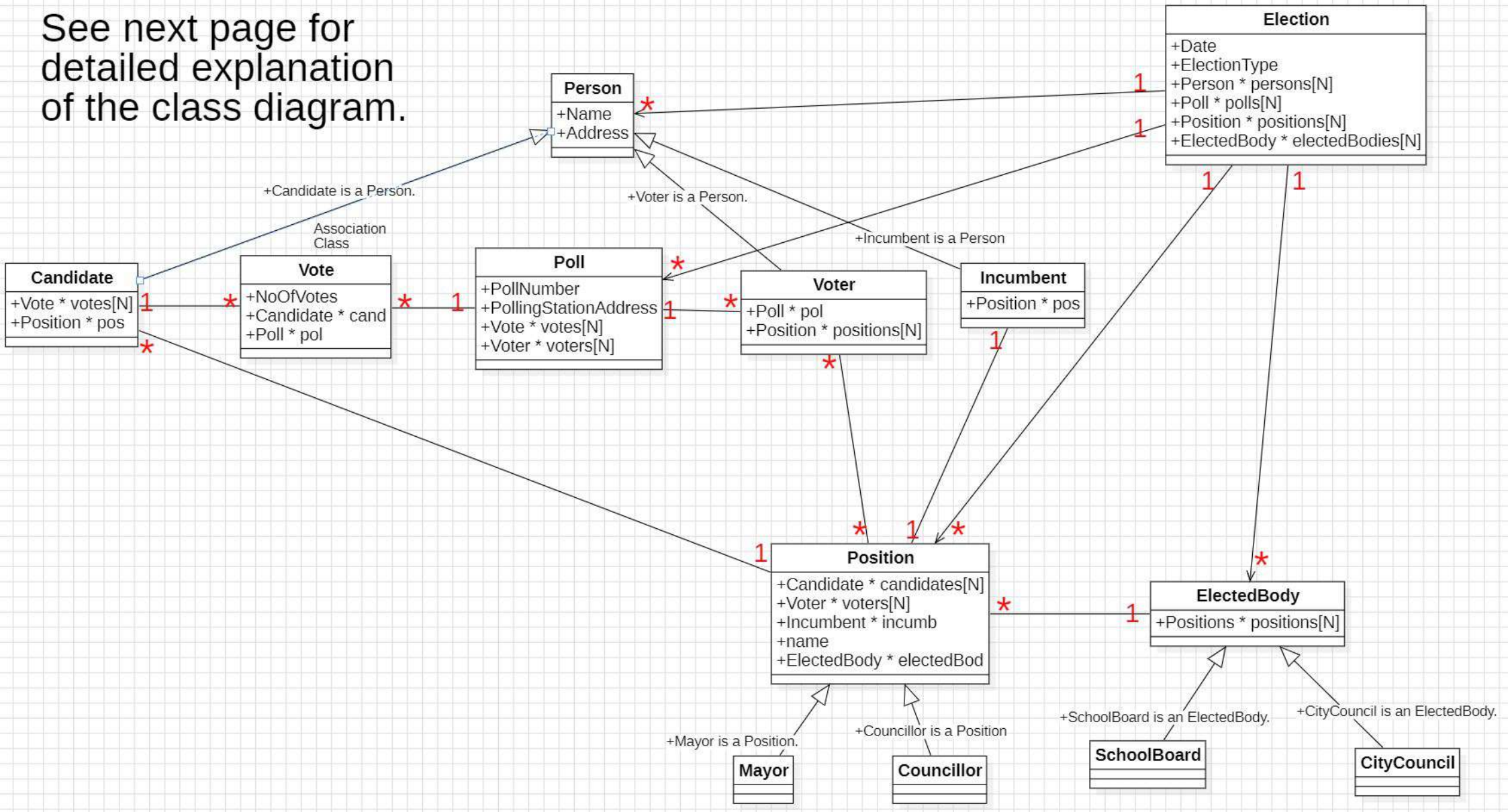
See next page for detailed explanation of the class diagram.



