# SOFTWARE SYSTEM FOR ENGINEERING JOINT SEAT ALLOCATION

UCS2201 – Fundamentals and Practice of Software Development

A PROJECT REPORT

Submitted By

Sweety Y 3122 22 5001 146

Varsha G 3122 22 5001 152

Vishnu Praba A J 3122 22 5001 164

Yuvapriya N 3122 22 5001 167

Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

Kalavakkam – 603110

July 2023

**Sri Sivasubramaniya Nadar College of Engineering**

**(An Autonomous Institution, Affiliated to Anna University)**

# BONAFIDE CERTIFICATE

Certified that this project report titled "ENGINEERING JOINT SEAT ALLOCATION" Is the bonafide work of "Sweety Y (3122 22 5001 146) Varsha G (3122 22 5001 152) Vishnu Praba A J (3122 22 5001 164) Yuvapriya N (3122 22 5001 167)" who carried out the project work in the UCS2201 – Fundamentals and Practice of Software Development during the academic year 2022-23.

Internal Examiner External Examiner

Date:

# TABLE OF CONTENTS

## PROBLEM STATEMENT

Develop a software system for the engineering counselling and admission process for two sets of institutes (for example, say IITs and NITs) each having a set of different branches, each branch with a certain number of seats available. Number of candidates can be assumed as 5 times the total number of seats available. Each candidate can provide a list of preferences where each preference is a 2-tuple, (institute, branch). Admission to each set of institutes is based on its own qualifying exam (for example, JEE-Advanced and JEE-Main). Each candidate will have a specific rank in one or both merit lists.

## EXTENDED EXPLORATION OF PROBLEM STATEMENT

***Reference: Business Rules of JOSAA from Google.***

In the case of a single merit list, candidates are arranged in a sequential order based on their ranks. Seat allotment is then performed by going through the list in order and allowing each candidate to select from the available seats.

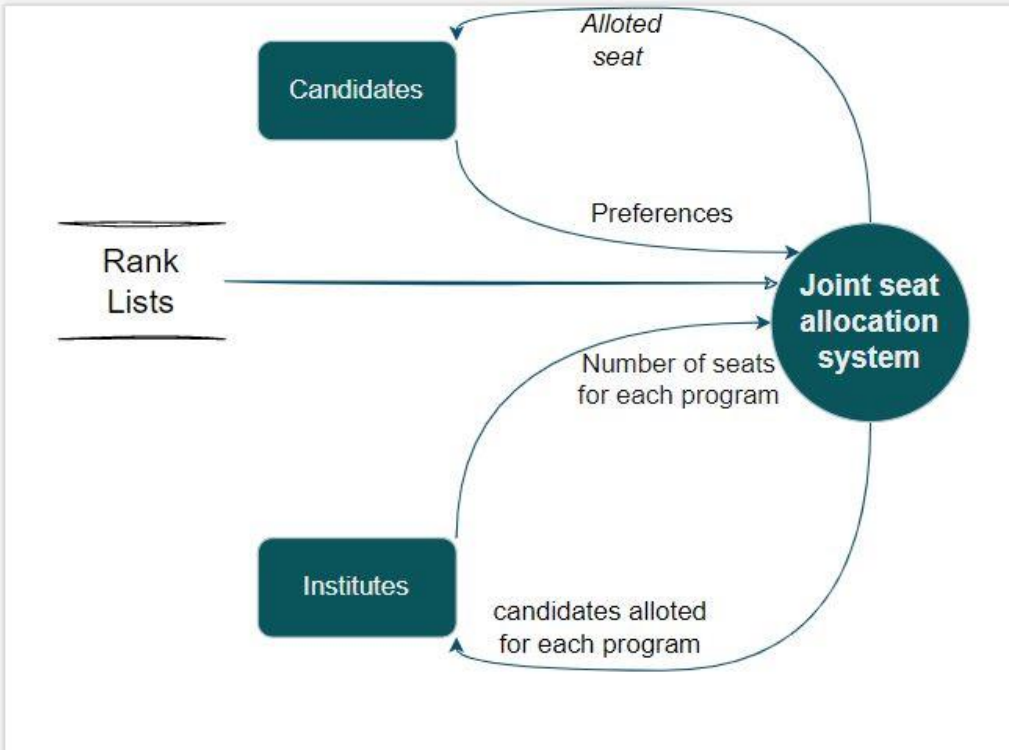
However, the situation becomes more complex when dealing with multiple merit lists, such as those for IITs, NITs, and other institutions that require an additional examination for admission. In this scenario, candidates need to create a common choice list across these multiple merit lists.

The challenge arises when there are many candidates, possibly in the hundreds of thousands, who must prioritize and select their preferred options from these different merit lists.

The system needs to be carefully designed and implemented to handle the complexity of multiple merit lists effectively. It requires robust algorithms, efficient data structures, and comprehensive databases to manage the candidates' choices and ensure that each candidate is allotted a suitable seat according to their preferences and ranks across all the participating institutions.

The Candidate-Proposing Deferred Acceptance (DA) algorithm is a mechanism used in many allocation problems, including school choice and matching markets. In the DA algorithm, candidates propose to their preferred options, and institutions provisionally accept the best candidates, iteratively improving the matches until stable allocations are achieved.

# DATA FLOW DIAGRAMS

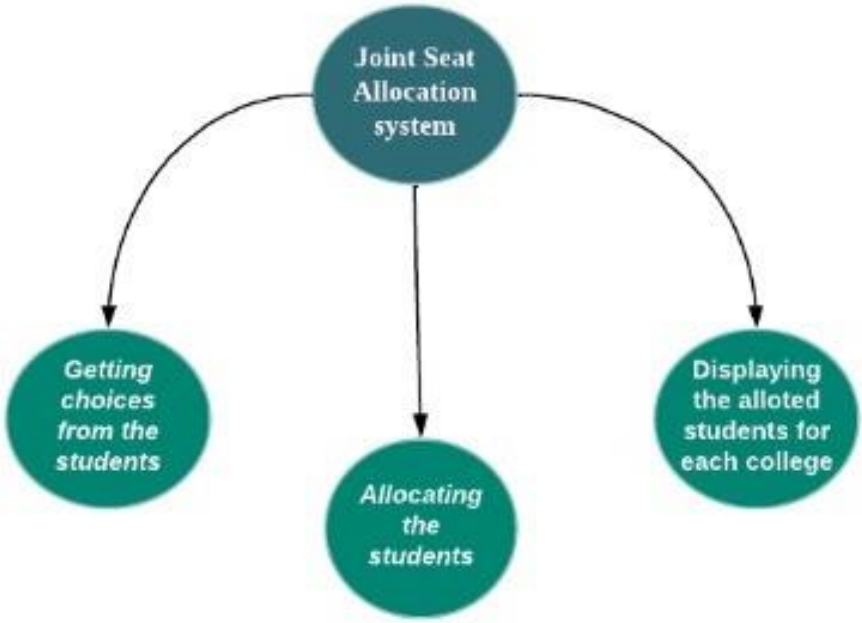



## Level 0

Student Login: This process represents the login functionality for students. It takes inputs such as the student's name and password. Based on the provided inputs, it determines the next action the student wants to perform (choice filling, allocation, or changing preferences).

