FootballGuru

Специфікація вимог до програмного забезпечення

Версія 1

10.04.2025

Шрамко І.В.

**ЗМІСТ**

[1 ВСТУП 3](#_p78hhgvnvu46)

[1.1 Огляд продукту 3](#_u2oplyael6tr)

[1.2 Мета 3](#_ifooyq28iuly)

[1.3 Межі 4](#_684ks0wk4gxk)

[1.4 Посилання 5](#_6ubrbq9kmr83)

[2 ЗАГАЛЬНИЙ ОПИС 6](#_syu9ocqrnaq)

[2.1 Перспективи продукту 6](#_livkc5sibhji)

[2.2 Функції продукту 6](#_giejusux3g5k)

[2.3 Характеристики користувачів 7](#_eclvmfspv9uy)

[2.4 Загальні обмеження 8](#_ucib5osacb3)

[2.5 Припущення й залежності 8](#_3ns4zx1j9hoj)

[3 КОНКРЕТНІ ВИМОГИ 10](#_8bm06jszionv)

[3.1 Вимоги до зовнішніх інтерфейсів 10](#_jklcnw9pz8c5)

[3.1.1 Інтерфейс користувача 10](#_abqt9s8bem26)

[3.1.2 Апаратний інтерфейс 10](#_q4zns6p6xtu1)

[3.1.3 Програмний інтерфейс 10](#_viz7anara0b7)

[3.1.4 Комунікаційний протокол 11](#_ut6d6hr2cfjc)

[3.1.5 Обмеження пам’яті 11](#_hixka0nnl6r1)

[3.1.6 Операції 11](#_qxlvac4nyw6q)

[3.2 Властивості програмного продукту 12](#_mrm8mpiv3l5p)

[3.3 Атрибути програмного продукту 12](#_oynjdlcvme3y)

[3.3.1 Надійність 12](#_oxkdxn2hi16y)

[3.3.2 Доступність 13](#_5atecwqbyjd7)

[3.3.3 Безпека 13](#_fucq89gnj071)

[3.3.4 Супроводжуваність 13](#_m66q14xfyqnf)

[3.3.5 Переносимість 13](#_mwvd5m21659i)

[3.3.6 Продуктивність 14](#_9gz8u4ssx2rj)

[3.4 Вимоги до бази даних 14](#_m36i38h6wc2v)

[3.5 Інші вимоги 15](#_d4lmi1iha8n9)

# **1 ВСТУП**

## 1.1 Огляд продукту

Програмна система FootballGuru призначена для автоматизації процесу підготовки та проведення тренувань на міні-футбольному полі з урахуванням реальних тактичних вимог. До її складу входять три мобільні застосунки, реалізовані на базі .NET MAUI: застосунок «Тренер» для планування та керування тренуванням, застосунок «Камера» для локальної обробки відеопотоку та «Екран» для відображення повідомлень гравцям. Серверна частина виконана як ASP.NET Core Web API з монолітною багаторівневою архітектурою і використовує SQL Server та MongoDB для зберігання даних.

Для обміну даними між мобільними вузлами система використовує автономну P2P-мережу на основі Wi-Fi Direct/TCP, що гарантує надійний зв’язок за відсутності Інтернету та передачу повідомлень у форматі JSON. Алгоритм обробки відеопотоку з EmguCV фільтрує кадри, визначає контури поля й гравців за кольором пов’язок, розраховує їх координати й відстані відносно меж поля. У разі виходу гравця за межі зони відповідальності застосунок «Камера» формує повідомлення про порушення і надсилає його для відображення у застосунку Екрану .

Інтерфейс користувача побудований за принципами MVVM і адаптивного дизайну, забезпечує інтуїтивну навігацію, швидкий доступ до створення схем, призначення ролей, зон та інструкцій.

## 1.2 Мета

Головною метою розробки системи FootballGuru є створення інтегрованого інструмента, який забезпечить тренерів можливістю ефективного контролю тактичної дисципліни команди в режимі реального часу та оперативного зворотного зв’язку гравцям під час тренувань. Система покликана автоматизувати наступні ключові процеси:

* Моніторинг позицій гравців на полі з використанням комп’ютерного зору.
* Автоматичне оповіщення гравців про вихід за межі визначеної зони відповідальності на основі порівняння фактичних координат із заданими межами.
* Поширення інструкцій тренера: відправлення попередньо підготовлених або довільних текстових повідомлень із підсвічуванням за кольором пов’язки гравця, що дозволяє швидке сприйняття коригувальних вказівок.

