

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
Факультет прикладної математики  
Кафедра прикладної математики

ПОЯСНЮВАЛЬНА ЗАПИСКА  
до курсової роботи  
із дисципліни «Бази даних та інформаційні системи»  
на тему  
«Тренування на природі»

Виконав:  
студент групи КМ-42  
Білоножко І.О.  
Керівник:  
старший викладач  
Терещенко І. О.

## ЗАВДАННЯ НА ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Курсова робота передбачає розробку цілісної робочої системи, що допомагає в деякій визначеній окремо кожним студентом області, використовуючи знання щодо клієнт-серверної архітектури та баз даних.

Згідно теми курсової роботи, а саме «Тренування на природі», розроблена система, призначена для ведення щоденника тренувань на природі та полегшення пошуку сприятливих для тренування місць завдяки наданій користувачу бази локацій для тренувань, яка може поповнюватись новими локаціями.

## АНОТАЦІЯ

В курсовій роботі розроблена цілісна робоча система, при розробці якої задіяні такі поняття як клієнт-серверна архітектура та бази даних. Розроблено Use Case діаграми, діаграму послідовності, концептуальну, логічну та фізичну діаграми «зв'язок-сутність». З технічної точки зору система присвячена розв'язанню проблеми цілісного функціонування системи, а з практичної точки зору – реалізації системи, що допомагає в веденні тренувань на природі.

## ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	5
ВСТУП .....	6
ОСНОВНА ЧАСТИНА.....	7
1 ПОСТАНОВКА ЗАДАЧІ.....	7
1.1 Категорії користувачів.....	8
1.2 Класи даних .....	8
2 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ.....	10
2.1 Scrum .....	10
2.2 Use Case.....	11
2.3 Діаграми послідовностей.....	12
3 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ .....	13
3.1 Опис сутностей.....	13
4 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ .....	16
ВИСНОВКИ.....	18
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ .....	20
ДОДАТОК А.....	20
ДОДАТОК Б .....	22

## СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

АВТОМАТИЗОВАНЕ РОБОЧЕ МІСЦЕ (АРМ) — індивідуальний комплекс технічних і програмних засобів, що призначений для автоматизації професійної праці фахівця і забезпечує підготовку, редагування, пошук і видачу на екран і друк необхідних йому документів і даних;

АВТОМАТИЗОВАНА ІНФОРМАЦІЙНА СИСТЕМА (АІС) - сукупність програмно-апаратних засобів, призначених для автоматизації діяльності, пов'язаної зі зберіганням, передачею та обробкою інформації. АІС є, з одного боку, різновидом інформаційних систем (ІС), з іншого - автоматизованих систем (АС), внаслідок чого їх часто називають ІС або АС;

АРХІТЕКТУРА КЛІЄНТ-СЕРВЕР – один із архітектурних шаблонів програмного забезпечення, що є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти: набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них; набір клієнтів, які використовують сервіси, що надаються серверами; мережа, яка забезпечує взаємодію між клієнтами та серверами;

БД – база даних;

SCRUM – це один із підходів гнучкої системи розробки, що заснована на емпіричному методі та призначений для розробки продуктів високої цінності в заплутаному середовищі; - це это набор принципов, на которых строится процесс разработки, позволяющий в жёстко фиксированные и небольшие по времени итерации, называемые спринтами (sprints), предоставлять конечному пользователю работающее ПО с новыми возможностями, для которых определён наибольший приоритет.

СПРІНТ - ітерація в скрам, в ході якої створюється функціональне зростання програмного забезпечення.

## ВСТУП

Існує близько чотирьохсот теорій старіння, але жодна з них не пояснює, чому цей процес запускається. Однак всі ми хочемо бути здоровими, сильними, молодими і жити якомога довше. Щоб цього досягти, як мінімум треба правильно харчуватися і займатися спортом. З останнім можуть допомогти не тільки платні тренажерні зали, але й самостійні тренування.

Тренування на свіжому повітрі особливо корисні і цьому є багато практичних та наукових підтверджень. Щоб притримуватись графіку тренувань досить зручно мати відповідні щоденники, а для теперішнього часу актуально мати останній в електронному виді. Такі системи можуть бути ще зручнішими завдяки додатковому функціоналу, наприклад, можливості вибору зручної локації для тренування, або пошуку фізичних вправ для конкретної групи м'язів.

Метою роботи є полегшення організації тренувань на природі за допомоги автоматизації цього процесу результатом якої є система із чітко визначеним корисним в рамках цієї предметної області функціоналом.

## ОСНОВНА ЧАСТИНА

### 1 ПОСТАНОВКА ЗАДАЧІ

В даній курсовій роботі реалізована система ведення журналу тренувань на природі, що розроблена задля більш зручної організації тренувань та економії часу для підбору зручних локацій для тренування завдяки базі локацій для тренувань на природі.

Серед основного функціоналу даної системи є наступне:

- а) створення тренування із відповідною інформацією щодо тренування, а саме локація, тип локації (наприклад, ліс, пляж, стадіон, парк), дата та опис тренування, де користувач може залишити нотатки щодо мети тренування, стилю або іншу потрібну йому інформацію;
- б) можливість перегляду локацій для тренування як списком, так і кожну локацію окремо;
- в) можливість перегляду журналу тренувань, тобто список тренувань, що вже були проведені або вже заплановані на майбутнє.

Функціонал, що описано вище, надано авторизованому користувачу. Але це буде детальніше описано далі. Цей функціонал є основним для даної системи, він задовольняє основну мету системи.

