



**SRM**  
INSTITUTE OF SCIENCE & TECHNOLOGY  
Deemed to be University u/s 3 of UGC Act, 1956

**SRM INSTITUTE OF SCIENCE & TECHNOLOGY**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**18CSC206J - SOFTWARE ENGINEERING AND  
PROJECT MANAGEMENT**

**Lab Report**

**Project Title : ONLINE VOTING SYSTEM**

**BATCH 3**  
**GROUP MEMBERS**

S.NO	REGISTRATION NO.	NAME
1	RA1811030010011	KALKI J
2	RA1811030010016	VISHAL S
3	RA1811030010024	BARGAV R

# PROBLEM STATEMENT

## ONE PAGE BUSINESS CASE TEMPLATE

DATE	07.01.2020
SUBMITTED BY	Kalki J, S.Vishal, R.Bargav
TITLE / ROLE	ONLINE VOTING SYSTEM



### THE PROJECT

In bullet points, describe the problem this project aims to solve or the opportunity it aims to develop.

- ✓ Burden of travelling is avoided.
- ✓ Avoid threats and scams.
- ✓ Risk in voting population.
- ✓ Ease to count votes.

### THE HISTORY -

In bullet points, describe the current situation.

- ✓ Traditional voting system is prone to lot of limitations.
- ✓ Distance and time is a huge problem.
- ✓ Insecurity on the time of voting such as scams, threats etc.
- ✓ Booth capturing is prevailing.

### LIMITATIONS -

List what could prevent the success of the project,

- ✓ Networking issues in unreachable parts of the city.
- ✓ Security of the software application (hack issues).
- ✓ Misusing /identity theft of a person.

### APPROACH -

List what is needed to complete the project.

- ✓ Software knowledge
  - HTML/ CSS
  - PHP
  - MYSQL
- ✓ Technical knowledge

## BENEFITS

In bullet points, list the benefits that this project will bring to the organization.

- ✓ Convenience in voting at will in any place at any time.
- ✓ Ease in counting voting.
- ✓ Fast estimation of voting.
- ✓ Hike in public turnout in voting.

## PROBLEM STATEMENT

The online voting system is the system implemented to make the voting system easy for both polling and also for the administrator to view and check the result for each area individually. Here the voting is done anywhere with in spite of the area in which they reside.

- The voters or the administrator can login only if they already have a user id and password. New user can be only created by the administrator.
- The voter can login with his user id only if he has not casted the vote already. If he has already casted the vote the voter status is changed to voted and he cannot login with that user id unless the administrator refreshes the voting system.
- The administrator controls all parts of the online voting system except the vote casting part which is restricted to the voters only. The administrator is the only one who can create new voter.
- The administrator is the soul controller of the online voting system. He can create new nominees for the election and he can view the individual election status for each locality. The graphs are enabled to show the election status easily.
- The voter when logs in he can view his personal details which cannot be edited by him he can only report the changes to the administrator who can edit the details. The voter is granted permission only to view his personal details and vote.
- The voter while voting is provided with two options, one he can vote to any one of the nominees in the ballot page and another he can register his vote as not willing to vote to avoid the illegal voting.
- The flow of the administrator and the voter completely differs based on the authentication type which they choose.

# STAKEHOLDERS & PROCESS MODELS

## PROJECT STAKEHOLDERS

Project Name: ONLINE VOTING SYSTEM				
Prepared by: J.Kalki, S.Vishal, R.Bargav				
Project Stakeholder Name	Specific Information needs	Project Interests	Impact on Project	Role
	Types and Frequency of communication	Specific areas of interest and participation	Positive, negative, influencer, supporter, Roadblock	Decision Maker, Collaborator, Participant, Consultant, Information Recipient
Developer	Direct and medium	Develop the code	Positive	Participant
Database Manager	Direct and high	Maintain the backend database	Positive	Collaborator and participant
Tester	Direct and low	Test the code	Positive, Supporter	Consultant and decision maker
Citizens	Indirect and low	Vote the leader	Influencer	Decision Maker
Politicians	Indirect and low	Apply for election	Influencer	Participant

## **USER STORY**

### **As an election organizer/voting holder/election officer:**

- I want an identity authentication in the voting system so that it can provide normal prohibition against non registered voters voting in any election.
- I want function that gives me the authority to access the information based on my identity and title.
- I want the voting system to have flexible voting policies,so that the voting can be conducted in different ways at different regions.
- I want a counter and a restriction for the voting system so that once a voter's vote is cast it's irreversible.
- I want a message sending function so that I can send messages to related voters.

### **As a voter/electorate:**

- I want an information reminder in the voting system so that once there is new information about the candidates and parties, the system can remind me.
- I want a deadline reminder in the voting system so that once the voting deadline is approaching, the system can remind me.

## PROJECT MODEL

Chosen Appropriate Model : **INCREMENTAL MODEL**

## COMPARATIVE STUDY WITH AGILE

Category	Incremental Model	Agile Model
Quality	Quality focus shifts between Analysis/Design phase to Coding/Design phase.	Quality focus on all aspects of requirements at any given time.
Risk	Risk identification and mitigation in dev and test phase of each phase	Every identification and mitigation in every sprint
Customer Feedback	At the end of every phase	At the end of every sprint
Goals	Completion of analysis and design for a set of features by one team followed by completion of code and test by another team	Completing the feature/story in all aspects within a sprint. Delivering the shippable product
Effort Estimation	Project Manager provides the estimation for each phase	Scrum manager facilitates and Team does the estimation.
Role switch	Role switch over rarely seen	Role switchover is common
Delivery	Delivering working software at the end of each requirements.	Demonstration and delivering working software at the end of every sprint.
Estimation	PM is responsible for getting the requirements completed.	Entire team is responsible for story completion in all aspects.

# **REQUIREMENTS FROM PROBLEM STATEMENTS**

## **Functional Requirements:**

1. The system must provide the voters with accurate data
2. Audit trails of who made changes to the database must be maintained
3. Server should not lag.
4. The system must provide standard error checking
5. Vulnerabilities should be solved as much as possible.
6. The system should provide documentation to inform users of system functionality and any change to the system.
7. Ease of GUI use that can be accessed via web browser must be established
8. Database should not face problems with CRUD operations.
9. The system must provide data integrity checks to ensure data remains consistent and updated.
10. The system should allow voting administrators to make updates to the voter information database

## **System Requirements:**

### **Hardware Requirements:**

A Compatible Computer : The application should be run on a computer with at least the following configurations:

1. 4 GB RAM .
2. 2.20 GHz Dual-Core CPU Processor / Pentium.
3. 500 GB HDD

### **Software Requirements**

The software requirements of the application are as follows:

1. Windows 7/8/8.1 32/64-bit Operating System
2. MySQL Server / XAMPP server
3. PHP application for coding platform
4. Minimum display area of 1028x720 px.
5. A compatible browser, for accessing the online module

## Non Functional Requirements:

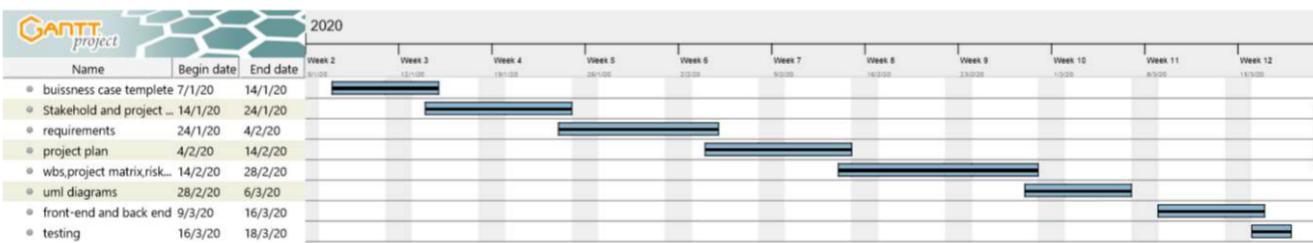
Category	Non-functional Requirements
Usability	<ul style="list-style-type: none"> <li>➤ The level of expertise of the user should be high.</li> <li>➤ The system shall provide a user-friendly interface.</li> <li>➤ The system software catalog has to be provided to the user.</li> </ul>
Reliability	<ul style="list-style-type: none"> <li>➤ The software shall be more reliable in terms of high degree of fault tolerance and be able to recover from hardware failures.</li> <li>➤ The data loss should be minimal to zero.</li> <li>➤ Any system handle exceptions should be notified with proper message and instruction to rectify.</li> <li>➤ Issues related to identity theft and hacking should be taken care.</li> </ul>
Performance	<ul style="list-style-type: none"> <li>➤ The response time of online voting should be less than 5 seconds and shall show no visible deterioration.</li> <li>➤ User tasks time should be adjustable.</li> <li>➤ Maximum of 500 citizens can access the software concurrently through parallel processing.</li> <li>➤ The database has to be sufficiently large to accommodate huge data.</li> <li>➤ Maximum latency period of 5 minutes is acceptable.</li> </ul>
Supportability	<ul style="list-style-type: none"> <li>➤ The software is compatible for extension of future updatations.</li> <li>➤ The database manager will regulate the database constantly.</li> <li>➤ The system should be able to work with other platforms as well.</li> </ul>
Implementation	<ul style="list-style-type: none"> <li>➤ The hardware platform is independent on any system.</li> <li>➤ Limited users can access the site and database has to be regularly updated by the database manager.</li> <li>➤ The tester will check and analyze the code developed by the developer now and then.</li> </ul>
Interface	<ul style="list-style-type: none"> <li>➤ The system should be able to interact with other existing systems.</li> <li>➤ Data is maintained in a well regulating database.</li> <li>➤ The system shall provide an easy-to-use interface (intuitive and navigable).</li> </ul>
Operation	<ul style="list-style-type: none"> <li>➤ The running system is managed by the database manager and the tester.</li> </ul>
Packaging	<ul style="list-style-type: none"> <li>➤ The system should be up and running and user can access during election period.</li> <li>➤ No installations required as the project is based on online activity.</li> <li>➤ No time constraints for installations.</li> </ul>
Legal	<ul style="list-style-type: none"> <li>➤ The software will be approved by the institution members.</li> <li>➤ If the software is down, there are liability issues associated with the administrator.</li> <li>➤ There is no license fees incurred for using specific algorithms and components. This project will be developed totally out of cost but with effective time.</li> </ul>

