**SCHOOL OF INFORMATION TECHNOLOGY**

**Bachelor of Information Technology**

Software Development Methods – IT6x23

**Semester Two – 2017**

**Programming Assignment**

**Student:** Christopher Simmons 21606081

**Date:** October 9

**Background**

Development for the following scenarios are completed and provided to you as starting point for the Epic Voting System, version 1 (at the end of the first iteration):

When the Voting System is run for the first time by the office administrator, it reads two text files (**staff.txt** and **candidates.txt**) containing the staff and candidate lists. This data is used by the program to validate staff, display the candidate list and update voting information.

This is a scenario of a staff casting a vote:

1. The main screen displays the system name, a welcome message and prompts the user to choose between a **Voter,** **Administrator** or **Help**.
2. The staff chooses **Voter** option.
3. The system requests the staff to enter the staff ID, which is done by the staff.
4. If the ID is a valid input, the system checks if the staff exists on its staff list and if the staff is eligible to vote. (**A staff cannot vote twice!**)
5. If the staff is eligible to vote, the system welcomes the staff by name and displays a list of candidates (with their codes) to choose from.
6. The voter then enters the code of the candidate he/she wishes to vote for.
7. The system displays the name of the candidate chosen and asks the staff to confirm the choice.
8. If the staff accepts the choice, the system records this vote, thanks the staff and returns to the main screen.

The management has realised that substantial resources are used in terms of time and money to conduct this manual process of voting. Hence, they have decided to computerise the process.

You are part of a team that is assigned to develop a prototype electronic voting system to elect staff representatives to the Board of Directors. The system will run on one PC in the administration area at head office where every staff will cast his/her vote over a period of seven days.

The development will be carried out in several iterations. First iteration of this project has already been completed for you. A demonstration and explanation of up-to-date development will be provided to you and a scenario of completed steps are also provided with this document.

This assignment concerns the second iteration which is to complete the basic model that uses plain text files to store persistent data. Once this basic model is tried out, a more advanced version is envisaged and planned in the future iterations. The final version of the voting system would be an on-line multi-user system where staff identity is authenticated using the Active Directory and remote access to voting system via a secure website.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Purpose of Test** | **Test Values** | **Expected results of test** | **Actual results of test** | **Corrective action** |
| **Staff** | **Staff Voting functions** |  |  | Complete |  |
| 1 | Staff class is modified correctly, to store information for password and date of voting. | Staff Class | Staff class is modified correctly, store information | Complete |  |
| 2 | User is prompted to enter the staff id & password.  User input must be validated. An appropriate error message is displayed if this check fails and the user is prompted to re-enter. | Votingnterface> manageVote | User is prompted to enter the staff id & password. | Complete |  |
| 3 | If a non-existent staff id is entered more than 3 times the system goes back to the main screen. | Votingnterface> manageVote | goes back to the main screen | Complete |  |
| 4 | System checks for a staff who has already voted.  If the staff has voted already, an appropriate message is displayed and program reverts to main screen | Votingnterface> manageVote | an appropriate message is displayed and program reverts to main screen | Complete |  |
| 5 | System records date for when staff voted correctly | VotingControl> recordVote | System records date for when staff voted correctly | Complete |  |
| 6 | System checks date correctly to ensure voting can only take place within the 7 days voting period allowed. |  | 7 days voting period allowed. | Complete |  |
| 7 | If the staff is eligible to vote, a list of candidates are displayed and with a prompt to enter the candidate code. | Votinginterface> getStaffVote | list of candidates are displayed prompt to enter the candidate code. | Complete |  |
| 8 | The program asks the staff to confirm the candidate selection before updating the vote. If the staff cancels the selection, the program reverts to the candidate list. | Votinginterface> getStaffVote | the program reverts to the candidate list. | Complete |  |
| 9 | On updating the vote, the program displays a confirmation message and reverts tomain screen. | Votinginterface> getStaffVote | the program displays a confirmation message reverts main screen. | Complete |  |
| 10 | This transaction is saved to file – candidate vote count is incremented by one, staff voted status is set to true and date of voting is recorded. | VotingControl>recordVote | This transaction is saved to file | Complete |  |
| 11 | If the staff or candidate data file cannot be written to, an appropriate error message is displayed. | VotingControl>savesStaffData and saveCandidateData | appropriate error message is displayed | Complete |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Purpose of Test** | **Test Values** | **Expected results of test** | **Actual results of test** | **Corrective action** |
| **Admin** | **Administration functions** |  |  |  |  |
| 1 | System administrator details (id , name, username & password) stored and retrieved from file. | Admin Class | stored and retrieved from file. | Complete |  |
| 2 | Upon a successful login, system administrator will have options to see voting results as well as options to update candidate, staff and admin accounts. | VotingInterface>manageAdmin | options to see voting results and options to update candidate, staff and admin accounts. | Complete |  |
| 3 | System administrator can add, view and delete admin account from the list stored in Admin.txt” file. | Manage Class | add, view and delete admin account | Complete |  |
| 4 | System administrator can view, add and delete staff information | Manage Class | view, add and delete staff information | Complete |  |
| 5 | System administrator can view, add and delete candidate information | Manage Class | view, add and delete candidate information | Complete |  |
| 6 | System administrator can Add, view and update the date range for voting can take place. | Manage Class | Add, view and update the date range for voting can take place | In-Complete | Make method to re input startDate and endDate.  ERROR wants a string doesn’t like Data Type |
| 7 | System administrator can produce report to display the voting results (display candidate code, name, department and showing total number of votes for each candidate as well as statistical percentages (total and for each candidate) | VotingInterface> printVoteResults | produce report to display the voting results | Complete |  |
| 8 | System administrator can produce report for list of staff that have voted (including date voted) and staff that did not place a vote. | VotingInterface> printVoteResults | produce report for list of staff that have voted |  |  |

