Banking System Design Document

Group 3

Table of Contents

1. Introduction

1.1. Goals and Objectives

Software must have a friendly GUI and communicate with the Server via ATM interface and Teller(Employee) Interface. Basic banking tasks must be done using the user interfaces and store and execute tasks in Server(Banking System). Use of a Debit Card is offered to client's customers to allow fast access to assets within the system. Software must be simple and clear design for easier implementation.

1.2. Statement of Scope

Decisions made in the design for this document are based on the following priorities: Usability, Maintainability, and Efficiency.

1.3 Software Context

Banking information and activities will be maintained in the Banking System(Server). The commands included in the server will be able to change and update customer information, connect to the atm, or employee interface. The customer information maintained will be checking the balance, withdrawing and depositing, closing and opening accounts, and updating identification information. The Customer information will be accessed through either the atm or employee interface (Client Side). The Teller will only be able to access the employee interface. The customer will be able to use the atm with a debit card for quick access of assets stored in the bank.

1.4. Major Constraints

Issue 1: How and where will multiple customer accounts, checking and savings, be held?

Option 1: All customer's account objects will be held in an array or list to allow for easy access. All checking accounts will be put into one list, and all saving accounts

will be put into another list. The lists will be put into the account class that is linked to the customers object.

Option 2: All customer's account information will be held in hashmaps while the server is running for easy information access. All employees will be put into one hashmap, and all customer's accounts will be put into another list. Debit Cards will also be placed into a hashmap. The data will be loaded into the hashmaps from data files. Data will also be saved every 5 minutes from the hashmaps into data files.

Solution: Option 2 was chosen to fix this issue.

2. Data Design

2.1 Client side

2.1.1 - Sending

The client interfaces: ATM, Teller(Employee) would receive information by a customer and functional commands within the UI will send to a message class where data is serialized and sent to Server for execution. Message objects are executed in Teller Class and sent to Server.

2.1.2 - Receiving

The client's UI would receive data from the server from Client Class(ATM, Employee) and display it to the Customer (ATM UI) or Teller (Employee UI).

2.2 Server side

2.2.1 - Sending

Data is serialized and using Message Class sent to Employee Class.

2.2.2 - Receiving

Date from Message class is deserialized and executed within the Banking System(Server). Data changes are saved within the Banking System(Server)'s data storage. Series of hash maps are used for easy access to objects to modify data.

2.2.3 - **Processing**

Banking System(Server) will do all functional tasks within BankingSystem(Server). The Banking System will receive data and the type of task it is executing. Data will be stored in hashmaps then lists and returned to the Client.

3. Architectural and Component-level Design

3.1 Program Structure

- The Banking System works on a Client-Server architecture; with ATM's and Employees acting as the clients, sending requests to the banking system

3.2 Description of ATM

3.2.1 ATM Processing Narrative

- Handles Member requests using Banking System Methods

  (Deposit, Withdraw, Check Balance)

3.2.2 ATM Interface Description

- Receives ATM card and verifies with Banking System

- Either Sends User to Next Screen or Responds with "Incorrect PIN"

3.2.3 ATM Processing Details

- None

3.3 Description of Banking System

3.3.1 Banking System Processing Narrative

- Handles ATM and Employee Requests own Methods:

  - Deposit

- Withdraw

- Check Balance

- Transfer Money

- Create/Close Account

- Verify Login

3.3.2 Banking System Interface Description

- Receives account numbers to act upon

- Responds with correlating output; (ie verified login, created account)

- Receives user data such as name and other info for account creation

3.3.3 Banking System Processing Details

- Checking if appropriate funds are available in a members account when a withdrawal request is made

- Checks that the debit card and PIN provided match the accounts current profile

- Logs each action taken or requested by itself, ATM and employees

3.4 Description of Employee

3.4.1 Employee Processing Narrative

- Acts as the middle man between the Member and the Banking System:

- Deposit

- Withdraw

- Check Balance

- Transfer Money

- Create/Close Account

- Check Logs

3.4.2 Employee Interface Description

- Re-directs member requests to the Banking System

- Sends Member data to Banking System for Account Creation/Deletion

3.4.3 Employee Processing Details

- Obtaining Member Info

4. User Interface Design

4.1. ATM

4.1.1 Description

- The ATM will show basic GUI options for the customers. It will perform deposit, withdraw, check balance for the bank customer as needed.

4.1.2 Images

None

4.1.3 Objects and Actions-

- Series of buttons will be located below the Panel. When clicked by the user, the screen panel is redirected to a new screen and the main panel is hidden.

4.1.4 Format

None

4.2 Teller (Employee)

4.2.1 Description

- The Employee interface will have basic GUI options to perform the requested work asked by bank customers**.**

4.2.2 Images

None

4.2.3 Objects and Actions

- It will have all the different buttons to perform the requested task and the textfields to get information from the customer.

4.2.4 Format

None

5. Restrictions, Limitations, and constraints

5.1.ATM Restrictions

- An ATM should have an active connection with the bank.
- If the customer entered invalid pin twice, end the session
- The customer needs teller permission to withdraw more than $1000.

5.2.TELLER

- Employee should provide their unique id to system to have access to perform the requested work for the bank customers
- Employee should verify/ get appropriate information from customer to close / open accounts