NATIONAL UNIVERSITY
COLLEGE OF COMPUTING AND
INFORMATION TECHNOLOGIES

Group Information		
Group Name:	EABank	
Project Title:	EABank System: A Secure E-Wallet Platform for Digital Transactions	
Group Leader:	Recaña, Fiona Mai R.	
Team Members		
UI/UX Designer:	Recaña, Fiona Mai R.	Sign and Date: 6/10/2025 
Front-End Developer:	Agsiwang, Kate Angel R.	Sign and Date: 6/10/2025 
Back-End Developer:	Aquino, Andrea Mae C.	Sign and Date: 6/10/2025 

Project Specifications:
(*Please list your project specifications in a numbered format. Each specification should include a clear and concise bullet description* ***).***

1. Login and Register Functionality
- Allows users to log in or register an account using mobile number and a 4-digit PIN that secures user accounts by verifying access credentials
- Validates input and prevents duplicate mobile number registrations
- Includes OTP Verification during registration to enhance account security
- “Show PIN” checkbox toggles password visibility
- Displays error messages for invalid or missing output
- Includes “Exit” Button to close the application

2. Main Menu Navigation
- Displays a personalized welcome message using the user’s registered name after login
- Provides access to key features like Cash In, Cash Out, Transfer, Check Balance, and Transaction History
- Includes a “Logout” button to return to the login/register screen

3. Cash In
- Prompts user to enter a valid, positive amount to deposit
- Updates user’s balance upon successful input
- Updates transaction history with a reference number and timestamp
- Displays a receipt-style dialog summarizing the transaction
- Provides input validation with error messages for invalid or negative amounts

4. Cash Out
- Prompt user to input for amount to withdraw
- Allows the user to select a pickup location (e.g., convenience stores or banks).
- Checks for sufficient balance before processing
- Deducts amount from the user’s balance if valid
- Updates transaction history with a reference number and timestamp
- Displays a receipt-style dialog summarizing the transaction
- Provides input validation with error messages for invalid or negative amounts

5. Transfer Funds
- Prompts the user for the recipient's mobile number and amount to transfer
- Validates recipient in the registered user list
- Verifies sufficient funds before processing
- Deducts from sender and adds to recipient’s balance
- Logs transaction on both users’ history records

6. Balance Inquiry (Check Balance)

- Displays the user's current balance in a pop-up dialog
- Accessible from the main menu at any time
- Reflects latest updates based on recent transactions.
- Design for fast, clear access to financial information

7. Currency Converter (Money Exchanger)

- Converts PHP to USD or JPY.
- Accepts valid PHP input and converts using fixed rates.
- Displays a styled receipt showing original and converted amounts.
- Includes date/time and success status.

8. Transaction History

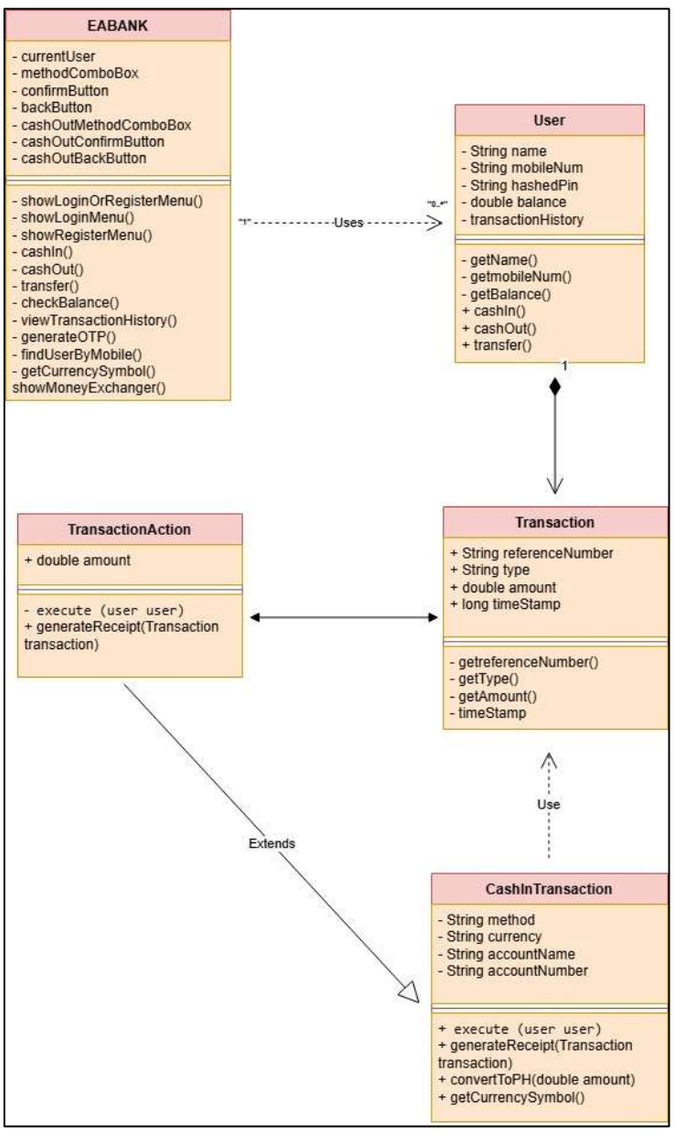
- Displays all user transactions chronologically list
- Includes transaction type, amount, reference number, and date/time.
- Informs the user if no transactions are available

