



# Final Presentation Election Software

Team Why?

Members: Richard Marshall, Levi Hagan,  
Hannah Posch

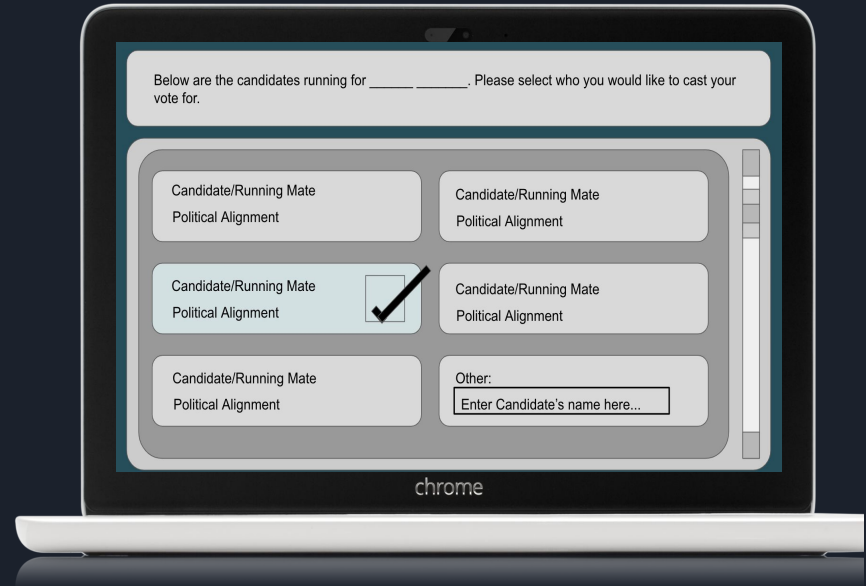


# Functional Requirements

- Voter Registration
  - Easy to use
  - Individual must be identified
- Casting a Ballot
  - Individual must be identified
  - Selection must be indicated as made or canceled
  - House and senate candidates should be included
  - Notify the voter when the selection is completed
  - Allow voter to review choices
  - Prevent over-voting
- Tallying the Ballots
  - Unofficial vote tally
  - Votes transferred to central location for official tally
- Certifying the Vote
  - Number of votes consistent with number of voters
  - Recounts possible

# Non-functional Requirements

- Security
- Performance
- Reliability
- Usability





# Analysis - Software Specification

- **Main Stakeholders**

- Government
- Voting Service
- Candidate

- **User**

- Citizens

## **Summary of System Features**

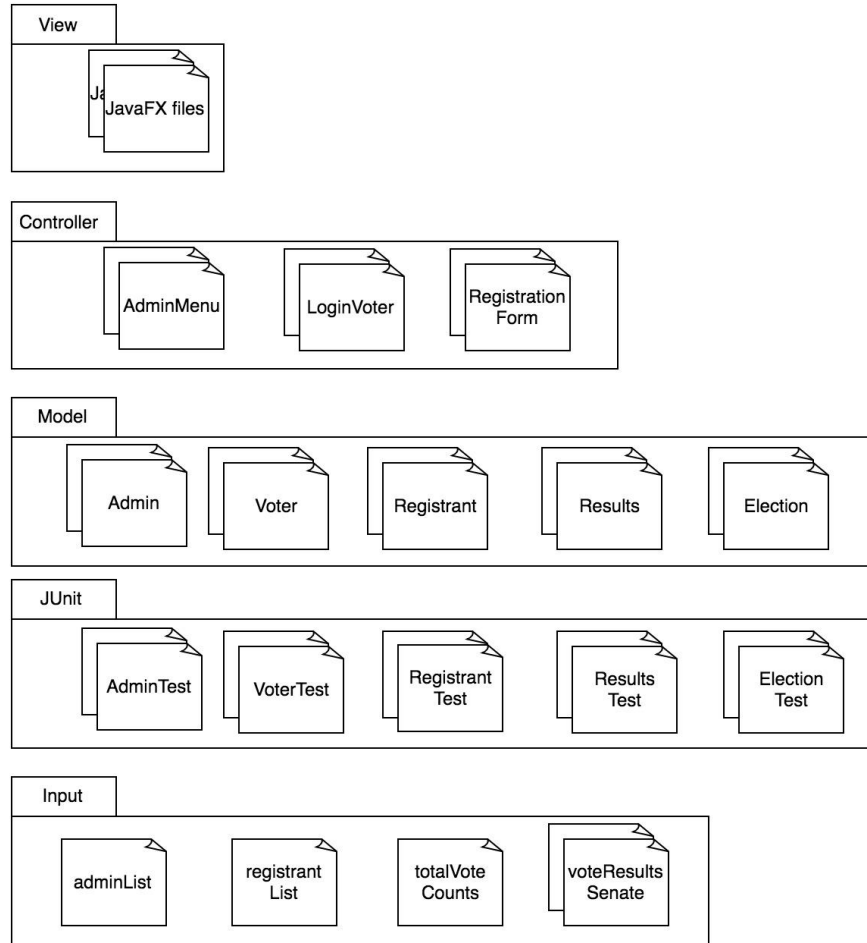
- Vote submission
- Voter authorization
- Admin authorization
- Registration
- Real-time unofficial voting tally



