

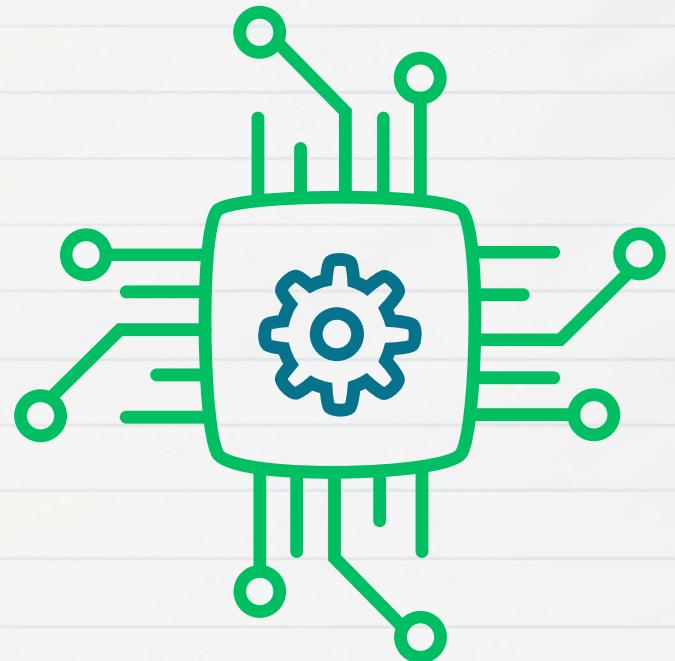
ATM MANAGEMENT SYSTEM

S.TEJASWI(RA2211003011107)
K.B.YASWANTH(RA2211003011104)
V.YASWANTH(RA2211003011123)



Overview:

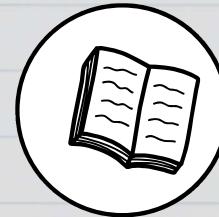
- ABSTRACT
- INTRODUCTION
- LITERATURE SURVEY
- METHODS AND METHODOLOGIES
- RESULT AND DISCUSSION
- CONCLUSION
- REFERENCE



ABSTRACT:

- THE AUTOMATED TELLER MACHINE (ATM) HAS REVOLUTIONIZED THE WAY INDIVIDUALS AND BUSINESSES CONDUCT FINANCIAL TRANSACTIONS. AS A CRITICAL COMPONENT OF MODERN BANKING, THE EFFICIENT MANAGEMENT OF ATMS IS VITAL TO ENSURE UNINTERRUPTED AND SECURE SERVICES. THIS PAPER PRESENTS AN EXPLORATION OF AN ATM MANAGEMENT SYSTEM DESIGNED TO STREAMLINE THE OPERATIONS, MAINTENANCE, AND SECURITY OF ATM NETWORKS. THROUGH A COMPREHENSIVE LITERATURE SURVEY, THIS RESEARCH AIMS TO IDENTIFY THE KEY CHALLENGES AND OPPORTUNITIES IN ATM MANAGEMENT, PROPOSING A SOLUTION TO ENHANCE THE OVERALL ATM EXPERIENCE.

INTRODUCTION:



THE AUTOMATED TELLER MACHINE (ATM) IS A UBIQUITOUS PRESENCE IN OUR DAILY LIVES, OFFERING UNPARALLELED CONVENIENCE AND ACCESSIBILITY TO FINANCIAL SERVICES. SINCE THE INCEPTION OF THE FIRST ATM IN THE LATE 1960S, THESE AUTOMATED BANKING TERMINALS HAVE UNDERGONE SIGNIFICANT TECHNOLOGICAL ADVANCEMENTS, BECOMING AN INTEGRAL PART OF MODERN BANKING SYSTEMS. THE SEAMLESS ACCESSIBILITY TO CASH WITHDRAWALS, ACCOUNT BALANCE INQUIRIES, FUND TRANSFERS, AND EVEN BILL PAYMENTS HAS TRANSFORMED THE WAY PEOPLE MANAGE THEIR FINANCES. WITH THE PROLIFERATION OF ATMS, EFFICIENT MANAGEMENT HAS BECOME A PARAMOUNT CONCERN FOR FINANCIAL INSTITUTIONS..

LITERATURE SURVEY

Paper Title	Author(s)	Year	Algorithm(s) Used	Dataset Used	Inference
ATM Security Management Using Fingerprint Recognition	Kumar, V., & Tarar, A.	2016	Fingerprint recognition, biometrics	Not specified	The study focuses on enhancing ATM security by implementing biometric authentication using fingerprint recognition.
A Survey on ATM Terminal Management	Verma, S., & Bajaj, K.	2017	Survey (methodology)	Not applicable	This survey provides an overview of the diverse range of algorithms and approaches used in ATM terminal management.