College Login: This process handles the login functionality for college administrators. It takes input in the form of the college name. Its purpose is to authenticate the college administrator and provide access to specific functionalities.

Main Function: This process acts as the main control of the system. It takes input in the form of the user's choice (student or college) and invoke the joint seat allocation system bubble.



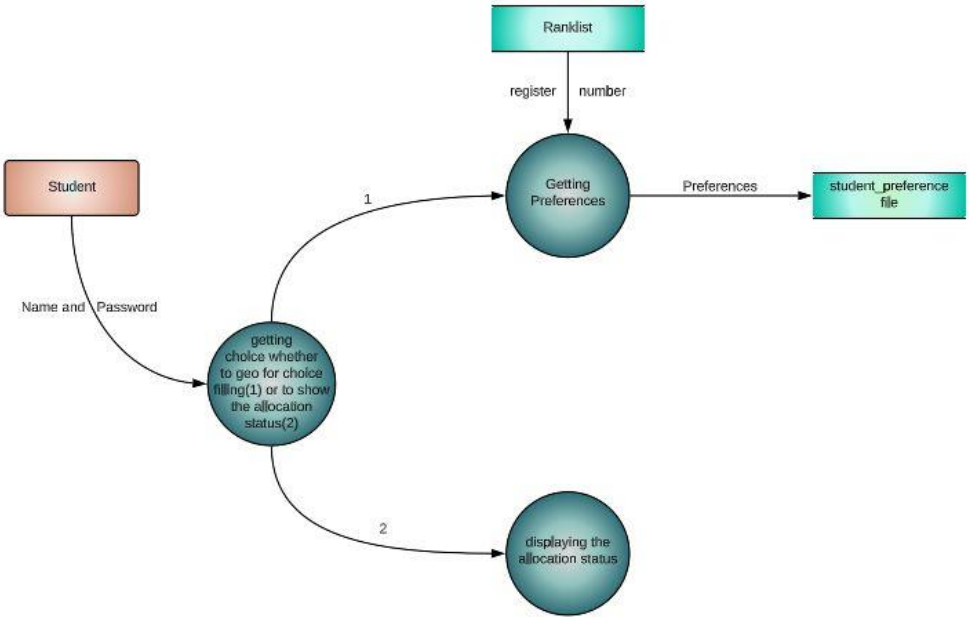

## Level 1

The joint seat allocation system invokes 3 bubbles

1.1: Getting choices from the students: This process gets the preferences from the students and shows their respective allocation.

1.2: Allocating the students: This is the main bubble where the algorithm works and allocate the seats according to their ranks.

1.3: Displaying the allotted students: This process shows the students allotted all the courses in a particular college when the college admin logins.




**Level 2.1**

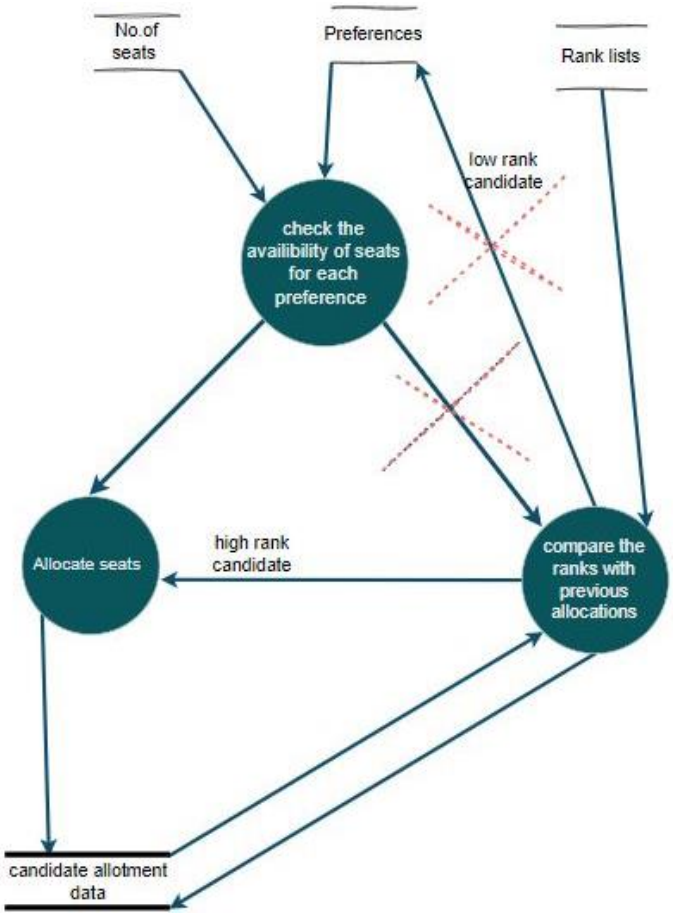
In this Authentication process as Student:

Getting choices: This process gets the name and password from the student and invoke the choice bubble.

Choice bubble: This asks the user to enter 1 for choice filling bubble and 2 for displaying allocation status bubble.

Choice filling bubble: Here the student's register number is taken from the ***rank list*** and asks the student to enter his/her choices and the preferences are stored in the ***student preference*** file.

Displays allocation status bubble: This bubble displays the college and course to which the student is allotted.



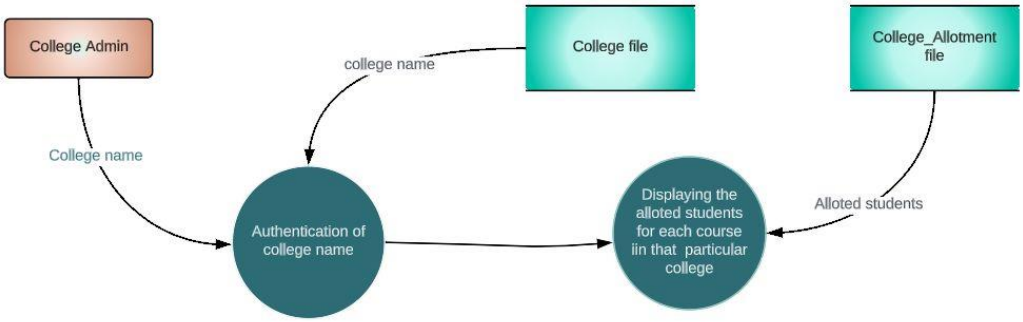

**Level 2.2**

In the allocating process:

Checking availability of seats: This bubble gets the preferences of the students from the ***"preferences"*** file and checks the availability of seats for the preference from getting info from the ***"No. of seats"*** file. If seat is available, then it invokes the allocate seat bubble. If there is no availability it invokes compare the ranks bubble.