Досягнення цієї мети забезпечить тренеру можливість не лише планувати вправи й візуалізувати тактичні схеми, але й автоматично відслідковувати їхнє виконання без постійного візуального контролю, що суттєво підвищить продуктивність і якість тренувального процесу.

## 1.3 Межі

Незважаючи на широкі функціональні можливості, платформа FootballGuru наразі обмежена підтримкою мобільних пристроїв під управлінням операційної системи Android: і застосунок «Тренер», і клієнти «Камера» та «Екран» працюють виключно на Android-платформі.

Хоча «Екран» спроектовано для відображення на телевізійних пристроях, обрано Android як універсальну ОС, що дозволяє запускати його на будь-яких Android-тюнерах чи приставках без потреби адаптувати під кожну пропрієтарну платформу телевізора.

Крім того, алгоритм кольорової ідентифікації гравців потребує попереднього калібрування під конкретні умови освітлення та навколишнього середовища: система наразі не адаптується автоматично до змін світлових параметрів і вимагає ручного налаштування порогових значень для коректного розпізнавання маркерів кольору.

Ці межі визначають поточну область застосування рішення і становлять базу для подальшого розширення його сумісності й адаптивності в наступних ітераціях.

## 1.4 Посилання

Проект FootballGuru будується на перехресті сучасних трендів у спортивних технологіях. Використання підходу з відстеженням позицій у реальному часі та аналізу тактичних помилок вирізняє систему серед існуючих аналогів і відповідає потребам цифрової трансформації тренувального процесу.

Платформа може бути легко інтегрована в інфраструктуру спортивних академій, тренувальних центрів і футзальних клубів для централізованого обліку показників виконання, збереження історії вправ та побудови аналітичних звітів. Особливо корисною вона стане для команд із високими вимогами до дисципліни та швидкого зворотного зв’язку, а також для молодіжних і юнацьких програм, де ключовим є своєчасна корекція помилок.

# 2 ЗАГАЛЬНИЙ ОПИС

## 2.1 Перспективи продукту

FootballGuru закладає фундамент для подальшого розвитку інтелектуальних спортивних рішень. У короткостроковій перспективі передбачено покращення процесу відслідковування та розпізнавання об’єктів на відео та розширення функцій аналітики: автоматичне збереження історії тренувань та побудова візуальних звітів на основі зібраних координат і порушень зон відповідальності. У середньостроковій – інтеграцію з зовнішніми сенсорними пристроями (GPS-трекери, акселерометри) та системами розпізнавання жестів для поглибленого аналізу рухів гравців. Довгостроково платформа може перетворитися на хмарну екосистему, яка об’єднає декілька команд і клубів, дозволяючи порівнювати статистику за сезонами та виступами, а також реалізувати модулі машинного навчання для прогнозування ризиків травм і розвитку індивідуальних тренувальних планів.

## 2.2 Функції продукту

Система реалізує наступні ключові функції, що забезпечують досягнення поставлених цілей:

* Управління гравцями та ролями: система забезпечує функціонал для управління записами про гравців та ролей. Тренер може додавати нових гравців, змінювати їхні персональні дані, призначати ролі (нападник, захисник, півзахисник тощо), доповнювати ролі індивідуальними інструкціями або видаляти застарілі.
* Визначення та налаштування зон відповідальності і конфігурація тренувань: через інтерфейс застосунку «Тренер» користувач може поділити віртуальне поле на секції та призначити кожному гравцю власну зону відповідальності.
* Автоматичне сповіщення про вихід із зони: перед початком вправ тренер задає для кожного гравця індивідуальну зону відповідальності на полі. Застосунок фоново відстежує положення кольорової мітки гравця У разі виходу за межі дозволеної області застосунок негайно відправляє сигнал про порушення.
* Надсилання інструкцій і повідомлень гравцям: тренер має можливість у будь-який момент надсилати гравцю коротке текстове повідомлення або обрати одну з попередньо створених команд-наборів. Повідомлення передається у напрямку застосунку «Екран», де воно відображається. Це дозволяє миттєво сповістити про необхідність коригувати дії на полі без додаткових голосових чи текстових каналів поза системою.

Загалом, ці функції утворюють єдину інтегровану платформу, що дозволяє тренеру не лише планувати та конфігурувати тренування, а й у реальному часі стежити за виконанням тактичних завдань і своєчасно коригувати дії гравців.

## 2.3 Характеристики користувачів

Тренер – досвідчений фахівець із тактичної підготовки, який використовує систему на планшеті або мобільному пристрої для створення схем, зон та інструкцій. Очікується, що тренер володіє базовими навичками роботи з мобільними додатками та має досвід планування тренувань.