# PROJECT PLAN

## Gantt Chart

### Tasks

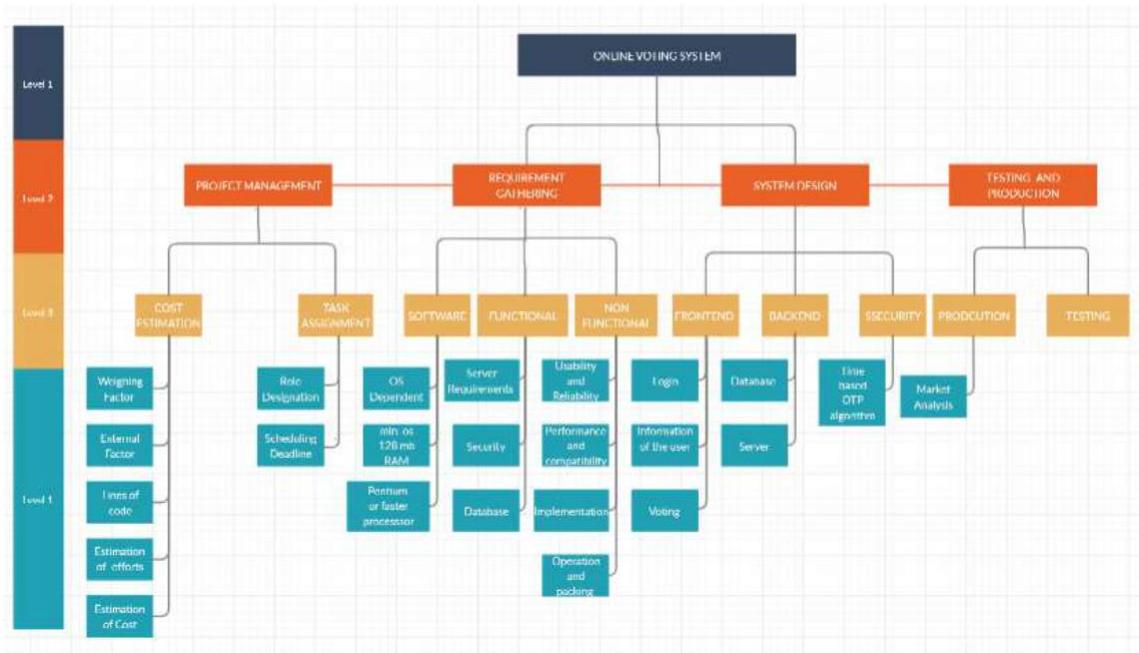
Name	Begin date	End date
buisness case template	7/1/20	14/1/20
Stakehold and project model	14/1/20	24/1/20
requirements	24/1/20	4/2/20
project plan	4/2/20	14/2/20
wbs,project matrix,risk mitigation	14/2/20	28/2/20
uml diagrams	28/2/20	6/3/20
front-end and back end	9/3/20	16/3/20
testing	16/3/20	18/3/20





# PROJECT EFFORT BASED ON RESOURCES

## Work Breakdown Structure



## Risk Analysis

Risk	Probability	RMMR
Hacker attack	50%	Appoint experts team to overcome the situation.
Environmental factors	30%	Replication to different data centres
Delivery deadline will be tightened	30%	Increase the amount of resources available

Risk	Probability	RMMM
Staff inexperienced	30%	Meet with experienced developer and experts of projects.
Size estimates may be significantly low	60%	Check the modules and revise the algorithm
Staff turnover will be high	60%	Meet with current staff to determine causes for turnover. Assign the backup staff member for every critical technologist.
Less reuse than planned	70%	Revise the components than to be used
Performance	20%	Consult the experienced software makers
Unavailability of team members	20%	Ensure that there is always an emergency person who can always take over the task assign is not available.
Reviews may not be conducted regularly	30%	Set dates for reviews for each documentation.

# ESTIMATION OF PROJECT METRICS

## COCOMO Model

### Step 1: Computation of Count-Total

Information Domain Values						
Measurement Parameter	Count	Simple ●	Average ●	Complex ●		Total
Number of user inputs	4	X	3	4	6	= 16.00
Number of user outputs	1	X	4	5	7	= 5.00
Number of user inquiries	1	X	3	4	6	= 4.00
Number of files	2	X	7	10	15	= 20.00
Number of external interfaces	1	X	5	7	10	= 7.00
Count=Total						52.00

Count Total

### Step 2: Finding Complexity Adjustment Values and Function Points

Complexity Weighting Factors // heading of the second table Rate each factor on a scale of 0 to 5: (0 = No influence, 1 = Incidental, 2 = Moderate, 3 = Average, 4 = Significant, 5 = Essential):						
Question	0	1	2	3	4	5
1. Does the system require reliable backup and recovery?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2. Are data communications required?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
3. Are there distributed processing functions?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4. Is performance critical?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
5. Will the system run in an existing, heavily utilized operational environment?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6. Does the system require on-line data entry?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
8. Are the master file updated on-line?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9. Are the inputs, outputs, files, or inquiries complex?	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10. Is the internal processing complex?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
11. In the code designed to be reusable?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
12. Are conversion and installation included in the design?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
13. Is the system designed for multiple installations in different organizations?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
14. Is the application designed to facilitate change and ease of use by the user?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Total 47.00						
Show Total of weighting Factor						
The Function Points is: Show Function Points 58.24						

### Step 3: Finding Lines Of Code

Programming Language	LOC/FP (average)	Select
Assembly Language	320	<input type="radio"/>
C	128	<input type="radio"/>
COBOL	105	<input type="radio"/>
Fortran	105	<input type="radio"/>
Pascal	90	<input type="radio"/>
Ada	70	<input type="radio"/>
Object-Oriented Languages	30	<input checked="" type="radio"/>
Fourth Generation Languages (4GLs)	20	<input type="radio"/>
Code Generators	15	<input type="radio"/>
Spreadsheets	6	<input type="radio"/>
Graphical Languages (icons)	4	<input type="radio"/>

LOC/FP:  1747.20

### Step 4: Selecting Complexity and estimating effort and duration

Software Project	a <sub>b</sub>	b <sub>b</sub>	c <sub>b</sub>	d <sub>b</sub>	Select
Organic	2.4	1.05	2.5	0.38	<input checked="" type="radio"/>
Semi-detached	3.0	1.12	2.5	0.35	<input type="radio"/>
Embedded	3.6	1.20	2.5	0.32	<input type="radio"/>

$$\text{Effort (E)} = a_b(KLOC)^{b_b} = \boxed{4.31} \quad \text{Duration (D)} = c_b(E)^{d_b} = \boxed{4.36}$$

Komodo Edit 11

Hence, our project is estimated to be completed in approximately 4 months having 4 manpower (3 teammates + 1 Faculty for guidance, in our case).

# DESIGN

## System Architecture

The project model is divided into three layers namely the presentation layer, application layer and data management layer. The Presentation layer also known as *client tier* is responsible for the presentation of data, receiving user events(votes) and controlling the user interface. Secondly, the application layer (*application server tier*) protects the data from direct access to the clients. It is the layer where most of the processing work occurs that is it enables access to the data tier so as to retrieve, modify or delete data from it which can then be sent to the presentation layer. Lastly, Data Management layer (Data server tier) is responsible for data storage using MySQL as database software.

### Components of System Architecture:

✓ **Client:**

Client is any user or program that wants to perform an operation over the system. Clients interact with the system through a presentation layer.

?

✓ **Presentation Layer:**

This layer is responsible for the presentation of data at the client side, i.e., it provides an interface for the end-user into the application.

?

