Software Requirements Specification Document

Software Requirements Specification Document

Turnitin

Document Details

Submission ID

trn:oid:::2945:319958188

Submission Date

Oct 19, 2025, 11:40 AM GMT+5

Download Date

Oct 19, 2025, 11:41 AM GMT+5

File Name

unknown_filename

File Size

21.1 KB

5 Pages

1,272 Words

7,995 Characters

1% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text

Match Groups

2 Not Cited or Quoted 1%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

1%	Internet sources
0%	Publications
1%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **2** Not Cited or Quoted 1%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

1%	Internet sources
0%	Publications
1%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers UNICAF on 2025-08-29	<1%
2	Internet www.nanpa.com	<1%

Software Requirements Specification Document (SRS)

Project Title: University E-Voting System
Course: CSE 327 - Software Engineering
Prepared By: Tamim Ahmed Rijan
Date: 19th October 2025

1. Introduction

1.1 Purpose

This document provides the complete Software Requirements Specification (SRS) for the University E-Voting System. The goal of this project is to create a secure and transparent web application which will be used for conducting university level elections electing student councilors, club members and department representatives. This document specifies the objectives, target users, performance expectations and operational constraints of CAST-1000 system. It will serve as a formal tool for developers, testers, project managers, analysts, learners/educators and stakeholders through the life cycle of the software.

1.2 Intended Audience

This Software Requirement Specification (SRS) is for the audience who have interest directly or indirectly in design, development and use of the University E-Voting System.

- ***Administrators:*** Responsible for managing elections, user accounts, and overseeing system operations.
- ***Developers:*** The technical team in charge of coding, integration, and maintenance of the system.
- ***Project Managers (PMs):*** Oversee project progress, ensure scope alignment, and monitor team activities.
- ***Testers / Quality Assurance Engineers (QA):*** Validate and verify that each requirement is implemented correctly.
- ***University Election Committee:*** Official university body responsible for election supervision and policy compliance.
- ***Students (Voters and Candidates):*** End users who will vote, register as candidates, and access election information.
- ***Stakeholders / Observers:*** Faculty advisors, university authorities, and others interested in election transparency.

This SRS is meant to act as a knowledge tool for all with an interest in the system (from developers through University Election Authorities) so that they understand what role they play and how the system can be effectively designed delivered and operated.

1.3 Intended Use

This document will be used by the audience mentioned above to understand the aims, features and working of University E-Voting system and this section describes how the SRS will be used to enable each group of stakeholders to fulfill their roles effectively.

1.3.1 Administrators

- Use the SRS to understand system requirements, deployment steps, and their administrative responsibilities.
- Manage elections, approve candidate registrations, verify voters, and monitor results.
- Follow security and maintenance guidelines defined in the document to ensure reliable operation**.**

1.3.2 Developers

- Understand all functional, non-functional, and interface requirements.
- Use it as the main reference for implementation, integration, and database design.
- Ensure system features such as login, voting, and result generation match the requirements exactly.

1.3.3 Project Managers (PMs)

- Use the SRS to define project **scope**, **timeline**, and **deliverables**.
- Track team progress and ensure development follows the defined requirements.
- Use it for communication between developers, testers, and the university election committee.

1.3.4 Testers / QA Engineers

- Use this document to develop **test cases** for functional and performance testing.
- Verify that all specified requirements have been met and work as intended.
- Identify defects or inconsistencies and provide feedback to developers.

1.3.5 University Election Committee

- Use the SRS to verify that the system design aligns with **university election policies** and regulations.
- Review how authentication, voting, and result publication ensure fairness and transparency.
- Prepare training materials or manuals for administrative staff based on this document.

1.3.6 Students (Voters & Candidates)

- Understand their interaction flow in the system, like: registration, authentication, and voting.
- Learn about vote privacy and system security to build trust in the digital election process.
- Know the conditions, rules, and responsibilities defined for participants.

