REPUBLIQUE DU CAMEROUN
****
PAIX-TRAVAIL-PATRIE
UNIVERSITE DE BUEA
****
FACULTÉ D'INGÉNIERIE ET DE TECHNOLOGIE
****

REPUBLIC OF CAMEROON
****
PEACE-WORK-FATHERLAND
UNIVERSITY OF BUEA
****
FACULTY OF ENGINEERING AND TECHNOLOGY
****

DEPARTMENT OF COMPUTER ENGINEERING
DÉPARTEMENT DE GÉNIE INFORMATIQUE
COURSE CODE: CEF 476
COURSE TITLE: SOFTWARE ENGINEERING AND DESIGN
LECTURER: Dr. HUGUES MARIE KAMDJOU

DESIGN AND IMPLEMENTATTION OF AN ONLINE ELECTION SOFTEWARE

GROUP NAMES

NAMES	**MATRICULE**
NDIKINTUM CARL NFON	FE20A073
NYENTY EYONG ARREHQUETTE	FE20A094
OBASIARREY M’ONEKE MARY ARREY-NJOK	FE20A095
TANDONGFOR SHALOM CHANGEH	FE20A111

TABLE OF CONTENT

TABLE OF CONTENT..1
TITLE: ONLINE ELECTION SOFTWARE.. 3
I.PROJECT DESCRIPTION... 3
Key Features:...3
Technology used:... 4
REQUIREMENT GATHERINGS... 4
1. Identify Stakeholders:.. 4
2. Conduct Interviews:... 5
3. Gather User Stories:.. 5
5. Document Requirements:.. 5
6. Review and Validate Requirements:.. 5
ACTORS AND PARTICULARITY OF THE SYSTEM................................6
1. Voters:...6
2. Election officials:.. 6
3. System administrators:.. 6
4. Auditors:...6
Particularity.. 7
II. REQUIREMENT ANALYSIS AND SPECIFICATIONS............................7
Functional Requirements:... 7
1. Voter Registration:.. 7
2. Voter Authentication:...7
3. Vote Casting:...8
4. Vote Counting:.. 8
5. Reporting:... 8
Non-functional Requirements:.. 8
1. Security:..8
2. Scalability:...8

3. Performance:.. 9
4. Availability:.. 9
5. Usability:.. 9
6. Compliance:..9
III.DESIGN..9
Use case description..9
1. Register to Vote:.. 9
2. Cast Vote:.. 10
3. View Results:... 10
4. Manage Elections:... 10
5. Candidate Management:...10
USE CASE DIAGRAM.. 11
SEQUENCE DIAGRAMS..12
Registration and authentication... 12
Description.. 12
Voting and verification..13
Description.. 13
Tabulation of results.. 14
Description.. 14
Create election..15
Description.. 16
State diagram...17
Description.. 17
Class diagram.. 19
Description.. 19
Object diagram.. 20
Description.. 20
Package diagram... 21
Description.. 21
Deployment diagram.. 22
Description.. 22

Data flow diagram..23

Description.. 23

IV. IMPLEMENTATION...24

Github link: https://github.com/CAMON-23/CEF-476-SOFTWARE-ENGINEERING-AND-DESIGN..24

UI design:.. 24

Implementation:... 26

Participation list...27

Conclusion...27

TITLE: ONLINE ELECTION SOFTWARE

I.PROJECT DESCRIPTION

An online election software is a platform that allows voters to cast their votes electronically, rather than using traditional paper ballots. The software typically includes features such as voter registration, ballot creation, vote casting, and vote counting. The goal of an online election software is to make the voting process more efficient, secure, and accessible to a wider range of voters.

Key Features:

- User authentication and authorization

- Administration dashboard for creating and managing elections

- Voter registration and verification

- Electronic voting system with the ability to handle a large number of voters

- Real-time results display and reporting

- Secure and private storage of election data

Technology used:

- Front-end: HTML, CSS, JavaScript, ReactJS

- Back-end: NodeJS

- Database: Oracle

This project can be expanded or modified depending on the specific requirements, such as the type of election or the number of users.

REQUIREMENT GATHERINGS

The requirements gathering process for an online election software involves several steps to ensure that the system meets the needs of all stakeholders. Some key steps in the process include,

1. Identify Stakeholders:

The stakeholders who will be using the system include election officials, voters, administrators, and system administrators.

2. Conduct Interviews:

Once stakeholders are identified, conduct interviews to understand their needs, goals, and requirements. This can include asking questions about user experience, system functionality, security, and other topics.

3. Gather User Stories:

User stories are specific examples of how a user would use the system to achieve their goals. They should be gathered from both election officials and voters, and they can help identify specific features and functionality that the system should have.

5. Document Requirements:

Once all requirements have been gathered, document them in a clear and concise format, including functional and non-functional requirements, use cases, and user stories.

6. Review and Validate Requirements:

Finally, review the requirements with stakeholders to ensure that all requirements have been captured accurately, and validate that the proposed solution meets the needs of all stakeholders.

ACTORS AND PARTICULARITY OF THE SYSTEM

In an online election software, there are several actors involved in the system. These actors include:

1. Voters:

These are the individuals who are eligible to vote in the election. They use the online election software to cast their votes.

2. Election officials:

These are the individuals responsible for managing the election. They use the software to create the ballot, manage voter registration, and monitor the voting process.

3. System administrators:

These are the individuals responsible for maintaining the online election software. They ensure that the system is secure, up-to-date, and functioning properly.

4. Auditors:

These are the individuals responsible for verifying the accuracy and integrity of the election results. They review the voting data and ensure that the results are accurate and free from tampering.

