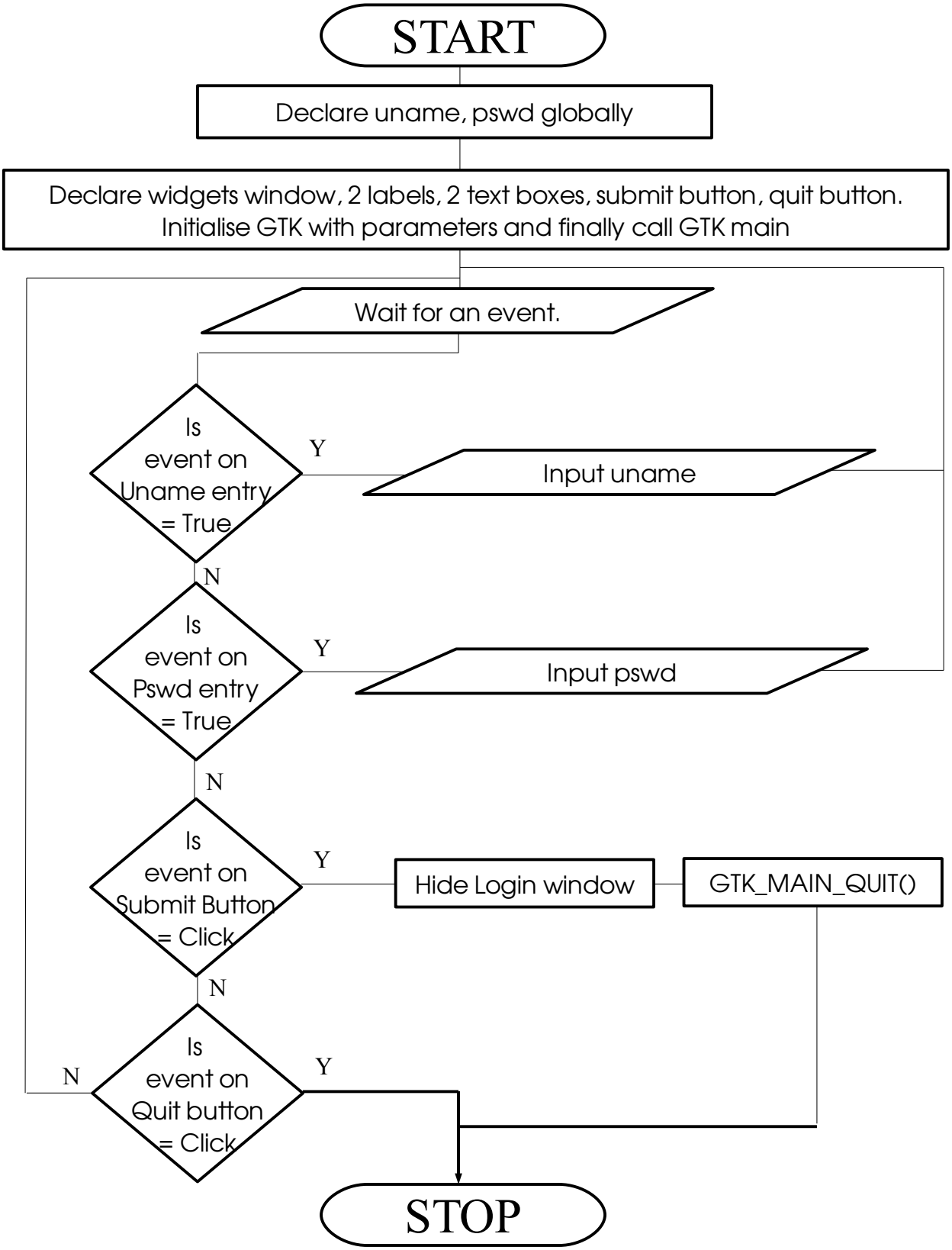
Flowchart for login module

(This module is responsible for starting graphical window for user login)



