Design Document

CS 4503 Fall 2017

Team Psylocke

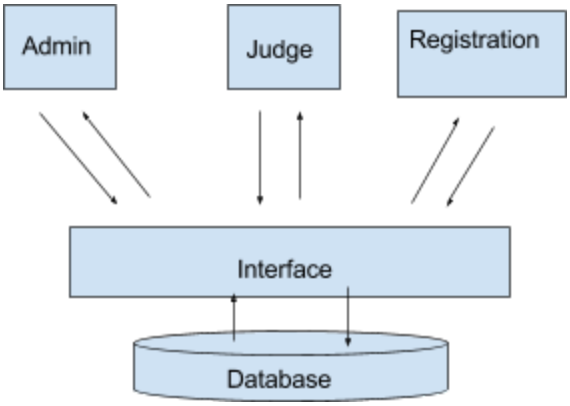
Anthony Mueller, Thomas Shaw, Alex Perdue, Jon Bolin, Michael Sun, John Talmage

1. Introduction

The purpose of this design is to demonstrate the intended high level architecture and entity relations for the Heartland Judging System webapp. This document will include architecture and entity relation diagrams showing the relationship between different parts of the system per specification in the SRS document, as well as the relationships between tables in the database. The audience of this document is software engineers and system architects who will be implementing and maintaining the described project.

2 Architecture

2.1 Introduction



The architectural design of the system will follow the repository model. There will be a central repository consisting of a MongoDB database to permanently store all information. A web app will interface with this database and allow for judging, registration, and administration. The web app will be written using Django.

2.2 Modules

2.2.1 Database

The database layer is responsible for holding all the persistent data for the system and defining the relationships between this data. The system will be using a MongoDB database. The data relationships are described in more detail below in section 2.2.3.

2.2.2 Interface

This module will provide to each subsystem the required access to the database as well as the visual components. This will outline the page layouts for the following interfaces: login, judging, admin, and registration

2.2.2.1 Login Page

The interface will include a shared login page, this page will consist of a username input box, a password input box as well as a button to allow for the scanning of QR codes. The user type will be determined and the appropriate user interface will be shown.

2.2.2.2 Judging Page

The judging interface opens up to multiple lists that provide various information about the entered projects. There will be a list of all categories, a list containing all participating entries, and a list of all the entries that have already been judged by the judge who is currently logged in. The bottom of the page will contain an evaluate button that will allow the judge to evaluate whichever entry they have selected.

2.2.2.3 Admin Interface

The admin interface will allow for exclusive management operations as well as the ability to access and manipulate all the entries in the database. There will be a button to generate a new login, consisting of a username, password, and QR code for any type of user. There will also be an edit team button, which will allow for the editing of existing entries/teams in the database. The admin will also get a button that will generate the scores for the final awards. This page also will have a button that allows the user to reset the database, there will be a confirmation message when the reset database button is pressed.

2.2.2.3 Registration Page

This page will include a list of all registered teams, and the ability to add new teams. Each team listed as registered will have a print button that allows for printing the QR code for each team. A delete button will allow the team to be deleted from the database. There will also be a register team button which will bring up an interface which asks for participants names, category, and the team name.

2.2.3 Subsystems

The architecture will contain three subsystems–registration, judging, and administration–which interact with the database.

2.2.3.1 Registration

The registration subsystem provides functionality to add teams, members, and judges to the central database. It will be responsible for generating the unique QR code that identifies teams and judges.

