

Software System For Engineering Joint Seat Allocation

UCS2201 – Fundamentals and Practice of Software Development

A PROJECT REPORT

Submitted By

Micheal Berdinth M 3122 22 5001 071

Prashanna Kumar S 3122 22 5001 094



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 “**Software System For Engineering Joint Seat Allocation**” is the bonafide work of “Micheal Berdinth M (3122 22 5001 071) and Prashanna Kumar S (3122 22 5001 094)” 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:19/07/2023

TABLE OF CONTENTS

Content	Page Number
1. Problem Statement	1
1.1 Abstract	2
1.2 Introduction	2
2. Extended exploration of problem statement	3
3. Analysis using Data Flow Diagrams	5
4. Overall Architecture Diagram	8
5. Description of each Module	9
6. Implementation	12
7. Validation through different test cases	15
8. Limitation of the solution provided	20
9. Observations from the Societal, Legal, Environmental and Ethical perspectives	21
10. Learning Outcomes	22
11. References	23

1.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.

Constraints:

- Seat allotment for a candidate must be from the list of choices given by the candidate
- Number of preferences given by each candidate is limited to the number of institutes times the number of branches in each institute
- Each candidate must be allotted only one of his/her choices
- All the available seats in all the branches in all the Institutes must be filled.
- If a student is denied a particular choice, then all those who were allotted that choice must be higher in the respective rank list (Merit should not be violated)

Input:

- Number of institutes
- Number of branches in each institute
- Number of seats for each branch in each institute
- Rank list of candidates for each qualifying exam

Output:

Allocation of institute and branch to each candidate as per policy

Abstract:

This project develops a counseling system for student seat allocation in educational institutes. It reads student choices and institute details from input files. Seats are allocated based on student preferences and ranks, considering both JEE Mains and JEE Advanced. Available seats are checked, and lower-ranked students may be replaced. The system generates a seat allocation report with college codes, names, branches, and student details. Limitations include memory complexity due to a 2D array for branches. Overall, it enhances seat allocation efficiency, promoting fairness and transparency in the admission process.

Introduction:

- Imagine a scenario where thousands of aspiring students are trying for admission to prestigious engineering institutes, such as the Indian Institutes of Technology (IITs) and National Institutes of Technology (NITs).
- The admission process involves numerous institutes, each offering different branches of engineering and a limited number of seats.
- The task of managing and allocating these seats to qualified candidates can be a daunting and complex process.
- This is where our software system comes into play, aiming to streamline the engineering counseling and admission process.

2. Extended exploration of problem statement

The objective of this project is to design and develop a software system to streamline the engineering counseling and admission process for two sets of institutes, namely IITs and NITs. The system aims to allocate seats to candidates based on their preferences and ranks obtained in the qualifying exams, JEE-Advanced and JEE-Main.

Key considerations in the problem statement include:

1. Multiple sets of institutes with different branches: The system should cater to two distinct sets of institutes, each offering various branches of study. For instance, IITs and NITs have their own specific branches such as Computer Science, Mechanical Engineering, etc. Each branch is associated with a predetermined number of available seats for admission.

2. Candidate preferences and rank lists: Candidates are required to provide a ranked list of preferences, indicating their desired institutes and branches. These choices are influenced by their individual interests and aspirations. Additionally, candidates hold ranks in either one or both of the qualifying exams, JEE-Advanced and JEE-Main, which significantly impact their eligibility for specific choices.

3. Constraints and policies: The seat allocation process must adhere to several constraints and policies. Firstly, it is necessary to assign a seat to each candidate from their list of preferences. Secondly, candidates are allowed to provide a maximum number of preferences, which is determined by the total

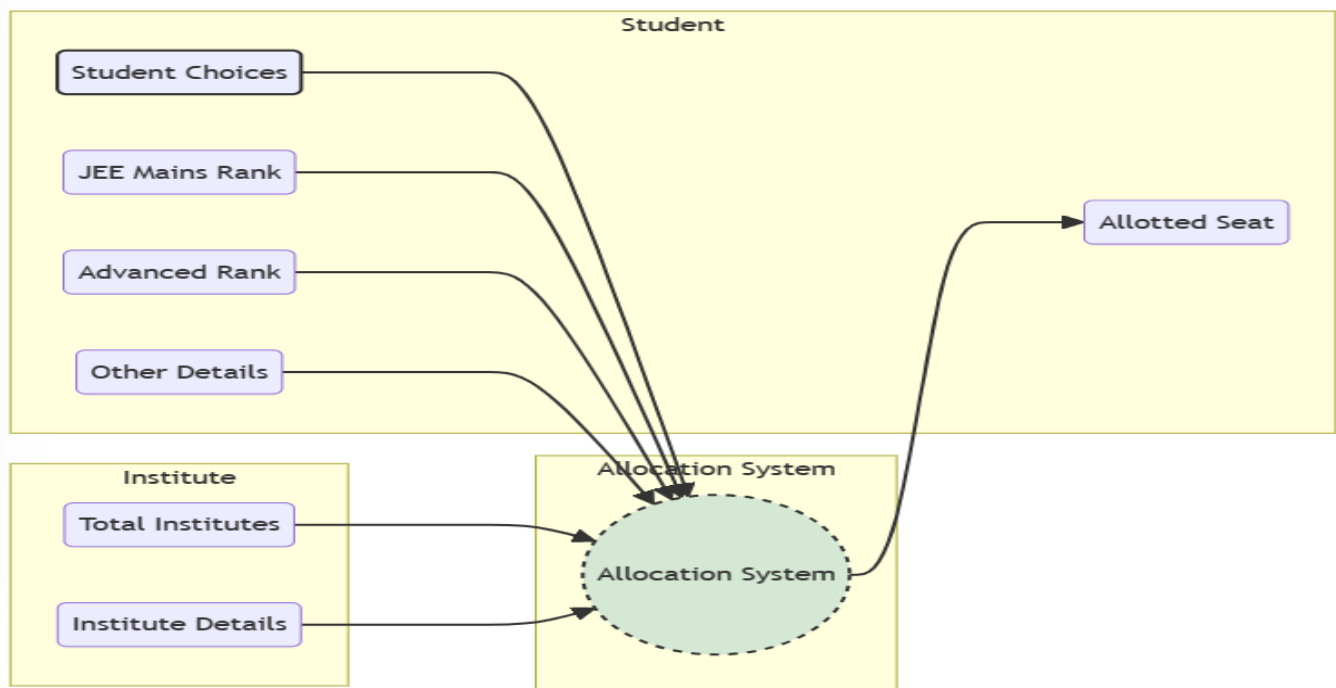
number of institutes multiplied by the number of branches per institute. This ensures candidates have ample options to choose from. Furthermore, each candidate can be allocated only one choice from their preferences. Lastly, it is crucial to fill all available seats across all branches in the institutes to optimize admission opportunities.

4. Maintenance of merit-based allocation: The system must uphold a merit-based allocation of seats to ensure fairness. In cases where a candidate is denied their preferred choice, it is essential to guarantee that all candidates who were allocated that choice possess higher ranks than him. This ensures that merit is respected throughout the allocation process.

