Чернівецький національний університет імені Юрія Федьковича

Факультет математики та інформатики

Кафедра математичного моделювання

**Лабораторна робота №2**

**«Написання UseCases. Побудова діаграм UseCases. Побудова діаграм взаємодії (Interaction Diagrams)»**

Виконав: студент 3 курсу

301 групи

спеціальності “Комп’ютерні науки”

Марчук Андрій

Викладач: Піддубна Л.А

**Чернівці – 2025**

***Написання UseCases***

*ЗАВДАННЯ*

Відповідно до обраної предметної області на основі проведеного аналізу вимог замовника (лабораторна робота №1) розробити три різні usecases (по одному в короткій, поверхневій та повній формах відповідно) для свого проекту. Повна форма опису має містити всі пункти наведені в таблиці 1. Головний успішний сценарій повинен мати не менше 10 кроків. Передбачити не менше 5 альтернативних сценаріїв

*Use Case №1* Перегляд особистої статистики спортсменом

*Коротка форма:*

Cпортсмен відкриває мобільний додаток, авторизується в системі та переходить у вкладку «Статистика». Система відображає персональні показники: кількість тренувань, результати матчів, прогрес фізичних навантажень, дані зі смарт-годинника (пульс, кроки, калорії). Користувач переглядає інформацію, обираючи фільтри за датою або видом активності.

**Поверхнева форма:**

**Головний сценарій:**

1. Спортсмен відкриває додаток
2. Входить в акаунт
3. Відкриває вкладку «Статистика»
4. Система підтягує дані з бази та зі смарт-годинника
5. Користувач обирає дату або тип діяльності
6. Система оновлює графіки, діаграми та статистичні таблиці, зокрема:
   * Таблиця відвідуваності тренувань
   * Таблиця змін у фізичних показниках (пульс, вага, кроки тощо)
   * Порівняльна таблиця результатів матчів за вибраний період
   * Таблиця прогресу по запланованих вправах
   * Таблиця активності (вказує дні, коли спортсмен був найбільш активним)
7. Система оновлює графіки, діаграми та статистичні таблиці, зокрема:
   * Таблиця відвідуваності тренувань
   * Таблиця змін у фізичних показниках (пульс, вага, кроки тощо)
   * Порівняльна таблиця результатів матчів за вибраний період
   * Таблиця прогресу по запланованих вправах
   * Таблиця активності (вказує дні, коли спортсмен був найбільш активним)

**Альтернативні сценарії:**

1. Користувач не авторизований — система потребує входу
2. Збій сервера даних — система повідомляє: «Дані часово недоступні»
3. Відсутні смарт-дані — система рекомендує синхронізувати гаджет

**Повна форма:**

**Use Case Name:** Перегляд особистої статистики спортсменом

**Scope:** Система управління спортивними клубами та спортсменами

**Level:** User-goal

**Primary Actor:** Спортсмен

**Stakeholders and Interests:**

* **Спортсмен:** хоче відстежувати свій прогрес та ефективність тренувань.
* **Тренер:** аналізує вплив тренувального процесу на результати спортсмена.
* **Адміністратор клубу:** слідкує за активністю спортсменів і відвідуваністю.

**Preconditions:**

* Користувач має обліковий запис.
* Пристрій підключений до інтернету.
* Гаджет для фітнес-моніторингу синхронізований (опціонально).

**Main Success Scenario:**

1. Користувач відкриває мобільний додаток
2. Авторизується в системі
3. Переходить до вкладки «Статистика»
4. Система виконує запит до бази даних
5. Система перевіряє, чи підключено смарт-годинник/браслет
6. Якщо підключено — отримує додаткові біометричні дані
7. Користувач обирає часовий проміжок (тиждень, місяць, квартал)
8. Система виводить графіки (вага, пульс, кількість тренувань тощо)
9. Табличні звіти за видами активності оновлюються
10. Користувач може зберегти або поділитися результатами (опціонально)

**Extensions:**

1a. Відсутнє інтернет-з’єднання → офлайн-режим з останніми збереженими даними.

5a. Смарт-годинник не синхронізовано → система пропонує це зробити

8a. Некоректний діапазон дат → система показує повідомлення про помилку та встановлює стандартні параметри.

10a. Не вдалося поділитись результатом — система показує помилку зв’язку або авторизації соцмережі.

**Special Requirements:**

* Всі візуальні елементи мають бути оптимізовані для мобільного перегляду
* Графіки — інтерактивні (з підказками при наведенні)
* Всі дані зберігаються з урахуванням політики конфіденційності

**Technology and Data Variations List:**

* Взаємодія з Apple Health / Google Fit API
* JSON-запити до внутрішнього REST API
* Відображення через D3.js або Chart.js

**Frequency of Occurrence:** Щодня або кілька разів на тиждень

**Miscellaneous:**

* Як реалізувати порівняння показників між гравцями однієї команди?
* Чи потрібна функція завантаження статистики у форматі PDF або Excel?

