



Assignment 2

Software Engineering

Dr. Adrian Fiech

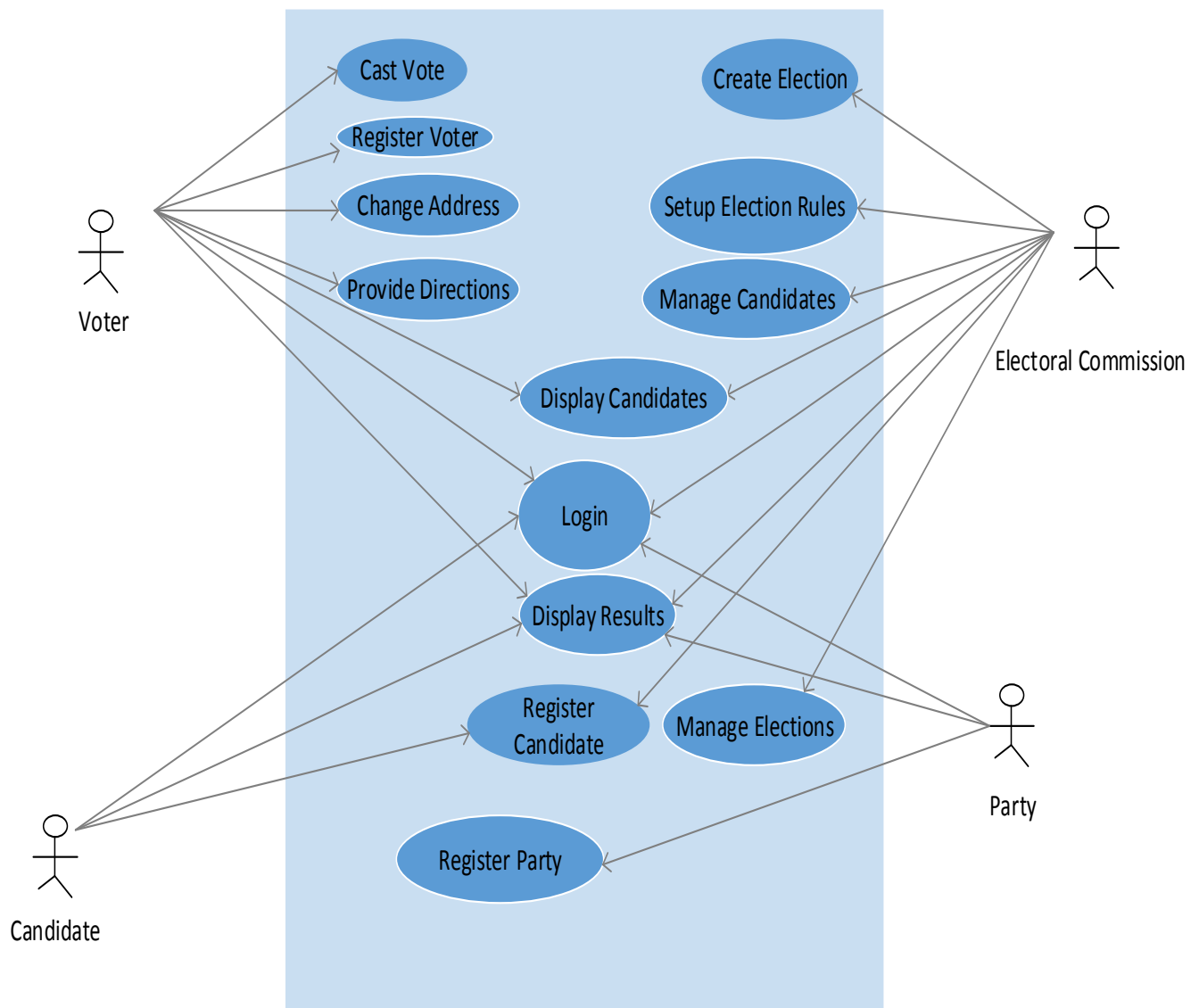
Group 1

Demarey Baker

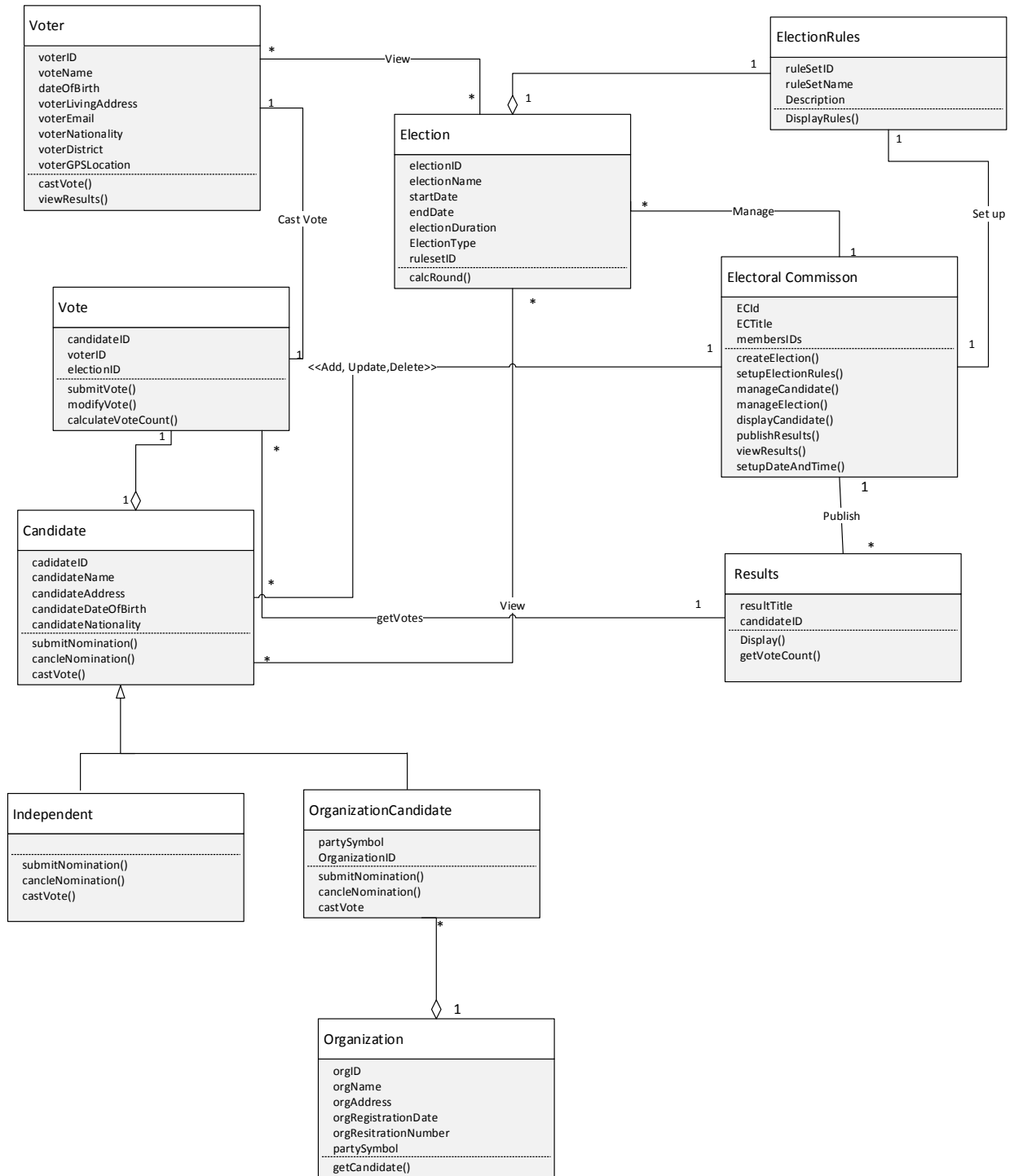
Hanieh Marvikhorasani

Keyvan Derakhshan Nik

*Use case diagram(revised version)*



## Online-Voting System \_ Class Diagram (revised version)



---

### *Description of three more Use Cases of online voting system*

---

**Use case name:** Register Voter

**Participating actor:** Voter

**Entry condition:** Voter is authenticated into the system.

**Flow of events:**

1. System shows a webpage to voter which he/she should provide requested information so can be registered as voter.
2. voter enters all requested information and submit it.
3. System validates voter information and sends verification email to voter.
4. voter verify the email by clicking on the link received in his/her email.
5. System adds the voter to the voter's list.

**Exit condition:** Voter is added to voter list.

#### **Alternative flow – Register Voter use case**

3a. System doesn't validate voter's information since the voter is already registered to the system.

