Project Initiation Document

TeamSync

# Contents

# Project Overview

## Project Name

TeamSync

## Project Owner

Jamal Ahmed

## Project Summary

### Project Description

TeamSync is a mobile application designed to streamline organization of football sessions by automatically assigning players to teams based on their individual ratings. The primary goal of TeanSync is to ensure that teams are balanced, leading to more competitive and enjoyable matches. This app will be tailored to people that regularly organize games and want to remove the hassle of manually balancing teams. Users can also see how well their performance is.

## Target Audience

**Casual and Semi – Competitive Football Players**

Individuals who participate in regular football sessions, whether it is for leisure or semi-competitive environment. These users seek a convenient way to ensure fair and balanced teams, enhancing the quality of their games.

**Match Organizers and Team Captains**

People responsible for organizing football sessions, with small groups of friends or local club.

**Youth and Amateur Clubs**

Coaches and administrators of youth or amateur football clubs who want to manage internal scrimmages or friendly matches with balanced teams, without the need for manual intervention.

## Long-term Vision

**Expansion to Other Sports**

The base feature of team-balancing can be implemented into other sports. The first choice other sport would be basketball; Hockey, lacrosse and rugby will be considered. This would allow TeamSync to become a versatile tool.

**Enhanced Analytics**

Future iterations and updates can include analytical features that offer detailed insights into performance. Using additional data (potentially from extra statistics provided and video data) Heatmaps, recommended positions, personalized tips, and basic match analysis.

**Social Features**

Allow users to find local teams, clubs and sessions. Furthermore, allow clubs to showcase trials and information to nearby players and groups. Give users the ability to create groups, schedule matches and tournaments.

**Monetization and Partnerships**

When the features above are implemented, request for premium subscriptions for potentially professional organizations using the application. For example, high tier premium subscription for enhanced analytical use. Also, for a larger integration of matches and players may request licensing, for example power league – licensing a custom/enhanced version of TeamSync to allow players, matches, leagues and tournaments to integrate seamlessly.

# Project Objectives

### Primary Objectives

* Create an easy-to-use mobile app that automatically balances teams based on individual player performance and ratings.
* Ensure that the app provides fair and competitive match setups, reducing the manual effort required by match organizers.
* Display key player statistics to allow users to track their performance over time.

### Secondary Objectives

* Integrate a basic MMR (Matchmaking Rating) system to track user progression.
* Offer users performance insights, such as roles based on their best skills and areas to improve.

# MoSCoW

### Must

* Automatic team balancing based on real-time player statistics and MMR.
* Ability to manually enter and update player performance after matches.
* Basic statistics display (goals, assists, etc.).
* App deployment on both Android and iOS.

### Should

* Add user roles based on individual performance, such as defense or offense-oriented roles.
* Provide detailed insights after each match, e.g., top performers.

### Could

* Integration with external APIs for player tracking (e.g., health data or advanced performance).
* Enhanced analytics including heatmaps or advanced video analysis.
* Social features like connecting teams or organizing larger tournaments.

### Will not

* Monetization (initial release will be free for users).
* Expansion to other sports or team types beyond football.

# Planning and Action Steps

### Immediate Next Steps

* Set up the development environment and repositories for both the front and backend (React Native and ASP.NET Core).
* Define the data models for player stats and match organization.
* Start working on the team-balancing algorithm (based on MMR and role classification).
* Produce a Software Architecture and design document.

### Development Steps

* **Front-end:** Design the basic mobile app interface (React Native). Use the Software Architecture and Design Document to build user flows for match creation, player stat entry, and team visualization.
* **Back-end:** Create API endpoints to handle player stats, match history, and team creation logic (ASP.NET Core).
* **Database:** Set up MongoDB to store player profiles, match history, and session data.

### Testing

* Set up Jest and React Native Testing Library for front-end tests and xUnit for backend testing.
* Unit test the team-balancing algorithm to ensure fairness and avoid large skill disparities.
* End-to-end testing to simulate full user sessions (from match creation to match completion).
* Usability testing to ensure the app is easy to use for casual and semi-competitive players.

### Deployment

* Deploy the backend to Heroku, and prepare the mobile app for release on the App Store and Google Play.
* Set up CI/CD pipelines using GitHub Actions for automated testing and deployments.

### Post-Launch Maintenance

* Monitor app performance, document user feedback and provide support options
* Schedule bi-weekly updates for bug fixes and feature additions.
* Implement advanced analytics features as part of future updates based on user demand.

# Success Criteria

### Functionality

* Teams are automatically balanced based on player ratings and stats.
* Users can enter and view their match performance data.
* The app is deployed and fully functional on both Android and iOS.

### Usability

* Users should be able to create a match session in under 2 minutes.
* The app should have an intuitive interface for viewing stats and managing sessions.

### Performance

* The app should load in under 3 seconds, even with large databases of player data.
* Match balancing should be completed in under 5 seconds.

### User Feedback

* Positive feedback on the fairness and balance of teams.
* Users report improved match quality due to balanced teams.

# Risks and Considerations

### Risks

**Balancing Algorithm Complexity:** Difficulty in achieving perfectly balanced teams across various skill levels could lead to user frustration.

**Adoption Rate:** If users do not find the app easy to use, they may revert to manual team balancing.

**Data Accuracy:** Inaccurate stats input from users may affect team balancing.

### Considerations

* Regular user feedback should be gathered post-launch to fine-tune the balancing algorithm.
* Scaling issues if user adoption exceeds initial expectations.

# Personal Motivation and Goals

### Learning Objective

Gain experience in building a full-stack mobile app with React Native and ASP.NET Core.

### Long-Term Vision

* Establish TeamSync as a core tool for small football leagues and casual players.
* Expand to other sports with minimal adaptation of the core functionality.
* Expand app functionality to potentially work with organizations and football clubs.

### Personal Challenge

* Ensure that the app remains efficient and fair in team balancing, even with many users.
* Successfully integrate complex data from multiple sources, gain proficiency in full stack .NET app development.

# Notes and Ideas for Future Updates