

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

**Волобоєв Микита Олександрович**

**УДК 004.51**

**«ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ВІДСТЕЖЕННЯ ЕФЕКТИВНОСТІ  
ТРЕНУВАНЬ СПОРТСМЕНІВ»**

**Автореферат кваліфікаційної роботи на здобуття  
ступеня вищої освіти «Бакалавр»**

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня кваліфікація  
«Бакалавр з інженерії програмного забезпечення»

**Миколаїв – 2021**

Кваліфікаційною роботою є рукопис.

Робота виконана в Чорноморському національному університеті імені Петра Могили Міністерства освіти і науки України на кафедрі інженерії програмного забезпечення.

Керівник: канд. техн. наук, доцент (б.в.з.)  
кафедри інженерії програмного  
забезпечення

\_\_\_\_\_

*підпис*

Горбань Гліб Валентинович

Рецензент канд. техн. наук, доцент, доцент  
кафедри інтелектуальних  
інформаційних систем

\_\_\_\_\_

*підпис*

Кондратенко Галина Володимирівна

Захист відбудеться « 24 » червня 2021р. о 9 год. на засіданні екзаменаційної комісії (ауд. 2 309 ) у Чорноморському національному університеті імені Петра Могили за адресою: вул. 68 Десантників, 10, Миколаїв, 54003.

З кваліфікаційною роботою можна ознайомитись в бібліотеці Чорноморського національного університету імені Петра Могили за адресою: вул. 68 Десантників, 10, Миколаїв, 54003.

Автореферат представлений « 18 » червня 2021р.

Секретар  
екзаменаційної комісії

викладач

(наук. ступінь, вчене звання)

І. О. Кандиба

(прізвище та ініціали)

## **ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ**

**Об'єктом кваліфікаційної роботи** є показники ефективності спортивних тренувань спортсменів кафедри фізичного виховання.

**Предметом кваліфікаційної роботи** є процеси збору та аналізу спортивних результатів.

**Актуальність теми кваліфікаційної роботи** полягає в підвищенні ефективності роботи тренерів факультету фізичного виховання та спорту, виконання постанови КМУ «Цифрова трансформація сфери фізичної культури і спорту» від 4 листопада 2020 р. № 1089, шляхом розробки системи, яка має допомогти зі збереженням та аналізом даних досягнень у тренуваннях спортсменів ЧНУ.

**Метою кваліфікаційної роботи** є підвищення ефективності тренувального процесу спортсменів шляхом розробки системи, яка дозволяла б переглядати статистичну інформацію тренування студентів, порівняння її між факультетами, групами та або роками, виконання постанови Кабінету Міністрів України зі стратегії розвитку фізичної культури і спорту.

Для досягнення визначеної мети необхідно вирішити такі завдання:

- 1) Дослідити об'єктну та предметну область.
- 2) Проаналізувати аналоги, що вже існують.
- 3) Спроекувати систему та базу даних.
- 4) Розробити користувацький інтерфейс.
- 5) Побудувати та відлагодити систему.

**Апробація результатів кваліфікаційної роботи.** Результат роботи був представлений на XV міжнародній науковій конференції «Ольвійський форум-2021: стратегії країн Причорноморського регіону в геополітичному просторі».

КРБ викладена на 86 сторінки, вона містить 6 розділи, 31 ілюстрацій, 5 таблиця, 51 джерел в переліку посилань та спеціальна частина з охорони праці.

## **Основний зміст КРБ**

У першому розділі дипломної роботи розглянуто об'єкт та предмет досліджень, проведено аналіз схожих систем з відслідковування тренувань, сформовано специфікацію вимог.

Зараз є безліч пристроїв, які можуть полегшити збір даних, такі як тренажери з вбудованою електронікою, яка дозволяє відстежувати показники та реєструвати їх. Але їх небагато та для таких завдань, як наприклад, віджимання, або біг на коротку дистанцію, та подібних ним таких пристроїв ще не придумали і в більшості випадків для реєстрації результатів таких завдань необхідний тренер, який буде фіксувати результат.

На жаль в інформаційну еру головним пристроєм для фіксування результатів тренування спортсмена є папір та ручка. Після чого, в кращому випадку, результат переноситься в якийсь журнал.

Після накопичення інформації, виникає потреба в її аналізі, яку згодом можна використати для вдосконалення тренувань спортсменів. А збереження її на паперових носіях цьому не сприяє – виникає необхідність в додаткових зусиллях для її аналізу.

Цю проблему якнайкраще може вирішувати за допомогою сучасних інформаційних технологій.

Через специфіку даної області, а саме через те, що тренування відбуваються не вдома, в більшості випадків, а наприклад в тренувальних залах або на свіжому повітрі (пробіжка на стадіоні або просто на вулиці), під рукою є тільки смартфон. Тому більшість рішень розроблені саме для смартфона, тому огляд буде застосунків для Android та IOS.

Огляд застосунків:

**WORKOUT TRAINER**

Переваги:

- велика кількість програм тренувань;
- підтримка Google Fit;
- детальний аудіо супровід;

- можливість завантаження тільки окремих тренувань;
- можливість створити своє тренування;
- можливість додавати вправи в обране.

