**Grand Valley Athletic Training Application Transfer Document**

This document is designed to help with the transference of the project to another team. In this document, the design decisions we’ve discussed with the sponsor will be explain in plain English for easy transferability. This includes ideas that have been implemented and are to be implemented. Be sure to also reference the PowerPoint presentations as they include more visual representations. It’s important to keep in mind that this game should be able to be used on any device through a web application.

**Overview**

The purpose of this application is help students of athletic trainers. It is designed like a Tamagotchi game, but it is an educational game. The user will have a group of athletes referred to as a team that will sustain injuries. It is up to the user to treat the athlete according to their training. Because this game requires expert knowledge, a secondary application is required to help the experts turn their information into a game for the user. To do this, an input system that is easy for experts to understand will be connected to a database that will connect to the user side application.

**Database**

The test database for this application will be included with this document among other files. This portion will explain the structure of each database and how the data was meant to be interpreted. Please also refer to the Entity Relationship Diagram.

* Game Data
  + This database is designed only to hold scenario IDs and IDs held by scenarios. The intent of this design is to help randomness. When an athlete gets a random injury, or the professor needs to send a new injury. This table is referenced whenever a new scenario id is needed.
  + The different fields of the database were determined by the sponsors. Determining the difference between a sprain, strain, dislocation, fracture and -itis is the main goal of treatment. Once those are determined, treatment can be suggested.
  + The database further differentiates these injuries by determining a location. The base locations decided on by the sponsors are leg, arm, torso, head, hand, and foot. Some of these locations may not be relevant to the kind of injury but that is done for flexibility for the sponsors.
* SUser
  + SUser stands for super user and is the login that the professors use to create scenarios. In their data, it contains the last time they logged in and user ID. It also contains a list of players that are considered in their class. The players are represented only by their player id.
* Players
  + Players are users that play the game. This data contains the players name, ID and the last time they played. This data also contains a list of athletes they have. The athletes are only represented by their athlete ID.
* Athletes
  + Athletes are the characters the players owns and performs treatments on. The athlete contains a lot of customization options for the athlete such as race, skin color, sex, skin color, shirt color and pants color. This data is used within a SVG to make a custom athlete.
  + There is last played and last injured on the athlete. Last injured is use to determine when the player should be randomly injured. If the athlete has not been injured in a long time, the would signal the athlete to receive an injury to continue the game loop.
  + The main part of the game comes from current scenario and last scenario. Last scenario is used to make sure there is not a repeat injury and that the athlete receives a different injury for the sake of variety. Current scenario determines what injury the athlete currently has. When the athlete is treated, it returns the athlete to zero.
* Scenario
  + Scenario is the most complicated data in the application. This contains information on the HOPRS method and how the game mechanics work. For now, this part of the document will only explain what each data piece represents. Later, the minigame will explain how the data is used in its entirety.
  + The first part of the data is the scenario ID, name, and review. Review gives a quick description of what the scenarios is. This is used more for the super users when reviewing a scenarios. This also keeps track of when it was last edited and when it was created.
  + The data keeps track of what category the scenario is in. This is represented inside game data and is used for forcing injuries on athletes.
  + SData within the scenario contains all the data within the scenario used by the minigames.
  + History is the first part of HOPRS and contains information known about the athlete or that the athlete would explain.
    - HStory contains a string of the story displayed to the user.
    - HKeyword contains an array of strings. These are words of interest to the user.
    - HReveal contains an array of strings. These are sentences revealed to the user.
    - HAnswer contains an array of Booleans. These determine if the point of interest was correct or not.
  + Observations are what the user would be able to see when inspecting the athlete:
    - OPoint are points on the athlete’s body determined by and X and Y coordinate. This is attached to a description of the observation or a literal picture of the observation.
    - After the user finds the observations, they must use their training to determine what these observations mean. The answers are contained with OAnswers.
  + Palpation is like observation, but it involves touch instead of sight. Because sight cannot detect everything, especially within the human body, touch is sometimes required to find hidden injuries.
    - PPoint are like OPoints in observation. This point is determined by and X and Y coordinate and shows a message to the user once the point is found. The point also has a max pain threshold to simulate the pain a spot being touched.
    - As the user finds the palpations, they will need to report them. The answers are stored in an array or words.
  + ROM or Range of Movement is the most complex mini-game as it requires custom animations. ROM will store information about what will happen when testing the athletes range of movement.
    - If the athlete is too injured or in too much pain, this test can be skipped. This field is determined by the Boolean ROMSkip.
    - If the ROMSkip is false then a ROM test is required. ROMTest will determine what test is the proper test. ROMLimit explain the limit of movement the athlete performs.
  + Special Test is a trial and error mini-game. The player must determine the proper special tests to do. Special test contains an array answers the player must determine.
  + Treatment is determined once all the other tests are performed. The player will have to determine what treatment the athlete needs.
* This data is included in the transfer folder provided.

**Minigames**

This application contains a collection of mini games that represents the HOPRS method. HOPRS is an anagram for History, Observation, Palpation, Range of Motion and Special Tests. Each of these methods will be given their own specific minigame. This section will explain the intended design of the minigames after much discussion with the sponsors. Make sure to also refer to the power point for design of the mini-games.

* History Minigame:
  + The user will be given a story that they will need to cipher which keywords are important to the story and what words are superfluous. The user will be provided with a paragraph with certain words underlined.
  + The user will only click or unclick the words that are relevant. Once the user believes they have found all the relevant information, they will click submit. For the sake of the beginner course, the user will be notified that they have made a mistake before being allowed to continue.
  + The data described in the previous section will be on JSON object. The first object will be the string containing the entire story. The keywords provided in an array should be used as a link or button that allows the use to click them. Once they are clicked, they are given a description of the word they selected on the side. This helps determine if they are on the right track.
* Observation Minigame
  + The user will look over the avatar with their touch. As they scan the athlete, points of interest will pop up. The user can then click these points of interests and decide about what they found.
  + The user will use their knowledge to determine what the observed injury is. However, sometimes the injury is not observable because the injury is internal. After observation, the user will realize that there are no visual observations and to move onto the next mini-game, Palpations.
  + The JSON for this minigame will contain coordinates that the observations are located. Paired with the coordinates are information on that coordinate, either a description or image. The JSON also contains the answer the user needs to find before proceeding forward.
* Palpation Minigame
  + Palpation mini-game works almost exactly like the observation mini-game. The difference is that the palpation mini-game has a pain meter. The closer the user gets to the source of pain, the higher the pain meter will go.
  + When the user finds the point of pain, the diagnosis will pop up on the screen. The user must memorize the diagnosis and input their findings as their answer. The discovery will disappear once they stop touching the point of pain so memorizing the result is requirement for the game. The user can move on once they find a prognosis. The user must also memorize the pain rating for the range of motion check.
  + The JSON object will look like an observation object. The coordinates in the object will denote the location on the athlete avatar. The maximum pain level is the highest the pain level will go as the user searches for the location of the injury. The JSON object will also contain the answer that needs to be submitted as well.
* Range of Motion Minigame
  + Range of Motion is the most complicated mini-game as it requires a range of animations to display the range of motion of an athlete. The animation will be a set of instructions on the athlete character model that will be made from an SVG.
  + From the information the user obtained earlier, the user must determine which Range of Motion test the athlete should do. Once they find the correct test, they will notice the animation will have difficulty moving. Each animation will have an active and passive movement. Active movement is movement the athlete does on their own. Passive is movement the user (or athletic trainer) helps the athlete move. These are both required to do the test properly.
  + If the athlete is in too much pain, the user does not need to do this minigame.
  + Once the user finds the proper test to do, the user can move on.
  + The JSON object will contain the proper test the user needs to find to proceed. The JSON will also contain the restricted angle of movement. A human can normally bend their elbow 180 degrees upward from a downward position. This object would be less than 180 depending on the severity of the injury.
* Treatment
  + Once the user has done all the test, the user needs to determine how the athlete should be treated. The athlete will either need to be recommended to a specialist or if they need to ice and elevate the injury.
  + The JSON object for this minigame will contain the answers for how the athlete will be treated.
  + When the user finishes this minigame, they will return to the main screen and the athlete will be marked as treated with a cooldown.

**Instructor Side**

The instructor, expert or super user needs certain functionality to use this application effectively. A list of features the super users need are the ability to create new scenario, edit scenarios, delete scenarios, read scenarios, check on students/players and their teams, view notifications of the application, and broadcast scenarios to the students.

* **Create new scenarios:** The first option in the menu for the super user is the create a new scenarios option. When the super user first opens this option, they will be prompted with a pre-created scenario number, title, location of injury, type of injury and review. Location and type of injury were decided on with the experts because it falls in line with the curriculum. After the super user enters in the initial information, they can put in the information for the games.
  + **Scenario Number:** This ID is used as a reference a scenario by either the student or professors.
  + **Scenario Title:** The title of the scenario that help communicate what the scenario is about.
  + **Location of Injury:** This describes in generic terms where the location of the injury is on the human body.
  + **Type of Injury:** This describes the proper term for the type of injury sustained by the athlete.
  + **Review:** The super user gives a brief description of what is included within the scenario. This is used for the review portion of the application.
  + **HOPRS:** After the super user selects next, they will have to enter in information for the mini-games using the HOPRS method.
  + **History:** As shown above, the super user will input information to fill out the story for the history game. They will also fill out the keywords and answers to the question.
  + **Observation:** As shown above, the super user will input each coordinates and answers to the observation. The super user will select a point on a test athlete and attach an answer to that coordinate.
  + **Palpation:** As shown above, the super user will input each coordinate and answer for a palpation. The super user will select a point on a test athlete, a maximum pain threshold, and an answer.
  + **Range of Motion:** As shown above, the super user will select a set of tests that will have a limited range of motion. These tests will also be given an indicator of how limited the athlete’s movement is.
  + **Special Tests:** As shown above, the super user will select if there are any special tests required for this scenario.
  + **Treatment:** As shown above, the super user will select the answer for the treatment that the player will determine over the course of the scenario.
* **Edit/Delete Scenarios:** Edit screen works like the new scenario screen. The first screen you are taken to is a search screen where the user can search a scenario or delete one. If they choose to edit the scenario, they will be given the same screens as a new scenario except with the information filled out with the information of the scenario. If the user wants to delete a scenario, they will be given a confirmation screen before proceeding.
* **Read Scenarios:** Because professors may need to reference a scenario in plain English, the review system has been made that translates the data from the database and turns the information into plain English that fits on one page. If a student/player comes to the super users with a question about the scenario, this screen allows them to quickly provide an answer if it is not a technical issue.
* **Test Scenario:**  If the super user designs a scenario, they will need to test it to make sure that it makes sense for them and the players/students. On the screen, it will simulate how it would look on a phone screen that player would use. The super user simply must enter in the scenario ID or title that they would like to use, and the scenario will start from the beginning. This also allows the super user to work alongside with a student/player if there are any confusions.
* **Check Student and their Athletes:** For use in the classroom, its important for professor to see how often the students are using the application and how many of their athletes are currently injured. It is not intended to be used to keep track of grades.
* **Broadcast Scenario:** When a professor uses this for a classroom setting, they will need to send a scenario to everyone in the class. This system finds an uninjured athlete and gives them an injury. It does this once for every student in the class. The professor will be able to send either a specific scenario to a student’s athlete or a random one with given parameters.

**Student/Player Side**

The player is the main user and needs certain features to work properly. The main user needs to be able to view their team of athletes, switch to different athletes, treat those athletes, switch their settings, make notes and view progress of athlete treatment and create an athlete.

* **Treat an Athlete:** This section contains the mini-games the player will be playing to fix their athletes. This contains the HOPRS method and treatment. Refer to the mini-game section for more information on this.
* **Track Progress:** In the treatment section and team section, the student will need to be able to track their progress of the athlete. If the student stops treating the athlete, the system should keep track of what they have performed so far. This progress should also be viewable on the main screen when the athlete is selected and when they are not selected in the team menu.
* **Track Athlete through Teams:** The student will keep track of all their different athletes they have made. When they select an athlete, they will be taken to the main menu screen where they can move onto treatment. This screen keeps track of the status of all the athletes they have made. If the user wants to create an athlete, they will select “Add athlete” and customize an athlete within given parameters. They will also be able to delete all but one athlete from this screen.
* **Create Athlete:** When the user first starts the application, they will be forced to create an athlete as there should always be one athlete attached to a player. On the teams screen, there will be an option to add athlete.
* **Check and Add Reference:** The student will be able to take and track notes using the reference section. Reference will be on each screen as a student may need to take notes or check notes at any time. It would be inconvenient to go all the way back to the main menu to check their notes.
* **Change Settings:** The student may need to change some settings such as font size or color settings. This will be kept on the main menu.