

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему СТВОРЕННЯ ПРОГРАМНОЇ СИСТЕМИ
ПОШУКУ ДРУЗІВ ДЛЯ СПІЛЬНИХ ЗАНЯТЬ БІГОМ
DESIGN OF A DATA PROCESSING SOFTWARE SYSTEM
FOR FINDING RUNNING PARTNERS

Виконав(ла): студент(ка) 4 курсу, групи КНТ-222
Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

МЕДВЕДСВ П.Р.

(ПРИЗВИЩЕ та ініціали)

Керівник ШИТІКОВА О.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОРОТКИЙ О.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет КНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 122 Комп'ютерні науки
 (код і найменування)
 Освітня програма (спеціалізація) Комп'ютерні науки
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2026 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

МЕДВЕДСЬКА Платона Романовича
 (ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Створення програмної системи пошуку друзів для спільних занять бігом. Design of a Data Processing Software System for Finding Running Partners

керівник проєкту (роботи) к.т.н., ШИТІКОВА Олена Вікторівна,
 (науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 07 ” квітня 2026 року № 139

2. Строк подання студентом проєкту (роботи) 10 червня 2026 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Проєктування програмної системи. 3. Розробка програмної системи. 4. Експлуатація та тестування програмної системи.

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ШИТІКОВА О.В., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання « 20 » квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 2
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Платон МЕДВЕДЄВ
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Олена ШИТІКОВА
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 110 с., 3 табл., 24 рис., 3 дод., 10 джерел.

БІГ, АКТИВНІСТЬ, СПІЛЬНЕ ЗАНЯТТЯ, ПРОГРАМНА СИСТЕМА, ФРЕЙМВОРК.

Об'єкт роботи – процес побудови програмних систем пошуку друзів для спільних занять бігом. Предмет роботи – програмні системи пошуку друзів для спільних занять бігом. Мета роботи – збільшення кількості релевантних пропозицій під час пошуку друзів для спільних занять бігом за рахунок врахування цікавих любителям бігу місць за допомогою програмної системи.

Матеріали, методи та технічні засоби: набір технологій, що включає Python, Django та PostgreSQL.

Результати. Програмна система побудована як сукупність об'єднаних для виконання завдання пошуку друзів для спільних занять бігом підсистем представлень, моделей, форм та шаблонів. Результати проєктування програмної системи включають виконане моделювання, визначення структури роботи з даними.

Висновки. Створена програмна система призначена для використання її для накопичення власних даних щодо результатів бігових активностей користувачами з одного боку, а з іншого боку для пошуку друзів для спільних занять бігом, планування таких занять бігом на основі визначення основних точок і місць для таких занять, організації віддалених занять стосовно досягнення певного рівня показників під час занять бігом протягом заданого періоду часу.

Галузь використання. Створення платформи для обміну інформацією між любителями бігу щодо власних досягнень та пошуку друзів для спільних занять бігом.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 110 pages, 3 tables, 24 figures, 3 appendixes, 10 sources.

RUNNING, ACTIVITY, JOINT ACTIVITY, SOFTWARE SYSTEM, FRAMEWORK.

The object of the work is the process of building software systems for finding running partners. The subject of the work is software systems for finding running partners. The purpose of the work is to increase the number of relevant offers when searching for running partners by taking into account places of interest to running enthusiasts using the software system.

Materials, methods and technical means: a set of technologies that includes Python, Django and PostgreSQL.

Results. The software system is built as a set of subsystems of views, models, forms and templates combined to perform the task of running partners. The results of the software system design include the performed modeling, determination of the structure of working with data.

Conclusions. The created software system is intended for use to accumulate its own data on the results of running activities by users on the one hand, and on the other hand, to search for running partners, planning such running activities based on determining the main points and places for such activities, organizing remote activities in order to achieve a certain level of indicators during running activities within a given period of time.

Field of use. Creating a platform for exchanging information between running enthusiasts on their own achievements and searching for running partners.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Огляд існуючих програм для забезпечення пошуку друзів для спільних занять бігом	10
1.2 Порівняльний аналіз програм для забезпечення пошуку друзів для спільних занять бігом.....	13
1.3 Висновки за розділом 1	14
2 Проєктування програмної системи	15
2.1 Вибір засобів проєктування і розробки	15
2.2 Моделювання роботи з програмною системою пошуку друзів для спільних занять бігом.....	16
2.3 Визначення структури бази даних програмної системи пошуку друзів для спільних занять бігом	21
2.4 Висновки за розділом 2	37
3 Розробка програмної системи.....	38
3.1 Функціональна схема програмної системи пошуку друзів для спільних занять бігом	38
3.2 Рішення з реалізації програмної системи пошуку друзів для спільних занять бігом	39
3.3 Висновки за розділом 3	62
4 Експлуатація та тестування програмної системи	63
4.1 Призначення програмної системи.....	63
4.2 Умови виконання програмної системи.....	63
4.3 Експлуатація програмної системи для пошуку друзів для спільних занять бігом	63
4.4 Тестування програмної системи для пошуку друзів для спільних занять бігом	67

4.5 Висновки за розділом 4	71
Висновки	72
Перелік джерел посилення	73
Додаток А Технічне завдання	74
Додаток Б Текст програми	80
Додаток В Слайди презентації.....	104

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CRUD – Create – Read – Update – Delete;

БД – база даних;

ПС – програмна система.

ВСТУП

Заняття бігом є доволі популярними в Україні. Частина людей займається самотійно, а частина потребує партнерів. Відповідно частина людей, які займаються бігом знаходяться в пошуку друзів для спільних занять бігом. А ще частина людей не займається бігом через те, що не має партнерів. Тобто такі люди потребують підтримки. З іншого боку процес пошуку друзів полягає не у встановленні контакту з випадковим любителем бігу, а в тому, щоб врахувати рівень такого любителя бігу стосовно бігових активностей та врахувати інтереси. Перший фактор виражається в тому, що в програмній системі, яка використовується для підтримки пошуку друзів для спільних занять бігом необхідно забезпечити накопичення даних про минулі бігові активності. Другий фактор виражається в тому, що люди схильні бігати в різних цікавих їм місцях. Для когось це може бути стадіон, для когось парк, а для когось тривалий маршрут з різними точками на ньому, кожна з яких несе певну цікавість.

У роботі необхідно спочатку дослідити стан існуючих програмних рішень для пошуку друзів для спільних занять бігом стосовно цих двох факторів у першу чергу, а тоді визначити вимоги до програмної системи, що створюється.

Мета роботи полягає в збільшенні кількості релевантних пропозицій під час пошуку друзів для спільних занять бігом за рахунок врахування цікавих любителям бігу місць за допомогою програмної системи (ПС). Завдання роботи:

- аналіз проблеми розробки ПС пошуку друзів для спільних занять бігом;
- проєктування ПС пошуку друзів для спільних занять бігом;
- розробка ПС пошуку друзів для спільних занять бігом.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд існуючих програм для забезпечення пошуку друзів для спільних занять бігом

Існуючі програми, які організовують підтримку занять бігом, можна розподілити на 2 групи. До першої групи належить ті програми, які дозволяють концентруватися на проведенні офіційних змагань, зокрема марафонів, півмарафонів. Такі програми дозволяють шукати самі змагання, переглядати результати участі любителів бігу в таких змаганнях. Прикладом такого рішення є Timeto [1] (рис. 1.1). Таке рішення не дозволяє встановлювати знайомства між любителями бігу, можна тільки отримати інформацію про їх результати.

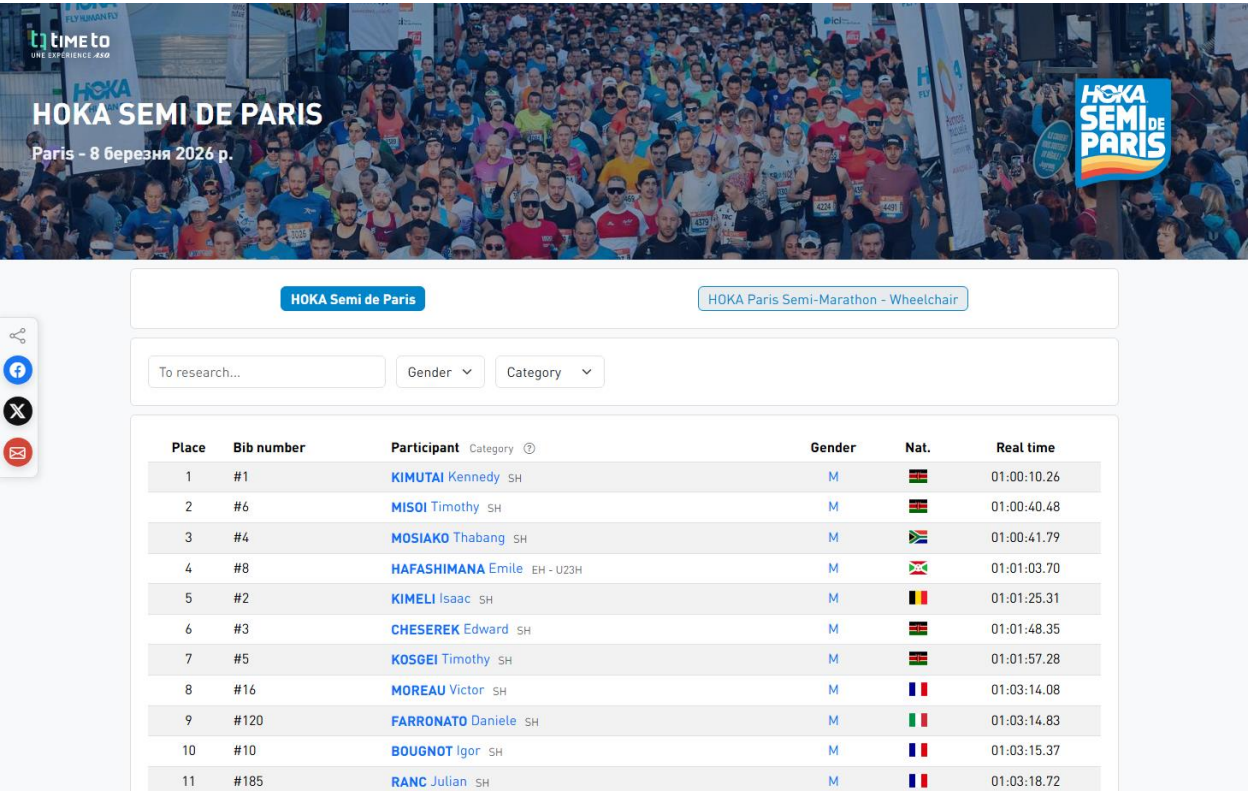


Рисунок 1.1 – Програма Timeto

За подібною логікою з ширшими інструментами пошуку організована програма FindARace [2] (рис. 1.2).

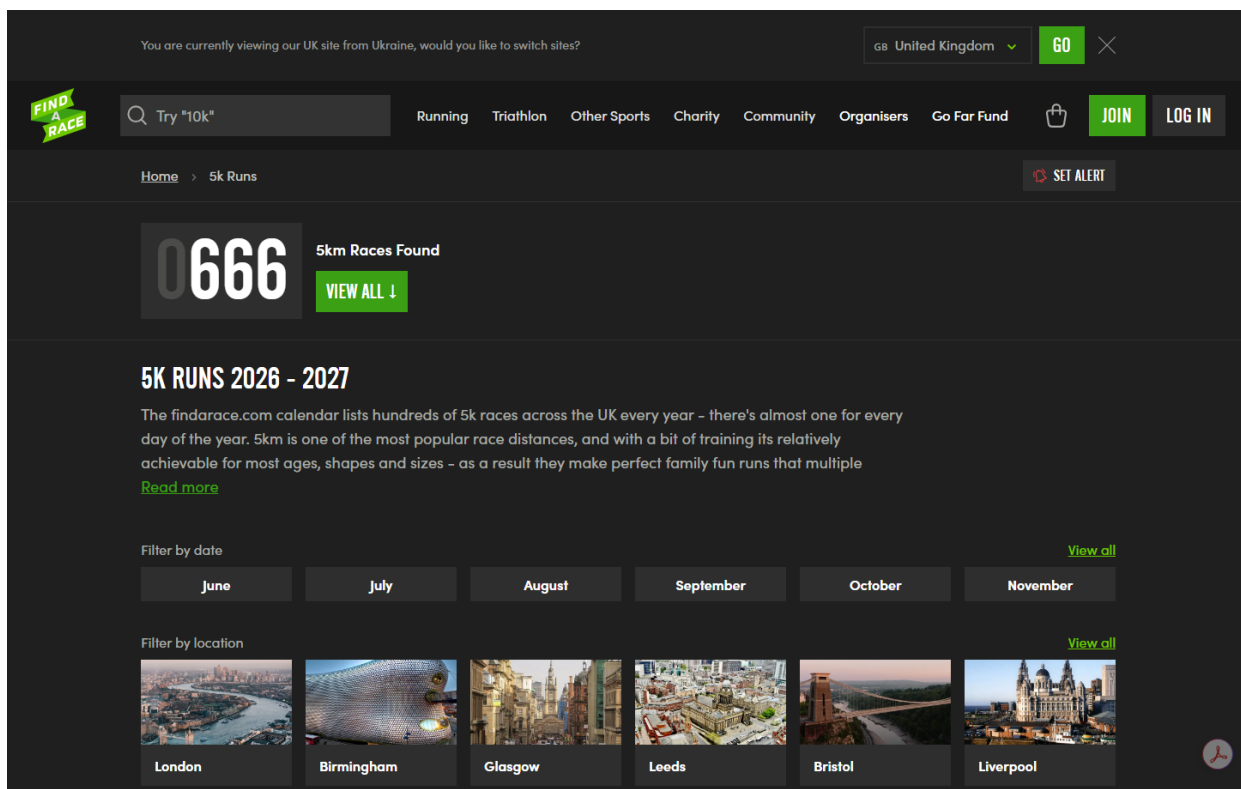


Рисунок 1.2 – Програма FindARace

Вебсайт асоціації International Trail Running Association [3] (рис. 1.3) має свої особливості, пов'язані з тим, що в основі тут вже лежить сам любитель бігу.

Відповідно вебсайт дозволяє шукати спортсменів за конкретними прізвищами, можна переглядати результати виступів спортсмена на змаганнях (рис. 1.4). Таким чином можна розширювати власну інформацію, проте в такому випадку мова йде саме про спортсменів, відповідно звичайних любителів бігу знайти на такому вебсайті не вдасться. Хоча якщо любитель бігу бере участь в змаганнях, то він також може бути представлений у результатах. Доступна статистика щодо виступів спортсменів є розширеною, можна також порівнювати спортсменів.

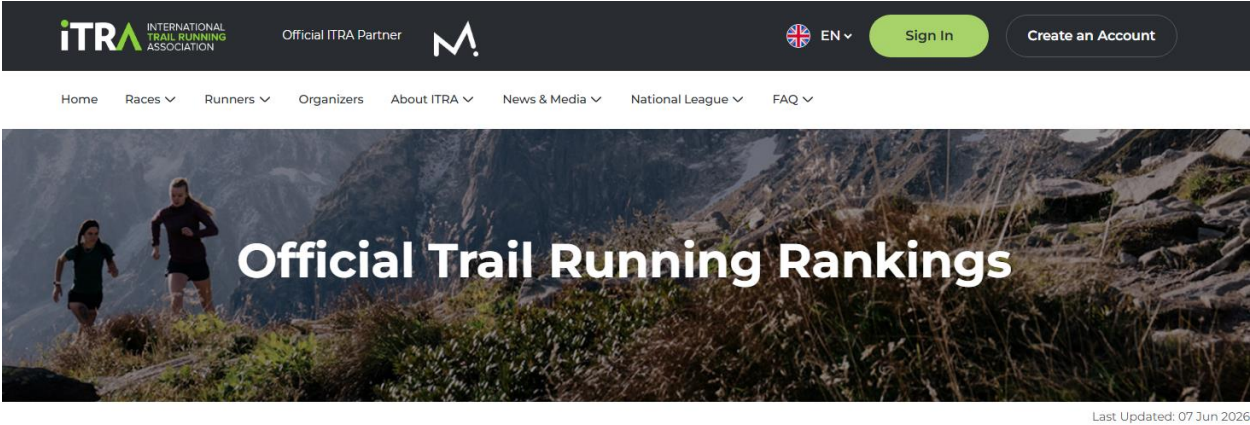


Рисунок 1.3 – Вебсайт ITRA

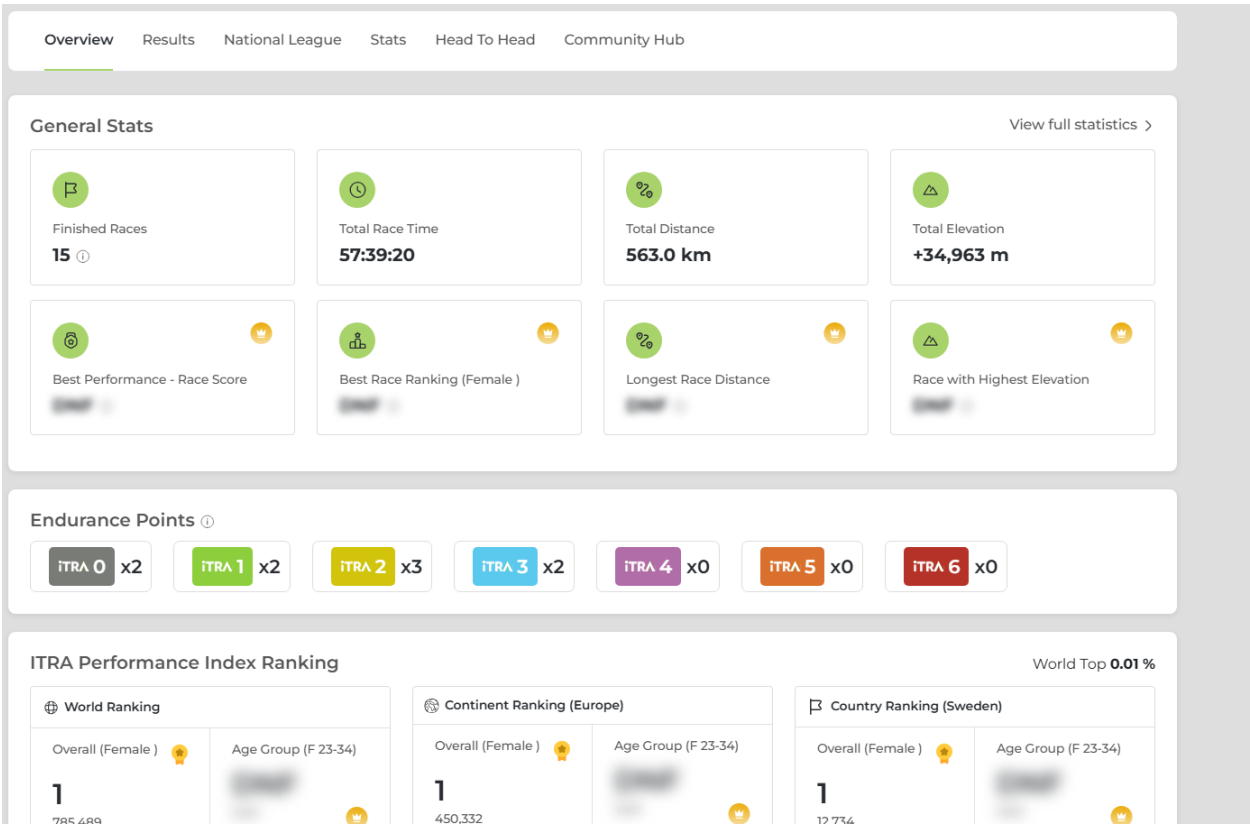


Рисунок 1.4 – Відображення результатів спортсмену на вебсайті ITRA

Другою групою рішень є ті, де навпаки аналізуються дані звичайних користувачів, які не є професійними спортсменами, відповідно вони дозволяють отримати багато статистики як, наприклад, програма Runalyze [4] (рис. 1.5).

