Software Requirements Specification

Heartland Judging System

Psylocke

Version: (1.0) **Date: (9/24/2017)**

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to explain the requirements and features of the Heartland registration and judging system.

1.2 Scope

The Heartland Judging System is a webapp. It shall provide an interface for the judges of the Heartland Gaming Expo to score games based on their type, quality, and other criteria. It shall allow registrars to register teams, and allow an administrator to administer the competition. The system shall be accessible offline and able to sync with an online database. The goal of the project is to build a user-friendly interface that is able to interact with the database while at the same type provide a stable environment to both online and offline.

1.3 Definitions, Acronyms, and Abbreviations.

Backend - Computational logic side of the program

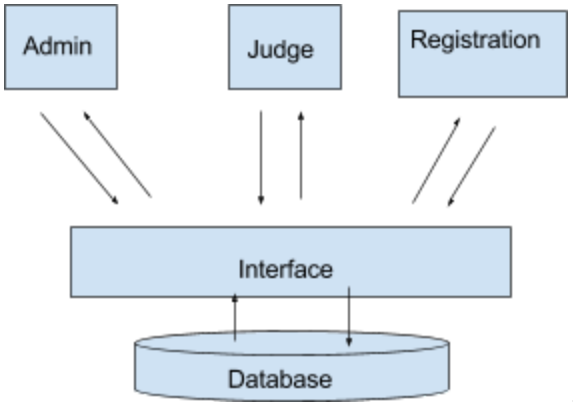
Frontend - The part of the program that the user interacts with directly

1.4 Overview

The rest of the this software requirements document will cover the overall description and specific requirement for the system. The overall description will provide a general overview of the system and interfaces for the system. The specific requirements will provide a more defined design and structure for the interfaces and how they will interact with each other.

2. The Overall Description

2.1 Product Perspective



2.1.1 System Interfaces

The Heartland judging system frontend application will interact with a backend system that manages business logic and interactions with the database.

2.1.2 Interfaces

The interface shall be a GUI webapp that is used to interact with the administration, registration and judging systems.

2.1.3 Hardware Interfaces

The judging portion of the system should be able to run on low-end Android tablets. The registrar and administration interfaces should be accessible from a computer.

2.1.4 Software Interfaces

There are no specific legacy software interfaces for this application.

2.1.5 Communications Interfaces

All systems shall use standard network interfaces, with tablets being able to directly communicate if manually synchronized to the database.

2.1.6 Memory Constraints

This application should be able to store information on local memory.

2.1.7 Operations

The front-end application shall have three modes of operation:

- Admin mode, for editing the categories, generating logins, viewing results, and wiping the database
- Registrar mode, for registering teams and printing team name placards
- Judging mode, for scoring teams.

2.1.8 Site Adaptation Requirements

There are no adaptation for any installations.

2.2 Product Functions

The Heartland judging app will provide an interface for an administrator to lay out the categories being judged and generate logins for judges and registrars. The registrars will be able to register teams into categories, and generate name placards for the teams, which include the team name and a QR code that can be used by the application to identify the team and category. Judges will be able to log into the application with their QR code, and judge a subset of the projects available. Their scores for these projects will be sent back to the database. The administrator shall be able to compile the judges' scores and adjust weights in category scores to determine winners.

2.3 User Characteristics

The registration portal shall be used by volunteers, who have little technical experience. The interface should be as easy to use as possible. The judging interface shall be used by volunteers from gaming industry and professors. This section should be made as intuitive as possible. The administrator shall be a knowledgeable user, allowing them to have direct access to advanced database manipulation features.

2.4 Constraints

The system shall be designed to work on laptops and low-end Android tablets provided to judges. Due to spotty internet connections at the event, the application should also be able to locally store data for the possibility of a dropped connection and later synchronize that data with the database.

2.5 Assumptions and Dependencies

The APIs for the database system and interface applications used shall remain consistent for several years. This way, the system can be used in future years without significant maintenance.

