Project Report On

Online Voting System

Submitted in partial fulfillment of the requirements for the award of the Degree of

#### Bachelor of Technology

In

Computer Science & Engineering

***Submitted by:***

### Sourabh Kr. Mahata Saurabh Kr. Dutta

**Roll: 1205382 Roll: 1205385**

**Rituparna Chakraborty Vivek**

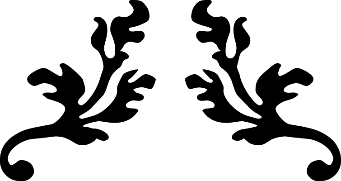
**Roll: 1205383 Roll: 1205217**

***under the guidance of Prof. Arup A. Acharya* Assistant Professor**

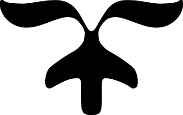
### School Of Computer Engineering

**SCHOOL OF COMPUTER ENGINEERING KIIT UNIVERSITY**

**Bhubaneswar-24 December 2015**



ONLINE VOTNG SYSTEM





**Vote for a change, Vote digitally,**

**Online Voting System**

**FIG. 1 Online Voting System**

**Project Mentor:** Prof. Arup A. Acharya

**Team Members:** Sourabh Kumar Mahata (1205382)

Rituparna Chakraborty (1205383) Saurabh Kumar Dutta (1205385) Vivek (1205217)

**Branch:** Computer Science & Engineering

# ABSTRACT

In this online voting system people can cast their vote through internet. Anyone who wants to have access to the features of this website has to go through two way authentication system. First the user has to input the **username, password and UID number** which when matched with the credential of the user in the database takes the user into the portal.

The portal enables the voter to cast their vote to their choice of candidate. A complete profile of the candidate along with their party details are uploaded in the website so that voters can have a look over their profile and can make proper judgment for their valuable vote. A proper heed is paid to the fact that no voter is allowed to cast their vote more than once. A feedback mechanism is also implied in this system so that voters can provide their feedback regarding the portal as well as regarding the candidate which can be seen only by the Election Commissioner i.e. the admin of this portal of the E voting system.

Keywords: UID, E Voting System, feedback mechanism.

# ACKNOWLEDGEMENT

We take this opportunity to express our deep gratitude to all those helping hands without whom this project would have not been what it is.

We take immense pleasure to express our thankfulness to our mentor **Prof. Arup**

**A. Acharya** for his constant motivation, timely suggestions which made our sail smooth through the odds we faced in our project.

I would like to thank our **Dean Prof. Samresh Mishra** for giving us this opportunity to enhance our skills by giving this project and for motivating us throughout the time and also thank the complete faculty of Computer Science and Engineering who have provided us the knowledge to successfully complete the project

Last but not the least we would like to thank our friends who not only provided their helping hands in our project as and when required but also for becoming the first user of this portal and providing us with all the feedback which helped us improve our portal.

##### Sourabh Kr. Mahata Saurabh Kr. Dutta

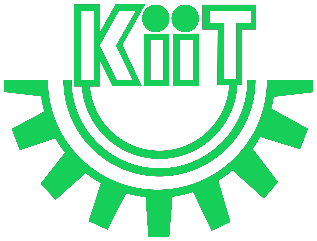
**(1205382) (1205385)**

**Vivek Rituparna Chakraborty**

**(1205217) (1205383)**

**School of Computer Engineering KIIT UNIVERSITY**

**Bhubaneswar-24**



**CERTIFICATE**

***This is to certify that the project entitled “Online Voting System” is being carried out by Rituparna Chakraborty (1205383), Vivek (1205217), Saurabh Kr. Dutta (1205385) and Sourabh Kr. Mahata (1205382) in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering at School of Computer Engineering, KIIT University, Bhubaneswar during the academic year 2010-2011 under my supervision. The matter embodied in this project is original and has not been submitted for the award of any other degree.***

**Signature of the Dean Signature of the Mentor**

**----------------------------- ------------------------------**

**(Dr. Samresh Mishra) (Prof. Arup A. Acharya) School of Computer Engineering Assistant Professor**

**Table of Contents**

##### Introduction

* 1. [FAQ Related to OnlineVoting 8](#_TOC_250031)
  2. Purpose of the Project 10
  3. Scope of the Project 10

1. [Objective 11](#_TOC_250030)
2. [Software Development Lifecycle 12](#_TOC_250029)
3. [Software Requirement Specification](#_TOC_250028)
   1. [Questionnaires 14](#_TOC_250027)
   2. [System Requirements 15](#_TOC_250026)
   3. [Functional Requirements 16](#_TOC_250025)
   4. [Non Functional Requirements 18](#_TOC_250024)
   5. [Features of the Project 19](#_TOC_250023)
4. [Design](#_TOC_250022)
   1. [UML Design](#_TOC_250021)
      1. [Use Case Diagram 20](#_TOC_250020)
      2. [Class Diagram 22](#_TOC_250019)
      3. [Activity Diagram 23](#_TOC_250018)
      4. [Sequence Diagram 25](#_TOC_250017)
   2. [Database Design](#_TOC_250016)
      1. [ER Diagram 30](#_TOC_250015)
      2. [Table Diagram 31](#_TOC_250014)
   3. [Test Case Design](#_TOC_250013)
      1. LOGIN Module 33
      2. [VOTE Module 34](#_TOC_250012)
      3. [FEEDBACK Module. 34](#_TOC_250011)
      4. ADMIN Module 35
   4. [GUI Design](#_TOC_250010)
      1. [LOGIN/SIGN UP Page 36](#_TOC_250009)
      2. ADMI Portal 38
      3. [FEEDBACK Page 39](#_TOC_250008)
      4. [REGISTRATION Page. 40](#_TOC_250007)
5. [Coding](#_TOC_250006)
   1. [Database Connection 41](#_TOC_250005)
   2. [Login process 42](#_TOC_250004)
   3. [Session Destroy 43](#_TOC_250003)
6. [Testing 44](#_TOC_250002)
7. [Future Work 50](#_TOC_250001)
8. [Conclusion 51](#_TOC_250000)
9. References 52

**List of figures:**

FIG.1. Online Voting System 2

FIG.2. Iterative Waterfall Model 12

FIG.3. Use Case diagram representing functions of Admin, Candidate & Voter 21

FIG.4. Class Diagram representing Classes, their Attributes, Methods and Relationship among Objects 22

FIG.5. Activity diagram for Online Voting System 24

FIG.6. Sequence diagram for Login System 25

FIG.7. Sequence diagram for Registration System 26

FIG.8. Sequence diagram for Cast Vote System 27

FIG.9. Sequence diagram for Feedback System 28

FIG.10. Sequence diagram for Update System 29

FIG.11. ER diagram for Online Voting System 30

FIG.12. Login page of Online Voting System 36

FIG 13. HOME page for Online Voting System 37

FIG.14. Admin Portal of Online Voting System 38

FIG.15. Feedback page of Online Voting System 39

FIG.16. Registration page for Candidate. 40

FIG.17. Registration page for Voter 40

FIG.18. Database Connection Code Snippet 41

FIG.19. Login process Coding 42

FIG.20. Session destroy Coding 43

# List of Tables:

TABLE 1. Attributes of ADMIN 31

TABLE 2.Attributes of CANDIDATE 31

TABLE 3. Attributes of VOTER 32

TABLE 4. FEEDBACK module 32

TABLE 5.Test Cases for LOGIN Process 33

TABLE 6.Test Cases for VOTE Module 34

TABLE 7.Test Cases for FEEDBACK Button 34

TABLE 8.Test Cases for ADMIN Portal 35

TABLE 6.Test Cases for VOTE Module. 34

TABLE 7.Test Cases for FEEDBACK Button 34

TABLE 8.Test Cases for ADMIN Portal 35

# Introduction

Online Voting System is an online voting technique in which a person above the age of 18 years and of any sex can vote without being physically present at the polling station. Using these system citizens of India can vote through online without visiting polling booth. A centralized database is maintained by election commission of India where citizens information is maintained when ever citizen is using online voting system his/her information is authenticated with the data present in database if user is not in the list he cannot use online voting system.

Advantages of using this application is time saving and voting percentage can be increased and high security can be implied for preventing false voting.

This application contains two level of user’s administrator level and voter level where each level has different functionality. Election commission of India will update voter’s details to database regularly.

A person can cast his/her vote in two ways:

* + internet voting at traditional polling sites.
  + internet voting for any location.

# FAQ Related to Online Voting

Q1.I know about voting by 'Postal ballot'. What is Online Voting System?

