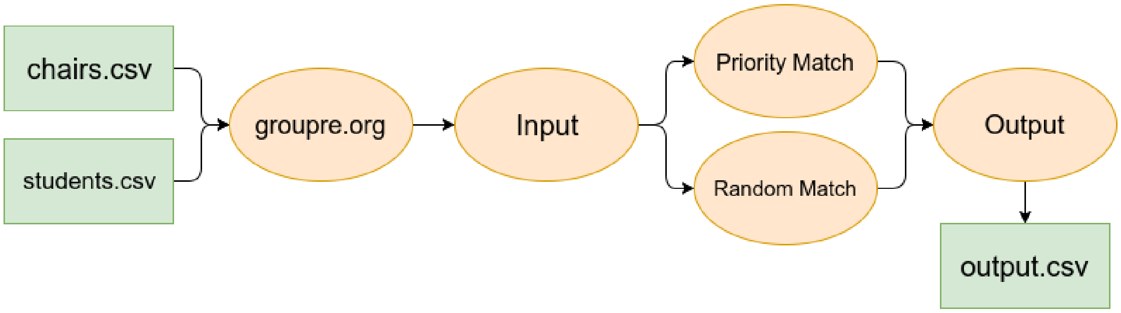
GROUPRE DESIGN DOCUMENT:

Architecture:

- Abstract Overview – this diagram outlines Groupre's basic flow. Inputs are received in the form of two .csv files – chairs and students – and passed to the web interface. From there they are matched according to the priorities laid out in the Functional Spec document before being output as a combined .csv.

Abstract Overview

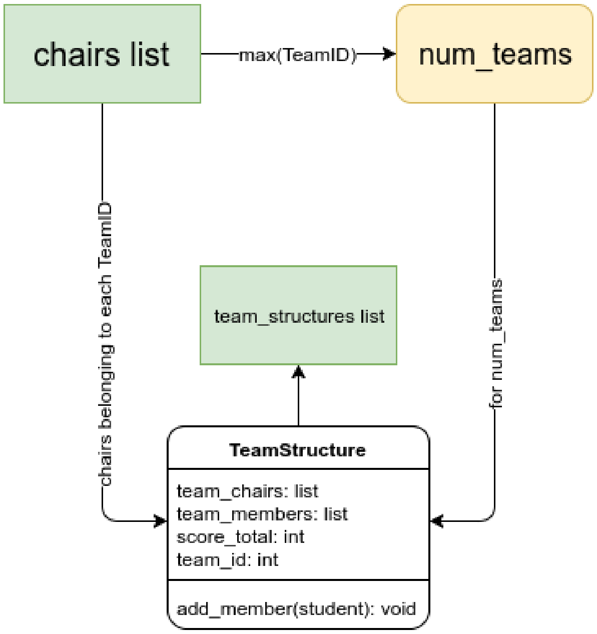
Groupre Web App (alt.)



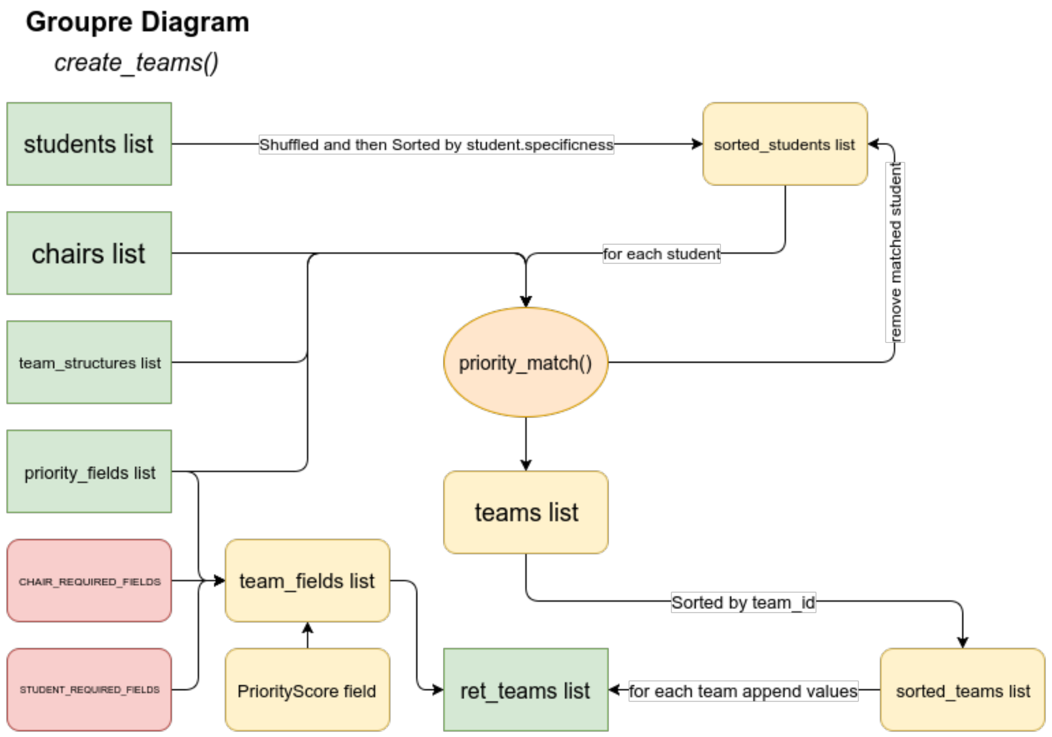
- Architectural Diagram – this diagram is a high level overview of the app as it pertains to the creation of the team structures. Chairs list is input, which sets the maximum number of teams.