3. Specific Requirements

3.1 External Interfaces

- 3.1.1 The Opening Interface
    - 3.1.1.1 the opening interface for the app shall include login command button along with the username and password textboxes, and a QR scanner command button.
        - 3.1.1.1.1 textbox_1:
            - **Name**: “username_textbox”
            - **Purpose**: takes the input for username
            - **Source of input**: keyboard input
            - **Valid range and accuracy**: Valid characters include letters, digits, underscores, and dots.
            - **Relationship to other inputs/outputs**: “username_textbox” along with “password_textbox” represent data stored in the database that is called when the login button is clicked.
        - 3.1.1.1.2 textbox_2:
            - **Name**: “password_textbox”
            - **Purpose**: takes the input for password
            - **Source of input**: keyboard input
            - **Valid range and accuracy**: not-applicable
            - Valid characters include letters, digits, underscores, and dots.
            - **Relationship to other inputs/outputs**: Along with “username_textbox” it represent data stored in the database that is called when the login button is clicked.
        - 3.1.1.1.3 commandButton_1:
            - **Name**: “login_button”
            - **Purpose**: logs the user into the system
            - **Source of input**: click/touch
            - **Valid range and accuracy**: not applicable

        - **Relationship to other inputs/outputs**: takes the input in the textboxes so it can be compared to the information inside the database
    - 3.1.1.1.4 commandButton_2:
        - **Name**: “qrScanner_button”
        - **Purpose**: logs the user into the system
        - **Source of input**: scanning from camera
        - **Valid range and accuracy**: QR code needs to be fully visible and the pixels need to be clear for the scanner to take the input
        - **Relationship to other inputs/outputs**: the QR code is compared to the QR code stored in the system.
- 3.1.2 Judging Interface
    - 3.1.2.1 The judging interface shall include multiple listboxes for listing games, categories, and games already judged by the user, as well as command buttons.
        - 3.1.2.1.1 listbox_1:
            - **Name:** “category_listbox”
            - **Purpose:** list of the categories of the games
            - **Source of input:** the list is connected to the database of the system and can be edited by admins.
            - **Valid range and accuracy:** not applicable
            - **Relationship to other inputs/outputs:** the list is connected to the database system, and it also connected to “name_listbox”
        - 3.1.2.1.2 listbox_2:
            - **Name:** “name_listbox”
            - **Purpose:** list containing names of participating games
            - **Source of input:** the list is connected to the database of the system and can be edited by admins.
            - **Valid range and accuracy:** not applicable
            - **Relationship to other inputs/outputs:** the list is connected to the database system, and it also connected to “category_listbox”
        - 3.1.2.1.3 listbox_3:
            - **Name:** “judged_listbox”
            - **Purpose:** list containing all the games judged by the user
            - **Source of input:** the list is connected to the database of the system and is changed whenever the user evaluation/reevaluates a new game.
            - **Valid range and accuracy:** only the games judged by the specific user
            - **Relationship to other inputs/outputs:** the list is connected to the database system and specifically to data the user made/changed.

        - 3.1.2.1.4 commandButton_1
            - **Name:** “evaluate_button”
            - **Purpose:** opens game evaluation window
            - **Source of input:** click/touch
            - **Valid range and accuracy:** not-applicable
            - **Relationship to other inputs/outputs:** opens evaluation window for the user
    - 3.1.2.2 Once a game has been chosen, this screen opens up showing information about the game and it includes evaluation system for evaluating the game or editing previous evaluation.
        - 3.1.2.2.1 commandButton_1
            - **Name:** “submit_button”
            - **Purpose:** submits current evaluation
            - **Source of input:** click/touch
            - **Valid range and accuracy:** not-applicable
            - **Relationship to other inputs/outputs:** stores all the information done on evaluation window in the database
        - 3.1.2.2.1 commandButton_2
            - **Name:** “cancel_button”
            - **Purpose:** exits the screen
            - **Source of input:** click/touch
            - **Valid range and accuracy:** not-applicable
            - **Relationship to other inputs/outputs:** -
        - 3.1.2.2.3 slider_1
            - **Name:** “rating_system”
            - **Purpose:** The slider is used to rate the game ranging from 1-10
            - **Source of input:** click/touch
            - **Valid range and accuracy:** -
            - **Relationship to other inputs/outputs:** -
- 3.1.3 Admin Interface
    - 3.1.3.1 The admin interface includes exclusive access to edit, generate logins, and do calculations.
        - 3.1.3.1.1 commandButton_1
            - **Name:** “generate_login”
            - **Purpose:** Generate login username,password, and QR code
            - **Source of input:** click/touch
            - **Valid range and accuracy:** -
            - **Relationship to other inputs/outputs:** The generated items are stored in the database
        - 3.1.3.1.2 commandButton_2
            - **Name:** “edit_team”
            - **Purpose:** Provides the admin with options to edit a team
            - **Source of input:** click/touch

