Team 1: Attendance Management System

An attendance management system is one of the major projects for university-level graduates. The project can be built using Java. It uses txt file as a storage of data and uses file I/O classes along with Object I/O classes to read and write the files.

Abstract: This project is an application that manages the attendance of any student in school or college. There are three type of users with the application. 1. Admin, 2. Faculty, and 3. Student. There should be POJO classes for Admin, Faculty and Student. Admin will create Faculty and Student Objects and save them in the file(user.txt) It has an admin feature that holds the access to do any kind of changes like update, delete, and add user records to the file. The classes and their contents are as follows: -

1. **User class**
    a. Data Members
        i. username
        ii. password
        iii. userType(1 for admin, 2 for Faculty and 3 for student)
    b. Methods
        i. Constructor
        ii. Getters and setters
        iii. Login() to search the objects from the User.txt file and check whether the input username and password matches or not. If they match, the menu for the user has to be displayed. Otherwise, should ask to re-login)
        iv. save() to save a new user object to the file
        v. printData() to print the details of the user object

2. **Faculty Class**
    a. Data members
        i. User
        ii. Name
        iii. Subject
    b. Methods
        i. Constructor
        ii. printDetails() to print the details of the Faculty
        iii. createFaculty() to add new faculty object in to the txt file for faculties
        iv. searchFaculty() to search and display the details of a particular faculty.

3. **Student Class**
    a. Data members

        i. User
        ii. Name
        iii. Course
    b. Methods
        i. Constructor
        ii. printDetails() to print the details of the student
        iii. createStudent() to add new students object in to the txt file for students
        iv. searchStudent() to search and display the details of a particular student

4. **Admin Class**
    a. Data Members
        i. User
        ii. Name
    b. Methods
        i. Constructor
        ii. printDetails() to print the details of the admin

5. **Attendance Class**
    a. Data Members
        i. Student
        ii. Date
        iii. attendanceStatus(Present/absent)
        iv. subject
    b. Methods
        i. Constructor
        ii. markAttendance() to save the attendance object of the students in linkedlist of attendance(two separate lists for absentees and present)
        iii. showAbsentees() to display the absent students details
        iv. showPresenties() to display the present students details
        v. checkAttendance() to view his/her (student's) attendance data from the linkedlist of attendance.

Write a main function, which will create an object of admin user and save in the txt file of admin. Then, ask the user to login with user name and password. Check for the successful login and allow to proceed with the menu. If login fails, ask the user to try login again.

Menu for different types of users are as follows:-

Admin:-

1. Create Faculty

2. Create Student
3. Search A Faculty
4. Search A Student
5. Print Details of all Faculties
6. Print Details of all Students
7. Print all User Data
8. Exit

Faculty:-

1. Mark Student Attendance
2. Print details of all Absentees
3. Print details of present students
4. Search a Student
5. Print Details of all Students
6. Exit

Student:-

1. Check Attendance
2. Print his/her details
3. Exit

Team 2: Supermarket Billing System

This application is a Java project that is usually built for keeping the sales recording made on a daily basis. It uses text files for recording the data of the users, products, and orders made by the user (customer).

Abstract: The Java application is implemented to keep a record of the products, status of the products orders, and user's history. This Application is made in terms that it displays records of bills made on that particular day, items added to the new bill also have an automated system that calculates the bill with GST and other applied taxes and has a print button to print the copy statement of the bill. It has an admin module that is responsible for adding, updating, or deleting records of the bill.
The classes and their contents are as follows: -

1. **User class**
    a. Data Members
        i. username
        ii. password
        iii. userType(1 for admin, 2 for Customer)
    b. Methods
        i. Constructor
        ii. Getters and setters
        iii. Login() to search the objects from the User.txt file and check whether the input username and password matches or not. If they match, the menu for the user has to be displayed. Otherwise, should ask to re-login)
        iv. save() to save a new user object to the file
        v. printData() to print the details of the user object

2. **Customer Class**
    a. Data members
        i. User
        ii. Name
        iii. Address
    b. Methods
        i. Constructor
        ii. printDetails() to print the details of the Customer
        iii. createCustomer() to add new Customer object in to the txt file for Customers
        iv. searchCustomer() to search and display the details of a particular Customer.

