

EMPLOYEE SWIPE SYSTEM

◇ PROJECT OVERVIEW:

The Employee Swipe System is a mainframe-based application developed to efficiently monitor and record employee attendance using virtual swipe-in and swipe-out functionalities. This system is built using TSO (Time Sharing Option) and CICS (Customer Information Control System), providing a real-time, multi-user, transaction-processing environment.

◇ OBJECTIVE:

- To develop a secure and centralized attendance tracking system.
 - To replace manual register entries with an automated, mainframe-based process.
 - To allow employees to swipe in/out digitally via terminal access.
 - To provide the HR department with accurate reports of working hours, absenteeism, and late entries.
-

◇ TECHNOLOGIES USED:

- Mainframe Environment
 - TSO/ISPF for editing and submitting jobs.
 - CICS for developing the front-end transaction interface.
 - COBOL for business logic and data handling.
 - VSAM files for data storage (Employee records and logs).
-

◇ SYSTEM OPERATION:

1. *Login Screen:* Employees access the CICS interface via TSO and log in using their credentials.
2. *Swipe In/Out:* Two primary transaction codes are used:
 - Swipe In (SWPIN)
 - Swipe Out (SWPOUT)
3. *Validation:* The system validates the employee ID against the master VSAM file.
4. *Timestamp Logging:* Upon a valid swipe, the system logs the timestamp in a transaction file.
5. *Error Handling:* Invalid entries prompt relevant error messages (e.g., duplicate swipe, unauthorized ID).

6. *HR Access: HR personnel can retrieve daily, weekly, or monthly reports using batch jobs or a CICS reporting interface.*
-

◇ Variables and Data Types Table:

VARIABLE NAME	DESCRIPTION	COBOL DATA TYPE / PICTURE CLAUSE	USAGE CONTEXT
EMP-ID	EMPLOYEE IDENTIFICATION NUMBER	PIC X(6)	UNIQUE ID FOR EACH EMPLOYEE
EMP-NAME	EMPLOYEE FULL NAME	PIC X(30)	DISPLAY ON REPORTS/SCREENS
SWIPE-DATE	DATE OF SWIPE	PIC 9(8) (FORMAT: YYYYMMDD)	SWIPE LOG DATE
SWIPE-TIME	TIME OF SWIPE	PIC 9(6) (FORMAT: HHMMSS)	SWIPE IN/OUT TIME
SWIPE-TYPE	TYPE OF ACTION (IN/OUT)	PIC X(3)	INDICATES "IN" OR "OUT"
TRANS-ID	CICS TRANSACTION ID	PIC X(4)	E.G., T128 OR R128
MSG-TEXT	MESSAGE DISPLAY TEXT	PIC X(80)	FOR ERROR/SUCCESS MESSAGES
RESPONSE-CODE	RETURN CODE FROM OPERATIONS	PIC S9(08) COMP	FOR INTERNAL VALIDATION RESULTS
VSAM-STATUS	STATUS CODE FOR VSAM FILE OPERATIONS	PIC X(2)	TO HANDLE FILE READ/WRITE STATUS

PROGRAM CODE:

