PocketScores Desktop Application

Software Requirements Documentation

PardonMyCode

12/01/2020

Clayton Winters, Brandon Banner, George Bell

UNCG Honor Code

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

1. Introduction to PocketScores

1.1. Title

1.1.1. Project Name:

PocketScores

1.1.2. Team Name:

PardonMyCode

1.1.3. Date:

12/01/2020

1.1.4. Team Members:

Clayton Winters, Brandon Banner, and George Bell

1.1.5. Stakeholders/Company:

University of North Carolina Greensboro Honor Code

1.2. Purpose:

The purpose of the PocketScores application is to simply inform anyone, anywhere, with live scores. While not wanting to deal with all the huge scores applications on the internet. Most of the popular sports scores applications (i.e.. ESPN, CBS Sports, and Yahoo Sports) take minutes to load all their different options (I.e., fantasy sports, live TV, and personal preferences) but the PocketScores application is an easy quick option to check your favorite teams score, and every NFL score at a glance, updated in real-time just click on the matchup for detailed live scoring. If someone were busy during the game, they can also open this application the next morning and look at all the scores from the night before. This application allows anyone to check past scores from the entire season, and future match ups to come.

Introduction (Continued)

1.3.Intended Audience:

Professor Ike Quigley with the class of CSC 340 sections 01 and 02.

1.4.Definitions/Jargon:

Some words and abbreviations that are often used that some people may not know.

T.O.	Time Outs
Quarter	The game is split into four quarters
Event	Game
Home team	Team playing at their field
Away team	Team traveling to opponents field
NFL	National Football League
NCAAF	National Collegiate Athletic Association Football (College Football)
NCAAB	National Collegiate Athletic Association Basketball (College Basketball)
NBA	National Basketball Association
Line to Gain	Yards to first down line
Game Time	Start time of game in Eastern Standard Military Time

1.5.Project Scope:

To construct an alpha version of an application that will display live scoring from any NFL game in the current season. This application will also show matchups and scores from past and future games on the NFL sport list view. The user will be able to mouse click on the game to view detailed scoring of that particular game.

Introduction (Continued)

1.6. Technical Challenges:

Connecting PocketScores to an online API to gather our up-to-date application while we have never worked with an API previously it has been a learning experience. Our API needs to return up to date live scoring from the NFL. Another challenge we ran into was creating a different number of scoreboards for our list of games that are scheduled. As a team we have never worked with JavaFXML before.

1.7. References:

- RapidAPI.com
- StackOverFlow.com
- Professor Ike Quigley

2. Overall Description

2.1. Product Features:

The user can quickly open this application, click their sport, then the application will display a list of games that are scheduled for that certain sport. The most important feature is our search for date design. If the user is curious about when or where their favorite team is playing their next game, the user can scroll through each day or the user can search for the date in our calendar drop down in the sport list view. While a game is being played this app allows the user to click on any active game for more live information in our live scoreboard view which displays the quarter and how much time is remaining in the game. The scoreboard view also displays how many points each team scored per quarter and total points by each team.

Overall Description (Continued)

2.2.User Characteristics:

Anyone that is interested in sports could use this application. PocketScores is an application to inform the everyday sports fan with up to the minute scores of their favorite team or check in on their rival's game. Someone who is looking to find NLF scores quick and efficiently, to be more involved in NLF conversations with family and friends. Anyone that is interested in sports would love this application.

2.3.Operating Environment:

This application is designed to be used anywhere with a computer and internet connection.

2.4.Design & Implementation Constraints:

- The Rundown API dropped support for the free version of the API.
- The Sportspage Feed (SportsFeed) APIs will continue to work correctly.
- Implement number of HBox needed per number of games.
- Erase prior HBox information to replace.

2.5.Assumptions & Dependencies:

NFL season will continue through the COVID19 global pandemic. Our RESTful API will continue to function correctly and not charge us a thousand dollars in the process.

3. Functional Requirements

3.1. Primary:

The main purpose of this application is to get live scoring from an online API for a live view of every game. It's designed to display live NFL scores. It will also display past scores from the current season as well as future scheduled games from the current season. Our newest feature is the drop-down calendar available on our NFL list view. The drop down calendar will replace looking scrolling through every available date.

3.2. Secondary:

Displays scores from every Sport (NBA, NCAAF, NCAAB, MLS, NHL) and be able to save users personalized feed with all the content you care about from your favorite teams and leagues.

4. Technical Requirements

4.1. Operating Systems/Compatibility:

- NetBeans12.1 IDE
    - Gluon plugin on NetBeans
- Java compatible device.
- Gluon SceneBuilder.
- Java FX 15 (or later) Library
- JFoenix library

Technical Requirements (Continued)

4.2.Interface Requirements

4.2.1.User Interface:

User will need a mouse and/or touch screen to use application.

4.2.2.Hardware Interface:

User will need a display to view the application and a computer to run the application.

4.2.3.Software Interface:

Our interface is connect to multiple RESTful API's.

5. Nonfunctional Requirements

5.1. Performance Requirements

Safety/Recovery Requirements

5.2.Security Requirements

N/A

5.3.Policy Requirements:

CSC 340: Software Engineering Style Guide

5.4.Software Quality Attributes

5.4.1. Availability:

Application is available at anytime.

5.4.2. Correctness:

While this application is not life saving, it is essential to our team that the application works accurately.

Nonfunctional Requirements (Continued)

5.4.3. Maintainability:

This application has great potential for expansion, however there may be difficulty with implementing sorting methods.

5.4.4.Reusability:

This application has good reusability, the models are flexible enough so that an API change would not affect the program.

5.4.5. Portability:

Due to the application not being an executable application, the technical requirements hinder the portability of the application.

5.5.Process Requirements

5.5.1.Development Process Used:

Spiral

5.5.2.Time Constraints:

Due to COVID-19, team members are in different remote locations, creating scheduling difficulty.

5.5.3.**Cost and Delivery Date**:

Cost of total application: $100.00, delivery date is 12/01/2020 at 7:00pm est.

All SRS/SRD should be:

- Correct: A method of analysis that ensures that the software meets the requirements identified.

- Unambiguous: There is only one interpretation of what the software will be used for and it is communicated in a common language.
- Complete: There is a representation for all requirements for functionality, performance, design constraints, attributes, or external interfaces.
- Consistent: Must be in agreement with other documentation, including a systems requirements specification and other documents.
- Ranked for Importance and/or Stability: Since all requirements are not of equal weight, you should employ a method to appropriately rank requirements.
- Verifiable: Use measurable elements and defined terminology to avoid ambiguity.
- Modifiable: A well-defined organizational structure of the SRS document that avoids redundancies can allow easy adaptation.
- Traceable: Ability to trace back to the origin of development and move forward to the documents produced from the SRS.
- Legible and Professionally Presented: Must use a consistent font and style. Must have proper formatting of tables and charts. Must be grammatically correct. Use active tense and concise sentences.