✓ **Resource manager:**

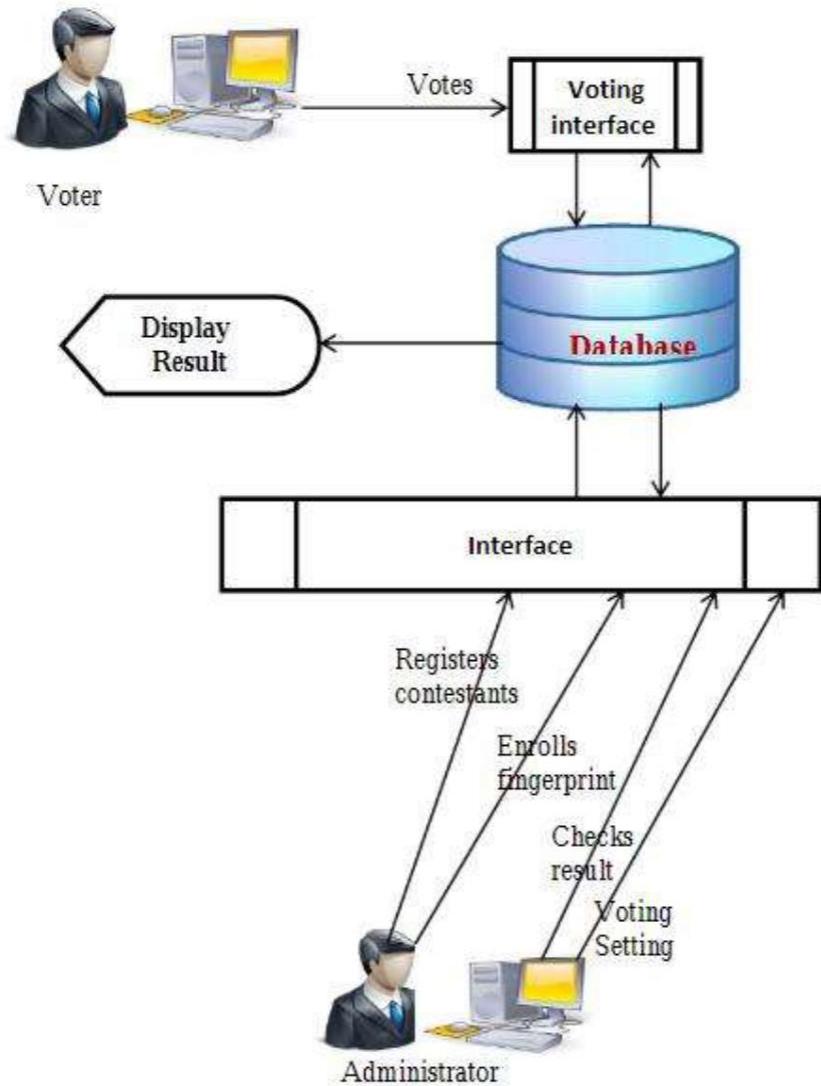
The resource manager deals with the organization (storage, indexing and retrieval) of the data necessary to support the application logic. This resource manager is typically a database.

?

✓ **Application logic:**

The application logic figures out what the system actually does. It takes care implementing the business rules and establishing the business processes. UoM online voting system is designed and implemented according to the three tier architecture.

# System Architectural Diagram



# UML USE CASE DIAGRAMS & USE CASE SCENARIOS



## **Use Case Scenario 1: Voting**

### **Voting Actor:**

Any person that is allowed to vote

### **Goal:**

To cast their votes in a safe and secure manner.

### **Preconditions:**

The process is password protected. The voter must know her/his PIN, without which they cannot vote.

### **Scenario:**

1. The voter enters the website address in his browser.
2. The voter selects the state to which he/she belongs.
3. The user is allowed to have a look at the tutorial section which is optional
4. The voter enters the Name, SSN, State ID, Date of Birth, and Gender.
5. If the input of the voter matches the records, he/she is allowed to login.
6. The voter is allowed to choose one of two options: Party Selection or Individual Selection.
7. The voter casts her/his vote to their favorite choice under a selection.
8. The voter navigates to all the pages and votes to his/her choice under each category.
9. The voter checks the final screen of the vote.
10. The voter is allowed to edit his/her vote any number of times.
11. If she/he is satisfied with the final vote screen, he/she casts the vote.
12. If the vote reaches the server, a message is displayed to the voter that his vote has reached the ballot.
13. The voter logs out.

### **Exceptions:**

1. The voter may enter the wrong details.
2. The voter might try to select options more than the allowable ones.
3. The voter's connection with the server may terminate before the vote reaches the server.
4. The voter's connection with the server may terminate in the course of the session.
5. After the vote is cast, the voter may try to navigate back to cast another vote.

### **Event:**

If the voter is not identified in three attempts, the process stops and the voter needs to contact the election conducting authority to restart the process.

### **Frequency:**

Used as many number of times as there are voters.

### **Secondary Actors:**

Election conducting staff who are contacted by voters in case of difficulties.

## **Use Case Scenario 2: Configuration**

**Actor:**

Configurator (usually an authorized person of the election commission).

**Goal:** To configure the voting system by entering the offices for which voting is to be done and configuring the candidates for the offices.

**Preconditions:** There are no preconditions while installing.

**Scenario:**

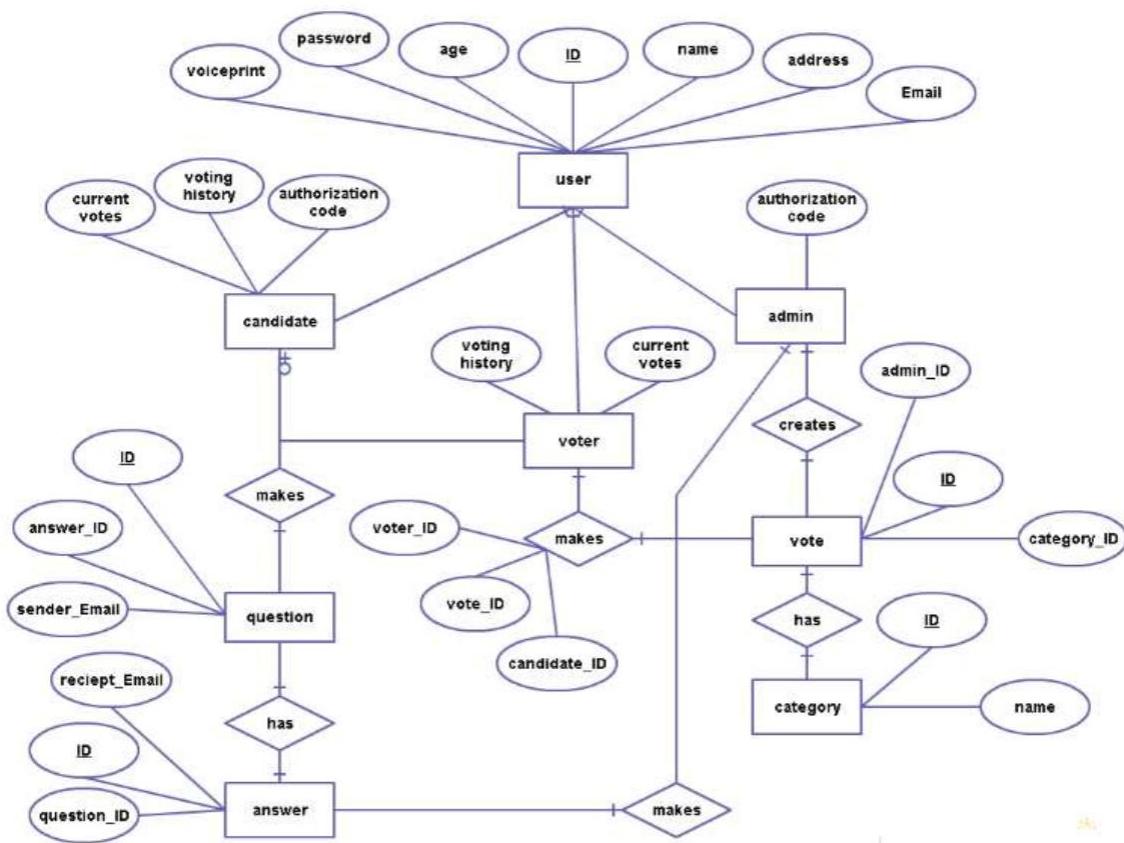
1. The actor clicks the button “Configure.”
2. The actor clicks either “Single Configuration” or “Multiple Configuration” button based on whether the election is held for a single province or a multiple province.
3. If the “Multiple Configuration” button is pressed, the actor is prompted to enter the common offices and the offices that are specific to that province.
4. The actor enters the criteria based on which provinces are distinguished.
5. The actor is allowed to add a new office or edit an existing office by pressing “Add New Office” or “Existing” button respectively.
6. The name of the office and the number of candidates for that office are entered.
7. The actor clicks the next button which allows him/her to enter the name of the candidates and the party to which they belong.

Exceptions: There are no exceptions.

