
Admin select the area name.
Admin click the save button.
System display the message.

Figure 3.15: Delete area System Sequence Diagram

3.14.15 Delete Election

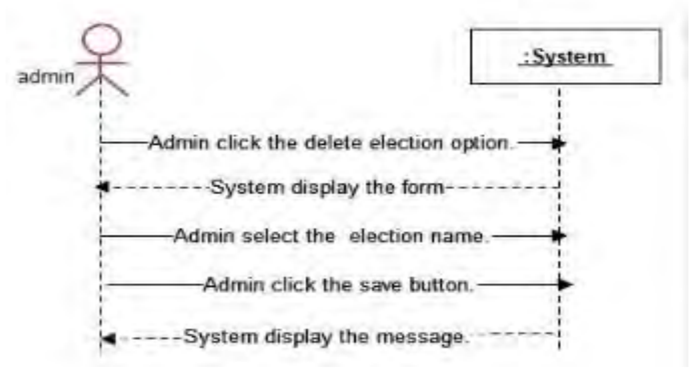

admin
:System
Admin click the delete election option.
System display the form
Admin select the election name.
Admin click the save button.
System display the message.

Figure 3.16: Delete election System Sequence Diagram

3.14.16 Count Vote

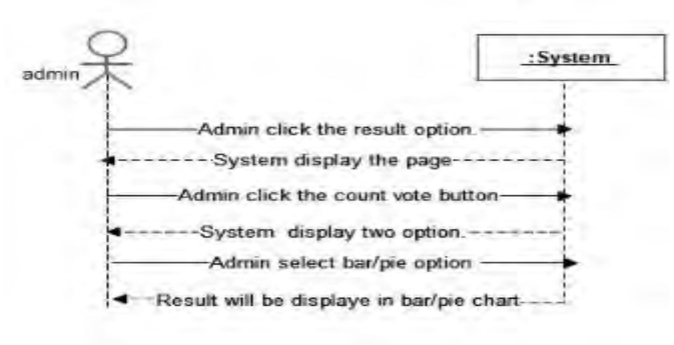

admin
:System
Admin click the result option.
System display the page
Admin click the count vote button
System display two option.
Admin select bar/pie option
Result will be displaye in bar/pie chart

Figure 3.17: Count votes System Sequence Diagram

3.15 Domain Model

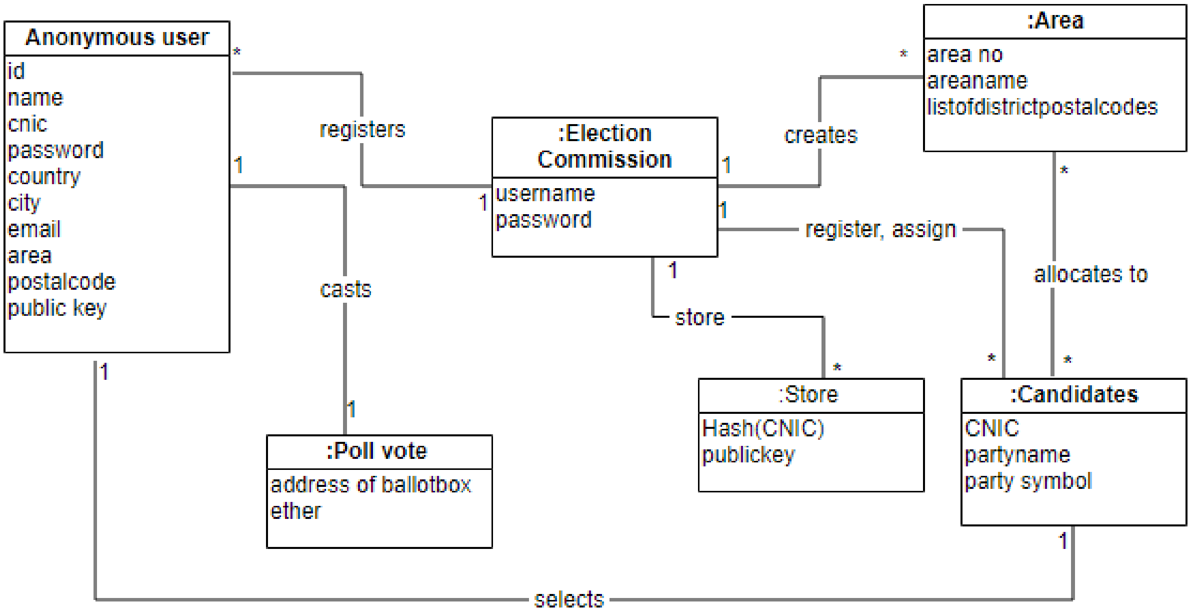


Figure 3.18: Domain Model

3.16 Database Requirements:

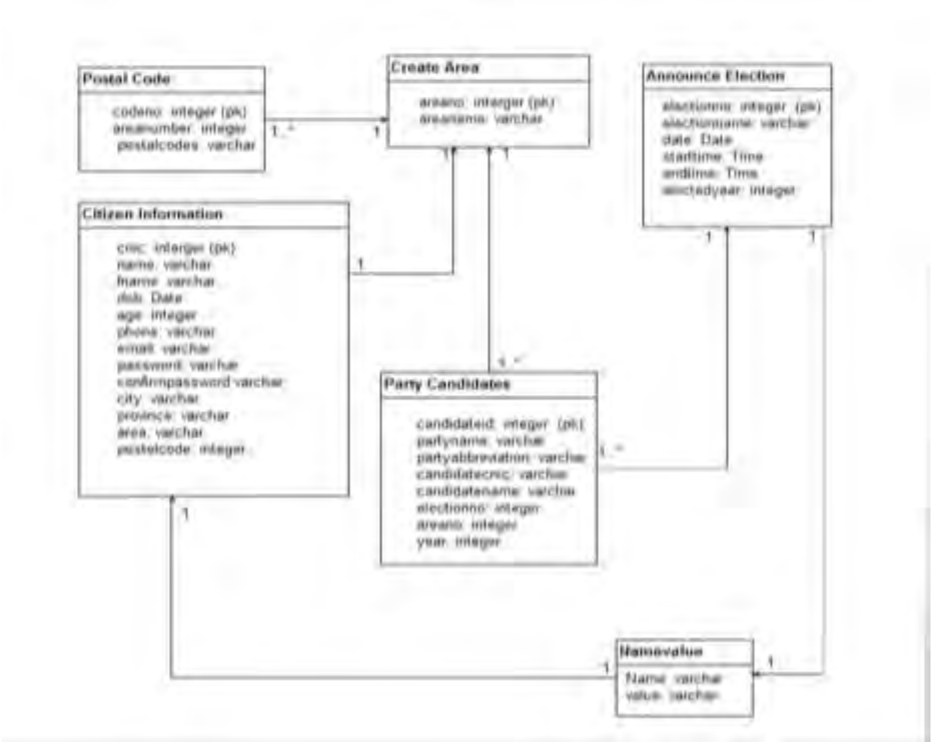


Figure 3.19: ERD

Chapter 4: Software Design Description

Introduction

This chapter gives the complete description of software design. This section explains the design of user interface and interactions diagrams.