Ans. e-Voting is voting through an electronic system where shareholders can vote on resolutions of companies requiring voting through Postal Ballot as per extant rules and regulations without having the necessity of sending their votes through post.

Q2.How will I benefit from the Online Voting system?

Ans. i. Ease of operation. With User ID and password, you can login to e-Voting system through internet and cast your vote from your residence, office etc. ii.Sufficient time to vote as you can vote till the end of voting cycle. iii.Elimination of postal ballots which leads to getting lost in-transit. iv.Paperless mode of casting vote thus no waste of paper occurs.

Q3.Will proxy be able to cast vote in e-Voting system?

Ans. e-Voting system brings flexibility, convenience and ease of operation for the shareholder to cast vote through internet. Thus, eliminating the need to appoint a proxy.

Q4. Once I cast my vote on e-Voting System, can I modify my vote before the closing of the e-Voting cycle if I have already cast my vote once?

Ans. No. Vote once casted will be considered final and cannot be modified.

Q5. How will the results be made available at the end of the voting period?

Ans. Each time a voter cast his/her vote for his/her preferred candidate, the votes are counted in the backend and saved in the database. When the time period expires then total votes for each candidates are summed up and displayed in the portal.

# Purpose

The purpose of this project is to present a complete automated solution of the manual voting procedure followed in India by using this application. This web application on E Voting presents a very user friendly interface through which the citizens of India can cast their vote to their desired candidate through any GPRS enabled device that they have and that supports the system. The main idea behind this application is to grab the interest of the youth so that their participation can be increased with this interactive means.

# Scope

1. Less effort and less man power required as the entire procedure is automated.
2. Increasing the number of voters as individual find it easier and more convenient. 3.Decrease in forgery as everything is well preserved in the database. 4.Easier to track votersas we can sit in one place and see how many people are voting.
3. Instant Poll result as there is no need to manually calculate the results.
4. Paperless voting process.

# Objective

The goal of the E-Voting or OVS as a product is to automate the voting process, help in solving fraud problems, decreasing the voting time, and the process of counting. The purpose of this software is to increase the overall voting percentage across the country, as in the present system people have to visit the booth to cast their vote and those people who live out of their hometown are not able to cast their vote during the elections.

This system encompasses legal, regulatory, behavioral and sociological aspects of the current voting system, while adding additional convenience and security to the overall voting process.

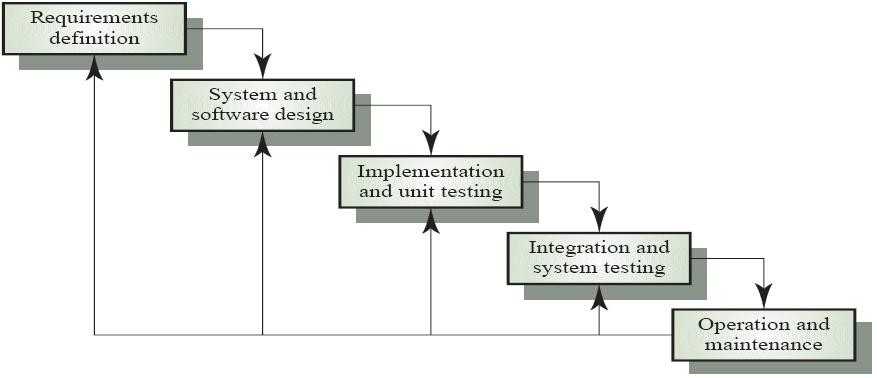
This system is designed to improve the current voting process in the following ways:

* Allow voters to vote from any poll site in the country without the use of absentee ballots.
* Reduce the number of legitimate votes not counted by reducing the number of over-votes and eliminating vote tampering.
* Improve the registration process by allowing voters to check their registration status prior to voting and centralizing registration databases.
* Increase voter confidence and improve the voting experience.

The main objective of our website is online voting and checkout the results after elections were computed. Other objectives are to deliver information about elections and to deliver information about different political parties and their leaders.

# Software Development Lifecycle

The software development life cycle[1] used here is iterative waterfall model. The process starts with an implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added.



**FIG. 2 Iterative Waterfall Model**

### Why did we use Iterative waterfall model?

#### Requirements of the complete system are clearly defined and understood.

* It was useful as the project was big, so it was beneficial to develop each phase of the system efficiently.
* As major requirements were defined, however we could easily identify some more details with time.

# Software Requirement Specification

A software requirements specification (SRS)[1] is a description of a software system to be developed, laying out functional and non- functional requirements, and may include a set of use cases that describe interactions the users will have with the software.

* 1. **Questionnaires**

1. What is the privacy of every user?
2. What info is needed during registration of contestant and voters?
3. What is the hierarchy of the users?
4. What are the features you want to manage while the voting is going on?
5. How is the timer useful to constraint duplicacy?
6. Can the different users edit their profile?
7. Will there be a confirmation after giving a vote?
8. Would you like to track the contestant (work doing to get votes)?
9. Can the voters ask their query?
10. How will the voters be informed about the date and time of voting? 11.For the above do we need any social plugin?
11. Can the contestants interact with the voters?
12. Will the profile details of the contestant be available? 14.How will the results be evaluated after the voting?

15.Who all can be the admin of this e-voting website in the back end? 16.Who all are the users in the front end except the voters?

1. Can they create an account from before and be in touch with the latest updates on the website?
2. How do we prevent duplicacy of votes by the voters?
3. Can we use adhaar number to prevent duplicacy by using adhaar API?
4. Can the voters interact with the candidates if they wish to do so? Can we have a live chat session between them?
5. Do we need a time limit for the voters? How will this time limit concept prevent duplicacy?
6. How much security can be provided to the data in the database?

## System Requirements

Different kinds of tools and software required to run this application:

1. HTML &CSS[4] (for designing the UI of the application)
2. Minimum PHP 5.3[3][5] (Server Side Scripting)
3. Apache Server (to host the system)
4. Web Browsers (like Firefox, Chrome etc.)
5. TCP/IP protocol is used communicate with the server. Client can send request to the server through TCP/IP Protocol and server responds to the client through TCP/IP Protocol.
6. A centralized database is maintained to store all the data which can be queried using MYSQL.

# Functional Requirements

##### Registration

*Description:* A user can register as a voter or contestant

##### Registration as a Voter

*Description:* A voter can register themselves by providing their name, DOB, address, email id and phone number.

*Input:* Personal Details

*Output:* Voter Registration and UID generation.

##### Registration as a Contestant

*Description:* A candidate can register themselves by providing their name, DOB, address, email id, phone number and party name.

*Input:* Personal Details

*Output:* Candidate Registration and UID generation.

##### Login

*Description:* The user of the application i.e. admin, voter and candidate have to login through their respective username, password and UID provided at the time of registration.

##### Update Profile

*Description:* The user can update any of the details of their profile except their DOB and UID after successfully logging into their account.

##### Update Voter Profile

*Description:* The voter can log in into their accounts and then can do necessary changes they want in their profile.

*Input:* username, password and UID

*Output:* Updated Voter profile

##### Update Candidate Profile

*Description:* The Candidate can log in into their accounts and then can do necessary changes they want in their profile.*Input:* username, password and UID *Output:* Updated Candidate profile.

##### Cast Vote

*Description:* The voter after successful log in can cast their vote to their preferred candidate.

*Input:*vote

*Output:* vote submitted

##### Feedback

*Description:* The voter can submit their feedback regarding the improvements they want in the existing system and also can report any candidate to ELECTION COMMISION OF INDIA

*Input:*feedback

*Output:* feedback submit successful.

##### R.3Organizing

*Description:* The admin (Election Commissioner) allots a time and day for the election.

*Input:* time and day

*Output:* time and day for each voter is set

# Non Functional Requirements

##### Performance Requirement:

Performance of the system depends on the response time, speed of the vote submission and the load it can withstand. This version of the system has limited vote submission speed hence it can work in small network efficiently, but can be extended to large network.

##### Safety Requirement:

Application needs to have a check on:

1. No other server is running on same port in which Apache-MySQL is running otherwise it will stop Apache’s functionality.
2. If Web Forms with the services processing form input are consistent.

##### Security Requirements:

Security is maintained by generating a unique ID for the user which he can use while login along with his password.In case of system crash the data in the database should be back up so that no data is lost.

##### Quality Requirements:

The application should have simple and user friendly GUI so that the end users find it easy to understand the functionality of the application and can carry out their task.