2.2.3.2 Judging

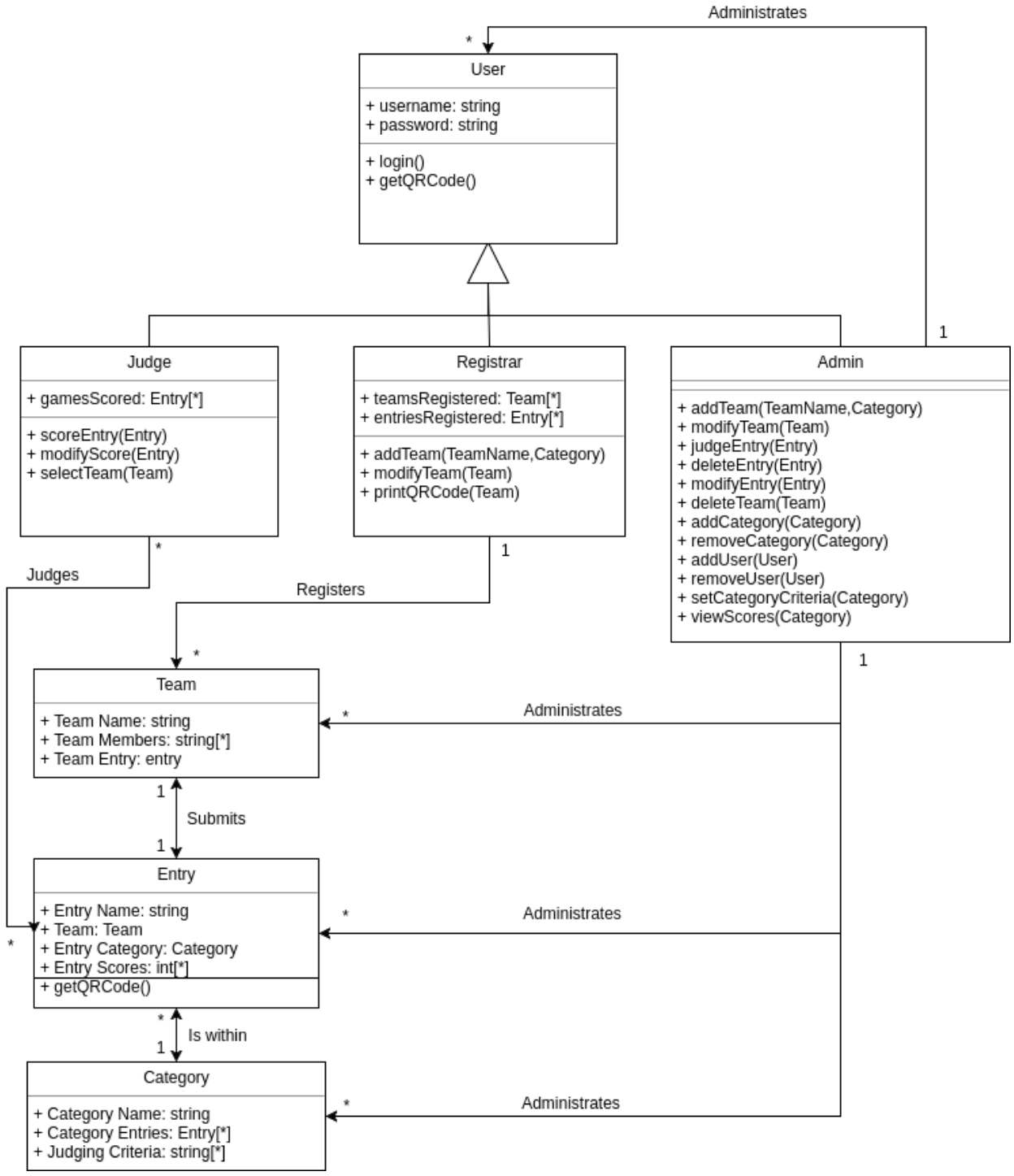
The judging subsystem provides functionality for judges to score the entries of the Heartland Gaming Expo. The module will provide information to the GUI for the users (judges). It will also be required to communicate judges' scores to the database so the changes can be permanently

stored. The system will also support updating of scoring information if judges go back and change scores. In future releases, the judging system will cache data locally in addition to uploading to the database.

2.2.3.3 Administration

The administrative subsystem will provide the necessary functionality for an admin-level user to modify and control the database information outside of the normal registration and judging process.