Функціонал, що є допоміжним або обслуговуючим основний функціонал, наведено в наступному переліку:

- а) авторизація користувача;
- б) реєстрація користувача (додавання користувача до БД);
- в) редагування та видалення користувача;
- г) додавання та видалення локації до БД;
- д) видалення тренувань з БД.

Не весь перелічений функціонал є доступним будь-якому користувачу. Кожен користувач має свій дозволений набір функцій. Це детальніше описано в наступному підрозділі.

## 1.1 Категорії користувачів

В системі визначено наступні групи користувачів:

- а) неавторизований користувач;
- б) авторизований користувач;
- в) адміністратор.

Користувачів кожного виду може бути один і більше, тобто маємо три групи користувачів.

Неавторизований користувач має наступні можливості:

- а) перегляд списку локацій для тренувань із обмеженою інформацією;
- б) реєстрація/авторизація в системі.

Авторизований користувач має ширший спектр функцій:

- а) перегляд списку локацій;
- б) перегляд окремої локації;
- в) додавання тренування;
- г) перегляд журналу (списку) тренувань, що були та будуть проведені.

Адміністратор окрім функціоналу авторизованого користувача має доступ до наступного функціоналу:

- а) додавання, редагування та видалення користувача;
- б) додавання, редагування та видалення локації;
- в) додавання, редагування та видалення тренування.

При виконанні етапу курсової роботи отримано досвід аналізу користувачів та ролей у інформаційній системі на прикладі інформаційної системи «Тренування на природі».

## 1.2 Класи даних

Система працює з великою кількістю даних. Для функціонування системи треба було визначити з якими даними буде працювати система, зокрема які дані потрібні в базі даних. Було створено наступні основні таблиці (в БД таблиці мають інші назви, але для більш сприйнятливого розуміння в цьому підрозділі використовуються наступні назви):

- а) користувачі;
- б) локації для тренувань;



- в) типи локацій для тренувань;
- г) тренування.

Кожна таблиця БД представляється як сукупність рядків і стовпців, де рядки відповідають екземпляру об'єкта, конкретної події (тобто окремому запису), а стовпці - атрибутам (ознаками, характеристиками, параметрами) об'єкта, події. Під об'єктами можемо розуміти в даному випадку конкретного користувача, конкретну локацію, конкретний тип локації, під подією – конкретне тренування користувача.

Отже, користувач має наступні атрибути:

- а) електронна адреса;
- б) ім'я;
- в) фамілія;
- г) дата народження;
- д) номер телефону;
- е) роль;
- ж) статус «видаленого користувача» чи ні.

Останні два атрибути є важливими для функціонування системи, але для користувача системи вони не мають ніякого сенсу. Вони існують для того, щоб визначати яким функціоналом наділяти авторизованого користувача (як звичайного користувача чи як адміністратора) та чи існує такий користувач серед активних (невидалених), бо з таблиці з користувачами насправді користувачі не видаляються в цілях безпеки та цілісності БД, їм лише може присвоюватись статус видаленого користувача.

Таблиця з локаціями для тренувань має наступні атрибути:

- а) назва локації;
- б) тип локації;
- в) адреса локації;
- г) площа локації;
- д) плата за вхід на локацію;
- е) наявність тренажерів;
- ж) наявність водойми;
- з) наявність басейну;
- и) опис до локації.

Таблиця з типами локацій складається з одного атрибуту та вміщує в себе різні типи, наприклад, «пляж», «стадіон», «поле», «парк», «басейн» і т.д.

Таблиця з тренуваннями складається з полів:

- а) користувач, що призначив тренування;

- б) локація;
- в) дата;
- г) опис до тренування.

## 2 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ

### 2.1 Scrum

Ті, хто займається управлінням проектами, та й просто управлінням, добре знають, наскільки складно організувати злагоджену роботу. Через відсутність злагодженості постійно порушуються плани, відбувається відставання від графіка, завдання різних підрозділів дублюються.

Методика Scrum покликана вирішити всі ці проблеми. Scrum - це протилежність класичного поетапного підходу, що застосовується у реалізації проектів. Проект розбито на відповідні спринти, що зображені у Додатку А на рисунку А.1, рисунку А.2 та рисунку А.3.

Перший спринт (рисунок А.1) за планом формується з реалізації функціоналу щодо локацій та тренувань – їх додавання, редагування та видання, що є основним функціоналом та основною ідеєю системи.

Спринти складаються з тасків (іншими словами, завдань), і кожен має свої обмеження.

Окремо розпишемо кожен таск.

а) Додавання локації:

- 1) Назва локації, адреса, тип локації: будь-які символи; кількість символів до 50 включно; Опис: до 2000; Тип локації – комбобокс;
- 2) Площа локації: float;
- 3) Наявність басейну, водоймів, тренажерів, доріжок: в БД можуть мати значення 0 (не наявні) або 1 (наявні), використовується комбобокс;
- 4) Кнопка “Save” відкликається на наведення курсором – з’являється рамочка.

б) Відображення списку локацій:

- 1) Назва локації, адреса: будь-які символи; кількість символів до 50 включно;
- 2) Посилання «view» на детальний перегляд локації;

в) Відображення локації:

- 1) Назва локації, адреса, тип локації: будь-які символи; кількість символів до 50 включно; Опис: до 2000; При натисканні на
- 2) Наявність басейну, водоймів, тренажерів, доріжок: в БД можуть мати значення 0 (не наявні) або 1 (наявні), використовується комбобокс; На клієнті це два строкові значення «yes» або «no». При редагуванні – комбобокс із варіантами «yes» та «no».
- 3) Площа локації: float;
- 4) Кнопка «Edit info» реагує на натискання (градієнт).