4.1 Introduction

Software design description is a representation of the software design. Software design is used for communication information to its users. It indicates how framework will be organized to fulfill the requirements. The software design description comprises of two stages. First stage defines the system architecture and data architecture. Second stage define the detailed design and algorithms and codes are developed for defined architecture.

4.1.1 Design Overview

Software design is an iterative process. It is used to translate requirements into blueprints for developing the system. It represents how end user will interact with the system. Design begins with the requirement model and for clarity, correctness and consistence with requirements software products are reviewed. Requirements are translated clearly through designing sequence diagrams, class diagram and user interfaces interactions.

4.1.2 Requirement Traceability Matrix

Table 20: Requirement Traceability matrix

Requirement ID	Requirement Name	Sequence Diagram	Interface
UC1	Register User	Yes	Yes
UC2	Verify user against double registration	No	No
UC3	Authentication of user	Yes	No
UC4	Create Area	Yes	Yes
UC5	Announce Election	Yes	Yes
UC6	Show Candidate list	No	Yes
UC7	Poll vote	Yes	Yes
UC8	Add Area	Yes	Yes
UC9	View Area	Yes	Yes
UC10	Delete Area	Yes	Yes
UC11	Add candidate	Yes	Yes
UC12	View Candidate	Yes	Yes
UC13	Update Candidate	Yes	Yes
UC14	View Candidate	Yes	Yes
UC15	Add Election	Yes	Yes

UC16	View Election	Yes	Yes
UC17	Update Election	Yes	Yes
UC 18	View Election	Yes	Yes

4.2 System Architecture Design

Architecture diagram shows how the system works. The architecture for this project is shown in fig

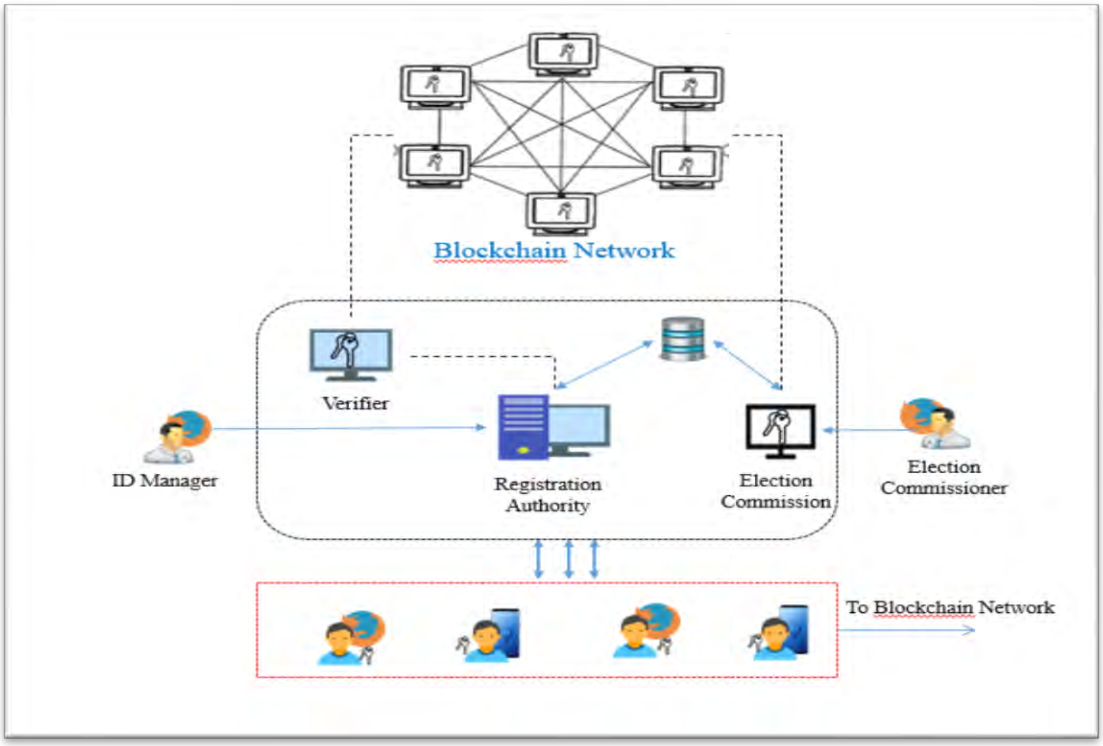


Figure 4.1: System Architecture

4.3 User Interface Design

User interface creates communication between user and computer. User interface design begins with the identification of user, task, and environmental requirements.

4.3.1 Admin Login Interface

Admin need username and password to login to their profile.