The input to the system includes the number of institutes, the number of branches within each institute, the seat capacity for each branch, and the rank lists of candidates for the qualifying exams. The expected output is the allocation of institutes and branches to individual candidates based on the established policies and constraints.

3. Analysis using Data Flow Diagrams:

Level :0



The Level 0 DFD illustrates the flow of data between the student, the allocation system, and the institute. It captures the main inputs and outputs of the system.

Inputs from the Student:

- **Choices:** The student provides their preferred choices for institutes.
- **JEE Mains Rank:** The student submits their rank in the JEE Mains examination.
- **Advanced Rank:** If applicable, the student provides their rank in the Advanced examination.
- **Other Details:** Additional information about the student (e.g., personal details, preferences).

Inputs from the Institute:

- **Total Institutes:** The allocation system receives the total number of institutes available for seat allocation.
- **Institute Details:** The institute shares its name, code, branches, and seat count with the allocation system.

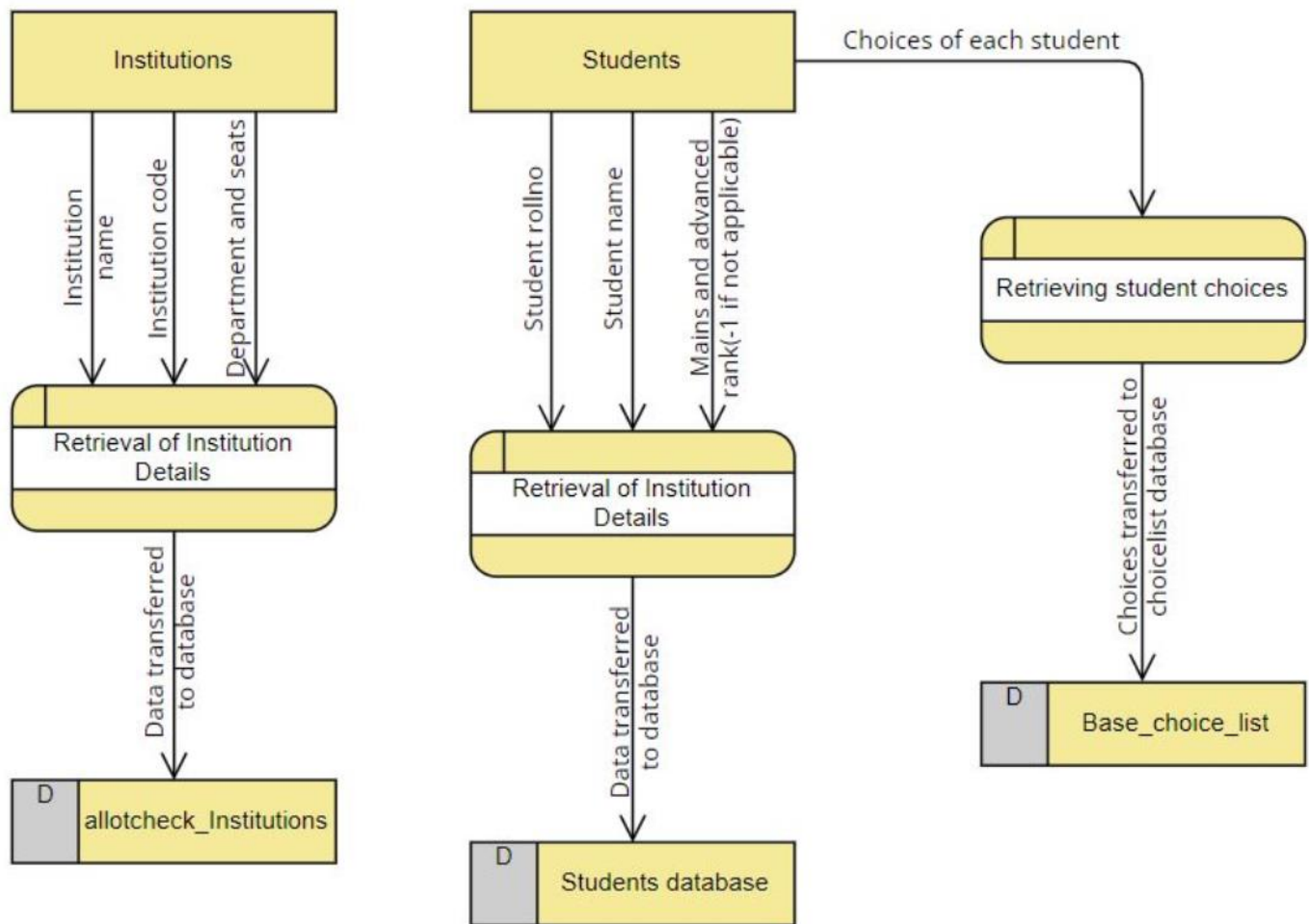
The Allocation System:

- **Processes the inputs and performs the allocation of seats using an algorithm (e.g., Gale-Shapley algorithm).**
- **Determines the allocated seat based on the student's choices, ranks, and institute details.**
- **Generates the output of the allocated seat.**

Outputs to the Student:

- **Allotted Seat:** The allocation system provides the student with the details of the seat allocated to them (e.g., institute name, branch).

Level-1 DFD:



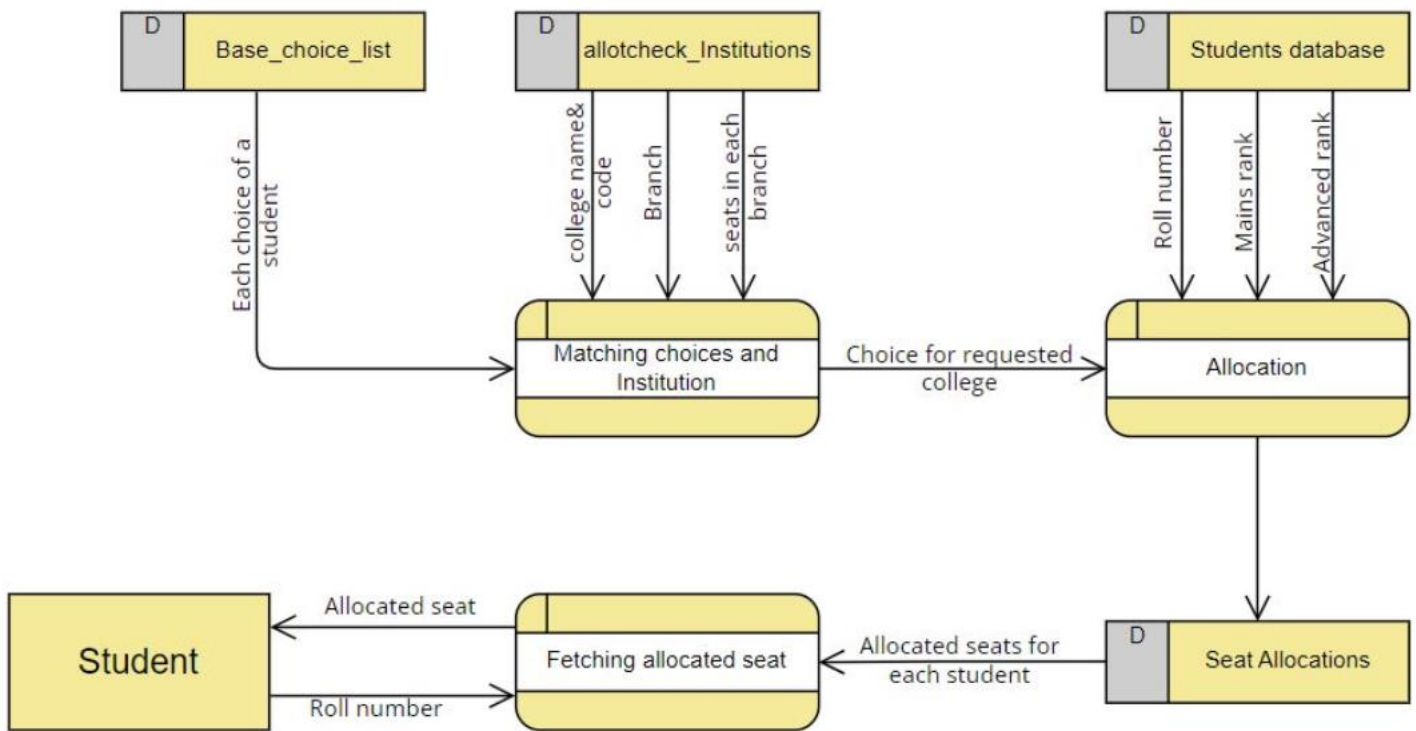
→ In Level 1 Data Flow Diagram (DFD), the system obtains data related to institutes and students for the seat allocation process. The institutes' data, including their names, codes, departments, and available seats, is acquired and transferred to the "allotcheck_Institutions.txt" file.