Project Documentation

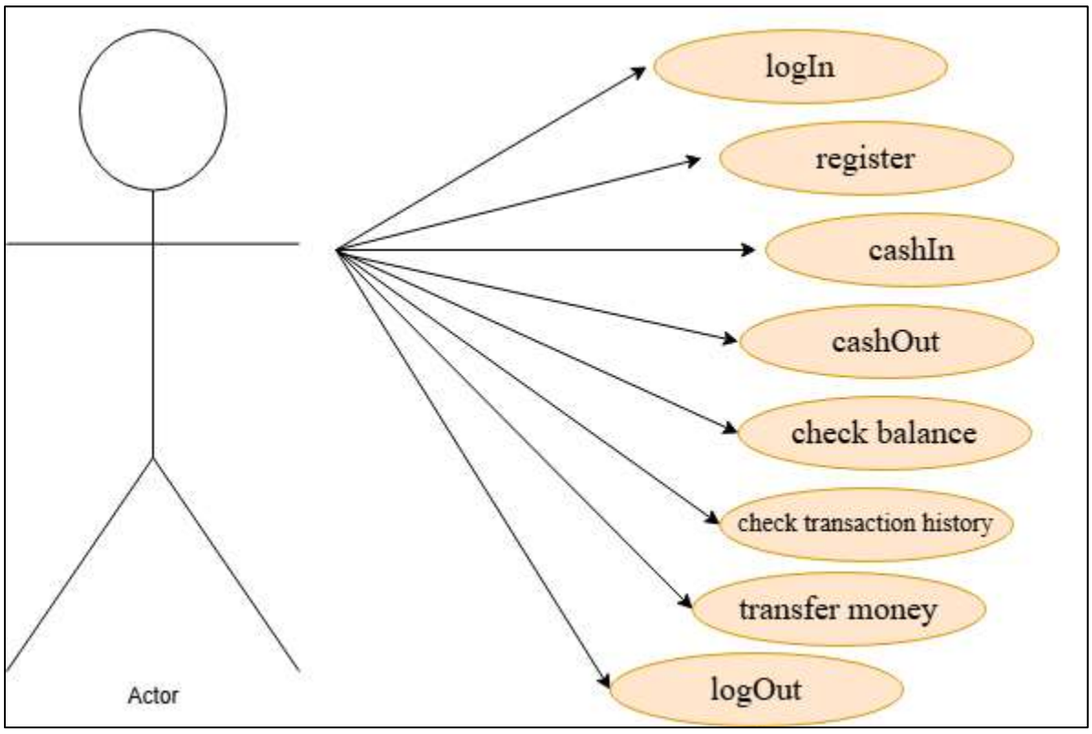
Unified Modeling Language (UML) Diagrams

(All diagrams must be uploaded in PNG format. You may use https://draw.io/ or similar tools to create your UML diagrams.)

