# PocketSports
## A Digital Coaching App

## Software Requirements Document
## Version 1.0

**Team Members:**

Garrett Gmeiner - ggmeiner2021@my.fit.edu

Tyler Ton - tton2021@my.fit.edu

Parker Cummings - pcummings2021@my.fit.edu

Taylor Carlson - tcarlson2021@my.fit.edu

**Faculty Advisor:**

Fitzroy Nembhard - fnembhard@fit.edu

**Client:**

Brad MacArthur - bmacarthur@fit.edu

Florida Institute of Technology

9/30/2024

# Table of Contents

# 1. Introduction

## 1.1 Purpose

The purpose of PocketSports is to provide a comprehensive digital solution for sports teams, specifically designed to streamline the coaching process. The application will allow coaches to manage teams, create and execute practice plans, track performance metrics, and enhance player development through drills and performance analysis. The platform is aimed at simplifying team communication and organization while also offering features to help optimize training sessions.

## 1.2 Scope

PocketSports will serve coaches, players, and parents by offering a set of tools for team management, drill creation, practice planning, and performance tracking. The app will support multiple sports such as basketball, lacrosse, and volleyball. PocketSports aims to provide customizable features to suit different team needs. The platform will include a drill library, team management functionalities, and advanced AI/ML-driven insights. The app will be accessible through a web-based interface, ensuring flexibility and ease of access.

## 1.3 Definitions, Acronyms, and Abbreviations

PocketSports: The name of the digital coaching app.

AI: Artificial Intelligence.

ML: Machine Learning.

UX/UI: User Experience/User Interface.

CSV: Comma-Separated Values.

JSON: JavaScript Object Notation.

MERN stack: MongoDB, Express, React, Node

## 1.4 References

- Agile Software Development methodologies for iterative software design cycle.
- Sports Coaching best practices (various sports organizations and publications).

### 1.5 Overview

This document offers a comprehensive overview of the PocketSports application, detailing its core features and specific requirements. It outlines the product's purpose, scope, and functionality, while also addressing key technical and operational requirements. In the following sections, we will explore the product's constraints and the primary functions it supports. The first section will give a brief overview to the purpose and scope of the product as well as any key definitions and references. In the second section, we will be focusing on the overall description of the product such as the functions and overall constraints. A deeper dive into Reliability requirements and Higher-order Language Requirements will also be included. In the third section and final section, the external interfaces will be elaborated with functional, non-functional, and performance requirements. objectives. This document is essential for anyone involved in the design, development, or implementation of the PocketSports app.

## 2. Overall Description

### 2.1 Product Perspective

The PocketSports application is designed as a digital solution for managing sports teams, practices, and player development, specifically targeting coaches, players, parents, and team owners. Standing as a comprehensive coaching tool, it integrates several existing tools into one unified platform, simplifying team management and communication. The product is intended to function as a web application, providing accessibility across devices. PocketSports aims to streamline traditional team operations like drill creation, practice management, and performance tracking, offering positive user experience and real-time collaboration. The app's design also supports growth by incorporating additional sports and features, making it a scalable platform for various levels of sports organizations.

### 2.2 Product Functions

Some of the product functions include Team Management, where users can create teams, assign roles (owner, coach, player, parent), and manage rosters efficiently. Team owners have the ability to invite members via email and control team settings. Another product function would be drill/play creation. In drill/play creation, coaches can design custom drills/plays using interactive tools like animated field views, tags, and drag-and-drop functionalities for strategic planning. Another core product function hovers around practice plan management where coaches can create and edit practice plans, pulling from a bank of saved drills. More core functions include stat tracking and goal setting. With stat tracking and goal setting, both team-level and individual stats can be tracked over time. Coaches and players can set performance goals and monitor progress through visual reports and graphs. The final feature would be Year Planner. In the year planner, coaches can allocate focus areas, track team goals, and ensure alignment with performance objectives throughout the season.