Efficient ATM Network Management System Using SNMP	Sharma, R., & Choudhary, A.	2019	Simple Network Management Protocol (SNMP)	Not specified	The paper discusses the implementation of SNMP for efficient remote monitoring and management of ATM networks.
Enhancing ATM Security with Biometric Authentication	Patel, S., & Shah, P.	2018	Fingerprint and iris recognition	Not specified	The paper suggests that integrating biometric authentication, such as fingerprint and iris recognition, enhances ATM security and user authentication.
Security Enhancement of ATM Transactions Using EMV Chip	Reddy, A. M., & Kumari, V.	2017	EMV chip technology	Not specified	The paper highlights the role of EMV chip technology in securing card-based ATM transactions, reducing the risk of card skimming and fraud.

Methods & Methodologies

1. REQUIREMENT ANALYSIS:

GATHER AND DOCUMENT THE SPECIFIC NEEDS AND EXPECTATIONS OF STAKEHOLDERS, INCLUDING BANKS, ATM SERVICE PROVIDERS, USERS.

2. SYSTEM DESIGN:

- UTILIZE UNIFIED MODELING LANGUAGE (UML) DIAGRAMS TO MODEL THE SYSTEM'S STRUCTURE AND INTERACTIONS.
- CHOOSE AN APPROPRIATE ARCHITECTURAL PATTERN TO DEFINE THE SYSTEM'S OVERALL STRUCTURE

3. DEVELOPMENT:

- APPLY OBJECT-ORIENTED PROGRAMMING (OOP) PRINCIPLES FOR MODULARITY AND MAINTAINABILITY.
- USE VERSION CONTROL SYSTEMS (E.G., GIT) TO TRACK CODE CHANGES AND COLLABORATE WITH A DEVELOPMENT TEAM.

4. DATABASE DESIGN:

- SELECT A SUITABLE RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS) FOR STRUCTURED DATA STORAGE.
- IMPLEMENT DATABASE NORMALIZATION TECHNIQUES TO EFFICIENTLY ORGANIZE DATA.

5. TESTING AND QUALITY ASSURANCE:

- CONDUCT UNIT TESTING TO VALIDATE INDIVIDUAL COMPONENTS.
- PERFORM INTEGRATION TESTING TO ENSURE DIFFERENT MODULES WORK TOGETHER COHESIVELY.
- UTILIZE SECURITY TESTING (E.G., VULNERABILITY SCANNING AND PENETRATION TESTING) TO IDENTIFY AND ADDRESS POTENTIAL SECURITY RISKS.
- INVOLVE USERS IN USER ACCEPTANCE TESTING (UAT) TO VERIFY THAT THE SYSTEM MEETS THEIR REQUIREMENTS.

CODE:

main.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4 #include <semaphore.h>
5 #include <string.h>
6
7 #define NUM_ACCOUNTS 3
8
9 - typedef struct {
10     float balance;
11     char password[20];
12     char owner[50];
13 } Account;
14
15 Account accounts[NUM_ACCOUNTS];
16
17 sem_t atm_semaphore;
18
19 - void* withdraw(void* arg) {
20     int account = ((int)arg);
21     float amount = 0;
22     char password[20];
23
24     printf("Enter your password: ");
25     scanf("%s", password);
26 }
```

```
27 if (strcmp(password, accounts[account].password) != 0) {
28     printf("X Incorrect password for Account %d\n", account);
29     pthread_exit(NULL);
30 }
31
32 printf("Enter the withdrawal amount for Account %d: $", account);
33 scanf("%f", &amount);
34
35 sem_wait(&atm_semaphore);
36 if (accounts[account].balance >= amount) {
37     accounts[account].balance -= amount;
38     printf("✓ Withdrawal from Account %d: $%.2f\n", account, amount);
39 } else {
40     printf("X Insufficient funds in Account %d\n", account);
41 }
42 printf("Account Status After Transaction:\n");
43 printf("Owner: %s\n", accounts[account].owner);
44 printf("Balance: $%.2f\n", accounts[account].balance);
45 sem_post(&atm_semaphore);
46
47 pthread_exit(NULL);
48 }
49
50 void deposit(int account) {
51     float amount = 0;
52     printf("Enter the deposit amount for Account %d: $", account);
```

```
53     scanf("%f", &amount);
54
55     sem_wait(&atm_semaphore);
56     accounts[account].balance += amount;
57     printf("✓ Deposit to Account %d: $%.2f\n", account, amount);
58     printf("Account Status After Transaction:\n");
59     printf("Owner: %s\n", accounts[account].owner);
60     printf("Balance: $%.2f\n", accounts[account].balance);
61     sem_post(&atm_semaphore);
62 }
63
64 void displayBalance(int account) {
65     printf("Account Information:\n");
66     printf("Owner: %s\n", accounts[account].owner);
67     printf("Balance: $%.2f\n", accounts[account].balance);
68 }
69
70 int main() {
71     pthread_t threads[NUM_ACCOUNTS];
72     int account_numbers[NUM_ACCOUNTS] = {0, 1, 2};
73
74     sem_init(&atm_semaphore, 0, 1);
75
76     printf("Welcome to the ATM!\n");
77     printf("Welcome to the ATM!\n");
78     printf("Welcome to the ATM!\n");
```