Гравець – користувач який впродовж гри буде бачити на «Екрані», повідомлення про відхилення та інструкції в процесі вправ. Повідомлення мають бути максимально лаконічними та помітними, щоб їх можна було швидко побачити в процесі руху.

Технічний адміністратор – відповідає за розгортання серверної частини, підключення БД, оновлення програмного забезпечення та керування правами доступу.

## 2.4 Загальні обмеження

Загальні обмеження системи:

* Підтримуються пристрої з ОС Android та Android TV.
* Процес тренування має змогу відбуватись без підключення до Інтернету.
* Всі дані зберігаються на сервері; резервне копіювання має виконуватися вручну.
* Наявність механізмів обмеження доступу та безпеки.
* Врахування обмеженого обчислювального ресурсу мобільних пристроїв.

## 2.5 Припущення й залежності

При розробці системи були враховані наступні припущення та залежності: наявність Wi-Fi Direct на всіх мобільних пристроях, по можливості із сертифікованими стеками P2P, Виробники пристроїв зберігають API Wi-Fi Direct без непередбачених змін у прошивках.

Стабільна робота камери – рівень освітлення, контрастність та мінімальна вібрація не порушують точність пошуку контурів.

Підтримка серверного середовища (.NET 8.0, ASP.NET Core) та баз даних (SQL Server 2019+, MongoDB 5.0+) – оновлення цих компонентів відповідає політиці вендорів щодо сумісності.

# 3 КОНКРЕТНІ ВИМОГИ

## 3.1 Вимоги до зовнішніх інтерфейсів

### 3.1.1 Інтерфейс користувача

Мобільні застосунки «Тренер», «Камера» та «Екран» реалізовано на базі .NET MAUI із шаблоном MVVM. Інтерфейс повинен бути інтуїтивно зрозумілим та адаптивним до різних розмірів екранів. Для тренера передбачені екрани планування тренування (вибір схеми, призначення гравців, встановлення зон, інструкцій), перегляд активних сесій і відображення списку підключених пристроїв. Інтерфейс «Камери» та «Екрану» мінімалізовано: лише статус підключення, сторінка отримання параметрів і сповіщення про порушення або інструкції .

### 3.1.2 Апаратний інтерфейс

Для коректної роботи потрібен модуль камери з достатньою роздільною здатністю та частотою оновлення кадрів. Пристрої мають підтримувати Wi-Fi Direct і TCP/IP-стек. Для збереження локальних файлів (конфігурацій JSON/XML) застосунок «Тренер» повинен отримати дозвіл на доступ до файлової системи пристрою .

### 3.1.3 Програмний інтерфейс

Вся серверна логіка реалізована з використанням REST API, що надає стандартизований механізм взаємодії між клієнтською та серверною частинами. Комунікація відбувається через HTTP-запити, а дані передаються у форматі JSON. Це дозволяє легко масштабувати систему або інтегрувати її з іншими сервісами в майбутньому. Усі запити проходять перевірку на автентифікацію, яка реалізована за допомогою JWT-токенів. Таким чином досягається гнучкість, безпека та підтримуваність системи.

### 3.1.4 Комунікаційний протокол

Між мобільними вузлами («Тренер» - «Камера» - «Екран») використовується peer-to-peer мережа Wi-Fi Direct із передачею даних по TCP, що гарантує мінімальні затримки й автономність без Інтернету. Клієнт–серверна взаємодія («Тренер» - сервер) відбувається через протоколи HTTP/HTTPS.

### 3.1.5 Обмеження пам’яті

Мобільні застосунки мають бути оптимізовані для роботи з великими обсягами даних в умовах обмежених ресурсів мобільних пристроїв, особливо застосунок «Камера» на стороні якого відбувається обробка даних з відео. Зі сторони сервера система спроектована таким чином, щоб обробляти великі об’єми даних. Зокрема, сервер повинен бути здатен підтримувати не менше 10 000 записів у базі даних без погіршення продуктивності. У разі зростання кількості користувачів можна передбачити масштабування на рівні бази даних та серверної інфраструктури.

### 3.1.6 Операції

У системі для тренера передбачений такий набір операцій як управління гравцями та ролями, керування процесу тренування та надсилання інструкцій, створення та налаштування тренувань, що включає в себе вибір граців та конфігурацію зон. Гравці в свою чергу під час тренувань можуть побачити надіслані тренером інструкції та автоматично згенеровані повідомлення про відхилення від позиції, які були виявлення під час опрацювання відео в реальному часі.

