Skillcourt Backend

Design Document

Andy Martinez & Matthew Santiago

Professor Masoud Sadjadi

Florida International University

Senior Project – Spring 2015

## Legal Notices

Copyright © by Florida International University – Senior Project – SkillCourt Backend

All rights reserved. No part of the SkillCourt Backend Project may be reproduced or transmitted in any form or by any means without prior written consent.

## 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 which is easy to understand and maintain.

Table of Contents

[Legal Notices i](#_Toc418169593)

[Abstract ii](#_Toc418169594)

[Table of Contents iii](#_Toc418169595)

[1. Introduction 1](#_Toc418169596)

[1.1 Problem Definition 1](#_Toc418169597)

[1.2 Design Methodology 1](#_Toc418169598)

[1.3 Terminology 2](#_Toc418169599)

[1.4 Overview of Document 2](#_Toc418169600)

[2. System Design 3](#_Toc418169601)

[2.1. Overview 4](#_Toc418169602)

[2.2. Subsystem Decomposition 5](#_Toc418169603)

[2.3. Hardware and Software Mapping 7](#_Toc418169604)

[2.4. Persistent Data Management 7](#_Toc418169605)

[2.5. Security and Privacy 8](#_Toc418169606)

[3. Detailed Design 10](#_Toc418169607)

[3.1. Overview 10](#_Toc418169608)

[3.2. Static Model 12](#_Toc418169609)

[3.3. Dynamic Model 12](#_Toc418169610)

[3.4. Code Specification 19](#_Toc418169611)

[4. Glossary 20](#_Toc418169612)

[5. Appendix 21](#_Toc418169613)

[5.1. Appendix A – Use case diagram 21](#_Toc418169614)

[5.2. Appendix B – Implemented Use Cases 22](#_Toc418169615)

[5.3. Appendix C – Detailed Class interfaces 39](#_Toc418169616)

[5.4. Appendix D - Diary of Meetings and tasks 40](#_Toc418169617)

[6. References 43](#_Toc418169618)

## **Introduction**

Soccer has become a world-wide sporting phenomena with events like the World Cup massing viewers of over 900 million and over 200 teams participating in the games that lead up to it. On a much smaller level, soccer is not surprisingly a favorite past-time for countless people of all ages. Due to its popularity as a sport, there is a large demand for guidance and coaching for becoming better.

### Problem Definition

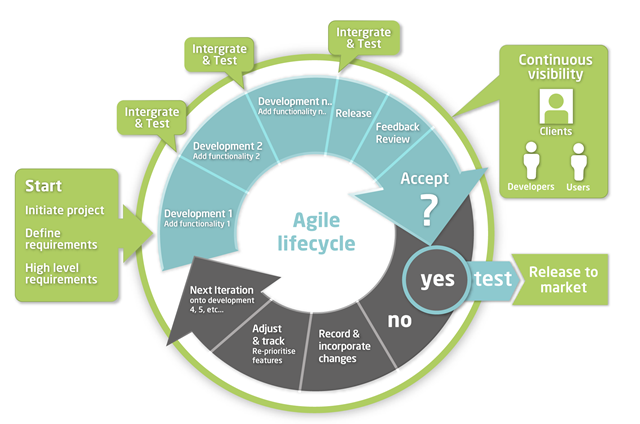
As it stands, training for soccer can be a very cumbersome feat for the average person to endeavor unaided. Without proper guidance, progression dwindles and a person may not feel obliged to continue. With so few places to turn outside of professional help, a new method is needed to fill the void.

SkillCourt is both a new method and an aid to soccer coaching. Offering the routines for honing the cognitive skills one needs for soccer, SkillCourt can be an all-in-one self-trainer for anyone. While SkillCourt alone can be a great method for training, coaches can also join and receive information vital to furthering their players’ progression.

### Design Methodology

Agile development methodology was used to begin the process of creating and adding functionality to all parts of the system. Agile is an iterative and incremental development process which allows developing functionalities and testing it along the way. Following this process was key to delivering a quality product that conforms to all standards.

Since the process is iterative, it allows small and concise changes to be made leading to the overall progression of the project as a whole. Below is the methodology diagram used for the completion of the product:

 ***Figure 1*** *Agile Methodology*

For SkillCourt, we used Mingle, a ThoughtWorks agile/scrum service, to track and document the project’s progress and requirements. Mingle allowed for us to break the development process into smaller, digestible, and easy to integrate pieces.

### Terminology

**Pad Simulator**: A virtual device which will take the place of physical SkillCourt pads for testing showcasing purposes. This device offers all of the features a SkillCourt Arena will offer.

**SkillCourt**: A system which uses SkillCourt Pads and a player interface for training soccer.

**SkillCourt Arena**: A 20’x20’ room with SkillCourt Pads on the walls used for SkillCourt routines.

**SkillCourt Pad**: A physical device with a flat surface that can measure and transmit when and how much pressure it received.

**Player**:

**Coach**:

### Overview of Document

In chapter 1, the main problem is introduced, along with the design methodology used and definitions relating to the project. In chapter 2, the system is introduced in terms of the system’s architecture, with the subsystem decomposition, hardware and software mapping, persistent data management and privacy/security aspects explained. Chapter 3 delves into the behavior of each subsystem described and the static and dynamic models used are explained. Chapter 4 is a glossary of domain-specific terms. In the appendix, miscellaneous material such as use case diagrams, use cases being implemented, and documented class interfaces can be found. Lastly, a diary of meetings are references can be found at the end of the document.

## **System Design**

This chapter gives insight into the system’s architectural patterns used. SkillCourt was divided into subsystems, each one with specific functionality that adds richness to the sports training process. In this chapter, an overview of the system’s design is introduced. Then, the decomposition of the system into subsystems is explained. Moreover, hardware and software mapping and persistent data management aspects of the project are discussed. Finally, the security and privacy issues of the system are explained.

The Mobile Application shall allow:

• players to create a new player account

• players to log in with their account credentials

• players to log in as a guest player

• players to view their account information

• players to change their account information

• players to view their statistics

• players to connect to SkillCourt Pads

• players to select default routines

• players to play default routines

• players to select custom routines

• players to play custom routines

• players to select coach routines

• players to play coach routines

• players to view game score and info

• players to disconnect from pads

• players to log out

The Website shall allow:

• coaches to create a new coach account

• coaches to log in with their account credentials

• coaches to log off

• coaches to view player roster

• coaches to view their players’ info

• coaches to view their players’ statistics

• coaches to view their custom routines

• coaches to create custom routines

• players to log in with their account credentials

• players to log off

• players to create a new player account

• players to view their account info

• players to view their account statistics

• players to edit their account info

• players to create custom routines

• players to view their custom routines

• users to view a public player’s statistics

• users to view a public player’s account info

The Simulator shall allow:

• the app to begin a routine

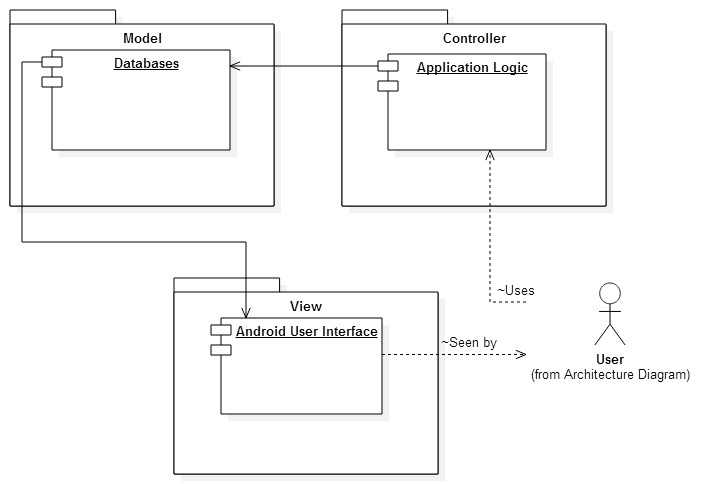
• the app to receive routine statistics

• the app to connect to pads

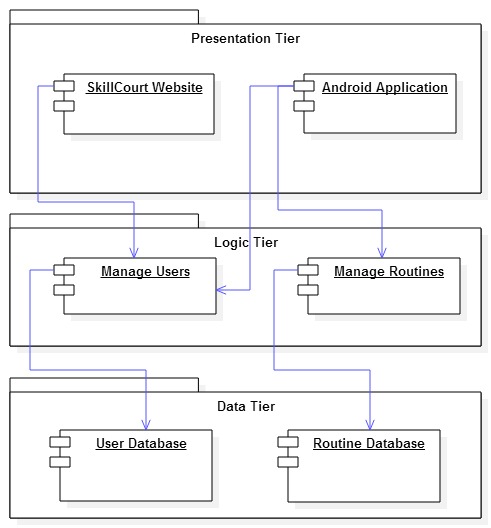
• the player to arrange pads as desired

### Overview

This version of SkillCourt is the first release, so all functionalities resemble work done in Ver. 1. SkillCourt was implemented using the Model-View-Controller architecture.

   
Figure 2 MVC Architecture

Also used by the system is a 3-tier architecture which is used to obtain data from the database. The following diagram demonstrates this architecture:

  
Figure 3 3-tier Architecture

### Subsystem Decomposition

SkillCourt is broken down into 3 systems: The App, the Website, and the Simulator. Each system can further be broken down into subsystems. The following diagrams represent the subsystems for each of the 3 systems.

The SkillCourt app is a mobile application which offers interaction between the player and the simulator as well as the database. As the core UX for the player, the app offers a subsystem for the following:

**Account** **Management**

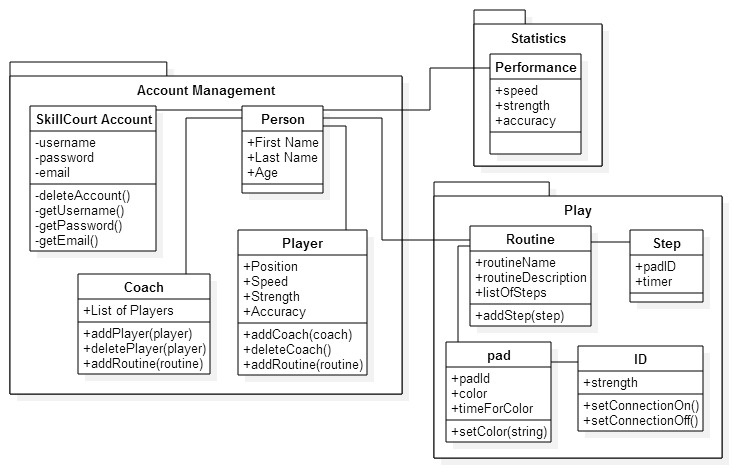
This subsystem gives the player control of all account information associated with the player’s account. This includes personal information, such as first and last name; account information, like username, password, email address; coach information, like coach’s username; and player information, ranging from position played to average statistics.

**Statistics**

As a player, statistics are a way of gauging the level of performance achieved. The statistics subsystem gives access to a range of player’s statistical information throughout the player’s lifetime with SkillCourt.

**Play**

The final subsystem of the app, play gives the player access to the game. It allows the player to send routine information to the simulator and allows the app to receive information about the played routine from the simulator

**   
Figure 4** App Subsystem Decomposition Diagram

The next system is the Website system. This systems is further divided into 3 subsystems. These include Account Management, Player Statistics, and User Routines.

**Account Management**

While this subsystem is similar to the App’s account management subsystem, it differs in that a different set of users has access. Coaches have access to the website, so this subsystem considers coaches as well as players.

**Player Statistics**

As a player, statistics are a way of gauging the level of performance achieved. The statistics subsystem gives access to a range of player’s statistical information throughout the player’s lifetime with SkillCourt.

**User Routines**

As a SkillCourt coach or player, customized variations of the default routines can be created for use by players on the App. This subsystem allows for the creation of new custom user routines which will set the variables for them to the user’s choosing.

The final system for SkillCourt is the simulator. The simulator is a system which emulates the SkillCourt Arena. It is broken down as follows:

**SkillCourt Room**

The SkillCourt room represents the 20’ by 20’ arena which has SkillCourt pads on the walls. This room is represented by a 5x5 square representing the floor and 4 3x5 “walls” surrounding it. Each square on the floor and the walls are a pad.

**Statistics**

As a player, statistics are a way of gauging the level of performance achieved. The statistics subsystem is a generator of the player’s statistics from information obtained while the player interacts with the simulator. Data gathered includes force, time between strikes, number of shots attempted, number of correct targets hit, etc.

**Routine**

The routine subsystem parses information related to a routine sent by the app. The routine information parsed includes the routine type, how long it will be played, the level of difficulty, etc.

### Hardware and Software Mapping

The main components of SkillCourt are the android mobile app, the PHP website, the simulator, and the MySQL database. Both the app and the website are connected to the database and do so using PHP scripts located on the server. All models in each component are mapped to tables in the MySQL environment set up on the same machine. The browser communicated with the server using HTTP.

### Persistent Data Management

#### Coach Information Storage

The system stores the information from the coach when he/she registers to make an account. This information includes the coach’s username, email, password, and some personal information such as first name, last name, and age

#### Player Information Storage

The system stores the information from the player when he/she registers to make an account. This information includes the player’s username, email, password, and some personal information such as first name, last name, and age. Additionally, the system will store the position of the player in the field.

#### Routine Storage

When a player creates a custom routine, the system will store this new routine. It will save to the database the routine name given by the player as he/she creates it, a unique ID created by the system and a string that will be converted to the series of steps in the routine. Additionally, the system will automatically have stored several routines for the player to choose from.

#### Performance Statistics Storage

The performance of the player while using the pads will be stored by the system. Statistics from the performance will include strength, speed, accuracy, etc.

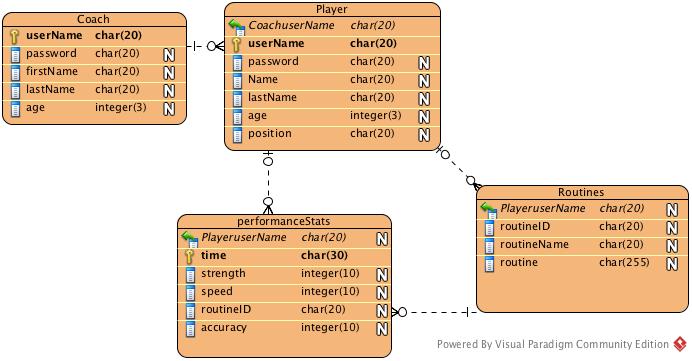


Figure 5: Database table

### Security and Privacy

#### Credential Authentication

The SkillCourt-Backend system will be accessed by players and coaches who have previously registered their credentials by making an account. In the mobile device, only players will be allowed to log in. As they enter the application, the system will provide them with a form to sing in using their previously created username and password. The system will then verify their credentials and verify that they have previously registered as “Players.” Once logged in, players will be restricted to access only their information and statistics. They will not have access to other player’s information.

In the case of coaches, they will not be allowed to sign in through the mobile device application. As they try to log in with their credentials, the system will check that they are not registered as “Player”, but as “Coach” and will return a message stating they cannot log in through the mobile device application. Additionally, users will be allowed to continue on the application as “guest,” but they will have no access to information.

In the webpage, both players and coaches will be allowed to log in by submitting their username and password. In both cases the system will verify that they have registered as players or coaches and will give each the appropriate access to information. Once logged in, players will only be allowed to access their information, while coaches will be allowed to access the information of all players they are connected to. No guests will be allowed on the webpage. In both, the webpage and the mobile application, users will no registered credentials will not be allowed to access any information

#### Data Encryption

Currently, no data is being encrypted. All sensitive data is being stored in plain text in the database.

## **Detailed Design**

This chapter introduces the system in terms of subsystems and the relationships among them. Initially the system is decomposed into several subsystems, each of which is described in terms of behavior and structure. Then the static model is introduced in terms of subsystems and the descriptions of each. After, the dynamic model is presented in terms of sequence diagrams. Finally the class interfaces and constraints for the main control object in each subsystem are presented.

### Overview

**Account Management**

This subsystem gives the player control of all account information associated with the player’s account. This includes personal information, such as first and last name; account information, like username, password, email address; coach information, like coach’s username; and player information, ranging from position played to average statistics.

**Statistics**

As a player, statistics are a way of gauging the level of performance achieved. The statistics subsystem gives access to a range of player’s statistical information throughout the player’s lifetime with SkillCourt.

**Play**

The final subsystem of the app, play gives the player access to the game. It allows the player to send routine information to the simulator and allows the app to receive information about the played routine from the simulator

The next system is the Website system. This systems is further divided into 3 subsystems. These include Account Management, Player Statistics, and User Routines.

While this subsystem is similar to the App’s account management subsystem, it differs in that a different set of users has access. Coaches only have access to the website, so this subsystem considers coaches as well as players.

**Player Statistics**

As a player, statistics are a way of gauging the level of performance achieved. The statistics subsystem gives access to a range of player’s statistical information throughout the player’s lifetime with SkillCourt.

**User** **Routines**

As a SkillCourt coach or player, customized variations of the default routines can be created for use by players on the App. This subsystem allows for the creation of new custom user routines which will set the variables for them to the user’s choosing.

The final system for SkillCourt is the simulator. The simulator is a system which emulates the SkillCourt Arena. It is broken down as follows:

**SkillCourt Room**

The SkillCourt room represents the 20’ by 20’ arena which has SkillCourt pads on the walls. This room is represented by a 5x5 square representing the floor and 4 3x5 “walls” surrounding it. Each square on the floor and the walls are a pad.

**Statistics**

As a player, statistics are a way of gauging the level of performance achieved. The statistics subsystem is a generator of the player’s statistics from information obtained while the player interacts with the simulator. Data gathered includes force, time between strikes, number of shots attempted, number of correct targets hit, etc.

**Routine**

The routine subsystem parses information related to a routine sent by the app. The routine information parsed includes the routine type, how long it will be played, the level of difficulty, etc.

### Static Model

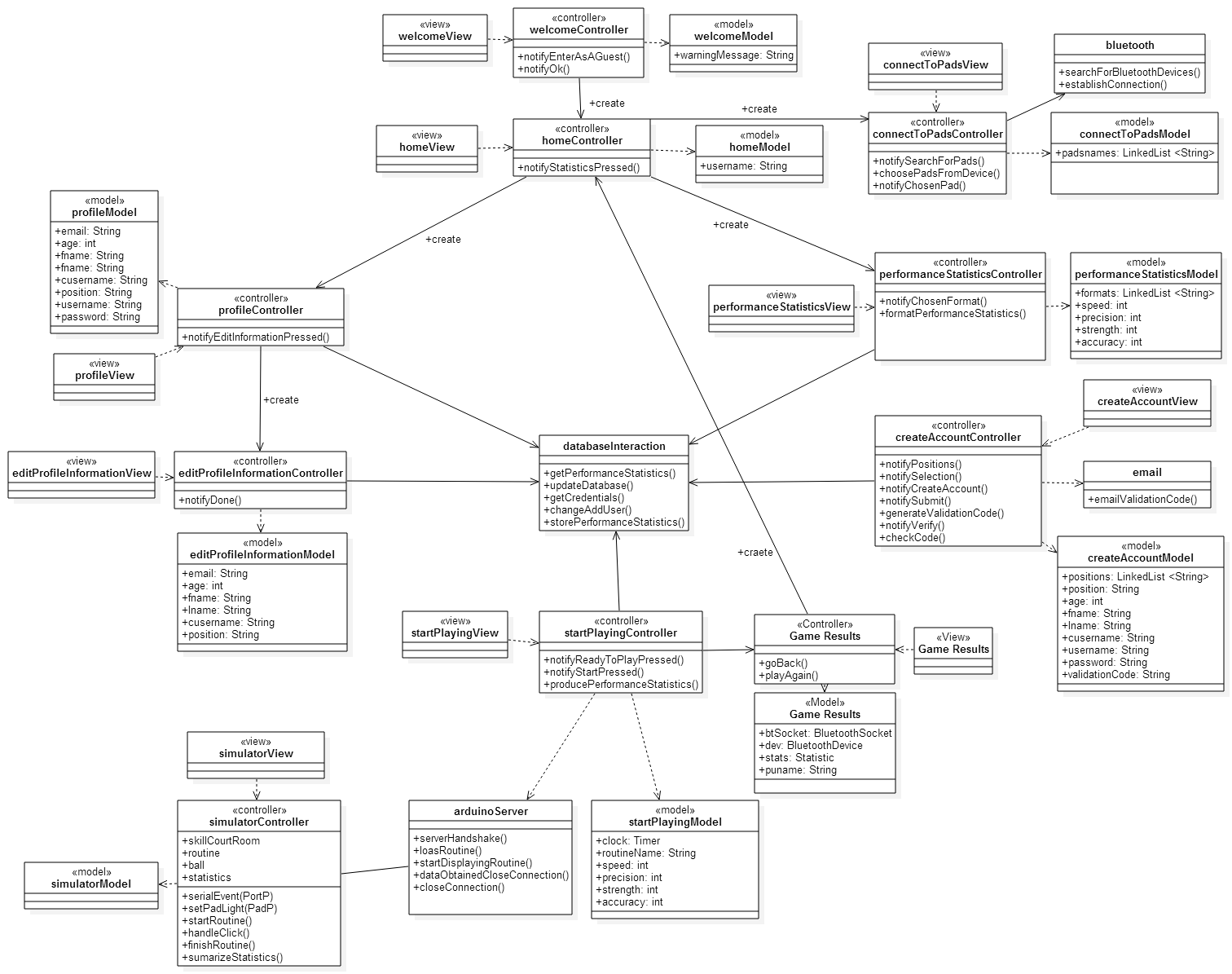
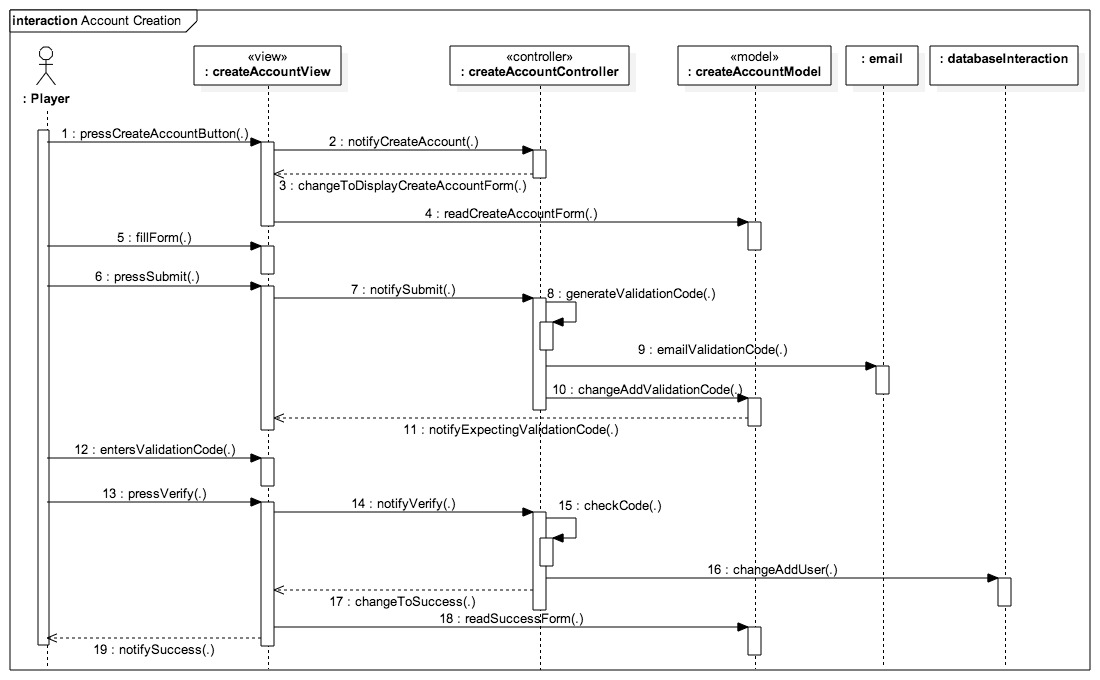
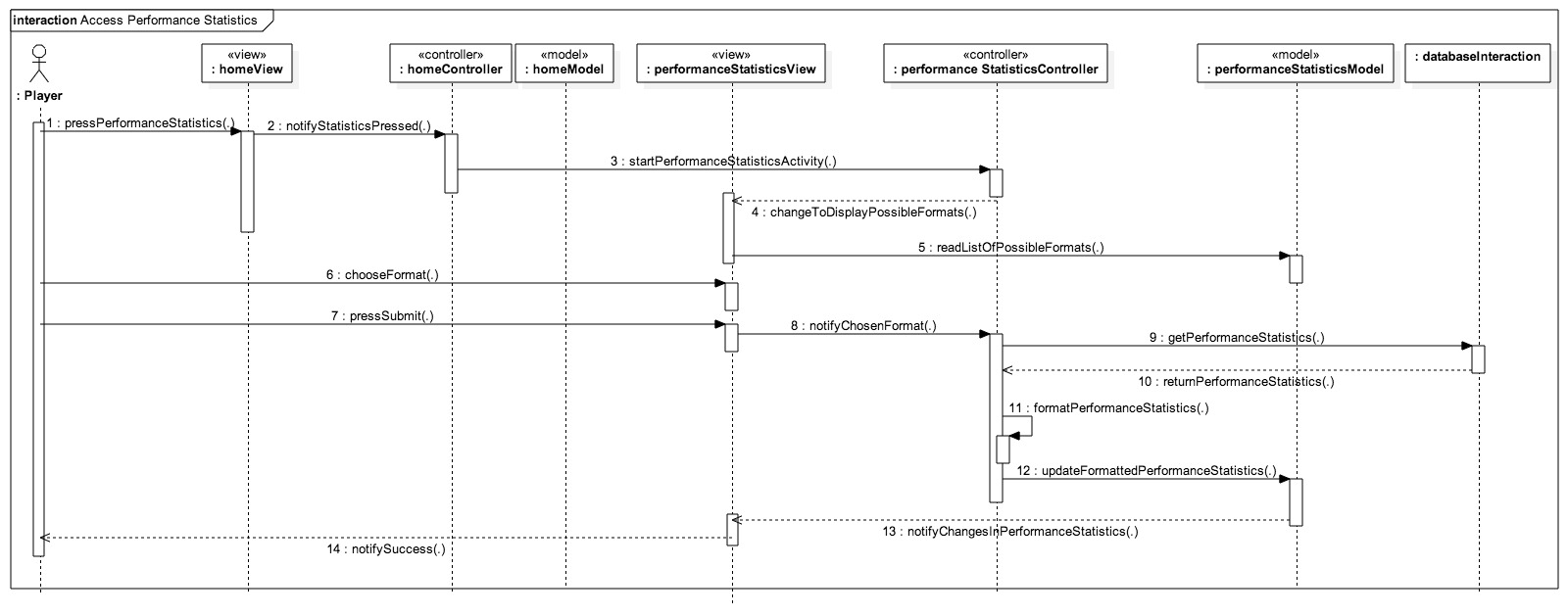
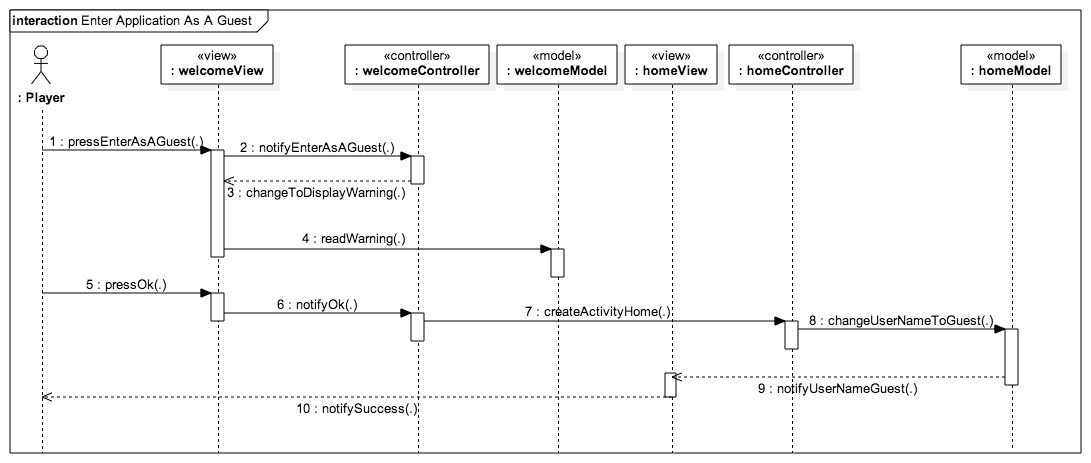


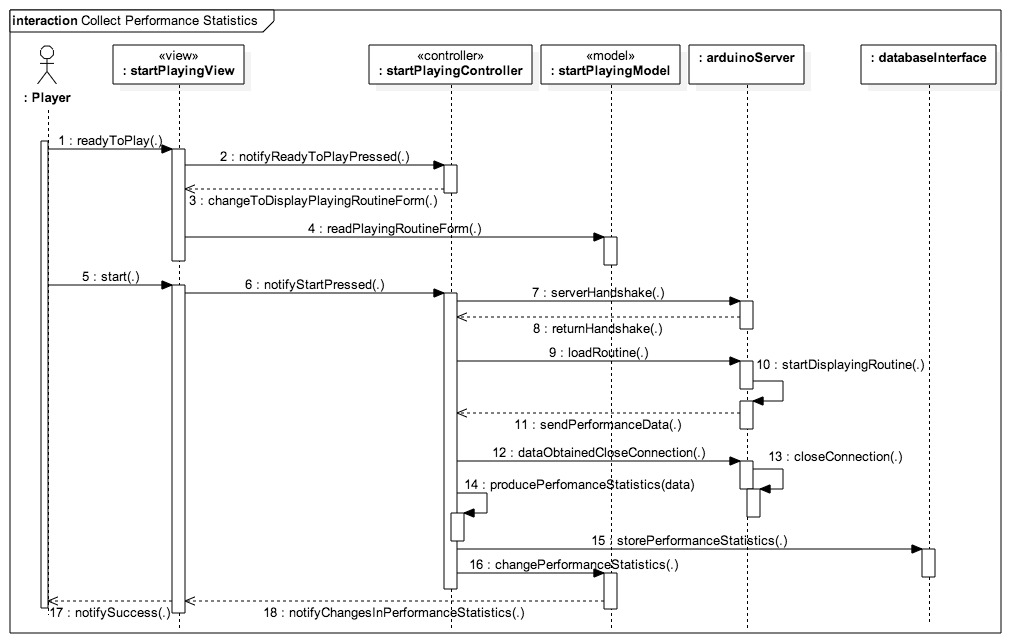
Figure 5 Class Diagram

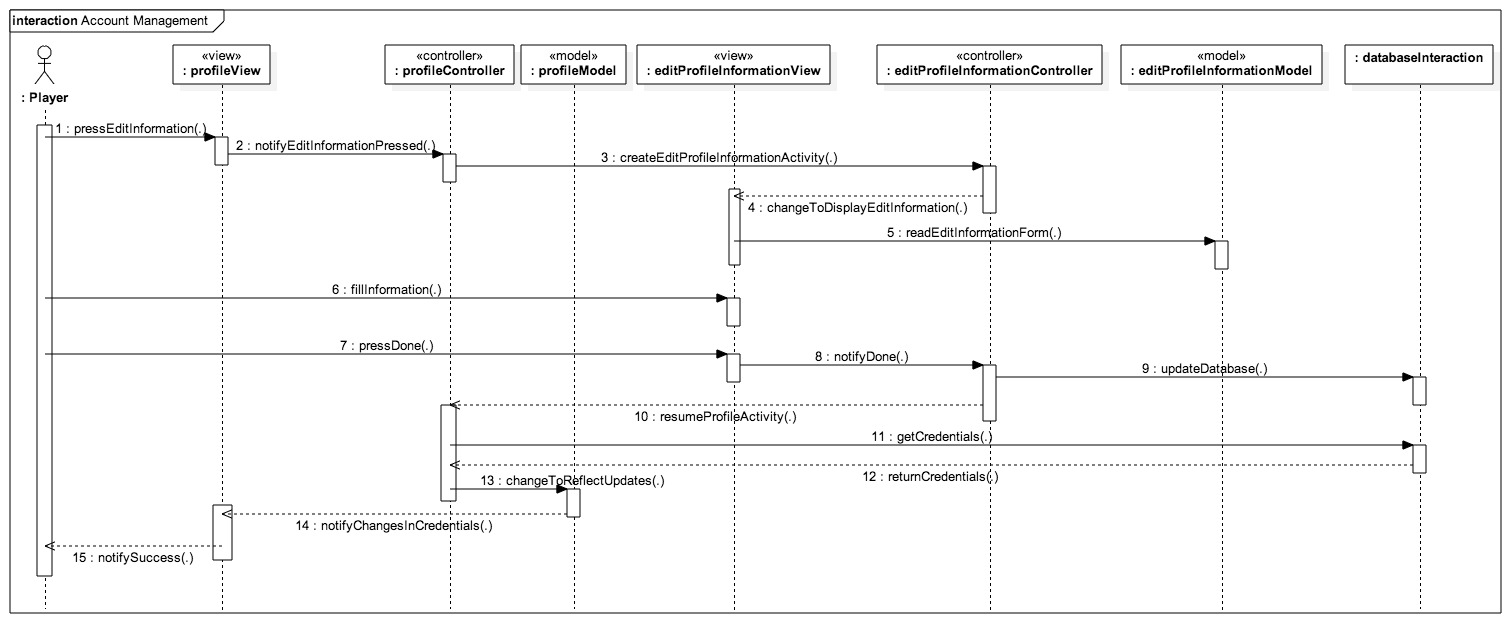
### Dynamic Model

Figure 6 Account Creation Sequence Diagram

****  
Figure 7 Access Player Statistics Sequence Diagram

  
Figure 8 Enter Application as A Guest Sequence Diagram

  
Figure 9 Collect Performance Statistics Sequence Diagram

   
Figure 10 Account Management Sequence Diagram

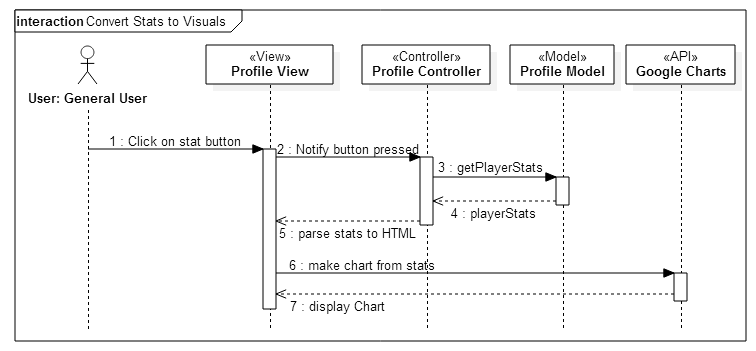
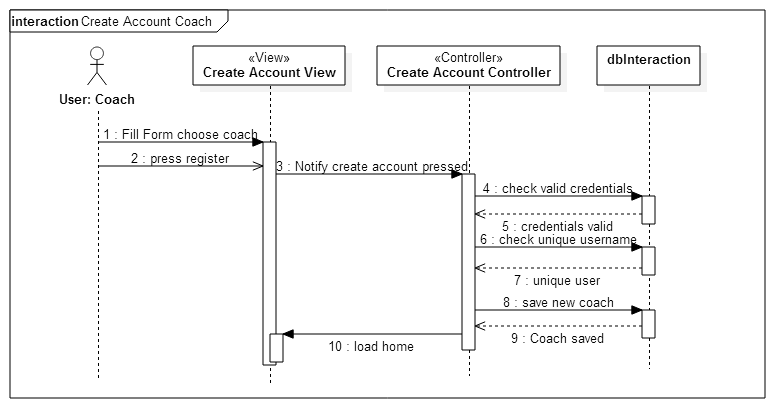
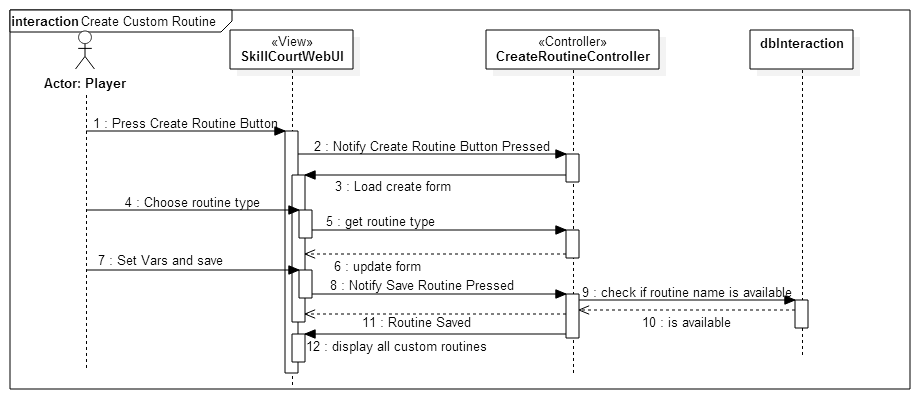
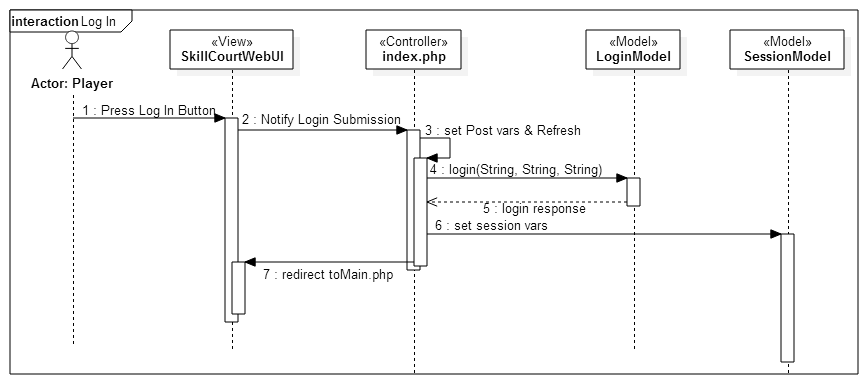
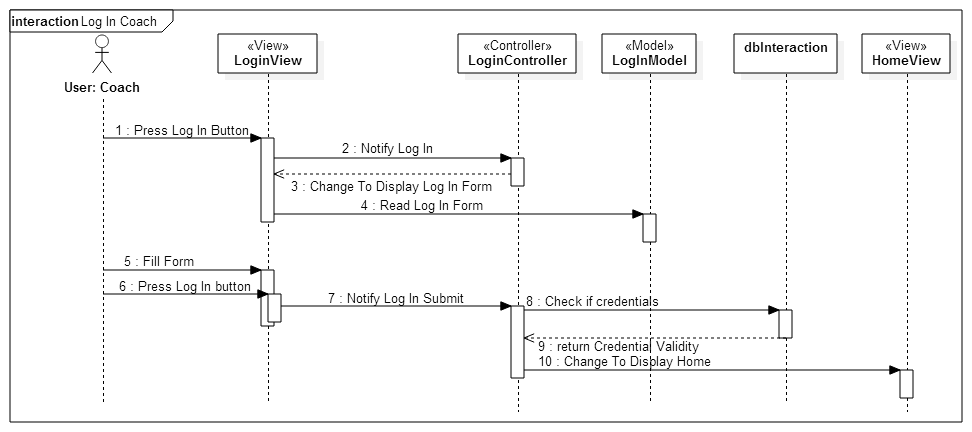
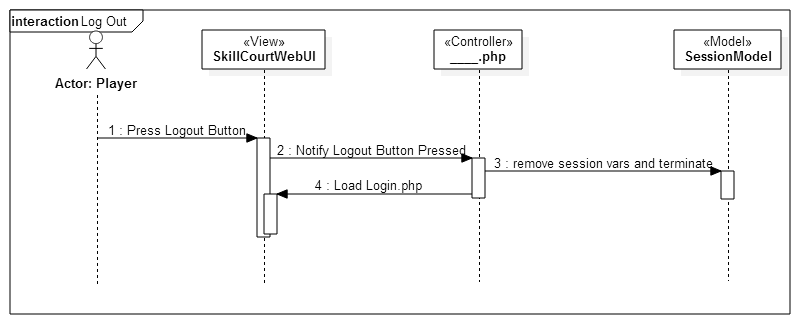
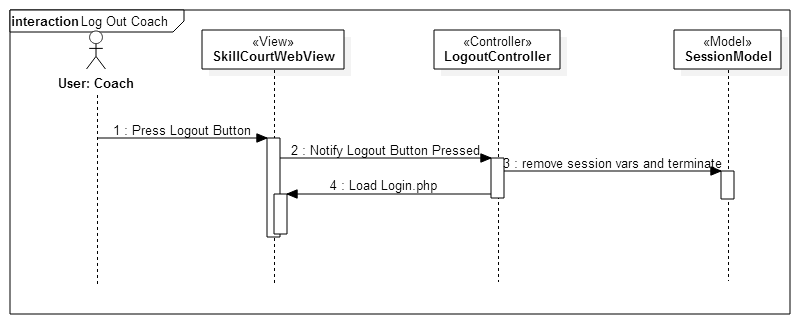
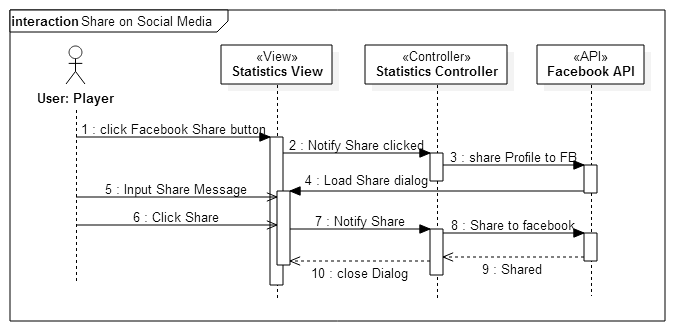
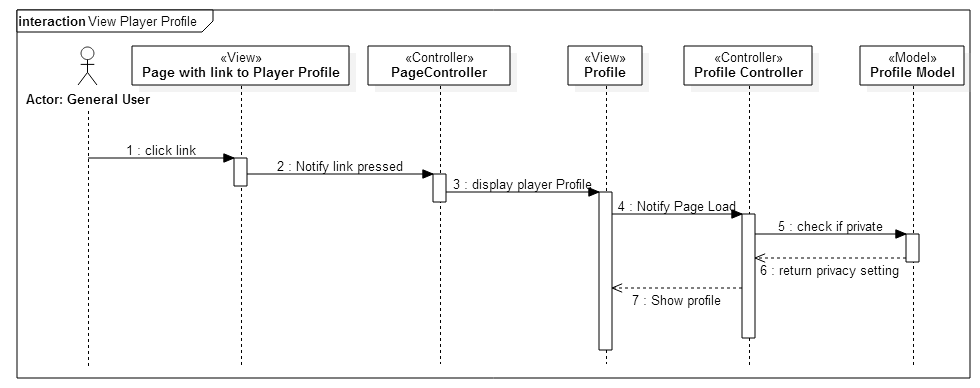
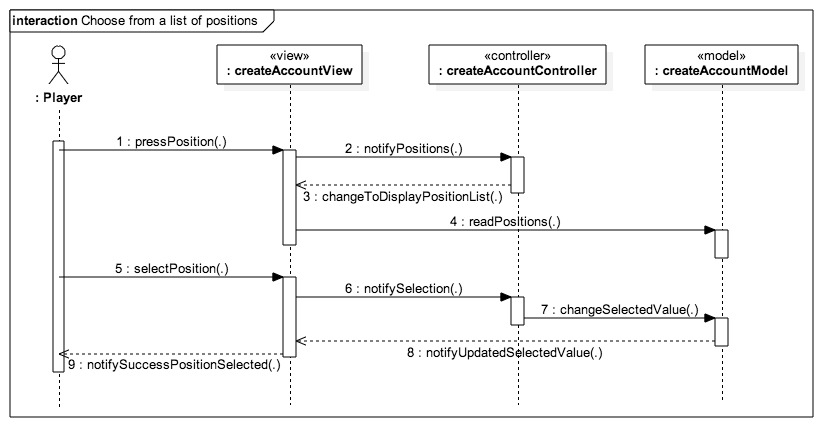
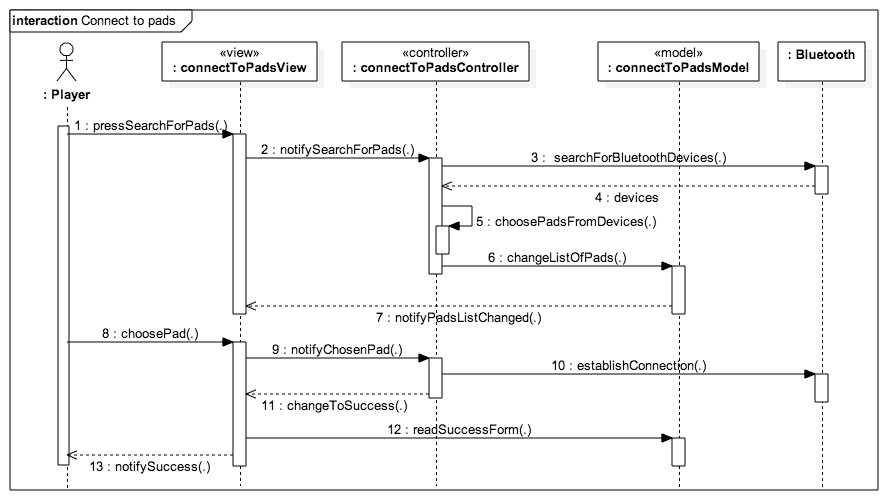


Figure 11 Convert Stats to Visuals Sequence Diagram

  
Figure 12 Create Account Coach Sequence Diagram  
  
Figure 13 Create Custom Routine Sequence Diagram

  
Figure 14 Player Login Sequence Diagram  
  
Figure 15 Coach Login Sequence Diagram  
Figure 16 Player Logout Sequence Diagram  
Figure 17 Coach Logout Sequence Diagram  
Figure 18 Share to Social Media Sequence Diagram  
Figure 19 View Player Profile Sequence Diagram

  
Figure 20 Choose from a list of positions

  
Figure 21 Connect to Pads

### Code Specification

## Glossary

|  |  |
| --- | --- |
| **Class Diagram** | A pictorial representation of all the classes in the system |
| **Functional Requirement** | A function supported by the system, where a function is a set of inputs, the behavior, and outputs. |
| **Non-Functional Requirement** | A requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. |
| **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 |

## Appendix

### Appendix A – Use case diagram

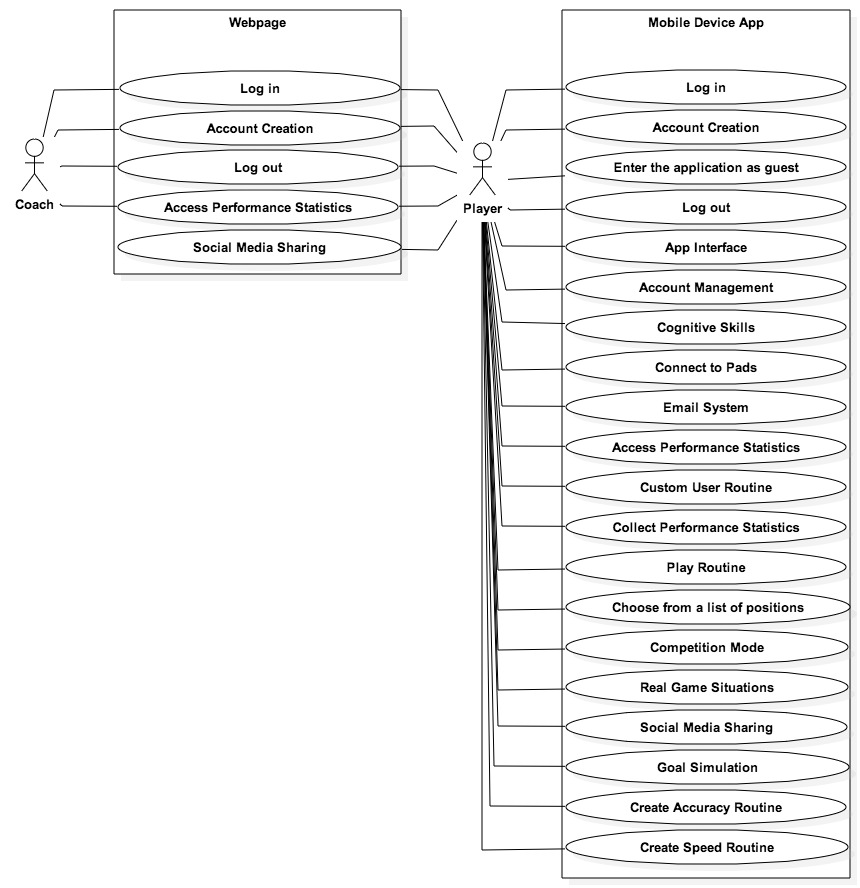


Figure 13 Use Case Diagram

### Appendix B – Implemented Use Cases

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #109 | |
| **Name:** | Access Performance Statistics | |
| **Details:** | This use case will allow the user to access his statistics | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * Player has logged in with his credentials | |
| **Description**:   * As a player I want to be able to access the statistics of my performances so that I can isolate weaknesses and see my improvements to focus in the areas of training that I need the most.   **Steps:**   * Use case begins when the user presses the “Statistics” button * The use case ends when the system accesses and displays the player’s statistics | |
| **Post-Conditions**:   * A window with the user’s statistics data is displayed | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #112 | |
| **Name:** | Account Creation | |
| **Details:** | This use case allows the players to create their SkillCourt accounts using their credentials | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * Player has entered the application in the mobile device | |
| **Description**:   * As a player, I want to be able to create an account so that I can log in and log out.   **Steps:**   * Use case begins when user presses the “Create Account” button * The system should provide the user with a form to be filled, which includes name, username, email address, password, date of birth, and coach username (optional). * The user should fill out the provided form and press “Submit” * System should record the user’s information in the database and increment the number of members by 1 * The use case ends when the system provides the user with a confirmation of the new account creation. | |
| **Post-Conditions**:   * The user information must be stored in the database | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #141 | |
| **Name:** | Enter Application as Guest | |
| **Details:** | This use case will allow the user to enter the application as a guest without logging in to an account | |
| **Actors:** | Player | |
| **Pre-Conditions:**   * Player has entered the application in the mobile device | |
| |  | | --- | | **Description**:   * As a player, I want to be able to enter the application as a guest so that I can play without having to enter my credentials.   **Steps:**   * Use case begins when user presses the “Log in as Guest” button * The system should show a warning saying that if the user enters as a guest, his performance will not be recorded, and present two options: “Accept”, “Go back” * The user presses “Accept” * The use case ends when the system displays the Guest Home View. | | |
| **Post-Conditions**:   * The system is aware that the user is a guest and is ready to handle it as such | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #142 | |
| **Name:** | Connect to Pads | |
| **Details:** | This use case allows the player to find pads that are closed to him and connect to them via Bluetooth. | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * The player has already logged into the account or entered as a guest | |
| **Description**:   * As a player I want my application to connect to the pads so that I can send information to the pads regarding what routine to play.   **Steps:**   * Use case begins when user presses the search for pads button. * The system will display a list of possible pads. * The user will select the pad he wants to connect to. * The use case ends when the system uses the Bluetooth address to establish a connection. | |
| **Post-Conditions**:   * The system has all the information needed from the pad in order to start to transfer data. | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #143 | |
| **Name:** | Log in | |
| **Details:** | This use case allows a player to log into their SkillCourt account. | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * The player has created an account already | |
| **Description**:   * As a player, I want to be able to log into my SkillCourt account so that I can have data saved to my account to track my progress.   **Steps:**   * Use case begins when the user hits the log in button on the home page * The system asks the user to input a correct username/password combination * The user enters the username/password combination * The use case ends when the system has accepted a correct username/password combination and has redirected the user to the welcome activity | |
| **Post-Conditions**:   * The player has access to his account. | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #144 | |
| **Name:** | Log out | |
| **Details:** | This use case allows a user to log out of the skillcourt app | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * The player has logged in | |
| **Description**:   * As a player, I want to be able to log out of the app so that others may log in.   **Steps:**   * Use case begins when the player hits the log out button on the upper bar. * The system deletes the local user-associated data * The use case ends when the system has redirected the user to the home page. | |
| **Post-Conditions**:   * Another player may log in. | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #145 | |
| **Name:** | Collect Performance Statistics | |
| **Details:** | This use case will allow the user to have his performance statistics stored. | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * The user is already connected to a set of pads | |
| **Description**:   * As a player I want to be able to record my performance statistics so that they are stored.   **Steps:**   * Use case begins when the user chooses the routine he wants to play * The Android application will use the name of the routine to get the routine’s specifications from the database and will transmit every step to the pads * The pads will receive the routine and will start illuminating the corresponding lights * The user will hit the illuminated pads * The pads will send information about the user’s hits to the Android application * The Android application will store the information provided by the pads in the database * The use case ends when the pads inform the Android application that the time for the entire routine has expired | |
| **Post-Conditions**:   * All performances of the player are stored in the database | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #192 | |
| **Name:** | Account Management | |
| **Details:** | This use case allows the player to change his account information after his account has been created | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * The player has already created an account and is logged into said account | |
| **Description**:   * As a user I want to be able to modify my account information so that I can make changes and updates as needed   **Steps:**   * Use case begins when user presses the edit account button. * The system will display the user’s account information * The user will press edit Information * The system will allow the user to modify his information * The user will change the information he wants to change and press done * The use case ends when the system moves to User’s profile view. | |
| **Post-Conditions**:   * The user’s new account settings will be properly changed in the database | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #197 | |
| **Name:** | Recover Username/Password | |
| **Details:** | This use case allows a player to get the credentials to the account if it was forgotten. | |
| **Actors:** | Player | |
| **Pre-Conditions**   * I’ve made a SkillCourt account already | |
| **Description**:   * As a player, I want to recover my account credentials so that I can have them in case I forget them.   **Steps:**   * Use case begins when a user clicks on the “Trouble Logging In?” button on the login interface * The system redirects the user to the account recovery activity. * The player enters the email address associated with the account. * The use case ends when the system has sent an email to the associated email address containing the username and password | |
| **Post-Conditions**:   * The player can log into his account. | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #264 | |
| **Name:** | Start Routine | |
| **Details:** | This use case allows a user to select a routine to start. | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * The player has logged in and connected to pads. * The player is on the Play Activity | |
| **Description**:   * As a player, I want to be able to choose a routine to play so that I can train my skills with SkillCourt   **Steps:**   * Use case begins when the user has arrived on the play interface * The user selects the type of routine to play. * The user selects a difficulty * The user selects how long to play the routine * The user presses play. * The use case ends when the user has configured SkillCourt to the desired game configuration and the system has accepted it. | |
| **Post-Conditions**: | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #273 | |
| **Name:** | Higher Two Rows | |
| **Details:** | This use case allows the player to play the higher two rows routine in the simulator | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * Connection between mobile device and pads has been established * The user has selected to play the higher two rows game, and has selected which level and number of rounds will be played. | |
| **Description**:   * As a player, I want to practice elevating the ball and also my reaction time to the visual cues   **Steps:**   * Use case begins when the master pad receives a command from the mobile device to start this specific routine. * The simulator will verify if the setup of the room is appropriate for the specified routine. * The entire room with start a count of 10 seconds to allow player to position himself. * Pads for this routine will start illuminating with the correct colors to expect the player’s hit. * The use case ends when the simulator starts playing the routine | |
| **Post-Conditions**:   * The user is capable of playing the routine | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend # 274 | |
| **Name:** | Lowest Row Routine | |
| **Details:** | This use case allows the player to play the lowest row routine in the simulator | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * Connection between mobile device and pads has been established * The user has selected to play the lowest rows game, and has selected which level and number of rounds will be played. | |
| **Description**:   * As a player I want to be able to practice my skills with reaction time   **Steps:**   * Use case begins when the master pad receives a command from the mobile device to start this specific routine. * The simulator will verify if the setup of the room is appropriate for the specified routine. * The entire room with start a count of 10 seconds to allow player to position himself. * Pads for this routine will start illuminating with the correct colors to expect the player’s hit. * The use case ends when the simulator starts playing the routine | |
| **Post-Conditions**:   * The user is capable of playing the routine | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #340 | |
| **Name:** | Chase Routine | |
| **Details:** | This use case allows the player to play the chase routine in the simulator | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * Connection between mobile device and pads has been established * The user has selected to play the chase routine, and has selected which level and number of rounds will be played. | |
| **Description**:   * As a player, I want to play a chase routine so that I can practice my coordination with the ball   **Steps:**   * Use case begins when the master pad receives a command from the mobile device to start this specific routine. * The simulator will verify if the setup of the room is appropriate for the specified routine. * The entire room with start a count of 10 seconds to allow player to position himself. * Pads for this routine will start illuminating with the correct colors to expect the player’s hit. * The use case ends when the simulator starts playing the routine | |
| **Post-Conditions**:   * The user is capable of playing the routine | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #393 | |
| **Name:** | Create Bluetooth Data Transfer between App and Simulator | |
| **Details:** | This use case allows the player to establish a Bluetooth data transfer between the master pad and the mobile device. | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * The user should have already logged in or entered the application as a guest. | |
| **Description**:   * As a player, I want to be able to send and receive information to and from the pads via Bluetooth so that I can play and receive my performance statistics.   **Steps:**   * Use case begins when the user presses “Play.” * The mobile device sends the routine to the master pad * The use case ends when the master pad receives the routine | |
| **Post-Conditions**:   * The master pad has the necessary data to know the specifications of the routine to be played. | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #426 | |
| **Name:** | View Game Results | |
| **Details:** | This use case allows the user to view how he did in a game. | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * The player has played a game | |
| **Description**:   * As a player, I want to be able to see how I did in a game I just completed.   **Steps:**   * Use case begins when the player finishes a SkillCourt routine. * The system receives the player’s statistics from that game. * The system redirects the view to the game results screen * The system displays the player’s statistics to the screen * The use case ends when the player views the statistics on the results screen | |
| **Post-Conditions**:   * The player can view the statistics from the routine. | |

#### The webpage

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #124 | |
| **Name:** | Share on Social Media | |
| **Details:** | This use case allows a user to share his profile on Facebook | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * The player has logged into the website | |
| **Description**:   * As a player, I want to be able to share my profile on social media so that my friends can see how I am doing   **Steps:**   * Use case begins when the player hits the “share” button on the statistics page * The system opens a dialog to log into facebook * The system opens a dialog that allows the player to enter a personalized message * The use case ends when the player hits the “post” button | |
| **Post-Conditions**:   * The system posts the player’s profile link to facebook. | |
|  | |
| **Use case ID:** | SkillCourt-Backend #134 | |
| **Name:** | Login – SkillCourt Website | |
| **Details:** | This use case allows a player to log into the website | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * The player has created a skillcourt account | |
| **Description**:   * As a user I want to be able to access my SkillCourt data on the web so that I can access my SkillCourt account from the comfort of my computer   **Steps:**   * Use case begins when the player enters the SkillCourt website. * The player inputs his proper credentials and presses login * The use case ends when the system redirects the player to the Main SkillCourt page | |
| **Post-Conditions**:   * The player has logged into his account | |
|  | |
| **Use case ID:** | SkillCourt-Backend #338 | |
| **Name:** | Website Logout | |
| **Details:** | This use case allows the user to log out of his account from the website | |
| **Actors:** | Player | |
| **Pre-Conditions**:   * The player has logged into the website | |
| **Description**:   * As a player, I want to be able to log out of my account so that others can log in.   **Steps:**   * Use case begins when the player hits the log out button * The system deletes all user info associated to that session * The system redirects the user to the login page * The use case ends when the player is on the login page. | |
| **Post-Conditions**:   * Another player can login | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #337 | |
| **Name:** | Logout – Coach | |
| **Details:** | This use case allows a coach to log out of the website. | |
| **Actors:** | Coach | |
| **Pre-Conditions**:   * The coach has logged in | |
| **Description**:   * As a coach, I want to be able to log out of my account so that I can keep my account secure and allow others to log in.   **Steps:**   * Use case begins when the coach hits the log out button * The system deletes all user info associated to that session * The system redirects the user to the login page * The use case ends when the coach is on the login page. | |
| **Post-Conditions**:   * Another player or coach can login | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #338 | |
| **Name:** | Create Account – Coach | |
| **Details:** | This use case allows a coach to create an account | |
| **Actors:** | Coach | |
| **Pre-Conditions**:   * The coach is on the login page | |
| **Description**:   * As a coach, I want to be able to create an account so that I can have access to my players through skillcourt.   **Steps:**   * Use case begins when the coach hits the register button on the login page. * The coach enters proper credentials and selects to create a coach account * The use case ends when the coach hits the create button | |
| **Post-Conditions**:   * The system stored the coach’s information into the database | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #339 | |
| **Name:** | Login – Coach | |
| **Details:** | This use case allows a coach to log into his account | |
| **Actors:** | Coach | |
| **Pre-Conditions**:   * The coach has created an account | |
| **Description**:   * As a coach, I want to be able to log in to out of the SkillCourt website so that I can get access to my account.   **Steps:**   * Use case begins when the coach is on the log in page * The coach enters his proper credentials * The use case ends when the system redirects the user to the main page | |
| **Post-Conditions**:   * The coach has access to his account | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #392 | |
| **Name:** | Create Player Profile | |
| **Details:** | This use case allows users to view a player’s profile | |
| **Actors:** | General | |
| **Pre-Conditions**:   * The user has a link to a player’s profile | |
| **Description**:   * As a user, I want to be able to view player’s account information   **Steps:**   * Use case begins when the user follows a link to a player’s profile * The system loads the players profile and uses GET to obtain the player’s information to display * The system loads the player’s information and parses it to HTML * The use case ends when the profile page has loaded | |
| **Post-Conditions**:   * The user can see the player’s profile. | |

|  |  |  |
| --- | --- | --- |
| **Use case ID:** | SkillCourt-Backend #396 | |
| **Name:** | Convert Statistics to Visuals | |
| **Details:** | This use case allows the system to create visualized graphs for users to see. | |
| **Actors:** | General | |
| **Pre-Conditions**:   * The user is on a player’s profile * The player is on his own stats page | |
| **Description**:   * As a player, I want to be able to be able to see my statistics so that I can track my progress. * As a user, I want to be able to see a player’s statistics so that their progress can be visualized.   **Steps:**   * Use case begins when a player or user clicks on the “Force” or “Accuracy” button on the statistics or profile page, respectively * The system retrieves the player’s statistics data from the database and parses it to the Google Charts API * The use case ends when the system displays the graph. | |
| **Post-Conditions**:   * A user can see the selected player’s statistic chart. | |

### Appendix C – Detailed Class interfaces

### Appendix D - Diary of Meetings and tasks

1. **Meeting 1:**

Monday 1/19/15

@8:57pm

Meeting on Mingle starts

In attendance: Andy Martinez, Matthew Santiago

@9:08pm

Arranging possible meeting times for Requirements Elicitation:

* Tuesday 1/20 9:00 pm
* Wednesday 1/21 before 2:00 pm
* Wednesday 1/21 after 7:30 pm

@9:23pm

Chose for Andy Martinez to be Scrum Master for the first sprint

@9:28pm

Sent message to @Product Owner for their preferred meeting time:

* Awaiting reply…
* **1/20 @10:43am** Reply Received, Meeting Tuesday 1/20 at 9:00 pm confirmed

@9:33pm

Meeting Dismissed

1. **Meeting 2:**

Tuesday 1/20/15

@9:00 pm

Conference Call Started

In attendance: Matthew Santiago, Andy Martinez, Jaime Borras, Gummi

@9:02 pm

Introductions

@9:12 pm

Stories:

1. Skill development (accuracy, speed, intensity)
   1. User chooses skill to train from Android App
   2. Measure pressure, time
   3. 3 colors red blue green
   4. Change to red when underperforming
   5. Ball should take 5 seconds to hit the pad, turn red if user takes too long
2. Tracking and analysis
   1. accuracy, speed, intensity, reaction time
3. Single Player/2 Player
   1. Social Media Connection
4. Mapping a game
   1. Pad light up, hit pad with ball in proper time, another pad lights up, hit with ball in proper time, continue until you make goal (certain number of successful iterations)
   2. Beginner, Intermediate, and advanced levels
5. Simulator (simulate a pad)
   1. Android application?
6. (Backend look at particular game (real game) and simulate a player or play within that game)

@9:32 pm

Set meeting with Gummi on 1/21 at 11:00 am in GL 693

@9:38 pm

Meeting Dismissed

1. **Meeting 3:**

Thursday 1/22/15

@9:00

Conference Call start

In attendance: Andy Martinez, Matthew Santiago, Jaime Borras

@9:07

Review stories

Predetermined Routines:

* Separate Goal Simulation into its own story

Performance Statistics:

* Add statistics about specific game

Competition Mode:

* Separation of social media into a new story is good
* Have different routines for single and multiplayer

Custom User Routine:

Pads Simulator:

Real Game Simulation:

Social Media Sharing:

Website:

* (Differentiate between coach and player?)
* Webpage access

@9:46

Meeting Dismissed

1. **Meeting 4:**

Tuesday 1/27/15

@ 6:30 pm

Meeting starts

In attendance: Jaime Borras, Andy Martinez, Matthew Santiago

**Github upload schedule:**

Matthew upload from 10:00pm - 10:59pm

Andy upload from 11:00pm - 11:59pm

**Weekly in-person meeting:**

Thursday between 2:00pm - 6:00pm to prepare for weekly meetings with project owners

**Coding standards**

Comment Convention:

/\*\*

\* Comment goes here

\* and here

\*/

fun()

{

random code

}

**Indentation:**

1 tab per pair of curly braces

**Variable names:**

private \_variableName

public variableName

**Reports:**

**Matthew:** Feasibility report & Initial System Design

**Andy:**  Project Plan & Initial Object Design

**Read up on:**

Android BlueTooth library

Java Databases

1. **Meeting 5:**

Tuesday 1/30/15

@ 4:00 pm

Conference Call starts

In attendance: Jaime Borras, Andy Martinez, Matthew Santiago, Gummy

Complete?

## References