# Analysis - Software Specification

High-Level Goal	Priority	Problems and Concerns	Current Solutions
Fast, reliable secure vote submission	High	<ul style="list-style-type: none"><li>-Secure system not susceptible to hacking and voter fraud</li><li>-System bugs that prevent accurate voting results or cause inability to vote</li><li>-Reduced submission speed as more systems come online</li><li>-Inability to authorize voter</li></ul>	Current voting systems supply basic voting needs, but do not prevent all problems described.
Easy to use interface, and able to locate voting results	High	<ul style="list-style-type: none"><li>-User interface is too complicated and user is unable to complete task</li><li>-Voting results are not provided when requested</li><li>-Administrator not able to authenticate identity to access results</li></ul>	Current voting systems do meet most of these needs, however, if we could make our system more efficient and easy to use, then that is our current goal.

# System Design - Architectural Design





# System Design - Physical Design

The UI of our software had several requirements. All of which we were able to achieve.

- Color Blind Friendly Color Palette (Monochromatic)
- Large Sized Font that is able to be read from 1m away
- Touch Screen Compatible
- Simple Design Layout



# JUnit Testing - Software Validation

- AdminTestList
- AdminMenuControllerTest
- AdminSessionTest
- AdminTest
- BallotTest
- CandidateTest
- RegisteringSessionTest
- RegistrantListTest
- RedistrantTest
- RegistrationFormControllerTest
- VoterListTest
- VoterTest





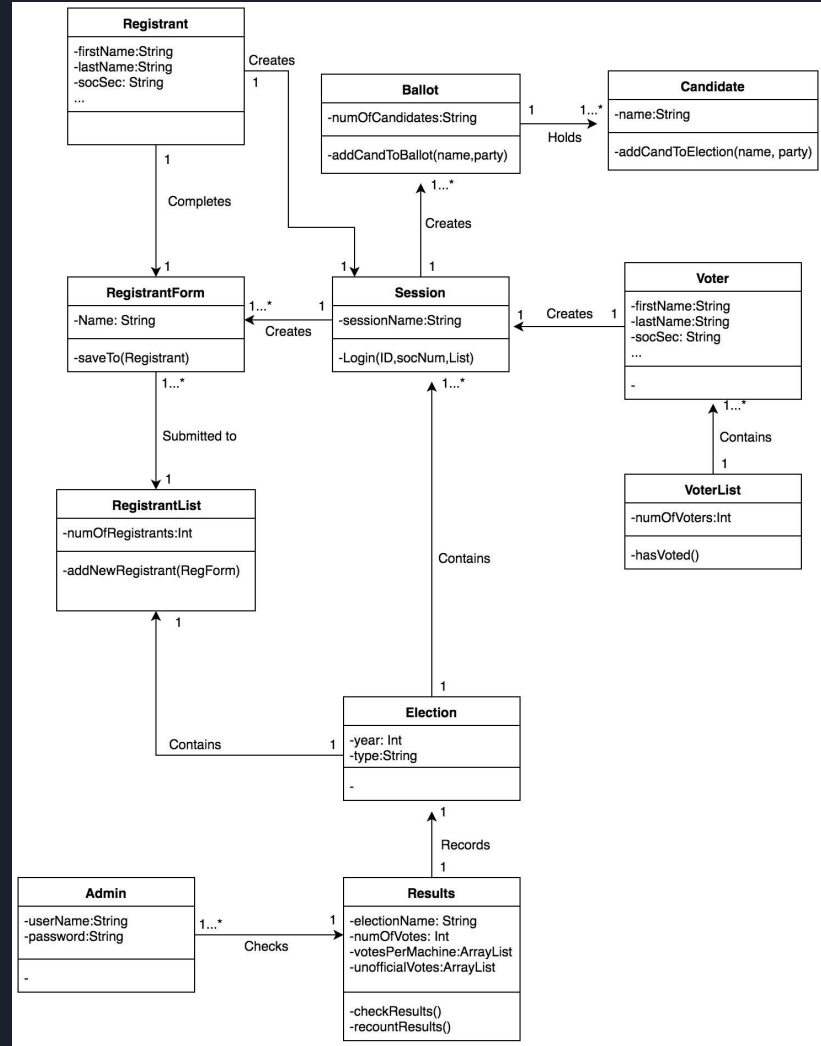
# Software Evolution

During the evolution of our Voter software many ideas were proposed and implemented

- Encryption on all passwords and sensitive data
- The Help menu as a popup.
- Admin Control/ Menu
- The Registration system
- Vote Tally to file.
- Multiple checks on valid input and entry

# Objects

- Registrant
- RegistrantList
- RegisteringSession
- Voter
- VoterList
- VotingSession
- Admin
- AdminList
- AdminSession
- Election
- Candidate
- Ballot
- ResultsDisplay
- MachineDisplay



# Demo



Questions

