Case Study: MyVote – Voting System

Group 6

Team member Emp ID	**Team member Name**	**Team Member Email Id**
46283010	**Mona Supriya**	**supriya.mona@capgemini.com**
46283152	Omni Gupta	omni.gupta@capgemini.com
46283149	Ruchi kumari	ruchi.h.kumari@capgemini.com
46283119	Snehlata Verma	snehlata.verma@capgemini.com
46289245	Sucharita Behera	sucharita.behera@capgemini.com
46283009	Priti Kumari	priti.g.kumari@capgemini.com

PROBLEM STATEMENT

1.2 O BJECTIVE

To create MyVote system for the users so they can nominate themselves as well as vote for the candidate. The application is to be developed as an executable file compiled on Linux. There are 2 entities Admin and User.

1.3 ABSTRACT OF THE PROJECT

1. Admin should be able to create election polls and choose nominees.
2. Users should be able to nominate themselves as candidate.
3. Each user is voter as well. Voter should have authentication to be able to vote. Once vote cannot revote.
4. Admin should calculate the result and publish it to user.
5. Handle data and errors properly. Show appropriate messages to the user.
6. Display good input, and output messages, and reports in a proper format.
7. Security features should be implemented wherever possible. For example, user passwords can be stored in an encrypted format.

1.4 FUNCTIONAL COMPONENTS OF THE PROJECT

Following is a list of functionalities of the system. Wherever the description of functionality is not adequate; you can make appropriate assumptions and proceed.

1. When MyVote starts it displays “Login Screen” with two two input messages -

```
-------------------------Login Screen--------------------------
Username
Password
```

System should maintain comma separated users.txt where each line stores one user and password. Be default this file contents one line – admin, admin – denoting admin user and password.

Whenever username and password is entered, system authenticates it with entry in “users.txt” file.

- If match is found and if admin user then “Admin Screen” is displayed.
- If match is found and if not admin user then “User Screen” is displayed.
- It match is not found then message “Invalid User or password” is displayed and system exits.

2. When admin user login MyVote starts it displays “Admin Screen”

```
-------------------------Admin Screen--------------------------
   1. Create Users
   2. Create Polls
   3. Calculate Poll Results
   0. Quit

   Enter your option : <option>
```

option = 1 (Create Users)

Admin will create multiple users. By default first password is same as username. These users information should be appended to “users.txt”.

For Example -

```
Enter username : sdlele
Do you want to add more users (y/n)? y
Enter username : smmohite
Do you want to add more users (y/n)? n
```

After this content of “users.txt” will be -

```
admin,admin
sdlele,sdlele
smmohite,smmohite
```

option = 2 (Create Polls)

Admin will fill following details of poll -

1. Poll Name String(10)
2. Poll Description String(50)
3. Ward No Number
4. PollDate Date (in DD/MM/YY format)

Do you want to add more polls (y/n)?

These details will be appended to comma separated “polls.txt” file.

For Example -
If admin enters two polls then polls.txt file will have following two lines -

```
Poll1,First poll,23,23/04/2019
Poll2,Second poll,40,12/12/2019
```

option = 3 (Calculate Poll Results)

Poll results will be calculated and a line appended to file “pollresults.txt” in following format -

PollNo, CandiateUsername1, Candidate1Votes, CandiateUsername2, Candidate2Votes, CandiateUsername3, Candidate3Votes,....

Above list should be stored in descending order of votes ob tained by each candidates. Number of votes calculated from “votes.txt” file.

Poll result is then displayed in following format -

```
Poll No – Poll1n
Poll Description – First poll
Ward No. - 23
Poll Date – 23/04/2019
Candidate Name              Number of votes
-----------------------------------------------------
abc                          1000
def                            450
ggggggggg                      330
-----------------------------------------------------
```

3. When normal user login MyVote starts it displays “User Screen”

```
-------------------------User Screen--------------------------
    1. Nominate
    2. Vote
    3. Display poll results
    4. Change password
    0. Quit

    Enter your option : <option>
```

option = 1 (Nominate)
This option will display all available polls. User will select poll number for which he/she wants to nominate. A line will be appened in “candidates.txt” file with format – Poll No,username

option = 2 (Vote)
This option will display all polls whose voting is on today’s date. After use selects poll number – its candidates will be shown from “candidates.txt” file.
User selects the candidate number he/she wants to vote for. This vote will be recorded by appending a line in file “votes.txt” with format –
PollNo,username,candiatename

option = 3 (Display poll results)
Ask poll number from user then display result of poll from “polls.txt” and “pollresults.txt” file in following format -

```
Poll No – Poll1
Poll Description – First poll
Ward No. - 23
Poll Date – 23/04/2019
Candidate Name                  Number of votes
-----------------------------------------------------
abc                              1000
def                                450
ggggggggg                          330
-----------------------------------------------------
```

option = 4 (Change password)
When admin creates new user by default password is same as username. If user wants to change password he/she will select this option. MyVote system should ask for new password. It then updates the line in “users.txt” for current user with username,newPassword

Assumptions: <Write assumptions made>

Technical Requirements -

1) C programming language
2) Use file input/output operations to read and write data.
3) Use multiple Linked Lists to read data from files at the beginning and write updated data to

files before applicaton ends.

4) Use dynamic memory allocation.

Non Functional Requirements

1) Multi-file multi-directory solution is expected. Modular and maintainable code (comments) and all coding standards should be followed.

2) makefile to build application. Two-step compilation process - .o and then executable should be generated.

3) Use valgrind tool on application executable to detect memory leak. Final valgrind report to be submitted in “reports” directory.

4) Level 0 DFD (context diagram), Level 1 DFD, Flow diagram and pseudocode for 2 complex functions logic.

5) SRS in pdf format, RTM, Plan, Presentation. MOMs

6) HLD_LLD Document (optional)

7) Design review, Code review, Inspection Logs of design and code reviews

8) Unit test cases and Integration test cases in UT_IT document. Both types of test cases i.e. sunny and rainy should be present in this document

Set Up Checklist for Project

Software Requirement:

Vi Editor, ctags, splint, valgrind, gcc, make, git account

Minimum System / Hardware Requirements:

Laptop with access to internet and Linux OS