## 2.3 User characteristics

### 2.3.1 The Player

The users of this app that are players will be active members of an organized sports team in an official league with other teams. These players must hold a position on a team, and can hold multiple positions if their skills apply to several positions on the team. The player will belong to a roster of a team of no less than 1 other player, directed by no less than one coach. In use of the app, the players will be able to set goals for themselves for a season. The players will be able to track statistics from each game that they played, given to them via a box score window that will appear on the app. Players will look at their statistics over multiple games and use their knowledge of their skills as well as these statistics to make the most educated decisions about their goals. The players can also view practice results, and how these results affected the goals that they set. Via the schedule feature, players can easily view every upcoming event for the season including practices, games, and other team-related events. The player should not be required to have highly developed technical skills, as the app should be completely feature-driven through a simple point-and-click UI.

### 2.3.2 The Coach

The users of this app that are coaches will be active designated coaches of a team from some governing organization of an official sports league. For a team using Pocket Sports, there must be at least one coach for the team. The coach will serve as the administrator of the app, setting games and practice times for the players of the team. The coach must be active in ensuring the calendar feature is updated with current practices, games, and time/location changes to either of those events. The coach will use the statistics feature and the team roster feature to create goals for the team as well as for individual players if they wish. Using the Top Performers feature on the homepage as well as a number of other sources, the coach can create well-educated goals that better align with the directives of the entire team. With these features, the coach can heavily benefit from aligning his views with those of the players. The coach will not be required to have a high technical ability, besides being able to utilize the point-and-click UI to use the features for their players.

### 2.3.3 The Owner

The owner's user experience will be very similar to that of the coach, with some elevated administrator privileges. For example, if a coach were to be fired or somehow removed from the position, the owner of the team would be responsible, as the overall administrator, for removing them from the system and adding in any new individuals to replace that position. The owner will also be responsible for keeping the coaches and players updated on league information, as the owner will likely be the one meeting with league officials and managers of the overall division/league itself. As the coach is like an administrator for the application, the owner can be thought of as a super administrator, managing the administrators as they are changed throughout the lifetime of the team. The owner will have the ability to add/remove coaches as well as players in the absence of a coach. The owner will have the same notification privileges as a coach, allowing them to keep the team (players, coaches, and parents included) updated with the most recent information about games, schedule changes, etc.

### 2.3.4 The Parent

The parent will have a similar user experience to the player, with less privileges than a standard member of the team. The parents will be able to designate themselves to a player, and keep updated with the goals they create, the progress of those goals, as well as the games and schedules of the team. The parents can contact the coaches/owners with concerns of their player so that the coaches and owners can be held accountable for keeping everyone updated with the most recent information. The parents will be able to view statistics and other things a standard player can view, without being able to update/change any of the goals themselves. Those features are reserved for the players themselves.

## 2.4 Constraints

There are several constraints that come into play with the PocketSports app.

### 2.4.1 Regulatory Policies

The application must adhere to the U.S. Government Section 508. The PocketSports App must be usable by users with disabilities including blindness, deafness, or other applicable disabilities.

### 2.4.2 Hardware Limitations

The PocketSports app will be limited to use on three different groups of hardware devices: Personal Computers (including laptops), tablets, and smartphones. The devices running the application are required to have at least 4 GBs of RAM, 15 GBs of SSD, and must have a display to view the UI of the application

### 2.4.3 Interfaces to other Applications

The PocketSports application must be able to connect and interface with several different applications. The database holding information for the coach and players will use MongoDB, an external service that must be connected to create, retrieve, update, and delete data from the application. For contact of the players by the coach for new goals, updated practice/game times and locations, and other information, the system will use an email service to send notifications when things are changed.

### 2.4.4 Parallel Operations

The PocketSports application must be able to avoid race conditions or deadlocks when parallel operations are occuring. If a database is being updated or things are being added, the rest of the operations must still be available for use to either players or coaches

### 2.4.5 Audit Functions

The audit and event tracking of the PocketSports app must be able to track events of the application such as team additions, new practice plans, etc. If a coach implements a new practice plan, that plan must be accessible at all times until the plan is deleted from the application.

