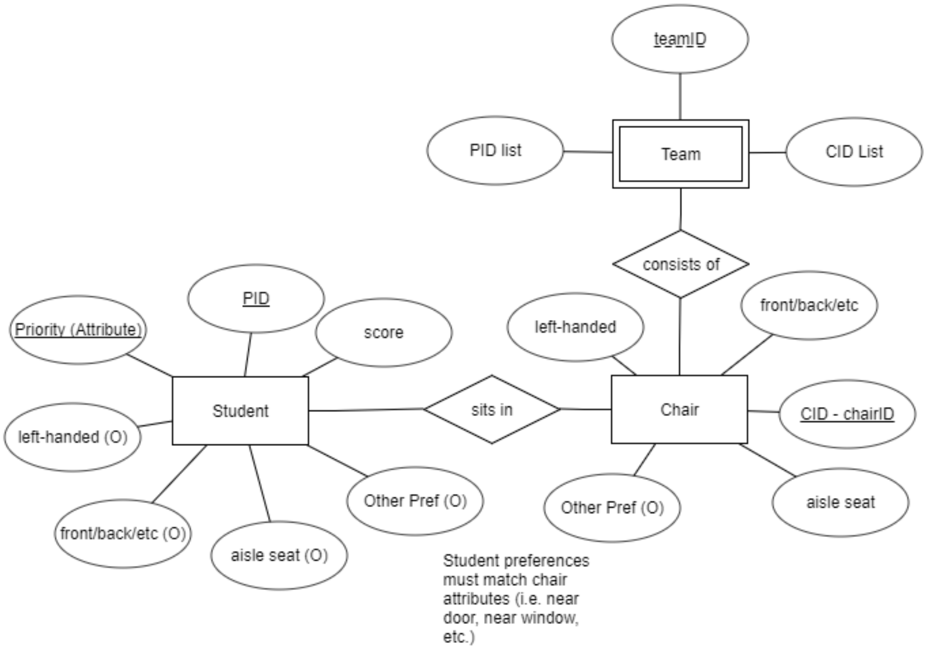
GROUPRE DESIGN DOCUMENT:

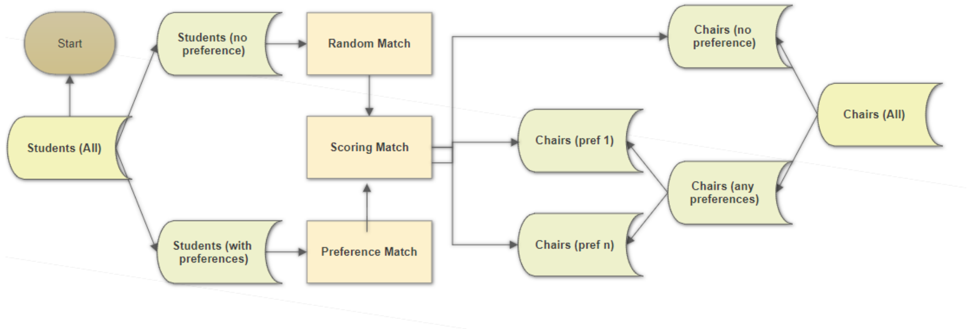
Architecture:

- Relationships – this diagram outlines the relationships between the various data fields; specifically the student, what team the student is on, and where the student is going to sit.

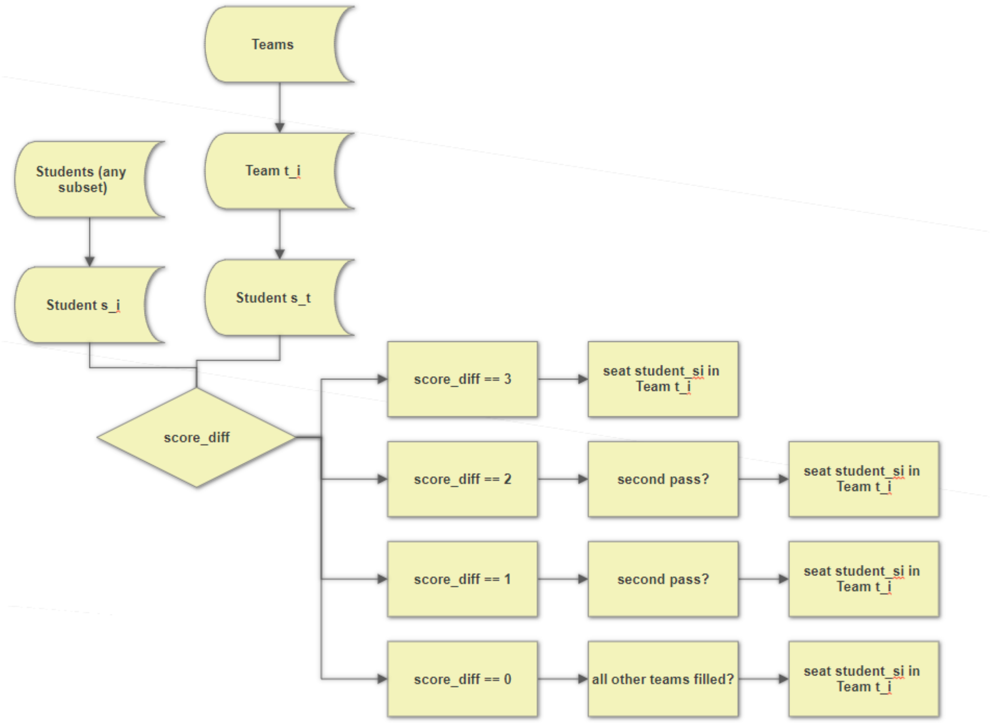


Student preferences must match chair attributes (i.e. near door, near window, etc.)

- Actual architecture – this diagram is a high level overview of the sorting algorithm, by preference, score, and availability of seating.



- Student Composition – this is a diagrammatic overview of the sorting mechanism as it relates to the pretest score administered by the instructor(s) on day 1.



Decomposition:

- Modules:
  1. Input – In this module, data is taken in in the form of a .csv file, along with basic user input. Data is parsed, stored in an array, and passed to the algorithm.
  2. Algorithm – Data which has been passed to the algorithm is analyzed, prioritized, and sorted based on established criteria (seat preference, priority of preference, seat availability, instructor requirements). Data is then sent to output.
  3. Output – Finally, data which has been input, analyzed, and sorted is sent to the output module which, in turn, displays relevant data to end-user in the form of a spreadsheet. This will eventually be displayed as an actual seating chart.
- Data:

1. Student name.
2. Student PID.
3. Seat number.
4. Handedness.
5. Aisle.
6. Team ID.
7. Pretest score.

Detailed Module Definitions:

1. Input Module – the module which specifically takes user input.
2. Algorithm Module – the module which performs analysis and sorting of data.
3. Output Module – the module which receives sorted data and provides that data to the end-user.

Detailed Data Definitions:

1. Student name – the student’s given name.
2. Student PID – the student’s PID.
3. Seat number – the specific seat number, in row/column format.
4. Handedness – whether the student is left handed or not.
5. Aisle – whether the student request/requires an aisle seat.
6. Team ID – the team number assigned to the student.
7. Pretest score – the score (from 1 to 4) the student achieved on the assigned pretest.

- Design Decisions:

- At this point, the decision has been made to code the program in Python and achieve basic offline functionality first, before proceeding to online integration. This integration will be built and tested on CloudApps. The program will functionally remain the same – in Python – but will be called or ‘wrapped’ in PHP in order to be web-capable.