Title: ABABIL.INV.03.03 - REPAYMENT FOR MARKUP BASE ACCOUNT USING Sub-GL(Sub-type ADJUSTABLE_DEBIT)

Scope: The purpose of this test is to verify the functionality of repayment for a mark-up based Investment account using SUb-GL(Sub-type ADJUSTABLE_DEBIT).

Test Environment

Name Of The Web Based Application Under Test: ABABIL

Nature Of The Web Based Application Under Test: Core Islamic Banking Software Solutions

Name Of The Web Browser: CHROME (Current Version), MOZILLA FIREFOX (Current Version)

Name Of The OS: Windows 10

Test Procedure and Verification

Explanation of terms:

S<number>: stands for identification of a test procedure step.

V<number>: stands for identification for the corresponding verification(s).

V<number>: N/A stands for verification is not required for this step.

Input/Output Dataset Information:

The input/output datasets referenced in this test case are stored, by the test case name, in [Storage Name], under "[Storage Location Path]".

The tools referenced in this test case are stored, by tool classification, in [Storage Name], under "[Storage Location Path]".

[Detail Run]

S1: Launch, Login And Navigate To Ababil → **General Ledger** Module:

Logon to Ababil application.

Proceed for Day End till the first installment date.

V1: Once the Ababil home page appears, click on the icon of the "**General Ledger**" module. Then click the main menu, and select **GL Activity Mapping.**

Select activity as **Repayment** and select a GL number from the list. Copy the GL account number.

Take a screenshot of the page and save to the designated storage for record keeping. [Screenshot 01]

S2: Find the Sub_GL for the selected GL.

Search the own branch Sub-GL from **GL Account.**

V2: Verify that the **GL Account Tree View** label appears as per expectation. Search GL by mapped GL account number that is stored previously.

Find the Sub_GL by clicking on that GL and select a Sub_GL which **Sub_Type** is **ADJUSTABLE_DEBIT**.

Store the sub_GL for repayment.

Take a screenshot of the page and save to the designated storage for record keeping. [Screenshot 02]

S3: Credit the repayment amount to the sub_GL from **Cash.**

Go to **Cash** module > **Cash Deposit**

Provide the account number, amount and all other mandatory fields data and proceed to deposit.

V3: Verify that a success message will appear as "**Cash Received Successfully.**"

Store the voucher number.

S4: Search MARK-UP based account.

Go to **Account** UI.

V4: Filter by account Status **Activated.**

Select an account number for repayment.

Take a screenshot of the page and save to the designated storage for record keeping. [Screenshot 03]

S5: Select **Repayment**

V5: Verify that "**Financing Repayment"** appears at the top of the page.

Provide the account number. Also verify that **Account title, Classification status, Account detail, Rate of return, Currency, Open date, Expiry date** & **EMI Size** fields are fetched by data automatically.

Take a screenshot of the page and save to the designated storage for record keeping. [Screenshot 04]

S6: Repayment of an EMI from Sub-GL providing **Reference number**.

Provide a comparatively larger amount than the voucher's adjustable amount.

V6: Verify that a pop up will appear as "**Please generate repayment schedule". (If overdue payment)**

Also verify that "**Financing account schedule created successfully**" appears as a pop up for successful Repayment Schedule Generation.

Provide the Sub_GL .

Verify that a new dropdown parameter appears named as **Reference Number** and the same voucher number appeared for cash deposit is showing in the dropdown.

Verify that a pop up will appear as "**Insufficient balance**"

Take a screenshot of the page and save to the designated storage for record keeping. [Screenshot 05]

S7: Insert the voucher's adjustable amount as repayment amount.

V7: Verify that a pop up will appear as "**Please generate repayment schedule". (If overdue payment)**

Also verify that "**Financing account schedule created successfully**" appears as a pop up for successful Repayment Schedule Generation.

Provide the Sub_GL .

Verify that a new dropdown parameter appears named as **Reference Number** and the same voucher number appeared for cash deposit is showing in the dropdown.

Take a screenshot of the page and save to the designated storage for record keeping. [Screenshot 06]

S8: Click **Submit**

V8: Verify that a pop up will appear after clicking **Submit** that holds the label **Confirmation.** After selecting **Yes,** verify that a pop up appears holding the message "**Financing account repayment done successfully**"

Save the voucher number.

Take a screenshot of the page and save to the designated storage for record keeping. [Screenshot 07]

S9: Click **Submit**.

V9: Verify that a pop up will appear as **Task sent for verification.**

Save the task ID.

Take a screenshot of the page and save to the designated storage for record keeping. [Screenshot 08]

S10: Verify and Accept From My Task:

If the task is sent to verify then login as a verified user and verify the required details. Then click on the accept button.

V10: Verify that the success popup message will be displayed "**Repayment saved with voucher number**".

Save the Voucher ID.

Take a screenshot of the pages and save to the designated storage for record keeping. [Screenshot 08]

S11: Risk and Hazard:

Throughout this testing, make sure all labeling, including messages, icons and messages of operation guidelines are accurate, written in short concise sentences, and written in simple and familiar words.

V11: Verify following items wherever appropriate:

Throughout this testing verify the AUT based on the following viewpoints:

i) Make sure that the user interface is simple, easy to understand and screen designs are clear, concise, consistent, complete and unambiguous.

ii) Make sure that the abbreviations, symbols, text and acronyms placed on or displayed by the AUT are consistent and unambiguous.

iii) Make sure that the AUT provides immediate and clear feedback following user entries, whenever necessary.

iv) Make sure that the operation steps are easy-to-remember.

v) Make sure that the prompts, menus, etc. are used to cue the user regarding important steps.

vi) Make sure that the AUT does not hang during run time or "strand" the user.

vii) Make sure that the AUT provides the users useful information in the case of an error. Make sure that the AUT provides conspicuous mechanisms for correction and troubleshooting guidance.

viii) Make sure that the AUT does not overload or confuse the users with information that is unformatted, densely packed or presented too briefly.

ix) Make sure that the use of symbols, icons, colors and abbreviations are acceptable to convey information reliably, precisely and quickly.

x) Make sure that dedicated display mechanisms are used for highly critical and time sensitive information.