→ On the student side, the system prompts the students to provide their roll numbers, names, and ranks in both JEE Mains and Advanced exams (if applicable). These student details are collected to facilitate the seat allocation process. Additionally, the system retrieves each student's choices regarding their preferred institutes and programs.

→ The retrieved student choices are then transferred and stored in the "base_check_institutions.txt" file. This file serves as the base for further processing and allocation of seats to students based on their preferences and available seats in the respective institutes.

→ By organizing and storing this data, the system enables efficient matching of student choices with the institutes' available seats, facilitating the allocation process based on student preferences and merit.

Level- 2 DFD:

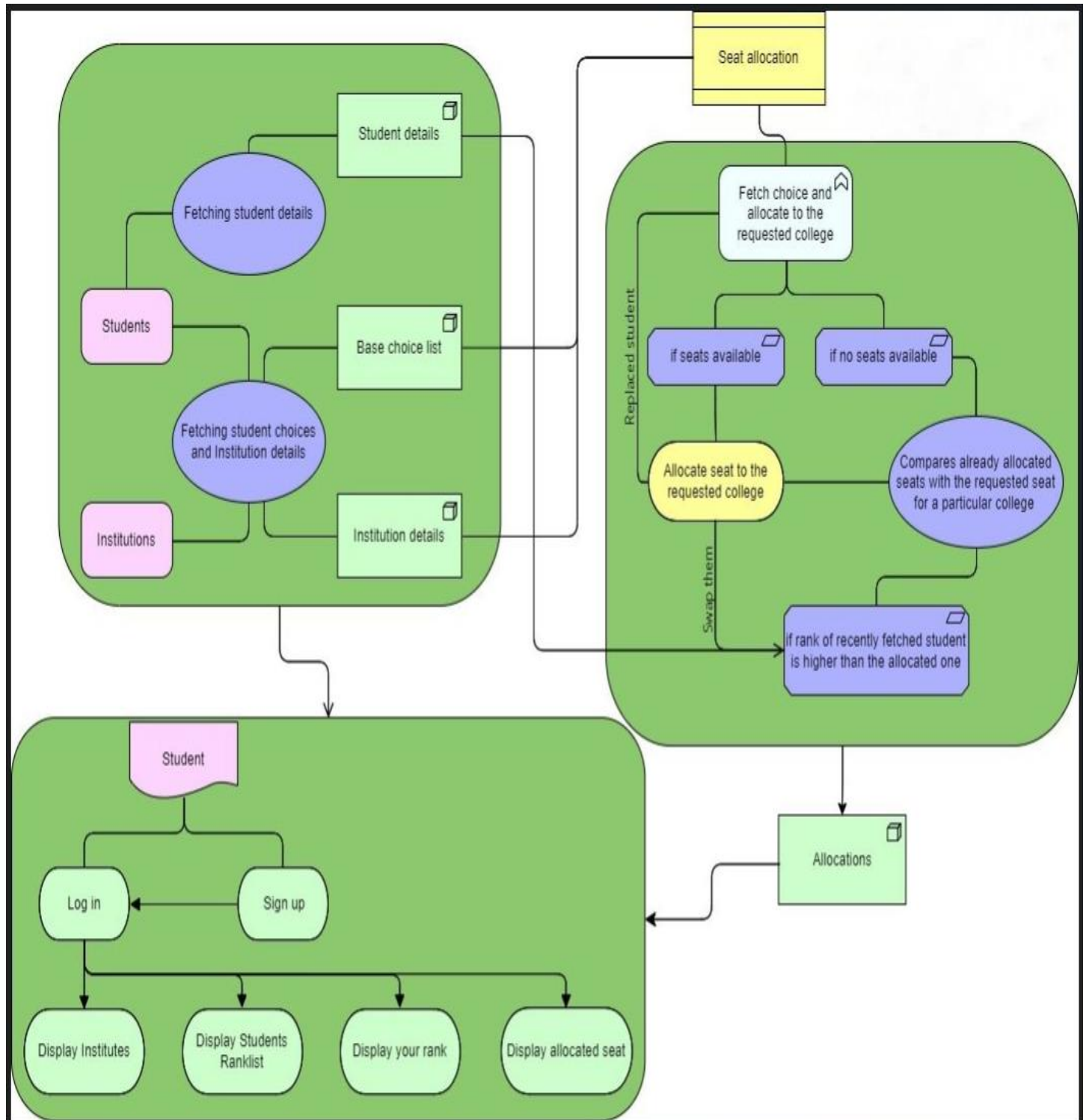


In Level 2 Data Flow Diagram (DFD), the system processes the data related to student choices, institute details, and student information to perform the seat allocation process. It occurs in the following steps,

1. The system retrieves each student's choice from the "base_choice_list.txt" file. Each choice consists of the student's roll number and the institute code and branch preference.
2. Simultaneously, the system obtains the institute details, including the institute's name, code, branches, and seat counts in each branch. This information is fetched from the allotCheck_Institutions.txt file.
3. The student's roll number, JEE Mains rank, and JEE Advanced rank are passed along with the student's choices and institute details to the "Allocation Module".
4. The "Allocation Module" utilizes the Gale-Shapley algorithm to determine the eligible seat allocation for each student. The algorithm considers the student's preferences, institute availability, and ranking to allocate seats optimally.
5. Once the seat allocation process is completed, the system retrieves the allocated seat information based on the student's roll number. This information includes the allocated institute's code, name, and branch.
6. The allocated seat information is then made available to the student, who can fetch it by providing their roll number.