1.3.7 Stakeholders / Observers

- Refer to this document to gain insight into the project's scope, functionality, and outcome.
- Ensure that the E-Voting System supports ethical, fair, and transparent elections.
- Use it as a communication tool to align expectations between university authorities and the project team.

1.4 Product Scope

The objective of E-Voting System is to eliminate the traditional paper-based voting system and bring a centralized online based election system. It will help the election cycle from candidate registration to result publication, in a comfortable and secure way.

1.4.1 Purpose

This system aims to computerize all processes involved in the university-wide voting. It offers a quick, safe and easy solution to enable students in casting their votes, and administrators in managing the election effectively. Fair participation, reducing human error and increase access for every student.

1.4.2 Benefits and Objectives

- **Enhancing Election Efficiency:** Reduces time and effort required for managing and counting votes.
- **Ensuring Fairness and Transparency:** Prevents manipulation and guarantees one-person-one-vote.
- **Improving Accessibility:** Allows students to vote securely from anywhere within the university network.
- **Reducing Paper Usage:** Promotes an eco-friendly election process by eliminating paper ballots.
- **Immediate Results:** Automatically generates and displays election outcomes once voting ends.
- **Data Security:** Protects voter information through encryption and secure authentication.

1.4.3 Alignment with Institutional Goals

This system is consistent with the university’s aspiration for digital transformation, student engagement and transparent governance. It aligns with the institution's vision of leveraging technology to enhance administrative effectiveness and student involvement. With this online voting system, the university shows agility and it clear it wants to embrace new technologies for decision making.

1.4.4 Relating to Educational Strategies

The university enhances the approach of deploying smart reality solutions towards academic and administration services through deployment of University E-Voting System. The technology encourages fairness, inclusivity and education among students. It is also symbolic of the university’s emphasis on contemporary learning experiences and the cultivation of a culture that promotes trust and accountability in a student governance.

1.5 Risk Definitions

The Software Requirements Specification identifies potential risks that could affect the success, security, and reliability of the University E-Voting System. These risks may arise from user behavior, technical limitations, or administrative challenges. If not properly managed, such risks could compromise the fairness, accessibility, or transparency of the university’s election process.

Risk Type	Description	Mitigation Strategy
Security Risk	Unauthorized access or data manipulation	Use encryption (SSL/TLS), hashed tokens, and RBAC policies
System Failure	Server crash during election period	Daily backups and cloud redundancy
Privacy Risk	Voter identity linked to vote	Separate vote table from user records
Network Risk	Connectivity issues for users	Auto-reconnect and token-based resumption
Data Loss	Database corruption	Versioned backups and transaction logs
Administrative Risk	Untrained staff mismanaging data	Provide training and user manuals

1.5.1 User Inactivity

There's a possibility that students, or candidates may not actively take part in election. The lack of voter turnout in itself contributes to undermining the trustworthiness and full representativeness of both results and is not helping much when it comes to trusting democratic systems.

1.5.2 Administrator Workload

The administrators can have a heavy load when they are checking candidate nominations, maintaining the voter list and handling potential system crashes in the election time. This may result in delays or errors unless it is underpinned by effective tools and automation.

1.5.3 Communication Breakdown

Lack of communication between developers, testers and the university election committee can lead to wrongly identified requirements that need not be met or improper implementation. Frequent meetup and documentation is important to mitigate this risk.

1.5.4 System Failure

And also there may be some unknown server crashes, or network outages which breaks the voting. This is not as good when voters could lose access during pending elections." "There should be redundant systems and failover capability.

1.5.5 Security Vulnerabilities

Hackers can try to tamper election data or access it. Poor validation of authentication or weak encryption result in unauthorized access, double voting or data fraud.

1.5.6 Data Loss

System failures, voter or results data loss may occur as a result of system errors, accidental deletion and database corruption. Backup and restoration schedules need to be established so that this does not happen.

1.5.7 Changing Stakeholder Needs

Active requirements are added, post-development start date (by vote, admin or election-committee members). Regular contact with constituents is needed to handle these shifting expectations.