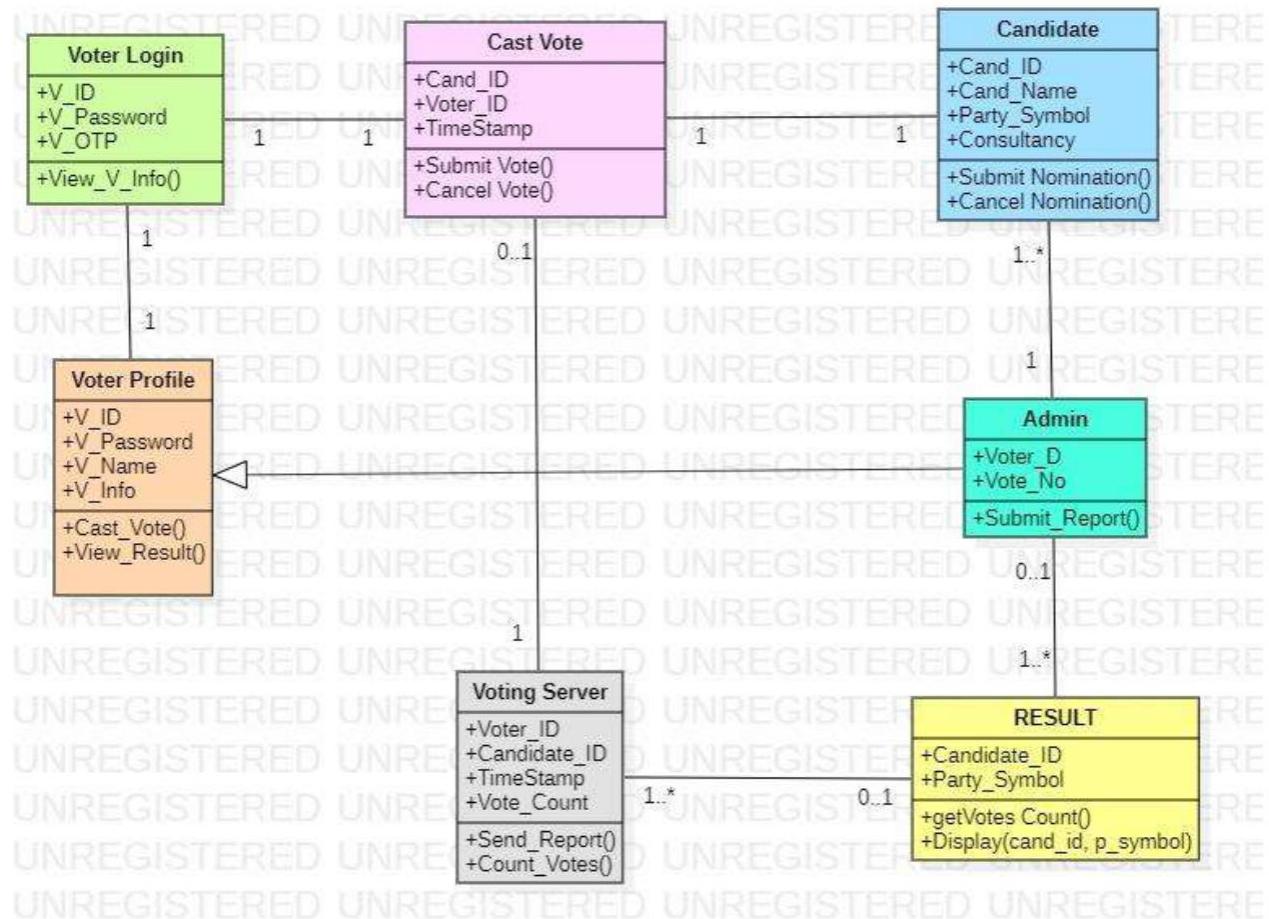
**Frequency:** Usually once.

