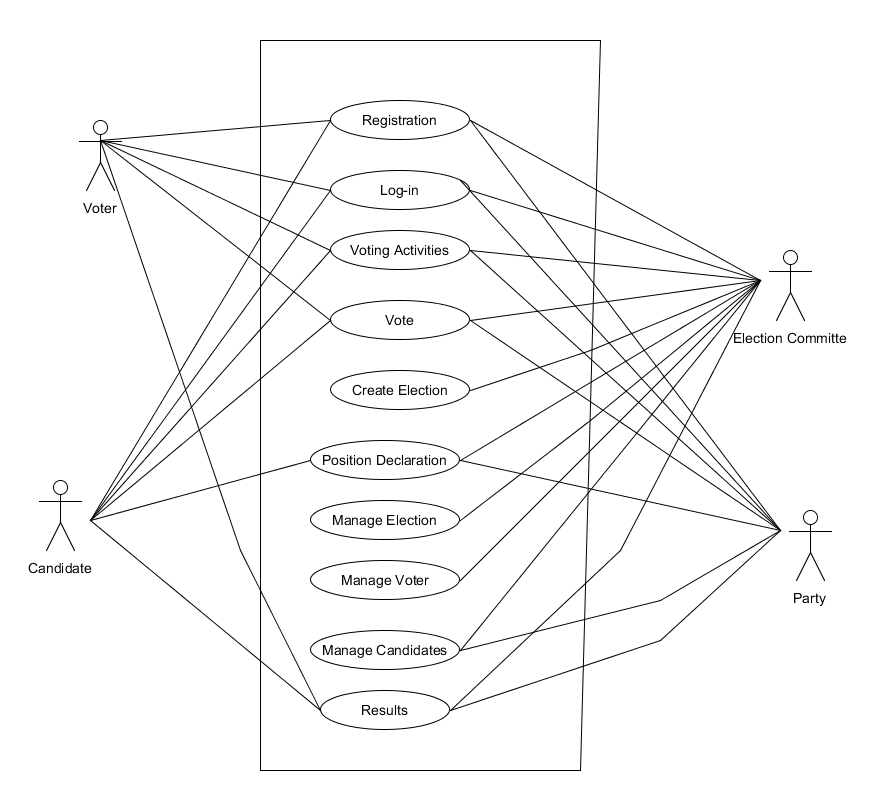
Assignment 1

Group 2

1. The online voting system has been developed for the people to be able to vote online, whether it be a parliamentary or student union elections.
2. The election committee decides to create the election, initialize the name and checks if they meet the eligibility criteria for the specific election.
3. The candidate must register himself with the election committee and declare his eligibility and position: presidential or district.
4. The Election Commission generates a revised list of candidates up for voting based on the rules of the particular elections.
5. The election commission also decides on the duration and directions of the election.
6. The voters must go online and register himself for the election.
7. To register the voters must meet the eligibility criteria set up for him by Election Commission.
8. The voter can log in to his account and fetch all the details regarding the elections, i.e. the list of candidates, directions, duration of election and rounds of elections.
9. Finally, the votes are counted and the committee shall declare a winner. The declared results can be seen by the voter, election committee and candidate.

**Use Case Diagram For the Online Voting System:**



**Detailed description of three use cases:**

|  |  |
| --- | --- |
| **Name of Use Case:** | **Creating Elections** |
| Participating Actors | Election committee |
| Entry Conditions: | Following last year's minutes and the Committee decides about for holding elections |
| Flow of Events | 1.Once the date of elections fixed, the Candidates submit their nomination  2. The Candidate should update their details and check the voter list |
| Exit Conditions | The Election committees should have initial list of the candidates |
| Alternate flow | 1a. No candidate submit their nomination  1a1. The elections will be reschedule  2a. The minimum number of Candidates are not enrolled  2a1. The election will be rescheduled |