<

01/26/2026

02/01/2026

>

Week 5, 2026

▼

🔍 Search

🔄 Automatic Sync

➕ Add workout

	Ty...	Distan...	Durati...	Pace	Pace	avg...	HR	Elev.	CS	Cadence	Grou...	Ver...	GCB	Title	T...	V...	PC	TE	PE	Req...
26.01 🇨🇦 Mon																				
27.01 🇨🇦 Tue	🏃 IT	15.1 km	1:10:57	4:42/km		70 %		96 m	0.0	186 spm	222 ms	7.7 %	48.1L/51.9R %	8x1k in 4:10, 500...	93	📈	-4/+4	3.8	36.8	43.63
28.01 🇨🇦 Wed	🏃 DL	10.0 km	54:21	5:25/km		62 %		141 m	1.9	180 spm	233 ms	8.8 %	48.5L/51.5R %		47	📈	-4/-2	2.6	14.3	34.07
29.01 🇨🇦 Thu																				
30.01 🇨🇦 Fri	🏃 RL	4.3 km	21:27	5:01/km		67 %		38 m	0.3	182 spm	224 ms	8.7 %	48.4L/51.7R %	4x200m in 4:10/km	25	📈	-3/-1	2.5	20.7	33.00
🇨🇦 31.01 🇨🇦 Sat	🏃 WK	50.0 km	4:06:45	4:56/km		74 %		68 m	0.0	186 spm	217 ms	7.8 %	48.8L/51.2R %	Rodgau Ultramarat...	383	📉	±0/-5	5.0	58.9	46.44
01.02 🇨🇦 Sun																				
av. Laufen		84.7 km	7:05:00	5:01/km		71 %		349 m		184 spm	222 ms	8.0 %	48.6L/51.4R %		579	📉			44.3	42.27

Рисунок 1.5 – Програма Runalyze [4]

При визначенні структури опису даних, які використовуються пристроями за біговими активностями, було використано популярний формат таких даних FIT [5].

1.2 Порівняльний аналіз програм для забезпечення пошуку друзів для спільних занять бігом

Для порівняльного аналізу програм для забезпечення пошуку друзів для спільних занять бігом виділено вебсайт ITRA та програму Runalyze. Альтернативою є розробка ПС цієї роботи (табл. 1.1). Розробка відрізняється від аналогів тим, що вона дозволяє фактично виконувати пошук друзів для спільних занять бігом, оскільки фактично аналоги на це не орієнтовані. Вони дозволяють збирати статистику, представляють її певним чином, що сприяє інформуванню користувачів, може бути підґрунтям для створення контакту, але безпосередньо за організацію таких контактів такі рішення не відповідають.

Таблиця 1.1 – Порівняння програм для забезпечення пошуку друзів для спільних занять бігом

Можливість	ITRA	Runanalyze	Розробка
Історія офіційних змагань з бігу	+	+	+
Історія звичайних бігових активностей	—	+	+
Пошук користувачів	Спортсмени	—	Любителі бігу
Визначення улюблених місць для бігу	—	—	+
Створення спільного заняття з бігу	—	—	+
Створення змагання за показниками	—	—	+

1.3 Висновки за розділом 1

Результатом аналізу предметної області розробки ПС пошуку друзів для спільних занять бігом є виділені особливості розробки порівняно з розглянутим набором аналогів.

2 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

2.1 Вибір засобів проєктування і розробки

Розробку ПС пошуку друзів для спільних занять бігом вирішено виконувати мовою Python, щоб сприяти швидкості розробки та мати можливість універсального доступу з різних типів пристроїв за рахунок звернення до серверу вебсистеми. Оскільки Python має ряд фреймворків для веброзробки, то було виконано порівняння двох з них: Django [6]-[7] та FastAPI [8]-[9] (табл. 2.1). Рішення було прийнято на користь Django, знову ж таки щоб сприяти швидкості розробки за рахунок монолітного принципу організації, який може мати свої переваги і недоліки порівняно з модульним, але в даному випадку за рахунок наявності вбудованих засобів, потрібних для реалізації цієї ПС, сприяє тому, щоб час розробки скоротився, адже не вимагає вибору рішення з альтернатив, встановлення його, налагодження, тестування роботи, що пов'язано з додатковими витратами часу.

Таблиця 2.1 – Порівняння фреймворків для забезпечення розробки ПС пошуку друзів для спільних занять бігом

Критерій	Django	FastAPI
Тип побудови	Монолітний	Модульний
Забезпечення роботи з даними	Об'єктно-реляційне відображення для взаємодії з базою даних вбудовано	Можливе для взаємодії з базою даних, але не за замовчуванням
Наявність адміністративної панелі	Вбудована	Не за замовчуванням
Сценарій використання	Великі, середні застосунки	Мікросервіси

2.2 Моделювання роботи з програмною системою пошуку друзів для спільних занять бігом

При моделюванні роботи з ПС пошуку друзів для спільних занять бігом вирішено було не виділяти ролі користувачів. Тобто всі користувачі працюють в одній і тій самій ролі, маючи доступ до всієї широти функціональних можливостей. При тому це означає, що вони потребують реєстрації та авторизації для роботи з ПС.

Окремою роллю є адміністратор, однак це особлива роль, яка фактично має у ПС пошуку друзів для спільних занять бігом окрему підсистему для взаємодії. Це означає, що інтерфейс адміністративної панелі створюється окремо засобами Django, відповідно сценарії такої роботи є типовими та передбачають виконання сценаріїв Create – Read – Update – Delete (CRUD). CRUD-сценарії є типовими та відповідають всім сутностям даних, які будуть описані далі.

Основний потік сценаріїв користувача для роботи з ПС пошуку друзів для спільних занять бігом включає:

- реєстрація власного пристрою для фіксації занять бігом;
- відображення профілю користувача, який займається бігом;
- пошук серед користувачів, які займаються бігом;
- авторизація;
- відображення власних пристроїв для фіксації занять бігом;
- відображення деталізованої статистики щодо занять бігом користувача за рік;
- відображення активностей з занять бігом користувача;
- відображення деталізованої активності з занять бігом користувача;
- додавання виконаної активності з занять бігом користувача;
- додавання запису фіксації активності з занять бігом користувача;
- зв'язування виконаної активності з занять бігом користувача з результатом офіційного змагання;

- додавання результату участі користувача в офіційному змаганні з бігу;
- відображення результатів офіційного змагання з бігу за рік проведення;
- відображення даних офіційного змагання з бігу;
- підписування на користувача, який займається бігом;
- відображення користувачів, які займаються бігом та на яких підписався даний;
- відображення користувачів, які займаються бігом та які підписалися на даного;
- підтвердження дружби у відповідь на підписку від користувача, який займається бігом;
- виділення точки, яку під час заняття бігом долають користувачі;
- пошук точок, які під час занять бігом долають користувачі;
- пошук місць, які під час занять бігом долають користувачі;
- відображення даних точки, яку під час занять бігом долають користувачі;
- відображення даних місця, яке під час занять бігом долають користувачі;
- створення відгуку про точку, яку під час занять бігом долають користувачі;
- створення відгуку про місце, яке під час занять бігом долають користувачі;
- відображення відгуків за точкою, яку під час занять бігом долають користувачі;
- відображення відгуків за місцем, яке під час занять бігом долають користувачі;
- відображення відгуків про точки, які під час занять бігом долають користувачі та які створив цей користувач;
- відображення відгуків про місце, які під час занять бігом долають

користувачі та які створив цей користувач;

- відмічування місця, яке під час занять бігом долають користувачі;
 - відмічування точки, яку під час занять бігом долають користувачі;
 - пошук офіційних змагань з бігу;
 - надсилання запиту за спільним заняттям з бігу;
 - підтвердження запиту за спільним заняттям з бігу;
 - відображення надісланих запитів за спільними заняттями з бігу;
 - відображення отриманих запитів за спільними заняттями з бігу;
 - відображення визначених спільних занять з бігу;
 - пропонування змагання для занять бігом за показниками;
 - надсилання запиту на участь у змаганні для занять бігом за показниками;
 - підтвердження запиту на участь у змаганні для занять бігом за показниками;
 - відображення змагань для занять бігом за показниками, в яких бере участь користувач;
 - відображення отриманих запитів на участь у змаганні для занять бігом за показниками;
 - відображення надісланих запитів на участь у змаганні для занять бігом за показниками;
 - відображення деталізованого заняття бігом за показниками;
 - відображення деталізованого спільного заняття з бігу;
 - відображення змагань для занять бігом за показниками за користувачами з підписок;
 - відображення відмічених користувачем точок;
 - відображення відмічених користувачем місць;
 - відображення активностей з бігу за користувачами з підписок
- (рис. 2.1).

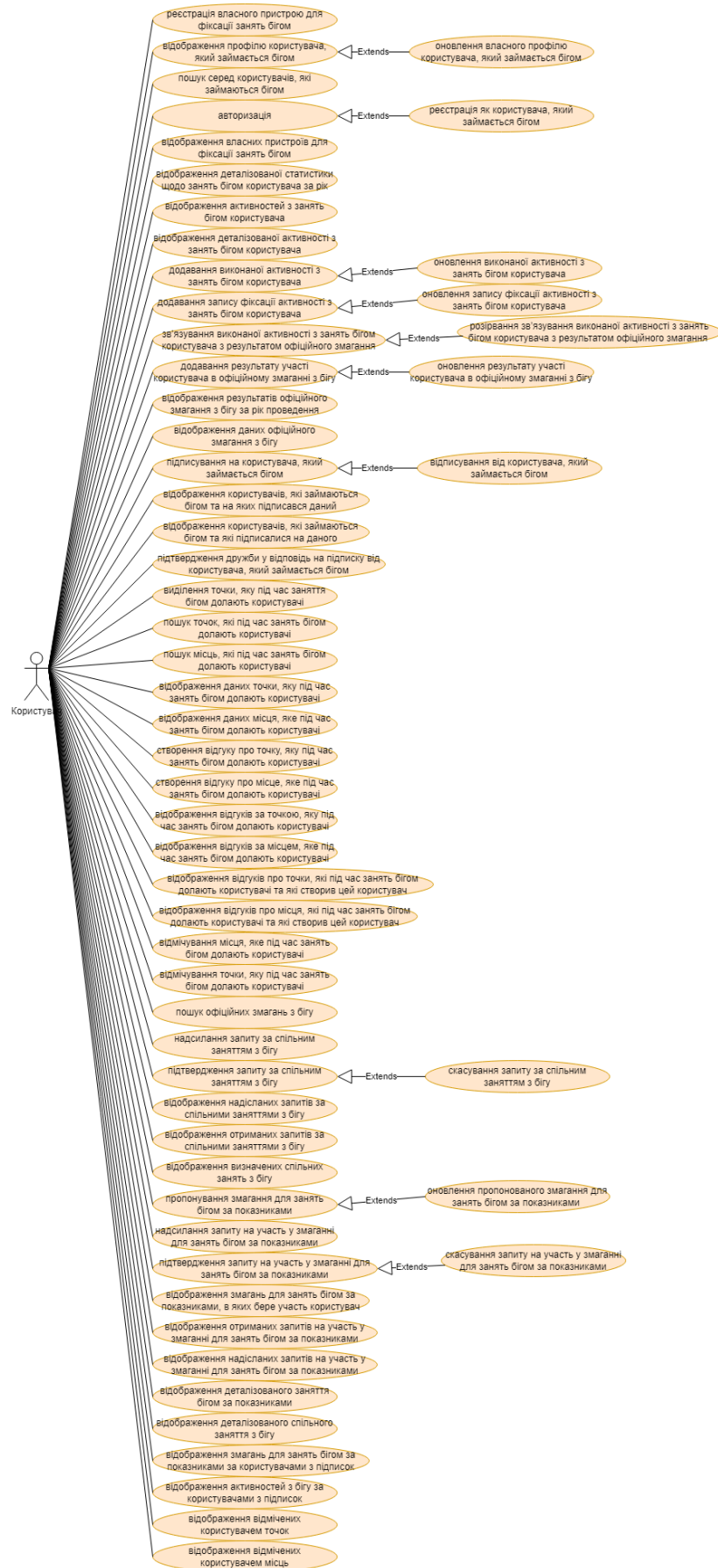


Рисунок 2.1 – Діаграма use case для користувача ПК пошуку друзів для спільних занять бігом

Альтернативні сценарії користувача для роботи з ПС пошуку друзів для спільних занять бігом:

- реєстрація як користувача, який займається бігом, альтернативно для сценарію авторизації, що необхідно тоді, коли користувач, який займається бігом, не може авторизуватись, тому що не має акаунту в програмі або має, але потребує використання іншого;

- скасування запиту за спільним заняттям з бігу альтернативно для сценарію підтвердження запиту за спільним заняттям з бігу, що необхідно тоді, коли запит за спільним заняттям з бігу поступив користувачу, але він не має бажання його виконувати, або тоді коли, користувач сам подав запит іншому користувачу, але вже не бажає його виконувати;

- оновлення пропонованого змагання для занять бігом за показниками альтернативно для сценарію пропонування змагання для занять бігом за показниками, що необхідно тоді, коли пропонування змагання для занять бігом за показниками виконане, але необхідно уточнити цю пропозицію;

- скасування запиту на участь у змаганні для занять бігом за показниками альтернативно для сценарію підтвердження запиту на участь у змаганні для занять бігом за показниками, що необхідно тоді, коли запит на участь у змаганні для занять бігом за показниками поступив користувачу, але він не має бажання його виконувати, або тоді коли, користувач сам подав запит іншому користувачу, але вже не бажає його виконувати;

- відписування від користувача, який займається бігом, альтернативно для сценарію підписування на користувача, який займається бігом, що необхідно тоді, коли таке підписування цим користувачем на іншого користувача виконане, але більше не є активним;

- оновлення результату участі користувача в офіційному змаганні з бігу альтернативно для сценарію додавання результату участі користувача в офіційному змаганні з бігу, що необхідно тоді, коли таке додавання результату участі користувача в офіційному змаганні з бігу вже виконано, але потрібно оновити внесені дані через деякі неточності;

- розірвання зв'язування виконаної активності з занять бігом користувача з результатом офіційного змагання альтернативно для сценарію зв'язування виконаної активності з занять бігом користувача з результатом офіційного змагання, що необхідно тоді, коли зв'язування виконаної активності з занять бігом користувача з результатом офіційного змагання виконане, але це було помилкою;
- оновлення виконаної активності з занять бігом користувача альтернативно для сценарію додавання виконаної активності з занять бігом користувача, що необхідно тоді, коли ця виконана активність з занять бігом цього користувача додана, але містить якісь неточності;
- оновлення запису фіксації активності з занять бігом користувача альтернативно для сценарію додавання запису фіксації активності з занять бігом користувача, що необхідно тоді, коли запис фіксації цієї активності з занять бігом цього користувача виконано, але він містить якісь неточності;
- оновлення власного профілю користувача, який займається бігом, альтернативно для сценарію відображення профілю користувача, який займається бігом, що необхідно тоді, коли профіль користувача, який займається бігом, має якісь неточності на поточний момент.

2.3 Визначення структури бази даних програмної системи пошуку друзів для спільних занять бігом

База даних (БД) була створена з врахуванням тих даних, які необхідні для забезпечення всіх сценаріїв роботи, які були окреслені у розділі 2, логікою роботи з системою PostgreSQL та описана нижче.

У складі БД таблиця Runningdevicemanufacturer відповідає за збереження даних виробників пристроїв, які фіксують активності за заняттями бігом, на основі виділення таких стовпців таблиці з наступними характеристиками:

- `id` – ідентифікатор виробника пристрою, який здатен фіксувати активності за заняттями бігом;

- `runningdevicemanufacturername` – назва виробника пристрою, який здатен фіксувати активності за заняттями бігом.

У складі БД таблиця `Runningdevicetype` відповідає за збереження даних типів пристроїв, які здатні фіксувати активності за заняттями бігом, на основі виділення таких стовпців таблиці з наступними характеристиками:

- `id` – ідентифікатор типу пристрою, який здатен фіксувати активності за заняттями бігом;

- `runningdevicetypename` – назва типу пристрою, який здатен фіксувати активності за заняттями бігом.

У складі БД таблиця `Runningdevice` відповідає за збереження даних пристроїв, які здатні фіксувати активності за заняттями бігом, на основі виділення таких стовпців таблиці з наступними характеристиками:

- `id` – ідентифікатор пристрою, здатного фіксувати активності за заняттями бігом;

- `runningdevicemanufacturer` – виробник цього пристрою, здатного фіксувати активності за заняттями бігом, встановлений зв'язком з таблицею `Runningdevicemanufacturer` типу один-до-багатьох;

- `runningdevicetype` – тип цього пристрою, здатного фіксувати активності за заняттями бігом, встановлений зв'язком з таблицею `Runningdevicetype` типу один-до-багатьох;

- `runningdevicename` – назва або модель пристрою, здатного фіксувати активності за заняттями бігом.

У складі БД таблиця `Runningpartner` відповідає за збереження даних користувачів, які займаються бігом, на основі виділення таких стовпців таблиці з наступними характеристиками:

- `id` – ідентифікатор користувача, який займається бігом;

- `runningpartneraccount` – акаунт користувача, який займається бігом, в Django, встановлений зв'язком з таблицею `User` типу один-до-одного;

- runningpartnerphoto – фотографія користувача, який займається бігом;
- runningpartnerbirthdate – дата народження користувача, який займається бігом;
- runningpartnernationality – національність користувача, який займається бігом, встановлений зв'язком з таблицею Runningcountry типу один-до-багатьох.

У складі БД таблиця Runningpartnerdevice відповідає за збереження даних конкретних пристроїв, які фіксують заняття бігом у конкретних користувачів, на основі виділення таких стовпців таблиці з наступними характеристиками:

- id – ідентифікатор конкретного пристрою, який здатен фіксувати заняття бігом у конкретного користувача, який займається бігом;
- runningpartner – користувач, який займається бігом та який має цей пристрій для фіксації занять, встановлений зв'язком з таблицею Runningpartner типу один-до-багатьох;
- runningdevice – пристрій, який використовується цим користувачем, який займається бігом, для фіксації занять бігом, встановлений зв'язком з таблицею Runningdevice типу один-до-багатьох;
- runningdevicesn – серійний номер конкретного пристрою, який здатен фіксувати заняття бігом у конкретного користувача;
- runningpartnerdevicecreated – час створення запису про наявність цього пристрою, який здатен фіксувати заняття бігом, у цього користувача, який займається бігом.

У складі БД таблиця Runningpartnerstatistic відповідає за збереження даних річної статистики користувача, який займається бігом, на основі виділення таких стовпців таблиці з наступними характеристиками:

- id – ідентифікатор річної статистики користувача, який займається бігом;

– `runningpartner` – користувач, який займається бігом та за яким збережено цю річну статистику, встановлений зв'язком з таблицею `Runningpartner` типу один-до-багатьох;

– `runningpartnerstatisticyear` – рік, якого стосується ця статистика занять бігом за користувачем;

– `runningpartnercompetitions` – кількість офіційних змагань, в яких користувач взяв участь протягом цього року;

– `runningpartnerdistance` – загальна сумарна дистанція в кілометрах, яку в межах офіційних змагань та звичайних занять бігом користувач подолав протягом цього року;

– `runningpartnerlongestrace` – найбільший за довжиною в кілометрах забіг, в якому взяв участь цей користувач протягом року;

– `runningpartneractivedays` – кількість днів цього року, в які користувач займався бігом;

– `runningpartnercompetitiontime` – загальний сумарний час, протягом якого в межах офіційних змагань бігом користувач займався бігом протягом цього року;

– `runningpartnertotaltime` – загальний сумарний час, протягом якого в межах офіційних змагань та звичайних занять бігом користувач займався бігом протягом цього року.

У складі БД таблиця `Runningactivitytype` відповідає за збереження даних типів активностей з бігу на основі виділення таких стовпців таблиці з наступними характеристиками:

– `id` – ідентифікатор типу активності з бігу, яка здатна фіксуватися пристроями та виділена в програмі для такої категоризації;

– `runningactivitytypename` – назва типу активності з бігу, яка здатна фіксуватися пристроями та виділена в програмі для такої категоризації.

