Automated Voting System

Name	Email	ID
Merna Abdelbadie	mernaabdelbadie@aucegypt.edu	900203731
Menatalla Aly	Mennatalla@aucegypt.edu	900191214
Salma Mohamed	salmamagdy@aucegypt.edu	900194112
Andra Kodsy	andra_hany@aucegypt.edu	900182621
Mariam Eskander	mairamehab@aucegypt.edu	900182354
Rana Emam	rana_emam10@aucegypt.edu	900182823

Table of Contents

Table of Figures

Overview Of The System:

Our project is an automated voting system. We have decided to choose it as our project because, as Egyptians, we know that the automated voting system in Egypt is not applied yet, and due to several factors such as the huge population and the issues in processing the votes in the paper-based voting system, we have seen that it is rather a good initiative to create a system that can be later deployed in Egypt. Therefore, our possible client is the Egyptian government as we are trying to help in easing the voting process for both the government and the population.

The system flow is as follows: the user will first be asked to enter his ID and password. Then, he will choose the person he/she wants to vote for if he/she hasn't voted yet. Finally, the voter will receive a confirmation message that he successfully voted. If the voter has already voted and tried to vote again, he/she will receive a message telling them that they already voted and can not vote again. On the other side, the government will be updated with the votes. The system also shows people the vote count of every candidate before voting and allows users to communicate with customer support via chat in order to resolve any technical issues or just send their feedback on the voting process.

Risks:

- **Security:** As security in this phase is just done using a password (no face recognition), this might cause people to use others' IDs to vote for their candidates. However, if the system were to be deployed in real life, the authentication process using face recognition would be implemented.
- **Accessibility:** not all people throughout Egypt have internet access or good connections
- **System failure:** if many queries are required in a shorter period of time, it might cause the system to fail in a critical time which will lead to people not being able to vote.
- **Convenience:** the system might not be convenient for people who are sight impaired or people who do not know how to read or use technology.

Product Backlog:

Functional Requirements:

a) **Login:** Users are required to log in using their national ID and password:
   i) Error messages would appear if:
      1) National ID has letters
      2) National ID is invalid
      3) Password is invalid
      4) Both National ID and Password are invalid
      5) Password does not match National ID
   ii) Promote integrity by showing a message saying that "By submitting, I acknowledge that the national ID entered is mine."

b) **Data Processing And Storage:** When a user enters information, the system keeps that information in a database. The administrator and electoral committee should be able to view all of the eligible voters in the system.

c) **Uploading Information About The Election Candidates:** Users can see all information about the elected candidates, such as their names and description.

d) **Voting Process**: Users will be able to vote after reading about the candidates by choosing between the different candidates' names provided.

e) **Identify How Much Needed To Vote:** If the voter opens the application and does not vote in 5 minutes, the page automatically will be closed.

f) **Voter Confirmation:** The voter should receive a confirmation that his/her vote has been recorded.

g) **Security For The Prevention Of Multiple Votes:** Each voter can vote only one time. A message should appear to voters who vote that they already voted and cannot vote again.

h) **Customer support:** Users can access Customer support via chat.
   i) Only one feedback per session will be allowed.
   ii) Feedback, along with the National ID of the user, should be written in order for the feedback to be sent.
   iii) It will not be allowed to have letters in the National ID
   iv) A confirmation message must appear once a feedback is sent.

i) **Flexible Cancellation Policy:** Voters have 5 minutes to choose their candidate before submitting; otherwise, the page will close automatically

j) **Announcing The Result:** The system shows the vote count for each candidate. After showing the results, the page should close automatically.

