Carleton
UNIVERSITY
Canada's Capital University

SYSC 5709

ADVANCED TOPICS IN SOFTWARE ENGINEERING

SOFTWARE DEVELOPMENT IN C

INSTRUCTOR: CRISTINA RUIZ MARTIN

TOPIC: ELECTRONIC POLLING SYSTEM

Project by

Vignesh Balaji	Soma Sundaram Ravindran	Priyanga Soundararajan	Aswanth Ramamoorthy
Student Id:101139628	Student Id: 101137129	Student Id: 101139629	Student Id:101142819
Git Id: vigneshbalaj	Git Id: somasundaramravindra	Git Id: priyangasoundararaja	Git Id: aswanth-ramamoorthy

I. Problem Statement:

Most of the highly populated regions in the world face a hard time when elections are held. We have designed an electronic polling system which is fully automated and can be used from any place by the voter. The main objective of the project is to provide the voters with highly authentic and reliable polling system with the ease of voting on the go. This will ensure high voting rates as well as prevent proxy voting in highly populated regions. This technology aims to speed up ballot counting, reduce the cost of paying staff to count votes manually and provide disabled voters with greater accessibility. In this system, we will be creating a CSV file of voters and nominees. Each voter will be provided with an option to login with their registered / government issued credentials. Once logged in, the system will fetch the list of nominees present in the province of the voter's residence for whom he/she is eligible to vote. The voter can choose from the list to vote and after casting his vote his session will be terminated. Based on predefined conditions, the user will be prevented from casting duplicate votes. The application will also be provided in a user-friendly format for the voter to access from anywhere so that one does not fail to cast his vote.

II. Purpose of this software:

- To provide the voters with a convenient voting method from anywhere in the world in a user-friendly format.
- To provide a highly authentic system which makes use of the user's government issued credentials.
- To achieve 100% votes with the ease of accessibility.
- To prevent proxy voting / duplicate voting.
- To save time and money of citizens who live abroad and in rural areas.

III. Project Description:

The requirement here is to develop an e-polling machine with an efficient algorithm.

1) User will be prompted with admin and voter login options
2) Voter will be able to login into the system using their government issued credential.
3) Once successfully logged into the system the user can see the list of government issued ID proofs, the user can proceed to vote by entering the number in his desired ID proof.
4) Based on the users' ID proof, the system will direct him to his constitution/province and show the list of candidates along with the parties nominated for the constituency.
5) Voter can cast the vote to the desired candidate. On successful voting a confirmation is shown to the user.
6) Voter can cast only one vote. The session will terminate immediately after the first vote.
7) If the user logs in as admin, he will be authorized to view a file with a list of all candidates, province-wise along with the number of votes for each.
8) Results are displayed based on the user's input

IV. Assumptions:

1) It is assumed that functionality to add, update and delete the list of voters and candidate records (nominees) are provided by an existing system and is not part of this problem.
2) The list of voters as well as the candidates of the constitution are maintained in the database/file
3) Every citizen eligible to vote is provided with a secure identification number by the government.
4) The user credentials are maintained in a highly secure server and is accessible only by an authorized official (Admin).

V. Releases:

The project is released in two modules, with a more secure system in the second release.

Release 1:

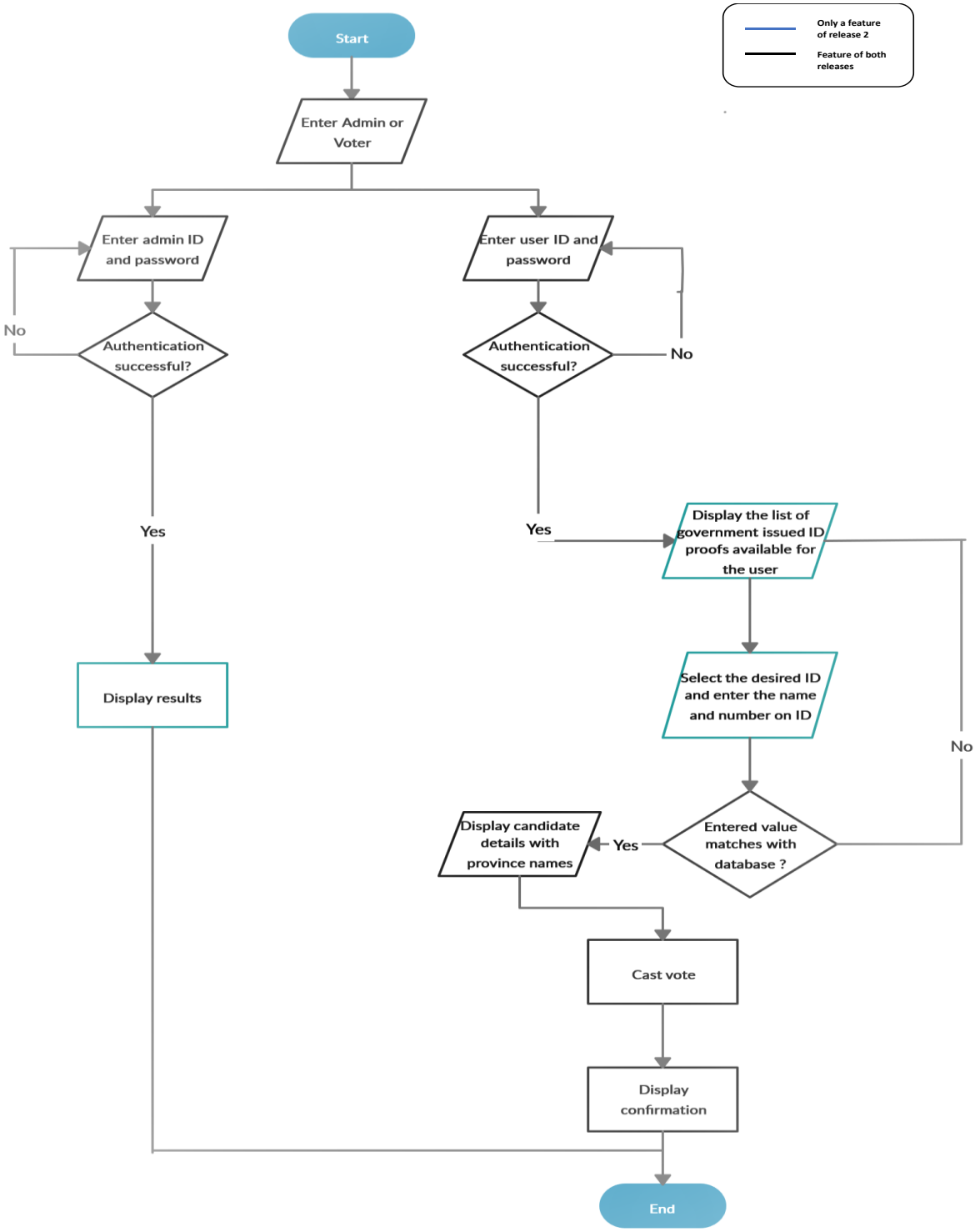
1) A prompt will appear where he/she will choose whether the user is a voter or admin.
2) At this stage, only user login is enabled and admin will be able to login but admin features will be available only in release 2
3) The system reads the credentials from the user to authenticate and begin the session.
4) The system will check the users' credentials and fetch the constituency/province to which he/she belongs.
5) The system will display all the parties and its candidates for that constituency/province.
6) Now the user can cast his vote to his desired candidate.
7) On successful voting, a confirmation is received.

Release 2:

8) A prompt will appear where he/she will choose whether the user is a voter or admin.
9) A two-step authentication system for the voters.
10) The system will display the name along with the list of secondary IDs on successful login.
11) The user can select any one of his desired ID and enter its value to enter the voting session. Then follows the step 2 to 5.
12) Finally, the total number of votes for each candidate will be displayed, province-wise upon Admins' request.

