МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Факультет інформатики і обчислювальної техніки

Кафедра обчислювальної техніки

**Пояснювальна записка**

до курсового проекту

з дисципліни

«Бази даних - 2. Програмні додатки»

на тему:

«Web-сервіс онлайн тренувань»

|  |  |
| --- | --- |
| Керівник :  Болдак Андрій Олександрович | Виконав :  Кафтанатій Богдан |
| Допущений до захисту | студент гр. ІП-42, ФІОТ |
| “\_\_\_“ \_\_\_\_\_\_\_\_ 201\_ \_\_\_\_\_\_  підпис | 3 курс  № IП-4211 |
| Захистив з оцінкою  \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_  оцінка підпис  “\_\_\_“\_\_\_\_\_\_\_\_\_ 201\_ | Підпис: |

Київ – 2016

ЗМІСТ

[РОЗДІЛ 1 4](#_Toc468369333)

[1.1. Вступ 4](#_Toc468369334)

[1.1.1. Мета 4](#_Toc468369335)

[1.1.2. Контекст 4](#_Toc468369336)

[1.2. Короткий огляд продукту 4](#_Toc468369337)

[1.3. Ділові правила 4](#_Toc468369338)

[1.3.1. Призначення продукту 4](#_Toc468369339)

[1.3.2. Політика взаємовідносин з клієнтом 5](#_Toc468369340)

[1.3.3. Характеристика ділового процесу 5](#_Toc468369341)

[1.4. Сценарії 5](#_Toc468369342)

[1.4.1. Сценарій вибору тренера 5](#_Toc468369343)

[1.4.2. Сценарій редагування розкладу тренувань 6](#_Toc468369344)

[1.4.3. Сценарій контролю тренувального процесу клієнта 7](#_Toc468369345)

[1.4.4. Сценарій публікації статті в системі. 8](#_Toc468369346)

[1.5. Функціональність 8](#_Toc468369347)

[1.6. Практичність 9](#_Toc468369348)

[1.7. Надійність 9](#_Toc468369349)

[1.7.1. Захист інформації користувача 9](#_Toc468369350)

[1.7.2. Навантаження на систему 9](#_Toc468369351)

[1.7.3. Забезпечення достовірності даних 9](#_Toc468369352)

[РОЗДІЛ 2 10](#_Toc468369353)

[2.1. Загальна схема прецедентів 10](#_Toc468369354)

[2.2. Опис прецедентів 11](#_Toc468369355)

[2.3. Діаграма бізнес-сутностей 14](#_Toc468369356)

[2.4. Реляційна модель 15](#_Toc468369357)

[2.5. Специфікація таблиць бази даних 15](#_Toc468369358)

[РОЗДІЛ 3 18](#_Toc468369359)

[3.1. Реляційно-об’єктне відображення 18](#_Toc468369360)

[3.2. Специфікація підсистем додатку 19](#_Toc468369361)

[3.2.1. Підсистема Beans 19](#_Toc468369362)

[3.2.2. Підсистема ManagedBeans 20](#_Toc468369363)

[РОЗДІЛ 4 21](#_Toc468369364)

[4.1. Реєстрація користувача 21](#_Toc468369365)

[4.2. Вхід в систему 21](#_Toc468369366)

[4.3. Вибір тренера 22](#_Toc468369367)

[4.4. Перегляд списку тренувань 22](#_Toc468369368)

[4.5. Збереження тренування 23](#_Toc468369369)

[4.6. Редагування розкладу тренувань 24](#_Toc468369370)

[4.7. Обмін повідомленнями з тренером 24](#_Toc468369371)

[СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ 26](#_Toc468369372)

[ДОДАТКИ 27](#_Toc468369373)

[SQL код створення таблиць бази даних 27](#_Toc468369374)

[Код програми 28](#_Toc468369375)

# РОЗДІЛ 1

Запити зацікавлених осіб

## Вступ

У даному документі описуються запити зацікавлених осіб по відношенню до розроблюваної системи «Онлайн тренер», у якості яких виступають клієнти, адміністратор, тренери.

### Мета

Метою документа є визначення основних вимог до функціональності, продуктивності, експлуатаційної придатності, а також визначення бізнес правил і технічних обмежень створюваної системи.

### Контекст

Перелік вимог, перерахованих у цьому технічному документі, є основою технічного завдання на розробку системи “Онлайн тренер“.

## 1.2. Короткий огляд продукту

Сервіс онлайн тренувань являє собою систему, яка призначена для проведення тренувального процесу клієнта з віддаленим тренером. Система допомагає фіксувати розклад та результати тренувань, та спрощує взаємодію між тренером та клієнтом.

Доступ до системи мають:

1. Адміністратор
2. Тренери
3. Клієнти

## 1.3. Ділові правила

### Призначення продукту

Забезпечення повноцінної взаємодії клієнтів та тренерів, без необхідності безпосередньої присутності. Система дозволяє побудувати тренувальну програму клієнтів, надаючи їм потрібну інформацію та допомогу віддаленого тренера. Перевагами даної системи є те, що клієнт має можливість швидко знаходити інформацію про основні аспекти тренувальної програми; клієнт може створити персональну сторінку та обрати власного тренера; тренери, у свою чергу, можуть складати/редагувати програму тренувань клієнтів та отримувати інформацію від клієнтів, про їх успішність.