У складі БД таблиця `Runningactivity` відповідає за збереження даних конкретних активностей з бігу у користувачів на основі виділення таких стовпців таблиці з наступними характеристиками:

– id – ідентифікатор конкретної активності з бігу в користувача, якою він займався;

– runningactivitytype – тип активності з бігу, якою займався користувач цього разу, встановлений зв'язком з таблицею Runningactivitytype типу один-до-багатьох;

– runningpartnerdevice – пристрій, на який фіксувалась ця активність з занять бігом, встановлений зв'язком з таблицею Runningpartnerdevice типу один-до-багатьох;

– runningactivitystarttime – час початку заняття бігом користувачем за цією активністю;

– runningactivityendtime – час завершення заняття бігом користувачем за цією активністю;

– runningactivitydistance – сумарна дистанція, яку протягом цього заняття бігом зміг подолати цей користувач, в кілометрах;

– runningactivitynote – нотатка від користувача щодо особливостей занять бігом за цією активністю;

– runningactivityheartratemax – максимальна кількість ударів серця на хвилину, зафіксована протягом цієї активності з бігу в цього користувача;

– runningactivityheartrateaverage – середня кількість ударів серця на хвилину, зафіксована протягом цієї активності з бігу в цього користувача;

– runningactivityspeedmax – максимальна швидкість, яку розвинув цей користувач за цією активністю протягом всього заняття.

У складі БД таблиця Runningactivitycompetitionresult відповідає за збереження даних відповідності зафіксованих активностей з занять бігом у користувачів результатам офіційних спортивних змагань, в яких вони також були зафіксовані, на основі виділення таких стовпців таблиці з наступними характеристиками:

– id – ідентифікатор відповідності зафіксованої активності з занять бігом у користувача результату офіційного спортивного змагання, в якому він також був зафіксований;

– `runningactivity` – активність з занять бігом у користувача, якій відповідає цей офіційний результат спортивного змагання, встановлений зв'язком з таблицею `Runningactivity` типу один-до-багатьох;

– `runningcompetitionresult` – офіційний результат спортивного змагання, за яким зафіксовано цю активність з бігу у користувача, встановлений зв'язком з таблицею `Runningcompetitionresult` типу один-до-багатьох.

У складі БД таблиця `Runningrecord` відповідає за збереження даних фіксацій записів активності з занять бігом користувача на основі виділення таких стовпців таблиці з наступними характеристиками:

– `id` – ідентифікатор зафіксованого запису активності з занять бігом у користувача;

– `runningactivity` – активність з занять бігом, за якою було зафіксовано цей запис, встановлена зв'язком з таблицею `Runningactivity` типу один-до-багатьох;

– `runningrecordlatitude` – широта точки, в якій перебував цей користувач під час заняття бігом за цією активністю, коли було зафіксовано цей запис спостереження;

– `runningrecordlongitude` – довгота точки, в якій перебував цей користувач під час заняття бігом за цією активністю, коли було зафіксовано цей запис спостереження;

– `runningrecordaltitude` – висота точки, в якій перебував цей користувач під час заняття бігом за цією активністю, коли було зафіксовано цей запис спостереження;

– `runningrecordlap` – коло, на якому перебував цей користувач під час заняття бігом за цією активністю, коли було зафіксовано цей запис спостереження;

– `runningrecordtime` – записана часова мітка під час заняття бігом за цією активністю, коли було зафіксовано цей запис спостереження;

– `runningrecordheartrate` – кількість ударів серця на хвилину в користувача під час заняття бігом за цією активністю, коли було зафіксовано цей запис спостереження;

– `runningrecorddistance` – кількість кілометрів, подолана користувачем під час цього заняття бігом за цією активністю з моменту останнього запису, коли така активність фіксувалась.

У складі БД таблиця `Runningcompetitiontype` відповідає за збереження даних типів офіційних змагань з бігу на основі виділення таких стовпців таблиці з наступними характеристиками:

– `id` – ідентифікатор типу офіційних змагань з бігу;

– `runningcompetitiontypename` – назва типу офіційних змагань з бігу.

У складі БД таблиця `Runningcountry` відповідає за збереження даних країн, в яких відбуваються змагання з бігу та які за національністю мають бігуни, на основі виділення таких стовпців таблиці з наступними характеристиками:

– `id` – ідентифікатор країни, в якій можуть відбуватися змагання з бігу та яку за національністю можуть мати бігуни;

– `runningcountryname` – назва країни, в якій можуть відбуватися змагання з бігу та яку за національністю можуть мати бігуни.

У складі БД таблиця `Runningcity` відповідає за збереження даних міст, в яких відбуваються змагання з бігу та які за національністю мають бігуни, на основі виділення таких стовпців таблиці з наступними характеристиками:

– `id` – ідентифікатор міста, в якому можуть відбуватися змагання з бігу;

– `runningcitycountry` – країна, в якій знаходиться це місто, в якому можуть відбуватися заняття з бігу, встановлена зв'язком з таблицею `Runningcountry` типу один-до-багатьох;

– `runningcityname` – назва міста, в якому можуть відбуватися змагання з бігу.

У складі БД таблиця Runningcompetition відповідає за збереження даних офіційних змагань з бігу на основі виділення таких стовпців таблиці з наступними характеристиками:

- id – ідентифікатор офіційного змагання з бігу;
- runningcompetitiontype – тип змагання з бігу, якому відповідає це, встановлений зв'язком з таблицею Runningcompetitiontype типу один-до-багатьох;
- runningcompetitioncity – місто, в якому відбувається це офіційне змагання з бігу, встановлене зв'язком з таблицею Runningcity типу один-до-багатьох;
- runningcompetitiontitle – назва цього офіційного змагання з бігу;
- runningcompetitionweb – вебадреса офіційного змагання з бігу;
- runningcompetitiondistance – відстань, на яку розраховане це змагання з бігу;
- runningcompetitionlaps – кількість кіл, яку потрібно подолати під час цього змагання з бігу.

У складі БД таблиця Runningcompetitionyear відповідає за збереження даних конкретних років проведення офіційних змагань з бігу на основі виділення таких стовпців таблиці з наступними характеристиками:

- id – ідентифікатор конкретного року проведення офіційного змагання з бігу;
- runningcompetition – офіційне змагання з бігу, яке проводиться цього року, встановлене зв'язком з таблицею Runningcompetition типу один-до-багатьох;
- runningcompetitiondate – час початку проведення цього офіційного змагання з бігу цього року.

У складі БД таблиця Runningcompetitionresult відповідає за збереження даних результатів участі конкретних бігунів у змаганні певного року на основі виділення таких стовпців таблиці з наступними характеристиками:

- `id` – ідентифікатор результату участі бігуна в змаганні з бігу певного року;
- `runningcompetitionyear` – рік проведення змагання з бігу, в якому було зафіксовано цей результат, встановлений зв'язком з таблицею `Runningcompetitionyear` типу один-до-багатьох;
- `runningpartner` – користувач, який займається бігом і результат якого зафіксовано в цьому записі, встановлений зв'язком з таблицею `Runningpartner` один-до-багатьох;
- `runningcompetitionresultplace` – місце, яке зайняв цей користувач під цього проведення змагання з бігу;
- `runningcompetitionresulttime` – час, за який пробіг дистанцію на цьому змаганні з бігу цей користувач (необхідно зберігати окремо тут, а не за активністю, тому що користувач може мати тільки офіційний результат, а сама активність через ті чи інші причини може бути не зафіксована або користувач взагалі не використовує пристрої);
- `runningcompetitionresultdistance` – відстань, яку подолав за цей час цей користувач під час цього змагання з бігу.

У складі БД таблиця `Runningpartnerfollow` відповідає за збереження даних певної форми дружби між користувачами, які займаються бігом, на основі виділення таких стовпців таблиці з наступними характеристиками:

- `id` – ідентифікатор зафіксованої дружби в певній формі між користувачами;
- `runningpartner` – користувач, який займається бігом та якому був адресований початково створений запис про дружбу в програмі, встановлений зв'язком з таблицею `Runningpartner` один-до-багатьох;
- `runningpartnerfollower` – користувач, який займається бігом та який початково створив цей запис про дружбу в програмі, встановлений зв'язком з таблицею `Runningpartner` один-до-багатьох;

– runningfollowtype – визначення форми дружби між користувачами, які займаються бігом: підписка одного користувача на іншого, взаємне затвердження і таким чином визначення дружби.

У складі БД таблиця Runningpoint відповідає за збереження даних окремо виділених точок, які під час занять бігом долають користувачі, на основі виділення таких стовпців таблиці з наступними характеристиками:

– id – ідентифікатор окремо виділеної точки, яку під час заняття бігом долає користувач;

– runningpointtitle – назва окремо виділеної точки, яку під час заняття бігом долає користувач;

– runningpointlongitude – довгота розміщення окремо виділеної точки, яку під час заняття бігом долає користувач;

– runningpointlatitude – широта розміщення окремо виділеної точки, яку під час заняття бігом долає користувач;

– runningpointaltitude – висота розміщення окремо виділеної точки, яку під час заняття бігом долає користувач;

– runningpointreviewscore – усереднена оцінка, виставлена цій окремо виділеній точці за відгуками бігунів.

У складі БД таблиця Runningplace відповідає за збереження даних місць, які під час занять бігом долають користувачі, на основі виділення таких стовпців таблиці з наступними характеристиками:

– id – ідентифікатор місця, яке під час заняття бігом долають користувачі;

– runningplacetype – тип місця, яке під час заняття бігом долають користувачі, встановлений зв'язком з таблицею Runningplacetype один-до-багатьох;

– runningplacetitle – назва місця, яке під час заняття бігом долають користувачі;

– runningplacelongitude – довгота розташування місця, яке під час заняття бігом долають користувачі;

- `runningplacelatitude` – широта розташування місця, яке під час заняття бігом долають користувачі;

- `runningplacealtitude` – висота розташування місця, яке під час заняття бігом долають користувачі;

- `runningplacereviewscore` – усереднена оцінка, виставлена цьому місцю, яке під час заняття бігом долають користувачі.

Відмінністю місця від точки є те, що місце є певною локацією в місті (парк або деяке місце в ньому, заклад тощо), а точка є певною точкою на мапі, де може бути гарний вид на місто або щось подібне, але де ніякого міського об'єкту не розташовано. Тобто місця є загально визнаними позиціями, а точки створюються самими користувачами, щоб підкреслити свій власний досвід та можливо розділити його з іншими користувачами.

У складі БД таблиця `Runningplacetype` відповідає за збереження даних типів місць, які під час заняття бігом долають користувачі, на основі виділення таких стовпців таблиці з наступними характеристиками:

- `id` – ідентифікатор типу місць, які під час заняття бігом долають користувачі;

- `runningplacetype` – назва типу місць, які під час заняття бігом долають користувачі.

У складі БД таблиця `Runningplacereview` відповідає за збереження даних відгуків про місця, які під час занять бігом долають користувачі, на основі виділення таких стовпців таблиці з наступними характеристиками:

- `id` – ідентифікатор відгуку про місце, яке під час заняття бігом долають користувачі;

- `runningplace` – місце, яке під час заняття бігом долають користувачі і якого стосується відгук, встановлене зв'язком з таблицею `Runningplace` один-до-багатьох;

- `runningpartner` – користувач, який створив цей відгук про місце, встановлений зв'язком з таблицею `Runningpartner` один-до-багатьох;

- runningplacereviewtime – час створення відгуку про місце, яке під час заняття бігом долають користувачі;

- runningplacereviewtext – текст відгуку про місце, яке під час заняття бігом долають користувачі;

- runningplacereviewscore – оцінка у відгуку про місце, яке під час заняття бігом долають користувачі.

У складі БД таблиця Runningplacereviewphoto відповідає за збереження даних фотографії, доданої до відгуку про місце, яке під час заняття бігом долають користувачі, на основі виділення таких стовпців таблиці з наступними характеристиками:

- id – ідентифікатор фотографії, доданої до відгуку про місце, яке під час заняття бігом долають користувачі;

- runningplacereview – відгук про місце, яке під час заняття бігом долають користувачі, до якого додана ця фотографія, встановлений зв'язком з таблицею Runningplacereview один-до-багатьох;

- runningplacereviewphotodata – дані фотографії, доданої до відгуку про місце, яке під час заняття бігом долають користувачі.

У складі БД таблиця Runningpointreview відповідає за збереження даних відгуків про точки, які під час занять бігом долають користувачі, на основі виділення таких стовпців таблиці з наступними характеристиками:

- id – ідентифікатор відгуку про точку, яку під час заняття бігом долають користувачі;

- runningpoint – точка, яку під час заняття бігом долають користувачі і якої стосується відгук, встановлена зв'язком з таблицею Runningpoint один-до-багатьох;

- runningpartner – користувач, який створив цей відгук про точку, встановлений зв'язком з таблицею Runningpartner один-до-багатьох;

- runningpointreviewtime – час створення відгуку про точку, яку під час заняття бігом долають користувачі;

– `runningpointreviewtext` – текст відгуку про точку, яку під час заняття бігом долають користувачі;

– `runningpointreviewscore` – оцінка у відгуку про точку, яку під час заняття бігом долають користувачі.

У складі БД таблиця `Runningpointreviewphoto` відповідає за збереження даних фотографії, доданої до відгуку про точку, яку під час заняття бігом долають користувачі, на основі виділення таких стовпців таблиці з наступними характеристиками:

– `id` – ідентифікатор фотографії, доданої до відгуку про точку, яку під час заняття бігом долають користувачі;

– `runningpointreview` – відгук про точку, яку під час заняття бігом долають користувачі, до якого додана ця фотографія, встановлений зв'язком з таблицею `Runningpointreview` один-до-багатьох;

– `runningpointreviewphotodata` – дані фотографії, доданої до відгуку про точку, яку під час заняття бігом долають користувачі.

У складі БД таблиця `Runningplacesave` відповідає за збереження даних відмічування місць, які під час занять бігом долають користувачі, самими користувачами на основі виділення таких стовпців таблиці з наступними характеристиками:

– `id` – ідентифікатор відмічування місця, яке під час заняття бігом долають користувачі, певним користувачем;

– `runningplace` – місце, яке під час заняття бігом долають користувачі і яке було відзначено цим користувачем, встановлений зв'язком з таблицею `Runningplace` один-до-багатьох;

– `runningpartner` – користувач, який займається бігом і який відзначив це місце, встановлений зв'язком з таблицею `Runningpartner` один-до-багатьох;

– `runningsavetype` – тип відмічування цього місця, яке під час заняття бігом долають користувачі, певним користувачем: публічний або приватний.

Відмічування відрізняється від відгуку тим, що відмічуванням користувач показує собі місце або точку для повернення, тобто щоб

переглянути її знов, або всім надає інформацію про те, що це не просто точка, про яку він хоче вказати свою власну думку або яку він долав, а що це точка, в якій він любить бігати.

У складі БД таблиця Runningpointsave відповідає за збереження даних відмічування точок, які під час занять бігом долають користувачі, самими користувачами на основі виділення таких стовпців таблиці з наступними характеристиками:

- id – ідентифікатор відмічування точки, яку під час заняття бігом долають користувачі, певним користувачем;
- runningpoint – точка, яку під час заняття бігом долають користувачі і яку було відзначено цим користувачем, встановлений зв'язком з таблицею Runningpoint один-до-багатьох;
- runningpartner – користувач, який займається бігом і який відзначив цю точку, встановлений зв'язком з таблицею Runningpartner один-до-багатьох;
- runningsavetype – тип відмічування цієї точки, яку під час заняття бігом долають користувачі, певним користувачем: публічний або приватний.

У складі БД таблиця Runningrequest відповідає за збереження даних запитів на спільне заняття бігом на основі виділення таких стовпців таблиці з наступними характеристиками:

- id – ідентифікатор запиту на спільне заняття бігом;
- runningpartner – користувач, який займається бігом та який створив цей запиті на заняття бігом, встановлений зв'язком з таблицею Runningpartner один-до-багатьох;
- runningpartnerto – користувач, який займається бігом та якому адресований цей запиті на заняття бігом, встановлений зв'язком з таблицею Runningpartner один-до-багатьох;
- runningrequesttitle – назва запиту на спільне заняття бігом;
- runningrequeststarttime – пропонуванний час початку спільного заняття бігом;
- runningrequestnote – примітка до спільного заняття бігом;

– `runningrequeststate` – стан запиту на спільне заняття бігом.

У складі БД таблиця `Runningrequestpoint` відповідає за збереження даних точок, пропонованих для спільного заняття бігом, на основі виділення таких стовпців таблиці з наступними характеристиками:

– `id` – ідентифікатор пропонованої точки для спільного заняття бігом;

– `runningrequest` – запит на спільне заняття бігом, стосовно якого пропонується ця точка, встановлений зв'язком з таблицею `Runningrequest` один-до-багатьох;

– `runningpoint` – точка, пропонується для спільного заняття бігом встановлена зв'язком з таблицею `Runningpoint` один-до-багатьох.

У складі БД таблиця `Runninggoal` відповідає за збереження даних пропонованих змагань для занять бігом за показниками на основі виділення таких стовпців таблиці з наступними характеристиками:

– `id` – ідентифікатор пропонованого індивідуального змагання для занять бігом за показниками;

– `runningpartner` – користувач, який створив це змагання для занять бігом за показниками, встановлений зв'язком з таблицею `Runningpartner` один-до-багатьох;

– `runningperiodtype` – тип періоду змагання для занять бігом за показниками: протягом доби, тижня, місяця, року;

– `runninggoalstarttime` – час початку змагання для занять бігом за показниками;

– `runninggoaldistance` – відстань в кілометрах, яку потрібно подолати під час цього змагання з бігу за показниками;

– `runninggoaltime` – час, протягом якого потрібно займатись бігом у цьому змаганні за показниками;

– `runninggoalperiod` – можливість досягти вказаних показників за декілька разів: наприклад, якщо змагання протягом доби, то можливо потрібно пробігти певну дистанцію за раз або бігти не зупиняючись протягом певного часу (рис. 2.2).

У складі БД таблиця Runninggoalpartner відповідає за збереження даних учасників таких змагань з бігу за показниками на основі виділення таких стовпців таблиці з наступними характеристиками:

- id – ідентифікатор внесення учасника до змагання з бігу за показниками;
- runninggoal – заняття з бігу за показниками, до якого відноситься ця участь, встановлене зв'язком з таблицею Runninggoal один-до-багатьох;

– `runningpartner` – користувач, який займається бігом і який входить до цього змагання з бігу за показниками або якому запропоновано увійти до такого змагання, встановлений зв'язком з таблицею `Runningpartner` один-до-багатьох;

– `runninggoalpartnerstate` – стан внесення учасника до змагання з бігу за показниками: запропоновано, підтверджено, скасовано;

– `runninggoaldistance` – відстань, яку подолав цей користувач як учасник цього змагання з бігу за показниками;

– `runninggoaltime` – час, протягом якого займався бігом цей користувач як учасник цього змагання з бігу за показниками.

2.4 Висновки за розділом 2

Проектування ПС пошуку друзів для спільних занять бігом завершено з отриманням результатів, які включають вибір технологій, до яких в першу чергу відносяться мова Python і фреймворк Django, моделювання роботи користувачів з ПС в частині визначення сценаріїв роботи за функціональними вимогами, визначення структури БД.

3 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

3.1 Функціональна схема програмної системи пошуку друзів для спільних занять бігом

Функціональна схема ПС пошуку друзів для спільних занять бігом побудована таким чином, щоб процес розпочинався з авторизації, до якої користувач не може виконувати ніяких дій. Після авторизації йому відкривається сторінка його власного профілю та можливість вибрати один з доступних розділів меню для роботи (рис. 3.1).