1. Class Diagram



2. Use Case Diagram



User Interface Screenshots

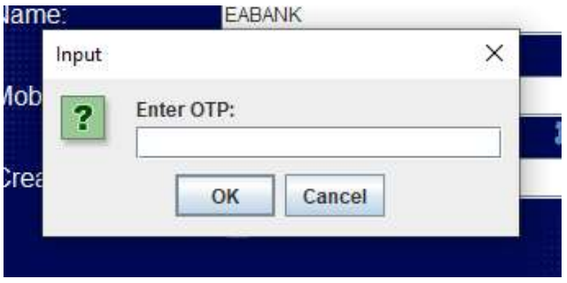
(Provide screenshots of the user interface that corresponds to your specifications. Each UI should include a relevant description explaining its purpose and the specific functionality it supports)

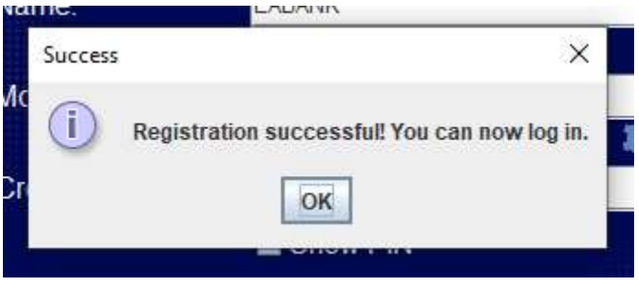
User Interface 1

Description:

This is the main menu screen of the MKA EABANK SYSTEM. It is the first screen that appears when the app is opened. It has three buttons: Login, Register, and Exit. The Login button is for users who already have an account. The Register button is for new users who want to sign up. The Exit button closes the app. This screen helps users get started and choose what they want to do next in the system.

User Interface 2





Description:

This is the Register Functionality which is used for registering a new account in the MKA EABANK SYSTEM. First, the user fills in their name, mobile number, and creates a 4-digit PIN. They can also choose to show or hide the PIN while typing. After clicking the "Register" button, the system generates a one-time password (OTP), which is shown in a pop-up message on the screen. The user must then enter this OTP in the provided box to confirm their identity. Once the correct OTP is entered, a message appears confirming that the registration was successful. This process helps secure the account and ensures that the mobile number belongs to the user.

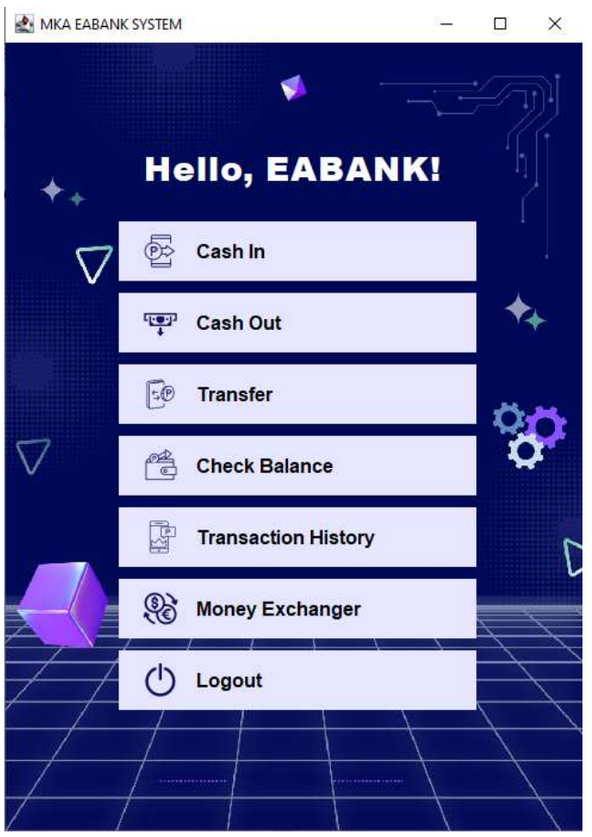
NATIONAL UNIVERSITY
COLLEGE OF COMPUTING AND
INFORMATION TECHNOLOGIES

User Interface 3

Description:

This is the Login Functionality which is used for logging into the MKA EABANK SYSTEM. The user is asked to enter their mobile number and 4-digit PIN that they created during registration. There is an option to show or hide the PIN while typing. Once the user enters the correct information and clicks the "Login" button, the system checks if the details are correct. If they are, the user is allowed to access their account. This screen helps protect user accounts by making sure only people with the correct PIN can log in.