Allocate seats: This process allocates the student to that preference and store that in the***"candidate allotment"*** file.

Compare the rank: This process compares the rank of the already allotted least rank student with the current student's rank by accessing the ***"rank lists"*** file. And passes the high rank candidate to the allocate seat bubble and low rank to preference file to get the next preference of that candidate.




**Level 2.3**

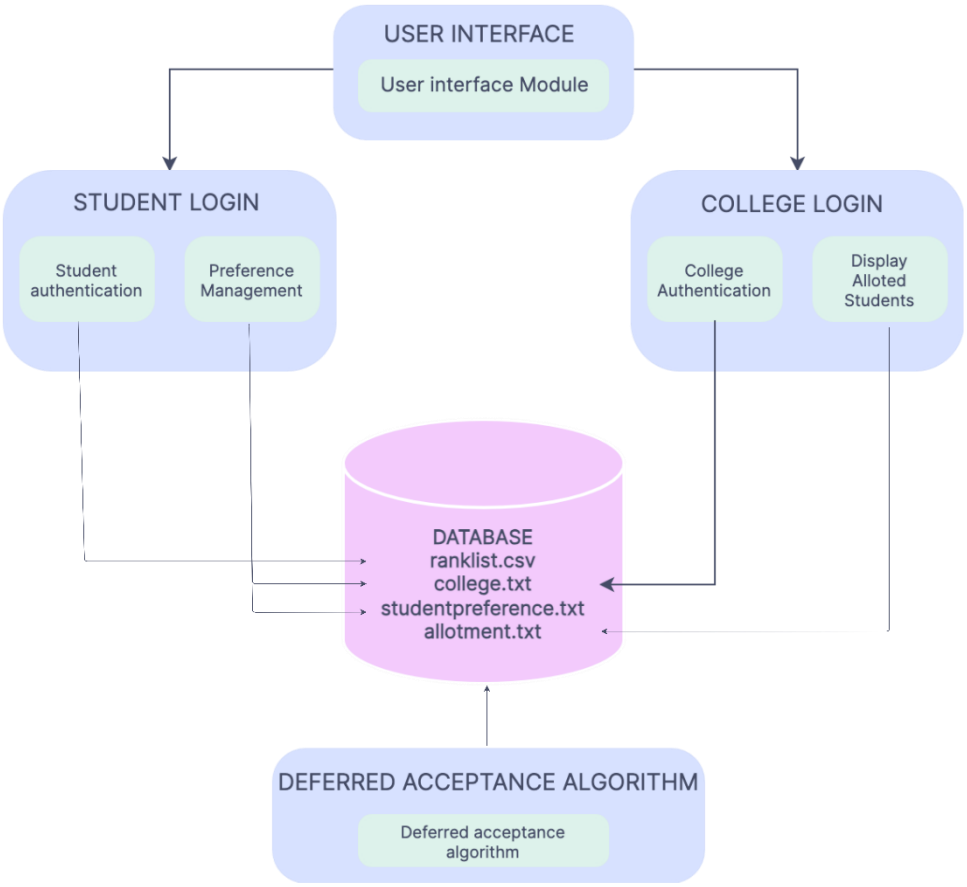
Displaying the allocated students:

Checking for valid college name: This process gets the name of the college from the college admin and check whether the name is valid or not by accessing the ***college*** file.

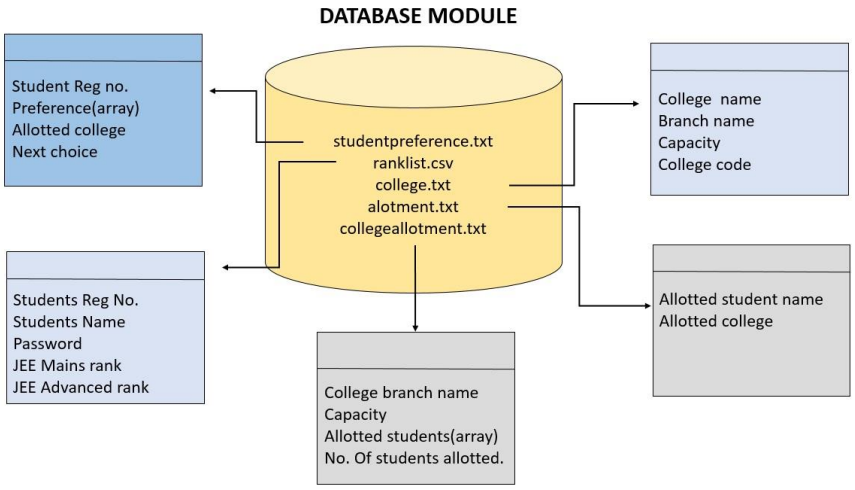
Displaying allotted students: This process displays the students allotted for each branch in the college by accessing the ***"college allotment"*** file.