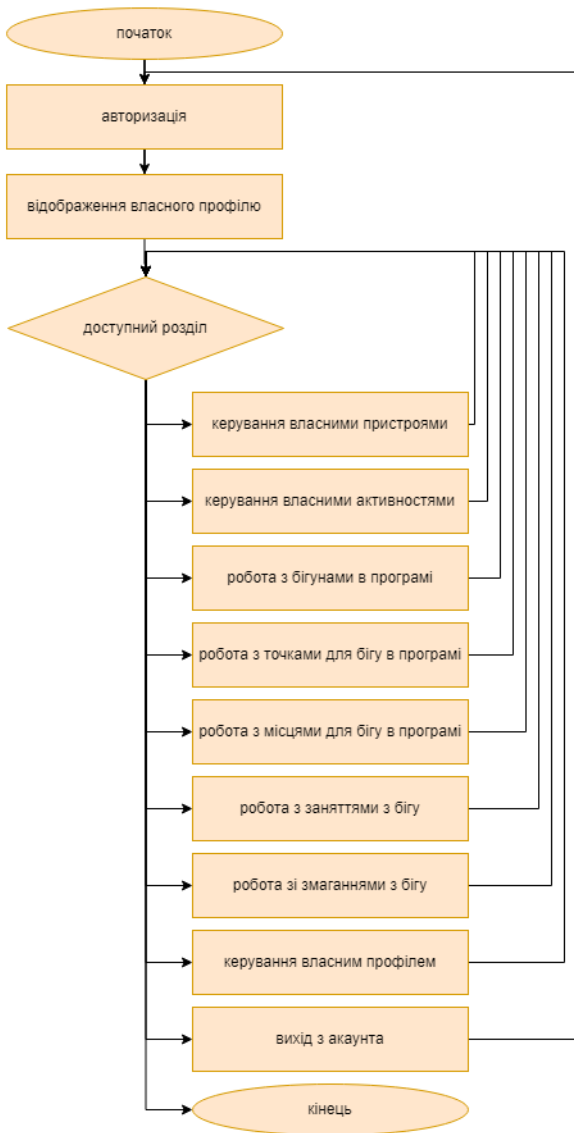


Рисунок 3.1 – Функціональна схема ПС пошуку друзів для спільних занять бігом

3.2 Рішення з реалізації програмної системи пошуку друзів для спільних занять бігом

Набір рішень з реалізації ПС пошуку друзів для спільних занять бігом базується в першу чергу на виборі фреймворку Django для основи розробки, оскільки він визначає структуру системи, формуючи підсистеми. До основних підсистем належать підсистема моделей ПС, підсистема представлень, підсистема шаблонів.

У підсистемі моделей ПС клас `Runningdevicemanufacturer` відповідає за збереження даних виробників пристроїв, які фіксують активності за заняттями бігом, маючи тільки поле `runningdevicemanufacturername`, що має тип `CharField` і яким задана назва виробника пристрою, який здатен фіксувати активності за заняттями бігом.

У підсистемі моделей ПС клас `Runningdevicetype` відповідає за збереження даних типів пристроїв, які здатні фіксувати активності за заняттями бігом, маючи тільки поле `runningdevicetypename`, що має тип `CharField` і яким задана назва типу пристрою, який здатен фіксувати активності за заняттями бігом.

У підсистемі моделей ПС клас `Runningdevice` відповідає за збереження даних пристроїв, які здатні фіксувати активності за заняттями бігом, маючи такі поля.

Поле `runningdevicemanufacturer` має тип `ForeignKey` і ним встановлений виробник цього пристрою, здатного фіксувати активності за заняттями бігом, шляхом зв'язування з моделлю `Runningdevicemanufacturer`.

Поле `runningdevicetype` має тип `ForeignKey` і ним встановлений тип цього пристрою, здатного фіксувати активності за заняттями бігом, , шляхом зв'язування з моделлю `Runningdevicetype`.

Поле `runningdevicename` має тип `CharField` і ним встановлена назва або модель пристрою, здатного фіксувати активності за заняттями бігом.

У підсистемі моделей ПС клас Runningpartner відповідає за збереження даних користувачів, які займаються бігом, маючи такі поля.

Поле runningpartneraccount має тип OneToOneField і ним встановлений акаунт користувача, який займається бігом, в Django, шляхом зв'язування з моделлю User.

Поле runningpartnerphoto має тип ImageField і ним встановлена фотографія користувача, який займається бігом.

Поле runningpartnerbirthdate має тип DateField і ним встановлена дата народження користувача, який займається бігом.

Поле runningpartnernationality має тип ForeignKey і ним встановлена національність користувача, який займається бігом, шляхом зв'язування з моделлю Runningcountry.

У підсистемі моделей ПС клас Runningpartnerdevice відповідає за збереження даних конкретних пристроїв, які фіксують заняття бігом у конкретних користувачів, маючи такі поля.

Поле runningpartner має тип ForeignKey і ним встановлений користувач, який займається бігом та який має цей пристрій для фіксації занять, шляхом зв'язування з моделлю Runningpartner.

Поле runningdevice має тип ForeignKey і ним встановлений пристрій, який використовується цим користувачем, який займається бігом, для фіксації занять бігом, шляхом зв'язування з моделлю Runningdevice.

Поле runningdevicesn має тип CharField і ним встановлений серійний номер конкретного пристрою, який здатен фіксувати заняття бігом у конкретного користувача.

Поле runningpartnerdevicecreated має тип DateTimeField і ним встановлений час створення запису про наявність цього пристрою, який здатен фіксувати заняття бігом, у цього користувача, який займається бігом.

У підсистемі моделей ПС клас Runningpartnerstatistic відповідає за збереження даних річної статистики користувача, який займається бігом, маючи такі поля.

Поле `runningpartner` має тип `ForeignKey` і ним встановлений користувач, який займається бігом та за яким збережено цю річну статистику, шляхом зв'язування з моделлю `Runningpartner`.

Поле `runningpartnerstatisticyear` має тип `PositiveSmallIntegerField` і ним встановлений рік, якого стосується ця статистика занять бігом за користувачем.

Поле `runningpartnercompetitions` має тип `PositiveSmallIntegerField` і ним встановлена кількість офіційних змагань, в яких користувач взяв участь протягом цього року.

Поле `runningpartnerdistance` має тип `DecimalField` і ним встановлена загальна сумарна дистанція в кілометрах, яку в межах офіційних змагань та звичайних занять бігом користувач подолав протягом цього року.

Поле `runningpartnerlongestrace` має тип `DecimalField` і ним встановлений найбільший за довжиною в кілометрах забіг, в якому взяв участь цей користувач протягом року.

Поле `runningpartneractivedays` має тип `PositiveSmallIntegerField` і ним встановлена кількість днів цього року, в які користувач займався бігом.

Поле `runningpartnercompetitiontime` має тип `TimeField` і ним встановлений загальний сумарний час, протягом якого в межах офіційних змагань бігом користувач займався бігом протягом цього року.

Поле `runningpartnertotaltime` має тип `TimeField` і ним встановлений загальний сумарний час, протягом якого в межах офіційних змагань та звичайних занять бігом користувач займався бігом протягом цього року.

У підсистемі моделей ПС клас `Runningactivitytype` відповідає за збереження даних типів активностей з бігу, маючи тільки поле `runningactivitytypename`, що має тип `CharField` і яким задана назва типу активності з бігу, яка здатна фіксуватися пристроями та виділена в програмі для такої категоризації.

У підсистемі моделей ПС клас `Runningactivity` відповідає за збереження даних конкретних активностей з бігу у користувачів, маючи такі поля.

Поле `runningactivitytype` має тип `ForeignKey` і ним встановлений тип активності з бігу, якою займався користувач цього разу, шляхом зв'язування з моделлю `Runningactivitytype`.

Поле `runningpartnerdevice` має тип `ForeignKey` і ним встановлений пристрій, на який фіксувалась ця активність з занять бігом, шляхом зв'язування з моделлю `Runningpartnerdevice`.

Поле `runningactivitystarttime` має тип `DateTimeField` і ним встановлений час початку заняття бігом користувачем за цією активністю.

Поле `runningactivityendtime` має тип `DateTimeField` і ним встановлений час завершення заняття бігом користувачем за цією активністю.

Поле `runningactivitydistance` має тип `DecimalField` і ним встановлена сумарна дистанція, яку протягом цього заняття бігом зміг подолати цей користувач, в кілометрах.

Поле `runningactivitynote` має тип `CharField` і ним встановлена нотатка від користувача щодо особливостей занять бігом за цією активністю.

Поле `runningactivityheartratemax` має тип `PositiveSmallIntegerField` і ним встановлена максимальна кількість ударів серця на хвилину, зафіксована протягом цієї активності з бігу в цього користувача.

Поле `runningactivityheartrateaverage` має тип `PositiveSmallIntegerField` і ним встановлена середня кількість ударів серця на хвилину, зафіксована протягом цієї активності з бігу в цього користувача.

Поле `runningactivityspeedmax` має тип `DecimalField` і ним встановлена максимальна швидкість, яку розвинув цей користувач за цією активністю протягом всього заняття.

Поле `runningcompetitionresult` має тип `OneToOneField` і ним встановлений офіційний результат спортивного змагання, за яким зафіксовано цю активність з бігу у користувача, шляхом зв'язування з моделлю `Runningcompetitionresult`.

У підсистемі моделей ПС клас Runningrecord відповідає за збереження даних фіксацій записів активності з занять бігом користувача маючи, такі поля.

Поле runningactivity має тип ForeignKey і ним встановлена активність з занять бігом, за якою було зафіксовано цей запис, шляхом зв'язування з моделлю Runningactivity.

Поле runningrecordlatitude має тип DecimalField і ним встановлена широта точки, в якій перебував цей користувач під час заняття бігом за цією активністю, коли було зафіксовано цей запис спостереження.

Поле runningrecordlongitude має тип DecimalField і ним встановлена довгота точки, в якій перебував цей користувач під час заняття бігом за цією активністю, коли було зафіксовано цей запис спостереження.

Поле runningrecordaltitude має тип DecimalField і ним встановлена висота точки, в якій перебував цей користувач під час заняття бігом за цією активністю, коли було зафіксовано цей запис спостереження.

Поле runningrecordlap має тип PositiveSmallIntegerField і ним встановлене коло, на якому перебував цей користувач під час заняття бігом за цією активністю, коли було зафіксовано цей запис спостереження.

Поле runningrecordtime має тип DateTimeField і ним встановлена записана часова мітка під час заняття бігом за цією активністю, коли було зафіксовано цей запис спостереження.

Поле runningrecordheartrate має тип PositiveSmallIntegerField і ним встановлена кількість ударів серця на хвилину в користувача під час заняття бігом за цією активністю, коли було зафіксовано цей запис спостереження.

Поле runningrecorddistance має тип DecimalField і ним встановлена кількість кілометрів, подолана користувачем під час цього заняття бігом за цією активністю з моменту останнього запису, коли така активність фіксувалась.

У підсистемі моделей ПС клас Runningcompetitiontype відповідає за збереження даних типів офіційних змагань з бігу, маючи тільки поле

runningcompetitiontypename, що має тип CharField і яким задана назва типу офіційних змагань з бігу.

У підсистемі моделей ПС клас Runningcountry відповідає за збереження даних країн, в яких відбуваються змагання з бігу та які за національністю мають бігуни, маючи тільки поле runningcountryname, що має тип CharField і яким задана назва країни, в якій можуть відбуватися змагання з бігу та яку за національністю можуть мати бігуни.

У підсистемі моделей ПС клас Runningcity відповідає за збереження даних міст, в яких відбуваються змагання з бігу та які за національністю мають бігуни, маючи такі поля.

Поле runningcitycountry має тип ForeignKey і ним встановлена країна, в якій знаходиться це місто, в якому можуть відбуватися заняття з бігу, шляхом зв'язування з моделлю Runningcountry.

Поле runningcityname має тип CharField і ним встановлена назва міста, в якому можуть відбуватися змагання з бігу.

У підсистемі моделей ПС клас Runningcompetition відповідає за збереження даних офіційних змагань з бігу, маючи такі поля.

Поле runningcompetitiontype має тип ForeignKey і ним встановлений тип змагання з бігу, якому відповідає це, шляхом зв'язування з моделлю Runningcompetitiontype.

Поле runningcompetitioncity має тип ForeignKey і ним встановлене місто, в якому відбувається це офіційне змагання з бігу, шляхом зв'язування з моделлю Runningcity.

Поле runningcompetitiontitle має тип CharField і ним встановлена назва цього офіційного змагання з бігу.

Поле runningcompetitionweb має тип URLField і ним встановлена вебадреса офіційного змагання з бігу.

Поле runningcompetitiondistance має тип DecimalField і ним встановлена відстань, на яку розраховане це змагання з бігу.

Поле `runningcompetitionlaps` має тип `PositiveSmallIntegerField` і ним встановлена кількість кіл, яку потрібно подолати під час цього змагання з бігу.

У підсистемі моделей ПС клас `Runningcompetitionyear` відповідає за збереження даних конкретних років проведення офіційних змагань з бігу, маючи такі поля.

Поле `runningcompetition` має тип `ForeignKey` і ним встановлене офіційне змагання з бігу, яке проводиться цього року, шляхом зв'язування з моделлю `Runningcompetition`.

Поле `runningcompetitiondate` має тип `DateTimeField` і ним встановлений час початку проведення цього офіційного змагання з бігу цього року.

У підсистемі моделей ПС клас `Runningcompetitionresult` відповідає за збереження даних результатів участі конкретних бігунів у змаганні певного року, маючи такі поля.

Поле `runningcompetitionyear` має тип `ForeignKey` і ним встановлений рік проведення змагання з бігу, в якому було зафіксовано цей результат, шляхом зв'язування з моделлю `Runningcompetitionyear`.

Поле `runningpartner` має тип `ForeignKey` і ним встановлений користувач, який займається бігом і результат якого зафіксовано в цьому записі, шляхом зв'язування з моделлю `Runningpartne`.

Поле `runningcompetitionresultplace` має тип `PositiveSmallIntegerField` і ним встановлене місце, яке зайняв цей користувач під цього проведення змагання з бігу.

Поле `runningcompetitionresulttime` має тип `TimeField` і ним встановлений час, за який пробіг дистанцію на цьому змаганні з бігу цей користувач

Поле `runningcompetitionresultdistance` має тип `DeciamlField` і ним встановлена відстань, яку подолав за цей час цей користувач під час цього змагання з бігу.

У підсистемі моделей ПС клас Runningpartnerfollow відповідає за збереження даних певної форми дружби між користувачами, які займаються бігом, маючи такі поля.

Поле runningpartner має тип ForeignKey і ним встановлений користувач, який займається бігом та якому був адресований початково створений запис про дружбу в програмі, шляхом зв'язування з моделлю Runningpartner.

Поле runningpartnerfollower має тип ForeignKey і ним встановлений користувач, який займається бігом та який початково створив цей запис про дружбу в програмі, шляхом зв'язування з моделлю Runningpartner.

Поле runningfollowtype має тип BooleanField і ним встановлене визначення форми дружби між користувачами, які займаються бігом: підписка одного користувача на іншого, взаємне затвердження і таким чином визначення дружби.

У підсистемі моделей ПС клас Runningpoint відповідає за збереження даних окремо виділених точок, які під час занять бігом долають користувачі, маючи такі поля.

Поле runningpointtitle має тип CharField і ним встановлена назва окремо виділеної точки, яку під час заняття бігом долає користувач.

Поле runningpointlongitude має тип DecimalField і ним встановлена довгота розміщення окремо виділеної точки, яку під час заняття бігом долає користувач.

Поле runningpointlatitude має тип DecimalField і ним встановлена широта розміщення окремо виділеної точки, яку під час заняття бігом долає користувач.

Поле runningpointaltitude має тип DecimalField і ним встановлена висота розміщення окремо виділеної точки, яку під час заняття бігом долає користувач.

Поле runningpointreviewscore має тип DecimalField і ним встановлена усереднена оцінка, виставлена цій окремо виділеній точці за відгуками бігунів.

У підсистемі моделей ПС клас Runningplace відповідає за збереження даних місць, які під час занять бігом долають користувачі, маючи такі поля.

Поле runningplacetype має тип ForeignKey і ним встановлений тип місця, яке під час заняття бігом долають користувачі, шляхом зв'язування з моделлю Runningplacetype.

Поле runningplacetitle має тип CharField і ним встановлена назва місця, яке під час заняття бігом долають користувачі.

Поле runningplacelongitude має тип DecimalField і ним встановлена довгота розташування місця, яке під час заняття бігом долають користувачі.

Поле runningplacelatitude має тип DecimalField і ним встановлена широта розташування місця, яке під час заняття бігом долають користувачі.

Поле runningplacealtitude має тип DecimalField і ним встановлена висота розташування місця, яке під час заняття бігом долають користувачі.

Поле runningplacereviewscore має тип DecimalField і ним встановлена усереднена оцінка, виставлена цьому місцю, яке під час заняття бігом долають користувачі.

У підсистемі моделей ПС клас Runningplacetype відповідає за збереження даних типів місць, які під час заняття бігом долають користувачі, маючи тільки поле runningplacetype_name, що має тип CharField і яким задана назва типу місць, які під час заняття бігом долають користувачі.

У підсистемі моделей ПС клас таблиця Runningplacereview відповідає за збереження даних відгуків про місця, які під час занять бігом долають користувачі, маючи такі поля.

Поле runningplace має тип ForeignKey і ним встановлене місце, яке під час заняття бігом долають користувачі і якого стосується відгук, шляхом зв'язування з моделлю Runningplace.

Поле runningpartner має тип ForeignKey і ним встановлений користувач, який створив цей відгук про місце, шляхом зв'язування з моделлю Runningpartner.

Поле `runningplacereviewtime` має тип `DateTimeField` і ним встановлений час створення відгуку про місце, яке під час заняття бігом долають користувачі.

Поле `runningplacereviewtext` має тип `CharField` і ним встановлений текст відгуку про місце, яке під час заняття бігом долають користувачі.

Поле `runningplacereviewscore` має тип `PositiveSmallIntegerField` і ним встановлена оцінка у відгуку про місце, яке під час заняття бігом долають користувачі.

У підсистемі моделей ПС клас `Runningplacereviewphoto` відповідає за збереження даних фотографії, доданої до відгуку про місце, яке під час заняття бігом долають користувачі, маючи такі поля.

Поле `runningplacereview` має тип `ForeignKey` і ним встановлений відгук про місце, яке під час заняття бігом долають користувачі, до якого додана ця фотографія, шляхом зв'язування з моделлю `Runningplacereview`.

Поле `runningplacereviewphotodata` має тип `ImageField` і ним встановлені дані фотографії, доданої до відгуку про місце, яке під час заняття бігом долають користувачі.

У підсистемі моделей ПС клас `Runningpointreview` відповідає за збереження даних відгуків про точки, які під час занять бігом долають користувачі, маючи такі поля.

Поле `runningpoint` має тип `ForeignKey` і ним встановлена точка, яку під час заняття бігом долають користувачі і якої стосується відгук, шляхом зв'язування з моделлю `Runningpoint`.

Поле `runningpartner` має тип `ForeignKey` і ним встановлений користувач, який створив цей відгук про точку, шляхом зв'язування з моделлю `Runningpartner`.

Поле `runningpointreviewtime` має тип `DateTimeField` і ним встановлений час створення відгуку про точку, яку під час заняття бігом долають користувачі.

Поле `runningpointreviewtext` має тип `CharField` і ним встановлений текст відгуку про точку, яку під час заняття бігом долають користувачі.

Поле `runningpointreviewscore` має тип `PositiveSmallIntegerField` і ним встановлена оцінка у відгуку про точку, яку під час заняття бігом долають користувачі.

У підсистемі моделей ПС клас `Runningpointreviewphoto` відповідає за збереження даних фотографії, доданої до відгуку про точку, яку під час заняття бігом долають користувачі, маючи такі поля.

Поле `runningpointreview` має тип `ForeignKey` і ним встановлений відгук про точку, яку під час заняття бігом долають користувачі, до якого додана ця фотографія, шляхом зв'язування з моделлю `Runningpointreview`.

Поле `runningpointreviewphotodata` має тип `ImageField` і ним встановлені дані фотографії, доданої до відгуку про точку, яку під час заняття бігом долають користувачі.