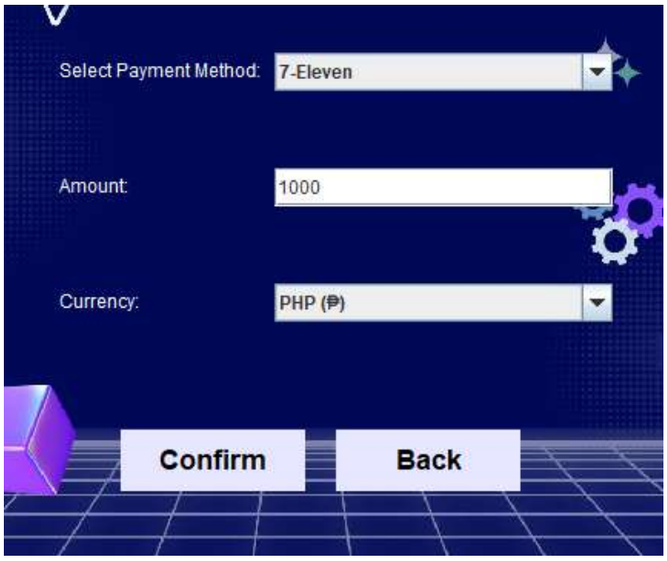
User Interface 4

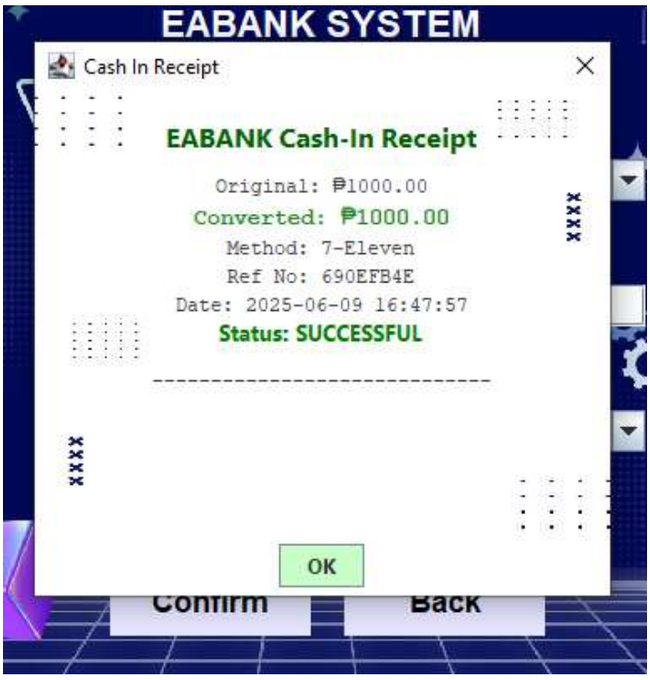


Description:

This screen is the main menu of the MKA EABank System, shown after the user logs in. It shows a welcome message using the user's name from registration. The menu gives access to important features like Cash In to add money, Cash Out to withdraw money, Transfer to send money to other users, Check Balance to view the current account balance, and Transaction History to see past transactions. It also includes a Money Exchanger feature, which allows users to convert money between different currencies. At the bottom, there is a Logout button that safely logs the user out and returns them to the login or register screen.

User Interface 5





Description:

This is the “Cash In” functionality which is used to cash into a user’s account. The user can choose where they want to cash in, such as at 7-Eleven or by using a credit card. They type in the amount they want to cash in and the currency (PHP). Then, they can click the "Confirm" button to continue or "Back" to go to the previous screen. After confirming, a receipt will appear. The receipt shows the amount, payment method, reference number, date, time, and the status of the transaction (like “Successful”). This screen helps users’ cash in easily and safely.