3. **Admin Class**
    a. Data Members

      i. User
      ii. Name
   b. Methods
      i. Constructor
      ii. printDetails() to print the details of the admin

4. **Product**
   a. Data Members
      i. productid
      ii. description
      iii. price
      iv. quantity
      v. GSTrate
   b. Methods
      i. Constructor
      ii. saveProduct() to save the Product in the text file.
      iii. printData() to search and print the details of the product
      iv. updateProduct() to search and change the quantity of a product after a purchase, in the text file
5. Purchase
   a. Data Members
      i. Product
      ii. Qty
      iii. Amount
   b. Methods
      i. Constructor
      ii. savePurchase() to save the purchase of an item in the linkedlist for that customer
      iii. printData() to print one Purchase detail
6. Bill
   a. Data Members
      i. PurchaseList (linked List)
      ii. Customer
      iii. totalAmount
      iv. date
   b. Methods
      i. Constructor
      ii. saveBill() to save the customer bill to a text file
      iii. printBill()

Write a main function, which will create an object of admin user and save in the txt file of admin. Then, ask the user to login with user name and password. Check for the successful login and allow to proceed with the menu. If login fails, ask the user to try login again.

Menu for different types of users are as follows:-

Admin:-

1. Create Customer
2. Search A Customer
3. Print Details of all Customers
4. Add Products
5. Exit

Customer:-

1. Search for the Products
2. See all products
3. Buy Products (check the quantity available for a purchase and deduct the qty of current purchase from the available quantity. Don’t allow purchase, in case of no sufficient stock)
4. Create Bill (calculate the total amount)
5. Exit

Team 3: Online Voting System

The Voting System can be the best project for college students. This project is designed to automate the voting process where multiple parties are added and then with the maximum votes, a leader is chosen.

Abstract: The main objective to build this application is to reduce the time at the voting booth. The application has different sections which has options for a login to enter the system, different parties with their names, an option to choose among them, and then to submit the entry. The user has to vote for the preferred party anonymously, but the voter's information and total votes will be stored in the text file.

The classes and their contents are as follows: -

1. **User class**
    a. Data Members
        i. username
        ii. password
        iii. userType(1 for admin, 2 for Student)
    b. Methods
        i. Constructor
        ii. Getters and setters
        iii. Login() to search the objects from the User.txt file and check whether the input username and password matches or not. If they match, the menu for the user has to be displayed. Otherwise, should ask to re-login)
        iv. save() to save a new user object to the file
        v. printData() to print the details of the user object

2. **Student Class**
    a. Data members
        i. User
        ii. Name
        iii. Course
    b. Methods
        i. Constructor
        ii. printDetails() to print the details of the Student
        iii. create Student() to add new Student object in to the txt file for Students
        iv. searchStudent() to search and display the details of a particular Student.

3. **Admin Class**
    a. Data Members

        i. User
        ii. Name
    b. Methods
        i. Constructor
        ii. printDetails() to print the details of the admin

4. **Candidate Class**
    a. Data Members
        i. PartyName
        ii. Student
    b. Methods
        i. Constructor
        ii. saveCandidate() to save the details of the candidate in the text file.
        iii. printData() to display the details of a candidate

5. **Vote Class**
    a. Data Members
        i. Student
        ii. Candidate
    b. Methods
        i. Constructor
        ii. saveVote() to save the vote to a text file
        iii. sum() to calculate and print the total vote received by a particular candidate

Write a main function, which will create an object of admin user and save in the txt file of admin. Then, ask the user to login with user name and password. Check for the successful login and allow to proceed with the menu. If login fails, ask the user to try login again.

Menu for different types of users are as follows:-

Admin:-

1. Add Student
2. Add Candidate
3. Search A Student
4. Search A Candidate
5. Print Details of all Students
6. Find the election result
7. Exit

Customer:-

1. See all Candidates
2. Vote for a Candidate
3. Exit