DESIGN REPORT

The modules and their respective roles in the engineering counseling and admission process:

1. Candidate Registration Module:
- Role: Admin
- Responsibilities:
  - Create a CSV file to store candidate registration details.
  - Validate and store candidate's personal details and exam ranks.
  - Check for duplicate registrations.

2. Preference Submission Module:
- Role: Student
- Responsibilities:
  - View the list of available institutes and branches.
  - Submit preferences by selecting institute and branch codes.
  - Validate and store the submitted preferences.

3. Seat Allocation Algorithm with Gale-Shapley Algorithm:
- Role: Admin
- Responsibilities:
  - Fetch the preferences submitted by candidates.
  - Determine candidate eligibility based on exam ranks and other criteria.
  - Perform the seat allocation algorithm using the Gale-Shapley algorithm.
  - Allocate seats to candidates based on their preferences and availability.
  - Update seat availability and admission status in the database.

4. Reporting/Display Allocation Module:
- Role: Admin
- Responsibilities:
  - Fetch necessary data from the database.
  - Generate reports or display allocation details.
  - Format and present the information in a clear manner.

In summary, the modules and their roles are as follows:
- Candidate Registration Module: Admin performs the module.
- Preference Submission Module: Students perform the module.
- Seat Allocation Algorithm with Gale-Shapley Algorithm: Admin performs the module.
- Reporting/Display Allocation Module: Admin performs the module.

These modules and their roles help facilitate the candidate registration, preference submission, seat allocation, and reporting aspects of the engineering counseling and admission process.

1. Candidate Registration Module (Admin):

1. Create a CSV file in accordance with the candidate to ensure the registration of details.
2. Validate the candidate's personal details (e.g., name, registration number, rank) to ensure they are complete and in the correct format. (Make sure the data is enriched and void of missing data)
3. Validate the exam ranks to ensure they are within the acceptable range.
4. Check if the candidate has already registered or if their information exists in the database. (Prevent entry errors such as re-entry or mis-entry)
5. If the candidate is new and needs to be added, generate a unique registration ID along with name and rank for them and append these data to the original pre-existing CSV data consisting of individuals mentioned.
6. Store the candidate's personal details and exam ranks generated along with the registration ID in the database.

2. Preference Submission Module (Student):

1. Fetch the list of available institutes and branches from the database.
2. Display the list of available institutes and branches to the candidate along with their corresponding codes.
3. Allow the candidate to submit their preferences by entering the institute and branch codes.
4. Validate the preferences to ensure they are within the available options.
5. Check if the candidate has already submitted their preferences.
6. If the preferences are new, store them in the database associated with the candidate's registration ID.

3. Seat Allocation Algorithm with Gale-Shapley Algorithm (Admin):

1. Fetch the list of preferences submitted by candidates.
2. Determine the eligibility of candidates based on their exam ranks and any other relevant criteria (e.g., eligibility to JEE ADV).
3. Sort candidates based on their ranks on both exams (JEE ADV,MAINS).
4. Initialize an empty set of allocated seats and an empty set of unmatched candidates.
5. Start with the highest-ranked candidate and their first preference.
6. Check if there are available seats in the preferred branch of the chosen institute.
7. If seats are available:
   a. Allocate a seat to the candidate and add it to the allocated seats set.
   b. Update the seat availability and mark the candidate as admitted.
   c. Continue to the next candidate.
8. If seats are not available:
   a. Add the candidate to the unmatched candidates set.

b. Move to the next preference of the candidate.
9. Repeat steps 6-8 until all candidates have either been allocated a seat or are unmatched.
10. For each unmatched candidate:
    a. Retrieve the institute and branch of their current preference.
    b. Find the candidate who currently holds the seat in that branch.
    c. Determine if the unmatched candidate has a higher rank than the current seat holder.
    d. If the unmatched candidate has a higher rank:
        i. Remove the current seat holder from the allocated seats set and mark them as unmatched.
        ii. Allocate the seat to the unmatched candidate and add it to the allocated seats set.
        iii. Update the seat availability and mark the unmatched candidate as admitted.
11. Repeat steps 10 until there are no further changes in seat allocations.
12. Update the admission status of each candidate in the database.

By incorporating the Gale-Shapley algorithm, the seat allocation process ensures stability, meaning that there are no pairs of candidates and institutes who would both prefer each other over their current match. This algorithm helps to optimize seat allocations and reduce the possibility of candidates being left unmatched despite their higher preferences.

4. Reporting/Display allocation Module (Admin):

1. Fetch the necessary data from the database, such as seat allocation information, candidate details, etc.
2. Generate an allocation page consisting of allocation details in an organized manner.
3. Format the reports to display the relevant information clearly.

ARCHITECTURAL DIAGRAM