У підсистемі моделей ПС клас `Runningplacesave` відповідає за збереження даних відмічування місць, які під час занять бігом долають користувачі, маючи такі поля.

Поле `runningplace` має тип `ForeignKey` і ним встановлене місце, яке під час заняття бігом долають користувачі і яке було відзначено цим користувачем, шляхом зв'язування з моделлю `Runningplace`.

Поле `runningpartner` має тип `ForeignKey` і ним встановлений користувач, який займається бігом і який відзначив це місце, шляхом зв'язування з моделлю `Runningpartner`.

Поле `runningsavetype` має тип `BooleanField` і ним встановлений тип відмічування цього місця, яке під час заняття бігом долають користувачі, певним користувачем: публічний або приватний.

У підсистемі моделей ПС клас `Runningpointsave` відповідає за збереження даних відмічування точок, які під час занять бігом долають користувачі, маючи такі поля.

Поле `runningpoint` має тип `ForeignKey` і ним встановлена точка, яку під час заняття бігом долають користувачі і яку було відзначено цим користувачем, шляхом зв'язування з моделлю `Runningpoint`.

Поле `runningpartner` має тип `ForeignKey` і ним встановлений користувач, який займається бігом і який відзначив цю точку, шляхом зв'язування з моделлю `Runningpartner`.

Поле `runningsavetype` має тип `BooleanField` і ним встановлений тип відмічування цієї точки, яку під час заняття бігом долають користувачі, певним користувачем: публічний або приватний.

У підсистемі моделей ПС клас `Runningrequest` відповідає за збереження даних запитів на спільне заняття бігом, маючи такі поля.

Поле `runningpartner` має тип `ForeignKey` і ним встановлений користувач, який займається бігом та який створив цей запиті на заняття бігом, шляхом зв'язування з моделлю `Runningpartner`.

Поле `runningpartnerto` має тип `ForeignKey` і ним встановлений користувач, який займається бігом та якому адресований цей запиті на заняття бігом, шляхом зв'язування з моделлю `Runningpartner`.

Поле `runningrequesttitle` має тип `CharField` і ним встановлена назва запиту на спільне заняття бігом.

Поле `runningrequeststarttime` має тип `DateTimeField` і ним встановлений пропонуванний час початку спільного заняття бігом.

Поле `runningrequestnote` має тип `CharField` і ним встановлена примітка до спільного заняття бігом.

Поле `runningrequeststate` має тип `CharField` і ним встановлений стан запиту на спільне заняття бігом.

У підсистемі моделей ПС клас `Runningrequestpoint` відповідає за збереження точок, пропонуваних для спільного заняття бігом, маючи такі поля.

Поле `runningpoints` має тип `ManyToManyField` і ним встановлені точки, пропоновані для спільного заняття бігом, шляхом зв'язування з моделлю `Runningpoint`.

У підсистемі моделей ПС клас `Runninggoal` відповідає за збереження даних пропонованих змагань для занять бігом за показниками, маючи такі поля.

Поле `runningpartner` має тип `ForeignKey` і ним встановлений користувач, який створив це змагання для занять бігом за показниками, шляхом зв'язування з моделлю `Runningpartner`.

Поле `runningperiodtype` має тип `CharField` і ним встановлений тип періоду змагання для занять бігом за показниками: протягом доби, тижня, місяця, року.

Поле `runninggoalstarttime` має тип `DateTimeField` і ним встановлений час початку змагання для занять бігом за показниками.

Поле `runninggoaldistance` має тип `DecimalField` і ним встановлена відстань в кілометрах, яку потрібно подолати під час цього змагання з бігу за показниками.

Поле `runninggoaltime` має тип `TimeField` і ним встановлений час, протягом якого потрібно займатись бігом у цьому змаганні за показниками.

Поле `runninggoalperiod` має тип `BooleanField` і ним встановлена можливість досягти вказаних показників за декілька разів.

У підсистемі моделей ПС клас `Runninggoalpartner` відповідає за збереження даних учасників таких змагань з бігу за показниками, маючи такі поля.

Поле `runninggoal` має тип `ForeignKey` і ним встановлене заняття з бігу за показниками, до якого відноситься ця участь, шляхом зв'язування з моделлю `Runninggoal`.

Поле `runningpartner` має тип `ForeignKey` і ним встановлений користувач, який займається бігом і який входить до цього змагання з бігу за

показниками або якому запропоновано увійти до такого змагання, шляхом зв'язування з моделлю Runningpartner.

Поле `runninggoalpartnerstate` має тип `CharField` і ним встановлений стан внесення учасника до змагання з бігу за показниками.

Поле `runninggoaldistance` має тип `DecimalField` і ним встановлена відстань, яку подолав цей користувач як учасник цього змагання з бігу за показниками.

Поле `runninggoaltime` має тип `TimeField` і ним встановлений час, протягом якого займався бігом цей користувач як учасник цього змагання з бігу за показниками.

У підсистемах представлень та шаблонів ПС для пошуку друзів для спільних занять бігом реалізовано наступні функції. Вони описані нижче за розподілом за сценаріями використання, тому що взаємодіють спільно за кожним сценарієм. Підсистема представлень готує дані для сценарію, а підсистема шаблонів отримує ці дані та заповнює ними сторінку, яку передає користувачу.

Функція відображення профілю користувача, який займається бігом, у підсистемі представлень реалізована як `runningpartner_view` з параметром `runningpartner_login`, який встановлює логін користувача, який займається бігом, і який може бути не заданий, але тоді користувач повинен бути авторизований, звідки беруться ці дані. У підсистемі шаблонів, створені за допомогою шаблону [10], реалізовано відповідний шаблон `runner.html`.

Функція відображення відмічених користувачем точок у підсистемі представлень реалізована як `runningpartner_saved_points_view` з параметром `runningpartner_login`, який встановлює логін користувача, який займається бігом, і який може бути не заданий, але тоді користувач повинен бути авторизований, звідки беруться ці дані. У підсистемі шаблонів реалізовано відповідний шаблон `runnerpoints.html`.

Функція відображення відмічених користувачем місць у підсистемі представлень реалізована як `runningpartner_saved_places_view` з параметром

`runningpartner_login`, який встановлює логін користувача, який займається бігом, і який може бути не заданий, але тоді користувач повинен бути авторизований, звідки беруться ці дані. У підсистемі шаблонів реалізовано відповідний шаблон `runnerplaces.html`.

Функція відображення користувачів, які займаються бігом та на яких підписався даний у підсистемі представлень реалізована як `runningpartner_follows_view` з параметром `runningpartner_login`, який встановлює логін користувача, який займається бігом, і який може бути не заданий, але тоді користувач повинен бути авторизований, звідки беруться ці дані. У підсистемі шаблонів реалізовано відповідний шаблон `runnerfollows.html`.

Функція відображення користувачів, які займаються бігом та які підписалися на даного у підсистемі представлень реалізована як `runningpartner_followers_view` з параметром `runningpartner_login`, який встановлює логін користувача, який займається бігом, і який може бути не заданий, але тоді користувач повинен бути авторизований, звідки беруться ці дані. У підсистемі шаблонів реалізовано відповідний шаблон `runnerfollowers.html`.

Функція відображення власних пристроїв для фіксації занять бігом у підсистемі представлень реалізована як `runningpartner_devices_view`. У підсистемі шаблонів реалізовано відповідний шаблон `runnerdevices.html`.

Функція відображення активностей з занять бігом користувача у підсистемі представлень реалізована як `runningpartner_activities_view` з параметром `runningpartner_login`, який встановлює логін користувача, який займається бігом, і який може бути не заданий, але тоді користувач повинен бути авторизований, звідки беруться ці дані. У підсистемі шаблонів реалізовано відповідний шаблон `runneractivities.html`.

Функція відображення відгуків про точки, які під час занять бігом долають користувачі та які створив цей користувач, у підсистемі представлень реалізована як `runningpartner_points_reviews_view` з параметром

`runningpartner_login`, який встановлює логін користувача, який займається бігом, і який може бути не заданий, але тоді користувач повинен бути авторизований, звідки беруться ці дані. У підсистемі шаблонів реалізовано відповідний шаблон `runnerpointreviews.html`.

Функція відображення відгуків про місця, які під час занять бігом долають користувачі та які створив цей користувач, у підсистемі представлень реалізована як `runningpartner_places_reviews_view` з параметром `runningpartner_login`, який встановлює логін користувача, який займається бігом, і який може бути не заданий, але тоді користувач повинен бути авторизований, звідки беруться ці дані. У підсистемі шаблонів реалізовано відповідний шаблон `runnerplacesreviews`.

Функція відображення відгуків за точкою, яку під час занять бігом долають користувачі, у підсистемі представлень реалізована як `runningpartner_points_reviews_view` з параметром `runningpoint_id`, який встановлює ідентифікатор точки, яку під час занять бігом долають користувачі. У підсистемі шаблонів реалізовано відповідний шаблон `runnerpointreviews.html`.

Функція відображення відгуків за місцем, яке під час занять бігом долають користувачі, у підсистемі представлень реалізована як `runningplace_reviews_view` з параметром `runningplace_id`, який встановлює ідентифікатор місця, яке під час занять бігом долають користувачі. У підсистемі шаблонів реалізовано відповідний шаблон `runnerplacereviews.html`.

Функція реєстрації власного пристрою для фіксації занять бігом у підсистемі представлень реалізована як `runningpartnerdevice_add_view` з параметрами `runningdevice_id`, який встановлює ідентифікатор пристрою для фіксації занять бігом, та `runningdevicesn`, який задає його серійний номер. У підсистемі шаблонів реалізовано відповідний шаблон `runnerdeviceadd.html`.

Функція пошуку серед користувачів, які займаються бігом, у підсистемі представлень реалізована як `runningpartner_search_view`. У підсистемі шаблонів реалізовано відповідний шаблон

runningpartnersearch.html, параметри ж пошуку передаються через запит до сторінки.

Функція оновлення власного профілю користувача, який займається бігом, у підсистемі представлень реалізована як runningpartner_edit_view. У підсистемі шаблонів реалізовано відповідний шаблон runneredit.html. У підсистемі форм реалізовано клас RunningpartnerForm.

Функція відображення деталізованої статистики щодо занять бігом користувача за рік у підсистемі представлень реалізована як runningpartnerstatistic_view з параметром runningpartnerstatistic_id, який встановлює ідентифікатор статистики щодо занять бігом користувача за рік. У підсистемі шаблонів реалізовано відповідний шаблон runningpartnerstatistic.html.

Функція відображення деталізованої активності з занять бігом користувача у підсистемі представлень реалізована як runningactivity_view з параметром runningactivity_id, який встановлює ідентифікатор активності з занять бігом користувача. У підсистемі шаблонів реалізовано відповідний шаблон runningactivity.html.

Функція додавання виконаної активності з занять бігом користувача у підсистемі представлень реалізована як runningactivity_add_view. У підсистемі шаблонів реалізовано відповідний шаблон runningactivityadd.html. У підсистемі форм реалізовано клас RunningactivityForm.

Функція оновлення виконаної активності з занять бігом користувача у підсистемі представлень реалізована як runningactivity_edit_view з параметром runningactivity_id, який встановлює ідентифікатор активності з занять бігом користувача. У підсистемі шаблонів використовується відповідний шаблон runningactivityadd.html.

Функція додавання запису фіксації активності з занять бігом користувача у підсистемі представлень реалізована як runningrecord_add_view з параметром runningactivity_id, який встановлює ідентифікатор активності з занять бігом користувача. У підсистемі шаблонів реалізовано відповідний

шаблон `runningrecordadd.html`. У підсистемі форм реалізовано клас `RunningrecordForm`.

Функція оновлення запису фіксації активності з занять бігом користувача у підсистемі представлень реалізована як `runningrecord_edit_view` з параметром `runningrunningrecord_id`, який встановлює ідентифікатор запису фіксації активності з занять бігом. У підсистемі шаблонів використовується відповідний шаблон `runningrecordadd.html`.

Функція зв'язування виконаної активності з занять бігом користувача з результатом офіційного змагання у підсистемі представлень реалізована як `runningactivitycompetitionresult_add_view` з параметрами `runningactivity_id`, який встановлює ідентифікатор активності з занять бігом користувача, та `runningcompetitionresult_id`, який задає ідентифікатор результату офіційного змагання. У підсистемі шаблонів використовується відповідний шаблон `runningactivitycompetitionresultadd.html`.

Функція розірвання зв'язування виконаної активності з занять бігом користувача з результатом офіційного змагання у підсистемі представлень реалізована як `runningactivitycompetitionresult_delete_view` з параметром `runningactivity_id`, який встановлює ідентифікатор активності з занять бігом користувача, та `runningcompetitionresult_id`, який задає ідентифікатор результату офіційного змагання.

Функція додавання результату участі користувача в офіційному змаганні з бігу у підсистемі представлень реалізована як `runningcompetitionresult_add_view`. У підсистемі шаблонів реалізовано відповідний шаблон `runningcompetitionresultadd.html`. У підсистемі форм реалізовано клас `RunningcompetitionresultForm`.

Функція оновлення результату участі користувача в офіційному змаганні з бігу у підсистемі представлень реалізована як `runningcompetitionresult_edit_view` з параметром `runningcompetitionresult_id`, який встановлює ідентифікатор результату участі користувача в офіційному

змаганні з бігу. У підсистемі шаблонів використовується відповідний шаблон `runningcompetitionresultadd.html`.

Функція відображення результатів офіційного змагання з бігу за рік проведення у підсистемі представлень реалізована як `runningcompetitionyear_detail_view` з параметром `runningcompetitionyear_id`, який встановлює ідентифікатор проведення офіційного змагання з бігу за рік. У підсистемі шаблонів реалізовано відповідний шаблон `runningcompetitionyear.html`.

Функція відображення даних офіційного змагання з бігу у підсистемі представлень реалізована як `runningcompetition_view` з параметром `runningcompetition_id`, який задає ідентифікатор офіційного змагання з бігу. У підсистемі шаблонів реалізовано відповідний шаблон `runningcompetition.html`.

Функція підписування на користувача, який займається бігом, у підсистемі представлень реалізована як `runningpartnerfollow_add_view` з параметром `runningpartner_login`, який встановлює логін користувача, який займається бігом.

Функція відписування від користувача, який займається бігом у підсистемі представлень реалізована як `runningpartnerfollow_delete_view` з параметром `runningpartner_login`, який встановлює логін користувача, який займається бігом.

Функція підтвердження дружби у відповідь на підписку від користувача, який займається бігом у підсистемі представлень реалізована як `runningpartnerfollow_touch_view` з параметром `runningpartner_login`, який встановлює логін користувача, який займається бігом.

Функція виділення точки, яку під час заняття бігом долають користувачі, у підсистемі представлень реалізована як `runningpoint_add_view`. У підсистемі шаблонів реалізовано відповідний шаблон `runningpointadd.html`. У підсистемі форм реалізовано клас `RunningpointForm`.

Функція пошуку точок, які під час занять бігом долають користувачі, у підсистемі представлень реалізована як `runningpoints_search_view`. У

підсистемі шаблонів реалізовано відповідний шаблон `runningpointssearch.html`.
Параметри пошуку передаються в сторінку через параметри запити.

Функція пошуку місць, які під час занять бігом долають користувачі, у підсистемі представлень реалізована як `runningplaces_search_view`. У підсистемі шаблонів реалізовано відповідний шаблон `runningplacessearch.html`.
Параметри пошуку передаються в сторінку через параметри запити.

Функція відображення даних точки, яку під час занять бігом долають користувачі, у підсистемі представлень реалізована як `running_point_view` з параметром `runningpoint_id`, який задає ідентифікатор точки, яку під час занять бігом долають користувачі. У підсистемі шаблонів реалізовано відповідний шаблон `runningpoint.html`.

Функція відображення даних місця, яке під час занять бігом долають користувачі, у підсистемі представлень реалізована як `running_place_view` з параметром `runningplace_id`, який задає ідентифікатор місця, яке під час занять бігом долають користувачі. У підсистемі шаблонів реалізовано відповідний шаблон `runningplace.html`.

Функція створення відгуку про точку, яку під час занять бігом долають користувачі, у підсистемі представлень реалізована як `runningpoint_review_add_view` з параметром `runningpoint_id`, який задає ідентифікатор точки, яку під час занять бігом долають користувачі. У підсистемі шаблонів реалізовано відповідний шаблон `runningpointreviewadd.html`. У підсистемі форм реалізовано клас `RunningpointreviewForm`.

Функція створення відгуку про місце, яке під час занять бігом долають користувачі, у підсистемі представлень реалізована як `runningplace_review_add_view` з параметром `runningplace_id`, який задає ідентифікатор місця, яке під час занять бігом долають користувачі. У підсистемі шаблонів реалізовано відповідний шаблон `runningplacereviewadd.html`. У підсистемі форм реалізовано клас `RunningplacereviewForm`.

Функція відмічування місця, яке під час занять бігом долають користувачі, у підсистемі представлень реалізована як `runningplacesave_view` з параметрами `runningplace_id`, який задає ідентифікатор місця, яке під час занять бігом долають користувачі, та `runningsavetype`, який задає тип відмічування (публічний або приватний).

Функція відмічування точки, яку під час занять бігом долають користувачі, у підсистемі представлень реалізована як `runningpointsave_view` з параметрами `runningpoint_id`, який задає ідентифікатор точки, яку під час занять бігом долають користувачі, та `runningsavetype`, який задає тип відмічування (публічний або приватний).

Функція пошуку офіційних змагань з бігу у підсистемі представлень реалізована як `runningcompetitions_search`. У підсистемі шаблонів реалізовано відповідний шаблон `runningcompetitionssearch.html`. Параметри пошуку передаються в сторінку через параметри запити.

Функція надсилання запити за спільним заняттям з бігу у підсистемі представлень реалізована як `runningrequest_send_view` з параметром `runningpartner_login`, який встановлює логін користувача, який займається бігом і якому адресовано запит.

Функція підтвердження запити за спільним заняттям з бігу у підсистемі представлень реалізована як `runningrequest_plus_view` з параметром `runningrequest_id`, який задає ідентифікатор запити за спільним заняттям з бігу.

Функція скасування запити за спільним заняттям з бігу у підсистемі представлень реалізована як `runningrequest_minus_view` з параметром `runningrequest_id`, який задає ідентифікатор запити за спільним заняттям з бігу.

Функція відображення надісланих запитів за спільними заняттями з бігу у підсистемі представлень реалізована як `runningrequests_sent_view`. У підсистемі шаблонів реалізовано відповідний шаблон `sentrunningrequests.html`.

Функція відображення отриманих запитів за спільними заняттями з бігу у підсистемі представлень реалізована як `runningrequests_received_view`.

У підсистемі шаблонів реалізовано відповідний шаблон `receivedrunningrequests.html`.

Функція відображення визначених спільних занять з бігу у підсистемі представлень реалізована як `runningrequest_commons_view`. У підсистемі шаблонів реалізовано відповідний шаблон `commonrunningrequests.html`.

Функція пропонування змагання для занять бігом за показниками у підсистемі представлень реалізована як `runninggoal_add_view`. У підсистемі шаблонів реалізовано відповідний шаблон `runninggoaladd.html`. У підсистемі форм реалізовано клас `RunninggoalForm`.

Функція оновлення пропонованого змагання для занять бігом за показниками у підсистемі представлень реалізована як `runninggoal_edit_view` з параметром `runninggoal_id`, який задає ідентифікатор пропонованого змагання для занять бігом за показниками. У підсистемі шаблонів використовується відповідний шаблон `runninggoaladd.html` та клас форми `RunninggoalForm`.

Функція надсилання запиту на участь у змаганні для занять бігом за показниками у підсистемі представлень реалізована як `runninggoalpartner_add_view` з параметром `runninggoal_id`, який задає ідентифікатор змагання для занять бігом за показниками. У підсистемі шаблонів реалізовано відповідний шаблон `runninggoalpartneradd.html`.

Функція підтвердження запиту на участь у змаганні для занять бігом за показниками у підсистемі представлень реалізована як `runninggoalpartner_plus_view` з параметром `runninggoalpartner_id`, який задає ідентифікатор запиту на участь у змаганні для занять бігом за показниками.

