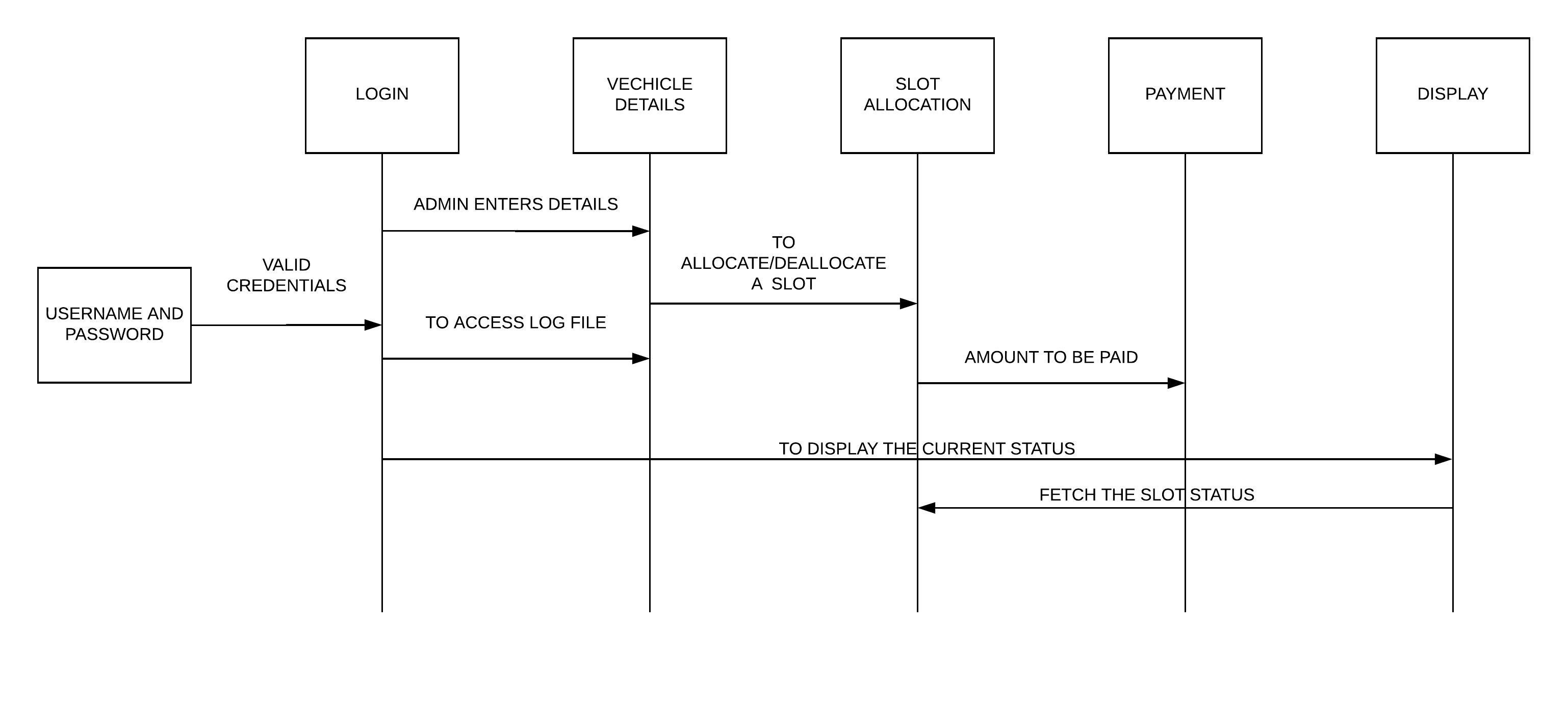
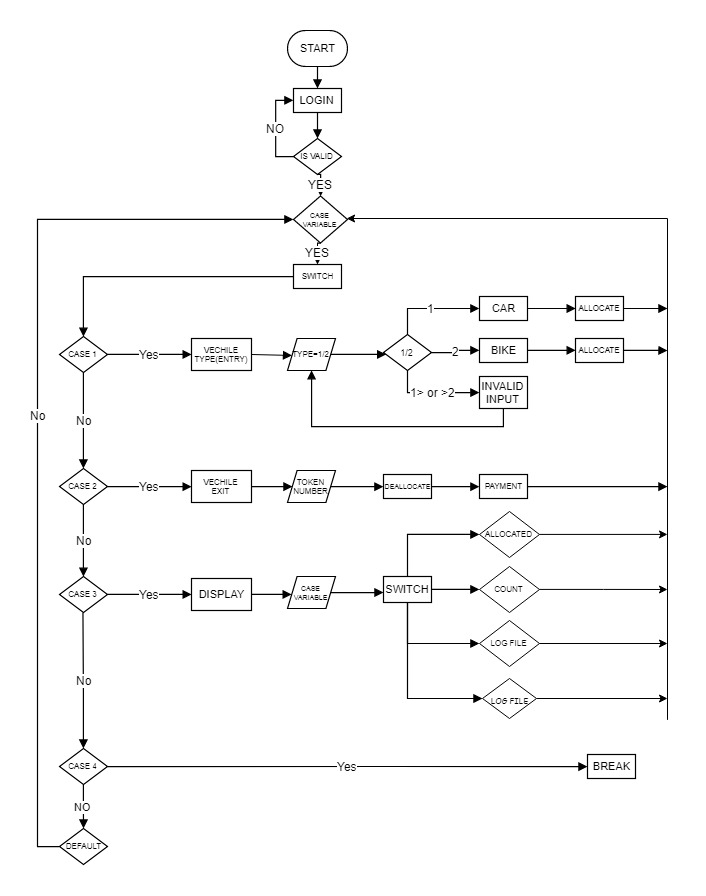
**VEHICLE PARKING MANAGEMENT SYSTEM**