- **Valid range and accuracy:** -
- **Relationship to other inputs/outputs:** Connected to the databases of all the teams and has the access to modify them.

■ 3.1.3.1.3 commandButton_3
- **Name:** “score_calculation”
- **Purpose:** Based on a scheme, the button calculates the scores for the teams and outputs a list of them.
- **Source of input:** click/touch
- **Valid range and accuracy:** -
- **Relationship to other inputs/outputs:** Has access to the databases team scores

■ 3.1.3.1.4 commandButton_4
- **Name:** “database_reset”
- **Purpose:** Reset database
- **Source of input:** click/touch
- **Valid range and accuracy:** -
- **Relationship to other inputs/outputs:** Resets the database. Password and other inputs might be required for this action.

- 3.1.4 Registration Interface
  - 3.1.4.1 Includes a list of all registered team, as well as the option to register a new one.
    - 3.1.4.1.1 listbox_1
      - **Name:** “registered_teams”
      - **Purpose:** Shows all registered teams
      - **Source of input:** click/touch
      - **Valid range and accuracy:** -
      - **Relationship to other inputs/outputs:** displays a list of all the registered teams from the database. A team can be accessed to print items or perform a deletion.
    - 3.1.4.1.2 commandButton_1
      - **Name:** “print”
      - **Purpose:** Prints the contestant’s name and QR code.
      - **Source of input:** click/touch
      - **Valid range and accuracy:** -
      - **Relationship to other inputs/outputs:** -
    - 3.1.4.1.3 commandButton_2
      - **Name:** “delete”
      - **Purpose:** Performs a deletion operation
      - **Source of input:** click/touch
      - **Valid range and accuracy:** -
      - **Relationship to other inputs/outputs:** Deletion operation modifies the current team as well as making changes in the

database. In the case of a deletion of the entire team, the whole team is removed from the database.

- 3.1.4.1.4 commandButton_3
  - **Name:** “register_team”
  - **Purpose:** Registers a new team.
  - **Source of input:** click/touch
  - **Valid range and accuracy:** Team name, individuals, and at least one category must be provided
  - **Relationship to other inputs/outputs:** All new information are added to the database

3.2 Functions

- 3.2.1 Phase 1
  - 3.2.1.1 Backend
    - 3.2.1.1.1 Adding contestants: The backend system shall accept the registration of a new team, which includes a contestant name and category, from a valid logged-in registrar or administrator. The team will be added to the database, and a QR code linked to the team will be generated.
    - 3.2.1.1.2 Removing contestants: The backend system shall accept requests to remove a provided team, as long as the request comes from a logged-in registrar or administrator.
    - 3.2.1.1.3 Accepting judge scores: The backend system shall accept a set of scores for a team provided those scores come from a logged-in judge with permissions to judge the team, or a logged-in administrator. The scores will be recorded to the database, perhaps overwriting previous scores from that judge.
    - 3.2.1.1.4 Calculating results: The backend system shall be able to calculate winners in given categories from the judge scores, and provided those winners to a administrator in a human-readable fashion when requested.
    - 3.2.1.1.5 Generate logins: When requested by an administrator, the system shall be able to create a number of usernames, passwords, and QR codes with appropriate permissions for registrars and judges.
    - 3.2.1.1.6 Purging database: The backend system shall be able to purge the database of all contestants, scores, and non-admin logins when requested by a logged-in administrator.
  - 3.2.1.2 Login Screen
    - 3.2.1.2.1 Login screen: When the application is started, the system shall present username and password fields for login, as well as an option to login using a QR code.
    - 3.2.1.2.2 If the credentials provided do not match known backend credentials, the system shall show the user an error message and

