Adam Hayes, Ariella Hanna

Professor Chang, 1631

Email Voting Software Plan

1. **Scope**

The University of Pittsburgh’s department of Computer Science would like to have a voting application to tally votes towards presentations and posters displayed on this day. This project will provide a software application that will allow attendees with smart phones to vote for their favorite displays by sending an email containing the candidate ID of their choice to a specific computer, which at the end of the voting process will tally the votes and display the winner and the runner ups, in descending order of votes. This project will be completed by the end of the 2017 spring semester, as dictated by the project specifications.

* 1. **Functions**

The software will be able to perform the following actions:

Poll for and collect email messages from smart phones.

Isolate and retrieve phone number and candidate ID from messages.

Maintain a database of votes with source phone number as key and candidate ID as value.

Protect against repetitive voters by checking against keys in the database.

Check for admin commands to begin/end vote polling process.

Upon termination of vote polling, display candidates by descending number of votes.

* 1. **Performance**

This system should be able to handle simultaneous and constant email input for the duration of the voting period. This system should also be able to handle the same number of vote messages as there are CS Day attendees, and should be able to process and display the results within a reasonable amount of time with respect to the size of the database.

* 1. **Limitations**

This product will not be able to accept incoming emails from devices that are not smart phones, as there would be no phone number to process. This product will also only be able to process email messages as votes if the email has a specific format that will be predetermined by the developers.

1. **Tasks**

Write:

Polling Function

Email message parsing function

Database update function

Database validity check function

Vote tallying function

User Interface

1. **Resources**
   1. **Hardware**

Development: Personal computers of developers, computer lab computers for

Different operating systems

Deployment: Specified computer, smartphone email applications

(unspecified number)

**3.2 Software**

Code sharing service (Github)