### Політика взаємовідносин з клієнтом

Користувачі: адміністратор, тренер, клієнт.

Клієнт після реєстрації отримує власну сторінку, де може переглянути профілі усіх тренерів та обрати певного за бажанням. Також є можливість створення власної програми тренувань або дієти у себе на сторінці.

Тренер має доступ до власної сторінки, може редагувати свій профіль. Є можливість переглядати профілі клієнтів, вносити коментарі у зіставлені програми/дієти. Також тренери мають можливість пропонувати до публікації на сайті статті або певні цікаві матеріали.

Адміністратор має персональну сторінку, з якої може переглядати сторінки усіх клієнтів та тренерів системи. Також адміністратор може переглянути перелік всіх запропонованих до публікації статей.

### Характеристика ділового процесу

Керівництво системою здійснюється адміністратором. Він несе відповідальність за збереження інформації будь-якого роду. Клієнти та тренери мають можливість лише редагувати власні профілі, а останні, крім цього, можуть додавати коментарі до програм своїх клієнтів та пропонувати статті для публікації.

## 1.4. Сценарії

### Сценарій вибору тренера

Учасники: клієнт, тренер, система.

Передумова: зареєстрований клієнт не має тренера.

1. Клієнт авторизується в системі за допомогою свого логіну та паролю.
2. Клієнт переглядає список всіх доступних тренерів.
3. Клієнт може фільтрувати доступних тренерів по їх оцінці.
4. Клієнт може поспілкуватися з тренером, дізнатися інформацію, що його цікавить.
5. Клієнт обирає тренера.
6. Система зберігає інформацію про вибраного тренера до профілю клієнта.
7. Система обновлює профіль клієнта в базі даних.

Результат: інформацію про обраного тренера збережена в базі даних.

Виключні ситуації:

* В системі не зареєстровано жодного тренера.

### Сценарій редагування розкладу тренувань

Учасники: клієнт, тренер, система.

Передумова: клієнт зареєстрований в системі.

1. Редагування розкладу клієнтом:
   1. Клієнт авторизується в системі за допомогою свого логіну та паролю.
   2. Клієнт переходить до підсистеми розкладу тренувань.
   3. Клієнт редагує свій розклад тренувань за допомогою додавання або видалення фізичних вправ, які мають бути виконані в певний день.
   4. Клієнт зберігає розклад в системі.
   5. Система оновлює нові дані розкладу в базі даних.
2. Редагування розкладу тренером:
   1. Тренер авторизується в системі за допомогою свого логіну та паролю.
   2. Тренер переходить до підсистеми роботи зі всіма його клієнтами.
   3. Тренер обирає клієнта та переходить до його профілю.
   4. Тренер переглядає розклад тренувань обраного клієнта.
   5. Тренер вносить зміни до розкладу та зберігає їх як пропоновані клієнту.
   6. Система надсилає зміни до клієнта, як пропоновані тренером.
   7. Клієнт переглядає зміни та може зберегти їх або видалити.

Результат: розклад клієнта оновлений.

### Сценарій контролю тренувального процесу клієнта

Учасники: клієнт, тренер, система.

Передумова –  клієнт має виконане тренування не збережене в системі.

1. Клієнт авторизується в системі за допомогою свого логіну та паролю.
2. Клієнт переходить до підсистеми тренувань.
3. Якщо клієнт має фітнес браслет або тренувальний додаток на мобільному телефоні:
   1. Клієнт синхронізує дані з браслету/телефону з базою даних.
4. Інакше:
   1. Клієнт обирає додання тренування вручну.
   2. Клієнт вводить потрібні дані для збереження тренування.
   3. Система зберігає отримані данні в базі даних.
5. Тренер авторизується в системі за допомогою свого логіну та паролю.
6. Тренер переходить до системи роботи з клієнтами.
7. Тренер переглядає список тренувань клієнта.
8. Тренер робить відповідні висновки та замітки.
9. Тренер відсилає результати аналізу тренування клієнту.

Результат: клієнт оновив свою тренувальну історію. Тренер переглянув тренувальні успіхи клієнта.

### Сценарій публікації статті в системі.

Учасники: модератор, тренер, система.

Передумова –  клієнт має виконане тренування не збережене в системі.

1. Тренер авторизується в системі за допомогою свого логіну та паролю.
2. Тренер переходить до системи відображення всіх публікацій.
3. Тренер обирає створення нової публікації.
4. Тренер вводить необхідні дані для публікації.
5. Система надсилає публікацію на розгляд адміністратору.
6. Адміністратор авторизується в системі за допомогою свого логіну та паролю.
7. Адміністратор переходить до підсистеми запропонованих публікацій.
8. Адміністратор переглядає всі запропоновані публікації, та публікує їх, або повертає на доопрацювання автору.
9. Система зберігає перевірені публікації в базі даних.

Результат: всі запропоновані публікації переглянуті та всі публікації, які пройшли перевірку відображено в системі.

