

Міністерство освіти і науки України
Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

Кваліфікаційна робота
за освітнім ступенем «бакалавр»
на тему:
**МОБІЛЬНИЙ ЗАСТОСУНОК ДЛЯ ПЕРСОНАЛІЗОВАНОГО
ТРЕНУВАННЯ ГРАВЦІВ У БІЛЬЯРД**

Виконав:

здобувач вищої освіти IV курсу
групи ПМ-41
спеціальності 113 «Прикладна математика»
Олександр Юрійович Смірнов

Науковий керівник:

старший викладач Тетяна Анатоліївна Кирик

Рівне – 2026

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА АКТУАЛЬНОСТІ РОЗРОБКИ	9
1.1. Особливості тренувального процесу у більярді	9
1.2. Роль персоналізації у спортивному тренуванні	11
1.3. Аналіз існуючих програмних рішень для підтримки тренувань у більярді	12
1.4. Обґрунтування доцільності розробки мобільного застосунку	14
РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ ТА МЕТОДІВ РЕАЛІЗАЦІЇ МОБІЛЬНОГО ЗАСТОСУНКУ	17
2.1. Загальна архітектура програмного застосунку	17
2.2. Вибір мобільної платформи та середовища розробки	18
2.3. Проєктування моделей даних	20
2.4. Застосування методів прикладної математики для аналізу результатів	22
2.5. Алгоритм формування персоналізованих рекомендацій	24
2.6. Візуалізація статистичних даних	26
2.7. Організація користувацького інтерфейсу	27
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ	30
3.1. Загальна структура мобільного застосунку	30
3.2. Реалізація підменю «Бібліотека»	32
3.3. Реалізація підменю «Створити»	34
3.4. Реалізація підменю «Рекомендації»	37
3.5. Реалізація підменю «Цілі»	38
3.6. Реалізація підменю «Статистика»	40
3.7. Взаємодія між модулями застосунку	42
РОЗДІЛ 4. ТЕСТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ	44
4.1. Мета та підхід до тестування	44
4.2. Функціональне тестування основних модулів	44
ВИСНОВКИ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій мобільні застосунки стали невід'ємною частиною повсякденного життя людини. Вони активно використовуються у різних сферах діяльності, зокрема в освіті, бізнесі, медицині, спорті та індивідуальному плануванні. Особливо актуальним є застосування мобільних технологій у спортивній підготовці, де важливу роль відіграють систематичне накопичення даних, аналіз результатів, відстеження прогресу та персоналізація тренувального процесу.

Більярд є складним техніко-тактичним видом спорту, який поєднує точність рухів, просторове мислення, контроль сили удару, здатність прогнозувати траєкторії руху куль, концентрацію уваги та психологічну стійкість. Досягнення стабільних результатів у більярді потребує регулярного виконання спеціалізованих вправ, аналізу помилок, контролю динаміки прогресу та поступового ускладнення тренувальних завдань. При цьому важливо не лише виконувати вправи, а й мати можливість об'єктивно оцінювати їх результати у довготривалій перспективі.

На практиці тренувальний процес у більярді часто ґрунтується на суб'єктивних відчуттях гравця або тренера. Результати можуть записуватися нерегулярно, зберігатися у вигляді окремих нотаток або взагалі не фіксуватися. Такий підхід ускладнює аналіз прогресу, не дозволяє швидко визначати слабкі сторони гравця та знижує ефективність планування подальших тренувань. Крім того, без системного обліку результатів складно зрозуміти, чи справді певна вправа покращується, залишається на стабільному рівні або демонструє негативну динаміку.

Особливої актуальності набуває задача створення такого програмного засобу, який дозволяє не лише фіксувати результати тренувань, а й аналізувати їх за допомогою математичних методів. Побудова часових рядів, обчислення середніх значень, використання ковзного середнього та оцінювання трендових

кривих дають змогу зменшити вплив випадкових коливань і сформувати більш об'єктивне уявлення про зміну рівня підготовки гравця. На основі таких даних можна автоматично формувати персоналізовані рекомендації щодо вправ, які потребують додаткової уваги.

На сьогодні існує обмежена кількість програмних рішень, орієнтованих саме на підтримку тренувального процесу у більярді. Більшість із них або зосереджені на демонстрації готових вправ, або не надають достатньо гнучких інструментів для створення власних тренувальних завдань, аналізу статистики, постановки індивідуальних цілей та формування рекомендацій на основі динаміки результатів. Також не всі рішення враховують специфіку різних більярдних дисциплін, зокрема дисципліни «Піраміда», яка має власні технічні особливості та вимоги до виконання ударів.

У зв'язку з цим актуальною є розробка мобільного застосунку для персоналізованого тренування гравців у більярд, який поєднує каталог вправ, можливість створення власних тренувальних схем, фіксацію результатів, статистичний аналіз, систему рекомендацій та відстеження індивідуальних цілей. Такий застосунок може бути корисним як для гравців-початківців, так і для більш досвідчених спортсменів, які прагнуть систематизувати власні тренування та приймати рішення на основі об'єктивних даних.

Розроблений мобільний застосунок передбачає використання декількох основних функціональних розділів. Розділ «Бібліотека» містить перелік тренувальних вправ із візуальними схемами та описами. Розділ «Створити» дає користувачеві можливість самостійно формувати нові вправи за допомогою розміщення куль на більярдному столі, побудови траєкторій руху та додавання назви й опису вправи. Розділ «Рекомендації» аналізує статистику виконання вправ і формує список пріоритетних напрямів для покращення. Розділ «Цілі» дозволяє користувачеві задавати конкретні цільові показники для вправ і автоматично

відстежувати їх виконання. Розділ «Статистика» забезпечує візуальне представлення результатів у вигляді графіків та трендових кривих.

Метою кваліфікаційної роботи є розробка мобільного застосунку для персоналізованого тренування гравців у більярд із використанням методів прикладної математики для аналізу результатів, формування рекомендацій, візуалізації прогресу та підтримки індивідуального планування тренувального процесу.

Для досягнення поставленої мети у роботі необхідно вирішити такі завдання:

- проаналізувати особливості тренувального процесу у більярді та визначити вимоги до програмного засобу його підтримки
- дослідити існуючі мобільні застосунки та підходи до фіксації, аналізу й візуалізації спортивних результатів
- обґрунтувати вибір мобільної платформи, середовища розробки та засобів реалізації користувацького інтерфейсу
- спроектувати структуру мобільного застосунку та моделі даних для збереження вправ, результатів тренувань, користувацьких цілей і рекомендацій
- реалізувати модуль бібліотеки вправ із можливістю перегляду схем, описів та запису результатів тренування
- реалізувати модуль створення користувацьких вправ із використанням інтерактивного розміщення куль, побудови суцільних і пунктирних ліній та збереження опису вправи
- розробити алгоритм формування персоналізованих рекомендацій на основі аналізу трендових кривих і динаміки результатів користувача
- реалізувати систему постановки та автоматичного відстеження виконання індивідуальних цілей
- забезпечити візуалізацію статистичних даних у вигляді графіків, часових рядів і згладжених трендових ліній

- провести тестування основних функціональних модулів мобільного застосунку та перевірити коректність роботи реалізованих алгоритмів.

Об'єктом дослідження є процес тренування гравців у більярд.

Предметом дослідження є методи та програмні засоби персоналізації тренувального процесу з використанням мобільних технологій, статистичного аналізу результатів, побудови трендових кривих та формування рекомендацій.

У роботі використано методи аналізу предметної області, проектування програмного забезпечення, об'єктно-орієнтованого моделювання, статистичної обробки даних, побудови часових рядів, згладжування результатів за допомогою ковзного середнього, а також методи функціонального тестування мобільних застосунків.

Методи дослідження. У кваліфікаційній роботі застосовано комплекс методів дослідження для досягнення поставленої мети та виконання визначених завдань.

Аналіз і порівняння: для дослідження особливостей тренувального процесу у більярді, визначення потреб користувачів та порівняння функціональних можливостей існуючих програмних рішень для підтримки спортивних тренувань.

Узагальнення та систематизація: для формування вимог до мобільного застосунку, визначення основних функціональних модулів, структури підменю та ключових сценаріїв використання.

Моделювання: для проектування архітектури застосунку, визначення моделей даних, опису взаємодії між функціональними модулями та побудови схем роботи системи.

Методи прикладної математики: для обробки тренувальних результатів, обчислення середніх значень, побудови часових рядів, використання ковзного середнього та оцінювання трендових кривих.

Проектування користувацького інтерфейсу: для розробки структури екранів мобільного застосунку, організації навігації між підменю, створення візуальних схем вправ та забезпечення зручної взаємодії користувача із застосунком.

Експериментальне тестування: для перевірки працездатності основних функцій застосунку, зокрема створення вправ, збереження результатів, побудови статистики, формування рекомендацій та автоматичного контролю виконання цілей.

Апробація та впровадження результатів дослідження. Основні теоретичні та практичні результати дослідження були представлені на XIX Всеукраїнській науково-практичній конференції здобувачів вищої освіти і молодих учених «Наука, освіта, суспільство очима молодих» (м. Рівне, 14 травня 2026 р.), а також на звітній науковій конференції викладачів, співробітників і здобувачів вищої освіти Рівненського державного гуманітарного університету (м. Рівне, 14 травня 2026 р.).

Практичне значення роботи полягає у створенні мобільного застосунку, який може використовуватися для систематизації тренувального процесу гравців у більярд. Застосунок дозволяє зберігати вправи та результати їх виконання, створювати власні тренувальні завдання, аналізувати прогрес, отримувати рекомендації щодо покращення слабких сторін і контролювати досягнення поставлених цілей. Це сприяє підвищенню ефективності тренувань та забезпечує більш усвідомлений підхід до спортивної підготовки.

