**Final Project Report**

Student Name: 김동건

Student ID: 2020315519

**1. Briefly describe the project purpose:**

The project is a simulation of an ATM system named SKKUATM, where users can create accounts, log in, deposit, withdraw, and transfer funds. It also includes functionality for updating and saving account information and transaction histories.

**2. Draw the logic flow of the program (with flowchart):**

success

Update account info

Back

Validate data(current, new, confirm), back

fail

Error!

Initialize

Change Pin

Error!

success

success

success

Change pin

fail

Error!

Validate data

Validate data

Validate data

transfer

deposit

withdraw

login

Initialize

ATM menu

Create account

success

Validate data

Initialize create account page

Log out

Deposit, withdraw, transfer, change pin, log out

Login or create account

Initialize SKKUATM

**3. Provide screenshots for each screen with brief description:**

<ATM screen>

In here, we can create account, and based on the account, log-in.

텍스트, 스크린샷, 소프트웨어, 멀티미디어이(가) 표시된 사진

자동 생성된 설명

<create account screen>

텍스트, 스크린샷, 소프트웨어, 운영 체제이(가) 표시된 사진

자동 생성된 설명

In here, we can create account

<ATM menu>

This is a ATM menu tha involves transaction table. In here, we can deposit, withdraw, and transfer, log out

스크린샷, 텍스트, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

<Chang PIN>

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

In here, we can Change PIN.

<accountData.txt>

This is the accountData.txt that stores the account data of the accounts.

스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**4. Explain the code of the main functionalities**

SKKU ATM : the main class for the application that initiates the user interface and handles account creation and login.

CreatePage: A class that can assigns account number and temp PIN number to the user.

AccountInfo : A class that stores details about each user account, including the account number, PIN, balance, name, and date of birth.

menu : A class that creates the user interface for account operations like deposit, withdraw, and transfer, log out after a successful login.

ChangePIN: A class that can change PIN.

AccountStorage : Handles the file I/O operations to save and load account details from a file named "accountData.txt".

TransactionHistory : Manages the recording of all transactions per account and notifies listeners about any updates.

TransactionRecord : Represents a single transaction with all related details such as date, type, amounts, etc.

AccountValidator : Contains static methods to validate account numbers and PINs.

Notification and its subclasses(DepositNotification, WithdrawalNotification, TransferNotification) : mplements a notification system where different types of transactions trigger corresponding notifications.

PersonInfo : A class that could potentially be used to store personal information about users or customers.

**5. Explain what is included in your project and why it is used (Polymorphism, Inheritance, File I/O, etc)**

Polymorphism: This is used in the Notification class hierarchy where different types of notifications inherit from a base class and provide their specific implementation of sending notifications.

Inheritance: The DepositNotification, WithdrawalNotification, and TransferNotification classes inherit from the Notification class.

File I/O: AccountStorage uses file I/O to save and load account data to and from a text file.

GUI (Graphical User Interface): The SKKUATM and menu, CreatePage, ChangePin classes use Swing to create a graphical interface for user interactions.

Thread: In the SKKUATM class, multi-threading is implemented using SwingWorker in the btnLoginAccount action listener; this allows the login process to run in the background, preventing the GUI from freezing during long operations. Also, The SwingUtilities.invokeLater() method in the menu class is used to safely update the GUI from a non-Event Dispatch Thread (EDT). It schedules the execution of the enclosed code block (which creates and sends a Notification and updates errorArea) to be executed later on the EDT, ensuring thread safety in Swing applications.