DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KATHMANDU UNIVERSITY

DHULIKHEL, KAVRE

COMP 102: FINAL PROJECT REPORT

SUBMITTED BY:

NAME: PRASHANT SHRESTHA (49)

SHISHIR TAMRAKAR (53)

SUGANDA ADHIKARI ()

(GROUP - C)

SUBMITTED TO:

NAME: MANISH JOSHI / PRATIT RAJ GIRI

(LECTURER)

DATE: 16th AUGUST, 2021

Table of contents

- Front Page 1

- Table of contents 2

1. Summary 3

2. Introduction 3

    2.1 Objectives 4

3. Case Study 5

    3.1 First Case 5

    3.2 Second Case 5

    3.3 Third Case 6

4. Attributes of Application 7

    4.1 Language Used 7

    4.2 File Handling 7

    4.3 Booking Time 8

5. Libraries Used.....8

6. Algorithm and Flowchart 10

7. Conclusion 23

8. Source Code 24

9. References 35

1. SUMMARY

The report is based about the program for meeting hall booking system. This report presents all the idea, logic and thinking that went behind the completion of the project. Project was break down in different parts and was divided among three of us. C language was used for the development of the program. The program is capable of taking inputs from the users, process it for the validity of booking time, save those data and then show them whenever asked by the user. Portability, clarity is some of the key attributes of the program.

2. INTRODUCTION

Booking system application is an application that allows program organizers (individual , departments) to self book the place on desired time, desired date according to the their respective needs. Advantages of a booking system:

- Hall can be booked at a time that is most convenient for the them
- Whole booking process can be done within minutes
- Checking one's availability to avoid double booking
- Saves time and money

The created application is for booking the meeting hall (Kathmandu University CV Raman Auditorium). Booking is available for all the departments of Kathmandu University. The application books date and time provided by the departments if it is available at the given time and date. Application is capable enough to save the given data, check it with the store date and time for the availability of the hall and then

return the appropriate outcome. Booked time and dates along with other information such as name of department and the purpose for the booking can be viewed with a simple instruction from the user. To make the application more advanced file handling was used. File handling is storing of data in a file using a program. All the data were stored in a other file using file handling. File handling makes it easier to fetch/extract data from a file to work with it in the program.

2.1. OBJECTIVES

To make the application better and flexible, at first structure of the program was discussed. Doing so, it made the process of writing program easier and clear. Further, the main parts/features of the application were easier to distinguish and work more on them to make them user friendly, flexible and efficient. Some of the objectives of the application are as follow:

- To make a hall booking system
- To make the application user friendly
- Effective process to check booking time
- Efficient and proper data management and storage
- More control and options for the user

3. CASE STUDY

3.1 CASE I

In the first case, we will be looking at a **Hotel Room Booking System.** The hotel room booking system is developed to help the visitors/customers to book the room of their desire whenever they want. As manual procedure of booking a room in hotel can be time consuming and frustrating, this system helps the customers to book the room through online medium. As it is an online platform, it can be accessed 24*7. Further payment methods and data management system is hugely benefited from this system. Due to centralized database in this kind of system, it adds an extra edge to readability and access of data.

The system takes the input such as location, time, rating of hotels, number of rooms from the user and then process those data to provide all the information about the hotels and rooms available which fits to the need of customer. Further, customer can manually type the name of the hotel and search for the rooms available at the given date and time. Payment method is also made simple due to this system as customer gets more than option for the method of payment. As all the data are saved in the server, those data can be accessed quickly whenever needed, which would not be same in case of manual booking at the hotel.

3.2 CASE II

The **Bus Sewa Nepal** is an online booking system currently running in Nepal. It is the first of its kind in Nepal. It is an online real time ticket

booking platform which was powered by small heaven travel and tours Pvt. Ltd.

The system is developed to make the travelling easier and comfortable for local as well as foreigners. The system provides option for number of buses travelling through major districts of Nepal such as Kathmandu, Jhapa, Pokhara and so on. After the starting and ending destination is provided by the customer/traveler, the system provides the information about the busses available. After selecting desired bus, the customer can further select the seat of their desired if it is available. Just like any other booking system, payment can be done in multiple ways. When the booking is confirmed, the system saves the information about the customer and then passes the required information to the bus company.

3.3 CASE III

Pathao is an online ride sharing application. Though pathao application was developed to help people to share ride and earn through it, the application has added further platform which is food delivery. All the booking procedure and payment can be done online.

Pathao helps people find a ride to their destination quickly and easily. The application is simple to use and is linked with other applications such as khalti for the payment procedure. Customers can book a ride with just few clicks in the application. Destination, time and location to be picked are few of the details to be filled by the user to book a ride. As the application is linked to other application, payment method is easy and convenient. It made the travelling easier and quicker as people don't have to wait long to find a medium to travel. With just few clicks whole the booking and payment method can be completed. All the information about the travel is stored in the server. Those

information are well managed and stored properly and can be accessed easily by the concern authorities when needed.

4. ATRRIBUTES OF APPLICATION

4.1 LANGUAGE USED

The whole application was created using C programming language. C is a general purpose language that was originally developed by Dennis Ritchie for the Unix operation system. It has an excellent support of high level as well as low level functionality, which makes it suitable for the application. The inherent flexibility and tolerance of this language makes it even more suitable. It is a simple language and provides faster execution. Another feature of C programming is that it can extend itself. A C program contains various functions which are part of a library. We can add our features and functions to the library. We can access and use these functions anytime we want in our program. This feature makes it simple while working with complex programming.

4.2 FILE HANDLING

File handling is used in the application to make the application more flexible and effective. File handling helps to create, update, read, and delete the files stored on the local file system through the program. Using file handling operation such as creation, opening, reading, writing and deleting files can be done easily through the program.

In the application file handling was used to access a file, store data on the file and use the store data for further works such as checking the availability of booking time. File handling was surely the most important part. File handling made storing and accessing the data effortless whenever required.

4.3 BOOKING TIME

Application allows to book any time between 6:00 - 18:00. If the requested time is not booked by any other departments before, the application will ask for final confirmation to book the hall at that particular time. But if the requested time is already booked by other department then the program will notify that the requested time is not available. Further, all the times available for booking can be accessed by checking the booked time which is just one click away.

5. LIBRARIES USED

A library in C is a collection of header files, exposed for use by other programs. The library therefore consists of an interface exposed in a.h file (named the "header"). They are inbuilt functions in C programming.

The libraries used in the application development are:

I. **Standard Input Output (stdio.h)**

Standard Input Output is a header file which has the necessary information to include the input/output related functions in our program e.g. printf, scanf etc. In order to use functions such as printf or scanf in the program, stdio.h header file must be included in the source code. Otherwise, the program doesn't know what is the definition of printf or scanf and it will throw error.

II. **Standard library (stdlib.h)**

The header file stdlib.h stands for standard library. It has the information of memory allocation/freeing functions. The stdlib.h header files defines the macro NULL, which yield a null pointer constant, and represents a pointer value that is guaranteed not to point to a valid address in memory.

III. String (string.h)

string.h is the header in the C standard library for the C programming language which defines on variable type (size_t), one macro (NULL) and various functions (such as strcmp, strcpy) for manipulating arrays of characters.