Particularity

The particularity of an online election software is that it relies on technology to facilitate the voting process. This means that the system must be designed to be secure, reliable, and accessible to all voters.
-The software must be able to protect against fraud and ensure the privacy of voters.
- Additionally, the system must be designed to be user-friendly. - The use of technology in the voting process also means that the system must be able to handle a large volume of traffic and be able to withstand potential cyber attacks.

II. REQUIREMENT ANALYSIS AND SPECIFICATIONS

Functional Requirements:

1. Voter Registration:

The system must allow eligible voters to register themselves for the election and verify their identity.

2. Voter Authentication:

The system must authenticate the voters' identity and eligibility during the voting process.

3. Vote Casting:

The system should provide an interface for voters to cast their votes.

4. Vote Counting:

The system must count the votes accurately to determine the winner of the election.

5. Reporting:

The system should be able to generate reports on the election results.

Non-functional Requirements:

1. Security:

The system must ensure the confidentiality, integrity, and availability of the voting process and results.

2. Scalability:

The system must be able to handle a large number of many users, especially during voting periods.

3. Performance:

The system must be responsive and provide an ideal user experience.

4. Availability:

The system must always be available.

5. Usability:

The system must be user-friendly and accessible for all users.

6. Compliance:

The system must comply with all legal requirements and rules for online elections.

III.DESIGN

Use case description

The use case diagram typically the following use cases:

1. Register to Vote:

The voter creates an account on the online election software and provides the necessary information to make him eligible to vote.

2. Cast Vote:

The voter selects the candidate of their choice and submits their vote through the online election software.

3. View Results:

The voter, candidate, and admin can view the election results after the voting period has ended.

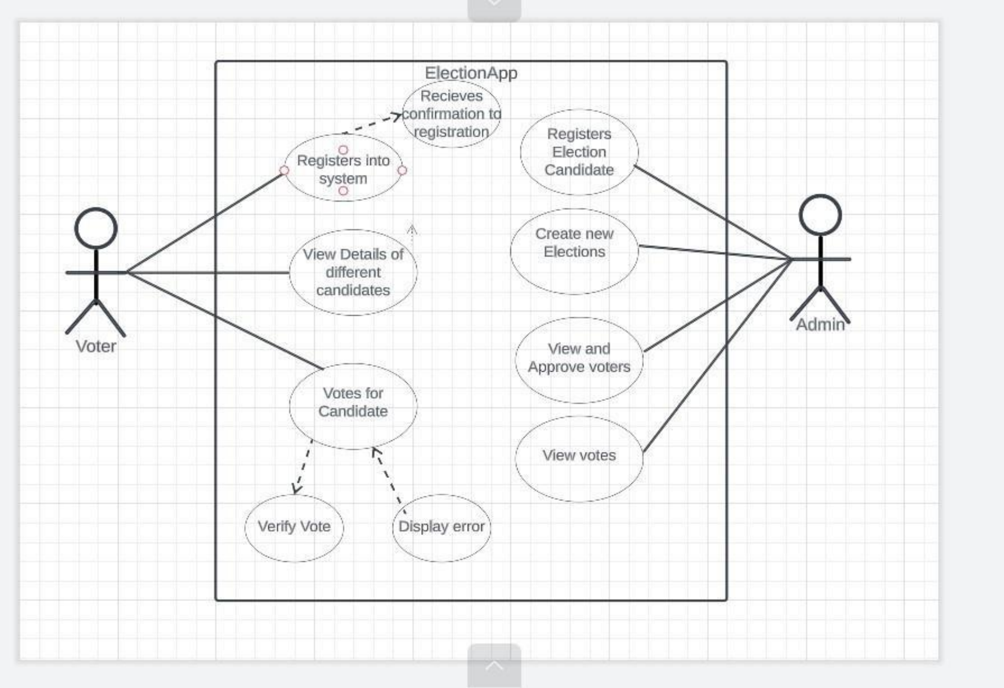
4. Manage Elections:

The administrator manages the election process, including setting up the election, managing voter registration, and ensuring the security of the voting process.

5. Candidate Management:

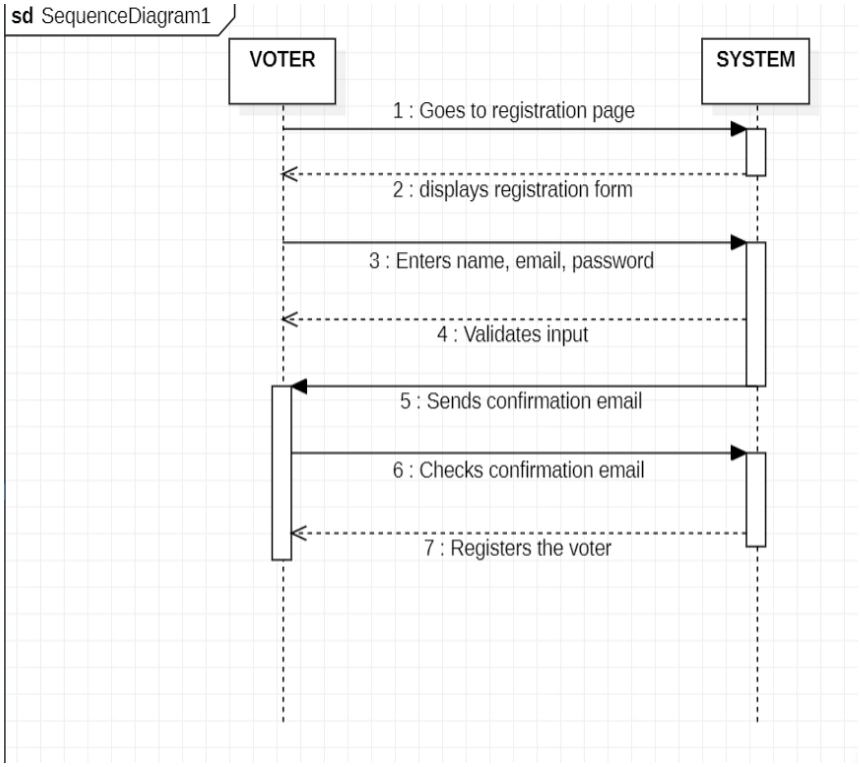
The administrator manages the candidates running in the election, including adding and removing candidates from the ballot.