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. У першому розділі розглядаються особливості предметної області, специфіка тренувального процесу у більярді та аналізуються існуючі програмні рішення. У другому розділі обґрунтовується вибір технологій, описуються моделі даних і математичні методи, використані для аналізу результатів. Третій розділ присвячений проектуванню та реалізації

основних функціональних модулів мобільного застосунку. У четвертому розділі наведено тестування застосунку, перевірку роботи основних сценаріїв і оцінку коректності реалізованого функціоналу. У висновках узагальнено результати виконаної роботи та визначено можливі напрями подальшого розвитку застосунку.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА АКТУАЛЬНОСТІ РОЗРОБКИ

1.1. Особливості тренувального процесу у більярді

Більярд належить до технічно складних видів спорту, у яких результат значною мірою залежить від точності рухів, просторового мислення, здатності прогнозувати траєкторії руху куль, контролю сили удару та стабільності психологічного стану гравця. На відміну від багатьох динамічних спортивних дисциплін, у більярді вирішальне значення мають не лише фізичні якості спортсмена, а й уміння багаторазово відтворювати однакові технічні дії з високим рівнем точності.

Тренувальний процес у більярді зазвичай складається з виконання окремих вправ, спрямованих на розвиток конкретних навичок. До таких навичок належать точність удару, контроль виходу битка, серійність, робота з різними кутами, виконання ударів від борту, контроль сили, а також здатність приймати правильні тактичні рішення залежно від розташування куль на столі. Кожна вправа має власну мету, умови виконання та спосіб оцінювання результату.

Для частини вправ основним показником є відсоток успішних ударів. Наприклад, користувач може виконати певну кількість спроб і зафіксувати кількість результативних ударів. У такому випадку результат тренування доцільно подавати у вигляді відсоткового значення. Для інших вправ більш важливим є рекордний показник, наприклад максимальна кількість успішних дій поспіль або найкращий досягнутий числовий результат. Такий поділ вправ потребує гнучкої моделі збереження даних, оскільки різні типи тренувань не завжди можуть оцінюватися за однаковою шкалою.

Однією з ключових проблем тренувального процесу у більярді є складність об'єктивної оцінки прогресу. Результат окремого тренування може залежати від багатьох випадкових факторів: фізичної втоми, концентрації, психологічного

стану, якості обладнання, складності конкретної позиції або навіть зовнішніх умов. Через це один високий або низький результат не завжди є достатньою підставою для висновку про реальний прогрес або спад. Більш обґрунтовані висновки можна робити лише на основі накопиченої історії результатів та аналізу їх динаміки в часі.

На практиці багато гравців не ведуть системного обліку тренувань. Результати можуть запам'ятовуватися приблизно, записуватися у нотатках або не фіксуватися взагалі. Такий підхід ускладнює оцінку слабких і сильних сторін гравця. Крім того, без збереження історії тренувань складно визначити, які вправи дійсно покращуються, які залишаються на стабільному рівні, а які потребують додаткової уваги.

Для вирішення цієї проблеми доцільним є використання цифрових засобів, які дозволяють фіксувати результати, накопичувати історію тренувань, будувати графіки та аналізувати зміну показників. У контексті програмної інженерії подібні системи мають не лише зберігати дані, а й забезпечувати зручну взаємодію користувача з інформацією, оскільки практична цінність програмного продукту значною мірою залежить від його зрозумілості, надійності та відповідності потребам користувача [28, 29].

Особливо важливим є те, що більярдні тренування часто потребують візуального представлення вправи. Гравцеві недостатньо лише текстового опису, оскільки розташування куль, напрям удару та передбачувана траєкторія руху мають суттєве значення для правильного виконання завдання. Тому ефективний мобільний застосунок для більярду має поєднувати текстові описи, візуальні схеми, можливість запису результатів та інструменти подальшого аналізу.

1.2. Роль персоналізації у спортивному тренуванні

Персоналізація тренувального процесу передбачає адаптацію вправ, навантаження, цілей та рекомендацій до індивідуальних особливостей конкретного

користувача. У більярді це особливо важливо, оскільки різні гравці можуть мати різний рівень технічної підготовки, різні слабкі сторони та різні пріоритети у розвитку. Для одного гравця проблемною може бути точність ударів у певних позиціях, для іншого — контроль битка після удару, для третього — стабільність серійного виконання вправ.

Традиційний підхід, при якому всі користувачі отримують однаковий набір вправ, має обмежену ефективність. Він може бути корисним на початковому етапі, але не завжди дозволяє врахувати реальну динаміку прогресу конкретного гравця. Якщо застосунок лише надає список вправ, але не аналізує результати їх виконання, він виконує роль електронного довідника, а не повноцінного інструменту підтримки тренувального процесу.

Більш ефективним є підхід, за якого застосунок накопичує індивідуальні результати користувача і на їх основі формує висновки щодо подальших тренувань. Для цього можуть використовуватися статистичні методи: обчислення середніх значень, побудова часових рядів, згладжування даних та оцінювання тренду. Такі методи дозволяють перейти від суб'єктивної оцінки до аналізу, який ґрунтується на фактичних даних [16, 17, 24].

У межах розроблюваного застосунку персоналізація реалізується через декілька функціональних механізмів. По-перше, користувач може переглядати бібліотеку вправ і фіксувати результати їх виконання. По-друге, він може створювати власні вправи з урахуванням конкретних тренувальних потреб. По-третє, застосунок аналізує статистику та формує рекомендації щодо вправ, які потребують додаткової уваги. По-четверте, користувач може встановлювати цілі для конкретних вправ і відстежувати їх виконання.

Важливим елементом персоналізації є система рекомендацій. Її завдання полягає не в тому, щоб випадково пропонувати вправи зі списку, а в тому, щоб визначати пріоритети на основі динаміки результатів. Якщо тренд певної вправи

демонструє погіршення, така вправа повинна отримати вищий пріоритет у тренувальному плані. Якщо результати залишаються стабільними, вправу доцільно залишити у списку рекомендованих для підтримання рівня. Якщо ж спостерігається позитивна динаміка, вправа може бути віднесена до категорії покращення.

Такий підхід дозволяє зробити тренування більш усвідомленим. Користувач бачить не лише окремі результати, а й загальну тенденцію. Це допомагає уникнути ситуації, коли гравець тренує лише ті вправи, які йому подобаються або добре вдаються, і водночас ігнорує вправи, що потребують найбільшої уваги.

Окрему роль відіграє модуль цілей. Постановка конкретної цілі дозволяє користувачеві визначити бажаний результат для певної вправи. Наприклад, це може бути досягнення визначеного відсотка успішних ударів або певного рекордного значення. Після введення нового результату застосунок автоматично порівнює його із заданою ціллю і визначає, чи було її виконано. Такий механізм підвищує мотиваційну цінність застосунку та робить тренувальний процес більш структурованим.

1.3. Аналіз існуючих програмних рішень для підтримки тренувань у більярді

На сучасному етапі розвитку мобільних технологій існують окремі програмні продукти, спрямовані на підтримку тренувального процесу у більярді. Вони можуть містити бібліотеки вправ, схеми розташування куль, текстові описи завдань або інструменти для ведення тренувальних записів. Проте більшість таких рішень мають обмежений функціонал і не завжди забезпечують повний цикл роботи з тренувальними даними: від створення вправи до аналізу результатів і формування рекомендацій.

Одним із відомих мобільних застосунків для тренувань у більярді є Drill Room [19]. Даний застосунок орієнтований переважно на гравців в американський пул і містить набір вправ для відпрацювання технічних елементів гри. Його перевагою є наявність структурованого каталогу тренувальних завдань і візуальних схем, які допомагають користувачеві зрозуміти умови виконання вправи.

Разом з тим, функціональні можливості подібних застосунків мають низку обмежень. По-перше, вони часто орієнтовані на конкретну дисципліну більярду, наприклад американський пул. Водночас дисципліна «Піраміда» має власні особливості, зокрема інші розміри куль, специфіку ударів, більші вимоги до точності та відмінну логіку позиційної гри. Через це вправи, створені для пулу, рідко можуть бути безпосередньо використані для тренувань у піраміду.

По-друге, існуючі рішення не завжди надають достатньо гнучкі інструменти для створення власних вправ. Для більярду це є суттєвим обмеженням, оскільки гравець або тренер часто потребує можливості змодельовати конкретну ігрову ситуацію. Така ситуація може включати індивідуальне розташування куль, декілька можливих траєкторій, різні типи ударів і пояснення умов виконання вправи. Без можливості створення власних тренувальних схем застосунок залишається обмеженим лише попередньо підготовленим набором завдань.

По-третє, значна частина застосунків недостатньо розвиває аналітичний компонент. Наявність списку вправ сама по собі не забезпечує персоналізації. Для цього потрібно аналізувати історію результатів, будувати графіки, оцінювати зміну показників і формувати висновки щодо подальшого тренування. Візуалізація даних має бути не декоративною, а інформативною, тобто допомагати користувачеві швидко зрозуміти стан прогресу та проблемні ділянки [20].

По-четверте, у багатьох рішеннях відсутній механізм постановки й автоматичного контролю індивідуальних цілей. Без такого механізму користувач

може бачити статистику, але не завжди має чітку систему орієнтирів. Наявність цілей дозволяє перетворити тренування з набору окремих вправ на послідовний процес досягнення конкретних результатів.

З погляду користувацького досвіду важливо, щоб застосунок був не лише функціональним, а й зручним. Інтерфейс має бути зрозумілим, не перевантаженим зайвими елементами і придатним для швидкого використання під час тренування. Загальні принципи зручності програмних інтерфейсів передбачають простоту навігації, передбачувану поведінку елементів, зрозумілу структуру екранів та мінімізацію зайвих дій з боку користувача [25, 26].

Таким чином, аналіз існуючих рішень показує, що на ринку є застосунки, які частково вирішують задачу підтримки тренувань у більярді, однак вони не завжди поєднують усі необхідні функції в єдиній системі. Зокрема, недостатньо представлені рішення, які одночасно підтримують створення власних вправ, фіксацію результатів, статистичний аналіз, побудову рекомендацій, постановку цілей і візуалізацію прогресу

1.4. Обґрунтування доцільності розробки мобільного застосунку

Розробка мобільного застосунку для персоналізованого тренування гравців у більярд є доцільною з огляду на потребу в системному підході до фіксації та аналізу тренувальних результатів. Мобільний пристрій є зручним інструментом для такого завдання, оскільки він завжди доступний користувачеві та може використовуватися безпосередньо під час тренування. Це дозволяє швидко записувати результати, переглядати схеми вправ, аналізувати статистику та контролювати досягнення цілей.