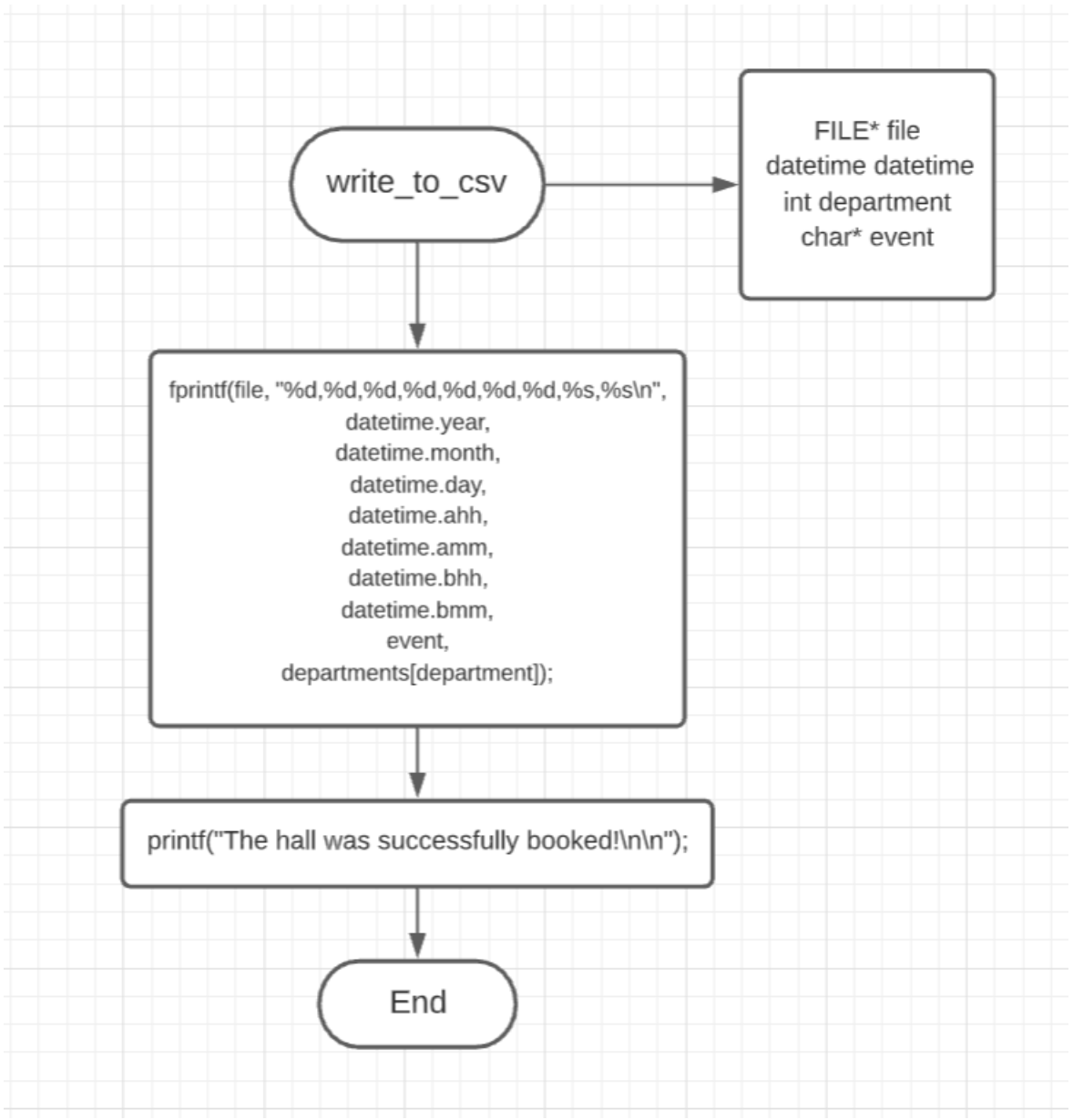
6. ALGORITHM AND FLOWCHART

write_to_csv(FILE *file, datetime datetime, int department, char* event)

Algorithm:

1. Start
2. Write all the booking data passed as parameters to the data.csv file
3. Display “The hall was successfully booked.”
4. Stop

Flowchart:



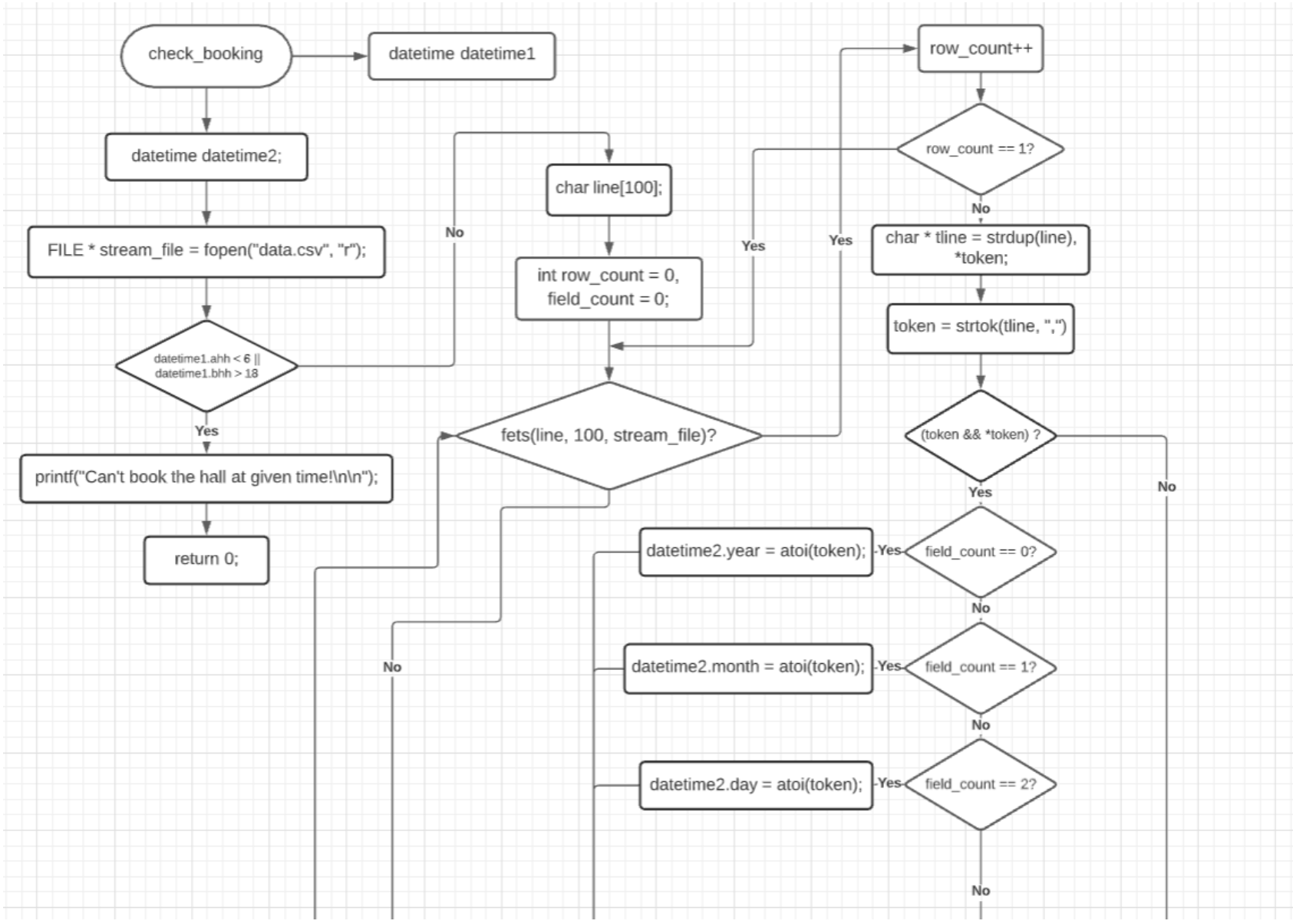
check_booking():