1. ASSEMBLER CODE[MAPSET]:

PRINT NOGEN

OZA128R DFHMSD TYPE=&SYSPARM,TIOAPFX=YES,LANG=COBOL,MODE=INOUT,

STORAGE=AUTO,CTRL=(FREEKB,FRSET),MAPATTS=COLOR,

DSATTS=COLOR

MPAGE DFHMDI SIZE=(24,80),LINE=1,COLUMN=1

**DFHMDF POS=(05,30),LENGTH=22,INITIAL='EMPLOYEE SWIPE SYSTEM',
ATTRB=(PROT,NORM)**

DFHMDF POS=(06,30),LENGTH=22,INITIAL='***',
ATTRB=(PROT,NORM)**

**DFHMDF POS=(08,10),LENGTH=10,INITIAL='OPTIONS: ',
ATTRB=(PROT,NORM)**

**DFHMDF POS=(11,30),LENGTH=13,INITIAL='1.SWIPE IN. ',
ATTRB=(PROT,NORM)**

**DFHMDF POS=(13,30),LENGTH=14,INITIAL='2.SWIPE OUT. ',
ATTRB=(PROT,NORM)**

**DFHMDF POS=(15,30),LENGTH=18,INITIAL='3.NEW REGISTRATION',
ATTRB=(PROT,NORM)**

**DFHMDF POS=(17,30),LENGTH=09,INITIAL='4.EXIT. ',
ATTRB=(PROT,NORM)**

**DFHMDF POS=(20,05),LENGTH=17,INITIAL='ENTER THE OPTION:',
ATTRB=(PROT,NORM)**

**OPTION DFHMDF POS=(20,23),LENGTH=02,INITIAL=' ',
ATTRB=(UNPROT,NORM,IC),PICIN='X(2)',PICOUT='X(2)'**

**DFHMDF POS=(23,05),LENGTH=10,INITIAL='MESSAGE:',
ATTRB=(PROT,NORM)**

**MSG1 DFHMDF POS=(23,17),LENGTH=23,INITIAL='_____ ',
ATTRB=(PROT,NORM),PICOUT='X(23)'**

LGON DFHMDF SIZE=(24,80),LINE=1,COLUMN=1

**HEAD2 DFHMDF POS=(03,04),LENGTH=21,INITIAL='SWIPED IN SUCCESSFUL',
ATTRB=(PROT,NORM),COLOR=YELLOW**

**DFHMDF POS=(05,04),LENGTH=08,INITIAL='EMPID :',
ATTRB=(PROT,NORM)**

**EMPID DFHMDF POS=(05,15),LENGTH=07,INITIAL='____ ',
ATTRB=(PROT,NORM)**

**DFHMDF POS=(07,04),LENGTH=10,INITIAL='EMP NAME:',
ATTRB=(PROT,NORM)**

EMPNAME DFHMDF POS=(07,15),LENGTH=23,INITIAL=' ',

ATTRB=(PROT,NORM)

DFHMDF POS=(09,04),LENGTH=06,INITIAL='DATE:',

ATTRB=(PROT,NORM)

DATE1 DFHMDF POS=(09,15),LENGTH=13,INITIAL=' ',

ATTRB=(PROT,NORM)

DFHMDF POS=(11,04),LENGTH=05,INITIAL='TIME:',

ATTRB=(PROT,NORM)

TIME1 DFHMDF POS=(11,15),LENGTH=10,INITIAL=' ',

ATTRB=(PROT,NORM)

DFHMDF POS=(15,04),LENGTH=17,INITIAL='ENTER Y TO EXIT:',

ATTRB=(PROT,NORM)

RETURN DFHMDF POS=(15,22),LENGTH=01,INITIAL='_',

ATTRB=(UNPROT,NORM),PICIN='X(01)',PICOUT='X(01)'

SWIN DFHMDI SIZE=(24,80),LINE=1,COLUMN=1

HEAD1 DFHMDF POS=(03,38),LENGTH=10,INITIAL='SWIPE IN',

ATTRB=(PROT,NORM)

DFHMDF POS=(05,35),LENGTH=14,INITIAL='ENTER DETAILS',

ATTRB=(PROT,NORM)

DFHMDF POS=(08,15),LENGTH=07,INITIAL='EMPID: ',

ATTRB=(PROT,NORM)

EID DFHMDF POS=(08,25),LENGTH=06,INITIAL='_____ ',

ATTRB=(UNPROT,NORM,IC),PICIN='X(06)',PICOUT='X(06)'

DFHMDF POS=(09,15),LENGTH=07,INITIAL='NAME: ',

ATTRB=(PROT,NORM)

NAME DFHMDF POS=(09,25),LENGTH=20,INITIAL='_____ ',

ATTRB=(UNPROT,NORM),PICIN='X(20)',PICOUT='X(20)'

DFHMDF POS=(23,05),LENGTH=10,INITIAL='MESSAGE: ',

ATTRB=(PROT,NORM)

MSG2 DFHMDF POS=(23,17),LENGTH=23,INITIAL='_____ ',

ATTRB=(PROT,NORM)

REG DFHMDI SIZE=(24,80),LINE=1,COLUMN=1

REGH DFHMDF POS=(05,30),LENGTH=18,INITIAL='ENTER THE DETAILS',

ATTRB=(PROT,NORM)

DFHMDF POS=(07,25),LENGTH=17,INITIAL='EMP ID :',

ATTRB=(PROT,NORM)

EMPIDR DFHMDF POS=(07,43),LENGTH=06,INITIAL='_____ ',

ATTRB=(UNPROT,NORM,IC),PICIN='X(06)',PICOUT='X(06)'

DFHMDF POS=(09,25),LENGTH=17,INITIAL='NAME :',

ATTRB=(PROT,NORM)

NAMER DFHMDF POS=(09,43),LENGTH=20,INITIAL='_____ ',

ATTRB=(UNPROT,NORM),PICIN='X(20)',PICOUT='X(20)'

DFHMDF POS=(11,25),LENGTH=17,INITIAL='PHONE NO. :',

ATTRB=(PROT,NORM)

PHONER DFHMDF POS=(11,43),LENGTH=10,INITIAL='_____ ',

ATTRB=(UNPROT,NORM),PICIN='X(10)',PICOUT='X(10)'

DFHMDF POS=(14,25),LENGTH=17,INITIAL='DATE OF BIRTH :',

ATTRB=(PROT,NORM)