г) Видалення, редагування локації:

- 5) Назва локації, адреса, тип локації: будь-які символи; кількість символів до 50 включно; Опис: до 2000; При натисканні на
- 6) Наявність басейну, водоймів, тренажерів, доріжок: в БД можуть мати значення 0 (не наявні) або 1 (наявні), використовується комбобокс; На клієнті це два строкові значення «yes» або «no». При редагуванні – комбобокс із варіантами «yes» та «no».
- 7) Площа локації: float;
- 8) Кнопка «Edit info» реагує на натискання (градієнт).

## 2.2 Use Case

Розроблені при виконанні даного етапу курсової роботи Use Case UML діаграми складаються з наступних елементів:

- а) актор;
- б) процес;
- в) перехід.

Актор – це об'єкт ІС, що виконує первні процеси і поділяється на:

- master (адміністратори);
- slave (решта).

Процес описує інтерфейс, що надає користувачу певний функціонал.

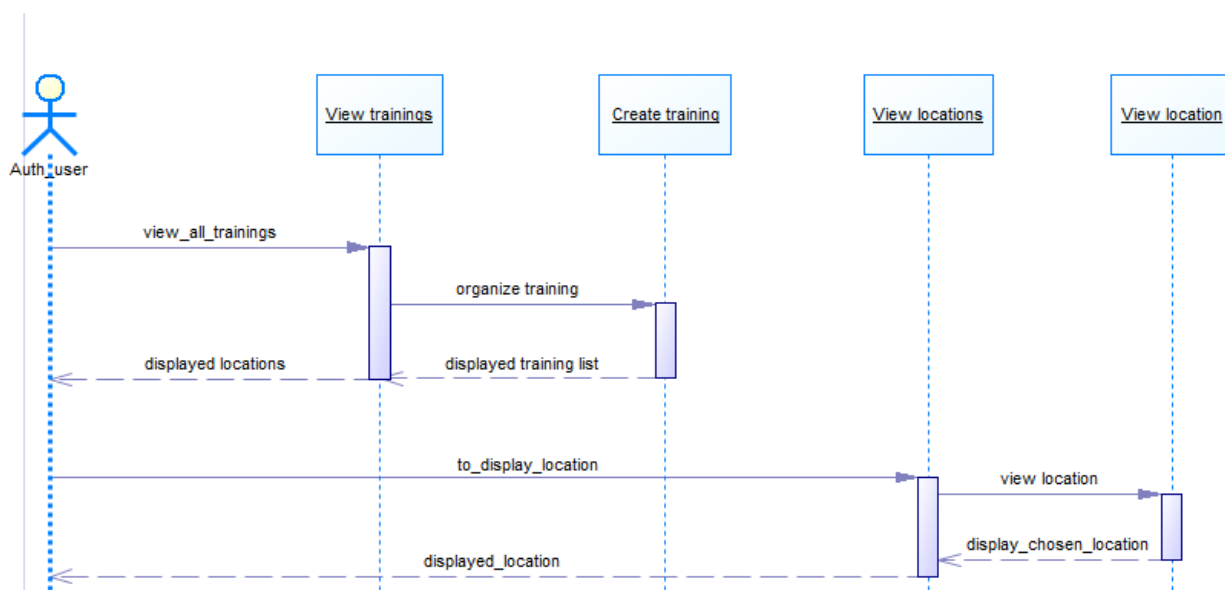
Перехід вказує перехід користувача між інтерфейсами та взаємозв'язок між інтерфейсами.

Діаграми побудовані для неавторизованого користувача, авторизованого та адміністратора. Під час виконання даного завдання неавторизовані користувачі мають обов'язкову точку входу – авторизацію. Всі інші процеси залежать від конкретного користувача.

Відповідні діаграми представлені в Додатку Б на рисунках Б.1, Б.2 та Б.3.

### 2.3 Діаграми послідовностей

Діаграма послідовностей для авторизованого користувача має дві паралельні



послідовності.

Перша послідовність показує дії користувача, який одразу організовує тренування. Він може перед тим як створити новий запис із тренуванням подивитися журнал усіх тренувань. Друга послідовність – це перегляд списку локацій та потім із списку локацій можна окремо переглянути окрему локацію. Діаграма послідовності представлена на рисунку 2.3.1.

Рисунок 2.3.1 – Діаграма послідовності для авторизованого користувача

### 3 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ

Розроблена в рамках виконання етапу курсової роботи діаграма «сутність-зв'язок» (ERD) для процесу «Тренування на природі» складається із:

- а) сутностей (кількість – 5);
- б) зв'язків.

Основний елемент діаграми ERD - це сутність: множина об'єктів зі спільними характеристиками, які називаються атрибутами. В роботі було використано зв'язки між сутностями наступних типів:

а) один до багатьох – реалізується імпортуванням ключа сутності до сутності, в якій відповідає хоча б двом об'єктам відповідає об'єкт з першої сутності;

б) багато до багатьох - реалізується за допомогою введення додаткової сутності, яка імпортує ключові атрибути пов'язаних сутностей та з'єднана з ними зв'язком «один до багатьох»;

в) один до одного - реалізується за допомогою введення додаткової сутності, яка імпортує ключові атрибути пов'язаних сутностей та з'єднана з ними зв'язком «один до багатьох», але для сутності, яка знаходиться зі сторони «багато» реалізуються програмно алгоритми перевірки на відсутність дублікатів об'єктів сутності, яка стоїть зі сторони «один» (тому, фактично сторона «багато» за сенсом є аналогічною стороні «один»).