# DETAILED DESIGN

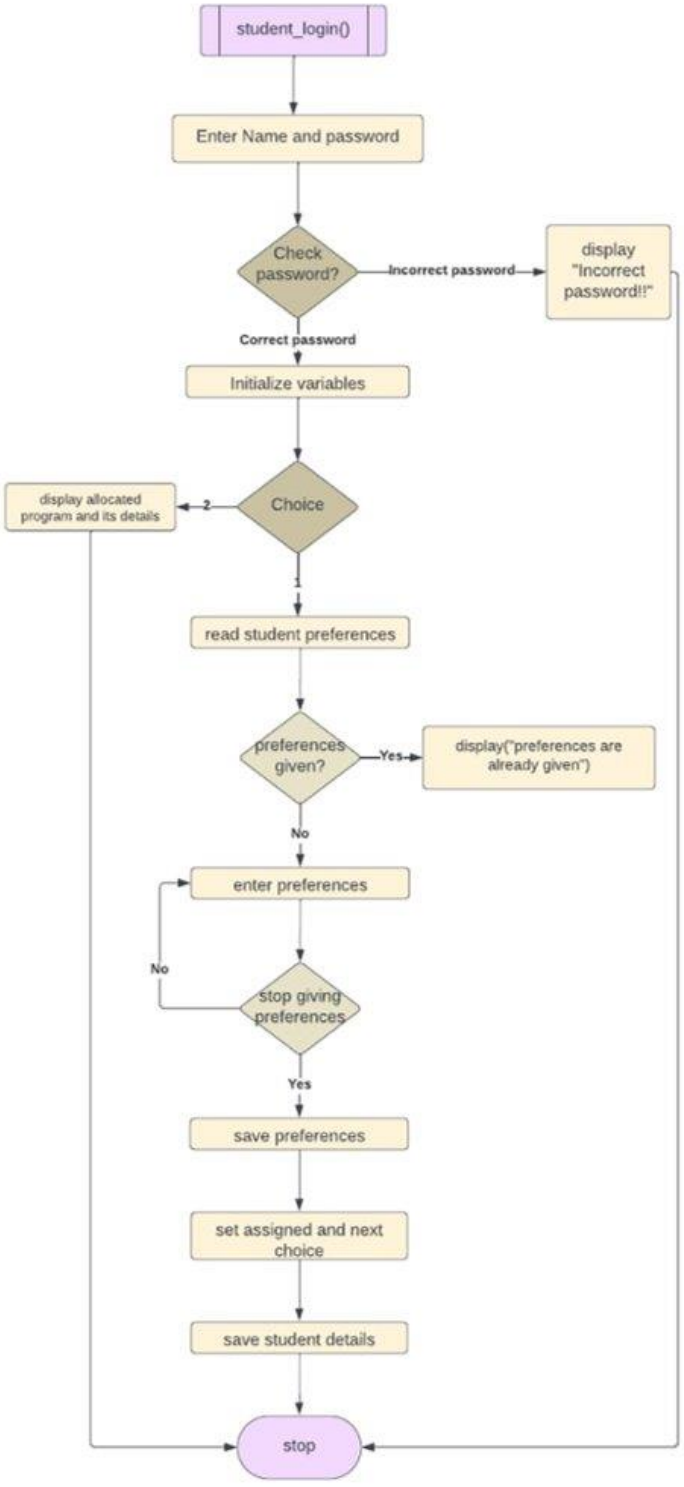
## Architectural Design



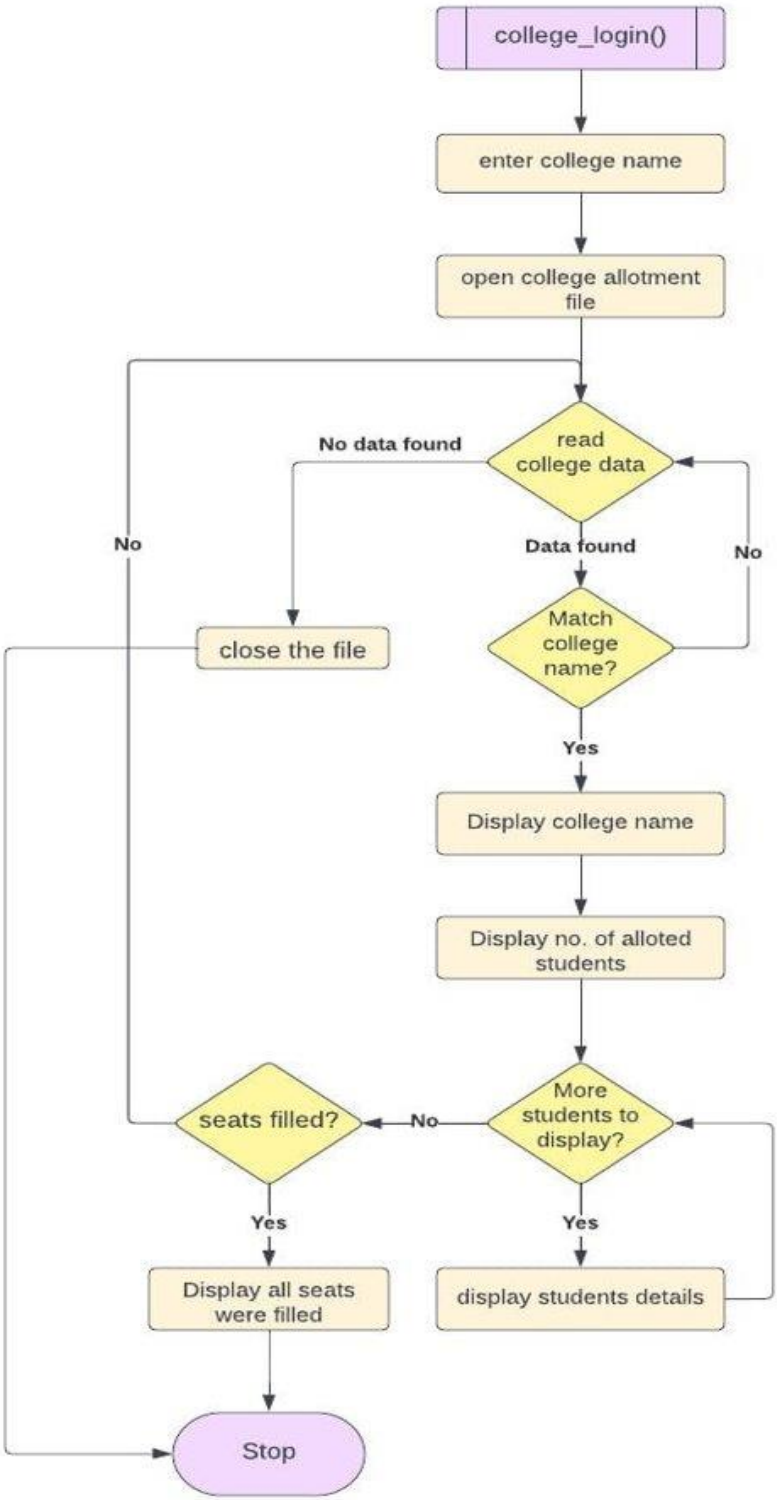

## Database Module



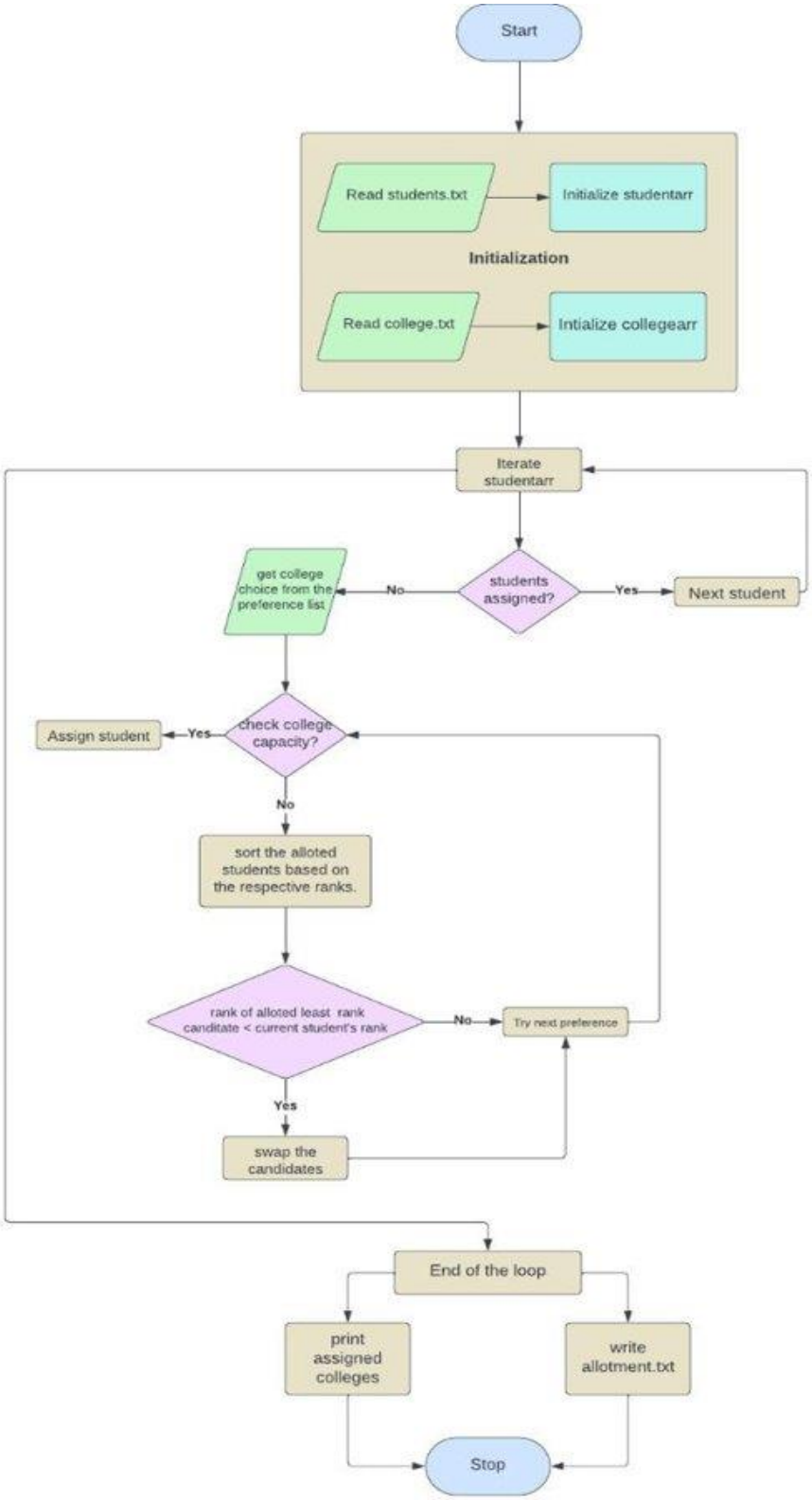
student_login() module



college_login() module



# Main Module


Start
Read students.txt
Initialize studentarr
Initialization
Read college.txt
Intialize collegearr
Iterate studentarr
students assigned?
Yes
Next student
No
get college choice from the preference list
check college capacity?
Yes
Assign student
No
sort the alloted students based on the respective ranks.
rank of alloted least rank canditate < current student's rank
No
Try next preference
Yes
swap the candidates
End of the loop
print assigned colleges
write allotment.txt
Stop

# DETAILED DESCRIPTION OF MODULES

**Database module:**

**File-based Data Storage:**

- The code uses several text files to store and retrieve data.

These files include:

- ***"studentpreference.txt"***: Stores student preference information.
- ***"College":*** Store the information of college course (college code, capacity).
- ***"Allotment.txt"***: Stores the allocation details of students.
- ***"college_allotment.txt"***: Stores the allocation details of colleges.
- ***"ranklist.csv"***: Contains data related to students' names, ranks, and passwords.

**Reading Data:**

- The code includes functions such as ***'no_of_students()'***, ***'check_password()'***, ***'is_present()'***, and ***'check_advanced()'*** that read data from ***files.no_of_students()*** read the number of students from the ***"studentpreference.txt"*** file.
- ***'check_password()'*** reads the ***"ranklist.csv"*** file to check if the entered student's name and password match.
- ***'is_present()'*** checks if a given program is present in the ***"D:\college.txt"*** file and returns the corresponding program number.
- ***'check_advanced()'*** reads the ***"ranklist.csv"*** file to check if a student has written the JEE Advanced exam.

**Writing Data:**

- The code includes functions such as ***'write_clg()'*** and ***'display_alloted_program()'*** that write data to files.
- ***'write_clg()'*** writes the allocated program information to the console.
- ***'display_alloted_program()'*** reads the ***"Allotment.txt"*** file to display the allocated program for a given student.

**File Format:**

- The code assumes a specific file format for each file it reads. For example, ***"ranklist.csv"*** is expected to have columns for student names, ranks, and passwords.
- Other files also have predefined formats to store data in specific fields. While the code does not use a traditional database management system like SQL or NoSQL databases, it leverages file operations to simulate data storage.
- However, it's important to note that this file-based storage approach has limitations in terms of scalability, and data consistency compared to dedicated database systems.