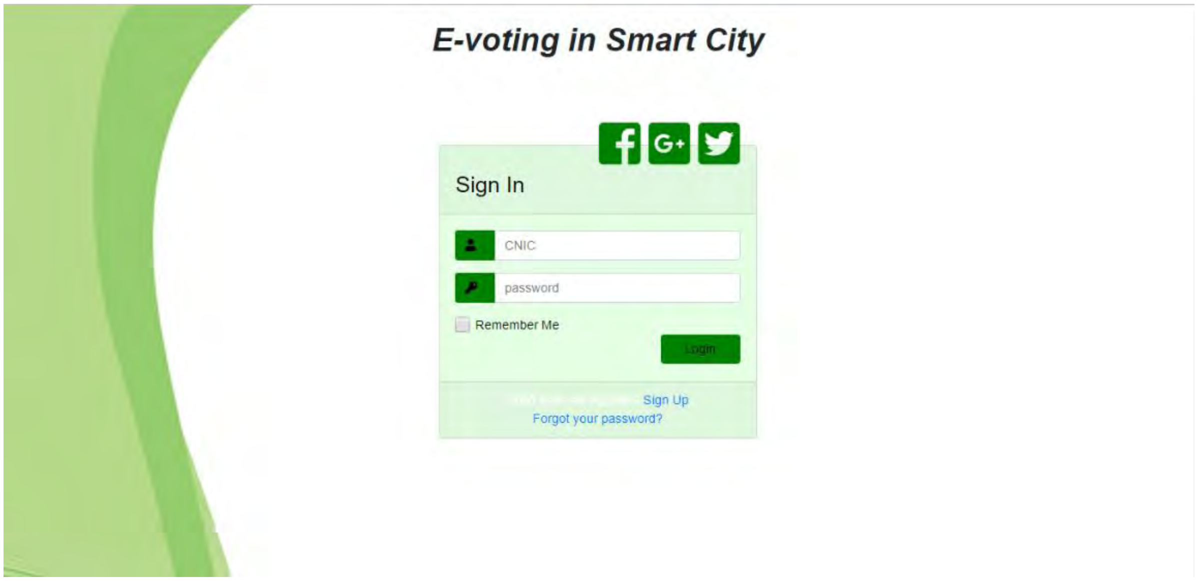
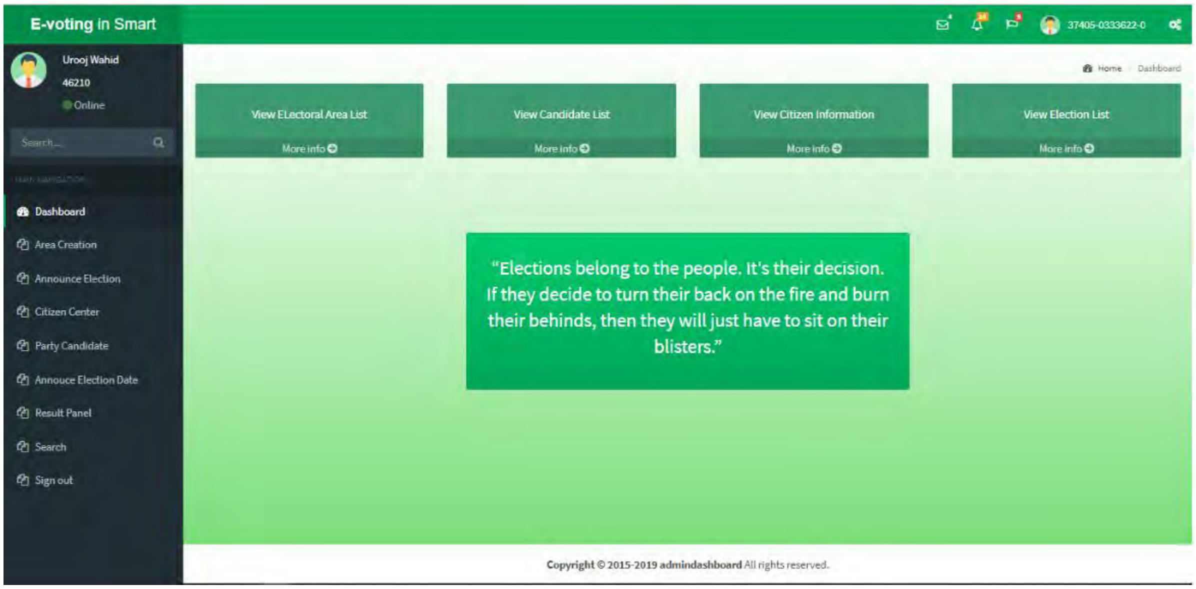


Figure 4.2: Admin login Interface

4.3.2 Admin Dashboard Interface

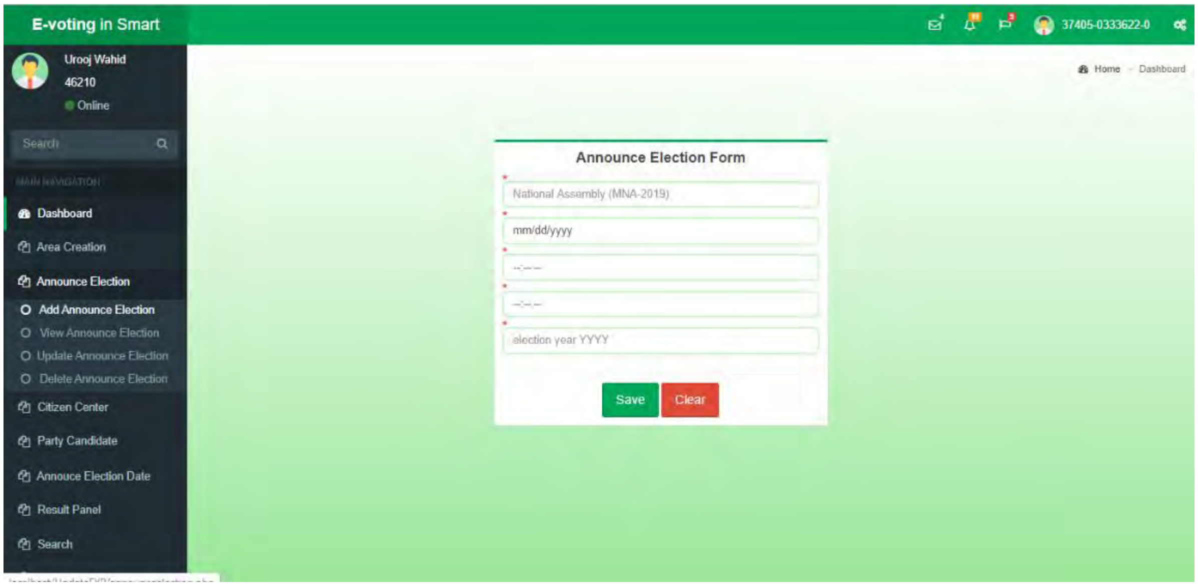


"Elections belong to the people. It's their decision. If they decide to turn their back on the fire and burn their behinds, then they will just have to sit on their blisters."

Copyright © 2015-2019 admindashboard All rights reserved.

Figure 4.3: Admin dashboard Interface

4.3.3 Announce Election Interface



Announce Election Form

National Assembly (MNA-2019)