* 1. **Features of the Project**

1. Secure Login
2. Only authenticated user can vote
3. A voter can vote only once
4. Instant poll result
5. Feedback System
6. Contestant details and parties
7. User can update their details

# DESIGN

Software design is the process of implementing software solutions to one or more set of problems. One of the important parts of software design is the software requirements analysis (SRA)[1]. It is a part of the software development process that lists specifications used in software engineering. We have thus described different forms of designs here.

# UML DESIGN

UML stands for Unified Modeling Language [1][6] which is used in object oriented software engineering. Although typically used in software engineering it is a rich language that can be used to model an application structures, behavior and even business processes. The different UML diagrams are as follows:

# USE CASE DIAGRAM:

A use case diagram [1][6] at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system.

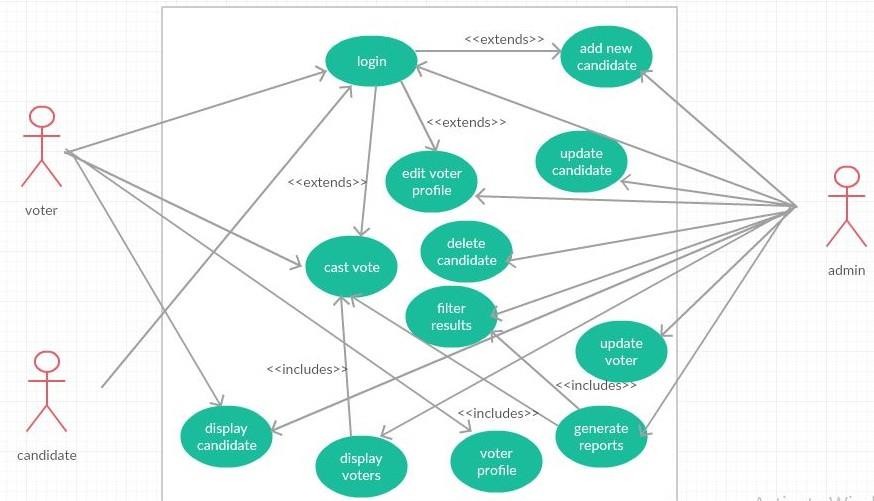
Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

Now when the initial task is complete use case diagrams are modelled to present the outside view.

So in brief, the purposes of use case diagrams can be as follows:

* + - * Used to gather requirements of a system.
      * Used to get an outside view of a system.
      * Identify external and internal factors influencing the system.
      * Show the interacting among the requirements are actors.

##### Use case of different actors using the system:



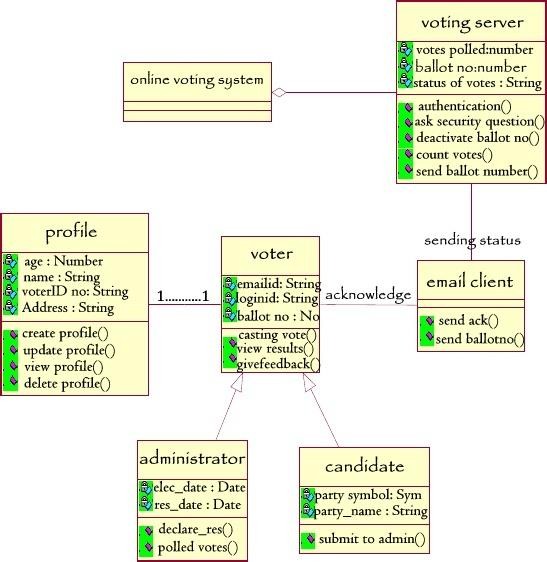
**FIG. 3 USE CASE DIAGRAM FOR ONLINE VOTING SYSTEM**

The diagram shows the interaction of the users (admin, candidate and voter) with the different features of the system such as “generate report” that can be accessed by the admin, “update voter” can be done by the admin, “display candidate” can be viewed by the admin, candidate and the voter etc.

# CLASS DIAGRAM

A class diagram[1][6] in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

* The top part contains the name of the class.
* The middle part contains the attributes of the class. They are left aligned and the first letter is lower case.
* The bottom part gives the methods or operations the class can take or undertake. They are also left aligned and the first letter is lower case.



**FIG. 4 CLASS DIAGRAM FOR ONLINE VOTING SYSTEM**

The class diagram shows how different classes of the system interact and exchange message in order to successfully implement the system.

## ACTIVITY DIAGRAM

Activity diagrams[1][6] are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

Activity diagrams show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity. They may be used to detail situations where parallel processing may occur in the execution of some activities. Activity diagrams are useful for business modelling where they are used for detailing the processes involved in business activities.

##### The following sections describe the elements that constitute an activity diagram.

**Actions:** An action represents a single step within an activity. Actions are denoted by round-cornered rectangles.

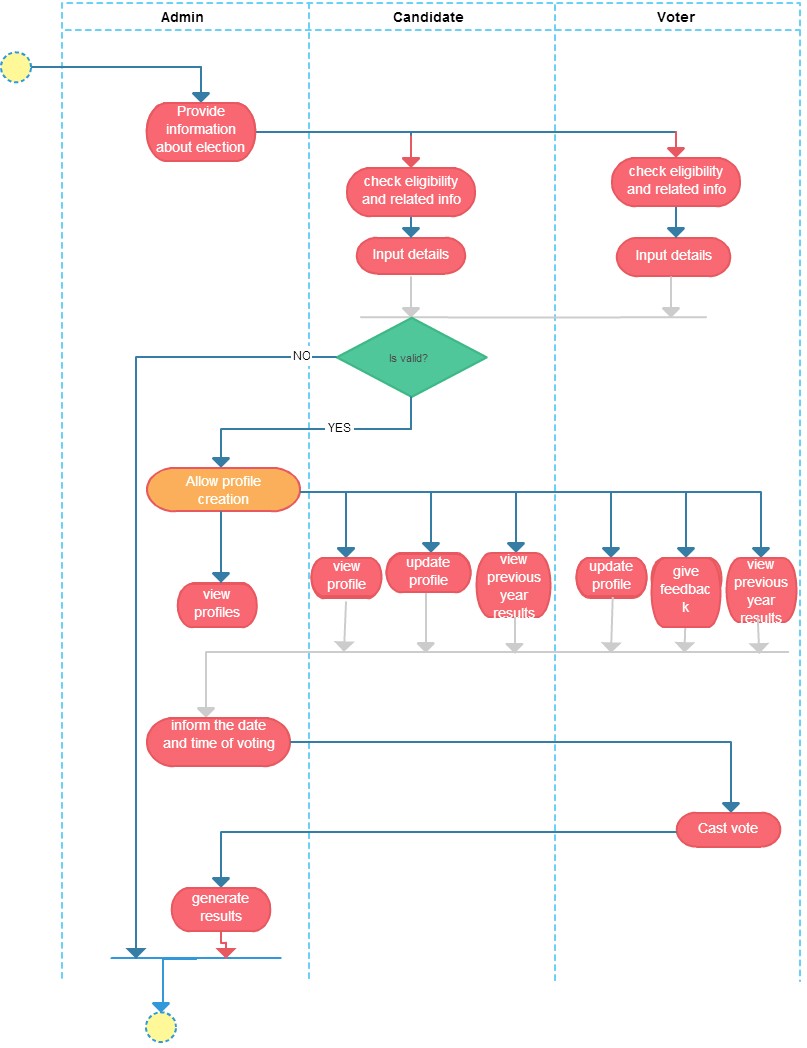
**Control Flow:** A control flow shows the flow of control from one action to the next. Its notation is a line with an arrowhead.

**Initial Node**: An initial or start node is depicted by a large black spot, as shown below.

**Final Node:** There are two types of final node: activity and flow final nodes. The activity final node is depicted as a circle with a dot inside.

**Decision and Merge Nodes:** Decision nodes and merge nodes have the same notation: a diamond shape. They can both be named. The control flows coming away from a decision node will have guard conditions which will allow control to flow if the guard condition is met.

**Fork and Join Nodes**: Forks and joins have the same notation: either a horizontal or vertical bar. They indicate the start and end of concurrent threads of control.