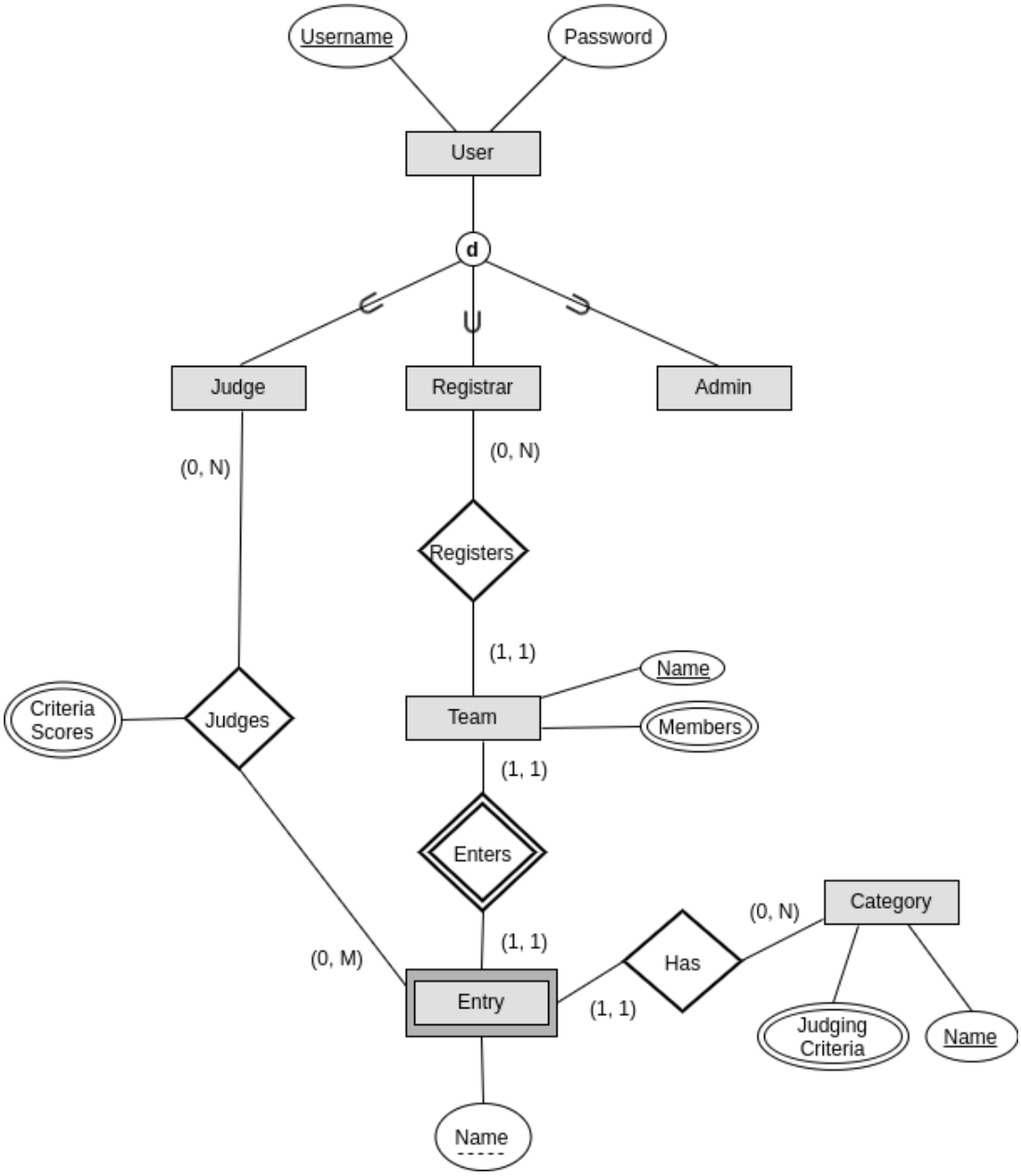
3 Class Diagrams



3.1 Database Class

We are using MongoDB as our database. MongoDB provides a Python API which will allow for interfacing with the database. The database is not shown in the diagram, but each User entity will interact with the database.

3.1.1 Schema



3.1.2 Schema Information

The data in the database shall be organized in the following tables and columns

3.1.2.1 User: Holds the data for a single user of the system including information such as their username and password, name, and whether they are a judge, registrar, or admin.

3.1.2.2 Judge: Inherits from user, includes games scored and ability to review scores.

3.1.2.3 Registrar: Inherits from user, includes teams and entries registered.

3.1.2.4 Admin: Inherits from user.

3.1.2.5 Team: Holds data for a single team in the competition, including team name, team members, and entry.

3.1.2.6 Entry: Holds data for single entry in the competition, including entry name, team that registered it, and the category.

3.1.2.7 Category: Holds data for a category of entries, including the entries in the category and the judging criteria.

3.2 User Class

The User class will be a superclass of Judge, Registrar, and Admin. It will contain a unique username as well as a password. Common functionality between each of the subclasses will be made available here, such as getQRCode and login.

3.3 Judge Class

The Judge class will track which games have been scored by this judge, and will contain functionality to submit scores for an entry, to modify scores previously submitted by this judge, and to select a previously-unjudged team for judgement.

3.4 Registrar Class

The Registrar class will track which teams it registers along with the entries for that team. It will create teams and entries and add them to the database.

3.5 Admin Class

The Admin class will have full reign over the database. It will be able to create and delete other Users, create Teams, Entries, and Categories, modify existing database entries, and view the aggregate final scores on a per-category basis.

3.6 Team Class

The Team class will contain information on the team's name, members, and entry by the team.

3.7 Entry Class

The Entry class will contain information on the entry's name, the team that entered it, and the category to which it belongs.

3.8 Category Class

The Category class will contain the category's name, what entries are in that category, and what criteria the category will be judged on.