Недоліки:

- необхідність авторизації при першому запуску;
- недостатня простота використання;
- тільки англійська мова;

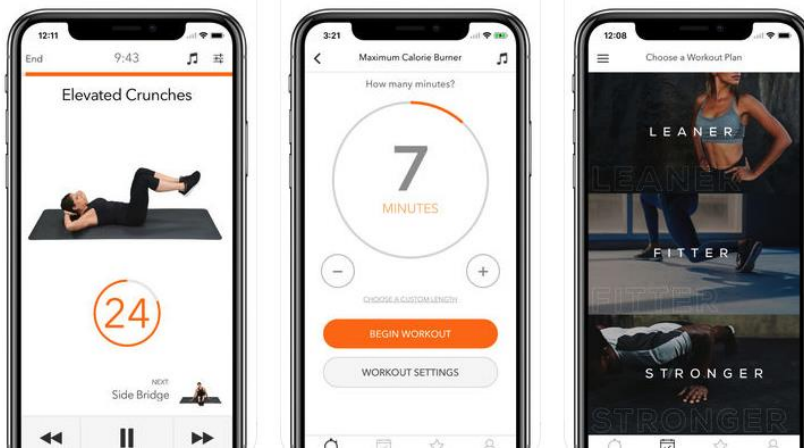


Рисунок 1 – Інтерфейс застосунку Workout Trainer

Фітнес – тренер FitProSport

Переваги:

- велика кількість програм тренувань;
- наявність довідника вправ;
- можливість створення своєї тренування або вправи;
- просунута статистика;
- наявність експорту даних.

Недоліки:

- неможливість додавання вправи або тренувань в обране;
- відсутність соціальної складової;
- повна версія сильно розширює можливості;
- відсутність аудіо супроводу.



Рисунок 2 – Інтерфейс застосунку FitProSport

## NIKE+ TRAINING CLUB

### Переваги:

- відсутність вбудованих покупок;
- велика кількість програм тренувань;
- підтримка Google Fit;
- детальний аудіо супровід;
- можливість завантаження тільки окремих тренувань;
- наявність розділу з корисними статтями;
- відсутність рекламних матеріалів;
- наявність програми тренувань з можливістю налаштування.

### Недоліки:

- необхідність авторизації при першому запуску;
- неможливість створити своє тренування;
- відсутність довідника вправ;
- велика вага застосунку.

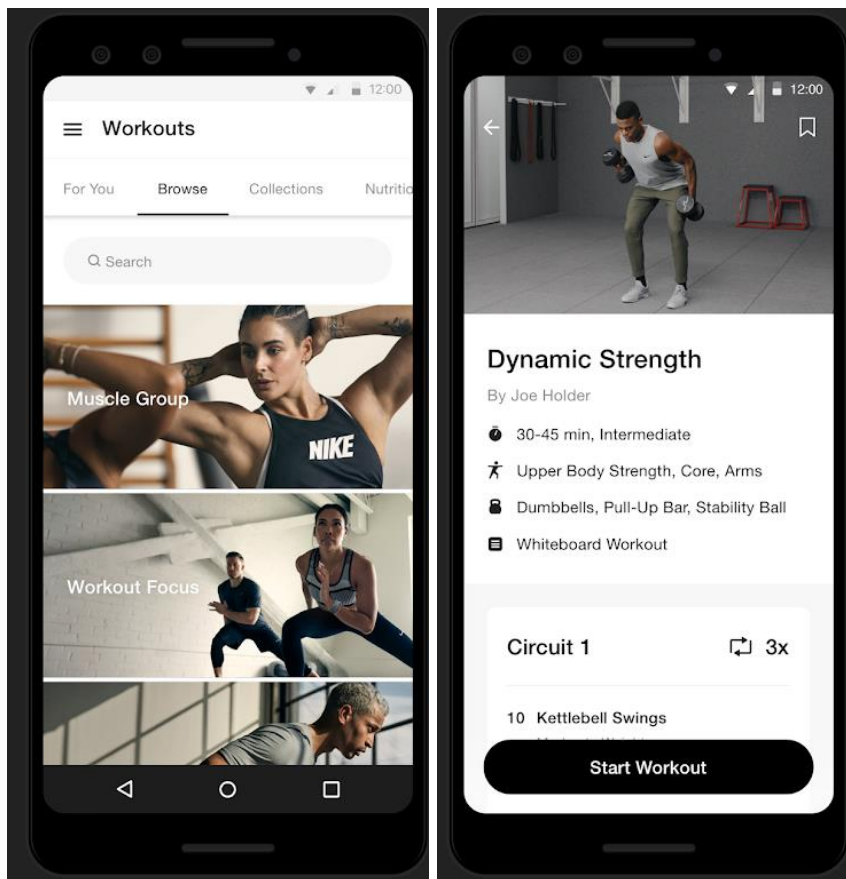


Рисунок 3 – Інтерфейс застосунку NIKE+ TRAINING CLUB

Також у якості недоліку всім розглянутим застосункам можна додати те, що переглянути результати тренувань може лише користувач, на смартфоні якого і встановлений застосунок.