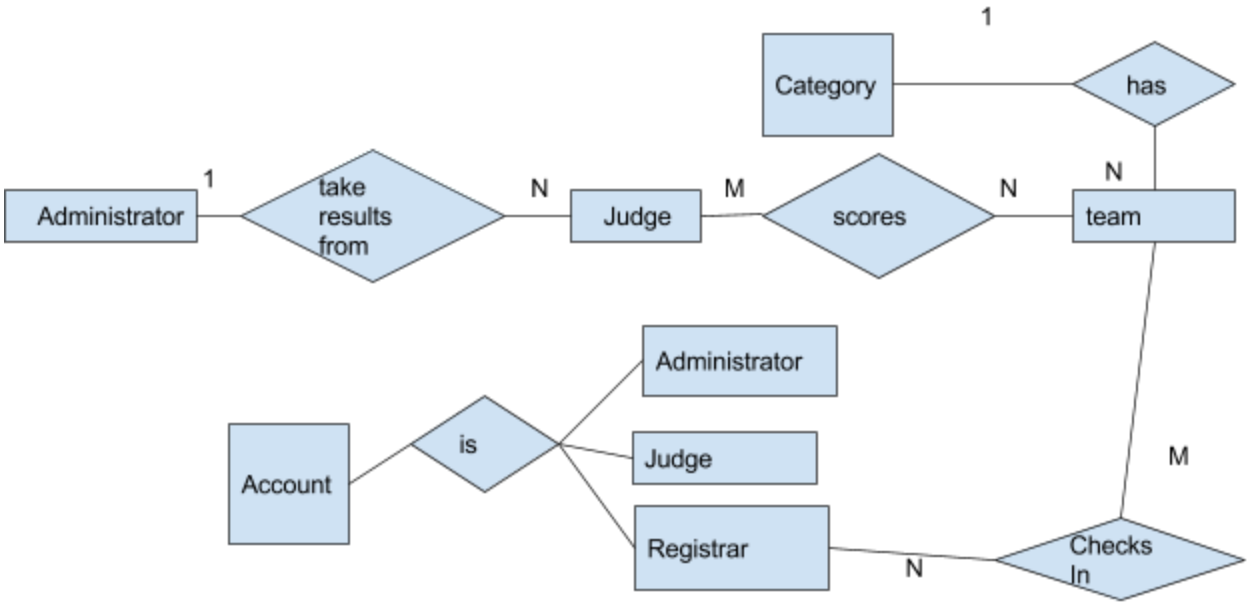
return to the login screen to allow the user to retry entering credentials.
        - 3.2.1.2.3 If the provided credentials match known credentials, the system shall redirect the user to the appropriate the role start page.
    - 3.2.1.2 Registration
        - 3.2.1.2.1 Contestant overview screen: The team overview screen shall display all currently registered contestants their currently registered category. It shall have a button to access the contestant registration screen.
            - 3.2.1.2.1.1 Print contestant placard button: Each contestant entry shall have a button that allows a registrar to print a placard that contains the contestant's name and an identifying QR code.
            - 3.2.1.2.1.2 Delete contestant button: Each contestant entry shall have a button that allows the registrar to delete the contestant's record from a database, in case of a mistake in registration.
        - 3.2.1.2.2 Contestant registration screen: This screen provides fields to create a new team name within a category to enter into the database, as well as a button to submit the provided information to the backend system.
    - 3.2.1.3 Judging
        - 3.2.1.3.1 Judge Main Menu: The judges main menu shall allow judges to search for teams manually or scan a QR code to access a Scoring Card for given team.
            - 3.2.1.3.1.1: QR Code Reader: Provide a button that brings up the tablet's camera to scan a team's QR code
            - 3.2.1.2.1.2 Contestant directory: Underneath the QR Code reader, the system will show a list of all currently known contestants, with an indication of what contestants have already been judged by this user.
            - 3.2.1.2.1.3 When a judge taps a team in the contestant directory or scans a QR code, the system will take the judge to the score entry page for the selected contestant.
            - 3.2.1.2.1.4 In the event of an improper QR code scan, the system should display an error message and take the judge back to the main menu.
        - 3.2.1.3.2 Score Entry: When a contestant is selected, the system will provide the judge with all scoring criteria for the team.
            - 3.2.1.3.2.1 Depending on what category the contestant is in, different judging criteria will be displayed, each with a slider going from 1 to 10.
            - 3.2.1.3.2.2 Submit button: At the bottom of the page will be a submit button that allows the judge to submit their scores for the contestant.