|  |  |  |
| --- | --- | --- |
| Enter "v" to Vote as staff  OR "a" to login in as system adminirstaror  OR "h" help options : | **public** **void** commenceVoting()  {  **boolean** systemQuit = **false**;  **while** (!systemQuit)  {  String input = **null**;  System.***out***.println("\n\t\t============== Main Menu =====================\n\n");  System.***out***.print("Enter \"v\" to Vote as staff \nOR \"a\" to login in as system adminirstaror \nOR \"h\" help options : ");  input = getInput();  **if** (input.equalsIgnoreCase("V"))  {  //New Staff Login  // validatestaff();  manageVote();  //Doesn't go to managePass  // managePass();  }  **else** **if** (input.equalsIgnoreCase("A"))  {  validateAdmin();  systemQuit = manageAdmin();  }  **else** **if** (input.equalsIgnoreCase("H"))  {  help();  systemQuit = **false**;  }  **else**  {  System.***out***.println("Your input was not recognised");  }  }  }  **public** **void** manageVote()  {  **boolean** moveOn = **false**;  **int** counter = 0;  //loop vote  **while** (moveOn == **false** || counter ==3)  {  System.***out***.print("Please enter your staff ID :");  **try**  {  theStaff = vc.getStaff(Integer.*parseInt*(getInput()));    System.***out***.print("Please enter your staff password :");  theStaff = vc.getStaffPass(Integer.*parseInt*(getInput()));        // password here  // theStaff = vc.getStaffPass(getInput());  **if**(theStaff.hasVoted() == 1)  {  System.***out***.println("\nYou have voted and cannot vote again\nGood bye...!");  moveOn = **true**;  }  **else** **if** (theStaff.hasVoted() == 0)  {  getStaffVote();  moveOn = **true**;  }  **else** **if**(counter ==3)  {  moveOn = **true**;  }  **else**  {  System.***out***.println("There seems to be a problem. Contact your administrator");  }  //3 times and then go back to main menu need a counter? loop?      }  **catch**(NumberFormatException e)  {  System.***out***.println("Invalid entry - you must enter a number\nPlease try again");  ++counter;  }  **catch**(NullPointerException e)  {  System.***out***.println("Error! Staff ID not found.\nPress ENTER to try again or \"q\" to QUIT : ");  ++counter;  //counter here?  **if** ("q".equalsIgnoreCase(getInput()))  {  System.***out***.println("Good bye!");  moveOn = **true**;  }  }  }  System.***out***.print("going back to voting screen...");  }  **private** **void** getStaffVote()  {  **int** candidateCode;  **boolean** retry = **true**;  displayVotingScreen();  //Date Log  **while** (retry)  {  System.***out***.print("\n\nEnter your candidate's code OR enter Q to quit voting : ");  **try**{  String input = getInput();  **if** (input.equalsIgnoreCase("Q"))  {  retry = **false**;  }  **else**  {  candidateCode = Integer.*parseInt*(input);  theCandidate = vc.getCandidate(candidateCode);  System.***out***.print("\nYou have selected " + theCandidate.getName()+ ". \n\nEnter Y to confirm or any other key to Cancel, then press ENTER : ");  **if** (getInput().equalsIgnoreCase("y"))  {  vc.recordVote();  System.***out***.println("\n\nThanks for voting " + theStaff.getName() + ". Bye!!!");  retry = **false**;  }  }  }  **catch**(NumberFormatException e)  {  System.***out***.println("That was not a number you entered\nPlease try again");  }  **catch**(NullPointerException e)  {  System.***out***.println("This candidate code does not exit\nPlease try again");  }  **catch**(Exception e){  System.***out***.println("We have a problem, please contact your administrator");  }  }  } | Today is the Day Bud.  Start: Sun Jan 01 00:00:00 NZDT 2017  End: Tue Oct 31 00:00:00 NZDT 2017  Today: Thu Oct 19 15:50:43 NZDT 2017  Please enter your staff ID :100  Please enter your staff password :100  Welcome Dong Long!  Code Candidate Name  ==== ==============  1 Jack Lee  2 Emma Burns  3 Jim Young  4 Susan Smith  5 Henry Jones  Enter your candidate's code OR enter Q to quit voting :5  You have selected Henry Jones.  Enter Y to confirm or any other key to Cancel, then press ENTER : y  Thanks for voting Dong Long. Bye!!!  going back to voting screen... |