## Функціональність

Основні вимоги щодо функціональності, пред’явлені зацікавленими особами до предмету розробки відносяться до категорії:

* Клієнт
* Тренер
  + 1. **Можливості для клієнта**

Клієнт є зареєстрованим повноцінним користувачем системи і має наступні можливості:

* Створення та редагування розкладу тренувань
* Збереження та редагування історії тренувань
* Вибір персонального тренера
* Обмін інформацією з тренером
  + 1. **Можливості тренера**

Тренер також є зареєстрованим користувачем системи. Його можливості:

* Перегляд списку та профілів всіх своїх клієнтів
* Редагування розкладу клієнта
* Обмін інформацією з клієнтом

## Практичність

Доступ до сервісу повинен здійснюватися з усіх актуальних версій браузерів.

Для користувацького інтерфейсу мають виконуватись наступні вимоги:

* Бути зрозумілим і не мати надлишковий функціонал
* Правильно відображатися на екранах різної роздільної здатності

## Надійність

### Захист інформації користувача

Сервіс повинен надавати надійні засоби захисту особистої інформації про користувачів, включаючи особисту інформацію. Вся інформація має бути максимально захищена.

### Навантаження на систему

Система має витримувати значні навантаження, обслуговуючи велику кількість користувачів.

### Забезпечення достовірності даних

Система повинна надавати кожному користувачу доступ лише до тих ресурсів, на які він має право. Для забезпечення даного механізму система може запросити підтвердження особи користувача та ввести пароль.

# РОЗДІЛ 2

Розробка інформаційного забезпечення

## 2.1. Загальна схема прецедентів

Загальна схема прецедентів показує можливості дій акторів. Основним видом діяльності клієнта є збереження своєї історії тренувань. Користувач також може зберігати та редагувати розклад тренувань, переглядати список тренерів та обирати особистого тренера. Тренер може переглядати історії тренувань своїх клієнтів та пропонувати зміни до розкладу. Схема прецедентів зображена на рис. 2.1.



Рисунок 2.1 Реляційна модель бази даних

## 2.2. Опис прецедентів

Нижче представлені прецеденти з вказаними передумовами, результатом, виключними ситуаціями та детальним описом послідовності дій.

Прецедент №1. Авторизація

Таблиця 2.1.

|  |  |  |
| --- | --- | --- |
| ID | 1 | |
| Назва | Авторизація | |
| Учасники | Користувач, система | |
| Передумова | – | |
| Результат | Користувач авторизований в системі | |
| Основний сценарій | Дії користувача | Дії системи |
|  | 1. Користувач вводить дані для авторизації у формі входу |  |
|  |  | 2. Система перевіряє дані та авторизує користувача |
| Виключні ситуації | Введено неіснуючий логін.  Введено неправильний пароль. | |

Прецедент №2. Реєстрація

Таблиця 2.2.

|  |  |  |
| --- | --- | --- |
| ID | 2 | |
| Назва | Реєстрація | |
| Учасники | Користувач, система | |
| Передумова | – | |
| Результат | Користувач зареєстрований в системі | |
| Основний сценарій | Дії користувача | Дії системи |
|  | 1. Користувач вводить дані для реєстрації у форму |  |
|  |  | 2. Система перевіряє дані та створює запис про новий обліковий запис в базі даних |
| Виключні ситуації | Користувач з таким логіном вже існує. | |

Прецедент №3. Вибір персонального тренера

Таблиця 2.3.

|  |  |  |
| --- | --- | --- |
| ID | 3 | |
| Назва | Вибір персонального тренера | |
| Учасники | Клієнт, система | |
| Передумова | Клієнт авторизований в системі | |
| Результат | Клієнт обрав тренера | |
| Основний сценарій | Дії клієнта | Дії системи |
|  | 1. Клієнт переходить до підсистеми тренерів. |  |
|  |  | 2. Система відображає всіх тренерів зареєстрованих в системі |
|  | 3. Клієнт обирає тренера |  |
|  |  | 4. Система оновлює профіль клієнта та зберігає зміни до бази даних |
| Виключні ситуації | В системі не зареєстровано жодного тренера. | |

Прецедент №4. Створення розкладу тренувань

Таблиця 2.4.

|  |  |  |
| --- | --- | --- |
| ID | 4 | |
| Назва | Створення розкладу тренувань | |
| Учасники | Клієнт, система | |
| Передумова | Клієнт авторизований в системі | |
| Результат | Клієнт створив власний розклад тренувань | |
| Основний сценарій | Дії клієнта | Дії системи |
|  | 1. Клієнт переходить до підсистеми розкладу тренувань. |  |
|  |  | 2. Система відображає поточний розклад клієнта |
|  | 3. Клієнт вносить зміни до свого розкладу |  |
|  |  | 4. Система оновлює профіль клієнта та зберігає зміни до бази даних |
| Виключні ситуації | – | |

Прецедент №5. Збереження історії тренувань

Таблиця 2.5.

