**Khalifa University**

**Electrical and Computer Engineering**

**ECE230 – Object Oriented Programming**

**Course Project- Fall 2022**

**JAVA Application: Banking Accounts Management System**

**Demo and Report Submission Deadline: December 5, 2022**

Banking Accounts Management System is a software product designed to improve the quality of banking service for the individual customers in the ABBank. This application will not replace any traditional Bank operation but it works along with it for better experience.

Every customer has a unique bank account number; however, he/she may open multiple saving/current accounts. In contrast, bankers have only one account to serve the customers.

Write a Banking Accounts Management System java application (ABBanking) using object-oriented programming. The system should be simple and provide authenticated access to accounts. It should always be scalable to grow with changing system requirements. The design specifications of the application are left open for each team to decide on but each must satisfy the following minimum requirements:

**Functionality Minimum requirements:**

**Banker:**

* Define the Bank and the banker details.
* Banker should be able to login only if he provided the correct credentials.
  + Banker account : User name : banker password: banker1234
* Your application should allow the bankers to open/withdraw a new bank account for customers of two different types (Current and/or saving account).
* Bankers should be able to view all the details and transaction history of all the accounts and customers registered.
* Banker should be able to **search** for an account by name or account number
* Bankers should approve or reject customers’ requests like:
  + Opening a new saving account.
  + Closing Saving account.
  + Loans requests (based on Eligibility).
  + …etc

“**Note: All requests should be displayed in an isolated window, “Notification Window”, where the new requests (Not seen) are highlighted in red color. The banker can access details of a request by clicking on it. Make sure to remove highlights from notifications that were already seen.”**

**Customer:**

* New customers (With no account) can submit a request to the banker to open a bank account from a sign up page in the first page of the Application.
* The sign up form should include the customer details like name, date of birth, salary details, salary certificate, account type (Current/Saving), email, phone number, ..etc. The request should be sent to the banker for approval/rejection.
* Registered customers should use login information (bank account number and pin assigned by the banker) to access the application.
* Registered bank customers should be able to do the following:
  + View all his/her accounts details.
  + Check accounts activities for a certain period.
  + View e-statement of an account activity for 1 month, 2 months, or 3 months.
  + Transfer money to utility (Water and electricity, Etisalat, du,..etc)
  + Pay fees of university.
  + Transfer money to another account in the same bank
  + Transfer money between his/her accounts.
  + Receive money from another account in the same bank. A notification is to be received to inform about the deposited amount.
  + Request a loan.
  + Receive money as interest in saving account (make it every 2 minutes for testing purposes). A notification is to be received to inform about the deposited amount and approvals/rejections of his/her requests. i.e. loan requests.
* A “Notification Window” where the user will receive notifications of having deposit/withdraw actions on any of his accounts as well as notifications of requests that have been sent to the banker such as loan request, open additional account request, ..etc.

“**Note: All notifications should be displayed in an isolated window, “Notification Window”, where the new notifications (Not seen) are highlighted in red color. The customer can access details of a notification (If needed) by clicking on it. Make sure to remove highlights from notifications that were already seen.”**

**Database(Files):**

* The bank database should be structured as branch of folders. Each customer should have a folder under his/her name that contains all detail about the customer, including authentication data. All transactions, additional accounts, loans, ..etc should be stored inside his/her folder where the program can access the customer’s data. All customers’ folders should be saved in a directory called “Customers”. The banker should have an isolated folder, “Bankers”, where all related data is stored.

**User Interface Design:**

* Simple, professional and clear design.
* Make your design interactive by using buttons, selection buttons, menus, lists, ..etc
* Use Tables to show the statements, customers, transactions history,…etc
* Data validation is a must in all fields (ex. Phone number format, time format, email address, names, age…etc).
* Use calendar for entering the dates.

**Testing Application:**

* Before the demonstration you need to prepare enough data permanently stored in your system like:
  + Banker account
  + Create accounts for different utilities, University, 5 customers.
  + Two accounts (1 saving and 1 current) for yourself.
  + Some interactions in your statements from previous time.

**Advanced Tasks for Bonus Marks (maximum bonus is 10%):**

* Use Android Studio instead of Eclipse for designing your application
* Send Email/sms or real push notifications to the bank customer to inform any transaction or change in the account.
* Build a Database instead of files.

**Grading scheme:**

* Interactive Interface design of the application (where menu, drop lists, buttons, date pickers..etc) are used whenever applicable. (25**%)**
* Functionality of the application. Make sure that all the above functionalities are included. Grading of this part will be from the project’s demonstration. So make sure that the demo is showing all functionalities. **(50%)**
* Usage of Classes Inheritance, interface, polymorphism, arrays, vectors,…etc., as much as applicable (at least 4). **(10%)**
* Class Diagram, report write up and screenshots. **(15%)**
* Penalty of -20% will be applied for every day late submission.
* Submissions will be on Blackboard.

**Group Size:**

* Group of 4 students registered by **Monday 14th of November 2022**. Names should be written in a google sheet shared below:

<https://docs.google.com/spreadsheets/d/1Z3lnrWtzKKqO7ZEcDy3nec02PE8M8SSUV1WnwE1hdjQ/edit?usp=sharing>

* All members of each team must be from the same Lab section.

**Project Deliverables and Dates:**

• **5/12/2022**: Submission of a zip folder containing all the project files as well as a formal report (containing introduction ,class diagram, source codes and screenshots of the application in addition to description of each part in the code, conclusion).

• **5-8/12/2022 -**10 minutes demo followed by a 5 min Q&A session will be allocated for each team to evaluate their project by Instructor/ Lab Instructors/ Teaching Assistants.

Demos will be conducted during the lab time.