

PROJECT REPORT

Computer Programming Lab (CSL-113)



PROJECT TITLE: “Bank Easy Pro Management System”

BS (CS) – 1B

Group Members

Name	Enrollment
1. Wardha Khalid	02-134242-096
2. Maheen Fatima	02-134242-011
3. Musavir Waseem	02-134242-008

Submitted to:

Ma'am Mehwish Saleem

BAHRIA UNIVERSITY KARACHI CAMPUS

Department of Computer Science

ABSTRACT

INTRODUCTION

BankEase Pro is a user-friendly and efficient bank management system designed to handle basic banking operations. It provides essential functionalities like creating, updating, and deleting accounts, as well as adding or withdrawing money. This system is ideal for beginners who want to learn programming concepts while working on a practical project. It ensures smooth account management and simplifies financial tasks. With its straightforward interface, BankEase Pro is a great tool for understanding how banking processes work, making it both educational and practical. It's a simple yet effective way to manage accounts and improve programming skills at the same time.

PROBLEM STATEMENT:

Managing banking operations manually is inefficient, prone to errors, and leads to customer dissatisfaction due to delays in tasks like account creation, updates, and transactions. The lack of a streamlined system also increases the chances of mismanagement and data loss. To solve these challenges, an automated solution like BankEase Pro is needed. It simplifies account management by offering functionalities such as creating, updating, and deleting accounts, along with secure money transactions. This ensures accuracy, improves efficiency, and enhances user experience, providing a reliable and practical approach to modern banking operations.

METHODOLOGY:

- **Requirement Analysis:** Identify functionalities like account creation, update, deletion, adding, and withdrawing money.
- **System Design:** Plan a user-friendly interface and define data structures for storing account information securely.
- **Development:** Use a programming language like Python or Java to implement the core functionalities.
- **Testing:** Test the system with various inputs to ensure reliability, error handling, and proper functionality.
- **Implementation:** Deploy the system in a suitable environment for use.
- **Maintenance:** Update the system to fix bugs, improve performance, and add features as needed.

PROJECT SCOPE:

BankEase Pro is a basic bank management system designed to automate essential banking tasks. It includes functionalities like creating, updating, and deleting accounts, along with adding and

withdrawing money. The project focuses on simplifying account management for small-scale operations or educational purposes. It's scalable for adding advanced features like transaction history, account security, and multi-user access.

SOURCE CODE:

```
WMM CP (Global Scope) main()
1  #include <iostream>
2  #include <string>
3  #include <fstream>
4  #include <iomanip>
5  #include <cctype>
6  using namespace std;
7
8  struct account {
9      int accno;
10     string name;
11     char acctype;
12     int deposit;
13 } info_acc[3];
14
15 void intro() {
16     cout << "\n\n MADE BY : WARDHA KHALID, MAHEEN FATIMA and MUSAVIR WASEEM!";
17     cout << "\n UNIVERSITY : BAHRIA UNIVERSITY KARACHI";
18     cin.get();
19 }
20
21 void openaccount(){
22
23     for (int i = 0; i < 3; i++) {
24         cout << "\nPlease! Enter The account No. : ";
25         cin >> info_acc[i].accno;
26
27         cout << "\nPlease! Enter The Name of The account Holder: ";
28         cin.ignore();
29         getline(cin, info_acc[i].name);
30
31         cout << "\nPlease! Enter Type of The account (C/S): ";
32         cin >> info_acc[i].acctype;
33         info_acc[i].acctype = toupper(info_acc[i].acctype);
34
35         while (info_acc[i].acctype != 'C' && info_acc[i].acctype != 'S') {
36             cout << "Invalid account type! Please enter C for Current or S for Saving: ";
37             cin >> info_acc[i].acctype;
38             info_acc[i].acctype = toupper(info_acc[i].acctype);
39         }
40     }
41 }
```

```

36         cout << "Invalid account type! Please enter C for Current or S for Saving: ";
37         cin >> info_acc[i].acctype;
38         info_acc[i].acctype = toupper(info_acc[i].acctype);
39     }
40
41     cout << "\nPlease! Enter The Initial amount (>=440 for Saving and >=1000 for Current): ";
42     cin >> info_acc[i].deposit;
43
44     while ((info_acc[i].acctype == 'S' && info_acc[i].deposit < 440) ||
45            (info_acc[i].acctype == 'C' && info_acc[i].deposit < 1000)) {
46         cout << "Invalid initial deposit. Please re-enter: ";
47         cin >> info_acc[i].deposit;
48     }
49
50     cout << "\n\nAccount Created Successfully.\n";
51 }
52
53 ofstream outFile("accounts.txt");
54 if (!outFile) {
55     cerr << "Error opening file for writing!" << endl;
56     return;
57 }
58 for (int i = 0; i < 3; i++) {
59     if (info_acc[i].accno != -1) {
60         outFile << info_acc[i].accno << " "
61             << info_acc[i].name << " "
62             << info_acc[i].acctype << " "
63             << info_acc[i].deposit << endl;
64     }
65 }
66 outFile.close();
67 cout << "Account data saved to file successfully!\n";
68 }
69 }
70
71 void display() {
72     cout << "\n\n";
73     cout << "Displaying information of all account holders in Fast Bank!\n";

