NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

DESIGN DOCUMENTATION

Bộ môn Công nghệ phần mềm
Khoa Công nghệ thông tin
Đại học Khoa học tự nhiên TP HCM

Table of Contents

DESIGN DOCUMENTATION

The document aims to focus on the following topcis:

- ✓ Create Design Documentation.
- ✓ Complete Design Documentation with following contents:
  - Conceptual Model
  - Architecture Design
  - Data Design
  - User Interface Design
- ✓ Read and understand the Design Documentation.

1 Member Evaluation Table

StudentID	Name	Contribution (%)	Sign
22127060	Lê Hoàng Đạt	25	Đạt
22127088	Phạm Quang Duy	25	Duy
22127270	Nguyễn Quang Minh	25	Minh
22127389	Nguyễn Phúc Thành	25	Thành

2 Conceptual Model

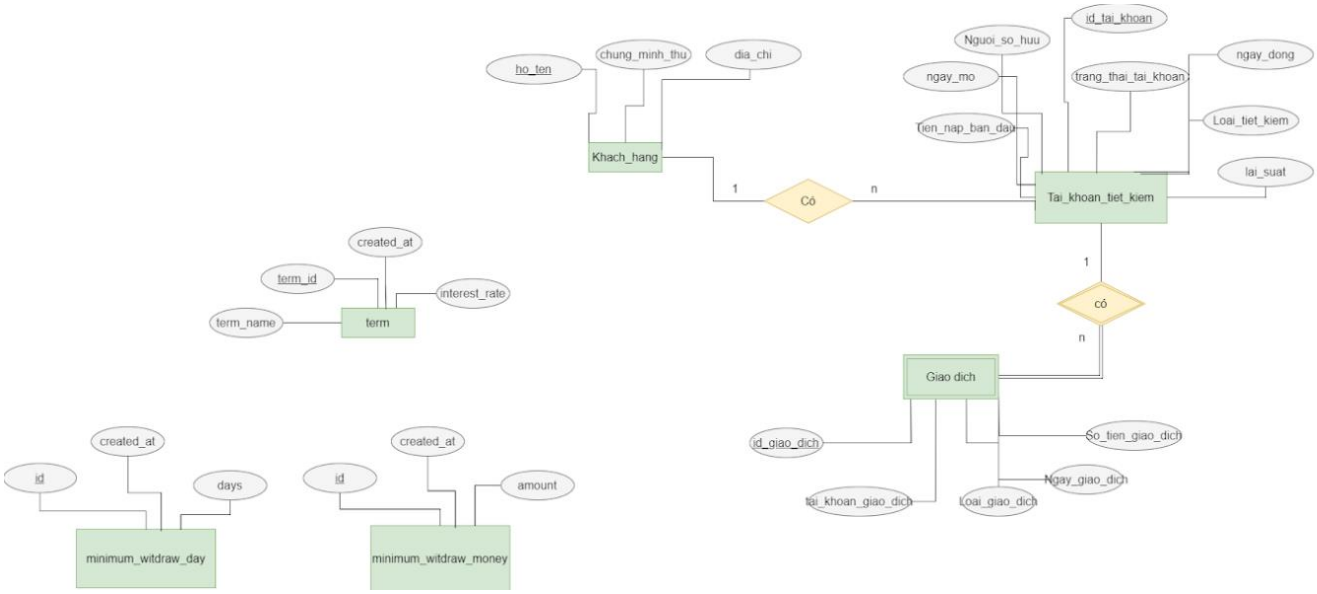


Figure 1 Link To View Detail Image

3 Architecture Design

3.1 Architectural Diagram:

3.1.1 System Decomposition Tree:

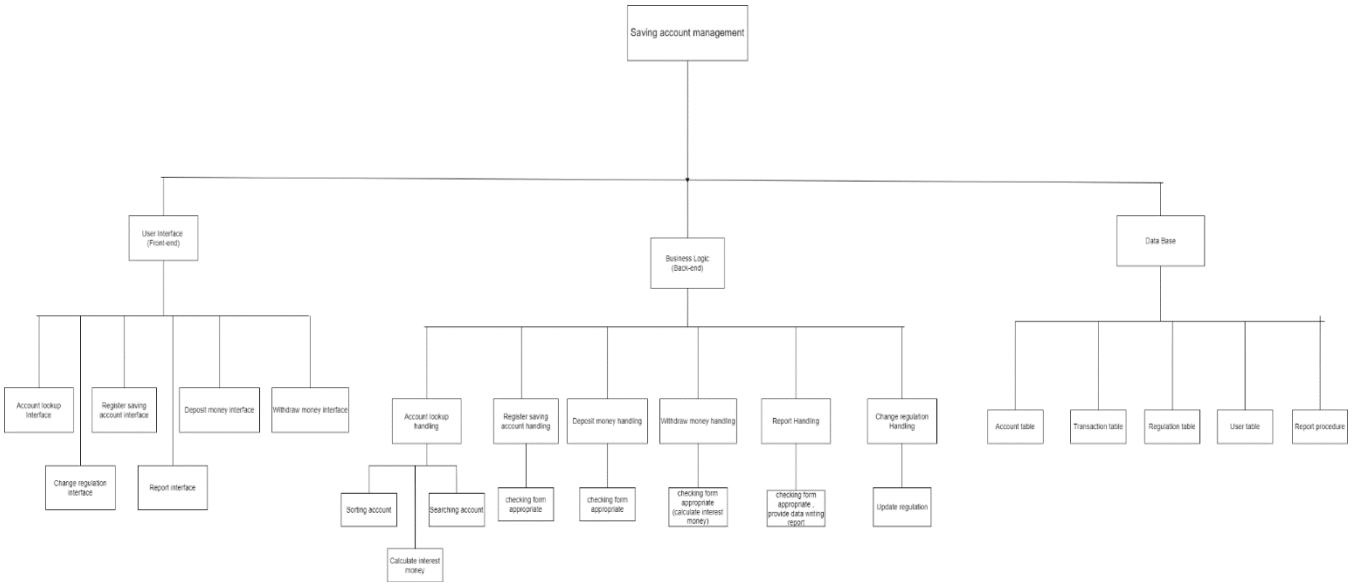
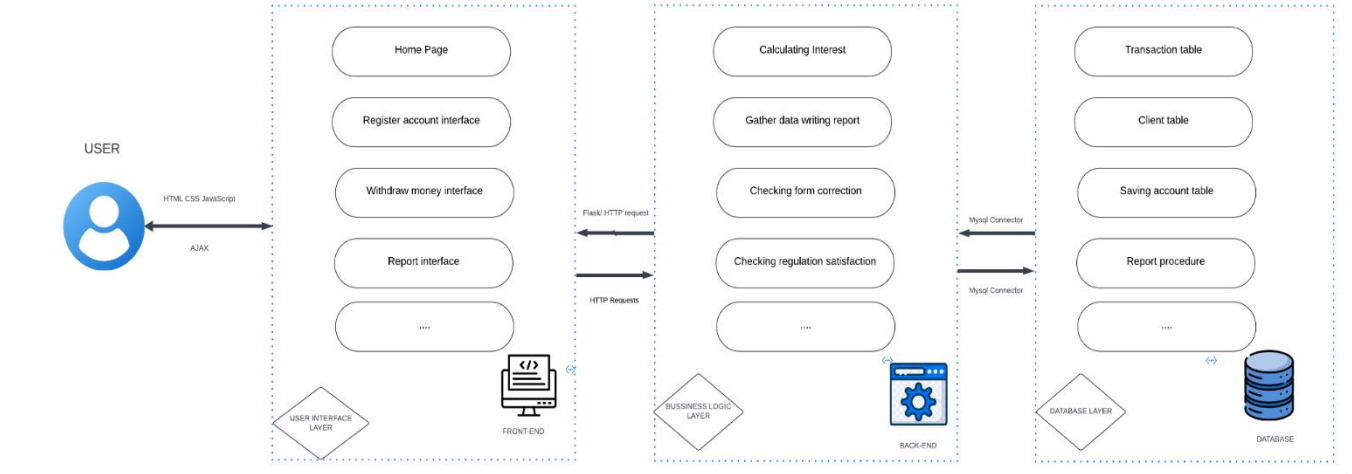


Figure 2 Link To View Detail Image

3.1.2 Overall architectural model:



3.1.3 Detail system operation:

1. User Interaction with the User Interface (Front-end):

- User (User): Users interact with the application through a web interface, choosing a kind of form on the home page.
- Provide early checking in this phase for basic errors like: number in name, day created in the past, amount of money not meeting minimum money in regulation,...

2. Business Logic Processing (Back-end):

- Flask HTTP Requests: The back-end is built on Flask (a Python framework) and receives HTTP requests from the front-end of a specific form. These requests include registering new accounts, withdrawing money, and generating reports, ...
- Business Logic Processing(Late checking): Once the back-end receives a request, it processes these requests, including calculating interest, and ensuring regulations are complied, withdrawing money needed to match the amount of money in account, withdrawal day needed to satisfy minimum withdrawal day, etc... .

3. Communication with the Database (Database):

- MySQL Connector: The back-end uses MySQL Connector to interact with the database. Database queries are executed to add, update, or retrieve data related to accounts, transactions, and reports.
- Database Tables: The database includes tables for transactions, customers, savings accounts, and reporting procedures.

4. Data Flow and Responses:

- Response from Server: After processing the requests, the back-end sends responses back to the front-end, which may be the requested data, confirmation of actions taken, or error messages if any.
- Updating the User Interface: The front-end updates the user interface based on the server's response, displaying new or updated information to the user.

3.1.4 Special mechanisms:

1. Application of Design Patterns:

- **Singleton Pattern**:

  - Purpose: To ensure that only one instance of the database connection is created throughout the system's operation. This helps manage resources efficiently and avoids the wastage associated with creating multiple connections simultaneously.
  - Application in the System: In the database management module, the Singleton will be used to create and maintain a connection to the MySQL database. This ensures that all database queries use the same connection, guaranteeing consistency and efficiency.
- **Factory Pattern**:
  - Purpose: To create an interface for generating objects within the system, allowing subclasses to decide which class will be instantiated. This design pattern makes the system more flexible in terms of expansion without needing to modify existing code.
  - Application in the System: Applied in creating different types of transactions such as deposits and withdrawals. The Factory Pattern manages the instantiation of transaction objects based on the type of transaction requested, thereby supporting easy expansion when new transaction types are added.
- **Decorator Pattern**:
  - Purpose: To add functionality to an object without changing the existing code of the classes. This makes the code more flexible for expansion.
  - Application in the System: Can be applied to add features such as logging or data security for transactions. For example, when a transaction is performed, a Decorator can automatically add a detailed logging feature without the need to modify the original transaction class's code.

2. Client-Server:

- **Model-View-Controller (MVC):**
  - Purpose: To separate the application's processing logic (Model), the user interface (View), and the control (Controller, BE). This reduces dependencies between the business processing part and the user interface, making the application easier to manage and scale.

