**Project phase 2 implementation direction**

**(50 marks)**

Please consider the following bulleted list under each step as a requirement to be delivered.

**Step 1.** **Implementing different functionalities using stored procedure and user defined function. (15 points)**

* At least 2 stored procedures. One of the stored procedures should be implemented with argument passing. The other stored procedure must be handling an exception.
* At least 1 user defined function

**Step 2**. **Create different set of triggers (minimum 2 numbers) to monitor the different DML and DDL activates in the database** **(10 point)**

**Some examples**

* Create trigger that report a message/audit entry during a new customer registration and new account creation
* Create trigger that report a message/audit entry that confirm loan payment is made
* Create trigger that report data update during transaction performance on saving or checking account
* Create trigger that report data read happened from a table of your choice

**Step 3** **Create index based on frequently used attribute for three of any table** **(10 point)**

* Replace the default cluster index with non-key attribute for one table. If you used GUI please provide details on which table you implemented it
* Create Composite clustered index for one of the table by removing the default clustered index. . If you used GUI please provide details on which table you implemented it
* Create non clustered composite index for one of the table you have. If you used GUI please provide details on which table you implemented it

**Step 4**. **Create different level of users and assign appropriate privilege. (10 point)**

A minimum of 2 user should be there.

* For instance, customer can read transactions, but they can’t update or delete. Accountant can read, edit or delete records etc. Create a user as **customer\_yourID** and password **customer**. When you login with this account you should be able to read and write only on selected tables that are related to customer such as customer, account, loan and payment tables. Provide testing query script after you enforced the privileges.
* Create a user as **accountant\_yourID** and password **accountant**. When you login with this account you should be able read all tables but cannot update account, payment and loan tables. Provide testing query script after you enforced the privileges.

**Step 5**: **Recovery Model and Backup ( 5 point)**

Set the recovery model for your database as full recovery model. Take a full backup of your database.

***Deliverables:*** A .bak file of the DB, that includes the database scripts/stored procedures/functions/triggers etc.

Provide step by step compiled screenshot of your work, for any feature implemented with GUI (If you created the users and assigned the privilege from GUI, or set the recovery and backup using GUI)