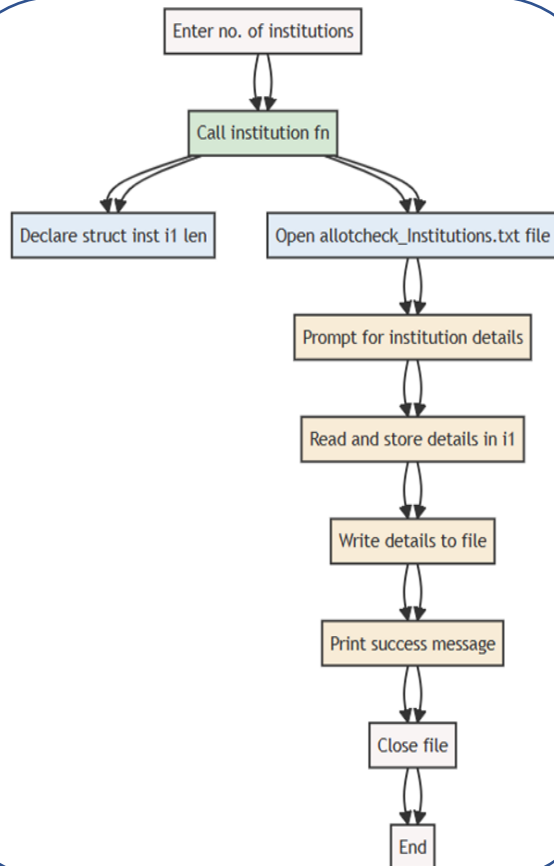
By using the Gale-Shapley algorithm and considering the student's preferences and ranking, the system ensures a fair and efficient allocation of seats in institutes. This process aims to provide students with the best possible seat allocation based on their choices and merits.

Overall Architecture Diagram



5. Description for each Module

Institution Module



Input:

The user is prompted to enter the number of institutions

For each institution, the user is prompted to enter the institution code, institution name, department name, and number of seats.

Process:

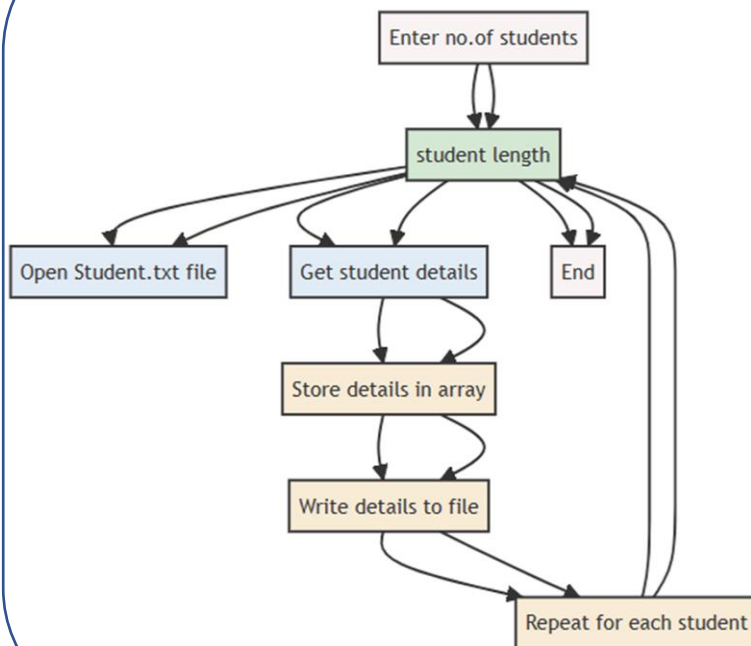
The institution function reads the user inputs and stores them in the i1 array of structures.

It writes the institution details to the "allotcheck_Institutions.txt" file.

Output:

If the file "allotcheck_Institutions.txt" is created successfully, a success message is printed.

Student Module



Input:

The user is prompted to enter the number of students.

Process:

The program prompts the user to input the details for each student, including roll number, first name, second name, mains rank, and advanced rank (if applicable).

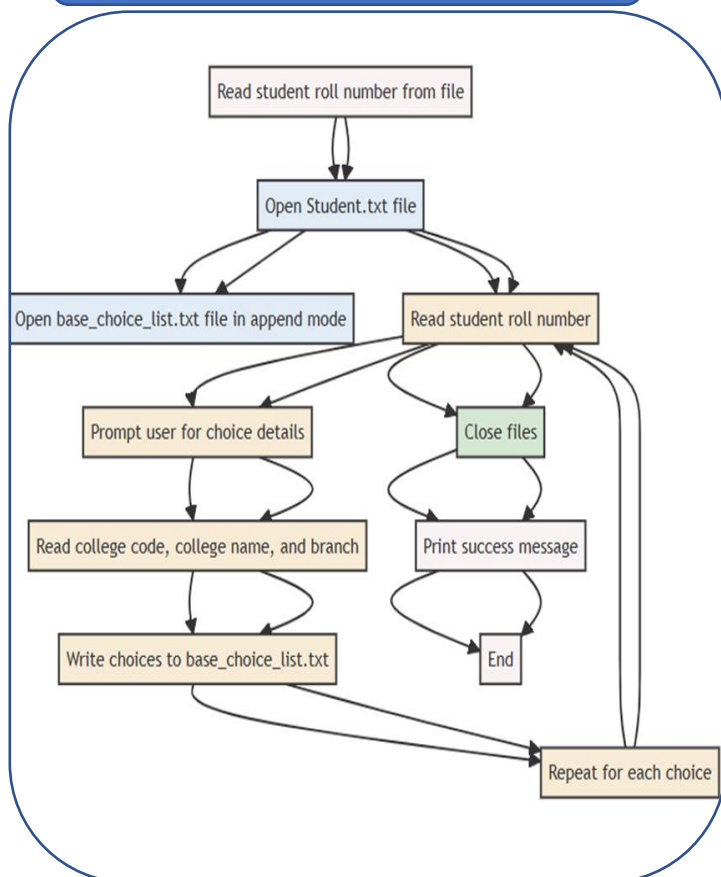
The program creates an array of students structure objects to store the student details.

The entered student details are written to the "Student.txt" file.

Output:

The program stores the student details entered by the user in the "Student.txt" file.

Choice list importing module



Input:

The input is read from the "Student.txt" file, which contains student roll numbers.

Process:

The code reads the roll number from the file and prompts the user to enter the choices for each student. It reads the college code, college name, and branch for each choice using scanf.