  - Application in the System:
  - Model: Manages data, logic, and rules of the application.
  - View: Displays data (user interface) provided by the Model and sends user events (like keystrokes, mouse clicks) to the Controller.
  - Controller: Handles input events from the View, translating them into actions to be performed by the Model in .
- **Drawing wide range of Diagram using google API:**
  - Purpose: Provide an alternate way for bankers to view the many complex statistical and analysis data from the report.
  - Application: Provide option to choose between table or diagram in report form.

3.2 Class Diagram:

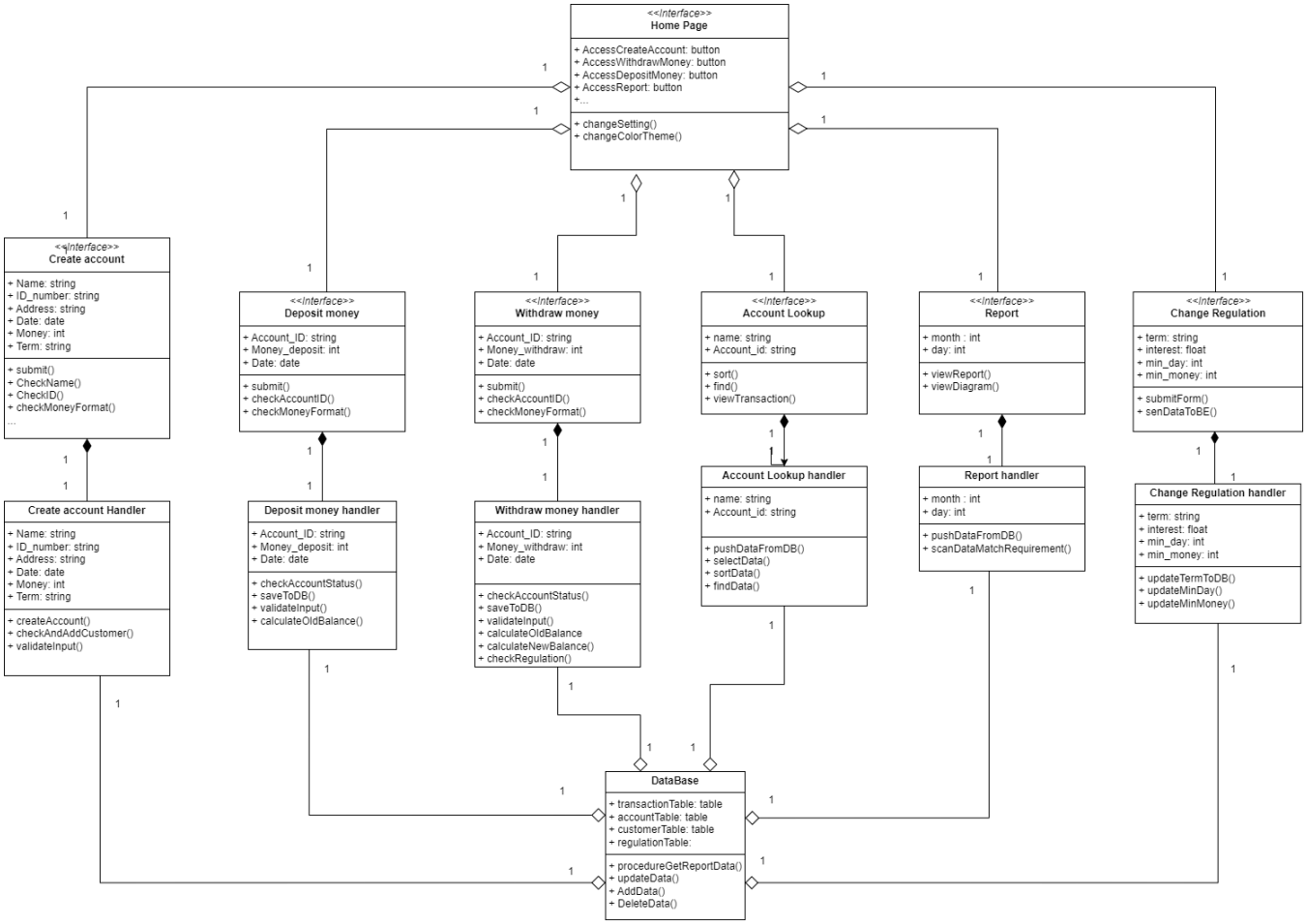


Figure 3 Link To View Detail Image

3.3 Specifying object classes:

3.3.1 Class Deposit money handler:

The DepositMoneyHandler class has a composition relationship with the DepositMoney class because the primary purpose of the BE is to transmit data to the FE.

Order	Attribute	Type	Constraints	Meaning
1	Account_ID	protected	Not Null, Have to be Valid account_id	Attribute used to identify the account performing the transaction
2	Money_deposit	protected	Not Null, Not negative	This attribute represents the amount of money involved in the transaction
3	Date	protected	Not Null, Valid Day	This attribute is used to represent the transaction date

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	check_account_status	protected	No Constraint	This method is used to check whether an account is still valid or not, preventing deposit from a disabled account.
2	CalculateOldBalance	protected	No Constraint	Calculate the remaining balance

				after previous transactions.
3	validate_input	protected	No Constraint	This method is used to check whether inputs are correct, valid.
4	validate_deposit_conditions	protected	No Constraint	This method is used to check whether deposit conditions are passed: term, amount.
5	SaveToDB	protected	# Check if the sending date is valid # Check the deposit conditions # Check if it is a non-term ledger # Check if the deposit amount meets the minimum required amount	Save the deposit transaction to the database for easy review of the history and to calculate the current amount in the non-term account held

3.3.2 Class Withdraw money handle:

The WithdrawMoneyHandler class has a composition relationship with the WithdrawMoney class because the primary purpose of the BE is to transmit data to the FE.

Order	Attribute	Type	Constraints	Meaning
1	Account_ID	protected	ID have to exist Account on active	Attribute used to identify the account performing the transaction
2	Money_withdraw	protected	Have to smaller than remaining money in account	This attribute represents the amount of money involved in the transaction
3	Date	protected	Have to be valid date	This attribute is used to represent the transaction date

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	calculateOldBalance	protected	#Check if the date is valid #Check if the ID is valid #Check the account status #Check the type of savings #Check the 15-day condition #Check the interest rate	Calculates the old balance by adjusting the initial deposit with the total deposits and withdrawals, and then applies interest rate adjustments

				based on the specified interest_rate, expired_time, and month.
2	validate_input	protected	No Constraint	This method is used to check whether inputs are correct, valid.
3	CheckRegulation	protected		This method is used to check whether withdraw conditions are passed: amount, date, term-perriod , 15-days conditions.

3.3.3 Class CreateAccountHandler:

The CreateAccountHandler class has a composition relationship with the CreateAccount class because the primary purpose of the BE is to transmit data to the FE.

Order	Attribute	Type	Constraints	Meaning
1	Name	protected	Customer information have to match the current data (if this customer already in data)	Attribute used to present account_user's name
2	Date	protected	The opening date cannot be later than the current date	This attribute is used to represent the create date
3	Money	protected	Bigger than minimum deposit amount is	This attribute is used to represent the first money send to account
4	Term	protected	Term have to be valid term	Indicate term of the account

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	CheckAndAddCustomer	protected	#Check if the date is valid #First deposit money is valid #Check the valid term	This function is used to check whether the current customer is already in the dataset. If so, it verifies if the provided customer information is accurate; if not, it adds the new customer to the dataset

2	CreateAccount	protected	#Check if the date is valid #First deposit money is valid #Check the valid term #Check if customer's information is valid	A function to add the customer's and account's data into the database
3	validate_input	protected	No Constraint	This method is used to check whether inputs are correct, valid.

3.3.4 Class HomePage:

The class does not associate or compose with other classes because it operates independently and is not affected when other classes are deleted.

Order	Attribute	Type	Constraints	Meaning
1	AccessCreateAccount	protected	No ConStraint	Button to get to class CreateAccount
2	AccessWithdrawMoney	protected	No Constraint	Button to get to class WithdrawMoney
3	AccessDepositMoney	protected	No Constraint	Button to get to class DepositMoney
4	AccessReport	protected	No Constraint	Button to get to class Report

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	AccessToForm	portected	No Constraint	Press The Button to Access Another Class

3.3.5 Class CreateAccount:

Order	Attribute	Type	Constraints	Meaning
1	Name	protected	Only letters are accepted and numbers are not allowed	The information is stored to be sent down to the BE processing section to create an account Button to get to class WithdrawMoney Button to get to class DepositMoney Button to get to class Report
2	Id_Number	protected	Have to be in right format	
3	Address	protected	No Constraint	
4	Date	protected	No Constraint	
5	Money	protected	Have to be formated	
6	Term	protected	No ConStraint	

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	Submit	portected	No Constraint	Pass the values and attributes to the CreateAccountHandler class.
2	CheckName	protected	No Constraint	Check if the name consists of only letters and contains no numbers
3	CheckID	protected	No Constraint	Check if the ID consists of only digit
4	CheckMoneyFormat	protected	No Constraint	Check if Money input is format

3.3.6 Class DepositMoney:

Order	Attribute	Type	Constraints	Meaning
1	Account_ID	protected	Only letters are accepted and numbers are not allowed	The information is stored to be sent down to the BE processing section to create an account Button to get to class WithdrawMoney Button to get to class DepositMoney Button to get to class Report
2	Money_Deposit	protected	Only digits are accepted	
3	Date	protected	No Constraint	

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	Submit	portected	No Constraint	Pass the values and attributes to the DepositMoneyHandler class.
3	CheckAccountID	protected	No Constraint	Check if the ID consists of only digit
4	CheckMoneyFormat	protected	No Constraint	Check if Money input is format

3.3.7 Class WithdrawMoney:

Order	Attribute	Type	Constraints	Meaning
1	Account_ID	protected	Only letters are accepted and numbers are not allowed	The information is stored to be sent down to the BE processing section to create an account Button to get to class WithdrawMoney Button to get to class DepositMoney Button to get to class Report
2	Money_Deposit	protected	Only digits are accepted	
3	Date	protected	No Constraint	

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	Submit	portected	No Constraint	Pass the values and attributes to the DepositMoneyHandler class.
3	CheckAccountID	protected	No Constraint	Check if the ID consists of only digit
4	CheckMoneyFormat	protected	No Constraint	Check if Money input is format

3.3.8 Class AccountLookUp:

Order	Attribute	Type	Constraints	Meaning
1	Account_ID	protected	Only letters are accepted and numbers are not allowed	The information is stored to be sent down to the BE processing section to create an account Button to get to class **AccountLookUpHandler** Button to get to class DepositMoney Button to get to class Report
2	Name	protected	Only letters are accepted and numbers are not allowed	

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	Sort	portected	No Constraint	To pass values into the selected AccountLookUpHandler class to sort by the selected column
2	find	protected	No Constraint	Search for a person or account based on the selected value and pass the value into the AccountLookUpHandler
3	viewTransaction	protected	No Constraint	Check if Money input is format

3.3.9 Class Report:

Order	Attribute	Type	Constraints	Meaning
1	Month	protected	No Constraint	The information is stored to be sent down to the BE processing section to create an account Button to get to class **ReportHandler** Button to get to class DepositMoney Button to get to class Report
2	Date	protected	No Constraint	

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	ViewReport	portected	No Constraint	View statistics by month or by day
2	ViewDiagram	protected	No Constraint	View charts by each type of report.