DOBR DFHMDF POS=(14,43),LENGTH=10,INITIAL='_____ ',

ATTRB=(UNPROT,NORM),PICIN='X(10)',PICOUT='X(10)'

DFHMDF POS=(16,25),LENGTH=17,INITIAL='DATE OF JOINING :',

ATTRB=(PROT,NORM)

DOJR DFHMDF POS=(16,43),LENGTH=10,INITIAL='_____ ',

ATTRB=(UNPROT,NORM),PICIN='X(10)',PICOUT='X(10)'

DFHMDF POS=(20,05),LENGTH=16,INITIAL='ENTER Y TO EXIT:',

ATTRB=(PROT,NORM)

EXR DFHMDF POS=(20,22),LENGTH=01,INITIAL='_',

ATTRB=(UNPROT,NORM),PICIN='X(01)',PICOUT='X(01)'

DFHMDF POS=(23,05),LENGTH=08,INITIAL='MESSAGE:',

ATTRB=(PROT,NORM)

RMI DFHMDF POS=(23,15),LENGTH=25,INITIAL='_____ ',

ATTRB=(PROT,NORM)

REGM DFHMDI SIZE=(24,80),LINE=1,COLUMN=1

DFHMDF POS=(10,30),LENGTH=19,INITIAL='EMPLOYEE NOT EXIST',

ATTRB=(PROT,NORM)

```

DFHMDF POS=(15,25),LENGTH=29,
INITIAL='ENTER Y FOR NEW REGISTRATION:',
ATTRB=(PROT,NORM)
RM DFHMDF POS=(15,55),LENGTH=01,INITIAL='_',
ATTRB=(UNPROT,NORM),PICIN='X(01)',PICOUT='X(01)'
EXIT DFHMDI SIZE=(24,80),LINE=1,COLUMN=1
DFHMDF POS=(12,30),LENGTH=13,INITIAL='SYSTEM ENDED',
ATTRB=(PROT,NORM)
DFHMSD TYPE=FINAL
END

```

2. MAIN PROGRAM:

IDENTIFICATION DIVISION.

PROGRAM-ID. KSEQPG.

DATA DIVISION.

WORKING-STORAGE SECTION.

COPY OZA128R.

COPY DFHBMSCA.

COPY DFHAID.

01 WS-COMMAREA.

10 WS-TRANSID PIC X(04).

LINKAGE SECTION.

01 DFHCOMMAREA.

10 LS-TRANSID PIC X(04).

PROCEDURE DIVISION.

MAIN-PARA.

MOVE LOW-VALUES TO MPAGEI MPAGEO.

PERFORM 1000-SEND-PARA

THRU 1000-SEND-END.

PERFORM 2000-RECEIVE-PARA

THRU 2000-RECEIVE-END.
PERFORM 9999-RETURN-PARA
THRU 9999-RETURN-END.

1000-SEND-PARA.

EXEC CICS SEND

MAP('MPAGE')
MAPSET('OZA128R')
ERASE
END-EXEC.

1000-SEND-END.

EXIT.

1900-RETURN-PARA.

EXEC CICS RETURN
TRANSID('O128')
END-EXEC.

1900-RETURN-END.

EXIT.

2000-RECEIVE-PARA.

EXEC CICS SEND
MAP('MPAGE')
MAPSET('OZA128R')
END-EXEC
EXEC CICS RECEIVE
MAP('MPAGE')
MAPSET('OZA128R')
END-EXEC
EVALUATE OPTIONI
WHEN 1
EXEC CICS LINK

```
PROGRAM('OZA128T1')
END-EXEC

WHEN 2
EXEC CICS LINK
PROGRAM('OZA128T2')
END-EXEC

WHEN 3
EXEC CICS LINK
PROGRAM('OZA128T3')
END-EXEC

WHEN 4
EXEC CICS LINK
PROGRAM('OZA128T4')
END-EXEC

WHEN OTHER
MOVE 'INVALID KEY' TO MSG10
EXEC CICS SEND
MAP('MPAGE')
MAPSET('OZA128R')
ERASE
END-EXEC
END-EVALUATE.

2000-RECEIVE-END.

EXIT.
```

```
9999-RETURN-PARA.
EXEC CICS RETURN END-EXEC.
9999-RETURN-END.
EXIT.
```

3. SUBPROGRAM-1:

IDENTIFICATION DIVISION.

PROGRAM-ID. SWIPEI.

DATA DIVISION.

WORKING-STORAGE SECTION.

COPY OZA128R.

COPY DFHBMSCA.

COPY DFHAID.

