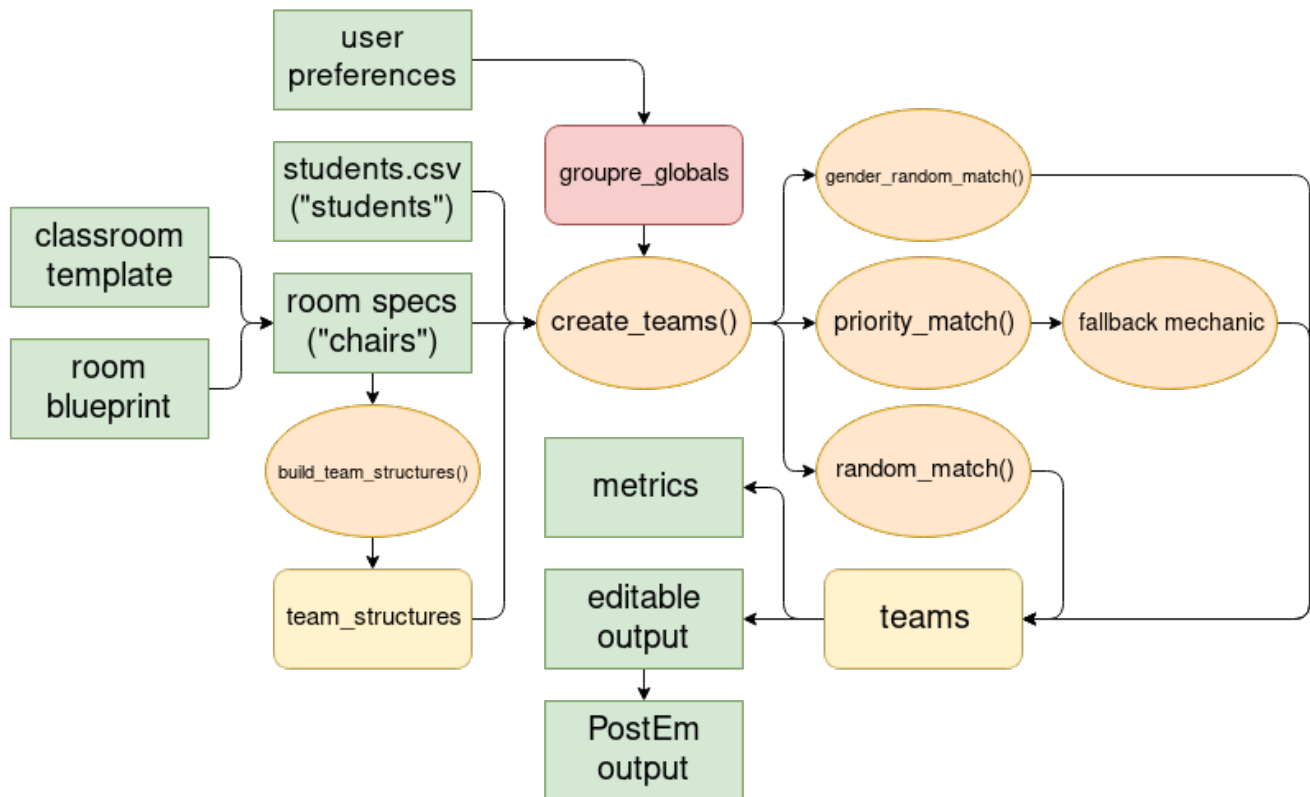# groupre Design Document

Architecture

Abstract Overview

The above diagram provides a rough overview of how the groupre module backe-nd and its web front-end interact for any general usage of the program. Green boxes denote input into or output from the program, orange ovals denote functionality within the groupre module, yellow curved-boxes denote some of the data structures used, and the red curved-box denotes a global module used by groupre.

Essentially, the user is expected to build a "room blueprint" if none exists, and then apply their own "room template" to that room with any specific additional criteria and their desired team layout. These two inputs are combined to form the "room specs" that will be used to match students to

teams. Students are input via a text-csv file format outlined in the User Manual. The user is also expected to have optional flags that they can enable, such as whether or not they want to enable the fallback mechanic, and then specify at what length the fallback should be allowed to go before not being applied.

This input is processed by groupre after being sent to the server from the web front-end, starting with some data-structure creation and then further processing through a number of matching algorithms depending on user input and the preferences that the students have.

The output of these matching algorithms forms the teams which are then sent back to the client so that they can view and/or modify the output before downloading a finalized output in the format of a Sakai PostEm file. A metrics file is also available depending on user input.

Decomposition

Modules

- Input

  - In this module, data is taken into the program as arguments, which denote options that the user has chosen as well as their specified input csv files that will be processed. This data is parsed and stored into data structures, which are maintained in lists. These internal data structures are then passed into groupre's sorting algorithms based on their attributes.

- Matching Algorithms

  - The matching algorithms used by groupre process the data collected by the input module, prioritize them based on their attributes, and sort them based on their attributes in comparison to possible target chairs as well as other students that are already within teams. The output of these matching algorithms is then sent back to the main module of groupre, which handles both input and output for final output file generation.

- Output

○   The results of the matching algorithms are taken and allocated into a text csv format, which is then returned either to the user in the case of command-line operation or to the web-app for download or displaying back to the user.

Design Decisions

The back-end of our project is 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, offering 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 grid-like 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. After 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.