Online Election System in College

Technical Report
No.Du/IP/B.sc(H) CS-IV/2016

April 2016

Department of Computer Science

Delhi-110054

1.Report number: No. Du/IP/B.sc(H) CS-IV-2016	2. Report Date: 12 April 2015
3. Report type: Technical report	4. Subject/Area: Software Engineering
5. Title: Online Election System in college.	
6. Author: Eva Chaudhary	7.Guide: Ms. Vimala Parihar
8. Name of Institute: Deptt. of Computer Science Indraprastha College for Women, Delhi-110054	
9. Abstract: **"Online Election System in college"** will be an online election technique in which the students can give his/her vote online and can also apply as a contestant online The result will be computed and declared online.	
10: Keywords: Voter, Contestant, Result, Vote Count	
11. Total no of pages: 53	12. Signature of author
Signature of Guide	

ACKNOWLEDGEMENT

We would like to sincerely acknowledge the help of our Software Engineering teacher Ms. Vimala Parihar. Without her help we couldn‟t have gone through with the magnanimous task of designing our software “Online Election System in College”. It was our honor to work under her guidance. Her rigorous commitment to our project and constant availability for help was a big help.

We would also like to thank our lab assistants Mr. Vipin Kumar Sharma, Mr. Vivek Kumar and Mr. Aniket for they supported and encouraged us to achieve our goal of successfully completing the task at hand and facing our challenges, so that we would not procrastinate in obtaining the desired result.

CERTIFICATE

This is to ceify that Eva Chaudhary successfully carried out the completion of the project entitled “Online Election System in College” under my supervision. The project has been submitted as per the requirements of Lab based on Software Engineering in the fourth semester of B.sc(H) Computer Science.

Teacher-In-Charge

(Ms. Vimala Parihar)

INDEX

S.NO.	TOPIC	PAGE NO.
	REQUIREMENT ANALYSIS	
1	Introduction	
2	SRS(Software requirement specification)	
3	Process model	
	DESIGN SPECIFICATION	
4	DFD(Data flow diagrams)	
5	Data dictionary	
	PROJECT MANAGEMENT	
6	Function point	
7	Effort estimation	
8	Risk table	
9	Timeline chart	
	DESIGN ENGINEERING	
10	Architectural design	
11	Data design and data table	
	CODE DEVELOPMENT	
12	Flow charts	
	TESTING	
13	Flow graph	
14	Cyclomatic complexity calculation	
15	White box testing test cases	
16	**FUTURE SCOPE**	
17	**REFERENCES**	

INTRODUCTION

“Online Election System in college” will be an online election technique in which the students can give his/her vote online and can also apply as a contestant online without going to the college desk.

This will surely consume less time as whole day is being consumed on the day of election .

There will be a database which will be maintained in which all the names of the students with their complete information will be stored and each student will be provided a USER ID and a log in password so that he/she will give his/her vote .

The students who will apply as a contestant for a particular position undergoes a criteria in which they will be tested that whether he/she will be capable for being nominated .

The scope of this project will be that surely increase voting percentage.
Online election system will be fast enough to calculate the results and reduce human efforts.

Software Requirement Specifications

Introduction

Software Requirements Specification is a perfect detailed description of the behaviour of the system to be developed. That is SRS document is an agreement between the developer and the customer covering the functional and non functional requirements of the software to be developed.

SRS is considered as a contract between the customer and the developer. It is a document that captures complete description about how the system is expected to perform.

Purpose

This SRS will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

Online Election System is a web-based application that allows the students to give his/her vote online and can also apply as a contestant without going to the college desk. This will surely consume less time as whole day is being consumed on the day of election.

A second purpose is to make it more difficult to commit fraud and cheating during an election. In a mannual system, sometimes people are register in more than one area and thus cast the vote multiple times. By creating an online database covering it will be possible to eliminate the double casting of votes. It would also give students option to cast an empty vote if they don't like to give the vote to any of the candidates.

This SRS has been built according to user needs. It will help the software developer to develop and fully understand the expectations of the user.

Scope

This software can be used by any college to make election computerized.

Proposed system

In this software the student is given an identity card that contains unique card ID. When the user enter the card ID, it display all the information about the particular student i.e. his personal details and also gives the information about all the candidates who have contested for election based on student union or student section.

In addition, this software is designed such a way that an individual is not allowed to vote more than once i.e. once an individual has finished voting, his/her personal information will be disabled. Data in this software is completely secured which means that only authorized person can update the information.

Advantages of the Software

1. It increases the counting speed.
2. It reduces the human effort.
3. There is no chance of wastage of votes.
4. It save lot of time.
5. Result will be announced within a short period of time

Hardware and Software requirement

Hardware requirement

Processor : Intel Core Duo 2.0 Ghz or more

RAM : 1 GB or more

Hard Disk : 80 GB or more

Monitor : 15” CRT or LCD monitor

Keyboard : Normal and Multimedia

Mouse : Compatible mouse

Software Requirement

Front end : Visual Basic 2008 Express edition

With SQL Server Compact Edition

Microsoft SDK 4.0

Back end : MS SQL Server

Operation System : Windows XP with server pack 3 or Windows 7

Main Modules

These are the modules that will be required in the project .

- user/voter
- registered contestant
- administrator

1. USER/VOTER MODULE

- **LOGIN**: The user/voter module helps the voters to easily give their votes. This module will provide them unique ID and password so that they can easily login and give their vote to their favourite candidates. If they don't want to give the vote to any of the participating candidates then they can press the **NOTA** button .

- **View contestants details :** This module allows the voters to check out the details of the participating candidates if they don't know anything about them in any case, so they can easily give their votes to the contestant whom they like.

- **Enter vote :** Using this option, a voter can give his/her vote .

- **Contact information:** He/she will get his/her username and other details.

- **Logout:** The user can logout after the voting process.

2. CONTESTANT MODULE

- **LOGIN** : This module helps the contestants to easily give their votes. This module will provide them unique ID and password so that they can easily login and give their vote .

- **Enter details** : This module will help the contestant to give information about themselves such as:

    1. **NAME AND AGE**
    2. **CONTACT INFORMATION**
    3. **AGENDA**
    4. **AIM**
    5. **POST FOR WHICH THEY ARE STANDING**
    6. **WHY SHOULD PEOPLE VOTE THEM?**

  So that the voter easily undestand their goal and vote them.

- **Logout:** After giving the vote and information about themselves they can logout from the site.

3. ADMINISTRATOR MODULE

- **Provide login-ID and password :**After the registeration of the voters/contestants, each student will be provided a unique ID and password along with their username.

- **Upload details** : After all the contestant enter their details and all the details will be uploaded on the election site .

- **False voting checker :** If one user will try to vote more than once with the same login ID and password, then his/her login ID will be blocked and he/she will be automatically log-out from the site and an error will be displayed .

- **Eligiblity of candidates /voters** : By this module the administartor will check whether the candiate is eligible to stand particpate in the election or not by following the particular criteria and whether the voters are eligible for voting or not .

PROCESS MODEL

When a product or a system is being built, it is important to go through a series of predictable steps- a road map that helps to create a timely, high quality and reliable software.

A software process model is a standardised format for
• planning
• organising, and
• running a development project.

It is a description of the sequence of activities carried out in a software engineering project, and the relative order of these activities. It provides a fixed generic framework that can be tailored to a specific project.

A variety of different process models for software engineering are available. Each model attempts bring order to an inherently chaotic activity. The different process models are

- Waterfall model
- Prescriptive model
- Incremental model
- RAD model
- Prototyping
- Spiral model
- Component-based model, etc..

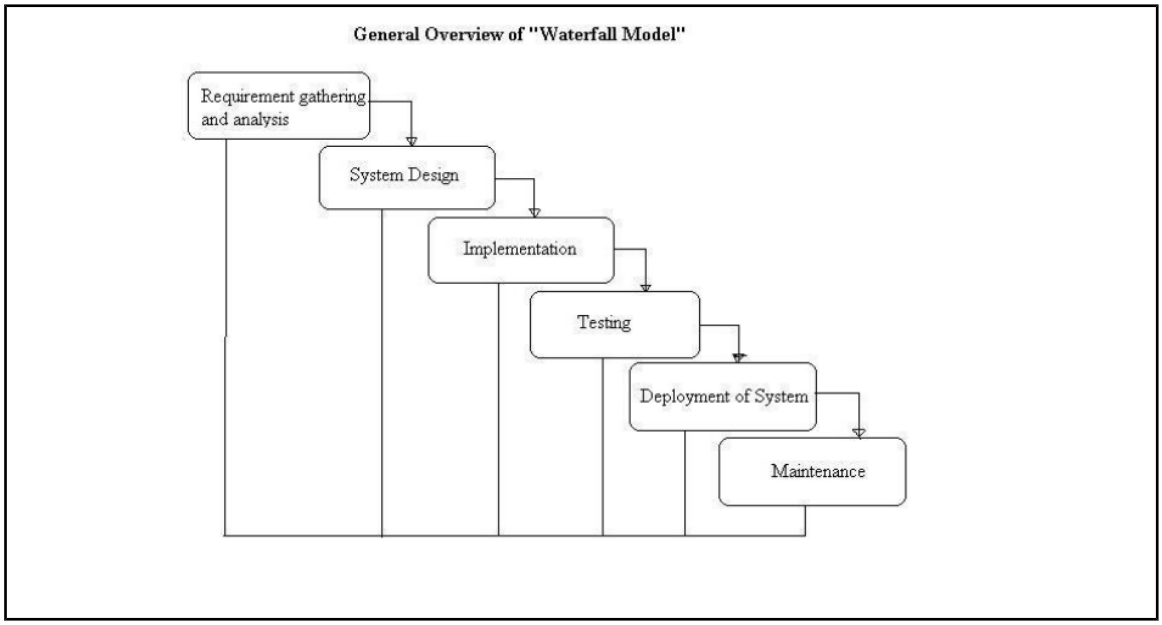
For our project, we have chosen Waterfall Model as our process model.

The waterfall model is a model which was developed for software development; that is to create software. It is called as such because the model develops systematically from one phase to other in a downward fashion, like a waterfall.

The most probable phases through which it progresses downwards are

- Definition Study/Analysis
- Basic Design
- Technical Design/Detailed Design

- Construction
- Testing
- Integration
- Management
- Maintenance.



WATERFALL MODEL

Before the advent of this method, the software development in the computer companies suffered
from a haphazard integrated software network like cluttered knitting. However with this method
they hoped to bring clarity in their projects.

Phases

As said earlier the waterfall model has been structured on multiple phases especially to help out the software construction companies to develop an organized system of construction. By following this method, the project will be divided into many stages thus easing out the whole process. For example you start with Phase I and according to this model, one only progresses to the next Phase once the previous one has been completed. This way one moves progressively to the final stage and once that point is reached, you cannot turn back; similar to the water in a waterfall.

Why Waterfall Model ?

It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of model is basically used for the for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model the testing starts only after the development is complete. In waterfall model phases do not overlap.

Brief Description of the Phases of Waterfall Model ; Definition Study / Analysis

During this phase research is being conducted which includes brainstorming about the software, what it is going to be and what purpose is it going to fulfill.

Basic Design: If the first phase gets successfully completed and a well thought out plan for the software development has been laid then the next step involves formulating the basic design of the software on paper.