Ще одним недоліком є неможливість використовувати на інших пристроях, які не використовують ОС Android та IOS.

У **другому розділі** досліджена предметна область, здійснено моделювання предмету дослідження, описані обрані засоби та інструменти для розробки системи, розглянуто варіанти та сценарії використання, процес моделювання бізнес логіки системи з використання UML діаграм, прототип користувацького інтерфейсу, моделі та база даних.

Ключовим бізнес-процесом системи, що проєктується, вважається процес збору та аналізу спортивних результатів.

Суб'єктом бізнес-процесу є «Програмне забезпечення відстеження ефективності тренувань спортсменів».

На вхід надходять дані:

- список спортсменів;
- список тестів;
- дані результатів тренувань.

Механізмами виконання процесу вважаються:

- тренери, викладачі, адміністратори.

Керуванням є:

- правила проведення контрольних тестів факультету фізичного виховання.

На виході отримуємо:

- вивід обробленої інформації результатів тестування.

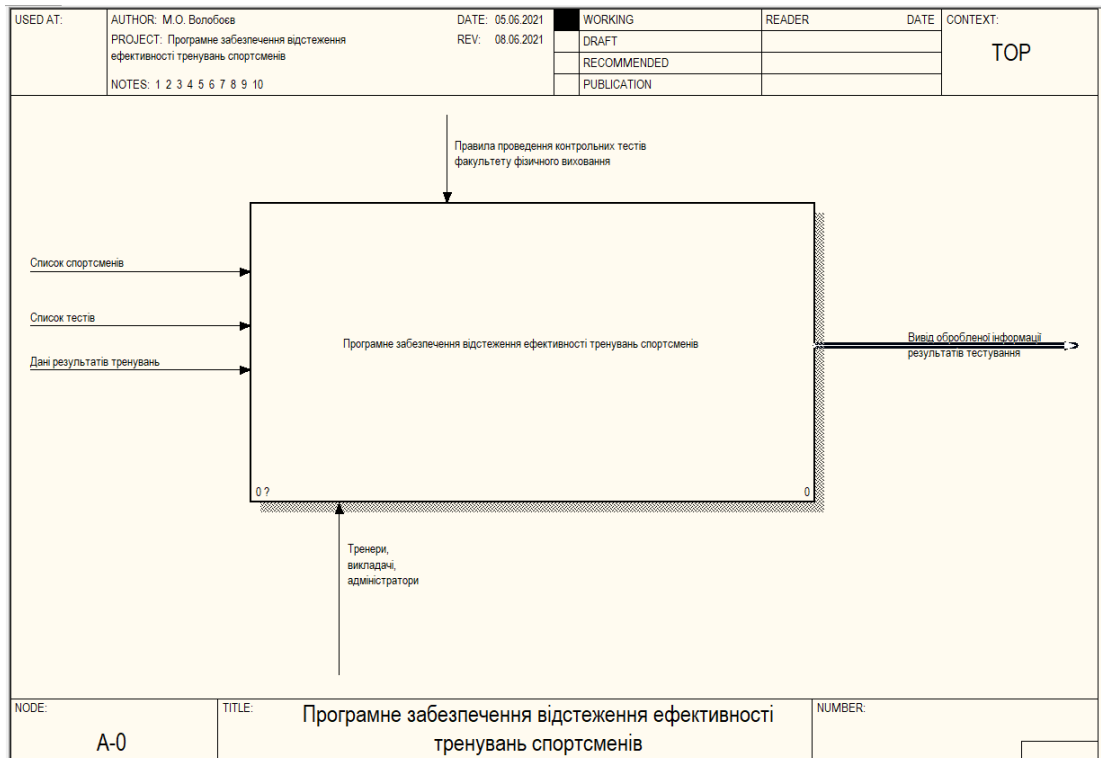


Рисунок 4 – Діаграма IDEF0 «Програмне забезпечення відстеження ефективності тренувань спортсменів»

Провідними бізнес-функціями процесу «збору та аналізу спортивних результатів» стануть:

- 1) Створення бази з спортсменами.
- 2) Створення бази з тестами.
- 3) Фіксація результатів тренувань спортсменів.
- 4) Формування запиту.