```
79
80 ~   for (int i = 0; i < NUM_ACCOUNTS; i++) {
81     printf("Enter owner's name for Account %d: ", i);
82     scanf("%s", accounts[i].owner);
83     printf("Enter the initial balance for Account %d: $", i);
84     scanf("%f", &accounts[i].balance);
85     printf("Set a password for Account %d: ", i);
86     scanf("%s", accounts[i].password);
87 }
88
89     int choice;
90     int account;
91
92 ~   while (1) {
93     printf("Select Account (0, 1, 2) or -1 to exit: ");
94     scanf("%d", &account);
95
96 ~     if (account == -1) {
97         break;
98     }
99
100 ~    if (account < 0 || account >= NUM_ACCOUNTS) {
101        printf("Invalid account selection.\n");
102        continue;
103    }
104 }
```

```
105     printf(" 1. Withdraw\n");
106     printf(" 2. Deposit\n");
107     printf(" 3. Check Balance\n");
108     printf(" 4. Exit\n");
109
110     printf("\n");
111
112     printf("Enter your choice: ");
113     scanf("%d", &choice);
114
115     switch (choice) {
116         case 1:
117             pthread_create(&threads[account], NULL, withdraw, &account_numbers[account]);
118             pthread_join(threads[account], NULL);
119             break;
120         case 2:
121             deposit(account);
122             break;
123         case 3:
124             displayBalance(account);
125             break;
126         case 4:
127             break;
128         default:
129             printf("Invalid choice.\n");
130     }
131
132     printf("\n");
133     printf(" Thank you for using the ATM! \n");
134     printf("\n");
135
136     sem_destroy(&atm_semaphore);
137
138     return 0;
139 }
```

Result & Discussion:

Welcome to the ATM!

Enter owner's name for Account 0: Teju
Enter the initial balance for Account 0: \$10000
Set a password for Account 0: Manju
Enter owner's name for Account 1: Yaswanth
Enter the initial balance for Account 1: \$50000
Set a password for Account 1: Yash
Enter owner's name for Account 2: Jagan
Enter the initial balance for Account 2: \$200000
Set a password for Account 2: Pawankalyan
Select Account (0, 1, 2) or -1 to exit: 0

- 1. Withdraw
- 2. Deposit
- 3. Check Balance
- 4. Exit

Enter your choice: 2
Enter the deposit amount for Account 0: \$2000
✓ Deposit to Account 0: \$2000.00
Account Status After Transaction:
Owner: Teju
Balance: \$12000.00

```
Select Account (0, 1, 2) or -1 to exit: 0
```

- 1. Withdraw
- 2. Deposit
- 3. Check Balance
- 4. Exit

```
Enter your choice: 3
```

```
Account Information:
```

```
Owner: Teju
```

```
Balance: $12000.00
```

```
Select Account (0, 1, 2) or -1 to exit: 1
```

- 1. Withdraw
- 2. Deposit
- 3. Check Balance
- 4. Exit

```
Enter your choice: 1
```

```
Enter your password: Yash
```

```
...Program finished with exit code 0
```

```
Press ENTER to exit console.
```

CONCLUSION:

ATM (AUTOMATED TELLER MACHINE) MANAGEMENT IN OPERATING SYSTEMS IS A CRITICAL ASPECT OF THE FINANCIAL INDUSTRY THAT INVOLVES THE EFFICIENT AND SECURE OPERATION OF ATM NETWORKS. THIS MANAGEMENT ENCOMPASSES A WIDE RANGE OF TASKS AND CONSIDERATIONS TO ENSURE THAT ATMS PROVIDE RELIABLE AND SECURE SERVICES TO CUSTOMERS. ATM MANAGEMENT IN OPERATING SYSTEMS ENCOMPASSES A BROAD RANGE OF FUNCTIONS, FROM ENSURING THE PHYSICAL AND DIGITAL SECURITY OF ATM MACHINES TO PROVIDING A USER-FRIENDLY AND EFFICIENT INTERFACE FOR CUSTOMERS. IT IS A COMPLEX AND CRITICAL PART OF THE FINANCIAL INDUSTRY THAT DEMANDS CAREFUL PLANNING, CONTINUOUS MONITORING, AND ADHERENCE TO INDUSTRY STANDARDS AND REGULATIONS TO MAINTAIN THE TRUST OF CUSTOMERS AND ENSURE THE SMOOTH OPERATION OF ATMS

REFERENCES AND FUTURE ENHANCEMENTS:

- Deitel, P. J., Deitel, H. M., & Choffnes, D. R. (2004). Operating Systems. Prentice Hall.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts. Wiley.
- Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems. Pearson.
- **Future Enhancements for ATM Management in Operating Systems:**
 - **Enhanced Security:** Incorporate advanced security measures such as biometric authentication, multi-factor authentication, and encryption to further protect user data and transactions.
 - **Improved User Interfaces:** Develop more user-friendly and intuitive interfaces with touchscreen capabilities and accessibility features for a wider range of users.
 - **AI and Machine Learning:** Implement AI and machine learning algorithms for transaction monitoring, fraud detection, and predictive maintenance to enhance system reliability and security.