USE CASE DIAGRAM



SEQUENCE DIAGRAMS

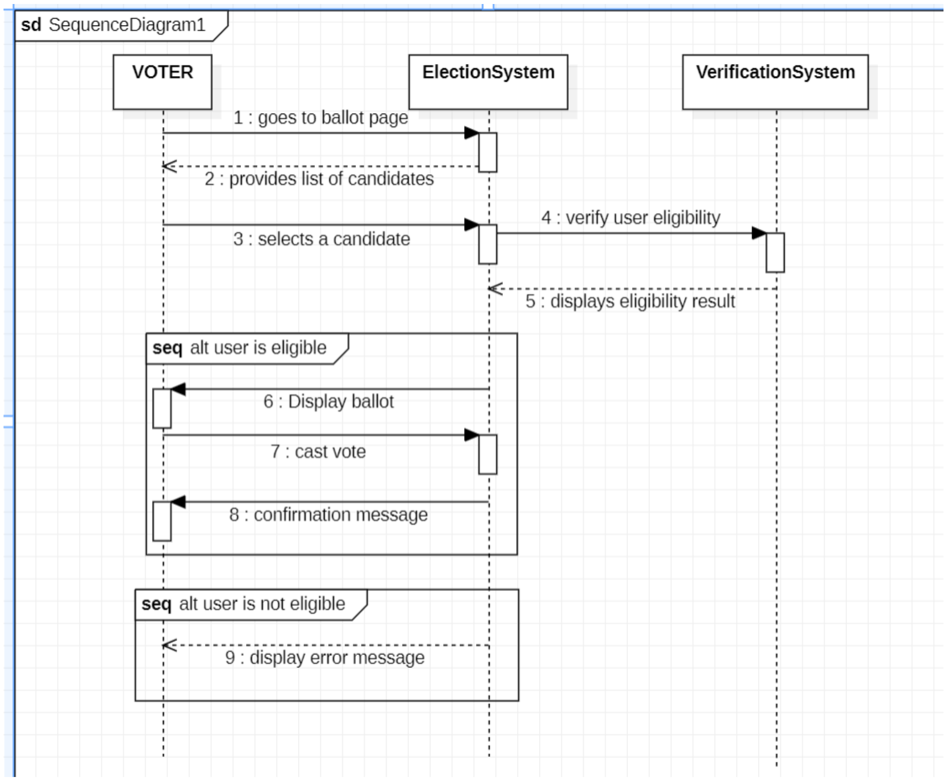
Registration and authentication


sd SequenceDiagram1
VOTER
SYSTEM
1 : Goes to registration page
2 : displays registration form
3 : Enters name, email, password
4 : Validates input
5 : Sends confirmation email
6 : Checks confirmation email
7 : Registers the voter

Description

- User goes to the registration page.
- The ElectionSystem displays the registration form.
- The User then fills out the registration form with their names, email and password.
- The ElectionSystem validates the input and sends a confirmation email. -The user checks the confirmation email
- Finally, the user is registered.

Voting and verification

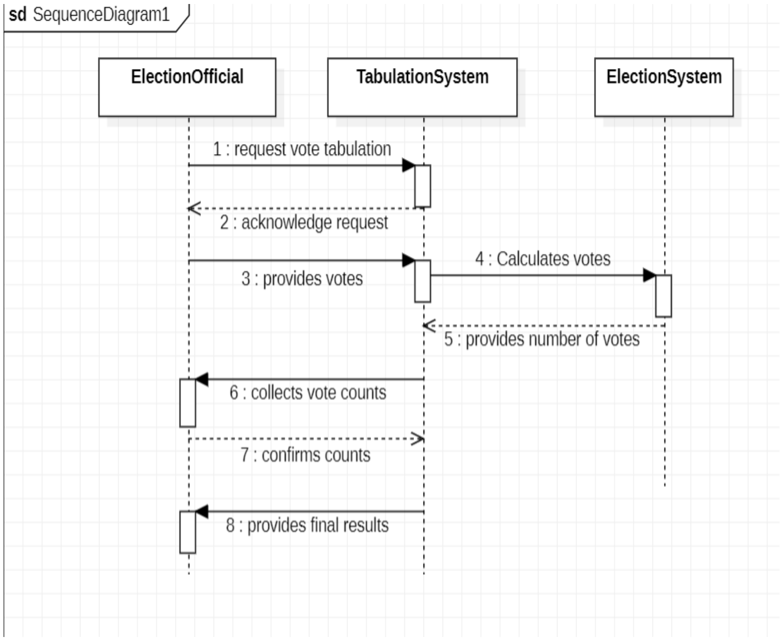


Description

- The user goes to the ballot page.
- The ElectionSystem responds by providing the User with a "Provide list of candidates" message, which displays the available candidates for voting.
- The User then selects a candidate and sends a "Select candidate" message to the ElectionSystem.
- The ElectionSystem now initiates the verification process by sending a "Verify user's eligibility" message to the VerificationSystem.
- The VerificationSystem performs the eligibility verification and sends an "Eligibility verification result" message to the ElectionSystem.

- Based on the result, If the User is eligible, the ElectionSystem sends a "Display ballot" message to the User, which allows them to cast their vote. The User sends a "Cast vote" message to the ElectionSystem.
- After voting, the ElectionSystem sends a "Confirmation message" to the User.
- If the User is not eligible, the ElectionSystem sends a "Display error message" indicating the ineligibility.