1. System displays error message that your name is already on voter's list and you cannot register again.

3b. System doesn't validate voter's information since according to Election Rules voter is not eligible to cast vote.

1. System displays error message including "you are not eligible to vote for this election".

4a. Voter doesn't receive verification email.

1. Voter requests system to re-send verification email.

**Use case name:** Display Results

**Participating actor:** Voter, Candidates, Electoral Commission

**Entry condition:** Voter/Candidates/Electoral Commission is authenticated into the system.

**Flow of events:**

1. Voter/Candidates/ Electoral Commission asks system to show the results of the selected election.
2. System displays results.

**Exit condition:** Results are shown to Voter/ Candidates/Electoral Commission.

#### **Alternative flow – Display Results use case**

2a. System cannot display results.

1. System displays an error message which the results cannot be displayed at the moment and they should try again later.

---

**Use case name:** Set up election rules

**Participating actor:** Electoral Commission

**Entry condition:** Electoral Commission is authenticated into the system.

**Flow of events:**

1. Electoral commission selects an election (which is created before) that rules are going to be set up for that.
2. System shows a list of elections which were created before.
3. Electoral Commission selects an election and performs create election rules.
4. System displays a webpage which different kinds of rules can be defined there.
5. Electoral commission enters all the rules (rules associated with Voter, Candidates and election) and submits it.

**Exit condition:** Rules are recorded into the system as a rule set for chosen selection.

#### **Alternative flow – Set up election rules use case**

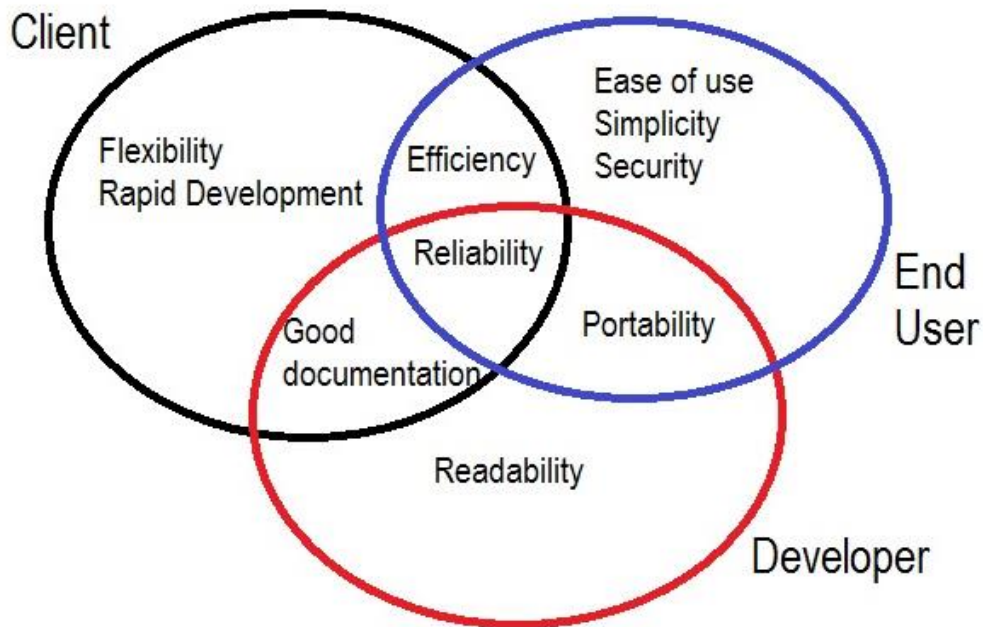
3a. Electoral commission wants to modify or remove or add some more rules to the rule set which was created before.