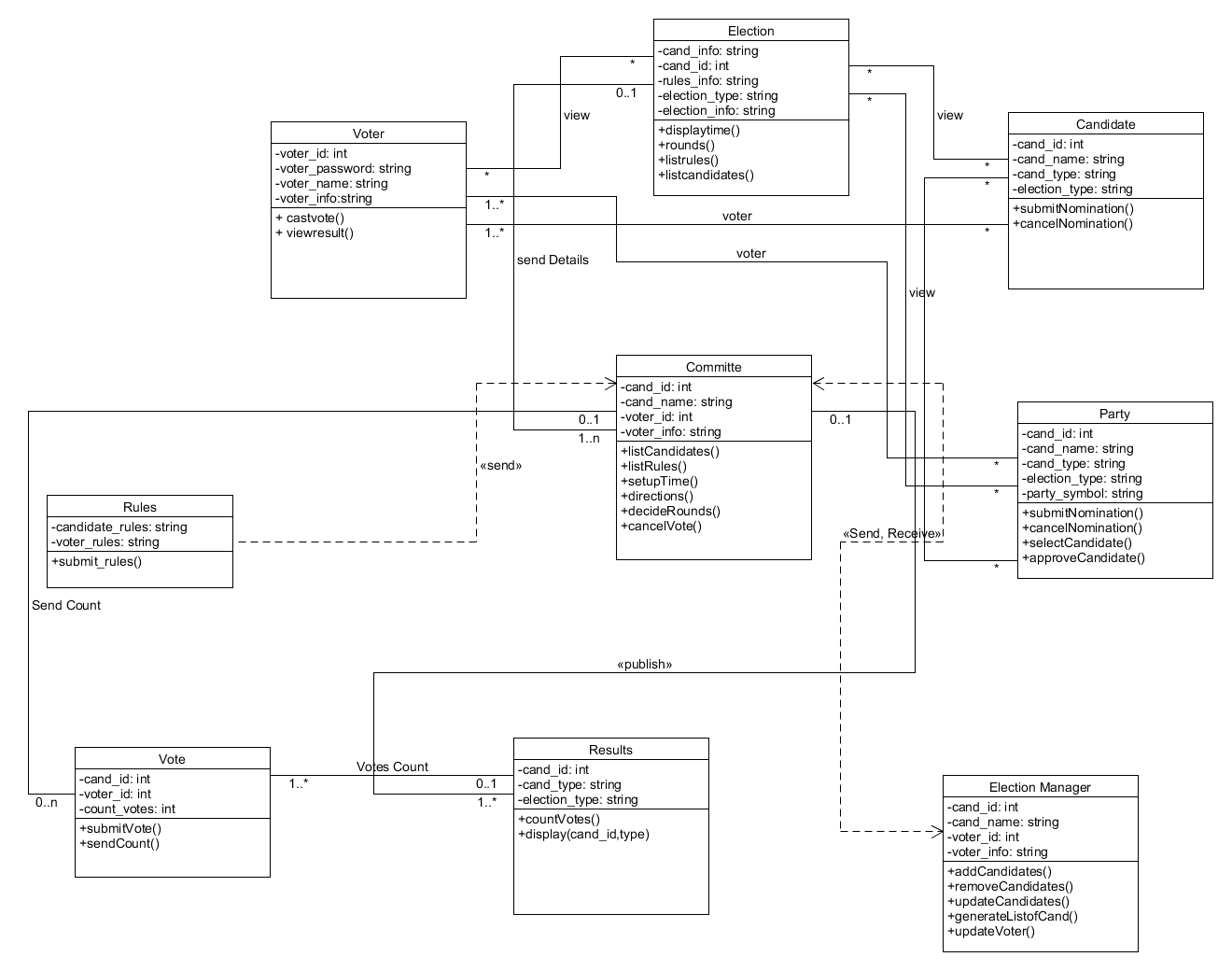
|  |  |
| --- | --- |
| **Name of Use Case:** | **Managing Elections** |
| Participating Actors | Election Committee |
| Entry Conditions: | The committee has the initial list of candidates |
| Flow of Events | 1.The Election committee sets up the rules required for the candidates to follow.  2.The Election committee sets up the rules for the voters.  3. The election committee processes the initial list of candidates based on rules.  4.The Election committee generates revised list of candidates. |
| Exit Conditions | The election committee should have a revised list of candidates |
| Alternate flow | 3a. Candidate withdraws from election.  3a1. The election committee reschedules the election.  4a. The minimum number of candidates aren’t processed.  4a1. The election committee reschedules the election. |

|  |  |
| --- | --- |
| **Name of Use Case:** | **Voting Activity** |
| **Participating Actors** | Voter, Election Committee, Candidate, Party |
| Entry Conditions: | Election Committee declares the activity. |
| Flow of Events | 1. The system prompts the voter to give the vote.  2. The voter chooses the candidate from candidate list.  3. The system sends a message to voter indicating that the vote is polled.  4. The server automatically deactivates the ballot number. |
| Exit Conditions | 1.Provides time frame for voting  2. Providing updation of voter’s information  3. The EC is asked for their(voters) correct information. |
| Alternate flow | 1a. The voter fails to select a candidate,then  1a1. The system will prompt with message “invalid vote” |