3.3.10 Class ChangeRegulation:

Order	Attribute	Type	Constraints	Meaning
1	Term	protected	The term must be one of the terms in the regulations	The information is stored to be sent down to the BE processing section to create an account Button to get to class **ReportHandler** Button to get to class DepositMoney Button to get to class Report
2	Interest	protected	Interest must be calculated for each term	
3	min_day	protected	The min_day must comply with the min_day in the regulation data.	
4	min_money	protected	The min_money must comply with the min_money in the regulation data.	

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	SubmitForm	protected	No Constraint	Send the attributes and variables down to the BE in the ChangeRegulationHandler class
2	SentDataToBE	protected	No Constraint	Search for a person or account based on the selected value and pass the value into the AccountLookUpHandler

3.3.11 Class AccountLookUpHandler:

The AccountLookUpHandler class has a composition relationship with the AccountLookUp class because the primary purpose of the BE is to transmit data to the FE

Order	Attribute	Type	Constraints	Meaning
1	Sort_colum	protected	Check if the selected column is valid; if not, sort by the default column	Used for sorting when querying the database.
2	Account_ID	protected	No Constraint	Used for searching within the database

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	PushDateFromDB	protected	No Constraint	Send a query to the database connected with the BE to retrieve information
2	selectData	protected	No Constraint	Select data based on the chosen filter
3	SortData	protected	No Constraint	Send a query to the database connected with the BE and sort the results.

3.3.12 Class Reporthandler:

The ReportHandler class has a composition relationship with the Report class because the primary purpose of the BE is to transmit data to the FE.

Order	Attribute	Type	Constraints	Meaning
1	Month	protected	No Constraint	Attach to the query to filter by month
2	Date	protected	No Constraint	Used for searching within the database

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	PushDataFromDB	protected	No Constraint	Send a query to the database connected with the BE to retrieve information
2	ScanDataMatchRequirement	protected	No Constraint	Send a query with specific condition to the database connected with the BE to retrieve information

3.3.13 Class ChangeRegulationHandler:

The ChangeRegulationHandler class has a composition relationship with the ChangeRegulation class because the primary purpose of the BE is to transmit data to the FE.

Order	Attribute	Type	Constraints	Meaning
1	term	protected	No Constraint	Store the information that needs to be changed for future updates.
2	interest	protected	No Constraint	
3	min-day	protected	No Constraint	
4	min-money	protected	No Constraint	

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	UpDateTermToDB	protected	No Constraint	Send a query to the database connected with the BE to save information Save new minday to DB Save new minMoney to DB
2	UpdateMinDay	protected	No Constraint	
3	UpdateminMoney	protected	No Constraint	

3.3.14 Class DataBase:

The DataBase class does not have associate or composite relationships with other classes because it operates independently, and other classes being affected does not impact the database.

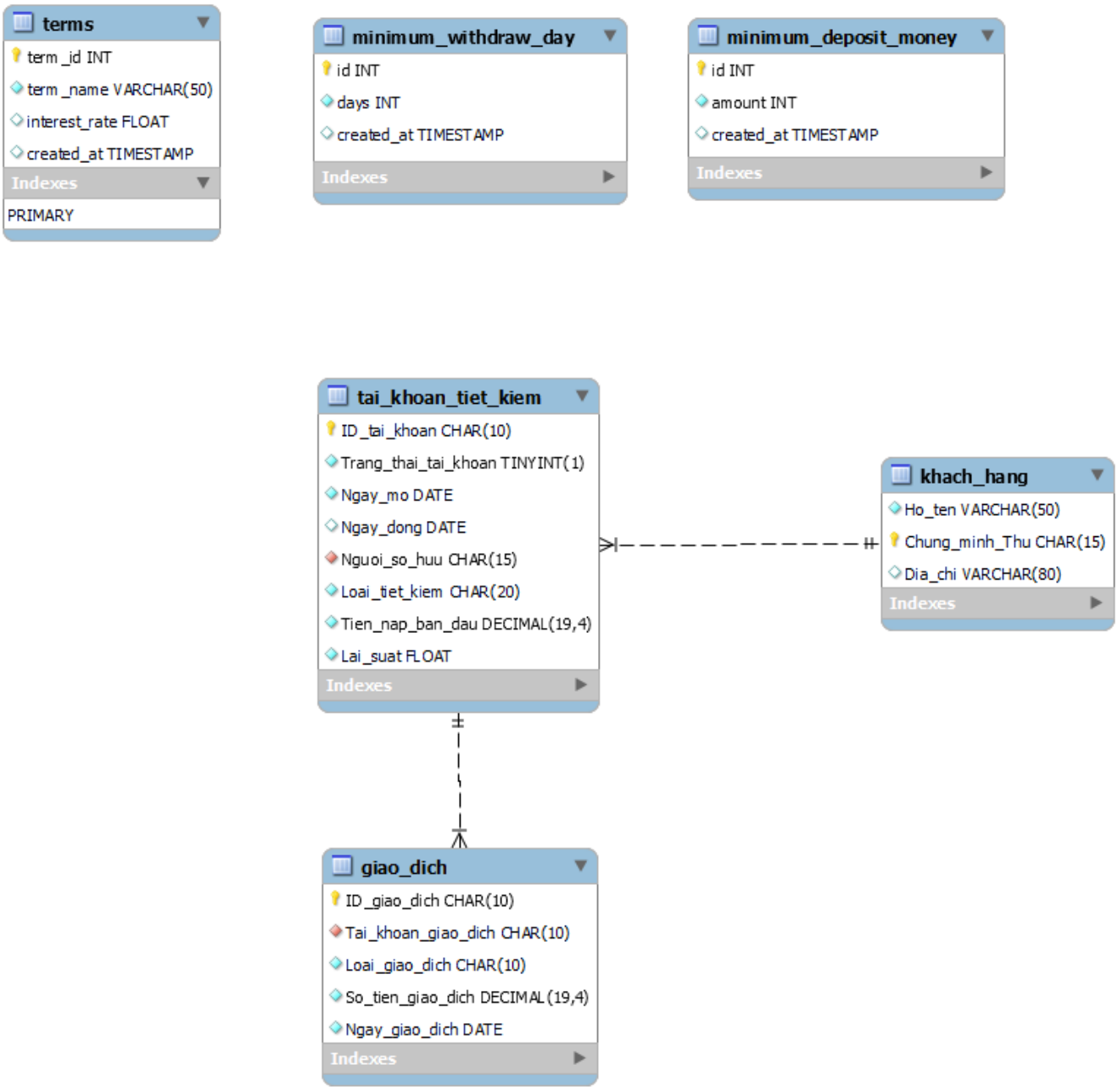
Order	Attribute	Type	Constraints	Meaning
1	transactionTable	protected	No Constraint	Store the information that needs to be changed for future updates.
2	AccountTable	protected	No Constraint	
3	customerTable	protected	No Constraint	
4	RegulationTable	protected	No Constraint	

[List of main methods]

Order	Method	Type	Constraints	Meaning
1	procedureGetReportData	portected	No Constraint	Use a procedure in SQL to retrieve data based on the procedure
2	UpdateData	protected	No Constraint	Modify the data that needs to be updated.
3	AddData	protected	No Constraint	Add new data to the database.
4	DeleteData	protected	No Constraint	Delete data in the database

4 Data design

4.1 Data diagram:



4.2 Data specification:

1. Terms:

- **Columns:**
  + term_id INT: Primary key. Unique identifier for each term.
  + term_name VARCHAR(50): Name of the term, up to 50 characters.
  + interest_rate FLOAT: Interest rate associated with the term.

  created_at TIMESTAMP: Timestamp indicating when the term was created.
- **Explanation:**
  + This table stores information about different savings terms, including their names and associated interest rates.

2. minimum_withdraw_day:

- **Columns:**
  + id INT: Primary key. Unique identifier for each record.
  + days INT: The minimum number of days required before a withdrawal can be made.
  + created_at TIMESTAMP: Timestamp indicating when the record was created.
- **Explanation:**
  + This table defines the minimum number of days a deposit must be held before it can be withdrawn.

3. minimum_deposit_money:

- **Columns:**
  + id INT: Primary key. Unique identifier for each record.
  + amount INT: The minimum deposit amount required.
  + created_at TIMESTAMP: Timestamp indicating when the record was created.
- **Explanation:**
  + This table specifies the minimum deposit amount for different accounts or terms.

4. tai_khoan_tiet_kiem (Savings Account):

- **Columns:**
  + ID_tai_khoan CHAR(10): Primary key. Unique identifier for each savings account.

- + Trang_thai_tai_khoan TINYINT(1): Account status (e.g., active or inactive).
- + Ngay_mo DATE: Account opening date.
- + Ngay_dong DATE: Account closing date.
- + Nguoi_so_huu CHAR(15): Owner's identification number, 15 characters.
- + Loai_tiet_kiem CHAR(20): Type of savings account.
- + Tien_nap_ban_dau DECIMAL(19,4): Initial deposit amount with up to 19 digits and 4 decimal places.
- + Lai_suat FLOAT: Interest rate for the savings account.

– **Explanation:**

- + This table holds information about individual savings accounts, including account details and financial information.

5. khach_hang (Customer):

– **Columns:**

- + Ho_ten VARCHAR(50): Customer's full name, up to 50 characters.
- + Chung_minh_Thu CHAR(15): Customer's ID number, 15 characters.
- + Dia_chi VARCHAR(80): Customer's address, up to 80 characters.

– **Explanation:**

- + This table contains information about customers, including their personal and contact details.

6. giao_dich (Transaction):

– **Columns:**

- + ID_giao_dich CHAR(10): Primary key. Unique identifier for each transaction.
- + Tai_khoan_giao_dich CHAR(10): Foreign key referencing tai_khoan_tiet_kiem. The savings account associated with the transaction.
- + Loai_giao_dich CHAR(10): Type of transaction (e.g., deposit, withdrawal).
- + So_tien_giao_dich DECIMAL(19,4): Transaction amount with up to 19 digits and 4 decimal places.
- + Ngay_giao_dich DATE: Date of the transaction.

– **Explanation:**

+ This table records each transaction made on savings accounts, including the type and amount of each transaction.

