



ENG 101: Programming for Engineers

Final Project Description

Fall 2021



Project Title: Interactive Banking System

Submission Deadline: Before 6:00pm of November 22, 2021

Project Description:

The main objective is to implement a functional banking system using the fundamental concepts of object-oriented programming studied in class. In this project, each team is required to develop a console-based banking system, where in the user can perform the following main actions:

- Withdraw/Deposit cash
- Display remaining balance
- Modify PIN code
- Transfer cash between accounts

Similar to most ATM terminals, the program should intake some input from the user and depending on the user's action request, it should interact with the database (in this case, the database.csv file provided) to execute the action. Upon running the program, the screen should provide the following options on display:

```
Welcome to the Bank of <Your Team Name>

Card number (6 digits):
```

As shown above, the name you selected for your team should be displayed in place of <Your Team Name>. At first, the user should be asked to enter his/her card number (we assume it to be a 6-digit number). If the format of the number entered is verified with the database, then the user should be asked to enter the corresponding PIN code (we assume it to be a 4-digit number).

```
Welcome to the Bank of <Your Team Name>

Card number (6 digits): 581058
PIN code (4 digits):
```

If either the card number or PIN code is incorrect, then the program should display the error and ask the user to enter the number as follows:

```
Welcome to the Bank of <Your Team Name>

The card number or PIN code is incorrect. Try again!
Card number (6 digits):
```

Once the card number and PIN code match the entry in the database provided, the next screen should display the following options to select from:

```
The Bank of <Your Team Name>

Select one of the following options:

1. Withdraw Cash          2. Deposit Cash
3. Modify PIN Code        4. Show Balance
5. Transfer Between Accounts 6. End Session

Enter option number:
```

In what follows, each of the 6 actions will be discussed in detail:

1. Withdraw Cash:

If the user selects this option (by entering 1), then the program should display the following:

```
The Bank of <Your Team Name>

Current balance: 1,234,567 KZT

Enter amount to withdraw: 5000
```

The user should see the current amount remaining in the account, followed by the amount to be withdrawn. If the amount requested to be withdrawn exceed the current balance, an appropriate message should be displayed to the user, thus informing and allowing him/her to withdraw a lesser amount. Once the amount is withdrawn, it should be deducted from the balance and updated in the database. In the above example, if the user withdraws 5,000 KZT, the amount stored in the csv file should be $1,234,567 - 5,000 = 1,229,567$ KZT. If the action is successful, you can display something similar to what is shown below, allowing the user to return to the main menu (by typing 'y') or end the session (by typing 'n'):

```
The Bank of <Your Team Name>

Cash withdrawn successfully!
Remaining balance: 1,229,567 KZT

Would you like to return to the main menu? (Y/N)
```

2. Deposit Cash:

This action is very similar to withdrawing cash, except that the amount entered by the user should be summed with the current balance in the account and displayed accordingly.

3. Modify PIN:

The user usually selects this option if he/she wants to activate a new card or change the PIN code of an existing card. Therefore, your program should check the PIN code of the card number provided by the user as input. If the corresponding PIN code is a 4-digit number in the database, it implies that the card is an already active card and the

user intends to change the code to a new 4-digit code (probably due to security reasons!). However, if the PIN code in the database is a 4-character code (such as 2a*1, #ssa, 0_2!), it means that the card is new and has to be activated (by changing the 4-character code to a pure 4-digit code) before being used for any transactions. Once the PIN code is set, the display should have an option for the user to either return to the main menu page or exit.

4. Show Balance:

If the user opts for this action, the program should display the following information, which should be retrieved from the database (Assume USD to KZT to USD xchange rate to be 1 USD = 450 KZT):

```
The Bank of <Your Team Name>

Card owner: Pete Jones
Card number: 581058
Current balance in KZT: 1,229,567
Current balance in USD: 2,732
Last transaction: - 5000 KZT

Would you like to return to the main menu? (Y/N)
```

5. Transfer Between Accounts:

Under this option, the user is required to enter the 6-digit number of the beneficiary's account and the amount to be transferred as shown below:

```
The Bank of <Your Team Name>

**Commission for fund transfers over 20,000 KZT is 500 KZT

Beneficiary's card number: 912510
Transfer amount (in KZT): 15000

Beneficiary's name: Aresh Dadlani
Conform transfer? (Y/N): Y

Transfer successful!

Would you like to return to the main menu? (Y/N)
```

There are a number of points to be noticed while implementing this option. Firstly, the program should check and verify if the beneficiary's card number exists in the database or not. If not, it should display the error to the user and ask to re-enter the correct card number. Secondly, the fund transferred should not exceed the current amount available in the sender's account. Thirdly, if the fund being transferred is more than 20,000 KZT, the program should automatically deduct 500 KZT from the sender's account after making the transfer. For instance, the example shown above, no commission will be taken from the sender's balance as the fund being transferred is exactly 20,000 KZT. Also, keep in mind that if the sender intends to transfer fund more than 20,000 KZT, there should be at least 500 KZT remaining in the sender's account that will be then reduced for the commission fee. Otherwise, the transactions should be carried out successfully and the user should be informed about it with an appropriate display message on the screen.

Finally, if a transfer is made between cards from two different banks (indicated using the 'Bank ID field in the csv file), then 200 KZT is deducted from the sender's account irrespective of the amount being transferred.

6. End Session

If the user selects this option, the program should show a good bye message and return to the very first screenshot, where a new user has to enter a card number and PIN code to initiate an action.

Policy and Requirements:

- Each team should comprise of not more than 3 students. Though students may choose to work individually, working in teams will not only help you manage time, but also be more efficient. Therefore, we highly recommend students to team up and start working on the project ASAP.
- Students who fail to register their team on the shared excel sheet by the deadline will be randomly grouped up by the instructors. Therefore, kindly make sure you find the compatible team members on time.
- The project will be graded based on the extent to which elements of OOP (classes, encapsulation, inheritance, polymorphism, exception handling,...) have been implemented. So, make sure you think about the project in terms of objects from the very beginning.
- We have zero tolerance for non-genuine code. The code should be originally yours. Any code found to be taken off the Internet or from other sources will not be graded.
- Each team will be graded based on the following items: (i) presentation, (ii) individual contribution, (iii) teamwork, (iv) report, and (v) OOP knowledge.
- The representative of each team is responsible to upload the final (working) source code of their project along with the report to Moodle before the specified deadline. The code uploaded by the representative of each team will only be considered for evaluation during the online presentation.
- Each project will be evaluated by the instructors via Zoom. Therefore, all team members should be present and ready to answer questions raised by the instructors during the scheduled online session.
- The report should detail the design of the program developed by the team and explicitly highlight the main contributions of each team member towards the implementation of the project. Kindly note that 'preparing the presentation slides' or 'completing the report' are not considered as individual contributions. Each student should be equally involved in implementing specific modules of the project. A sample report will be uploaded to Moodle as reference.
- During the 15-minute online presentation, each team is required to prepare and present a 5-slide presentation that should explicitly include: (i) team name and members, (ii) contribution of each member in the team, (iii) examples of OOP concepts applied in the project, (iv) challenges faced and strengths of working as a team, (v) any additional features added to the project.
- Additional features (such as sound effects, graphical interface, ...) will be considered for bonus points ONLY IF the project has been implemented with perfection. It is at the instructors' discretion to whether or not give any bonus points.
- Each team should use the original database.csv file during the online project presentation.
- A Q&A session will be conducted on Friday, October 29 to clarify any further questions with regard to the final project.

❖❖ Good Luck ❖❖