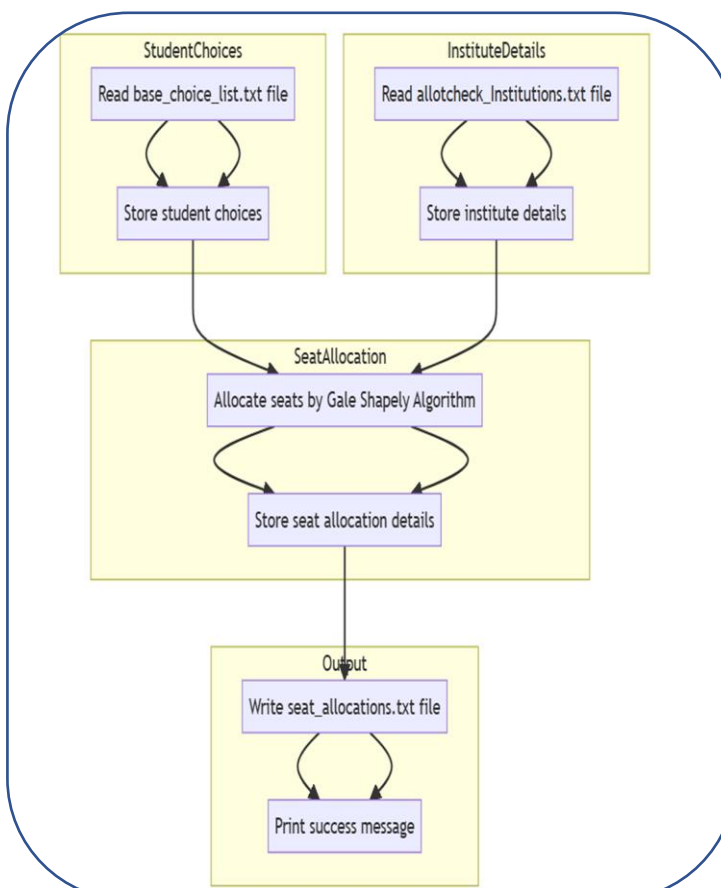
It calls the writeChoicesToFile function to write the choice details to the "base_choice_list.txt" file.

Output:

The output is a success message indicating that the choices have been written to the "base_choice_list.txt" file.

The final output is the success message: "Choices have been written to base_choice_list.txt."

Allocation module



Input:

The input is read from two files: "base_choice_list.txt" and "allotcheck_Institutions.txt".

"base_choice_list.txt" contains student choices, including roll numbers, ranks, college codes, college names, and branches.

"allotcheck_Institutions.txt" contains institute details, including codes, names, branches, and seat counts.

Process:

The code reads student choices and institute details from their respective files.

It allocates seats to students based on their choices and ranks.

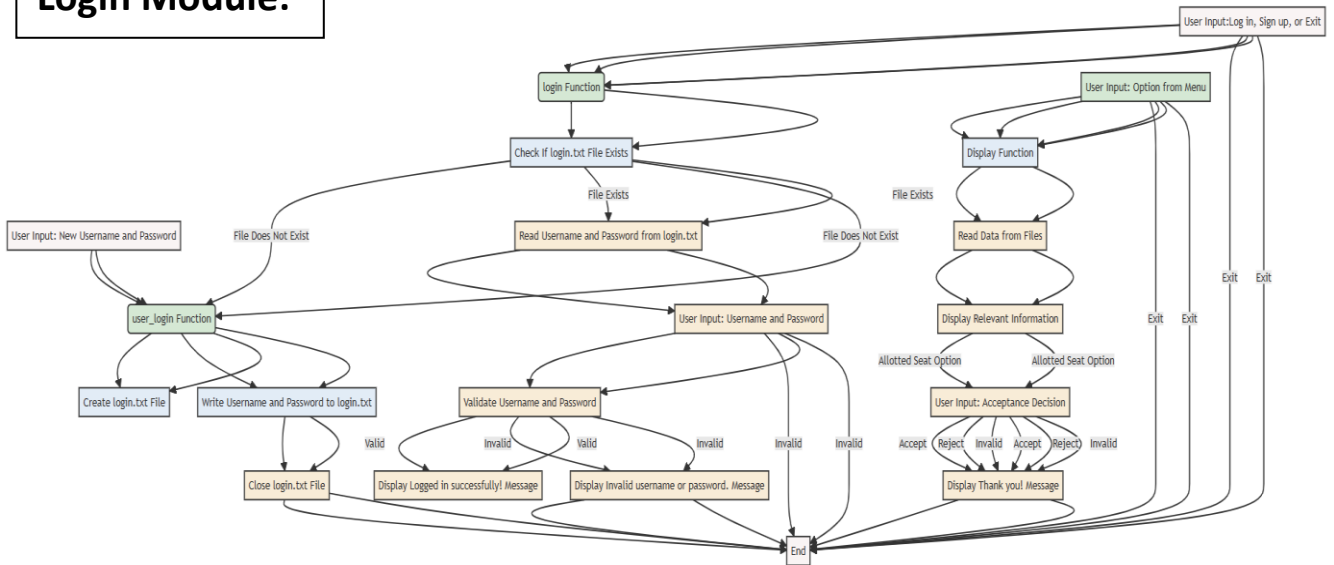
It writes the seat allocation details to the "seat_allocations.txt" file.

Output:

The output is the successful completion of the seat allocation process.

The seat allocation details are written to the "seat_allocations.txt" file.

Login Module:



Input:

New Username and Password

Roll Number

Acceptance Decision

Process:

user_login Function:

User input: New Username and Password

Create login.txt file

Write Username and Password to login.txt

login Function:

Check if login.txt file exists

If not, call user_login Function

If exists, prompt for Option

Read Username and Password from login.txt

Validate Username and Password

Return success or failure flag

display Function:

User input: Option

Read data from files

Display relevant information and Prompt for Acceptance Decision

Output:

Logged in message

Institutions: Display institution information

Ranklist : Display students ranklist

Your Rank: Display individual student rank

Allotted Seat: Display allocated seat

6. Implementation

6.1 We organized data using various language constructs such as arrays, structures ,array of structures and files.

6.1.1 Arrays:

Arrays are used to store multiple values of the same type, we used arrays to store student information, such as roll number, first name, second name, mains rank, and advanced rank.

For example:

```
struct students {  
    int roll_no;  
    char first_name[20];  
    char second_name[20];  
    int mains_rank;  
    int adv_rank;  
};
```

By using arrays, the code can efficiently store and access multiple student records in a structured manner.

6.1.2 Structures:

Structures are used to define custom data types that can hold multiple variables with different data types. We used structures to define the data structures for login credentials, institutions, departments, students, and choices.

For example:

```
struct login {  
    char username[50];  
    char password[9];  
};  
struct inst {  
    char code[5];  
    char name[20];  
    struct dept d[5];  
};
```

By using structures, the code can group related data together and create more meaningful and organized representations of complex entities.

6.1.3 Array of Structures:

Array of structures is used to store multiple instances of a structure. We used array of structures to store multiple student records and multiple choices.

For example:

```
struct students s1[count];  
  
struct choice choices[MAX_CHOICES];
```

By using array of structures, the code can handle and manage a collection of related data elements efficiently.