Algorithm for login window

```
//GLOBAL VARIABLE DECLARATIONS

static char uname[14]="",pswd[6]="";
const char *uname_entry,*pswd_entry;
int login_retcode = 0;

//GLOBAL FUNCTION DECLARATIONS

submit_uname()
{     GET TEXT FROM THE TEXTBOX AND COPY IT TO THE GLOBAL UNAME VARIABLE
}

submit_pswd()
{     GET TEXT FROM THE TEXTBOX AND COPY IT TO THE GLOBAL PSWD VARIABLE
}

submit()
{     CHECK THE VALIDITY OF THE UNAME AND PASS VAR AND QUIT.
}

exiting_login()
{     QUITTING THE LOGIN WINDOW
}

void login_window_content()
{     DECLARE THE WIDGETS
      CREATE WINDOW
      CONNECT THE WIDGETS WITH SIGNALS
      FORMAT THE WIDGETS IN THE WINDOW AND ADD CONTAINERS
      SHOW WINDOW
}

//MAIN FUNCTION STARTS HERE

int login_window(int argc,char *argv[])
{     CALL WINDOW CONTENT AND CALL GTK INIT.
      CALL LOGIN WINDOW CONTENT
      GTK_MAIN();
}
```

Space Complexity

h/w stack

FUNCTIONS	Parameters	Return address	Return type
main	2	1	1
Login_window_content	2	1	1
Exiting_login	2	1	1
Submit	2	1	1
Submit Pswd	2	1	1
Submit Uname	2	1	1

Global variables

Uname	: 14
Pswd	: 6
login_retcode	: 2
uname_entry	: 14
pswd_entry	: 6
labels	: 19
TOTAL :	65

Total Space Complexity:

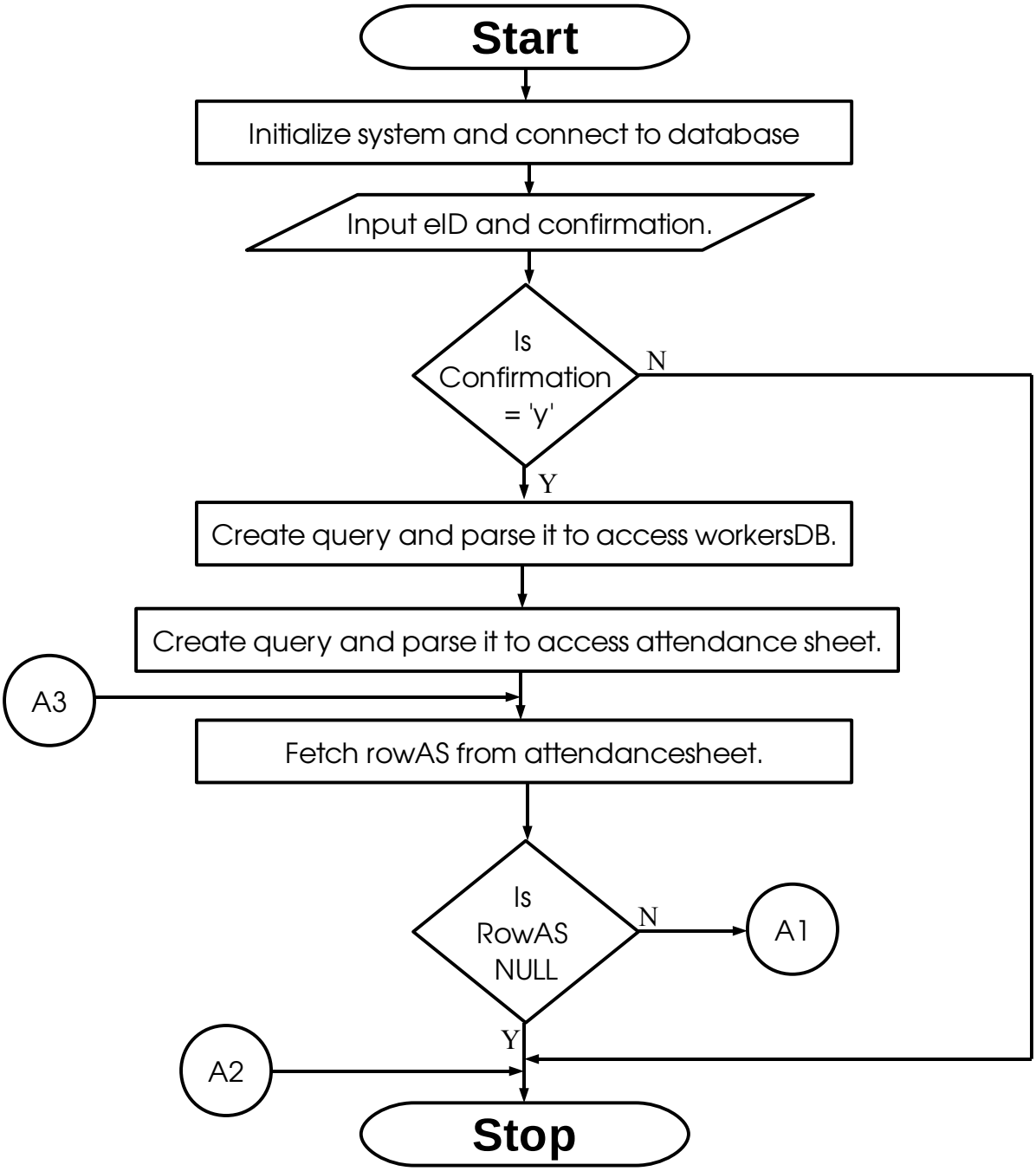
65 + 23 + g

= 88 + g

