1. **SOFTWARE REQUIREMENTS SPECIFICATION (SRS)**

**Brief introduction**: This section of the documentation briefly covers the requirements for the development of the application to facilitate student campus nominations for yearly elections at the university. The application aims to provide a secure and efficient voting platform while ensuring the principle of a secret ballot and addressing privacy concerns raised in the provided case study.

**Scope of project:** The scope of the project includes the development of a Python-based application that allows currently registered students to participate in campus nominations for yearly elections. The application will be developed utilizing the Flask web framework for Python. Flask is a lightweight and flexible framework that provides essential tools for building web applications. Its minimalistic design allows for greater flexibility and customization, making it suitable for the development of the voting platform with specific requirements. The application will provide safety features to ensure that each student can cast only one vote and maintain the principle of a secret ballot. Additionally, the application will integrate security measures to protect user data and address concerns regarding privacy breaches.

**Functional requirements**:

* **User authentication** - **Verification of Identity**: The authentication system shall validate the identity of users by requiring them to provide their university credentials. **Login Mechanism:** Users shall be presented with a login interface where they can input their username and password. **Integration with University Credentials:** The application shall integrate with the university's authentication infrastructure to verify the validity of user credentials. This integration ensures that only students with valid university accounts can access the voting platform. **Registration Status Check**: Before allowing users to authenticate, the system shall verify whether they are currently registered students. This check ensures that only eligible students can participate in the elections. **Session Management**: Upon successful authentication, the system shall establish a session for the user, allowing them to interact with the application without needing to log in repeatedly during the same session.
* **Voting Process: User Authorization**: Only authenticated users shall be allowed to access the voting functionality of the application. Upon successful authentication, users will be granted permission to cast their votes. **Candidate Selection**: Each student shall have the opportunity to select one candidate for each available position in the election. The application shall present a list of nominated candidates for each position, allowing users to make their selections. **Vote Recording**: Once a user casts their vote, the application shall record it securely in the system's database. This recording ensures the integrity of the voting process and prevents duplicate or fraudulent votes. Vote **Modification Restriction:** After submitting their votes, users shall not be permitted to modify their selections. This restriction maintains the confidentiality and integrity of the voting process, preventing any attempts to alter votes after they have been cast. **Anonymous Voting**: The application shall enforce the principle of a secret ballot, ensuring that each user's vote remains anonymous. This anonymity fosters trust and integrity in the electoral process, allowing users to vote without fear of reprisal or coercion.
* **Candidate Nomination**: The application shall provide a mechanism for the nomination of candidates by eligible students. This mechanism includes a nomination form where students can input relevant details such as their name, position they are running for and a brief statement outlining their candidacy. Authorized administrators shall review and approve candidate nominations before they appear on the ballot.
* **Results Display**: After the voting period ends, the application shall display the election results. Results shall be displayed in a manner that preserves the anonymity of individual voters.

**NON-FUNCTIONAL REQUIREMENTS**

* **Security:** The application shall implement strong encryption mechanisms to protect user authentication credentials and voting data. Access controls shall be enforced to restrict unauthorized access to sensitive functionalities and data. Regular security audits and vulnerability assessments shall be conducted to identify and address potential security threats.
* **Usability:** The user interface of the application shall be intuitive and user-friendly, allowing students to navigate and interact with the system easily. Clear instructions and prompts shall be provided to guide users through the voting process.
* **Reliability:** The application shall be highly reliable, with minimal downtime and data loss. Data backups shall be performed regularly to ensure the integrity and availability of voting data.
* **Performance**: The application will be able to handle concurrent user interactions during peak voting times without significant performance degradation. Response times for user interactions, such as logging in, casting votes, and viewing results, shall be kept minimal to enhance user experience.