CIS 634-OBJECT ORIENTED SOFTWARE ENGINEERING

Software Requirement Specification (SRS) Document

“UNI VOTE” - a college voting system

PROJECT MEMBERS:

SIREESH ODURU

NIKHIL KUMAR BAMANDLAPALLI

SUJIT REDDY MUDHUNOOR

INTRODUCTION

This section should provide an overview of the entire document.

1.1 Purpose

To automate the voting process to certify proprietary guarantee system of secured and efficient elections which are conducted in College or Institution.

1.2 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.

1.1 Definitions, acronyms, and abbreviations

SRS – System Requirement Specification

This SRS is easy guide to students (us), the professors who can act as our client and the project head. It will provide clear workflow of the application. Thus, this SRS is organized as follows: -

Chapter 1 shows the end-users (developers & clients) to understand the development stages of the application.

Chapter 2 gives an overall idea of the product.

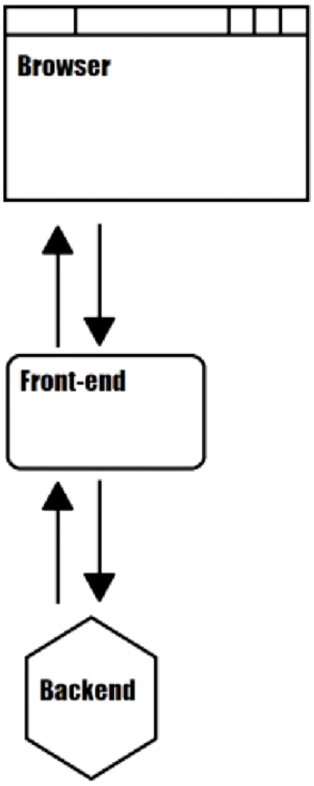
Chapter 3 and 4 represents the requirements and functionalities of the OV.

2. OVERALL DESCRIPTION

2.1 Product Perspective

The product is designed in such a way that it will put back the archaic voting system, anyhow it can be tested at small range (University and so on) and after being successful it might be executed at large range (Country).

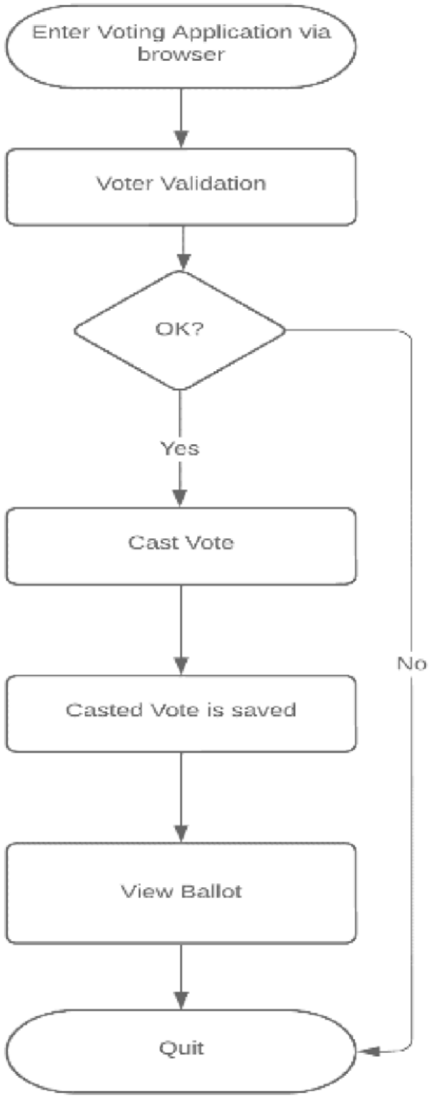
Here is the structure of our application that we are building.



Application consists of front-end that is designed with HTML, CSS and JavaScript. Along with PHP connectivity with the back end.

2.2 Product Functionality

As per the voter's perspective, the system is helpful for them to cast their votes and see to whom they voted. The main aim of this application is that voters can be assured that their votes will never be manipulated in any situations, and they can ultimately be a part of safe, secure and reliable elections.



2.3 End-Users and Characteristics

The three main end-users of this application would be: -

Administrator: - Set election positions, add/remove voters and add/remove candidates.