**Summary of Use cases:**

|  |  |
| --- | --- |
| Registration | The voters will provide their details. |
| Login | Once the user is registered, as per the user login authentication, users will be given specific rights to the system |
| Vote | Every registered voter should be able to vote. The system should accept only one vote per voter. |
| Manage Candidate | Candidates can be added, updated or removed.. It also lets the candidates view or update their details. |
| Position Declaration | Candidates apply for particular position (independent or party) and the election commission finalize the selective candidate. |
| Results | Once the voting process has been completed , tally all the casting vote and publish the results. |

**Domain Modelling for Online Voting System:**



Assignment 2

Group 2

**Description of use cases:**

|  |  |
| --- | --- |
| **Name of Use Case:** | **Results** |
| Participating Actors | **Election Manager** |
| Entry Conditions: | The Election Manager is logged into the System. |
| Flow of Events | 1. The Election Manager selects an election.  2. The System presents the casted votes to the Election Manager for the election. |
| Exit Conditions | The votes are casted within the time frame. |
| Alternate flow | None. The System's state is not changed. |

|  |  |
| --- | --- |
| **Name of Use Case:** | **Position Declaration** |
| Participating Actors | **Candidates** |
| Entry Conditions: | The party needs to nominate the Candidate. |
| Flow of Events | 1. The Candidates will apply to election committee for accepting their candidacy .  2. The System will represent the eligibility criterias for candidacy. |
| Exit Conditions | The Candidate names will be declared . |
| Alternate flow | 1a. No Candidate is available from the party.  1a.1 The system will prompt up with the message “No suitable Candidates” |

|  |  |
| --- | --- |
| **Name of Use Case:** | **Manage Candidate** |
| Participating Actors | **Election Committee** |
| Entry Conditions: | The Candidates are selected by the party and election committee . |
| Flow of Events | 1. The election committee assigns symbol for the Candidates.  2. The voter see the symbol and Candidates name. |
| Exit Conditions | The election committee is ready for the election from the Candidates side. |
| Alternate flow | 1a. any Candidate is assigned by the party.  1a.1 The system will prompt up with the message “Independent Candidates” |

|  |  |
| --- | --- |
| **Name of Use Case:** | **Vote** |
| Participating Actors | **Voter** |
| Entry Conditions: | Voter casts their vote in the election. |
| Flow of Events | 1. The Voter is presented with the appropriate ID that was created in Registration.  2. The Voter enters their vote to the System.  3. The System asks the Voter to confirm that they want to submit their Vote.  4. The Voter confirms the submission.  5. The System records the vote. |
| Exit Conditions | The Voter's vote is recorded in the System and record that the Voter has voted in this election. |
| Alternate flow | 1. The Voter exits the System. Nothing is recorded by the System  2. The Voter does not confirm the submission.  The Voter may make changes to their ballot or exit the system. |

|  |  |
| --- | --- |
| **Name of Use Case:** | **Login** |
| Participating Actors | Voter, Election Manager (EM) |
| Flow of Events | 1. Voter and Election Officer enters their login information. 2. The System verifies their login information. 3. Once the voter has registered, they can access the system, he/she can choose a candidate from the dropdown list available in the site according to their preference. 4. The Election Officer is granted access to administrative functions in their domain. |
| Entry Condition | Voter and Election Manager (EM) should have registered to become a member of the system. |
| Exit Condition | Voter should have a successful vote and Election Officer has access to administrative functions in their domain. |