Technical Design / Detail Design: After the basic design gets approved, then a more elaborated technical design can be planned. Here the functions of each of the part are decided and the engineering units are placed for example modules, programs etc.

Construction / Implementation: In this phase the source code of the programs is written. Testing: At this phase, the whole design and its construction is put under a test to check its functionality. If there are any errors then they will surface at this point of the process.

Integration: in the phase of Integration, the company puts it in use after the system has been successfully tested.

Management and Maintenance: Maintenance and management is needed to ensure that the system will continue to perform as desired. Through the above mentioned steps it is clearly shown that the Waterfall model was meant to function in a systematic way that takes the production of the software from the basic step going downwards towards detailing just like a Waterfall which begins at the top of the cliff and goes downwards but not backwards.

Advantages of the Waterfall Model

- This model is simple and easy to understand and use.
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables
- and a review process.
- In this model phases are processed and completed one at a time. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are very well understood.

Disadvantages of the Waterfall Model

Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.

- No working software is produced until late during the life cycle. High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

When to use the waterfall model

- This model is used only when the requirements are very well known, clear and fixed.
- Product definition is stable.
- Technology is understood.
- There are no ambiguous requirements
- Ample resources with required expertise are available freely
- The project is short.

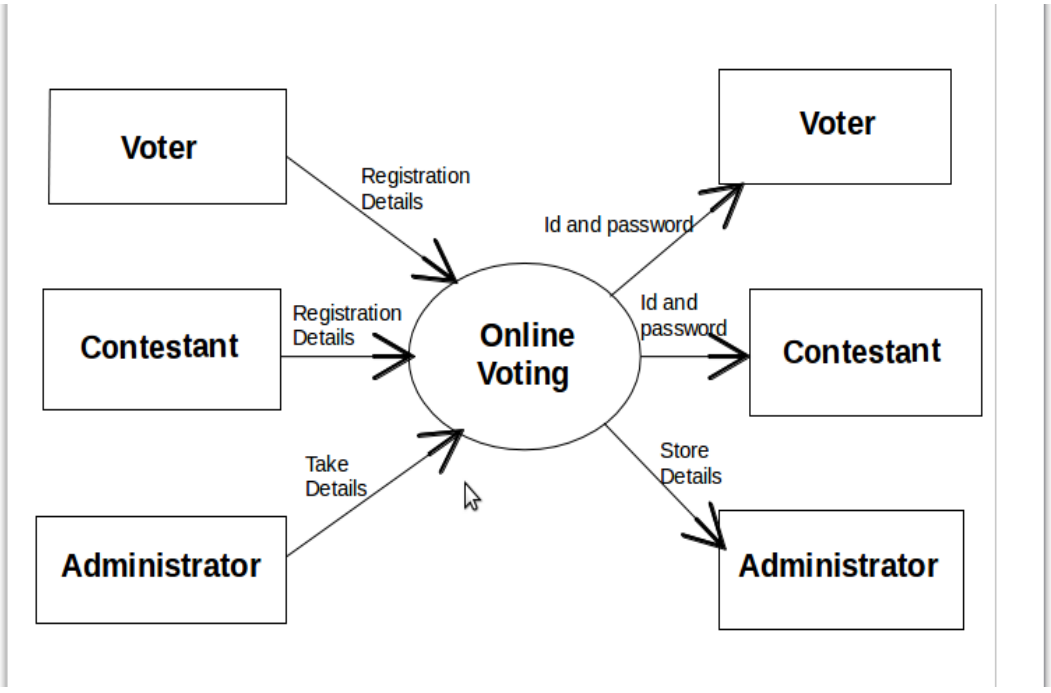
Data Flow Diagram

What is a data flow diagram?
A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.

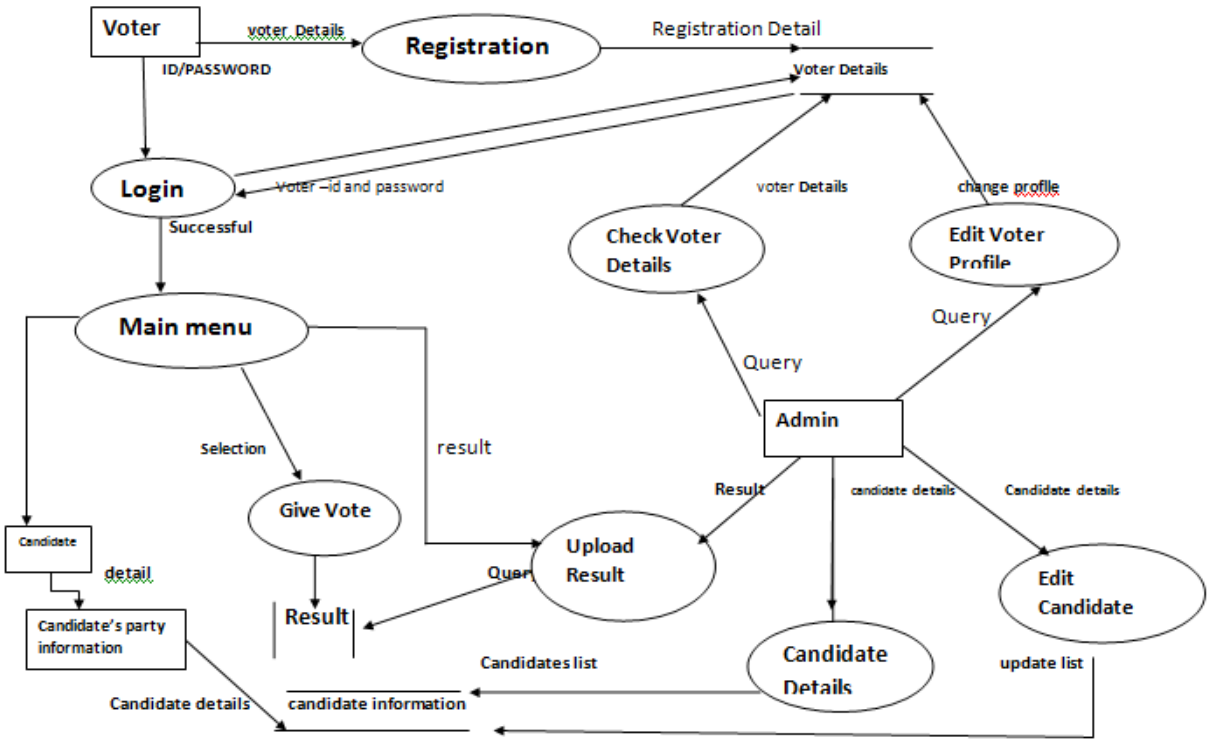
Context Diagram
A context diagram is a top level (also known as "Level 0") data flow diagram. It only contains one process node ("Process 0") that generalizes the function of the entire system in relationship to external entities.

DFD Levels
The first level DFD shows the main processes within the system. Each of these processes can be broken into further processes until you reach pseudo code.