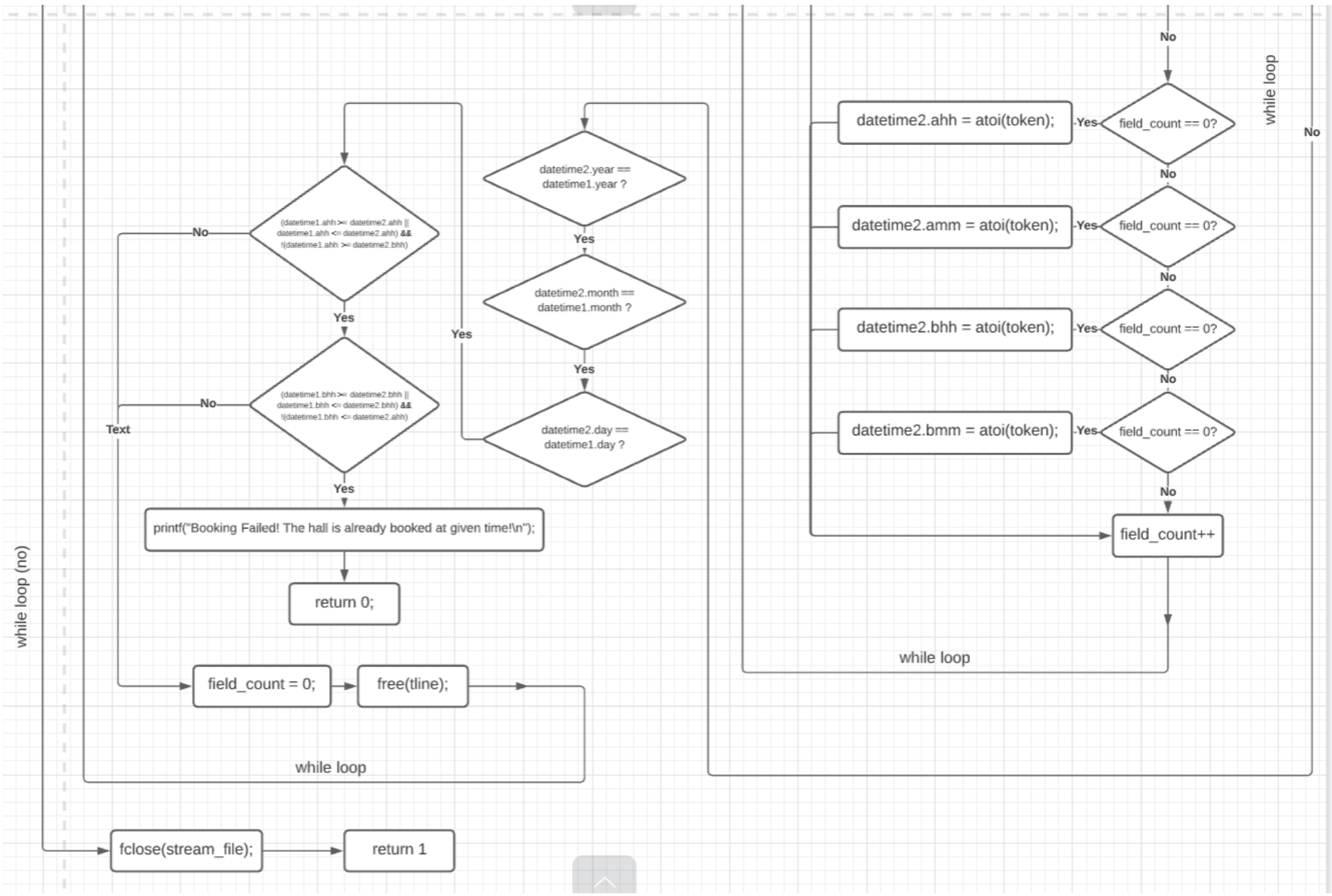
Algorithm:

1. Start
2. Declare datetime struct variable: datetime2
3. FILE * stream_file = fopen(“data.csv”, “r”)
4. If (datetime1.ahh < 6 || datetime1.bhh > 18) is true, goto next step else goto step 8
5. Printf(“Can’t book the hall at given time!”);
6. Return 0
7. End
8. Declare char line[100]
9. Declare int row_count = 0, field_count = 0
10. If(fgets(line, 100, stream_file)) is true goto next step else goto step 48
11. Row_count++
12. If row_count is equal to 1 goto step 10 else goto next step
13. Char * tline = strdup(line), *token;
14. token = strtok(tline, “,”)
15. if(token && *token) is true then goto next step else goto step
16. if field_count == 0 goto next step else goto step 19
17. datetime2.year = atoi(token)
18. goto step 36
19. if field_count == 1 goto next step else goto step 22
20. datetime2.year = atoi(token)
21. goto step 36
22. if field_count == 2 goto next step else goto step 25
23. datetime2.year = atoi(token)
24. goto step 36
25. if field_count == 3 goto next step else goto step 28
26. datetime2.year = atoi(token)
27. goto step 36
28. if field_count == 4 goto next step else goto step 31
29. datetime2.year = atoi(token)

30. goto step
31. if field_count == 5 goto next step else goto step 34
32. datetime2.year = atoi(token)
33. goto step 36
34. if field_count == 6 goto next step else goto step 36
35. datetime2.year = atoi(token)
36. field_count++
37. if(datetime2.year == datetime1.year) goto next step else goto step 45
38. if(datetime2.month == datetime1.year) goto next step else goto step 45
39. if(datetime2.day == datetime1.year) goto next step else goto step 45
40. if((datetime1.ahh >= datetime2.ahh || datetime1.ahh <= datetime2.ahh) && !(datetime1.ahh >= datetime2.bhh)) goto next step else goto step 45
41. if((datetime1.bhh >= datetime2.bhh || datetime1.bhh <= datetime2.bhh) && !(datetime1.bhh <= datetime2.ahh))goto next step else goto step 45
42. printf(“Booking Failed!”)
43. return 0
44. End
45. field_count = 0
46. free(tline)
47. goto step 10
48. fclose(stream_file)
49. return 1
50. End

Flowchart:


check_booking
datetime datetime1
datetime datetime2;
FILE * stream_file = fopen("data.csv", "r");
datetime1.ahh < 6 || datetime1.bhh > 18
Yes
No
printf("Can't book the hall at given time!\n\n");
return 0;
char line[100];
int row_count = 0, field_count = 0;
fets(line, 100, stream_file)?
Yes
No
row_count++
row_count == 1?
Yes
No
char * tline = strdup(line), *token;
token = strtok(tline, ",")
(token && *token) ?
Yes
No
field_count == 0?
datetime2.year = atoi(token);
field_count == 1?
datetime2.month = atoi(token);
field_count == 2?
datetime2.day = atoi(token);