**Design Goals:**

|  |  |
| --- | --- |
| **Reliability** | Our main design goal is, the system will be reliable and should be work as we have desired. |
| **Usability** | The system should be user-friendly and with a basic computer skill user could access the system. |
| **Availability** | Our system is hosted on server. We guarantee that all functionality is available 100% of the time in a given period, excluding special planned maintenance. |
| **Portability** | It is portable to any platforms as we have used HTML, CSS, JavaScript and node.js to build our online voting system. |
| **Performance** | To handle server side, we have used node.js for our system. Again, for database we have used MongoDB for storing data. The both technologies are update and perfect match with each other. So, data transfer rate is much higher because of their mutual connectivity.  The data retrieval rate from the database is less than two minutes as data is organised in the database in efficient way. The system will perform in efficient manner with responsive and stable condition. Before it is deployed, the system will be tested thoroughly for defects and bugs. |
| **Scalability** | The whole system architecture has used model view controller(MVC). So, there will be no effect as the number of users being increased with respect to time. The webpage directly been connected with the database so the it will help the data traffic as well. |
| **Security** | Simply Voting takes secrecy of the vote very seriously. All voter information is removed from our servers if we choose to have the election deleted. We never make use of voter information for anything other than voting and never share such information with third parties. Security testing will be done to make sure the system is not vulnerable to any attacks. |