7. Relationships:

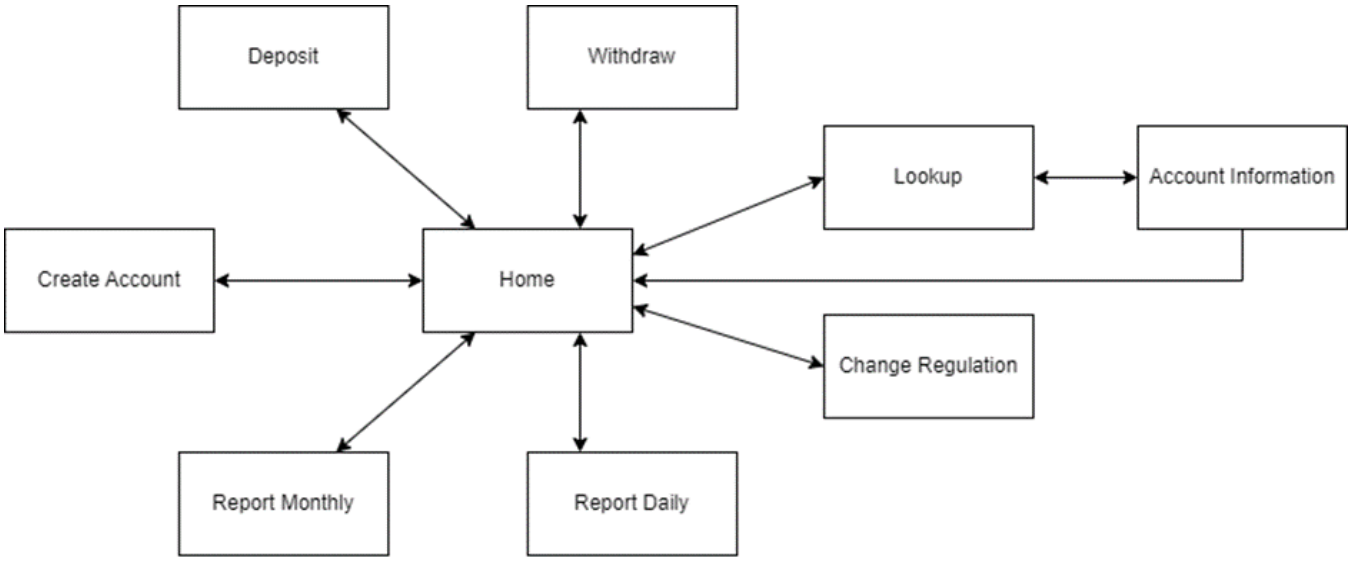
- The tai_khoan_tiet_kiem table is linked to the khach_hang table via the Nguoi_so_huu column, which holds the customer ID. One customer can have many accounts
- The giao_dich table is linked to the tai_khoan_tiet_kiem table via the Tai_khoan_giao_dich column, representing transactions associated with each savings account.

5 User interface design

5.1 Diagram and screen list:

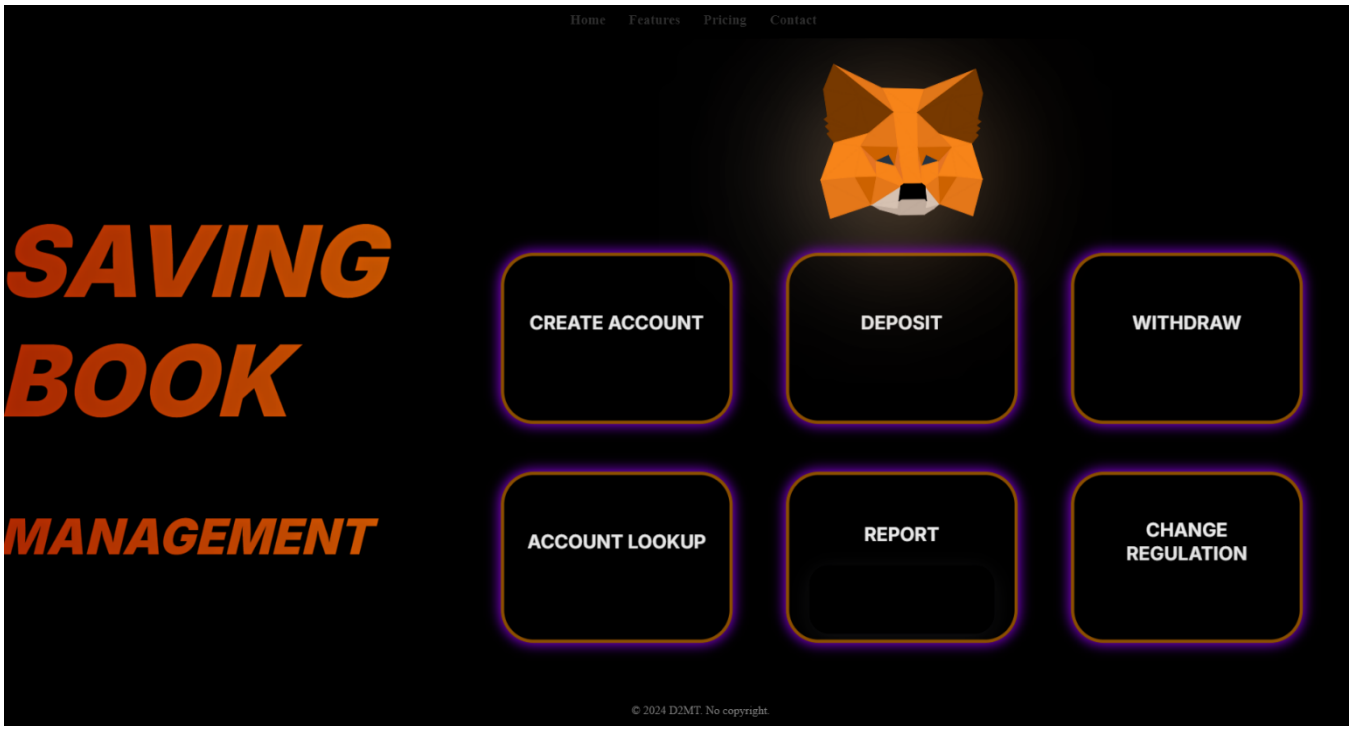
Order	Screen	Meaning
1	Home	The main screen for navigating to other functional screens.
2	Create Saving Account	Screen for creating customer accounts and savings accounts.
3	Deposit	Screen for depositing money into a savings account.
4	Withdraw	Screen for withdrawing money from a savings account.
5	Lookup Account	Screen for viewing account information.
6	Account Information	Screen for detailed transaction information of each account.
7	Report Daily	Screen for reporting total income and total expenses by day.
8	Report Monthly	Screen for reporting the number of accounts opened and closed each day of the month.
9	Change Regulation	Screen for changing the regulations.

Diagram:



5.2 Interface screen specifications:

5.2.1 Home:



1. User Interface Controls:

a. Input Controls:

– **Command:**

+ **Button:** The six rectangular buttons labeled "CREATE ACCOUNT," "DEPOSIT," "WITHDRAW," "ACCOUNT LOOKUP," "REPORT," and "CHANGE REGULATION."
+ **Link:** Each rectangular button leads to one screen that has a specific functionality as its name.

b. Output Controls:

– **Simple Output:**

+ **Label:**
  - "SAVING BOOK MANAGEMENT": The title of the page.
  - The buttons "CREATE ACCOUNT," "DEPOSIT," "WITHDRAW," "ACCOUNT LOOKUP," "REPORT," and "CHANGE REGULATION" have text labels inside them indicating their functionality.

- **Complex Output:**
  + **GridView:** Buttons navigate to website's functions.

2. User Interface Design Guidelines:

a. Color Usage:

- **Consistency:** The color palette is uniform with a dark background and alternating warm text colors (orange, purple), creating an eye-catching and cohesive design.
- **Simplicity:** The design uses a simple color palette primarily involving black, orange, and purple.
- **Do not use too many colors (4/6):** The design adheres to this guideline, using a limited set of colors.
- **Be careful of contrast colors:** The contrast between the dark background and the bright text/buttons ensures readability and visibility.

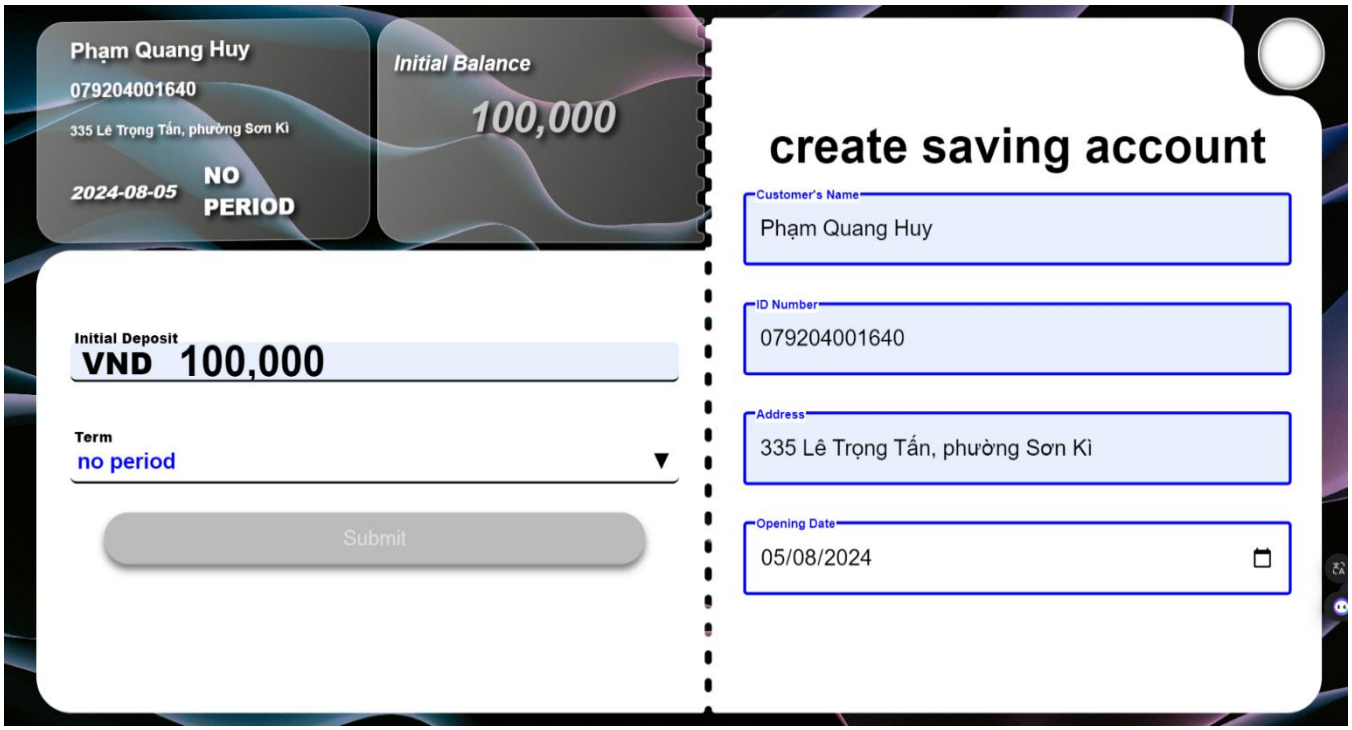
b. Message Usage

- **Consistency:** The text labels and by most users.