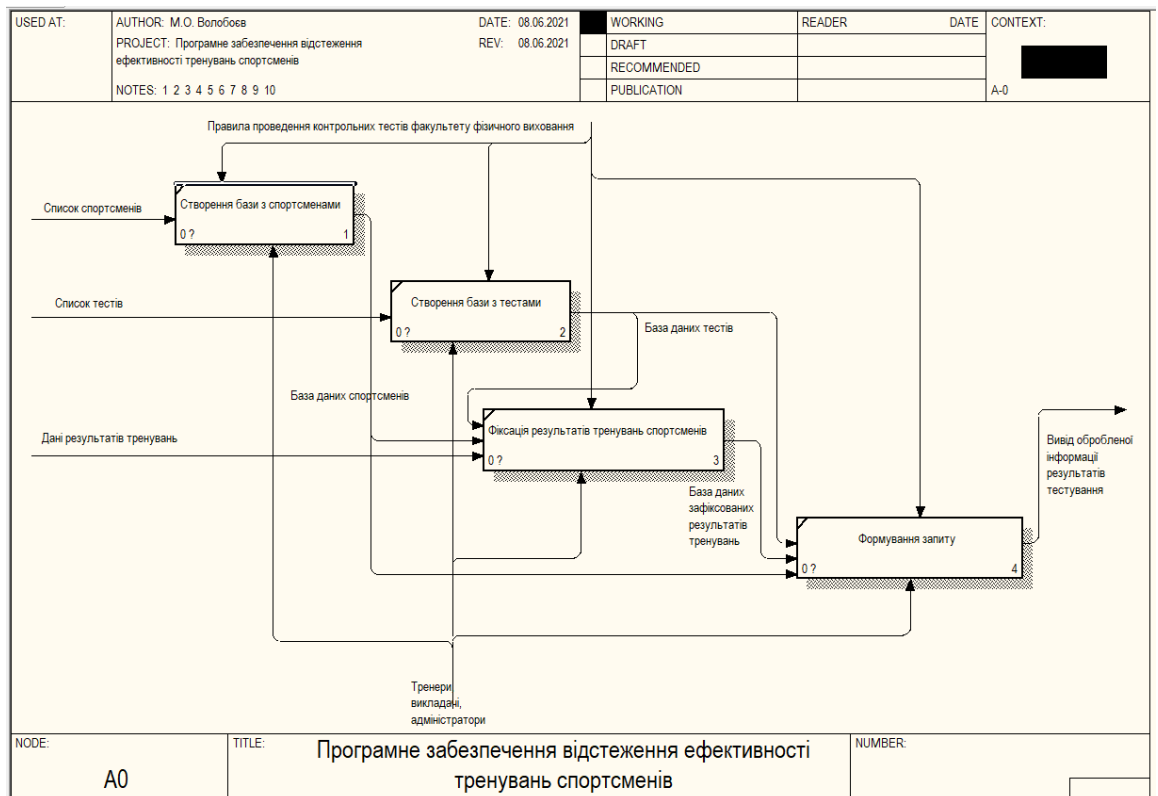


Рисунок 5 – Повна діаграма IDEF0 «Програмне забезпечення відстеження ефективності тренувань спортсменів»

Для розробки серверної частини було вирішено обрати бібліотеку ASP.NET Core на мові C#.

ASP.NET Core – безкоштовне та відкрите для використання програмне забезпечення каркаса вебзастосунків, розроблене корпорацією Microsoft і співтовариством. Це модульна структура, яка працює як на повній платформі .NET Framework, так і на платформі ASP.NET Core. Бібліотека реалізує шаблон архітектурний Model-View-Controller, основною метою якого є поділ системи на три взаємопов’язані частини: модель даних (Model), вигляд або інтерфейс користувача (View) та модуль керування (Controller). Цей архітектурний шаблон застосовується для поділу даних або моделей від вигляду (інтерфейсу користувача) так, щоб зміни в моделях даних могли здійснюватися без змін інтерфейсу користувача, а зміни в інтерфейсі користувача мінімально впливали на роботу з даними.

Для побудови графіку використовується використовується Google Chart. Google Chart – це безкоштовна бібліотека діаграм, яка основана на

чистому JavaScript, має широкий вибір типів графіків, відмічається легкістю та гнучкістю у використанні. Завдяки тому, що для побудови графіки використовується її підтримують усі сучасні браузері.

Для розробки застосунку побудуємо Usecase-діаграму з декількома варіантами використання системи.