**Entity Objects:**

* Voter
* Election
* Candidates name
* Party

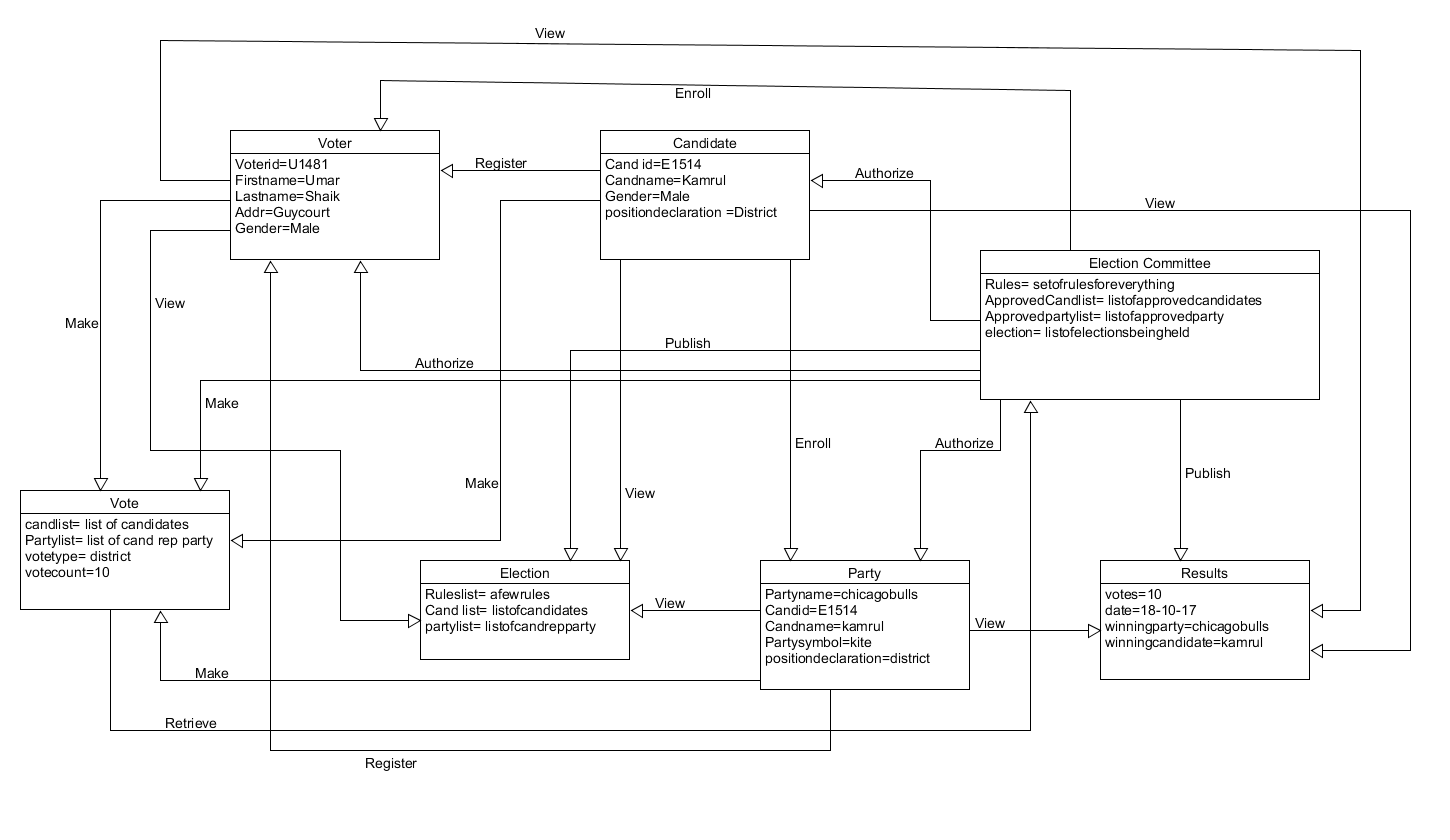
**Boundary Objects:**

* Vote
* Results

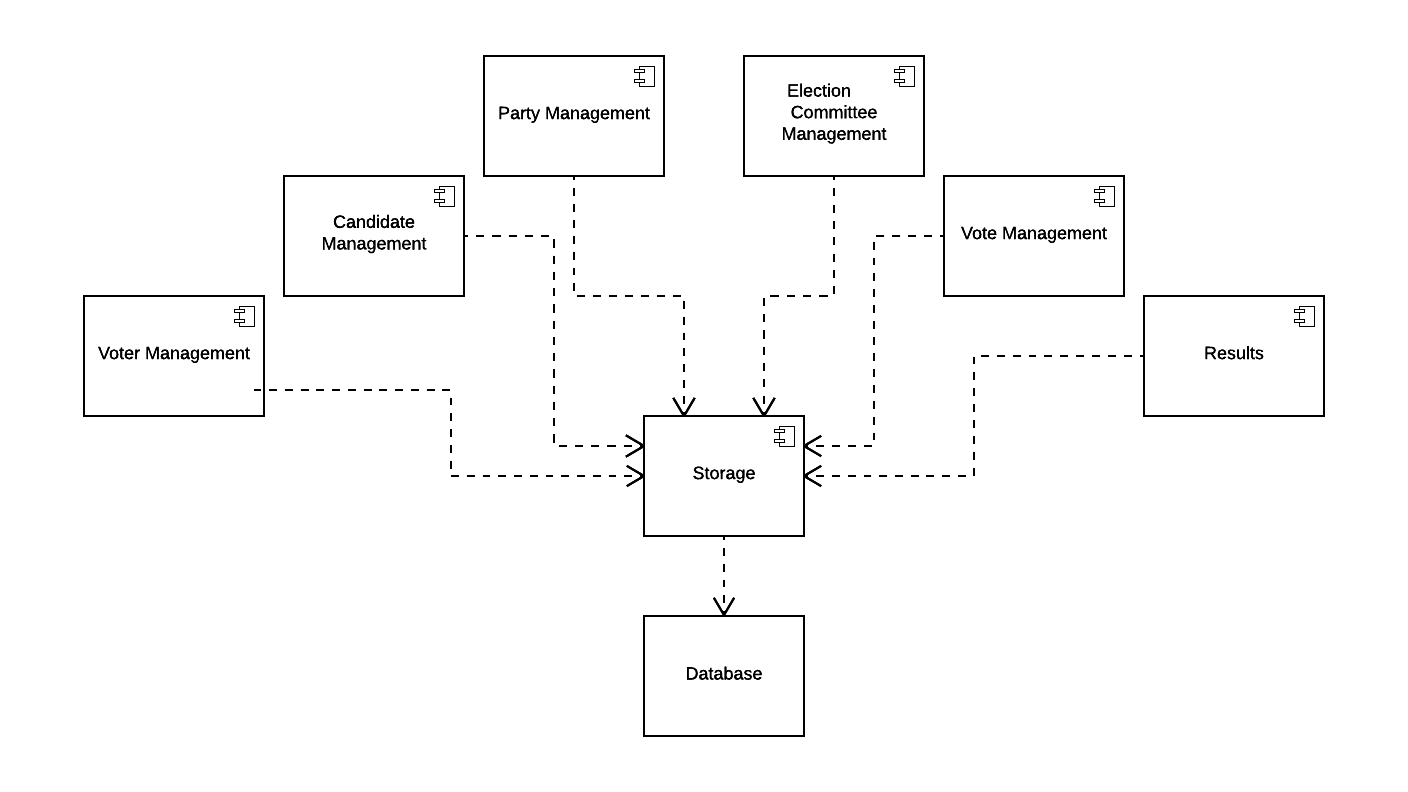
**Control Objects:**

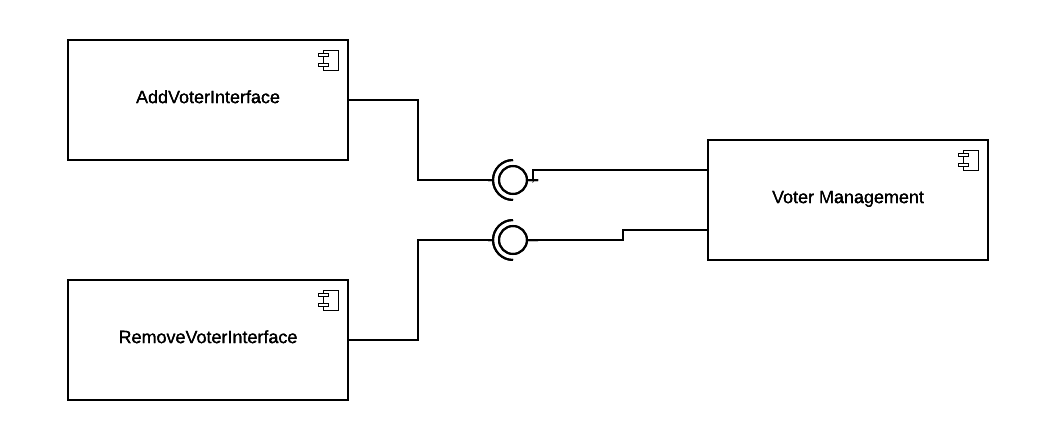