6.1.4 Files:

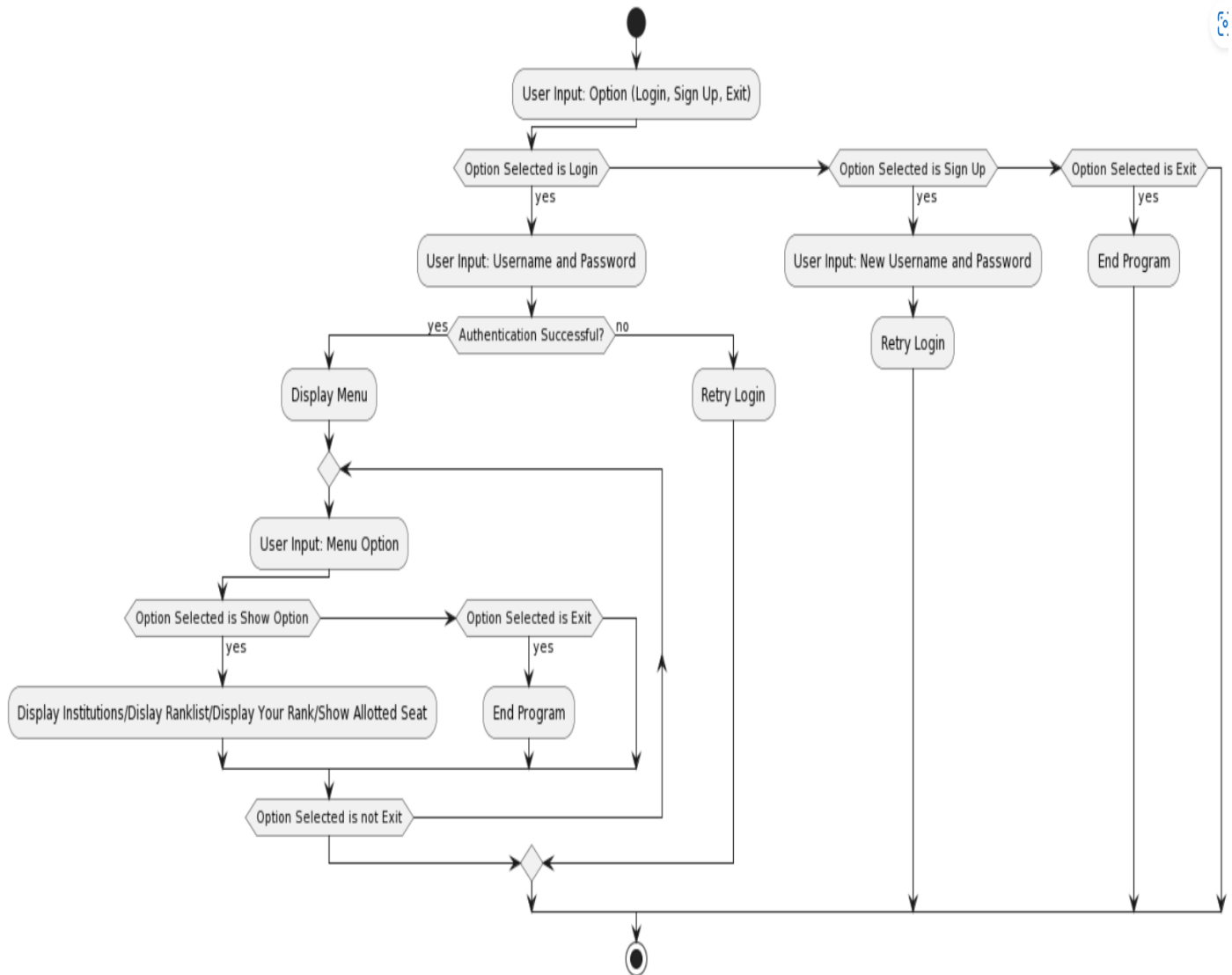
Files are used to read from and write to external files. We used files to store login credentials, student information, institution information, and seat allocations.

For example:

```
FILE *f1;  
  
f1 = fopen("login.txt", "w+");  
  
FILE *f2;  
  
f2 = fopen("Student.txt", "r");  
  
FILE *f3;  
  
f3 = fopen("allotcheck_Institutions.txt", "r");  
  
FILE *f4;  
  
f4 = fopen("allocations.txt", "r");
```

By utilizing files, the code can persist data beyond the runtime of the program and retrieve it when needed. This allows for data storage, retrieval, and manipulation even across different program executions.

6.2 User Interface Design:



6.3 Platform used for Code Development,

We used Replit, an online integrated development environment (IDE), for coding the project. Replit provided a user-friendly interface, collaborative tools, etc... It allowed remote collaboration, simplified code editing, execution, and debugging. The version control feature helped manage code changes. Overall, Replit enhanced productivity and facilitated the successful implementation of the seat allocation system

7. Validation through different test cases

```
1 1 Prashanna Kumar 51 101
2 2 Micheal Berdiananth 45 10
3 3 Divya S 13 25
4 4 Ganesh Hayagrivan 69 -1
5 5 Subasri G.S 85 42
6 6 Krishnaraj S 89 -1
7 7 Barath R 32 102
8 8 Anil velaa 54 50
9 9 Jayaprakash K 48 97
10 10 Rajesh G 67 -1
11 11 Shalini S 95 180
12 12 Abhishek R 88 168
13 13 Divya R 79 185
14 14 Ajay S 65 120
15 15 Nithya K 73 140
16 16 Sanjay M 70 -1
17 17 Priya R 85 160
18 18 Arjun S 92 175
19 19 Nithin K 81 150
20 20 Rahul S 76 -1
```

Here's the student details provided with Roll number, name, mains rank and advanced rank.

If a student didn't approach for JEE advanced exam, his/her rank will be assigned with -1.

Students who wrote both JEE mains and advanced exam are only eligible for choosing both IIT institutes as well as NIT institutes.

On the other hand, student who wrote only mains exam are eligible for choosing only NIT institutes. Here's the sample given below.

```
1 Roll Number: 1 College Code: 1001 College Name: IITMadras Branch: cse
2 Roll Number: 1 College Code: 1002 College Name: IITBombay Branch: cse
3 Roll Number: 1 College Code: 2001 College Name: NITTrichy Branch: cse
4 Roll Number: 1 College Code: 2002 College Name: NITWarangal Branch: it
5 Roll Number: 1 College Code: 1003 College Name: IITTirupathi Branch: cse
6 Roll Number: 2 College Code: 2001 College Name: NITTrichy Branch: mech
7 Roll Number: 2 College Code: 1001 College Name: IITMadras Branch: cse
8 Roll Number: 2 College Code: 1001 College Name: IITMadras Branch: eee
9 Roll Number: 2 College Code: 1002 College Name: IITBombay Branch: cse
10 Roll Number: 2 College Code: 2001 College Name: NITTrichy Branch: ece
11 Roll Number: 3 College Code: 2001 College Name: NITTrichy Branch: mech
12 Roll Number: 3 College Code: 2001 College Name: NITTrichy Branch: cse
13 Roll Number: 3 College Code: 2002 College Name: NITWarangal Branch: cse
14 Roll Number: 3 College Code: 1002 College Name: IITBombay Branch: mech
15 Roll Number: 3 College Code: 1003 College Name: IITTirupathi Branch: it
16 Roll Number: 4 College Code: 2001 College Name: NITTrichy Branch: cse
17 Roll Number: 4 College Code: 2002 College Name: NITWarangal Branch: mech
18 Roll Number: 4 College Code: 2003 College Name: NITKurukshetra Branch: it
19 Roll Number: 4 College Code: 2001 College Name: NITTrichy Branch: cse
20 Roll Number: 4 College Code: 2002 College Name: NITWarangal Branch: cse
```