(datetime1.ahh >= datetime2.ahh || datetime1.ahh <= datetime2.ahh) && !(datetime1.ahh >= datetime2.bhh)
No
Yes
(datetime1.bhh >= datetime2.bhh || datetime1.bhh <= datetime2.bhh) && !(datetime1.bhh <= datetime2.ahh)
No
Text
Yes
printf("Booking Failed! The hall is already booked at given time!\n");
return 0;
field_count = 0;
free(tline);
while loop
while loop (no)
datetime2.year == datetime1.year ?
Yes
datetime2.month == datetime1.month ?
Yes
datetime2.day == datetime1.day ?
Yes
No
No
datetime2.ahh = atoi(token);
Yes
field_count == 0?
No
datetime2.amm = atoi(token);
Yes
field_count == 0?
No
datetime2.bhh = atoi(token);
Yes
field_count == 0?
No
datetime2.bmm = atoi(token);
Yes
field_count == 0?
No
field_count++
while loop
while loop
No

fclose(stream_file);
return 1

main():

Algorithm:

1. Start
2. typedef struct{int year,day,month,ahh,amm,bhh,bmm;}
3. char department[15][50]
4. int flag_loop=1,command I,department
5. char event[50]
6. int lno, ctr
7. char ch,fname[10],temp[10]
8. clear screen
9. if flag_loop==1 then
10. output list of commands
11. read command
12. switch(command)
13. case 1
14. case 2
15. case 3
16. case 4
17. default
18. else end

case 1

1. clear screen
2. Display list of departments from the department[15][20]
3. Read a number and store it to department
4. Department = department -1
5. Read year, month and day from the user and store it to datetime.year, datetime.month and datetime.day respectively
6. Read time from which the events starts and store it in datetime.ahh and datetime.amm
7. Read time for when the event ends and store it to datetime.bhh and datetime.bmm

8. Confirm to book the hall with provided data
9. If confirm ==1 then
10. Clear screen
11. File *csv_file = fopen("data.csv","a");
12. If !csv_file is equal to 1 then
13. Print "There was an error reading the file."
14. break
15. Else
16. Write_to_csv(csv_file,determine,department,event)
17. Fclose(csv_file)
18. Break
19. Else break

case 2

1. Clear screen
2. Display "List of Booking"
3. Display headers "SN,Date,From,To,Event,Department with proper indentation
4. File*printf_file=fopen("data.csv","r")
5. Declare Char line[100]
6. Declare int row_count=0,field_count=0
7. If fgets(line,100,print_file) is true
8. Go to next step 10
9. else go to next step 44
10. row_count++
11. if(row_count==1) then
12. go to step 7
13. else
14. declare char *tline = strdup(line)
15. declare char *token
16. display SN(row_count-1)
17. token = strtok(tline,",")

18. if(token&&*token) is true go to next step 20
19. else go to next step 40
20. if(field_count==0) is true goto next step else goto step 22
21. print value of token(year)
22. if(field_count==1) is true goto next step else goto step 24
23. print value of token(month)
24. if(field_count==2) is true goto next step else goto step 26
25. print value of token(day)
26. if(field_count==3) is true goto next step else goto step 28
27. print value of token(from : hour hand)
28. if(field_count==4) is true goto next step else goto step 30
29. print value of token(from : minute hand)
30. if(field_count==5) is true goto next step else goto step 32
31. print value of token(to : hour hand)
32. if(field_count==6) is true goto next step else goto step 34
33. print value of token(to : minute hand)
34. if(field_count==7) is true goto next step else goto step 36
35. print value of token(event)
36. if(field_count==8) is true goto next step else goto step 38
37. print value of token(department)
38. field_count++
39. goto step step 18
40. field_count = 0
41. print new line
42. free(tline)
43. goto step 7
44. fclose(print_file)
45. break

case 3

1. declare ctr=0
2. strcpy(fname, "data.csv")
3. strcpy(temp, "temp.csv")

4. fptr1 = fopen(fname,"r")
5. fptr2=fopen(temp,"w")
6. read the line number user want to remove and store it to lno
7. lno++
8. declare str[100]
9. if(!feof(fptr1)) is true go to step 11 else goto next step
10. else go to step 18
11. if (fgets(str,100,fptr1)) return true goto next step else goto step
12. if!(feof(fptr1)) is true go to next step 9
13. ctr++
14. if (ctr != lno) go to next step
15. else go to step 9
16. fprintf(fptr2,"%s",str)
17. go to step 9
18. fclose(fptrr1)
19. fclose(fptrr2)
20. remove(fname)
21. rename(temp,fname)
22. system clear
23. display the delelted line number(lno-1)
24. break

case 4

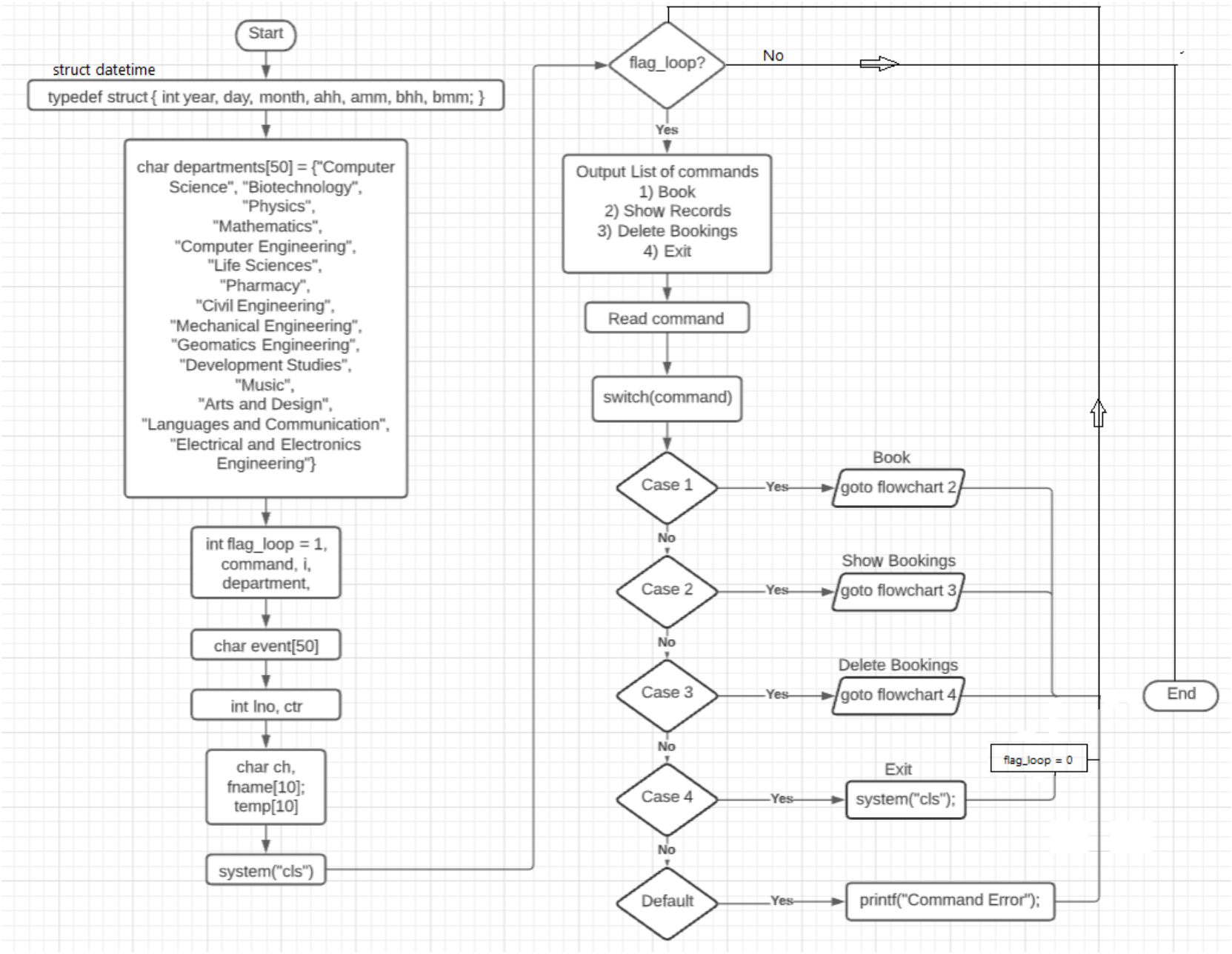
1. clear screen
2. flag_loop = 0
3. break