**Detailed explanation of the class diagram:**

**Statement 1:** The system will manage elections for a variety of different elected bodies (e.g. school boards, city councils, etc.).

**Solution:** There is a Parent class named 'ElectedBody' whose children are 'SchoolBoard' and 'CityCouncil'.

**Statement 2:** Each elected body can have various positions (also called seats, e.g. mayor, councillor, etc.).

**Solution:** There is a Parent class named 'Position' whose children are 'Mayor' and 'Councillor'. The 'ElectedBody' class has a one-to-many relation with the 'Position' class.

**Statement 3:** Elections are scheduled for a specific date, and usually several (or all) positions are voted on together; however, sometimes there may be the need for a by-election (e.g. to elect a particular councillor because the incumbent – the previous person who held the position – has resigned).

**Solution:** There is a class named 'Election' which has an attribute of 'Date' and an attribute of 'ElectionType' which indicates whether it's a normal election or a by-election.

**Statement 4**: The system will keep track of candidates for each seat.

**Solution:** There is a class named 'Candidate'. The 'Position' class has a one-to-many relation with the 'Candidate' class. The 'Position' class has an array of 'Candidate' pointers which we can use to track all candidates competing for that position/seat.

**Statement 5:** The system will also record who is the incumbent for a seat, since newspaper reporters are interested in reporting whether incumbents have won again or lost.

**Solution:** There is a class named 'Incumbent' which is the child of the 'Person' class. There is a one-to-one relation between 'Incumbent' and 'Position'. The 'Position' class has a pointer to its incumbent which can be used to tell whose is currently occupying that position/seat.

**Statement 6:** The system records the name and address of each candidate and incumbent.

**Solution:** There is a class named ‘Person’ with attributes ‘Name’ and ‘Address’. The ‘Candidate’ class and the ‘Incumbent’ class are children of the ‘Person’ class.

**Statement 7:** The system will also keep track of the list of eligible voters.

**Solution:** There a class named ‘Voter’ who is a child of ‘Person’ class. There is a one-to-many relation between ‘Election’ class and ‘Person’ class. The ‘Election’ class has an array of ‘Person’ pointers which can be used to keep track of all the persons(voters).

**Statement 8:** Each voter can only vote for certain positions (e.g. a particular council seat that represents their area).

**Solution:** There is a many-to-many relation between ‘Voter’ class and ‘Position’ class. One voter can vote for multiple positions and one position can be voted by multiple voters. The ‘Voter’ class contains an array of ‘Position’ pointers which can be used to tell that what are the allowed positions on which the voter can cast vote.

**Statement 9:** Each voter is also assigned to vote at a specific poll – each poll has a number and is located in a polling station.

**Solution:** There is a class named ‘Poll’ which has an attribute ‘PollNumber’ and an attribute ‘PollingStationAddress’. There is one-to-many relation between the ‘Poll’ class and the ‘Voter’ class. The ‘Voter’ class has ‘Poll’ pointer that can be used to tell which poll is assigned to a voter.

**Statement 10:** The system records the name and address of each voter.

**Solution:** There is a class named ‘Person’ with attributes ‘Name’ and ‘Address’. The ‘Voter’ class is a child of the ‘Person’ class.

**Statement 11:** Finally, the system will keep track of the number of votes for each candidate at each poll.

**Solution:** There is an Association class named 'Vote' between 'Candidate' class and 'Poll' class. There is a one-to-many relation between the 'Candidate' class and the 'Vote' class. There is a one-to-many relation between the 'Poll' class and the 'Vote' class.

The 'Vote' class stores three items:

- NoOfVotes: the number of votes gained.
- 'Candidate' pointer: Points to the candidate who gained votes.
- 'Poll' pointer: Points to the 'Poll' at which the candidate gained votes.