Tabulation of results

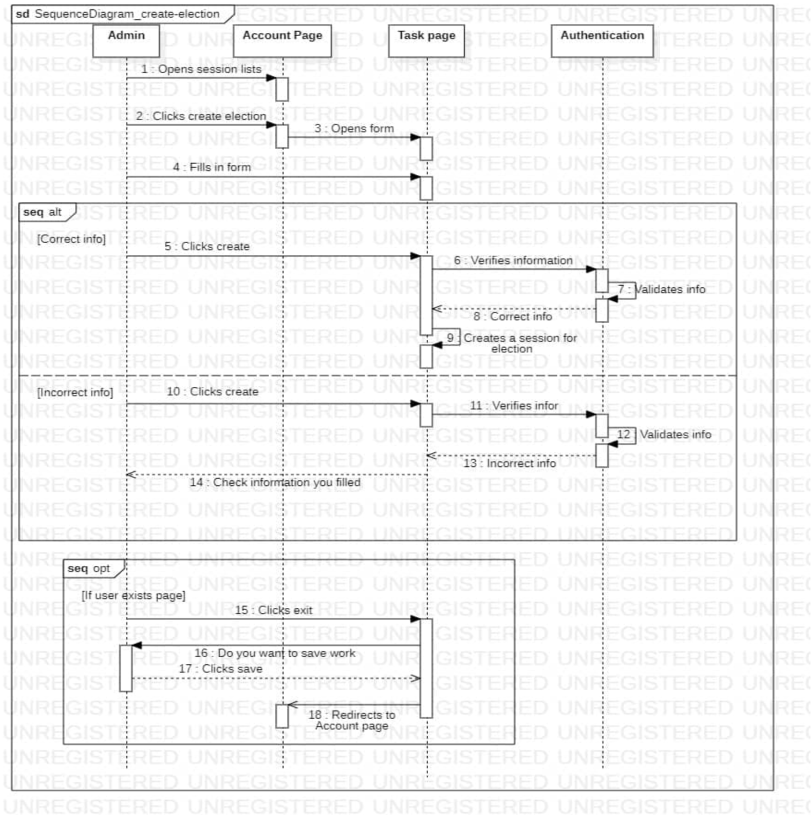


Description

- The ElectionOfficial sends a "Request vote tabulation" message to the TabulationSystem.
- The TabulationSystem acknowledges and sends an "Acknowledge request" message.
- The ElectionOfficial then sends a "Provide votes" message to the TabulationSysytem.

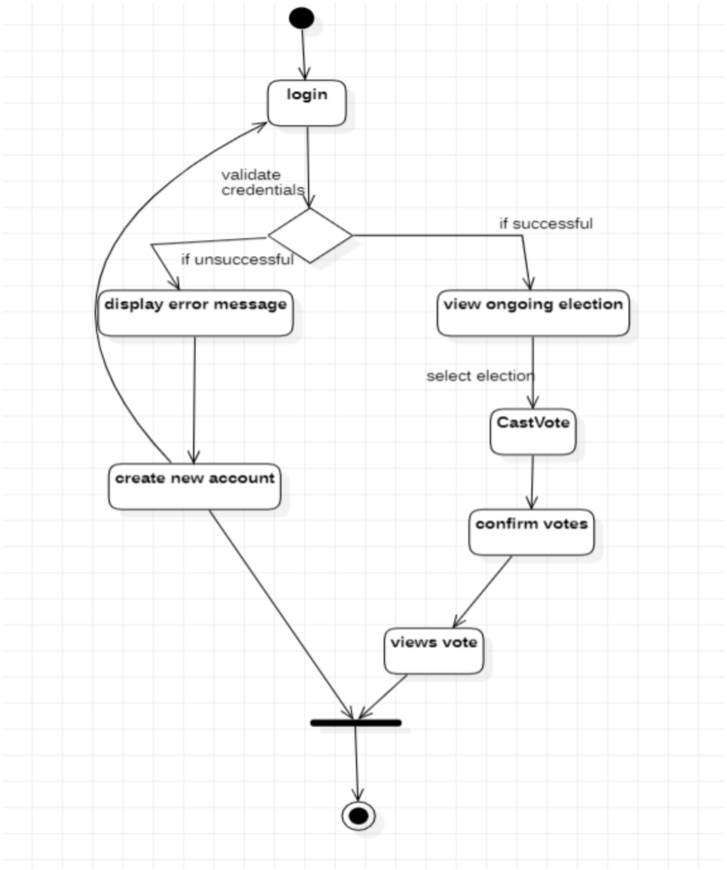
- The TabulationSystem uses this data to calculate the votes for each candidate.
- The ElectionSystem sends a "Provide number of votes" message.
- The TabulationSystem collects the vote counts and sends a "Collect vote counts" message to the ElectionOfficial.
- The ElectionOfficial confirms the vote counts by responding with a "Confirm vote counts" message.
- Finally, the TabulationSystem provides the final election results to the ElectionOfficial with a "Provide final results" message.