0-LEVEL DFD

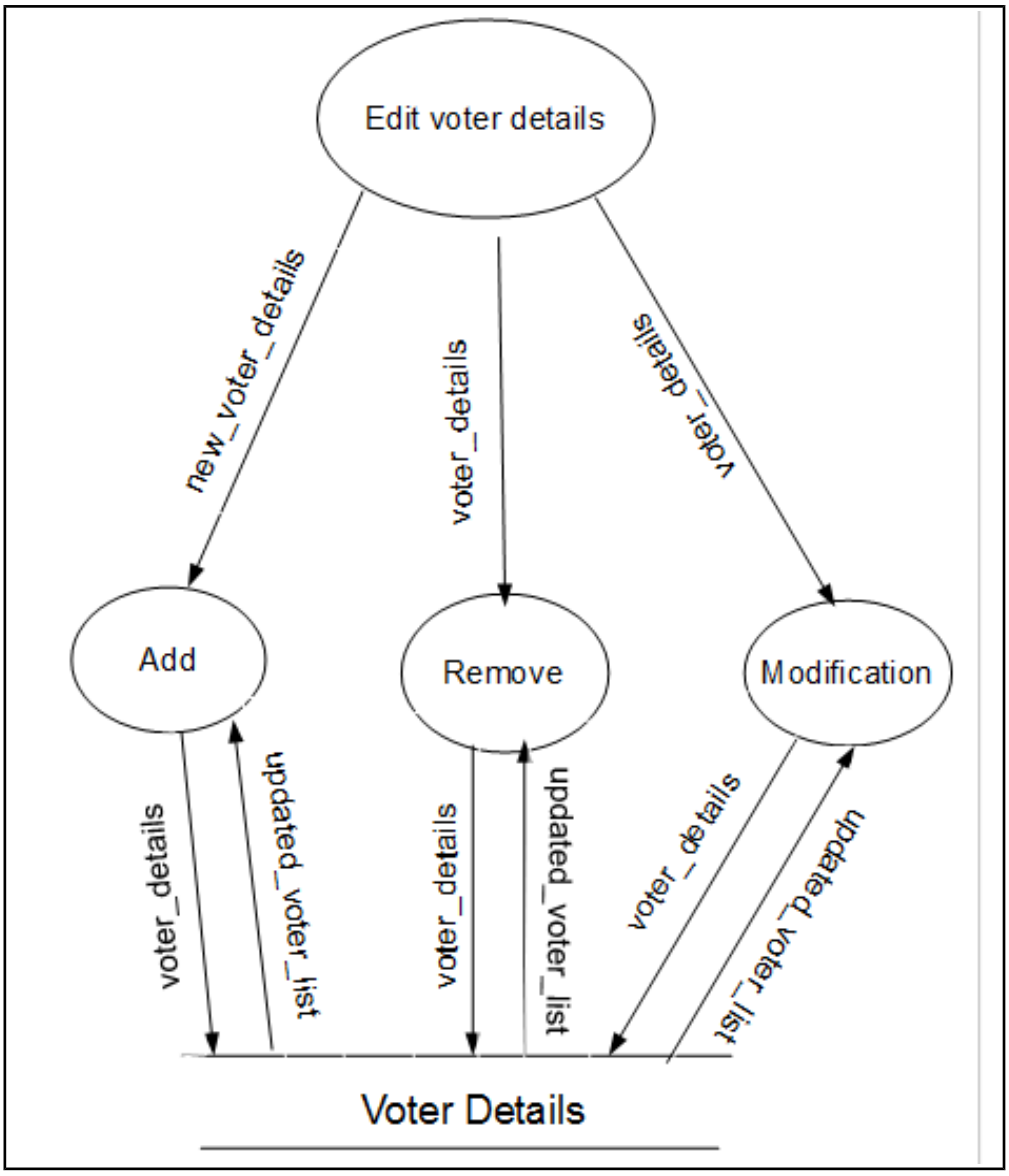


1-LEVEL DFD

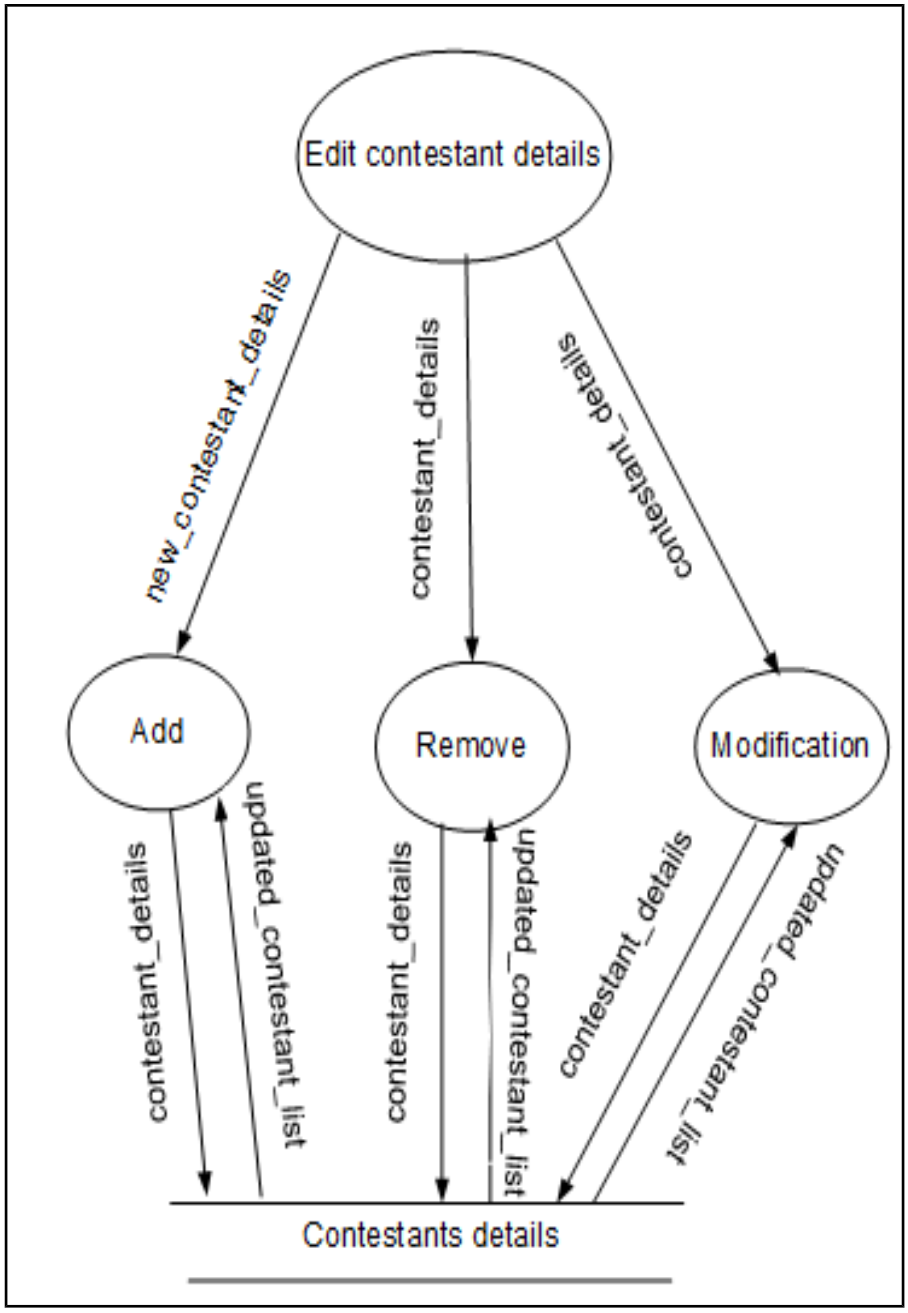

Voter
voter Details
Registration
Registration Detail
ID/PASSWORD
Voter Details
Login
Voter –id and password
voter Details
change profile
Successful
Check Voter Details
Edit Voter Profile
Main menu
Query
Query
Admin
Selection
result
Give Vote
Result
candidate details
Candidate details
Candidate
detail
Upload Result
Edit Candidate
Candidate's party information
Result
Candidate Details
Candidates list
update list
Candidate details
candidate information

2-LEVEL DFD

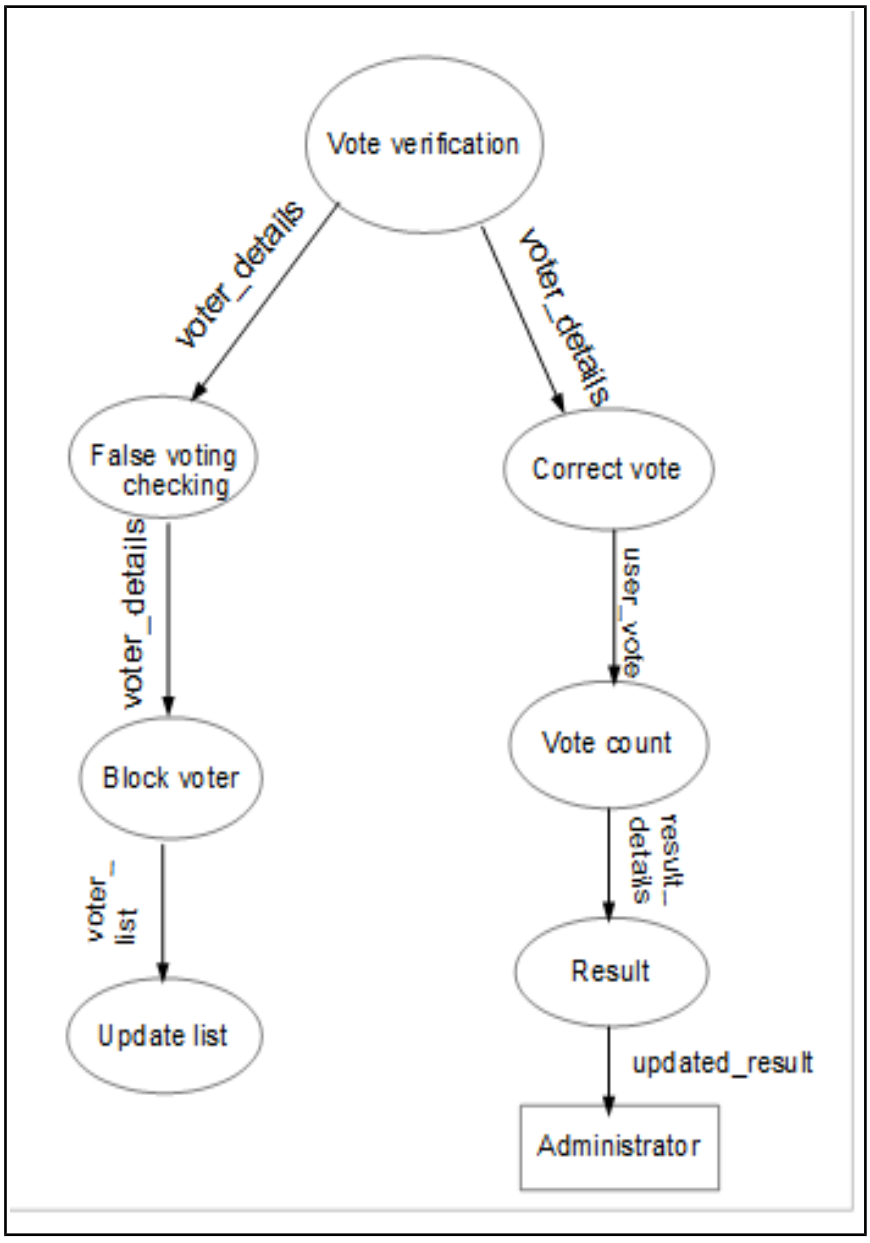
- **Edit Voter Details**



- **Edit Contestant Details**



- **Vote Verification**



DATA DICTIONARY

A set of information describing the contents, format, and structure of a database and the relationship between its elements, used to control access to and manipulation of the database.

The data dictionary is a crucial component of any relational database. Ironically, because of its importance, it is invisible to most database users. Typically, only database administrators interact with the data dictionary.

Most of the data dictionary contains the following information:

- Name - the primary name of the data or control item, the data store or an external entity
- Alias - other names used for the first entry.
- Where used - a listing of the processes that use the data or control item and how it is used external.
- Description - a notation for representing content.

NAME	ALIAS	WHERE USED	DESCRIPTION
Identity and Password for voter	ID/Password	Issued during log in and registration	It is specific to each voter and it is used as a log in detail.
Identity and Password for candidate	ID/Password	Issued during log in and registration	It is specific to each candidate and it is used as a log in detail.
Result of on line voting	Result	Declared at the end of the vote count procedure.	Declares the winner of the specific positions.
Details of the voter	voter_Details	Used at the time of registration and log in	Contains: First,Middle and Last Name of the voter, age, College

			id, student id, department name
Details of the contestant	contestant_Details	Used at the time of registration and log in	Contains: First,Middle and Last Name of the candidate, age, College id, student id, department name, Position name
Details of the Registration	Registration_Details	Used at the time of registration	Contains : Registration ID, Voter/Candidate Name, Student ID
Query generated by voter/Candidate	Query	Used at the time of checking and editing voter details	Contains: Questions generated by voter/ candidate.
Details of a new voter	new_voter_details	Used at the time of editing and updating voter list.	Contains: First,Middle and Last Name of the voter, age, College id, student id, department name
Details of a new contestant	new_contestant_details	Used at the time of editing and updating contestant list.	Contains: First,Middle and Last Name of the candidate, age, College id, student id, department name, Position name
Updated list of the voters	updated_voter_list	Used at the time of editing and updating the voter list.	Contains the list of voters and their details.
Updated list of the contestants	updated_contestant_list	Used at the time of editing and updating the contestant details.	Contains the list of contestants and their details.
Vote of the user	user_vote	Used at the time of voting and verification of votes.	It is the vote given by the user- voter or contestant.
List of the voters.	voter_list	Used at the time of voting and verification of votes.	It is the list of the voters.
Result updated.	updated_result	Used at the time of vote verification, counting of votes and declaration of results.	It is the updated result.

FUNCTION POINT

What Are Function Points?

- Function Points measure software size by quantifying the functionality provided to the user based solely on logical design and functional specifications.
- Function point analysis is a method of quantifying the size and complexity of a software system in terms of the functions that the system delivers to the user.
- It is independent of the computer language, development methodology, technology or capability of the project team used to develop the application.

Uses of Function Points

- Measure productivity (ex. Number of function points achieved per work hour expended)
- Estimate development and support (cost benefit analysis, staffing estimation)
- Monitor outsourcing agreements (Ensure that the outsourcing entity delivers the level of support and productivity gains that they promise)
- Normalize other measures (Other measures, such as defects, frequently

require the size in function points)

Analyzing the Information Domain

WEIGHTING FACTOR

Measurement parameter	count	SIMPLE	AVERAGE	COMPLEX	TOTA
No. Of User Inputs	=3	**3**	4	6	9
No. Of User Outputs	=3	**4**	5	7	12
No. Of user inquiries	=3	**3**	4	6	9
No. Of files	=3	**7**	10	15	21
No. Of External files	=0	**5**	7	10	0
Count Total					**51**

where,

Number of user inputs
Each user input that provides distinct application oriented data to the software is counted. Inputs should be distinguished from inquiries, which are counted separately.

Number of user outputs
Each user output that provides application oriented information to the user is counted. In this context output refers to reports, screens, error messages, etc. Individual data items within a report are not counted separately.

Number of user inquiries.
An inquiry is defined as an on-line input that results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry is counted.