01 WS-BUFF.

10 WS-KEY PIC X(06).

10 SP1 PIC X(01) VALUE SPACES.

10 WS-NAME PIC X(20).

10 SP2 PIC X(01) VALUE SPACES.

10 WS-DATE PIC X(11).

10 SP3 PIC X(01) VALUE SPACES.

10 WS-TIME PIC X(09).

01 WS-COMMAREA.

10 WS-TRANSID PIC X(04).

01 WS-RESP PIC S9(08) COMP.

77 WS-ABSDT PIC S9(15) COMP.

77 WS-FDATE PIC X(10).

77 WS-FTIME PIC X(08).

LINKAGE SECTION.

01 DFHCOMMAREA.

10 LS-TRANSID PIC X(04).

PROCEDURE DIVISION.

MAIN-PARA.

EXEC CICS ASKTIME
ABSTIME(WS-ABS DT)
END-EXEC

EXEC CICS FORMATTIME
ABSTIME(WS-ABS DT)
TIME(WS-FTIME)
TIMESEP(:)
DDMMYYYY(WS-FDATE)
DATESEP(/')
END-EXEC

MOVE LOW-VALUES TO SWINI SWINO.
MOVE 'SWIPE IN' TO HEAD1I.
PERFORM 1000-RECEIVE-PARA
THRU 1000-RECEIVE-END.

1000-RECEIVE-PARA.
EXEC CICS SEND
MAP('SWIN')
MAPSET('OZA128R')
ERASE
END-EXEC

EXEC CICS RECEIVE
MAP('SWIN')
MAPSET('OZA128R')
END-EXEC

MOVE LOW-VALUES TO WS-BUFF.
MOVE EIDI TO WS-KEY.

```
EXEC CICS READ
  DATASET('OZA128F')
  INTO(WS-BUFF)
  RIDFLD(WS-KEY)
  RESP(WS-RESP)
  UPDATE
  LENGTH(LENGTH OF WS-BUFF)
END-EXEC
```

```
MOVE NAMEI TO WS-NAME.
MOVE WS-FDATE TO WS-DATE.
MOVE WS-FTIME TO WS-TIME.
```

```
EXEC CICS REWRITE
  DATASET('OZA128F')
  FROM(WS-BUFF)
  RESP(WS-RESP)
END-EXEC
```

```
IF WS-RESP = DFHRESP(NORMAL) THEN
  MOVE 'EMPLOYEE DATA ADDED!' TO MSG2I
  MOVE LOW-VALUES TO LGONI LGONO
  MOVE 'SWIPED IN SUCCESSFUL' TO HEAD2I
  MOVE WS-KEY TO EMPIDO
  MOVE WS-NAME TO EMPNAMEO
  MOVE WS-DATE TO DATE1O
  MOVE WS-TIME TO TIME1O
```

```
EXEC CICS SEND
  MAP('LGON')
  MAPSET('OZA128R')
  ERASE
```

END-EXEC

EXEC CICS RECEIVE

MAP('LGON')

MAPSSET('OZA128R')

END-EXEC

IF RETURNI = 'Y'

EXEC CICS LINK

PROGRAM('OZA128T')

END-EXEC

EXEC CICS RETURN END-EXEC

END-IF

ELSE

EXEC CICS LINK

PROGRAM('OZA128T5')

END-EXEC

END-IF.

1000-RECEIVE-END.

EXIT.

9999-END-PARA.

EXEC CICS RETURN END-EXEC.

9999-END.

EXIT.

4. SUBPROGRAM-2:

IDENTIFICATION DIVISION.

PROGRAM-ID. SWIPEO.

DATA DIVISION.

WORKING-STORAGE SECTION.

COPY OZA128R.

COPY DFHBMSCA.

COPY DFHAID.

01 WS-BUFF.

10 WS-KEY PIC X(06).

10 SP1 PIC X(01) VALUE SPACES.

10 WS-NAME PIC X(20).

10 SP2 PIC X(01) VALUE SPACES.

10 WS-DATE PIC X(11).

10 SP3 PIC X(01) VALUE SPACES.

10 WS-TIME PIC X(09).

01 WS-COMMAREA.

10 WS-TRANSID PIC X(04).

01 WS-RESP PIC S9(08) COMP.

77 WS-ABSDT PIC S9(15) COMP.

77 WS-FDATE PIC X(10).

77 WS-FTIME PIC X(08).

LINKAGE SECTION.

01 DFHCOMMAREA.

10 LS-TRANSID PIC X(04).

PROCEDURE DIVISION.

MAIN-PARA.

EXEC CICS ASKTIME

ABSTIME(WS-ABSDT)

END-EXEC

EXEC CICS FORMATTIME

ABSTIME(WS-ABSDT)

TIME(WS-FTIME)

TIMESEP(:')

DDMMYYYY(WS-FDATE)

DATESEP('/')

END-EXEC

MOVE LOW-VALUES TO SWINI SWINO.