Create election



Description

- The above sequence diagram is for the Creation of elections: this is done by the Administrator. He/She have tabs on their account on which their

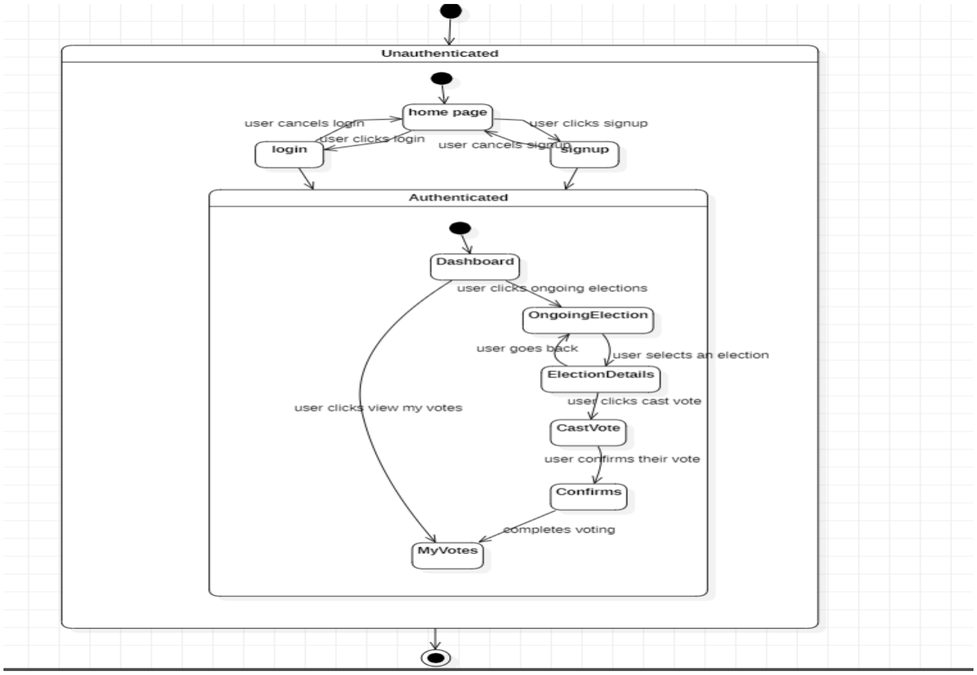


Description

- Users log into the platform.
- Users view the ongoing elections.
- Users cast their vote in a specific election.
- User confirms their vote
- Users view the results of an election.
- Logout: User logs out of the platform.
- End: Represents the end of the activity.

- The system displays an error message when an error occurs during the process.
- The system displays a confirmation message when an action is successful.

State diagram

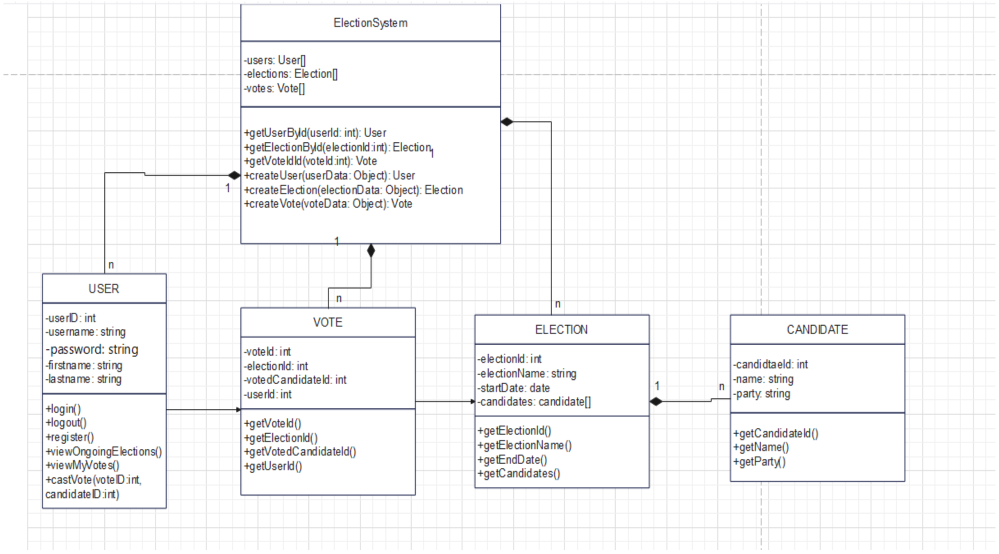


Description

- The initial state is "Unauthenticated," where users start when they access the online election platform.
- From the "Unauthenticated" state, users can navigate to the "Home" state, where they have options to either "Login" or "Signup."

- If a user successfully logs in or signs up, they move to the "Authenticated" state.
- In the "Authenticated" state, users have access to the "Dashboard," where they have options to view "Ongoing Elections" or "My Votes."
- When users choose to view "Ongoing Elections," they go to the "OngoingElections" state, where they can see a list of ongoing elections.
- From the "OngoingElections" state, users move to the "ElectionDetails" state to view more information about that election.
- They can then go back to the "OngoingElections" state or choose to "Cast Vote" in the "CastVote" state.
- Users confirm their vote in the "Confirmation" state, then move to the "MyVotes" state.
- In the "MyVotes" state, users can view their votes
    - Users view the results of an election.
    - Logout: User logs out of the platform.
    - End: Represents the end of the activity.
    - The system displays an error message when an error occurs during the process.
    - The system displays a confirmation message when an action is successful.