### 2.4.6 Control Functions

The PocketSports app must implement several controls for its features to ensure quick and efficient addition, updating, and deletion of resources. For building rosters/adding players, the app must be able to do so in a quick manner, and hit no errors in adding players that do not previously exist on the team. The app must be able to accordingly update the schedules for players as the coaches add events to the calendar. The app must be able to validate information of the players to ensure no players are being added to the roster that do not exist. The practice plan functionalities must work together with the practice results and goals feature to allow concurrent information between the three. All inputs to practice plans such as time, repetition, etc. must be validated within certain ranges.

### 2.4.7 Higher-order Language Requirements

The PocketSports application must use languages that support its dynamic features and allow for multiple processes to be run concurrently. The choice of language must be able to support external data types such as exporting PDFs of practice plans and sending email/phone notifications for changes to events in the application.

### 2.4.8 Signal Handshake Protocols

For the transfer of critical data between points of the application, there must be acknowledgement from the receiver to the sender of the data that it was received and digested properly. For notifications and information being sent in bulk, the system must be able to handle occurrences of overload where messages cannot be sent. The system must then queue the messages until the overload has been remedied and then continue sending the data.

### 2.4.9 Reliability Requirements

The system must be fault-tolerant, in that when an overload, network error, or other kind of system failure occurs, there shall be alternate instances of the application that will take the place of the active application server. The system must recover from failures within 1 hour of the failure occurring, offloading the data to an alternate instance of the application.

### 2.4.10 Criticality of the Application

If a system failure occurs, there is no threat to lives or wellbeing of coaches or players, however it may greatly affect the concurrency of the team through the players and coaches. Coaches may have to alternate to other forms of more complex communication for getting information to their players.

### 2.4.11 Safety and Security Considerations

Although there is little safety concern with the use of the application, there are heavy security constraints that must be implemented for the protection of user data. Because the signup and login for the app includes emails, and the app requires saved passwords for access to its features, there must be encryption of the database through code methods, such as Javascript's bcrypt library. This library allows simple hashing of passwords for web applications that require signup and login.

## 2.5 Assumptions and dependencies

For the use of the application, several assumptions are made about the system, the user, and other aspects of the software. The system is connected to a wireless network for communication between the coaches and players, as well as communication between players. Any device the system is running on is assumed to have sufficient SSD and RAM to run the application's processes and store application data, as well as running the Windows or iOS operating system. The team's configuration is assumed to be one single head coach (with possible assistant coaches) and more than one player on the team. The application is also assuming that there is access to box score statistics after the completion of each game via a website, API, or some other kind of access to the data.

# 3. External Interfaces

## 3.1 External Interfaces

PocketSports Digital Coaching App has two main external interfaces, the player user interface and the coach user interface. Both have different functions and features that benefit the users.

### 3.1.1 Player User Interface

The first feature of the user interface (UI) is the goal tracking system, located on the dashboard. Through visual aids, players can track their progress and view updates or insights to help improve their skills. To create a goal, the player clicks the "Create Goal" button and inputs personal goals, including the name and the target amount. The goal will then appear on the dashboard for tracking and will be stored in a database. It will automatically update as the coach or player enters stats.Each goal is customizable to fit individual needs, but players cannot input zero or negative values. The units of measurement depend on the sport and specific objectives the player sets. For instance, in basketball, users can choose to set goals related to three-pointers, free throws, and more. Although there is no time limit for goal creation, if the player does not save the goal, it will not be stored in the database or displayed on the dashboard.Once a goal is achieved, it will transform into an achievement or award and will be removed from the active goals list. To complete a goal, the player must input the relevant information when editing a goal, but can only upload information for personal goals. The coach is the only one who can update goals for the team. The UI is designed to be clean and intuitive, ensuring players can easily track progress and view insights. A button will be provided to create new goals, with a small pop-up form where players can fill in goal details. Dropdown menus will allow them to choose units, and fields will enable them to enter the name and target value of the goal. A "Save" button will store the goal and display it on the dashboard. Additional buttons will allow users to filter between all goals, team goals, or personal goals. Upon completion, achievements will be displayed as badges, and the total goal completion rate will be visualized in a diagram. Goal amounts must be entered as integers, not words, so the system can accurately process them into percentages and provide updates. Once a goal is successfully saved and stored in the database, a success message will appear. If there is an error in saving or an invalid input, the interface will display an error message.

