Election management system
8/4/2022

Assigned TA Name: **Dalia Elalfy**

Team ID:44

Team members:

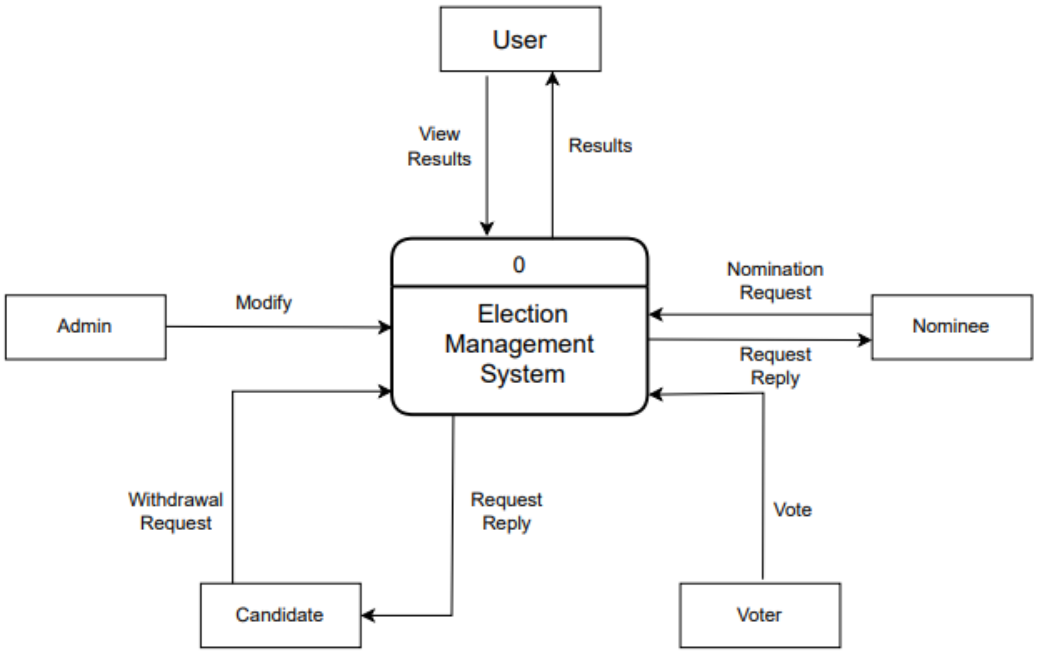
1. Team Members' name: **اسراء حسن محمد محمد**
   Seat numbers: 20191700095
   Department: Computer system
2. Team Members' name: **سهيلة سيد يوسف محمد**
   Seat numbers: 201917003006
   Department: Is
3. Team Members' name: **اسراء جابر هاني محمد**
   Seat numbers: 20191700094
   Department: Scientific computing
4. Team Members' name: **سمر نصر هاشم شهاب دياب**
   Seat numbers: 20191700296
   Department: Computer system
5. Team Members' name: **سميره يحيي خليل علي**
   Seat numbers: 20191700300
   Department: Cs
6. Team Members' name: **احمد رضا البحيري يوسف**
   Seat numbers: 20201700040
   Department: None
7. Team Members' name: **جابرييل مارك انطون مرقص**
   Seat numbers: 20191700831
   Department: None

SRS Document

1. Introduction

An election is a formal group decision-making process by which a population chooses an individual or multiple individuals to hold public office. Elections have been the usual mechanism by which modern representative democracy has operated since the 17th century. And our goal is to ease the collection and calculation of votes to deduct the winner of the election.

2. Context Diagram



3. User Requirements

The software has several users can be increased by signing up, and a number of candidates, each user can choose only one candidate and the candidate with the highest number of electors will win the election. The software has an admin who can modify, delete the candidates and the electors

4. Functional Requirements

1. *Login User:*

- Description: This function ensures the existence of the user and his type
- Inputs: (Username, Password)
- Source: User
- Pre-condition: Exist in the database of the software
- Post-condition: Will specify User's type and pass him to available functions
- Output: "Welcome Mr. *Username*"

2. *Vote for Candidate:*

- Description: This function finds the wanted candidate
- Inputs: (Candidate's name)
- Source: User
- Pre-condition: User should be logged in as a Voter
- Post-condition: Alter the number of votes in the system
- Output: "You've voted for *Candidate*"

- ***Calculate Votes(include):***

- Description: This takes the name from the VOTE function and adds one to the candidate vote counter
  - Inputs: (Candidates address in the system)
  - Source: VOTE function
  - Pre-condition: Executing Vote function correctly
  - Post-condition: Save the new average number of each vote
  - Output: None

3. *View Results:*

- Description: Give a percentage of each candidates' votes
- Inputs: (Choosing it from the menu)
- Source: User
- Pre-condition: Exist in the database of the software
- Post-condition: None
- Output: Display results

4. *Display Candidate:*

- Description: Displays each candidates info and his stats in the election
- Inputs: (Choosing it from the menu)
- Source: User
- Pre-condition: Exist in the database of the software
- Post-condition: None
- Output: Each candidates info and his stats in the election

5. Nominate as Candidate:

- Description: Add request to Admin
- Input: Personal information
- Source: User
- Pre-condition: None
- Post-condition: Either acceptance and transferred into candidates or rejected
- Output: “Thank you for submitting your info”

6. Check Request:

- Description: Parent of 2 functions of alteration
- Inputs: New data about the candidate
- Source: User
- Pre-condition: User should be logged in as an Admin
- Post-condition: None
- Output: “Request has been fulfilled”