Розроблюваний застосунок має на меті поєднати декілька функціональних напрямів. Першим напрямом є бібліотека вправ, яка забезпечує користувачеві

доступ до підготовлених тренувальних завдань. Другим напрямом є створення власних вправ, що дозволяє адаптувати застосунок до конкретних потреб гравця або тренера. Третім напрямом є фіксація та збереження результатів. Четвертим напрямом є статистичний аналіз, який включає побудову графіків, обчислення середніх значень і трендових кривих. П'ятим напрямом є формування рекомендацій та контроль індивідуальних цілей.

На відміну від простого електронного списку вправ, такий застосунок виконує роль інструменту підтримки тренувального процесу. Він не лише зберігає інформацію, а й допомагає користувачеві інтерпретувати її. Завдяки аналізу трендів можна виявляти вправи, у яких результати погіршуються, залишаються стабільними або покращуються. Це дозволяє формувати персоналізовані рекомендації та визначати пріоритети для подальших занять.

Практична доцільність розробки також полягає у можливості використання застосунку в індивідуальних тренуваннях. Гравець може самостійно створювати вправи, фіксувати результати, стежити за прогресом і контролювати досягнення поставлених цілей. Для тренера такий застосунок може бути корисним як допоміжний інструмент для структурування завдань і пояснення вправ за допомогою візуальних схем.

Важливою перевагою є локальне збереження даних. Оскільки основний функціонал застосунку орієнтований на індивідуальне використання, відсутність обов'язкової залежності від серверної частини спрощує архітектуру та робить застосунок доступним навіть без підключення до мережі Інтернет. Такий підхід відповідає принципам створення програмних систем, у яких архітектурні рішення мають бути узгоджені з реальними потребами користувача та призначенням продукту [28, 29].

Окремо слід зазначити значення візуального компонента. Для більярду схема вправи має важливе практичне значення, оскільки вона дозволяє користувачеві

швидко зрозуміти розташування куль і логіку виконання удару. Тому реалізація інтерактивного модуля створення вправ із можливістю розміщення куль і побудови траєкторій підвищує практичну цінність застосунку. Такий підхід робить програмний продукт не лише засобом обліку результатів, а й інструментом планування тренувальних завдань.

Таким чином, розробка мобільного застосунку для персоналізованого тренування гравців у більярд є актуальною та практично значущою задачею. Вона поєднує елементи мобільної розробки, прикладної математики, спортивної аналітики, проєктування користувацького інтерфейсу та тестування програмного забезпечення. Реалізація такого застосунку дозволяє підвищити рівень системності тренувального процесу, забезпечити об'єктивніший аналіз результатів і створити основу для подальшого розвитку функціональності.

РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ ТА МЕТОДІВ РЕАЛІЗАЦІЇ МОБІЛЬНОГО ЗАСТОСУНКУ

2.1. Загальна архітектура програмного застосунку

Проектування мобільного застосунку для персоналізованого тренування гравців у більярд потребує вибору такої архітектури, яка забезпечує зручність використання, можливість подальшого розширення функціоналу та коректну обробку тренувальних даних. Оскільки застосунок орієнтований на індивідуальне використання одним користувачем, основний функціонал реалізовано на стороні мобільного пристрою без обов'язкової взаємодії з віддаленим сервером.

У межах роботи застосунок побудовано як локальний мобільний програмний продукт, що поєднує користувацький інтерфейс, модулі прикладної логіки, локальне збереження даних та інструменти статистичного аналізу. Такий підхід дозволяє забезпечити доступність основних функцій навіть без підключення до мережі Інтернет, що є важливим для використання застосунку безпосередньо під час тренування.

Загальна архітектура застосунку складається з таких основних рівнів:

- рівень користувацького інтерфейсу;
- рівень прикладної логіки;
- рівень моделей даних;
- рівень локального збереження інформації;
- рівень статистичного аналізу та формування рекомендацій;
- рівень візуалізації результатів.

Рівень користувацького інтерфейсу відповідає за взаємодію користувача із застосунком. До нього належать екрани бібліотеки вправ, створення нової вправи, рекомендацій, цілей та статистики. Саме цей рівень забезпечує введення даних, вибір вправ, розміщення куль, побудову траєкторій, перегляд графіків і керування цілями.

Рівень прикладної логіки виконує обробку дій користувача. Він відповідає за перевірку введених результатів, створення нових вправ, обчислення статистичних показників, визначення виконання цілей та підготовку даних для відображення у вигляді графіків. Такий розподіл дозволяє відокремити логіку роботи застосунку від елементів інтерфейсу, що спрощує супровід і подальший розвиток програмного продукту [28, 29].

Рівень моделей даних описує основні сутності предметної області: вправи, результати тренувань, траєкторії, цілі, рекомендації та записи виконаних цілей. Чітке проєктування моделей даних є важливою умовою коректної роботи застосунку, оскільки саме на основі цих моделей здійснюються обчислення, побудова графіків і формування персоналізованих рекомендацій.

Рівень статистичного аналізу забезпечує обробку накопичених результатів. У ньому реалізуються обчислення відсотка успішності, середніх значень, ковзного середнього та нахилу трендової лінії. Ці показники використовуються для оцінки прогресу користувача та визначення категорії рекомендацій для кожної вправи.

Загалом обрана архітектура відповідає принципам модульності та розподілу відповідальності між компонентами системи. Це дозволяє незалежно розширювати окремі функціональні частини застосунку, наприклад додавати нові типи вправ, нові способи візуалізації або додаткові алгоритми аналізу результатів.

2.2. Вибір мобільної платформи та середовища розробки

Для реалізації мобільного застосунку було обрано платформу iOS. Такий вибір обґрунтований стабільністю платформи, наявністю розвинених засобів для створення графічного інтерфейсу, підтримкою роботи з жестами користувача, широкими можливостями для побудови кастомних елементів інтерфейсу та зручними інструментами тестування.

Розробка застосунку здійснювалася у середовищі Xcode, яке є офіційним інтегрованим середовищем розробки для програмних продуктів екосистеми Apple [14]. Xcode надає інструменти для написання коду, створення інтерфейсу, налагодження, запуску застосунку на симуляторі та реальному пристрої, а також для виконання тестування.

Як основну мову програмування було використано Objective-C. Ця мова є класичною мовою розробки для iOS та забезпечує повноцінну взаємодію з фреймворками платформи Apple [1]. Використання Objective-C у межах роботи є доцільним, оскільки мова дозволяє реалізовувати як стандартні екрани застосунку, так і кастомні графічні елементи, необхідні для відображення більярдного столу, куль, ліній траєкторій та графіків статистики.

Для створення користувацького інтерфейсу застосовано UIKit — основний фреймворк для побудови інтерфейсів iOS-застосунків [12]. UIKit забезпечує роботу з екранами, контролерами, кнопками, таблицями, колекціями, жестами та іншими елементами інтерфейсу. У межах розроблюваного застосунку UIKit використовується для реалізації вкладкової навігації, списків вправ, форм введення результатів, екранів статистики, цілей і рекомендацій.

Окрему роль відіграють можливості iOS щодо роботи з графікою та жестами. Для модуля створення вправ важливо забезпечити інтерактивну взаємодію користувача з об'єктами на екрані: розміщення куль, вибір типу лінії, побудову траєкторій та редагування схеми вправи. Для цього можуть використовуватися стандартні механізми обробки дотиків, жестів та переміщення елементів інтерфейсу [5,6].

Для побудови графічних ліній та допоміжних елементів доцільним є використання засобів UIKit і Core Graphics. Зокрема, UIBezierPath може застосовуватися для побудови ліній, кривих, контурів та інших графічних об'єктів

[11]. Core Graphics, у свою чергу, надає нижчорівневі можливості для малювання, роботи з координатами, кольорами та графічним контекстом [3].

Таким чином, вибір платформи iOS, середовища Xcode, мови Objective-C та фреймворку UIKit є обґрунтованим з огляду на поставлені завдання. Обрані технології дозволяють реалізувати як стандартну логіку мобільного застосунку, так і специфічні функції, пов'язані з інтерактивним створенням більярдних вправ та візуалізацією статистичних даних.

2.3. Проєктування моделей даних

Одним із ключових етапів розробки застосунку є проєктування моделей даних. Моделі визначають, які об'єкти існують у системі, які властивості вони мають і яким чином взаємодіють між собою. Для застосунку персоналізованого тренування гравців у більярд основними сутностями є вправа, результат тренування, користувацька схема, траєкторія, ціль та рекомендація.

Базовою сутністю є вправа. Вона містить назву, опис, тип оцінювання результату, візуальну схему та додаткові параметри. Вправи можуть бути вбудованими, тобто заздалегідь підготовленими у бібліотеці, або користувацькими, створеними самим користувачем у модулі «Створити». Такий поділ дозволяє поєднати готовий набір тренувальних завдань із можливістю індивідуального налаштування.

Для кожної вправи визначається тип оцінювання. У межах застосунку доцільно розглядати два основні типи:

- вправи на точність;
- вправи на рекордний результат.

Для вправ на точність користувач вводить кількість спроб та кількість успішних виконань. На основі цих даних відсоток результативності обчислюється за формулою (2.1):

$$P = (N / H) \cdot 100\% \quad (2.1)$$

де Р — відсоток результативності, Н — кількість успішних ударів, N — загальна кількість спроб.

Для рекордних вправ користувач вводить числове значення досягнутого результату. Це може бути кількість успішних дій поспіль, максимальне значення серії або інший числовий показник, який характеризує виконання вправи. Такий підхід дозволяє підтримувати різні види тренувальних завдань без необхідності створювати окрему логіку для кожної вправи.

Окремою сутністю є запис тренування. Він містить посилання на вправу, дату виконання, введені користувачем значення та обчислений результат. Збереження дати є важливим, оскільки саме вона дозволяє будувати часові ряди, аналізувати динаміку результатів і формувати трендові криві.

Для модуля створення вправ необхідно передбачити модель користувацької схеми. Вона включає інформацію про розташування куль на більярдному столі та набір побудованих траєкторій. Кожна куля може описуватися координатами на площині столу, типом або кольором. Координати доцільно зберігати у відносному вигляді, наприклад у діапазоні від 0 до 1 по ширині та висоті столу. Це дозволяє коректно масштабувати схему на різних розмірах екрана.