1. Electoral commission performs modify rules.
2. System shows a webpage including all the rules Electoral Commission can modify rules, remove them or add more rules.

---

### *Design Goals*

---



**Simplicity, ease of use:** Since most part of end users of our system are ordinary people including different ranges of age and education, Online-voting system has to be easy to work with and simple so users (Voter, Candidates and electoral commission) prefer to participate in voting online instead of doing manually. Our system will be simple and organized with every detail mentioned clearly. For example: Users can easily register as voter or candidate with clear instructions and simple steps.

**Reliability:** Our system will be reliable and we will make sure required services will be available all the time.

**Flexibility:** The system has to be flexible to the possible future change requests from client.

**Efficiency:** when it comes to voting time (since voting will be done in specific date and time), many voters want to cast their votes at a specific time. Our system has to be efficient to handle the load of requests simultaneously

**Good documentation:** Our system contains proper documentation; every requirement, design goals and implementation details are well documented.

**Security:** Since we will store sensitive data (voters and candidates' private data) into our repositories, it is important for them to be sure that their information is securely stored and their privacy will be preserved.

**Portability:** Given the fact that users of the system use different devices with different platforms such as smartphones, tablets, laptops, and etc., the system has to be extremely portable. It should support different browsers and be responsive (should be adaptive to any screen size).

**readability** Given the fact that users of the system use different devices with different platforms such as smartphones, tablets, laptops, and etc., the system has to be extremely portable.

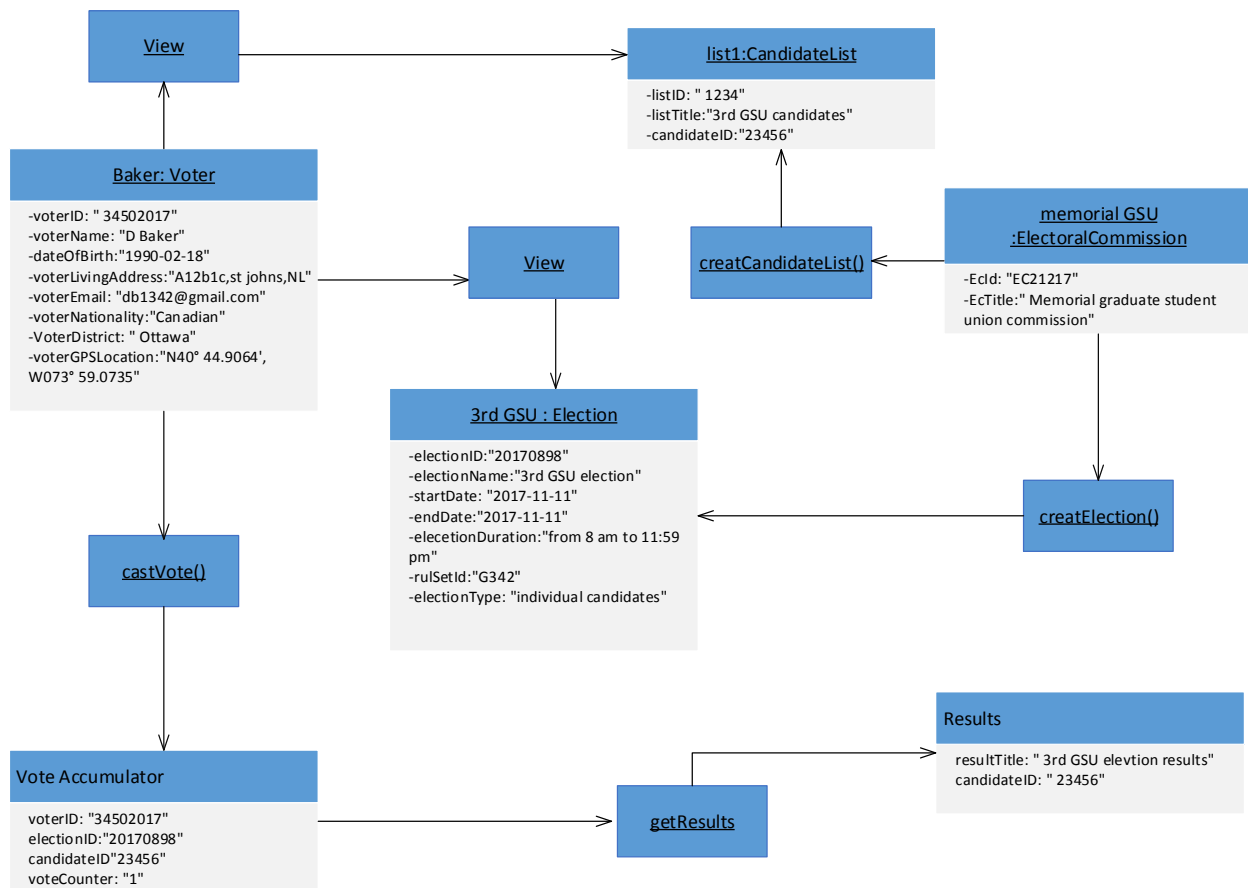
**Rapid development:** Our system is following rapid development approach due to limited time.