Number of files

Each logical master file (i.e., a logical grouping of data that may be one part of a large database or a separate file) is counted.

Number of external interfaces.

All machine readable interfaces (e.g., data files on storage media) that are used to transmit information to another system are counted.

To compute **function points (FP)**, the following relationship is used:

$$\mathbf{FP = count\ total\ X\ [0.65+.01\ X\ \sum(Fi)]}$$

The Fi (i =1 to 14) are “complexity adjustment values” based on response to the following questions:

1. Does the system require reliable backup recovery? =4
2. Are data communications required? =5
3. Are there distributed processing functions? =5
4. Is performance critical? =0
5. Will the system run in an existing, heavily utilized operational environment? =1
6. Does the system require on-line data entry? =5
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations? = 5
8. Are the master files updated on-line? =0
9. Are the inputs, outputs, files, or inquiries complex? =0
10. Is the internal processing complex? =0
11. Is the code designed to be reusable? =0
12. Are conversion and installation included in the design? =0
13. Is the system designed for multiple installations in different organizations? =2
14. Is the application designed to facilitate change and ease of use by the user? =3

Each of these questions is answered using a scale that ranges from

0- Not present, or no influence,
1- Incidental influence,
2- Moderate influence,
3- Average influence,

4- Significant influence,
5- Strong influence throughout.

The weighting factors for input and output, for user inquiries, number of files and external interfaces weighting factor is **simple**.
FP = count total X [0.65+.01 X ∑(Fi)]
∑(Fi)=30
FP =51 X [0.65+.01 X 23]
FP=**48.45**

EFFORT ESTIMATION

Effort Estimation is a method for estimating the duration of software engineering projects. It is best suited to producing initial estimates for the length of a job based on known time duration for preparing a specification.
For our software, we calculate Effort (E) using the following formula:

$$E = -12.88 + 0.405\ FP$$

As calculated earlier, the function point (FP) of our project = 48.45
Hence, Effort (E) = -12.88 + 0.405 (48.45)
E = **6.742**
Total months = 40% of effort
= 0.4*6.742
= **2.69**
Total members = **3**
Members per month = 2.69/3 = **0.89**

Risk Table

Risk is a probabilistic event i. e. it may occur or may not occur. We frequently have optimistic tendency to simply not see risk or wish they will not occur, which later on leads us to trouble. Hence it is advisable to identify the critical areas, assess their probabilities, estimate the impact and plan the contingency plan.
The first step is Risk identification. Next each risk is analyzed to determine the likelihood that it will occur and the damage it will do if it occurs. Risks are then, ranked by probability and impact. Finally a plan is developed to manage those risks with high probability, moderate as well as low impact.

These risk analysis activities assist us in developing a strategy for dealing risks. An effective strategy is the RMMM plan.
It Includes:

- Risk identification
- Risk monitoring
- Risk management and contingency plan

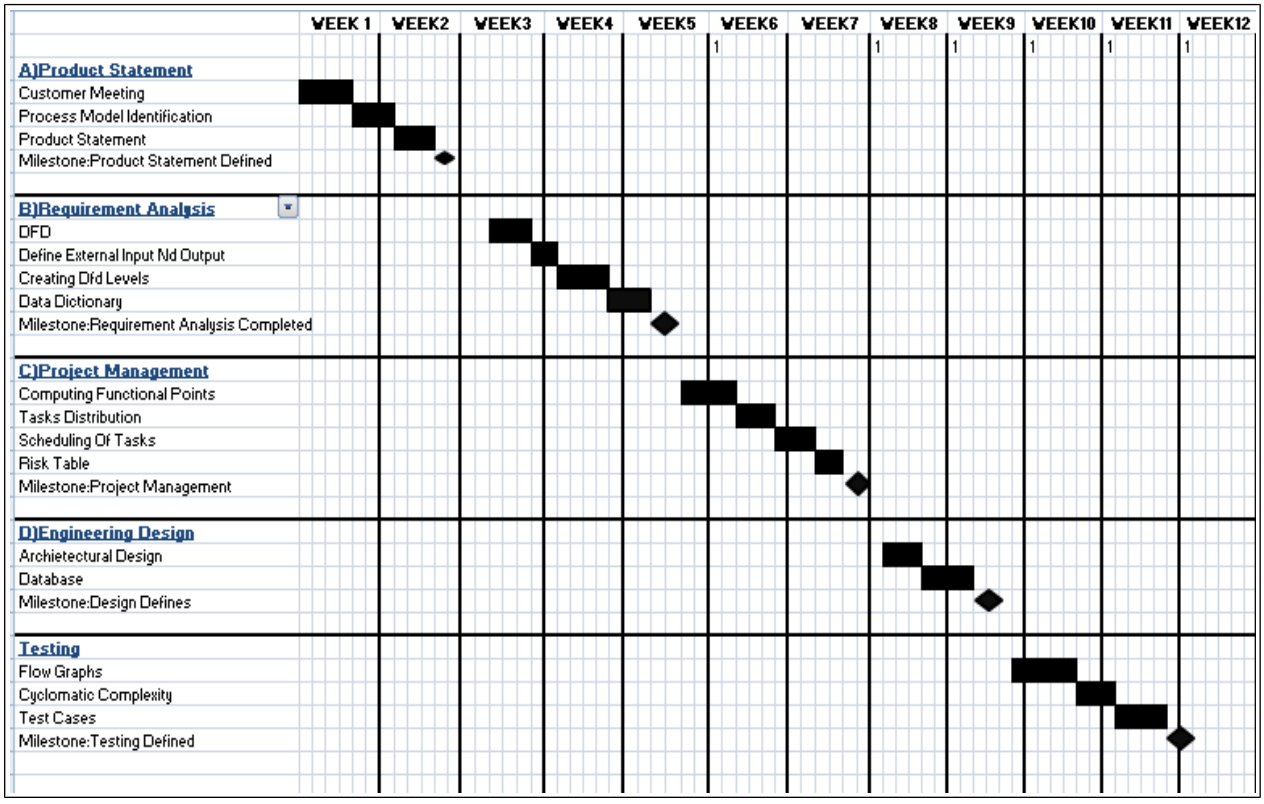
Impact: Five Categories –

- Negligible (1)
- Normal (2)
- Marginal (3)
- Critical (4)
- Catastrophic (5)

Risk	Probabilty	Impact	RMMM
1. Login Risk	20.00%	2	1. User can recover his/her password. Also, user should save it as some other safe place so that in case the user forgets his/her password, he/she can refer to that place. 2. User can recover password answering the recovery question. 3. User can re-register.
2. Loss of Database	20.00%	4	1. Restore data from the backup database. 2. Monitoring the database whenever some user logs in or whenever the software is being updated. 3. The database has to be restored.
3. Technical Risk	20.00%	3	1. Any operation performed by the user must be auto-saved. 2. Software must save user data after some time by default. 3. Auto-saving of the data must be done.
4. Congestion risk	40.00%	4	1. Proper internet facility

			should be available. 2. There should be multiple links for voting. 3. The system should be able to recover instantly if any fault occurs.

TIMELINE CHART



Architectural Design

Software architectural design represents the structure of the data and program components that are required to build a computer-based system.

An architectural design model is transferable that is it can be applied to the design of other systems. It represents a set of abstractions that enable software engineers to describe architecture in predictable ways.

Architecture design for the software Online Election System is shown on the next page.

Database Design

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data.

In the relational model these are the tables and view. In an object database the entities and relationships map directly to object classes and named relationships.

However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system (DBMS).

The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

- Determine the relationships between the different data elements.
- Superimpose a logical structure upon the data on the basis of these relationships.

ER-DIAGRAM

Database designs also include ER (entity-relationship model) diagrams. An ER diagram is a diagram that helps to design databases in an efficient way.

Attributes in ER diagrams are usually modelled as an oval with the name of the attribute, linked to the entity or relationship that contains the attribute.
Within the relational model the final step can generally be broken down into two further steps that of determining the grouping of information within the system, generally determining what are the basic objects about which information is being stored, and then determining the relationships between these groups of information, or objects. This step is not necessary with an Object database.

Database table for ER-DIAGRAM

CONTESTANT details

Name	char	NULL
Age	integer	NULL
College-ID	char	NULL
Position	char	NULL
Sex	char	NULL
Year	integer	NULL
Department	char	NULL

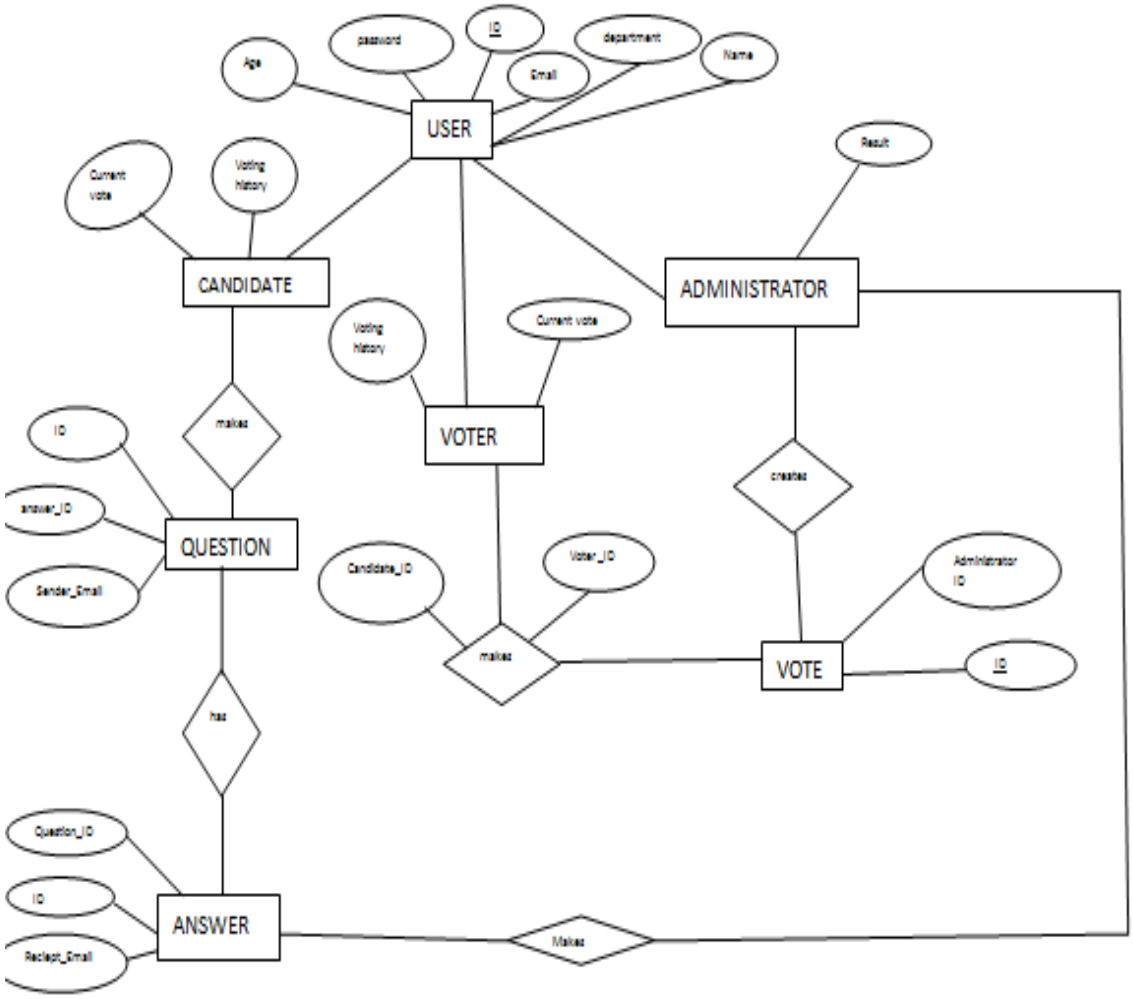
VOTER Details

Name	char	NULL
Age	integer	NULL
College-ID	char	NULL
Sex	char	NULL
Year	integer	NULL
Department	char	NULL