mm/dd/yyyy

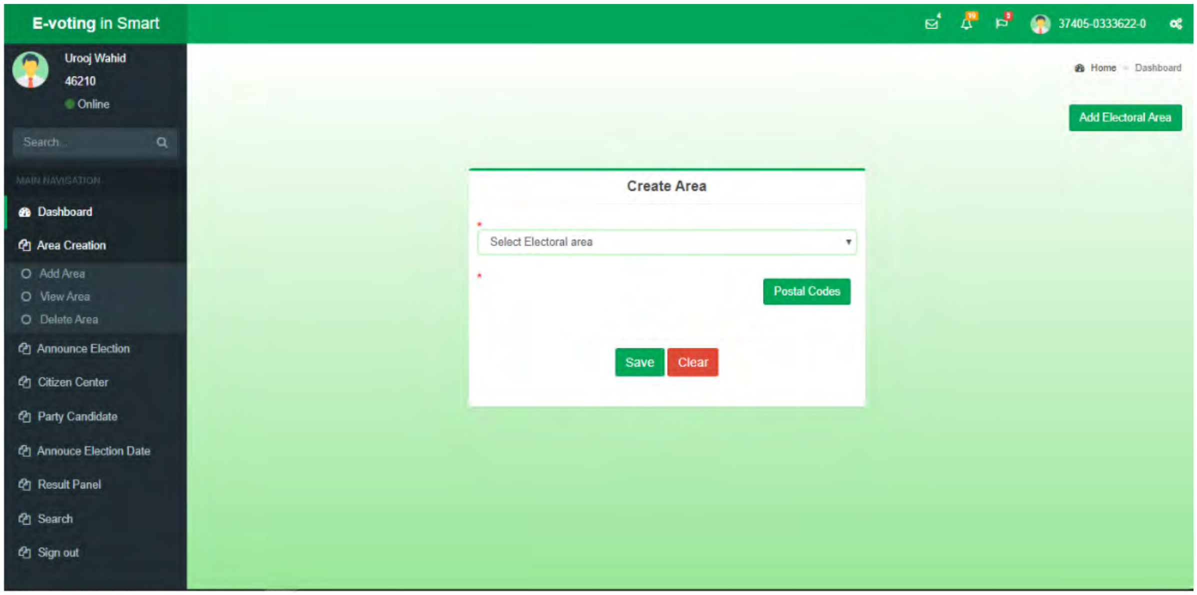
--:-- --

--:-- --

election year YYYY

Figure 4.4: Announce Election Interface

4.3.4 Electoral Area Interface



Create Area

Select Electoral area

Figure 4.5: Electoral Area Interface

4.3.5 Register Candidate

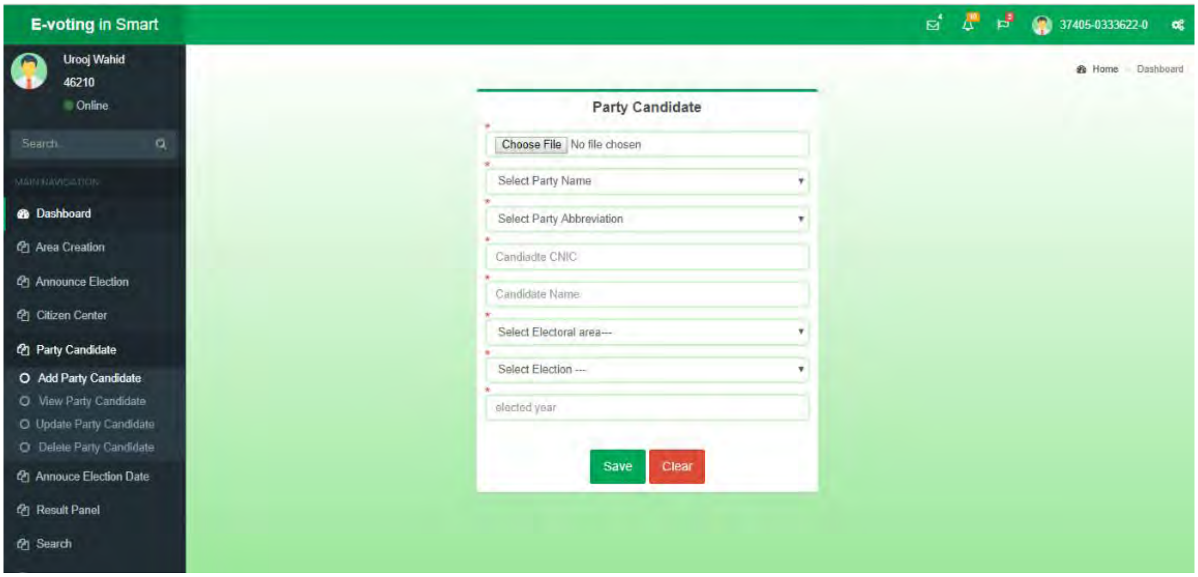


Figure 4.6: Register Candidate Interface

4.3.6 Main Website Interface

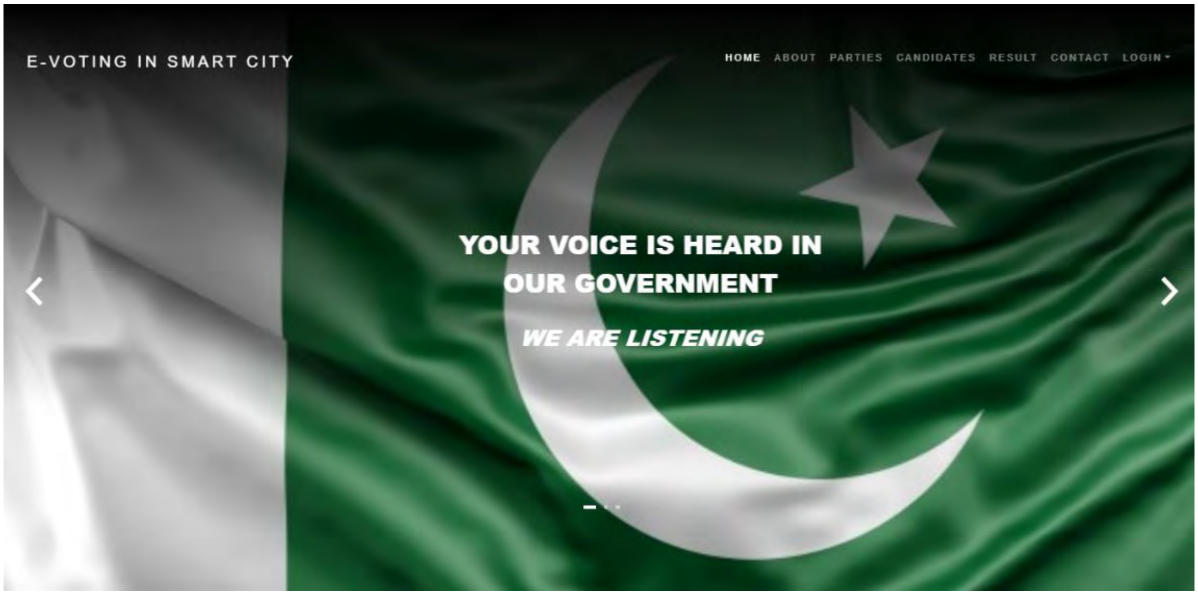
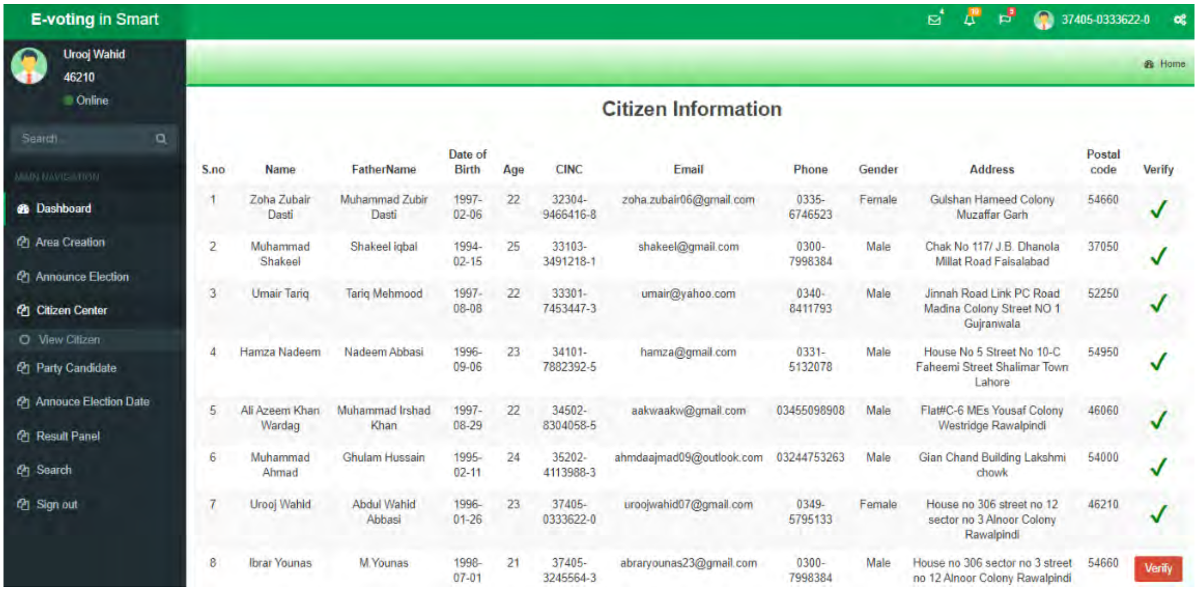