**student_login() module:**

- The function starts by reading the student's name and password as input.
- It then checks if the entered name and password match the records using the ***'check_password'*** function. If the match is found, the process continues; otherwise, it displays an ***"Incorrect Name or password!"*** message and exits the function.
- If the password match is successful, the function proceeds to check if the student has already given their preferences. It reads the student preferences from a file using ***'fread'*** and compares the entered password with the preferences stored in the file. If the preferences are found, it displays a message stating ***"You have already given the preferences."*** sets ***'prefcheck'*** flag to 0 and continues to the next step.
- If the student has not given preferences (***'prefcheck'*** flag is 1), the function prompts the student to choose between preference filling and allocation by entering '1' or '2' as the 'allot' choice. If the student chooses preference filling (choice '1'), the function enters a loop where the student can enter their preferences. It prompts the student to enter a program (college name with branch), checks if the program is valid using ***'is_present'*** function, and ensures that the same preference is not entered twice. If a duplicate preference is detected, it displays a message and continues to the next iteration of the loop. If the program is invalid, it displays an ***"Invalid Program Name"*** message and provides an option to view the list of institutes and branches using the ***'show'*** function. Once the preferences are entered or the loop is stopped, the student's preferences are stored in the file along with other relevant data.
- If the student chooses allocation (choice '2'), the function calls the ***'display_alloted_program'*** function to display the program allocated to the student based on their name.
- Finally, if the password match is unsuccessful, it displays an ***"Incorrect Name or password!"*** message and exits the function. The flowchart represents the high-level flow of the ***student_login()*** function, illustrating the sequence of steps and decision points based on the input and conditions

**college_login() module:**

- The function starts by reading the college name from the user.
- It opens the ***'college_allotment.txt'*** file, which contains information about college allotments and assigned students.
- The function iterates through the college entries in the file (assumed to be 16 colleges based on the loop condition). For each college entry, it reads the college information using ***'fread'*** into the clg structure.
- The function checks if the entered college name matches the current college's name using case-insensitive comparison with ***'strstr'*** and ***'strupr'*** functions. If a match is found, it proceeds to display the college name.
- It displays the number of students allotted to the branch in the college ***('clg.num_stu')***.