**Secondary Actors:** Software staff.

# E-R MODELING

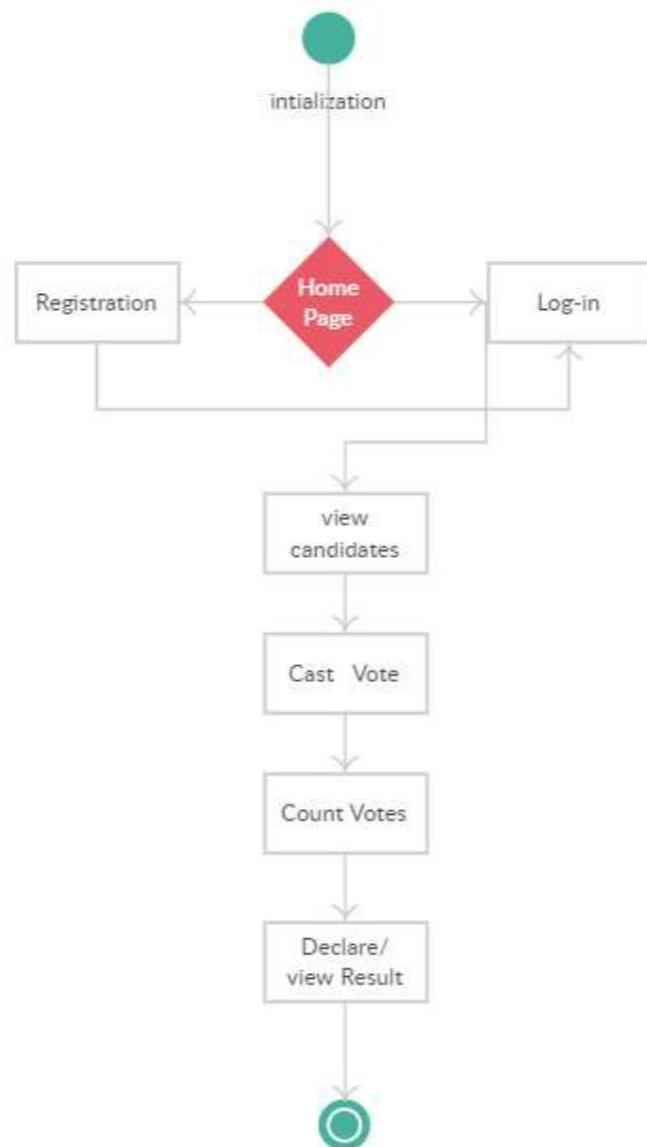


## DOMAIN CLASSES - CLASS DIAGRAM

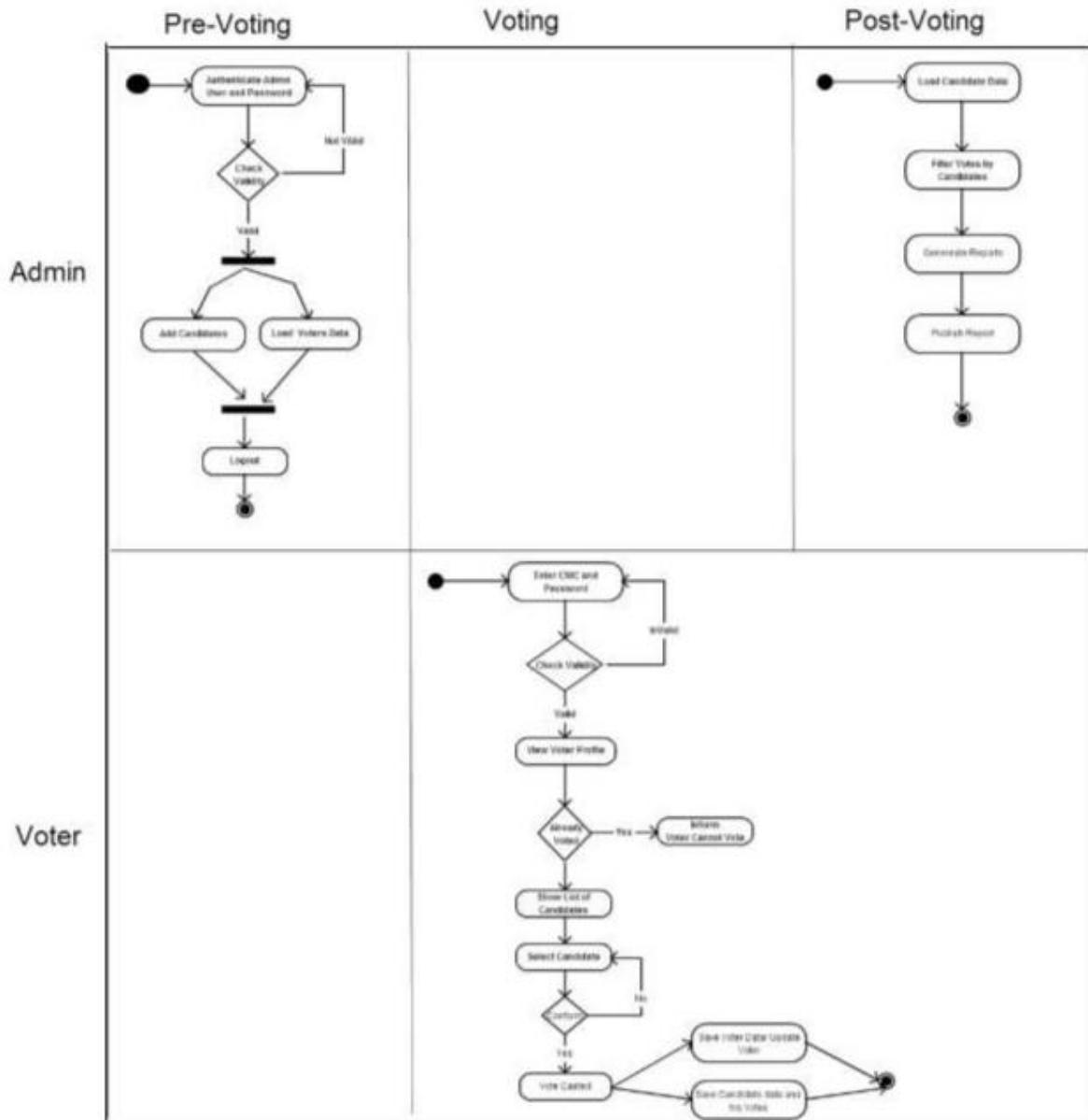


# STATECHART AND ACTIVITY MODELING

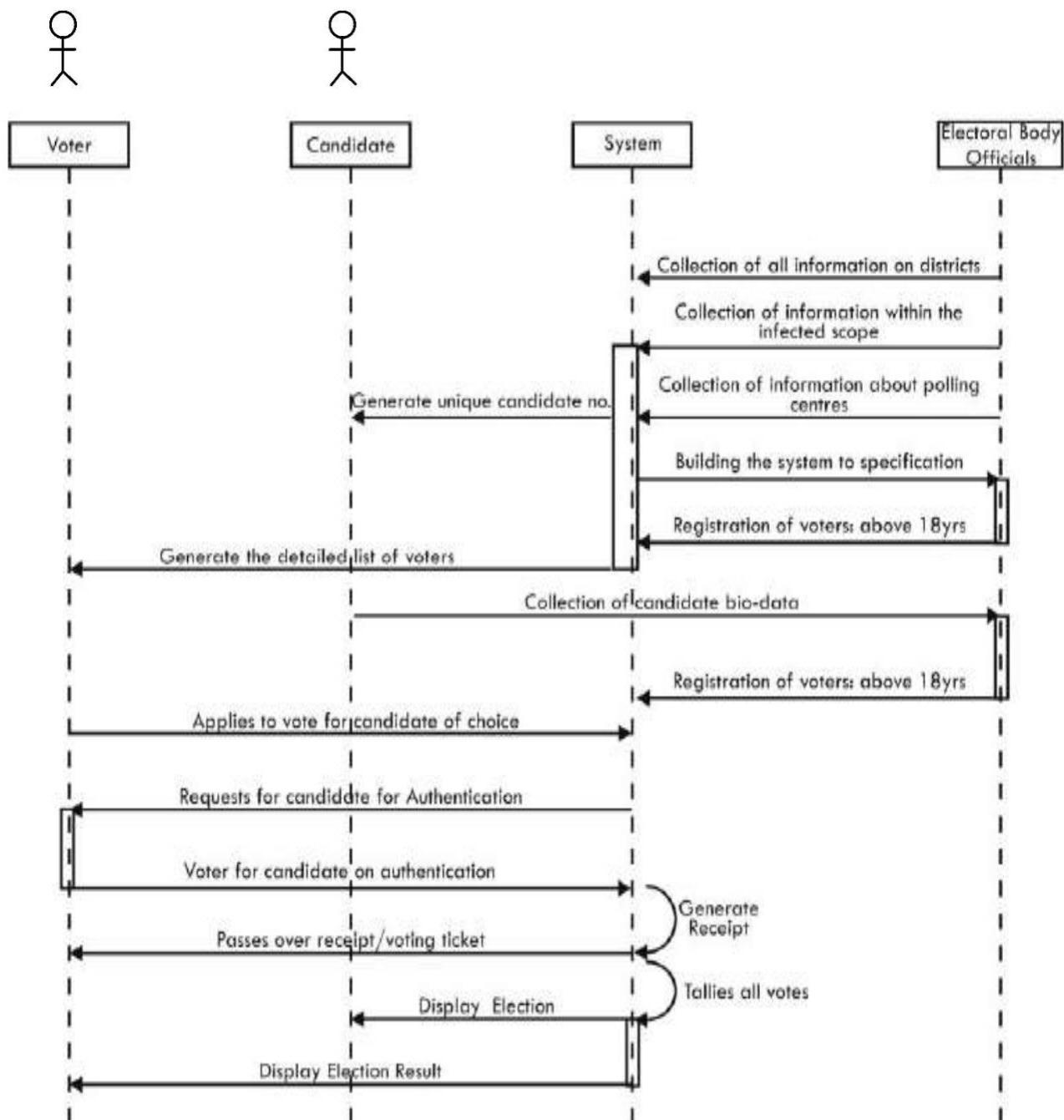
## Statechart diagram



# Activity diagram

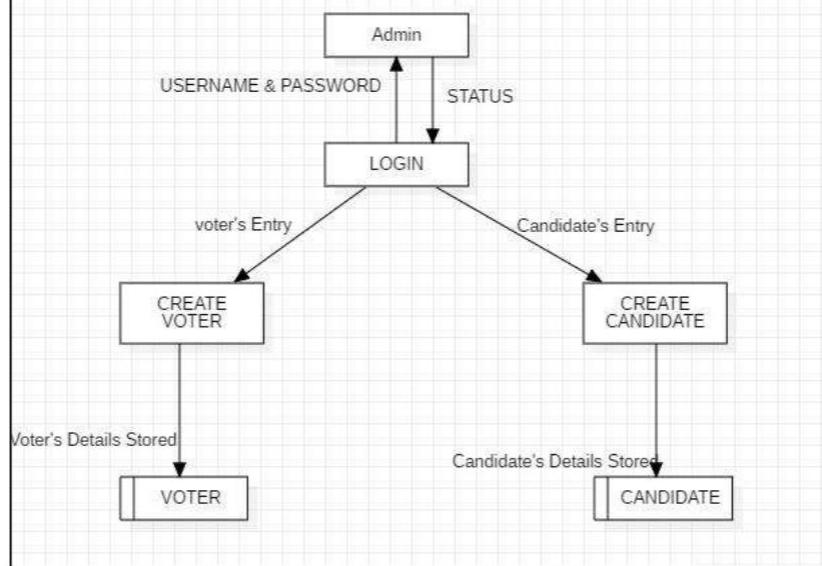


# SEQUENCE DIAGRAM

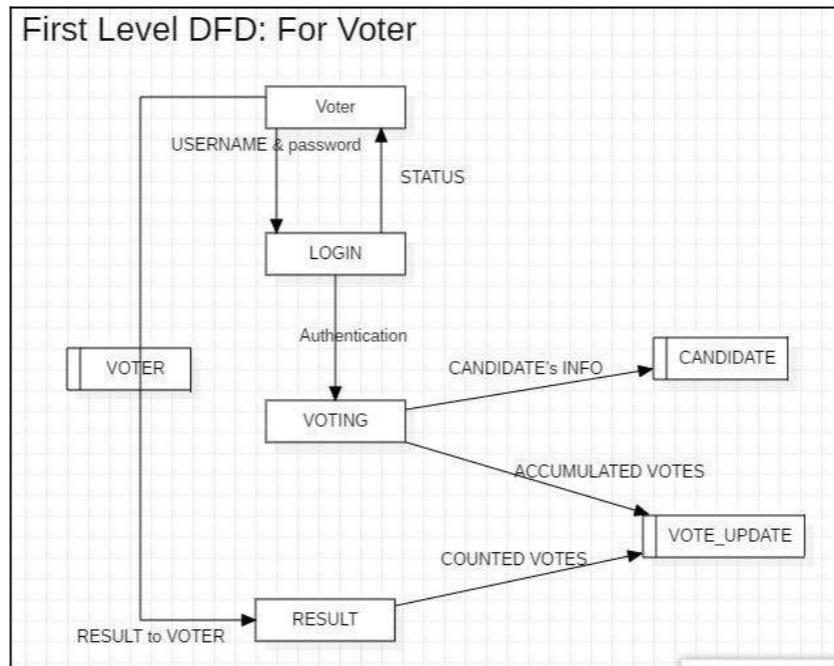


# DATA FLOW DIAGRAM

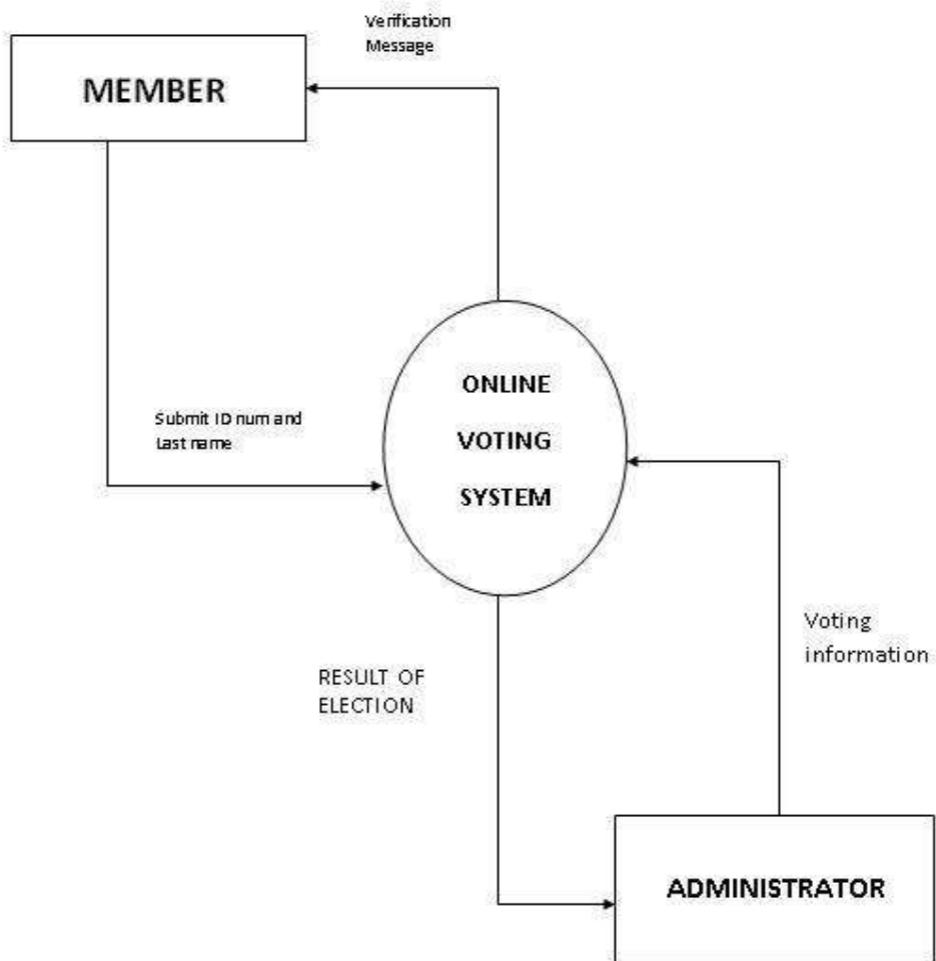
First Level DFD:For Admin



First Level DFD: For Voter



## Context Diagram

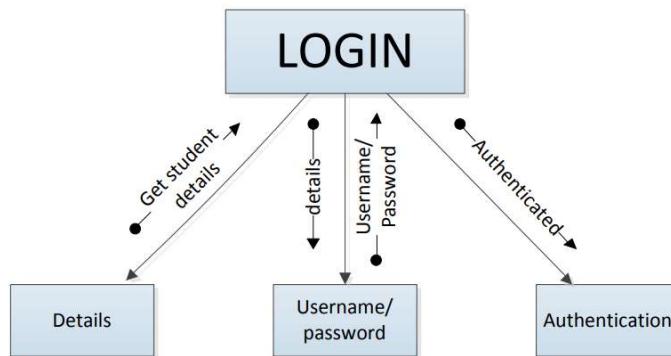


# ESTIMATION OF TEST COVERAGE METRICS AND STRUCTURAL COMPLEXITY

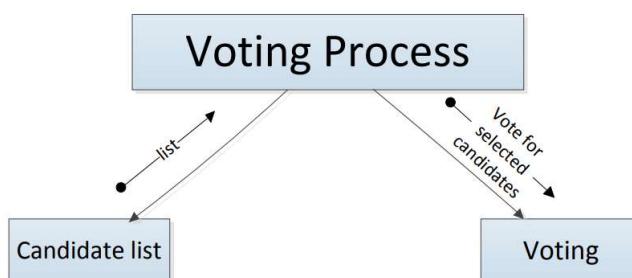
## Control Flow Graph

The first-level of control flow results in a very high level structure, where each subordinate module has a lot of processing to do. To simplify these modules, they must be factored into subordinate modules that will distribute the work of a module. Each of the input, output and transformation modules must be considered for factoring. A subordinate input module is created for each input data stream coming into this new central transform, and a subordinate transform module is created for the new central transform. The new input modules now created can then be factored again, until the physical inputs are reached. The factoring of the output modules is symmetrical to the factoring of the input modules. The goal is to determine sub transforms that will together compose the overall transform and then repeat the process for the newly found transforms, until we reach the atomic modules.