3. Interaction with screen:

- Select a function to navigate to the screen for performing that function.

5.2.2 Create Account:



1. User Interface Controls:

a. Input Controls:

- **Command:**
  - **Button:**
    - Gray "Submit" button at the bottom.
    - Round "Home" button at the top right corner.
  - **Link:** Linked to the Home page via the round button at the top right corner.
- **Typing:**
  - **TextBox:**
    - Input field "Customer's Name": account owner's name.
    - Input field "ID Number": customer's identification number.
    - Input field "Address": customer's address.
    - Input field "Opening Date": default value is the current date.
    - Input field "Initial Deposit" with the note "VND amount?": the initial deposit amount when opening the savings account.
  - **Selection:**

- **ListBox:** Item "Term" with the note "choose a term".

b. Output Controls:

– **Simple Output:**

+ **Label:** "Create saving account": the title of the savings account opening page.
+ **TextBox:** Two gray boxes at the top left display the customer's entered information:
  - First box: Customer information, opening date, and savings type.
  - Second box: Initial deposit amount.
+ **MessageBox:** Notification when submission is successful.

2. User Interface Design Guidelines:

a. Color Usage:

– **Consistency:** Colors are used consistently with a dark background and light-colored input forms. The dark background contrasts well with white and blue text, making information clear and readable.

– **Simplicity:** The design uses a simple color palette primarily consisting of white, black, and blue. This simplicity helps users focus on the main information.

– **Do not use too many colors (4/6):** The design follows this principle by using a limited color palette with a dark background and light text, emphasizing important information.

– **Be careful of contrast colors:** The dark background with light text enhances readability, while key details like deposit amount, customer name, address, and savings type are highlighted in blue and white.

b. Message Usage:

– **Consistency:** Labels and input field names are consistently formatted in terms of font and color, helping users easily identify and use the interface functions.

– **Politeness:** The neutral interface suits all ages, and its simple presentation creates a comfortable experience, despite the absence of explicit politeness messages.

– **Simplicity:** Messages are straightforward and easy to understand. Labels on input fields and displayed information are concise yet complete.

- **Informative:** Labels on input fields clearly indicate their functions. Users can easily understand what information to enter in each field.
- **Use user language:**
  + **General:** The language used is general and understandable by most users.

c. Data Validation:

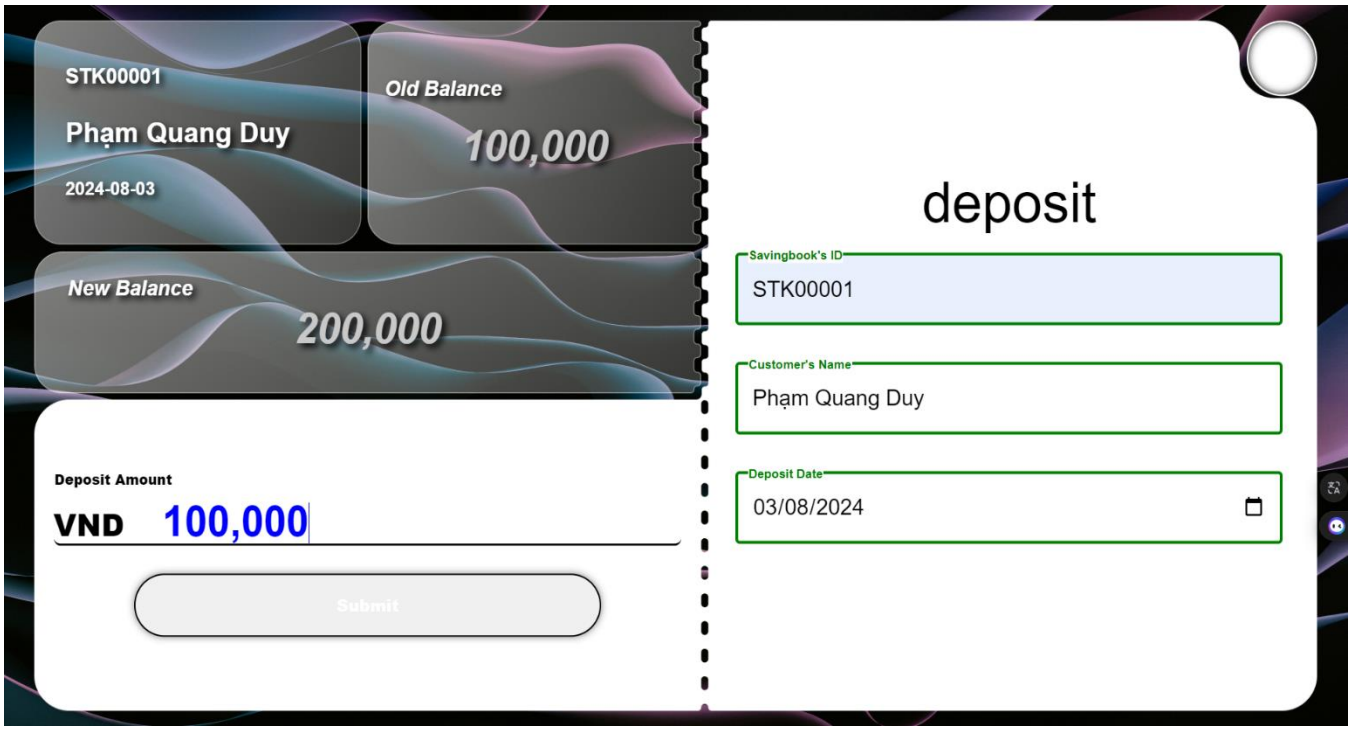
- **Validation constraints:**
  + **Natural constraints:**
    - Customer names must not contain numbers. (1)
    - ID numbers must not contain letters or special characters. (2)
    - Initial deposit amount must not contain letters or special characters and must not be 0. (3)
  + **Business constraints:**
    - The name and address of the customer with the same ID number must be consistent (An ID number cannot exist for multiple customers with different information). (4)
    - Initial Deposit must not be lower than the minimum required amount. (5)
- **Types of validation:**
    - **Early checking:** Client-side validation to provide immediate feedback:
      - Constraints: (1), (2), (3)
    - **Late checking:** Server-side validation to ensure data integrity after submission:
      - Constraints: (4) and (5)

3. Interaction with screen:

- **Step 1**: Enter customer information: "Customer's Name", "ID Number", "Address". Customer information will be displayed in the first gray box at the top left.
- **Step 2**: Enter savings account information: "Initial Deposit", "Term". Account information will be displayed in the second gray box at the top left.
- **Step 3**: If there are no real-time validation errors, click "Submit". If errors are detected, they will be displayed in red below the corresponding input field.
- **Step 4**: A success message will be displayed if the submission is successful

5.2.3 Deposit:

1. User Interface Controls:



a. Input Controls:

- **Command:**
  - **Button:**
    - Gray "Submit" button at the bottom.
    - Round "Home" button at the top right corner.
  - **Link:** Linked to the Home page via the round button at the top right corner.
- **Typing:**
  - **TextBox:**
    - Input field "Savingbook's ID": the ID of the savings book.
    - Input field "Customer's Name": the customer's name, which is a read-only field that auto-fills when the savings book ID is entered.
    - Input field "Deposit Date": default value is the current date.
    - Input field "Deposit Amount" with the note "VND amount?": the amount deposited into the savings account.

b. Output Controls:

- **Simple Output:**
  - **Label:**
    - "Deposit": the title of the deposit page.
    - "Old Balance": displays the previous balance.
    - "New Balance": displays the new balance after the deposit.
  - **TextBox:** Three gray boxes at the top left displaying customer-entered information:
    - First box: Customer information and deposit date.
    - Second box: Account balance.
    - Third box: New account balance after the deposit.
  - **MessageBox:** Notification when submission is successful.

2. User Interface Design Guidelines:

a. Color Usage:

- **Consistency:** Colors are used consistently with a dark background and light-colored input forms. The dark background contrasts well with white and blue text, making information clear and readable.
- **Simplicity:** The design uses a simple color palette primarily consisting of black, white, and blue. Important information is highlighted with different colors and font styles.
- **Do not use too many colors (4/6):** The design adheres to this principle, using a limited set of colors, including a dark background and light text. This helps focus on important information.
- **Be careful of contrast colors:** The contrast between the light background and black/blue text, or vice versa, ensures readability and clarity. Important information like deposit amount, previous balance, and new balance are highlighted in blue and white.

b. Message Usage:

- **Consistency:** Labels and input field names are consistently formatted in terms of font and color, helping users easily identify and use the interface functions.

- **Politeness:** The interface is neutral; suitable for a wide range of ages and users. While there are no explicit politeness messages, the simple presentation and language usage make users feel comfortable.
- **Simplicity:** Messages are straightforward and easy to understand. Labels on input fields and displayed information are concise yet complete.
- **Informative:** Labels on input fields clearly indicate their functions. Users can easily understand what information to enter in each field.
- **Use user language:**
  + **General:** The language used is general and understandable by most users.

c. Data Validation:

- **Validation constraints:**
  + **Natural constraints:**
    - Deposited amount must not contain letters or special characters and must not be 0. (1)
  + **Business constraints:**
    - Customer ID must follow the format "STK + xxxxx". For example: STK00012. (2)
    - Customer information must already be saved in the database. (3)
    - Deposit amount must not be lower than the minimum required amount. (4)
    - Deposits are only allowed if the account type is no-period. (5)
    - Check if the account is closed. (6)
- **Types of validation:**
  + **Early checking:** Client-side validation to provide immediate feedback:
    - Constraints: (1), (2)
  + **Late checking: Server-side validation to ensure data integrity after submission:**
    - Constraints: (3), (4)
    - Server-side validation with immediate feedback: (5), (6)

3. Interaction with screen:

- **Step 1**: Enter customer information: "ID Number", and the system will auto-fill "Customer's Name" if the account exists in the database. Customer information will be displayed in the first gray box at the top left along with the current account balance in the second gray box.
- **Step 2**: Enter information to deposit into the savings account: "Deposit Amount". The new balance after the deposit will be displayed in the third gray box.
- **Step 3**: If there are no real-time validation errors, click "Submit". If errors are detected, they will be displayed in red below the corresponding input field.
- **Step 4**: A success message will be displayed if the submission is successful.