* Committee
* Rules
* Election Manager

**Object modelling:**

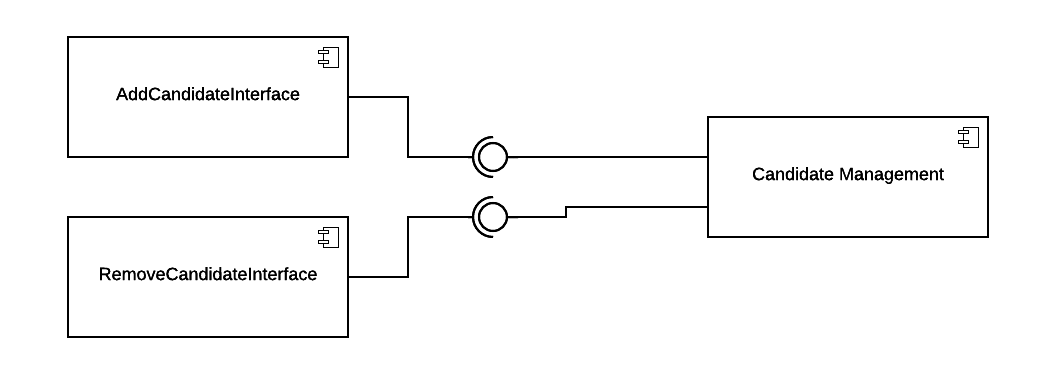


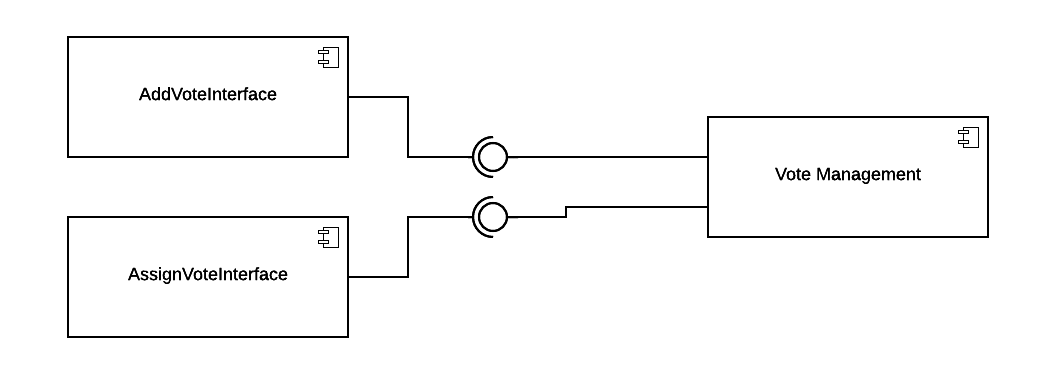
**System Decomposition:**

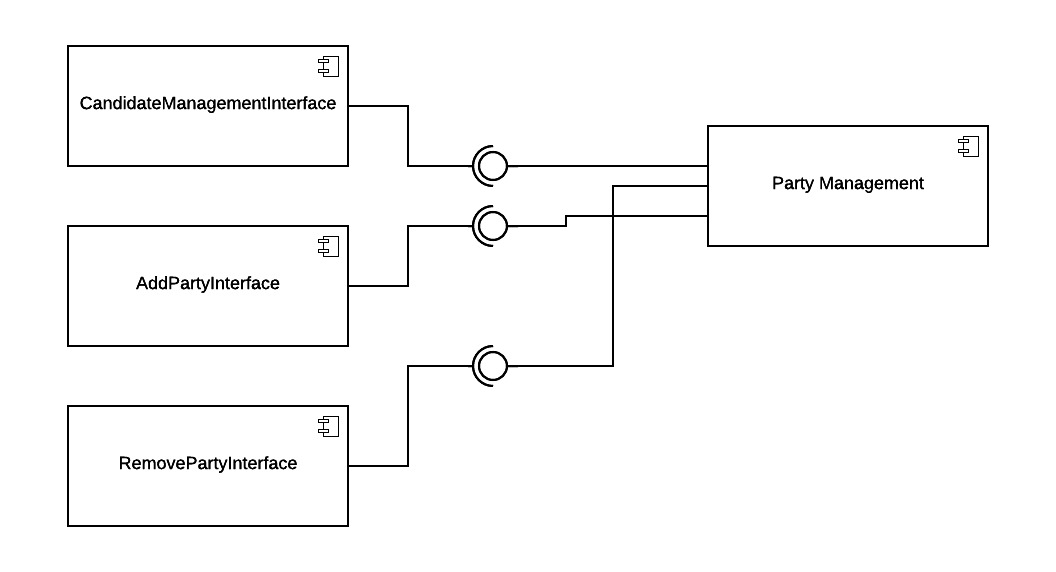