Non-Functional Requirements:

a) **Reliability:** The system shall be strong enough to have a high degree of error tolerance. For instance, the system should not allow inaccurate entries and should identify the inaccurate input and produce an error message.

b) **Security:** The system should implement measures to thwart hackers and unauthorized users. The administrator would only permit approved users to vote (age 18 and above). The system must be strong enough to withstand a variety of fraudulent activities. The system must also be transparent and understandable enough for voters.

c) **Performance:** The average response time for an automated voting system should be around 5 seconds. When a user queries a system, the response time is the amount of time the user must wait before receiving a response.

d) **Integrity:** Voter registration is only authorized by the system administrator. To safeguard the databases, the system needs to be both physically and logically secure. Before gaining access to the system, the administrators must authenticate themselves.

e) **Scalability:** The system must be able to grow to accommodate the government's changing needs while continuing to fulfill the original intent for which it was designed.

f) **Usability:** The system should be clear to users. The system must have an intuitive user interface so that users do not strain when interacting with it. Service notifications and error messages must be concise, professional, and free of complexity. The user interface must be simple to use and navigate.

g) **Availability and Accessibility:** The system must be operational during the voting period.

h) **Interoperability:** The system ought to be compatible with other current systems. Backward and forward compatibility should be guaranteed.

i) **Transparency:** Voters should be able to possess general knowledge and understanding of the voting process.

j) **Accuracy:** The system should correctly record and count all votes.

The Detailed Architecture:

The high architecture pattern that is used in the project is the layered architecture. There are three layers in the architecture: The presentation layer, the persistence layer (using MySQL), and the database layer.

In the presentation layer, there is the user interface which is responsible for the login along with the candidate view, voting, and vote count view. Once the user interface receives requests, it will send them to the persistence layer, as the user interface does not know where the data is or how it will be retrieved. The persistence layer, which contains both the user database access object and the administrator database access object, will then execute SQL queries based on the information received from the user interface or the administrator. The executed SQL queries will then be sent to the database layer that contains the database itself so that the required information will be retrieved. The data would flow back to the persistence layer, followed by the presentation layer to be shown to the users.

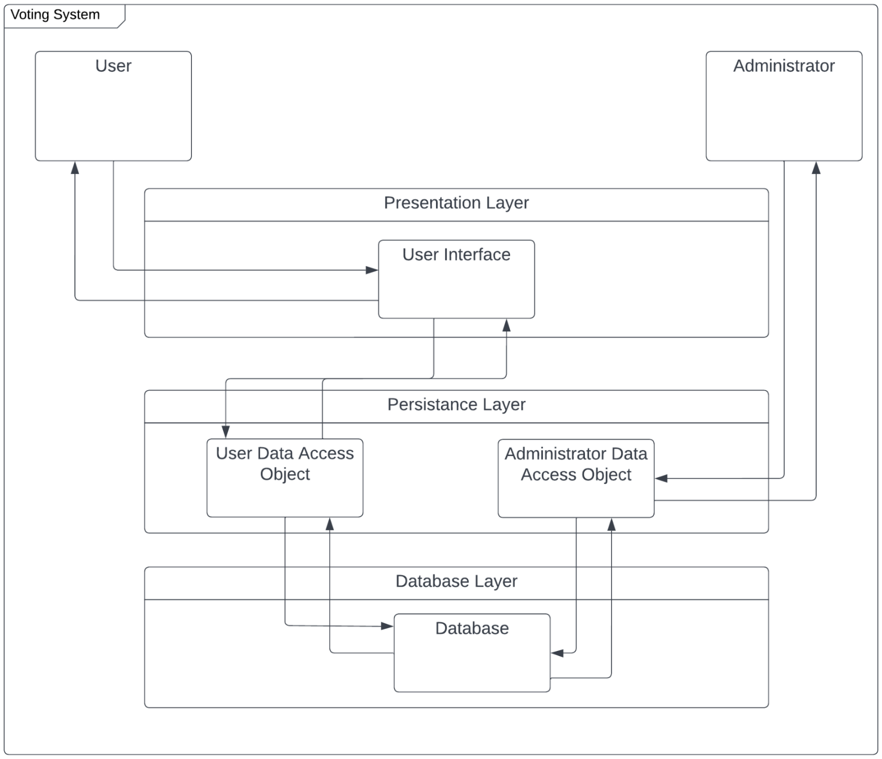

Figure 1: Detailed Architecture

The Detailed Design:

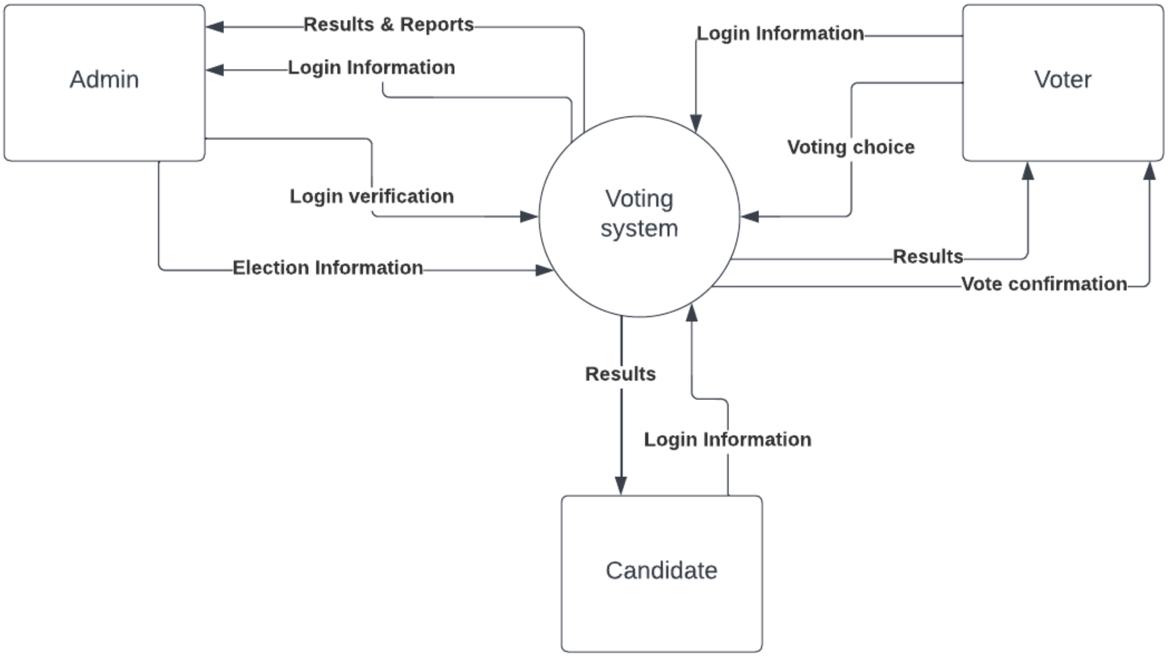

Figure 2: Context Data Flow Diagram

Our voting system interacts with three external entities, admin, voter, and candidate. As seen above, this diagram describes the data flows to and from the voting system and said external entities. Said data flows are based on the functional and non-functional requirements specified earlier.