```

```

72     cout << "\n\n";
73     cout << "Displaying information of all account holders in Fast Bank!\n";
74     cout << "\n";
75
76     cout << left << setw(15) << "Account No."
77         << setw(25) << "Name"
78         << setw(15) << "Account Type"
79         << setw(15) << "Deposit Amount" << endl;
80
81     for (int i = 0; i < 3; i++) {
82         if (info_acc[i].accno != -1) {
83             cout << left << setw(15) << info_acc[i].accno
84                 << setw(25) << info_acc[i].name
85                 << setw(15) << info_acc[i].acctype
86                 << setw(15) << info_acc[i].deposit << endl;
87         }
88     }
89 }
90
91 void read() {
92     ifstream inFile("accounts.txt");
93     if (!inFile) {
94         cerr << "Error opening file for reading! Make sure 'accounts.txt' exists." << endl;
95         return;
96     }
97
98     int i = 0;
99     while (inFile >> info_acc[i].accno) {
100         inFile.ignore();
101         getline(inFile, info_acc[i].name, ' ');
102         inFile >> info_acc[i].acctype >> info_acc[i].deposit;
103         i++;
104         if (i >= 3) break;
105     }
106
107     inFile.close();
108     cout << "Account data loaded from file successfully!\n";
109 }

```

```

109 }
110
111
112
113
114 void sortaccounts() {
115
116     for (int i = 0; i < 3 - 1; i++) {
117         for (int j = 0; j < 3 - i - 1; j++) {
118             if (info_acc[j].accno > info_acc[j + 1].accno) {
119                 swap(info_acc[j], info_acc[j + 1]);
120             }
121         }
122     }
123     cout << "Displaying sorted accounts\n";
124     display();
125 }
126
127 void search() {
128
129     int key;
130     cout << "Enter account number you want to search: ";
131     cin >> key;
132
133     bool found = false;
134     for (int i = 0; i < 3; i++) {
135         if (info_acc[i].accno == key) {
136             cout << "\nAccount found:\n";
137             cout << left << setw(15) << "Account No."
138                 << setw(25) << "Name"
139                 << setw(15) << "Account Type"
140                 << setw(15) << "Deposit Amount" << endl;
141             cout << left << setw(15) << info_acc[i].accno
142                 << setw(25) << info_acc[i].name
143                 << setw(15) << info_acc[i].acctype
144                 << setw(15) << info_acc[i].deposit << endl;
145             found = true;
146             break;
147

```

```

146         break;
147     }
148 }
149 if (!found) {
150     cout << "\nAccount number " << key << " not found!\n";
151 }
152 }
153
154 void updatedeposit() {
155
156     int accountNumber, moreAmount;
157     cout << "\nEnter the account number you want to deposit into: ";
158     cin >> accountNumber;
159
160     bool found = false;
161     for (int i = 0; i < 3; i++) {
162         if (info_acc[i].accno == accountNumber) {
163             cout << "Enter the amount you want to deposit: ";
164             cin >> moreAmount;
165
166             if (moreAmount > 0) {
167                 info_acc[i].deposit += moreAmount;
168                 cout << "\nDeposit successful! Updated account details:\n";
169                 display();
170             }
171             else {
172                 cout << "Invalid deposit amount. Please enter a positive value.\n";
173             }
174
175             found = true;
176             break;
177         }
178     }
179
180     if (!found) {
181         cout << "Account number " << accountNumber << " not found. Please try again.\n";
182     }
183 }
184