MOVE 'SWIPE OUT' TO HEAD1I.

PERFORM 1000-RECEIVE-PARA

THRU 1000-RECEIVE-END.

1000-RECEIVE-PARA.

EXEC CICS SEND

MAP('SWIN')

MAPSET('OZA128R')

ERASE

END-EXEC.

EXEC CICS RECEIVE

MAP('SWIN')

MAPSET('OZA128R')

END-EXEC

MOVE LOW-VALUES TO WS-BUFF.

MOVE EIDI TO WS-KEY.

```
EXEC CICS READ
  DATASET('OZA128F1')
  INTO(WS-BUFF)
  RIDFLD(WS-KEY)
  RESP(WS-RESP)
  UPDATE
  LENGTH(LENGTH OF WS-BUFF)
END-EXEC
```

```
MOVE NAMEI TO WS-NAME.
MOVE WS-FDATE TO WS-DATE.
MOVE WS-FTIME TO WS-TIME.
```

```
EXEC CICS REWRITE
  DATASET('OZA128F1')
  FROM(WS-BUFF)
  RESP(WS-RESP)
END-EXEC.
```

```
IF WS-RESP = DFHRESP(NORMAL) THEN
  MOVE 'EMPLOYEE DATA ADDED!' TO MSG2I
  MOVE LOW-VALUES TO LGONI LGONO
  MOVE 'SWIPED OUT SUCCESSFUL' TO HEAD2I
  MOVE WS-KEY TO EMPIDO
  MOVE WS-NAME TO EMPNAMEO
  MOVE WS-DATE TO DATE1O
  MOVE WS-TIME TO TIME1O
```

```
EXEC CICS SEND
  MAP('LGON')
  MAPSET('OZA128R')
```

ERASE

END-EXEC

EXEC CICS RECEIVE

MAP('LGON')

MAPSSET('OZA128R')

END-EXEC

IF RETURNI = 'Y'

EXEC CICS LINK

PROGRAM('OZA128T')

END-EXEC

EXEC CICS RETURN END-EXEC

END-IF

ELSE

MOVE 'ERROR IN EMP FILE : ' TO MSG2I

EXEC CICS LINK

PROGRAM('OZA128T5')

END-EXEC

END-IF.

1000-RECEIVE-END.

EXIT.

9999-END-PARA.

EXEC CICS RETURN END-EXEC.

9999-END.

EXIT.

5. SUBPROGRAM-3:

IDENTIFICATION DIVISION.

PROGRAM-ID. ERR.

DATA DIVISION.

WORKING-STORAGE SECTION.

COPY OZA128R.

COPY DFHBMSCA.

COPY DFHAID.

LINKAGE SECTION.

01 DFHCOMMAREA.

10 LS-TRANSID PIC X(04).

PROCEDURE DIVISION.

MAIN-PARA.

MOVE LOW-VALUES TO REGMI REGMO.

PERFORM 1000-SEND-PARA

THRU 1000-SEND-END.

PERFORM 2000-RECEIVE-PARA

THRU 2000-RECEIVE-END.

PERFORM 9999-END-PARA

THRU 9999-END.

1000-SEND-PARA.

EXEC CICS SEND

MAP('REGM')

MAPSET('OZA128R')

ERASE

END-EXEC.

1000-SEND-END.

EXIT.

2000-RECEIVE-PARA.

EXEC CICS RECEIVE

MAP('REGM')

MAPSET('OZA128R')

END-EXEC

IF RMI = 'Y'

EXEC CICS LINK

PROGRAM('OZA128T3')

END-EXEC

END-IF.

2000-RECEIVE-END.

EXIT.

9999-END-PARA.

EXEC CICS RETURN END-EXEC.

9999-END.

EXIT.

6. SUBPROGRAM-4:

IDENTIFICATION DIVISION.

PROGRAM-ID. NREG.

DATA DIVISION.

WORKING-STORAGE SECTION.

COPY OZA128R.

COPY DFHBMSCA.

COPY DFHAID.

01 WS-BUFF1.

10 WS-KEY PIC X(06).
10 FILLER PIC X(01).
10 WS-NAME PIC X(20).
10 FILLER PIC X(01).
10 WS-PHONE PIC X(10).
10 FILLER PIC X(01).
10 WS-DOB PIC X(10).
10 FILLER PIC X(01).
10 WS-DOJ PIC X(10).
10 FILLER PIC X(20).

01 WS-COMMAREA.

10 WS-TRANSID PIC X(04).

01 WS-RESP PIC S9(08) COMP.

LINKAGE SECTION.

01 DFHCOMMAREA.

10 LS-TRANSID PIC X(04).

PROCEDURE DIVISION.

MAIN-PARA.

MOVE LOW-VALUES TO REGI REGO.