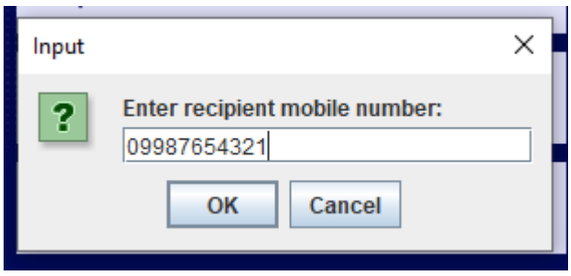
NATIONAL UNIVERSITY
COLLEGE OF COMPUTING AND
INFORMATION TECHNOLOGIES

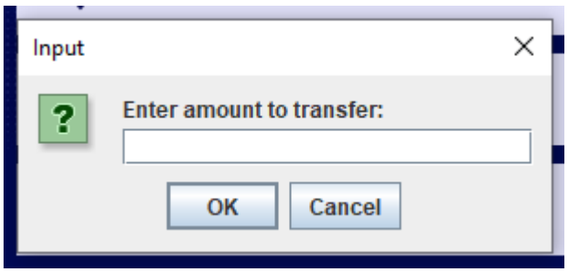
User Interface 6

Description:

This UI highlights the “Cash Out” process of a MKA EABank System. Users can select any outlet such as Uncle John’s, Family Mart, 7-Eleven, or in an own Machine of MKA. After choosing where they can get the money from the system, they will enter the amount, choose a currency. In a system there has a feature where they can convert the money on what currency they want to. After that all process, user will confirm the transaction and a digital receipt will appear, showing the transaction details being made by the user. It includes amount, reference number, date, and status of transaction in the system.

User Interface 7





Description:

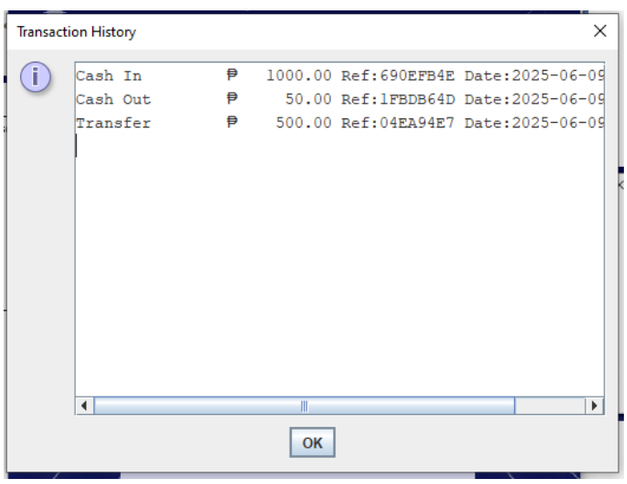
This UI highlights the “Transfer” which is used to transfer funds to another user. First, the user is asked to enter the recipient’s mobile number and the amount to transfer. The system checks if the mobile number belongs to a registered user and makes sure the sender has enough balance. If everything is correct, the money is taken from the sender’s account and added to the recipient’s account. A receipt is then shown with the transfer amount, recipient name, reference number, date, time, and status (like “Successful”). This transaction is saved in both users’ histories for record-keeping.

User Interface 8

Description:

This shows a “balance receipt” after checking the user’s account. It displays the current balance along with the account name, date, time, and status (like “Checked”). The balance reflects the latest updates based on recent transactions. This receipt is helpful for users who want to quickly see how much money they have. It can be accessed anytime from the main menu, making it easy and fast to check financial information.