|  |  |  |
| --- | --- | --- |
| ID | 5 | |
| Назва | Збереження історії тренувань | |
| Учасники | Клієнт, система | |
| Передумова | Клієнт авторизований в системі | |
| Результат | Клієнт зберіг інформацію про тренування в системі | |
| Основний сценарій | Дії клієнта | Дії системи |
|  | 1. Клієнт переходить до підсистеми історії тренувань |  |
|  |  | 2. Система відображає список всіх збережених тренувань клієнта |
|  | 3. Клієнт обирає збереження нового тренування |  |
|  |  | 4. Система відображає форму створення нового тренування |
|  | 5. Клієнт вносить дані про тренування до форми |  |
|  |  | 6. Система зберігає тренування в базі даних |
| Виключні ситуації | – | |

Прецедент №6. Обмін повідомленнями

Таблиця 2.6.

|  |  |  |
| --- | --- | --- |
| ID | 6 | |
| Назва | Обмін повідомленнями | |
| Учасники | Користувач, система | |
| Передумова | Користувач авторизований в системі | |
| Результат | Повідомлення збережена в базі даних | |
| Основний сценарій | Дії клієнта | Дії системи |
|  | 1. Користувач переходить до підсистеми листування |  |
|  |  | 2. Система відображає список всіх діалогів користувача |
|  | 3. Користувач обирає діалог |  |
|  |  | 4. Система відображає форму для вводу повідомлення |
|  | 5. Користувач вносить дані до форми |  |
|  |  | 6. Система зберігає повідомлення в базі даних |
| Виключні ситуації | – | |

## 2.3. Діаграма бізнес-сутностей

Дана діаграма створюється на стадії бізнес моделювання. Вона відображає основні сутності та взаємозв’язки між ними. В даному випадку основними сутностями є Client, Trainer, Training і Exercise які взаємодіють між собою. Діаграма бізнес-сутностей проекту зображена на рис 2.2.



Рисунок 1.2 Діаграма бізнес сутностей

## 2.4. Реляційна модель

Реляційна модель бази даних (рис. 2.3) зображує структуру таблиць бази даних, взаємодію між ними та поля кожної з таблиць. Наведена діаграма має багато схожого з діаграмою бізнес-сутностей. Кожній основній бізнес-сутності відповідає таблиця баз даних.

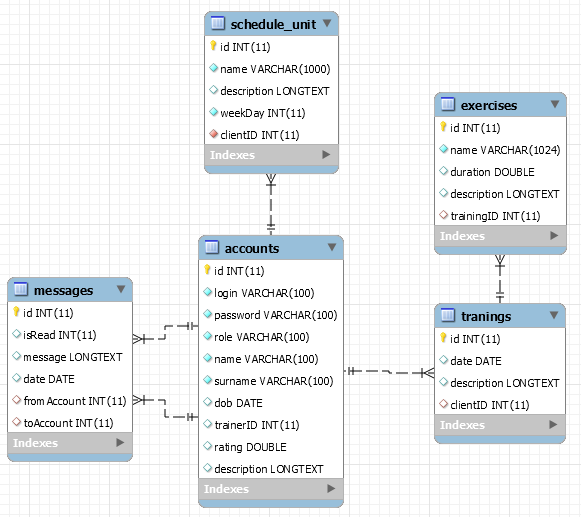


Рисунок 2.3 Реляційна модель бази даних

## 2.5. Специфікація таблиць бази даних

Специфікація таблиць баз даних включає в себе інформацію про назви колонок таблиць, їхній тип, інформацію про те чи є ця колонка первинним ключем, чи поле може бути пустим, чи значення поля автоматично збільшується. Специфікації наведені нижче.

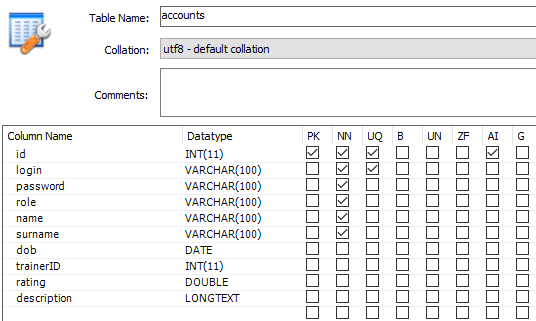


Рисунок 2.4 Таблиця «Accounts»

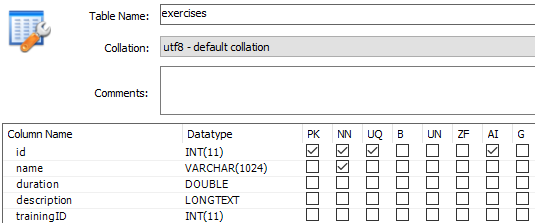


Рисунок 2.5 Таблиця «Exercises»

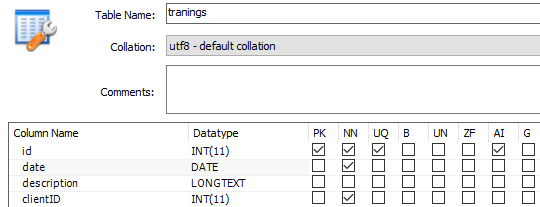


Рисунок 2.6 Таблиця «Trainings»