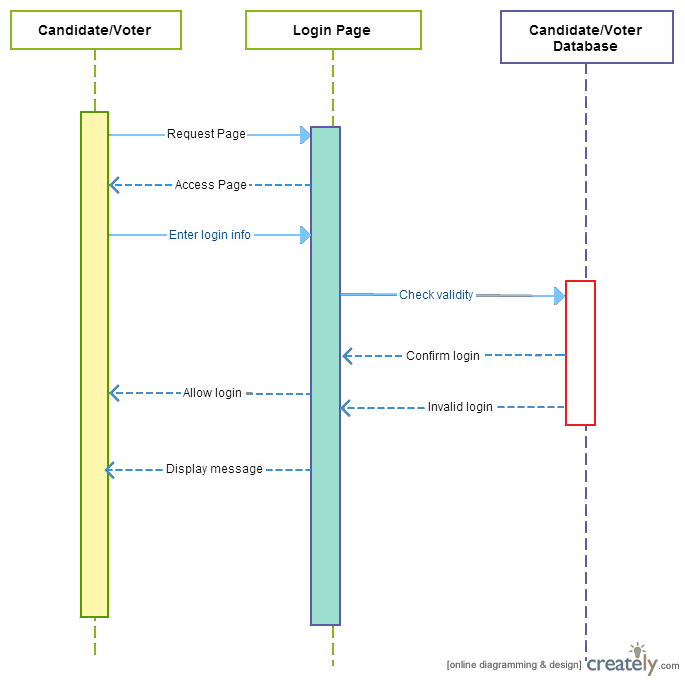
**FIG.5 ACTIVITY DIAGRAM FOR ONLINE VOTING SYSTEM**

The activity diagram shows how the activities flow in order to successfully execute the system. This shows the step by step activity required to be carried out by the users i.e. the admin, candidate and the voter.

# SEQUENCE DIAGRAM:

A Sequence diagram [1][6] shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

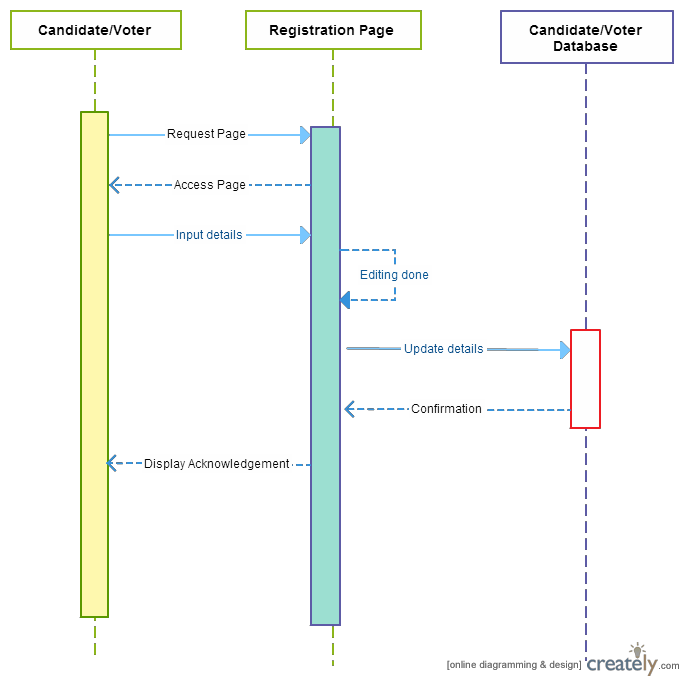
## LOGIN MODULE:



**FIG.6 SEQUENCE DIAGRAM FOR LOGIN SYSTEM**

This sequence diagram carries out the functionality of the Login Page and shows how the candidate/voter uses the login page and how the candidate/voter interact with the database for a successful or unsuccessful login.

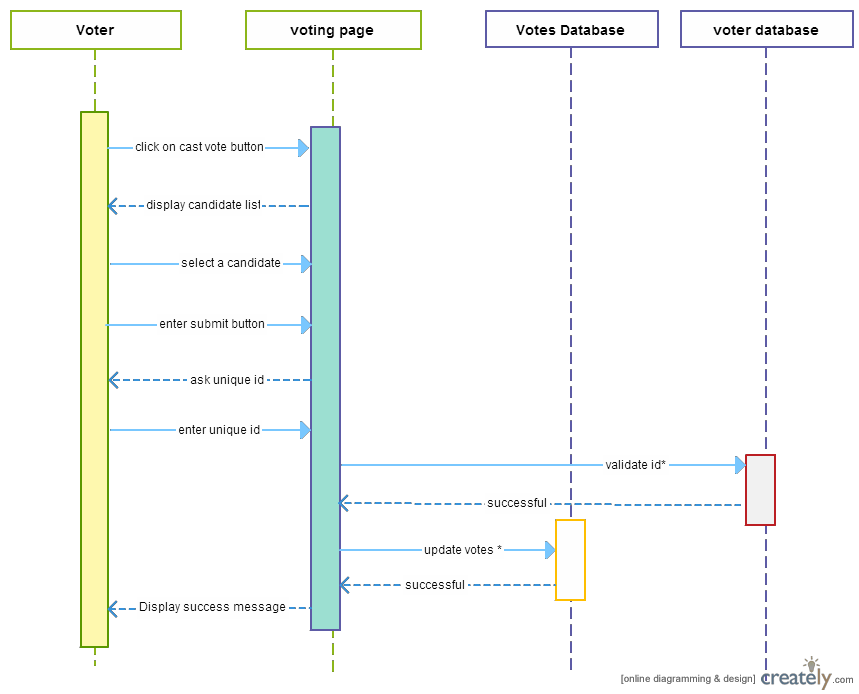
## REGISTRATION SYSTEM MODULE:



**FIG.7 SEQUENCE DIAGRAM FOR REGISTRATION SYSTEM**

This sequence diagram carries out the functionality of the Registration System and shows how the candidate/ voter uses the registration page and how the candidate/ voter interact with database to register themselves in the portal.

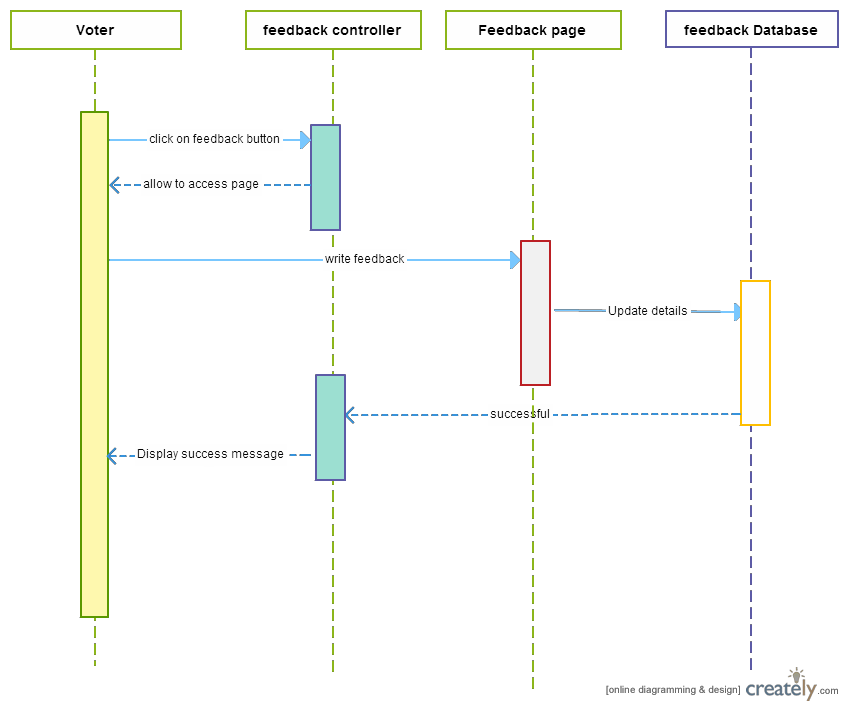
## CAST VOTE MODULE:



**FIG.8 SEQUENCE DIAGRAM FOR CAST VOTE MODULE**

This sequence diagram reflects the functionality of the casting votes and shows how the voter, the voting system, vote database and voter database exchange messages so that a voter can vote to the required candidate.