Figure 4.7: Main Website Interface

4.3.7 View Citizen

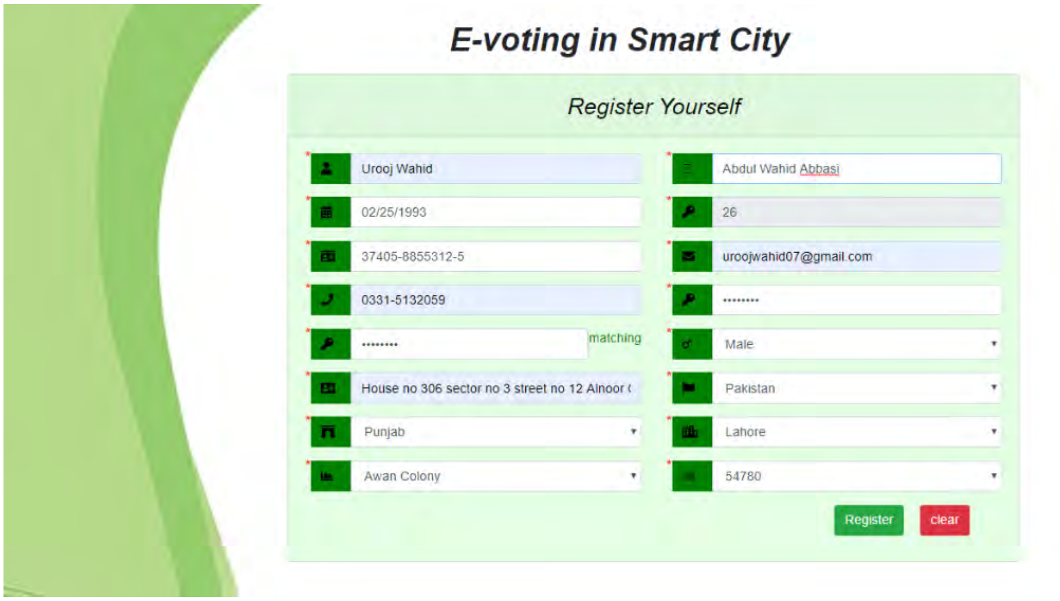


Citizen Information

S.no	Name	FatherName	Date of Birth	Age	CINC	Email	Phone	Gender	Address	Postal code	Verify
1	Zoha Zubair Dasti	Muhammad Zubir Dasti	1997-02-06	22	32304-9466416-8	zoha.zubair06@gmail.com	0335-6746523	Female	Gulshan Hameed Colony Muzaffar Garh	54660	✓
2	Muhammad Shakeel	Shakeel iqbal	1994-02-15	25	33103-3491218-1	shakeel@gmail.com	0300-7998384	Male	Chak No 117/ J.B. Dhanola Millat Road Faisalabad	37050	✓
3	Umair Tariq	Tariq Mehmood	1997-08-08	22	33301-7453447-3	umair@yahoo.com	0340-8411793	Male	Jinnah Road Link PC Road Madina Colony Street NO 1 Gujranwala	52250	✓
4	Hamza Nadeem	Nadeem Abbasi	1996-09-06	23	34101-7882392-5	hamza@gmail.com	0331-5132078	Male	House No 5 Street No 10-C Faheemi Street Shalimar Town Lahore	54950	✓
5	Ali Azeem Khan Wardag	Muhammad Irshad Khan	1997-08-29	22	34502-8304058-5	aakwaakw@gmail.com	03455098908	Male	Flat#C-6 MEs Yousaf Colony Westridge Rawalpindi	46060	✓
6	Muhammad Ahmad	Ghulam Hussain	1995-02-11	24	35202-4113988-3	ahmdaajmad09@outlook.com	03244753263	Male	Gian Chand Building Lakshmi chowk	54000	✓
7	Urooj Wahid	Abdul Wahid Abbasi	1996-01-26	23	37405-0333622-0	uroojwahid07@gmail.com	0349-5795133	Female	House no 306 street no 12 sector no 3 Alnoor Colony Rawalpindi	46210	✓
8	Ibrar Younas	M.Younas	1998-07-01	21	37405-3245564-3	abraryounas23@gmail.com	0300-7998384	Male	House no 306 sector no 3 street no 12 Alnoor Colony Rawalpindi	54660	Verify

Figure 4.8: view Citizen

4.3.8 User Registration Form



E-voting in Smart City

Register Yourself

Urooj Wahid	Abdul Wahid Abbasi
02/25/1993	26
37405-8855312-5	uroojwahid07@gmail.com
0331-5132059	••••••••
•••••••• matching	Male
House no 306 sector no 3 street no 12 Alnoor (	Pakistan
Punjab	Lahore
Awan Colony	54780

Figure 4.9: Registration Form interface

4.4 Sequence Diagram

Sequence diagram describes the interactions between the actors and the objects in a system and the object themselves. It shows the objects participating in the interaction by their lifelines and the messages that they send each other

4.4.1 Register user

The user sends digitally signed public key and personal information (name, father name, area, country, date of birth…etc.) to the registration authority where registration authority extracts CNIC and take hash of it and digitally signed hash and public key and send to the verifier. A verifier then adds the user in the blockchain if user is not already registered and return hash and public key back to registration authority and registration authority then saves all in the information in the local database and create profile and return it to the user as shown in figure.

