SOFTWARE REQUIREMENTS SPECIFICATION

Rainbow Boys

David Skocic

1.0 Introduction

1.1 Goals and Objectives

The primary goals of this software are to create a user-friendly and reliable way to insert and output data for the gymnasts in the rainbow program at Gymnastics World. The objectives include designing a secure and robust back end where data will not be lost and creating an intuitive front-end that is easy to learn and allows for quick access to all required functionalities.

1.2 Statement of Scope

This software is designed to manage the records of gymnasts and their progress throughout the Rainbow Program at Gymnastics World. It will handle inputs such as class information, such as class times and coaches, gymnast information, such as name and birthdate, and gymnast progress, such as which skills they have learned and when. This software will ensure this data is reliably stored and the user can easily enter and access all data.

1.3 Software Context

This software will operate independently on only the machine which it is installed and will only be handled by a small number of trained users. It will serve as the primary system of record keeping for the boys Rainbow Program, lending to more consistent record keeping and better organization among in the Rainbow Program boys at Gymnastics World. Eventually it may be extended to the girls' side of the Rainbow Program and will be hosted online so users may access data from anywhere.

1.4 Major Constraints

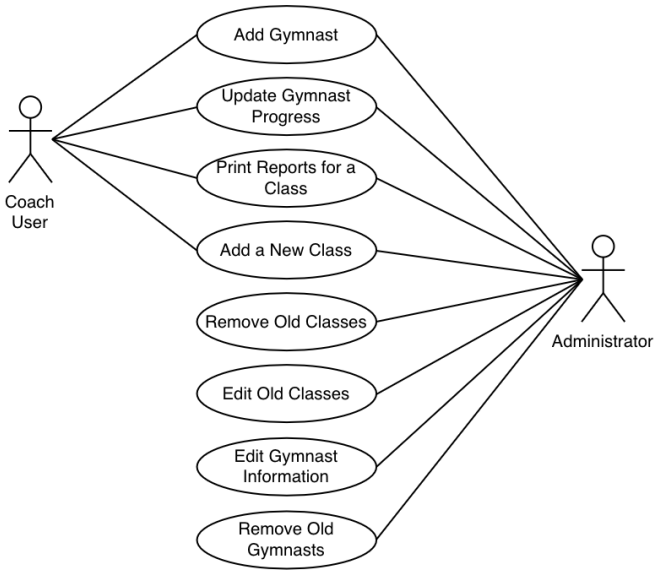
Various constraints will shape the creation of this software. These include the kinds of systems that might be running this software, so it must be portable to minimize compatibility issues. The system also must integrate with old data that is stored in a different database so all old data can be transferred to this new Rainbow Boys system. The time and resource limitations of only having a single team member may also impact the prioritization of features that are included in this software.

2.0 Usage Scenario

2.1 User Profiles

There are only two types of users in this system. There will be a small number of regular users who are coaches are trained on the software to do their typical data entry tasks. They will have permission to add gymnasts, track progress, create new classes, and create reports. There will also be administrator users, likely only one or two, who have all the permissions of coaches, along with the permission to remove an old gymnast or class or edit it because there is not often a need for this and it could lead to data loss.

2.2 Use-Cases



- Case 1: Adding a Gymnast
    - Admins and coaches may click a button to add a gymnast, fill in their data and press 'submit' to add a gymnast to the system.
- Case 2: Updating a Gymnast's Progress
    - Admins and coaches may select a gymnast to update their progress, then fill in checkboxes to represent learning a skill.
- Case 3: Printing Progress Reports
    - Admins and coaches may press print while selecting a group of gymnasts and a pdf will be generated of all the progress for each gymnast.
- Case 4: Adding a New Class
    - Admins and coaches click a button to add a new class where they can enter information about the class and click 'submit' to create it.

- Case 5: Removing Old Classes
    - When an admin enters their password, an option will be available to select a class and click remove to get rid of a class in the database.
- Case 6: Editing Existing Classes
    - When an admin enters their password, an option to change class information appears so the admin may change class names or times.
- Case 7: Editing Existing Gymnast Information
    - When an admin enters their password the option to edit gymnast information appears where they can change names and birthdates.
- Case 8: Removing an Old Gymnast
    - When an admin enters their password, a button to remove a gymnast will appear. They will be asked to confirm deletion, so no data is lost.

2.3 Special Usage Considerations

Due to the long-term use that this software should see, the system must be able to handle and efficiently query a large amount of data. Also, since the user should not need any software knowledge to use this product, it must be very user-friendly to accommodate anyone who may be trained to use it.

