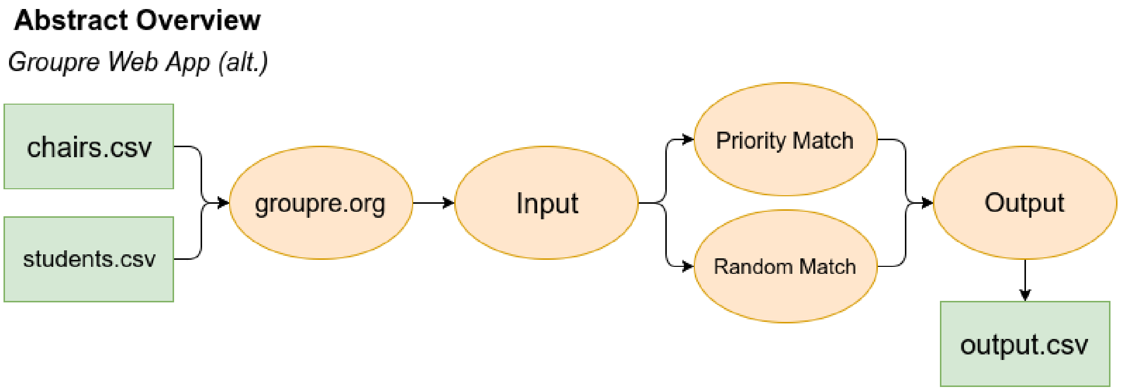
Sprint 6 5 December 2017

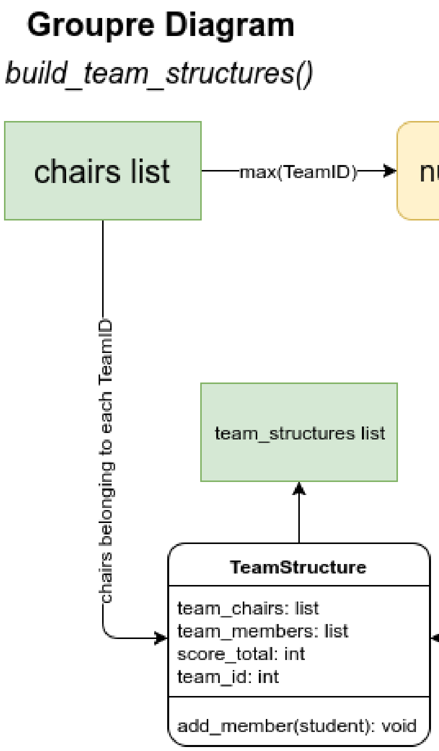
GROUPRE DESIGN DOCUMENT:

Architecture:

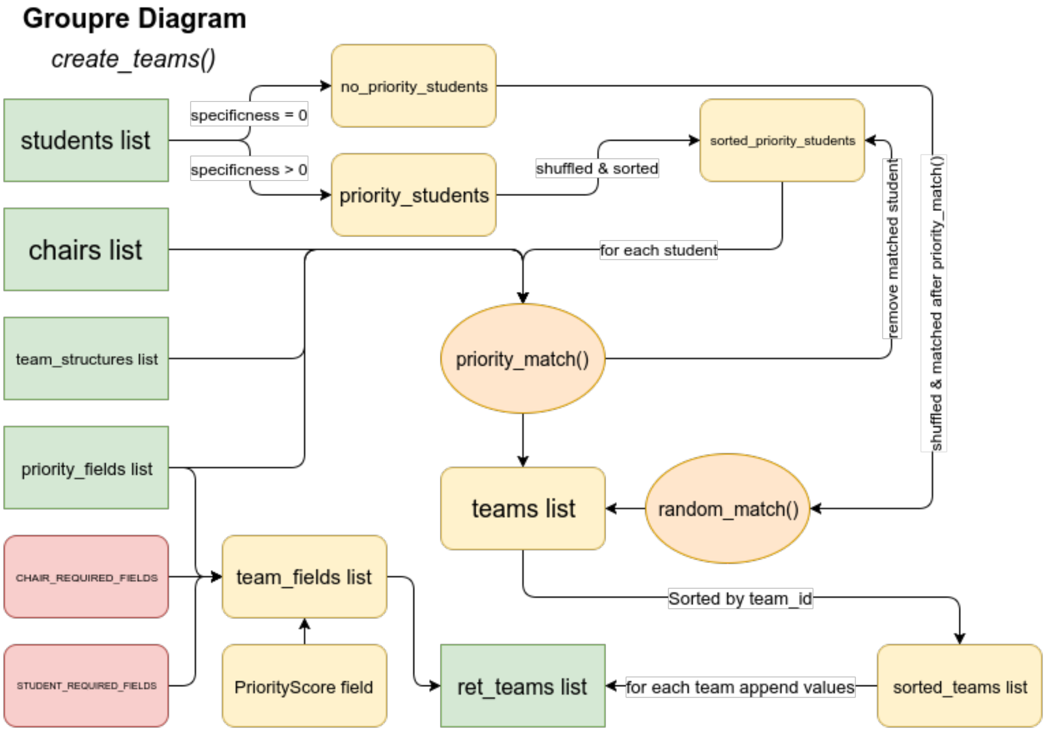
- Abstract Overview – this diagram outlines Groupre's basic flow. Inputs are received in the form of two .csv files – chairs and students – and passed to the web interface. From there they are matched according to the priorities laid out in the Functional Spec document before being output as a combined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