```

```
WMM CP (Global Scope) main()
183 }
184
185 void withdraw() {
186     int accountNumber, deductAmount;
187     cout << "\nEnter the account number you want to withdraw from: ";
188     cin >> accountNumber;
189
190     bool found = false;
191     for (int i = 0; i < 3; i++) {
192         if (info_acc[i].accno == accountNumber) {
193             cout << "Enter the amount you want to withdraw: ";
194             cin >> deductAmount;
195
196             if (deductAmount > 0 && deductAmount <= info_acc[i].deposit) {
197                 info_acc[i].deposit -= deductAmount;
198                 cout << "\nWithdrawal successful! Updated account details:\n";
199                 display();
200             }
201             else {
202                 cout << "Invalid withdrawal amount. Please ensure it's positive and doesn't exceed your deposit.\n";
203             }
204
205             found = true;
206             break;
207         }
208     }
209
210     if (!found) {
211         cout << "Account number " << accountNumber << " not found. Please try again.\n";
212     }
213 }
214
215 void delaccount() {
216     int delaccount;
217     cout << "\nEnter the account number you want to delete: ";
218     cin >> delaccount;
219
220     bool found = false;
```

```
WMM CP (Global Scope) main()
220
221     bool found = false;
222     for (int i = 0; i < 3; i++) {
223         if (info_acc[i].accno == delaccount) {
224             info_acc[i].accno = -1;
225             info_acc[i].name = "";
226             info_acc[i].acctype = '\0';
227             info_acc[i].deposit = 0;
228
229             cout << "Account number " << delaccount << " deleted successfully!\n";
230             found = true;
231             break;
232         }
233     }
234
235     if (!found) {
236         cout << "Error! Account number " << delaccount << " not found. Please enter the correct account number.\n";
237     }
238 }
239
240 int main() {
241     cout << "\n*****\n";
242     cout << "\nWelcome To Bank Ease PRO Management System!\n";
243     cout << "*****\n";
244     cout << "\n.....system is starting ...../n";
245
246     intro();
247
248     int choice;
249     do {
250         cout << "\n\n\t\tMAIN MENU";
251         cout << "\n\t\t01. INPUT USER DETAILS TO CREATE NEW ACCOUNT";
252         cout << "\n\t\t02. SORT ACCOUNTS";
253         cout << "\n\t\t03. SEARCH FOR AN ACCOUNT";
254         cout << "\n\t\t04. DISPLAY ACCOUNTS";
255         cout << "\n\t\t05. DEPOSIT MORE AMOUNT";
256         cout << "\n\t\t06. WITHDRAW FROM DEPOSIT AMOUNT";
257     }
```



```
WMM CP (Global Scope) main()
257 cout << "\n\t\t06. WITHDRAW FROM DEPOSIT AMOUNT";
258 cout << "\n\t\t07. DELETE AN ACCOUNT";
259 cout << "\n\t\t08. EXIT";
260 cout << "\n\t\tSelect Your Option (1-8): ";
261 cin >> choice;
262
263 switch (choice) {
264 case 1:
265     openaccount();
266     break;
267 case 2:
268     sortaccounts();
269     break;
270 case 3:
271     search();
272     break;
273 case 4:
274     display();
275     break;
276 case 5:
277     updatedeposit();
278     break;
279 case 6:
280     withdraw();
281     break;
282 case 7:
283     delaccount();
284     break;
285 case 8:
286     cout << "\n.....system is closing ...../n";
287     cout << "\nThank you for using BANK EASE PRO :) \n";
288     break;
289 default:
290     cout << "\nInvalid choice. Please enter a valid option (1-8).\n";
291     break;
292 }
293 } while (choice != 8);
294
295 return 0;
```

```
287 cout << "\nThank you for using BANK EASE PRO :) \n";
288 break;
289 default:
290     cout << "\nInvalid choice. Please enter a valid option (1-8).\n";
291     break;
292 }
293 } while (choice != 8);
294
295 return 0;
296 }
297
```