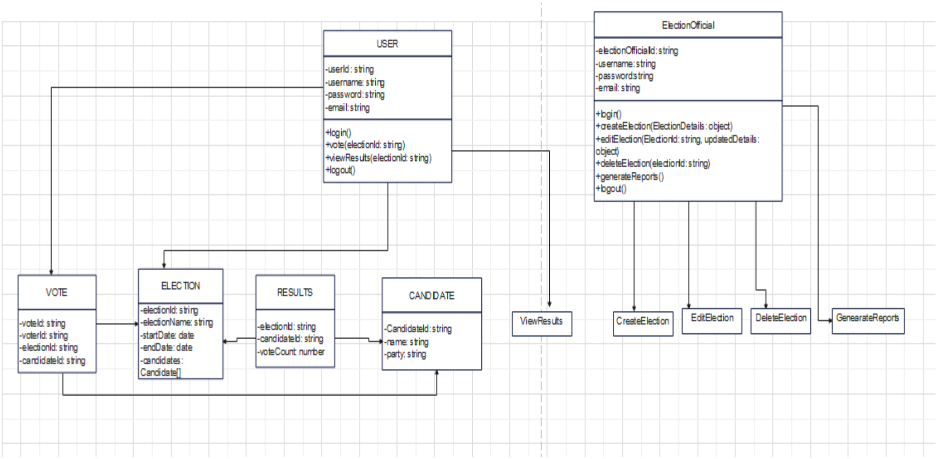
Class diagram



Description

- User class: Represents a user of the system with attributes such as userId, username, password, email, firstName, and lastName. It has methods for login, logout, registration
- Election class: Represents an election with attributes like electionId, electionName, startDate, endDate, and a list of candidates. It has methods to retrieve information about the election and its candidates.
- Vote Class: Represents a user's vote for a specific election with attributes like voteId, electionId, votedCandidateId, and userId.
- Candidate class: Represents a candidate in an election with attributes such as candidateId, name, and party.
- ElectionSystem class: Represents the main system that manages users, elections, and votes. It maintains collections of users, elections, and ballots and provides methods for creating new instances and retrieving existing ones.

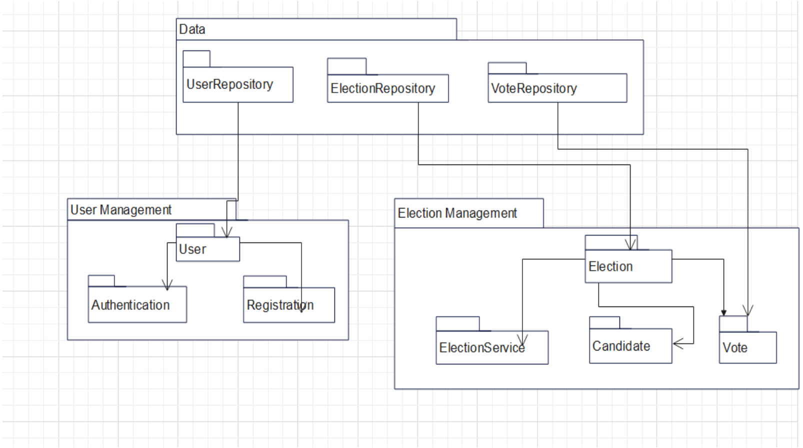
Object diagram



Description

- User: Represents a user of the platform, with attributes such as userId, username, password, and email. It has methods for login, voting, viewing results, and logout.
- ElectionOfficials: has attributes such as ElectionOfficailId, username, password, and email. It has methods for login, creating, editing, and deleting elections, generating reports, and logout.
- Election: Represents an election, with attributes such as electionId, electionName, startDate, endDate, and candidates
- Candidate: Represents a candidate in an election, with attributes such as candidateId, name, and party.
- Vote: Represents a vote cast by a user, with attributes such as voteId, voterId, electionId, and candidateId.
- Result: Represents the result of an election, with attributes such as electionId, candidateId, and voteCount.

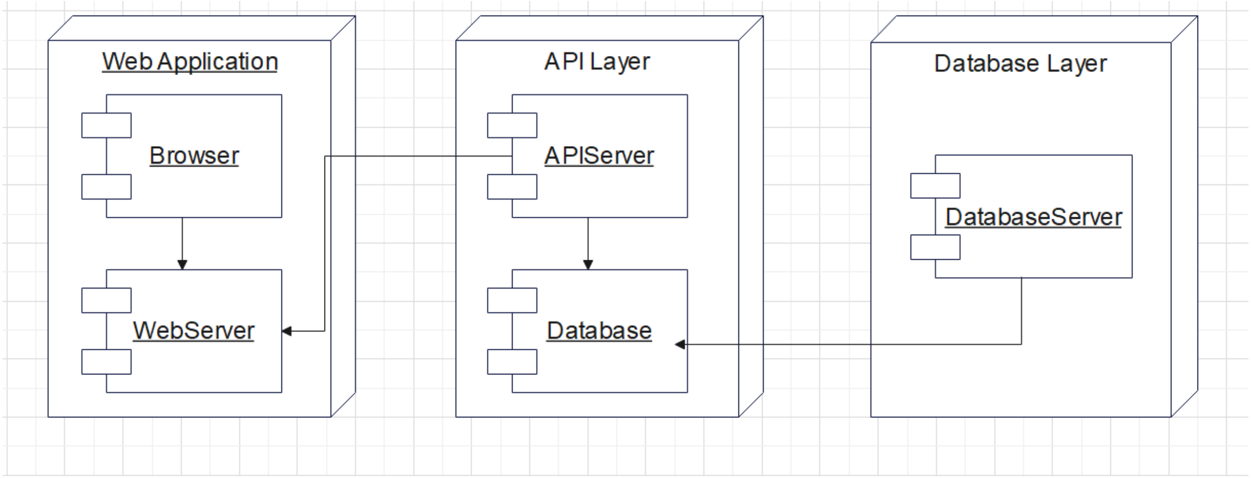
Package diagram



Description

- UserManagement: Contains the User interface, Authentication interface, and Registration interface for managing user-related operations.
- ElectionManagement: Includes the Election interface, Candidate interface, vote interface, and ElectionService interface for managing elections and related entities.
- Data: Represents the package responsible for data storage and retrieval, including the UserRepository, ElectionRepository, and VoteRepository Interfaces.

