Team Matchmakers

PA School Matcher

Version 1.1

August 25, 2018

1. **Problem domain:**

With the rise of students applying to Physician Assistant school (PA School), comes a rise in software needed to assist students with getting into PA school. The cost of applying to one PA program, on average, is about $150. For this reason, the PA Brian Palm, requests that user friendly software be built to give students an idea of programs they are likely or unlikely to get into.

The PA program matcher will be an application, written in JavaScript, that will take in student’s academic information (Core GPA, Science GPA, GRE scores and shadowing hours/experience hours) and match them with PA programs they are likely to get into. The application will give a rating of Good, Average or Poor as far as chances a student has of getting into a PA program. The application will list out the schools in order of chances of getting in, with programs the applicant has a good chance of getting in at the top and programs the applicant has a poor chance of getting in at the bottom.

The application will also give an overall score for the student that is based on how many schools the student has a good chance of getting into. Optionally, the application will compare the student to other students who have taken the assessment in their area and students who have taken the assessment nationally. The application should give feedback on ways at which the student can improve their chances of getting into a PA school program, i.e. improve science GPA, or retake GRE. Optionally, the client would like for the application to have links to study material to help students improve whatever area is holding them back from being accepted into PA school.

A database is needed to store information on the schools as well as information of applicants that use this site. The database will likely be created using MySQL. Both the database and the application should have a way for the client to update information and reset student information for a new application cycle (delete student information after 3 or 4 months).

1. **Scenarios:**
   * Applicant uses the application to view their overall odds of getting into PA school
   * Applicant gets score that compares them to applicants in their area and nationally
   * Applicant gets information on how to improve themselves
   * Client updates the database on schools information
2. **Functional Requirements:**

**MUST**

* + The system must accept the Grade Point Average (GPA) and Graduate Record Examination (GRE) scores.
    1. RAT: Allows the user to enter their GPA and GRE scores into the website.
  + The system must compare student information to requirements of PA programs.
    1. RAT: Allows the user to see how he or she compares to requirements of PA programs.
  + The system must return PA schools with a score of Good, Average, or Poor based on the user’s chance of getting into it.
    1. RAT: Allows the user to see their chances of getting into specific PA schools.
  + The system must return a percentage of PA schools that the user received a Good score on.
    1. RAT: Allows the user to see a percentage of PA schools he or she has a good chance of getting into.
  + The system must JSON files to store school’s information in.
    1. RAT: Allows for a large amount of information to be stored and accessed at a later time.
  + The system and database must be able to be updated easily by the client.
    1. RAT: Allows the client to maintain the system and JSON files whenever there are changes.

**COULD**

* + The system could accept the Healthcare Experience (HCE) hours.
    1. RAT: Allows that user to enter their HCE hours into the website.
  + The system could accept Volunteer Hours.
    1. RAT: Allows the user to enter their Volunteer Hours into the website.
  + The system could compare the user’s information to average scores of those who have been accepted into PA schools.
    1. RAT: Allows the user to see how their scores compare to average scores.
  + They system could compare the user to other applicants nationally.
    1. RAT: Allows the user to see how he or she compares to other people applying to PA schools at a national scale.
  + The system could ask for the user’s zip code to show PA schools in the local area.
    1. RAT: Allows the user to see PA school he or she may be able to get into that are in the local area.
  + The system could ask for the user’s zip code to compare the user to other applicants in the local area.
    1. RAT: Allows the user to see how he or she compares to other people applying to PA school at a local scale.
  + The system could link the user to resources that would help improve specific scores.
    1. RAT: Allows the user to receive links to resources after viewing how his or her scores compare to the minimum requirements.

1. **Non-Functional Requirements:**
   * Results must be displayed in less than 3 seconds.
     1. RAT: Allows for quick assessment.
   * Downtime must be less than 3 hours a week.
     1. RAT: Allows for a reliable and more usable website.
   * The text fields for the user enter data must be displayed on one webpage.
     1. RAT: Allows for easy usability and avoids having to load multiple pages.
2. **Feasibility Concerns:**
   * Technology
     1. System is unable to correctly give a rating to the user.
   * Schedule
     1. There may not be enough time to fulfill all of the optional (COULDs) that the client would like to add to expand the system.
3. **Target Audience:**
   * PA school applicants
   * Future PA school applicants (Those wanting to know if they are on the right track to get into PA school)
   * Someone considering PA school as an approach to the medical field

1. **Deliverable and Deadlines:** 
   * **August 31, 2018** - Client Requirements
   * **Week of Sep 24th** - Requirements phase: team presentations of rationale for platform selection, feasibility and requirements report and 1st iteration plan.
   * **Week of Oct 24th** - Client checkpoint: Functional demo
   * **Week of Nov 26th** - Final product delivery and presentations
   * **Thursday Nov 29th, 2pm** - Demos at CREATE symposium