default

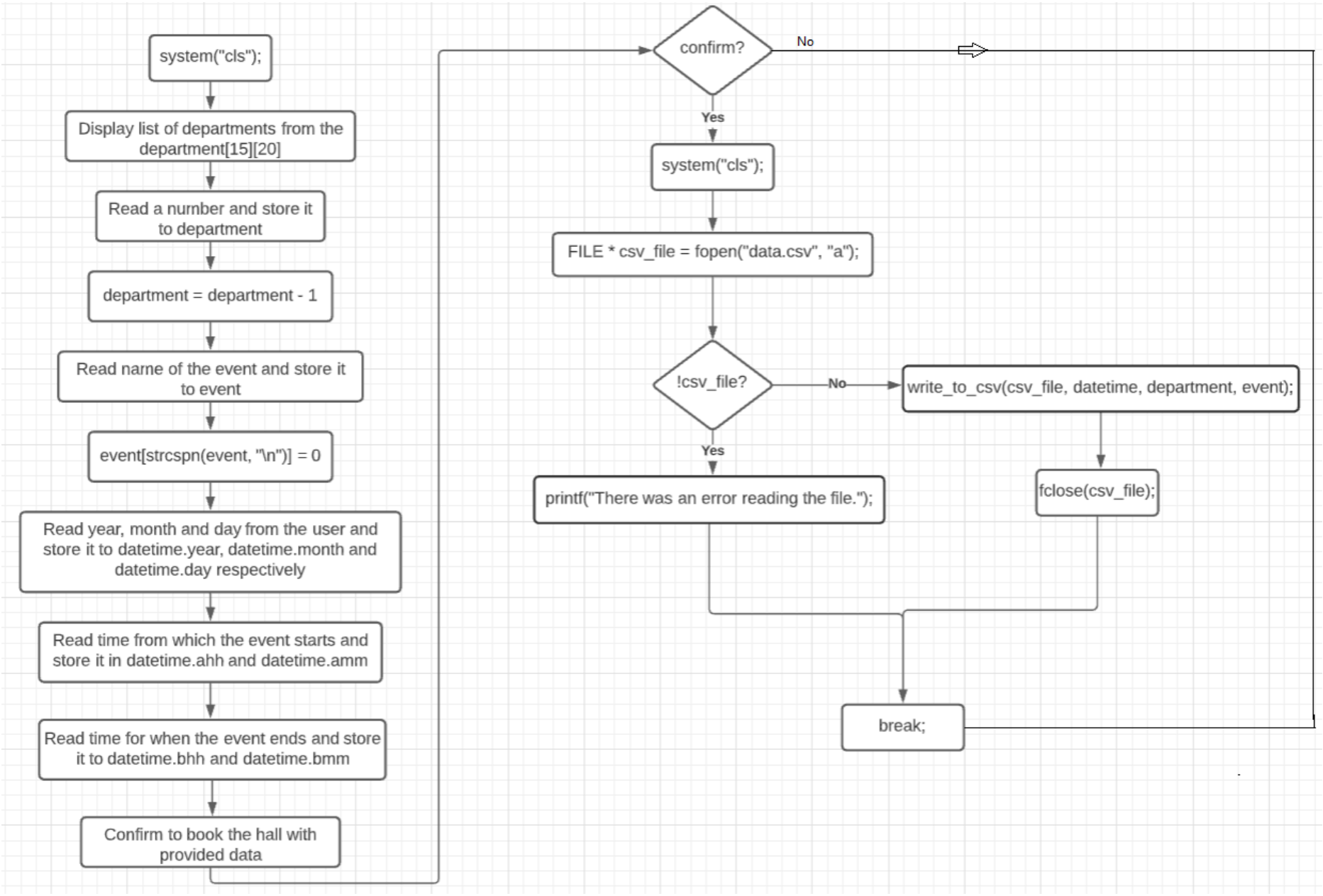
1. print "Command Error"
2. break

Flowchart:

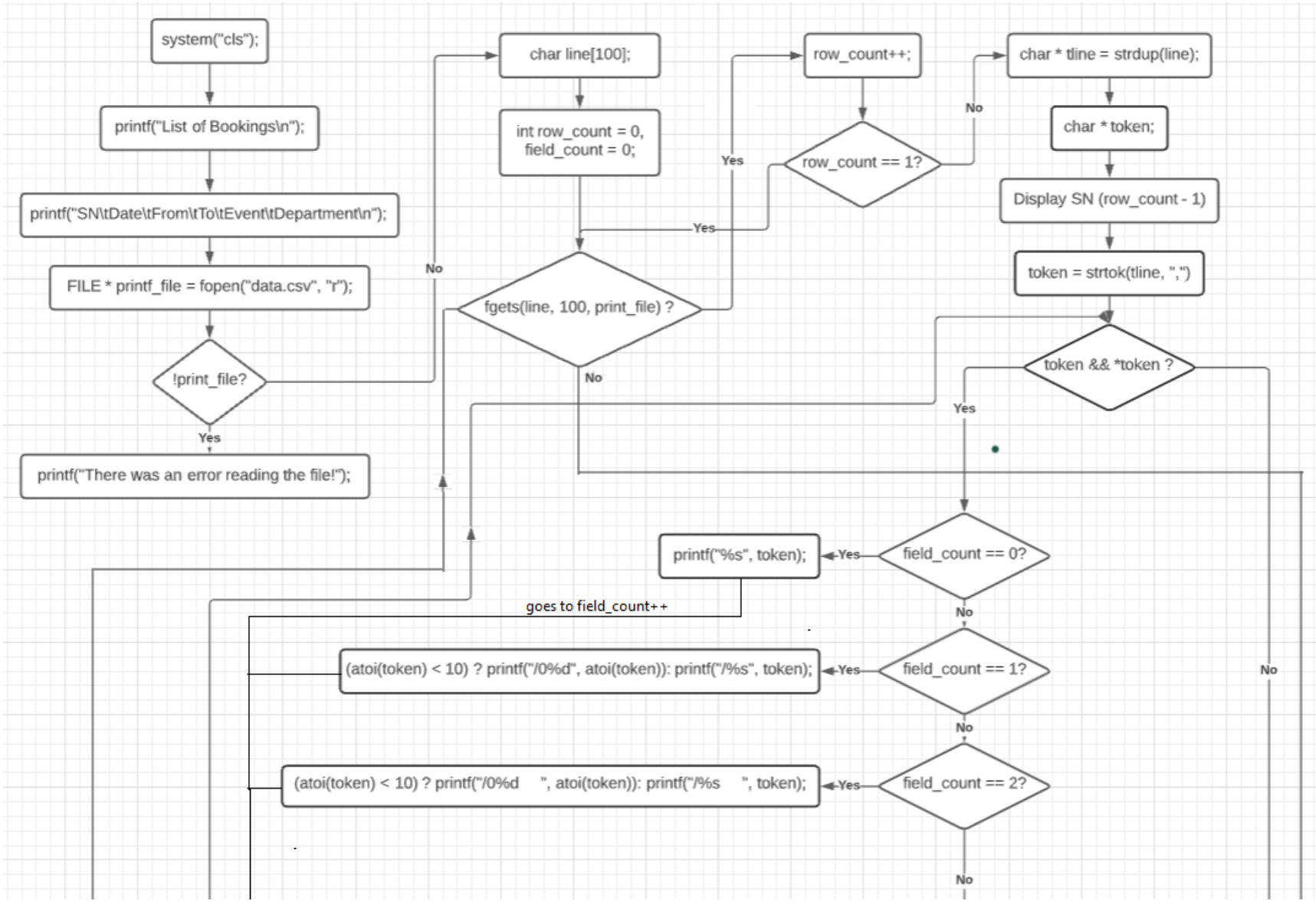
(Flowchart 1)