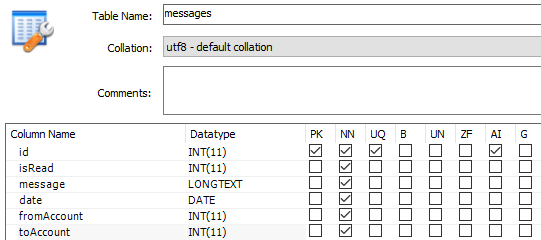


Рисунок 2.7 Таблиця «Messages»

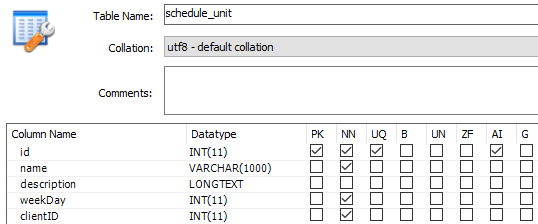


Рисунок 2.8 Таблиця «Schedule\_Unit»

# РОЗДІЛ 3

Розробка програмного продукту

## 3.1. Реляційно-об’єктне відображення

Для реляційно-об’єтного відображення використовується бібліотека JPA (Java Persistence API). JPA надає можливість легко встановити зв’язок з будь-якою базою даних та встановити відображення між об’єктно-орієнтованою моделлю та традиційною реляційною моделлю баз даних. На рисунку 3.1 зображено діаграму класів сутностей системи.

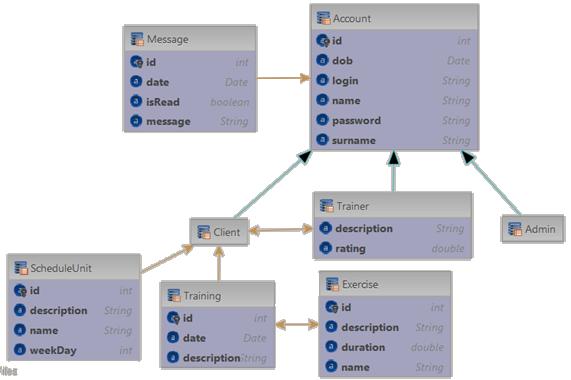


Рисунок 3.1. Реляційно-об’єктне відображення

## 3.2. Специфікація підсистем додатку

### 3.2.1. Підсистема Beans

Класи, що представлені тут наслідують абстрактний клас AbstractDAO і використовуються для маніпулювання з таблицями бази даних. Кожен клас реалізує всі основні CRUD операції для своєї сутності.



*Рисунок 3.2. Діаграма класів підсистеми Beans*

### 3.2.2. **Підсистема ManagedBeans**



Класи, що представлені тут містять методи призначені для отримання даних потрібних для відображення на web-сторінці.

# РОЗДІЛ 4

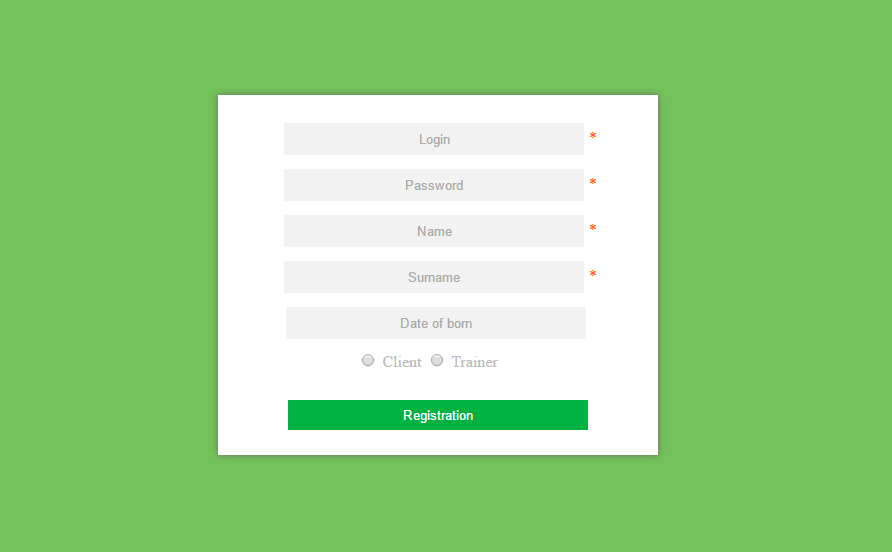
Ілюстрація роботи програми

Для ілюстрації роботи програми в цьому розділі наведено графічні сценарії роботи проекту.

## 4.1. Реєстрація користувача

1. Користувач натискає «Registration» на головній сторінці системи.

2. Система відображає користувачу форму реєстрації.



*Рисунок 4.1. Реєстрація*

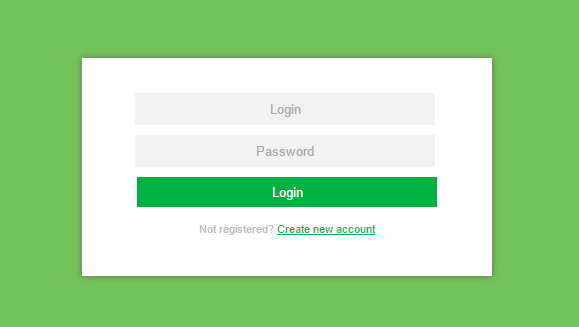
3. Користувач заповнює всі надані поля даними.

