Project Realization Report

Objective: The objective of this project is to create a Java application that manages and analyzes voting data. The application has two main roles: **admin and voter**. **The admin** has the ability to add, delete, and edit elections, voters, and candidates. They can also view winner of the voting for each election. **The voter**, on the other hand, can log in using their username and password and vote in an election. Each voter is allowed to vote only once per election. The application provides functionalities such as calculating the winner of an election, updating voting records, deleting voting records, and retrieving all voting records.

System Structure: The system is structured around several key classes:

Candidate: This class represents a candidate in an election. It includes properties such as the candidate's name, age, gender, elect status, and photo.

Voter: This class represents a voter in the system. It includes properties such as the voter's name, age, gender, and voting status.

Election: This class represents an election. It includes properties such as the election's name, date, and status.

Vote: This class represents a vote in an election. It includes properties such as the voter's id, candidate's id, and election's id.

CandidateController: This class is responsible for managing the operations related to Candidate entities.

VoterController: This class is responsible for managing the operations related to Voter entities.

ElectionController: This class is responsible for managing the operations related to Election entities.

VotingService: This class handles the logic of voting, including the method calculateWinner that determines the winner of an election.

CandidateService: This class provides services related to the Candidate class, such as adding, updating, and deleting candidates.

VoterService: This class provides services related to the Voter class, such as adding, updating, and deleting voters.

ElectionService: This class provides services related to the Election class, such as adding, updating, and deleting elections.

CandidateRepository: This class interacts with the database to perform operations related to the Candidate class.

VoterRepository: This class interacts with the database to perform operations related to the Voter class.

ElectionRepository: This class interacts with the database to perform operations related to the Election class.

VotingRepository: This class interacts with the database to perform operations related to the Vote class.

These classes are related in that they all contribute to the overall functionality of the ElectionMaster application. The Candidate, Voter, Election, and Vote classes represent the entities of the application, while the Controller classes manage the operations related to these entities, the Service classes provide services related to these entities, and the Repository classes interact with the database to perform operations related to these entities.

Fulfillment of Assessment Criteria: The project fulfills the assessment criteria in the following ways:

- It is an object-oriented program written in Java.
- It uses classes and methods to encapsulate the functionality of managing elections, candidates, and voters.
- It provides a user interface for interacting with the system.
- It separates the business logic from the user interface.
- It uses a database to store and retrieve data (in memory database).
- It includes Javadoc comments for all classes and methods.

Major Program Versions: The major versions of the program that were submitted to GitHub are as follows:

Version 1.0: Initial version with basic functionality for managing candidates, elections, and voters.

Version 1.1: Added user interface for interacting with the system and database integration for storing and retrieving data.

Version 1.2: Added additional features such as calculating the winner of an election.

Javadoc Documentation: The Javadoc documentation is included in the project under the root directory of the project folder name called **doc** , providing detailed information about the classes and methods in the program.