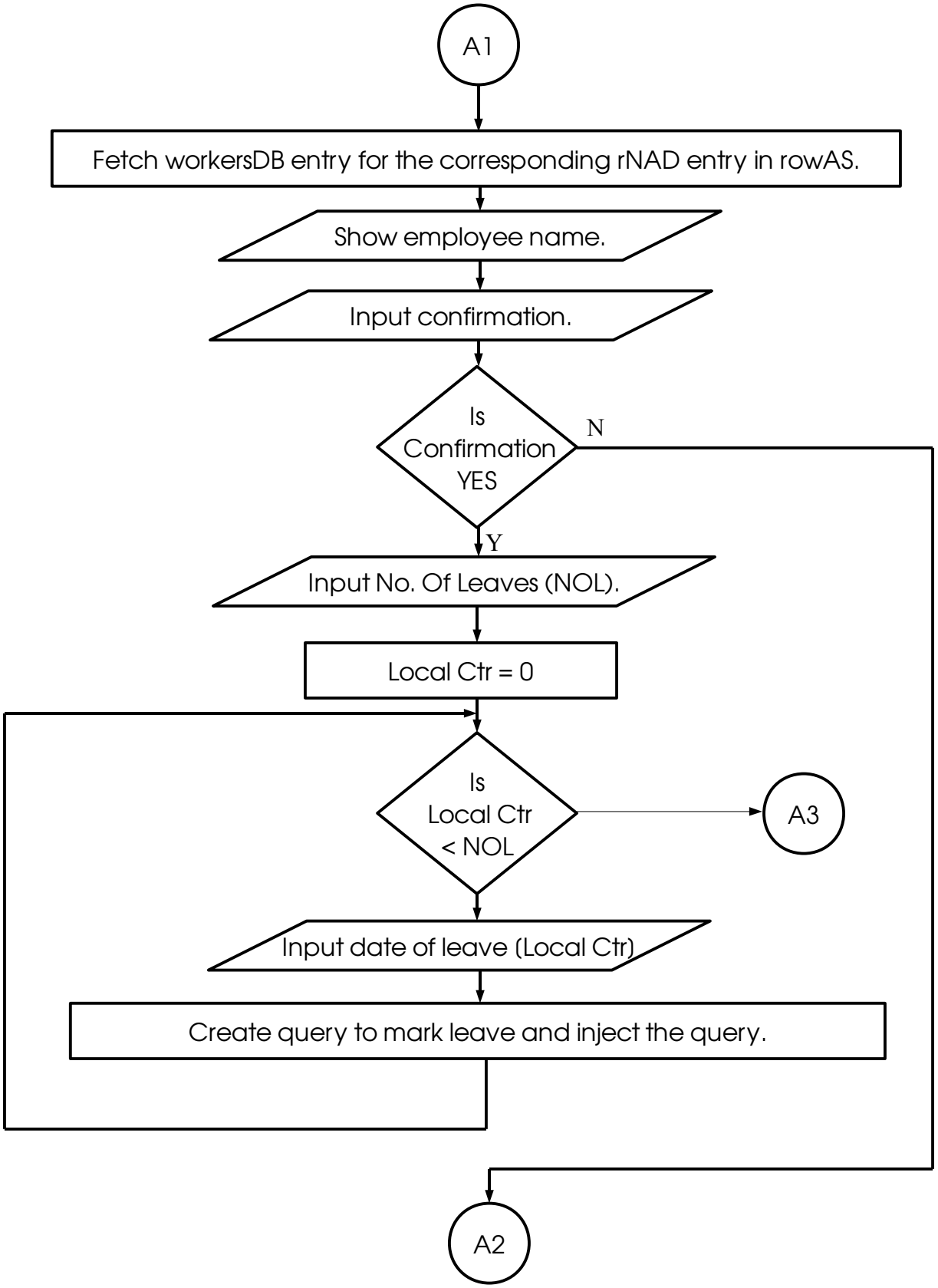
(where g is the GTK constant)