4. Система перевіряє правильність та унікальність даних користувача та створює нового користувача.

5. Система перенаправляє користувача на головну сторінку системи.

## 4.2. Вхід в систему

1. Користувач переходить за посиланням яке потребує входу в систему.
2. Система відображає форму логіну.
3. Користувач заповнює всі поля даними.

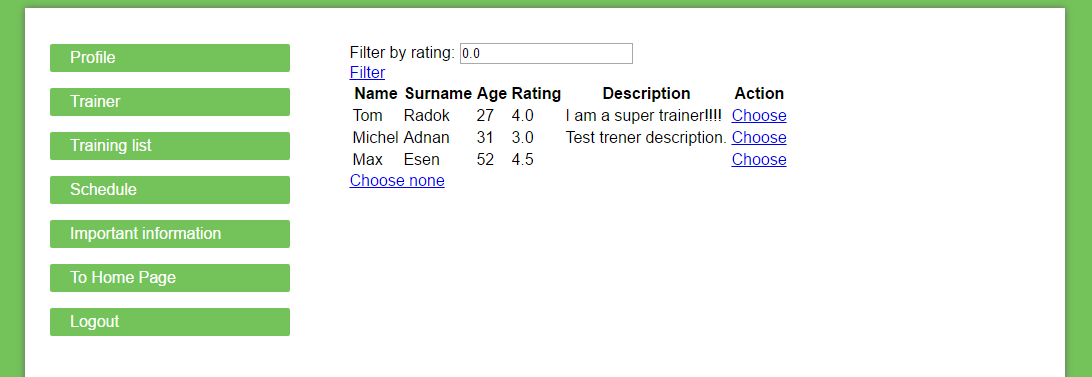


*Рисунок 4.2. Вхід в систему*

1. Система перевіряє правильність введених даних.
2. Система перенаправляє користувача до його особистого кабінету.

## 4.3. Вибір тренера

1. Клієнт переходить до вибору тренера.
2. Система відображає список всіх тренерів.



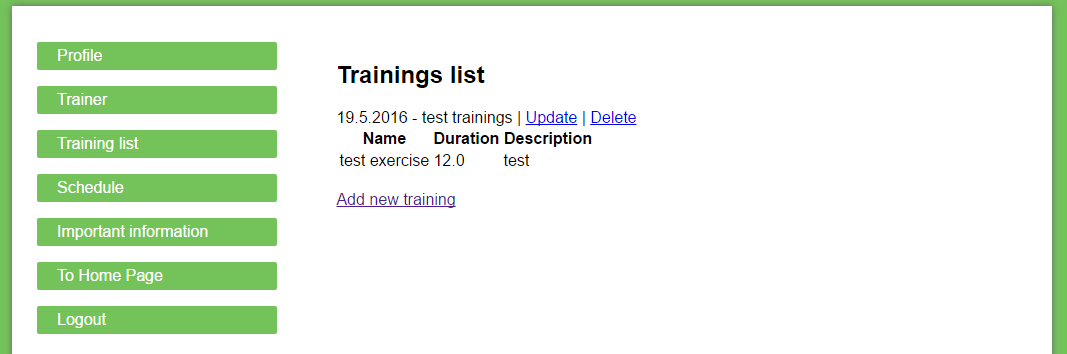
*Рисунок 4.3. Список тренерів*

1. Користувач обирає тренера.
2. Система додає до профілю клієнта інформацію про обраного тренера та оновлює профіль в базі даних.

## 4.4. Перегляд списку тренувань

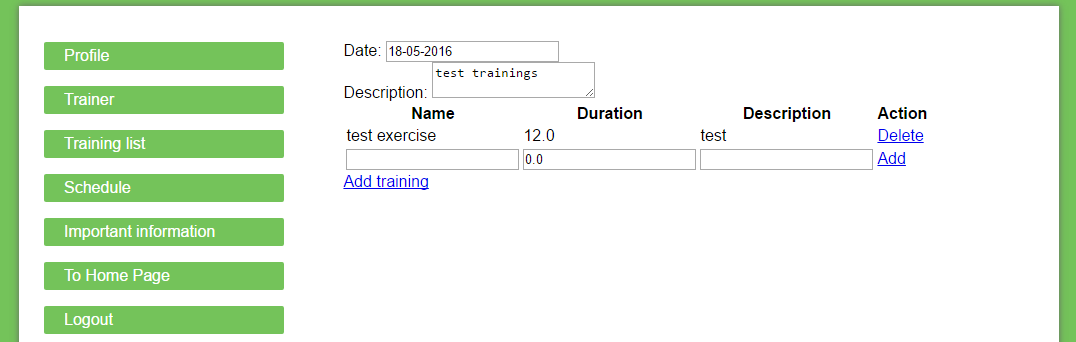
1. Користувач переходить до перегляду тренувань.

2. Система відображає список всіх тренувань.



