THOMPSON RIVERS UNIVERSITY

SENG 3210

Applied Software Engineering (Winter 2024)

Voting Android Application

Eric Jude Jacob T00691963

Felix Beauchemin-Berthelot T00684599

2024-02-16

1) Brainstorm:

Different components that we want to implement in this project:

Welcome screen with login page for access into the application.

- Prompt the user with a terms of service page, listing information on privacy and the policies of the applications.
- Login page with required fields of information, name, driver's license, and voter ID to verify identity.

User authorization and authentication:

- The authorization will be done using a specified QR code on the voter information card, and will be 2-factored authenticated with a valid ID through an ID scanner module.
- Both name and address will be verified to match both pieces of identification to validate the voter’s information.

Databases:

User Database:

- Database will be created and will store the information of the user with a composite key made of the person’s name and address. The database will contain more information like the city and the postal code to determine the jurisdiction of the voter.
- A database will be used to map the postal codes to the jurisdiction.
- Another database will be used to provide the electoral candidate information based on the jurisdiction.

Voting Screen:

- The voter will be prompted with the electoral candidates for their jurisdiction.
- Users will be able to press and select their vote, and confirm it with a button.
- Votes will be stored in the electoral candidate database, and votes will be counted through the aggregate function COUNT().

Other considerations:

Voters can only vote once, this will be enforced by a check if the combination of their address and their name matches with a composite key already in the user database. If it already exists, a message will be prompted to the user that they are not eligible to vote.

2) Tasks and Modules:

Welcome Screen:

The welcome screen will be a simple view with the following information:

- It will contain the terms of service and user agreement sections.
- It will Provide rules and regulations behind the application.

- It will instruct the user on how to login.

Login Page:

- It requires users to provide a picture of their voter information card.
- Utilizes barcode readers (future implementation) to read the barcode on the voter information card.
- Requires a picture of two separate IDs for address and name verification.

Identity Verification:

- Login requires name, driver's license, and voter ID.
- Two-factor authentication using voter information card QR code and ID scanner.

User Database:

Class: User

- Stores the user information such as name, address, driver's license, and voter ID after the voter already voted.
- Implements methods to check if the user already exists in the database, to ensure users can’t vote more than once.

Database: UserDatabase

- Stores user data with a composite key of name and address to uniquely identify people.

Postal Code Mapping:

Class: PostalCodeMapper

- Maps postal codes to jurisdictions. This will be used to determine which jurisdiction the user lives in.
- implements methods for retrieving the jurisdiction information based on the input postal codes.

Database: PostalCodeDatabase

Stores postal code and their corresponding jurisdiction.

Candidate Information:

Class: Candidate

- Represents electoral candidates with the following attributes: name, party and platform.
- Implement methods to retrieve information about the candidate.

Database: CandidateDatabase

Stores candidate profiles, profile pictures, description.

Voting Process:

Class: VotingProcess

- Manages the voting process, including the user's selected votes and the storing of the vote in the CandidateDatabase.

Implements user interface elements for voting interaction like giving the user the ability to read the candidate's description and other information by reading from the CandidateDatabase

Class: VoteConfirmation
Displays a confirmation to the and controls a button so that the users can confirm their vote. This is meant to ensure that no votes are accidental.

Vote Counting:
Class: VoteCounter
Uses the database aggregate function "COUNT()" to count the votes of each candidate.

One Vote per Voter:
Class: VoterRegistry

- Manages voter registration and eligibility, like confirming Canadian citizenship and age of the voter.
- Implements a method to check if the user already exists in the UserDatabase meaning they already voted, in that case prompt error message to the user.

3) Task Distribution:

Welcome Screen: [Felix and Eric | Before: February 23rd 2024]
Login Page: [Felix and Eric | Before: February 23rd 2024]

User Database: [Felix and Eric | Before: February 30th 2024]
Postal Code Mapping: [Felix and Eric | Before: February 30th 2024]
Candidate Information: [Felix | Before: February 30th 2024]

Voting Process:[Eric | Before: March 5th 2024]
Vote Counting: [Felix | Before: March 5th 2024]
One Vote per Voter: [Felix | Before: March 5th 2024]

Identity Verification: [Work in progress, if we have time to implement it at the end.]
Additional Considerations: [Work in progress, extra features will be done at the end of the project if we have extra time.]