**“SMART LOCKER SYSTEM”**

**By Team G:**

Billona, Ezekiel

Mamerto, Mark Nelson

Pontay, Ashley

Salanatin, Nathaly Pearl

**TABLE OF CONTENTS**

I. **SCENARIO/BACKGROUND**

I.I. Scenario

I.II. Current Problems

I.III. Target Beneficiaries or users of the system

**II. OBJECTIVES**

**III. SCOPE AND LIMITATIONS**

**IV. RESULTS AND DISCUSSIONS**

**V. GROUP EVALUATION**

**VI. CONCLUSION AND RECOMMENDATION**

**VII. APPENDICES**

1. **SCENARIO/BACKGROUND**

Lockers is a narrow storage compartment for storing clothing or other personal items. It is commonly used on university campuses for students to safely store their clothing or other personal belongings.

A student locker tends to contain things that are valuable to them, like pictures, books, homework, and other belongings they may need. However, with an increasing number of students on campus, managing locker access and usage has become a difficult task. A locker is supposed to be a safe box to store items without having to worry that it may be stolen or ruined.

The proposed system would implement a better locker login system that will verify students' TUPM ID and password before granting locker access. A more secured locker is undoubtedly the greatest benefit that comes with installing lockers for universities.

**PLACE TO IMPLEMENT**

* The proposed system would be implemented inside the Technological University of The Philippines.

**CITE THE DEVICES TO BE UTILIZE**

* "Printer display.exe" EXAMPLE IN EMU8086

It will serve as counting device of how often students enter their passwords incorrectly.

**OPERATION**

The proposed system will monitor locker usage patterns by tracking login date and time (operation), providing insights on peak usage times and trends to optimize locker availability and management on campus.

The Proposed system will also include input and output operations for students to enter their TUPM ID and password, which the system will verify before granting access to the lockers. This ensures that only authorized students have access to the lockers, which improves security and lowers the risk of theft or vandalism. The system will also be able to reset passwords for students who have forgotten their login credentials, allowing them to immediately access their lockers.

1. **SPECIFIC OBJECTIVES**

* The proposed system is designed to create a simple login system for lockers that verifies a students' user ID and password.
* The proposed system is mainly to share and discuss a simple "locker system" for everyone.
* To accomplish the program within a certain time.
* to develop a well function program.

1. **SCOPE AND DELIMITATIONS**

The general intent of this study is to create a safe box to store items without having to worry that it may be stolen or ruined.

The proposed system is delimited to weight inputs up to 125 only. Any input weight greater than this value will result in an incorrect output. Additionally, the program only accepts weight input in whole numbers, decimals are not supported.

1. **RESULTS AND DISCUSSION**
2. Screen shots of output

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

1. Technical problems encountered and applied solutions.

**Technical Problem Encountered:** "Pressure gauge and switches"

We have made various ways to insert the keyboard input, the Pressure gauge, and the switches (Emulation kit) in the "smart locker system" but it only generates numerous errors in the codes.

**Solution:**

We are looking for an alternative solution about the "keyboard input and pressure gauge" by simply entering sure in the window console of input and what specific weight he will put in his vault.

Since the "key board input and pressure gauge" is not working in our codes, The members have decided that the user should simply input it in our console, then if the user makes a mistake, our system will quickly pop up a warning message to the administrator.

1. **GROUP EVALUATION**

|  |  |
| --- | --- |
| Functionality | 20 |
| Reliability | 23 |
| Usability | 22 |
| Efficiency | 21 |
| **TOTAL** | **86 pts** |

1. **CONCLUSIONS AND RECOMMENDATIONS**

"Smart locker system" is a code is written in the Assembly language for the emu8086 architecture and appears to be a program for a vault security system. It initializes data for messages and passwords for three vaults, entry for a weight, date and time, and warning messages. It then displays a title and prompts the user to select a vault and enter a password. If the password is correct, the user is granted access to the vault. If the password is incorrect, the program counts the number of attempts and locks the vault after three failed attempts. It also displays an overload prompt if the entered weight exceeds 50 kg. The program finishes with the posting of a farewell message.