Use Case 2: Купівля квитка фанатом на матч

**Коротка форма:**

Фанат заходить у мобільний додаток, авторизується, відкриває вкладку "Квитки", обирає матч, місце, оплачує через Mono або LiqPay. Після успішної оплати система надсилає електронний квиток.

**Поверхнева форма:**

**Головний сценарій:**

1. Користувач відкриває мобільний додаток.
2. Входить до свого акаунта.
3. Обирає вкладку «Квитки».
4. Обирає матч із переліку доступних.
5. Переглядає план стадіону.
6. Обирає місце.
7. Натискає кнопку «Придбати».
8. Обирає спосіб оплати (Mono або LiqPay).
9. Завершує оплату.
10. Система відображає повідомлення про успішну покупку.
11. Квиток зберігається у вкладці «Мої квитки».

**Таблиці, які використовуються:**

* Tickets: ID, користувач, матч, місце, ціна, статус
* Matches: ID, дата, команди, місце проведення
* StadiumSeats: сектор, ряд, місце, доступність
* Transactions: спосіб оплати, час, сума, статус
* Users: ID, роль, email

**Альтернативні сценарії:**

1. Невірні дані авторизації
2. Місце вже зайняте
3. Проблема з платіжним шлюзом
4. Втрата інтернет-з’єднання
5. Матч скасовано

**🔹 Повна форма:**

* **Use Case Name:** Купівля квитка на матч
* **Scope:** Мобільний додаток фаната
* **Level:** User-goal
* **Primary Actor:** Фанат
* **Stakeholders:**
  + Фанат — хоче придбати квиток зручно та швидко
  + Адміністратор клубу — стежить за продажами
  + Фінансовий відділ — контролює оплати
* **Preconditions:** користувач авторизований; матч ще не відбувся
* **Main Scenario:** див. поверхневу форму
* **Extensions:** див. альтернативні сценарії
* **Special Requirements:**
  + Підтримка LiqPay/MonoPay
  + Генерація QR-коду квитка
* **Technology Variations:**
  + Apple Pay, Google Pay (опціонально)
  + JSON API для збереження квитків
* **Frequency of Occurrence:** 60%
* **Miscellaneous:** Як обробляти повернення квитків?

Use Case 3: Додавання тренування тренером

**Коротка форма:**

Тренер заходить у додаток, авторизується, переходить у вкладку "Тренування", обирає дату, час, тип, додає опис, обирає команду — і зберігає. Спортсмени бачать подію у своєму розкладі.

**Поверхнева форма:**

**Головний сценарій:**

1. Тренер входить у мобільний додаток
2. Переходить у вкладку «Тренування»
3. Натискає кнопку «Додати»
4. Вибирає дату, час
5. Обирає команду
6. Обирає тип тренування (кардіо, ігрове, технічне тощо)
7. Додає опис та програму
8. Натискає «Зберегти»
9. Система додає подію до розкладу команди
10. Спортсмени отримують повідомлення

**Таблиці, які використовуються:**

* Trainings: ID, тренер, команда, дата, час, тип, опис
* Users: ID, ім’я, роль
* Notifications: ID, користувач, тип, повідомлення
* Teams: ID, назва, учасники

**Альтернативні сценарії:**

1. Тренер не має доступу до команди
2. Проблема з підключенням
3. Відсутні спортсмени у складі
4. Не збережено тренування
5. Дублікат події