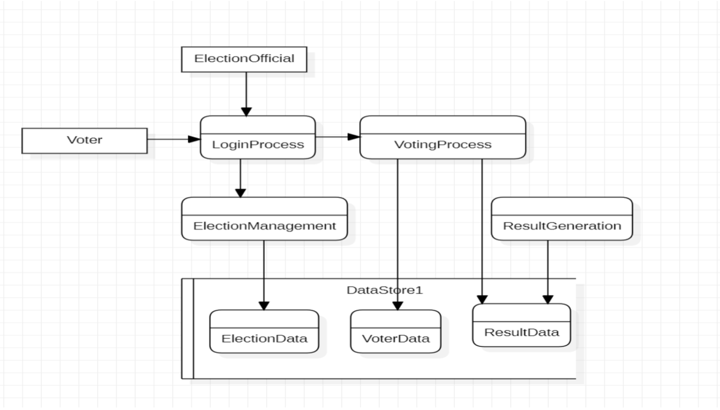
Deployment diagram



Description

- Web Application: Represents the user-facing web application that runs on the WebServer.
- API Layer: Represents the server-side APIs responsible for handling client requests and interacting with the Database.
- Database Layer: Represents the database server and the associated database.

Data flow diagram



Description

- LoginProcess: Handles the login functionality and authentication for both voters and ElectionOfficials.
- ElectionManagement: Manages the creation, modification, and deletion of elections, interacting with the ElectionData store.
- VotingProcess: Handles the voting process, interacting with the VoterData store to record user votes and the ResultData store to update the vote count.
- ResultGeneration: Generates election result reports based on the data stored in the ResultData store.

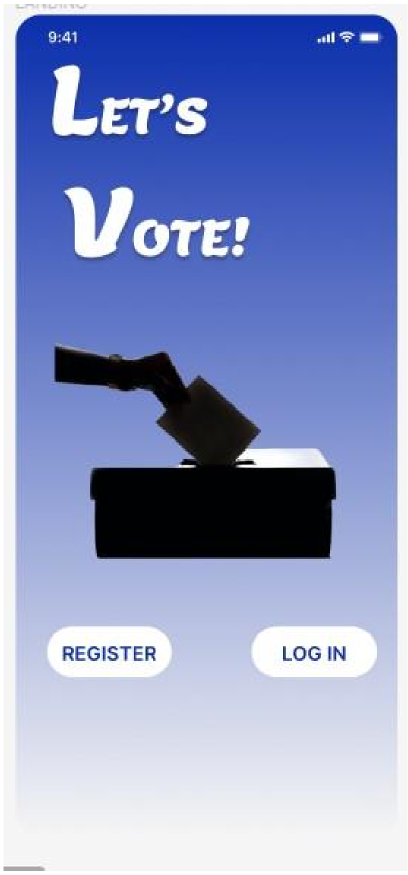
IV. IMPLEMENTATION

Github link:

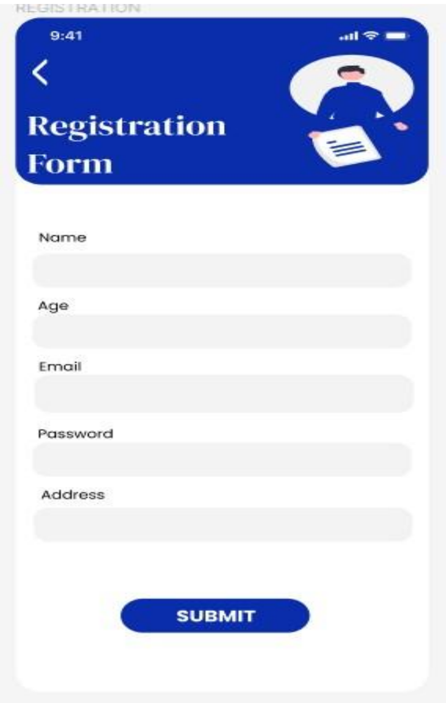
https://github.com/CAMON-23/CEF-476-SOFTWARE-ENGINEERING-AND-DESIGN

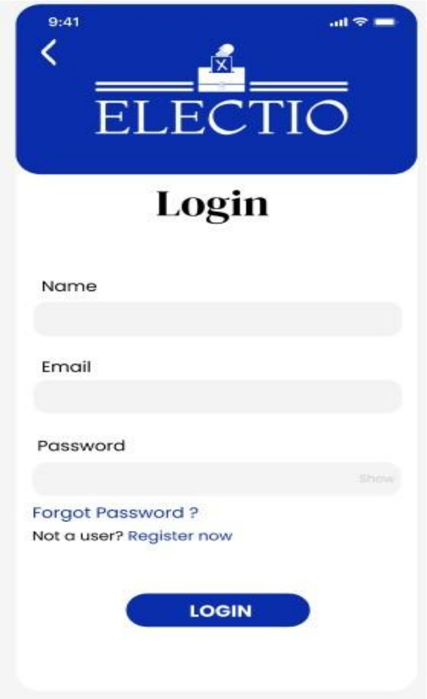
UI design:

The user interface was done using figma. The blue color means loyalty, honesty, trust, responsibility, calm, commitment, wisdom, and serenity. So this color was chosen to portray these different qualities for the users to have confidence in the application.

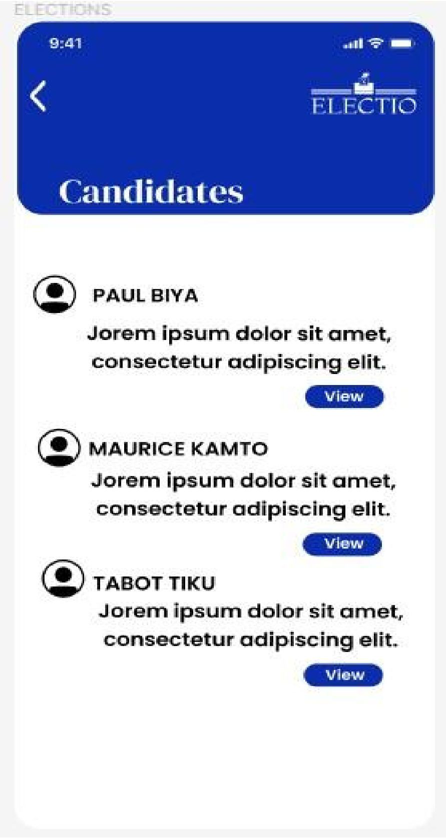


This design is the welcome screen of the Electio application which gives users the option to register as new voters or login to their already existing accounts.