1. **APPENDICES**
2. Source Code

|  |  |
| --- | --- |
| **-START OF CODE-**  include emu8086.inc  name "printer"  .MODEL SMALL  .DATA  SIZE EQU 10  HEAD DB ' ',0dh, 0ah  DB ' [<---------- Security lock ---------->]',0dh, 0ah, '$'  MSG1 DB 13, 10, " Select your Vault:",13,10, 0dh, 0ah, " 1. Vault A",13,10," 2. Vault B",13,10," 3. Vault C",13,10, 0dh, 0ah," Input Only [1-3]: $"  MSG2 DB 13, 10,0dh, 0ah, ' Enter your Password: $'  MSG3 DB 13, 10, ' Invalid vault number. Please enter a number between 1 and 3.$'  MSG4 DB 13, 10, ' Wrong Password! Access denied$'    MS1 DB 13, 10,' ',13,10, ' [<---------- Welcome to Vault# 1 ---------->]',0dh, 0ah, '$'  MS2 DB 13, 10,' ',13,10, ' [<---------- Welcome to Vault# 2 ---------->]',0dh, 0ah, '$'  MS3 DB 13, 10,' ',13,10, ' [<---------- Welcome to Vault# 3 ---------->]',0dh, 0ah, '$'    MSG6 DB 13, 10, ' Too Long password!$'  MSG7 DB 13, 10, " You've entered the wrong password three times.", 0dh, 0ah  DB " Your account and locker are temporarily locked." , 0dh, 0ah  DB " Please contact the admin to recover your locker." , 0dh, 0ah  DB " The lock occurred on:$"  GOODBYE DB 0dh, 0ah, " Press any key to exit... $"  input db 13,10," Enter the weight [in KG]: $"  overweight\_msg db 0ah, 0dh, 0dh, 0ah, " We would like to inform you that your baggage",13,10, " has been found to be overweight according to our regulations.",13,10, " The weight limit for checked baggage is 50kg.",13,10,"$",13,10  welcome\_msg db 0ah, 0dh,0dh, 0ah, " We are excited to welcome you to TUP-Manila!",13,10, " We hope you have a great time as a student here. If you need",13,10, " anything, don't hesitate to reach out to us.$",13,10  TEMP\_Pass DB 1 DUP(?)  PassSize = $-Temp\_Pass  Password1 DB 1  Password2 DB 2  Password3 DB 3  Counter DB 0  vault\_select1 db 0    msg db " WARNING! WARNING! WARNING! ", 0dh, 0ah, 0dh, 0ah  msg\_vaul1 db ' ALERT:',13,10, ' Unauthorized access attempt detected.',13,10, 0dh, 0ah, ' Possible passcode ',13,10, ' input error by student or unauthorized individual.',13,10, 0dh, 0ah, ' Please investigate and take necessary actions.', 0dh, 0ah    curr\_date db 0dh,0ah, ' Error Occurred at Date: '  day db 0, 0, '-'  mont db 0, 0, '-'  year db 0, 0,    curr\_time db 0dh,0ah,' Error Occurred at Time: '  hour db 0, 0, ':'  minu db 0, 0, ':'  seco db 0, 0, ' '      msg\_end db 0      .CODE  MAIN PROC  MOV AX,@DATA  MOV DS,AX  MOV AX,0000H      Title:    LEA DX,HEAD  MOV AH,09H  INT 21H    Vault\_Select:  prompt: LEA DX,MSG1  MOV AH,09H  INT 21H    CALL scan\_num  MOV vault\_select1, CL    CMP vault\_select1, 1  JE PASS\_PROMPT  CMP vault\_select1, 2  JE PASS\_PROMPT  CMP vault\_select1, 3  JE PASS\_PROMPT    JMP ERRORMSG    PASS\_PROMPT:LEA DX,MSG2  MOV AH,09H  INT 21H  Pass\_INPUT: CALL scan\_num  CMP CL,0FH  JAE TooLong  CMP vault\_select1, 1  JE CheckPass1  CMP vault\_select1, 2  JE CheckPass2  CMP vault\_select1, 3  