| Enter "v" to Vote as staff  OR "a" to login in as system adminirstaror  OR "h" help options : h | **public** **void** help() {    **boolean** retry = **true**;    // displayVotingScreen();    **while** (retry)  {  String input = **null**;  // System.out.println("\n\n");  System.***out***.println("\n\t\t===================== Help =====================\n\n");  System.***out***.print("Enter \"vh\" Vote help \nOR \"ah\" adminirstaror help \nOR \"Q\" to quit : ");    **try** {  input = getInput();  **if** (input.equalsIgnoreCase("VH"))  {  System.***out***.println("\nTo vote goto main eVoting System menu,\npress \"v\" then type in your staff id and password");  retry = **true**;  }    **else** **if** (input.equalsIgnoreCase("AH"))  {  System.***out***.println("\nTo administrate goto main eVoting System menu,\npress \"a\" then type in your adminirstaror username and password");  retry = **true**;  }  **else** **if** (input.equalsIgnoreCase("Q"))  {    retry = **false**;  }  **else**  {  System.***out***.println("Your input was not recognised");  }  }    **catch**(NumberFormatException e)  {  System.***out***.println("Enter a number");  }  **catch**(NullPointerException e)  {  System.***out***.println("Error! Staff ID not found.\nPress ENTER to try again or \"q\" to QUIT : ");  **if** ("q".equalsIgnoreCase(getInput()))  {  System.***out***.println("Good bye!");  retry = **true**;  }  }  }    System.***out***.print("going back to voting screen");  } | Enter "vh" Vote help  OR "ah" adminirstaror help  OR "Q" to quit :  If Type vh  “To vote goto main eVoting System menu,  press "v" then type in your staff id and password”  If Type ah  “To administrate goto main eVoting System menu,  press "a" then type in your adminirstaror username and password” |
| ============== Main Menu =====================  Enter "v" to Vote as staff  OR "a" to login in as system adminirstaror  OR "h" help options : a  You have entered Administration space.  Enter username or "Q" to quit : adminMike  Please enter password : security123  ===================== Manage =====================  "C" :To continue voting enter  "Stop" :To end voting enter  "VR" :To View Voting Result  "HV" :To View Who Voted  "US" :To Update Staff  "UA" :To Update Admin  "UC" :To Update Candidate  "Q " :To Quit Back To Main Menu  Enter your choice: | **private** **void** validateAdmin()  {  **boolean** adminQuit = **false**;  **while** (!adminQuit)  {  System.***out***.print("\nYou have entered Administration space. \nEnter username or \"Q\" to quit : " );  String input = getInput();      **for** (**int** i = 0; i <vc.getadmins().size() ;i++)  {  theAdmin = vc.getAdminUsername(input);  **if** (theAdmin != **null**)  {  **break**;  }  }    **try**  {  // System.out.print(theAdmin.getPasWord());  **if** (input.equalsIgnoreCase("q"))  {  adminQuit = **true**;  }  **else**  {  String password = **null**;    System.***out***.print("\nPlease enter password : ");  password = getInput().trim();  **if**((theAdmin.getPasWord()).equals(password))  {    adminQuit = **true**;  }  **else**  {  System.***out***.println("Incorrect username.\"" + password+"\" does not match password stored for this admin");  }  }  }  **catch**(Exception e)  {  System.***out***.println("Incorrect username. Username \"" + input + "\" does not exit!");  }  }  }  **private** **boolean** manageAdmin()  {  **boolean** adminQuit = **false**;  **boolean** systemQuit = **false**;      **while** (!adminQuit){  System.