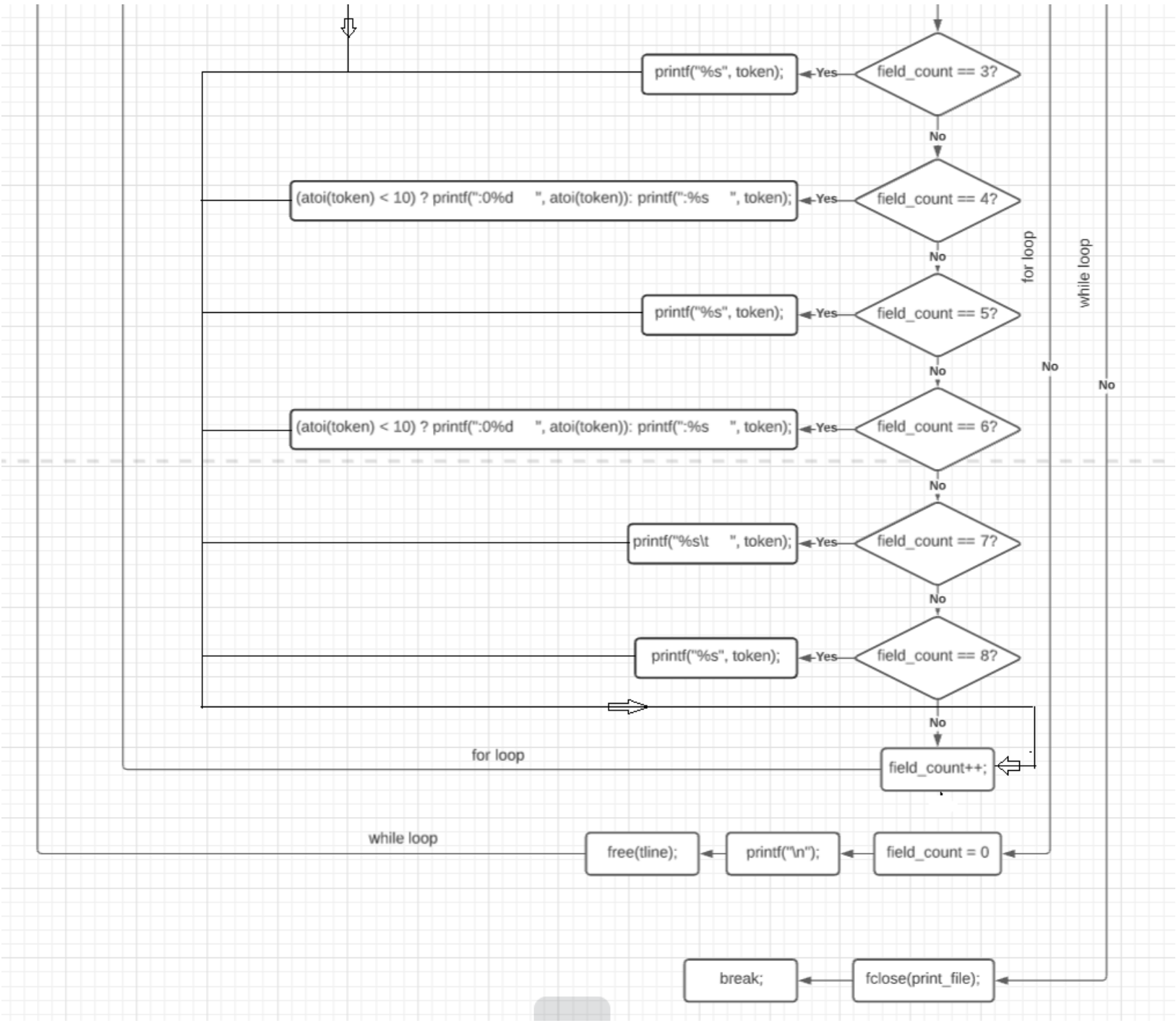
(Flowchart 2)


system("cls");
Display list of departments from the department[15][20]
Read a number and store it to department
department = department - 1
Read name of the event and store it to event
event[strcspn(event, "\n")] = 0
Read year, month and day from the user and store it to datetime.year, datetime.month and datetime.day respectively
Read time from which the event starts and store it in datetime.ahh and datetime.amm
Read time for when the event ends and store it to datetime.bhh and datetime.bmm
Confirm to book the hall with provided data
confirm?
No
Yes
system("cls");
FILE * csv_file = fopen("data.csv", "a");
!csv_file?
No
Yes
write_to_csv(csv_file, datetime, department, event);
fclose(csv_file);
printf("There was an error reading the file.");
break;