Окремі зв'язки характеризуються обов'язковістю з боку однієї сутності та необов'язковості з боку другої сутності. Присутні також зв'язки, які характеризуються обов'язковістю з обох боків.

#### 3.1 Опис сутностей

В даній роботі були задіяні наступні сутності: користувач, локація для тренування, тип локації для тренування, тренування. В базі даних сутності представлені таблицями з відповідними атрибутами (характеристиками).

Розберемо кожну сутність окремо.

Сутність «Користувач» представляє усіх користувачів системи: як загальних користувачів, так і адміністраторів. Вони усі мають спільні якості як людини, але можуть мати різні ролі в системі, яким надається відповідний функціонал системи. Отже розділяє

усіх користувачів на загальних та адміністраторів окремий атрибут таблиці “Users” - це атрибут “User\_type”, що визначає роль при авторизації. До того ж первинним ключем користувача є його електронна пошта. Інші атрибути (поля) користувачів досить стандартні: ім’я, фамілія, дата народження, пароль від системи, номер телефону. Ще є наступні атрибути: статус видалений користувач чи ні та роль користувача.

Наступна сутність «Локація» також має атрибути, такі як назва, адреса, тип локації, що до речі є зовнішнім ключем з відповідної таблиці з типами локацій, наявність водоймів, басейнів, доріжок, тренажерів, плати за відвідування, опис.

Одна з найважливіших таблиць є таблиця «Trainings». Її ключем є дата тренування а атрибутом є опис тренування. Концептуальна ERD представлена на рисунку 3.1.1.

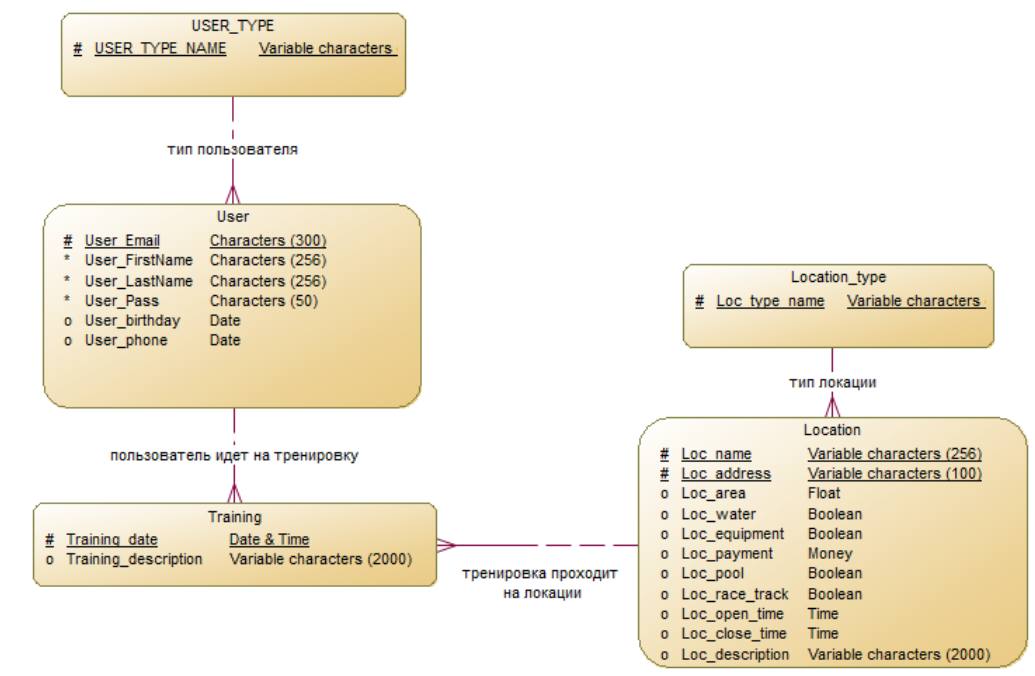


Рисунок 3.1.1 – Концептуальна ERD

Основні зв’язки далі ще описані.

Зв’язок між сутностями (таблицями) користувача та тренування є один до багато, названо згідно логіці зв’язку «користувач запланував тренування», є зв’язком один до багато, є залежність по ключу тренування від користувача.

Зв’язок між сутностями (таблицями) локації та тренування є один до багато, названо згідно логіці зв’язку «тренування заплановано на локації», є зв’язком один до багато, є залежність по ключу тренування від локації.



## 4 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ

На діаграмах нижче представлені логічні та фізична діаграми. Різниця між двома логічними діаграмами «сутність-зв'язок» полягає у оптимізації логічної діаграми за допомогою введеного атрибуту Loc\_ID, тобто індекса локації, який є ключем та допомагає не передавати великі масиви даних при передачі ключів в інші сутності. Отже, логічна неоптимізована модель представлена на рисунку 4.1, логічна оптимізована за допомогою введеного штучного атрибуту – на рисунку 4.2, фізична модель – на рисунку 4.3.

