**Software Requirements Specifications for College Football Coach**

**December 8th, 2018**

**Team Members**

Matthew Klopfenstein

Keith Radulski

Gabriel Siewert

Table of Contents

[1. Introduction 3](#_Toc532559150)

[1.1 Purpose 3](#_Toc532559151)

[1.2 Scope 3](#_Toc532559152)

[1.3 References 3](#_Toc532559153)

[2. General Description 3](#_Toc532559154)

[2.1 Product Perspective 3](#_Toc532559155)

[2.2 Product Functions 4](#_Toc532559156)

[2.3 User Characteristics 4](#_Toc532559157)

[2.4 General Constraints 4](#_Toc532559158)

[3. Functional Requirements 4](#_Toc532559159)

[4. Non-functional Requirements 5](#_Toc532559160)

[5. Database Diagrams 6](#_Toc532559161)

[5.1 Database Schema 6](#_Toc532559162)

[5.2 Database ER Diagram 8](#_Toc532559163)

# **Introduction**

## **Purpose**

The purpose of this document is to describe the various characteristics and requirements of the College Football Coach Android mobile app. This document is intended for a user of the app who has at least a surface level familiarity with the app.

## **Scope**

This product is intended to be a relatively simple game, displaying the results of each game throughout the season, and updating team rankings and player ratings. It does not display more in-depth statistics such as the player roster for every team, the MVP for a season, or player depth charts.

## **References**

This Android mobile app was modeled after two existing apps, the citations for which are listed below.

Easwaran, A. (2018). *College Football Coach*. Apple.

Jones, A. (2016). *Football Coach*. Palm Bay, FL: Google.

# **General Description**

## **Product Perspective**

This product was developed with the intent of designing an entertaining, user-friendly game. The product is meant for fans of college football and/or fans of fantasy football games. The product was implemented using Android Studio, and the source code was written in Java and XML.

## **Product Functions**

This product will allow a user to do all of the following.

1. Select a college football team to be the “coach” of
2. Play through each week of a football season for multiple seasons
3. Observe game results, changing team rankings and player ratings, season results, and the updating of their player roster

## **User Characteristics**

This product is intended for fans of college football, who have a moderate amount of knowledge regarding the sport. Users of this product should be familiar with the general structure of a football season, how Win-Loss statistics work, and how football games are scored.

## **General Constraints**

This product may not support multiple simultaneous instances of the app being run. This product’s target SDK is Android 27. The minimum SDK supported is 4, phones with versions older than this may not be able to run the application. Older phones may experience delay when the play game button is running the “play all games” logic. No login information or password is necessary.

# **Functional Requirements**

1. The product will display a main screen with options for the user to select.
2. The product will allow the user to reset all data (i.e. reset entire database).
3. The product will allow the user to view a “Game Info” page.
4. The product will allow the user to select a team when starting a new game.
5. The product will allow the user to play through multiple seasons of college football.
6. The product will allow the user to view the top 25 ranked teams at any time during a season.
7. The product will allow the user to view information about their team, including the Win-Loss statistics, Rank, and Player Roster.
8. The product will display the schedule for the user’s team during a season.
9. The product will allow the user to play through each week during a season.
10. The product will display the result of the game that the user’s team played in upon selecting “Play Week,” including the score, and whether the user’s team won or lost.
11. The product will display and update the current week number and current year.
12. The product will dynamically add games to the schedule depending on the performance of the user’s team throughout the season. This could be either or both a conference championship game or participation in the college football playoffs.
13. The product will also display your overall head coaching record across all seasons during your “tenure” as head coach of a particular team.
14. The product will display the players on the user’s team who are leaving at the end of a season, only seniors will leave.
15. The product will display the freshman players incoming to the user’s team before a new season begins.
16. The product will display the players on the user’s team that did not leave, along with the amount that their player rating changed over the course of the previous season.
17. The product will display the results of a season, including the National Champion, the results of the National Championship game, and the user’s team’s Rank and Win-Loss statistics.

# **Non-functional Requirements**

1. The product shall have a “Game Info” page wherein users can learn about how the product can be used.
2. The product shall use the Room Persistence Library (a wrapper class around SQLite to interact with Java objects) to keep data persistent across app runs.
3. The product will support only one team to be “coached” by the user.
4. The product will use the Varsity font for the title page.
5. The product logo will appear on the main screen.
6. The product color scheme will consist of dark blue and light blue.

# **Database Diagrams**

## **5.1 Database Schema**

CREATE TABLE Teams (

name VARCHAR(64) PRIMARY KEY,

abbreviation VARCHAR(5),

conference VARCHAR(32),

division VARCHAR(32),

wins INT,

losses INT,

conWins INT,

conLosses INT,

offRating REAL,

defRating REAL,

rankingVotes INT

);

CREATE TABLE Game (

home VARCHAR(64),

away VARCHAR(64),

homeScore INT,

awayScore INT,

week INT,

PRIMARY KEY(home, away, week),

FOREIGN KEY(home) REFERENCES Teams,

FOREIGN KEY(away) REFERENCES Teams

);

CREATE TABLE Players (

id INT PRIMARY KEY,

rating REAL,

firstName VARCHAR(32),

lastName VARCHAR(32),

position CHAR(2),

year CHAR(2)

);

CREATE TABLE State (

playerTeam VARCHAR(16) PRIMARY KEY,

newgame INT,

careerWins INT,

careerLosses INT,

year INT,

week INT.

difficulty VARCHAR(4)

);

CREATE TABLE oldTeams (

name VARCHAR(64) PRIMARY KEY,

abbreviation VARCHAR(5),

conference VARCHAR(32),

division VARCHAR(32),

wins INT,

losses INT,

conWins INT,

conLosses INT,

offRating REAL,

defRating REAL,

rankingVotes INT

);

## **5.2 Database ER Diagram**