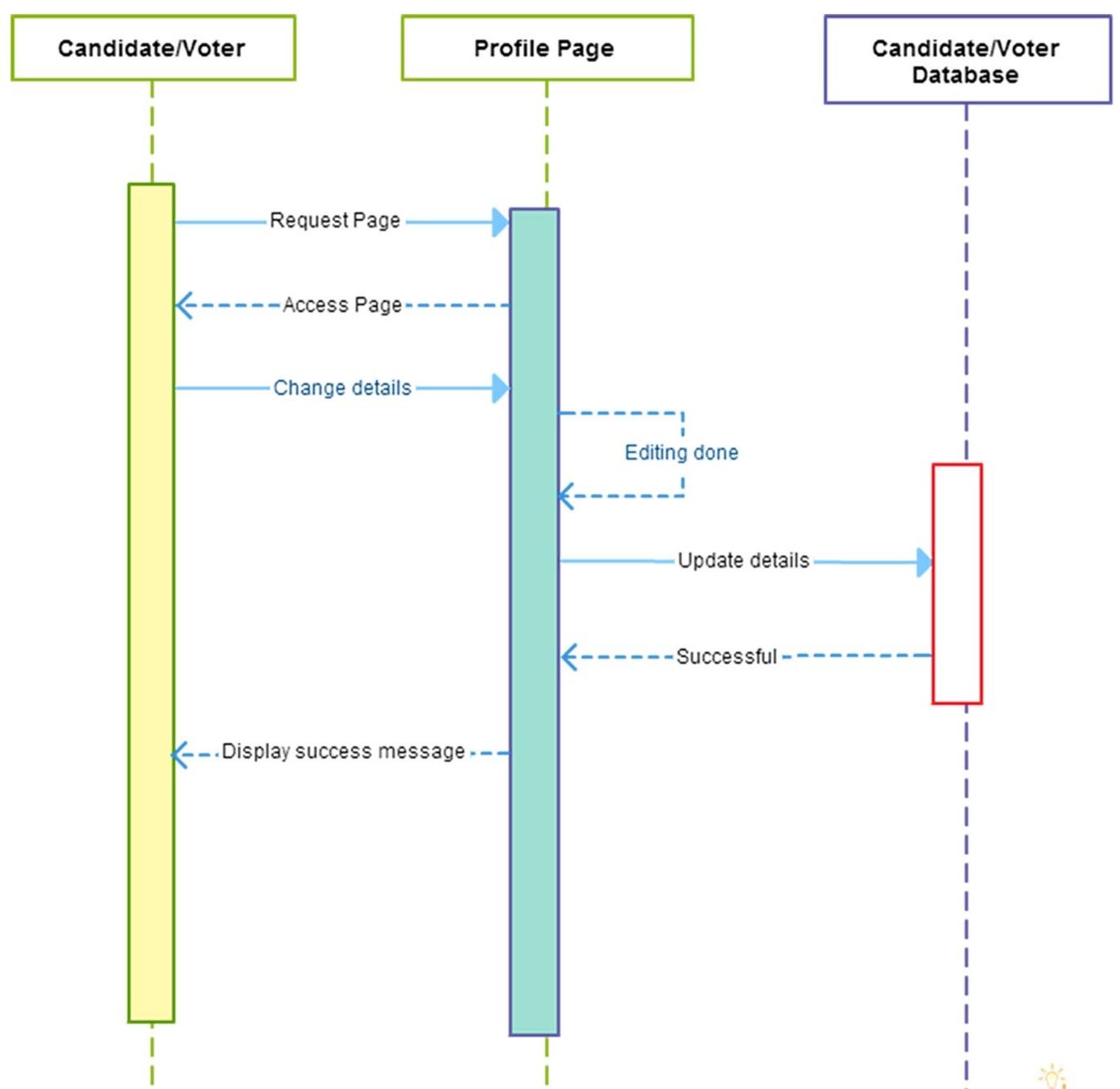
## FEEDBACK MODULE:



**FIG.9 SEQUENCE DIAGRAM FOR FEEDBACK MODULE**

This sequence diagram carries out the functionality of the Feedback System and shows how the vote, the feedback controller, feedback system and feedback database exchange messages in order to successfully give a feedback.

## UPDATE SYSTEM:



**FIG.10 UPDATE MODULE FOR ONLINE VOTING SYSTEM**

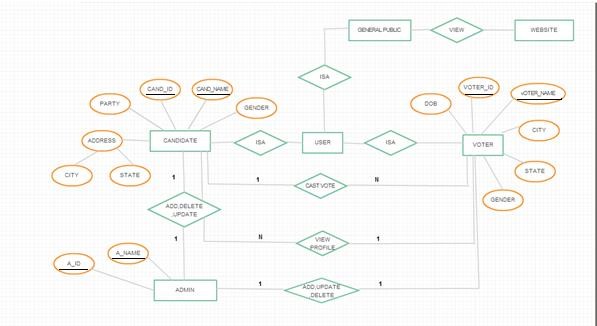
This sequence diagram carries out the functionality of the Update System and shows how the candidate/voter, the profile page and the candidate/voter database exchange messages in order to update details of their profile

# DATABASE DESIGN

Database design[2] is the design of how persistently stored so that it can be accessed by different programs and users over a period of time.

## ER DIAGRAM

In software engineering, an entity–relationship model (ER model)[2] is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them. It has been reduced till second normal form (2NF) as it in 1NF and has no non-prime attribute which is dependent on any proper subset of any candidate key.



**FIG.11 ER DIAGRAM FOR ONLINE VOTING SYSTEM**

This ER Diagram shows how different entities like admin, candidate and voter have dependency upon each other like one-to-many, many-to-many etc.

# TABLE DIAGRAM

The table diagram allows you to visualize a database to which you are connected. It shows all the attributes of a database along with its type and constraints.

## DATABASE TABLE:

The different database tables for the Online Voting System are as follows:

## ATTRIBUTES OF ADMIN:

**TABLE 1 ATTRIBUTES OF ADMIN**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **CONSTRAINT** |
| Username | varchar(10) | Primary key,Not Null |
| Password | varchar(32) | Not Null |
| first name | varchar(10) | Not Null |
| last name | varchar(10) | Not Null |

The table diagram shows the attributes of the Admin of the system

# ATTRIBUTES OF CANDIDATE:

**TABLE 2 ATTRIBUTES OF CANDIDATE**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **CONSTRAINT** |
| user\_id | varchar(32) | Primary Key,Not Null |
| Name | varchar(32) | Not Null |
| City | varchar(32) | Not Null |
| Dob | Date | Not Null |
| party | varchar(32) | Not Null |
| Votes | int(11) |  |
| password | varchar(32) | Not Null |

The table diagram shows the attributes of the candidate of the system.

# ATTRIBUTES OF VOTER:

**TABLE 3 ATTRIBUTES OF VOTER**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **CONSTRAINT** |
| user\_id | varchar(32) | Unique,Not Null |
| name | varchar(32) | Not Null |
| City | varchar(32) | Not Null |
| Dob | date | Not Null |
| password | varchar(32) | Not Null |

The table diagram shows the attributes of the voter of the system.

## FEEDBACK MODULE:

**TABLE 4 FEEDBACK MODULE FOR ONLINE VOTING SYSTEM**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **CONSTRAINT** |
| s.no | int(11) | Unique, Not Null |
| Uid | int(11) | Unique,Not Null |
| firstname | varchar(32) | Not Null |
| lastname | varchar(32) | Not Null |
| email | varchar(32) | Not Null |
| Date | date | Not Null |
| comment | varchar(1024) | Not Null |

The table diagram shows the attributes of the feedback mechanism of the system.

# TEST CASE DESIGN

A test case[1], in software engineering, is a set of conditions under which

a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do.

## LOGIN:

**TABLE 5 TEST CASES FOR LOGIN MODULE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl no** | **Test case name** | **Test Procedur e** | **Pre- condition** | **Expected Result** | **Reference** |
| **1.** | all three text  box empty | No  credential | Login page  is loaded | Invalid credentials  message popup | Login |
| **2.** | Username ok but password &  UID field empty | Only username | Login page is loaded | Empty password and UID field message popup | Login |
| **3.** | Password ok but username and UID  field empty | Only password | Login page is loaded | Empty username and UID field message popup | Login |
| **4.** | UID ok but username & password field empty | Only UID | Login page is loaded | Empty username and password  field message popup | Login |
| **5.** | Username & password ok but UID field empty | Username & password | Login page is loaded | Empty UID  field message popup | Login |
| **6.** | Password and UID ok but  username field empty | Password & UID | Login page is loaded | Empty username field message popup | Login |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **7.** | Username & UID ok but password  field empty | Username & UID | Login page is loaded | Empty password field | Login |

## VOTE MODULE:

**TABLE 6 TEST CASES FOR VOTE BUTTON**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl no.** | **Test case name** | **Test Procedure** | **Pre- condition** | **Expected Result** | **Reference** |
|  |  |  |  |  |  |
| **1.** | Cast his/her  vote | Call voted page with casting | Login page is loaded | Voted Successfully | Voting |
| **2.** | Not casted his/her  vote | Call voted page with no casting | Login page is loaded | Unsuccessful | Voting |

## FEEDBACK MODULE:

**TABLE 7 TEST CASES FOR FEEDBACK BUTTON**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl no.** | **Test case name** | **Test Procedure** | **Pre- condition** | **Expected Result** | **Reference** |
|  |  |  |  |  |  |
| **1.** | Given feedback | Call feedback page with feedback  given | Voted page is loaded | Given feedback successfully | Feedback |
| **2.** | No feedback | Call feedback page with no  feedback | Voted page is loaded | Not given feedback | Feedback |