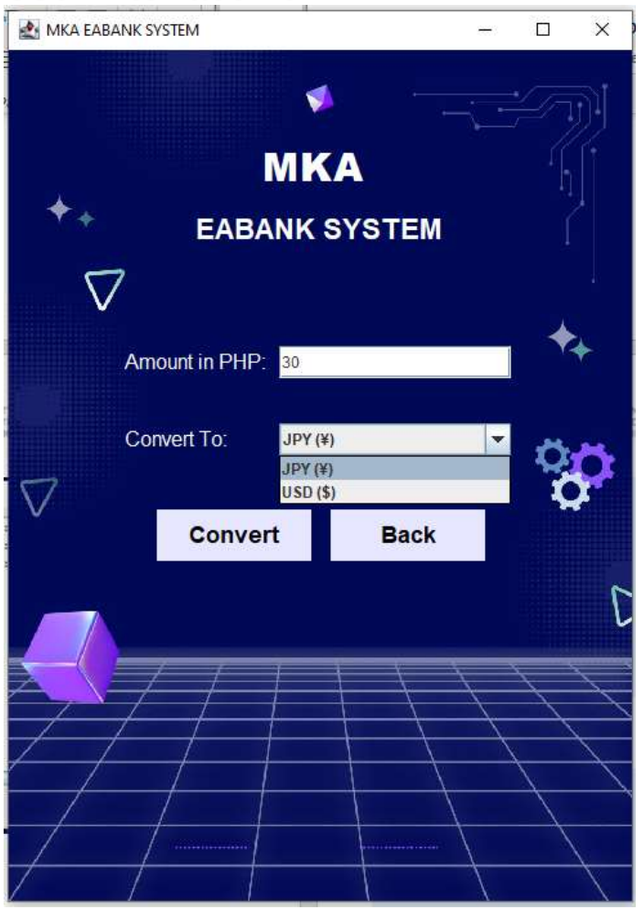
User Interface 9

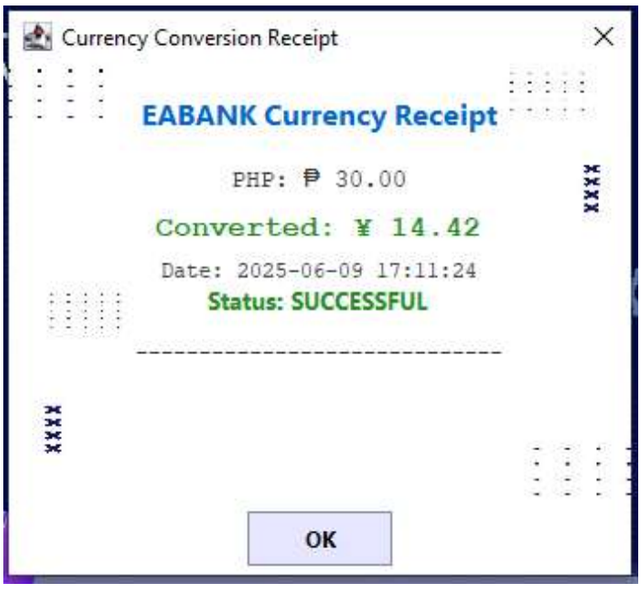


Description:

This UI highlights the “Transaction History” which shows the user’s transaction history. It displays a list of all past transactions in order by date and time. Each entry includes the type of transaction (like Cash In or Transfer), the amount, reference number, and the date and time it happened. If the user has no past transactions, the system will show a message to let them know. This feature helps users keep track of their account activity and monitor their spending or deposits easily.

User Interface 10





Description:

This “Currency Converter” is used for converting Philippine Peso (PHP) to other currencies such as Japanese Yen (JPY) and US Dollar (USD), The user types the amount in PHP and chooses the currency they want to convert to from a list. After clicking the "Convert" button, a small receipt appears on the screen. This feature helps users easily check the value of their money in other currencies and gives them a copy of the conversion details for future reference.

NATIONAL UNIVERSITY
COLLEGE OF COMPUTING AND
INFORMATION TECHNOLOGIES

Code Snippets - Model

*(Attach screenshots of code related **only to the Model** component of your application. Each snippet should include a brief explanation of its role and the feature it supports.)*

Code Snippet 1: User Class

```
// User Class
static class User {
    private final String name;
    private final String mobile;
    private final String hashedPin;
    private double balance;
    private final ArrayList<Transaction> transactionHistory;

    public User(String name, String mobile, String pin) {
        this.name = name;
        this.mobile = mobile;
        this.hashedPin = hashPin(pin);
        this.balance = 0.0;
        this.transactionHistory = new ArrayList<>();
    }

    private String hashPin(String pin) {
        return Integer.toHexString(pin.hashCode());
    }

    public boolean checkPin(String pin) {
        return this.hashedPin.equals(hashPin(pin));
    }

    public String getName() {
        return name;
    }

    public String getMobile() {
        return mobile;
    }

    public double getBalance() {
        return balance;
    }
```

```
    public void cashIn(double amount) {
        balance += amount;
        transactionHistory.add(new Transaction(type:"Cash In", amount));
    }

    public void cashOut(double amount) {
        balance -= amount;
        transactionHistory.add(new Transaction(type:"Cash Out", amount));
    }

    public void transfer(User recipient, double amount) {
        this.balance -= amount;
        recipient.balance += amount;
        transactionHistory.add(new Transaction(type:"Transfer", amount));
        recipient.transactionHistory.add(new Transaction(type:"Received", amount));
    }

    public ArrayList<Transaction> getTransactionHistory() {
        return transactionHistory;
    }

    public Transaction getLastTransaction() {
        if (transactionHistory.isEmpty())
            return null;
        return transactionHistory.get(transactionHistory.size() - 1);
    }
}
}
```

Description:

User Class
- It keeps track all the private information of the user such as name, mobile, pin, balance, and list of transaction history. - It encapsulates and guarantees that the user's information and transaction in the system are handle securely.
- It offers a method for a secure authentication of the user, cash in/out, transfers, and tracking of transaction.

Code Snippet 2: Transaction Class

```java
// Transaction Class
static class Transaction {
    private final String referenceNumber;
    private final String type;
    private final double amount;
    private final LocalDateTime timestamp;

    public Transaction(String type, double amount) {
        this.referenceNumber = UUID.randomUUID().toString().substring(beginIndex:0, endIndex:8).toUpperCase();
        this.type = type;
        this.amount = amount;
        this.timestamp = LocalDateTime.now();
    }

    public String getReferenceNumber() {
        return referenceNumber;
    }

    public String getType() {
        return type;
    }

    public double getAmount() {
        return amount;
    }

    public String getTimestamp() {
        return timestamp.format(DateTimeFormatter.ofPattern(pattern:"yyyy-MM-dd HH:mm:ss"));
    }

    @Override
    public String toString() {
        return String.format(format:"%-15s ₱%10.2f Ref:%s Date:%s", type, amount, referenceNumber, getTimestamp());
    }
}
```

Description:

Transaction Class
- It manages the transaction of users using the EABANK system.
- It enables to list all the transaction made by user that can promote accountability and trust in the application.
- It records all the transactions type, amount, timestamp, and reference number that appears on the receipt.

Code Snippet 3: Transaction Action Class

```
abstract class TransactionAction {
    protected double amount;

    public TransactionAction(double amount) {
        this.amount = amount;
    }

    public abstract void execute(EABANKSYSTEM.User user);

    public abstract String generateReceipt(EABANKSYSTEM.Transaction transaction);
}
```

Description:

Transaction Action Class
- It is an abstract class that serves as a blueprint for all transaction related operation within the EABank System.
- It establishes a standardized structure for how various financial actions, like depositing or withdrawing money should be handled.
- It ensures that all transaction types can produce a receipt providing a clear record for the user

Code Snippet 4: Cash In Transaction Class

```
class CashInTransaction extends TransactionAction {
    private String method;
    private String currency;
    private String accountName;
    private String accountNumber;

    public CashInTransaction(double amount, String method, String currency, String accountName, String accountNumber) {
        super(amount);
        this.method = method;
        this.currency = currency;
        this.accountName = accountName;
        this.accountNumber = accountNumber;
    }

    @Override
    public void execute(EABANKSYSTEM.User user) {
        double convertedAmount = convertToPHP(amount, currency);
        user.cashIn(convertedAmount);
    }

    @Override
    public String generateReceipt(EABANKSYSTEM.Transaction transaction) {
        return String.format(
                format:"--- CASH IN RECEIPT ---%nMethod: %s%nOriginal Amount: %s %.2f%nConverted to PHP: ₱ %.2f%nAccount: %s
                method, getCurrencySymbol(currency), amount,
                convertToPHP(amount, currency),
                accountName, accountNumber.substring(accountNumber.length() - 4),
                transaction.getReferenceNumber(),
                transaction.getTimestamp());
    }
```

```
    public double convertToPHP(double amount, String currency) {
        return switch (currency) {
            case "PHP (₱)" -> amount;
            case "USD ($)" -> amount * 55.5;
            case "JPY (¥)" -> amount * 0.48;
            default -> 0;
        };
    }

    public String getCurrencySymbol(String currency) {
        return switch (currency) {
            case "PHP (₱)" -> "₱";
            case "USD ($)" -> "$";
            case "JPY (¥)" -> "¥";
            default -> "";
        };
    }
}
```

Description:

Cash In Transaction Class
- It handles depositing funds into a user's account.
- It facilitates the "Cash In" feature, allowing users to add money to their balance.
- It supports multi-currency deposits where users can deposit money in various currencies, including Philippine Peso (PHP), US Dollars (USD), and Japanese Yen (JPY).
- It processes the cash-in operation, updating the user's account balance with the converted amount.
- It provides detailed receipt generation.