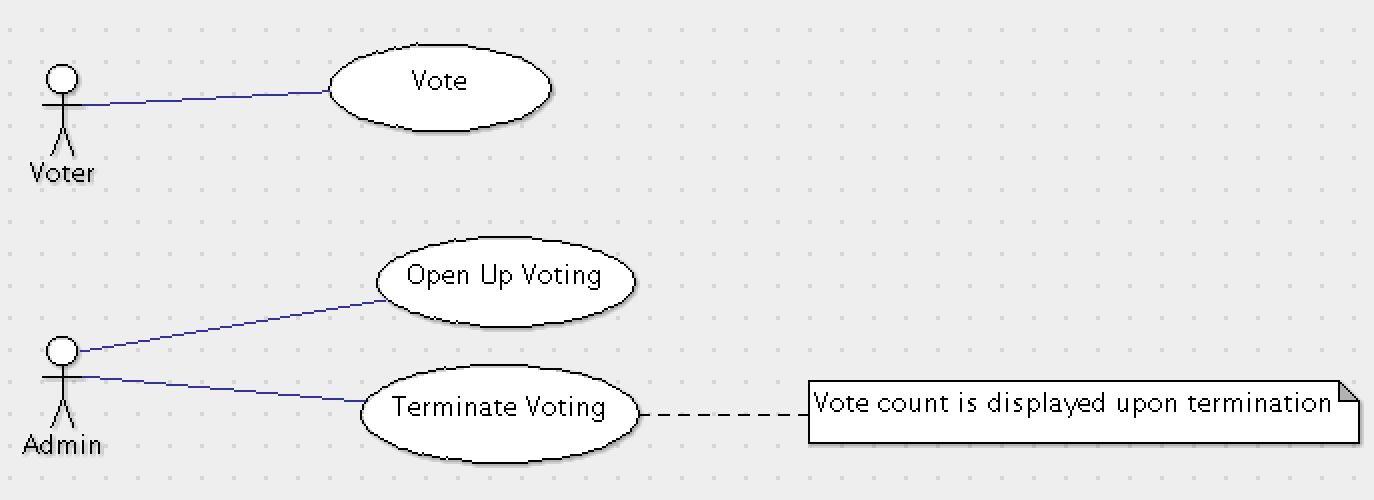
Most recent JDK

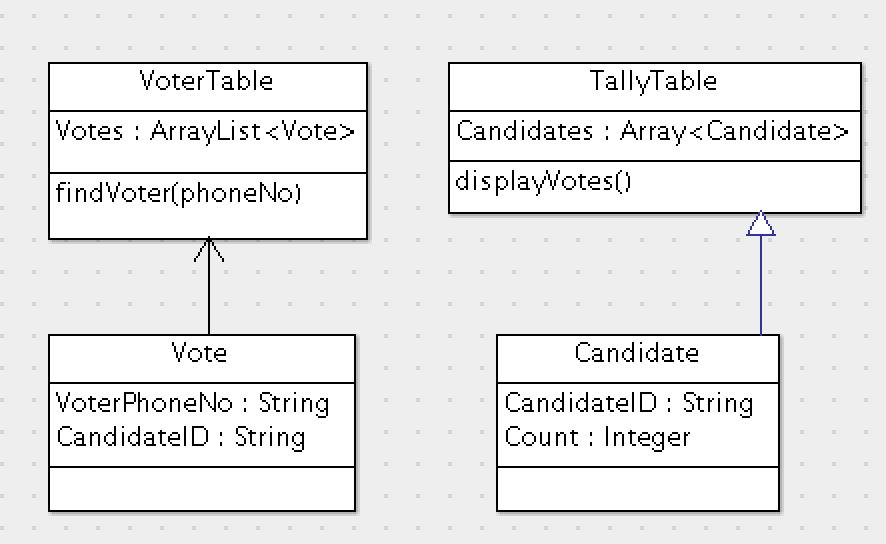
**3.3 People**

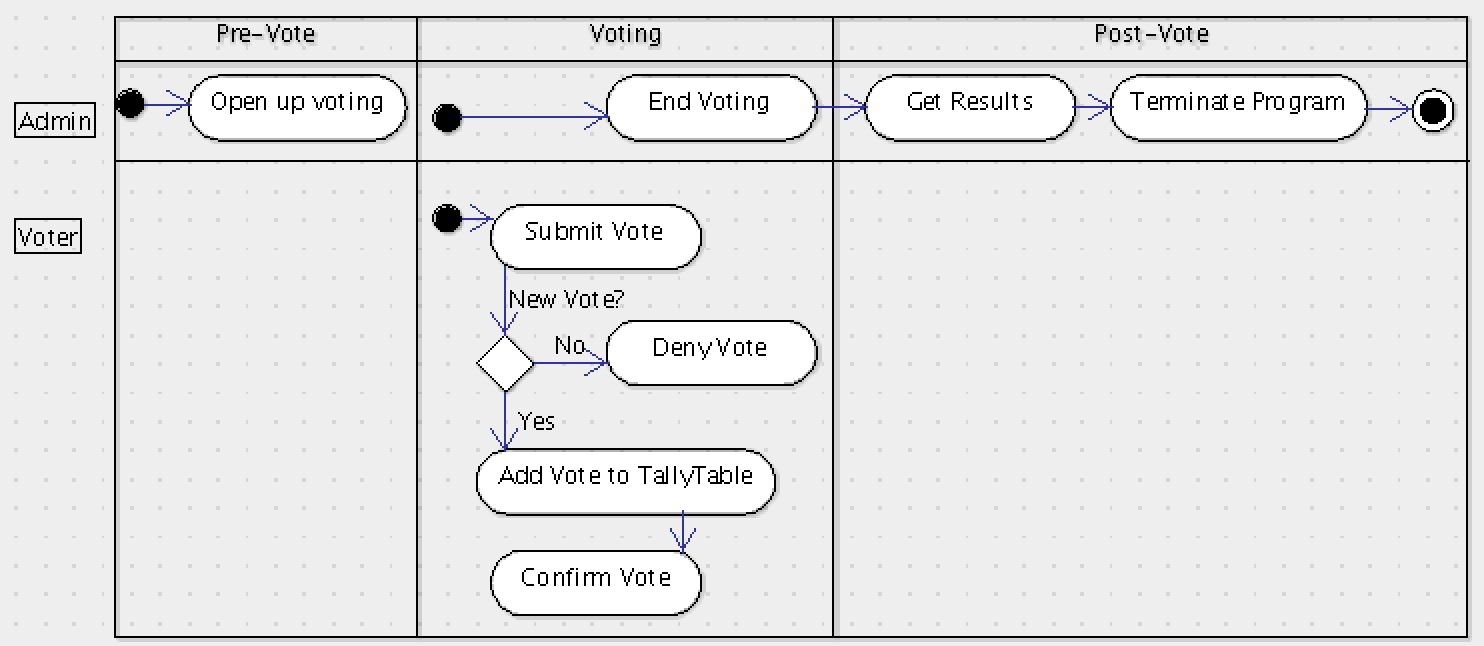
Software Developers – Adam Hayes, Ariella Hanna

Quality Assurance Team – Adam Hayes, Ariella Hanna

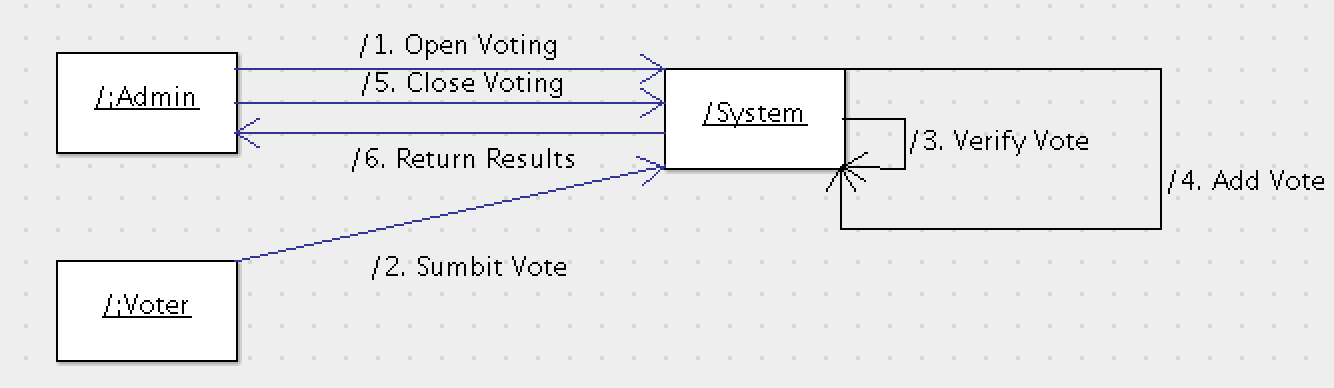
1. **UML Diagrams**
   1. **Use Case Diagram**

**4.2 Class Diagram**

**4.3 Activity Diagram**



**4.4 Collaboration Diagram**

**4.5 Deployment Diagram**