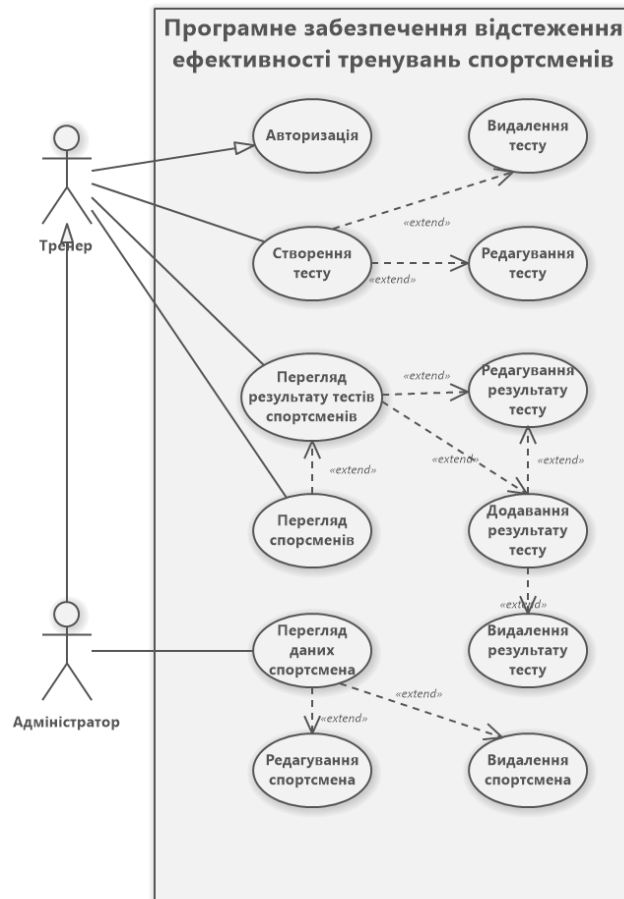


Рисунок 6 – Діаграма варіантів використання

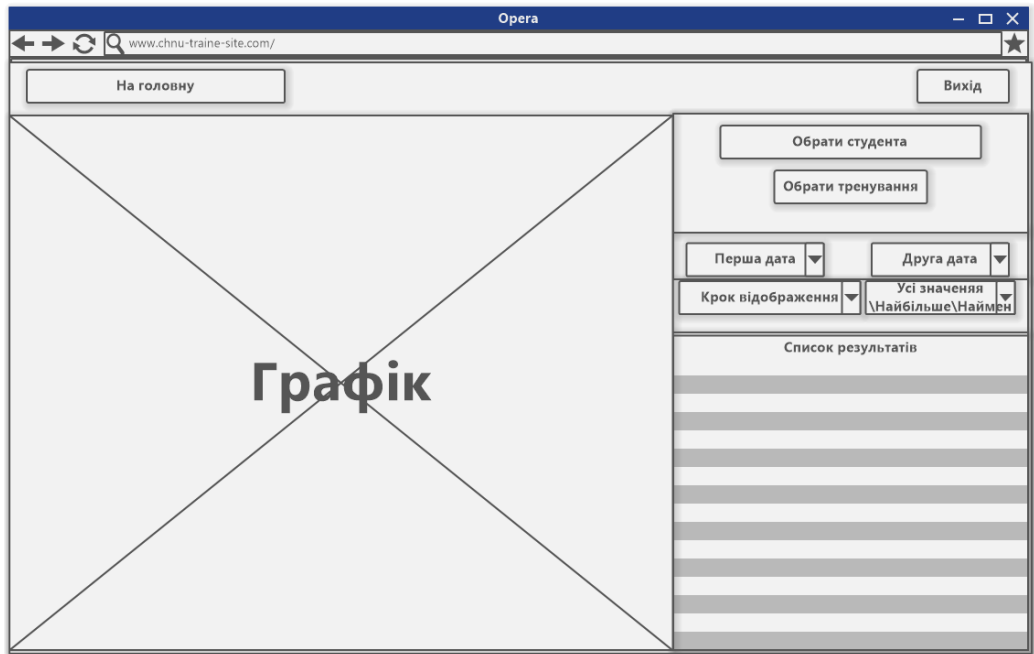


Рисунок 7 – Прототип сторінки перегляду результатів тренувань

Ця сторінка дозволяє переглянути результати тренувань за допомогою графіку, також в лівому нижньому куті, «Список результатів», можна буде переглянути список результатів які будуть відображатися на графіку.

Для перегляду необхідно:

- обрати студентів, натиснувши на кнопку «Обрати студента», що перенесе на сторінку, де можна буде обрати студентів;
- обрати тренування, натиснувши на кнопку «Обрати тренування», яка також перенесе на окрему сторінку, де можна обрати необхідне тренування;
- обрати проміжок, за який час показати графік: обирається перша дата в «Перша дата» та друга дата в «Друга дата», якщо перша дата пізніша за другу то виводиться повідомлення з помилкою, в якій буде вказано на цю помилку;
- крок відображення, натиснувши на меню, що випадає, можна буде обрати з яким кроком буде відображатися графік: рік, місяць, тиждень, день;

- обрати, які результати відображати: якщо за вказаний крок у спортсмена є декілька результатів за обраним тестом, можна обрати який з них необхідно відобразити у графіці: найбільше значення – буде обраний найбільший результат, найменше значення – буде обраний найменший результат, середнє значення – буде підрахований з усіх значень середній результат, усі значення – будуть виведені усі збережені раніше результати тестів.

Сторінка оновлюється щоразу як змінюється будь-який фільтр відображення графіку.