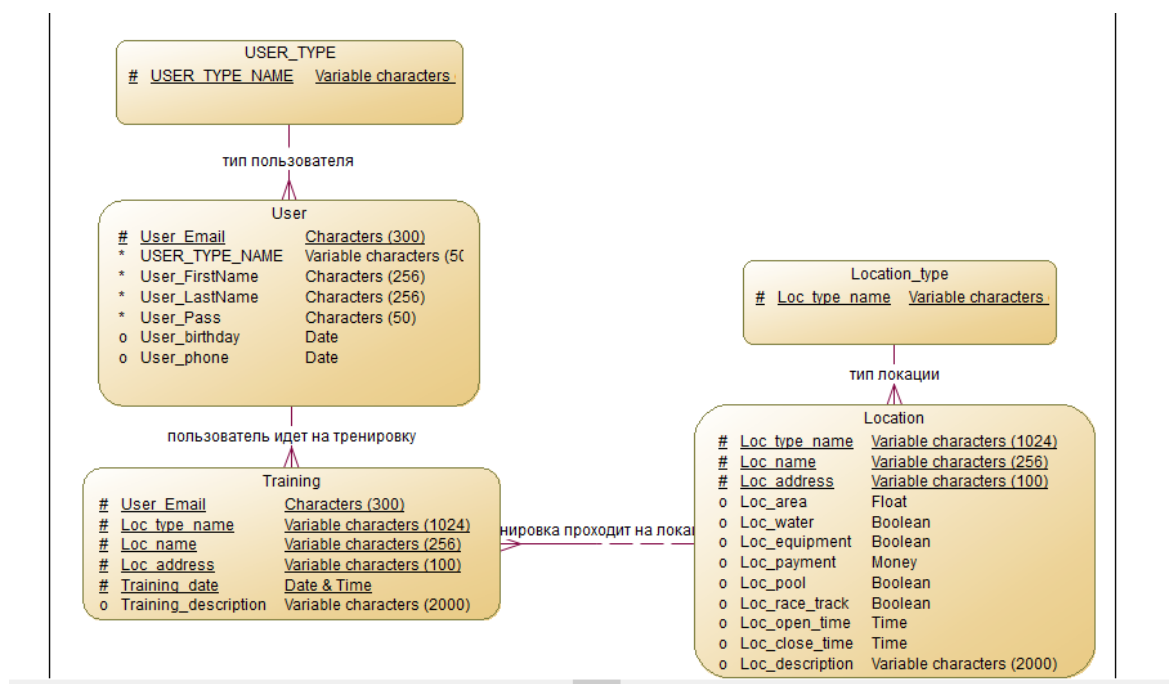


Рисунок 4.1 – Логічна ERD (неоптимізована)



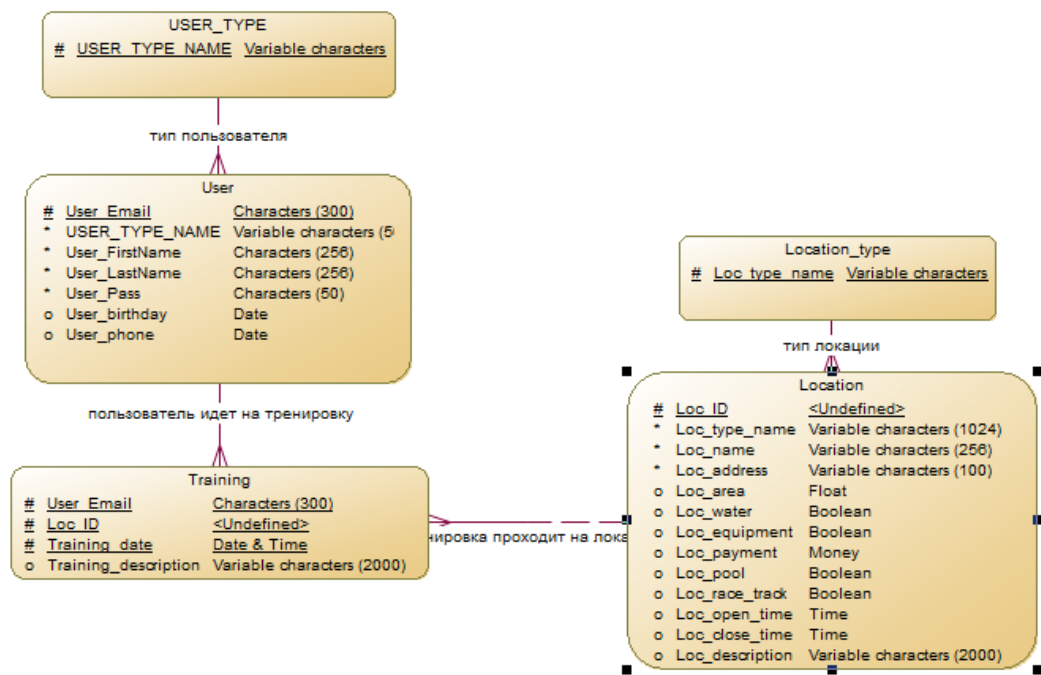


Рисунок 4.2 – Логічна ERD (оптимізована)