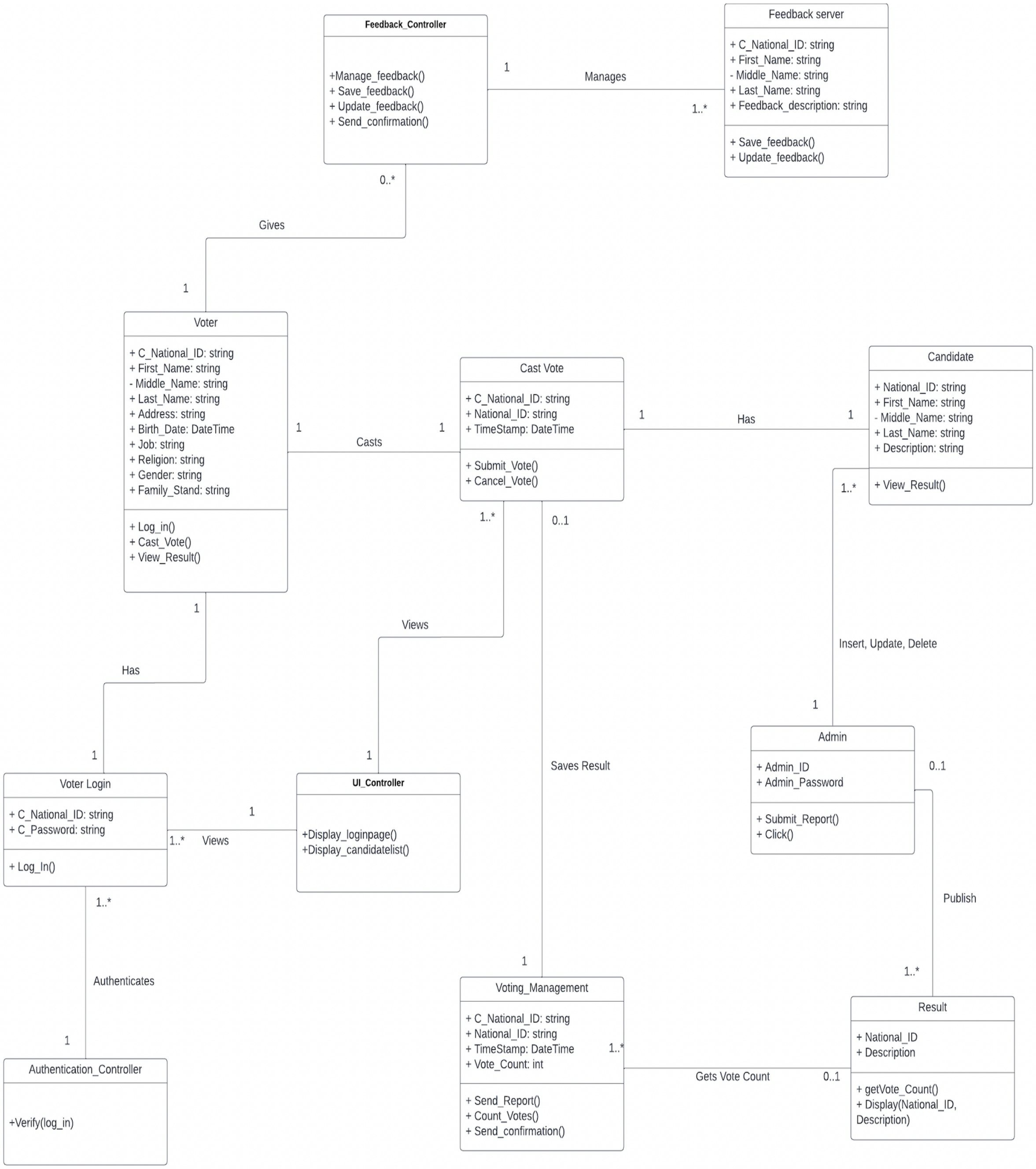

Figure 3: Class Diagram

This diagram describes how we wish to implement our system using object-oriented design. We decided on the base and controller classes based on the use case diagram and requirements specified. The base classes are used to decouple the functionality of the system. The UI controller class is used to serve as an interface for the system for the user. Other controller classes are used for needed functionalities like verifying login information, managing the feedback collected from the user, and managing the voting functionality.