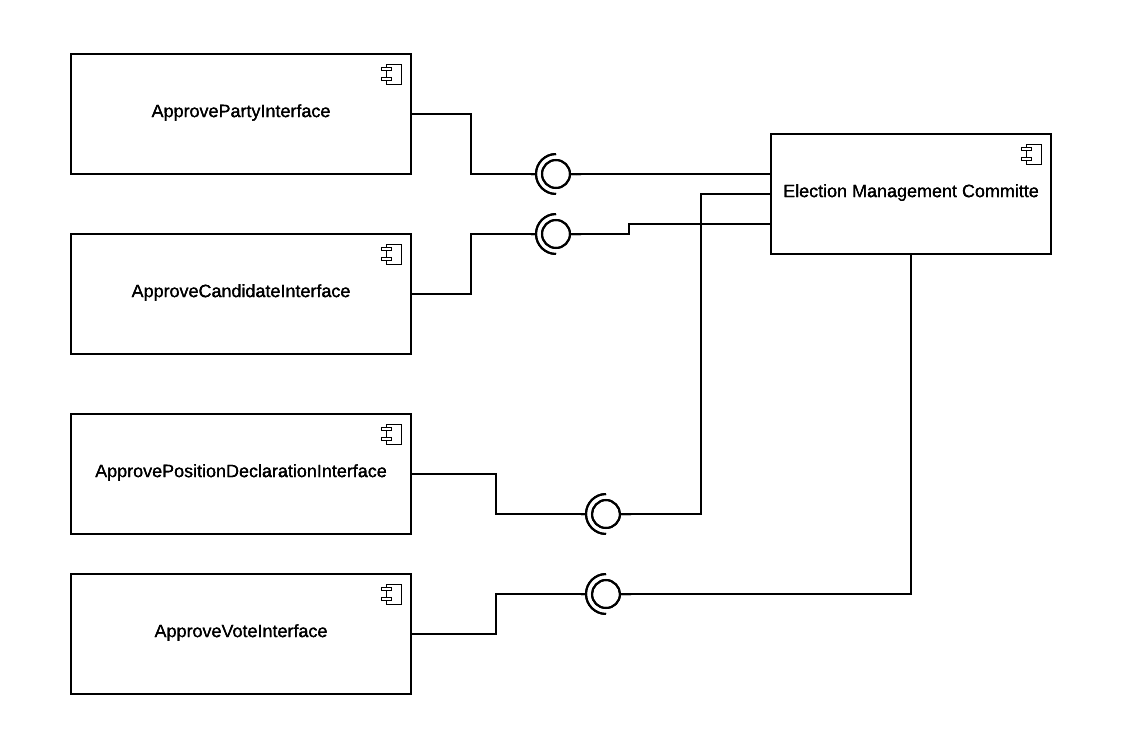


The vote management system provides services to the other modules. Add and Remove voter interfaces require services from vote management to add and remove voter