---

## *(Essential) Object Modeling*

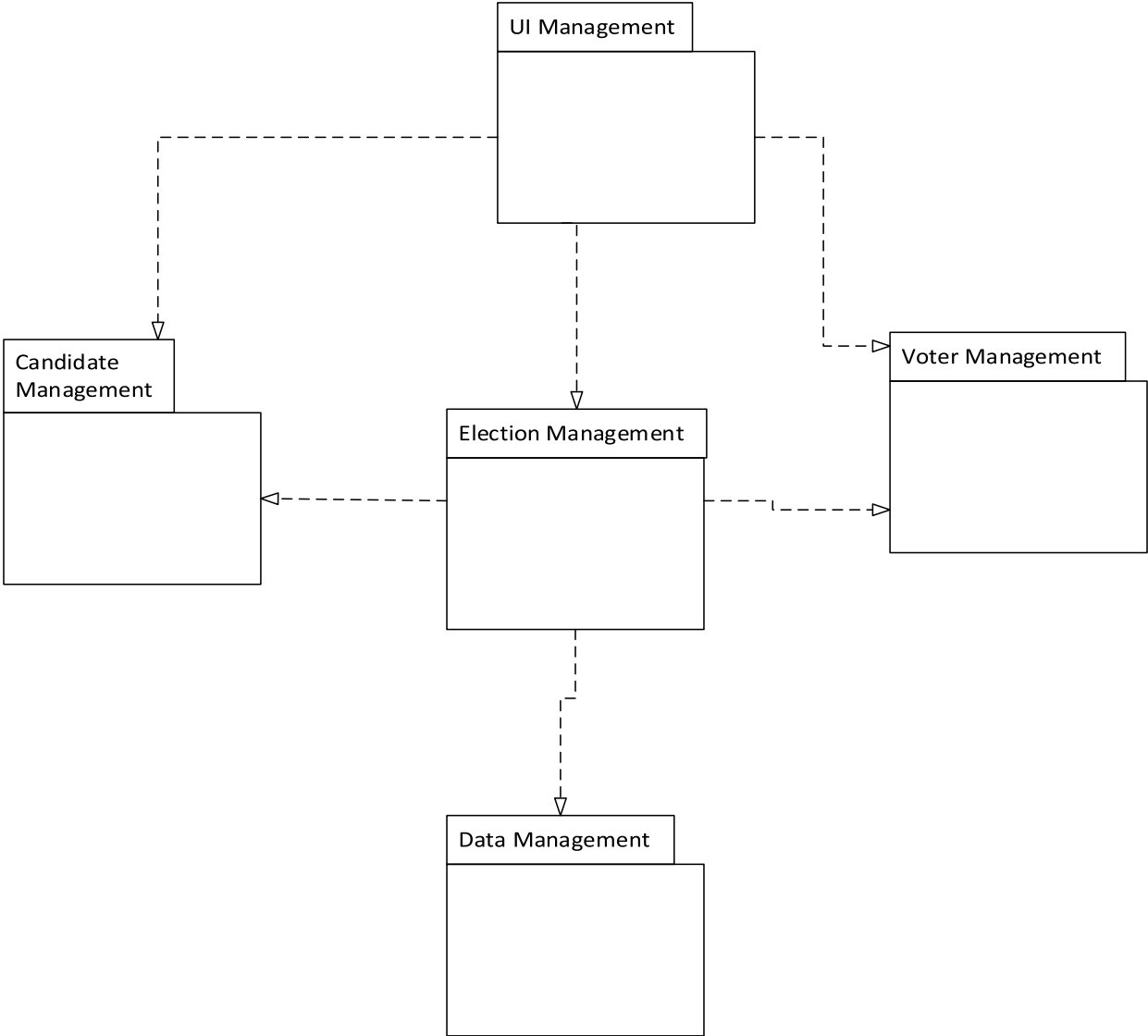
---

Following diagram displays a snapshot of our system when Electoral Commission created an event and a voter cast vote.





*System Decomposition*



**User interface management subsystem:** This subsystem contains the user interface of our whole system. On this subsystem Voters, candidates and electoral commission can view and modify information based on their access levels. For example, they can see ongoing elections, results, candidate list and etc.