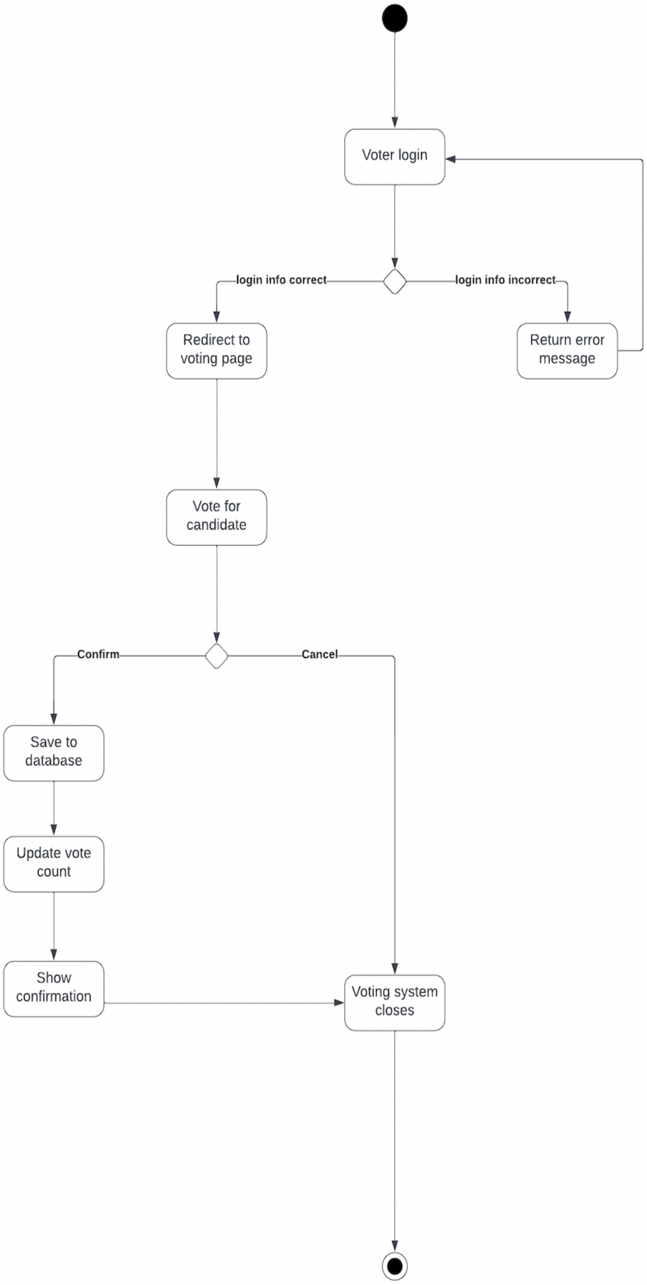

Figure 4: Activity Diagram

This activity diagram describes the flow of the voting process. It involves logging in and includes how the system should respond when incorrect information is entered. Moreover, it also includes the supposed system response if the voter wishes to cancel their voting session. Normally the voter should be able to log in and vote for their desired candidate. The system saves and updates the vote count and subsequently shows voting confirmation to the user.