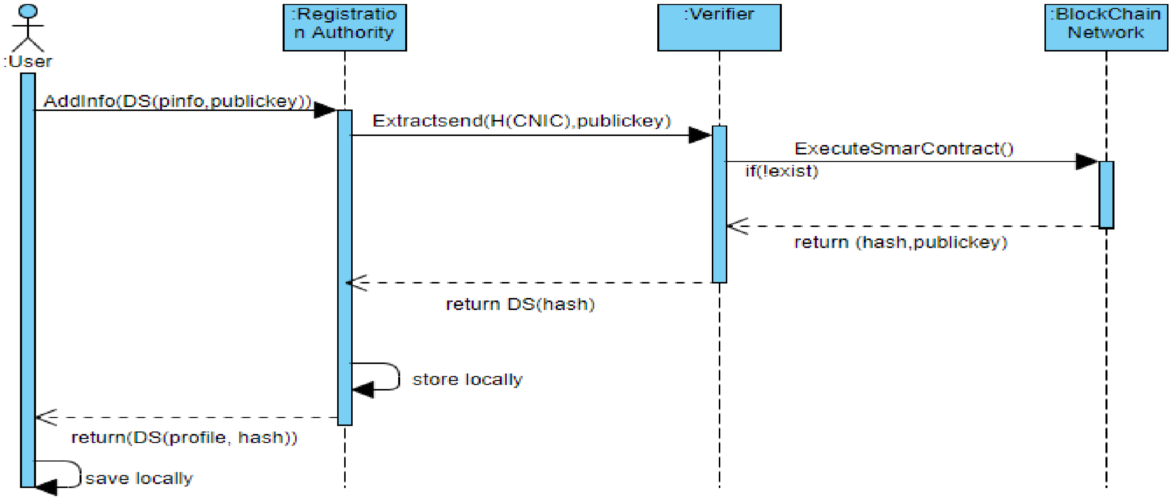


Figure 4.10: Sequence Diagram Register user

4.4.2 Authentication of user

User generate random number called nonce to avoid duplication and send digitally signed nonce and hash to election commission. Election commission extract information and checks in the blockchain and return a verification message and display profile as shown in figure.

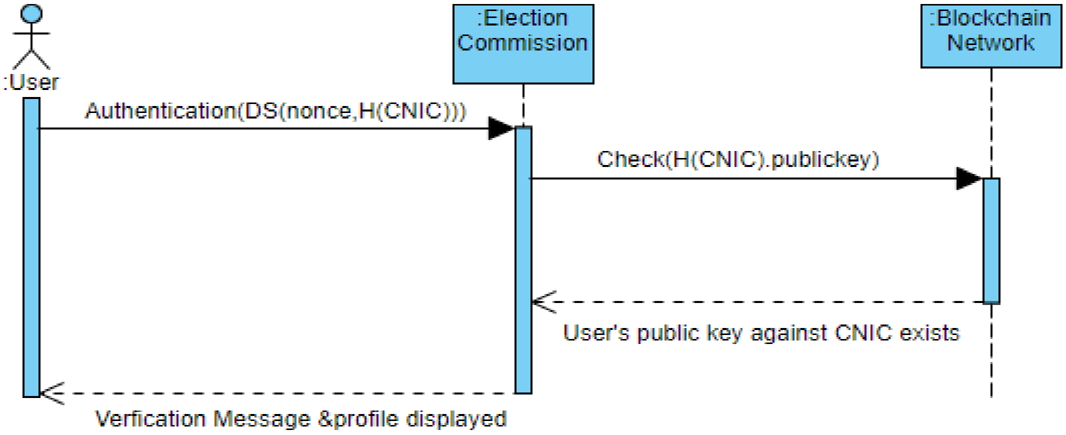

:User
:Election
Commission
:Blockchain
Network
Authentication(DS(nonce,H(CNIC)))
Check(H(CNIC).publickey)
User's public key against CNIC exists
Verfication Message &profile displayed

Figure 4.11: Sequence Diagram Authenticate User

4.4.3 Assign Candidate to Area

Admin authenticate himself and the assign area to the selected candidate for the election commission.

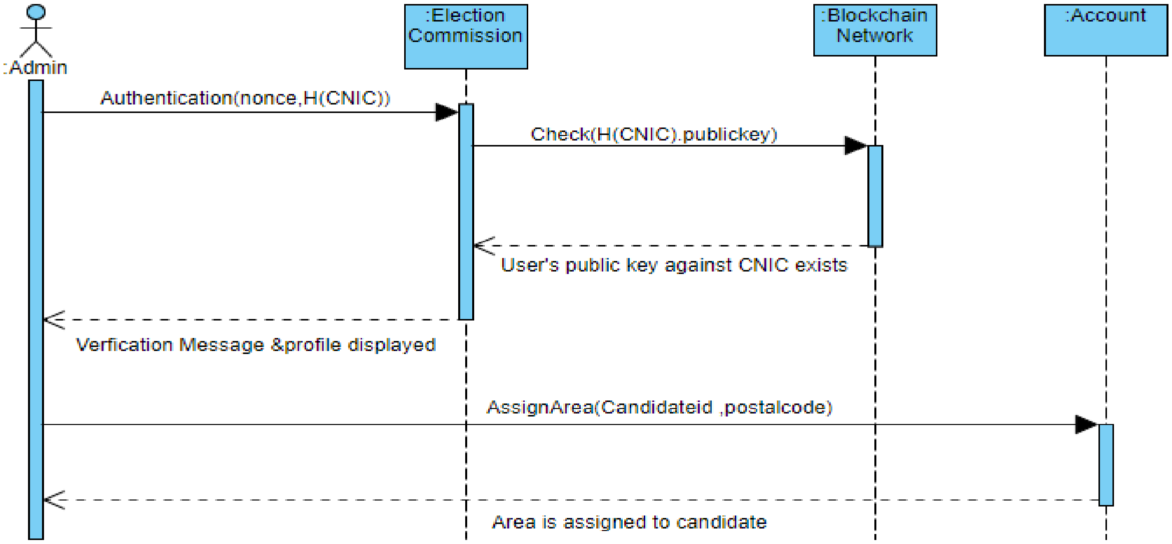

:Admin
:Election Commission
:Blockchain Network
:Account
Authentication(nonce,H(CNIC))
Check(H(CNIC).publickey)
User's public key against CNIC exists
Verfication Message &profile displayed
AssignArea(Candidateid ,postalcode)
Area is assigned to candidate