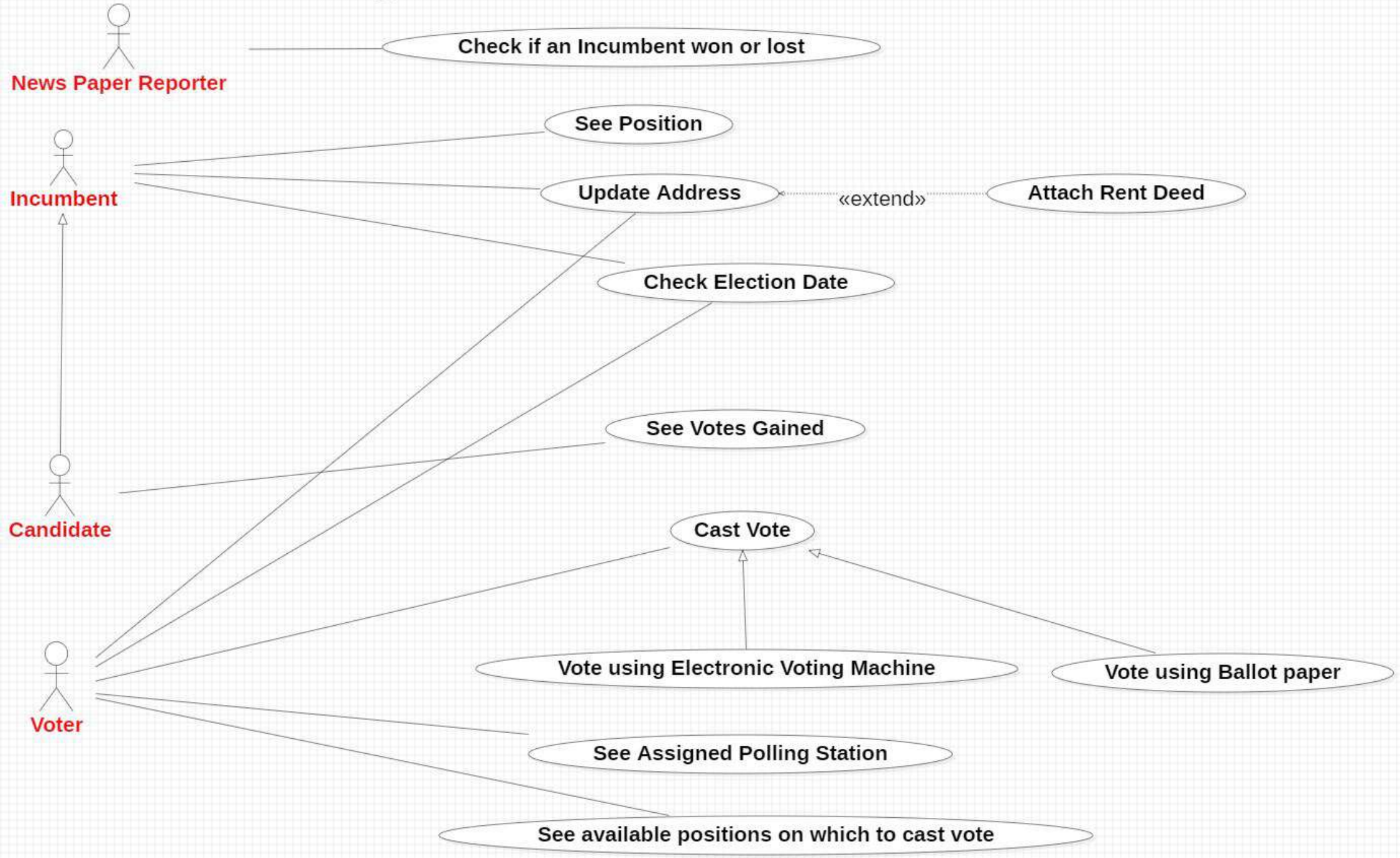
The 'Candidate' class has an array of 'Vote' pointers which be used to track the number of votes of each candidate at each poll.

The 'Poll' class has an array of 'Vote' pointers which can be used to track the number of votes at each Poll for each candidate.

**Statement 12:** However, under no circumstance will it record which voter voted for which candidate, nor whether a voter voted at all

**Solution:** No such information is being stored.

# Use Case Diagram



# Use Case Diagram Continued