5.2.4 Withdraw:

1. User Interface Controls:

a. Input Controls:

- **Command:**
  - **Button:**
    - Gray "Submit" button at the bottom.
    - Round "Home" button at the top right corner.
  - **Link:** Linked to the Home page via the round button at the top right corner.
- **Typing:**
    - **TextBox:**
      - Input field "Savingbook's ID": the ID of the savings book.
      - Input field "Customer's Name": the customer's name, which is a read-only field that auto-fills when the savings book ID is entered.
      - Input field "Withdraw Date": default value is the current date.
      - Input field "Withdraw Amount" with the note "VND amount?": the amount withdrawn from the savings account.

b. Output Controls:

- **Simple Output:**
  - **Label:**
    - "Withdraw": the title of the withdrawal page.
    - "Old Balance": displays the previous balance.
    - "New Balance": displays the new balance after the withdrawal.
  - **TextBox:** Three gray boxes at the top left displaying customer-entered information:
    - First box: Customer information and deposit date.
    - Second box: Account balance "Old Balance".
    - Third box: New account balance after the withdrawal "New Balance".
  - **MessageBox:** Notification when submission is successful.

2. User Interface Design Guidelines:

a. Color Usage:

- **Consistency:** Colors are used consistently with a dark background and light-colored input forms. The dark background contrasts well with white and blue text, making information clear and readable.
- **Simplicity:** The design uses a simple color palette primarily consisting of black, white, and blue. Important information is highlighted with different colors and font styles.
- **Do not use too many colors (4/6):** The design adheres to this principle, using a limited set of colors, including a dark background and light text. This helps focus on important information.
- **Be careful of contrast colors:** The contrast between the light background and black/blue text, or vice versa, ensures readability and clarity. Important information like withdrawal amount, previous balance, and new balance are highlighted in blue and white.

b. Message Usage:

- **Consistency:** Labels and input field names are consistently formatted in terms of font and color, helping users easily identify and use the interface functions.

- **Politeness:** The interface is neutral; suitable for a wide range of ages and users. While there are no explicit politeness messages, the simple presentation and language usage make users feel comfortable.
- **Simplicity:** Messages are straightforward and easy to understand. Labels on input fields and displayed information are concise yet complete.
- **Informative:** Labels on input fields clearly indicate their functions. Users can easily understand what information to enter in each field.
- **Use user language:**
  + **General:** The language used is general and understandable by most users.

c. Data Validation:

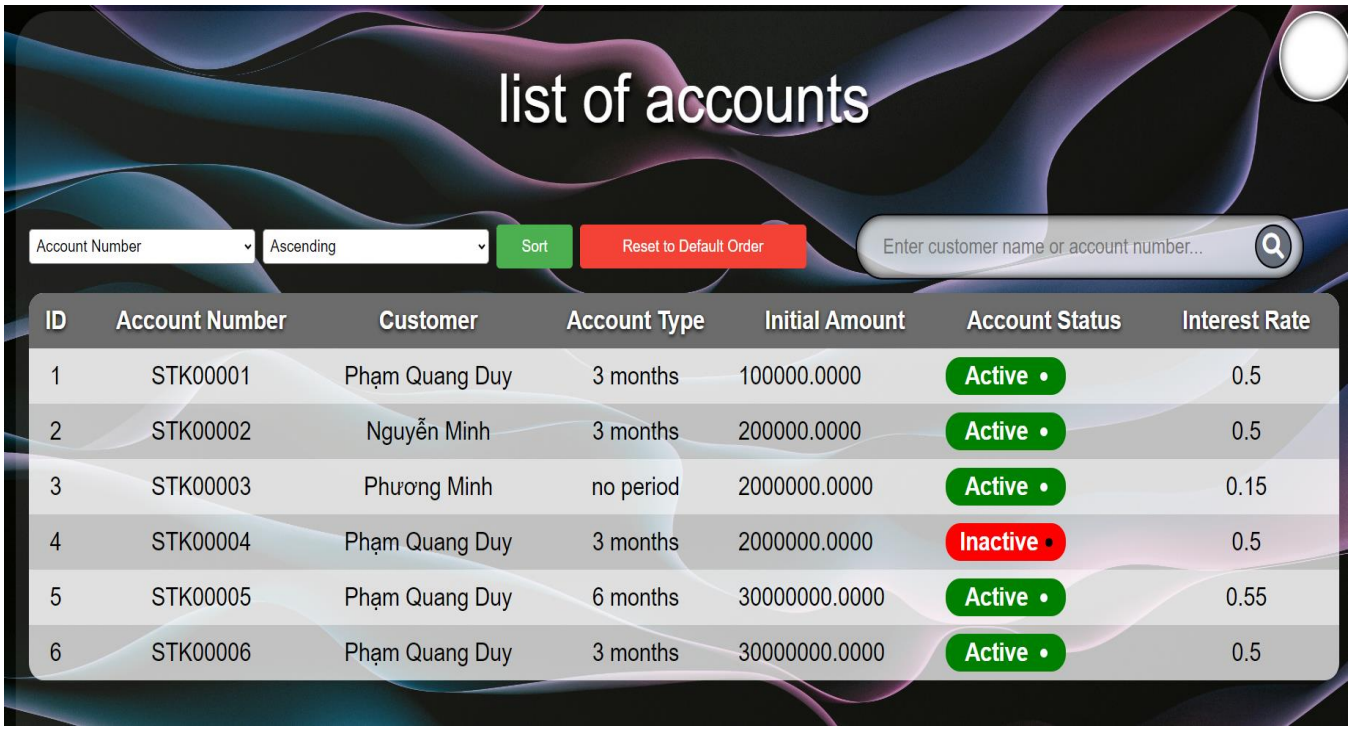
- **Validation constraints:**
  + **Natural constraints:**
    - Withdrawn amount must not contain letters or special characters and must not be 0. (1)
  + **Business constraints:**
    - Customer ID must follow the format "STK + xxxxx". For example: STK00012. (2)
    - Customer information must already be saved in the database. (3)
    - For no-period savings accounts: The withdrawal amount must not exceed the account balance. (4)
    - For term savings accounts: The withdrawal amount must equal the account balance (withdraw the entire amount). (5)
    - The withdrawal transaction must be at least 15 days after the last deposit/opening transaction. (6)
    - The savings account has not met the minimum period for withdrawal. (7)
    - Check if the account is closed. (8)
- **Types of validation:**
  - **Early checking:** Client-side validation to provide immediate feedback:
    - Constraints: (1), (2)

- **Late checking:** Server-side validation to ensure data integrity after submission:
  - Server-side validation with immediate feedback: (3), (4)
  - Server-side validation with feedback after submission: (5), (6), (7), (8)

3. Interaction with screen:

- **Step 1**: Enter customer information: "ID Number", and the system will auto-fill "Customer's Name" if the account exists in the database. Customer information will be displayed in the first gray box at the top left along with the current account balance in the second gray box.
- **Step 2**: Enter information to withdraw from the savings account: "Withdraw Amount". The new balance after the withdrawal will be displayed in the third gray box.
- **Step 3**: If there are no real-time validation errors, click "Submit". If errors are detected, they will be displayed in red below the corresponding input field.
- **Step 4**: A success message will be displayed if the submission is successful.

5.2.5 Lookup Account:



1. User Interface Controls:

a. Input Controls:

- **Command:**
  - **Button:**
    - "Sort" button to sort the account list by ID or in ascending order.
    - "Reset to Default Order" button to reset the list to the default order.
  - **Link:** Linked to the Home page via the round button at the top right corner.
- **Typing:**
  - **TextBox:** Search box "Enter customer name or account number..." allows users to enter the customer's name or account number to search.
  - **Selection:**
    - **ListBox:** Present for selecting items and sorting options.

b. Output Controls:

- **Simple Output:**

+ **Label:** "ID", "Account Number", "Customer", "Account Type", "Initial Amount", "Account Status", "Interest Rate" are the details of an account.

– **Complex Output:**

+ **ListView:** Account list table with detailed information.

2. User Interface Design Guidelines:

a. Color Usage:

– **Consistency:** The interface uses consistent colors, mainly dark tones, with light panels and black text to display information. Interface elements such as titles, buttons, and search boxes follow a consistent color scheme, making it easy for users to recognize and use.

– **Simplicity:** The design is simple and easy to understand. Important information such as account number, customer name, account type, initial amount, account status, and interest rate are clearly displayed in a table.

– **Do not use too many colors (4/6):** Uses a limited color palette. The background is black, text is white, and account status uses green (Active) and red (Inactive). Buttons are also consistently colored with green and red, avoiding visual clutter.

– **Be careful of contrast colors:** The contrast between the dark background and light text helps users read information easily. Button colors and account statuses are also designed to stand out and be easily recognizable with green and red.

b. Message Usage:

– **Consistency:** Messages are used consistently and are easy to understand. Labels such as "Account Number", "Customer", "Account Type", "Initial Amount", "Account Status", "Interest Rate" are used consistently.

– **Politeness:** The interface is neutral to suit a wide range of ages, using simple language that makes users feel comfortable.

– **Simplicity:** Messages are concise and easy to understand. Labels and information in the table are written briefly but comprehensively.

– **Informative:** Provides necessary information for users to perform actions. Information about savings accounts is displayed fully and clearly.

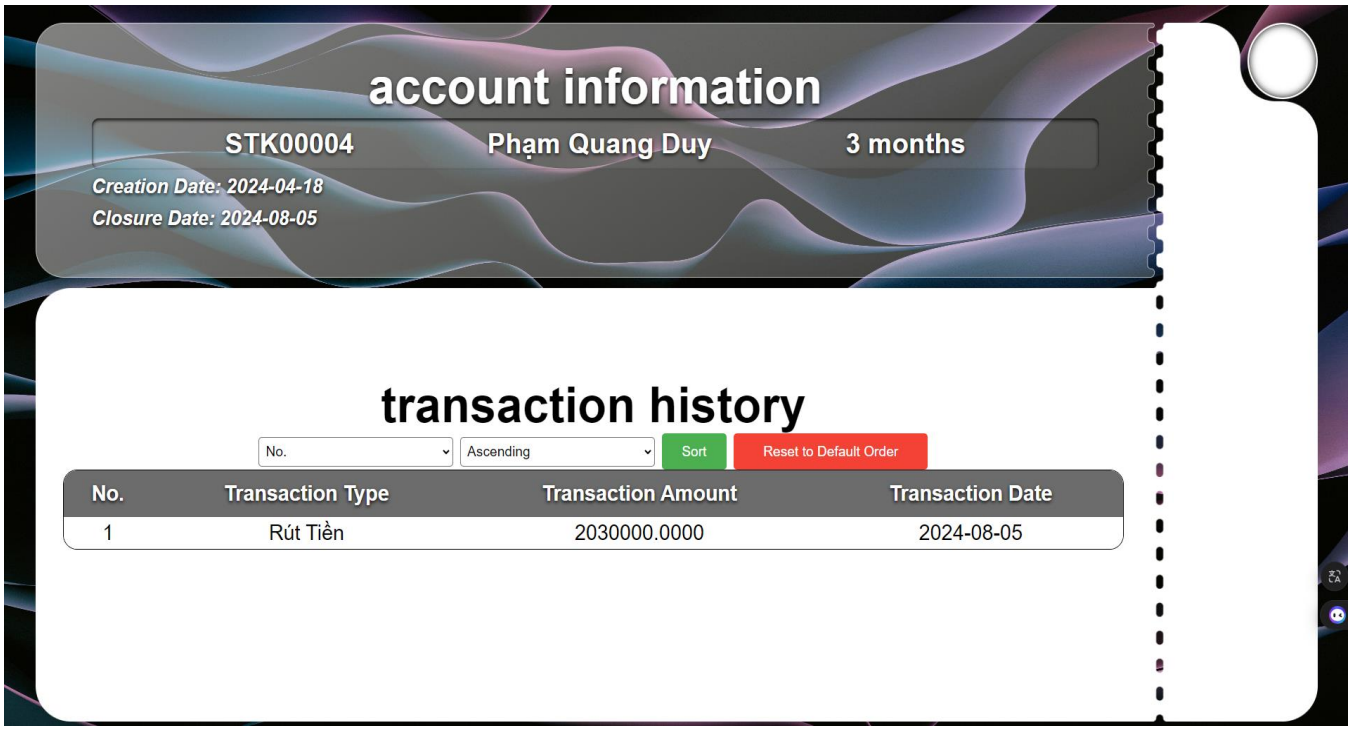
– **Use user language:**

+ **General:** The language used is general and understandable by most users.

