**SOFTWARE REQUIREMENTS SPECIFICATION**

**“UNI VOTE” - a college voting system**

**PROJECT MEMBERS:**

**SIREESH ODURU**

**NIKHIL KUMAR BAMANDLAPALLI**

**SUJIT REDDY MUDHUNOOR**

**1.0 Introduction**

This section provides an overview of the entire requirement document. This document describes all data, functional and behavioral requirements for software.

**1.1 Goals and objectives**

This Online Voting web application can work as automated voting system for small scale to large scale areas like institutions/colleges/universities (President, Vice President), organizations (team leads, manager), or public use (to elect public representatives).

**1.2 Statement of scope**

The Scope of the project is as follows: - To build a system through which the voters can cast their vote to their respective leader’s sitting anywhere in the world. To provide corruption free election. To avoid using archaic methods of voting.

**1.3 Software context**

The system working is like usual manual election system of voting where candidates with different position contest in election and voters will elect them as per their choice. The system has two phase of the user interface which are the administrator and voter’s side. The admin user is responsible to add and manage the data of the system and the voter side which is where the voters will vote their candidate and submit their vote.

**2.0 Usage scenario**

This section provides a usage scenario for the software. It organized information collected during requirements elicitation into use-cases.

**2.1 User profiles**

The three main end-users of this application would be: - Administrator: - Set election positions, add/remove voters and add/remove candidates. Voters: - Enter the application and cast their votes. Candidates: - Added to Application by administrator with required details.

**Use-cases**

Graphic Use Case Model

Diagram

Description automatically generated

Textual Description for each use case

**Use Case UC-1: Register**: - End User creates a new account on the website. As soon as user create account admin will give approval. End User Contestant/ Candidate register for contesting in election admin give approval.

Use Case UC-1: Register

Initiation Actor: End User

Actor’s Goal: - To register a user account to access the voting dashboard.

Participating Actors: Database

Preconditions: - The user is not yet registered and has a valid email address.

Postconditions: - The user has an account

Flow of Events for Main Success Scenario for User:

End User: - inputs the required details.

System: - verifies the information and send confirmation mail to user email and wait for admin to confirm the status.

Admin: - verifies the given details of the user and change status of the user.

System: - activates the user account for login

Flow of Events for Candidate:

End User: - inputs the required details

System: - verifies the information and send confirmation mail to candidate email and wait for admin to confirm the status

Admin: - verifies the given details of the candidate and change status.

System: - activates the candidate to display his detail in user dashboard to cast vote

Flow of Event for Alternative Scenario:

End User/Candidate: - enter invalid user details

System: - rejects the request and show error.

**Use Case UC-2: Login**: - End User Logs into the website. This will allow them to access the full features of the site, such as voting, checking ballot.

Use Case UC-2: Login

Initiating Actor: - End User, Administrator

Actor’s Goal: - To login into the system.

Participating Actors: Database

Preconditions: - The user has already created an account.

Postconditions: - The user will be logged into the site to use its functionalities

Flow of Events for Main Success Scenario:

End User/Admin: - enters the user login and password

System: - does a basic validation to make sure that the password and username are not empty string

System: - verifies the entered data by checking the Database. It then logs the user in and displays the End User/Admin dashboard

Flow of Events for Alternative Scenario:

End User/Admin: - enter an invalid username and password

System: - shows an error

**Use Case UC-3: Vote**: - End User Logs into the site and get access to vote to candidate of their choice.

**Use Case UC-4: Admin**: - Admin Logs into the site and manage votes, voters, candidates, positions and other details of the application.

**3**. **Functional Model and Description**

Functional Requirements

|  |  |  |
| --- | --- | --- |
| Functional Requirement ID# | Functional Name | Functional Requirements Description |
| ADMIN | | |
| 1 | Login | The System will validate the entered credential if matched enter dashboard or try again |
| 2 | Votes | See the cased votes and can reset for next election |
| 3 | Voters | Add/Remove voters from the System |
| 4 | Positions | Add/Remove positions from the System |
| 5 | Candidates | Add/Remove Candidates from the System |
| VOTER SIDE | | |
| 1 | Login | The System will validate the entered credential if matched enter dashboard or try again |
| 2 | Vote | Cast vote to candidate |

Non-functional requirements

Performance Requirements: -

* Online Voting shall support up to a million voters at a time.
* Response time of server should be less.

Safety and Security: -

* Votes must be placed once per user.
* Votes may not be altered in any situation.
* Voting rules are immutable Security
* Data us decentralized
* Unfeasibly hard to hack

**Hardware Interfaces:**

The required interface is through a computer system. The operating system can be Windows/Linux or MAC. Software Interfaces the OV runs on Apache/tomcat server.

**Communications Interfaces**

• A stable internet connection

• Communication interfaces include HTTP server hosted on Apache/tomcat server.

**System Requirements**

Operating System: Windows/Mac or Linux Processor x86 or x64

RAM 512 MB (minimum), 1GB (recommended).

Hard Disk up to 200 MB

PHP 5.3.3 Server

Apache Tomcat Database

MySQL

**4. Restrictions, Limitations, and Constraints**

This application is developed with keeping in mind with regards of the end-user as mentioned below: -

• The end-user who is in charge to maintain the project must be familiar with the working of system.

• Easy to use application with low constraints.

• The budget is as low as possible however if end-user want more functionalities the budget constraint may vary with each functionality added.

• UI is in core English Language only.

• Simple UI constraint to keep designing budget low if asked for more specific design change the budget constraint may vary.

**Assumptions and Dependencies**

• User must have basic knowledge of computer operating.

• Roles and tasks are pre-defined.

• The voting results will be managed and calculated by the logic.

• Administrator is deployed in the application by default.