Figure 4.12: Sequence Diagram Assign candidates to Area

4.4.4 Poll Vote

User will authenticate their selves and will cast a vote as shown in figure

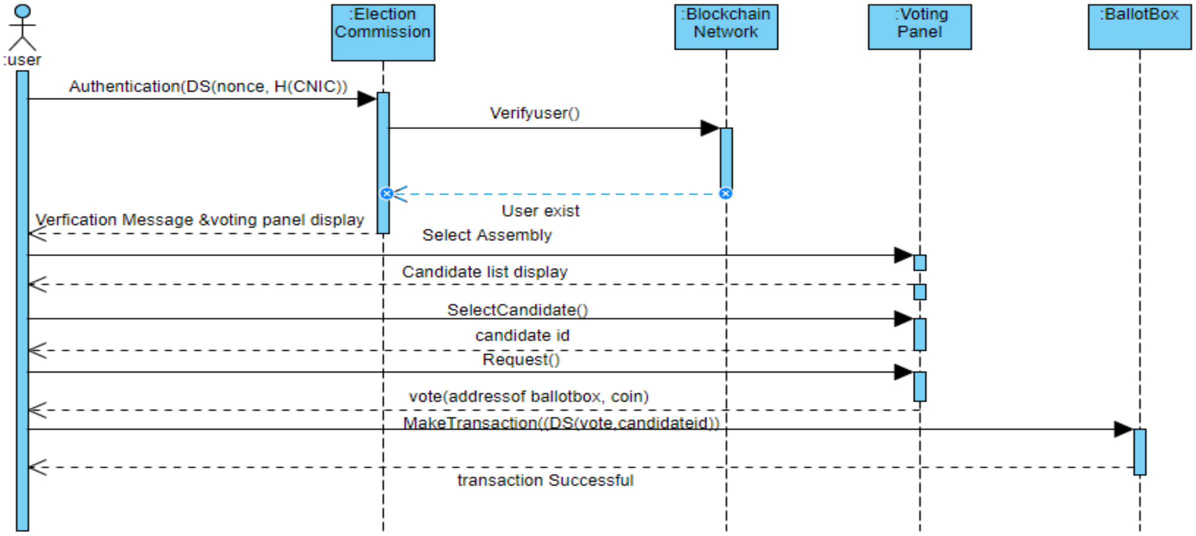


Figure 4.13: Sequence Diagram Poll Vote

4.5 Class Diagram

Smart contracts are used instead classes. Smart contracts use object-oriented approach as class. Programming language used in this project is solidity which is a contract-oriented language. Concepts used in contract-oriented programming are almost the same as they are in object-oriented programming. In this case we will use class diagrams for showing smart contracts on paper and modelling the static view of application.

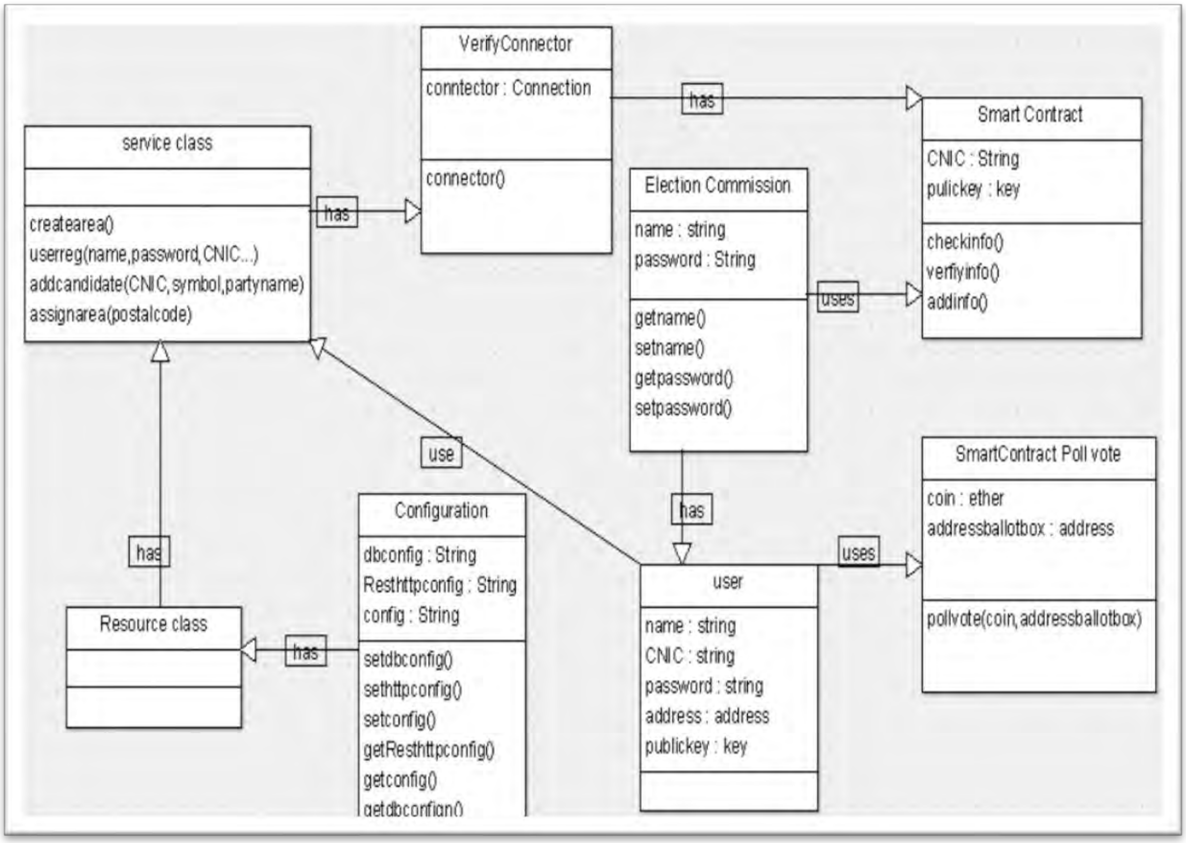


Figure 4.14: Class diagram

This chapter covers the complete description of software design. It has provided a detailed description regarding architecture design, components of the system and user interface description. Finally, interaction between the object and human actor are shown by interaction diagram and relationship between the instances is shown by class diagram.