Flowchart for Sanction_leave Module

(This module is responsible for sanctioning the leaves for a particular employee for a given month)



Sanction_leave Module (Continued)



Space Complexity

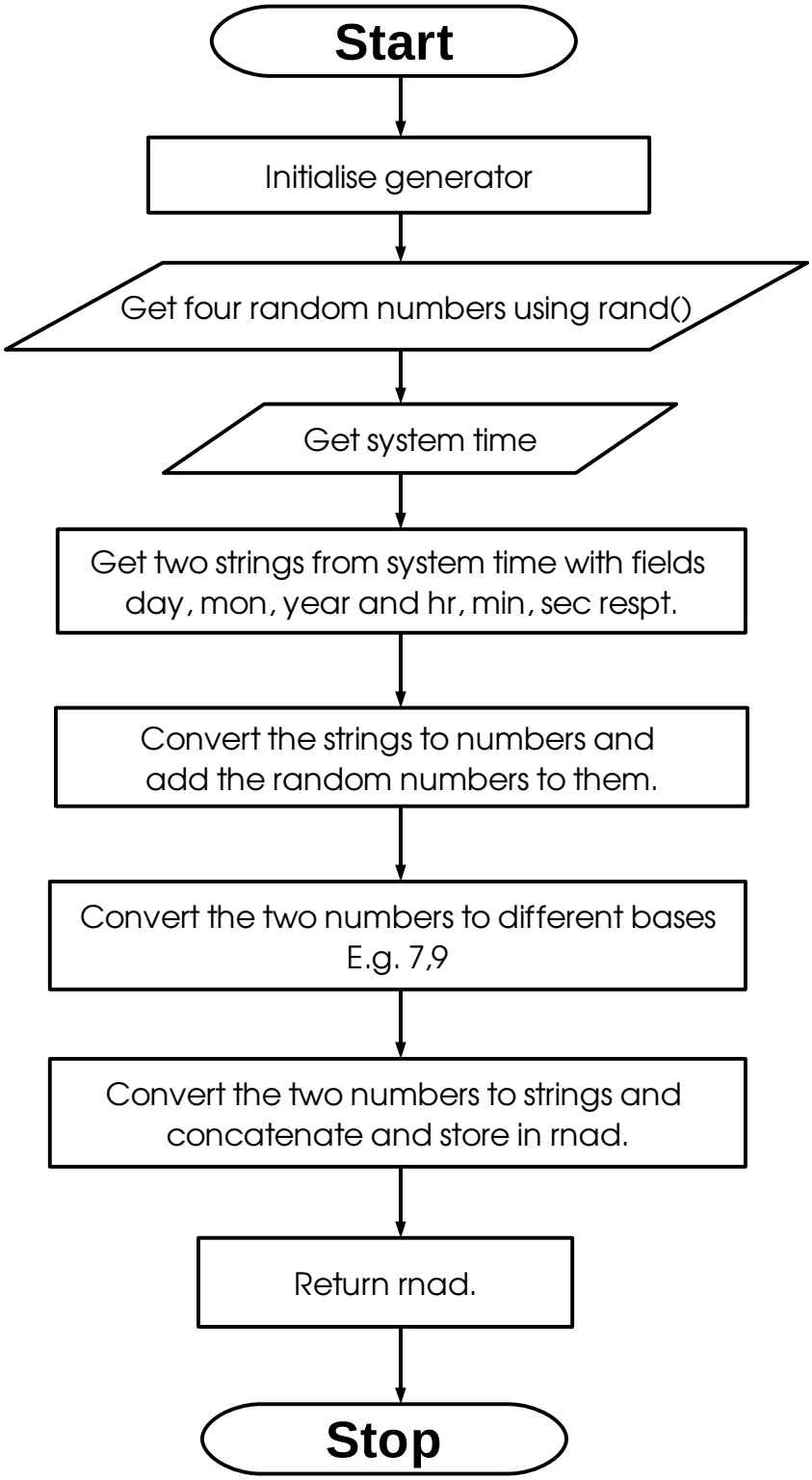
Server	10
User	5
Password	5
Database	13
Choice	1
EmpID	10
QuerySL	100
QuerySL2	100
QuerySLbck	100
retCodeSL	1
rowAS_SLstr	50
Leaves	1
Leavesbck	1
Localctr	1
day_SL	1
free_res_SL	1
day_SLstr	4
SheetSelectSL	60
TOTAL	464