Функція скасування запиту на участь у змаганні для занять бігом за показниками у підсистемі представлень реалізована як `runninggoalpartner_minus_view` з параметром `runninggoalpartner_id`, який задає ідентифікатор запиту на участь у змаганні для занять бігом за показниками.

Функція відображення змагань для занять бігом за показниками, в яких бере участь користувач, у підсистемі представлень реалізована як

active_runninggoals_view. У підсистемі шаблонів реалізовано відповідний шаблон activerunninggoals.html.

Функція відображення отриманих запитів на участь у змаганні для занять бігом за показниками у підсистемі представлень реалізована як received_runninggoals_view. У підсистемі шаблонів реалізовано відповідний шаблон receivedrunninggoals.html.

Функція відображення надісланих запитів на участь у змаганні для занять бігом за показниками у підсистемі представлень реалізована як sent_runninggoals_view. У підсистемі шаблонів реалізовано відповідний шаблон sentrunninggoals.html.

Функція відображення деталізованого заняття бігом за показниками у підсистемі представлень реалізована як runninggoalpartner_detail_view з параметром runninggoalpartner_id, який задає ідентифікатор цього заняття бігом за показниками. У підсистемі шаблонів реалізовано відповідний шаблон runninggoalpartner.html.

Функція відображення деталізованого спільного заняття з бігу у підсистемі представлень реалізована як runningrequest_detail_view з параметром runningrequest_id, який задає ідентифікатор цього спільного заняття з бігу. У підсистемі шаблонів реалізовано відповідний шаблон runningrequestdetail.html.

Функція відображення змагань для занять бігом за показниками за користувачами з підписок у підсистемі представлень реалізована як runninggoals_by_saves_view. У підсистемі шаблонів реалізовано відповідний шаблон savesrunninggoals.html.

Функція відображення активностей з бігу за користувачами з підписок у підсистемі представлень реалізована як runningactivities_by_saves_view. У підсистемі шаблонів реалізовано відповідний шаблон savesrunningactivities.html.

Виконання CRUD-операцій забезпечується у ПС за рахунок адміністративної панелі, яка є частиною створеної системи одразу на основі створення її за допомогою фреймворку Django.

3.3 Висновки за розділом 3

Реалізацію ПС пошуку друзів для спільних занять бігом завершено з отриманням результатів, які включають створені підсистеми моделей, представлень, форм, шаблонів, які описані в даному розділі.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

4.1 Призначення програмної системи

ПС призначена для використання її для накопичення власних даних щодо результатів бігових активностей користувачами з одного боку, а з іншого боку для пошуку друзів для спільних занять бігом, планування таких занять бігом на основі визначення основних точок і місць для таких занять, організації віддалених занять стосовно досягнення певного рівня показників під час занять бігом протягом заданого періоду часу.

4.2 Умови виконання програмної системи

Умови виконання ПС пошуку друзів для спільних занять бігом включають необхідність виділення сервера, де має бути забезпечено підтримку мови Python, бази даних, робота з якою має виконуватись через систему PostgreSQL. Технічні засоби, необхідні для організації роботи серверу, включають вимоги до процесору щодо наявності хоча би 2 ядр, оперативної пам'яті, вивільненої для підтримки роботи програмної системи пошуку друзів для спільних занять бігом у обсязі хоча би 4 Гб, та щодо жорсткого диску стосовно наявності на ньому хоча би 50 Гб вільного простору.

4.3 Експлуатація програмної системи для пошуку друзів для спільних занять бігом

Після своєї авторизації користувач ПС для пошуку друзів для спільних занять бігом може працювати з розділами меню, які об'єднують основні функції роботи.

Розділ активностей дозволяє одразу вносити свої власні бігові активності та переглядати бігові активності за тими користувачами, на яких

підписався цей користувач.

За кожним любителем бігу є сторінка профілю, яка представляє по ньому основні дані. Вона доступна кожному користувачу з розділу профілю, тоді він може перейти до його редагування. У випадку ж відображення сторінки іншого користувача доступні засоби підписування на нього, пропонування спільного заняття бігом (рис. 4.1).

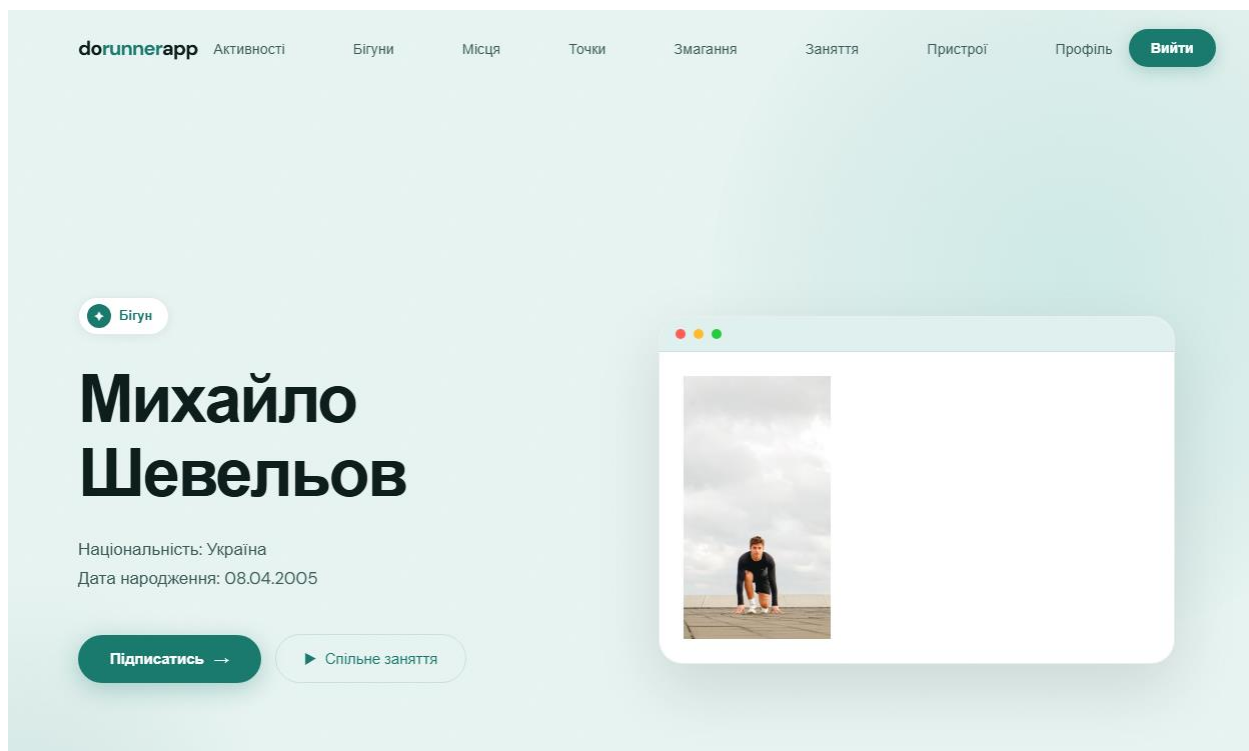


Рисунок 4.1 – Відображення профілю користувача, який займається бігом (презентаційна картка)

Стосовно інших складових загалом на цій сторінці присутні зони з виділенням 3 елементів та кнопками для переходу на сторінки відображення повних даних. Зокрема це стосується відображення бігових активностей (рис. 4.2). Так само відображаються створені користувачем відгуки за місцями, точками, в тому числі за збереженими місцями, точками результати участі користувача в змаганнях.

З розділу бігунів користувачу доступна можливість переходу до пошуку бігунів та відображення підписок (тих, хто на нього підписався, або

тих, на кого він підписався). При відображенні тих, хто підписався на цього користувача, доступна можливість підтвердити такий зв'язок та стати друзями, або створений такий зв'язок можна розірвати (рис. 4.3).

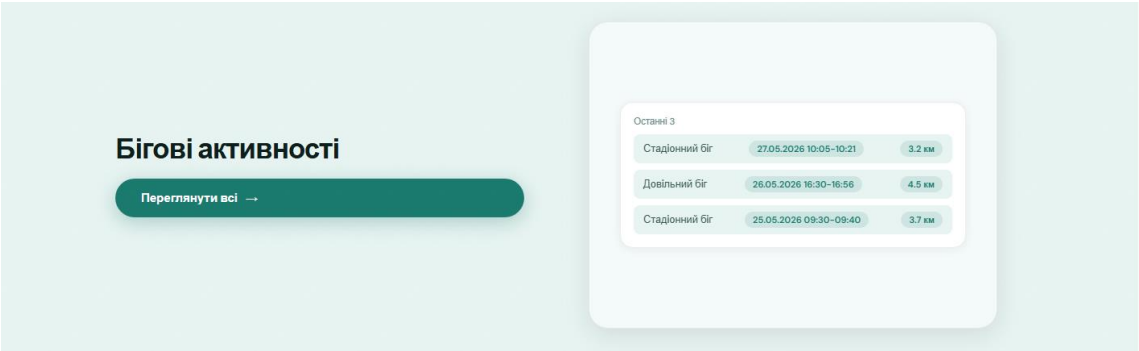


Рисунок 4.2 – Відображення профілю користувача, який займається бігом (перелік бігових активностей)

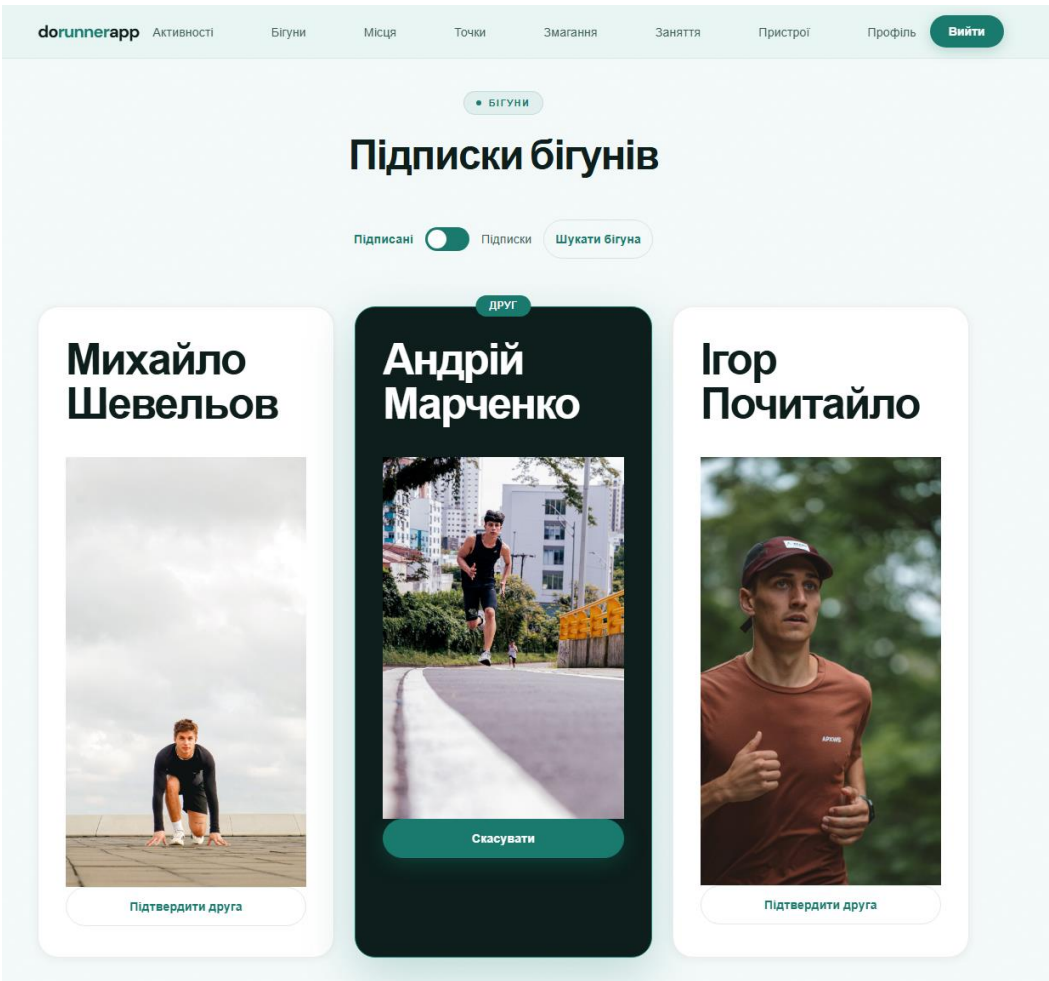


Рисунок 4.3 – Відображення користувачів, які займаються бігом та які підписалися на даного

З розділу точок користувач може перейти або до пошуку таких точок для занять бігом, або до власних створених відгуків за точками, або до відзначених ним точок. Також доступний перехід до створення нової точки з виділенням її за координатами.

З розділу змагань доступна можливість додати результати власної участі в офіційному змаганні, виконати пошук серед доступних змагань, переглянути змагання за показниками, визначені користувачами, на яких оформлено підписки, переходу до запитів на змагання.

З розділу занять доступна можливість переглянути надіслані, отримані запити на спільні заняття бігом, а також вже сформовані спільні заняття бігом.

З розділу пристроїв доступна можливість переглянути ті пристрої, які використовуються користувачем для фіксації занять бігом, а також додати новий.

З розділу місць користувач може перейти або до пошуку таких місць для занять бігом, або до власних створених відгуків за місцями, або до відзначених ним місць. За результатами пошуку можна перейти до місця, відобразити відгуки за ним (рис. 4.4).

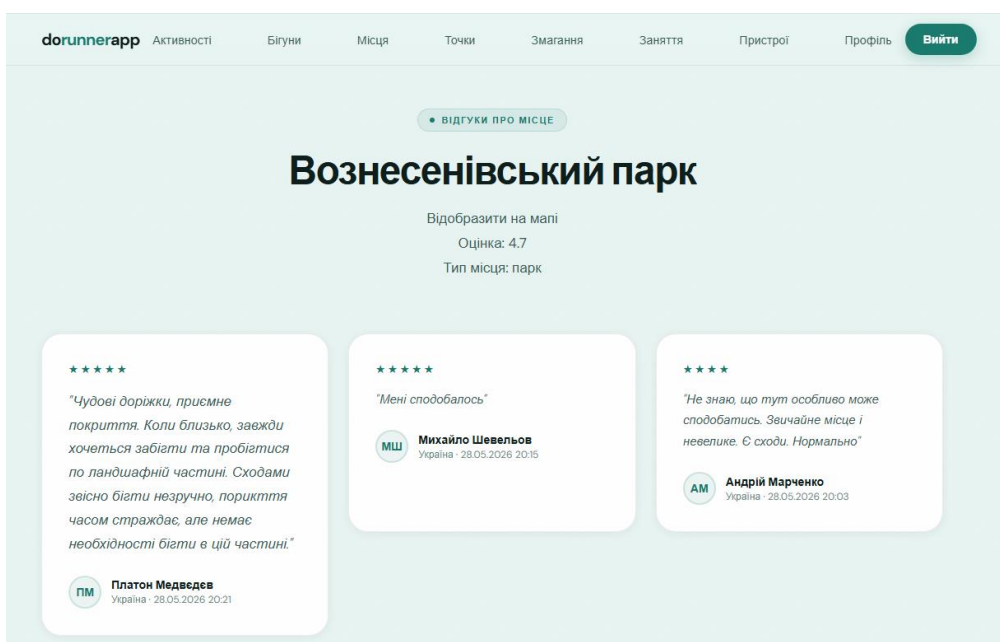


Рисунок 4.4 – Відображення відгуків за місцем, яке під час занять бігом долають користувачі

4.4 Тестування програмної системи для пошуку друзів для спільних занять бігом

Чек-лист тестування ПС для пошуку друзів для спільних занять бігом (табл. 4.1) містить результати проходження тестів за кожною функцією ПС. Позначки «+» свідчать про успішність завершення кожного тесту.

Таблиця 4.1 – Чек-лист тестування ПС для пошуку друзів для спільних занять бігом

Функція під час тестування	Результат тестування
1	2
Реєстрація як користувача, який займається бігом	+
Реєстрація власного пристрою для фіксації занять бігом	+
Відображення профілю користувача, який займається бігом	+
Пошук серед користувачів, які займаються бігом	+
Оновлення власного профілю користувача, який займається бігом	+
Авторизація	+
Відображення власних пристроїв для фіксації занять бігом	+
Відображення деталізованої статистики щодо занять бігом користувача за рік	+
Відображення активностей з занять бігом користувача	+
Відображення деталізованої активності з занять бігом користувача	+
Додавання виконаної активності з занять бігом користувача	+
Оновлення виконаної активності з занять бігом користувача	+

Продовження таблиці 4.1

1	2
Додавання результату участі користувача в офіційному змаганні з бігу	+
Оновлення результату участі користувача в офіційному змаганні з бігу	+
Відображення результатів офіційного змагання з бігу за рік проведення	+
Відображення даних офіційного змагання з бігу	+
Підписування на користувача, який займається бігом	+
Відписування від користувача, який займається бігом	+
Відображення користувачів, які займаються бігом та на яких підписався даний	+
Відображення користувачів, які займаються бігом та які підписалися на даного	+
Підтвердження дружби у відповідь на підписку від користувача, який займається бігом	+
Виділення точки, яку під час заняття бігом долають користувачі	+
Пошук точок, які під час занять бігом долають користувачі	+
Пошук місць, які під час занять бігом долають користувачі	+
Відображення даних точки, яку під час занять бігом долають користувачі	+
Відображення даних місця, яке під час занять бігом долають користувачі	+
Створення відгуку про точку, яку під час занять бігом долають користувачі	+
Створення відгуку про місце, яке під час занять бігом долають користувачі	+

Продовження таблиці 4.1

1	2
Відображення відгуків за точкою, яку під час занять бігом долають користувачі	+
Відображення відгуків за місцем, яке під час занять бігом долають користувачі	+
Відображення відгуків про точки, які під час занять бігом долають користувачі та які створив цей користувач	+
Відображення відгуків про місця, які під час занять бігом долають користувачі та які створив цей користувач	+
Відмічування місця, яке під час занять бігом долають користувачі	+
Відмічування точки, яку під час занять бігом долають користувачі	+
Пошук офіційних змагань з бігу	+
Надсилання запиту за спільним заняттям з бігу	+
Підтвердження запиту за спільним заняттям з бігу	+
Скасування запиту за спільним заняттям з бігу	+
Відображення надісланих запитів за спільними заняттями з бігу	+
Відображення отриманих запитів за спільними заняттями з бігу	+
Відображення визначених спільних занять з бігу	+
Пропонування змагання для занять бігом за показниками	+
Оновлення пропонованого змагання для занять бігом за показниками	+
Надсилання запиту на участь у змаганні для занять бігом за показниками	+

Кінець таблиці 4.1

1	2
Підтвердження запиту на участь у змаганні для занять бігом за показниками	+
Скасування запиту на участь у змаганні для занять бігом за показниками	+
Відображення змагань для занять бігом за показниками, в яких бере участь користувач	+
Відображення отриманих запитів на участь у змаганні для занять бігом за показниками	+
Відображення надісланих запитів на участь у змаганні для занять бігом за показниками	+
Відображення деталізованого заняття бігом за показниками	+
Відображення деталізованого спільного заняття з бігу	+
Відображення змагань для занять бігом за показниками за користувачами з підписок	+
Відображення активностей з бігу за користувачами з підписок	+
Відображення відмічених користувачем точок	+
Відображення відмічених користувачем місць	+
Додавання запису фіксації активності з занять бігом користувача	+
Оновлення запису фіксації активності з занять бігом користувача	+
Зв'язування виконаної активності з занять бігом користувача з результатом офіційного змагання	+
Розірвання зв'язування виконаної активності з занять бігом користувача з результатом офіційного змагання	+

4.5 Висновки за розділом 4

ПС для пошуку друзів для спільних занять бігом була перевірена стосовно тих можливостей, які були реалізовані та описані в попередньому розділі. Для цього було реалізовано тестування за функціями ПС, яке при виділенні результатів за кожною функцією з використанням відповідного сценарію роботи ПС призвело до успішних результатів за кожною позицією.