RESULT

Position	integer(3)	NULL
Candidate Name	char	NULL
MaximumVote	integer	NULL
Winner name	char	NULL

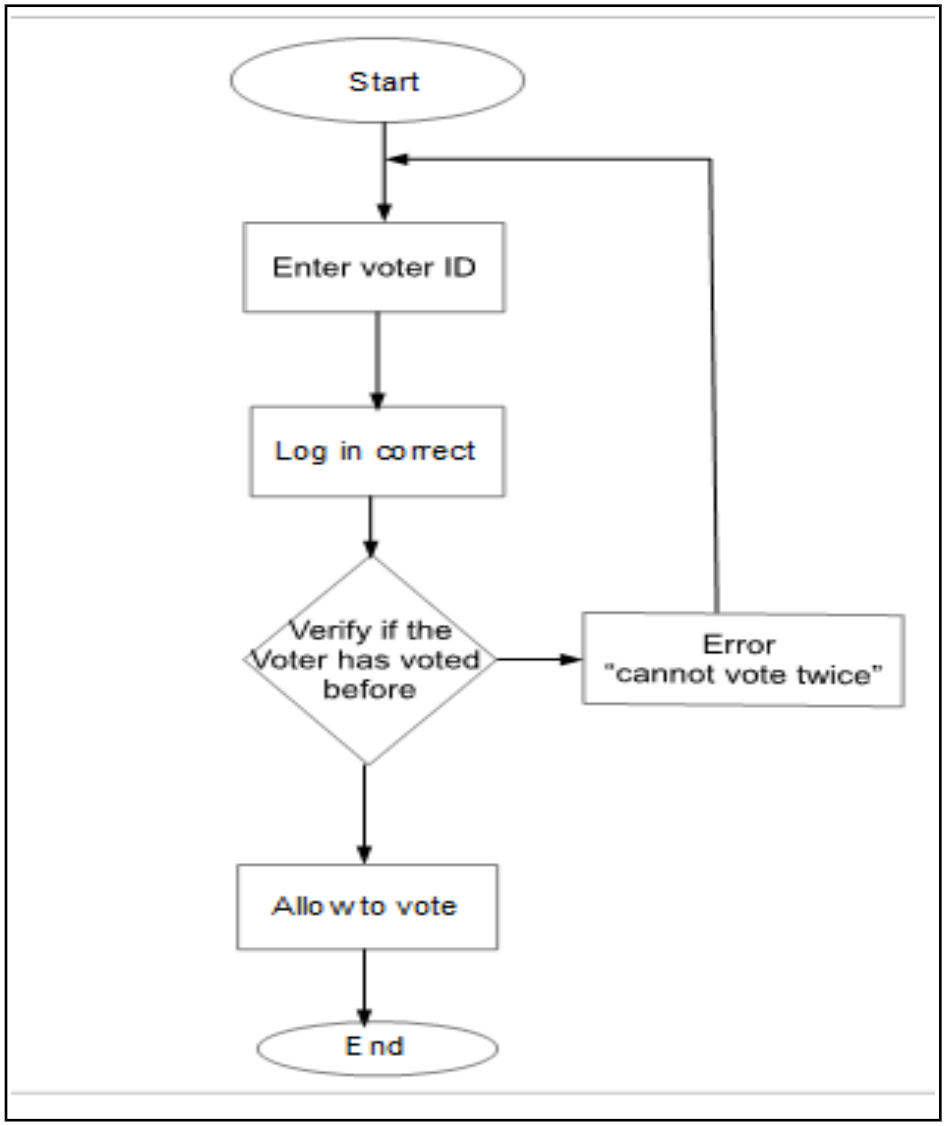
ER-DIAGRAM



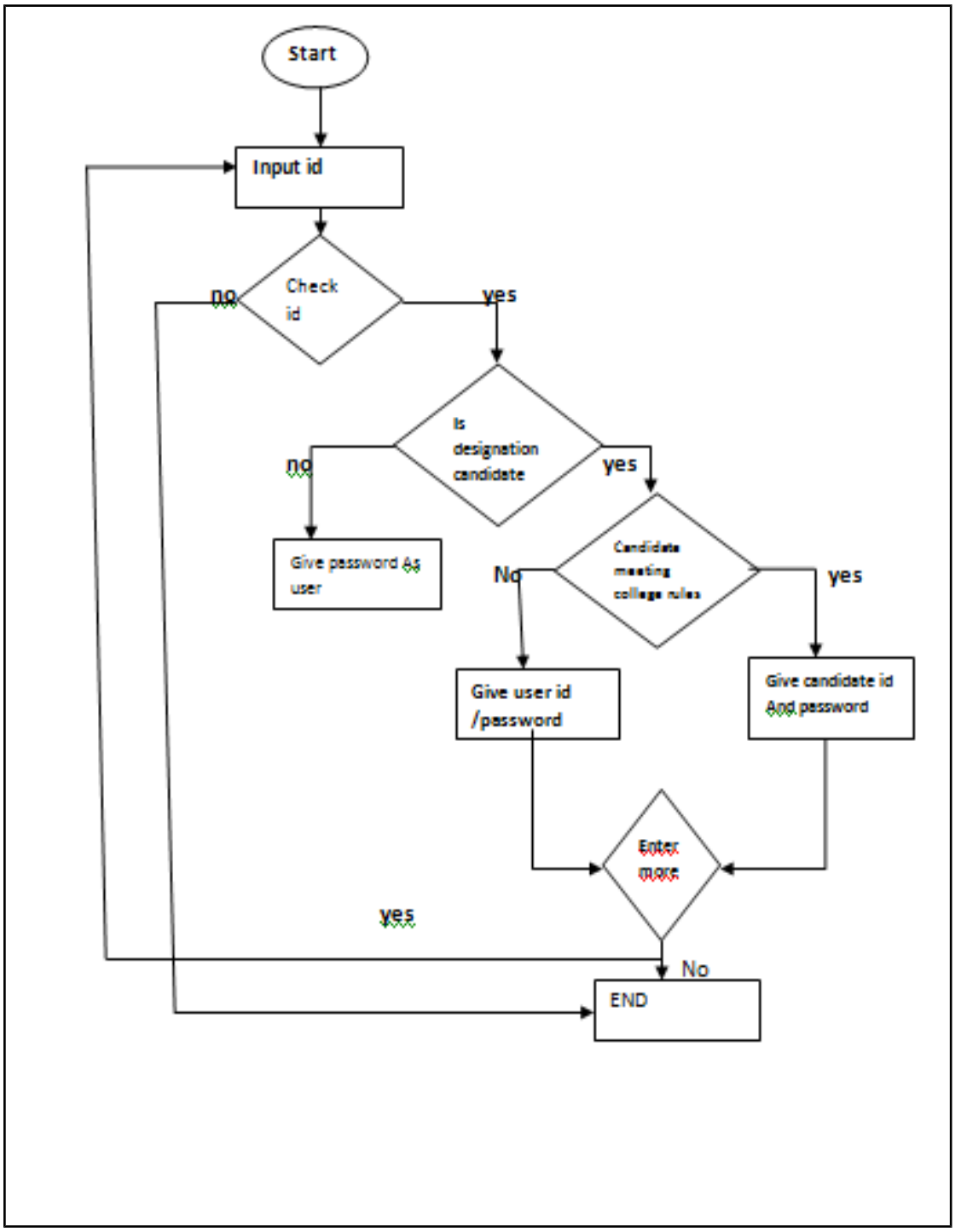
I

Flow Charts

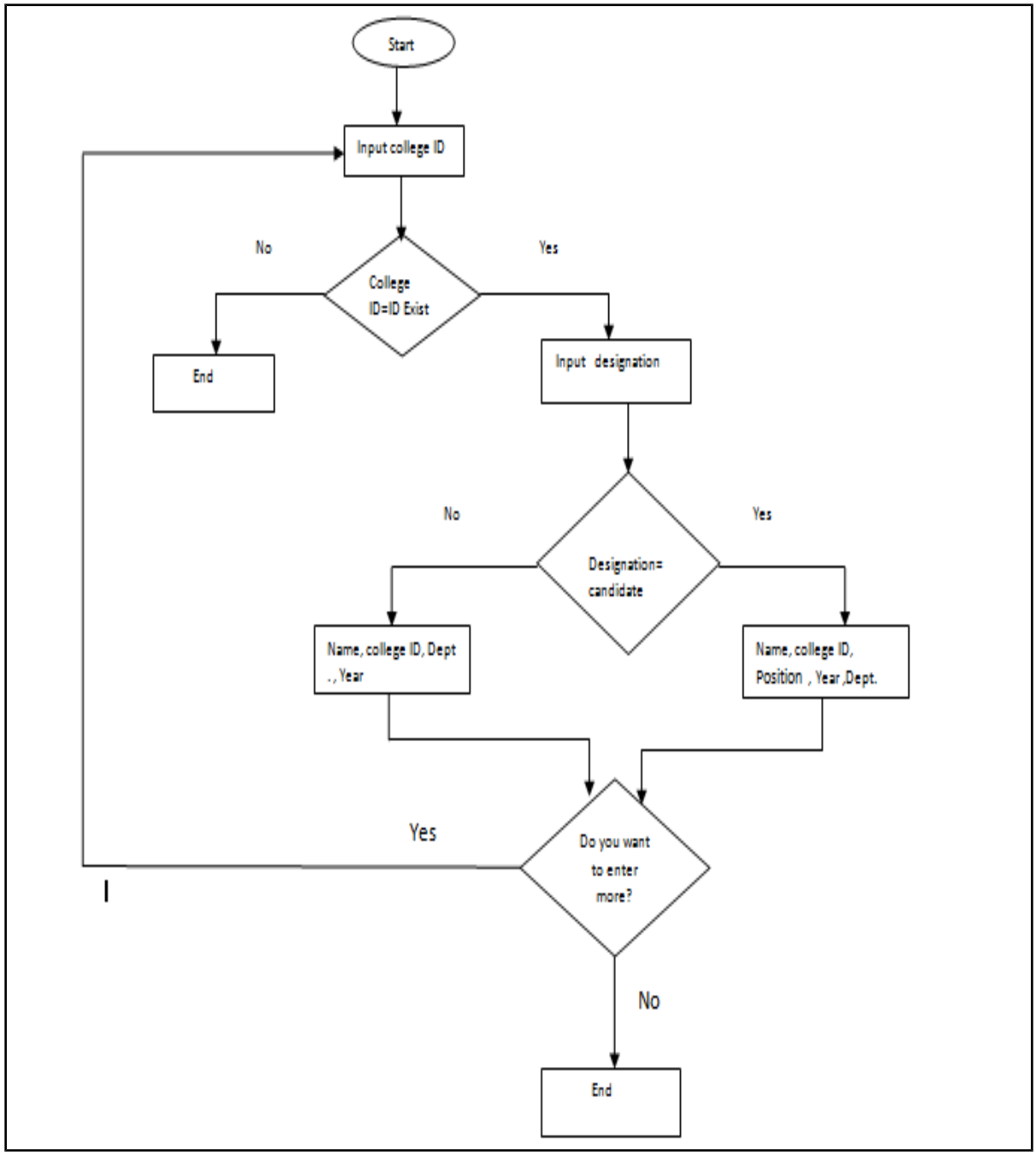
- **Give Vote**



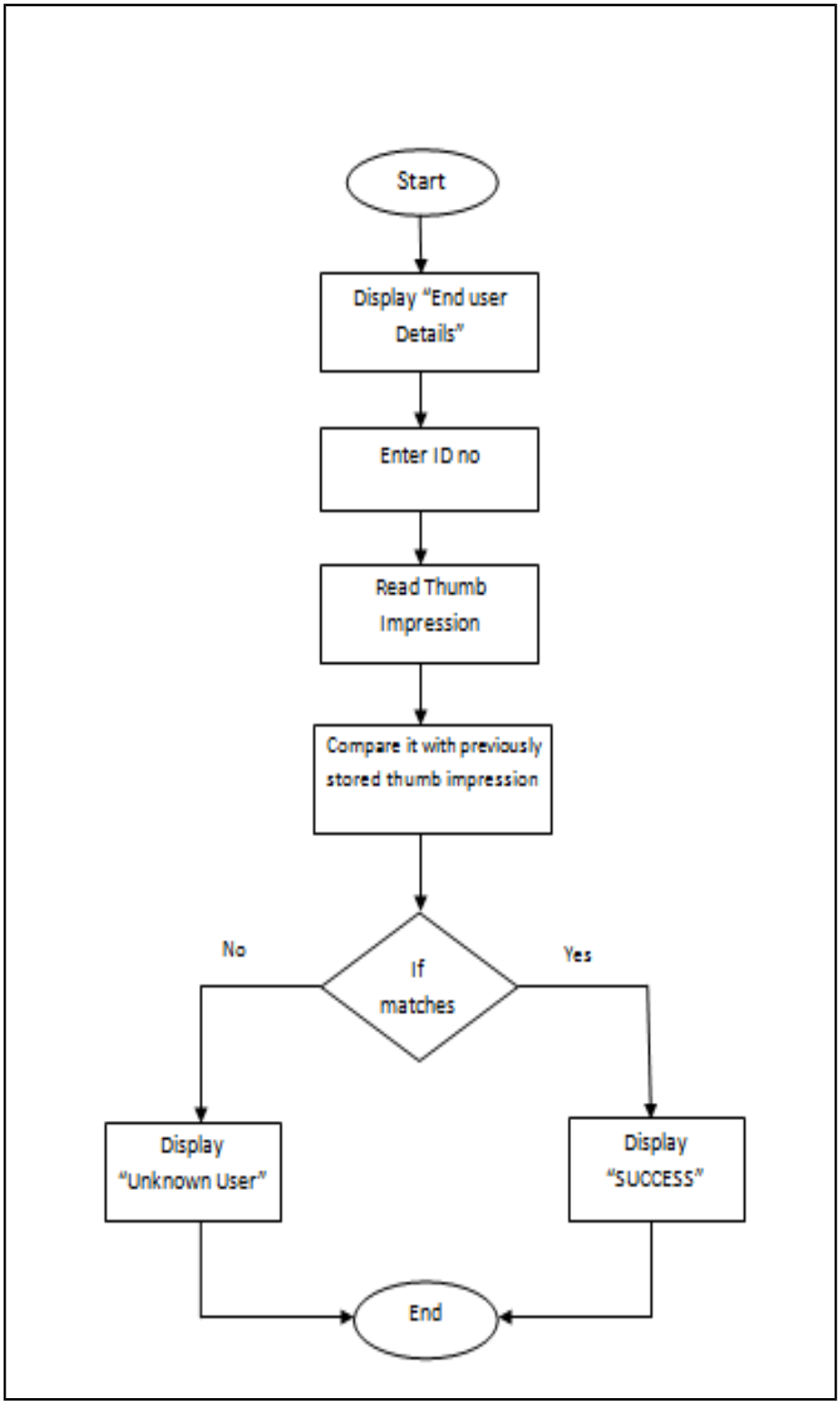
- **Login**



- **Registeration**

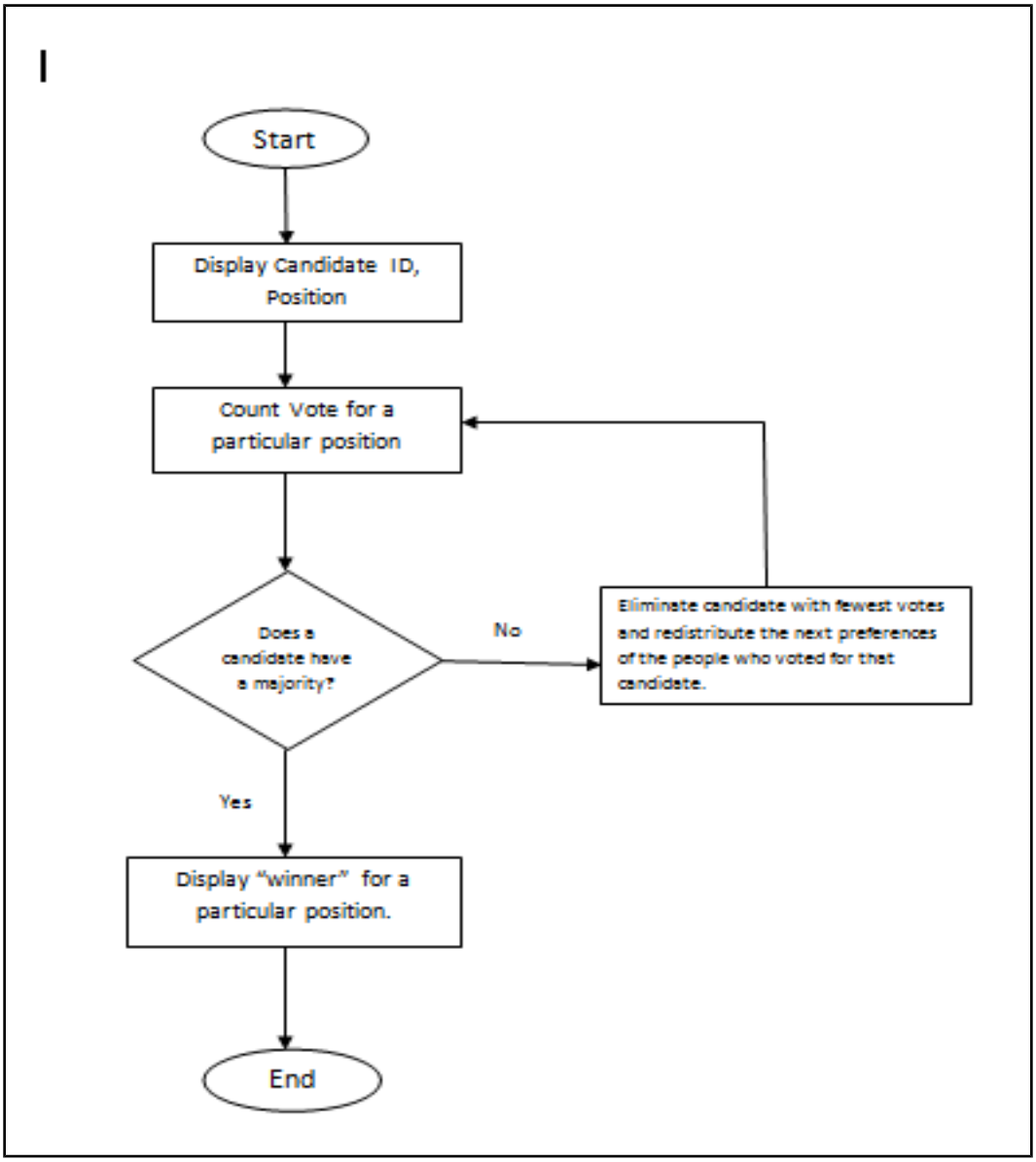


- **Check Voter Details**

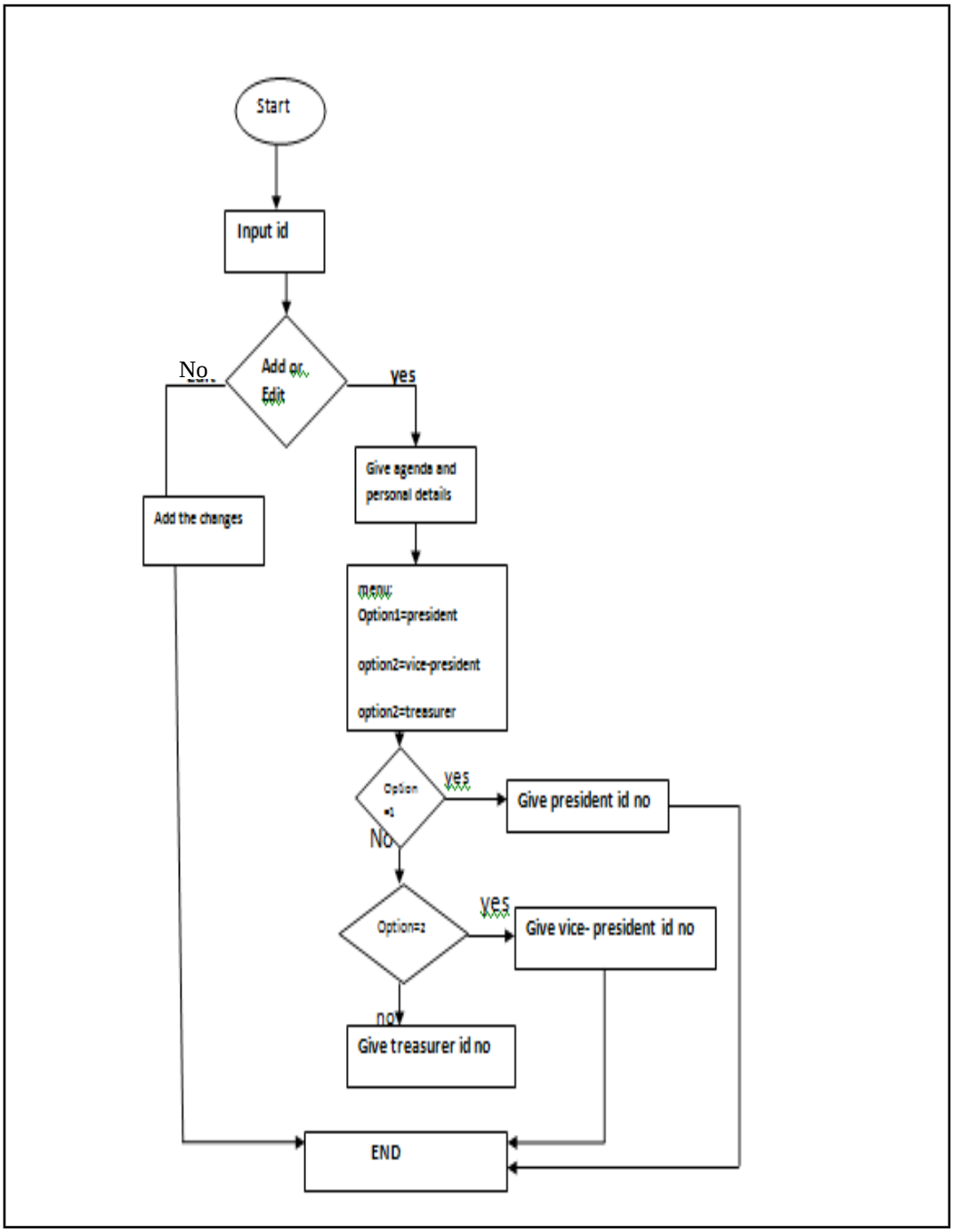


- **Upload Result**

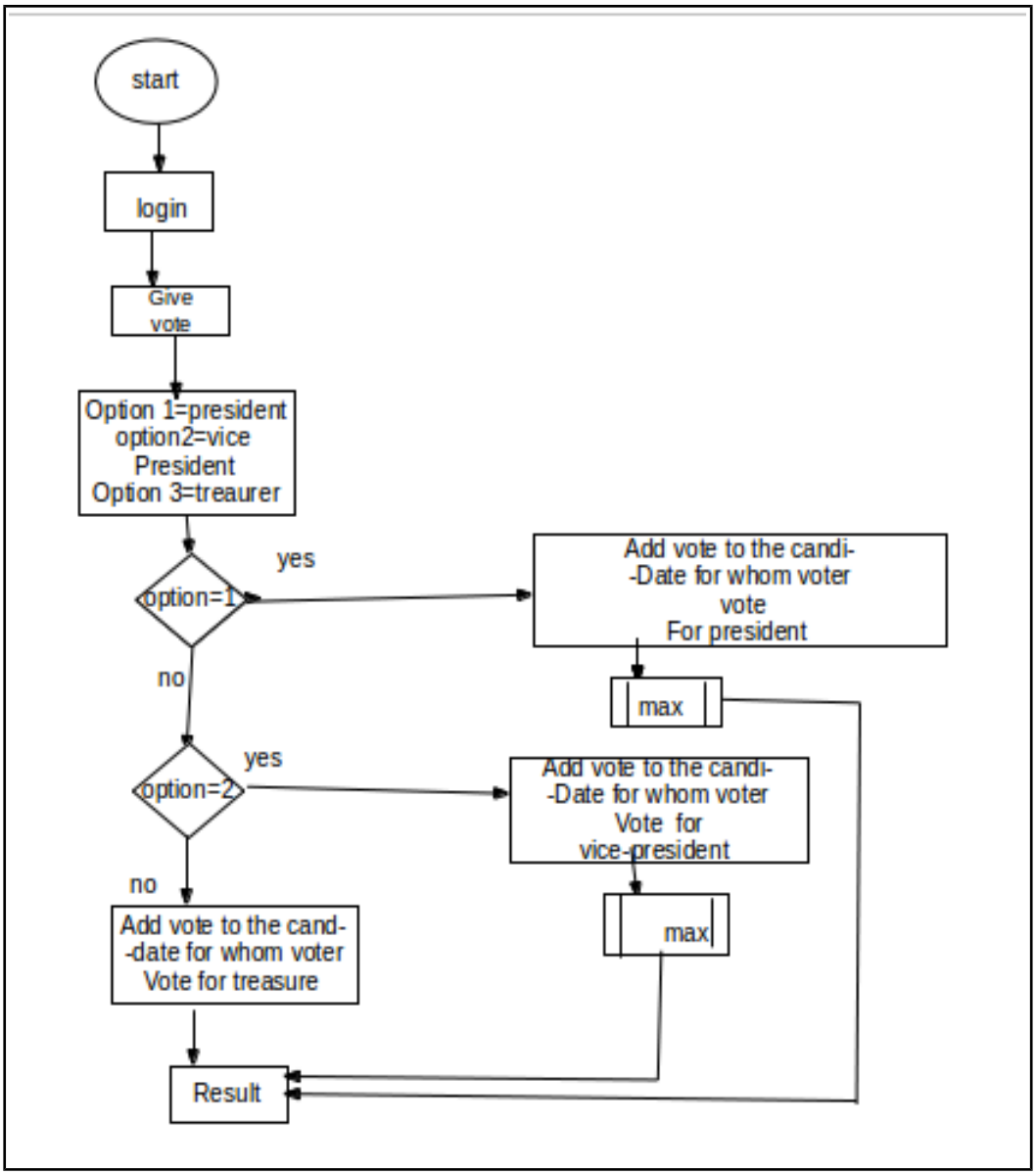
I



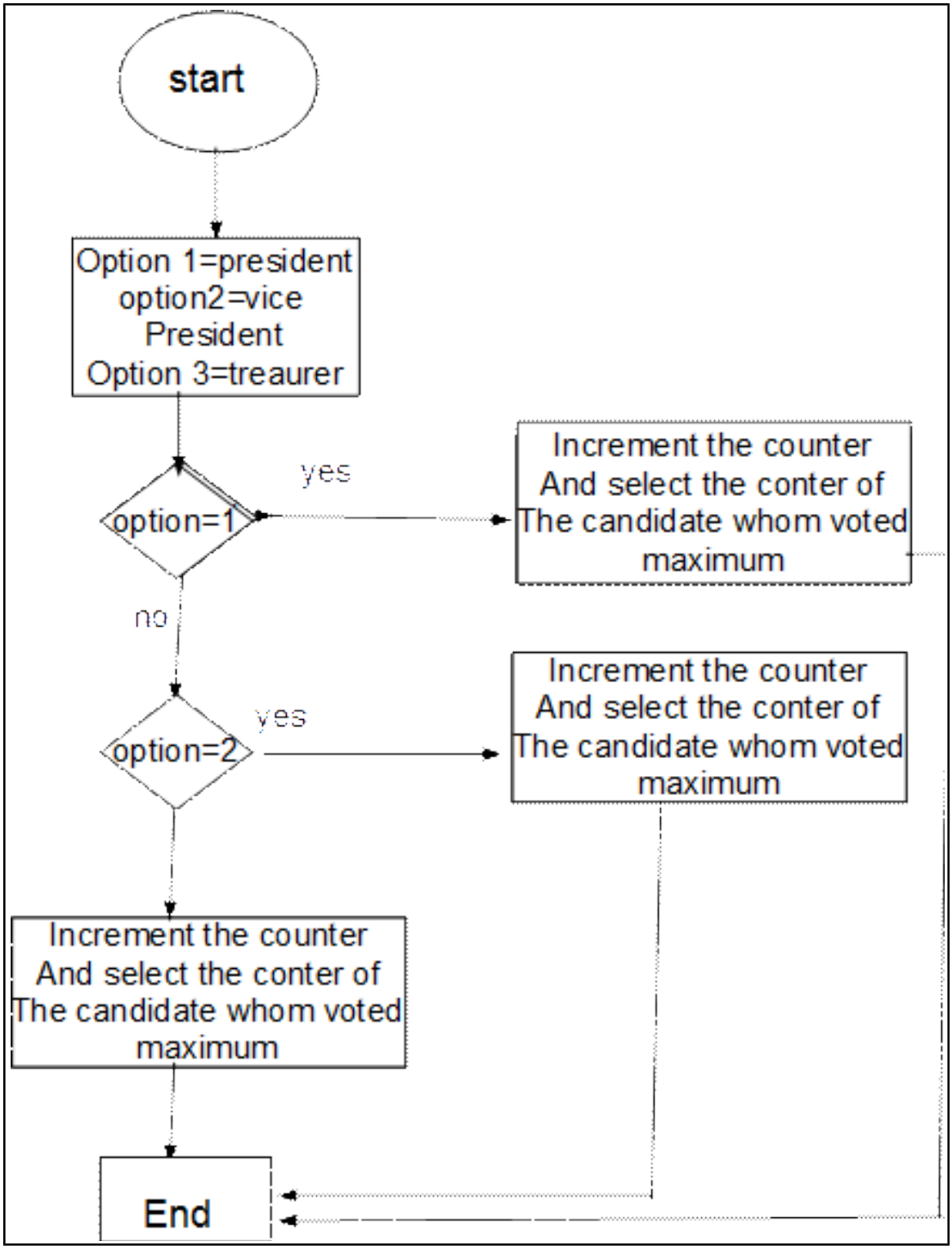