- It iterates through the assigned students for that branch and displays their information using the ***'check_pass*** function'.
- After displaying the assigned students, the function checks if all the seats in the branch (***'clg.capacity'***) have been filled.
- If the capacity matches the number of students allotted (***'clg.num_stu'***), it displays a message indicating that all seats have been filled.
- If no match is found for the entered college name, it displays an ***"Invalid college name"*** message. Finally, the function closes the college file and exits.

**Main module:**

- The function starts by declaring variables and opening the studentpreference.txt file to read student preferences.
- It reads the student preferences from the file into the studentarr array.
- The function opens the college.txt file to read college information.
- It reads the college information from the file into the collegearr array.
- The function initializes variables and arrays for further processing.
- It enters a loop for stable matching until all students are assigned.
- Within the loop, it checks unassigned students.
- If an unassigned student is found, it checks if the chosen college has available capacity.
- If there is available capacity, the student is assigned to the college.
- If there is no available capacity, the function sorts and reassigns students based on preferences.
- The student's preferences are updated, the function writes the allotment information to the Alloted.txt file.
- It writes the college allotment information to the college_allotment.txt file.
- Finally, the function displays the allotment information to the console.

The function ends.

## IMPLEMENTATION

1. **Explanation of how the data is organized and the Rationale behind the selection of a particular language construct (Arrays, Structures, Array of Structures, Files etc.)**

- Arrays: Arrays are used to store data related to preferences, assigned students, and other information. For example, the ***'pref'*** array in the '**Student'** structure stores the preferences of a student, and the ***'assigned_stu'*** array in the '**College'** structure stores the names of students assigned to a particular branch in a college.

- Structures: Structures are used to define custom data types that encapsulate related information. The '**Student'** structure stores information about a student, including their name, preferences, assigned branch, and next choice. The '**College'** structure stores information about a college, including its name, capacity, assigned students, and the number of students allotted to a branch.

- Array of Structures: An array of structures is used to store multiple instances of the same structure type. In this code, an array of '**Student'** structures (**studentarr**) and an array of '**College'** structures (**collegearr**) are used to store information about multiple students and colleges, respectively.

- Files: Files are used for reading and writing data. The code reads data from files like ***"studentpreference.txt", "Allotment.txt",*** and ***"college_allotment.txt"***. It also writes the allocated seats and assigned students to the ***"Allotment.txt"*** file.

2. **Explanation of any other libraries or APIs that have been used**

**The code includes several header files:**

- ***stdio*.h**: Provides input/output functions like ***printf*** and ***scanf***.
- ***string*.h**: Provides string manipulation functions like ***strcpy*** and ***strcmp***.
- ***stdlib*.h**: Provides general-purpose functions like ***atoi*** and ***fclose***.
- ***strings*.h**: Provides additional string manipulation functions like ***strupr***.
- **"*functions*.h"**: Contains additional functions used in the code. The content of this header file is not provided in the given code snippet.

The **"*functions*.h"** header file likely contains function prototypes and definitions for various operations related to student and college preferences, password checking, file handling, and more. Since the actual content of this header file is not provided in the code snippet, it is not possible to provide a detailed explanation of its contents.

**Explanation of Each function in *"functions.h"*:**

- The code starts with various header file inclusions, function prototypes, and structure definitions for the Student and College data types.
- The ***'no_of_students'*** function reads the student preference data from a file and returns the count of students.
- The ***'trim'*** function is a utility function used to remove leading and trailing spaces from a string.
- The ***'rank'*** function reads a CSV file containing rank information and returns the rank value for a given student name and college choice.
- The ***'sort'*** function is used to sort an array of assigned students based on their ranks for a particular college choice.
- The ***'prefers'*** function checks if a student prefers a particular choice over the previous assigned student for a given college.
- The ***'main'*** function is the entry point of the program. It performs the seat allocation process using the **Gale-Shapley algorithm**. It reads student and college data from files, processes student preferences, and allocates students to colleges based on their choices and availability of seats.
- The ***'student_login'*** function allows students to log in, provide their preferences, and view their allocated program. The college_login function allows college administrators to log in and view the allocated students for their respective colleges.
- The ***'check_password'*** function checks if a student's name and password match the data in a CSV file containing rank and password information.
- The ***'write_clg'*** function reads the college names and codes from a file and displays the allocated program based on the college code.
- The ***'display_alloted_program'*** function reads the allocated program data from a file and displays the program allocated to a particular student. The ***is_present*** function checks if a given program is present in a file containing a list of college programs.
- The ***'check_advanced'*** function checks if a student has appeared for the JEE Advanced exam by reading rank information from a CSV file. The show function displays the list of colleges and their programs from a file.

## 3. User interface design

- The user interface design in the code is based on console input/output.
- Students are prompted to enter their name and password for login. They are then provided with options for choice filling, allocation, or preference changing.
- College admins are prompted to enter the college name for login. They are then shown the number of students allocated to each branch in their college.

4. **Platform used for Code Development:** Dev C++

# DETAILED TEST CASES

Ranklist.csv:

| reg.no | JEE MAINS | JEE ADVAN | name | password |
|---|---|---|---|---|
| 2 | 2 | 0 | vishnuprak | vishnu24 |
| 3 | 3 | 4 | sweety | sweety03 |
| 6 | 6 | 8 | varsha | varsha07 |
| 7 | 7 | 5 | yuva | yuva18 |
| 10 | 10 | 9 | asvikka | asvikka02 |
| 1 | 1 | 2 | blesslin | blesslin04 |
| 4 | 4 | 3 | deepitha | deepitha03 |
| 5 | 5 | 6 | abirami | abirami09 |
| 8 | 8 | 7 | hariharan | hariharan21 |
| 9 | 9 | 10 | harish | harish34 |
| 12 | 12 | 11 | pranesh | pranesh37 |
| 13 | 13 | 14 | varshini | varshini56 |
| 16 | 16 | 15 | vidisha | vidisha23 |
| 17 | 17 | 18 | ramya | ramya16 |
| 20 | 20 | 19 | siva | siva18 |
| 11 | 11 | 12 | ram | ram22 |
| 14 | 14 | 13 | vishnu | vishnu345 |
| 15 | 15 | 16 | prabha | prabha123 |
| 18 | 18 | 17 | vikas | vikas15 |
| 19 | 19 | 20 | jana | jana23 |
| 22 | 22 | 21 | navasri | navasri17 |
| 23 | 23 | 24 | kavisika | kavisika17 |
| 26 | 26 | 25 | kamal | kamal06 |
| 27 | 27 | 28 | sindhu | sindhu45 |
| 30 | 30 | 29 | nive | nive10 |