OUTPUT:

```
Microsoft Visual Studio Debug Console

*****
"Welcome To Bank Ease PRO Management System!"
*****

.....system is starting ...../

MADE BY : WARDHA KHALID, MAHEEN FATIMA and MUSAVIR WASEEM!
UNIVERSITY : BAHRIA UNIVERSITY KARACHI

MAIN MENU
01. INPUT USER DETAILS TO CREATE NEW ACCOUNT
02. SORT ACCOUNTS
03. SEARCH FOR AN ACCOUNT
04. DISPLAY ACCOUNTS
05. DEPOSIT MORE AMOUNT
06. WITHDRAW FROM DEPOSIT AMOUNT
07. DELETE AN ACCOUNT
08. EXIT
Select Your Option (1-8): 1

Please! Enter The account No. : 666

Please! Enter The Name of The account Holder: Wardha Khalid

Please! Enter Type of The account (C/S): s

Please! Enter The Initial amount (>=440 for Saving and >=1000 for Current): 1600

Account Created Successfully.

Please! Enter The account No. : 444

Please! Enter The Name of The account Holder: Maheen Fatima
```



```
Please! Enter The Name of The account Holder: Maheen Fatima
Please! Enter Type of The account (C/S): s
Please! Enter The Initial amount (>=440 for Saving and >=1000 for Current): 2200

Account Created Successfully.
Please! Enter The account No. : 111
Please! Enter The Name of The account Holder: Musavir Waseem
Please! Enter Type of The account (C/S): c
Please! Enter The Initial amount (>=440 for Saving and >=1000 for Current): 2500

Account Created Successfully.
Account data saved to file successfully!

MAIN MENU
01. INPUT USER DETAILS TO CREATE NEW ACCOUNT
02. SORT ACCOUNTS
03. SEARCH FOR AN ACCOUNT
04. DISPLAY ACCOUNTS
05. DEPOSIT MORE AMOUNT
06. WITHDRAW FROM DEPOSIT AMOUNT
07. DELETE AN ACCOUNT
08. EXIT
Select Your Option (1-8): 4

-----
Displaying information of all account holders in Fast Bank!
```



Microsoft Visual Studio Debug Console



Displaying information of all account holders in Fast Bank!

Account No.	Name	Account Type	Deposit Amount
666	Wardha Khalid	S	1600
444	Maheen Fatima	S	2200
111	Musavir Waseem	C	2500

MAIN MENU

- 01. INPUT USER DETAILS TO CREATE NEW ACCOUNT
- 02. SORT ACCOUNTS
- 03. SEARCH FOR AN ACCOUNT
- 04. DISPLAY ACCOUNTS
- 05. DEPOSIT MORE AMOUNT
- 06. WITHDRAW FROM DEPOSIT AMOUNT
- 07. DELETE AN ACCOUNT
- 08. EXIT

Select Your Option (1-8): 2

Displaying sorted accounts

Displaying information of all account holders in Fast Bank!

Account No.	Name	Account Type	Deposit Amount
111	Musavir Waseem	C	2500
444	Maheen Fatima	S	2200
666	Wardha Khalid	S	1600

MAIN MENU

- 01. INPUT USER DETAILS TO CREATE NEW ACCOUNT
- 02. SORT ACCOUNTS
- 03. SEARCH FOR AN ACCOUNT
- 04. DISPLAY ACCOUNTS
- 05. DEPOSIT MORE AMOUNT
- 06. WITHDRAW FROM DEPOSIT AMOUNT



Microsoft Visual Studio Debug Console



```
05. DEPOSIT MORE AMOUNT
06. WITHDRAW FROM DEPOSIT AMOUNT
07. DELETE AN ACCOUNT
08. EXIT
```

```
Select Your Option (1-8): 3
```

```
Enter account number you want to search: 444
```

```
Account found:
```

Account No.	Name	Account Type	Deposit Amount
444	Maheen Fatima	S	2200

```
MAIN MENU
```

```
01. INPUT USER DETAILS TO CREATE NEW ACCOUNT
02. SORT ACCOUNTS
03. SEARCH FOR AN ACCOUNT
04. DISPLAY ACCOUNTS
05. DEPOSIT MORE AMOUNT
06. WITHDRAW FROM DEPOSIT AMOUNT
07. DELETE AN ACCOUNT
08. EXIT
```

```
Select Your Option (1-8): 5
```

```
Enter the account number you want to deposit into: 666
```

```
Enter the amount you want to deposit: 80
```

```
Deposit successful! Updated account details:
```

```
-----
Displaying information of all account holders in Fast Bank!
```

Account No.	Name	Account Type	Deposit Amount
111	Musavir Waseem	C	2500
444	Maheen Fatima	S	2200
666	Wardha Khalid	S	1680

444	Maheen Fatima	S	2200
666	Wardha Khalid	S	1680

MAIN MENU

01. INPUT USER DETAILS TO CREATE NEW ACCOUNT
02. SORT ACCOUNTS
03. SEARCH FOR AN ACCOUNT
04. DISPLAY ACCOUNTS
05. DEPOSIT MORE AMOUNT
06. WITHDRAW FROM DEPOSIT AMOUNT
07. DELETE AN ACCOUNT
08. EXIT

Select Your Option (1-8): 6

Enter the account number you want to withdraw from: 111

Enter the amount you want to withdraw: 900

Withdrawal successful! Updated account details:

Displaying information of all account holders in Fast Bank!