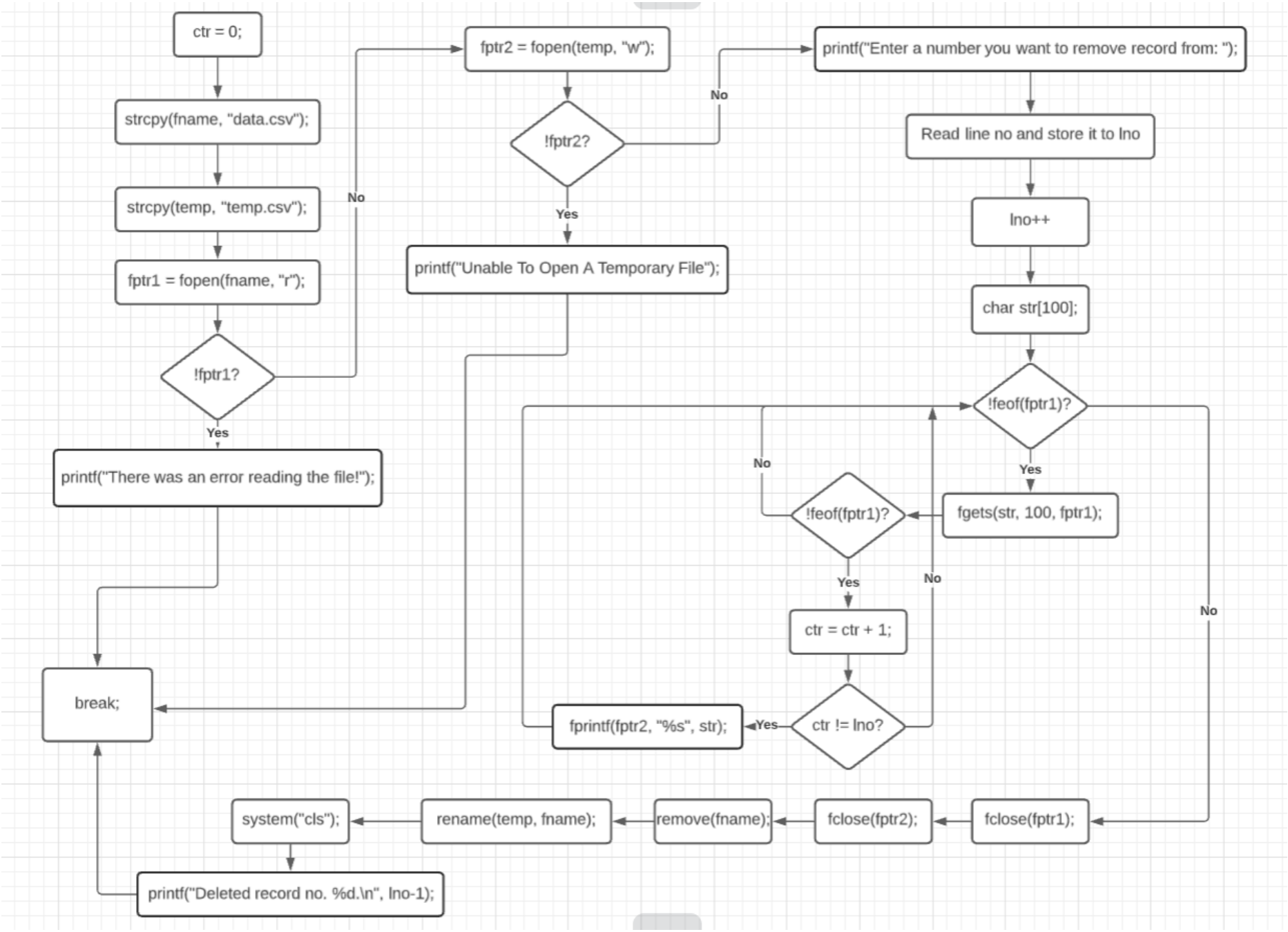
(Flowchart 3.1)



(Flowchart 3.2)



(Flowchart 4)


ctr = 0;
strcpy(fname, "data.csv");
strcpy(temp, "temp.csv");
fptr1 = fopen(fname, "r");
!fptr1?
No
Yes
printf("There was an error reading the file!");
break;
fptr2 = fopen(temp, "w");
!fptr2?
No
Yes
printf("Unable To Open A Temporary File");
printf("Enter a number you want to remove record from: ");
Read line no and store it to lno
lno++
char str[100];
!feof(fptr1)?
Yes
No
fgets(str, 100, fptr1);
!feof(fptr1)?
No
Yes
ctr = ctr + 1;
ctr != lno?
Yes
No
fprintf(fptr2, "%s", str);
No
fclose(fptr1);
fclose(fptr2);
remove(fname);
rename(temp, fname);
system("cls");
printf("Deleted record no. %d.\n", lno-1);

7. CONCLUSION

The application was developed to manage the details of booking dates and times along with other information related to booking. The application is not built openly and the user can access the information about other booking. This was done to give user more flexibility and information to make his decision. The main purpose of this application was to reduce manual work and save time and money. The application being the first one has some small issues where there is space for improvement.

The part of the application containing the booking time checking process can is one part which can be further improved and can be made more effective for the user. It can be tweak a bit by letting the user see the available time for the given day. Currently, user can only see the booked date and time manually so this is surely one of the portion which can be further improved to make the application more effective.

Currently, the option for specific change in the booking is not available. So in the near future, the option to change a particular data/information about the booking can be made available for the user. Further, if the booked time can be connected with goggle calendar or if a calendar is provided to user in the application, the user would be able to see the dates available for booking and make his decision on the available information. This is also one of few enhancements which can be made in the future to make the application effective, user friendly and efficient.

8. CODE

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>


typedef struct{

    int year;

    int month;

    int day;

    int ahh;

    int amm;

    int bhh;

    int bmm;

} datetime;


char departments[15][50] = {{"Computer Science"},

                    {"Biotechnology"},

                    {"Physics"},

                    {"Mathematics"},

                    {"Computer Engineering"},

                    {"Life Sciences"},

                    {"Pharmacy"},

                    {"Civil Engineering"},

                    {"Mechanical Engineering"},

                    {"Geomatics Engineering"},
```

```
        {"Development Studies"},
        {"Music"},
        {"Arts and Design"},
        {"Languages and Mass Communication"},
        {"Electrical and Electronics Engineering"}};

void write_to_csv(FILE* file, datetime datetime, int department, char* event) {
    fprintf(file, "%d,%d,%d,%d,%d,%d,%d,%s,%s\n", datetime.year, datetime.month, datetime.day, datetime.ahh, datetime.amm, datetime.bhh, datetime.bmm, event, departments[department]);
    printf("The hall was successfully booked!\n\n");
}

int check_booking(datetime datetime1) {
    datetime datetime2;
    FILE * stream_file = fopen("data.csv", "r");
    if(!stream_file) {
        printf("There was an error reading the file.\n");
        return 0;
    }
    if(datetime1.ahh < 6 || datetime1.bhh > 18) {
        printf("Can't book the hall at given time!\n\n");
        return 0;
    }
    char line[100];
```

```
int row_count = 0;
int field_count = 0;
while(fgets(line, 100, stream_file)) {
   row_count++;
   if(row_count == 1) {
      continue;
   }
   char * tline = strdup(line);
      char * token;
      for((token = strtok(tline, ",")); token && *token; token = strtok(NULL,
",")) {
         if(field_count == 0)
            datetime2.year = atoi(token);
         if(field_count == 1)
            datetime2.month = atoi(token);
         if(field_count == 2)
            datetime2.day = atoi(token);
         if(field_count == 3)
            datetime2.ahh = atoi(token);
      if(field_count == 4)
         datetime2.amm = atoi(token);
      if(field_count == 5)
         datetime2.bhh = atoi(token);
      if(field_count == 6)
         datetime2.bmm = atoi(token);
```

```
            field_count++;
        }

        if(datetime2.year == datetime1.year) {
            if(datetime2.month == datetime1.month) {
                if(datetime2.day == datetime1.day) {
                    if(((datetime1.ahh >= datetime2.ahh || datetime1.ahh <=
datetime2.ahh) && !(datetime1.ahh >= datetime2.bhh)) &&
                        ((datetime1.bhh >= datetime2.bhh || datetime1.bhh <=
datetime2.bhh) && !(datetime1.bhh <= datetime2.ahh))) {
                        printf("Booking Failed! The hall is already booked at given
time!\n");
                        return 0;
                    }
                }
            }
        }

        field_count = 0;
            free(tline);
        }
        fclose(stream_file);

        return 1;
    }
```

```
int main() {
    int flag_loop = 1;
    datetime datetime;
    int command, i, department;
    char event[50];