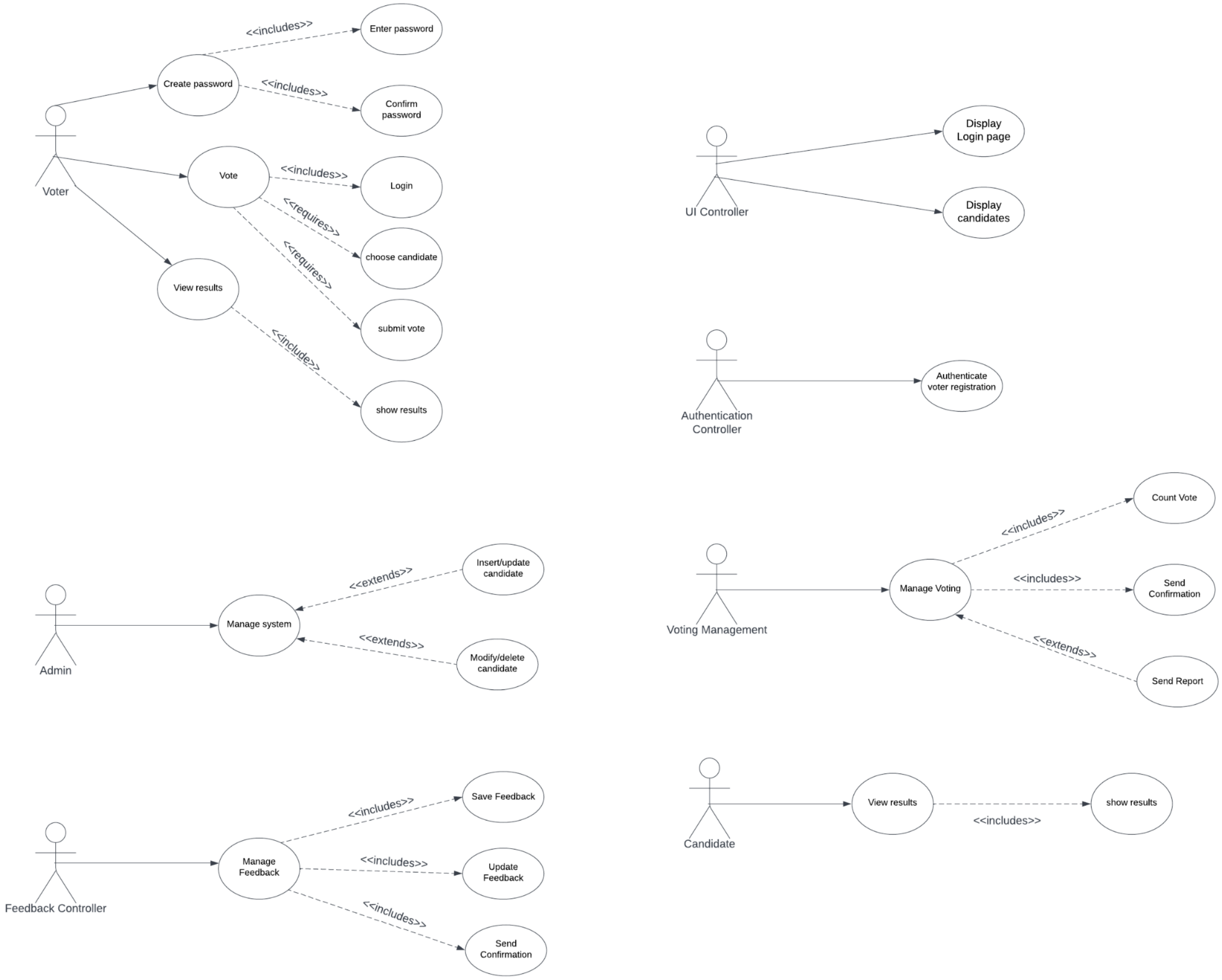

Figure 5: Use Case Diagram

This diagram includes the actors that will use the voting system, voter, admin, candidate, and controllers. It describes their functionalities or use cases. These use cases are based on the functional requirements specified earlier.