Траєкторія у схемі вправи може містити початкову точку, кінцеву точку та тип лінії. У застосунку передбачено два типи ліній: суцільна та пунктирна. Суцільна лінія може використовуватися для позначення основного напрямку удару або руху кулі, а пунктирна — для допоміжної траєкторії, прогнозованого руху або альтернативного напрямку. Такий поділ робить схему вправи більш інформативною та зрозумілою для користувача.

Ще однією важливою сутністю є ціль. Вона пов'язується з конкретною вправою і містить бажане числове значення, якого користувач прагне досягти. Для

вправ на точність ціль може задаватися у відсотках, а для рекордних вправ — у відповідних числових одиницях. Ціль має статус: активна або виконана. Якщо користувач вводить результат, що дорівнює або перевищує цільове значення, застосунок автоматично фіксує виконання цілі.

Рекомендація є похідною сутністю, яка формується на основі аналізу статистики. Вона містить вправу, значення тренду та категорію: «Критично», «Рекомендовано» або «Покращення». Рекомендації не є статичним списком, оскільки вони залежать від актуальних результатів користувача та можуть змінюватися після додавання нових записів тренування.

Таким чином, проєктування моделей даних забезпечує основу для реалізації всіх основних функцій застосунку: бібліотеки вправ, створення користувацьких схем, запису результатів, побудови статистики, формування рекомендацій і контролю цілей.

2.4. Застосування методів прикладної математики для аналізу результатів

Важливою складовою застосунку є аналіз тренувальних результатів із використанням методів прикладної математики. Оскільки результати тренувань можуть коливатися під впливом випадкових факторів, оцінювання прогресу лише за окремими значеннями є недостатньо надійним. Для більш об'єктивного аналізу необхідно розглядати динаміку результатів у часі.

Основою статистичного аналізу є часовий ряд. У контексті застосунку часовий ряд — це послідовність результатів певної вправи, впорядкованих за датою виконання. Кожна точка такого ряду містить дату та числове значення результату. Для вправ на точність значенням є відсоток успішності, а для рекордних вправ — досягнутий числовий показник.

Для узагальнення результатів може використовуватися середнє арифметичне (2.2):

$$\bar{x} = (x_1 + x_2 + \dots + x_n) / n \quad (2.2)$$

де $\bar{x}$ — середнє значення, x_i — окремий результат, n — кількість результатів.

Середнє значення дозволяє оцінити загальний рівень виконання вправи за певний період. Проте воно не завжди показує напрям зміни результатів. Наприклад, дві вправи можуть мати однакове середнє значення, але в першій результати поступово покращуються, а в другій — погіршуються. Тому для оцінки прогресу важливо аналізувати не лише середній рівень, а й тренд.

Для зменшення впливу випадкових коливань у застосунку використовується метод ковзного середнього. Його суть полягає в тому, що кожне згладжене значення обчислюється як середнє значення декількох сусідніх результатів (2.3):

$$MA_i = (x_i + x_{i-1} + \dots + x_{i-k+1}) / k \quad (2.3)$$

де MA_i — значення ковзного середнього, k — розмір вікна згладжування.

У межах застосунку доцільно використовувати невелике вікно згладжування, наприклад 3 останні результати. Такий підхід дозволяє згладити випадкові стрибки, але не приховує реальні зміни у динаміці тренувань. Якщо даних недостатньо, застосунок може використовувати фактичні значення без згладжування або будувати тренд лише після накопичення мінімальної кількості записів.

Для визначення загального напрямку зміни результатів використовується коефіцієнт нахилу трендової лінії. Його можна обчислити за допомогою лінійної апроксимації результатів. У найпростішому вигляді трендова лінія описується рівнянням (2.4):

$$y = ax + b \quad (2.4)$$

де y — значення результату, x — порядковий номер запису або час, a — коефіцієнт нахилу, b — початкове значення.

Саме коефіцієнт a є основним показником для визначення динаміки. Якщо a має додатне значення, результати мають тенденцію до покращення. Якщо a має від'ємне значення, спостерігається погіршення. Якщо значення a близьке до нуля, результати можна вважати стабільними.

Використання часових рядів, ковзного середнього та трендової лінії є поширеним підходом до аналізу експериментальних даних і дозволяє отримати більш об'єктивне уявлення про зміну показників у часі [16, 17, 24].

2.5. Алгоритм формування персоналізованих рекомендацій

Однією з основних відмінностей розроблюваного застосунку від простого електронного каталогу вправ є наявність модуля персоналізованих рекомендацій. Його завдання полягає у визначенні вправ, які потребують уваги користувача, на основі фактичної статистики виконання.

Алгоритм формування рекомендацій складається з декількох етапів. На першому етапі застосунок отримує всі збережені результати для кожної вправи. Якщо для вправи недостатньо даних, наприклад є лише один запис, така вправа не може бути надійно проаналізована за трендом. У цьому випадку вона може тимчасово не включатися до рекомендацій або позначатися як така, що потребує накопичення додаткової статистики.

На другому етапі результати впорядковуються за датою. Це дозволяє сформувати часовий ряд для кожної вправи. Якщо за один день було введено декілька результатів однієї вправи, застосунок може або враховувати їх як окремі записи, або обчислювати середнє значення за день. У межах даної роботи доцільним є використання окремих записів, оскільки кожен результат відображає конкретне виконання вправи.

На третьому етапі виконується згладжування даних за допомогою ковзного середнього. Це дозволяє зменшити вплив випадкових коливань, які можуть

виникати через втому, неуважність або інші тимчасові фактори. Згладжені дані краще відображають загальну тенденцію зміни результатів.

На четвертому етапі для кожної вправи обчислюється коефіцієнт нахилу трендової лінії. Саме цей коефіцієнт використовується для визначення категорії рекомендації. Для вправ на точність значення може аналізуватися у відсоткових пунктах. Для рекордних вправ доцільно використовувати нормалізований показник, щоб результати з різними шкалами можна було порівнювати між собою.

Для класифікації вправ використовується порогове значення ε , яке визначає допустиму зону стабільності. Якщо нахил тренду знаходиться в межах від $-\varepsilon$ до $+\varepsilon$, зміни вважаються незначними. У межах застосунку для вправ на точність можна використовувати допуск 0,5 процентного пункту на один запис. Це дозволяє не вважати незначні коливання реальним прогресом або спадом.

Категоризація виконується за такими правилами:

$$a < -\varepsilon$$

— вправа належить до категорії «Критично»;

$$-\varepsilon \leq a \leq \varepsilon$$

— вправа належить до категорії «Рекомендовано»;

$$a > \varepsilon$$

— вправа належить до категорії «Покращення».

Категорія «Критично» містить вправи, у яких спостерігається негативна динаміка. Такі вправи потребують першочергової уваги, оскільки результати користувача погіршуються. Категорія «Рекомендовано» містить вправи зі стабільною динамікою. Їх доцільно продовжувати виконувати для підтримання або

поступового покращення рівня. Категорія «Покращення» містить вправи з позитивною динамікою, тобто ті, у яких користувач демонструє прогрес.

Щоб модуль рекомендацій не дублював бібліотеку вправ, у кожній категорії відображається не більше чотирьох вправ. Це дозволяє зберегти компактність інтерфейсу та зосередити увагу користувача на найбільш важливих позиціях. У категорії «Критично» вправи сортуються від найгіршого нахилу до менш негативного, тобто спочатку показуються ті, де спад є найбільшим. У категорії «Покращення» вправи можуть сортуватися від найбільшого позитивного нахилу до меншого.

Такий алгоритм дозволяє формувати рекомендації не вручну, а на основі статистичних даних. Це підвищує практичну цінність застосунку, оскільки користувач отримує не просто список доступних вправ, а персоналізовану підказку щодо того, які елементи тренувального процесу потребують уваги.

2.6. Візуалізація статистичних даних

Візуалізація даних є важливою частиною мобільного застосунку, оскільки саме графічне представлення результатів дозволяє користувачеві швидко оцінити динаміку прогресу. Табличне або текстове подання результатів може бути точним, але воно не завжди зручне для швидкого аналізу. Графік, навпаки, дозволяє наочно побачити підйоми, спади, стабільні ділянки та загальний напрям зміни результатів.

У застосунку статистика подається у вигляді графіка, на якому відображаються фактичні результати та згладжена трендова крива. Фактичні результати показують конкретні значення, введені користувачем після тренувань. Трендова крива дозволяє оцінити загальний напрям зміни показників без надмірного впливу випадкових коливань.

По осі абсцис відкладається час або послідовність тренувальних записів, а по осі ординат — числове значення результату. Для вправ на точність це відсоток

успішності, а для рекордних вправ — відповідне числове значення. Такий підхід дозволяє використовувати єдиний механізм побудови графіків для різних типів вправ.

У застосунку передбачено вибір періоду аналізу: 7 днів, 30 днів або 90 днів. Короткий період дозволяє оцінити найближчі зміни, тоді як довший період дає змогу побачити загальну тенденцію. Це важливо, оскільки прогрес у більярді не завжди є рівномірним. У короткостроковій перспективі можливі коливання, тоді як довгостроковий графік краще демонструє реальний напрям розвитку.

Для збереження читабельності графіків необхідно враховувати кількість підписів, масштабування та візуальне розділення ліній. Надмірна кількість елементів може ускладнити сприйняття, тому графік має бути достатньо простим і водночас інформативним. Принципи ефективної візуалізації даних передбачають, що графічне представлення має допомагати користувачеві зрозуміти зміст даних, а не лише декоративно оформлювати екран [20].

Візуалізація також використовується у модулі рекомендацій. Хоча рекомендації подаються у вигляді списку, їх основою є аналіз трендових кривих. Тому графік статистики та список рекомендацій логічно пов'язані між собою: користувач може побачити загальний висновок у розділі «Рекомендації», а потім перейти до статистики конкретної вправи для детальнішого аналізу.

2.7. Організація користувацького інтерфейсу

Користувацький інтерфейс застосунку розроблено з урахуванням принципів зручності, послідовності та швидкого доступу до основних функцій. Оскільки застосунок використовується під час тренувального процесу, інтерфейс має бути зрозумілим і не вимагати від користувача зайвих дій.