Groupre Diagram

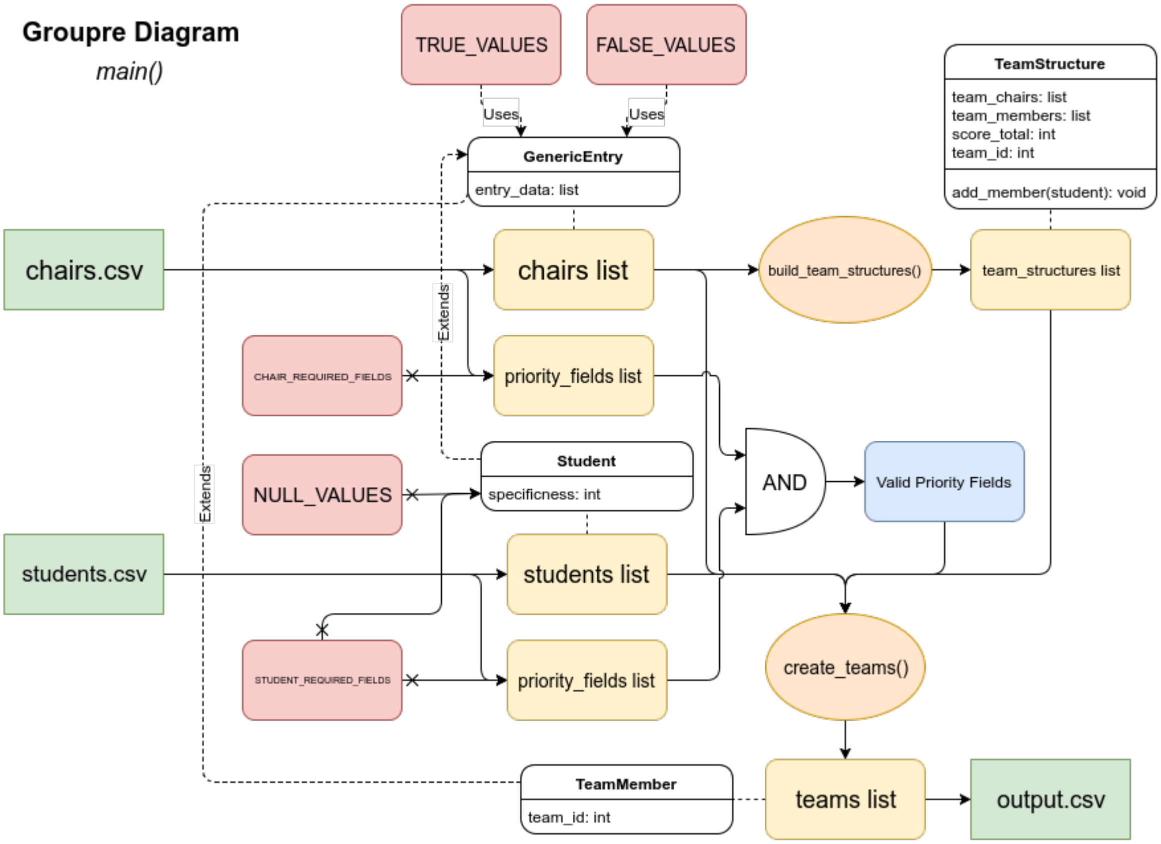
build_team_structures()



- Team Building Function – this function takes the input (in the form of the two lists; students and chairs) along with any specified team structures and priority fields, then sorts and matches the students into chairs. The priorities and teams are set by the end-user. After sorting, the sorted teams list is returned and output.



- Actual Groupre Function – this is a detailed diagrammatic overview of each step of the Groupre process.



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. Student creates student, chair creates chair, and the input module creates the relationship between the two of them.
  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.

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.

- 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 is built and tested on CloudApps, though concurrent offline development and testing continues. The program will functionally remain the same – in Python – but will be called or 'wrapped' in PHP in order to be web-capable. Flask will also be used in order to facilitate html to python compatibility.