# ADMIN PORTAL:

**TABLE 8 TEST CASE FOR ADMIN PORTAL**

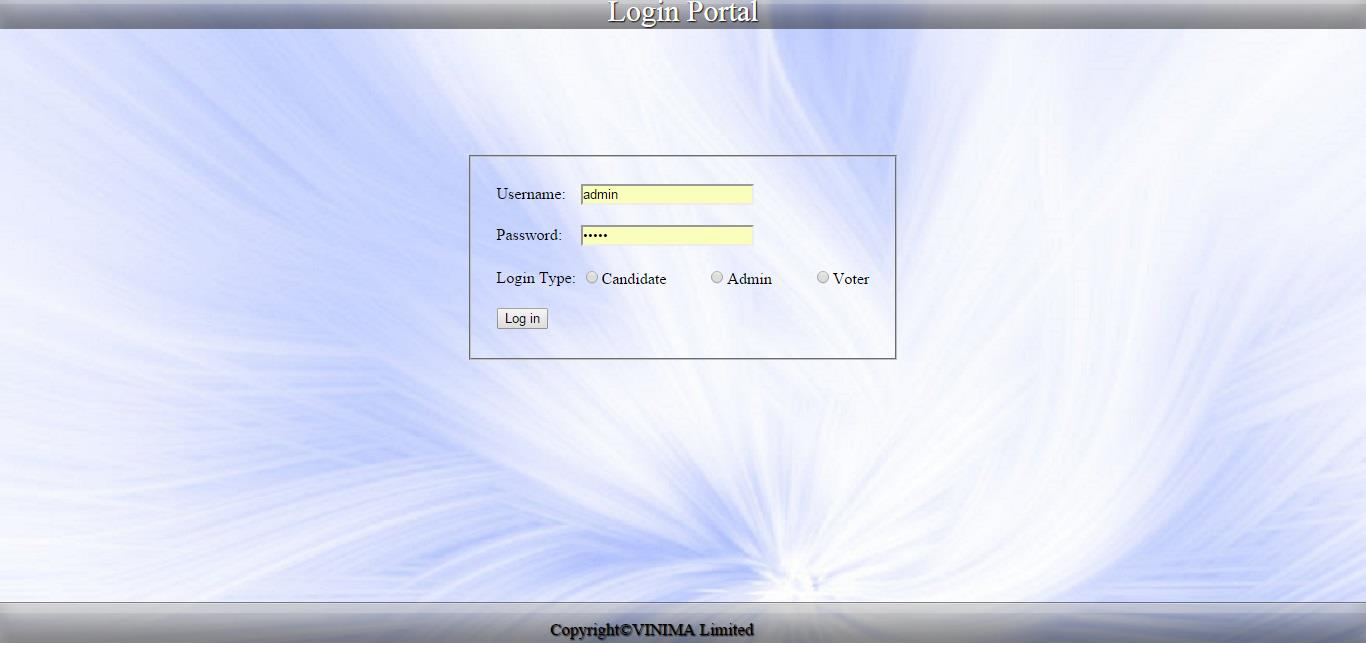
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl no**  **.** | **Test case name** | **Test Procedure** | **Pre- condition** | **Expected Result** | **Reference** |
|  |  |  |  |  |  |
| **1.** | Schedule | Call scheduling  page | Login page is  loaded | Scheduled successfully | Scheduling |
| **2.** | View voter’s details | Call details page for the  voter | Scheduled page is loaded | Viewed successfully | Viewing voter’s details |
| **3.** | View  candidate’s details | Call details page for the  candidate | Scheduled page is loaded | Viewed successfully | Viewing candidate’s details |
| **4.** | View live results | Call live results Page | Voter’s/Candidat e’s details page  is loaded | Viewed successfully | Viewing live results |

# 5.4. GUI DESIGN

A graphical user interface or GUI, is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation.

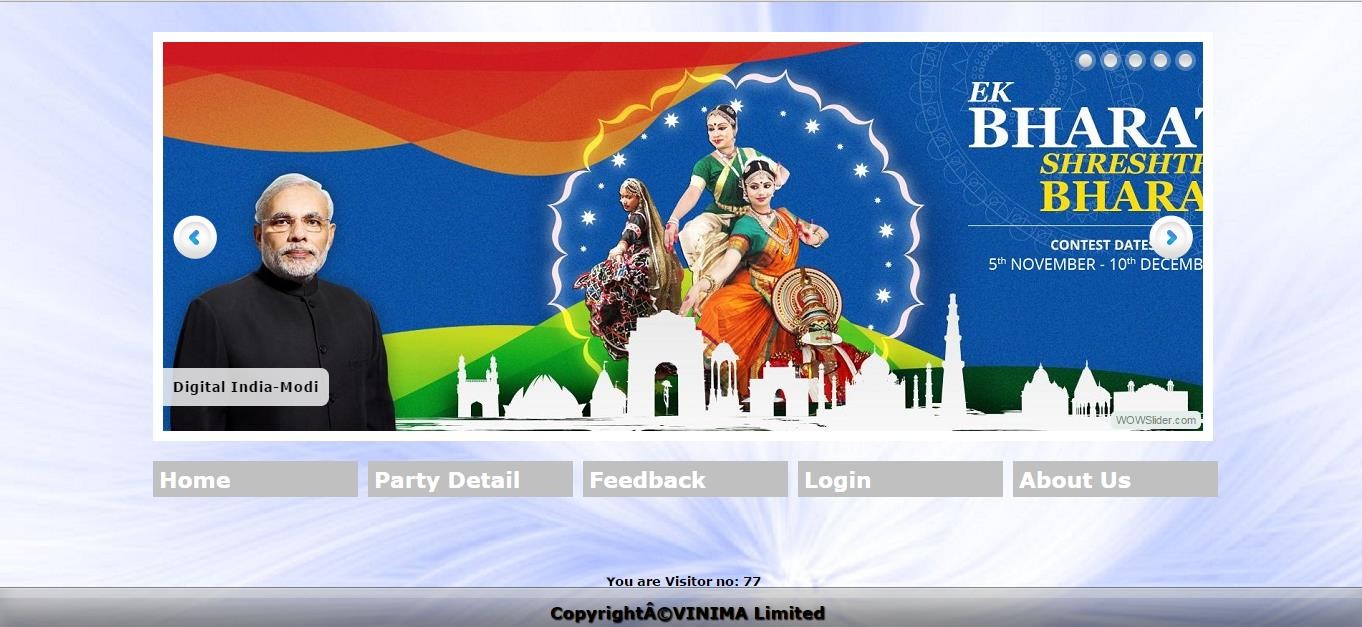
## 5.4.1 LOGIN/SIGN UP PAGE:

This is the first page of the application where the registered user can login into the application, if not registered then they can sign up and get themselves registered to use this portal. After they have successfully signed up, each user will be provided with a UID (unique identification) number. With the login details, each user have to provide the UID at the time of login. If all the credentials matches with the credential saved in the database, the user is directed to the home page of the portal and based on the type of user (i.e. admin, candidate and voter) they can use the features of the portal.



