

Brian Jahnke

Project Milestone

Abstract

The idea of this project comes from improving on the difficulties that are inherent in managing intramural sports at Marist through the site, Imleagues.com. The question I wish to solve for this project is to figure out how to make many of the processes that imleagues handles more automatic and efficient, thus providing a better experience for those participating in intramural sports at marist.

Introduction

As a person who has managed intramural sports at Marist for the past year, I've found that many difficulties come from the fact that many process are completely manual. As a result, some useful features found in imleagues are hardly taken advantage of because they require correspondence from multiple parties for them to be actually executed. This report shall outline a description of my proposed solution to these issues as well as DFA diagrams that illustrate my solution. Also included is the beginnings of my implementation of a representation of the Imleagues site.

Detailed System Description

The DFA will bring the user through the entire process of creating a team, building a roster, and entering that team in a league. Being a DFA, depending on what state the team is in the process, the program will inform the user what they must do in order to get to the next state. The DFA will cover certain things that the current imleagues site does not do automatically. For example, the DFA will not reach an accepting state if: the team does not have the requisite amount of players, the team does not have a unique name, the team applies for a league that is already full, a player on the roster is already on a team for that sport. Making these processes automatic should make the team building process much easier for students to get their teams in order before the deadline.

Requirements

Any modern computer with java installed on it.

Literature Survey

Imleagues.com is a site used by hundreds of colleges across the country, Marist included. Therefore, it will be the site I will be basing much of my project off of.

User Manual

Once the code starts running, the user will have to sign in to access their info. Certain commands will be available to them depending on whether they are a normal user or administrator. If the user's team is not in an accepted stated, the program will report to the users the steps it must take to do so.

Conclusion

So far, I have written the basic foundation of my project. I have outlined all of the major classes and most of their basic functions. I still need to add quite a few more functions, some sort of REPL/UI, and the DFA itself. Although the DFA for this project will likely be small, it should vastly improve the functionality of Imleagues and showcase the usefulness of a DFA no matter the size.