Основна навігація реалізована за допомогою вкладок. Такий підхід є типовим для iOS-застосунків і дозволяє швидко переходити між основними функціональними розділами. У застосунку передбачено п'ять основних підменю:

- «Бібліотека»;
- «Створити»;
- «Рекомендації»;
- «Цілі»;
- «Статистика».

Розділ «Бібліотека» містить список вправ. Його основне завдання — надати користувачеві швидкий доступ до готових тренувальних завдань. Візуальне оформлення вправ повинно бути достатньо інформативним, щоб користувач міг швидко знайти потрібну вправу та перейти до її виконання.

Розділ «Створити» має більш складну структуру, оскільки він передбачає інтерактивну роботу з графічними елементами. У цьому розділі користувач бачить більярдний стіл, набір куль, перемикач типу лінії, поле для введення назви та опису вправи. Основною вимогою до цього екрана є зрозумілість: користувач повинен швидко зрозуміти, що кулі можна розміщувати на столі, а лінії — використовувати для позначення траєкторій.

Розділ «Рекомендації» має показувати користувачеві не всі вправи, а лише найважливіші з погляду поточної статистики. Тому в ньому використовується поділ на категорії: «Критично», «Рекомендовано» та «Покращення». Така структура дозволяє користувачеві швидко зрозуміти пріоритети тренувального процесу.

Розділ «Цілі» призначений для керування індивідуальними цілями. Він поділяється на активні цілі та історію виконаних цілей. Такий поділ дозволяє користувачеві окремо бачити поточні завдання і вже досягнуті результати.

Виконані цілі можуть зберігатися як історія досягнень, що має мотиваційне значення.

Розділ «Статистика» забезпечує перегляд графіків і трендових кривих. Важливо, щоб користувач міг швидко обрати потрібну вправу та період аналізу. Тому доцільною є структура, у якій у верхній частині екрана відображається графік, а нижче розміщується список вправ для вибору.

Для оформлення інтерфейсу використовується єдина стилістика: темне тло, зелена кольорова палітра, візуальні акценти для активних елементів та контрастний текст. Такий підхід забезпечує цілісність зовнішнього вигляду застосунку. При цьому інтерфейс має відповідати загальним рекомендаціям щодо зрозумілості, послідовності та зручності мобільних застосунків [4, 7, 25, 26].

Таким чином, організація користувацького інтерфейсу спрямована на те, щоб поєднати функціональність і простоту використання. Застосунок має не лише виконувати аналітичні обчислення, а й подавати їх у формі, зрозумілій для користувача без додаткових пояснень.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ

3.1. Загальна структура мобільного застосунку

Розроблений мобільний застосунок для персоналізованого тренування гравців у більярд має модульну структуру, що відповідає основним сценаріям використання програми. На відміну від простого електронного списку вправ, застосунок поєднує декілька взаємопов'язаних функціональних частин: перегляд бібліотеки вправ, створення власних тренувальних схем, фіксацію результатів, аналіз статистики, формування рекомендацій та контроль індивідуальних цілей.

Основна навігація в застосунку реалізована за допомогою вкладкового інтерфейсу. Такий підхід є зручним для мобільного застосунку, оскільки користувач може швидко переходити між основними розділами без необхідності повертатися до проміжних екранів. Вкладкова структура також відповідає типовим принципам побудови iOS-застосунків, де основні функціональні області мають бути доступні з нижньої панелі навігації [7, 12]. Основна структура застосунку реалізована з використанням *UITabBarController*, який забезпечує перемикання між головними підменю програми. Кожне підменю представлено окремим класом-нащадком *UIViewController*, що дозволяє ізолювати логіку окремих екранів і спростити подальшу підтримку коду. Для екранів, які потребують переходу до детальної інформації, додатково використовується *UINavigationController*. Такий підхід дозволяє поєднати швидку навігацію між основними вкладками з можливістю відкривати детальні екрани вправ, статистики або цілей.

На рисунку 3.1 зображено загальну архітектуру мобільного застосунку та взаємодію його основних функціональних модулів.

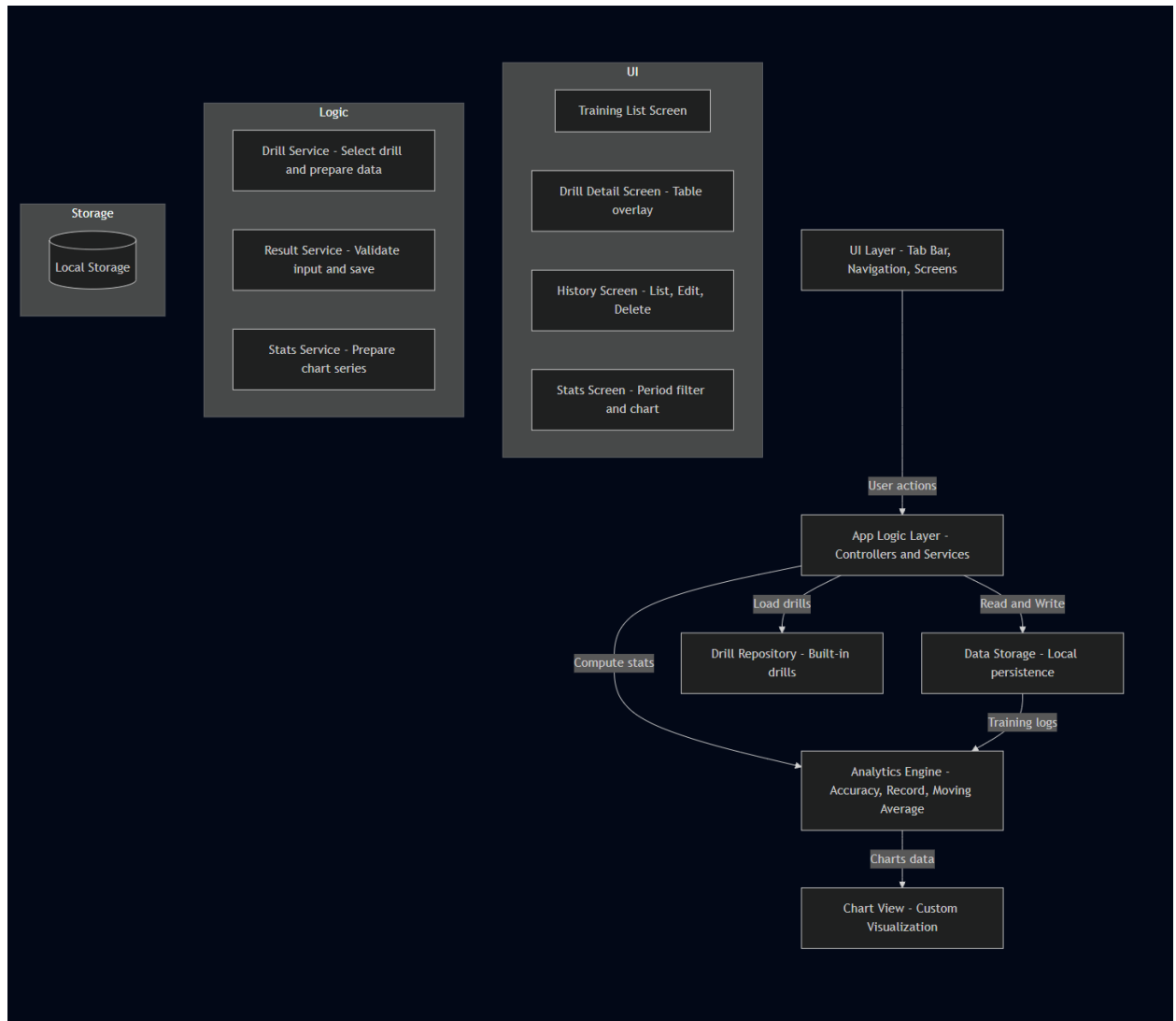


Рисунок 3.1. Схема архітектури мобільного застосунку

Архітектура застосунку передбачає поділ на рівень користувацького інтерфейсу, рівень прикладної логіки, рівень моделей даних, рівень локального збереження інформації та рівень аналітики. Такий поділ дозволяє спростити супровід програмного коду, оскільки кожен модуль виконує власну задачу. Наприклад, модуль статистики не відповідає за введення результатів, а модуль створення вправ не виконує самостійно розрахунок рекомендацій. Водночас усі

модулі працюють із єдиними моделями даних, що забезпечує узгодженість інформації в межах застосунку.

Основними моделями даних у застосунку є модель вправи, модель результату тренування, модель користувацької схеми, модель траєкторії, модель цілі та модель рекомендації. Саме ці сутності забезпечують зв'язок між інтерфейсними екранами та логікою обробки даних. Наприклад, створена користувачем вправа зберігається як окремий об'єкт і надалі може відображатися у бібліотеці, використовуватися для запису результатів, потрапляти до статистики та враховуватися під час формування рекомендацій.

Для реалізації застосунку використано платформу iOS, мову програмування Objective-C та фреймворк UIKit. Такий вибір дозволяє створювати екрани з нативними елементами інтерфейсу, реалізовувати навігацію між контролерами, працювати з дотиками користувача та будувати кастомні графічні компоненти [1, 12, 14].

3.2. Реалізація підменю «Бібліотека»

Підменю «Бібліотека» є базовою частиною застосунку, оскільки саме з нього користувач найчастіше починає роботу з вправами. Основне призначення цього розділу полягає у наданні доступу до переліку тренувальних завдань, які можуть бути використані під час занять більярдом.

Для реалізації списку вправ використано UIScrollView у поєднанні з вертикальним UIStackView. Така структура дозволяє динамічно додавати елементи списку та забезпечує автоматичне прокручування вмісту, якщо кількість вправ перевищує висоту екрана.

У бібліотеці вправи подаються у вигляді структурованого списку. Кожен елемент списку містить назву вправи та коротку візуальну або текстову інформацію, яка допомагає користувачеві швидко зорієнтуватися у змісті вправи.

Такий підхід дозволяє уникнути перевантаження інтерфейсу та водночас забезпечує швидкий доступ до потрібного тренувального завдання.

На рисунку 3.2 зображено підменю «Бібліотека», у якому користувач може переглядати список доступних вправ.