Account No.	Name	Account Type	Deposit Amount
111	Musavir Waseem	C	1600
444	Maheen Fatima	S	2200
666	Wardha Khalid	S	1680

MAIN MENU

01. INPUT USER DETAILS TO CREATE NEW ACCOUNT
02. SORT ACCOUNTS
03. SEARCH FOR AN ACCOUNT
04. DISPLAY ACCOUNTS
05. DEPOSIT MORE AMOUNT
06. WITHDRAW FROM DEPOSIT AMOUNT
07. DELETE AN ACCOUNT

```
Microsoft Visual Studio Debug Console
07. DELETE AN ACCOUNT
08. EXIT
Select Your Option (1-8): 7

Enter the account number you want to delete: 111
Account number 111 deleted successfully!

MAIN MENU
01. INPUT USER DETAILS TO CREATE NEW ACCOUNT
02. SORT ACCOUNTS
03. SEARCH FOR AN ACCOUNT
04. DISPLAY ACCOUNTS
05. DEPOSIT MORE AMOUNT
06. WITHDRAW FROM DEPOSIT AMOUNT
07. DELETE AN ACCOUNT
08. EXIT
Select Your Option (1-8): 4

-----
Displaying information of all account holders in Fast Bank!

Account No.      Name                Account Type  Deposit Amount
444              Maheen Fatima      S             2200
666              Wardha Khalid      S             1680

MAIN MENU
01. INPUT USER DETAILS TO CREATE NEW ACCOUNT
02. SORT ACCOUNTS
03. SEARCH FOR AN ACCOUNT
04. DISPLAY ACCOUNTS
05. DEPOSIT MORE AMOUNT
06. WITHDRAW FROM DEPOSIT AMOUNT
07. DELETE AN ACCOUNT
08. EXIT
Select Your Option (1-8): 8
```

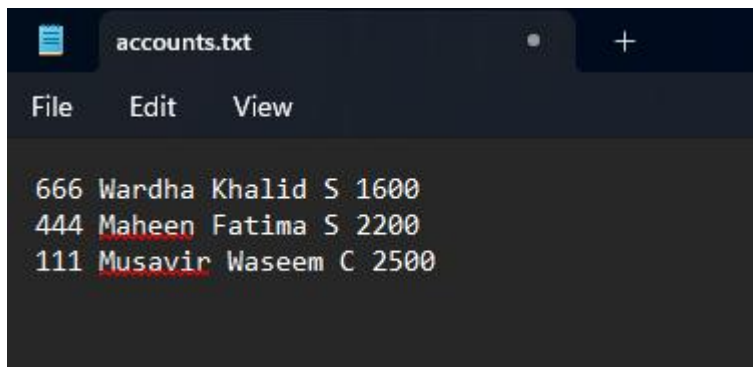
```
08. EXIT
Select Your Option (1-8): 8

.....system is closing ...../

Thank you for using BANK EASE PRO :)

C:\Users\user\Desktop\WMM CP\WMM CP\x64\Debug\WMM CP.exe (process 13412) exited with code 0 (0x0).
Press any key to close this window . . .|
```

TEXT FILE:



```
accounts.txt
File Edit View
666 Wardha Khalid S 1600
444 Maheen Fatima S 2200
111 Musavir Waseem C 2500
```

FUTURE DEVELOPMENT:

- **Transaction History:** Add functionality to record and display past transactions.
- **Authentication:** Implement user login and password protection for secure access.
- **Multi-User System:** Allow multiple users with role-based permissions.
- **Interest Calculation:** Automate interest for savings accounts.
- **Mobile Integration:** Develop a mobile app for easy access.

CONCLUSION

BankEase Pro is a user-friendly and efficient solution for basic bank management tasks. It simplifies operations like account creation, updates, and transactions, making banking more accessible and organized. This project provides a strong foundation for learning programming and system development. With future enhancements, it can evolve into a robust and practical tool for real-world