Ranklist

College.txt:

```
IITMadras,CSE 2 1
IITMadras,MECH 2 2
IITMadras,CIVIL 2 3
IITMadras,AIDS 2 4
IITBombay,CSE 2 5
IITBombay,IT 2 6
IITBombay,ECE 2 7
IITBombay,EEE 2 8
IITGuwahati,BIOTECH 2 9
IITGuwahati,IT 2 10
IITGuwahati,ECE 2 11
IITGuwahati,EEE 2 12
IITDelhi,MECH 2 13
IITDelhi,CIVIL 2 14
IITDelhi,BIOTECH 2 15
IITDelhi,AIDS 2 16
NITTrichy,CSE 2 17
NITTrichy,MECH 2 18
NITTrichy,CIVIL 2 19
NITTrichy,AIDS 2 20
NITWarangal,CSE 2 21
NITWarangal,IT 2 22
NITWarangal,ECE 2 23
NITWarangal,EEE 2 24
NITNagpur,BIOTECH 2 25
NITNagpur,IT 2 26
```

Test case 1:
Input:

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:varsha
Enter your Password:varsha07
Your register number is:6
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefernces

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:iitmadras,cse
Enter the Program:nittrichy,cse
Enter the Program:1

--------------------------------
Process exited after 22.87 seconds with return value 0
Press any key to continue . . .
```

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:vishnupraba
Enter your Password:vishnu24
Your register number is:2
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefernces

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:nittrichy,cse
Enter the Program:nitwarangal,cse
Enter the Program:1

--------------------------------
Process exited after 31.41 seconds with return value 0
Press any key to continue . . .
```

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:blesslin
Enter your Password:blesslin04
Your register number is:1
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefer

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:nittrichy,cse
Enter the Program:1

--------------------------------
Process exited after 14.67 seconds with return value 0
Press any key to continue . . .
```

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:yuva
Enter your Password:yuva18
Your register number is:7
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefernces

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:iitmadras,cse
Enter the Program:1

--------------------------------
Process exited after 17.2 seconds with return value 0
Press any key to continue . . .
```

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:sweety
Enter your Password:sweety03
Your register number is:3
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefernces

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:iitmadras,cse
Enter the Program:1

--------------------------------
Process exited after 20.05 seconds with return value 0
Press any key to continue . . .
```

Expected Output:

Since, 5 students have given their preferences. Seats were allotted to everyone as per their preferences except the student named" varsha" because of the availability of seats.

Output: Algorithm Check

```
sweety
Congrats!!
You have been alotted to IITMadras,CSE.

vishnupraba
Congrats!!
You have been alotted to NITTrichy,CSE.

yuva
Congrats!!
You have been alotted to IITMadras,CSE.

blesslin
Congrats!!
You have been alotted to NITTrichy,CSE.

varsha
```

Since no colleges have been allotted to “varsha”. The available seats in the other colleges will be displayed.

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:varsha
Enter your Password:varsha07
Your register number is:6
Enter 1 for Choice Filling 2 for Allocation:2
You are not alloted.
The colleges with available seats are:
IITMadras,MECH     2
IITMadras,CIVIL     2
IITMadras,AIDS     2
IITBombay,CSE     2
IITBombay,IT     2
IITBombay,ECE     2
IITBombay,EEE     2
IITGuwahati,BIOTECH     2
IITGuwahati,IT     2
IITGuwahati,ECE     2
IITGuwahati,EEE     2
IITDelhi,MECH     2
IITDelhi,CIVIL     2
IITDelhi,BIOTECH     2
IITDelhi,AIDS     2
NITTrichy,MECH     2
NITTrichy,CIVIL     2
```

Test case 2:
Input:

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY----------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:varsha
Enter your Password:varsha07
Enter 1 for Choice Filling 2 for Allocation1
Enter 1 to stop giving prefences and 3 to change any of ur prefernces

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:iitmadras,cse
Enter the Program:nittrichy,cse
Enter the Program:nitwarangal,cse
Enter the Program:1

--------------------------------
Process exited after 51.5 seconds with return value 0
Press any key to continue . . .
```

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:vishnupraba
Enter your Password:vishnu24
Your register number is:2
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefernces

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:nittrichy,cse
Enter the Program:nitwarangal,cse
Enter the Program:1

--------------------------------
Process exited after 31.41 seconds with return value 0
Press any key to continue . . .
```

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:blesslin
Enter your Password:blesslin04
Your register number is:1
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefer

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:nittrichy,cse
Enter the Program:1

--------------------------------
Process exited after 14.67 seconds with return value 0
Press any key to continue . . .
```

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:yuva
Enter your Password:yuva18
Your register number is:7
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefernces

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:iitmadras,cse
Enter the Program:1

--------------------------------
Process exited after 17.2 seconds with return value 0
Press any key to continue . . .
```

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:sweety
Enter your Password:sweety03
Your register number is:3
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefernces

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:iitmadras,cse
Enter the Program:1

--------------------------------
Process exited after 20.05 seconds with return value 0
Press any key to continue . . .
```

Expected output: Now, the college will be allotted to student named “varsha”.
OUTPUT:

```
sweety
Congrats!!
You have been alotted to IITMadras,CSE.

vishnupraba
Congrats!!
You have been alotted to NITTrichy,CSE.

yuva
Congrats!!
You have been alotted to IITMadras,CSE.

blesslin
Congrats!!
You have been alotted to NITTrichy,CSE.

varsha
Congrats!!
You have been alotted to NITWarangal,CSE.
```

Some User-friendly techniques that we have used:

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:Varsha
Enter your Password:varsha09
Your register number is:151
Enter 1 for Choice Filling 2 for Allocation:
```

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:varsha
Enter your Password:varsha07
Your register number is:6
Enter 1 for Choice Filling 2 for Allocation:
```

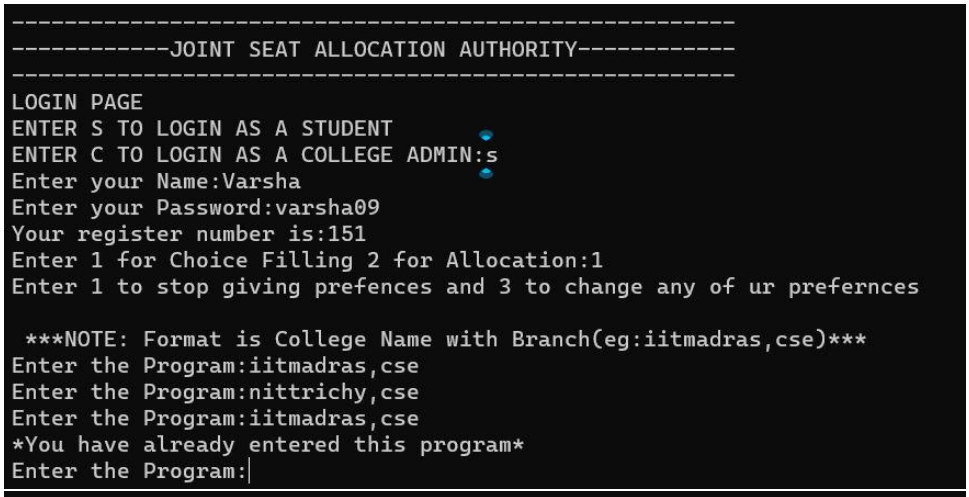
```
-------------------------------------------------
-----------JOINT SEAT ALLOCATION AUTHORITY-----------
-------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:Varsha
Enter your Password:varsha09
Your register number is:151
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefernces

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:vit,cse
Invalid Program Name
   Enter 1 to see the Institutes and Branches 2 to continue:2
Enter the Program:iitmadras,cse
Enter the Program:
```

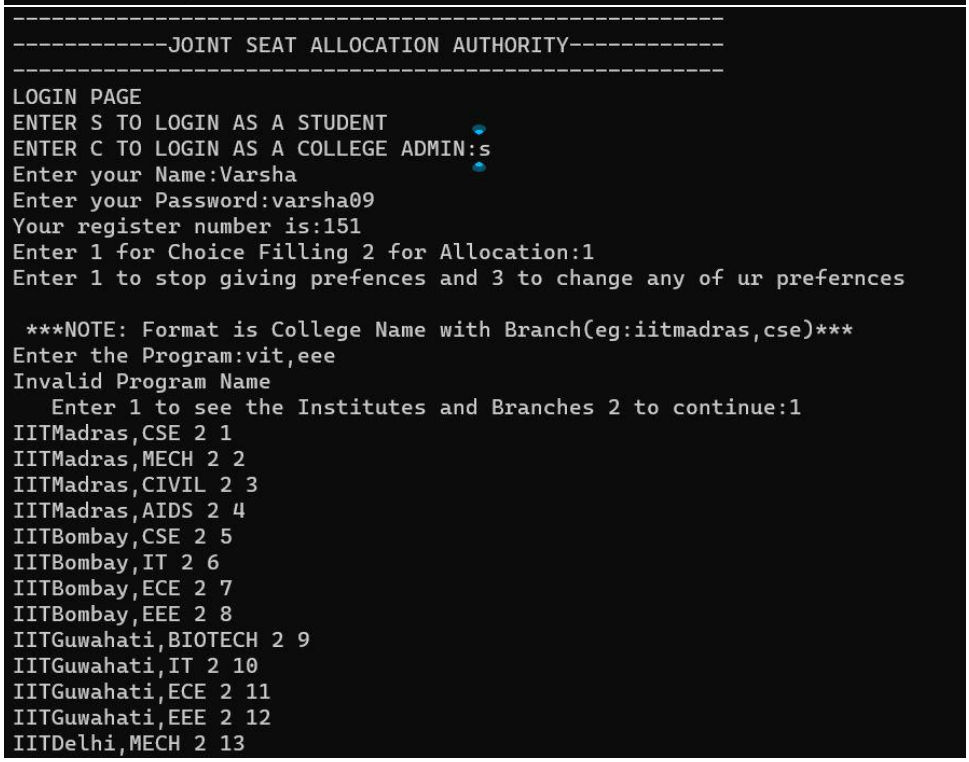
```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:varsha
Enter your Password:varsha07
Your register number is:6
Enter 1 for Choice Filling 2 for Allocation:1
You have already given the preferences
--------------------------------
Process exited after 8.346 seconds with return value 0
Press any key to continue . . .
```

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:Varsha
Enter your Password:varsha09
Your register number is:151
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefernces

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:iitmadras,cse
Enter the Program:nittrichy,cse
Enter the Program:iitmadras,cse
*You have already entered this program*
Enter the Program:
```

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:Varsha
Enter your Password:varsha09
Your register number is:151
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefernces

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:iitmadras,cse
Enter the Program:nitwarangal,cse
Enter the Program:3
enter the college name to be changed:nitwarangal,cse
enter the clg to be added:nittrichy,cse
Enter the Program:
```

```
------------------------------------------------------
------------JOINT SEAT ALLOCATION AUTHORITY------------
------------------------------------------------------
LOGIN PAGE
ENTER S TO LOGIN AS A STUDENT
ENTER C TO LOGIN AS A COLLEGE ADMIN:s
Enter your Name:Varsha
Enter your Password:varsha09
Your register number is:151
Enter 1 for Choice Filling 2 for Allocation:1
Enter 1 to stop giving prefences and 3 to change any of ur prefernces

 ***NOTE: Format is College Name with Branch(eg:iitmadras,cse)***
Enter the Program:vit,cse
Invalid Program Name
   Enter 1 to see the Institutes and Branches 2 to continue:2
Enter the Program:iitmadras,cse
Enter the Program:
```




## LIMITATIONS OF THE SOLUTION

- While the deferred acceptance algorithm is a powerful tool for joint seat allocation, it is not without its challenges and limitations.
- The Main challenge is the need for accurate and reliable data, which can be difficult to obtain in practice.
- It also assumes that participants act rationally and truthfully, which may not always be the case.
- These challenges and limitations must be carefully considered when designing and implementing a joint seat allocation system.

## OBSERVATIONS FROM THE SOCIETAL, LEGAL, ENVIRONMENTAL AND ETHICAL PERSPECTIVES

- We have considered various perspectives to ensure a holistic solution.
- From a societal standpoint, our solution focuses on fairness, transparency, and equal opportunities in the admission process.
- Candidate data should be collected, stored, and used responsibly, ensuring that it is only used for the intended purpose of the admission process and not shared or exploited for other purposes without proper consent.
- Our solution is also user friendly.

## LEARNING OUTCOMES

1. Understanding File Handling: The code provides practical experience in working with file operations, demonstrating how to read and write data to files using functions like `fopen`, `fread`, `fwrite`, and `fclose`. It focuses on file handling techniques using .csv and .txt files.

2. Algorithm Design: The code implements a stable matching algorithm for allocating students to colleges based on their preferences and capacities. It helps learners understand the logic and design principles behind such algorithms.

3. String Manipulation: The code involves various string manipulation operations, such as tokenizing, trimming, and comparison. By working with the code, learners can gain experience in manipulating strings and understanding different techniques.

4. User Input and Validation: The code handles user input for login credentials, preferences, and program choices. It demonstrates techniques for input validation, error handling, and ensuring data integrity.

5. Modular Programming: The code is organized into several functions that perform specific tasks, promoting modular programming practices. Analyzing the code enhances understanding of modular design and code organization.

6. Data Flow Diagrams (DFDs): Creating DFDs for the code helps visualize the system's data flow and understand the interactions between various components and processes.

7. Problem Solving and Optimization: The code implements an allocation algorithm that handles scenarios where preferences exceed capacities. Studying the code enhances problem-solving skills and provides insights into optimizing resource allocation algorithms.

## REFERENCES

*https://www.lucidchart.com/blog/how-to-design-software-architecture*

*https://en.wikipedia.org › wiki › Gale–Shapley_algorithm*

*https://in.docs.wps.com/l/sINWqnr_HAYCQ_aMG?sa=00&st=0t&v=v2*

*http://econweb.umd.edu/~vincent/econ415/Lecture415%2810%29MatchingO2O.pdf*