The second feature is the practice schedule, which provides players with detailed information on upcoming practices, including the time, location, and the practice plan for the day. The coach inputs this information through their system, which is then displayed on the player's interface. Players can browse practices and games throughout the season by searching through specific months, but cannot view dates outside the season's timeframe.Practice plans and related information are only displayed for days that the coach has scheduled and saved in the system, with all data securely stored in our database. Players will interact with a calendar interface, allowing them to select a specific day, month, or year. Upon selecting a date, the practice details and the coach's planned exercises for that day will appear below the calendar.To enhance usability, players will also receive reminders for upcoming practices and games, helping them stay prepared and engaged throughout the season.

The third feature is the practice results section, where players can review any stats or feedback recorded by the coach during practice. Players can select a specific date within the team's season to view practice details, including any stats or coach feedback from that day. Once a valid date is chosen, the system will automatically retrieve and display the data from the database, provided the coach has entered and saved the information. A small calendar will be available for date selection, and next to the calendar, a detailed list of exercises completed during practice will be shown, along with the corresponding stats and individualized coach feedback. All data is formatted as MM/DD/YYYY to ensure consistency and ease of reference.

### 3.1.2 Coach User Interface

The coach user interface is for a user who is the coach of a team and wants to be able to track their team goals and create drills/schedules. The user will be able to create any team goals and design, execute, and analyze practice plans throughout the sports season. The coach will also be able to manage and communicate with the players any stats or feedback.

The first feature on the user interface (UI) is the goal tracking dashboard, designed to help coaches create and track team progress toward achieving season goals. Using visuals, coaches can update stats and monitor the team's journey toward their objectives. To create a goal, the coach selects the "Create Goal" button, where they can input specific team goals for the season. This includes naming the goal and specifying a target value, which will then be displayed on the dashboard for both the coach and selected players to track progress. All goals are stored in a database and automatically updated when new stats are entered. Goals can be tailored to each team's needs, though values cannot be zero or negative. The measurement units will vary depending on the sport and the nature of the goal. For example, in basketball, a coach can set goals around three-pointers or free throws. There are no time restrictions for entering a goal, but if a goal isn't saved, it will not appear on the dashboard or in the database. Once a goal is achieved, it will convert into an achievement/award and no longer appear in the active goals list.To update progress, coaches can input stats that are relevant to the goal. The UI is designed to be intuitive and dynamic, ensuring that the layout is clean and organized, offering quick insights into progress. The "Create Goal" button opens an interactive form where the coach can input the goal's details, including a dropdown for units and a text field for the goal's name. The "Save" button stores the data in the database and displays the goal on the dashboard. Coaches can also update existing goals by clicking an edit button on the goal, which opens a form where they can adjust the target or enter new progress. Once completed, goals are marked as achievements, displayed similarly to badges, and the team's progress will be visualized in a diagram. The input for units and progress must be numeric to allow the system to calculate percentages and track updates accurately. Upon successfully saving a goal to the database, the interface will display a success message. If the goal fails to save or an input is invalid, the system will display an error message, ensuring smooth interaction with the UI.