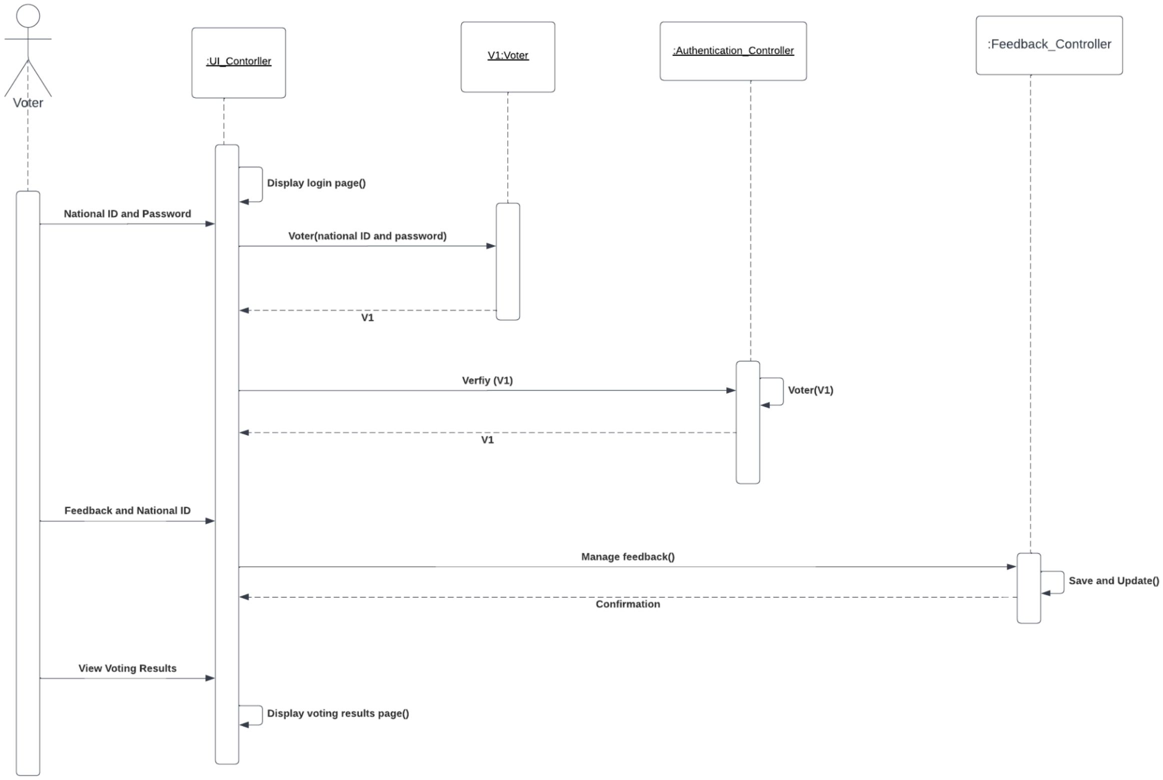

Figure 6: login sequence diagram

The sequence diagram explains how the system will work when the voter makes an action. First, the user will open the application, and the login page will be displayed. Then, he will enter the National ID and password, and the UI controller will pass this function to “v1 voter” which checks that this user is in the database and returns the results. After this action, the UI controller will pass the same function to the authentication controller, which verifies that the National ID matches the password for the same voter. Another action the voter may use is writing feedback using the national ID, the UI controller will send a “manage feedback” function to the feedback controller, which saves and uploads the feedback and sends a confirmation message. Finally, the voter can check the results, the UI controller will display the voting results page.