DELEVERABLE-2

VI. Flowchart:



VII. Process Flow:

1. Once the user enters the application, a prompt appears where he/she will choose whether the user is a voter or admin.
2. If the user chooses admin and authentication succeeds, the user will enter the application.
3. Admin will be asked to enter file name to view the election results.
4. Once he enters the correct file name, the vote count of each candidate under all provinces will be displayed.

Steps 2 to 4 are covered in 2nd release.

5. If the user chooses voter, he/she will be prompted to enter his/her government issued credentials.
6. Upon successful authentication, the user will be listed with his/her details and a set of other government issued ID proofs from which he/she can choose one to enter the application.
7. Once verified, the application will display the list of nominees and their respective party available in the voter's province.
8. The voter can then cast his/her vote.
9. The application will terminate with a confirmation message after voting.

Step 6 is covered in 2nd release

☐ User data, Candidate details, Results of poll and User authentication details are all managed using files.

VIII. Functions and descriptions:

1. Int user_login()

This function is used to verify or authenticate the user. This will also prevent proxy voting and fake votes.

- **Parameters:** None
- **Return 1:** User is authenticated
- **Return 0:** Authentication failed

2. **Int admin_login()**

This function allows only the admin to access the system.

- **Parameters:** None
- **Return 0:** When login fails
- **Return 1:** When login succeeds

3. **Int user_details(char* name)**

This function displays the details of the user like name, phone and province

- **Parameters:** char * name
- **Return 0:** When user details are found
- **Return 1:** When user details are not found

4. **int secondary_authentication(char* name)**

This function provides a secondary authentication to verify the user.

This function will take the 2nd government issued ID name and number as input and verifies the user login. This is done assuming that only the user has access to all his government issued ID details. Even if someone tries to cast a fake vote with the user's primary login details, the person may not have access to other ID details.

- **Parameter:** char* name
- **Return 1:** User is authenticated
- **Return -1:** Authentication failed

5. **Int candidate_details(char* province)**

Once the user is authenticated, only the list of nominees in the province to which the user belongs is displayed. This will prevent the user to cast vote to nominees in other provinces to which the user is not permitted to vote

- **Parameters:** char* province
- **Return 0:** Candidate details are displayed
- **Return 1:** Candidate details are not displayed

6. **Int cast_vote(char* province)**

This function enables the user to cast his/her vote to the desired candidate. It takes the primary user id number and province as input. Each casted vote is summed up candidate-wise and stored in the file.

- **Parameters:** char* province
- **Return 0:** When entered candidate ID is valid
- **Return 1:** When entered candidate ID is not valid

7. **int display_count()**

Only the admin is granted access to this function. It enables the admin to view the polling results

- **Parameters:** None
- **Return 0:** File name is invalid
- **Return 1:** File name is valid

Unit testing for each function has been done

IX. Function Allocation:

Function/Contribution	Release	Group Member
int user_login()	Release 1	Vignesh
int admin_login()	Release 1	Vignesh
int user_details(char* name)	Release 1	Aswanth
int candidate_details(char* province)	Release 1	Priyanga
int secondary_authentication(char* name)	Release 1	Aswanth
int cast_vote(char* province)	Release 1	Soma Sundaram
int display_count()	Release 2	Soma Sundaram
int attempt_count()	Release 2	Vignesh
Project documentation	Release 2	Priyanga
Developer manual	Release 2	Soma Sundaram
User manual	Release 2	Vignesh
Reusable functions	Release 1	Priyanga
Code integration	Release 2	Aswanth
Integration and usability testing (Scenario based)	Release 2	Priyanga
Make file	Release 2	Aswanth
Readme file	Release 2	Soma Sundaram
Doxygen	Release 2	Vignesh