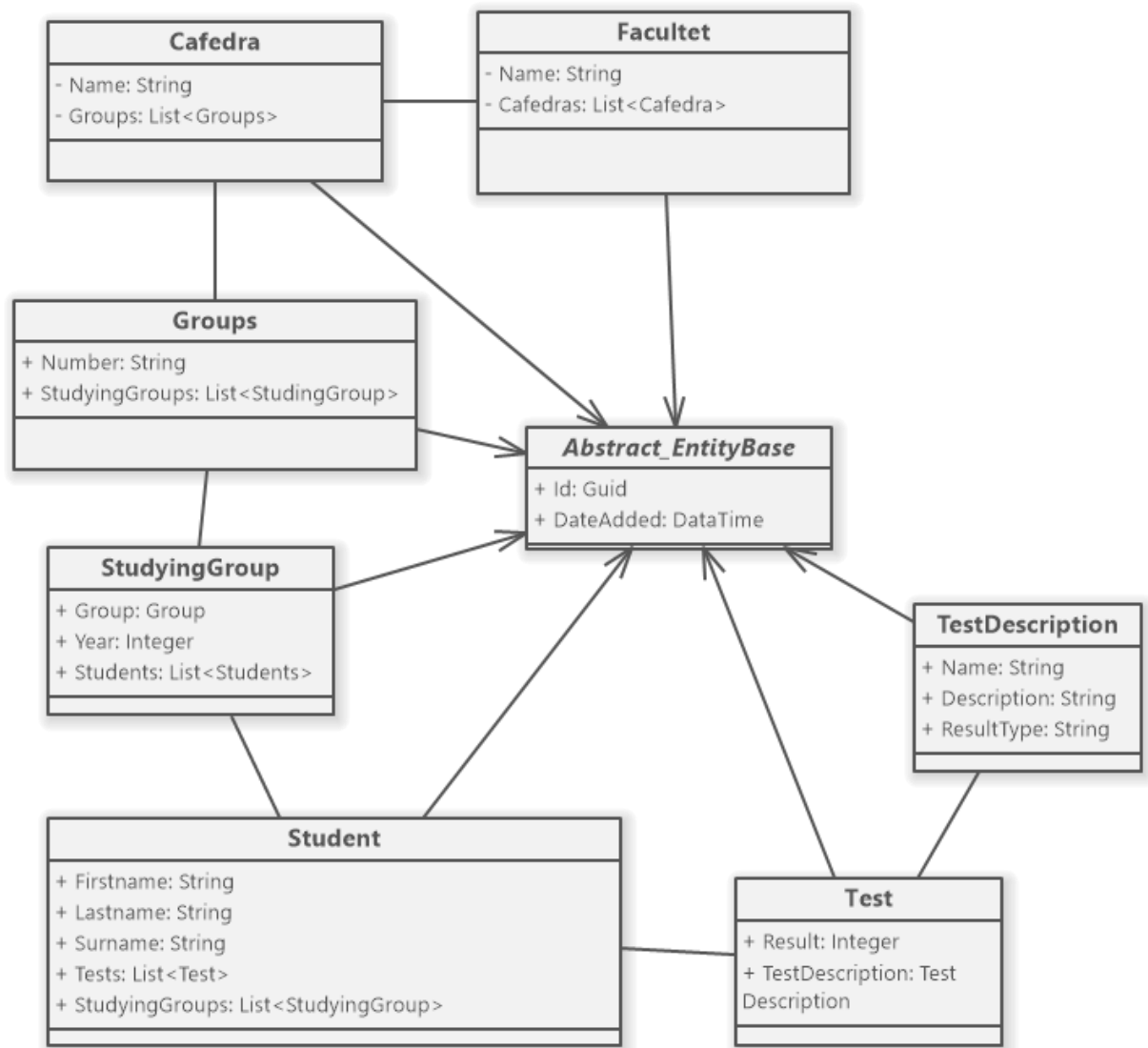


Рисунок 8 – Діаграма класів моделей

## Опис абстрактного класу EntityBase

**EntityBase** – абстрактний клас, необхідний для спрощення коду, так як він має властивості, які будуть мати всі інші класи моделей, тому для зменшення коду був створений окремий абстрактний клас.

**Id: Guid** – властивість типу **Guid**, яка зберігає унікальний ідентифікатор, що відповідає ідентифікатору в базі даних.

**DateAdded: DateTime** – властивість типу **DateTime**, яка зберігає час створення об'єкту, яка записана у базі даних.

## Опис класу Facultet

**Facultet** – сутність факультету, містить назву факультету та список кафедр, які до нього входять.

**Name: String** – властивість, містить в собі назву факультету.

**Cafedras: List<Cafedra>** – список кафедр, що входять до поточного факультету.

## Опис класу Cafedra

**Cafedra** – сутність кафедри, містить назву кафедри та список груп, які до нього входять.

**Name: String** – властивість, містить в собі назву кафедри.

**Groups: List<Group>** – список груп, що входять до поточної кафедри.

## Опис класу Group

**Group** – сутність групи, містить назву групи та список навчальної групи, які до неї входять.

**Number: String** – властивість, містить в собі номер групи.

**StudyingGroups: List<StudyingGroup>** – список груп з роками навчання.

## Опис класу Student

**Student** – сутність студента або спортсмена, містить ПІБ, список складених тестів та список груп з роками в яких він навчався.