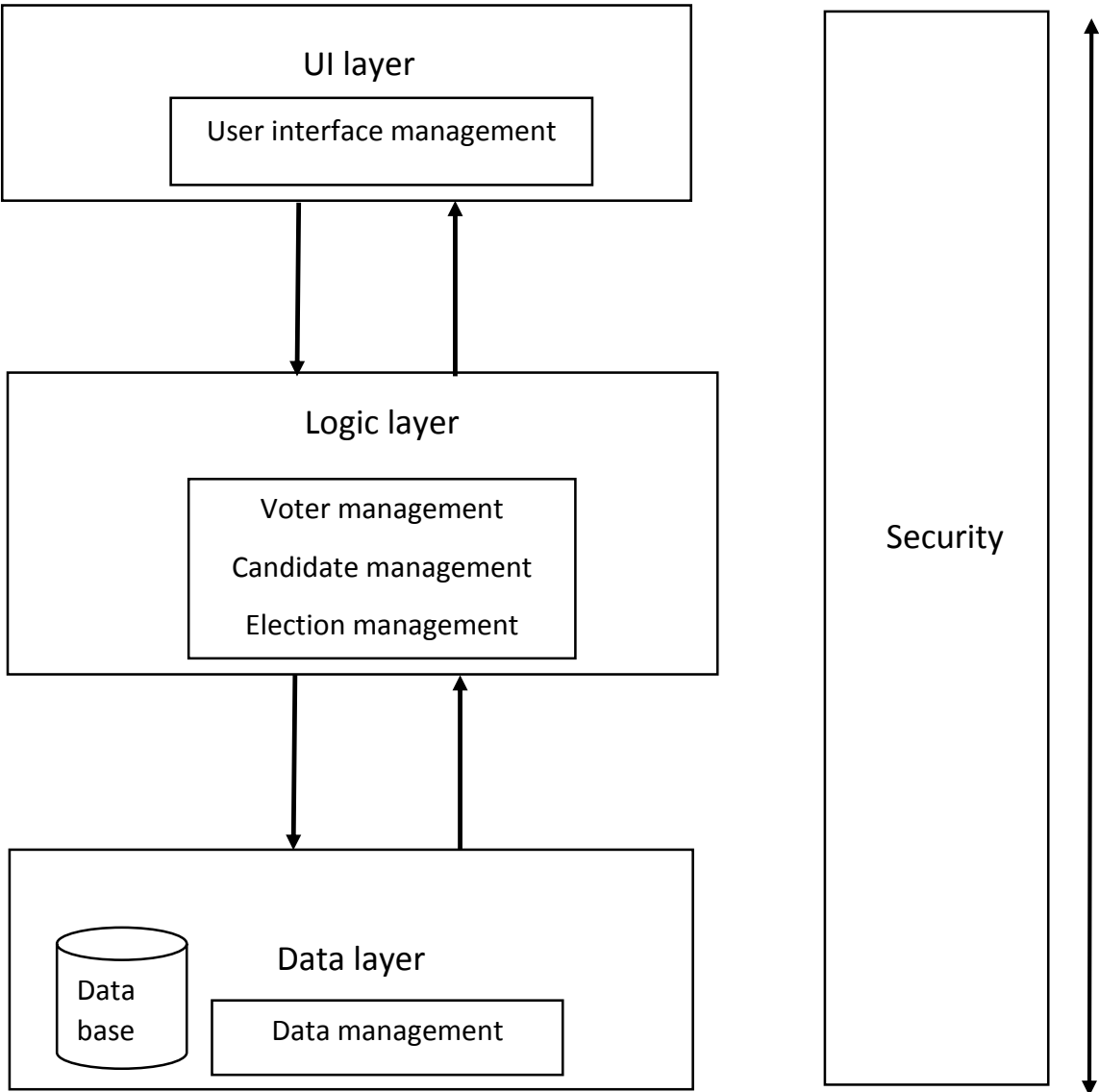
**Voter management subsystem:** In this subsystem, user can register as voter. User should provide their personal information for being registered in the system and being added to voters list. Also voter can cast vote, modify vote and see the result of election. Electoral commission also can manage voters like add or modify a voter in this subsystem. This subsystem is connected to UI subsystem and Data management subsystem. information is stored on Data management subsystem and displayed by UI subsystem.

**Candidate management subsystem:** In this subsystem candidates can register for the election. They should provide and submit required information in the subsystem. Electoral commission can check eligibility of the candidate in this subsystem and add them to candidate list. Candidates also can see all the information about ongoing election and results in this subsystem. This subsystem is connected to UI subsystem and Data management subsystem. information is stored on Data management subsystem and displayed by UI subsystem.

**Election management subsystem:** Electoral commission has only access to this subsystem. Electoral commission can create or modify election, create election rules and do all the things which is related to managing the election. This subsystem is closely working with Voter management subsystem and Candidate management subsystem. This subsystem is connected to UI subsystem and Data management subsystem. information is stored on Data management subsystem and displayed by UI subsystem.

**Data management subsystem:** This subsystem is controlling how data is being managed and stored in our system. The database is its main component. Three of our subsystems (Voter management, Election management and candidate management) are directly connected with this subsystem and interacts with its functionalities. For example, when a voter wants to register in our system, Voter management subsystem will interact with this subsystem to check if the user is registered before or not.

*Logical Architecture*



## Three-layer architecture:

For the existing system would be beneficial in finding a solution which fits our requirements.

The layers are explained below:

UI layer: The Web Browser implements the user interface.

Logic layer: The Web Server serves requests from the web browser.

Data layer: The Database manages and provides access to the persistent data.

Link to our prototype:

<http://www.cs.mun.ca/~dabaker/>

link to our Github:

<https://github.com/silvamari/OVS>