Case Study: C-DAC Admission System

This document proposes a modified C-DAC admisson system. Primary focus of the project is to allocate seats to the students for various C-DAC courses at various C-DAC centers based on their C-CAT ranking. Major modification considers eligibility of students for a particular course before giving center/course preferences and limiting only one counciling (with 2 rounds of seat allocation).

The basic data is made available in CSV files. Fields of CSV files are described below.

1. degrees.txt
    a. Degrees (text)
2. courses.csv
    a. Id (int)
    b. Name (text)
    c. Fees (text)
    d. Section (text) -- which CCAT section exam should be appeared for the course A, B or C
3. Eligibilities.csv
    a. Course name (text)
    b. Degree (text) -- qualifying degree for the course
    c. Min percentage (decimal)
4. Centers.csv
    a. Center Id (text) -- used as username for login
    b. Center name (text)
    c. Address (text)
    d. Coordinator (text)
    e. Password (text) -- used as password for login
5. Capacities.csv
    a. Center id (text)
    b. Course name (text)
    c. Capacity (max seats)
    d. Filled capcity (seats allocated after center allocation)
6. Students.csv
    a. Form no (int) -- used as username for login
    b. Name (text) -- used as password for login
    c. Rank a section (-1 indicate student not appeared for the section)
    d. Rank b section (-1 indicate student not appeared for the section)
    e. Rank c section (-1 indicate student not appeared for the section)
    f. Degree (text)
    g. Pecentage (decimal) -- degree score
    h. Allocated preference (int)
    i. Allocated course name (text)
    j. Allocated center id (text)
    k. Payment (decimal)

    l. Reported to center (int)
    m. Prn (text) -- Will be generated at the end as centerid_coursename_srno

7. Preferences.csv
    a. Student form no
    b. Preference no (1, 2, 3, …, 10)
    c. Course name
    d. Center id

System has three types of users i.e. Admin, Student and Center coordinator. Each user has the following relevant functionalities.

1. Student
    a. Register new student (Append student data at the end of Students.csv)
    b. Sign in
    c. List courses (as per eligibility)
    d. List centers
    e. Give preferences (Allowed to give preference only if the student is eligible for that course. Then append preferences in Preferences.csv)
    f. See allocated center/course
    g. Update payment details
2. Admin
    a. Sign in (username & password: admin)
    b. List courses & eligibilities
    c. List centers & capacities
    d. List students
    e. Update student ranks
    f. Allocate centers (Round 1)
    g. Allocate centers (Round 2)
    h. List allocated students
    i. List paid students
    j. List reported (at center) students
    k. Generate PRN
    l. List admitted students (with PRN) for given center
3. Center coordinator
    a. Sign In
    b. List courses (of that center)
    c. List students (allocated to that center)
    d. Update reported status
    e. List admitted students (with PRN)

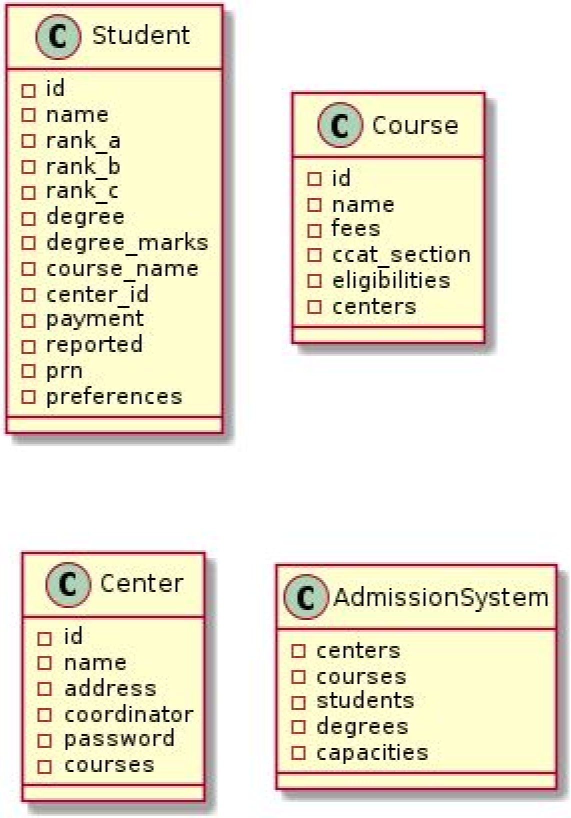
Case Study: C-DAC Admission System

Modified admission process (considered for this case study) is as follows:

1. Student registers for the C-DAC exam along with their academic details and exam category (A, B or C).
2. Students appear for exams of respective sections. The ranks (based on marks) are updated by C-DAC admin. If a student is not qualified/appeared in any section his/her rank is considered as -1.
3. As per valid rank and qualifying degree, students can give max 10 preferences for desired course at desired center as per eligibility criteria. The order of preference matters for center selection. (Data upto this step is already given in students.csv and preferences.csv file. For newly/registered added students (if any), data for these steps should be entered via this program in sequence).
4. C-DAC admin will allocate centers (Round-1) to the students as per their merit/rank (in respective sections) and their preferences. The allocated preference number, center and course should be updated (into students.csv). Also allocated seats should be updated (into capacities.csv).
5. If the center is allocated to the student, he/she should update payment of the first installment (Rs. 11,800/-) (into students.csv).
6. C-DAC admin will discard the seats allocated to unpaid students and allocate centers (Round-2) to the rest of students as per their merit/rank (in respective sections) and their preferences. The allocated preference number, center and course should be updated (into students.csv). Also allocated seats should be updated (into capacities.csv).
7. If the center is allocated to the student, he/she should update payment of the first installment (Rs. 11,800/-) (into students.csv).
8. All students allocated in Round-1 and Round-2 should also update the rest of fee payment (into students.csv).
9. On the first day of the course, the center coordinator will update the reporting (attendance) status of students (into students.csv).
10. Finally C-DAC admin will generate PRN number of all fully paid and reported students and update it (into students.csv).

Relevant data can be clubbed into structure/object/dictionary (depending on language choice). A suggested data in each object is given below. Data types should be chosen wisely depending on language used for implementation.

Case Study: C-DAC Admission System



Please consider the following points during implementing case study.

1. Implement the project as a menu driven console application. Depending on login, the corresponding menu should be visible to the user.
2. The whole data (from all files) must be loaded into the memory in appropriate data structures (collections). Entire processing will happen only in the memory and at the end (before exit) all files should be updated (as per changes done in memory).
3. Sample data is provided into CSV files, so that students need not to waste time for data entry. The data must be loaded from these files (as mentioned above).
4. Since the project focuses on the center/seat allocation algorithm, many menu items are skipped and that data is loaded from CSV files. As a future scope this data can be taken from the user (admin) like degrees, centers, courses, capacities and eligibilities. Also edit and delete features can be added for each of this data.
5. Students and mentors are encouraged to design and plan steps for implentation before coding begins.

Happy Programming!