**Firstname: String** – властивість, містить в собі ім'я спортсмена.

**Lastname: String** – властивість, містить в собі прізвище спортсмена.

**Surname: String** – властивість, містить в собі по батькові спортсмена.

**Tests: List<Test>** – список складених спортсменом тестів.

**StudyingGroups: List<StudyingGroup>** – список груп з роками навчання в яких перебував студент.

Опис класу TestDescription

**TestDescription** – сутність опису завдання, містить в собі назву завдання, опис та тип результату який потребує тест.

**Name: String** – властивість, містить в собі назву завдання.

**Description: String** – властивість, містить в собі опис завдання.

**ResultType: String** – властивість, містить в собі тип результату який потребує тест.

Опис класу Test

**Test** – сутність виконаного тесту, містить в собі результат, на який склав спортсмен та опис завдання.

**Result: Integer** – властивість, містить в собі результат складання тесту.

**TestDescription: TestDescription** – властивість, містить в собі сутність опису завдання.

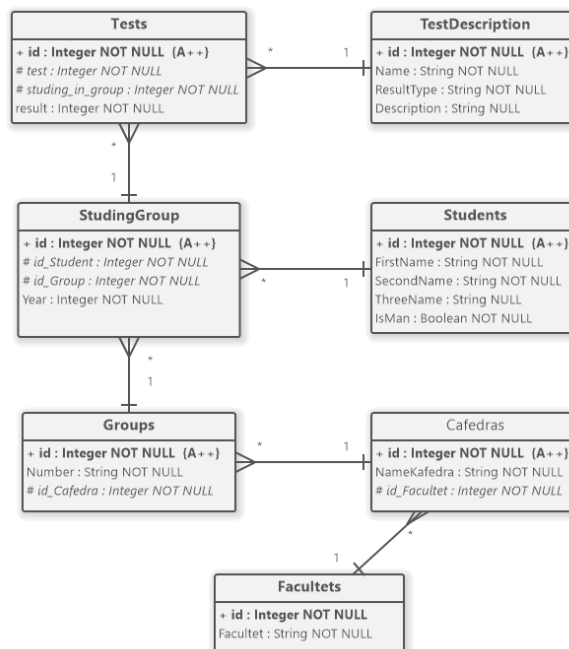


Рисунок 9 – ERD діаграма бази даних

Таблиця **Facultets** містить список назв факультетів.

Таблиця **Cafedras** містить список назв кафедр, має обов'язковий зовнішній ключ до таблиці **Facultets**, до яких факультетів належать кафедри.

Таблиця **Groups** містить список груп які взагалі існують, закріплені за кафедрами, має обов'язковий зовнішній ключ до таблиці **Cafedras**, до яких кафедр належать групи.

Таблиця **Students** містить в собі список студентів: дані про їх прізвище, ім'я та, не обов'язково для заповнення, по батькові, інформацію про те, що студент дівчина чи ні.

Таблиця **StudingGroup**, фіксує в собі переміщення студента по групах, містить в собі обов'язковий зовнішній ключ до таблиці **Students** та до таблиці **Groups** та рік навчання.

Таблиця **TestDescription** зберігає в собі список усіх створених тестів, містить назву тесту, тип результату, та не обов'язкове поле опису тесту.

Таблиця **Tests** зберігає в собі результати складених тестів спортсменами, має зовнішні ключі до таблиць **TestDescription** та **StudingGroup**.

У третьому розділі продемонстрована робота системи.

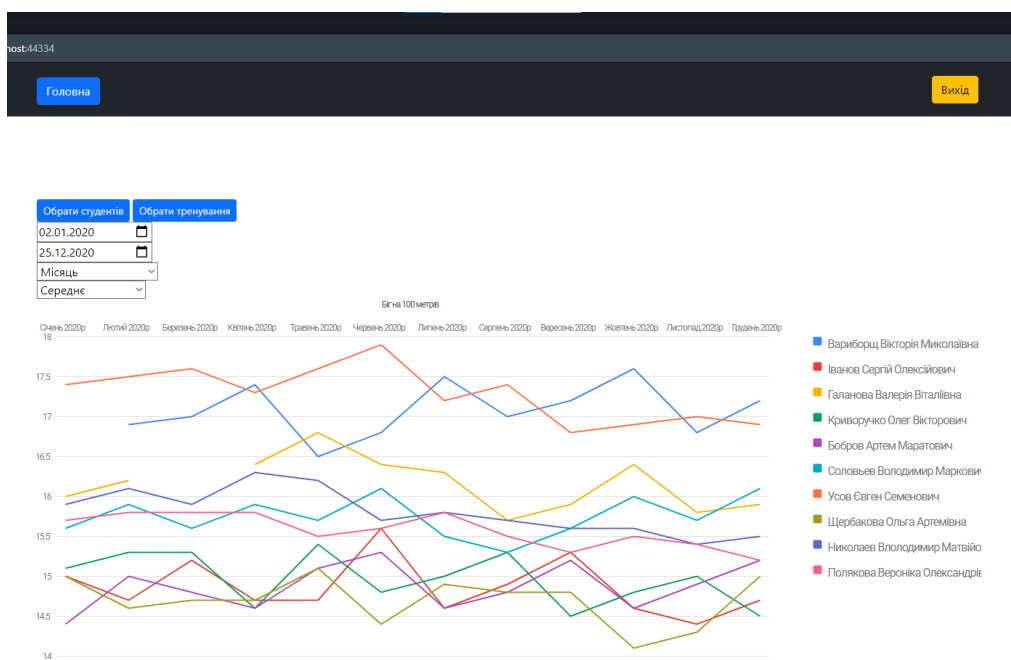


Рисунок 10 – Сторінка з графіком результату тренувань

На цій сторінці можна переглянути графік досягнень тренувань спортсменів за допомогою графіку. Для перегляду спочатку необхідно зробити:

- обрати спортсменів для перегляду, це можна зробити натиснувши на кнопку «Обрати студентів», після якого буде перенаправлено на сторінку з пошуком спортсменів де можна їх обрати, після того як вони будуть обрані та користувач підтвердить свій вибір його перенаправить на сторінку з переглядом графіку;