PERFORM 1000-SEND-PARA

THRU 1000-SEND-END.

PERFORM 2000-RECEIVE-PARA

THRU 2000-RECEIVE-END.

PERFORM 9999-END-PARA

THRU 9999-END.

1000-SEND-PARA.

EXEC CICS SEND

MAP('REG')
MAPSET('OZA128R')
ERASE
END-EXEC.
1000-SEND-END.
EXIT.

2000-RECEIVE-PARA.

EXEC CICS RECEIVE
MAP('REG')
MAPSET('OZA128R')
END-EXEC
MOVE LOW-VALUES TO WS-BUFF1
MOVE EMPIDRI TO WS-KEY
MOVE NAMEI TO WS-NAME
MOVE PHONERI TO WS-PHONE
MOVE DOBRI TO WS-DOB
MOVE DOJRI TO WS-DOJ

EXEC CICS WRITE
DATASET('OZA128F3')
FROM(WS-BUFF1)
LENGTH(LENGTH OF WS-BUFF1)
RIDFLD(WS-KEY)
RESP(WS-RESP)
END-EXEC.

MOVE LOW-VALUES TO WS-BUFF1
MOVE EMPIDRI TO WS-KEY

EXEC CICS WRITE
DATASET('OZA128F')

FROM(WS-BUFF1)
LENGTH(LENGTH OF WS-BUFF1)
RIDFLD(WS-KEY)
RESP(WS-RESP)
END-EXEC.

EXEC CICS WRITE
DATASET('OZA128F1')
FROM(WS-BUFF1)
LENGTH(LENGTH OF WS-BUFF1)
RIDFLD(WS-KEY)
RESP(WS-RESP)
END-EXEC.

IF WS-RESP = DFHRESP(NORMAL) THEN
MOVE 'REGISTERED SUCCESSFUL' TO RMII
EXEC CICS SEND
MAP('REG')
MAPSET('OZA128R')
ERASE
END-EXEC
EXEC CICS RECEIVE
MAP('REG')
MAPSET('OZA128R')
ERASE
END-EXEC
IF EXRI = 'Y'
EXEC CICS LINK
PROGRAM('OZA128T')
END-EXEC
END-IF
ELSE

MOVE 'REGISTRATION FAILED' TO RMII
EXEC CICS SEND
 MAP('REG')
 MAPSET('OZA128R')
 ERASE
END-EXEC
EXEC CICS RECEIVE
 MAP('REG')
 MAPSET('OZA128R')
 ERASE
END-EXEC
IF EXRI = 'Y'
 EXEC CICS LINK
 PROGRAM('OZA128T')
 END-EXEC
END-IF
END-IF.

2000-RECEIVE-END.

EXIT.

9999-END-PARA.

EXEC CICS RETURN END-EXEC.

9999-END.

EXIT.

7. SUBPROGRAM-5:

IDENTIFICATION DIVISION.

PROGRAM-ID. SWIPEO.

DATA DIVISION.

WORKING-STORAGE SECTION.

COPY OZA128R.

COPY DFHBMSCA.

COPY DFHAID.

01 WS-BUFF.

10 WS-KEY PIC X(06).

10 WS-DATA PIC X(74).

01 WS-COMMAREA.

10 WS-TRANSID PIC X(04).

01 WS-RESP PIC S9(08) COMP.

77 WS-FDATE PIC X(10).

77 WS-TIME PIC X(08).

77 WS-ABSDT PIC S9(15) COMP.

LINKAGE SECTION.

01 DFHCOMMAREA.

10 LS-TRANSID PIC X(04).

PROCEDURE DIVISION.

MAIN-PARA.

MOVE LOW-VALUES TO EXITI EXITO.

PERFORM 1000-EXIT-PARA

THRU 1000-EXIT-END.

PERFORM 9999-END-PARA

THRU 9999-END.

1000-EXIT-PARA.

EXEC CICS SEND

MAP('EXIT')

MAPSET('OZA128R')

ERASE

END-EXEC.

1000-EXIT-END.

EXIT.

9999-END-PARA.