ВИСНОВКИ

У виконаній роботі розроблено ПС пошуку друзів для спільних занять бігом, яка сприяє збільшенню кількості релевантних пропозицій під час пошуку друзів для спільних занять бігом за рахунок врахування цікавих любителям бігу місць. Оскільки користувачі можуть оцінювати такі місця, вказуючи на задоволення ними стосовно бігових активностей, також вони можуть відзначати окремі з цих місць, формувати з них як ключових точок маршрути для спільних занять бігом, то це дозволяє визначити інтереси любителів бігу стосовно таких локацій. Врахування ж їх як параметрів пошуку дозволяє визначити, хто з інших любителів бігу так сам позитивно оцінює ці місця та надавати таким чином в результатах пошуку перевагу таким любителям бігу. Це в підсумку має сприяти задоволеності любителів бігу стосовно використання цієї програмної системи для пошуку нових друзів для спільних занять бігом зокрема.

Програмна система побудована як сукупність об'єднаних для виконання завдання пошуку друзів для спільних занять бігом підсистем представлень, моделей, форм та шаблонів. Результати проєктування програмної системи включають виконане моделювання, визначення структури роботи з даними. Було отримано успішні результати тестування програмної системи пошуку друзів для спільних занять бігом.

Створена програмна система призначена для використання її для накопичення власних даних щодо результатів бігових активностей користувачами з одного боку, а з іншого боку для пошуку друзів для спільних занять бігом, планування таких занять бігом на основі визначення основних точок і місць для таких занять, організації віддалених занять стосовно досягнення певного рівня показників під час занять бігом протягом заданого періоду часу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Results TimeTo. URL: <https://results.timeto.com/> (date of access: 24.05.2026).
2. Find a Race. 1000's of Runs, Rides, Swims, Triathlons & More. URL: <https://findarace.com/> (date of access: 24.05.2026).
3. ITRA – International Trail Running Association – official world ranking of trail runners recognized by World Athletics. URL: <https://itra.run/> (date of access: 24.05.2026).
4. RUNALYZE – Data analysis for athletes. URL: <https://runalyze.com/#features> (date of access: 24.05.2026).
5. File Types. FIT SDK. Garmin Developers. URL: <https://developer.garmin.com/fit/file-types/activity/> (date of access: 24.05.2026).
6. Django at a glance. Django documentation. Django. URL: <https://docs.djangoproject.com/en/6.0/intro/overview/> (date of access: 25.05.2026).
7. Comparison of FastAPI with Django and Flask. *GeeksForGeeks*. URL: <https://www.geeksforgeeks.org/python/comparison-of-fastapi-with-django-and-flask/> (date of access: 25.05.2026).
8. FastAPI. URL: <https://fastapi.tiangolo.com/> (date of access: 25.05.2026).
9. Tutorial – User Guide – FastAPI. URL: <https://fastapi.tiangolo.com/tutorial/> (date of access: 25.05.2026).
10. Free Template 622 Clearwave. URL: <https://templatemo.com/tm-622-clearwave> (date of access: 28.05.2026).

ДОДАТОК А
Технічне завдання

Вступ

Заняття бігом є доволі популярними в Україні. Частина людей займається самостійно, а частина потребує партнерів. Відповідно частина людей, які займаються бігом знаходяться в пошуку друзів для спільних занять бігом. А ще частина людей не займається бігом через те, що не має партнерів. Тобто такі люди потребують підтримки. З іншого боку процес пошуку друзів полягає не у встановленні контакту з випадковим любителем бігу, а в тому, щоб врахувати рівень такого любителя бігу стосовно бігових активностей та врахувати інтереси. Перший фактор виражається в тому, що в програмній системі, яка використовується для підтримки пошуку друзів для спільних занять бігом необхідно забезпечити накопичення даних про минулі бігові активності. Другий фактор виражається в тому, що люди схильні бігати в різних цікавих їм місцях. Для когось це може бути стадіон, для когось парк, а для когось тривалий маршрут з різними точками на ньому, кожна з яких несе певну цікавість.

Програмної системи, яка б враховувала обидва ці фактори в повній мірі, на даний момент не існує. Тому потрібно розробити в цій роботі таку програмну систему для пошуку друзів для спільних занять бігом.

A.1 Підстави для розробки

Розробка виконується за темою дипломної кваліфікаційної роботи «Створення програмної системи пошуку друзів для спільних занять бігом», визначеною Наказом № 139 від 7.04.2026 року в Національному університеті «Запорізька політехніка».

A.2 Призначення розробки

Програмна система призначена для використання її для накопичення

власних даних щодо результатів бігових активностей користувачами з одного боку, а з іншого боку для пошуку друзів для спільних занять бігом, планування таких занять бігом на основі визначення основних точок і місць для таких занять, організації віддалених занять стосовно досягнення певного рівня показників під час занять бігом протягом заданого періоду часу.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Серед функцій програмної системи пошуку друзів для спільних занять бігом необхідно забезпечити:

- реєстрація як користувача, який займається бігом;
- реєстрація власного пристрою для фіксації занять бігом;
- відображення профілю користувача, який займається бігом;
- пошук серед користувачів, які займаються бігом;
- оновлення власного профілю користувача, який займається бігом;
- авторизація;
- відображення власних пристроїв для фіксації занять бігом;
- відображення деталізованої статистики щодо занять бігом користувача за рік;
- відображення активностей з занять бігом користувача;
- відображення деталізованої активності з занять бігом користувача;
- додавання виконаної активності з занять бігом користувача;
- оновлення виконаної активності з занять бігом користувача;
- додавання запису фіксації активності з занять бігом користувача;
- оновлення запису фіксації активності з занять бігом користувача;
- зв'язування виконаної активності з занять бігом користувача з результатом офіційного змагання;
- розірвання зв'язування виконаної активності з занять бігом користувача з результатом офіційного змагання;

- додавання результату участі користувача в офіційному змаганні з бігу;
- оновлення результату участі користувача в офіційному змаганні з бігу;
- відображення результатів офіційного змагання з бігу за рік проведення;
- відображення даних офіційного змагання з бігу;
- підписування на користувача, який займається бігом;
- відписування від користувача, який займається бігом;
- відображення користувачів, які займаються бігом та на яких підписався даний;
- відображення користувачів, які займаються бігом та які підписалися на даного;
- підтвердження дружби у відповідь на підписку від користувача, який займається бігом;
- виділення точки, яку під час заняття бігом долають користувачі;
- пошук точок, які під час занять бігом долають користувачі;
- пошук місць, які під час занять бігом долають користувачі;
- відображення даних точки, яку під час занять бігом долають користувачі;
- відображення даних місця, яке під час занять бігом долають користувачі;
- створення відгуку про точку, яку під час занять бігом долають користувачі;
- створення відгуку про місце, яке під час занять бігом долають користувачі;
- відображення відгуків за точкою, яку під час занять бігом долають користувачі;
- відображення відгуків за місцем, яке під час занять бігом долають користувачі;

- відображення відгуків про точки, які під час занять бігом долають користувачі та які створив цей користувач;
- відображення відгуків про місця, які під час занять бігом долають користувачі та які створив цей користувач;
- відмічування місця, яке під час занять бігом долають користувачі;
- відмічування точки, яку під час занять бігом долають користувачі;
- пошук офіційних змагань з бігу;
- надсилання запиту за спільним заняттям з бігу;
- підтвердження запиту за спільним заняттям з бігу;
- скасування запиту за спільним заняттям з бігу;
- відображення надісланих запитів за спільними заняттями з бігу;
- відображення отриманих запитів за спільними заняттями з бігу;
- відображення визначених спільних занять з бігу;
- пропонування змагання для занять бігом за показниками;
- оновлення пропонованого змагання для занять бігом за показниками;
- надсилання запиту на участь у змаганні для занять бігом за показниками;
- підтвердження запиту на участь у змаганні для занять бігом за показниками;
- скасування запиту на участь у змаганні для занять бігом за показниками;
- відображення змагань для занять бігом за показниками, в яких бере участь користувач;
- відображення отриманих запитів на участь у змаганні для занять бігом за показниками;
- відображення надісланих запитів на участь у змаганні для занять бігом за показниками;
- відображення деталізованого заняття бігом за показниками;
- відображення деталізованого спільного заняття з бігу;
- відображення змагань для занять бігом за показниками за

користувачами з підписок;

- відображення активностей з бігу за користувачами з підписок;
- відображення відмічених користувачем точок;
- відображення відмічених користувачем місць.

А.3.2 Умови експлуатації

Умови виконання програмної системи пошуку друзів для спільних занять бігом включають необхідність виділення сервера, де має бути забезпечено підтримку технологій, які будуть обрані в роботі, та зокрема бази даних.

А.3.3 Вимоги до складу та параметрів технічних засобів

Технічні засоби, необхідні для організації роботи серверу, включають вимоги до процесору щодо наявності хоча би 2 ядр, оперативної пам'яті, вивільненої для підтримки роботи програмної системи пошуку друзів для спільних занять бігом у обсязі хоча би 4 Гб, та щодо жорсткого диску стосовно наявності на ньому хоча би 50 Гб вільного простору.

А.4 Порядок контролю та приймання

Дипломна кваліфікаційна робота має бути виконана за 8 тижнів, а вимоги до окремих етапів виконання роботи до проведення захисту приводяться у календарному плані в завданні на роботу.

ДОДАТОК Б
Текст програми

```
from .models import Runningdevicemanufacturer,
    Runningdevicetype,
    Runningdevice,
    Runningpartner,
    Runningpartnerdevice,
    Runningpartnerstatistic,
    Runningactivitytype,
    Runningactivity,
    Runningrecord,
    Runningcompetitiontype,
    Runningcountry,
    Runningcity,
    Runningcompetition,
    Runningcompetitionyear,
    Runningcompetitionresult,
    Runningpartnerfollow,
    Runningpoint,
    Runningplace,
    Runningplacetype,
    Runningplacereview,
    Runningplacereviewphoto,
    Runningpointreview,
    Runningpointreviewphoto,
    Runningplacesave,
    Runningpointsave,
    Runningrequest,
    Runninggoal,
    Runninggoalpartner
from django.shortcuts import render
from django.core.paginator import Paginator
```

```

def runningpartner_view(request, runningpartner_login = None):
    if not request.user.is_authenticated:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Потрібна авторизація для роботи."
            }
        )
    if runningpartner_login:
        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount__username = runningpartner_login)
        auth_running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
        if auth_running_partner:
            auth_running_partner = auth_running_partner[0]
        else:
            return render(
                request,
                "dorunnerapp/error.html",
                {
                    "error": "Потрібна авторизація для роботи."
                }
            )
    else:
        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)

```

```

        auth_running_partner = running_partner[0]
    if not running_partner:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Не вдалось ідентифікувати такого
користувача."
            }
        )
    else:
        running_partner = running_partner[0]
        running_partner_devices =
Runningpartnerdevice.objects.select_related("runningpartner").select_related("run
ningdevice").select_related("runningdevice__runningdevicemanufacturer").select_
related("runningdevice__runningdevicetype").filter(runningpartner =
running_partner)[:3]
        running_partner_statistic =
Runningpartnerstatistic.objects.select_related("runningpartner").filter(runningpartn
er = running_partner).order_by("-runningpartnerstatisticyear")
        running_partner_activities =
Runningactivity.objects.select_related("runningactivitytype").select_related("runni
ngpartnerdevice").select_related("runningpartnerdevice__runningpartner").filter(ru
nningpartnerdevice__runningpartner = running_partner).order_by("-
runningactivityendtime")[:5]
        running_partner_competitions =
Runningcompetitionresult.objects.select_related("runningpartner").select_related("
runningcompetitionyear").select_related("runningcompetitionyear__runningcompe
tition").select_related("runningcompetitionyear__runningcompetition__runningco
mpetitiontype").select_related("runningcompetitionyear__runningcompetition__ru

```

```

nningcompetitioncity").filter(runningpartner = running_partner).order_by("-
runningcompetitionyear__runningcompetitiondate")[:3]

    running_partner_placereviews =
Runningplacereview.objects.select_related("runningplace").select_related("running
partner").filter(runningpartner = running_partner).order_by("-
runningplacereviewtime")[:3]

    for running_partner_placereview in running_partner_placereviews:
        running_partner_placereview.photos =
Runningplacereviewphoto.objects.select_related("runningplacereview").filter(runn
ingplacereview = running_partner_placereview)
        running_partner_placereview.save =
Runningplacesave.objects.select_related("runningplace").select_related("runningpa
rtner").filter(runningplace = running_partner_placereview.runningplace,
runningpartner = auth_running_partner)
        running_partner_pointreviews =
Runningpointreview.objects.select_related("runningpoint").select_related("running
partner").filter(runningpartner = running_partner).order_by("-
runningpointreviewscore "):[:3]

    for running_partner_pointreview in running_partner_pointreviews:
        running_partner_pointreview.photos =
Runningpointreview.objects.select_related("runningpointreview").filter(runningpoi
ntreview = running_partner_pointreview)
        running_partner_pointreview.save =
Runningpointsave.objects.select_related("runningpoint").select_related("runningpa
rtner").filter(runningpoint = running_partner_pointreview.runningpoint,
runningpartner = auth_running_partner)
        running_partner_follows =
Runningpartnerfollow.objects.select_related("runningpartnerfollower").select_relat
ed("runningpartner").filter(runningpartnerfollower = running_partner)[:3]

```

```

        running_partner_followers =
Runningpartnerfollow.objects.select_related("runningpartnerfollower").select_relat
ed("runningpartner").filter(runningpartner = running_partner)[:3]

        running_partner_saved_points =
Runningpointsave.objects.select_related("runningpoint").select_related("runningpa
rtner").filter(runningsavetype = True, runningpartner = running_partner)[:3]

        running_partner_saved_places =
Runningplacesave.objects.select_related("runningplace").select_related("runningpa
rtner").filter(runningsavetype = True, runningpartner = running_partner)[:3]

        import datetime

        running_partner_goals_now = datetime.datetime.now()

        running_partner_goals =
Runninggoal.objects.select_related("runningpartner").filter(runninggoalstarttime__
gte = running_partner_goals, runningpartner = running_partner)

    return render(
        request,
        "dorunnerapp/runner.html",
        {
            "runner": running_partner,
            "devices": running_partner_devices,
            "statiscitics": running_partner_statiscitics,
            "activities": running_partner_activities,
            "competitions": running_partner_competitions,
            "placereviews": running_partner_placereviews,
            "pointreviews": running_partner_pointreviews,
            "follows": running_partner_follows,
            "followers": running_partner_followers,
            "points": running_partner_saved_points,
            "places": running_partner_saved_places,
            "goals": running_partner_goals
        }
    )

```

```

        }
    )

def runningpartner_saved_points_view(request, runningpartner_login =
None):
    if not request.user.is_authenticated:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Потрібна авторизація для роботи."
            }
        )
    if runningpartner_login:
        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount__username = runningpartner_login)
        auth_running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
        if auth_running_partner:
            auth_running_partner = auth_running_partner[0]
        else:
            return render(
                request,
                "dorunnerapp/error.html",
                {
                    "error": "Потрібна авторизація для роботи."
                }
            )

```

```

else:
    running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
    auth_running_partner = running_partner[0]
if not running_partner:
    return render(
        request,
        "dorunnerapp/error.html",
        {
            "error": "Не вдалось ідентифікувати такого
користувача."
        }
    )
else:
    running_partner = running_partner[0]
    if runningpartner_login:
        running_partner_saved_points =
Runningpointsave.objects.select_related("runningpoint").select_related("runningpa
rtner").filter(runningsavetype = True, runningpartner = running_partner)
    else:
        running_partner_saved_points =
Runningpointsave.objects.select_related("runningpoint").select_related("runningpa
rtner").filter(runningpartner = running_partner)
        running_partner_saved_points =
Paginator(running_partner_saved_points, 10)
        running_partner_saved_points_current_page =
request.GET.get('page')

```

```

        running_partner_saved_points =
running_partner_saved_points.get_page(running_partner_saved_points_current_pa
ge)

        for running_partner_saved_point in running_partner_saved_points:

            running_partner_saved_point.save =
Runningpointsave.objects.select_related("runningpoint").select_related("runningpa
rtner").filter(runningpoint = running_partner_saved_point.runningpoint,
runningpartner = auth_running_partner)

        return render(
            request,
            "dorunnerapp/runnerpoints.html",
            {
                "runner": running_partner,
                "points": running_partner_saved_points
            }
        )

def runningpartner_saved_places_view(request, runningpartner_login =
None):
    if not request.user.is_authenticated:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Потрібна авторизація для роботи."
            }
        )
    if runningpartner_login:

```

```

        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount__username = runningpartner_login)

        auth_running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)

        if auth_running_partner:
            auth_running_partner = auth_running_partner[0]
        else:
            return render(
                request,
                "dorunnerapp/error.html",
                {
                    "error": "Потрібна авторизація для роботи."
                }
            )
    else:
        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
        auth_running_partner = running_partner[0]
    if not running_partner:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Не вдалось ідентифікувати такого
користувача."
            }
        )

```

```

else:
    running_partner = running_partner[0]
    if runningpartner_login:
        running_partner_saved_places =
Runningplacesave.objects.select_related("runningplace").select_related("runningpa
rtner").filter(runningsavetype = True, runningpartner = running_partner)
    else:
        running_partner_saved_places =
Runningplacesave.objects.select_related("runningplace").select_related("runningpa
rtner").filter(runningpartner = running_partner)
        running_partner_saved_places =
Paginator(running_partner_saved_places, 10)
        running_partner_saved_places_current_page =
request.GET.get('page')
        running_partner_saved_places =
running_partner_saved_places.get_page(running_partner_saved_places_current_pa
ge)

        for running_partner_saved_place in running_partner_saved_places:
            running_partner_saved_place.save =
Runningplacesave.objects.select_related("runningplace").select_related("runningpa
rtner").filter(runningplace = running_partner_saved_place.runningplace,
runningpartner = auth_running_partner)
        return render(
            request,
            "dorunnerapp/runnerplaces.html",
            {
                "runner": running_partner,
                "places": running_partner_saved_places
            }
        )

```

```

def runningpartner_follows_view(request, runningpartner_login = None):
    if not request.user.is_authenticated:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Потрібна авторизація для роботи."
            }
        )
    if runningpartner_login:
        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount__username = runningpartner_login)
    else:
        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
    if not running_partner:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Не вдалось ідентифікувати такого
користувача."
            }
        )
    else:
        running_partner = running_partner[0]

```

```

        running_partner_follows =
Runningpartnerfollow.objects.select_related("runningpartnerfollower").select_relat
ed("runningpartner").filter(runningpartnerfollower = running_partner)
        running_partner_follows = Paginator(running_partner_follows, 10)
        running_partner_follows_current_page = request.GET.get('page')
        running_partner_follows =
running_partner_follows.get_page(running_partner_follows_current_page)
        return render(
            request,
            "dorunnerapp/runnerfollows.html",
            {
                "runner": running_partner,
                "follows": running_partner_follows
            }
        )

def runningpartner_followers_view(request, runningpartner_login = None):
    if not request.user.is_authenticated:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Потрібна авторизація для роботи."
            }
        )
    if runningpartner_login:
        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount__username = runningpartner_login)
    else:

```

```

        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
        if not running_partner:
            return render(
                request,
                "dorunnerapp/error.html",
                {
                    "error": "Не вдалось ідентифікувати такого
користувача."
                }
            )
        else:
            running_partner = running_partner[0]
            running_partner_followers =
Runningpartnerfollow.objects.select_related("runningpartnerfollower").select_relat
ed("runningpartner").filter(runningpartner = running_partner)
            running_partner_followers = Paginator(running_partner_followers,
10)
            running_partner_followers_current_page = request.GET.get('page')
            running_partner_followers =
running_partner_followers.get_page(running_partner_followers_current_page)
            return render(
                request,
                "dorunnerapp/runnerfollowers.html",
                {
                    "runner": running_partner,
                    "follows": running_partner_followers
                }
            )