- обрати до якого тренування або завдання необхідно побудувати графік, це виконується на спеціальній сторінці, де обирається завдання та перенаправляється за схожим принципом як і при виборі спортсменів, на сторінку можна потрапити натиснувши на кнопку «Обрати тренування»;

- налаштувати проміжок часу, на протязі якого беруться з БД результати тренувань, необхідно обрати першу дату – початок пошуку та другу – остання дата пошуку;

- налаштувати «крок відображення» – дозволяє задати з яким кроком буде побудовано графік: з кроком в рік, місяць, тиждень та день;

- налаштувати як будуть оброблені результати тренувань, в випадаючому списку «Обрати значення»: можливо обрати усі значення – в графіці будуть використовуватися всі раніше збережені значення, найбільше значення – буде обрано найбільше значення результату на протязі одного кроку; найменше значення – принцип роботи як і в «найбільшому значенню», але обирає найменший результат; середнє значення – підраховує середнє значення на основі наявних результатів за обраний крок.



Рисунок 11 – Розрив в діаграмі



В разі, якщо у спортсмена не буде жодного запису виконаного обраного завдання, праворуч від графіку його ПІБ буде відображатись, але в самому графіці він буде відсутній. А в тому разі, якщо за спортсменом є зареєстровані результати тренувань, але деякі кроки відсутні, так в місце де відсутній результат буде розрив, або якщо на початку або в кінці графіку то як відсутній результат.

Графік не статичний, при наведенні на лінію одного зі спортсменів вся його лінія становиться більшою у порівнянні з іншими, що покращує навігацію по графіку. Також вона підсвічується уразі наведення курсору на ПІБ спортсмена праворуч від графіку.

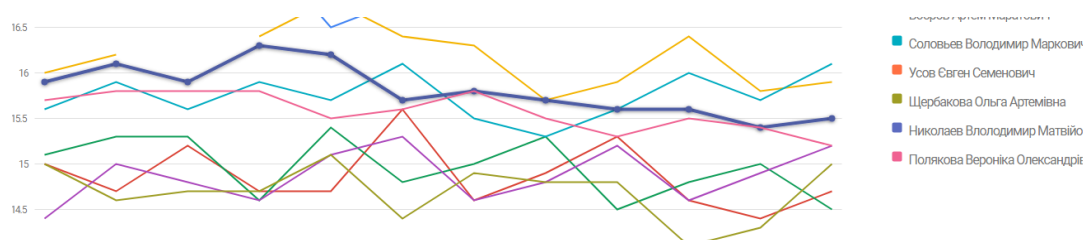


Рисунок 12 – Виділення результатів тренувань конкретного спортсмена

Також можливий перегляд результату одразу на графіку: необхідно перемістити курсор на одну з вершин і в спливаючій підказці буде вказано: на якому кроці, який спортсмен та результат.

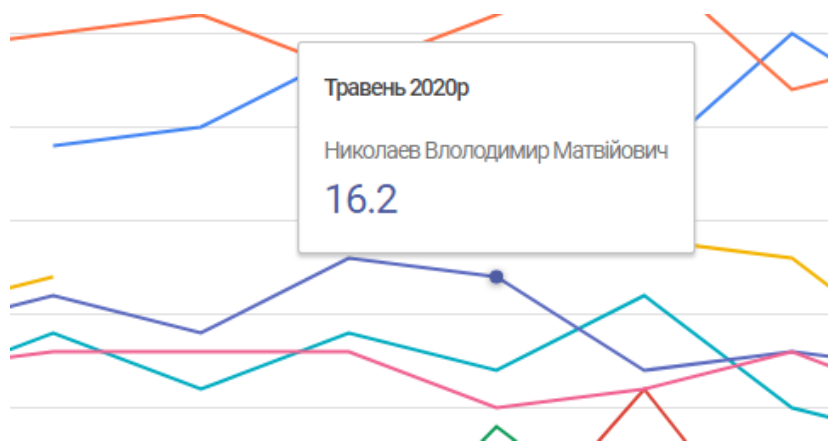


Рисунок 13 – Виділення результатів тренувань конкретного спортсмена

## **ВИСНОВКИ**

В ході виконання КРБ була досліджена об'єктна та предметна область. Розглянуто та обґрунтовано вибір інструментів для моделювання бізнес процесів. Застосовані методології аналізу бізнес-процесів з побудовою логічних моделей.

Проаналізовані програмні аналоги, виявлені їх переваги та недоліки для розробки системи. З урахуванням досліджених аналогів, на їх основі побудована специфікація вимог.

За допомогою UML-діаграм була спроектована система, визначено користувачів та їх діяльність в рамках системи. Використовуючи UML-діаграми класів та ERD-діаграми спроектована система, взаємодія між компонентами та створена база даних.

Був розроблений прототип користувацького інтерфейсу, на основі якого був побудований користувацький інтерфейс.

Оглядаючись на специфікацію вимог, були розглянуті та підібрані технології для клієнтської та серверної частини, виявлені їх переваги та недоліки при використанні у розробці системи. Підібрані інструменти для роботи з обраними технологіями. Побудована та відлагоджена система для відстеження ефективності тренувань спортсменів.