Chapter 5: Software Testing

This chapter describes software testing and software testing processes. This chapter further elaborates the acceptance test cases which are used to test the functional and non-functional requirements after coding of software

5.1 Introduction

Software test document involves the documentation of artefacts that should be developed before or during the testing of software. Software testing is the way toward assessing a framework or its component(s) with the expectation to discover whether it fulfills the predefined prerequisites or not. Testing is executing a framework so as to distinguish any errors or missing prerequisites in as opposed to the real necessities.

5.1.1 Test approach

Manual testing includes testing a software manually without using any automated tool. The tester takes over the role of an end-user and tests the software to distinguish any bug. Testers use test plans, to test a software to ensure the completeness of testing. There are different stages for manual testing such as unit testing, system testing, and user acceptance testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it. Unit testing is the process of testing program components, such as methods or object and classes. Individual method or function are simplest type of the component.

5.2 Test Plan Environment

Test planning is an activity that ensures that there is initially a list of tasks and milestones in a plan to track the progress of the project. Test plan determines the scope and the risk that need to be tested and are not to be tested. Deciding fail and pass criteria.

5.2.1 Testing tools and techniques

This setup consists of the physical setup which includes hardware, and logical setup that includes operating system, client operating system, block chain database, front end running environment, browser or any other software components required to run this software product.

5.3 Test Case

5.3.1 Register user

Table 21: Register user test case

ID	**T1**
Description	User will be registered to the system.
Tester	User
Setup	User open website.
Instructions:	1. Select registration option. 2. Enter name (abc) 3. Enter password (*****) 4. Enter CNIC (37405-033622-0) 5. Enter DOB (23-01-1998) 6. Enter Age 7. Enter address 8. Enter country (Pakistan)

	9. Enter city (Karachi) 10. Enter email (abc@gmail.com) 11. Enter postal code (74700) 12. Press save option
Expected Results	Message will be generated that you have successfully registered.
Actual Result	As expected.
Status	Pass

5.3.2. Create Area

Table 22: Create area test case

ID	**T2**
Description	Admin will create areas.
Tester	Admin
Setup	Admin must be login into their profile.
Instructions:	1. Click Area option. 2. Enter area number (NA-01) 3. Enter area name (national60) 4. Select district postal code (75700,98003) 5. Press save button
Expected Results	Message will be generated that you have successfully created area.
Actual Result	As expected.
Status	Pass

5.3.3 Add candidates

Table 23: Assign area to candidate test case

ID	T4
Description	Admin assign area to candidates.
Tester	Admin
Setup	Admin login into their profile.
Instructions:	1. Enter party image. 2. Enter party name. 3. Enter party symbol. 4. Enter Candidate CNIC. 5. Enter Candidate Name. 6. Select election number. 7. Select electoral areas. 8. Press save button.
Expected Results	Message will be generated that you have successfully assigned area to candidates.
Actual Result	As expected.
Status	Pass

5.3.4 Poll vote

Table 24: Poll vote test case

ID	T4
Description	User will cast a vote.
Tester	User
Setup	User login to their profile.
Instructions:	1. Select election (national assembly) 2. Select candidate. 3. Press voteGen. 4. Press poll vote
Expected Results	Message will be generated that you have successfully make a transaction to the ballot box.
Actual Result	As expected.
Status	Pass

Chapter 6: Software Implementation

This document describes the project implementation for developing the project planner and scheduler.

6.1 Language Selection

- **Html/CSS**

  Used for designing web pages
- **JavaScript**

  Used for scripting and validation
- **Solidity**

  Programming language used for the development of smart contracts
- **Ajax**

  Used to fetch data from database without reloading of the page.
- **PHP**

  Used for backend
- **Java**

  Used to write web services

6.2 Tools Selection

- Web Browser
- WampServer
- Sublime text
- Eclipse oxygen

Chapter 7: Conclusion and Future Enhancements

Introduction

This document describes the project conclusions and future enhancements i.e. what type of new features can be added with time.

7.1 Summary

E-voting for smart city is a decentralized application which operates using the blockchain, smart contracts and RESTAPI. Admin creates areas, register elected candidates and citizen and assign areas to citizens. Citizens provide their personal information, register themselves, manages their information in blockchain but can’t update their information in blockchain, and cast vote to the elected candidates only once, which is a bitcoin. After completion of the voting process, counting process will start and display the results.

7.2 Conclusion

- People no longer need to go the polling station to cast a vote.
- People from anywhere in the city can cast a vote, but each citizen must register himself/herself in the system.
- The present election system of our country involves paper voting which has several disadvantages. For instance, invalid votes, integrity of votes, misplace pf votes and the unreliability in the final counting of votes. These issues can be resolved using blockchain technology, which prove d to be more authenticate and reliable.

7.3 Future Enhancements

- This application can be developed for iOS and android.
- We can visualize the voting process.

References

https://en.wikipedia.org/wiki/Electronic_voting

[1] Electronic voting, https://en.wikipedia.org/wiki/Electronic_voting, accessed on 22/08/2019

[2] JILLIAN GODSIL, The Future of Blockchain Technology Is Promising Thanks to Artificial Intelligence, https://blockleaders.io/2019/07/08/the-future-of-blockchain-technology-is-promising-thanks-to-artificial-intelligence/, 8th July 2019

[3] André Jeppsson and Oskar Olsson, "Blockchains as a solution for

traceability and transparency", master thesis, DIVISION OF PACKAGING LOGISTICS, DEPARTMENT OF DESIGN SCIENCES, FACULTY OF ENGINEERING LTH, LUND UNIVERSITY, 2017

[4]Petri Helo, Yuqiuge Hao, "Blockchains in operations and supply chains: A model and reference implementation", Published in the journal of Computers & Industrial Engineering, Volume 136, Pages 242-251, October 2019

[5] Roman Beck, Michel AvitalMatti, RossiJason Bennett Thatcher, "Blockchain Technology in Business and Information Systems Research", published in the Business & Information Systems Engineering, Volume 59, Issue 6, pp 381–384

[6] M. Chaieb et al., "Verify-Your-Vote : A Verifiable Blockchain-based Online Voting Protocol, To cite this version : HAL Id : hal-01874855 Verify-Your-Vote : A Verifiable Blockchain-based Online Voting Protocol," 2018

[7] Ahmed Ben Ayed, A CONCEPTUAL SECURE BLOCKCHAIN- BASED ELECTRONIC VOTING SYSTEM, International Journal of Network Security & Its Applications (IJNSA) Vol.9, No.3, May 2017, DOI: 10.5121/ijnsa.2017.9301 1