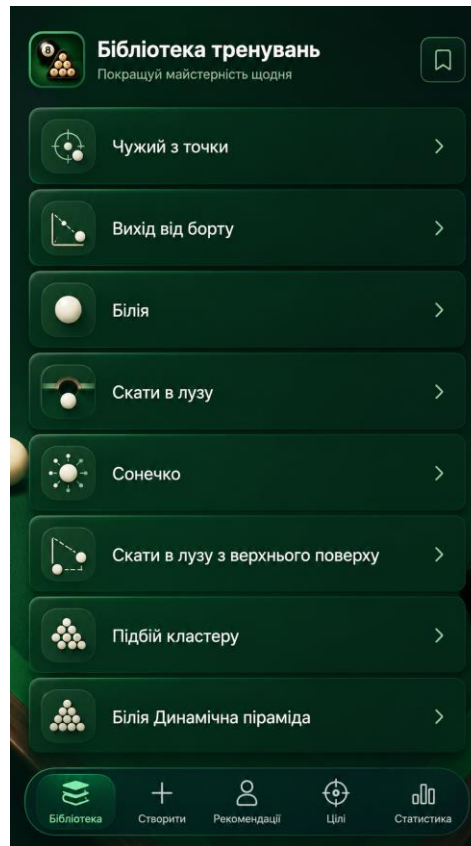


Рисунок 3.2. Підменю «Бібліотека» зі списком вправ

При виборі вправи відкривається детальний екран, на якому відображається її схема, опис та елементи для переходу до запису результату. Схема вправи є важливою частиною інтерфейсу, оскільки у більярді розташування куль і напрямки ударів мають практичне значення. Текстовий опис доповнює візуальну інформацію та пояснює умови виконання вправи.

Для побудови візуальної схеми використовується зображення більярдного столу, поверх якого розміщуються графічні елементи. До таких елементів можуть належати кулі, лінії напрямку удару та допоміжні траєкторії. Такий підхід дозволяє

не створювати окреме зображення для кожної вправи, а формувати схему з окремих компонентів.

Вправи у бібліотеці можуть бути двох типів: вбудовані та створені користувачем. Вбудовані вправи є частиною початкового набору застосунку, тоді як користувацькі вправи додаються через підменю «Створити». Це робить бібліотеку не статичним списком, а розширюваним набором тренувальних завдань.

Для кожної вправи зберігається тип оцінювання результату. Якщо вправа орієнтована на точність, користувач вводить кількість спроб та кількість успішних ударів. Якщо вправа має рекордний характер, користувач вводить числове значення досягнутого результату. Такий поділ дозволяє застосунку коректно обробляти різні види тренувальних даних.

3.3. Реалізація підменю «Створити»

Одним із ключових модулів застосунку є підменю «Створити». Його призначення полягає у наданні користувачеві можливості самостійно формувати власні більярдні вправи. Такий функціонал суттєво розширює можливості застосунку, оскільки користувач не обмежується лише заздалегідь підготовленою бібліотекою вправ.

Робоча область для створення вправи реалізована на основі окремого *UIView*, який виконує роль інтерактивного більярдного столу. Кулі представлені як окремі графічні елементи, розміщені поверх цієї області. Для їх переміщення використовується *UIPanGestureRecognizer*, який дозволяє відстежувати рух пальця користувача та змінювати координати відповідної кулі на екрані. Для побудови траєкторій використано *UIBezierPath* та шар *CAShapeLayer*.

На рисунку 3.3 зображено підменю «Створити», у якому користувач може розміщувати кулі на більярдному столі, будувати траєкторії та додавати опис нової вправи.



Рисунок 3.3. Підменю «Створити»

Інтерфейс підменю «Створити» складається з декількох основних частин. Центральною частиною екрана є більярдний стіл, який використовується як робоча область. На ньому користувач може розміщувати кулі та будувати лінії, що позначають напрямки руху. Нижче або поруч із робочою областю розташовані елементи керування: вибір типу лінії, поля для введення назви та опису вправи, а також кнопка збереження.

Розміщення куль реалізовано за принципом drag-and-drop. Користувач може переміщувати кулю по поверхні столу та встановлювати її у потрібній позиції. Такий спосіб взаємодії є природним для мобільного пристрою, оскільки користувач безпосередньо керує об'єктами на екрані за допомогою дотику. Подібний підхід відповідає загальним рекомендаціям щодо інтерактивної взаємодії з елементами інтерфейсу [5, 6].

Для збереження позицій куль використовуються координати відносно області більярдного столу. Це важливо для коректного відображення вправи на різних розмірах екрана. Наприклад, якщо куля розташована на 30% ширини та 60% висоти столу, ці відносні координати можна використати для повторного відтворення схеми незалежно від фактичного розміру зображення на екрані.

Окрім куль, користувач може додавати траєкторії. У застосунку передбачено два типи ліній: суцільна та пунктирна. Суцільна лінія використовується для позначення основного напрямку удару або руху кулі. Пунктирна лінія може використовуватися для позначення допоміжної траєкторії, прогнозованого руху після зіткнення або альтернативного напрямку. Такий поділ дозволяє зробити схему вправи більш змістовною та зрозумілою.

Побудова ліній реалізується за допомогою графічних засобів UIKit та Core Graphics. Зокрема, для створення ліній, контурів і кривих доцільно використовувати UIBezierPath, який дозволяє описувати графічні об'єкти у вигляді шляхів [11]. Для складнішого малювання може застосовуватися Core Graphics, який забезпечує роботу з графічним контекстом, координатами та стилями відображення [3].

Після розміщення елементів користувач вводить назву вправи та її опис. Назва використовується для відображення вправи у бібліотеці, статистиці, цілях і рекомендаціях. Опис містить пояснення правил виконання вправи, умов оцінювання та додаткових зауважень. Після натискання кнопки збереження створена вправа додається до локального сховища застосунку та стає доступною в підменю «Бібліотека».

Реалізація підменю «Створити» має важливе практичне значення. У реальному тренувальному процесі гравець або тренер часто стикається з конкретними ситуаціями, які потрібно окремо відпрацювати. Можливість швидко

створити власну схему дозволяє адаптувати застосунок до індивідуальних потреб і робить його більш гнучким інструментом тренування.

3.4. Реалізація підменю «Рекомендації»

Підменю «Рекомендації» призначене для автоматичного визначення вправ, які потребують уваги користувача. На відміну від бібліотеки, де відображаються всі доступні вправи, у цьому підменю показуються лише ті вправи, які є найбільш важливими з погляду поточної статистики. Для реалізації списку вправ використано *UIScrollView* у поєднанні з вертикальним *UIStackView* по прикладу підменю «Бібліотека».

У застосунку передбачено три категорії рекомендацій:

- «Критично»;
- «Рекомендовано»;
- «Покращення».

Категорія «Критично» містить вправи, у яких спостерігається негативна динаміка результатів. Це означає, що трендова крива для такої вправи має від’ємний нахил, який перевищує встановлений допуск. Такі вправи потребують першочергової уваги, оскільки результати користувача погіршуються.

Категорія «Рекомендовано» містить вправи, у яких результати залишаються відносно стабільними. У цьому випадку нахил трендової кривої знаходиться в межах допустимого інтервалу. Такі вправи доцільно продовжувати виконувати для підтримання рівня або поступового покращення.

Категорія «Покращення» містить вправи з позитивною динамікою. Це означає, що результати користувача покращуються, а трендова крива має додатний нахил, що перевищує встановлений допуск. Ця категорія дозволяє користувачеві бачити, у яких вправах він уже демонструє прогрес.

Алгоритм формування рекомендацій використовує результати тренувань, згладжені значення та коефіцієнт нахилу тренду. Для кожної вправи застосунок отримує історію результатів, впорядковує її за датою, обчислює згладжені значення та визначає напрям зміни показників. Такий підхід ґрунтується на аналізі часових рядів, що є поширеним методом роботи з даними, які змінюються в часі [16, 17, 24].

Щоб рекомендації залишалися компактними та не перетворювалися на дублювання бібліотеки, у кожній категорії відображається не більше чотирьох вправ. Це дозволяє користувачеві швидко зрозуміти основні пріоритети тренування. У категорії «Критично» вправи сортуються від найбільш негативного тренду до менш негативного. У категорії «Покращення» вправи можуть сортуватися від найбільшого позитивного тренду до меншого.

Практична цінність модуля рекомендацій полягає в тому, що користувач отримує не просто статистичну інформацію, а конкретну підказку щодо подальших дій. Якщо певна вправа потрапляє до категорії «Критично», це означає, що її варто включити до найближчих тренувань. Якщо вправа перебуває у категорії «Рекомендовано», її можна використовувати для підтримання стабільності. Якщо вправа знаходиться у категорії «Покращення», користувач бачить підтвердження позитивної динаміки.

3.5. Реалізація підменю «Цілі»

Підменю «Цілі» призначене для постановки та контролю індивідуальних тренувальних завдань. Наявність такого модуля дозволяє користувачеві не лише переглядати статистику, а й працювати з конкретними бажаними результатами. Для реалізації списку вправ використано *UIScrollView* у поєднанні з вертикальним *UIStackView* по прикладу підменю “Бібліотека”.

У застосунку цілі поділяються на дві основні категорії:

- «В процесі»;
- «Виконані цілі».

Категорія «В процесі» містить активні цілі, які користувач встановив для конкретних вправ, але ще не виконав. Кожна ціль пов'язана з певною вправою та має числове значення. Для вправ на точність це може бути бажаний відсоток успішних ударів, а для рекордних вправ — конкретне рекордне значення.

Категорія «Виконані цілі» містить історію досягнутих цілей. На відміну від активного списку, цей розділ може містити повтори однієї й тієї самої вправи, оскільки він виконує роль журналу досягнень. Наприклад, користувач може декілька разів встановлювати нові цілі для однієї вправи та поступово підвищувати бажаний результат.

Процес роботи з ціллю складається з декількох етапів. Спочатку користувач обирає вправу та задає цільове значення. Після цього ціль зберігається у списку активних. Коли користувач виконує вправу та записує новий результат, застосунок автоматично порівнює введене значення з цільовим. Якщо результат дорівнює або перевищує ціль, вона вважається виконаною.

Після виконання цілі застосунок додає відповідний запис до історії виконаних цілей. Такий запис може містити назву вправи, досягнутий результат, цільове значення та дату виконання. Для візуального позначення виконаної цілі може використовуватися зелена галочка або інший зрозумілий індикатор.