The second feature is the roster display on the roster page, which showcases the team roster and statistics. When the user clicks on the Roster tab, they are directed to a comprehensive list of players and coaches, all sourced from the database where the coach has previously input team member information.Upon selecting a player, the user will automatically see detailed information about that player, along with their individual statistics. Player stats will be prominently displayed at the bottom of the screen, while additional player information will appear directly beneath their name, creating a clear and organized layout for easy access to relevant details.

The third feature is the practice schedule for coaches, providing details on when and where practices occur, along with the daily practice plan. This information is generated from the coach's input when creating a new event, which is then communicated to all players. When a coach initiates the creation of a new event, a form will pop up with several fields to fill out, including the event name, location, date, and time. Additionally, the coach can specify the visibility of the event, choosing which players or coach assistants will have access to this information. Once the coach saves the details, they are stored in the database. A success message will appear if the information is successfully uploaded, while a failure message will indicate any issues, such as incorrect or missing fields. Coaches can search for practices and games throughout the season's months, accessing details for specific days; however, they cannot search outside the designated season range. Only those practices and events that the coach has inputted and saved will be displayed. A calendar interface allows the coach to select the desired day, month, and year, with relevant information shown below the calendar, including a list of exercises the coach has scheduled for that practice. To ensure preparedness, users will also receive reminders about upcoming practices and games.

The fourth feature focuses on creating and editing practice plans for the team. Coaches will have the ability to design and customize practice plans using various drawing functions and features. When initiating a new drill, a field view will be presented, allowing the coach to draw the drill and input any relevant information. Upon saving the drill, both the drill and the overall practice plan will be stored in the database, making them accessible on the plans dashboard. A success message will confirm that the plan has been saved correctly, while a failure message will indicate any issues with the upload process. Once saved, coaches can assign these practice plans to specific sessions, ensuring that the plans are displayed on the players' user interface.

### 3.1.3 Login

Users will be directed to a login page where they must enter the email and password associated with their account. This information will be submitted to the system for validation against the database. If the credentials match an existing account, the user will be granted access. However, if the information is not found, access will be denied. For users without an existing account, a registration page is available. Here, they will need to provide needed information, including their name, role (coach or player), email address, and password. To ensure accuracy, users must re-enter their password, and once all information is completed, it will be saved in the database. In the event that a user forgets their password, they can navigate to a password recovery page. This page will allow them to reset a password by sending an email containing a link. After following the link, users can enter and confirm a new password, which will be saved to their account for future access.

## 3.2 Functional Requirements

When a user submits their login credentials, the system shall verify the input against the stored data in the login database. If the credentials match an existing record, the system shall grant the user access to their account. If the credentials are invalid, the system shall display an error message prompting the user to re-enter their information.

When a user registers for an account, the system shall validate the provided email to ensure it is in a proper format and verify that the password meets the required criteria: a minimum of 8 characters, including at least one uppercase letter, one number, and one special character. Once the user submits the information, the system shall process and store the valid credentials in the login database for future authentication.

If a user forgets their password, they can navigate to the password recovery page. The system shall send a password reset link to the email address stored in the database. The link will prompt the user to enter and confirm a new password, which must meet the required criteria: a minimum of 8 characters, including at least one uppercase letter, one number, and one special character. Once validated, the new password will replace the existing one in the database for the corresponding email.

For any date input related to schedule or practice results, the system shall validate the date and check if there is any associated data stored in the database. If data exists for the selected date, the system shall retrieve and display the relevant information to the user. If no data is available, no information will be shown. Additionally, the selected date must fall within the defined season's range. Any dates outside of the season will be rejected, and the system will prompt the user to enter a valid date within the season.

For any buttons that create new objects, such as goals, events, plans, or roster entries, the system shall display a form for the user to input the required information. Once the form is completed and submitted, the system shall process and store the data in the appropriate database. Any numerical fields will be validated to ensure they contain valid numbers. For goals, numerical values will be used to track the percentage of goal completion.

When a user clicks on any object to display information, the system shall query the database to retrieve the relevant data associated with that object. The retrieved information will be displayed on the screen. If no related data is found, no information will be shown.