*Рисунок 4.4. Список тренувань*

3. Користувач обирає тренування та може переглянути його детальну інформацію та відредагувати тренування.

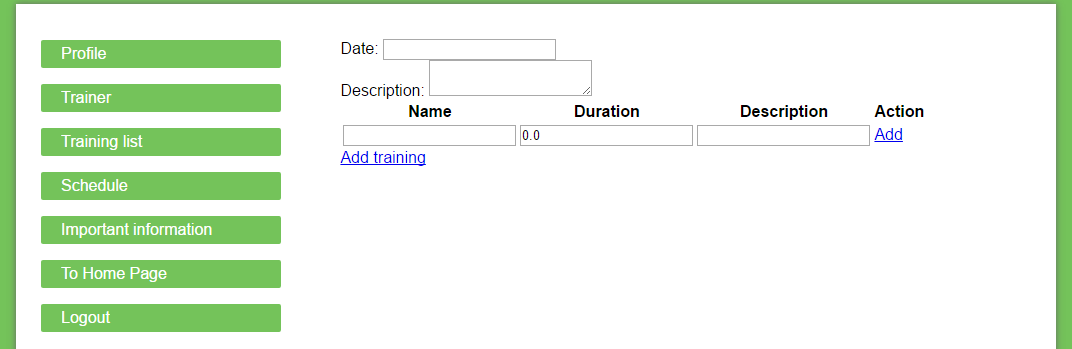


*Рисунок 4.5. Редагування тренування*

## 4.5. Збереження тренування

1. Користувач переходить до додання нового тренування.

2. Система відображає форму для вводу інформації про тренування.



*Рисунок 4.6. Додання тренування*

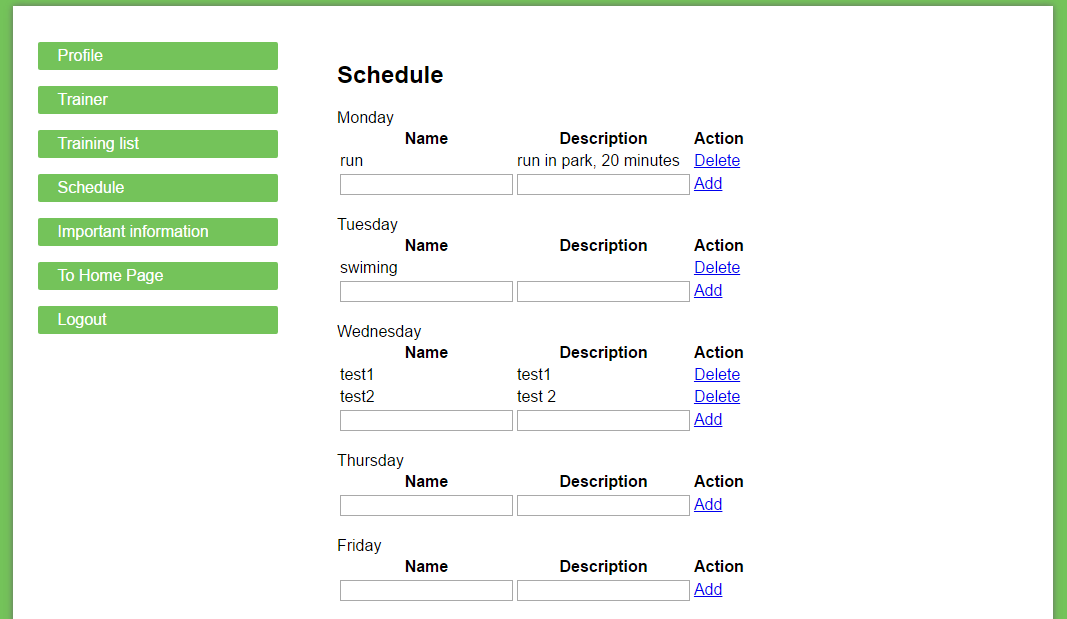
3. Користувач додає до тренування виконані вправи та зберігає тренування.

4. Система зберігає тренування за клієнтом.

## 4.6. Редагування розкладу тренувань

1. Клієнт переходить до редагування розкладу тренувань.

2. Система відображає поточний розклад клієнта в поля для внесення змін до нього.



*Рисунок 4.7. Редагування розкладу тренувань*

3. Користувач вводить необхідну інформацію.

4. Система зберігає зміни в базу даних.

## 4.7. Обмін повідомленнями з тренером

1. Користувач переходить до листування з своїм тренером.

2. Система відображає список всіх збережених повідомлень та форму для вводу нового повідомлення.