                - 3.2.1.3.2.3 Cancel button: At the bottom of the page will be a cancel button that allows the judge to cancel the scoring attempt.
        - 3.2.1.4 Administration
            - 3.2.1.4.1 Main Admin Screen: The main administrator screen should provide options to generate logins, reset the database, edit teams, edit scores, and calculate results.
- 3.2.2 Phase 2
    - 3.2.2.1 Caching scores in judging system
        - 3.2.2.1.1 The judging system should keep local copies of the judges score. In the event an acknowledgement of transmission was not received from the backend, the judging system should continue to attempt to send the scores until a confirmation message is received.
        - 3.2.2.1.2 The backend system should send an acknowledgement of receiving scores from a judge, and be resilient to potential losses of network connectivity.
    - 3.2.2.2 Registrars should only be able to remove contestants in a five-minute window after the contested is registered.
    - 3.2.2.3 Prioritizing judging order: The system should have a way to indicate in the judging system which teams do not have very many scores yet, and should be prioritized for judging.
    - 3.2.2.4 Calculating scores in alternative ways: The system should be able to calculate winners using alternative methods of scoring, such as normalizing each judge's scores, or throwing out high or low scores for each contestant
    - 3.2.2.5 Restricting Scoring Ranges: The system should not accept a score from a judge when the overall score is lower than the lowest secondary score, or higher than the highest secondary score. In the event a judge attempts to submit an out-of-bounds overall score, an error message should be displayed.
    - 3.2.2.6 QR codes printed at the registration step should be encoded with the category in addition to the team id, such that judging criteria for a contestant can be determined even without network access.

3.3 Performance Requirements

The application should be able to operate on a low-end tablet that could lose power at any moment. The application should also be able to support operation in an environment where internet access may be intermittent.

3.4 Logical Database Requirements



Judges shall be able to view all of their scores for all registrants they have judged. They shall be able to view all registrants that have registered since the last time the judge synchronized. Judges should be able to update their scores whenever logged in.

The administrator shall be able to add categories and teams. He/she shall have access to, and the ability to manipulate, all information.

A check-in account shall be able to add registrants to categories.

3.5 Design Constraints

The judging interface shall be clearly understandable on a tablet-size screen, and should efficiently make use of the resources of the android tablets to extend battery life.

3.6 Software System Attributes

3.6.1 Reliability

The judging interface shall allow judges to review submissions of project without connectivity. The tablets should be able to store judging scores and information in memory in case of battery failure.

3.6.2 Availability

The system shall allow judges to restart the application after failure with the loss of at most one entry.

3.6.3 Security

The system should maintain strict access control. The administrator shall have access to the whole system, Judges shall solely score teams and update their scores, and Registrators shall strictly have the ability to register and remove teams from categories. The system will be login and password protected, with the option of using a QR code to log in.

3.6.4 Maintainability

The backend database interface should be separated from the front-end interfaces to allow ease of updates to the user interface. For example, the backend code should not need to be changed to update the logo to a new year. In addition, categories and the judging criteria for those categories should be controlled by the database, thus allowing a change of categories or judging criteria without having to redeploy the application. Logins should also be managed through the system so no redeployment is needed for new logins.

3.6.5 Portability

The judging system shall be designed as a web-app, optimized for android. The registration and administration interfaces shall both be designed as web apps, meaning they should be accessible anywhere with internet access.