## 3.2 Властивості програмного продукту

Програмний продукт FootballGuru характеризується стабільною та безперебійною роботою в умовах польових тренувань без доступу до Інтернету, використовуючи автономну P2P-мережу на основі Wi-Fi Direct/TCP. Завдяки застосуванню .NET MAUI інтерфейси мобільних клієнтів відрізняються високою швидкістю реакції та ефективним використанням ресурсів пристроїв, а асинхронна обробка відеопотоку зменшує навантаження на CPU та пам’ять. Серверна частина забезпечує надійний захист даних та високу доступність, а поєднання SQL Server і MongoDB забезпечує оптимальне зберігання та швидке виконання запитів будь-якої складності. Така сукупність властивостей дозволяє системі залишатися масштабованою, адаптивною до майбутніх розширень і підтримувати високу якість користувацького досвіду на різних платформах.

## 3.3 Атрибути програмного продукту

### 3.3.1 Надійність

Система має вбудовані механізми для обробки помилок як на клієнтській, так і на серверній стороні. Всі вхідні дані проходять перевірку, щоб уникнути некоректного введення. У серверній частині використано конструкції, які дозволяють обробляти винятки та запобігати відмови застосунку при виникненні помилок.

### 3.3.2 Доступність

Офлайн-режим із локальним збереженням даних і автоматичне перепідключення при втраті Wi-Fi Direct забезпечують безперервність тренування.

### 3.3.3 Безпека

Уся система побудована з урахуванням безпеки. Для захисту особистих даних користувачів реалізовано автентифікацію та авторизацію, права доступу чітко розмежовані відповідно до ролі. Паролі зберігаються в хешованому вигляді. Також передбачено захист від SQL-ін’єкцій, весь трафік зашифровано через HTTPS.

### 3.3.4 Супроводжуваність

Код програми організований у вигляді модулів і компонентів, що значно спрощує його підтримку та розширення. Дотримання принципів чистого коду та наявність внутрішньої документації дозволяє іншим розробникам швидко орієнтуватися у структурі проекту. За потреби можна легко додати нові функції або змінити існуючі, не порушуючи роботу всієї системи.

### 3.3.5 Переносимість

Єдина кодова база на C# (.NET MAUI для клієнта, ASP.NET Core для сервера) дозволяє розгортати систему на різних платформах (Android, iOS, Windows, Linux) без значних змін. Серверну частину можна розгорнути на будь-якому сервері, який підтримує .NET та Microsoft SQL Server. Крім того, можлива контейнеризація системи з використанням Docker, що дозволяє запускати її на різних платформах, у тому числі у хмарних середовищах таких як Azure або AWS.

### 3.3.6 Продуктивність

Оптимізація потоків обробки відео, фільтрація кадрів і використання черг зменшують навантаження на CPU/пам’ять, забезпечуючи стабільний FPS та швидкий час відповіді користувача.

## 3.4 Вимоги до бази даних

Система використовує дві доповнюючі одна одну СУБД: Microsoft SQL Server для зберігання довідкових сутностей (тренери, гравці, ролі, інструкції, кольори пов’язок) та MongoDB для документного зберігання конфігурацій тренувань і тактичних схем. Реляційна модель спроектована з дотриманням принципів третьої нормальної форми, що виключає надлишкове дублювання даних і гарантує логічну цілісність через налаштовані зовнішні ключі та каскадні дії при оновленні або видаленні записів. Для підвищення швидкодії операцій на полях-відношеннях передбачені кластерні та некластерні індекси.

Документна база MongoDB організована у вигляді вкладених документів. Завдяки denormalized-підходу додавання та читання тренувальних даних відбувається за одного запиту, що мінімізує кількість звернень і прискорює повернення повного об’єкта конфігурації тренування.

Такий підхід поєднує строгі властивості цілісності реляційних даних із високою продуктивністю та гнучкістю документної моделі для складної ієрархії тренувальних конфігурацій.

## 3.5 Інші вимоги

Система повинна підтримувати локалізацію інтерфейсу українською та англійською мовами з можливістю динамічного перемикання під час роботи, забезпечуючи зручність використання в різних регіонах. Для гарантування безперервної роботи потрібно реалізувати резервне копіювання локальних даних тренера (конфігурацій тренувань) і бази даних за розкладом із можливістю відновлення до будь-якої попередньої версії. Для оцінки та оптимізації продуктивності необхідно вбудувати механізми моніторингу та метрик (CPU, пам’ять, час обробки кадру, мережеві затримки) із можливістю вивантаження логів і графіків у зовнішні системи аналітики. Цей комплекс додаткових вимог забезпечує не лише функціональну завершеність, а й надійність, підтримуваність й адаптивність платформи в умовах реального застосування на тренувальному майданчику.