```

```

def runningpartner_devices_view(request):
    if not request.user.is_authenticated:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Потрібна авторизація для роботи."
            }
        )
    running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
    if not running_partner:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Не вдалось ідентифікувати такого
користувача."
            }
        )
    else:
        running_partner = running_partner[0]
        running_partner_devices =
Runningpartnerdevice.objects.select_related("runningpartner").select_related("run
ningdevice").select_related("runningdevice__runningdevicemanufacturer").select_
related("runningdevice__runningdevicetype").filter(runningpartner =
running_partner)
        running_partner_devices = Paginator(running_partner_devices, 10)
        running_partner_devices_current_page = request.GET.get('page')

```

```

        running_partner_devices =
running_partner_devices.get_page(running_partner_devices_current_page)
        return render(
            request,
            "dorunnerapp/runnerdevices.html",
            {
                "runner": running_partner,
                "devices": running_partner_devices
            }
        )

```

```

def runningpartner_activities_view(request, runningpartner_login = None):
    if not request.user.is_authenticated:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Потрібна авторизація для роботи."
            }
        )
    if runningpartner_login:
        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount__username = runningpartner_login)
    else:
        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
    if not running_partner:
        return render(

```

```

        request,
        "dorunnerapp/error.html",
        {
            "error": "Не вдалось ідентифікувати такого
користувача."
        }
    )
else:
    running_partner = running_partner[0]
    running_partner_activities =
Runningactivity.objects.select_related("runningactivitytype").select_related("runni
ngpartnerdevice").select_related("runningpartnerdevice__runningpartner").filter(ru
nningpartnerdevice__runningpartner = running_partner).order_by("-
runningactivityendtime")
    running_partner_activities = Paginator(running_partner_activities, 10)
    running_partner_activities_current_page = request.GET.get('page')
    running_partner_activities =
running_partner_activities.get_page(running_partner_activities_current_page)
    return render(
        request,
        "dorunnerapp/runneractivities.html",
        {
            "runner": running_partner,
            "activities": running_partner_activities
        }
    )

def runningpartner_points_reviews_view(request, runningpartner_login =
None):
    if not request.user.is_authenticated:

```

```

        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Потрібна авторизація для роботи."
            }
        )
    if runningpartner_login:
        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount__username = runningpartner_login)
        auth_running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
        if auth_running_partner:
            auth_running_partner = auth_running_partner[0]
        else:
            return render(
                request,
                "dorunnerapp/error.html",
                {
                    "error": "Потрібна авторизація для роботи."
                }
            )
    else:
        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
        auth_running_partner = running_partner[0]
    if not running_partner:

```

```

        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Не вдалось ідентифікувати такого
користувача."
            }
        )
    else:
        running_partner = running_partner[0]
        running_partner_pointreviews =
Runningpointreview.objects.select_related("runningpoint").select_related("running
partner").filter(runningpartner = running_partner).order_by("-
runningpointreviewscore ")
        running_partner_pointreviews =
Paginator(running_partner_pointreviews, 10)
        running_partner_pointreviews_current_page = request.GET.get('page')
        running_partner_pointreviews =
running_partner_pointreviews.get_page(running_partner_pointreviews_current_pa
ge)

        for running_partner_pointreview in running_partner_pointreviews:
            running_partner_pointreview.photos =
Runningpointreview.objects.select_related("runningpointreview").filter(runningpoi
ntreview = running_partner_pointreview)
            running_partner_pointreview.save =
Runningpointsave.objects.select_related("runningpoint").select_related("runningpa
rtner").filter(runningpoint = running_partner_pointreview.runningpoint,
runningpartner = auth_running_partner)
        return render(
            request,

```

```

        "dorunnerapp/runnerpointreviews.html",
        {
            "runner": running_partner,
            "pointreviews": running_partner_pointreviews
        }
    )

def runningpartner_places_reviews_view(request, runningpartner_login =
None):
    if not request.user.is_authenticated:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Потрібна авторизація для роботи."
            }
        )
    if runningpartner_login:
        running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount__username = runningpartner_login)
        auth_running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
        if auth_running_partner:
            auth_running_partner = auth_running_partner[0]
        else:
            return render(
                request,
                "dorunnerapp/error.html",

```

```

        {
            "error": "Потрібна авторизація для роботи."
        }
    )

else:
    running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
    auth_running_partner = running_partner[0]
if not running_partner:
    return render(
        request,
        "dorunnerapp/error.html",
        {
            "error": "Не вдалось ідентифікувати такого
користувача."
        }
    )

else:
    running_partner = running_partner[0]
    running_partner_placereviews =
Runningplacereview.objects.select_related("runningplace").select_related("running
partner").filter(runningpartner = running_partner).order_by("-
runningplacereviewtime")
    running_partner_placereviews =
Paginator(running_partner_placereviews, 10)
    running_partner_placereviews_current_page =
request.GET.get('page')
```

```

        running_partner_placereviews =
running_partner_placereviews.get_page(running_partner_placereviews_current_pa
ge)

        for running_partner_placereview in running_partner_placereviews:

            running_partner_placereview.photos =
Runningplacereviewphoto.objects.select_related("runningplacereview").filter(runn
ingplacereview = running_partner_placereview)

            running_partner_placereview.save =
Runningplacesave.objects.select_related("runningplace").select_related("runningpa
rtner").filter(runningplace = running_partner_placereview.runningplace,
runningpartner = auth_running_partner)

        return render(
            request,
            "dorunnerapp/runnerplacesreviews.html",
            {
                "runner": running_partner,
                "placereviews": running_partner_placereviews
            }
        )

def runningplace_reviews_view(request, runningplace_id):
    if not request.user.is_authenticated:
        return render(
            request,
            "dorunnerapp/error.html",
            {
                "error": "Потрібна авторизація для роботи."
            }
        )
    else:

```

```

        auth_running_partner =
Runningpartner.objects.select_related("runningpartneraccount").filter(runningpartn
eraccount = request.user)
        if auth_running_partner:
            auth_running_partner = auth_running_partner[0]
        else:
            return render(
                request,
                "dorunnerapp/error.html",
                {
                    "error": "Потрібна авторизація для роботи."
                }
            )
        running_place =
Runningplace.objects.select_related("runningplacetype").filter(id =
runningplace_id)
        if not running_place:
            return render(
                request,
                "dorunnerapp/error.html",
                {
                    "error": "Не вдалось ідентифікувати таке місце для
занять бігом."
                }
            )
        else:
            running_place = running_place[0]
            running_placereviews =
Runningplacereview.objects.select_related("runningplace").select_related("running

```

```

partner").filter(runningplace = running_place).order_by("-
runningplacereviewtime")

    running_placereviews = Paginator(running_placereviews, 10)
    running_placereviews_current_page = request.GET.get('page')
    running_placereviews =
running_placereviews.get_page(running_placereviews_current_page)
    for running_placereview in running_placereviews:
        running_placereview.photos =
Runningplacereviewphoto.objects.select_related("runningplacereview").filter(runn
ingplacereview = running_placereview)
        running_partner_placereview.save =
Runningplacesave.objects.select_related("runningplace").select_related("runningpa
rtner").filter(runningplace = running_partner_placereview.runningplace,
runningpartner = auth_running_partner)
    return render(
        request,
        "dorunnerapp/runnerplacereviews.html",
        {
            "place": running_place,
            "placereviews": running_placereviews
        }
    )

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
кафедра програмних засобів

Тема дипломної кваліфікаційної роботи бакалавра:
Створення програмної системи пошуку друзів
для спільних занять бігом
Design of a Data Processing Software System for
Finding Running Partners

Виконавець:

Платон МЕДВЕДЄВ

Керівник: к.т.н.

Олена ШИТІКОВА

2026

Рисунок В.1 – Початковий слайд

Об'єкт, предмет, мета роботи

- Об'єкт роботи – процес побудови програмних систем пошуку друзів для спільних занять бігом.
- Предмет роботи – програмні системи пошуку друзів для спільних занять бігом.
- Мета роботи – збільшення кількості релевантних пропозицій під час пошуку друзів для спільних занять бігом за рахунок врахування цікавих любителям бігу місць за допомогою програмної системи.

2

Рисунок В.2 – Об'єкт, предмет, мета роботи

Завдання роботи

- аналіз проблеми розробки програмної системи пошуку друзів для спільних занять бігом;
- проєктування програмної системи пошуку друзів для спільних занять бігом;
- розробка програмної системи пошуку друзів для спільних занять бігом.

3

Рисунок В.3 – Завдання роботи

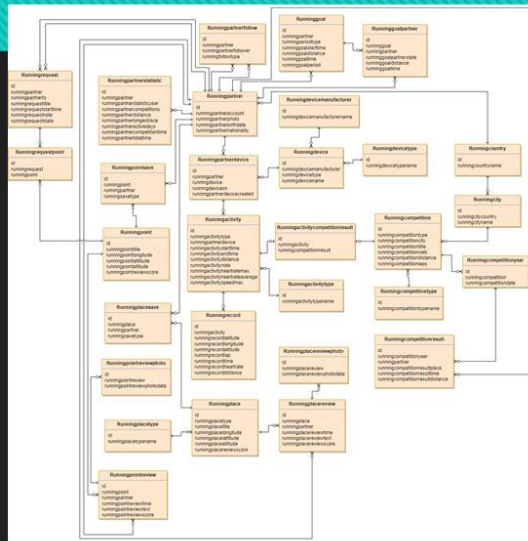
Порівняння програм для забезпечення пошуку друзів для спільних занять бігом

Можливість	ITRA	Runalyze	Розробка
Історія офіційних змагань з бігу	+	+	+
Історія звичайних бігових активностей	–	+	+
Пошук користувачів	Спортсмени	–	Любителі бігу
Визначення улюблених місць для бігу	–	–	+
Створення спільного заняття з бігу	–	–	+
Створення змагання за показниками	–	–	+

4

Рисунок В.4 – Порівняння програм для забезпечення пошуку друзів для спільних занять бігом

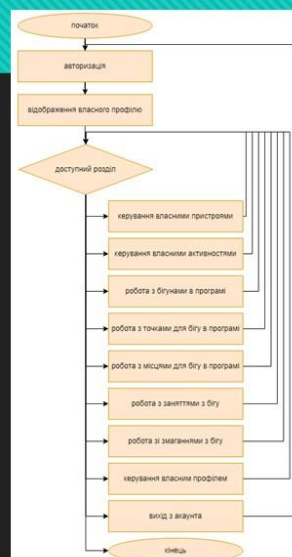
Схема бази даних програмної системи пошуку друзів для спільних занять бігом



7

Рисунок В.7 – Схема бази даних програмної системи пошуку друзів для спільних занять бігом

Функціональна схема програмної системи пошуку друзів для спільних занять бігом



8

Рисунок В.8 – Функціональна схема програмної системи пошуку друзів для спільних занять бігом

Відображення профілю користувача, який займається бігом

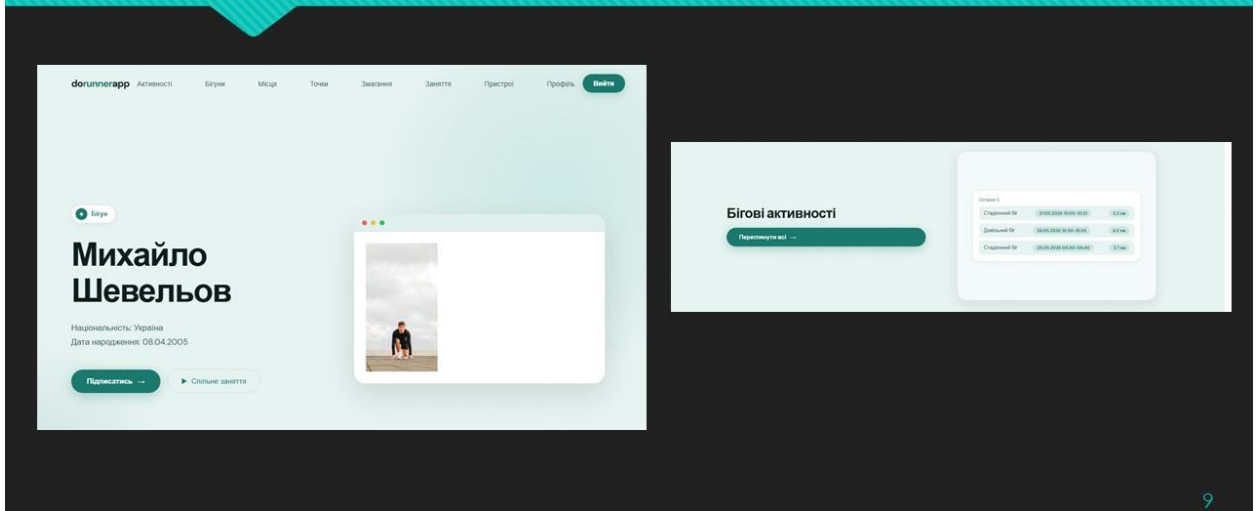


Рисунок В.9 – Відображення профілю користувача, який займається бігом

Відображення користувачів, які займаються бігом та які підписалися на даного, та відгуків за місцем, яке під час занять бігом долають користувачі

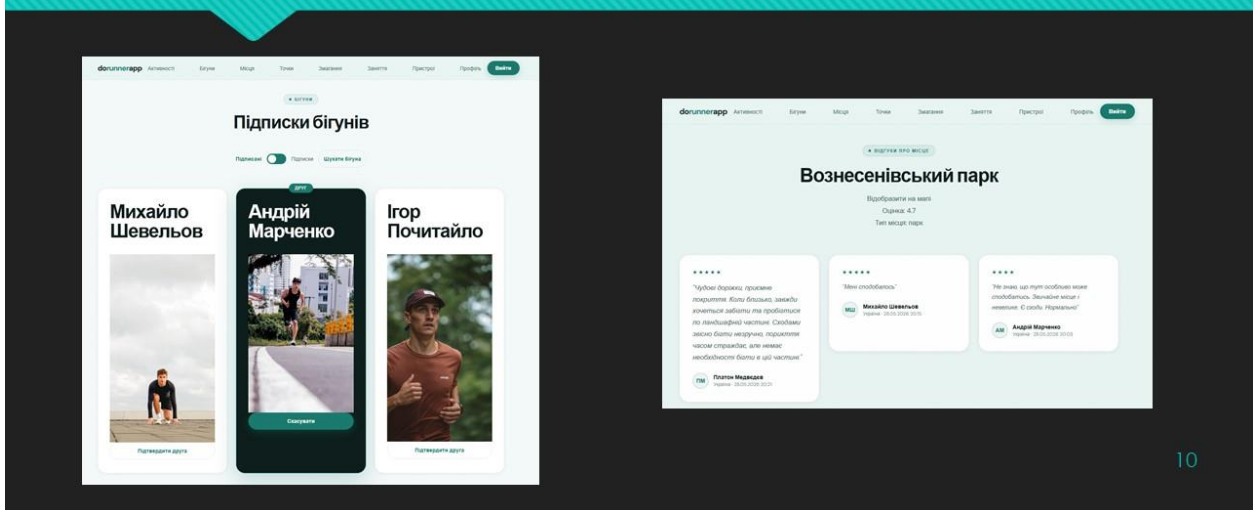


Рисунок В.10 – Відображення користувачів, які займаються бігом та які підписалися на даного, та відгуків за місцем, яке під час занять бігом долають користувачі

Тестування програмної системи для пошуку друзів для спільних занять бігом

Функція під час тестування	Результат тестування	Функція під час тестування	Результат тестування	Функція під час тестування	Результат тестування		
Функція під час тестування	Результат тестування	Додавання результату участі користувача в офіційному змаганні з бігу	+	Відображення відгуку за точкою, яку під час занять бігом додають користувачі	+	Підтвердження запиту на участь у змаганні для занять бігом за показниками	+
Регістрація як користувача, який займається бігом	+	Оновлення результату участі користувача в офіційному змаганні з бігу	+	Відображення відгуку за місцем, яке під час занять бігом додають користувачі	+	Скасування запиту на участь у змаганні для занять бігом за показниками	+
Регістрація власного пристрою для фіксації занять бігом	+	Відображення результатів офіційного змагання з бігу за рік проведення	+	Відображення відгуку про точки, які під час занять бігом додають користувачі та які створив цей користувач	+	Відображення змагань для занять бігом за показниками, в яких бере участь користувач	+
Відображення профілю користувача, який займається бігом	+	Відображення даних офіційного змагання з бігу	+	Відображення відгуку про місце, яке під час занять бігом додають користувачі та які створив цей користувач	+	Відображення отриманих запитів на участь у змаганні для занять бігом за показниками	+
Пошук серед користувачів, які займаються бігом	+	Підписування на користувача, який займається бігом	+	Відмінування місця, яке під час занять бігом додають користувачі	+	Відображення деталізованого змагання з бігу	+
Оновлення власного профілю користувача, який займається бігом	+	Відписування від користувача, який займається бігом	+	Відмінування точки, яку під час занять бігом додають користувачі	+	Відображення змагань для занять бігом за показниками за користувачами з підписок	+
Авторизація	+	Відображення користувачів, які займаються бігом та на яких підписався даний	+	Пошук офіційних змагань з бігу	+	Відображення активностей з бігу за користувачами з підписок	+
Відображення власних пристроїв для фіксації занять бігом	+	Відображення користувачів, які займаються бігом та які підписалися на даного	+	Підтвердження запиту за спільним заняттям з бігу	+	Відображення відмінутих користувачем точок	+
Відображення деталізованої статистики щодо занять бігом користувача за рік	+	Підтвердження друзів у відповідь на підписку від користувача, який займається бігом	+	Скасування запиту за спільним заняттям з бігу	+	Відображення відмінутих користувачем місць	+
Відображення активностей з занять бігом користувача	+	Виділення точки, яку під час заняття бігом додають користувачі	+	Відображення надісланих запитів за спільним заняттям з бігу	+	Додавання запису фіксації активності з занять бігом користувача	+
Відображення деталізованої активності з занять бігом користувача	+	Пошук точок, які під час занять бігом додають користувачі	+	Відображення отриманих запитів за спільним заняттям з бігу	+	Оновлення запису фіксації активності з занять бігом користувача	+
Додавання виконаної активності з занять бігом користувача	+	Пошук місць, які під час занять бігом додають користувачі	+	Відображення виконаваних спільних занять з бігу	+	Зв'язування виконаної активності з заняттям бігом користувача з результатом офіційного змагання	+
Оновлення виконаної активності з занять бігом користувача	+	Відображення даних точки, яку під час занять бігом додають користувачі	+	Пропонування змагання для занять бігом за показниками	+	Розв'язання зв'язування виконаної активності з заняттям бігом користувача з результатом офіційного змагання	+
		Відображення даних місця, яке під час занять бігом додають користувачі	+	Оновлення пропонування змагання для занять бігом за показниками	+		
		Створення відгуку про точку, яку під час занять бігом додають користувачі	+	Надмінування запиту на участь у змаганні для занять бігом за показниками	+		
		Створення відгуку про місце, яке під час занять бігом додають користувачі	+				

11

11

Рисунок В.11 – Тестування програмної системи для пошуку друзів для спільних занять бігом

Висновки

- У виконаній роботі розроблено ПС пошуку друзів для спільних занять бігом, яка сприяє збільшенню кількості релевантних пропозицій під час пошуку друзів для спільних занять бігом за рахунок врахування цікавих любителям бігу місць. Оскільки користувачі можуть оцінювати такі місця, вказуючи на задоволення ними стосовно бігових активностей, також вони можуть відзначати окремі з цих місць, формувати з них як ключових точок маршрути для спільних занять бігом, то це дозволяє визначити інтереси любителів бігу стосовно таких локацій. Врахування ж їх як параметрів пошуку дозволяє визначити, хто з інших любителів бігу так сам позитивно оцінює ці місця та надавати таким чином в результатах пошуку перевагу таким любителям бігу. Це в підсумку має сприяти задоволеності любителів бігу стосовно використання цієї програмної системи для пошуку нових друзів для спільних занять бігом зокрема.

12

Рисунок В.12 – Висновки