21 Roll Number: 5 College Code: 1001 College Name: IITMadras Branch: ece
22 Roll Number: 5 College Code: 1002 College Name: IITBombay Branch: ece
23 Roll Number: 5 College Code: 2002 College Name: NITWarangal Branch: cse
24 Roll Number: 5 College Code: 2002 College Name: NITWarangal Branch: it
25 Roll Number: 5 College Code: 1003 College Name: IITTirupathi Branch: cse
26 Roll Number: 6 College Code: 2001 College Name: NITTrichy Branch: cse
27 Roll Number: 6 College Code: 2002 College Name: NITWarangal Branch: cse
28 Roll Number: 6 College Code: 2001 College Name: NITTrichy Branch: ece
29 Roll Number: 6 College Code: 2002 College Name: NITWarangal Branch: it
30 Roll Number: 6 College Code: 2002 College Name: NITKurukshetra Branch: cse
31 Roll Number: 7 College Code: 1001 College Name: IITMadras Branch: cse
32 Roll Number: 7 College Code: 1002 College Name: IITBombay Branch: it
33 Roll Number: 7 College Code: 1002 College Name: IITBombay Branch: cse
34 Roll Number: 7 College Code: 1003 College Name: IITTirupathi Branch: it
35 Roll Number: 7 College Code: 1003 College Name: IITTirupathi Branch: cse
36 Roll Number: 8 College Code: 1001 College Name: IITMadras Branch: cse
37 Roll Number: 8 College Code: 1002 College Name: IITBombay Branch: it
38 Roll Number: 8 College Code: 2001 College Name: NITTrichy Branch: cse
39 Roll Number: 8 College Code: 2001 College Name: NITTrichy Branch: ece
40 Roll Number: 8 College Code: 2002 College Name: NITWarangal Branch: it

41 Roll Number: 9 College Code: 1001 College Name: IITMadras Branch: ece
42 Roll Number: 9 College Code: 1001 College Name: IITMadras Branch: cse
43 Roll Number: 9 College Code: 1001 College Name: IITMadras Branch: it
44 Roll Number: 9 College Code: 1002 College Name: IITBombay Branch: ece
45 Roll Number: 9 College Code: 2002 College Name: NITWarangal Branch: ece
46 Roll Number: 10 College Code: 2001 College Name: NITTrichy Branch: cse
47 Roll Number: 10 College Code: 2001 College Name: NITTrichy Branch: ece
48 Roll Number: 10 College Code: 2002 College Name: NITWarangal Branch: it
49 Roll Number: 10 College Code: 2003 College Name: NITKurukshetra Branch: cse
50 Roll Number: 10 College Code: 2002 College Name: NITWarangal Branch: cse
51 Roll Number: 11 College Code: 1001 College Name: IITMadras Branch: cse
52 Roll Number: 11 College Code: 1001 College Name: IITMadras Branch: mech
53 Roll Number: 11 College Code: 1002 College Name: IITBombay Branch: it
54 Roll Number: 11 College Code: 2001 College Name: NITTrichy Branch: cse
55 Roll Number: 11 College Code: 2002 College Name: NITWarangal Branch: cse
56 Roll Number: 12 College Code: 1001 College Name: IITMadras Branch: ece
57 Roll Number: 12 College Code: 1002 College Name: IITBombay Branch: cse
58 Roll Number: 12 College Code: 1002 College Name: IITBombay Branch: it
59 Roll Number: 12 College Code: 2001 College Name: NITTrichy Branch: cse
60 Roll Number: 12 College Code: 2002 College Name: NITWarangal Branch: cse

61 Roll Number: 13 College Code: 1001 College Name: IITMadras Branch: cse
62 Roll Number: 13 College Code: 2001 College Name: NITTrichy Branch: cse
63 Roll Number: 13 College Code: 2002 College Name: NITWarangal Branch: cse
64 Roll Number: 13 College Code: 1002 College Name: IITBombay Branch: mech
65 Roll Number: 13 College Code: 1003 College Name: IITTirupathi Branch: it
66 Roll Number: 14 College Code: 1001 College Name: IITMadras Branch: cse
67 Roll Number: 14 College Code: 1001 College Name: IITMadras Branch: mech
68 Roll Number: 14 College Code: 1002 College Name: IITBombay Branch: it
69 Roll Number: 14 College Code: 2001 College Name: NITTrichy Branch: cse
70 Roll Number: 14 College Code: 2002 College Name: NITWarangal Branch: cse
71 Roll Number: 15 College Code: 1001 College Name: IITMadras Branch: ece
72 Roll Number: 15 College Code: 1002 College Name: IITBombay Branch: ece
73 Roll Number: 15 College Code: 2002 College Name: NITWarangal Branch: cse
74 Roll Number: 15 College Code: 2002 College Name: NITWarangal Branch: it
75 Roll Number: 15 College Code: 1003 College Name: IITTirupathi Branch: cse
76 Roll Number: 16 College Code: 2001 College Name: NITTrichy Branch: ece
77 Roll Number: 16 College Code: 2002 College Name: NITWarangal Branch: cse
78 Roll Number: 16 College Code: 2003 College Name: NITKurukshetra Branch: cse
79 Roll Number: 16 College Code: 2002 College Name: NITWarangal Branch: it
80 Roll Number: 16 College Code: 2002 College Name: NITWarangal Branch: ece

81 Roll Number: 17 College Code: 1001 College Name: IITMadras Branch: cse
82 Roll Number: 17 College Code: 1001 College Name: IITMadras Branch: mech
83 Roll Number: 17 College Code: 1002 College Name: IITBombay Branch: it
84 Roll Number: 17 College Code: 2001 College Name: NITTrichy Branch: cse
85 Roll Number: 17 College Code: 2002 College Name: NITWarangal Branch: cse
86 Roll Number: 18 College Code: 1001 College Name: IITMadras Branch: ece
87 Roll Number: 18 College Code: 1002 College Name: IITBombay Branch: ece
88 Roll Number: 18 College Code: 2002 College Name: NITWarangal Branch: cse
89 Roll Number: 18 College Code: 2002 College Name: NITWarangal Branch: it
90 Roll Number: 18 College Code: 1003 College Name: IITTirupathi Branch: cse
91 Roll Number: 19 College Code: 1001 College Name: IITMadras Branch: cse
92 Roll Number: 19 College Code: 1002 College Name: IITBombay Branch: cse
93 Roll Number: 19 College Code: 2001 College Name: NITTrichy Branch: cse
94 Roll Number: 19 College Code: 2002 College Name: NITWarangal Branch: it
95 Roll Number: 19 College Code: 1003 College Name: IITTirupathi Branch: cse
96 Roll Number: 20 College Code: 2001 College Name: NITTrichy Branch: cse
97 Roll Number: 20 College Code: 2002 College Name: NITWarangal Branch: cse
98 Roll Number: 20 College Code: 2001 College Name: NITTrichy Branch: it
99 Roll Number: 20 College Code: 2002 College Name: NITWarangal Branch: it
100 Roll Number: 20 College Code: 2003 College Name: NITKurukshetra Branch: cse