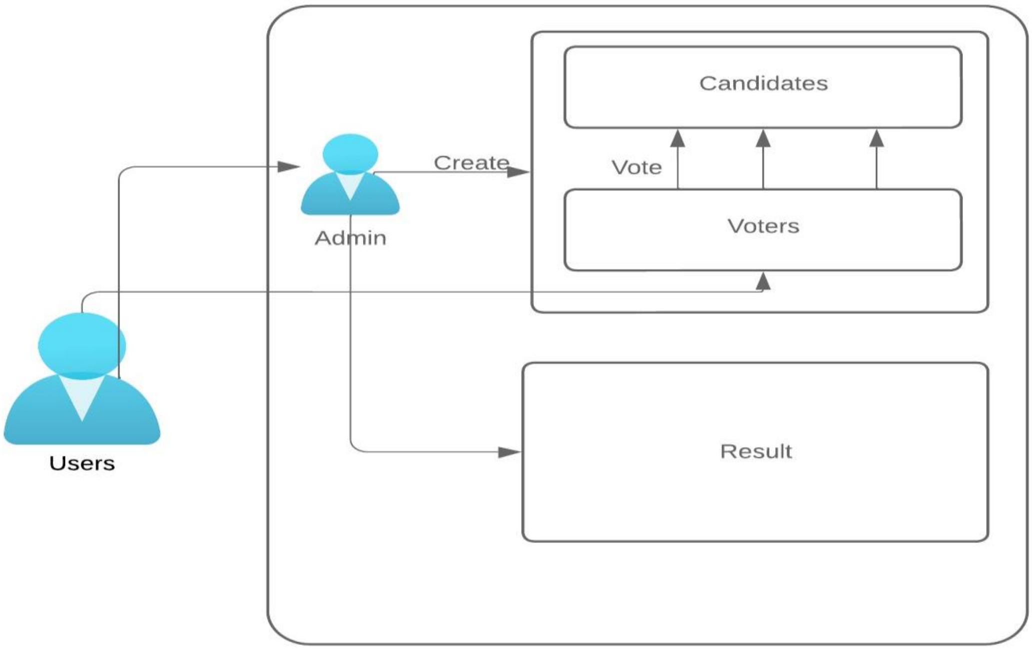
Voters: - Enter the application and cast their votes.

Candidates: - Added to Application by administrator with required details.

2.4 Operating Environment

The application can be installed on any machine, since it is a web application, it requires a stable internet connection. This application is compatible with Windows, MAC and Linux. There is no hardware component requirement other than computer to access the application.

The diagram shows the major components of the overall system.



2.5 Design and Implementation 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.

2.6 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.

3. Functional Requirements Description

3.1 Functional Requirement List

The following table shows the functional business requirements for the project: -

Functional Requirement ID #	Functional Name	Functional Requirement Description
	ADMIN SID E	
1	Login	The system will validate the entered credentials if matched enter into dashboard or try again
2	Votes	See the casted votes and also can reset for next election
3	Voters	Add/remove voters from the system.
4	Positions	Add/remove different positions for the elections.
5	Candidates	Add/remove the candidates.
	Voter Side	
1	Login	The system will validate the entered credentials if matched will login into the system or try again
2	Vote	Cast vote to candidate

3.1 **Hardware Interfaces**

- The required interface is through a computer system.
- The operating system can be Windows/Linux or MAC.

3.2 Software Interfaces

The system runs on Apache/tomcat server.

3.3 Communications Interfaces

- A stable internet connection
- Communication interfaces include HTTP server hosted on Apache/tomcat server.

3.4 System Requirements

Operating System	Windows/Mac or Linux
Processor	x86 or x64
RAM	512 MB (minimum), 1 GB (recommended)
Hard Disk	up to 200 MB
PHP	5.3.3
Server	Apache Tomcat
Database	MySQL

4. Other Non-Functional Requirements

4.1 Safety and Security

1. Votes must be placed once per user.
2. Votes may not be altered in any situation.
3. Voting rules are immutable.

Security: -

1. Data is decentralized.
2. Unfeasibly hard to hack.

4.2 Software Attributes

1. Availability.
2. Correctness.
3. Reusability
4. Testability.
5. Portability.
6. Reliability.
7. Usability.