3.0 Data Model and Description

3.1 Data Description

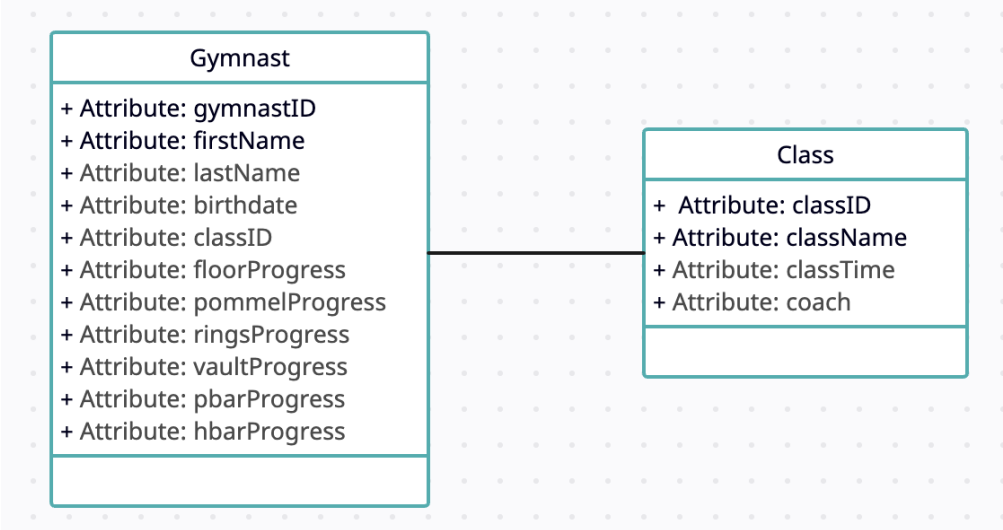
There will be two kinds of objects that will be managed by this software: Gymnasts and Classes. When the database is queried by the user, the code will create a gymnast object for every row of gymnast data that comes along with the query. Alongside this, there will be class objects that will simply hold the data about a class.

3.1.1 Data Objects

- Gymnast object
    - A gymnast object will have data from the database for a gymnast, it will pull together data from different tables to have the fields: gymnast ID, first and last name, birthdate, a reference to the class they belong to, and their progress on each of the six events.
- Class object
    - A class object will have data from the database about the class. It should only have the fields: class id, class name, class time, and coach.

3.1.2 Relationships

The following figure is the class diagram for this project. This could get expanded later to include a male and female gymnast, and to include coaches if more information is ever included with them



The Database Schema is in the following image, also subject to expansion when more features are added

3.1.3 Data Dictionary

- Gymnast
  - Attributes:
    - gymnastID: integer
      - unique identifier for each gymnast
    - firstName: String
      - The first name of the gymnast
    - lastName: String
      - The last name of the gymnast
    - birthdate: Date
      - The birthdate of the gymnast
    - Progress lists: Date[]
      - List of dates to track when each skill is learned
- Class
  - Attributes:
    - classID: integer
      - unique identifier for each class
    - className: String
      - A name for the class to give a user an idea of what group it is
    - classTime: Date
      - The time of the day that the class is
    - coach: String
      - The name of the coach that teaches the class

4.0 Functional Model and Description

4.1 Description of Major Functions

This section includes the requirements in order of priority for this software.

4.1.1 Progress Updating

The user will use textboxes and buttons from JavaFX to fill fields with data and click a 'submit' button to first update the data in the gymnast object then send this updated object to the database.

4.1.2 Class Creation

The user will enter the new class information to textboxes that clicks a 'submit' button implemented using JavaFX to create a new class object that is then inserted to the database.

4.1.3 Printing a Progress Report

The user can press a print button in the application that will export a selected group of gymnasts (a class) to a pdf file to print for a physical copy.

4.1.4 New Gymnast Addition

The user will enter the new gymnast information to textboxes then clicks a 'submit' button implemented using JavaFX to create a new class object that is then inserted to the database.

4.1.5 User-Friendliness

There is an old system that is currently used to track this information. Users of Rainbow Boys must report a 10 second time reduction per gymnast where they track their progress to ensure that it is easier to use than its predecessor

4.1.6 Simplicity

A new user should be able to be trained for 1 hour and be aware of how to use all the necessary functions to track student process.

4.1.7 Reliability

The old system that this one is replacing currently only stores data locally on the machine where the software is installed. This new system will have consistent (weekly or monthly) backups to the cloud to ensure data is never lost.

4.1.8 Efficiency

Querying and inserting data to and from the database will take 3 seconds or less so no time is wasted searching for data. This will be done using a PostgreSQL database, which should take care of questions about efficiency for this software.

4.2 Software Interface Description

4.2.1 External System Interfaces

The only external system interface is between the database and the application. This will be done using Java Database Connectivity (JDBC) to

execute queries. This is a complete library for accessing external databases using Java.

4.2.2 Human Interface

The user interface will be created using JavaFX so the database connectivity and the front end can all be done using the same language for easy connection. The user interface will have options for all the major functions of the program like adding classes and gymnasts, tracking progress, and printing reports. These will almost all have their own window that appears when an option is selected.

5.0 Restrictions, Limitations, Constraints

The main restriction of this project is the lack of resources to complete it. There is only a single developer that must work on the human interface, the database interface, and everything in between, so the attention to detail on any given section may not be as high as if there was a team for each. Another limitation of this project is that it must be portable because there are numerous systems which may host this software.