    //for case 3
    int lno, ctr;
    char ch;
    FILE *fptr1, *fptr2;
    char fname[10];
    char temp[10];

    printf("\e[1;1H\e[2J");
    printf("\n----------------------Kathmandu University CV Raman Auditorium-----------------\n\n");

    while(flag_loop) {
    printf("---List of commands---\n");
    printf("1) Book\n");
    printf("2) Show Records\n");
    printf("3) Delete Bookings\n");
    printf("4) Exit\n\n");

    printf("Command: ");
    scanf("%d", &command);
```

```
getchar();
printf("\n");

if(command > 4) {
    printf("Command Error!");
} else {
    switch (command)
    {
        case 1:
            printf("\e[1;1H\e[2J");       //system("cls"); | clears the screen
            printf("\n----------------------Kathmandu University CV Raman Auditorium----------------------\n\n");
            printf("----List Of Departments----\n");
            for(i = 0; i < 15; i++) {
               printf("%d) %s\n", i+1, departments[i]);
            }
            printf("-------------------------\n\n");
            printf("Enter a number corresponding to your department: ");
            scanf("%d", &department);
            getchar();
            department--;

            printf("Enter the event name: ");
            fgets(event, 50, stdin);
            event[strcspn(event, "\n")] = 0;       //removes line terminator from the event string
```

```
            printf("Enter a date(DD/MM/YYYY): ");
            scanf("%d/%d/%d", &datetime.day, &datetime.month,
&datetime.year);
            printf("From: ");
            scanf("%d:%d", &datetime.ahh, &datetime.amm);
            printf("To: ");
            scanf("%d:%d", &datetime.bhh, &datetime.bmm);
            getchar();


            char sure;
            printf("Are you sure you want to book the hall on %d/%d/%d from
%d:%d to %d:%d?\n(Y/N): ", datetime.day, datetime.month, datetime.year,
datetime.ahh, datetime.amm, datetime.bhh, datetime.bmm);
            scanf("%c", &sure);
            if(sure == 'Y') {
               printf("\e[1;1H\e[2J");
               if(check_booking(datetime)) {
                  FILE * csv_file = fopen("data.csv", "a");
                  if(!csv_file) {
                     printf("There was an error reading the file.");
                     break;
                  }
                  write_to_csv(csv_file, datetime, department, event);
                  fclose(csv_file);
               }
```

```
            }
            break;

        case 2:
            printf("\e[1;1H\e[2J");
            printf("\n----------Kathmandu University CV Raman Auditorium----------\n\n");
            printf("--------------------------------List Of Bookings--------------------------------\n\n");
            printf("SN      Date\t      From      To\t  Event\t\t\t Department\n\n");
            FILE * print_file = fopen("data.csv", "r");
            if(!print_file) {
                printf("There was an error reading the file.");
                break;
            }
            char line[100];
            int row_count = 0;
            int field_count = 0;
            while(fgets(line, 100, print_file)) {
               row_count++;
               if(row_count == 1) {
                  continue;
               }
               char * tline = strdup(line);
               char * token;
```

```
            printf("%d     ", row_count-1);
            for((token = strtok(tline, ",")); token && *token; token = strtok(NULL, ",")) {
               if(field_count == 0)
                  printf("%s", token);    //year
               if(field_count == 1)
                  (atoi(token) < 10) ? printf("/0%d", atoi(token)): printf("/%s", token);   //month
               if(field_count == 2)
                  (atoi(token) < 10) ? printf("/0%d          ", atoi(token)): printf("/%s    ", token); //day
               if(field_count == 3)
                  printf("%s", token);    //ahh
               if(field_count == 4)
                  (atoi(token) < 10) ? printf(":0%d     ", atoi(token)): printf(":%s ", token); //amm
               if(field_count == 5)
                  printf("%s", token);    //bhh
               if(field_count == 6)
                  (atoi(token) < 10) ? printf(":0%d     ", atoi(token)): printf(":%s ", token); //bmm
               if(field_count == 7)
                  printf("%s\t     ", token);  //event
               if(field_count == 8)
                  printf("%s", token);    //department

               field_count++;
```

```
            }
            field_count = 0;
            printf("\n");
            free(tline);
        }
        printf("-----------------------------------------------------------------------
---\n\n");
        fclose(print_file);
        break;

    case 3:
        ctr = 0;
        strcpy(fname, "data.csv");
        strcpy(temp, "temp.csv");

        fptr1 = fopen(fname, "r");
        if(!fptr1) {
            printf("File Not Found!");
            break;
        }
        fptr2 = fopen(temp, "w");
        if(!fptr2) {
            printf("Unable To Open A Temporary File");
            fclose(fptr1);
            break;
        }
```

```
printf("Enter a number you want to remove record from: ");
scanf("%d", &lno);
lno++;

char str[100];
while(!feof(fptr1)) {
   fgets(str, 100, fptr1);

   if(!feof(fptr1)) {        //checks if eof(end of file) is reached
      ctr++;
      if(ctr != lno) {
         fprintf(fptr2, "%s", str);
      }
   }
}

fclose(fptr1);
fclose(fptr2);
remove(fname);
rename(temp, fname);

printf("\e[1;1H\e[2J");
printf("Deleted record no. %d.\n", lno-1);
break;
```

```
        case 4:
            printf("\e[1;1H\e[2J");
            flag_loop = 0;          //exits out of the while loop (program termination)
            break;
        }
    }
}

    return 0;
}
```

Output

```
----------------------Kathmandu University CV Raman Auditorium----------------------

---List of commands---
1) Book
2) Show Records
3) Delete Bookings
4) Exit

Command: 1

----------------------Kathmandu University CV Raman Auditorium----------------------

----List Of Departments----
1) Computer Science
2) Biotechnology
3) Physics
4) Mathematics
5) Computer Engineering
6) Life Sciences
7) Pharmacy
8) Civil Engineering
9) Mechanical Engineering
10) Geomatics Engineering
11) Development Studies
12) Music
13) Arts and Design
14) Languages and Mass Communication
15) Electrical and Electronics Engineering
--------------------------

Enter a number corresponding to your department: 9
Enter the event name: Project Discussion
Enter a date(DD/MM/YYYY): 09/12/2021
From: 08:00
To: 11:00
Are you sure you want to book the hall on 9/12/2021 from 8:0 to 11:0?
(Y/N): Y


The hall was successfully booked!

---List of commands---
1) Book
2) Show Records
3) Delete Bookings
4) Exit

Command: 2
```

```
----------------------Kathmandu University CV Raman Auditorium----------------------

----------------------------------List Of Bookings----------------------------------

SN   Date         From    To      Event                    Department

1    2021/08/04   12:00   14:00   Orientation              Biotechnology

2    2021/09/04   09:00   10:00   Orientation Programme     Computer Engineering

3    2021/12/09   08:00   11:00   Project Discussion        Mechanical Engineering

------------------------------------------------------------------------------------

---List of commands---
1) Book
2) Show Records
3) Delete Bookings
4) Exit

Command: 3


Enter a number you want to remove record from: 2


 Deleted record no. 2.
 ---List of commands---
 1) Book
 2) Show Records
 3) Delete Bookings
 4) Exit

 Command: 2

 ----------------------Kathmandu University CV Raman Auditorium----------------------

 ----------------------------------List Of Bookings----------------------------------

 SN   Date         From    To      Event                    Department

 1    2021/08/04   12:00   14:00   Orientation              Biotechnology

 2    2021/12/09   08:00   11:00   Project Discussion        Mechanical Engineering

 ------------------------------------------------------------------------------------

 ---List of commands---
 1) Book
 2) Show Records
 3) Delete Bookings
 4) Exit

 Command:
```

9. REFERENCES

https://www.geeksforgeeks.org

https://stackoverflow.com