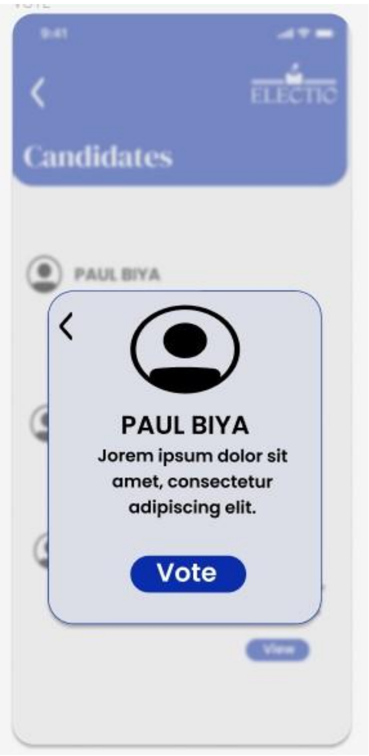
To the left, the two designs show the Login and registration forms available for its users.



Jorem ipsum dolor sit amet, consectetur adipiscing elit.

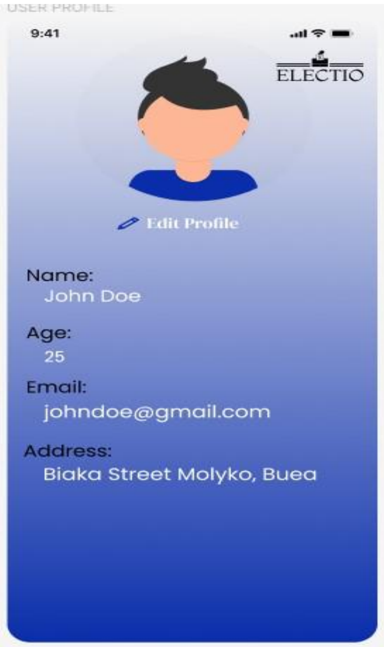
Jorem ipsum dolor sit amet, consectetur adipiscing elit.

Jorem ipsum dolor sit amet, consectetur adipiscing elit.



Jorem ipsum dolor sit amet, consectetur adipiscing elit.

These two designs are for the viewing of candidates and voting based on the preferences of the voter. The voter won’t be allowed to see the number of voters for a particular candidate. only the Admin will have such privilege.



Name:
John Doe

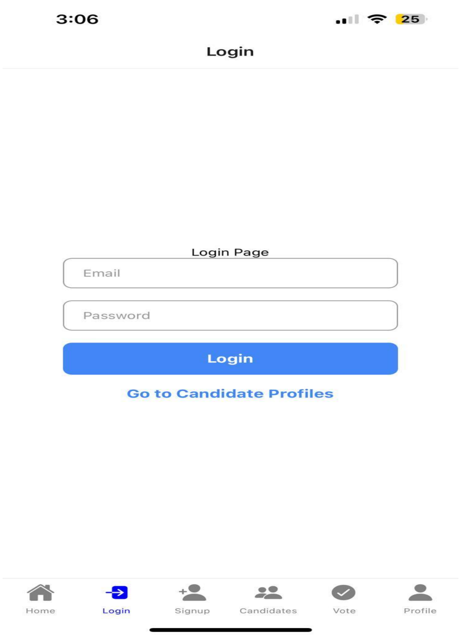
Age:
25

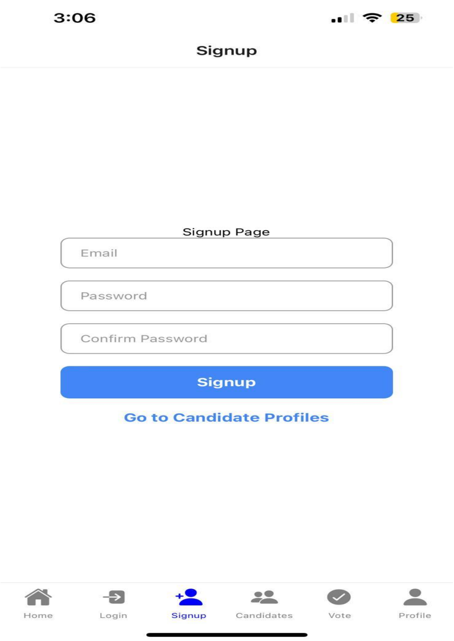
Email:
johndoe@gmail.com

Address:
Biaka Street Molyko, Buea

Here is the profile page of a voter.

Implementation:





Above are the implementation of both the Login and registration pages using React Native. The coding was done online, Below is a link to the code.

https://snack.expo.dev/@slicky/elections?platform=web

Participation list

Name	Work done
Ndikintum Carl Nfon	Project Description,key features, class diagram, object diagram, package diagram
Nyenty Eyong Arrehquette	Actors and particularities,functional requirements, use case diagram, state diagram, activity diagram, data flow diagram, UI design
Obasiarrey M’oneke Mary	Technology used, requirement gatherings, non-functional requirements, all 4 sequence diagrams, Implementation.
Tandongfor Shalom Changeh	Deployment diagram, UI design

Conclusion

Implementing an election mobile app in order to solve the problem of inefficiency, insecurity and inaccessibility to a wide range of voters, wasn’t an easy task. Irrespective of the shortcomings within the project we were able to learn from one another.