**🔹 Повна форма:**

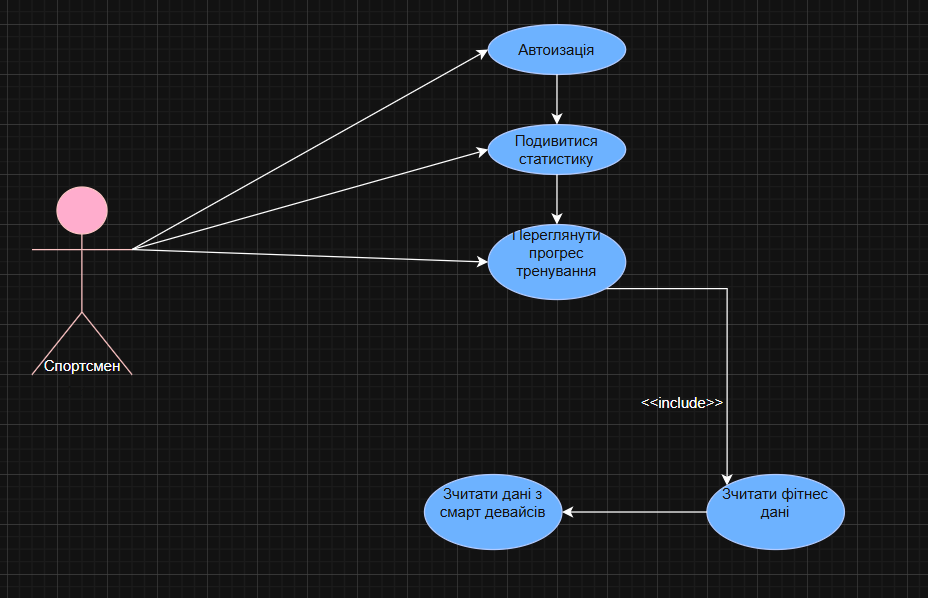
* **Use Case Name:** Додавання тренування
* **Scope:** Адмін-панель тренера
* **Level:** User-goal
* **Primary Actor:** Тренер
* **Stakeholders:**
  + Тренер — планує заняття
  + Спортсмени — бачать розклад
  + Адміністратор — бачить активність тренерів
* **Preconditions:** авторизований тренер; існує команда
* **Main Scenario:** див. поверхневу форму
* **Extensions:** див. альтернативні сценарії
* **Special Requirements:**
  + Автоповідомлення про нове тренування
  + Сумісність з календарем iOS/Android
* **Technology Variations:**
  + Використання Azure Notification Hub для push-сповіщень
* **Frequency of Occurrence:** 85%
* **Miscellaneous:** Як дозволити помічнику тренера створювати тренування?

**Побудова діаграм UseCases**

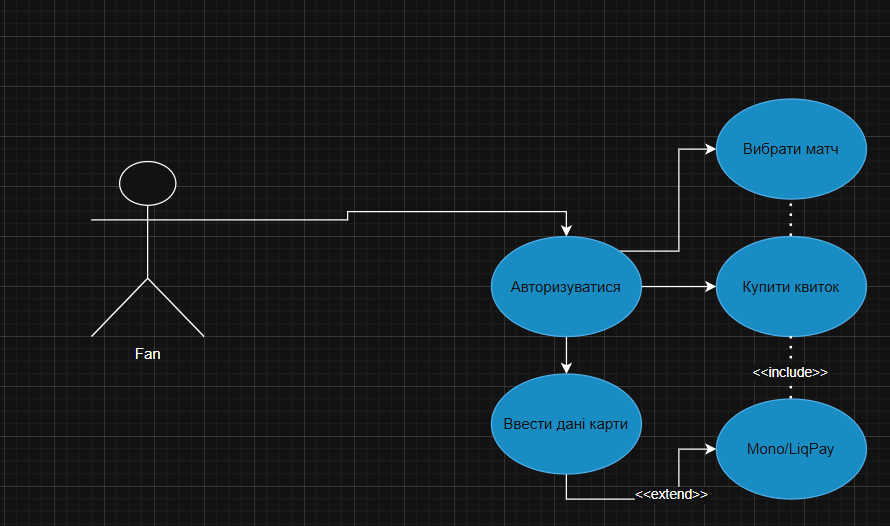
**ЗАВДАННЯ**

У середовищі app.diagrams.net створити діаграму варіантів використання для обраного варіанта комп’ютерної системи. Діаграма повинна містити усіх акторів (користувачів системи) та по три варіанти використання для кожного актора. Пов’язати варіанти використання та акторів, при цьому використати усі види зв’язків (unidirectional association, generalization, extend relationship, include relationship).