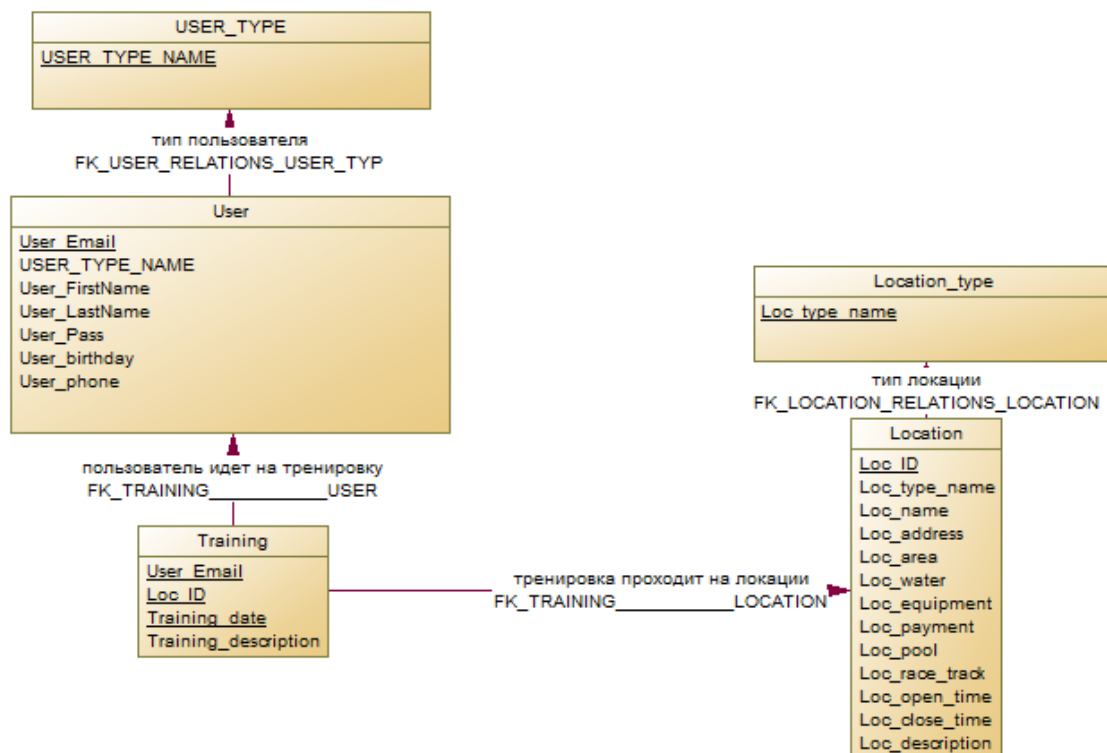


Рисунок 4.3 – Фізична ERD

## ВИСНОВКИ

В даній курсовій роботі реалізована система ведення журналу тренувань на природі, що розроблена задля більш зручної організації тренувань та економії часу для підбору зручних локацій для тренування завдяки базі локацій для тренувань на природі.

Серед основного функціоналу даної системи є наступне:

- г) створення тренування із відповідною інформацією щодо тренування, а саме локація, тип локації (наприклад, ліс, пляж, стадіон, парк), дата та опис тренування, де користувач може залишити нотатки щодо мети тренування, стилю або іншу потрібну йому інформацію;
- д) можливість перегляду локацій для тренування як списком, так і кожну локацію окремо;
- е) можливість перегляду журналу тренувань, тобто список тренувань, що вже були проведені або вже заплановані на майбутнє.

Функціонал, що описано вище, надано авторизованому користувачу. Але це детальніше описано в підрозділі 1.1. Цей функціонал є основним для даної системи, він задовольняє основну мету системи.

Функціонал, що є допоміжним або обслуговуючим основний функціонал, наведено в наступному переліку:

- е) авторизація користувача;
- ж) реєстрація користувача (додавання користувача до БД);
- з) редагування та видалення користувача;
- и) додавання та видалення локації до БД;
- к) видалення тренувань з БД.

Не весь перелічений функціонал є доступним будь-якому користувачу. Кожен користувач має свій дозволений набір функцій. Це детальніше описано в підрозділі 1.1.

Логіка даної системи описана в діаграмі «зв'язок-сутність», що має 5 сутностей (користувач, тип користувача, локація, тип локації, тренування), зв'язки між сутностями і множину атрибутів до кожної. Дані сутності та атрибути допомогли побудувати схему реалізації та взагалі функціонування системи задля оптимальності як з точки зору реалізації, так і з точки зору функціонування та використання. Дана діаграма описана детальніше в розділі 5. Саме починаючи з цього будувалась база даних. Перша концептуальна модель ER діаграми була основою для побудови логічної

та фізичної, а далі і схеми бази даних (побудови логіки та складання скрипту створення схеми БД).

Клієнт- серверна архітектура реалізована за допомогою мови програмування Java версії 8 та сервлетами Apache Tomcat, що мали практичне застосування для побудови серверної та клієнтської частин системи. Для роботи із БД використовували мову PL/SQL, базу даних Oracle та середу розробки SQL Developer.

## СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

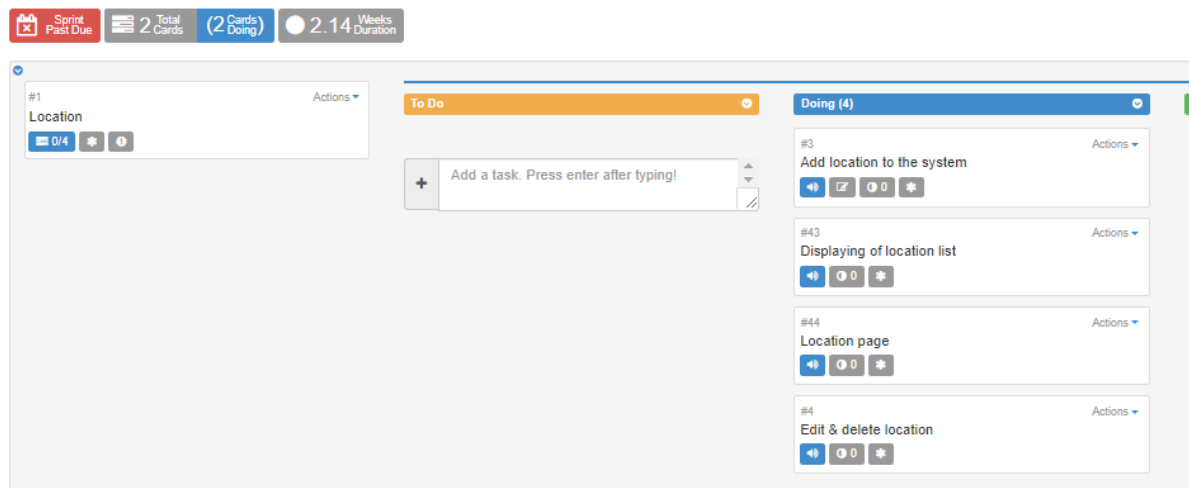
1. Форта Б. - Освой самостоятельно SQL. 10 минут на урок, 3-е издание. — Вильямс, 2005. — 287 с.

