SYNERGY
GLOBAL FINANCIAL SOLUTIONS

EBank backend processes

1. Login and Registration module

This module will be responsible for user details, including login, logout and registration. Bellow processes should be implemented:

- User should be blocked after 5 unsuccessful login attempts.
- User will have option to choose to login only from specific IP address (optional)
- Procedure for getting user details should be created
- Procedure for registration new users should be created. Validation must be added for unique user.
- Procedure for unblocking user should be created.
- There should be option to force password change - manual or system should force it 6 months after last password change.
- Log all changes made to the use
- Hints
    - Create table **sytb_user** containing details for the user:
        - user_id [number]
        - user_login_name [varhcar2]
        - password – must be hashed [varchar2]
        - email [varchar2]
        - user_status [varchar2] – E-enabled, B – blocked, BP – blocked for 5 times unsuccessful login attempts
        - date_created [date]
        - last_modification [date]
        - last_login [date]
        - last_logout [date]
        - force_password_change [varchar2]
        - ip_address_restriction [varchar2]
        - last_password_change [date]
    - Create package **sypks_user_login** with bellow public functions and procedures:
        - Procedure **pr_login** – will be called when user try to login , password should be validated. Error or success should be returned
        - Procedure **pr_logout**
        - Procedure **pr_get_user_info** – return details for the users
        - Procedure **pr_unblock_user** – unblock user after 5 unsuccessful login attempts.
        - Procedure **pr_register** – for registration f new users. Validations should be applied.
    - Scheduler Job should be created to check last password change and if it is not changed last 6 months to force it.
    - Logging changes –can be done with trigger and history table

2. Module for transactions and account balances

This module will maintain accounts and transactions in the system. Bellow processes should be implemented:

- Store accounts for each user with available balance.
- Calculate for each account monthly and yearly turnover
- Update current balance for each account after transaction. Operation should be thread synchronized in order to have correct value.
- Store transactions for each account. Two types of transaction – debit and credit. If type is debit system should give a error if there is no enough money in the account.
- Transaction cannot be modified and cannot be deleted
- Multi currency should be implemented – allow transactions from different currencies, not only account currency. Three currencies are supported – BGN , USD, EUR
- Charge module should be implemented – addition transaction for charge should be stored if there is any. Transaction can have maximum one charge or no charges.
- Hints
    - Create table for the accounts – **sytb_accounts**
        - user_id [number] - relation to table sytb_user
        - account_id[varchar2]
        - account_ccy [varchar2]
        - account_desc [varchar2]
        - ac_open_date [date]
        - account_status [varchar2] – E –enabled, F – frozen, C – closed, BD – blocked for debit, BC – blocked for credit
        - date_created [date]
        - date_last_modification [date]
        - modification_no[number]
        - acy_curr_balance[number] – available balance in account currency
        - lcy_curr_balance[number] – available balance in local currency
        - acy_month_turnover_cr[number] – credit turnover for current month in account currency
        - acy_month_turnover_dr[number] – debit turnover for current month in account currency
        - acy_year_turnover_cr[number] – credit turnover for current year in account currency
        - acy_year_turnover_dr[number] – debit turnover for current year in account curency
        - date_last_transaction[date]
        - iban_ac_no[varchar2]
    - Create package **sypks_accounts**
        - Procedure for creating account
        - Procedure for account modification
        - Scheduler Job to reset monthly and yearly turnovers for each account
    - Create table for the transactions – **sytb_transactions**

        - account_id[varchar2] – relation to table sytb_accounts
        - trn_id[number] – unique value for each transactions
        - trn_ref_no[varchar2] – unique value for group of transactions for one operation. For example if transaction has charge, transaction and charge will ne 2 records in the table but hey will have same trn_ref_no
        - drcr_indicator[varchar2] – debit or credit indicator
        - fcy_amount [number] – transaction amount in foreign currency
        - exchange_rate[number] -exchange rate for the transaction
        - lcy_amount[number] – transaction amount in local currency
        - trn_date [date] – date of transaction
        - trn_type[varhcar2] – T - real transaction or C - charge
        - trn_description[varchar2] – transaction description
        - charge_product [varchar2] – related to charges
    - Create table for the charges – **sytb_charge** (maintenance table, should be populated with all type of charges initially)
        - Charge_product[varchar2] – unique value for each charge
        - Charge_amt_lcy [number]– amount for the charge in local currency
        - Charge_descr [varchar2] – description of the charge
    - Create package **sypks_transactions**
        - procedure for inserting new transaction. Internally this transaction should update account balances and turnovers. Internally this procedure should check for the charge and insert one more record in the transaction table. Procedure should validate for available balance in the account for debit transactions.
        - Internal Procedure for checking account balance.
        - Internal Procedure for checking charges related to the transaction
        - procedure for calculating amount from one currency to another currency. Create local table for exchange rates for list of currencies – BGN, USD, EUR.

3. Module for sending mails, SMS and push notifications.