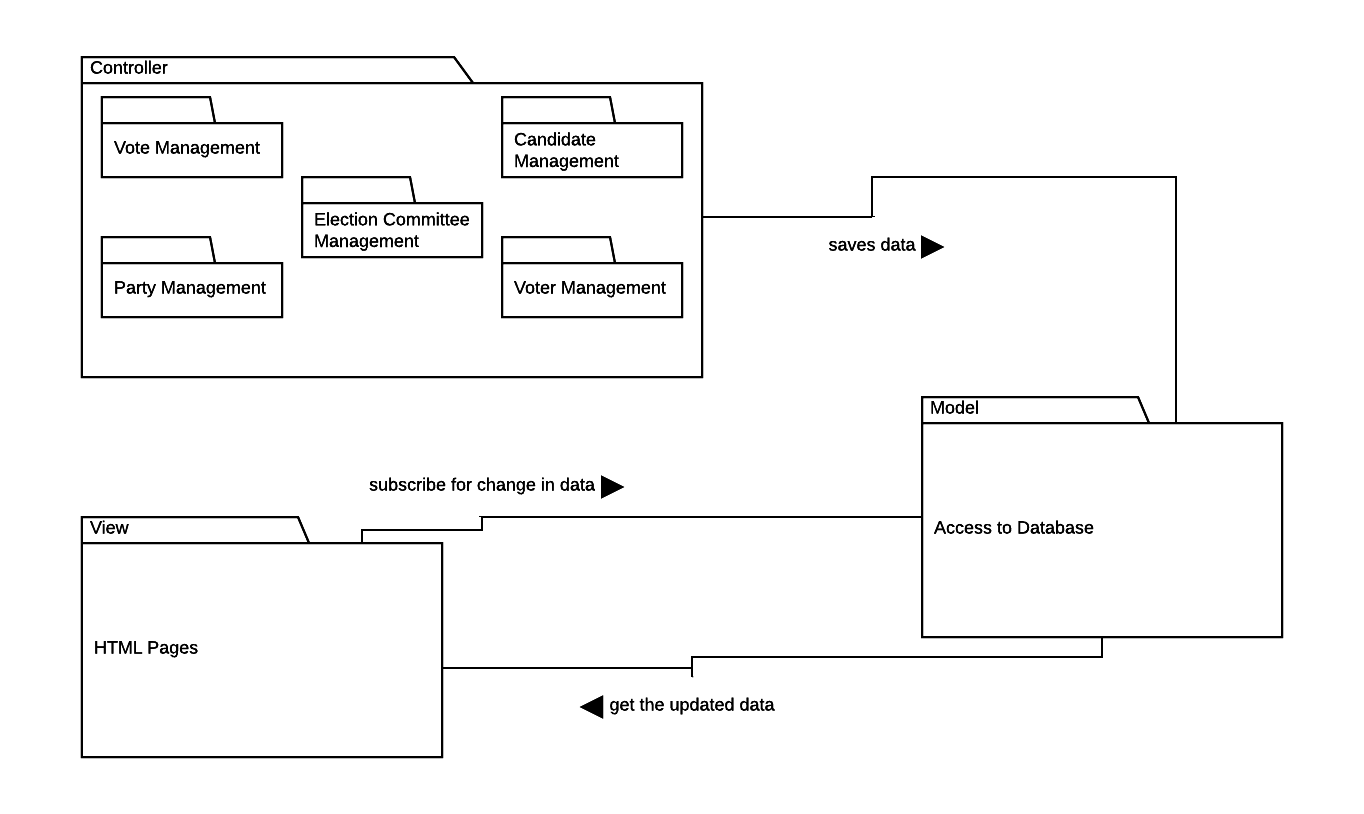








**Logical Architecture: Model View Controller(MVC)**



For Online Voting System, we are going to implement “Model View Controller” System.

Its much more maintainable if the control, data and presentation is separated. It also makes stuff easy for the programmer to comprehend and make changes.

Model: Handles connection with database

View: All HTML will be part of the view

Controller: node.js