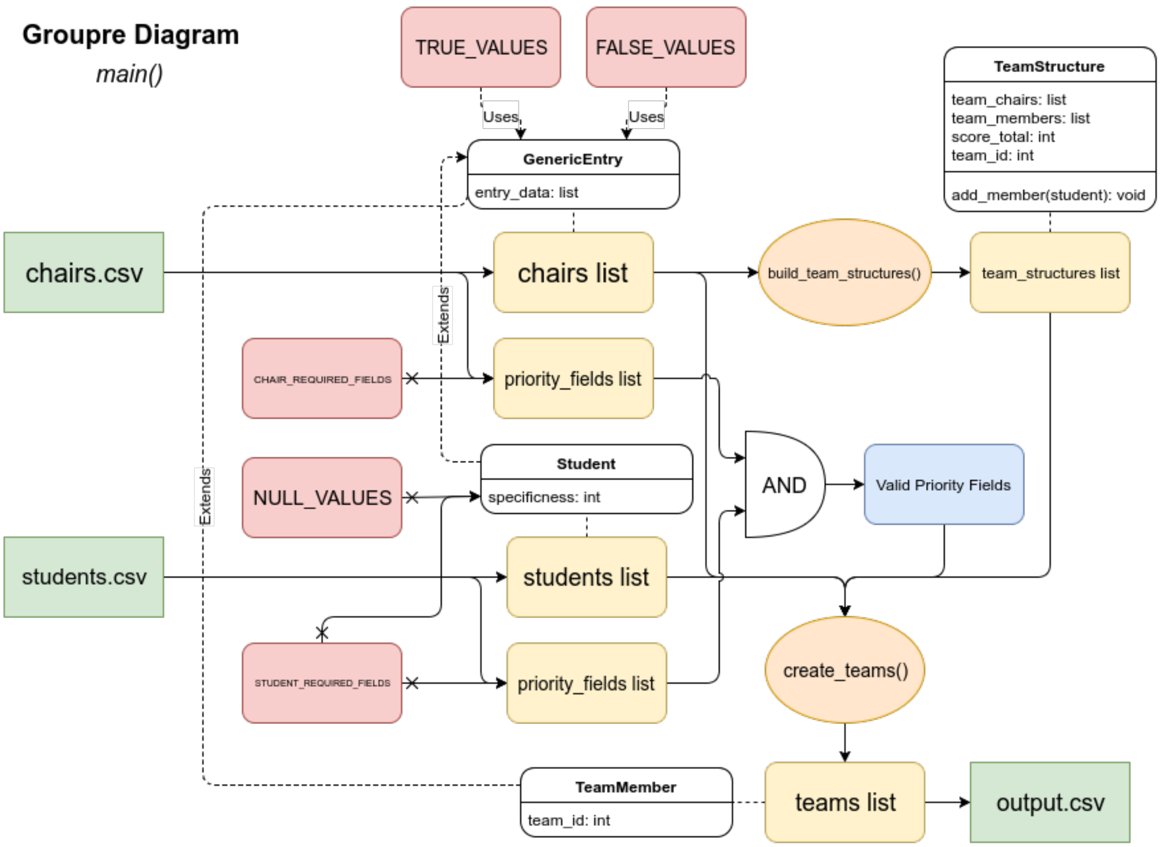
- Architectural Diagram – this diagram is a high level overview of the app as it pertains to the creation of the team structures. Chairs list is input, which sets the maximum number of teams.



- 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:
  - The back-end of our project is to be coded in Python, which will support basic offline functionality as a standalone module. This module is to then be integrated into a web-app deployed on Carolina CloudApps as a Python 3 application. As the back-end code is built upon, the web interface will be completed alongside it, ofering a more user-friendly visual tool for utilizing the groupre module. The web-app is to be designed and coded using a mixture of Python code in the Flask framework and JavaScript alongside basic HTML.
  - The GUI will be designed such that the user is able to dynamically construct the classroom in a gridlike fashion using standard keyboard-shortcuts for productivity. Their classroom design will be able to be stored for future use either by the user alone or by other users who wish to start with a template in the same room “Blueprint”. The user will be able to select and assign various rows and columns of Chairs to which they can assign attributes, which will be represented by colored icons. Afer the classroom is built, it will be used by the groupre

module as the Chairs input file. The output of the groupre module will eventually be interpreted such that it is displayed within the same blueprint that the user designed, so that they can make manual edits if necessary before downloading a finalized output file.