*Florida International University*

*School of Computing and Information Sciences*

CIS 4911 - Senior Capstone Project

Software Engineering Focus

Final Deliverable

SkillCourt

Team # 10

**Team Members**

Andres Ruggiero

Erico Oyarzun

**Product Owner**: Jaime Borras, Gummi Traustason

**Instructor**:Masoud Sadjadi

Copyright and trademark notices, restrictions on copying or distributing the documentation, information for contacting the issuing organization (reader’s comments), warranties, contractual obligations or disclaimers, and general warnings and cautions.

***Abstract***

*There is a lot involved with the training of soccer players. The current system for training is primitive usually involving an instructor and a physical field for playing. The primary objective is to produce a new, modern, and system for training soccer players. The system will be a program with features that will assist players for learning the skills required on their own.*

*Implementing this system is revolutionary to the way avid players train in the sport. With the functionality and portability that SkillCourt offers, the user can create a personalized regimen for improving skills. Thus, SkillCourt offers an overall improvement to both the soccer training and playing experience for players. Utilizing different engineering architectures and techniques has allowed to create a system which offers a user-friendly interface on top of a model that is easy to understand and maintain.*

**Table of Contents**

1. Introduction........................................................................................................................4
   1. Current System......................................................................................................4
   2. Purpose of New System........................................................................................5
2. User Stories.......................................................................................................................7
   1. Implemented User Stories.....................................................................................7
   2. Pending User Stories.............................................................................................9
3. Project Plan.....................................................................................................................10
   1. Hardware and Software Resources.....................................................................10
   2. Sprints Plan..........................................................................................................11
4. System Design.................................................................................................................20
   1. Architectural Patterns...........................................................................................20
   2. System and Subsystem Decomposition..............................................................21
   3. Persistent Data Storage.......................................................................................24
   4. Deployment Diagram...........................................................................................24
   5. Design Patterns.................................................................................................. 25
5. System Validation............................................................................................................27
6. Glossary...........................................................................................................................50
7. Appendix..........................................................................................................................52
   1. Appendix A UML Diagrams..................................................................................52
   2. Appendix B User Interface Design.......................................................................62
   3. Appendix C Sprint Review Reports......................................................................65
   4. Appendix D Sprint Retrospective Reports...........................................................67
   5. Appendix E Use Case..........................................................................................69
8. References………............................................................................................................84

1. Introduction

This section provides an overview and description of the functionalities of the first release of SkillCourt. Followed by the purpose and description of the second release of the system, explaining the new functionalities developed.

The following section offers a detailed description of the Product Backlog, which contains implemented and pending user stories of the Second Release of the system. Then, a section is dedicated to describe how the project plan was divided, as well as the software and hardware resources utilized. Next section describes the system design, architectural & design patterns and what is the system & subsystem decomposition of SkillCourt. Moreover, the subsequent section offers the System Validation test cases used. And finally this document concludes with a Glossary, Appendix and References section.

1.1 Current System

The market lacks any non-traditional training system for Soccer players. There are not many companies addressing this problem right now. Companies such as Axon Sports, for example, have tried to develop a training system that improves brain efficiency with games that use the touch technology. However no customizable interactive training system is commercially available for users to buy and use.

The traditional system used for training is primitive, usually involving an instructor and a physical field for playing. The limitations of the traditional training system include the lack of one to one interactivity between the coach and each of the players, and the lack of resources for individual players and low budget teams. Additionally, for games like Soccer that require a big field, the availability of space is crucial to training, and lack of space to train is a major setback in many cases, often leading to the inability to train as desired.

The purpose of the current system is to create a concept that will provide users an idea on what is the new SkillCourt training methodology about. The first release of SkillCourt contains three main subsystems. A SkillCourt Android Application that offers the ability to log in as a user or guest, select a routine from a list of two pre-defined ones and send the information to an Arduino Board via Bluetooth, which serves as an ideal Set of Pads. In the final product, the idea is that the Android Application sends the information to an actual set of pads, which will ultimately play the selected routine. But for the purpose of the first release, the Arduino Board (Set of Pads) served as a means of transferring the information to a Simulator. The second subsystem is a stand-alone desktop Simulator, which task is to play the selected routine. After playing the routine, the Android Application shows some statistics regarding the game played in the simulator.

The third sub-system is a SkillCourt Website that allows new users to register and existing users to manage and edit their account information.

1.2 Purpose of New System

The new system’s purpose is to offer users a solution that serves as a proof-of-concept on SkillCourt’s new training methodology. In the previous version it was troublesome for a new user to understand the idea of the new and innovative training solution that SkillCourt offers; because it requires the installation of a Desktop Simulator and Android Application, as well as an Arduino Board.

The current system’s audience was limited only to users that could purchase, connect and set-up the Arduino Controller with the simulator and Android Application. The new system offers any user with Internet connectivity, a browser with HTML 5 and a computer, the opportunity to experience SkillCourt’s proof of concept.

The new system is a Website that offers players the ability to play routines on the website, with an online version of the previously existing desktop simulator. The new online simulator also includes five move predefined routines to enhance the player's’ training capabilities.

The second release of SkillCourt introduces the concept custom coaching, offering them the ability to play custom routines designed to improve their personal weaknesses. The system also allows coaches to create & edit custom routines, as well as assigned them to their players. Both coaches and players have the ability to manage their account information on the website.

The online simulator also provides live feedback of your performance while playing a routine, giving awareness of the player’s current performance.

2. User Stories

The purpose of this section is to provide a description of all the user stories in the product backlog of the second release of SkillCourt. The user stories are divided into two sub-sections. The first subsection contains all the implemented user stories for the new system. Followed by the stories that are carried over from the previous release, including stories not implemented during the development of this project.

**2.1 Implemented User Stories**

The implemented user stories constitute all the user stories implemented in this version of SkillCourt.

**User Story # 1 - Add Simulator to Website (Rebuild simulator in Processing JS)**

In the previous version of SkillCourt, the user was able to play the game through a simulator on a local machine. The main goal of this user story was to take the simulator and put it online, so users have the ability to see a proof of concept online. In order to perform this story, we had to rebuild the entire simulator on processing.js (a JavaScript based language).

**User Story # 2 - Implement Routines**

A user should be able to play Home Chase and Home Fly routines using the simulator on the website, so that they can grasp an idea on how the game works.

**User Story # 3 - Create xCue and Ground Chase routines**

As a user, I should be able to select the difficulty level and game type for xCue and Ground Chase routines so that I may have an idea of how it would work in the room.

**User Story # 4 - Enhance Simulator**

As a user I should receive a new set of targets if I take too long of the current, so that the game does not become stagnant. A user should be able to select if they want to play with 3 or 4 walls, so that they may have an idea of how different setups affect the routines. As a user, they should have a clear idea of when the routines starts, so it will not affect their ability to play.

**User Story # 5 - Implement Coach Feedback for Online Simulator**

As a user, I should receive voice-feedback in the middle of the routine so that I feel encouraged or know how to improve my performance. The online simulator should also play a sound indicating that a player missed or hit a target.

**User Story # 6 - Integrate Simulator to Mobile App**

As a user I should be able to use the Mobile Application to select the routine I want to play, so that I may know how it works in the SkillCourt room. A user should be able to select the same options seen on the website when selecting a routine on the SkillCourt Android Application

**User Story # 7 - Implement Custom Coach Routine Wizard**

As a user I should be able to create fully custom routines of the SkillCourt website so that I may have a complete say in the movements of the player.

**User Story # 8 - Allow users to Play Custom Coach Routines**

As a user, I should be able to play the Custom Coach Routines made on the Custom Coach Wizard so that I may see how they would play in the SkillCourt Room.

**User Story # 9 - Allow Coach to Navigate the Website**

As a user (coach), I should be able to manage custom routines, view/manage my profile, and view/manage my players.

**User Story # 10 - Allow Player to Navigate the Website**

As a player I should be able to Log In, view assigned custom coach routines & profile, and play custom and default routines on the simulator.

**User Story # 11 - Make Website Accessible to All**

As a user I should be able to access the SkillCourt website through www.skillcourt.com, so that I may make an account and access the applications features.

**User Story # 12 - Choose from a list of positions**

As a user I should be able to select, from a predefined list, what type of position I play, so that my coach may adapt my custom routines to train my weaknesses.

**2.2 Pending User Stories**

**User Story # 13 - Integrate Live feedback to Mobile Application**

As a user, I should be able to see my live performance during a game on the mobile application, so that I may know my current game state.

**User Story # 14 - Implement Push Notifications for Custom Coach Routines**

As a player, I should receive a push notification when my coach assigns me a custom coach routine, so that I may be aware when new routines are available for me.

**User Story # 15 - Competition Sharing in Social Media for Rewards**

The mobile application should allow users to share results of competitions with Friends on social media in return for rewards.

**User Story # 16 - Add Advertising spots**

The mobile application and website should have spots reserved for the eventual addition of the advertising.

**User Story # 17 - Create iOS SkillCourt App**

iOS version of the Android SkillCourt Application.

3. Project Plan

As part of the project’s development process, we have created a structured and detailed plan to organize and divide our work. Since the methodology selected is Agile Scrum, our project plan is divided into five two-week Development Sprints and a final one-week Release Sprint. During the first five Sprints, the team is dedicated to retire User Stories from the Product Backlog depending on the priority of each. The final Sprint is dedicated to prepare all the materials for the Product Showcase.

**3.1 Hardware and Software Resources**

The hardware needed for the second release of SkillCourt includes an Android device with bluetooth capabilities, and an Arduino Microcontroller with a Bluetooth module. Two computers are also vital to the development process.

As a tool for project management and team collaboration, Mingle from Thoughtworks was used. For version control and source code management, the development team selected Github.

The system is divided into several subsystems, which requires a wide variety of software resources. The SkillCourt website was developed using HTML, CSS, JavaScript, PHP, JQuery and AJAX, and a XAMPP web server was also needed to host the website during development stages. Since the requirements of the project dictate that the website has to accessible from any computer, Amazon Web Services EC2 is used to host the SkillCourt system. Parse Cloud is used as a Back-end Service, not only offering Cloud access to the data model but also Application Analytics and push notification support.

Some changes were necessary to the Android Application and Arduino microcontroller, which requires the use of Android Studio and Arduino Software. Part of the project was also directed towards improving the User Interface of the website, which required the use of Webcode & Sketch.

**3.2 Sprint Plans**

### Sprint 1

(18/05/2015 - 29/05/2015)

**User Story # 2 - Implement Routines**

***Tasks***

* Implement Fly Routine
* Implement Chase Routine
* Implement Three Wall Chase Routine
* Implement Home Fly
* Implement Home Chase

***Acceptance Criteria***

* A user is able to play “Home Chase” routine, having the ability to select difficulty level and game type.
* A user is able to play “Home Fly” routine, having the ability to select difficulty level and game type.
* A user is able to play “Fly” routine, having the ability to select difficulty level and game type.
* A user is able to play “Chase” routine, having the ability to select difficulty level and game type.
* A user is able to play “Three Wall Chase” routine, having the ability to select difficulty level and game type.

***Modeling***

To view the sequence diagrams please refer to Appendix A.

### Sprint 2

(29/05/2015 - 12/06/2015)

**User Story # 1 - Add Simulator to Website (Rebuild simulator in Processing JS)**

***Tasks***

* Create Pad Class
* Create Wall Class
* Create Room Class
* Create Routine Class
* Create Game Class
* Create Statistics Class
* Detach Simulator
* Integrate Website and simulator
* Implement Floor/Ground Pads to guide/drive ball
* Develop live-feedback

***Acceptance Criteria***

* A user is able to play different routines using the simulator on the website.
* A user is able to play HomeChase routine using the simulator on the website.
* A user is able to play HomeFly routine using the simulator on the website.
* A user is able to play Chase routine using the simulator on the website
* A user is able to play Fly routine using the simulator on the website.
* A user is able to play Three Wall Chase routine using the simulator on the website.

***Modeling***

To view the sequence diagrams please refer to Appendix A.

**User Story # 3 - Create xCue and GroundChase Routines**

***Tasks***

* Implement “Ground Chase” Routine.
* Develop Difficulties for “Ground Chase” Routine.
* Implement “xCue” Routine.
* Develop Difficulties for “xCue” Routine.

***Acceptance Criteria***

* A user is able to play xCue routine with difficulties using the simulator on the website.
* A user is able to play GroundChase routine with difficulties using the simulator on the website.

***Modeling***

To view the sequence diagrams please refer to Appendix A.

**User Story # 4 - Enhance Simulator**

***Tasks***

* Add Stop Button to website
* Change how game starts
* Implement Timeouts
* Implement Custom Room

***Acceptance Criteria***

* The simulator should produce new targets if the current ones are taking too long to be hit.
* Users should be able to select game mode based on 3 or 4 walls and play in that customized room.
* The game has a countdown that is easy to understand by any user.

***Modeling***

To view the sequence diagrams please refer to Appendix A.

**User Story # 5 - Implement Coach Feedback for Online Simulator**

***Tasks***

* Implement mid-game Coach Feedback.
* Implement sound-feedback for successes and misses.

***Acceptance Criteria***

* The website provides corrective and/or encouraging feedback based on current routine.
* The website plays a sound when a user successfully hits or misses a target pad.

***Modeling***

To view the sequence diagrams please refer to Appendix A.

### Sprint 3

(12/06/2015 - 26/06/2015)

**User Story # 6 - Integrate Simulator to Mobile**

***Tasks***

· Update Arduino Sketch

· Research: Android Development & Bluetooth connectivity

· Update Application Settings

· Connect simulator

***Acceptance Criteria***

· A user is able to select a routine in the mobile application and play it in the simulator.

***Modeling***

To view the sequence diagrams please refer to Appendix A.

**User Story # 7 - Implement Custom Coach Routine Wizard**

***Tasks***

· Create GUI for Wizard

· Connect FUI to customWizard sketch

· Create/Implement StepCreator Interface

· Implement classes to create customRoutine

***Acceptance Criteria***

· A user should be able to create custom routines on the website

***Modeling***

To view the sequence diagrams please refer to Appendix A.

**User Story # 8 - Allow users to play custom coach routines**

***Tasks***

* Update UI on Play Custom Routine part of the website
* Create custom routine class
* Create/implement Master Game class
* Create Custom Game class
* Code Integration
* Implement Game Interface Class
* Design solution for plating custom game on website
* Integrate custom game class into Sketch

***Acceptance Criteria***

* A user should be able to play previously created Custom Routines on the online simulator.

***Modeling***

To view the sequence diagrams please refer to Appendix A.

### Sprint 4

(26/06/2015 - 10/07/2015)

**User Story # 6 - Integrate Simulator to Mobile App**

***Tasks***

· Update Arduino Sketch

· Research: Android Development & Bluetooth connectivity

· Update Application Settings

· Connect simulator

***Acceptance Criteria***

· A user is able to select a routine in the mobile application and play it in the simulator.

***Modeling***

To view the sequence diagrams please refer to Appendix A.

**User Story # 7 - Implement Custom Coach Routine Wizard**

***Tasks***

· Create GUI for Wizard

· Connect FUI to customWizard sketch

· Create/Implement StepCreator Interface

· Implement classes to create customRoutine

***Acceptance Criteria***

· A user should be able to create custom routines on the website

***Modeling***

To view the sequence diagrams please refer to Appendix A.

**User Story # 8 - Allow users to play custom coach routines**

***Tasks***

* Update UI on Play Custom Routine part of the website
* Create custom routine class
* Create/implement Master Game class
* Create Custom Game class
* Code Integration
* Implement Game Interface Class
* Design solution for plating custom game on website
* Integrate custom game class into Sketch

***Acceptance Criteria***

* A user should be able to play previously created Custom Routines on the online simulator.

***Modeling***

To view the sequence diagrams please refer to Appendix A.

### Sprint 5

(10/07/2015 - 24/07/2015)

**User Story # 9 - Allow coach to navigate website**

***Tasks***

* Research: Learn Parse
* Research: Learn AJAX/Jquery
* Create Coach page
* Implement Create Default Coach routines
* Implement Create custom Coach routines
* Implement Manage Coach routines
* Implement Edit coach routines
* Implement delete coach routines
* Implement unassign coach routines
* Implement assign coach routines

***Acceptance Criteria***

* A coach can view and manage routines.
* A coach can view and manage his/her players.
* A coach can view and manage his/her profile.

***Modeling***

To view the sequence diagrams please refer to Appendix A.

**User Story # 10 - Allow player to navigate website**

***Tasks***

* Research: Learn Parse
* Implement User Card
* Implement Signup Page
* Implement Login Page
* Implement Change Password Page
* Implement User Profile Page
* Implement Player Routines Page
* Implement Main Page
* Implement UI for Profile Page
* Implement UI for Signup Page
* Implement UI for Player Routines PAge
* Implement UI for Main Page
* Implement UI for LogIn Page
* Implement UI for Change Password Page
* Design Website UI

***Acceptance Criteria***

* A player is able to login to the website
* A player is able to view assigned custom & default routines
* A player is able to view and modify profile
* A player is able to play custom and default routines

***Modeling***

To view the sequence diagrams please refer to Appendix A.

**User Story # 11 - Make website accessible to all**

***Tasks***

* Change domain name
* Set up Amazon Web Services EC2 Instance
* Migrate MYSQL to DB Parse

***Acceptance Criteria***

* Website is accessible to all through the domain www.skillcourt.com
* Users may make accounts as players or coaches.
* Users may make use of the SkillCourt Simulator and SkillCourt Routine Wizard.

***Modeling***

To view the sequence diagrams please refer to Appendix A.

**User Story # 12 - Choose from a list of positions**

***Tasks***

* Implement positions attribute for a player

***Acceptance Criteria***

* Provide the player the ability to choose his position from a list of predefined ones.

***Modeling***

To view the sequence diagrams please refer to Appendix A.

4. System Design

The following chapter provides an insight into SkillCourt’s system architectural patterns. It was divided into three major subsystems, each one with specific functionality that adds value to soccer training process. First, an overview of the system design & architectural patterns is introduced. Followed by a description and a package diagram (Figure 2) of the system and subsystem decomposition. Moreover, a deployment diagram (Figure 4), which illustrates what subsystems will reside on each hardware component and how the different pieces are connected. Finally, a brief explanation on what design patterns were used and the reasons behind it.

**4.1 Architectural Patterns**

The architectural pattern selected to develop the new system is a 3-tier architectural pattern (Figure 1). The solution proposed for this release is a website that required the use of database connectivity to implement some of the new features. This design pattern was selected because it offers the system the ability to have three independent layers, each of them replaceable if needed.

Three-tier architecture was suitable for this project because the technologies used in some of the components were not initially defined. This allows the team to design the system considering a set of independent technologies for each layer (tier). The presentation tier responsibility is to display information to the user, which in our case is the content rendered by the browser. The second tier provides the SkillCourt website the ability to keep the logic module independent. And the third tier offers the ability to store data independent of the rest of the layers.

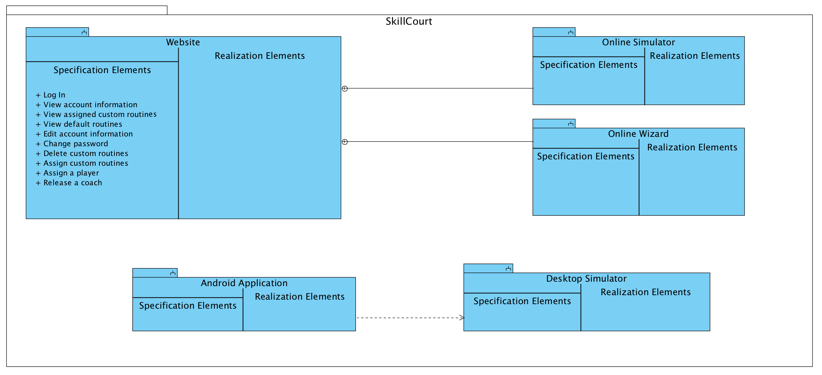


Figure #1 - Three - tier architecture

**4.2 System and Subsystem Decomposition**

SkillCourt is conformed of three major subsystems, each of them offering different functionalities. Below is a brief description of their functionality, followed by a UML package diagram that describes the system (Figure 2) .

1. **SkillCourt Website:** The first subsystem is a website that allows users to view & edit their account information, view & play custom & default routines assigned by their personal coach and play any of the seven predefined routines. The website’s purpose is to provide a proof-of-concept of what the new SkillCourt training methodology is about. The website also provides coaches to login and create, edit, delete and assign routines to their players. The website also contains the following subsystems:
   1. **Online Simulator:** The online simulator allows users to play custom and default assigned routines, as well as pre designed routines. It also provides players with live statistic feedback about their performance in the current game. The class diagram for this subsystem was developed using UML and represents the classes needed to implement it (Refer to Appendix A).
   2. **Online Wizard:** As the second subsystem of the website, the Online Wizard allows coaches to create custom routines, as well as edit previously existing custom routines. The class diagram for this subsystem was developed using UML and represents the classes needed to implement it (Refer to Appendix A).
   3. **Account Management:** This subsystem provides the user the ability to view & edit their personal and account information like first name, middle name, last name, gender, birthdate, username, password email address and coach information, if any.
   4. **Routine Management:** The routine management is designed to work for players and coaches, and both have different functionalities. The coach has the ability to sign & release a player, as well as assign & unassign routines to their players. In this subsystem, the player has the ability to view & play the assigned default or custom routines.
2. **Android Application**: The second subsystem is an Android Mobile Application that allows users to play new routines on the desktop simulator via the Arduino Controller. The User Interface and application logic were modified to adapt to the new simulator upgrades.
3. **Desktop Simulator**: As a third subsystem, the desktop simulator allows a user to play a default routine selected in the Android Application. The desktop simulator was upgraded to add Time Per Round game type, an option to remove a wall and ability to use the Ground Pads.

Figure #2 - System & Subsystem Package Diagram

**4.3 Persistent Data Storage**

For Data Storage the new version of SkillCourt uses Parse as a Back-end Service. This was chosen as a storage solution because it offers not only cloud access, but also includes APIs that handle encryption, push notification to mobile phones and application analytics. Below is the Entity-Relationship Diagram for the database:

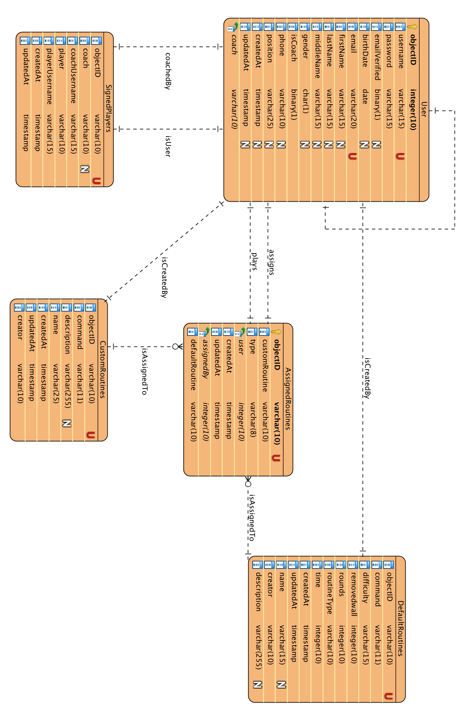


Figure #3 - Entity Relationship Diagram

**4.4 Deployment Diagram**

Using a UML deployment diagram, illustrate which subsystems will reside on each hardware component and show how the different pieces are connected.

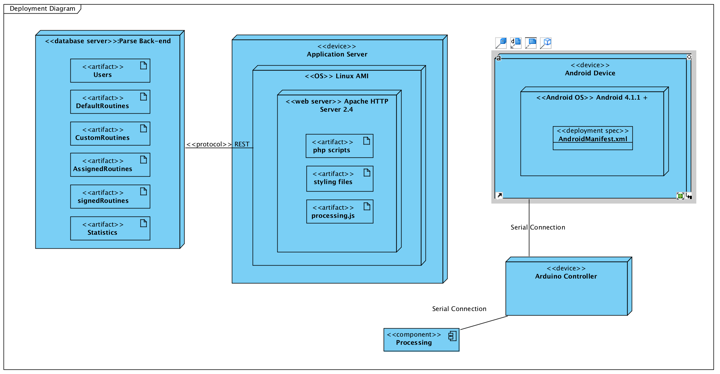


Figure #4 - SkillCourt Deployment Diagram

**4.5 Design Patterns**

The subsystem that contains the mobile application, desktop simulator, and Arduino controller was built using MVC architecture (Figure 5). This simplifies the role of each member and keeps their functionality organized.

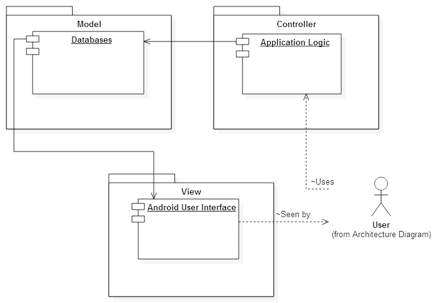


Figure # 5 - MVC Design Pattern

5. System Validation

The following section provides a detailed list of all the system validation scenarios that were performed on the system. The system validation is divided by each implement user story.

**User Story # 1 - Add Simulator to the Website**

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Chase\_Routine\_1** |
| **Purpose:** | **Test that a user is able to play a chase routine on the website** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Difficulty = {Novice}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Chase routine from dropdown menu**  **3. User selects Novice difficulty level**  **4. User presses play button** |
| **Expected Results:** | **The simulator should start the countdown and start the game** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Chase\_Routine\_2** |
| **Purpose:** | **Test that a user is able to play a chase routine on the website** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Difficulty = {Intermediate}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Chase routine from dropdown menu**  **3. User selects Intermediate difficulty level**  **4. User presses play button** |
| **Expected Results:** | **The simulator should start the countdown and start the game** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Chase\_Routine\_3** |
| **Purpose:** | **Test that a user is able to play a chase routine on the website** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Difficulty = {Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Chase routine from dropdown menu**  **3. User selects Advanced difficulty level**  **4. User presses play button** |
| **Expected Results:** | **The simulator should start the countdown and start the game** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Chase\_Routine\_4** |
| **Purpose:** | **Test that a user can’t play a Chase routine with rounds** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **None** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Chase routine from dropdown menu**  **3. User tries to select the Play By menu** |
| **Expected Results:** | **The dropdown should be blocked avoiding the user to choose rounds** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Chase\_Routine\_5** |
| **Purpose:** | **Test that a user can’t play a Chase routine with rounds** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **none** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Chase routine from dropdown menu**  **3. User tries to select the Time Per Round checkbox** |
| **Expected Results:** | **The Time Per Round checkbox should be blocked** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Fly\_Routine\_1** |
| **Purpose:** | **Test that a user is able to play a fly routine on the website** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Difficulty = {Novice}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Fly routine from dropdown menu**  **3. User selects Novice difficulty level**  **4. User presses play button** |
| **Expected Results:** | **The simulator should start the countdown and start the game** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Fly\_Routine\_2** |
| **Purpose:** | **Test that a user is able to play a fly routine on the website** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Difficulty = {Intermediate}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Fly routine from dropdown menu**  **3. User selects Intermediate difficulty level**  **4. User presses play button** |
| **Expected Results:** | **The simulator should start the countdown and start the game** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Fly\_Routine\_3** |
| **Purpose:** | **Test that a user is able to play a fly routine on the website** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Difficulty = {Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Fly routine from dropdown menu**  **3. User selects Advanced difficulty level**  **4. User presses play button** |
| **Expected Results:** | **The simulator should start the countdown and start the game** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Fly\_Routine\_4** |
| **Purpose:** | **Test that a user can’t play a Fly routine with rounds** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **None** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Fly routine from dropdown menu**  **3. User tries to select the Play By menu** |
| **Expected Results:** | **The dropdown should be blocked avoiding the user to choose rounds** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Fly\_Routine\_6** |
| **Purpose:** | **Test that a user can’t play a Fly routine with rounds** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **none** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Fly routine from dropdown menu**  **3. User tries to select the Time Per Round checkbox** |
| **Expected Results:** | **The Time Per Round checkbox should be blocked** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Three\_Wall\_Chase\_1** |
| **Purpose:** | **Test that a user is able to play Three Wall Chase time based** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Three Wall Chase} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Three Wall Chase Routine from the dropdown menu**  **3. User selects any difficulty type**  **4. User selects ‘time’ from ‘Play By’ dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts in time based mode** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Three\_Wall\_Chase\_2** |
| **Purpose:** | **Test that a user is able to play Three Wall Chase round based** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Three Wall Chase} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Three Wall Chase Routine from the dropdown menu**  **3. User selects any difficulty type**  **4. User selects ‘rounds’ from ‘Play By’ dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts in rounds mode** |
| **Actual Results:** | **As expected** |

**User Story # 2 - Implement Routines**

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Three\_Wall\_Chase\_3** |
| **Purpose:** | **Test that a user is able to play Three Wall Chase on timed rounds mode** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Three Wall Chase} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Three Wall Chase Routine from the dropdown menu**  **3. User selects any difficulty type**  **4. User select the ‘Time Per Round’ checkbox**  **5. User enters the amount of seconds per round**  **6. User selects the play button** |
| **Expected Results:** | **The game starts in time per round mode** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Home\_Chase\_1** |
| **Purpose:** | **Test that a user is able to play Home Chase time based** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Home Chase} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Home Chase Routine from the dropdown menu**  **3. User selects any difficulty type**  **4. User selects ‘time’ from ‘Play By’ dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts in time based mode** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Home\_chase\_2** |
| **Purpose:** | **Test that a user is able to play Home Chase round based** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Home Chase} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Home Chase from the dropdown menu**  **3. User selects any difficulty type**  **4. User selects ‘rounds’ from ‘Play By’ dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts in rounds mode** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Home \_Chase\_3** |
| **Purpose:** | **Test that a user is able to play Home Chase on timed rounds mode** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Home Chase} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Home Chase Routine from the dropdown menu**  **3. User selects any difficulty type**  **4. User select the ‘Time Per Round’ checkbox**  **5. User enters the amount of seconds per round**  **6. User selects the play button** |
| **Expected Results:** | **The game starts in time per round mode** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Home\_Fly\_1** |
| **Purpose:** | **Test that a user is able to play Home Fly time based** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Home Fly} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Home Fly Routine from the dropdown menu**  **3. User selects any difficulty type**  **4. User selects ‘time’ from ‘Play By’ dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts in time based mode** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Home\_Fly\_2** |
| **Purpose:** | **Test that a user is able to play Home Fly round based** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Home Fly} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Home Fly from the dropdown menu**  **3. User selects any difficulty type**  **4. User selects ‘rounds’ from ‘Play By’ dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts in rounds mode** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Home \_Fly\_3** |
| **Purpose:** | **Test that a user is able to play Home Fly on timed rounds mode** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Home Fly} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Home Fly Routine from the dropdown menu**  **3. User selects any difficulty type**  **4. User select the ‘Time Per Round’ checkbox**  **5. User enters the amount of seconds per round**  **6. User selects the play button** |
| **Expected Results:** | **The game starts in time per round mode** |
| **Actual Results:** | **As expected** |

**User Story # 3 - Create xCue and GroundChase Routines**

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Ground\_Chase\_1** |
| **Purpose:** | **Test that a user is able to play Ground Chase time based** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Ground Chase} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Ground Chase Routine from the dropdown menu**  **3. User selects any difficulty type**  **4. User selects ‘time’ from ‘Play By’ dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts in time based mode** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Ground\_Chase\_2** |
| **Purpose:** | **Test that a user is able to play Ground\_Chase round based** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Ground\_Chase} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Ground\_Chase from the dropdown menu**  **3. User selects any difficulty type**  **4. User selects ‘rounds’ from ‘Play By’ dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts in rounds mode** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_Ground\_Chase\_3** |
| **Purpose:** | **Test that a user is able to play Ground\_Chase on timed rounds mode** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Ground Chase} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Home Ground Chase from the dropdown menu**  **3. User selects any difficulty type**  **4. User select the ‘Time Per Round’ checkbox**  **5. User enters the amount of seconds per round**  **6. User selects the play button** |
| **Expected Results:** | **The game starts in time per round mode** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Live\_Statistics** |
| **Purpose:** | **Test that a user is able to see live feedback during a current game** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Any Routine}** |
| **Steps:** | 1. **User selects the simulator tab** 2. **User select any routine** 3. **User select the play button** |
| **Expected Results:** | **The game starts and the live feedback screen shows** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_XCue\_1** |
| **Purpose:** | **Test that a user is able to play XCue on time based mode** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Xcue} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects XCue Routine from the dropdown menu**  **3. User selects any difficulty type**  **4. User selects ‘time’ from ‘Play By’ dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts in time based mode** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_XCue\_2** |
| **Purpose:** | **Test that a user is able to play XCue round based** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {XCue} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects XCue from the dropdown menu**  **3. User selects any difficulty type**  **4. User selects ‘rounds’ from ‘Play By’ dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts in rounds mode** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Play\_XCue\_3** |
| **Purpose:** | **Test that a user is able to play XCue on timed rounds mode** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Ground Chase} Difficulty = {Novice, Intermediate, Advanced}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Home XCue from the dropdown menu**  **3. User selects any difficulty type**  **4. User select the ‘Time Per Round’ checkbox**  **5. User enters the amount of seconds per round**  **6. User selects the play button** |
| **Expected Results:** | **The game starts in time per round mode** |
| **Actual Results:** | **As expected** |

**User Story # 4 - Enhance Simulator**

|  |  |
| --- | --- |
| **Test Case ID:** | **Remove\_Wall\_1** |
| **Purpose:** | **Test that a user can’t remove a wall from an Xcue routine** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Xcue}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Xcue Routine from the dropdown menu**  **3. User tries to select the Time Per Round checkbox** |
| **Expected Results:** | **The Time Per Round checkbox should be blocked** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Ground\_Chase\_Remove\_Wall** |
| **Purpose:** | **Test that a user can remove any wall and play Ground Chase** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Ground Chase} Removed Wall = {North, South, East, West}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Ground Chase Routine from the dropdown menu**  **3. User clicks the ‘Removed Wall’ checkbox**  **4. User selects any wall option from the dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts without selected wall** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Home\_Fly\_Remove\_Wall** |
| **Purpose:** | **Test that a user can remove any wall and play Home\_Fly** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Home Fly} Removed Wall = {North, South, East, West}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Home Fly Routine from the dropdown menu**  **3. User clicks the ‘Removed Wall’ checkbox**  **4. User selects any wall option from the dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts without selected wall** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Home\_Chase\_Remove\_Wall** |
| **Purpose:** | **Test that a user can remove any wall and play Home\_Chase** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Home Chase} Removed Wall = {North, South, East, West}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Home Chase Routine from the dropdown menu**  **3. User clicks the ‘Removed Wall’ checkbox**  **4. User selects any wall option from the dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts without selected wall** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Three\_Wall\_Chase\_Remove\_Wall** |
| **Purpose:** | **Test that a user can remove any wall and play Three Wall Chase** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Three Wall Chase} Removed Wall = {North, South, East, West}** |
| **Steps:** | **1. User selects the simulator tab**  **2. User selects Three Wall Chase Routine from the dropdown menu**  **3. User clicks the ‘Removed Wall’ checkbox**  **4. User selects any wall option from the dropdown menu**  **5. User selects the play button** |
| **Expected Results:** | **The game starts without selected wall** |
| **Actual Results:** | **As expected** |

**User Story # 5 - Implement Coach Feedback for Online Simulator**

|  |  |
| --- | --- |
| **Test Case ID:** | **Coach\_Feedback\_1** |
| **Purpose:** | **Test that the online simulator plays a sound to indicate miss or hit** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Any}** |
| **Steps:** | 1. **User selects any routine from the dropdown menu** 2. **User Clicks the play button** |
| **Expected Results:** | **The simulator should play a sound indicating a miss or a hit** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Coach\_Feedback\_2** |
| **Purpose:** | **Test that the online simulator plays a sound to indicate midgame** |
| **Prerequisite:** | **User accesses the website as a player or guest** |
| **Test Data** | **Routine = {Any}** |
| **Steps:** | 1. **User selects any routine from the dropdown menu** 2. **User Clicks the play button** |
| **Expected Results:** | **The simulator should play a sound when the game is in the middle** |
| **Actual Results:** | **As expected** |

**User Story # 6 - Integrate Simulator to Mobile App**

|  |  |
| --- | --- |
| **Test Case ID:** | **Simulator\_Mobile\_App\_1** |
| **Purpose:** | **Test that a user can select a routine and play it in the simulator** |
| **Prerequisite:** | **User logs as a user or guest** |
| **Test Data** | **Routine = {Any}** |
| **Steps:** | 1. **User selects any routine from the dropdown menu on the phone** 2. **User Clicks the play button** |
| **Expected Results:** | **The desktop simulator should start to play** |
| **Actual Results:** | **As expected** |

**User Story # 7 - Implement Custom Coach Routine Wizard**

|  |  |
| --- | --- |
| **Test Case ID:** | **Custom\_Routine\_Wizard\_1** |
| **Purpose:** | **Test that a coach can add a new step and targets to a custom routine** |
| **Prerequisite:** | **User logs in to the website as a coach** |
| **Test Data** | **None** |
| **Steps:** | 1. **User selects the Wizard tab in the navigation bar** 2. **User clicks the + button to add a step** 3. **User selects desired targets in the simulator** 4. **User selects Finish Step button** |
| **Expected Results:** | **The simulator should update the number of existing rounds to reflect the new created step** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Custom\_Routine\_Wizard\_2** |
| **Purpose:** | **Test that a coach cannot create a custom routine with an empty step** |
| **Prerequisite:** | **User logs in to the website as a coach** |
| **Test Data** | **None** |
| **Steps:** | 1. **User selects the Wizard tab in the navigation bar** 2. **User selects the ‘Finish Routine’ button** |
| **Expected Results:** | **The simulator should warn you that a routine cannot be created with empty step** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Custom\_Routine\_Wizard\_3** |
| **Purpose:** | **Test that a coach cannot create a custom routine with an empty round** |
| **Prerequisite:** | **User logs in to the website as a coach** |
| **Test Data** | **None** |
| **Steps:** | 1. **User selects the Wizard tab in the navigation bar** 2. **User selects the ‘Finish Routine’ button** |
| **Expected Results:** | **The simulator should warn you that a routine cannot be created with empty round** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Custom\_Routine\_Wizard\_4** |
| **Purpose:** | **Test that a coach can create a target step** |
| **Prerequisite:** | **User logs in to the website as a coach** |
| **Test Data** | **None** |
| **Steps:** | 1. **User selects the Wizard tab in the navigation bar** 2. **User selects ‘target set’ from the dropdown menu** 3. **User starts selecting pads from several walls** |
| **Expected Results:** | **The simulator should start lighting the selected pads green from all the walls selected. The simulator should also block the Ground.** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Custom\_Routine\_Wizard\_5** |
| **Purpose:** | **Test that a coach can create a ground step** |
| **Prerequisite:** | **User logs in to the website as a coach** |
| **Test Data** | **None** |
| **Steps:** | 1. **User selects the Wizard tab in the navigation bar** 2. **User selects ‘ground set’ from the dropdown menu** 3. **User one ground pad** |
| **Expected Results:** | **The simulator should start lighting the selected ground pad. If another ground pad is selected, then the simulator lights off the previously selected one.** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Custom\_Routine\_Wizard\_6** |
| **Purpose:** | **Test that a coach is able to delete a round from a custom routine** |
| **Prerequisite:** | **User logs in to the website as a coach** |
| **Test Data** | **None** |
| **Steps:** | 1. **User selects the Wizard tab in the navigation bar** 2. **User select the + button to add a new round** 3. **User selects the ‘Delete Round’ button to delete a round** |
| **Expected Results:** | **The simulator should update the number of existing rounds to display the deleted round.** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Custom\_Routine\_Wizard\_7** |
| **Purpose:** | **Test that a coach is able to delete a step from a custom routine** |
| **Prerequisite:** | **User logs in to the website as a coach** |
| **Test Data** | **None** |
| **Steps:** | 1. **User selects the Wizard tab in the navigation bar** 2. **User select the + button to add a new round** 3. **User selects the ‘Delete Step’ button to delete a step** |
| **Expected Results:** | **The simulator should update the number of existing step to display the deleted step.** |
| **Actual Results:** | **As expected** |

**User Story # 8 - Allow users to play custom routines**

|  |  |
| --- | --- |
| **Test Case ID:** | **Custom\_Routine\_Wizard\_7** |
| **Purpose:** | **Test that a player is able to play a custom routine from the online simulator** |
| **Prerequisite:** | **User logs in to the website as a player.**  **User has assigned custom routines.** |
| **Test Data** | **None** |
| **Steps:** | 1. **User selects the Simulator tab in the navigation bar** 2. **User selects the ‘Custom’ tab routines in the simulator** 3. **User selects an assigned custom routine from the dropdown menu** 4. **User clicks play** |
| **Expected Results:** | **The simulator should start showing the assigned custom routine selected** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Custom\_Routine\_Wizard\_8** |
| **Purpose:** | **Test that a player is not able to play a custom routine from the online simulator** |
| **Prerequisite:** | **User logs in to the website as a player.**  **User has not been assigned custom routines.** |
| **Test Data** | **None** |
| **Steps:** | 1. **User selects the Simulator tab in the navigation bar** 2. **User selects the ‘Custom’ tab routines in the simulator** 3. **User will not see any custom routines in the dropdown menu** |
| **Expected Results:** | **The simulator should block the play button, and no options should show on the dropdown menu** |
| **Actual Results:** | **As expected** |

**User Story # 9 - Allow coach to navigate Website**

|  |  |
| --- | --- |
| **Test Case ID:** | **Coach\_Navigation\_1** |
| **Purpose:** | **Test that a coach can assign a routine to a player** |
| **Prerequisite:** | **User logs in to the website as a coach** |
| **Test Data** | **Coach = {username password}** |
| **Steps:** | 1. **User selects the ‘Routines’ tab from the navigation bar** 2. **User selects the desired routine to assign from the list** 3. **User clicks the ‘Assign’ button** 4. **User selects a player from the drop down menu** 5. **User selects the ‘Confirm’ Button** |
| **Expected Results:** | **The simulator should assign the custom routine to selected player, and update the ‘Assigned To’ dropdown menu.** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Coach\_Navigation\_2** |
| **Purpose:** | **Test that a coach can un-assign a routine to a player** |
| **Prerequisite:** | **User logs in to the website as a coach** |
| **Test Data** | **Coach = {username password}** |
| **Steps:** | 1. **User selects the ‘Routines’ tab from the navigation bar** 2. **User selects the desired routine to assign from the list** 3. **User clicks the ‘Assign’ button** 4. **User selects a player from ‘Assigned To’ the drop down menu** 5. **User selects the ‘Un-assign’ Button** |
| **Expected Results:** | **The simulator should unassign the custom routine to selected player, and update the ‘Assigned To’ dropdown menu.** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Coach\_Navigation\_3** |
| **Purpose:** | **Test that a coach can delete a custom routine** |
| **Prerequisite:** | **User logs in to the website as a coach** |
| **Test Data** | **Coach = {username password}** |
| **Steps:** | 1. **User selects the ‘Routines’ tab from the navigation bar** 2. **User selects the desired routine to assign from the list** 3. **User clicks the ‘Delete’ Button.** |
| **Expected Results:** | **The simulator should update the routines list to show the recently deleted item.** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Coach\_Navigation\_4** |
| **Purpose:** | **Test that a coach can login to the website** |
| **Prerequisite:** | **User has existing credentials** |
| **Test Data** | **Coach = {username password}** |
| **Steps:** | 1. **User clicks the ‘Login’ button** 2. **User enters username and password information** 3. **User clicks ‘Login’** |
| **Expected Results:** | **The website should redirect the user to the coach main page** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Coach\_Navigation\_5** |
| **Purpose:** | **Test that a coach can log out of the website** |
| **Prerequisite:** | **User is logged in to the website as a coach** |
| **Test Data** | **Coach = {username password}** |
| **Steps:** | 1. **User clicks the ‘Log out’ button** |
| **Expected Results:** | **The website redirects the user to the main page** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Coach\_Navigation\_6** |
| **Purpose:** | **Test that a coach can sign a new player** |
| **Prerequisite:** | **User is logged in to the website as a coach** |
| **Test Data** | **Coach = {username password}** |
| **Steps:** | 1. **User clicks the ‘Players’ tab in the navigation bar** 2. **User types the player’s username** 3. **User clicks the ‘Sign Player’ button** |
| **Expected Results:** | **The website alerts that a player was successfully signed** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Coach\_Navigation\_7** |
| **Purpose:** | **Test that a coach can release a player** |
| **Prerequisite:** | **User is logged in to the website as a coach** |
| **Test Data** | **Coach = {username password}** |
| **Steps:** | 1. **User clicks the ‘Players’ tab in the navigation bar** 2. **User selects the player from the ‘Signed Players’ dropdown menu** 3. **User clicks the ‘Release Player’ button** |
| **Expected Results:** | **The website alerts that a player was successfully released and updates the view** |
| **Actual Results:** | **As expected** |

# **User Story # 10 - Allow player to navigate Website**

|  |  |
| --- | --- |
| **Test Case ID:** | **Player\_Navigation\_1** |
| **Purpose:** | **Test that a player can login to the website** |
| **Prerequisite:** | **User has existing credentials** |
| **Test Data** | **player = {username password}** |
| **Steps:** | 1. **User clicks the ‘Login’ button on the website** 2. **User enters the username and password** 3. **User clicks the ‘Login’ button** |
| **Expected Results:** | **The website should redirect the user to the player’s page** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **User\_Navigation\_1** |
| **Purpose:** | **Test that a user can not login with invalid credential information** |
| **Prerequisite:** | **None** |
| **Test Data** | **None** |
| **Steps:** | 1. **User clicks the ‘Login’ button on the website** 2. **User enters a non-existing username or password** 3. **User clicks the ‘Login’ button** |
| **Expected Results:** | **The website should display a ‘Invalid Login Credential Error’** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **User\_Navigation\_2** |
| **Purpose:** | **Test that a user can change their password** |
| **Prerequisite:** | **User has an existing account** |
| **Test Data** | **Username = {username or password}** |
| **Steps:** | 1. **User clicks the ‘Login’ button on the website** 2. **User selects the ‘Forgot Username or password?’ link** 3. **User enters email address or username** 4. **User receives an email to change their account password** |
| **Expected Results:** | **The website should show a forgot password window** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **User\_Navigation\_3** |
| **Purpose:** | **Test that a user is able to create an account** |
| **Prerequisite:** | **User has no SkillCourt account** |
| **Test Data** | **Non-existing email and username** |
| **Steps:** | 1. **User selects ‘Register Now’ button on the website** 2. **User enters their personal and account information** 3. **User clicks the ‘Sign Up’ button** |
| **Expected Results:** | **The website should create an account and redirect the user to the ‘Login’ window** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Player\_Navigation\_2** |
| **Purpose:** | **Test that a player is able to view assigned routines** |
| **Prerequisite:** | **User is logged in as a player** |
| **Test Data** | **User = {username,password}** |
| **Steps:** | 1. **User selects the ‘Routines’ tab in the navigation** |
| **Expected Results:** | **The website redirect the player to the ‘Routines’ page** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Player\_Navigation\_3** |
| **Purpose:** | **Test that a player is able to play assigned custom routines** |
| **Prerequisite:** | **User is logged in as a player** |
| **Test Data** | **User = {username,password}** |
| **Steps:** | 1. **User selects the ‘Routines’ tab in the navigation** 2. **User selects a routine from the list of ‘Custom’ routines** 3. **User clicks the ‘Play’ button** 4. **In the simulator, user clicks ‘Play’ when ready to start routine** |
| **Expected Results:** | **The website should redirect the player to the simulator page** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Player\_Navigation\_4** |
| **Purpose:** | **Test that a player is able to play assigned default routines** |
| **Prerequisite:** | **User is logged in as a player** |
| **Test Data** | **User = {username,password}** |
| **Steps:** | 1. **User selects the ‘Routines’ tab in the navigation** 2. **User selects a routine from the list of ‘Default’ routines** 3. **User clicks the ‘Play’ button** 4. **In the simulator, user clicks ‘Play’ when ready to start routine** |
| **Expected Results:** | **The website should redirect the player to the simulator page** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **Player\_Navigation\_5** |
| **Purpose:** | **Test that a player is able to play a predefined routine** |
| **Prerequisite:** | **User is logged in as a player** |
| **Test Data** | **User = {username,password}** |
| **Steps:** | 1. **User selects the ‘Simulator’ tab in the navigation** 2. **User selects a routine from the pre defined ‘Routines’ dropdown** 3. **User clicks the ‘Play’ button in the simulator page** |
| **Expected Results:** | **The website should redirect the player to the simulator page** |
| **Actual Results:** | **As expected** |

|  |  |
| --- | --- |
| **Test Case ID:** | **User\_Navigation\_4** |
| **Purpose:** | **Test that a user is able view their personal and account information** |
| **Prerequisite:** | **User is logged in to the website** |
| **Test Data** | **User = {username, password}** |
| **Steps:** | 1. **User selects the ‘username’ tab** 2. **User clicks the ‘Edit’ button** 3. **User changes information in any of the available fields** 4. **User clicks the ‘Save’ button** |
| **Expected Results:** | **The website should update the player’s information** |
| **Actual Results:** | **As expected** |

# **User Story # 11 - Make website accessible to all**

|  |  |
| --- | --- |
| **Test Case ID:** | **Website\_accessibility\_1** |
| **Purpose:** | **Test that a user is able to access the website from the domain www.skillcourt.com** |
| **Prerequisite:** |  |
| **Test Data** | **www.skillcourt.com** |
| **Steps:** | 1. **User enters** [**www.skillcourt.com**](http://www.skillcourt.com) **from a browser** |
| **Expected Results:** | **The website’s main page should load** |
| **Actual Results:** | **As expected** |

# **User Story # 12 - Choose from a list of positions**

|  |  |
| --- | --- |
| **Test Case ID:** | **Website\_Positions\_1** |
| **Purpose:** | **When creating a user account, test that a user can select from a list of pre defined positions** |
| **Prerequisite:** | **None** |
| **Test Data** | **None** |
| **Steps:** | 1. **User selects the ‘Register Now’ button on the main page** 2. **User selects a position from the ‘Positions’ dropdown menu** |
| **Expected Results:** | **The website should show the player a list of predefined positions** |
| **Actual Results:** | **As expected** |

6. Glossary

Define any domain-specific terms that the audience of this document may be unfamiliar with. You can assume the audience of this document to be technical savvy. Examples of terms that do not need to be defined are: HTML, CSS, Web Server, DB, and the like.

|  |  |
| --- | --- |
| **App** | **Refers to the SkillCourt Application in the Mobile Device** |
| **Pad Simulator** | **An emulated device which will take the place of SkillCourt pads for testing showcasing purposes. This device will offer all of the features a SkillCourt Pad will offer** |
| **SkillCourt** | **A system which uses SkillCourt Pads and a player interface for training soccer** |
| **SkillCourt Room** | **A 20’x40’ room with SkillCourt Pads on the walls used for soccer training.** |
| **SkillCourt Pad** | **A physical device with a flat surface that can measure and transmit when and how much pressure it received** |
| **Class Diagram** | **A pictorial representation of all the classes in the system** |
| **Object Diagram** | **A pictorial representation of an instance of a class with example of how the data of the class will be populated** |
| **Sequence Diagram:** | **A pictorial representation of how processes operate with one another and the user during the course of a specific piece of functionality.** |
| **Use Case** | **List of steps defining the interaction between the user and the system to achieve a goal** |
| **Online Wizard** | **Refers to a simulation of a SkillCourt room that allows a coach to design custom routines.** |

7. Appendix

7.1 Appendix A - UML Diagrams

### 7.1.1 Static UML Diagrams

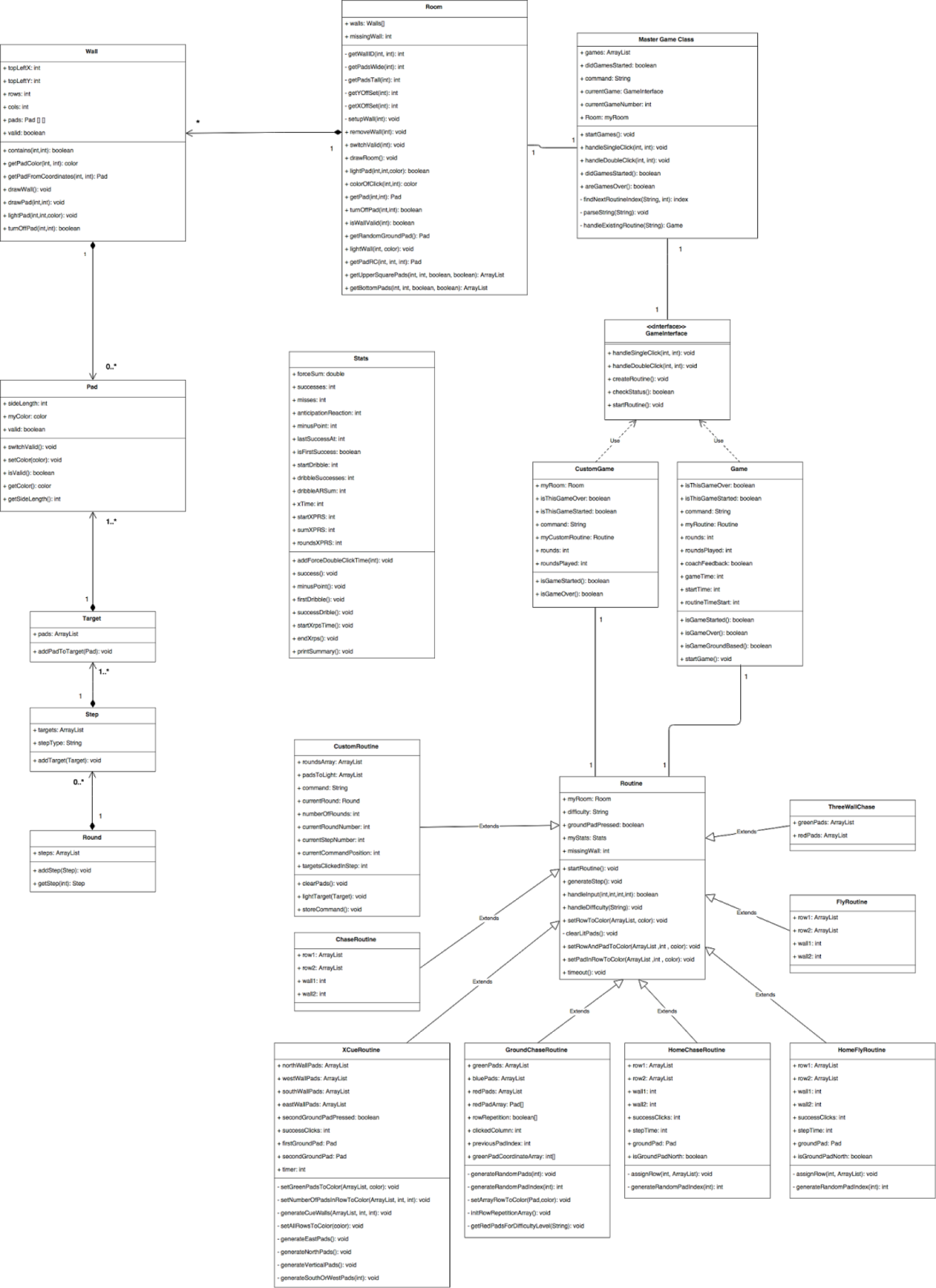
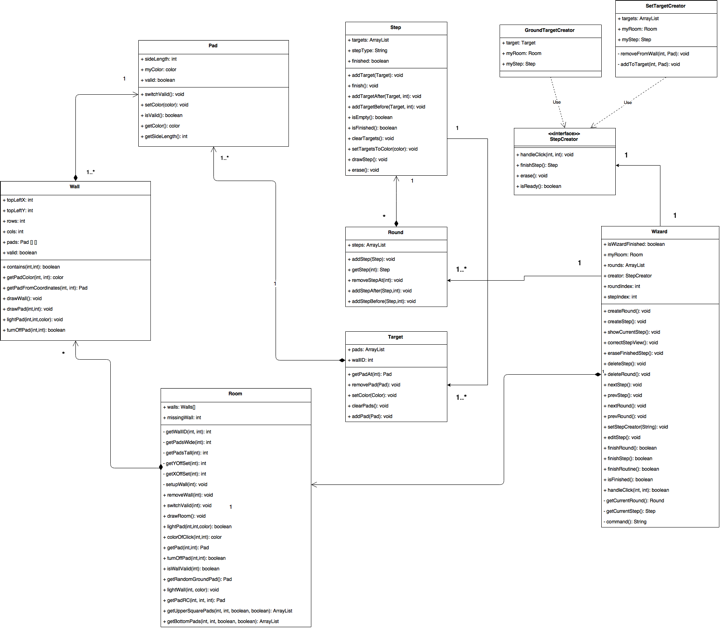


Figure # 6 - Online Simulator Class Diagram



### 

### Figure # 7 - Online Wizard Class Diagram

### 

### 

### 

### 

### 7.1.2 Dynamic UML Diagrams



Figure # 8 - Assign Coach Routine

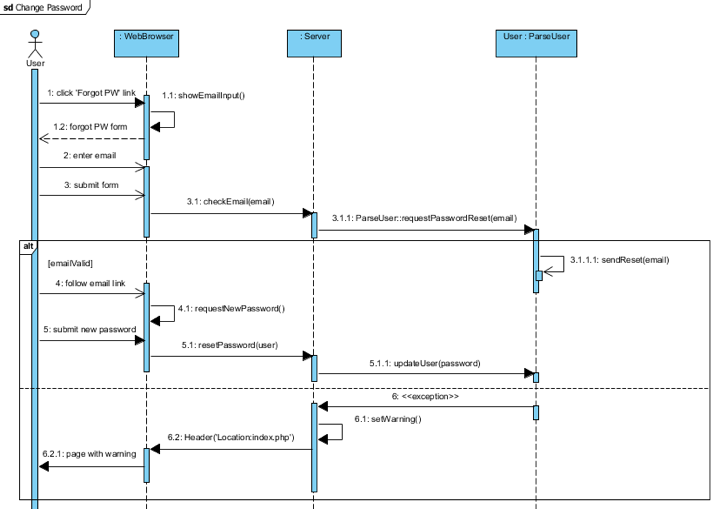


Figure # 9 - Change Password

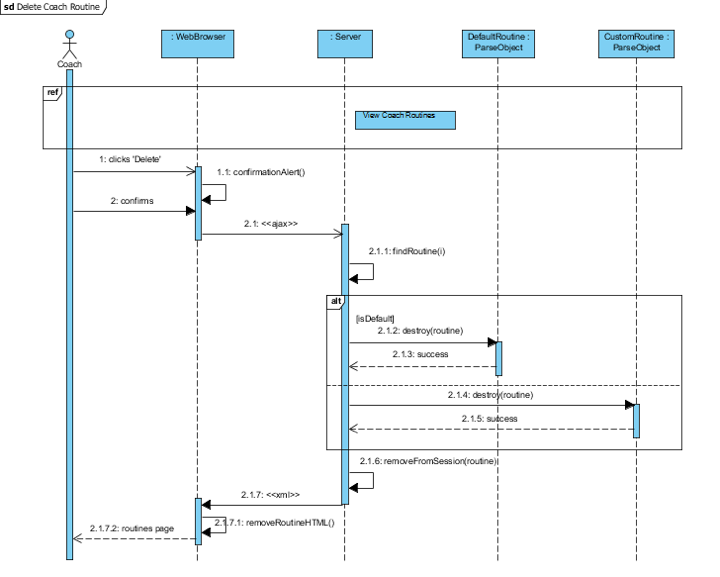


Figure # 10 - Delete Coach Routine

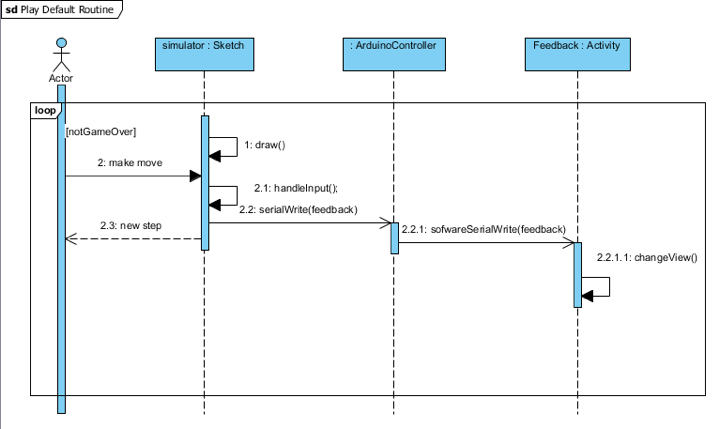


Figure # 11 - Play Default Routine

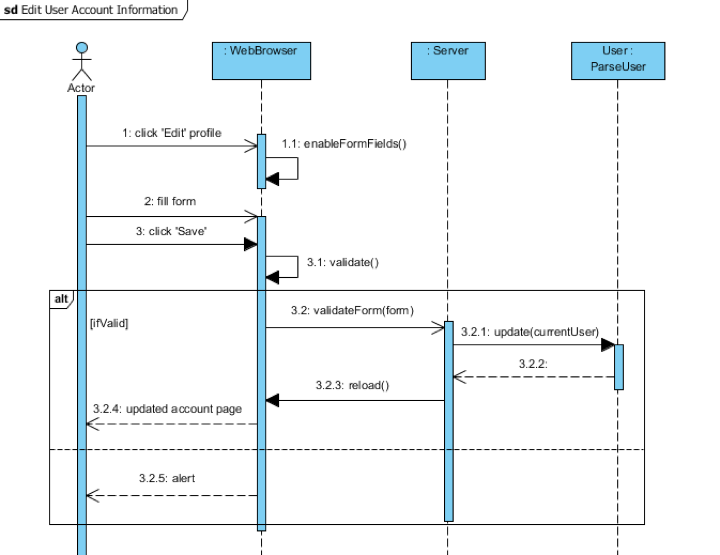


Figure # 12 - Edit User Account Information

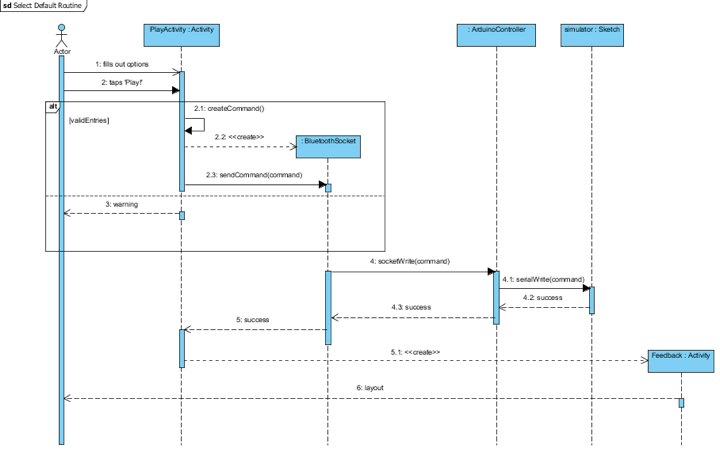


Figure # 13 - Select Default Routine

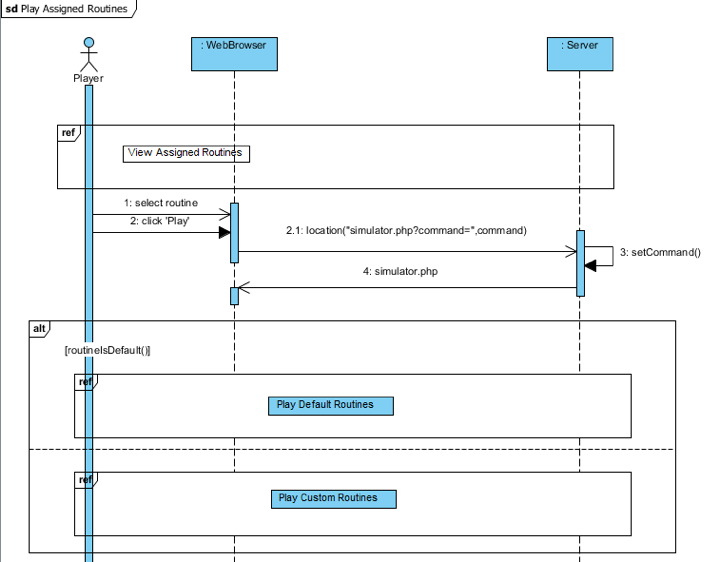


Figure # 14 - Play Assigned Routines

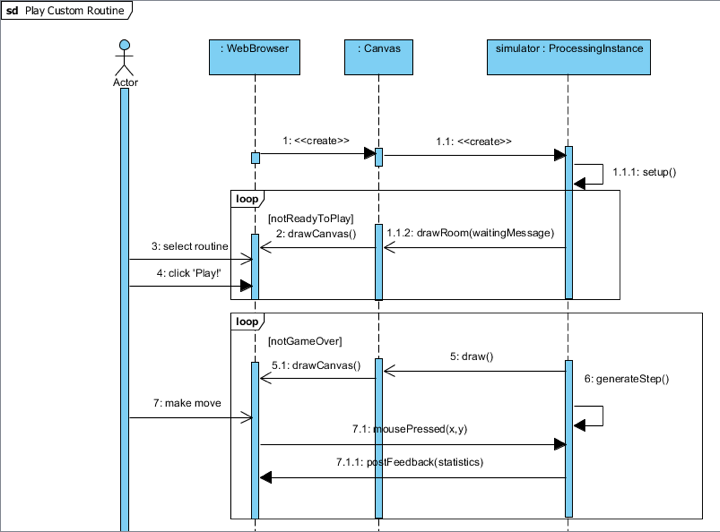


Figure # 15 - Play Custom Routine

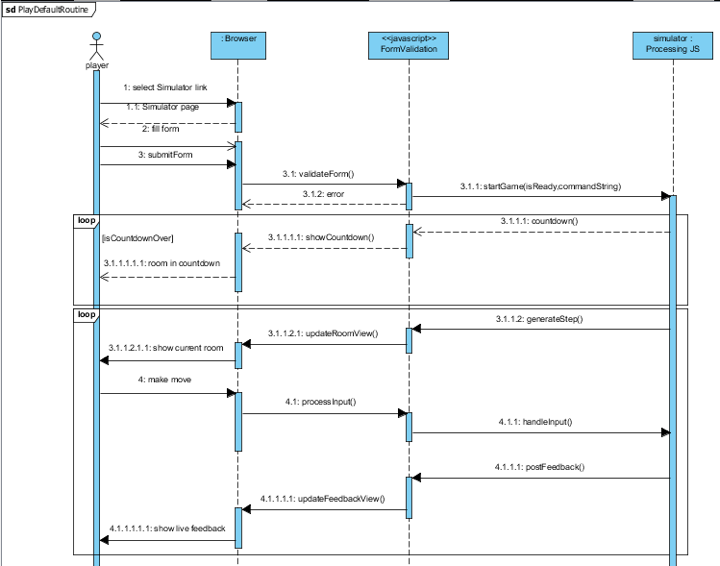


Figure # 16 - Play Default Routine

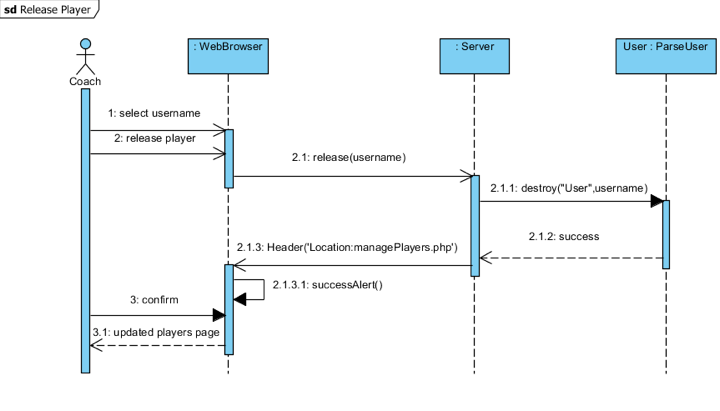


Figure # 17 - Release Player

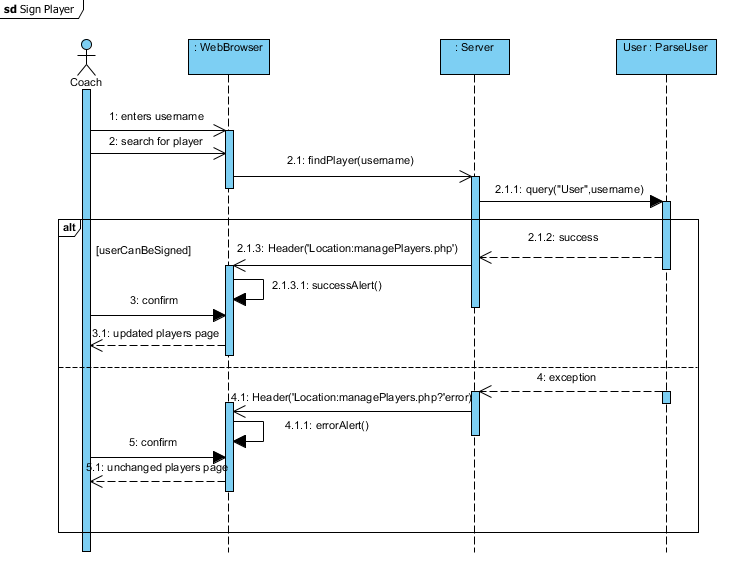


Figure # 18 - Sign Player

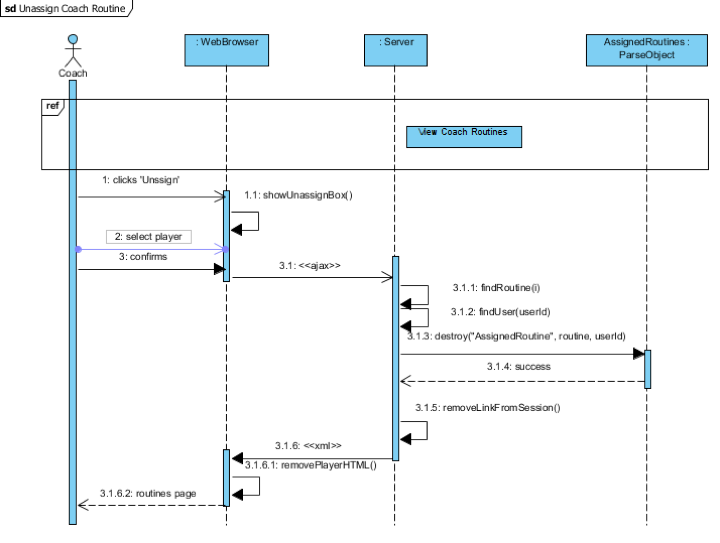


Figure # 19 - Unassign Coach Routine

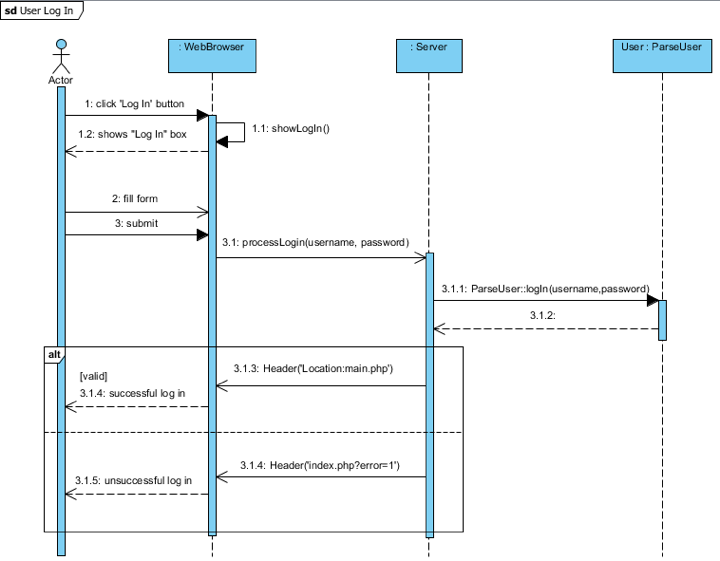


Figure # 20 - User Log In

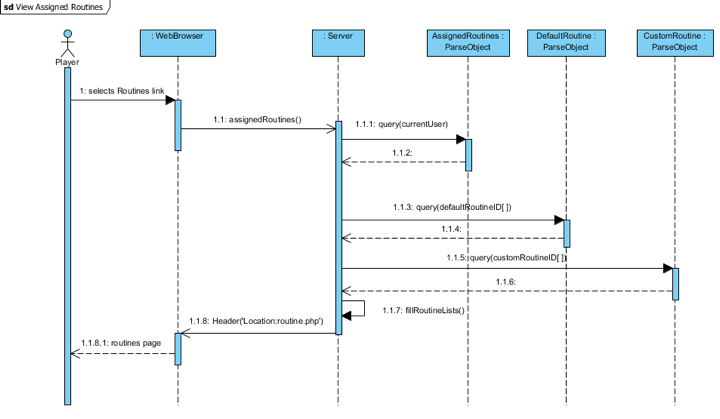


Figure # 21 - View Assigned Routines

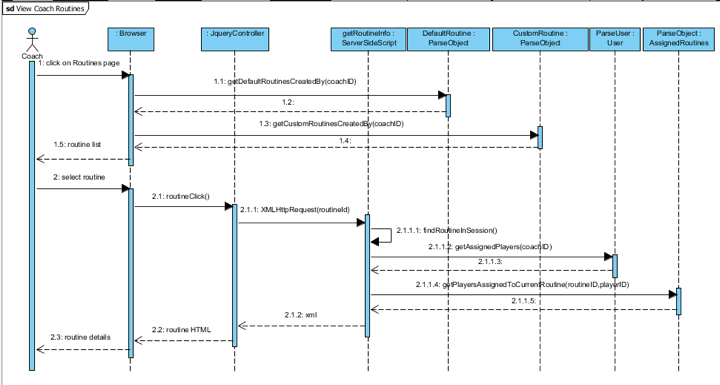


Figure # 22 - View Coach Routines

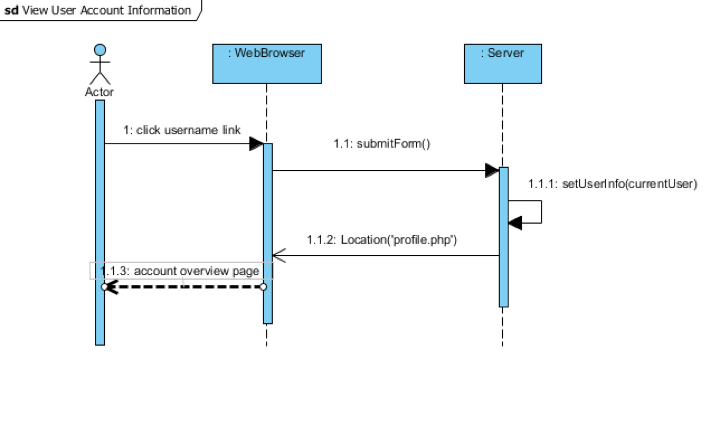


Figure # 23 - View User Account Information

7.2 Appendix B - User Interface Design

Include screenshots of the user interface of your system. For new versions of existing sytems, include only screenshots of the new or modified aspects of the user interface.

There’s no need for introducing this section.

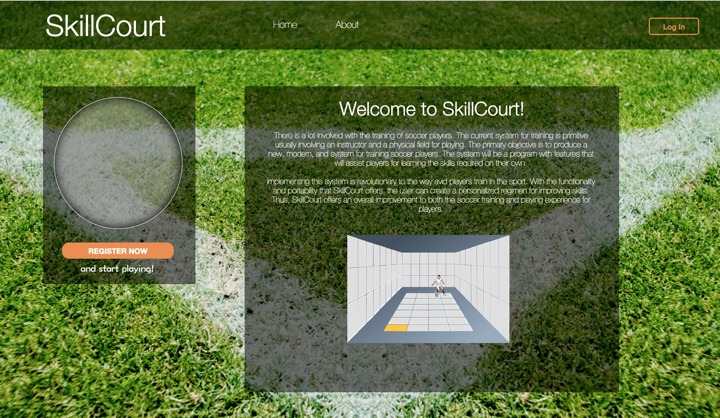


Figure # 24 - Main Page

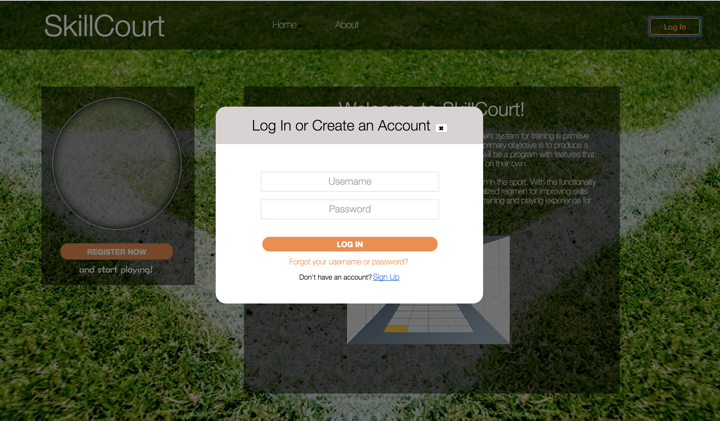


Figure # 25 - Login Page

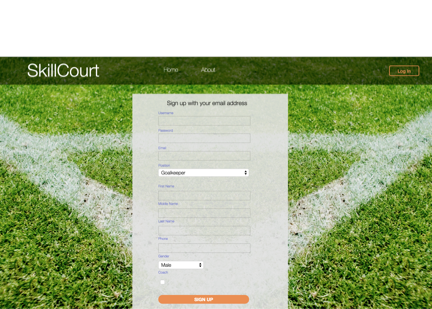


Figure # 26 - Sign Up Page

## 

Figure # 27 - Password Reset Page

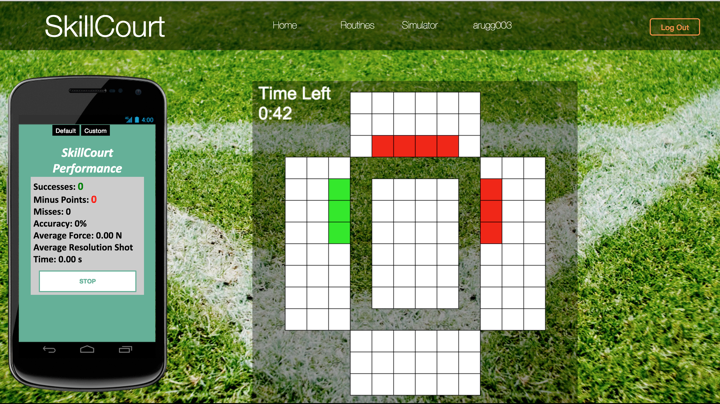


Figure # 28 - Online Simulator

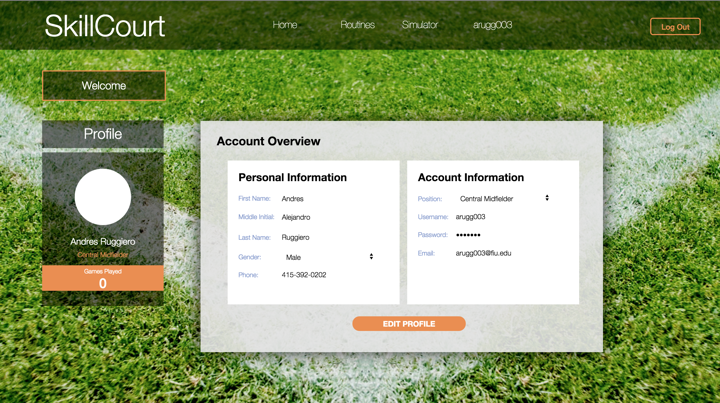


Figure #29 - Account Information Page

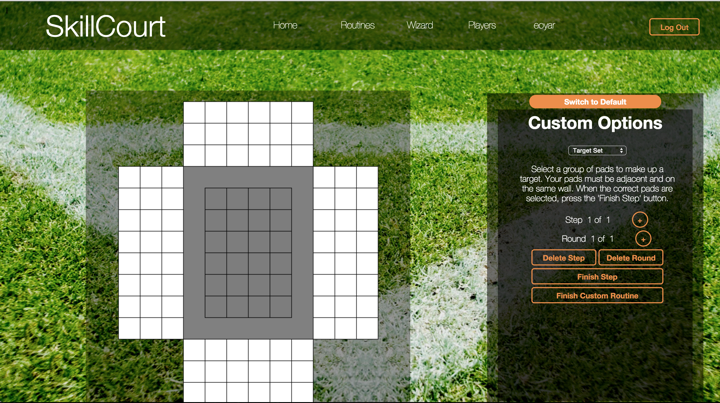


Figure #30 - Online Wizard

7.3 Appendix C - Sprint Review Reports

**Sprint 1 Report**

**Date:** May 29, 2015

**Attendees:** Masoud Sadjadi, Jaime Borras, Gummi Traustason, Erico Oyarzun and Andres Ruggiero

**Discussed Topics:**

The initial goal for Sprint 1 was to implement 4 user stories. After evaluating the user story “Add simulator to the website”, we realized that it may take more development time than it was already planned for. In order to add the simulator to the website, we had to rebuild it in a Javascript based language, Processing.js. The user story “Add Simulator to Website” was not 100% complete, but we were able to complete “Implement Routines”.

Due to this setback, the product backlog was modified. Three user stories “Add simulator to Website”, “Enhance Simulator” and “Enhance Mobile Application” were added to the product backlog.

**Sprint 2 Report**

**Date:** June 12, 2015

**Attendees:** Masoud Sadjadi, Jaime Borras, Gummi Traustason, Erico Oyarzun and Andres Ruggiero

**Discussed Topics:**

The initial goal for Sprint 2 was to implement five user stories. The first three were pending user stories carried over from sprint 1, and the remaining two were new user stories. One of the users stories, “Enhance Mobile Application”, was not completed during this sprint due to a miscalculation on the total development time. This user story was sent back to the product backlog.

**Sprint 3 Report**

**Date:** June 26, 2015

**Attendees:** Masoud Sadjadi, Jaime Borras, Gummi Traustason, Erico Oyarzun and Andres Ruggiero

**Discussed Topics:**

The purpose of Sprint 3 was to complete a SkillCourt Custom Wizard, allowing coaches the ability to create a custom version of pre defined routines and assign them to players. The complexity of this user story was not considered properly, causing miscalculations in development time. The results of Sprint 3 were four user stories, which functionality wasn’t was the product owners expected. Due to this setback, new user stories were created to fulfil the product owner's’ needs.

**Sprint 4 Report**

**Date:** July 10, 2015

**Attendees:** Masoud Sadjadi, Jaime Borras, Gummi Traustason, Erico Oyarzun and Andres Ruggiero

**Discussed Topics:**

The goal for Sprint 4 was to complete the new SkillCourt Custom Wizard, as well as allowing players to play the custom routines. It also included the integration of live feedback to the mobile application. The first four stories were implemented, and as a pending user story, “Integrate Live Feedback to Mobile App”, was sent back to the product backlog.

**Sprint 5 Report**

**Date:** July 23, Year

**Attendees:** Masoud Sadjadi, Jaime Borras, Gummi Traustason, Erico Oyarzun and Andres Ruggiero

**Discussed Topics:**

Sprint 5 was dedicated to integrate the development of the simulator and the online wizard to the website, as well as host the website using Amazon Web Services to provide public access to SkillCourt’s proof of concept. The user stories for this sprint were completed, and the functionality was shown to the product owners. No changes were made to the product backlog.

7.4 Appendix D - Sprint Retrospective Reports

**Sprint 1 Retrospective**

**Date:** May 29, 2015

**Attendees:** Andres Ruggiero and Erico Oyarzun

**Discussed Topics:**

The Sprint 1 was our first opportunity to start looking at the previous developed software, and get an idea on what new features the product owners required. Our planning was not accurate, resulting in the miscalculation of development time of some user stories. For the next coming sprints, the team agreed that each user story had to be carefully revised and planned before accepting them.

**Sprint 2 Retrospective**

**Date:** June 12, 2015

**Attendees:** Andres Ruggiero and Erico Oyarzun

**Discussed Topics:**

Sprint 2 was very productive, and most of the user stories were retired successfully. The team dedicated the proper balance between planning and coding. The only setback was the miscalculation of the time calculated to learn Android Development, which describes one of the user stories. For future sprints the team agreed that research and learning time for new technologies has to be well defined.

**Sprint 3 Retrospective**

**Date:** May 29, 2015

**Attendees:** Andres Ruggiero and Erico Oyarzun

**Discussed Topics:**

The development time for Sprint 3 stories were not calculated properly. The team encountered some issues that caused some stories to be incomplete, setting them back to the product backlog. For next sprint the team agreed to work more closely with the product owners to design more detailed user stories.

**Sprint 4 Retrospective**

**Date:** Jun 10, 2015

**Attendees:** Andres Ruggiero and Erico Oyarzun

**Discussed Topics:**

The stories for Sprint 4 were carefully thought and planned. The team had a proper balance between coding and designing a long-lasting solution for the custom routine wizard. Unfortunately, the Android learning curve caused one user story to be sent back to the product backlog. The team agreed that technologies that required more than 4 hours or learning time are not going to be taken as user stories.

**Sprint 5 Retrospective**

**Date:** Jun 24, 2015

**Attendees:** Andres Ruggiero and Erico Oyarzun

**Discussed Topics:**

The user stories for Sprint 5 were all completed, without affecting the product backlog. But some of the user stories contained small bugs. The team believes that this happened because testing time was not properly allocated to some user stories. Developers agree that planning should consider the proper testing time for each component.

7.5 Appendix E - Use Case

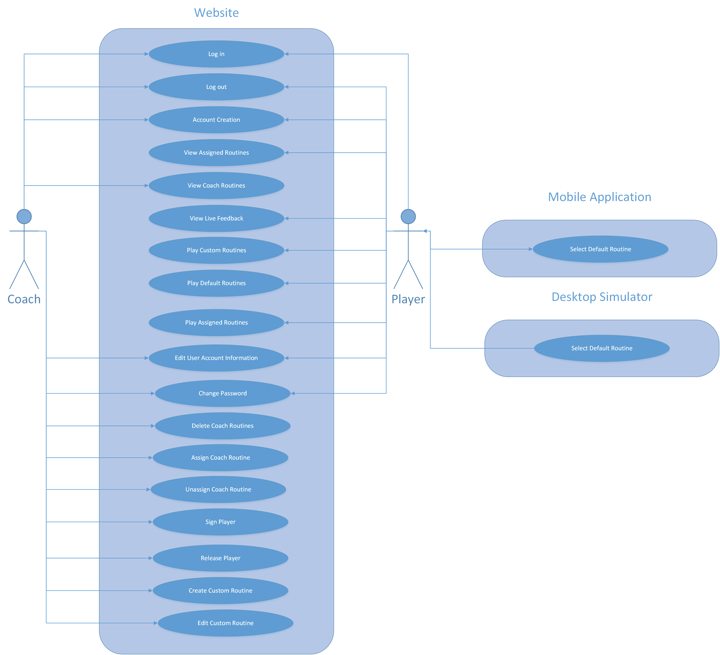


Figure #31 - Account Information Page

7.5.1 Online Simulator

|  |  |
| --- | --- |
| **Use Case: Play Default Routines** | **ID:** |
| **Actor: Player User** | |
| **Description: A Player User chooses and configures a default routine, then plays it** | |
| **Trigger: Player User selects ‘Simulator’ link** | |
| **Preconditions:**  **1. Player User is logged in** | |
| **Normal Course:**  **1.0 Player User selects ‘Simulator’ link**  **1.1 System displays ‘Simulator’ page**  **1.2 User fills in options to their liking**  **1.3 User clicks ‘Play!’ button**  **1.4 User plays chosen routine until game is over** | |
| **Post Conditions:**  **1. Statistics and feedback is available for user on ‘SkillCourt Performance’ screen** | |
| **Exceptions:**  **E1.0: User has ‘Time Per Round’ checkbox selected but leaves input blank or with invalid input (occurs at Normal Course 1.2)**  **1.1 System displays ‘Please fill out all options properly’ message.**  **2.1 Return to Normal Course 1.2**  **E2.0: User has leaves ‘Play By’ input blank or with invalid input (occurs at Normal Course 1.2)**  **1.1 System displays “Please fill out all options properly” message.**  **2.1 Return to Normal Course 1.2** | |

|  |  |
| --- | --- |
| **Use Case: Play Custom Routine** | **ID:** |
| **Actor: Player User** | |
| **Description: A Player User chooses a custom routine from their assigned list and plays them.** | |
| **Trigger: While on ‘Simulator’ page, a player selects the ‘Custom’ option** | |
| **Preconditions:**  **1. A Player User is logged in**  **2. The Player User is in the ‘Simulator’ page**  **3. The Player User has Custom Routines assigned** | |
| **Normal Course:**  **1.0 Player selects ‘Custom’ option**  **1.1 System displays list of user’s Custom Routines**  **1.2 User selects routine and clicks ‘Play!’**  **1.3 User plays routine until game is over** | |
| **Post Conditions:**  **1. User can choose to play Custom or Default routines on ‘Simulator’ page** | |

# 7.5.2 Website

|  |  |
| --- | --- |
| **Use Case: View Coach Routines** | **ID:** |
| **Actor: Coach User** | |
| **Description: A Coach User views routines they have created** | |
| **Trigger: Coach User selects** | |
| **Preconditions:**  **1. Coach User is logged in** | |
| **Normal Course:**  **1.0 Coach User selects ‘Routines’ navigation link**  **1.1 System displays ‘Coach Routines’ page**  **1.2 User selects a routine from list**  **1.3 System displays unique details depending on routine** | |
| **Post Conditions:**  **1. Coach can follow up with Assign, Unassign, Delete, and Edit options** | |

|  |  |
| --- | --- |
| **Use Case: User Log In** | **ID:** |
| **Actor: Coach User, Player User** | |
| **Description: A User logs in to the website** | |
| **Trigger: User selects ‘Log In’ on homepage** | |
| **Preconditions:**  **1. A User is registered with the website** | |
| **Normal Course:**  **1.0 User selects ‘Log In’ on homepage**  **1.1 System displays ‘Log In’ fields box**  **1.2 User fills and submits ‘Log In’ form**  **1.3 System validates credentials**  **1.4 System redirects user to ‘Main’ page** | |
| **Post Conditions:**  **1. User is logged in** | |
| **Exceptions:**  **E1.0: User enters invalid ‘Log In’ credentials (occurs at Normal Course 1.3)**  **1.1 System displays ‘Invalid Log In Credentials’ message.**  **2.1 Return to Normal Course 1.1** | |

|  |  |
| --- | --- |
| **Use Case: View User Account Information** | **ID:** |
| **Actor: Player User, Coach User** | |
| **Description: User views their account information** | |
| **Trigger: User clicks on their username on the navigation bar from any page** | |
| **Preconditions:**  **1. User is logged in** | |
| **Normal Course:**  **1.0 User selects the profile link on navigation bar**  **1.1 System displays ‘Account Overview’ information box** | |
| **Post Conditions:**  **1. The user’s personal and account information is displayed** | |

|  |  |
| --- | --- |
| **Use Case: View Assigned Routines** | **ID:** |
| **Actor: Player User** | |
| **Description: A Player User views their assigned custom and default routines** | |
| **Trigger: Player User selects the ‘Routines’ link on the navigation bar from any page** | |
| **Preconditions:**  **1. A Player User is logged in** | |
| **Normal Course:**  **1.0 A Player User selects ‘Routines’ from the navigation bar**  **1.1 System displays routine lists with Player User’s assigned routines** | |
| **Post Conditions:**  **1. Player User can view their assigned routines** | |

|  |  |
| --- | --- |
| **Use Case: View Live Feedback of Current Routine** | **ID:** |
| **Actor: Player User** | |
| **Description: A Player User views the feedback based on their performance on a current game** | |
| **Trigger: User performs the “Play Default Routine” Use Case** | |
| **Preconditions:**  **1. Player User is currently logged in**  **2. User is currently playing a Default Routine** | |
| **Normal Course:**  **1.0 User performs the “Play Default Routine” use case**  **1.1 Player User plays routine**  **1.2 System displays live feedback** | |
| **Post Conditions:**  **1. Feedback for user is displayed** | |

|  |  |
| --- | --- |
| **Use Case: Play Assigned Routines** | **ID:** |
| **Actor: Player User** | |
| **Description: A Player User plays routines assigned to them** | |
| **Trigger: A Player User performs the ‘View Assigned Routines’ use case** | |
| **Preconditions:**  **1. A Player User is logged in**  **2. User has routines assigned to them** | |
| **Normal Course:**  **1.0 A Player User performs the ‘View Assigned Routines’ use case**  **1.1 System displays assigned routines to Player User**  **1.2 User selects a routine to play and clicks on ‘Play’**  **1.3 System displays ‘Simulator’ page with filled in**  **1.4 User clicks ‘Play!’ to start chosen routine** | |
| **Post Conditions:**  **1. Player User plays a chosen assigned routine** | |

|  |  |
| --- | --- |
| **Use Case: Edit User Account Information** | **ID:** |
| **Actor: Player User, Coach User** | |
| **Description: A User edits their account information** | |
| **Trigger: A User performs the “View User Account Information” use case** | |
| **Preconditions:**  **1. User is logged in** | |
| **Normal Course:**  **1.0 User performs the “View User Account information” use case**  **1.1 System displays ‘Account Overview’**  **1.2 User clicks on ‘Edit Profile’ button**  **1.3 System allows User to edit ‘Account Overview’**  **1.4 User edits preferred account details and clicks on ‘Save’** | |
| **Post Conditions:**  **1. System displays updated ‘Account Overview’ page** | |
| **Exceptions:**  **E1.0: User enters invalid ‘Log In’ credentials (occurs at Normal Course 1.3)**  **1.1 System displays ‘Invalid Log In Credentials’ message.**  **2.1 Return to Normal Course 1.1** | |

|  |  |
| --- | --- |
| **Use Case: Change Password** | **ID:** |
| **Actor: Player User, Coach User** | |
| **Description: A User changes their password to a new one** | |
| **Trigger: User selects ‘Forgot Username or Password?’ link when logging in** | |
| **Preconditions:**  **1. User knows email address associated with account** | |
| **Normal Course:**  **1.0 User selects ‘Forgot Username or Password?’ link at Log In**  **1.1 System displays email field for User**  **1.2 User enters email and submits form**  **1.3 User follows link on email sent from ‘SkillCourt Password Reset’**  **1.4 System displays new password form**  **1.5 User completes and submits form** | |
| **Post Conditions:**  **1. User has new password** | |
| **Exceptions:**  **E1.0: User enters invalid email (occurs at Normal Course 1.2)**  **1.1 System displays ‘Invalid Credentials‘ message**  **2.1 Return to Normal Course 1.1** | |

|  |  |
| --- | --- |
| **Use Case: Delete Coach Routine** | **ID:** |
| **Actor: Coach User** | |
| **Description: A Coach User deletes a Default or Custom routine they have created** | |
| **Trigger: Coach User clicks ‘Delete’ button with routine selected** | |
| **Preconditions:**  **1. Coach User is logged in**  **2. User is on ‘Routines’ page**  **3. User must have routines created**  **4. Selected routine is not assigned to a player** | |
| **Normal Course:**  **1.0 Coach clicks ‘Delete’ button with routine selected**  **1.1 System displays confirmation alert to delete routine**  **1.2 User confirms deletion**  **1.3 System displays ‘Routines’ page with deleted routine missing** | |
| **Post Conditions:**  **1. List of routines has the desired routine missing** | |

|  |  |
| --- | --- |
| **Use Case: Assign Coach Routine** | **ID:** |
| **Actor: Coach User** | |
| **Description: A Coach User assigns a default or custom routine they have created to one or more of their signed players** | |
| **Trigger: Coach User clicks the ‘Assign’ button while a routine is selected** | |
| **Preconditions:**  **1. Coach User is logged in**  **2. User is on ‘Routines’ page**  **3. User must have routines created**  **4. Routine must not be assigned desired player(s)** | |
| **Normal Course:**  **1.0 Coach User clicks the ‘Assign’ button while a routine is selected**  **1.1 System displays list of available players**  **1.2 User selects player(s) to assign the routine to and clicks ‘Assign’** | |
| **Post Conditions:**  **1. Assigned player list for routine has new players** | |

|  |  |
| --- | --- |
| **Use Case: Unassign Coach Routines** | **ID:** |
| **Actor: Coach User** | |
| **Description: A Coach User unassigns a routine from a player** | |
| **Trigger: Coach User clicks the ‘Unassign’ button while a routine is selected** | |
| **Preconditions:**  **1. Coach User is logged in**  **2. User is on ‘Routines’ page**  **3. User has created routines and assigned them to players** | |
| **Normal Course:**  **1.0 Coach User clicks the ‘Unassign’ button**  **1.1 System displays list of players assigned to selected routine**  **1.2 User selects a player and clicks ‘Unassign’** | |
| **Post Conditions:**  **1. Assigned player list for the routine has player removed** | |
| **Use Case: Sign Player** | **ID:** |
| **Actor: Coach User** | |
| **Description: A Coach User signs a player to themselves** | |
| **Trigger: Coach selects ‘Players’ link on the navigation bar** | |
| **Preconditions:**  **1. A Coach User is logged in** | |
| **Normal Course:**  **1.0 Coach selects ‘Players’ link on the navigation bar**  **1.1 System displays ’Players’ page**  **1.2 User submits the username of the desired player under “Search for a player to sign”**  **1.3 System displays alert notifying of successful signing** | |
| **Post Conditions:**  **1. The Coach User is now the assigned Coach to the selected player** | |
| **Exceptions:**  **E1.0: User enters nonexistent username or username to a player already signed by another Coach User(occurs at Normal Course 1.2)**  **1.1 System displays appropriate message, waits for confirmation**  **2.1 Return to Normal Course 1.2** | |

|  |  |
| --- | --- |
| **Use Case: Release Player** | **ID:** |
| **Actor: Coach User** | |
| **Description: A Coach User releases a signed player** | |
| **Trigger: Coach selects ‘Players’ link on the navigation bar** | |
| **Preconditions:**  **1. A Coach User is logged in** | |
| **Normal Course:**  **1.0 Coach selects ‘Players’ link on the navigation bar**  **1.1 System displays ’Players’ page**  **1.2 User selects the username of the player from the list under “Search for a player to release”**  **1.3 System displays alert notifying of successful release** | |
| **Post Conditions:**  **1. The Coach User is no longer attached to the released player** | |

# 

# 

# 7.5.3 Online Wizard

|  |  |
| --- | --- |
| **Use Case: Create Custom Routine** | **ID:** |
| **Actor: Coach User** | |
| **Description: A Coach User creates a custom routine** | |
| **Trigger: Coach User clicks on ‘Wizard’ link on navigation bar** | |
| **Preconditions:**  **1. Coach User is logged in** | |
| **Normal Course:**  **1.0 Coach User clicks on ‘Wizard’ link**  **1.1 System displays the ‘SkillCourt Wizard’ page**  **1.2 User selects desired options to design routine and clicks on ‘Finish Custom Routine’**  **1.3 System saves routine and redirects Users to ‘Routines’ page** | |
| **Post Conditions:**  **1. Newly created routine is viewable in ‘Routines’ page** | |
| **Exceptions:**  **E1.0: User attempts to Finish Routine with incomplete steps or rounds (occurs at Normal Course 1.2)**  **1.1 System warns user to complete steps and rounds**  **2.1 Return to Normal course 1.2** | |

|  |  |
| --- | --- |
| **Use Case: Edit Custom Routine** | **ID:** |
| **Actor: Coach User** | |
| **Description: A Coach User edit a custom routine** | |
| **Trigger: Coach User clicks on ‘Edit’ button in ‘Routines’ page** | |
| **Preconditions:**  **1. Coach User is logged in**  **2. Coach has created Custom Routines** | |
| **Normal Course:**  **1.0 Coach User clicks ‘Edit’ with custom routine selected**  **1.1 System displays the ‘SkillCourt Wizard’ page with routine filled in**  **1.2 User selects desired options to edit routine and clicks on ‘Finish Editing Routine’**  **1.3 System saves routine and redirects Users to ‘Routines’ page** | |
| **Post Conditions:**  **1. Edited routine is viewable in ‘Routines’ page** | |
| **Exceptions:**  **E1.0: User attempts to Finish Editing Routine with incomplete steps or rounds (occurs at Normal Course 1.2)**  **3.1 System warns user to complete steps and rounds**  **4.1 Return to Normal course 1.2** | |

7.5.4 Desktop Simulator

|  |  |
| --- | --- |
| **Use Case: Play Default Routine** | **ID:** |
| **Actor: Guest User** | |
| **Description: A Guest User plays a default routine on the desktop simulator** | |
| **Trigger: Guest User performs the “Select Default Routine” use case** | |
| **Preconditions:**  **1. User has already selected routine options** | |
| **Normal Course:**  **1.0 Guest User performs the “Select Default Routine” use case**  **1.1 User taps ‘Play!’**  **1.2 System begins countdown on simulator followed by routine simulation**  **1.3 User plays routine until game is over** | |
| **Post Conditions:**  **1. Routine is over and must be reset** | |

7.5.5 Mobile Application

|  |  |
| --- | --- |
| **Use Case: Select Default Routine** | **ID:** |
| **Actor: Guest User** | |
| **Description: A Guest User selects default routine options** | |
| **Trigger: User taps ‘Start Playing’ Home screen** | |
| **Preconditions:**  **1. Bluetooth connection has been made** | |
| **Normal Course:**  **1.0 User taps ‘Start Playing’ on Home screen**  **1.1 System displays SkillCourt Routines Screen**  **1.2 User fills out options and taps ‘Play!’** | |
| **Post Conditions:**  **1. Desktop simulator begins simulating routine** | |

8. References

8.1 For the Website

<http://www.w3schools.com/>

8.2 For the Processing.js Sketck

<http://processingjs.org/reference/>