2.

## ДОДАТОК А

В даному додатку надані скріни спринтів, що розроблені для проектування. Детальний розгляд наведено в розділі 3.

Locations & trainings: operations



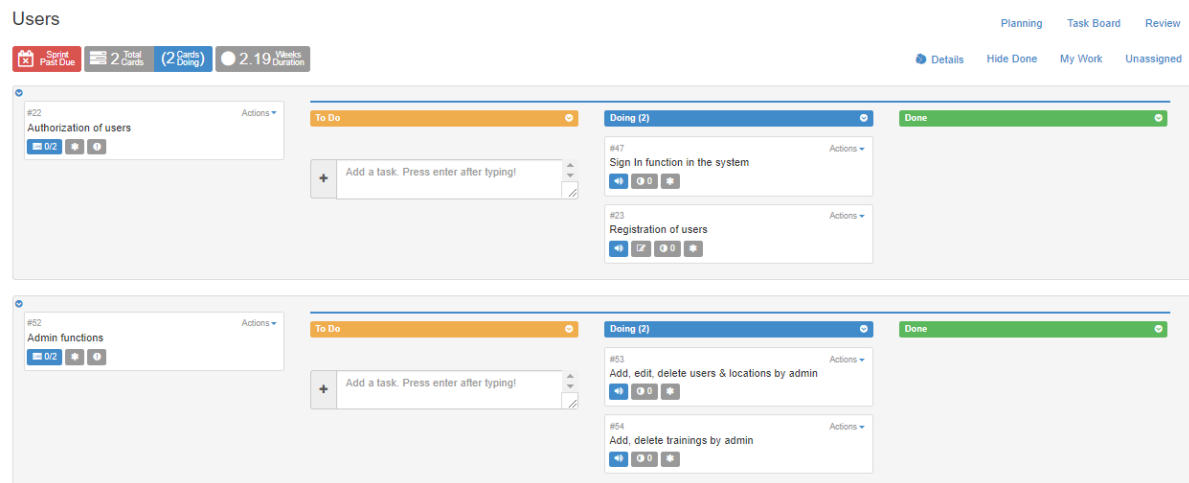


Рисунок А.1 – Scrum. Перший спрінт Location

Рисунок А.2 – Scrum. Перший спрінт Trainings

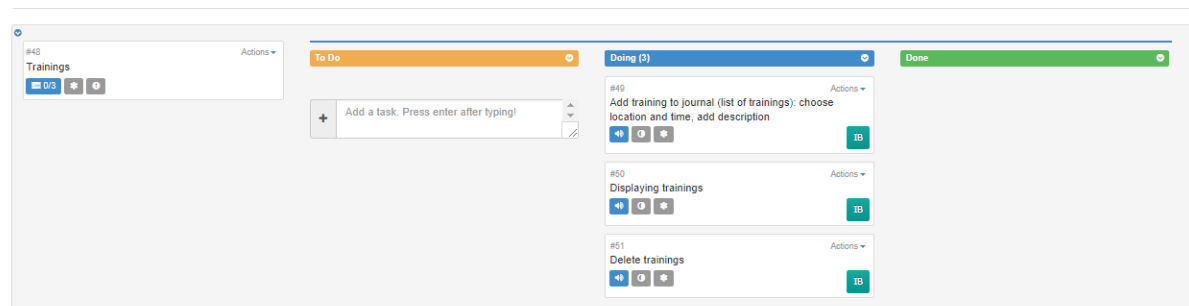


Рисунок А.3 – Scrum. Другий спрінт стосовно користувачів

## ДОДАТОК Б

Діаграма послідовності для неавторизованого користувача – на рисунку Б.1.

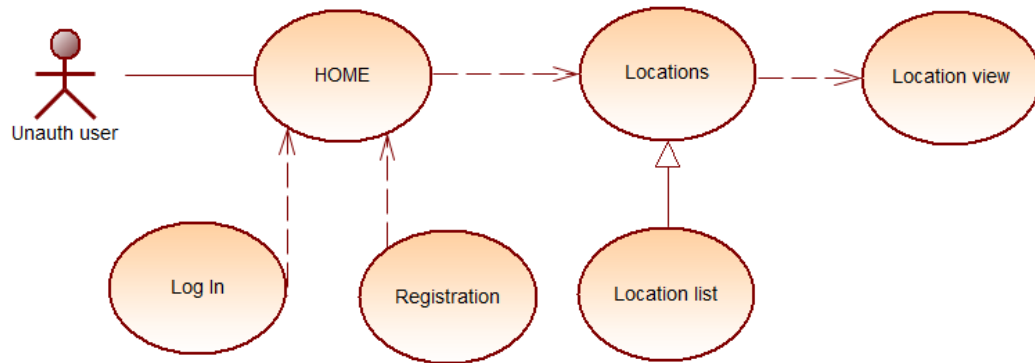


Рисунок Б.1 – Use Case діаграма для неавторизованого користувача

Діаграма послідовності для авторизованого користувача – на рисунку Б.2.

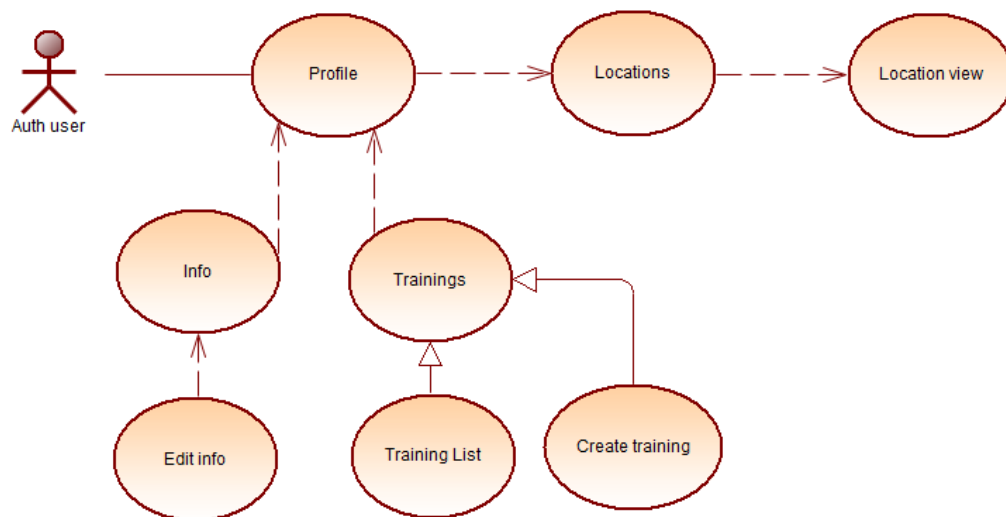


Рисунок Б.2 – Use Case діаграма для авторизованого користувача

Діаграма послідовності для адміністратора – на рисунку Б.3.

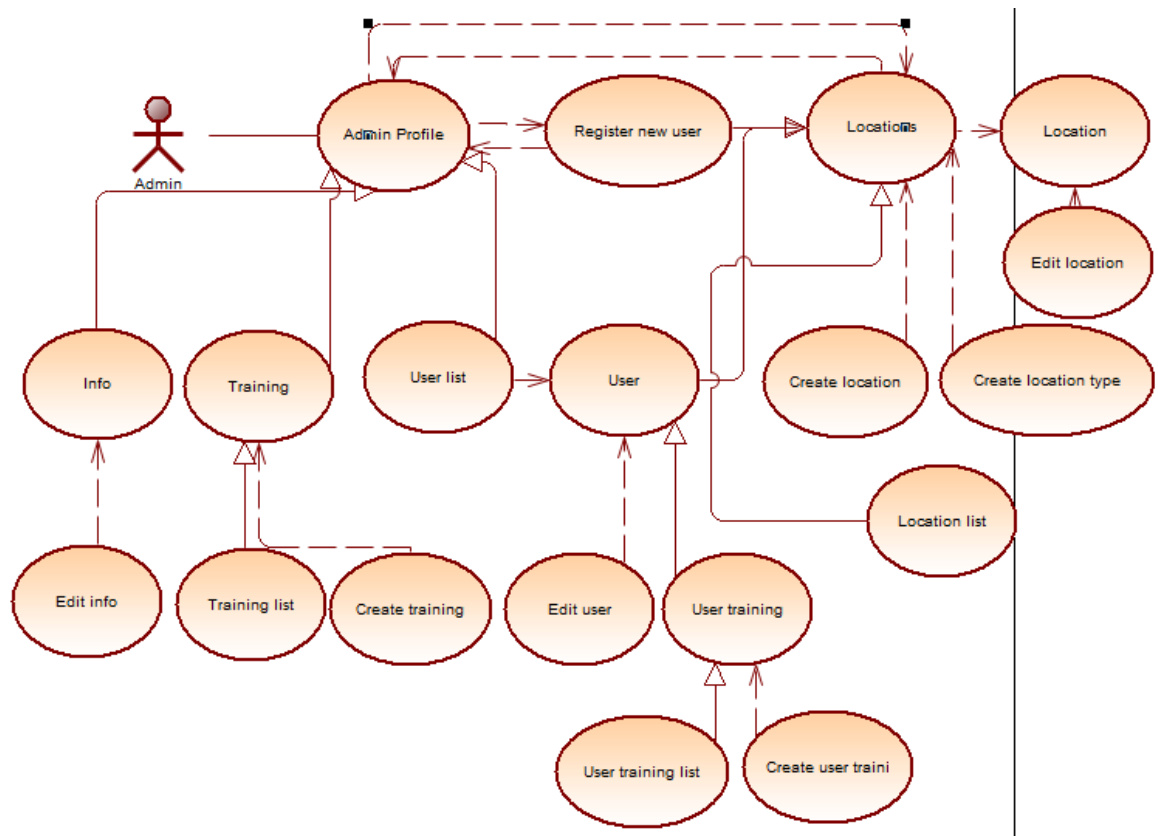


Рисунок Б.2 – Use Case діаграма для авторизованого користувача