When a user edits existing objects, such as plans or goals, the system shall process the new input and update the corresponding information in the database. Any numerical fields will be validated to ensure they contain valid numbers. For edited practice plans, users will be able to add drawings and continue creating drills from where they left off. Once the user saves the updates, the system shall store the revised plans and drawings. Any drawings associated with a practice date will be automatically updated for players to view.

## 3.3 Performance Requirements

This section outlines the performance requirements for the sports team management application, detailing both static and dynamic numerical metrics that are essential for optimal functionality and user experience. These metrics are essential for ensuring that the application remains reliable, efficient, and user-friendly, thereby enhancing the overall experience for all users involved in sports team management.

### 3.3.1 Terminal Support

The application must support a minimum of 500 terminals simultaneously, accommodating a wide range of devices including desktops, tablets, and smartphones. This ensures accessibility for all users, regardless of their device choice.

### 3.3.2 Simultaneous User Support

The system should be capable of supporting up to 1,000 simultaneous users without degradation in performance. This requirement is critical to ensure that coaches, players, and parents can access the application concurrently during peak usage times, such as practice schedules or team announcements.

### 3.3.3 Information Support

The application must efficiently handle a data volume of up to 10,000 records per team, encompassing user accounts, team rosters, drills, practice plans, and performance metrics. This includes various data types, such as text, images, and video, ensuring a comprehensive database that supports dynamic content and reporting features.

### 3.3.4 Dynamic Requirements

The application should achieve a response time of less than 2 seconds for 95% of user interactions, ensuring a smooth user experience. The system must be capable of processing user inputs and updates in real-time, with a maximum lag of 1 second for updates to team rosters and practice plans.

## 3.4 Non-functional requirements

The non-functional requirements for the PocketSports application are essential to ensure that the system meets performance standards and provides a quality user experience. These requirements encompass various aspects such as usability, security, maintainability, and compliance.

### 3.4.1 Usability

The UI must be intuitive and easy to navigate for both coaches and players, allowing for efficient access to features with minimal training required. The design should adhere to best practices in UX/UI, ensuring that information is presented clearly.

### 3.4.2 Security

User data must be encrypted both in transit and at rest. The application will employ industry-standard encryption methods, such as bcrypt, to protect sensitive information. The application must implement secure user authentication protocols, including multi-factor authentication (MFA) for coaches to prevent unauthorized access. Role-based access control will ensure that users can only access features relevant to their role.

### 3.4.3 Maintainability

The codebase must adhere to established coding standards and be well-documented to facilitate future updates and maintenance. This will include comments in code and separate documentation for each module. The application should follow a modular design that allows for easy addition or modification of features without affecting the entire system. This will aid in reducing maintenance effort and improving scalability.

### 3.4.4 Reliability

The system must be designed to handle failures gracefully, with backup mechanisms to ensure continuous availability. If a failure occurs, the system should automatically switch to a backup instance, maintaining functionality for users. Regular backups of user data should be implemented, allowing for recovery in case of data loss. The application must ensure that data can be restored within a defined timeframe, minimizing downtime.

### 3.4.5 Compliance

The application must adhere to relevant legal and regulatory requirements, such as GDPR for user data protection and any specific regulations applicable to sports organizations. The system should maintain audit logs of user actions, providing a traceable history of changes made to team rosters, practice plans, and goals. This will aid in compliance and security audits.

### 3.4.6 Interoperability

The application must support integration with third-party applications and services, such as email notifications and performance tracking tools, via well-defined APIs. This will enhance functionality and user engagement.

### 3.4.7 Portability

The application should be accessible on various devices and operating systems, including Windows, macOS, iOS, and Android. This ensures that users can access the app from their preferred devices without compatibility issues.

### 3.4.8 Localization

The application must support multiple languages to cater to a diverse user base. Localization efforts should include translations of UI elements and content, as well as the ability to format dates and numbers according to regional preferences.