Algorithm for SL

Please leave 1 page space for it I will be mailing you later

Flowchart for rnadgen module

(This module is responsible for generation of the rNAD which is associated with each worker in the workersDB.)



Space Complexity

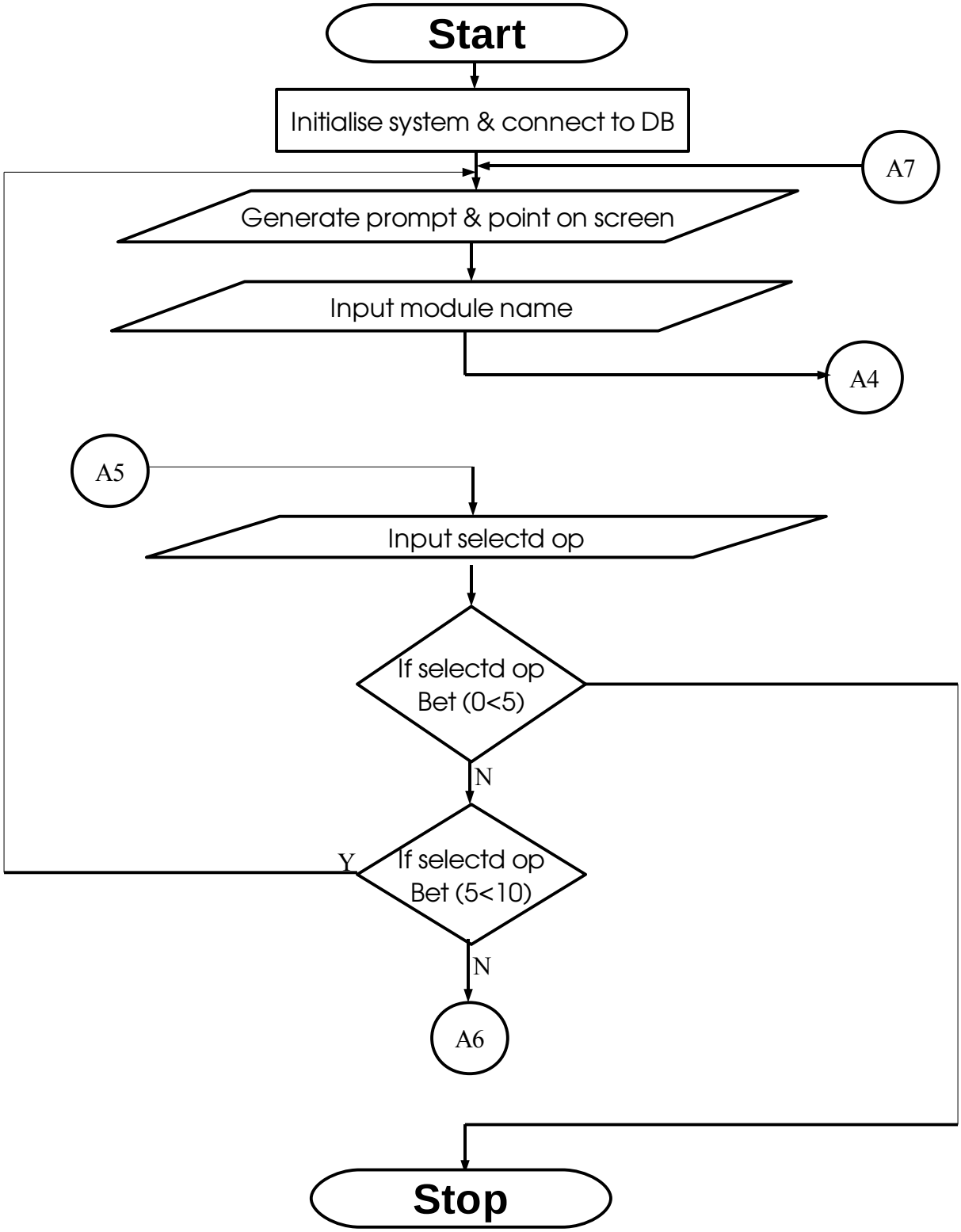
I will mail you later about this

Algorithm for rnadgen

```
GLOBAL rnad
Algo baseConvert()
{     //INPUT
            NUM, BASE

      LOOP TILL (m>0)
            rem=m%base;
            n=n*10+rem;
            m=m/base;
      END LOOP
      m=0;

      LOOP TILL (n>0)
            rem=n%10;
            m=m*10+rem;
            n=n/10;
      END LOOP
            RETURN m
}

Algo rnadGenerator()
{
         GENERATE RANDOM NUMBERS FOR r1,r2,r3,r4
         GET SYSTEM TIME IN t
         FORMAT TIME AND STORE IN arr1,arr2
         CONVERT arr1,arr2 TO NUMBERS AND STORE IN num1,num2
         ADD r1,r2,r3,r4 TO num1,num2
         CALL baseConvert FOR num1,num2
         CONVERT num1,num2 BACK TO STRING AND STORE IN str1,str2
         CONCAT str1,str2 AND STORE IN rnad

}
```