3. Interaction with screen:

- The screen will display a table of accounts stored in the database.
- Sort by selecting the item to sort and the sorting method (ascending/descending) and click “Sort”. Cancel sorting with the “Reset to Default Order” button.
- Search by entering the account ID or owner's name to have the system display the account.
- To navigate to “Account Information”, click on the row of the account you want to view.

5.2.6 Account Information:



1. User Interface Controls:

a. Input Controls:

- **Command:**
  - **Button:**
    - Sort" button to sort the account list by ID or in ascending order.
    - "Reset to Default Order" button to reset the list to the default order.
  - **Link:** Linked to the Home page via the round button at the top right corner.
- **Typing:**
  - **Selection:**
    - **ListBox:** Present for selecting items and sorting options.

b. Output Controls:

- **Simple Output:**
  - **Label:** "No.", "Transaction type", "Transaction Amount", "Transaction Date" are the transaction details of an account.

  + **TextBox:** A grey box at the top displays all account information such as ID, owner's name, savings type, open date, and close date (if closed).
- **Complex Output:**
  + **ListView:** A list of accounts with detailed information about each transaction, including transaction type, transaction amount, and transaction date.

2. User Interface Design Guidelines:

a. Color Usage:

- **Consistency:** The interface uses consistent colors, mainly dark tones, with light panels and black text to display information. Interface elements such as titles and buttons follow a consistent color scheme, making it easy for users to recognize and use.
- **Simplicity:** The design is simple and easy to understand. Important information such as account number, customer name, account type, and account status is clearly displayed in a table.
- **Do not use too many colors (4/6):** Uses a limited color palette. The background is black, text is white, and account statuses use green (Active) and red (Inactive). Buttons are also consistently colored with green and red, avoiding visual clutter.
- **Be careful of contrast colors:** The contrast between the dark background and light text helps users read information easily. Button colors and account statuses are also designed to stand out and be easily recognizable with green and red.

b. Message Usage:

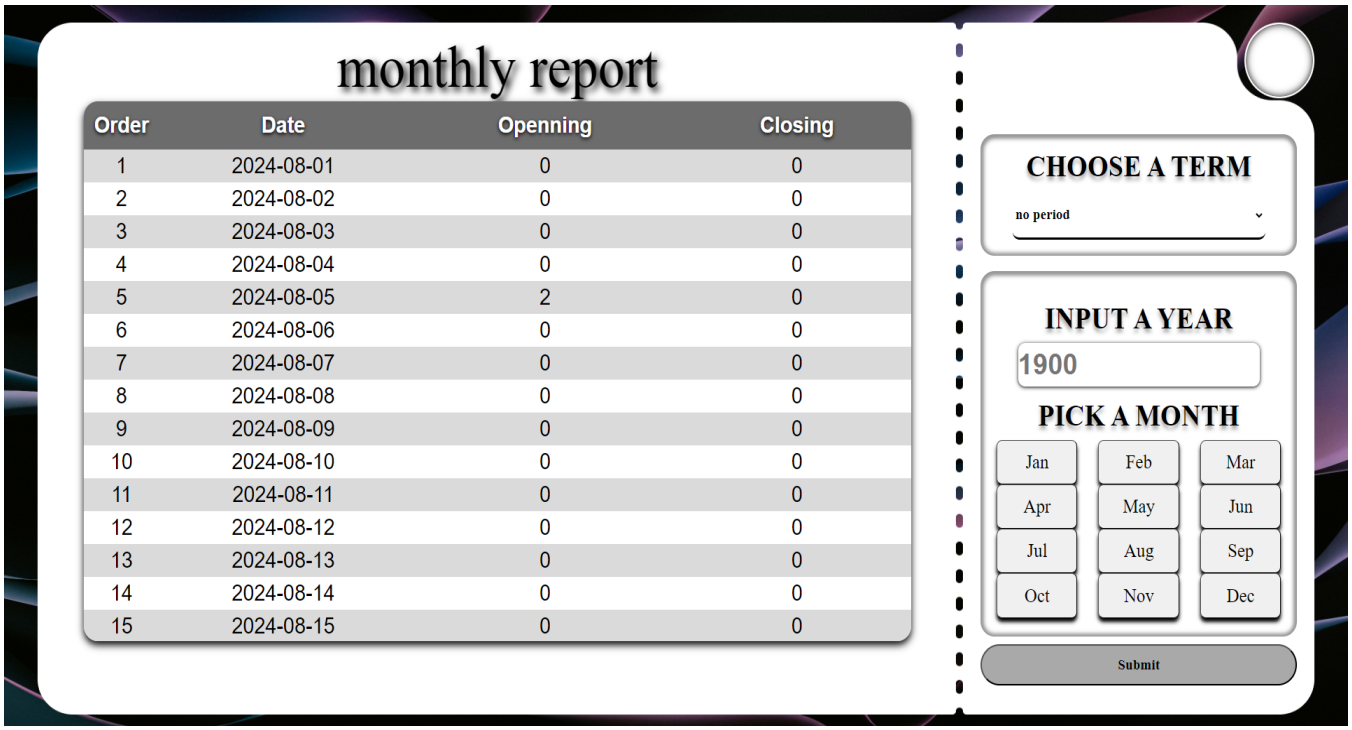
- **Consistency:** Messages are used consistently and are easy to understand. Labels such as "No.", "Transaction type", "Transaction Amount", "Transaction Date" are used consistently.
- **Politeness:** The interface is neutral to suit a wide range of ages, using simple language that makes users feel comfortable.
- **Simplicity:** Messages are concise and easy to understand. Labels and information in the table are written briefly but comprehensively.
- **Informative:** Provides necessary information for users to perform actions. Information about savings accounts is displayed fully and clearly.

- **Use user language:**
  - **General:** The language used is general and understandable by most users.

3. Interaction with screen:

- The screen will display a table of stored transactions for the account.
- Sort by selecting the item to sort and the sorting method (ascending/descending) and click "Sort". Cancel sorting with the "Reset to Default Order" button.
- All information will be displayed in the grey box.

5.2.7 Report Monthly:



1. User Interface Controls:

a. Input Controls:

- **Command:**
  - **Button:**
    - "Submit" button to send the selected date information to generate the report.
    - "Back to table" button to return to the table view when in chart report mode.
  - **Link:** Linked to the Home page via the round button at the top right corner.
- **Typing:**
  - **Spinner**: Year input can be entered or adjusted between the values 1900 and 2024.
  - **Selection:**
    - **ListBox:** "Choose a term" to select the type of savings for the report.
    - **ComboBox:** None present, but there is a month selection option.

b. Output Controls:

- **Simple Output:**
  - **Label:** Labels such as "Order", "Date", "Opening", "Closing" describe the columns in the report.
  - **TextBox:** No TextBox displays specific information, but there are cells in the table.
- **Complex Output:**
  - **ListView:** A table listing the days of the month and the corresponding number of accounts opened and closed.
  - **Report:** A column chart representing the monthly report by showing the number of accounts opened and closed each day if clicked from the table.

2. User Interface Design Guidelines:

a. Color Usage:

- **Consistency:** The interface uses consistent colors, primarily dark tones, light panels with black text to display information. Interface elements such as titles and buttons follow a specific color scheme, making it easy for users to recognize and use.
- **Simplicity:** The design is simple and easy to understand. Important information such as account number, date, total accounts opened, and closed are clearly displayed in a table. The chart format also clearly presents data with complete legends and indicators, effectively conveying information.
- **Do not use too many colors (4/6):** Uses a limited color palette. The background is black, text is white, or vice versa for tables. For charts, each category is represented by a distinctive, eye-catching color that remains pleasant for users. Buttons are also consistently colored to avoid visual clutter.
- **Be careful of contrast colors:** The contrast between the dark background and light text helps users easily read the information. The colors of the columns in the chart are also designed to stand out against the background.

b. Message Usage:

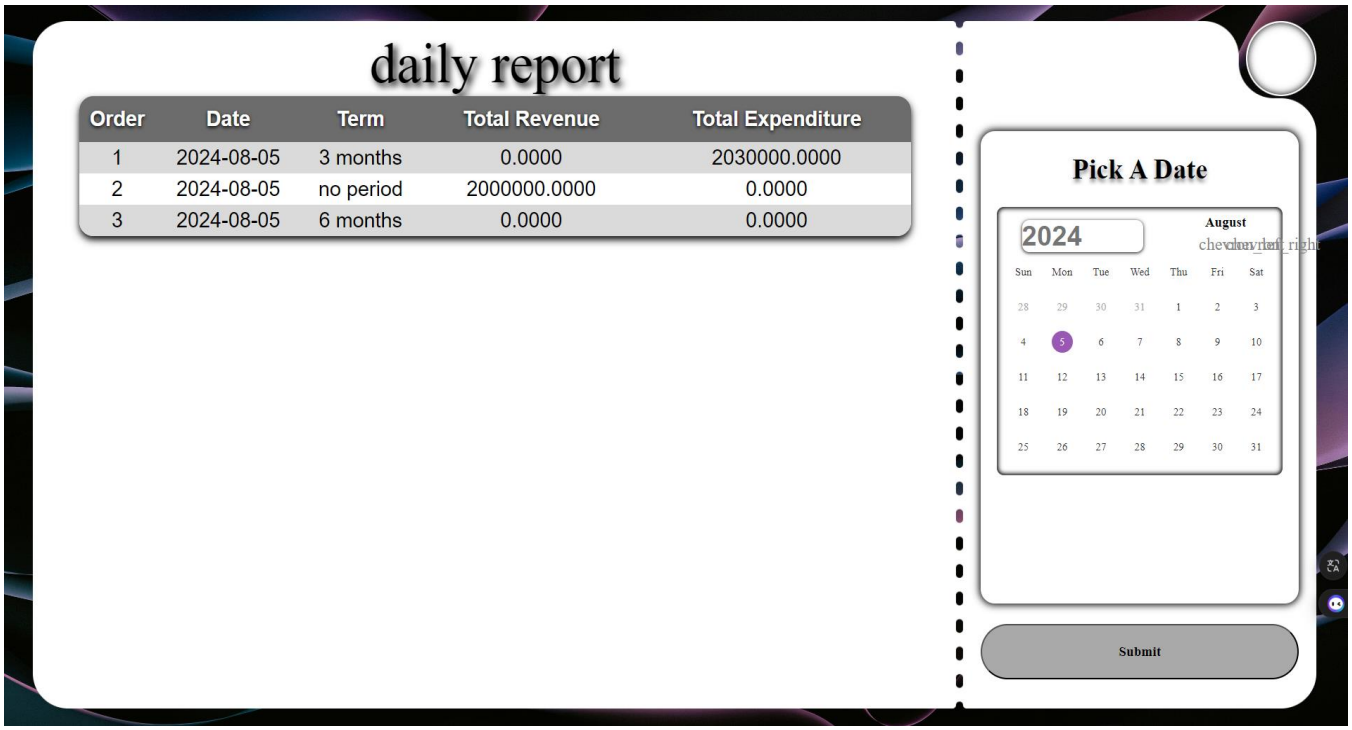
- **Consistency:** Messages are used consistently and are easy to understand. Labels such as "Order", "Date", "Opening", "Closing" are used consistently.

- **Politeness:** The interface is neutral to suit a wide range of ages, using simple language that makes users feel comfortable.
- **Simplicity:** Messages are concise and easy to understand. Labels and information in the table are written briefly but comprehensively.
- **Informative:** Provides necessary information for users to perform actions. Information about savings accounts is displayed fully and clearly.
- **Use user language:**
  + **General:** The language used is general and understandable by most users.

3. Interaction with screen:

- **Step 1**: Select the type of savings for the report.
- **Step 2**: Enter the year and month for the report.
- **Step 3**: Click "Submit".
- The system will display the information on the number of accounts opened/closed by each day of the month.
- The system will display column chart if user click on the table.

5.2.8 Report Daily:



1. User Interface Controls:

a. Input Controls:

- **Command:**
  - **Button:**
    - "Submit" button to send the selected date information to generate the report.
    - "Back to table" button to return to the table view when in chart report mode.
  - **Link:** Linked to the Home page via the round button at the top right corner.
- **Typing:**
  - **Spinner**: Year input can be entered or adjusted between the values 1900 and 2024.
  - **Selection:**
    - **ComboBox:** None present, but there is a day selection option.

b. Output Controls:

- **Simple Output:**

  - + **Label:** Labels such as "Order", "Date", "Term", "Total Revenue", "Total Expenditure" describe the columns in the report.
  - + **TextBox:** None present. Information is displayed in the table cells.
- **Complex Output:**
  - + **ListView:** A table listing the daily report with total revenue and total expenditure for each type of savings.
  - + **Report:** A column chart representing the daily report with total revenue and total expenditure for each day, displayed if clicked from the table.

2. User Interface Design Guidelines:

a. Color Usage:

- **Consistency:** The interface uses consistent colors, primarily dark tones, light panels with black text to display information. Interface elements such as titles and buttons follow a specific color scheme, making it easy for users to recognize and use.
- **Simplicity:** The design is simple and easy to understand. Important information such as account numbers, transaction dates, transaction types, total revenue, and total expenditure are clearly displayed in a table. The chart also presents data clearly with complete legends and indicators, effectively conveying information.
- **Do not use too many colors (4/6):** Uses a limited color palette. The background is black, text is white, or vice versa for tables. For charts, each category is represented by a distinctive, eye-catching color that remains pleasant for users. Buttons are also consistently colored to avoid visual clutter.
- **Be careful of contrast colors:** The contrast between the dark background and light text helps users easily read the information. The colors of the columns in the chart are also designed to stand out against the background.

b. Message Usage:

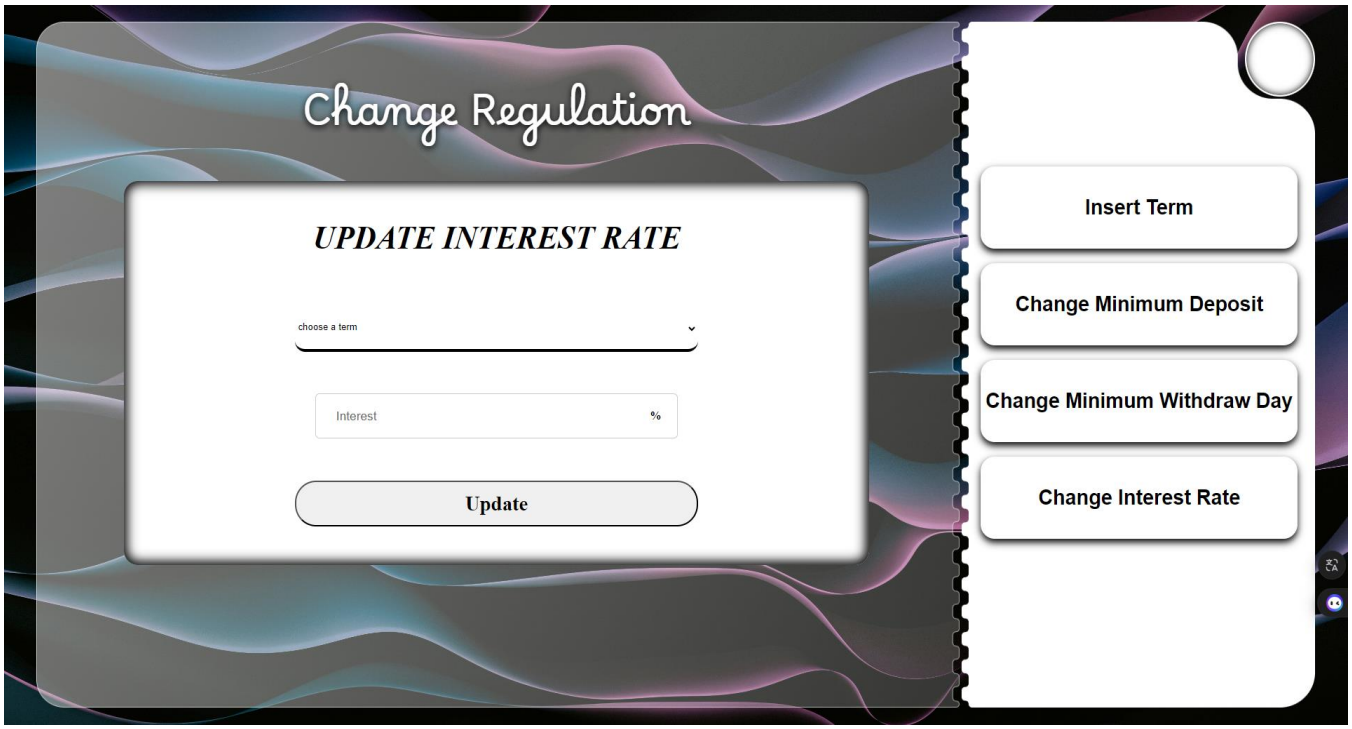
- **Consistency:** Messages are used consistently and are easy to understand. Labels such as "Order", "Date", "Term", "Total Revenue", "Total Expenditure" are used consistently.
- **Politeness:** The interface is neutral to suit a wide range of ages, using simple language that makes users feel comfortable.

- **Simplicity:** Messages are concise and easy to understand. Labels and information in the table are written briefly but comprehensively.
- **Informative:** Provides necessary information for users to perform actions. Information about savings accounts is displayed fully and clearly.
- **Use user language:**
  + **General:** The language used is general and understandable by most users.

3. Interaction with screen:

- **Step 1**: Enter the day, month, and year for the report.
- **Step 2**: Click "Submit".
- The system will display information on total revenue and total expenditure for the day, broken down by each type of savings.
- The system will display column chart if user click on the table.

5.2.9 Change Regulations:



1. User Interface Controls:

a. Input Controls:

- **Command:**

+ **Button:**
    - Button to switch between regulations for making changes.
    - "Confirm" button to confirm the entered information about terms and interest rates on the initial screen.
    - "Update" button on screens for changing minimum deposit amounts, minimum withdrawal days, and interest rates.
+ **Link:** Linked to the Home page via the round button at the top right corner.

– **Typing:**

+ **TextBox:**
    - Insert New Term: Enter the savings term and interest rate for the new savings type.
    - Update Interest Rate: Enter the interest rate for an existing savings term.
+ **Spinner**:
    - Change Minimum Deposit: Enter the new minimum deposit amount.
    - Change Minimum Withdraw Day: Enter the new minimum number of days from the most recent transaction.
+ **Selection:**
    - **ListBox:**
        - Update Interest Rate: Choose a term – select the savings type from available options.
    - **ComboBox:** None present, but there is a date selection option.

b. Output Controls:

– **Simple Output:**

+ **Label:** Labels such as "Term", "Interest", "Deposit", and "Day" to describe input fields.
+ **TextBox:** None present. Information is displayed in input fields.

2. User Interface Design Guidelines:

a. Color Usage:

- **Consistency:** The interface uses consistent colors, primarily dark tones with light text, creating a harmonious feel. Elements such as titles and buttons follow a specific color scheme, making it easy for users to recognize and use.
- **Simplicity:** The design is simple, making it easy for users to understand and use. Important information is clearly displayed in input fields and buttons.
- **Do not use too many colors (4/6):** Uses a limited color palette to avoid visual clutter. The dark background combined with light text ensures good contrast.
- **Be careful of contrast colors:** The contrast between the dark background and light text helps users easily read the information. Buttons are also consistently colored to avoid visual clutter.

b. Message Usage:

- **Consistency:** Messages are used consistently and are easy to understand. Labels such as "Term", "Interest", "Deposit", and "Day" are used uniformly.
- **Politeness:** The interface is neutral, using simple language to make users feel comfortable and accessible.
- **Simplicity:** Messages are concise and easy to understand. Labels and information in input fields are written briefly but comprehensively.
- **Informative:** Provides necessary information for users to perform actions. Information about terms and interest rates is clearly displayed.
- **Use user language:**
  + **General:** The language used is general and understandable to most users.

3. Interaction with screen:

- **Step 1**: Select a regulation you want to change from the options on the right.
- **Step 2**: Enter the relevant editing information for each type.
- **Step 3**: Click“Submit”.