***out***.println("\n\t\t===================== Manage =====================\n\n");    System.***out***.println("\"C\" :To continue voting enter");  System.***out***.println("\"Stop\" :To end voting enter");  System.***out***.println("\"VR\" :To View Voting Result");  System.***out***.println("\"HV\" :To View Who Voted");  System.***out***.println("\"US\" :To Update Staff");    System.***out***.println("\"UA\" :To Update Admin");  System.***out***.println("\"UC\" :To Update Candidate");  System.***out***.println("\"Q \" :To Quit Back To Main Menu");    System.***out***.print("\n\n");  System.***out***.println("Enter your choice: ");  String input = getInput();  **if** (input.equalsIgnoreCase("C"))  {  adminQuit = **true**;  }  **else** **if**(input.equalsIgnoreCase("Stop"))  {  //stop system  adminQuit = **true**;  systemQuit = **true**;  System.***out***.println("Voting System Closed");  }  **else** **if**(input.equalsIgnoreCase("US"))  {  adminQuit = **true**;  //ERROR Loops Choice  //FIXED 11/10/2017    m.ManageStaff();      }  **else** **if**(input.equalsIgnoreCase("UA"))  {  adminQuit = **true**;  //ERROR - goes back to main screen  //FIXED 11/10/2017  m.Manageadmin();    }  **else** **if**(input.equalsIgnoreCase("UC"))  {  adminQuit = **true**;  m.Managecandidate();    }  **else** **if**(input.equalsIgnoreCase("VR"))  {  adminQuit = **false**;  printVoteResults();  }  **else** **if**(input.equalsIgnoreCase("HV"))  {    **try** {  m.ViewAllRecord();  } **catch** (IOException e) {    e.printStackTrace();  adminQuit = **false**;  }  }  **else**  {  System.***out***.println("Good Bye... : \n\n");  adminQuit = **true**;  }  }  **return** systemQuit;  } | ===================== Manage =====================  "C" :To continue voting enter  "Stop" :To end voting enter  "VR" :To View Voting Result  "HV" :To View Who Voted  "US" :To Update Staff  "UA" :To Update Admin  "UC" :To Update Candidate  "Q " :To Quit Back To Main Menu  Enter your choice:  vr  VOTING STATS  =========================  Code Name Department Votes (Vote%)  \_\_\_\_ \_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_  1 Jack Lee Sales 1 (20%)  2 Emma Burns Log 0 (0%)  3 Jim Young Money 0 (0%)  4 Susan Smith Retail 0 (0%)  5 Henry Jones Storage 4 (80%)  Numbers on voting list: 17  Numbers voted: 5.0(29.41%)  ===================== Manage =====================  "C" :To continue voting enter  "Stop" :To end voting enter  "VR" :To View Voting Result  "HV" :To View Who Voted  "US" :To Update Staff  "UA" :To Update Admin  "UC" :To Update Candidate  "Q " :To Quit Back To Main Menu  Enter your choice:  hv  ------------------------------------------------------------------------------  ID Name Voted Date Password  ------------------------------------------------------------------------------  100 Jade Long 0 null 100  101 Barbara Jones 1 null 101  102 Amardeep Singh 0 null 102  103 Manoo Smith 0 null 103  104 Dean Kenwood 0 null 104  105 Zhuo Wang 0 null 105  106 Ann Sanderson 0 null 106  107 Kimberly Gibbs 0 null 107  108 Ayako Nakayama 1 04-57-17 108  109 Nikhil Singh 1 04-01-17 109  110 Steve Smith 1 null 110  111 Java King 0 null 111  112 Susan Brown 0 null 112  113 Cyrus Jin 0 null 113  114 Lisa Hiti 0 null 114  115 Sean Wingard 0 null 115  116 Dude 0 null 116  -------------------------------------------------------------------------------------  ===================== Manage =====================  "C" :To continue voting enter  "Stop" :To end voting enter  "VR" :To View Voting Result  "HV" :To View Who Voted  "US" :To Update Staff  "UA" :To Update Admin  "UC" :To Update Candidate  "Q " :To Quit Back To Main Menu  Enter your choice: |