**Epic Construction Case Study**

To assist with implementation of the following deliverable for Epic Construction Case Study provided to you in assignment one. The System administrator credentials Username and Password for Epic Voting System(EVS) is currently hardcoded as constant variables inside the program (*VotingInterface.Java)*. By the end of this workshop, this information must be stored and retrieved from text file.

* System administrator details (**id**, **name**, **username**, **password**) must be stored and retrieved from file.

This is similar to how EVS stores and retrieves data for candidates and staff from files. Study sections of code in EVS System that loads staff and candidate data as objects into the program and then save data back to file in comma delimited format. Methods used to achieve this are:

* public void loadStaffData()
* public Staff getStaff(int id)
* public void saveStaffData()
* public void loadCandidateData()
* public Candidate getCandidate(int candidateCode)
* public ArrayList<Candidate> getCandidates()
* public void saveCandidateData()

To implement this requirement, you first need to create a class to manage data for admin. Next create a file in the root directory of the program to store admin data in a comma delimited text file (“admin.txt”). Below is example of data stored inside “admin.txt”

100,Mike Smith,adminMike,security123

101,Kim Jones,kimman,kim123

102,Lisa Smith,lsmith,L%Smith\_123

104,Sam Lee,lee,security

You can then write code in VotingController and VotingInterface classes to create objects of admin class and validate user name password based on information retrieved from inside “admin.txt”. Here are some hints for code you need to write inside VotingController and VotingInterface classes:

**VotingController** class

* + Declare ArrayList to store array of admin objects
  + Declare object of admin class
  + Create a method called loadAdminData() that :
    - Reads content of “admin.txt” and split data for each line from file
    - Store data for each system administrator inside an object
    - Add admin objects to the ArrayList
  + Create method called getAdmins to return arraylist of admin objects. Sample code provided below:

**public** ArrayList<admin> getadmins(){

**return** admins;

}

* + Create a method called getAdminUsername() that receives one string parameter and returns admin object that matchs the username passed to it as parameter. Sample code provided below:

**public** admin getAdminUsername(String adminUserName)

{

Iterator<admin> it = admins.iterator();

**while**(it.hasNext())

{

theAdmin = (admin) it.next();

**if**((theAdmin.getUserName()).equals(adminUserName))

{

**return** theAdmin;

}

}

**return** **null**;

}

**VotingInterface** class

Next you need to modify validateAdmin() method inside VotingInterface.java to loop through admin ArrayList. In your loop, you need to check the Admin ArrayList and find the Admin object that matches username entered by the user. Display appropriate error message and ask for another input if user input does not match username for any Admin object stored in the program.

Once you find the Admin object with correct username, you can then ask user to enter the password and check that against password stored for that admin object inside the ArrayList in your program. Display appropriate error message and ask for another input if user input does not match password for Admin object stored in the program.