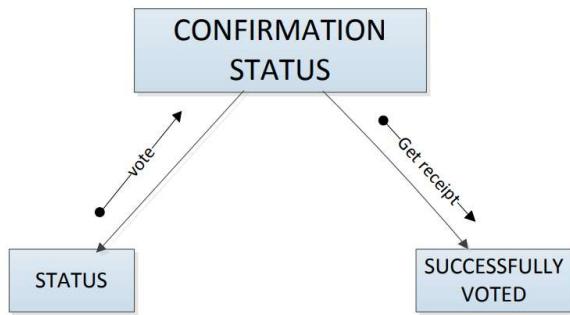
**Control flow of Input module**



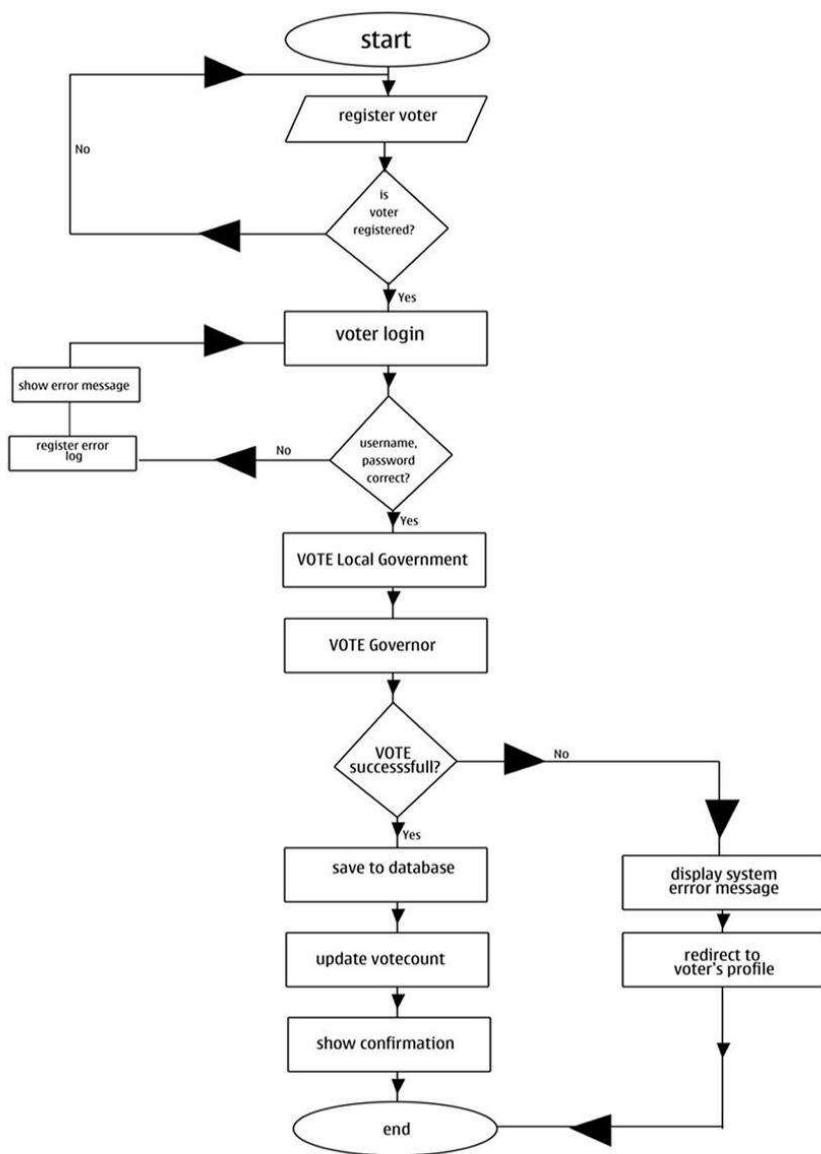
**Control flow of Transform Module**



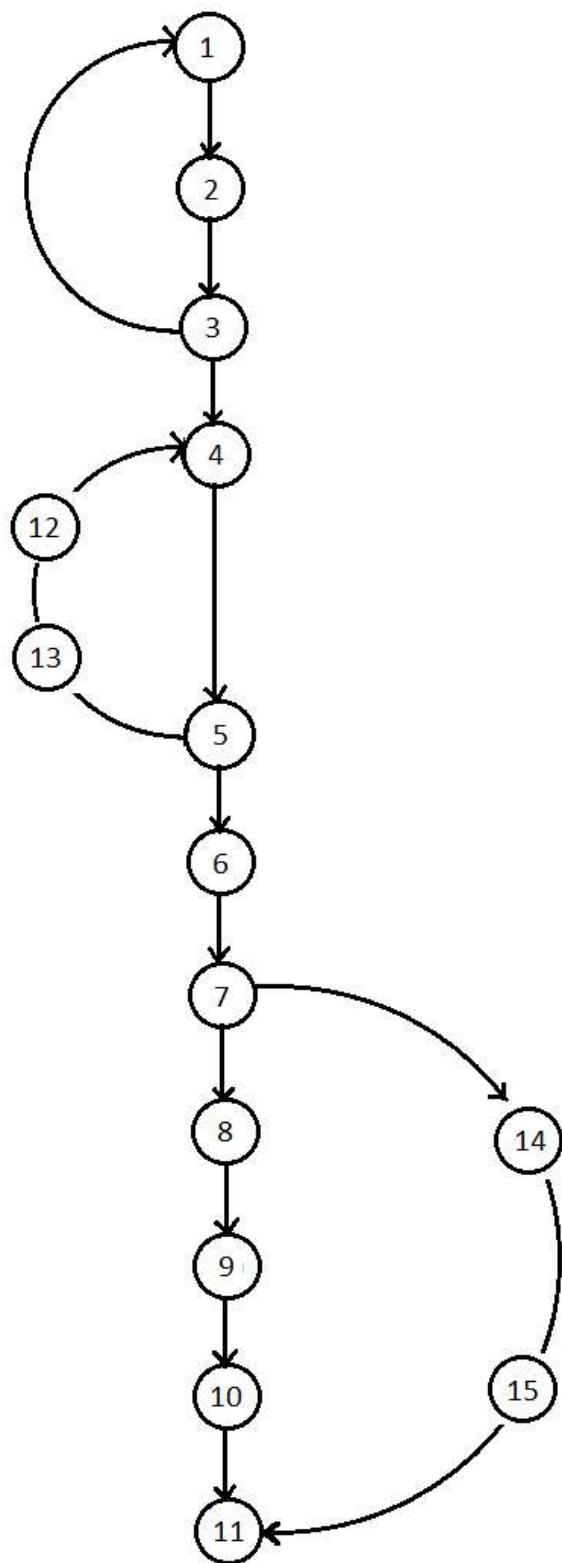
## Control flow of Output Module



## Complete Flow Chart



Corresponding Flow Graph



## McCabe's Cyclomatic Complexity

Cyclomatic complexity is a software metric used to measure the complexity of a program. It is a quantitative measure of independent paths in the source code of the program. Independent path is defined as a path that has at least one edge which has not been traversed before in any other paths. Cyclomatic complexity can be calculated with respect to functions, modules, methods or classes within a program. This metric was developed by Thomas J. McCabe in 1976 and it is based on a control flow representation of the program.

## Optimum Value of Cyclomatic Complexity

The cyclomatic complexity can be computed by using the algorithm.

1. The number of regions correspond to the cyclomatic complexity  $\square$ .
2. Cyclomatic complexity,  $V(G)$ , for a flow graph,  $G$  is defined as  
$$V(G) = E - N + 2 \quad (\text{where } E = \text{no. of edges}, N = \text{nodes})$$
3. Cyclomatic complexity,  $V(G)$ , for a flow graph  $G$ , is also defined as  
$$V(G) = P + 1$$

In this case,

1. Number of regions = 4
2. 
$$\begin{aligned} V(G) &= E - N + 2 \\ &= 17 \text{ edges} - 15 \text{ nodes} + 2 \\ &= 4 \end{aligned}$$
3. 
$$\begin{aligned} V(G) &= P + 1 \\ &= 3 \text{ predicate nodes} + 1 \\ &= 4 \end{aligned}$$

Therefore the Optimum Value of Cyclomatic Complexity is 4

## Independent Program Paths

An independent path is any path through the program that introduces atleast new set of processing statements or a new condition. When stated in terms of a graph, an independent path must move along atleast one edge that has not traversed before the path is defined. A set of independent paths flow graph illustrated in figure is :

path 1: 0-1-2-3-4-5-6-7-8-9-10-11-12  
path 2: 0-1-2-3-5-6-7-14-15-11  
path 3: 0-1-2-3-5-13-12-4  
path 4: 0-1-2-3-1

# **DESIGNING TEST SUITES**

## **Software Testing**

Software Testing is the process of running a system with the intention of finding errors or checking if it is working well by meeting the parameters established for it. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements. The main purpose of testing is to detect errors and error-prone areas in a system. Testing must be thorough and well-planned. A partially tested system is as bad as an untested system. And the price of an untested and under-tested system is high.

The implementation is the final and important phase. It involves user-training, system testing in order to ensure successful running of the proposed system. The user tests the system and changes are made according to their needs. The testing involves the testing of the developed system using various kinds of data. While testing, errors are noted and correctness is the mode.

The Online Voting System was designed to run on a web browser thus a XAMPP server (which is basically built to test websites to see if the expected functionalities are met before finally uploading it online) was installed to test the system and was confirmed to be logically and functionally correct and working.

In this project,

1. Unit testing
  - (i) White Box testing
  - (ii) Black Box testing
2. Integration testing
3. Validation testing
4. System testing
5. Acceptance testing are being tested .

# **Testing Framework**

## **Test Case Design 1:**

### **Test Case Description**

This test case deals with the entering of basic details related with the registered and logged in user . This process takes many inputs. The test case should check for proper inputs and verify whether the function for validating the inputs is called properly or not.

### **Expected Inputs**

Details Related With The logged in user in a Standard Format.

### **Expected Outputs**

- (i) Alert window for erroneous inputs.
- (ii) Confirmation of Successful Completion of Basic Details Entry Process.

### **Actual Test Results**

- (i) An alert window was shown whenever the user gave some erroneous data.
- (ii) Confirmation of Completion of Basic Details Entry was displayed on submission, implying that the functions were called properly.

## **Test Case Design 2:**

### **Test Case Description**

This test case deals with the updating of Basic Details related with the users trade by the java function. The test will check if the java function receives the updating details properly and checks if the updating is performed correctly or not.

### **Expected Inputs**

Updated Basic Details from the web server.

### **Expected Outputs**

- (i) The java function should return the correct row count.
- (ii) Row count denotes the number of rows being updated.

### **Actual Test Results**

- (i) The java function received the details to be updated properly.
- (ii) On updating the correct row count was returned by the function.

# **Master Test Plan**

## **Scope of Study**

The scope of the project is that it will use the ID and password created by user to register him/her in the voting site, through this all the details of voter are saved in database. And it will act as the main security to the votes system.

- i. **Advanced technology:** It is an advanced technology used now a day. It increases the internet knowledge of the users which is very necessary for current generation.
- ii. **Internet:** It is an online facility and hence very useful for the users. Voters can vote from anywhere at any time in India.
- iii. **Mails:** Election Commission can send the error report to a particular user if he\she entered false information.
- iv. **E-SMS:** People who have not internet connection they cannot check the emails or not have email they can be informed by SMS on their mobile. Today many websites provide free SMS to the mobile. Election Commission can use these to send any information.

## **Contract of Responsibility**

### **(a) General:**

It is the Contractor's responsibility to comply with all of the terms and conditions of this Contract. The Contractor is also expected to make every effort to support the mission and duties of the schemes of Online Voting System.

### **(b) Personnel and Staffing.**

Contractor is solely responsible for all matters concerning the recruitment, performance and retention of Contractor's personnel. Contractor must fully comply with all federal, state, and local laws/regulations regarding employment and immigration, including, but not limited to, nondiscrimination, compensation, taxation, and benefits.

Contractor understands and agrees that team members are not permitted to work under this Contract, nor can they have ownership in this project for or works under this Contract, either as a prime contractor or as a sub-contractor. Any breach of this provision shall result in immediate termination of this Contract and termination of involvement with this team.

Contractor warrants that all persons assigned to perform the services under this Contract are either lawful employees of Contractor or lawful employees of a subcontractor. All of Contractor's or any of subcontractor's personnel shall comply with the confidentiality requirements of the Agreement and the security requirements . In the event that any of Contractor's or subcontractor's personnel do not comply with such confidentiality and security requirements.

Contractor agrees that any obligation to provide health insurance to any individual employed, utilized or subcontracted by Contractor shall be the sole and exclusive responsibility of Contractor or the subcontractor of Contractor. Contractor warrants that all persons assigned to perform the services under this Contract are qualified to perform such Services.

Personnel assigned by Contractor shall have all professional licenses required to perform the Services. Contractor warrants that persons assigned by Contractor to provide the services to possess all licenses, training and certifications. Contractor understands and agrees that it is responsible for paying the wages of Contractor's personnel, and will be responsible for withholding and remitting all income and social security taxes from their wages and for paying workers' compensation insurance premiums, state and federal unemployment insurance taxes and the employer's share of social security taxes on the behalf of its personnel, as required by law.

Contractor recognizes that personnel who are assigned to provide the services may have access to certain information, which may be proprietary and confidential. Contractor will require each of the personnel to treat this information as confidential and agree not to disclose it to any third person in accordance with the terms of this Contract. Contractor agrees all proprietary rights which it may have in work product created or contributed to by personnel while on assignment.

# Description Of Main Processes

## Pre-election

### Planning

Election dates are fixed and the electoral processes are complex, and require careful planning strategies. Our experience in thousands of elections allows us to develop contingency plans, minimize risks, delivering a robust voter experience.