JE CheckPass3  JMP ERRORMSG  CheckPass1: MOV BH,00H  MOV DL,Password1[BX]  CMP CL,DL  JE CORRECT1  JMP INCORRECT    CheckPass2: MOV BH,00H  MOV DL,Password2[BX]  CMP CL,DL  JE CORRECT2  JMP INCORRECT    CheckPass3: MOV BH,00H  MOV DL,Password3[BX]  CMP CL,DL  JE CORRECT3  JMP INCORRECT    CORRECT1:  LEA DX,MS1  MOV AH,09H  INT 21H  CALL checkWeight  CORRECT2: LEA DX,MS2 | MOV AH,09H  INT 21H  CALL checkWeight  CORRECT3: LEA DX,MS3  MOV AH,09H  INT 21H  CALL checkWeight    checkWeight:  lea dx, input  mov ah, 09h  int 21h    mov dl, 10  mov bl, 0    scanNum:  mov ah, 01h  int 21h    cmp al, 13  je compare    mov ah, 0  sub al, 48    mov cl, al  mov al, bl    mul dl    add al, cl  mov bl, al    jmp scanNum    compare:  cmp bl, 50  jle welcome  jg overweight  overweight:    lea dx, overweight\_msg  mov ah, 09h  int 21h  mov dl, 0DH  mov ah, 02H  int 21H    mov dl, 0AH  mov ah, 02H  int 21H    jmp checkWeight  welcome:    lea dx, welcome\_msg  mov ah, 09h  int 21h    JMP TERMINATE    ERRORMSG:  LEA DX,MSG3  MOV AH,09H  INT 21H    MOV DL, 0DH  MOV AH, 02H  INT 21H    MOV DL, 0AH  MOV AH, 02H  INT 21H    JMP prompt  CALCU\_DECIMAL:  push cx  xor ah, ah  mov cl, 10  div cl  add ax, 3030h  pop cx  ret      LOCKED:  mov ah, 2ah  int 21h    add cx, 0f830h  mov ax, cx  call CALCU\_DECIMAL  mov w. year, ax  mov al, dh  call CALCU\_DECIMAL  mov w. mont, ax  mov al, dl  call CALCU\_DECIMAL  mov w. day, ax      mov ah, 2ch  int 21h  mov al, ch  call CALCU\_DECIMAL  mov w. hour, ax  mov al, cl  call CALCU\_DECIMAL  mov w. minu, ax  mov al, dh  call CALCU\_DECIMAL  mov w. seco, ax    mov si, offset msg  mov cx, offset msg\_end - offset msg  mov bh, 5  JMP PRINTER\_DEMO    PRINTER\_DEMO:  MOV dl, [si]  MOV ah, 5  INT 21h  INC si  LOOP PRINTER\_DEMO    MOV DX,OFFSET GOODBYE  MOV AH, 9  INT 21H    mov ax, 0  int 16h  MOV dl, 12  MOV ah, 5  INT 21h  JMP EXIT    INCORRECT:  LEA DX,MSG4  MOV AH,09H  INT 21H  MOV DL, 0DH  MOV AH, 02H  INT 21H    MOV DL, 0AH  MOV AH, 02H  INT 21H    INC Counter  CMP Counter,3  JGE LOCKED  JMP PASS\_PROMPT  TooLong: LEA DX,MSG6  MOV AH,09H  INT 21H    MOV DL, 0DH  MOV AH, 02H  INT 21H    MOV DL, 0AH  MOV AH, 02H  INT 21H  JMP PASS\_PROMPT    CLEAR\_SCREEN:    EXIT:  MOV AH, 4ch  INT 21h  DEFINE\_SCAN\_NUM  DEFINE\_GET\_STRING  TERMINATE:  END MAIN  **-END OF CODE-** |

1. Job Distribution

|  |  |  |
| --- | --- | --- |
| NAME | JOB DISTRIBUTION | E-Signature |
| **Billona, Ezekiel - Leader** | Developer/Programmer, Researcher, Quality assurance tester | Diagram  Description automatically generated with medium confidence |
| **Mamerto, Mark Nelson** | Developer/Programmer, Researcher, Quality assurance tester |  |
| **Pontay, Ashley** | Developer/Programmer, Quality assurance tester | Text, letter  Description automatically generated |
| **Salanatin, Nathaly Pearl** | Quality assurance tester, Researcher, Documentation, Video Editing |  |