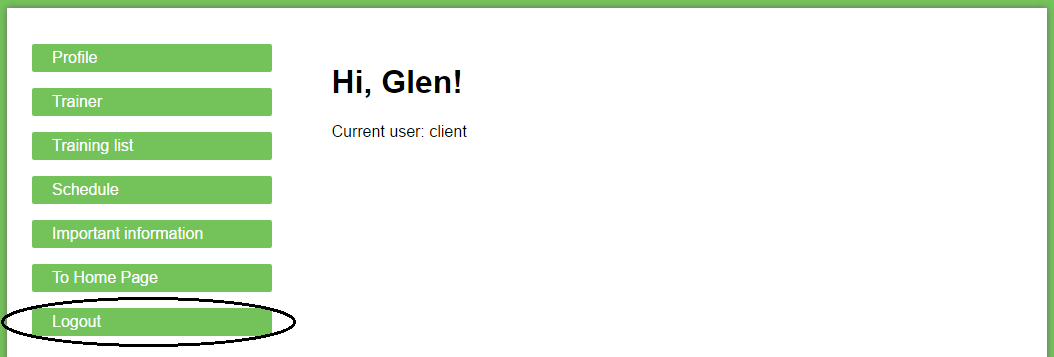
*Рисунок 4.8. Листування з тренером*

3. Користувач вводить повідомлення та натискає надіслати.

4. Система зберігає повідомлення в базі даних та оновлює сторінку.

4.8. Вихід з системи

1. Користувач натискає «Logout».



*Рисунок 4.8. Вихід з системи*

2. Система закінчує поточну сесію та перенаправляє користувача на головну сторінку.

# СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. «Прийоми обектно-орієнтованого проектування. Шаблони проектування.» - СПб: Пітер, 2001.-368 с.
2. Йенер М., Фидом А. «Шаблони проектування» .: Пер. с англ. - М.: Видавничий дім "Вільямс", 2010. - 421 с.
3. Ларман Крег. «Застосування UML и шаблони проектування» - СПб: Пітер, 2009. - 341 с.
4. Гранд Марк . «Шаблони проектування в JAVA» .: Пер. с англ. - М.: Видавничий дім "Вільямс", 2011. - 205 с.
5. Сервіс Coggle, електронний ресурс [https://coggle.it/]
6. Криві Безьє, електронний ресерс [https://learn.javascript.ru/bezier]
7. Іван Портянкін «Swing. Ефективні користувацькі інтерфейси» - Москва: Пітер, 2005 – 61с.
8. Герберт Шилдт «Swing. Керівництво для початківців» : Пер. с англ. - М.: Видавничий дім "Вільямс", 2007, - 571с.
9. Електронна документація Java – електронний ресурс [https://docs.oracle.com/javase/tutorial/]
10. Джошуа Блох «Ефективне програмування Java» - Видаництво «Лорі», 2014 – 166с.

# ДОДАТКИ

## SQL код створення таблиць бази даних

CREATE SCHEMA `trainer\_online` DEFAULT CHARACTER SET utf8 ;

CREATE TABLE `trainer\_online`.`accounts` (

`id` INT NOT NULL AUTO\_INCREMENT,

`login` VARCHAR(100) NOT NULL,

`password` VARCHAR(100) NOT NULL,

`role` VARCHAR(100) NOT NULL,

`name` VARCHAR(100) NOT NULL,

`surname` VARCHAR(100) NOT NULL,

`dob` DATE NULL,

`trainerID` INT NULL,

`rating` DOUBLE NULL,

`description` LONGTEXT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `id\_UNIQUE` (`id` ASC),

UNIQUE INDEX `login\_UNIQUE` (`login` ASC));

CREATE TABLE `trainer\_online`.`tranings` (

`id` INT NOT NULL AUTO\_INCREMENT,

`date` DATE NULL,

`description` LONGTEXT NULL,

`clientID` INT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `id\_UNIQUE` (`id` ASC),

INDEX `clientID\_idx` (`clientID` ASC),

CONSTRAINT `clientID`

FOREIGN KEY (`clientID`)

REFERENCES `trainer\_online`.`accounts` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE);

CREATE TABLE `trainer\_online`.`exercises` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(1024) NOT NULL,

`duration` DOUBLE NULL,

`description` LONGTEXT NULL,

`trainingID` INT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `id\_UNIQUE` (`id` ASC),

INDEX `trainingID\_idx` (`trainingID` ASC),

CONSTRAINT `trainingID`

FOREIGN KEY (`trainingID`)

REFERENCES `trainer\_online`.`tranings` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE);

CREATE TABLE `trainer\_online`.`schedule\_unit` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(1000) NOT NULL,

`description` LONGTEXT NULL,

`weekDay` INT NOT NULL,

`clientID` INT NOT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `id\_UNIQUE` (`id` ASC),

INDEX `clientID\_idx` (`clientID` ASC),

CONSTRAINT `schedule\_clientID`

FOREIGN KEY (`clientID`)

REFERENCES `trainer\_online`.`accounts` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

CREATE TABLE `trainer\_online`.`messages` (

`id` INT NOT NULL AUTO\_INCREMENT,

`isRead` TINYINT NOT NULL,

`message` LONGTEXT NOT NULL,

`date` DATE NOT NULL,

`fromAccount` INT NOT NULL,

`toAccount` INT NOT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `id\_UNIQUE` (`id` ASC),

INDEX `fromFK\_idx` (`from` ASC),

INDEX `toFK\_idx` (`to` ASC),

CONSTRAINT `fromFK`

FOREIGN KEY (`from`)

REFERENCES `trainer\_online`.`accounts` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `toFK`

FOREIGN KEY (`to`)

REFERENCES `trainer\_online`.`accounts` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE);

## Код програми