1	1001 IITMadras	<p>Here's the list of IIT and NIT institutes. Each institute is subdivided into branches along with number of seats in each branch. Each institute is given by its unique code and college name.</p> <p>These are fetched and stored in structures for further use in the allocation process.</p>	25	1003 IITTirupathi
2	cse 3		26	cse 3
3	it 3		27	it 3
4	eee 3		28	eee 3
5	ece 3		29	ece 3
6	mech 1		30	mech 3
7	2001 NITTrichy		31	2003 NITKurukshetra
8	cse 3		32	cse 3
9	it 3		33	it 3
10	eee 3		34	eee 3
11	ece 3		35	ece 3
12	mech 2		36	mech 3
13	1002 IITBombay		37	1004 IITDelhi
14	cse 3		38	cse 3
15	it 3		39	it 3
16	eee 3		40	eee 3
17	ece 3		41	ece 3
18	mech 3		42	mech 3
19	2002 NITWarangal			
20	cse 3			
21	it 3			
22	eee 3			
23	ece 3			
24	mech 3			

1	Roll Number: 1 College Code: 1001 College Name: IITMadras Branch: cse
2	Roll Number: 3 College Code: 2001 College Name: NITTrichy Branch: mech
3	Roll Number: 10 College Code: 2001 College Name: NITTrichy Branch: cse
4	Roll Number: 5 College Code: 1001 College Name: IITMadras Branch: ece
5	Roll Number: 7 College Code: 1001 College Name: IITMadras Branch: cse
6	Roll Number: 8 College Code: 1001 College Name: IITMadras Branch: cse
7	Roll Number: 9 College Code: 1001 College Name: IITMadras Branch: ece
8	Roll Number: 14 College Code: 1001 College Name: IITMadras Branch: mech
9	Roll Number: 15 College Code: 1001 College Name: IITMadras Branch: ece
10	Roll Number: 13 College Code: 2001 College Name: NITTrichy Branch: cse
11	Roll Number: 16 College Code: 2001 College Name: NITTrichy Branch: ece
12	Roll Number: 17 College Code: 1001 College Name: IITMadras Branch: mech
13	Roll Number: 18 College Code: 1001 College Name: IITMadras Branch: ece
14	Roll Number: 19 College Code: 1002 College Name: IITBombay Branch: cse
15	Roll Number: 20 College Code: 2001 College Name: NITTrichy Branch: cse
16	Roll Number: 2 College Code: 2001 College Name: NITTrichy Branch: mech
17	Roll Number: 4 College Code: 2002 College Name: NITWarangal Branch: mech
18	Roll Number: 6 College Code: 2002 College Name: NITWarangal Branch: cse
19	Roll Number: 11 College Code: 1002 College Name: IITBombay Branch: it
20	Roll Number: 12 College Code: 1002 College Name: IITBombay Branch: cse

→The above file shows the allocation of seats for each student. In the above case those who haven't written advanced exam (-1 in our case) are only allotted to NIT colleges. But for student who requested for NIT colleges as their first choice are also allotted accordingly based on their choice priority. (See the cases for students with roll number 4,6,10,16 & 20).

→Choices were allotted at first to each student until there are seats left. If a student request to a particular college where there are no seats left, immediately it checks mains or advanced rank according to his/her choice and compares it with of already allocated students' rank correspondingly. If the currently fetched student's rank is higher than any one of the already allotted student's rank, it replaces him/her with the current student.

→The replaced student is sent again to the iteration so that his second choice is fetched and allotted accordingly as per the seats available. Then the above process continues until every student is allocated to a particular college.

→The above attached file contains the allocation of each student to a particular college after the allocation process is held out.

→Among 20 students, student with roll number 1,7,8,11,13,14,17 and 19 opted for IITMadras cse branch. But only student with roll number 1, 7 and 8 are allocated to that seat since their ranks are higher compared to others who requested for.

8. Limitations of the solution provided

Scalability: The current solution may not be scalable enough to handle a large number of students and institutions. If the number of students or institutions increases significantly, the code may become inefficient and slow. It may be necessary to optimize the code or consider a more robust data management approach.

Error Handling: The current code does not have robust error handling mechanisms. It assumes that the input data is valid and does not account for potential errors or invalid input. Adding proper error handling, input validation, and exception handling can enhance the reliability and stability of the solution.

Security: The current login system uses a simple text file to store usernames and passwords, which may not provide adequate security for sensitive user information. It is recommended to implement more secure methods, such as using encrypted databases or hashing algorithms, to protect user credentials.

Data Persistence: The solution currently uses text files to store and retrieve data. While this approach works for small-scale projects, it may not be efficient for larger datasets. Consider using a more efficient and structured data storage solution such as a database management system (DBMS) to improve data organization and retrieval.

User Interface: The current solution relies on command-line interactions for input and output. While this may be sufficient for basic functionality, a more user-friendly graphical user interface (GUI) can enhance the user experience and make the system more intuitive and accessible.

9. Observations from the Societal, Legal, Environmental and Ethical perspectives

Societal Perspective:

Accessibility: The project aims to provide a fair and efficient seat allocation system, which can contribute to equal opportunities for students in accessing educational institutions.

Legal Perspective:

Data Protection: When dealing with student data and personal information, it is important to comply with relevant data protection which is done by our login module that process information only after a successful login.

Environmental Perspective:

Paperless System: Implementing a digital seat allocation system reduces the need for physical paperwork and document printing, contributing to environmental sustainability by minimizing paper waste.

Ethical Perspective:

Impartiality: The seat allocation system should be designed and implemented in a way that treats all students fairly, regardless of their background, gender, ethnicity, or other personal characteristics.

Confidentiality: Handling student data with strict confidentiality is crucial to protect their privacy and prevent any misuse or unauthorized access to sensitive information.

10. Learning Outcomes:

- Data Structures: Developing a seat allocation system involves managing and organizing data efficiently. We learned about data structures like arrays, structures, arrays of structures to store and manipulate student records, institution details, and seat allocations.
- Algorithms and Logic: Developing an efficient seat allocation algorithm is critical to ensure fair and optimized seat allocation. We learned about various algorithms and applied logical thinking to design and implement an algorithm (Gale Shapely Algorithm) that considers factors such as student preferences, seat availability, and ranking criteria.
- File Handling: Working with files is essential for storing and retrieving data in the seat allocation system. We have learned so much about how to read and write data to files, handle file errors, and maintain data integrity.
- Error Handling and Debugging: Throughout the development process, We have encountered many errors and unexpected behaviours. We have learned how to identify, diagnose, and debug issues in the code, improving our troubleshooting and problem-solving skills.
- Collaboration and Teamwork: Worked on a team project provided an opportunity to enhance collaboration and teamwork skills. We learned how to communicate effectively, delegate tasks, coordinate efforts, and work together towards a common goal.

11. References:

→Byron Gottfried, Programming with C ,3rd Edition.