**TEAM : S10**

**CONTROL FLOW DIAGRAM**

****

**FLOW CHART**

****

**MODULE 1 (LOGIN)**

**ALGORITHM**

**Step 1:** Start the program

**Step 2:** Get username and password from security

**Step 3:** Compare the username and password entered by the security with the existing password

**Step 4:** If the username and the password is correct, the security can login

**Step 5:** Otherwise, the security cannot login.

**Step 6:** Stop the program

**PSEUDO CODE**

BEGIN

Print “username”

Input uname

Print “password”

Input pass

If( uname==”admin1 or admin2”) &amp;&amp; if (pass==”pass”)

True: Print “login successfull”

False: Print “Invalid username or password”

END

**MODULE 2(VEHICLE DETAILS)**

**ALGORITHM**

**step 1:** start the program

**step 2:** get the registration number and vehicle type

**step 3:** record its entry time and write all details into log file

**step 4:** allocate token no and allocate parking

**step 5:** stop the program

**PSEUDO CODE**

BEGIN

case input

get regno and vehicle type

time\_t mytime = time(NULL)

initialize tokenno

get start time = ctime(&mytime)

fputs(text,fptr)

END

**MODULE 3 (SLOT ALLOCATION)**

**ALGORITHM:**

**Step 1:** Start

**Step 2:** Get the token number and the type of the vehicle

**Step 3:** If the type is two wheeler then store it in array “two”

**Step 4:** Free spaces are represented as “0”

**Step 5:** Traverse the array and place the vehicle in the free space

**Step 6:** Return the slot number

**Step 7:** Stop

**PSEUDOCODE**

BEGIN

Input tokennumber

Input type

If type==1

Loop i=0-2,j=0-50

a[i][j]=0

Loop i=0-2,j=0-50

If(two[i][j]==0) do two[i][j]=tokennumber

Display slotnumber

If type==2

Loop i=0-2,j=0-50

a[i][j]=0

Loop i=0-2,j=0-50

If(four[i][j]==0) do four[i][j]=tokennumber

Display slotnumber

End

**MODULE 4 (PAYMENT)**

**ALGORITHM**

**step 1:** start the program

**step 2:** get the registration number

**step 3:** check its entry time and note exit time

**step 4:** calculate total durations and calculate charges

**step 5:** display total charges

**step 6:** write all details into log file

**step 7:** stop the program

**PSEUDO CODE**

BEGIN

time\_t mytime = time(NULL)

get tokenno

get exit time = ctime(&mytime)

totalduration=endtime-starttime

parkingcharges=totalduration\*fare

fputs(text,fp)

**MODULE 5 (DISPLAY)**

**ALGORITHM**

**step 1:** start the program

**step 2:** get the switch case variable

**step 3:** based on the case variable execute the functions.

**step 4:** 1 deals with allocation representation,2 deals with total vehicle count in parking,3 arrival log file,4 with departure log file

**step 5:** display the required case.

**step 6:**  stop the program.

**PSEUDO CODE**

BEGIN

Input casevariable

switch casevariable

case 1: Display allocation representation

case 2: Display total vehicle count in parking

case 3: Display arrival log

case 4: Display departure log

END