### Voter Education

Voter education is needed to let the electors exercise their right to vote. Well-informed voters are essential for a healthy and democratic electoral process and to achieve high participation.

This requires transparency, which means not only an awesome website and down to earth documentation, but the strategic dissemination of information in multiple media formats.

### Election configuration

The configuration of the election can be really complex. Election configuration includes the layout and presentation of the options, the questions, the tally methods, etc. Years of experience have allowed us to develop a powerful solution that lets us configure all the aspects of the election.

### Personnel training

Effectively assisting the public during an election is one of the most important tasks to guarantee the right to vote. Correctly performing and orchestrating the internal processes of the election, such as executing the tally or managing the census and the issues that may arise, is not only essential but needs proper staff training.

### Voter registration

Guaranteeing the right to vote requires that eligible voters can participate in the election. The registration of thousands or millions of electors is not only complex, but usually requires a sizable amount of internet resources. Our voter registration solution grants legitimacy to the electoral process, keeping registration tasks simple and transparent.

## Election-day

### Internet Voting

Our solution for remotely casting votes through the Internet provides the highest security and privacy standards. Remote offline electronic voting is an efficient electoral solution for organizations, like the military or those living abroad, whose members are dispersed in a wide geographic area. It is also a good fit for smaller organizations lacking the resources to establish on-site polling stations..

## **Authentication**

Multiple authentication and signature mechanisms can be used, and sometimes more than one is used at once.

SMS one-time authentication codes, email, and more!

Postal delivery of authentication codes

Credentials can safely be delivered via postal mail. The credentials can be a codes card, so that it can securely be reused in future elections.

## **Post-election**

### **Auditing**

Maintaining the legitimacy and credibility of an online election requires not only top-notch security software, hardware and protocols, but the auditing by experts in technology and processes to guarantee the integrity of the election.

### **Reporting**

On election night everyone (candidates, the news, the public) is awaiting the results of the election. Accurately reporting these results in an understandable and timely manner can be crucial to the success of the whole operation. Our solution provides a professionally designed website to produce real-time, fast and secure election result updates.

## **Innovation**

### **Participation platform**

Sometimes a public or private organization needs to carry out electoral processes on a frequent basis. Our participation platform solution provides a secure, easy to use and cost-effective way of covering this need of both big and small organizations alike.

## Risks and Assumptions

- Attacks could target availability, confidentiality, or authentication of the system.
- Distributed denial of service attacks (DDoS) can overload servers, preventing voters from registering and especially if elections are held on a single day.
- Intruders could read personal information, submit false information, or even change info on voters.
- If separate servers are used for both verifying voters and counting votes, then the verification servers can be separately targeted for a DDoS.
- Attackers could also take the place of legitimate voters through phishing attacks, tricking users into revealing their credentials.
- DDoS attacks are possible to keep all voting locations from reporting, but the main threat is against integrity
- Intruders could potentially break into election servers and change previously cast votes.
- Attackers could potentially impersonate legitimate voters to cast false votes, or monitor network traffic to see how individuals voted.

### Assumptions and Dependencies

#### User side assumptions and dependencies.

- PC (Personal Computer) or workstation with GUI.
- A web browser with support for cookies.
- Working Internet connection.

#### Server side assumptions and Dependencies

- A web server with GUI, PHP and an http server installed .

# **Testing**

## **1. Unit testing**

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

### **(i) White Box Testing**

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

### **(ii) Black Box Testing**

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

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

## **Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail

## **Test objectives**

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

## **Features to be tested**

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

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

## **2. Integration testing**

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

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

### **Integration testing for Database Synchronization:**

- Testing the links that call the Change Username & password, Migration and Synchronization screens etc.
- The username should be retained throughout the application in the form of hidden variables or by using cookies.
- If the login user does not have enough privileges to invoke a screen, the link should be disabled.
- Any modification in the Master server should be reflected in the Slave server.
- The XML file should retrieve only the records, which have been modified.

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

### **3. Validation testing**

Validation testing provide a systematic demonstration that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Validation testing is centered on the following items:

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

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify :

Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

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

### **4. System test**

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

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

## **5. Acceptance test**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

### **Acceptance testing for Data Synchronization:**

- Users have separate roles to modify the database tables.
- The timestamp for all insertions and updating should be maintained.
- Users should have the ability to modify the privilege for a screen.
- Once the Synchronization starts, the Master server or Slave Server should not be stopped without notifying the other.
- The XML file should be generated in short time, i.e., before the next modification occurs.

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

# Manual Test Case

TEST SCENARIO		CHANGE PASSWORD IN ONLINE VOTING SYSTEM						
BATCH		B3						
REGISTRATION NO.		RA1811030010011, RA1811030010016, RA1811030010024						
DATE		18th April, 2020						
TEST SCENARIO ID	TS_CHPASS_001	TEST CASE SCENARIO	Verify the change password action					
TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POSTCONDITION	ACTUAL RESULT	STATUS
TC_CHPASS_001	Enter valid current password, valid new password and valid confirm new password.	Need a valid user account to do 'change password'	1. Enter current pass 2. Enter new pass 3. Enter confirm new pass 4. Click update	<Valid current pass> <Valid new pass> <Valid confirm new pass>	Password update should be successful	Password should be updated	Password update is successful	PASS
TC_CHPASS_002	Enter invalid current password, valid new password and valid confirm new password.	Need a valid user account to do 'change password'	1. Enter current pass 2. Enter new pass 3. Enter confirm new pass 4. Click update	<Invalid current pass> <Valid new pass> <Valid confirm new pass>	A message "The current password does not match" should be shown	An error message is popped out	As expected	PASS
TC_CHPASS_003	Enter valid current password, valid new password and invalid confirm new password.	Need a valid user account to do 'change password'	1. Enter current pass 2. Enter new pass 3. Enter confirm new pass 4. Click update	<Valid current pass> <Valid new pass> <Invalid confirm new pass>	A message "The new pass and confirm new pass do not match" should be shown	An error message is popped out	As expected	PASS
TC_CHPASS_004	Enter valid current password, invalid new password and invalid confirm new password.	Need a valid user account to do 'change password'	1. Enter current pass 2. Enter new pass 3. Enter confirm new pass 4. Click update	<Valid current pass> <Invalid new pass> <Invalid confirm new pass>	A message "Enter valid new password" should be shown	An error message is popped out	As expected	PASS
TC_CHPASS_005	Click update leaving current password, new password and confirm new pass fields blank	Need a valid user account to do 'change password'	1. No current pass 2. No new pass 3. No confirm new pass 4. Click update	<Nil> <Nil> <Nil>	A message "Fields should not be left blank" should be shown	An error message is popped out	As expected	PASS

# DEPLOYMENT REPORT

Deployment reports contain historical information about deployments, such as the total number of deployment runs and the average duration of those runs.

## Deployment average duration

This report provides average deployment times for applications that ran during a specified reporting period. The report provides both a tabular presentation and line graph of the data. It took 3 months in completing our project. Hence the deployment average duration for our project is 3 months.

## Deployment total duration

This report provides total deployment times for applications that ran during a specified reporting period. It took 14 attempts to run the application successfully. Therefore the deployment total duration is 14.

## Deployment count

This report provides information about the number of deployments that ran during a specified reporting period. It is the very first time we deployed our project. Hence the deployment count for our project is 1.

## Deployment detail

This report shows all deployments for the specified applications, environments, and users in the specified time period.

Report name	Description
All application deployments (advanced)	Displays detailed summary information for all application deployments.
All application deployments (basic)	Displays summary information for all application deployments.
Application deployments per asset	Displays applications deployed to a specified device or user.
Application infrastructure errors	Displays application infrastructure errors. These errors include internal infrastructure issues, or errors resulting from invalid requirement rules.
Application Usage Detailed Status	Displays usage details for installed applications.
Task sequence deployments containing application	Displays task sequence deployments that install a specified application.

## **CONCLUSION**

This Online Voting system will manage the Voter's information by which voter can login and use his voting rights. The system will incorporate all features of Voting system. It provides the tools for maintaining voter's vote to every party and it count total no. of votes of every party. There is a DATABASE which is maintained by the ELECTION COMMISSION OF INDIA in which all the names of voter with complete information is stored

In this user who is above 18 year's register his/her information on the database and when he/she want to vote he/she has to login by his id and password and can vote to any party only single time. Voting detail store in database and the result is displayed by calculation. By voting system percentage of voting is increased. It decreases the cost and time of voting process. It is very easy to use and It is vary less time consuming. It is very easy to debug.

## **REFERENCES**

- ◆ Software Engineering Theory and practices, Shari Lawrence pfleeger, Joanne M.Atlee, Third edition, Pearson Education,2006.
- ◆ Roger S. Pressman, Software Engineering- A Practitioner's Approach, McGraw-Hill International Edition, Seventh Edition, 2009.
- ◆ Mastering UML with Rational Rose 2002, Wendy Boggs, Michael Boggs, Sybex International Edition 2005
- ◆ <http://stsc.hill.af.mil/stscs.html> - project management and cost estimation details.