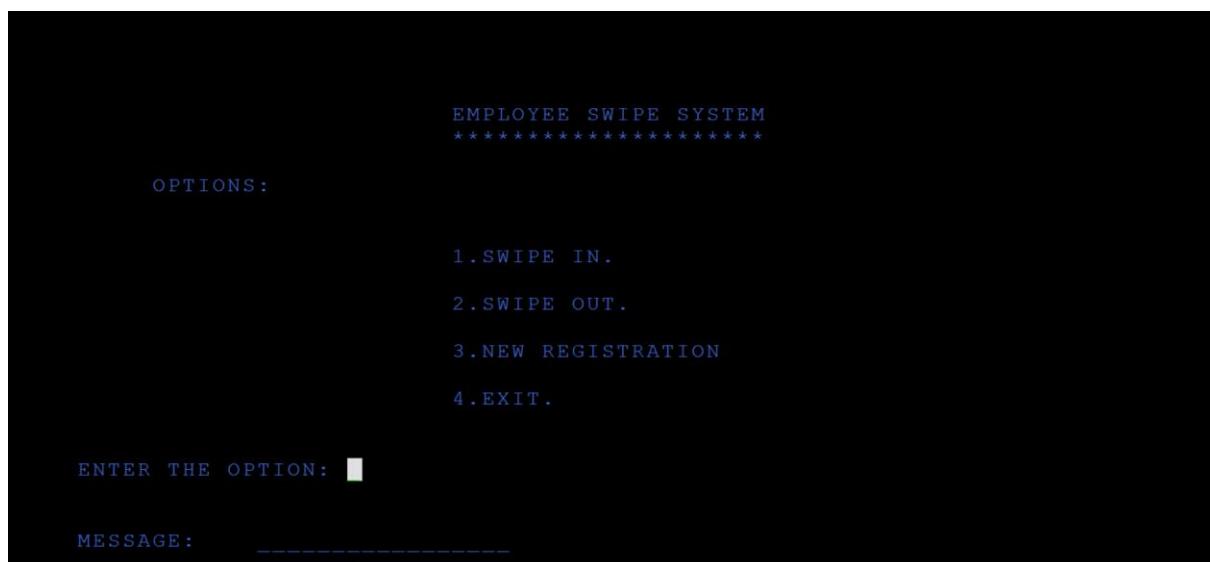
EXEC CICS RETURN END-EXEC.

9999-END.

EXIT.

THE FINAL OUTPUT DISPLAY :

1. *The menu page where the user can choose to swipe in/swipe out/new registration/exit based on the options they enter respectively. The options (1/2/3/4) can be entered in the respective field to choose the respective action.*



2. The option 1 is for swipe-in action.

```
EMPLOYEE SWIPE SYSTEM
* * * * *
OPTIONS :
1 . SWIPE IN .
2 . SWIPE OUT .
3 . NEW REGISTRATION
4 . EXIT .

ENTER THE OPTION: 1
MESSAGE : -----
```

3. The swipe-in page, where the user needs to enter their employee ID (empid) and employee name(name).

```
SWIPE IN
ENTER DETAILS

EMPID: [REDACTED]
NAME: [REDACTED]

MESSAGE : -----
```

4. The user enters the respective details on the respective fields.

```
SWIPE IN
ENTER DETAILS

EMPID: 000001
NAME: janish e [REDACTED]

MESSAGE : -----
```

5. After proper data validation the success message will be displayed with the swipe-in details like empid, emp name, date, time of the swipe-in.

```
SWIPIED IN SUCCESSFUL
EMPIID :      000001
EMP NAME :    JANISH E
DATE :        15/05/2025
TIME :        10:49:58

ENTER Y TO EXIT: _
```

6. After the success message the user can enter 'y' in the respective field to exit back to the menu page.

```
SWIPIED IN SUCCESSFUL

EMPIID :      000001

EMP NAME :    JANISH E

DATE :        15/05/2025

TIME :        10:49:58

ENTER Y TO EXIT: y
```

7. Option 2 is for swipe-out action.

```
EMPLOYEE SWIPE SYSTEM
*****
OPTIONS:
1 . SWIPE IN .
2 . SWIPE OUT .
3 . NEW REGISTRATION
4 . EXIT .

ENTER THE OPTION: 2
MESSAGE: -----
```