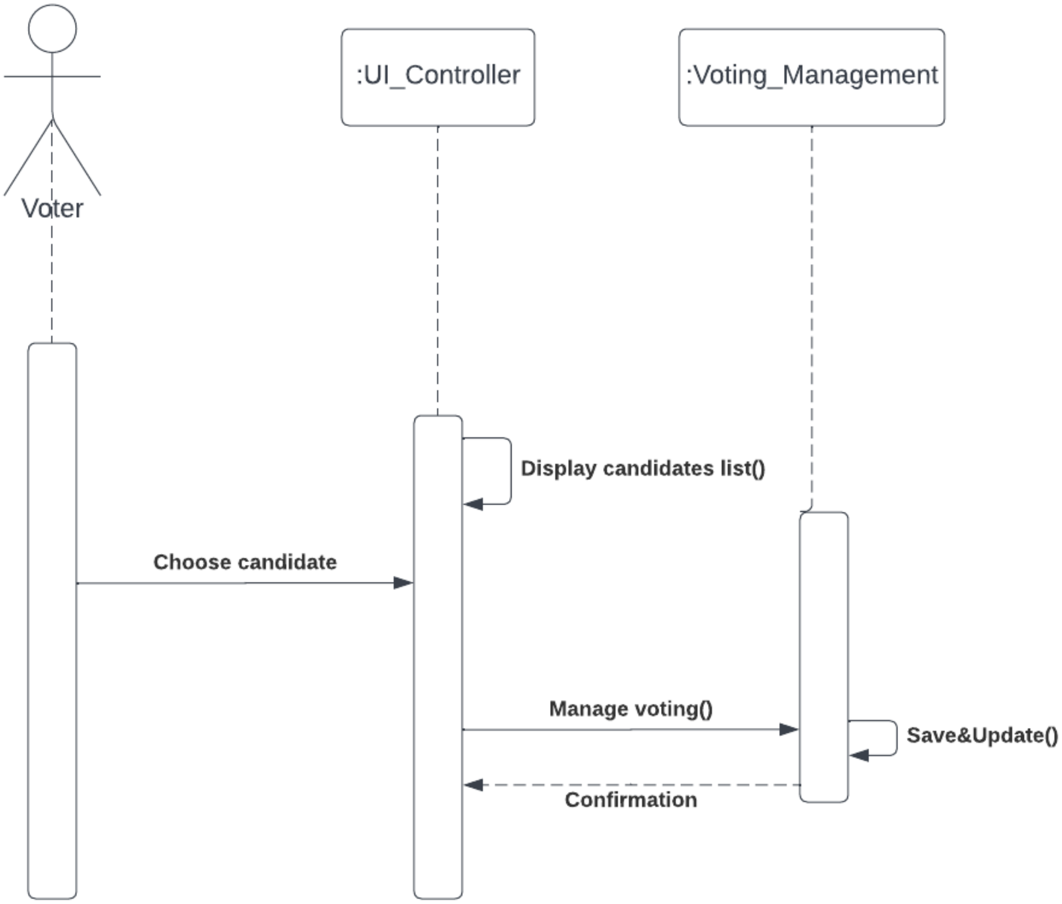

Figure 7: voting sequence diagram

This sequence diagram shows what happens to the system when the voter chooses a candidate. In the beginning, the UI controller will display the list of available candidates for the voter to choose one. After selecting a candidate, the UI controller passes this function (manage voting) to voting management, where data is saved and updated, then it returns a confirmation message to the voter.

The Refactored Source Code and Prototype:

Source code link:

https://drive.google.com/file/d/1pfA5Pu6_1LTuJbqqf762XbfftYEvF69p/view?usp=share_link

Prototype Video link (Shows sprint one and sprint 2 requirements achieved):

https://drive.google.com/file/d/1DQPHG2WjUuy14yjIi3R1HmYON0ISk8mc/view?usp=sharing

SQL Code Link:

https://drive.google.com/file/d/1RnX5f_AqQivZ7qjxQhJ_UvDpnsRw_sX7/view?usp=sharing

Test Cases (For all requirements in sprint 1 and 2):

Verify the login		
User entering invalid National ID	Entering letters or invalid characters	Error message: Invalid login National ID should only include numbers Try again
	National ID is not in the database	Error message: Invalid login Try again
User entering wrong password	Entering password that does not match the ID number	Error message: Invalid login Try again
User entering valid information	Entering the correct password and ID number	Redirecting to the next page
Customer support		
User pressing enter without writing anything	Not entering National ID and feedback	Message displayed and user will be asked to enter both feedback and national ID
User writing feedback and pressing enter	Not entering the national ID Just writing feedback	Message displayed asking the user to enter the national ID

User entering national ID and pressing enter	Not writing feedback Just entering the national ID	Message displayed asking the user to write feedback
User entering national ID and feedback and pressing enter	Both are entered	A message is displayed “feedback is sent”
User wants to re write feedback	Write feedback more than one time	User can only write one feedback per session
User entering invalid National ID and write feedback	Entering letters or invalid characters in national ID	Error message: Invalid login National ID should only include numbers Try again
	National ID is not in the database	Error message: Invalid login Try again
Results		
User press on show statistics	User want to the check voting process	Information about candidates and results are displayed
Voting process		
User stays more than 5 minutes on the voting page		Page automatically close
User voting for a candidate	Choosing only one candidate	A message displayed That the user successfully voted
The same user wants to re-vote	User will not be able to vote again	Message: “You already voted And can not vote again” Then the page will close

Plans for Future Expansion:

- Giving the voter access to cancel his/her voting after 5 mins submitting his/her voting.
- Giving the candidate the ability to vote for other candidates.
- For ease of navigation for the voters and candidates, the application will not close after each task/activity.
- After the voting period ends, a voting analysis would be conducted based on gender, age, economic status, city-based, etc., to determine any correlations or if there is anything unusual.
- To increase security, there would be a tracking analysis during the voting period to track if there are any unusual activities (e.g., a sudden rise in the vote count for a candidate)
- For easier use and access, we would implement Face ID for authentication and authorization access.