Важливо, що додавання запису до історії виконаних цілей відбувається автоматично. Користувач не додає виконану ціль вручну, оскільки це могло б призвести до помилок або дублювання некоректних даних. Натомість застосунок самостійно визначає факт виконання на основі введенного результату.

Модуль цілей має мотиваційне значення. Він дозволяє користувачеві бачити не лише загальну статистику, а й конкретні досягнення. Крім того, постановка цілей робить тренувальний процес більш структурованим, оскільки користувач

працює не просто над абстрактним покращенням, а над досягненням визначених показників.

3.6. Реалізація підменю «Статистика»

Підменю «Статистика» призначене для візуального аналізу результатів тренувань. Воно дозволяє користувачеві оцінити зміну показників у часі, побачити фактичні результати та визначити загальну тенденцію прогресу. Для малювання ліній та кривих використовується *UIBezierPath*, що дозволяє гнучко керувати формою графіка та стилем його відображення.

Основна область екрана статистики призначена для графіка. На ньому відображаються результати обраної вправи за певний період. Користувач може обирати період аналізу, наприклад 7, 30 або 90 днів. Це дозволяє аналізувати як короткострокові зміни, так і довготривалу динаміку (рис. 3.4).

Нижче графіка розміщується список вправ. При виборі конкретної вправи застосунок оновлює графік і показує статистику саме для неї. Такий підхід є зручним, оскільки користувач не переходить між багатьма окремими екранами, а може швидко перемикатися між вправами в межах одного підменю.

Для вправ на точність на графіку відображається відсоток успішності. Для рекордних вправ відображається числове значення результату. Це дозволяє використовувати спільний механізм побудови графіка, але з урахуванням типу вправи.

Окрім фактичних результатів, на графіку може відображатися згладжена трендова крива. Вона будується на основі ковзного середнього і дозволяє зменшити вплив випадкових коливань. Наприклад, якщо один результат був значно нижчим через втому або неуважність, трендова крива не повинна різко змінюватися лише через один запис.

Під час побудови графіка застосунок не враховує відсутні дні як нульові значення. Це важливо, оскільки користувач може тренуватися нерегулярно, і відсутність запису не означає поганий результат. Аналіз виконується лише за фактично збереженими тренуваннями.

Підменю «Статистика» також пов'язане з модулем рекомендацій. Рекомендації формуються саме на основі аналізу статистичних даних. Тому користувач може спочатку побачити вправу в категорії «Критично» або «Покращення», а потім перейти до статистики цієї вправи та переглянути детальнішу динаміку.



Рисунок. 3.4. Трендова крива (останні 90 днів)

3.7. Взаємодія між модулями застосунку

Оскільки застосунок складається з декількох взаємопов'язаних модулів, важливо забезпечити узгоджену взаємодію між ними. Один і той самий об'єкт даних може використовуватися в різних частинах програми. Наприклад, вправа, створена користувачем у підменю «Створити», повинна відображатися у бібліотеці, бути доступною для запису результатів, враховуватися у статистиці та брати участь у формуванні рекомендацій.

Основний сценарій взаємодії починається з вибору або створення вправи. Якщо користувач обирає готову вправу з бібліотеки, застосунок відкриває її детальний екран. Якщо користувач створює нову вправу, вона спочатку зберігається у локальному сховищі, після чого стає доступною в бібліотеці.

Після виконання вправи користувач вводить результат. Застосунок перевіряє коректність введених значень і зберігає запис тренування. Далі цей запис може бути використаний у трьох напрямках. По-перше, він додається до історії результатів вправи. По-друге, він враховується у статистиці та графіках. По-третє, він використовується для перевірки активних цілей.

Якщо для відповідної вправи існує активна ціль, застосунок порівнює новий результат із цільовим значенням. Якщо результат дорівнює або перевищує ціль, ціль вважається виконаною, а відповідний запис додається до історії виконаних цілей. Таким чином, користувачеві не потрібно вручну переносити ціль до завершених.

Після оновлення історії результатів може змінитися статистика вправи. Це, у свою чергу, може вплинути на рекомендації. Наприклад, якщо після декількох тренувань результати вправи погіршилися, вона може потрапити до категорії «Критично». Якщо результати поступово покращуються, вправа може перейти до категорії «Покращення».

На рисунку 3.5 зображено діаграму послідовностей, яка демонструє взаємодію користувача з основними модулями застосунку під час запису результату тренування.

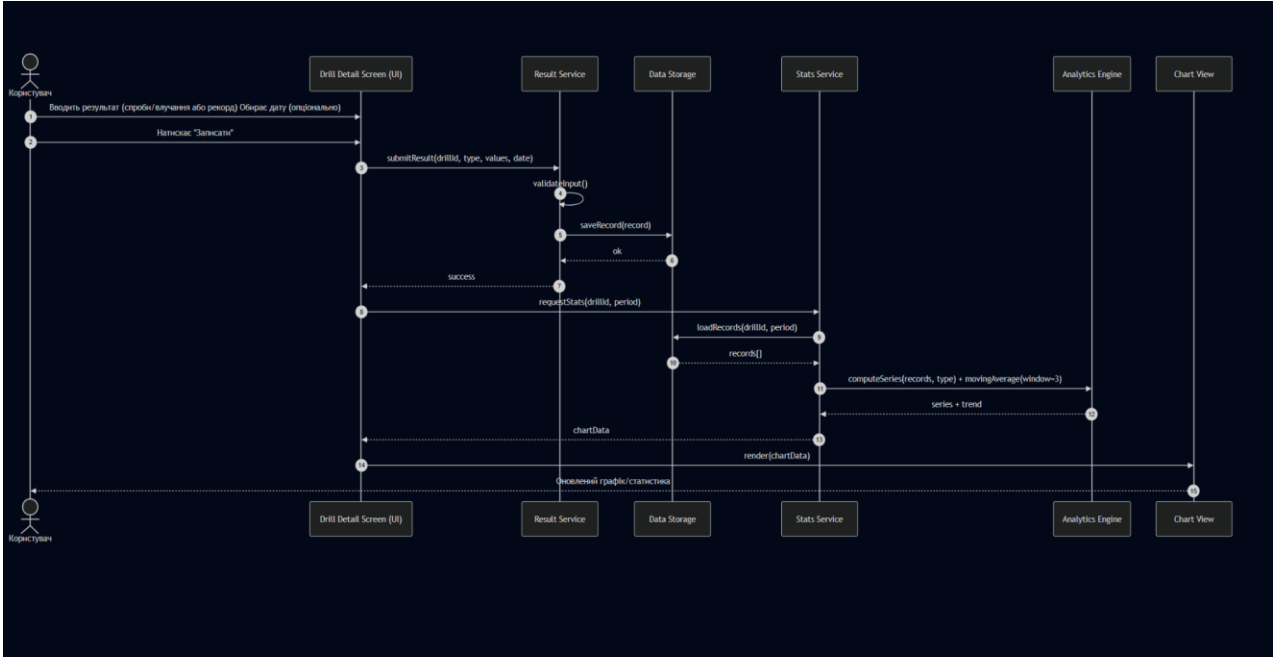


Рисунок. 3.5. Діаграма послідовностей

РОЗДІЛ 4. ТЕСТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ

4.1. Мета та підхід до тестування

Після реалізації основних функціональних модулів мобільного застосунку було проведено тестування з метою перевірки коректності його роботи, відповідності поставленим вимогам та стабільності основних користувацьких сценаріїв.

Основна увага під час тестування приділялася таким аспектам:

- коректність переходів між основними підменю застосунку;
- правильність відображення бібліотеки вправ;
- працездатність модуля створення користувацьких вправ;
- коректність введення та збереження результатів тренувань;
- правильність побудови статистичних графіків;
- коректність формування рекомендацій на основі трендових кривих;
- перевірка роботи модуля цілей;
- обробка граничних і помилкових ситуацій.

У межах даної роботи основний акцент було зроблено на функціональному тестуванні основних сценаріїв користувача, оскільки застосунок має відносно невеликий обсяг, локальну архітектуру та значну частину функціональності, пов'язану з користувацьким інтерфейсом. Для такого типу мобільного застосунку першочергове значення має перевірка того, чи коректно користувач може виконати основні дії: відкрити бібліотеку вправ, створити нову вправу, записати результат, переглянути статистику, отримати рекомендації та перевірити виконання цілей.

Повноцінне покриття застосунку юніт-тестами на даному етапі не є обов'язковим, оскільки більшість помилок у такому проєкті можуть виникати саме на рівні взаємодії між екранами, введення даних та відображення результатів. Такі

аспекти доцільніше перевіряти за допомогою ручного функціонального тестування та тестових сценаріїв [10, 15, 22, 23].

Водночас окремі алгоритмічні частини застосунку можуть бути виділені для подальшого юніт-тестування. До них належать обчислення відсотка результативності, розрахунок ковзного середнього, визначення нахилу трендової лінії, формування категорій рекомендацій та перевірка виконання поставлених цілей. Ці компоненти мають чіткі вхідні й вихідні дані, тому можуть бути протестовані ізольовано за допомогою автоматизованих тестів.

Таким чином, у межах кваліфікаційної роботи було обрано практичний підхід до тестування: основну увагу приділено перевірці користувацьких сценаріїв і працездатності функціональних модулів, а юніт-тестування розглядається як доцільний напрям подальшого вдосконалення проєкту.

Під час тестування також було визначено критерії успішного проходження перевірки для кожного функціонального блоку застосунку. Основним критерієм вважалася відповідність фактичної поведінки програми очікуваному результату: коректне відкриття екранів, збереження введених даних, правильне оновлення пов'язаних модулів та відсутність критичних помилок під час виконання основних дій. Для перевірки логіки роботи застосунку використовувалися підготовлені тестові набори даних із різними типами вправ, результатів і цільових значень. Це дозволило оцінити поведінку програми не лише за звичайних умов, а й у ситуаціях, коли дані мають різну кількість записів, різну динаміку зміни результатів або належать до різних типів оцінювання.

4.2. Функціональне тестування основних модулів

Функціональне тестування проводилося окремо для кожного основного підменю застосунку. Результати перевірки наведено в таблиці 4.1.