- **Contestant Information**


Start
Input id
Add or Edit
No
yes
Add the changes
Give agenda and personal details
menu
Option1=president
option2=vice-president
option2=treasurer
Option =1
yes
Give president id no
No
Option=2
yes
Give vice- president id no
no
Give treasurer id no
END

- **Vote count**


start
login
Give
vote
Option 1=president
option2=vice
President
Option 3=treaurer
option=1
yes
Add vote to the candi-
-Date for whom voter
vote
For president
max
no
option=2
yes
Add vote to the candi-
-Date for whom voter
Vote for
vice-president
max
no
Add vote to the cand-
-date for whom voter
Vote for treasure
Result

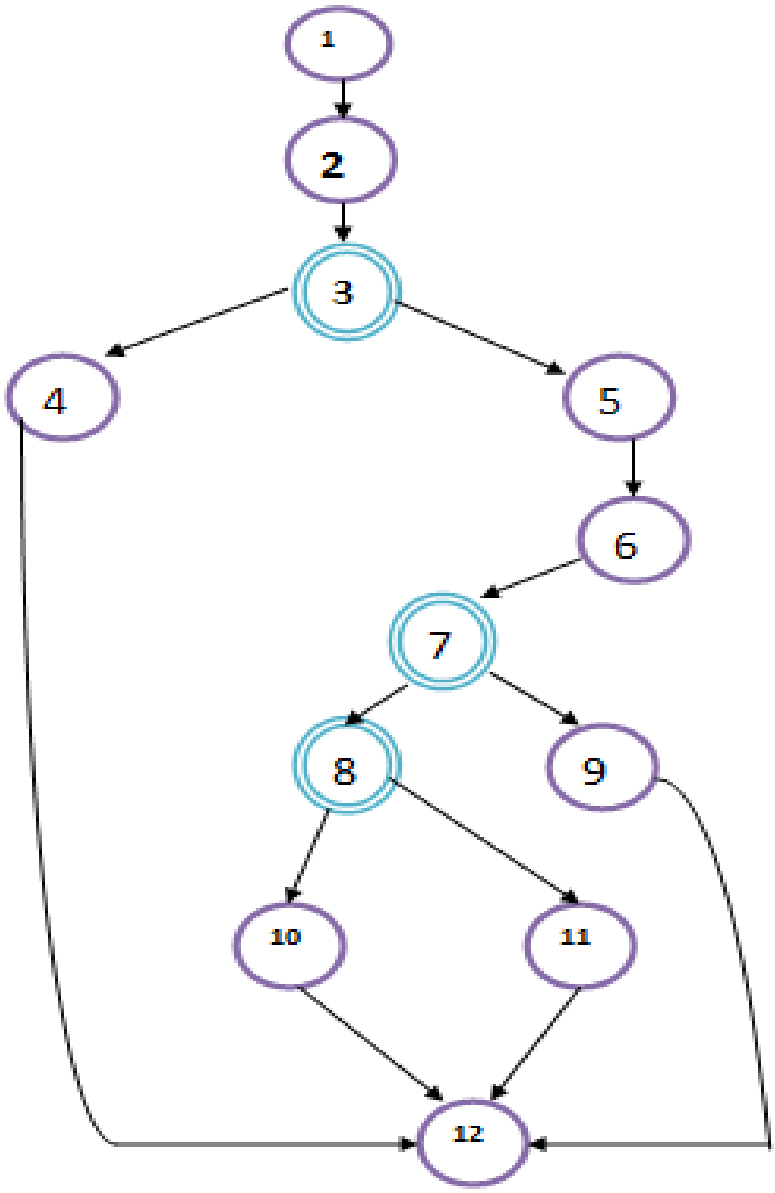
- **Maximum voted Contestant**



Flowgraph

In a control flow graph each node in the graph represents a basic block, i.e. A straight-line piece of code without any jumps or jump targets; jump targets start a block, and jumps end a block. Directed edges are used to represent jumps in the control flow. There are, in most presentations, two specially designated blocks: the entry block, through which control enters into the flow graph, and the exit block, through which all control flow leaves.