**FIG.12 LOGIN PAGE FOR ONLINE VOTING SYSTEM**

# HOME PAGE



**FIG.13 HOME PAGE FOR ONLINE VOTING SYSTEM**

# ADMIN PORTAL:

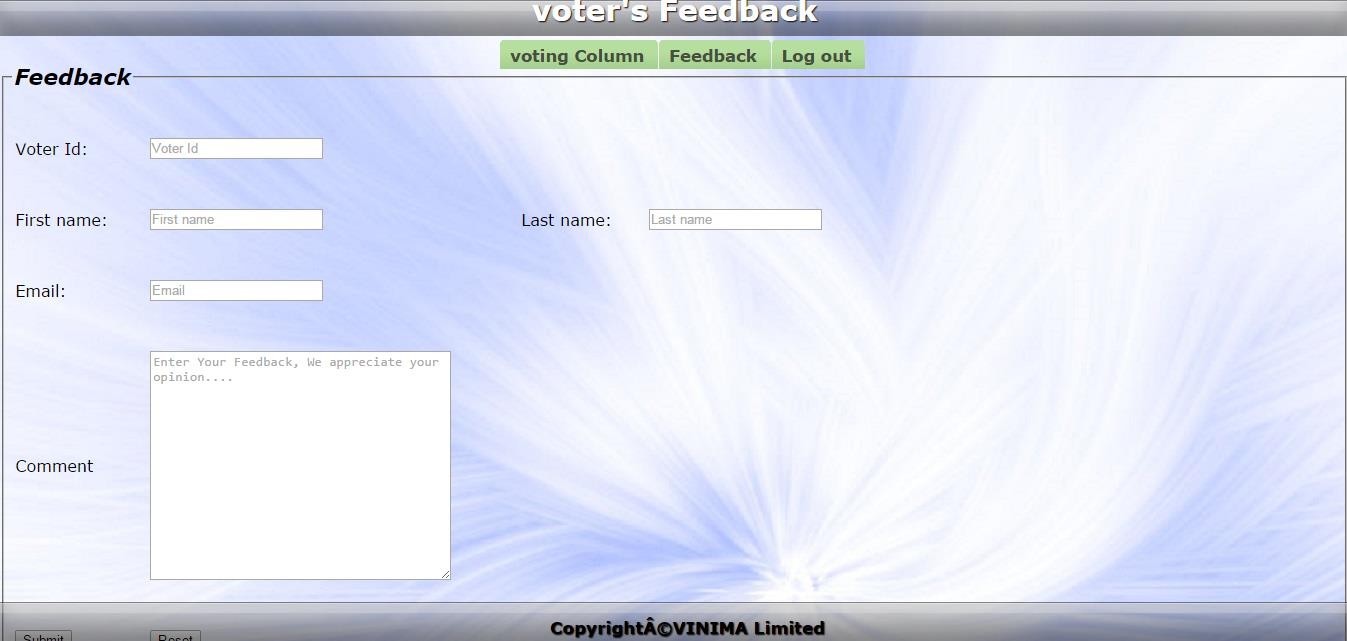
Admin portal is the portal where the admin can create the candidate so that their details along with their party manifesto is available to the voters at the time of polling. This page provides the admin with special functionality such as editing candidate details, deleting a candidate if requested from the Election Commission of India, scheduling election date and time within which a voter should cast his vote, after the date and time expires voter is not eligible for voting.



**FIG.14 ADMIN PORTAL**

# FEEDBACK PAGE:

When a user selects the feedback option then he/she is directed to this page where he/she can submit his/her views regarding the portal for further enhancement as well as they can report any candidate to the election commissioner.



**FIG.15 FEEBACK PAGE**

# REGISTRATION PAGE:

When the user visits the portal for the first time then he/she is required to register themselves into the portal so that their profile is created and then he/she can take full leverage of the features of the portal.



**FIG. 16 Registration Page for Candidate**



**FIG. 17 Registration Page for Voter**

# CODING

Coding[1] may refer to computer programming, the process of designing, writing, testing, debugging / troubleshooting, and maintaining the source code of computer programs.

## DATABASE CONNECTION:

**<!-- Document modified on 20-Nov-2015 -->**

**<!-- Author Sourabh, Saurabh, Vivek, Rituparna -->**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"** [**"http://www.w3.org/TR/xhtml/DTD/xhtml1-transitional.dtd"**](http://www.w3.org/TR/xhtml/DTD/xhtml1-transitional.dtd)**>**

**<html xnlns="**[**http://www.w3.org/1999/xhtml"**](http://www.w3.org/1999/xhtml)**>**

**<html>**

**<head>**

**<meta http-equiv="Connect-Type" connect="text/html; charset=utf-8" />**

**<title>Connect</title>**

**</head>**

**<body>**

**<?php**

**$host = "localhost";**

**$username = "root";**

**$password = "";**

**$db\_name = "evoting";**

**mysql\_connect($host, $username, $password) or die("Cannot connect to server"); mysql\_select\_db($db\_name) or die("Cannot connect to database");**

**?>**

**</body>**

**</html>**

**FIG.18 DATABASE CONNECTION**

## LOGIN PROCESS:

**<!-- Document modified on 20-Nov-2015 -->**

**<!-- Author Sourabh,Saurabh,Vivek,Rituparna -->**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"** [**"htt**](http://www.w3.org/TR/xhtml/DTD/xhtml1-transitional.dtd)**p**[**://www.w3.org/TR/xhtml/DTD/xhtml1-transitional.dtd**](http://www.w3.org/TR/xhtml/DTD/xhtml1-transitional.dtd)**">**

**<html xnlns="**[**http://www.w3.org/1999/xhtml**](http://www.w3.org/1999/xhtml)**">**

**<html>**

**<head>**

**<meta http-equiv="Connect-Type" connect="text/html; charset=utf-8" />**

**<title>LOGIN</title>**

**</head>**

**<body>**

**<?php**

**include 'index.php'; include 'connect.php'; ob\_start(); session\_start();**

**$current\_file = $\_SERVER['SCRIPT\_NAME'];**

**$http\_referer = $\_SERVER['HTTP\_REFERER']; if(isset($\_POST['username']) && isset($\_POST['password'])){**

**$username = $\_POST['username'];**

**$password = $\_POST['password'];**

**$password\_hash = md5($password);**

**$log = $\_POST['radio'];**

**if(!empty($username) && !empty($password) && $log == 'admin')**

**{**

**$query = mysql\_query("SELECT `id` FROM `admin` WHERE `username`='$username' AND `password`='$password\_hash'");**

**$query\_num\_rows = mysql\_num\_rows($query); if($query\_num\_rows == 0){**

**echo 'Invalid Username or Password Combinations. ';**

**}**

**else if($query\_num\_rows == 1){**

**$user\_id = mysql\_result($query, 0, 'id');**

**$\_SESSION['user\_id']=$user\_id; header('Location:login1.php');**

**}**

**else{**

**echo 'You must supply username and password';**

**}**

**}**

**?>**

**</body>**

**</html>**

**FIG.19 LOGIN PROCESS**

## SESSION DESTROY

**<!-- Document modified on 20-Nov-2015 -->**

**<!-- Author Sourabh,Saurabh,Vivek,Rituparna -->**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"** [**"http://www.w3.org/TR/xhtml/DTD/xhtml1-transitional.dtd"**](http://www.w3.org/TR/xhtml/DTD/xhtml1-transitional.dtd)**>**

**<html xnlns="**[**http://www.w3.org/1999/xhtml"**](http://www.w3.org/1999/xhtml)**>**

**<html>**

**<head>**

**<meta http-equiv="Connect-Type" connect="text/html; charset=utf-8" />**

**<title>Session Destroy</title>**

**</head>**

**<body>**

**<?php**

**include 'connect.php'; ob\_start(); session\_start();**

**$current\_file = $\_SERVER['SCRIPT\_NAME'];**

**$http\_referer = $\_SERVER['HTTP\_REFERER'];**

**//echo $http\_referer; session\_destroy(); header('Location:login.php');**

**?>**

**</body>**

**</html>**

**FIG.20 SESSION DESTROY**

# TESTING

Testing [1] is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

We have done two phases of testing, in the first phase some of the functionalities were not working correctly but in second phase all the functionalities were successfully implemented.

## PHASE-1 LOGIN:

**TABLE 9 TEST CASES FOR LOGIN MODULE FOR PHASE-1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no** | **Test case name** | **Test Procedure** | **Pre- conditio**  **n** | **Expected Result** | **Refer ence** | **Actual Result** | **Status** |
| **1.** | all three text box empty | No credential | Login page is loaded | Invalid credentials message  popup | Login | Invalid credentials message  popup | Successful |
| **2.** | Username ok but password & UID  field empty | Only username | Login page is loaded | Empty password and UID field message  popup | Login | Empty password and UID field message  popup | Unsuccess ful |
| **3.** | Password ok but username and UID field empty | Only password | Login page is loaded | Empty username and UID field message  popup | Login | Empty username and UID field message  popup | Successful |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **4.** | UID ok but | Only UID | Login | Empty | Login | Empty | Successful |
|  | username |  | page is | username |  | username |  |
|  | & |  | loaded | and |  | and |  |
|  | password |  |  | password |  | password |  |
|  | field empty |  |  | field |  | field |  |
|  |  |  |  | message |  | message |  |
|  |  |  |  | popup |  | popup |  |
| **5.** | Username | Username | Login | Empty UID | Login | Empty UID | Unsuccess |
|  | & | & | page is | field |  | field | ful |
|  | password | password | loaded | message |  | message |  |
|  | ok but UID |  |  | popup |  | popup |  |
|  | field empty |  |  |  |  |  |  |
| **6.** | Password | Password | Login | Empty | Login | Empty | Successful |
|  | and UID | & UID | page is | username |  | username |  |
|  | ok but |  | loaded | field |  | field |  |
|  | username |  |  | message |  | message |  |
|  | field empty |  |  | popup |  | popup |  |
| **7.** | Username | Username | Login | Empty | Login | Empty | Successful |
|  | & UID ok | & UID | page is | password |  | password |  |
|  | but |  | loaded | field |  | field |  |
|  | password |  |  |  |  |  |  |
|  | field empty |  |  |  |  |  |  |