Таблица 4.1

Функціональне тестування основних модулів застосунку

№	Модуль	Перевірка	Очікуваний результат	Результат
1	Бібліотека	Відкриття списку вправ	Список вправ відображається коректно	Пройдено
2	Бібліотека	Вибір вправи	Відкривається детальний екран вправи	Пройдено
3	Створити	Переміщення кулі по столу	Куля змінює позицію відповідно до дотику	Пройдено
4	Створити	Додавання суцільної лінії	На схемі з'являється суцільна траєкторія	Пройдено
5	Створити	Додавання пунктирної лінії	На схемі з'являється пунктирна траєкторія	Пройдено
6	Створити	Збереження вправи без назви	Застосунок не дозволяє зберегти некоректну вправу	Пройдено
7	Створити	Збереження коректної вправи	Вправа додається до бібліотеки	Пройдено
8	Результати	Введення результату для вправи на точність	Обчислюється відсоток результативності	Пройдено
9	Результати	Введення рекордного значення	Числовий результат зберігається	Пройдено

Продовження табл. 4.1

10	Статистика	Вибір періоду 7/30/90 днів	Графік оновлюється відповідно до періоду	Пройдено
11	Рекомендації	Наявність негативного тренду	Вправа потрапляє до категорії «Критично»	Пройдено
12	Рекомендації	Наявність стабільного тренду	Вправа потрапляє до категорії «Рекомендовано»	Пройдено
13	Рекомендації	Наявність позитивного тренду	Вправа потрапляє до категорії «Покращення»	Пройдено
14	Цілі	Створення активної цілі	Ціль додається до списку «В процесі»	Пройдено
15	Цілі	Результат дорівнює цілі	Ціль позначається як виконана	Пройдено
16	Цілі	Результат перевищує ціль	Ціль додається до історії виконаних	Пройдено

ВИСНОВКИ

Завершення розробки та аналіз результатів кваліфікаційної роботи дозволяють стверджувати, що поставлена мета була досягнута шляхом проєктування та реалізації мобільного застосунку для персоналізованого тренування гравців у більярд. У ході дослідження було встановлено, що ефективний тренувальний процес у більярді потребує не лише регулярного виконання вправ, а й систематичного обліку результатів, аналізу динаміки прогресу та визначення індивідуальних напрямів для покращення. Обґрунтовано доцільність використання мобільного застосунку як інструменту, що дозволяє поєднати збереження тренувальних даних, візуалізацію статистики, постановку цілей та формування персоналізованих рекомендацій.

У процесі виконання роботи було досліджено особливості тренувального процесу у більярді, розглянуто існуючі програмні рішення для підтримки спортивних тренувань та визначено їхні функціональні обмеження. Аналіз показав, що значна частина подібних застосунків орієнтована переважно на перегляд готових вправ і не забезпечує повного циклу роботи з тренувальними даними: створення власних вправ, фіксації результатів, аналізу трендових кривих, формування рекомендацій та контролю досягнення цілей. Це підтвердило актуальність розробки власного мобільного застосунку, орієнтованого на індивідуальні потреби користувача.

Було обґрунтовано вибір платформи iOS, середовища розробки Xcode, мови програмування Objective-C та фреймворку UIKit. Спроєктовано архітектуру застосунку, моделі даних для збереження вправ, результатів тренувань, користувацьких схем, траєкторій, цілей і рекомендацій. Застосування модульного підходу дозволило логічно розділити основні функціональні частини системи та забезпечити їхню взаємодію в межах єдиного програмного продукту.

Практична реалізація застосунку охоплює п'ять основних підменю: «Бібліотека», «Створити», «Рекомендації», «Цілі» та «Статистика». Реалізовано можливість перегляду вправ, створення власних тренувальних схем із розміщенням куль і побудовою траєкторій, введення результатів тренування, перегляду графіків статистики та контролю індивідуальних цілей. Особливе значення має модуль рекомендацій, який використовує накопичені результати та аналіз трендових кривих для розподілу вправ за категоріями «Критично», «Рекомендовано» та «Покращення». Це дозволяє користувачеві не лише переглядати статистику, а й отримувати практичні підказки щодо подальшого тренувального процесу.

У роботі застосовано методи прикладної математики для обробки тренувальних результатів. Зокрема, використано обчислення середніх значень, побудову часових рядів, метод ковзного середнього та оцінювання нахилу трендової лінії. Такі методи дозволяють зменшити вплив випадкових коливань окремих результатів і більш об'єктивно визначати загальну динаміку прогресу користувача. На основі цього застосунок може виявляти вправи з негативною, стабільною або позитивною тенденцією.

Проведене тестування підтвердило працездатність основних функціональних модулів застосунку. Було перевірено створення користувацьких вправ, збереження результатів, побудову статистичних графіків, формування рекомендацій, роботу з активними цілями та автоматичне перенесення виконаних цілей до історії. Також перевірено обробку граничних ситуацій, зокрема введення некоректних значень, відсутність статистичних даних та недостатню кількість результатів для побудови тренду. Результати тестування показали, що застосунок працює відповідно до очікуваної логіки та може використовуватися як практичний інструмент підтримки тренувань.

Практичне значення роботи полягає у створенні мобільного застосунку, який допомагає систематизувати тренувальний процес гравця у більярд. Застосунок дозволяє зберігати вправи та результати їх виконання, створювати індивідуальні тренувальні завдання, аналізувати прогрес у часі, визначати слабкі сторони та контролювати досягнення поставлених цілей. Таким чином, розроблений програмний продукт сприяє більш усвідомленому, структурованому та персоналізованому підходу до тренувань.

Узагальнюючи результати виконаної роботи, можна зробити висновок, що розроблений мобільний застосунок є завершеним програмним продуктом, який відповідає поставленим вимогам і готовий до практичного використання в індивідуальному тренувальному процесі. Подальший розвиток застосунку може передбачати синхронізацію даних між пристроями, додавання нових типів аналітики, підтримку інших більярдних дисциплін та вдосконалення алгоритмів персоналізованих рекомендацій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Apple Inc. About Objective-C. URL: <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html> (дата звернення: 01.04.2026).
2. Apple Inc. Apple Developer Documentation. URL: <https://developer.apple.com/documentation> (дата звернення: 01.04.2026).
3. Apple Inc. Core Graphics Documentation. URL: <https://developer.apple.com/documentation/coregraphics> (дата звернення: 01.04.2026).
4. Apple Inc. Designing for iOS. URL: <https://developer.apple.com/design/human-interface-guidelines/designing-for-ios> (дата звернення: 01.04.2026).
5. Apple Inc. Drag and Drop. URL: <https://developer.apple.com/design/human-interface-guidelines/drag-and-drop> (дата звернення: 01.04.2026).
6. Apple Inc. Drag and Drop in UIKit. URL: <https://developer.apple.com/documentation/uikit/drag-and-drop> (дата звернення: 01.04.2026).
7. Apple Inc. Human Interface Guidelines. URL: <https://developer.apple.com/design/human-interface-guidelines> (дата звернення: 01.04.2026).
8. Apple Inc. Instruments Tutorials. URL: <https://developer.apple.com/tutorials/instruments> (дата звернення: 01.04.2026).
9. Apple Inc. TestFlight. URL: <https://developer.apple.com/testflight> (дата звернення: 01.04.2026).
10. Apple Inc. Testing in Xcode. URL: <https://developer.apple.com/documentation/xcode/testing> (дата звернення: 01.04.2026).

- 11.Apple Inc. UIBezierPath Documentation. URL:
<https://developer.apple.com/documentation/uikit/uiBezierPath> (дата звернення: 01.04.2026).
- 12.Apple Inc. UIKit Framework Documentation. URL:
<https://developer.apple.com/documentation/uikit> (дата звернення: 01.04.2026).
- 13.Apple Inc. UserDefaults Documentation. URL:
<https://developer.apple.com/documentation/foundation/userdefaults> (дата звернення: 01.04.2026).
- 14.Apple Inc. Xcode Documentation. URL:
<https://developer.apple.com/documentation/xcode> (дата звернення: 01.04.2026).
- 15.Apple Inc. XCTest Documentation. URL:
<https://developer.apple.com/documentation/xctest> (дата звернення: 01.04.2026).
- 16.Box G. E. P., Jenkins G. M., Reinsel G. C., Ljung G. M. Time Series Analysis: Forecasting and Control. Hoboken : Wiley, 2015. 712 p.
- 17.Chatfield C. The Analysis of Time Series: An Introduction. Boca Raton : CRC Press, 2016. 352 p.
- 18.Cockburn A. Writing Effective Use Cases. Boston : Addison-Wesley, 2001. 304 p.
- 19.Drill Room. Official Mobile Application for Pool Training. URL:
<https://drillroomapp.com> (дата звернення: 01.04.2026).
- 20.Few S. Show Me the Numbers: Designing Tables and Graphs to Enlighten. Burlingame : Analytics Press, 2012. 316 p.
- 21.Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston : Addison-Wesley, 1994. 395 p.
- 22.ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality

- models. URL: <https://www.iso.org/standard/35733.html> (дата звернення: 01.04.2026).
- 23.ISTQB. Certified Tester Foundation Level Syllabus. Version 4.0.1. URL: https://istqb.org/wp-content/uploads/2024/11/ISTQB_CTFL_Syllabus_v4.0.1.pdf (дата звернення: 01.04.2026).
- 24.Montgomery D. C., Runger G. C. Applied Statistics and Probability for Engineers. Hoboken : Wiley, 2018. 784 p.
- 25.Moran K. Usability Testing 101. Nielsen Norman Group. URL: <https://www.nngroup.com/articles/usability-testing-101/> (дата звернення: 01.04.2026).
- 26.Nielsen J. Usability 101: Introduction to Usability. Nielsen Norman Group. URL: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> (дата звернення: 01.04.2026).
- 27.Object Management Group. Unified Modeling Language Specification. URL: <https://www.omg.org/spec/UML> (дата звернення: 01.04.2026).
- 28.Pressman R. S. Software Engineering: A Practitioner's Approach. New York : McGraw-Hill Education, 2014. 720 p.
- 29.Sommerville I. Software Engineering. 10th ed. Boston : Pearson Education, 2016. 816 p.