8. The swipe-out page has the fields for empid and name.

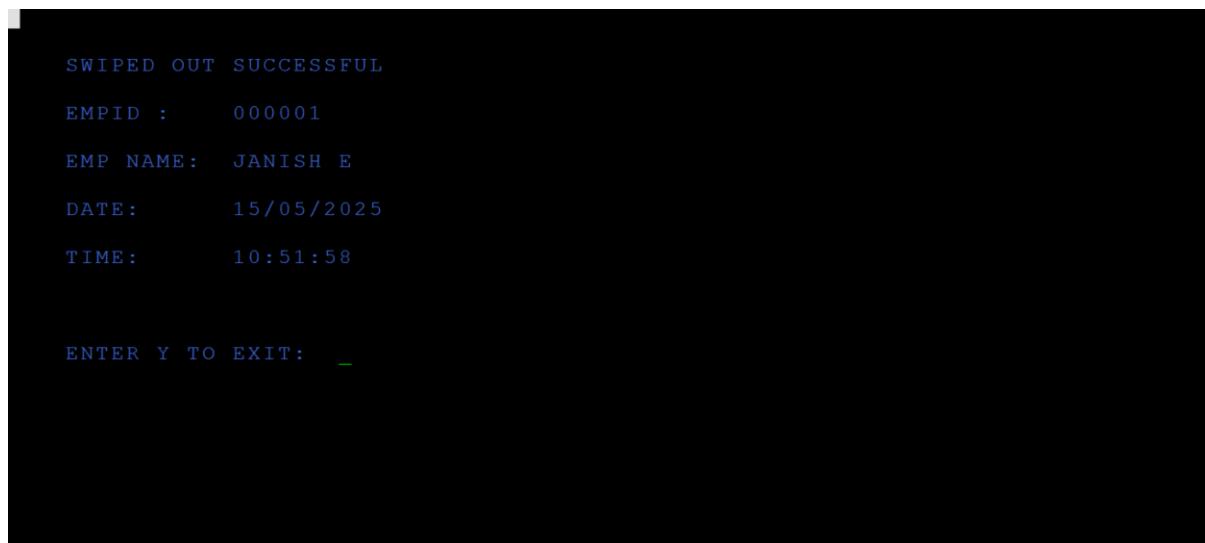
```
SWIPE OUT
ENTER DETAILS

EMPID: _____
NAME: _____
MESSAGE: -----
```

9. Press enter to end action After entering valid data in the respective fields.



10. The swipe-out success message will be displayed with the swipe-out date and time. The user can enter 'y' in the respective field to exit to the menu page.



SWIPED OUT SUCCESSFUL

EMPID : 000001

EMP NAME: JANISH E

DATE: 15/05/2025

TIME: 10:51:58

ENTER Y TO EXIT: Y

11. The option 3 is for the new registration of the employee whose details are not stored in the record.

EMPLOYEE SWIPE SYSTEM

OPTIONS:

- 1 . SWIPE IN.
- 2 . SWIPE OUT.
- 3 . NEW REGISTRATION
- 4 . EXIT.

ENTER THE OPTION: 3

MESSAGE: -----

12. This opens the registration page where the employee data can be filled in the respective fields and the data is recorded in the VSAM dataset.

ENTER THE DETAILS

EMP ID :

NAME :

PHONE NO. :

DATE OF BIRTH :

DATE OF JOINING :

ENTER Y TO EXIT:

MESSAGE:

ENTER THE DETAILS

EMP ID : 000002

NAME : pranesh akilan a r

PHONE NO. : 8667484160

DATE OF BIRTH : 31/10/2003

DATE OF JOINING : 01/06/2025

ENTER Y TO EXIT:

MESSAGE:

13. If the registration is done successfully with valid data the success message will be displayed.

```
ENTER THE DETAILS
EMP ID : 00002
NAME : PRANESH AKILAN A R
PHONE NO. : 8667484160

DATE OF BIRTH : 31/10/2003
DATE OF JOINING : 01/06/2025

ENTER Y TO EXIT: _  
MESSAGE: REGISTERED SUCCESSFUL
```

14. Enter 'y' In the respective field to exit and navigate to the menu page.

```
ENTER THE DETAILS
EMP ID : 00002
NAME : PRANESH AKILAN A R
PHONE NO. : 8667484160

DATE OF BIRTH : 31/10/2003
DATE OF JOINING : 01/06/2025

ENTER Y TO EXIT: y  
MESSAGE: REGISTERED SUCCESSFUL
```

15. If a new empid is entered in the swipe-in page, it will display that “Employee not exist” and guide the user to the registration page.



16. Now the user can register the new employee and save it to the record.

```
ENTER THE DETAILS
EMP ID : 000003
NAME : rakul v
PHONE NO. : 9487246880

DATE OF BIRTH : 16/10/2003
DATE OF JOINING : 01/06/2025

ENTER Y TO EXIT: _  
MESSAGE: -----
```

```
ENTER THE DETAILS
EMP ID : 000003
NAME : RAKUL V
PHONE NO. : 9487246880

DATE OF BIRTH : 16/10/2003
DATE OF JOINING : 01/06/2025

ENTER Y TO EXIT: _  
MESSAGE: REGISTERED SUCCESSFUL
```

17. Enter option 4 to exit from the system and terminate the transaction.

```
EMPLOYEE SWIPE SYSTEM
*****
OPTIONS:
1. SWIPE IN.
2. SWIPE OUT.
3. NEW REGISTRATION
4. EXIT.

ENTER THE OPTION: 4

MESSAGE: -----
```

```
SYSTEM ENDED
```

18. If a wrong entry is made, the error message will be displayed. This system doesn't allow duplicate data due to the implementation of KSDS.

```
SWIPE IN
ENTER DETAILS

EMPID:    000002
NAME:     RAKUL V

MESSAGE:   ERROR IN EMP FILE :
```