- ***Add Candidate(extends):***
  - Description: Add new candidates
  - Inputs: New candidate’s data
  - Source: User
  - Pre-condition: User should be logged in as an Admin
  - Post-condition: Create new object of Candidate
  - Output: Candidate Added

- ***Remove Candidate(extends):***
  - Description: Delete Candidates
  - Inputs: (Choosing it from the menu)
  - Source: User
  - Pre-condition: User should be logged in as an Admin
  - Post-condition: Remove object of Candidate
  - Output: “Candidate removed”

7. Add Information:

- Description: Candidate can add a little description about themselves
- Inputs: Text from candidate
- Source: User
- Pre-condition: User should be logged in as a Candidate
- Post-condition: Add new string of description at candidate objects
- Output: “Info has been added”

8. Retreat Request:

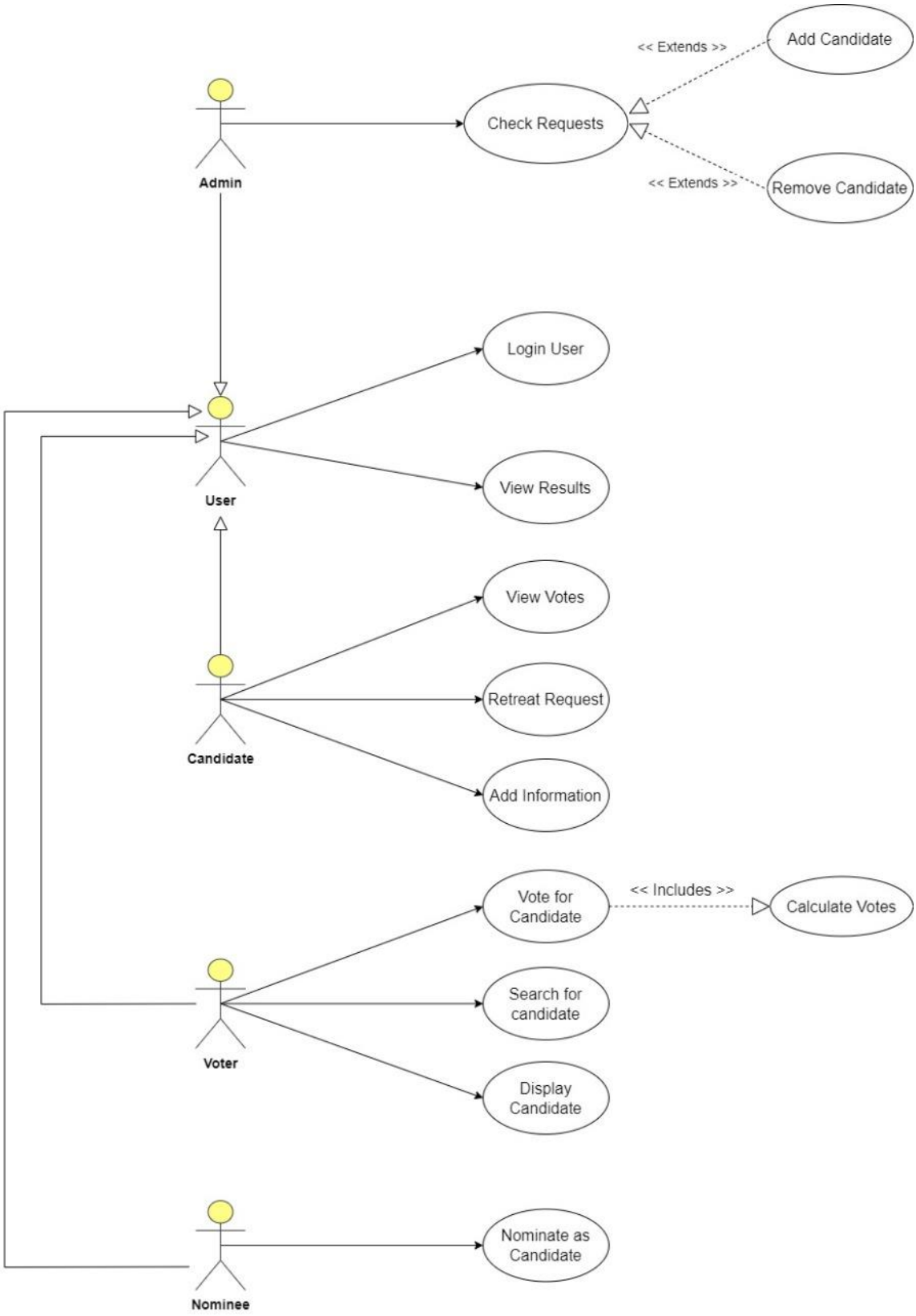
- Description: Withdraw
- Inputs: (Choosing it from the menu)
- Source: Users
- Pre-condition: User should be logged in as a candidate
- Post-condition: Remove object of Candidate
- Output: “Thank you for your Participation”

5. Non-Functional Requirements

1. **Security:**
   We established security and difference between each user type by making a log in page
2. **Ease of Use:**
   This app will establish ease of use that it will end training as an admin for 5 to 7 days
3. **Robustness:**
   In case of a failure or a crash the app will rework or execute again in an interval of 0 to 120 seconds
4. **Size:**
   This application will require a size of 200 mb of memory
5. **Cost effective:**
   This application will be cost effective that it will save man labor in counting and gathering of votes and will ease management where it will save papers and boxes where the votes will be submitted

6. Diagrams

1. Use Case diagram:



2. Sequence Diagram: