**Software Requirements Specification**

**for**

**Leveling The Playing Field**

**Version 2.0 approved**

**Prepared by Kristina Krokosz, Areli Muñoz,**

**Matt Cowans, Mark McDonough**

May 5, 2019

**Github Project Link:** https://github.com/arimunoz/SportsProj.git

**Table of Contents**

**Table of Contents........................................................................................................ ii**

**1. Introduction............................................................................................................ 1**

1.1 Purpose...................................................................................................................... 1

1.2 Intended Audience and Reading Suggestions............................................................ 1

1.3 Project Scope............................................................................................................. 1

1.4 References................................................................................................................. 1

**2. Overall Description................................................................................................. 3**

2.1 Product Perspective................................................................................................... 3

2.2 Product Features........................................................................................................ 3

2.3 User Classes and Characteristics.............................................................................. 3

2.4 Operating Environment.............................................................................................. 4

2.5 Design and Implementation Constraints.................................................................... 4

2.6 User Documentation.................................................................................................. 4

2.7 Assumptions and Dependencies............................................................................... 4

**3. External Interface Requirements......................................................................... 5**

3.1 User Interfaces........................................................................................................... 5

3.2 Software Interfaces.................................................................................................... 6

3.3 Communications Interfaces....................................................................................... 6

**4. Other Nonfunctional Requirements.........................................................................**

4.1 Security Requirements............................................................................................... 7

# **1.** **Introduction**

## **1.1** **Purpose**

Leveling the Playing Field is a website that assess the statistics of players and calculates into their base salary. We created an asp.NET Core web application implementing razor pages to calculate player’s salaries using a universal calculation. Basing our salary calculation from the players individual stats we can showcase their ideal salary vs their listed contract salary. Leveling the Playing Fields’ focus currently calculates salaries for MLB athletes, eventually our application can be used to calculate salaries for players across all leagues.

There is a homepage where you can login to have access to certain featured pages. Once you have logged in, you can manually type in the statistics for hitters or pitchers.

## **1.2** **Intended Audience**

We wanted Leveling the Playing Field to be used by anyone, however if professionals wanted to use this program for their own team; such as team manages, Third party agencies, and individual agents, they are to do so. They will be able to use Leveling the Playing Field in a way to promote a players growth over their seasons to make themselves a more marketable player during free agency.

## **1.3** **Project Scope**

The purpose of this project is to eliminate contract negotiations between third party companies, team league managers, and players individual managers. The goal is to calculate players salaries based on their statistics, so they are in theory earning their salary. This product can help teams better manage all players salaries and the teams’ overall salary expenses.

## **1.4** **References**

Ardalis. “Ardalis/OrganizingAspNetCore.” GitHub, 2 Nov. 2017, github.com/ardalis/OrganizingAspNetCore/tree/master/WithRazorPages., <https://github.com/ardalis/OrganizingAspNetCore/tree/master/WithRazorPages>

“ASP and ASP.NET Tutorials.” *ASP Tutorial*, [www.w3schools.com/asp/](http://www.w3schools.com/asp/).

Dotnet-Bot. “FormExtensions.BeginForm Method (System.Web.Mvc.Html).” FormExtensions.BeginForm Method (System.Web.Mvc.Html) | Microsoft Docs, docs.microsoft.com/en-us/dotnet/api/system.web.mvc.html.formextensions.beginform.,

<https://www.completecsharptutorial.com/asp-net-mvc5/html-beginform-example-tutorial-aspnet-mvc5.php>

“Html Agility Pack (HAP) Html Agility Pack.” *Zzz Projects*, html-agility-pack.net/.

“How to Install SQL Server on a Mac.” *Database.Guide -*, database.guide/how-to-install-sql-server-on-a-mac/.

<https://database.guide/how-to-install-sql-server-on-a-mac/>

Kumar, Mukesh. “CRUD Operations in Asp.Net Core 2 Razor Page with Dapper and Repository Pattern.” *Mukesh Kumar*, 1 Jan. 1969, www.mukeshkumar.net/articles/dotnetcore/crud-operations-in-asp-net-core-2-razor-page-with-dapper-and-repository-pattern.

“MLB Stats, Scores, History, & Records.” *Baseball,* [www.baseball-reference.com/](http://www.baseball-reference.com/)

“Online Courses for Creative, Technology, Business Skills.” *LinkedIn Learning*, www.linkedinlearning.com/.

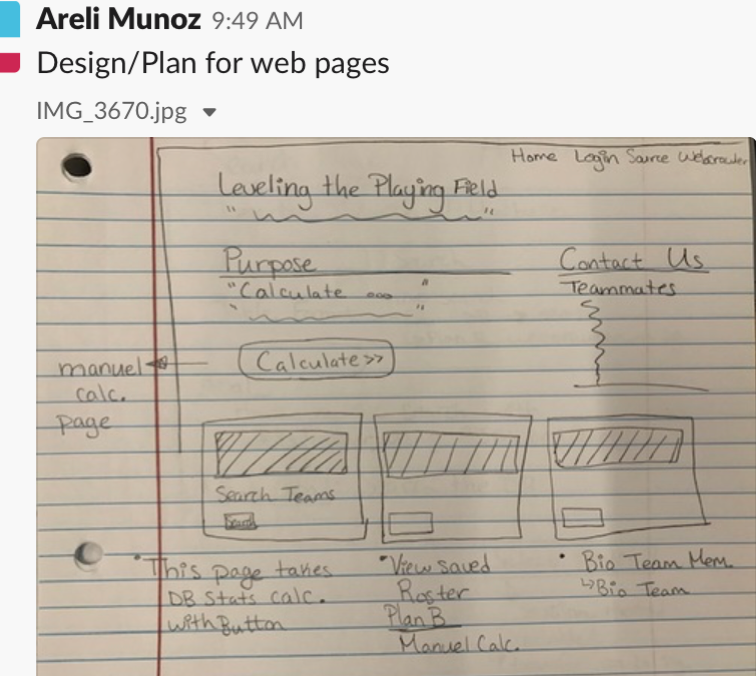
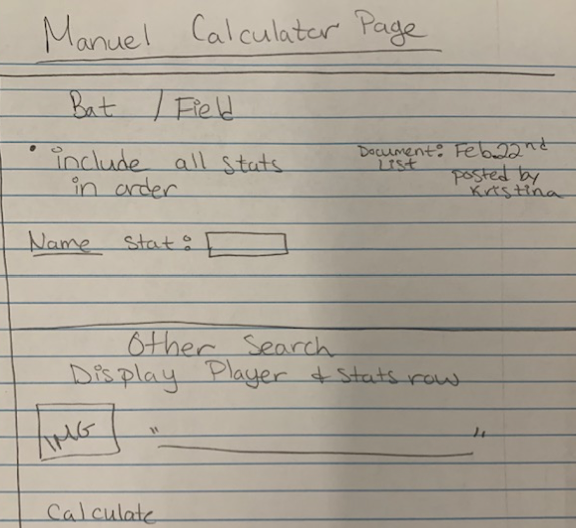
Your guide to using ASP.NET Core Razor Pages. “Welcome To Learn Razor Pages.” *An Introduction To ASP.NET Core Razor Pages | Learn Razor Pages,* 11 Jan. 2018, [www.learnrazorpages.com](http://www.learnrazorpages.com/), [www.learnrazorpages.com/razor-pages](http://www.learnrazorpages.com/razor-pages), <https://www.learnrazorpages.com/razor-pages/tag-helpers/input-tag-helper>

Jchadwick. “Welcome.” *Lynda.com - from LinkedIn*, 17 Nov. 2017, www.lynda.com/ASP-NET-tutorials/Welcome/630622/679579-4.html?autoplay=true.

Rick-Anderson. “Add Search to ASP.NET Core Razor Pages.” *Microsoft Docs*, docs.microsoft.com/en-us/aspnet/core/tutorials/razor-pages/search?view=aspnetcore-2.2.

# **2.** **Overall Description**

## **2.1** **Product Perspective**

This products is a self contained product that originally had the idea of being a generalized athlete calculator for all sports. We branched the idea to focus on just the MLB. 

## **2.2** **Product Features**

Product includes features such as search for players, calculate players salaries, and a login with validation. Each page is inaccessible unless logged into an account.

## **2.3** **User Classes and Characteristics**

The top classes used are the hitters and pitchers calculator page, home main calculation page, and the home index page. It is mandatory to login to access the hitters and pitchers calculation page.

* BatCalc.cshtml.cs
* PitcherCalc.cshtml.cs
* MainCalcPage.cshtml
* Index.cshtml

## **2.4** **Operating Environment**

The project is hosted on a linux server and will run on any web browser.

## **2.5** **Design and Implementation Constraints**

Leveling the Playing Field is developed in C# using the .Net Core 2.0. With the use of this framework, We are able to take advantage of the functionality of HTML, JavaScript, CSS, and Bootstrap with .NET Core.

## **2.6** **User Documentation**

Your guide to using ASP.NET Core Razor Pages. “Welcome To Learn Razor Pages.” *An Introduction To ASP.NET Core Razor Pages | Learn Razor Pages,* 11 Jan. 2018, [www.learnrazorpages.com](http://www.learnrazorpages.com/)

Take a look at courses in linkedinlearning and lynda.com.

User Manual: Look into Areli’s Documentation on Github; <https://github.com/arimunoz/SportsProj.git> Instructions.txt on the search feature within Kristina’s Documentation. Also, we have Matt’s Documentation on uploading to the server, database connection, and webcrawler.

## **2.7** **Assumptions and Dependencies**

*“MLB - Major League Baseball Teams, Scores, Stats, News, Standings, Rumors.” ESPN, ESPN Internet Ventures, www.espn.com/mlb/.*

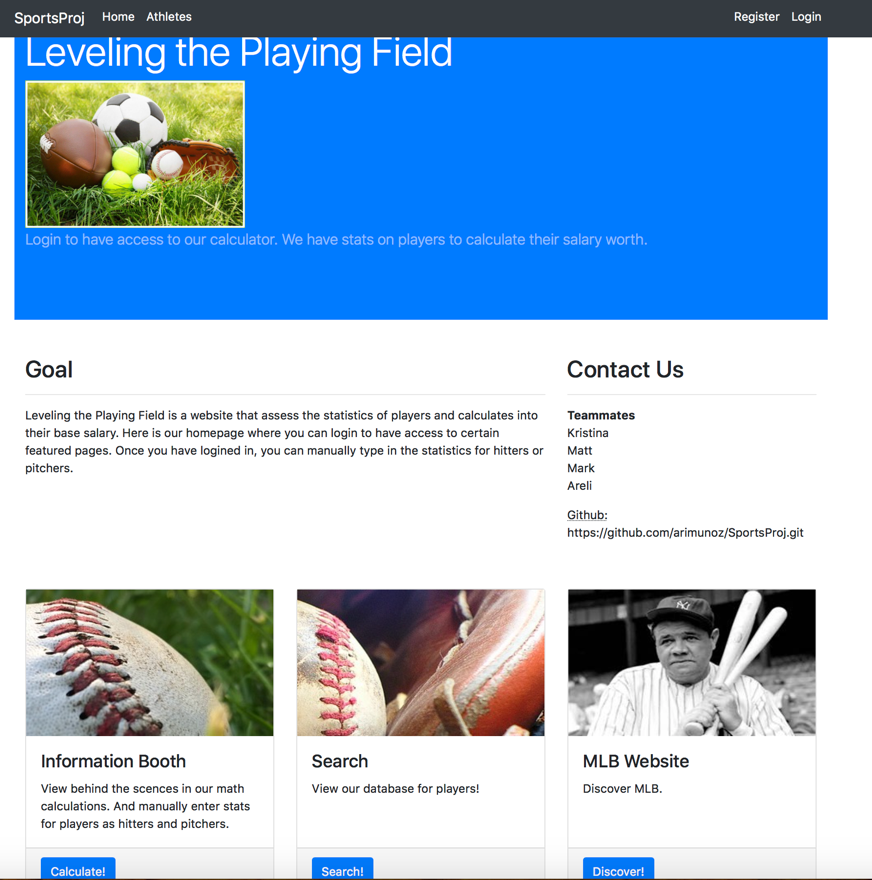
# 

# 

# 

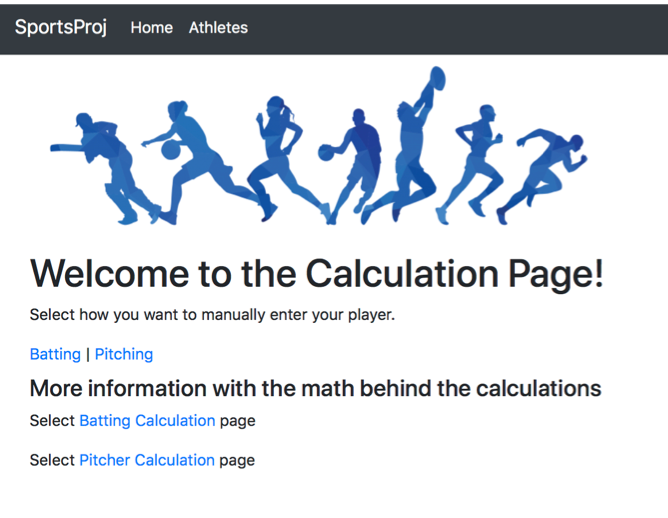
# **3.** **External Interface Requirements**

## **3.1** **User Interfaces**



The layout is displayed using bootstrap. Here the user can select whether they want to login, register, search, enter into another website, and enter into our main calculation page to have access to Hitters/Pitchers Page.

To click over to another page you need the asp-page tag along with the path of the files.



To limit featured pages all you need is [Authorize] in the .cshtml.cs PageModel.

## 

## **3.2** **Software Interfaces**

Leveling the Playing Field uses two different databases: MYSQL as well SQL Server Object Explorer, SQL Object Explorer is Microsoft's integrated database within visual studio and MYSQL is a database that can be hosted the the local machine. Inorder to connect the database with Razor pages a CRUD operation is required. By creating a model directory to hold the tag helper in order to seed the database. From there another directory is made for the Repository this is where the MYSQL statements are help within classes that are later called when its appropriate.

## **3.3** **Communications Interfaces**

The project parse stats from the ESPN MLB page, using the Nuget package HTML Agility Pack. With this package classes like HtmlDocument and HtmlNode are what is used most. Similar to an XML parser, it provides the selectSingleNode and selectNodes methods that accept XPath expressions. The HtmlDocument.Load() method also accepts a stream which is very useful in integrating with other stream oriented classes in the .NET framework. While HtmlEntity.DeEntitize() is another useful method for processing html entities correctly. From here we are able to parse any information from an HTML document and store it in a database.

# 

# 

# **4.** **Other Nonfunctional Requirements**

## **4.1** **Security Requirements**

Security issues could pertain to how asp.NET Core stores user passwords and login information. Upon creating an account the requirements consists of using one non alphanumeric, one digit, and one uppercase letter. The password must be between 6 and 100 characters long and the email validation checks for the @ symbol. Since the login/register were auto generated with the project, they are stored internally within asp.NET Core. Due to this we do not know if the passwords are encrypted and if they are to what extent.