**Sacco Performance Measurement and monitoring**

Uprise Sacco has been operational for more than 1 year. The Sacco members save and can request for loans, which they pay back within a specific period of time as per the memorandum that governs the Sacco. Currently, the sacco is struggling with keeping track of its performance and as such, it has decided to acquire a system to help them achieve the objective. The system is expected to do the following:

Using a command line interface, also known as the uprise client, members will be able to do the following

* Log into the system using the command   
  **login** *username password*
* Enter and submit their deposits by using command ‘  
  **deposit** *amount date\_deposited* *receipt\_number*
* Check their statement using command   
  **CheckStatement** *dateFrom DateTo*
* Request for a loan using the command below. The member sees the loan application number, generated by the server for further reference  
  **requestLoan** *amount paymentPeriod\_in\_months*
* Checking the status of a loan **LoanRequestStatus** *loan\_application\_number*
* If the **LoanRequestStatus** is granted, thedetails are displayed for member to accept or reject the loan by typing accept or reject. Once accepted, the loan is registered by the server as taken and expected installments and dates indicated.
* Any other command, which may be necessary can be added
* **Please do not substitute the commands with entering menu number and ensure that the menu is provided each time there is a response so that members don’t need to recall**

All requests are sent to the command line server, which receives all requests from the command line client. The server responds to the requests as below:

* If the username and password are both correct when a member wishes to login, the rest of the secure menu items are provided for the member to proceed else, the system will request the member to supply their member number and phone number registered. If there is a match, the member receives the password to use. Else, they are requested to return after a day, after the web system administrator has confirmed what the problem may be. The member is then supplied a reference number to use the next time they access the system such that it is used to follow up the submitted claim.
* On depositing, the server receives the message, confirms if there is a corresponding receipt number in he database and then if there is one, gives an acknowledgement to the member. Else, the member is requested to check later after new information is uploaded
* If checkstatement command, the server extracts the details and sends them to the client – organized by date and classified under loan status and Contribution Status. At the end of the statement, the performance of the member is given in percentage. Guided by the following:-
  + Percentage loan progress = (months cleared since loan repayment start period / total expected months to be cleared\*100);  
    eg. if the member has gone 20 months into the loan period and has only paid for 12 months, then   
    ***percentage loan progress = 12/20 \*100 = 60%***All members whose progress is less than 50 should see the warning as soon as they login
  + Percentage contribution progress is also calculated using a similar formula- replacing the loan period with contribution period
  + Total performance of the whole sacco, which gives the average of the entire sacco team is also given to each member
* The **requestLoan** command is received by the server, compiled until 10 applicants are received. It then uses the following to decide who to give and how much to give: -
  + Availability of funds – the Sacco will not give out loans if it has less than shs 2,000,000
  + The number of members who requested for loans
  + Members whose previous loan performance went below 50% in the previous term are given less priority
  + It is recommended that the available funds be shared among those who request equitably such that those who contribute the most to the Sacco receive a better amount.
  + After receiving the 10 applicants, the server automatically provides a recommended list of loan amount distribution and inserts it in the database. This list should first be approved by the systems administrator before the requesting members can see the output.
  + The server should provide the updates of the loan request process if the member was rejected immediately the member logs in.
  + If a member rejects a loan, the funds should be automatically reallocated to other members. The loan will only be registered if the member agrees to take the loan.
  + A member cannot take out a loan that is over ¾ of their total contributions

The web system does the following: -

* The web system uploads a list of the available deposits using a CSV file assumed to be provided by the bank. The file includes fields such as *member number, amount deposited, date, receipt number.* Also, at the beginning, the deposits that were made before the system was designed are uploaded in a CSV file.
* In case of pending requests such as login, deposits, etc. from members not previously addressed, the systems administrator is able to see on a dashboard new requests to enable them attend to them. For instance, a member or their deposit may not have been uploaded to the system when available. If it is the case, the member details are uploaded
* References that are not addressed within a period of 5 hours are given a red star against them.
* Use graphs and PDF reports to show an assessment of the sacco – The more elaborate, the better
* At the top of the hour (for coursework reasons), send an email of the report on performance of the sacco to all active members. An active member is one who has deposited funds for the last 6 months.

Using the above problem, you are required to do the following:

1. Write a design document and submit it on muele by **Deadline 3rd July 2023. If you fail to submit on muele, kindly don’t proceed with the project. Don’t submit any work to my email**
2. Implement the system components
   1. Client (java )
   2. Server (java)
   3. Web interface (PHP, Laravel framework )- design good logo for your sacco
3. Submit the code on git, - each member will be assessed using what they commit
4. Present the work (**23rd and 25th August 2023)** – marks will only be awarded to only those who present their projects. **If you fail to present your project, kindly don’t request for marks**

For inquiries, send an email to [mnsabagwa@cit.ac.ug](mailto:mnsabagwa@cit.ac.ug)