- **Flow graph for “candidate information”:**



Cyclomatic Complexity

Cyclomatic complexity is computed using the control flow graph of the program, the nodes of the graph correspond to individual groups of commands of a program and a directed edge connects two nodes if the second command might be executed immediately after the first command. Cyclomatic complexity may also be applied to individual functions, modules, methods or classes within a program.

One testing strategy, called Basis Path Testing by McCabe, who first proposed it, is to test each linearly independent path through the program; in this case, the number of test cases will equal the cyclomatic complexity of the program.

Formula is given by:

No. of paths = predicate nodes + 1

In the above given flow graph we have predicate nodes = 3
therefore, the number of paths will be 3+1 =4

V(G) for the flow graph is defined by-

V (G) =E – N+ 2

where,
E= NO. OF EDGES= 14
N= NO.OF NODES= 12

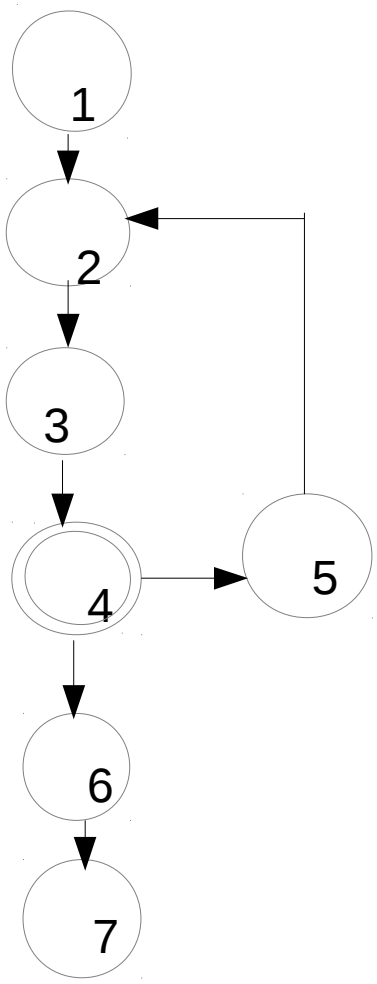
Therefore,

V(G)= 4, which means **INDEPENDENT PATH** is 4

Now the paths are :

Path 1 : 1->2->3->4->12
Path 2 : 1->2->3 ->5 ->6-> 7 ->8 ->10-> 12
Path 3 : 1- >2 ->3-> 5-> 6-> 7-> ->8-> 11-> 12
Path 4 : 1->2->3->5->6->9->12

- **Flow graph for “Give Vote”:**



Cyclomatic complexity

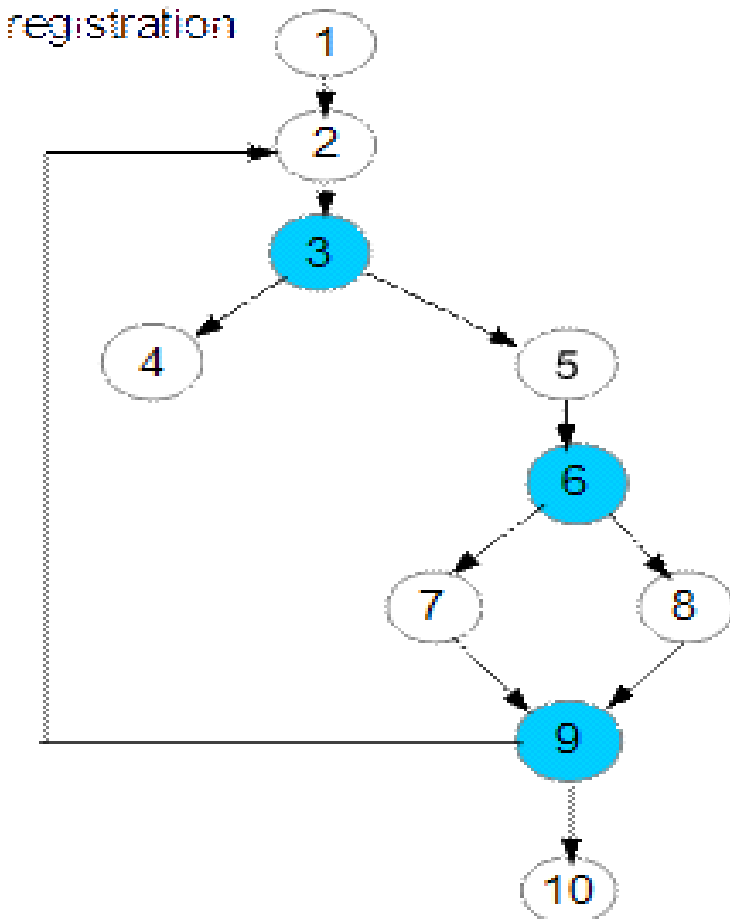
1. CC(Closed Region)= 2
2. CC(Edge-node+2)= 7-7+2= 2

Cyclomatic complexity=2

Independent paths

Path 1 : 1->2->3->4->6->7
Path 2 : 1->2->3->4->5->2->3->4->6->7

- **Flow graph for “Registration” :**



Cyclomatic complexity

1. CC(Closed Region)=4
2. CC(Edge-node+2)=12-10+2=4

Cyclomatic complexity=4

Independent paths

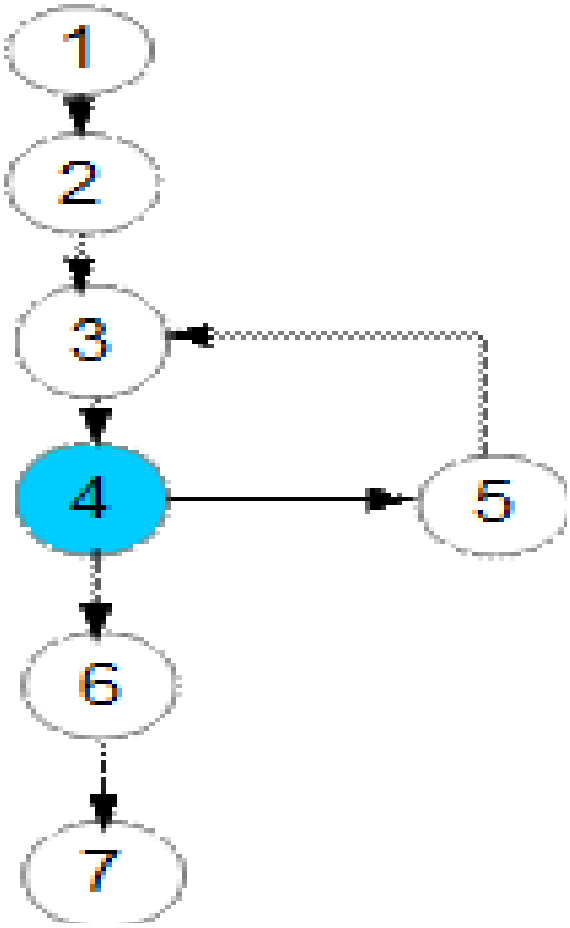
Path 1: 1->2->3->4

Path 2: 1->2->3->5->6->7->9->10

Path 3: 1->2->3->5->6->8->9->10

Path 4: 1->2->3->5->6->7->2.....

- **Flow graph for “Upload Result”**



Cyclomatic complexity

1.CC(Closed Region)=2
2.CC(Edge-node+2)=7-7+2=2
Cyclomatric complexity=2

Independent paths

Path 1: 1->2->3->4->6->7
Path 2: 1->2->3->4->5->3->4->6->7

Test Cases

White box testing(aka clear box testing, glass box testing, transparent box testing and structural testing) is a method of testing software that tests internal structures of working of an application, as opposed to its functionality(black box testing).

In white box testing, an internal perspective of the system as well as programming skills, are used to design test cases. The tester chooses inputs to ex, an internal perspective of the system as well as programming skills, are used to design test cases. The tester chooses inputs to excise the paths through the codes and determine the appropriate outputs. This is analogous to testing nodes in a circuit. Eg. In circuit testing (ICT).

White box testing can be applied at the unit, integration and system levels of the software process. Although traditional testers tended to think of white box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration and between subsystems during a system level test. Though this method of test design c an uncover many errors of problems. It has potential to miss unimplemented parts of the specifications or missing requirements.

- **Test case for “Give Vote”**

Path	Condition	Test cases	Statement	Status
1) 1->2->3->4->6->7	If voter has voted before	NO	Allow to vote	successful
2) 1->2->3->4->5->2->3->4->6->7	If voter has voted before	YES	Error message-cannot vote twice	continue with further voting

- **Test case for “Registration”**

Path	Condition	Test cases	Statement	Status
1) 1->2->3->4	College ID exist	NO	End the process	unsuccessful
2) 1->2->3->5->6->8->9->10	Colllege ID exist and designation is candidate	YES	Input candidate login ID	successful
3) 1->2->3->5-	Colllege ID exist and designation is candidate	NO	Input user login ID	successful

>6- >7- >9- >10				
4) 1- >2- >3- >5- >6- >7- >2.	If you want to enter more	YES	Input ID	Unsuccessful\successful

- **Test case for “Upload Result”**

Path	Condition	Test cases	Statement	Status
1) 1- >2- >3> 4- >6- >7	If candidate have majority	YES	Declare winner	Successful
2) 1- >2- >3- >4- >5- >3- >4- >6- >7	If candidate have majority	NO	Eliminate candidate with fewer votes and redistribute the next preferences of the people who voted for that candidate	Unsuccessful

FUTURE SCOPE

- This system can be used in election as it provide complete security and accurate result and also save time and expenditure.

- It provide a compact ,stable system of voting with a facility at home .

- It completely rules out the chance of invalid voting so this system helps in fair election and accurate result

REFERENCES

The theoretical details that helped through the course of the project have been taken from

- Software Engineering, a Practitioner‟s Approach- Roger S. Pressman.
- World Wide Web.
- Sample projects
- P. Jalote, An Integrated Approach to Software Engineering