## VOTE MODULE:

**TABLE 10 TEST CASES FOR VOTE BUTTON FOR PHASE-1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no.** | **Test case name** | **Test Procedure** | **Pre- condition** | **Expected Result** | **Reference** | **Actual Result** | **Status** |
|  |  |  |  |  |  |  |  |
| **1.** | Cast his/her vote | Call voted page with  casting | Login page is loaded | Voted Successfully | Voting | Voted Successfully | Successful |
| **2.** | Not casted his/her vote | Call voted page with no  casting | Login page is loaded | Unsuccessful | Voting | Unsuccessful | Unsuccessful |

## FEEDBACK MODULE:

**TABLE 11 TEST CASES FOR FEEDBACK BUTTON FOR PHASE-1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no**  **.** | **Test case name** | **Test Procedure** | **Pre- conditio n** | **Expected Result** | **Refere nce** | **Actual Result** | **Status** |
|  |  |  |  |  |  |  |  |
| **1.** | Given feedback | Call feedback page with feedback  given | Voted page is loaded | Given feedback successfully | Feedback | Given feedback successfully | Successful |
| **2.** | No feedback | Call feedback page with no  feedback | Voted page is loaded | Not given feedback | Feedback | Not given feedback | Unsuccessf ul |

# ADMIN PORTAL:

**TABLE 12 TEST CASES FOR ADMIN PORTAL FOR PHASE-1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no**  **.** | **Test case name** | **Test Procedure** | **Pre- condition** | **Expected Result** | **Referen ce** | **Actual Result** | **Status** |
|  |  |  |  |  |  |  |  |
| **1.** | Schedule | Call scheduling  page | Login page is  loaded | Scheduled  successfully | Scheduling | Scheduled  successfully | Successf  ul |
| **2.** | View voter’s details | Call details page for the  voter | Scheduled page is loaded | Viewed successfully | Viewing voter’s  details | Viewed successfully | Successf ul |
| **3.** | View  candidate’s  details | Call details page for the  candidate | Scheduled page is loaded | Viewed successfully | Viewing  candidate’s  details | Viewed successfully | Unsucce ssful |
| **4.** | View live results | Call live results Page | Voter’s/Candidat  e’s details page is loaded | Viewed successfully | Viewing live results | Viewed successfully | Successf ul |

## PHASE-2: LOGIN:

**TABLE 13 TEST CASES FOR LOGIN MODULE FOR PHASE-2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no** | **Test case name** | **Test Procedure** | **Pre- conditio**  **n** | **Expected Result** | **Refer ence** | **Actual Result** | **Status** |
| **1.** | all three text box empty | No credential | Login page is loaded | Invalid credentials message  popup | Login | Invalid credentials message  popup | Successful |
| **2.** | Username ok but password & UID  field empty | Only username | Login page is loaded | Empty password and UID field message  popup | Login | Empty password and UID field message  popup | Successful |
| **3.** | Password ok but username and UID field empty | Only password | Login page is loaded | Empty username and UID field message  popup | Login | Empty username and UID field message  popup | Successful |
| **4.** | UID ok but username & password field empty | Only UID | Login page is loaded | Empty username and password field  message popup | Login | Empty username and password field  message popup | Successful |
| **5.** | Username & password ok but UID field empty | Username & password | Login page is loaded | Empty UID field message popup | Login | Empty UID field message popup | Successful |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **6.** | Password and UID ok but username  field empty | Password & UID | Login page is loaded | Empty username field message  popup | Login | Empty username field message  popup | Successful |
| **7.** | Username | Username | Login | Empty | Login | Empty | Successful |
|  | & UID ok | & UID | page is | password |  | password |  |
|  | but |  | loaded | field |  | field |  |
|  | password |  |  |  |  |  |  |
|  | field empty |  |  |  |  |  |  |

## VOTE MODULE:

**TABLE 14 TEST CASES FOR VOTE BUTTON FOR PHASE-2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no.** | **Test case name** | **Test Procedure** | **Pre- condition** | **Expected Result** | **Reference** | **Actual Result** | **Status** |
|  |  |  |  |  |  |  |  |
| **1.** | Cast his/her vote | Call voted page with  casting | Login page is loaded | Voted Successfully | Voting | Voted Successfully | Successful |
| **2.** | Not casted his/her vote | Call voted page with no  casting | Login page is loaded | Unsuccessful | Voting | Unsuccessful | Successful |

## FEEDBACK MODULE:

**TABLE 15 TEST CASES FOR FEEDBACK BUTTON FOR PHASE-2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no**  **.** | **Test case name** | **Test Procedure** | **Pre- conditio n** | **Expected Result** | **Refere nce** | **Actual Result** | **Status** |
|  |  |  |  |  |  |  |  |
| **1.** | Given feedback | Call feedback page with feedback  given | Voted page is loaded | Given feedback successfully | Feedback | Given feedback successfully | Successful |
| **2.** | No feedback | Call feedback page with no  feedback | Voted page is loaded | Not given feedback | Feedback | Not given feedback | Successful |

# ADMIN PORTAL:

**TABLE 16 TEST CASES FOR ADMIN PORTAL FOR PHASE-2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no**  **.** | **Test case name** | **Test Procedure** | **Pre- condition** | **Expected Result** | **Referen ce** | **Actual Result** | **Status** |
|  |  |  |  |  |  |  |  |
| **1.** | Schedule | Call scheduling  page | Login page is  loaded | Scheduled  successfully | Scheduling | Scheduled  successfully | Successf  ul |
| **2.** | View voter’s details | Call details page for the  voter | Scheduled page is loaded | Viewed successfully | Viewing voter’s  details | Viewed successfully | Successf ul |
| **3.** | View  candidate’s  details | Call details page for the  candidate | Scheduled page is loaded | Viewed successfully | Viewing  candidate’s  details | Viewed successfully | Successf ul |
| **4.** | View live results | Call live results Page | Voter’s/Candidat  e’s details page is loaded | Viewed successfully | Viewing live results | Viewed successfully | Successf ul |

# FUTURE WORK:

* This web application on E Voting so far allows only one way authentication of the user i.e. through their username, password and unique identification number (UID). Right now only basic functionalities has been provided i.e. user registration, updating user profile, session check in order to avoid duplication of votes, candidate profile with party manifesto, previous election results.
* As an extension of this application we plan to incorporate a two way authentication system of the user. First through a usual way of authentication i.e. username, password and UID.
* Second with the help of real time face detection which will act as an intermediate layer between the first authentication system and home page so that even if a person with someone else username, password and UID tries to login would not be able to do so. This would minimize the forgery that happens during our election to some extent.
* Other feature that we like to add subsequently is providing the database with an extra security so that no tampering with the votes can be done in case someone tries to hack into this application.

# CONCLUSION:

This Online Voting application allows anyone who is above the age of 18 can cast their vote and leverage their voting rights. This application incorporates all the features of the voting system in India. It includes various features such as registration of user (voters/candidate), updating profile of the user (voter/candidate), a feedback mechanism which will enable voters to suggest any modification in the existing application as well as report any complaint against any candidate to the ELECTION COMMISION OF INDIA, a duplication avoidance mechanism which will restrict an user to cast his vote more than once.

The user is authenticated with the details he/she provided during the time of registration and the UID which is being assigned to him/her at that time. After a user is authenticated he/she can cast his/her vote to his/her preferred candidate at the time of election and within a small span of time after the voting time assigned by the admin (election commissioner) gets over the results are displayed.

The main aim of this application is to increase the participation of the internet savvy citizens of India who mostly neglect this procedure of voting when it is done manually.

# REFERENCE:

1. Fundamentals of Software Engineering, 4th Edition by Rajiv Mall PHI Learning Private Limited.
2. Database Systems Models, Languages, Design and Application Programming, 6th Edition by Ramez Elmasri and Shamkant B. Navathe Pearson Publication.
3. Learning PHP & MYSQL by Michele E. Davis and John A. Phillips O’RELLY publication.
4. HTML and CSS. [http://www.w3schools.com](http://www.w3schools.com/)
5. PHP and MYSQL. [https://phpacademy.org](https://phpacademy.org/)
6. Object Oriented System Development 1st Edition by Ali Bahrami TMH Publication.