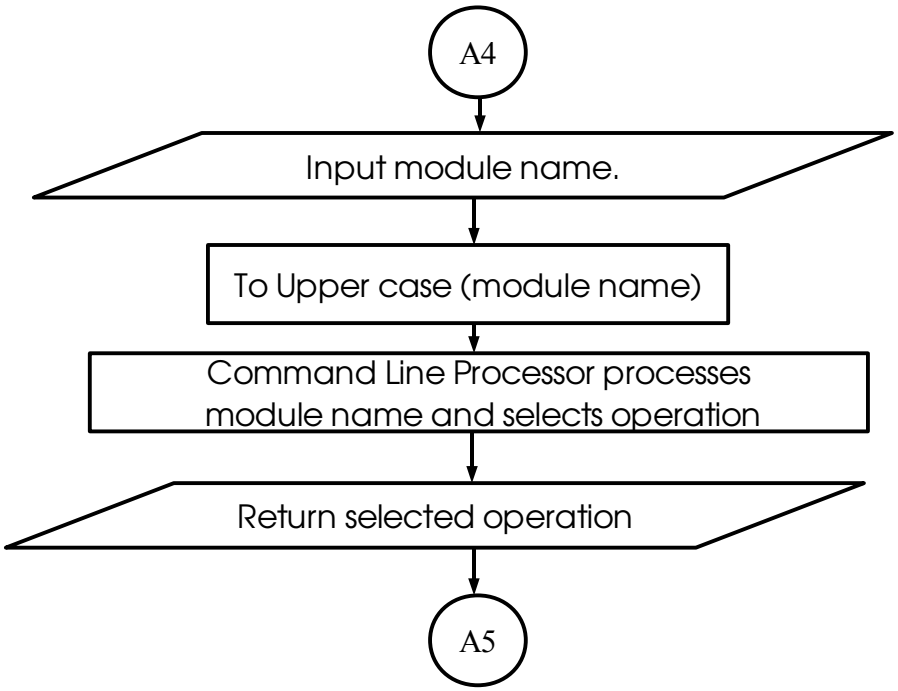
Flowchart for OPCC module

(This module is responsible for controlling different phases and some special features of omniPresence.)



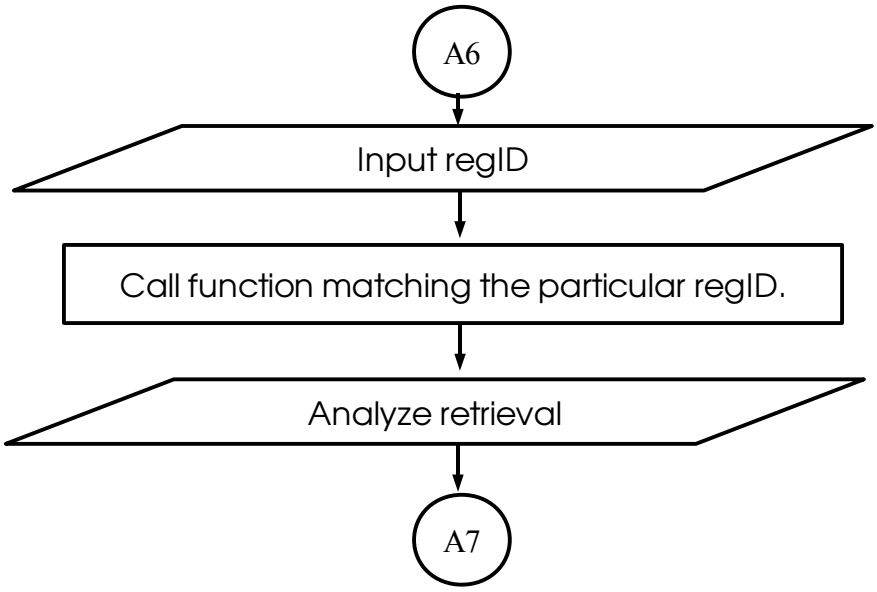
Command line Processor Module

(accepts input from the user and identifies if they are applicable to OPCC or not)



Registry Module

(brings into action those functions which the user wants to process)



Algorithm for omniPresence Command Central (O.P.C.C.)

```
//INCLUDE SYSTEM COMPONENTS (LIKE C HEADERS)
//INCLUDE CUSTOM COMPONENTS (LIKE P1, P2)
//GLOBAL VARIABLE DECLARATION
//INCLUDE OPCC COMPONENTS

ALGO OPCC
{     INITIALISE GRAPHICAL SYSTEM
      GENERATE PROMPT AND PARSE VALUES
      LOOP WHILE shutdownOPCC NOT 1
            GENERATE PROMPT AND TAKE INPUT
            SEND INPUT TO COMMAND LINE PROCESSOR
            ANALYSE OUTPUT RETURNED FROM COMMAND LINE PROCESSOR
                  IF (SELECTED OPERATION)
                        SEND TO REGISTRY MODULE
                  ELSE
                        CONTINUE OR EXIT
                  END IF
      END LOOP
      COLLECT AND SHUTDOWN SYSTEM
}

ALGO commandLineProcessor
{     //INPUT
            COMMANDLINE ASCII
      //OUTPUT
            SELECTED OPERATION

      //WORKING
            CONVERT COMMANDLINE TO UPPERCASE
            PERFORM CHECK OF THE COMMANDLINE FROM THE EXISTING FUNCTION LIBRARY
            IF (FOUND)
                  RETURN THE CORRESPONDING SELECTION CODE
            ELSE
                  RETURN NOT FOUND

}
```