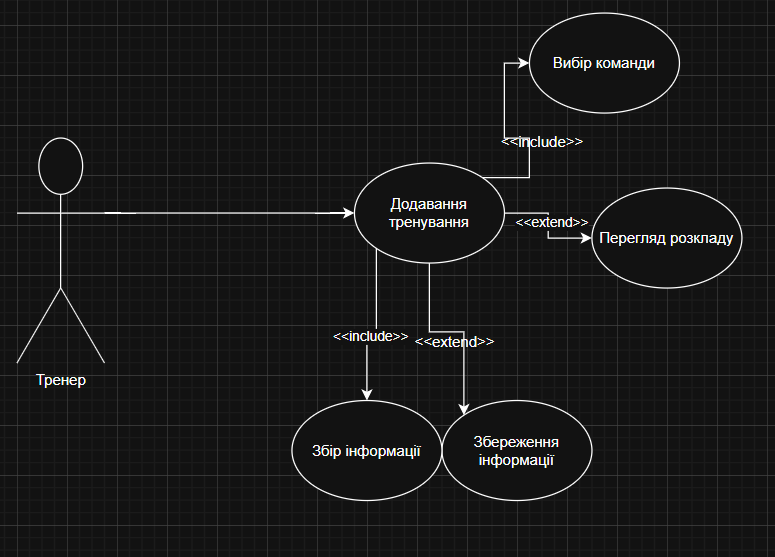
*Use Case №1*



Use Case №2



Use Case №3

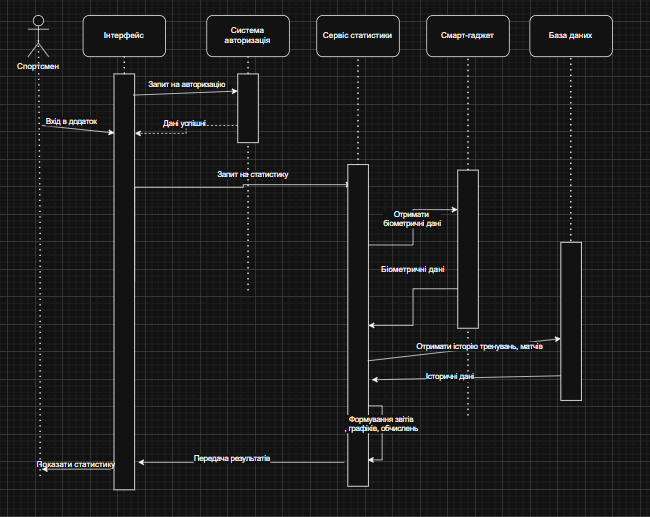


**Побудова діаграм взаємодії (Interaction Diagrams)**

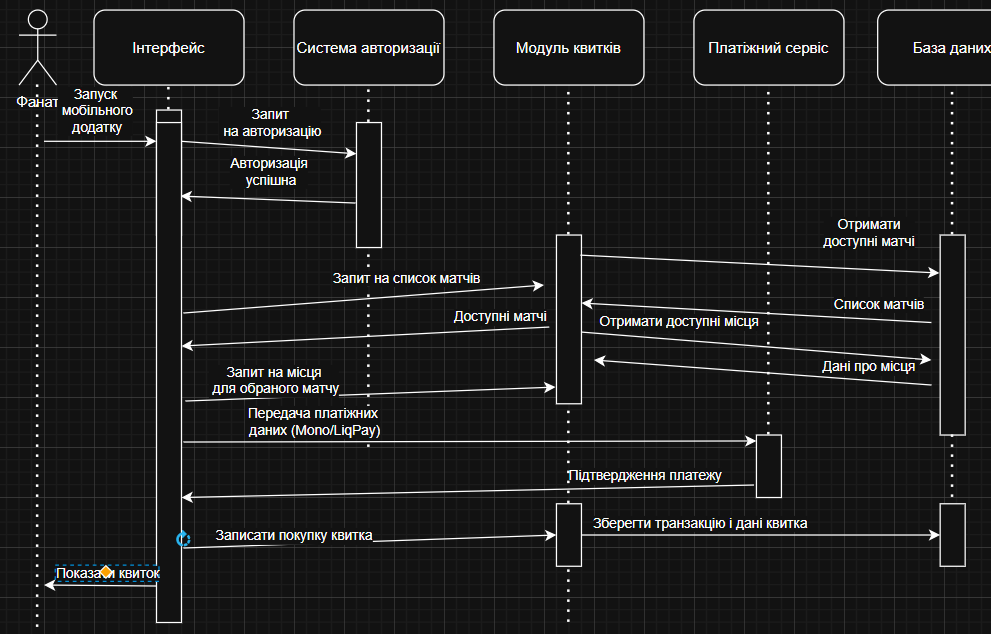
**ЗАВДАННЯ**

Для кожного варіанта використання на Usecase Diagram створити Sequence або Collaboration Diagram (тобто у проекті повинно бути не менше шести діаграм кооперації та послідовності). На кожній діаграмі взаємодії повинен бути головний актор (при наявності) та не менше 5 об’єктів. Кожна діаграма взаємодії повинна містити не менше 10 повідомлень, якими обмінюються об’єкти в процесі виконання сценарію. Загальна сума різних об’єктів у проекті повинна налічувати 12-15 об’єктів. Об’єкти та повідомлення на діаграмах повинні мати зрозумілі назви. При побудові діаграм використовувати прямі, рефлексивні та зворотні типи повідомлень, а також символи знищення об’єктів.

User Case №1



Use Case №2



Use Case №3

