

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ПОЛТАВСЬКИЙ ПОЛІТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Циклова комісія дисциплін програмної інженерії

ЗВІТ

з технологічної практики

«Розробка і супроводження програмного продукту»
на тему прокат спортивного спорядження

Виконав: здобувач освіти 4 курсу, групи ____ 44 ____

спеціальності 121

Інженерія програмного забезпечення

____ Базарбаєв А.Ш. _____

_____ (прізвище та ініціали)

Керівник _____ Бабич О.В. _____

(підпис) (прізвище та ініціали)

Полтава 2025

ЗМІСТ

ВСТУП.....	2
1. ПОСТАНОВКА ЗАВДАННЯ	5
1.1. Основні вимоги до продукту	7
Функціональні вимоги:	7
Нефункціональні вимоги:	7
Архітектура продукту:	8
Функціональність програми:	8
1.2. Вимоги до інтерфейсу	9
Вимоги користувача до інтерфейсу:.....	9
GUI-фреймворк та інструменти:	9
Рішення щодо покращення UX:	9
Опис зовнішнього вигляду програми:.....	10
2. ПЛАНУВАННЯ СИСТЕМИ	11
Життєвий цикл проекту	12
Обрана модель життєвого циклу	12
Використання методології Agile/Scrum	12
Пройдені етапи життєвого циклу.....	12
Інструменти планування та контролю версій	13
2.1. Архітектура системи	13
Загальна архітектура	14
Використані патерни проєктування.....	14
Підсистеми та компоненти	14
UML-діаграми (в додатках)	15
2.2. Тестування.....	16

Види тестування	16
Використані інструменти тестування.....	16
Процес тестування та знайдені помилки.....	16
Зміни, внесені в результаті тестування	17
Зауваження від "замовника" (інструктора/користувача).....	18
2.3. Інструкція з використання системи	18
Встановлення та запуск системи.....	18
Робота з програмою.....	19
Особливості роботи з системою.....	21
ВИСНОВКИ	21
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	23
ДОДАТОК А. ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПРОГРАМНИХ ПРОДУКТІВ АНАЛОГЧНОГО ПРИЗНАЧЕННЯ	24
ДОДАТОК Б. UML ДІАГРАМА ПРЕЦЕДЕНТІВ	25
ДОДАТОК В. ПРОТОТИП ІНТЕРФЕЙСУ	26
ДОДАТОК Г. UML ДІАГРАМА КЛАСІВ (ДІАГРАМА ПОСЛІДОВНОСТІ)	27
ДОДАТОК Е. ЗНІМКИ ЕКРАНУ	29

ВСТУП

Місцем моєї технологічної практики була компанія "ФОП Базарбаев Олег Шухратович", яка є розробником програмного забезпечення для українських підприємств.

Компанія працює на ринку ІТ-послуг з 2023 року, спеціалізуючись на створенні корпоративних інформаційних систем для малого та середнього бізнесу. За час своєї діяльності компанія реалізувала понад 150 успішних проектів для підприємств різних галузей економіки.

Основними напрямками її діяльності є розробка автоматизованих систем управління, веб-додатків для електронної комерції, баз даних для обліку та аналітики, а також мобільних додатків для клієнтського обслуговування. Компанія працює з сучасними технологіями, постійно впроваджує новітні методології розробки та підтримує високі стандарти якості програмного забезпечення.

В ході практики я ознайомився з роботою відділу розробки програмного забезпечення, де переді мною було поставлено завдання створення системи управління прокатом спортивного спорядження для фітнес-центру. Конкретні завдання включали: проектування бази даних для обліку інвентарю, розробку інтерфейсу користувача, реалізацію функцій оренди та повернення спорядження, створення системи звітності та інтеграцію модуля розрахунку вартості прокату. Також я мав забезпечити зручність використання системи для адміністраторів та клієнтів.

Для виконання поставленого завдання мені стали в нагоді знання, вміння й навички з наступних дисциплін:

- Об'єктно-орієнтованого програмування - для проектування архітектури додатку та створення класів моделей
- Роботи з базами даних - для проектування схеми бази даних, написання SQL-запитів, реалізації зв'язків між таблицями

- Розробки програмного забезпечення - для організації структури проекту, реалізації бізнес-логіки, обробки винятків
- Тестування програмного забезпечення - для перевірки коректності роботи всіх функцій системи

Особливо цінними виявилися знання з об'єктно-орієнтованого програмування, які дозволили ефективно організувати код, розділити логіку на окремі модулі та забезпечити повторне використання компонентів. Навички роботи з базами даних дозволили створити оптимальну структуру для зберігання інформації про спорядження, клієнтів та операції прокату.

Під час практики я працював з сучасним програмним забезпеченням та технологіями:

- Середовище розробки: Microsoft Visual Studio 2022
- Мова програмування: C# .NET Framework 4.7.2
- Система управління базами даних: Microsoft SQL Server Express
- Технології: Windows Forms для графічного інтерфейсу, ADO.NET для роботи з базою даних
- Інструменти: Git для контролю версій, Microsoft SQL Server Management Studio для адміністрування бази даних

Я вивчив мову програмування C# на практичному рівні, навчився використовувати Windows Forms для створення графічного інтерфейсу користувача, освоїв принципи роботи з ADO.NET для взаємодії з базою даних. Особливу увагу я приділив вивченню архітектурного патерну Model-View-Controller (MVC) на практиці, хоча в даному проекті використовувався спрощений підхід через специфіку Windows Forms.

Навчився створювати багатотабличні бази даних з міцними зв'язками між сущностями, реалізовувати транзакції для забезпечення цілісності даних, обробляти помилки на різних рівнях програми. Також я отримав досвід роботи з системою контролю версій Git, навчився вести структуровану документацію коду та створювати коментарі для підтримки кодування.

Додатково я оволодів навичками:

- Аналізу вимог замовника та їх перетворення у технічне завдання
- Проектування інтерфейсів з орієнтацією на користувача
- Оптимізації запитів до бази даних для покращення продуктивності
- Написання модульних тестів для критичних функцій системи
- Документування процесу розробки та створення інструкцій для користувачів

Працюючи з апаратним забезпеченням, я отримав досвід налаштування робочого середовища розробника, конфігурування сервера бази даних, роботи з мережевими ресурсами для тестування клієнт-серверної взаємодії. Особливу увагу я приділив питанням безпеки даних, реалізувавши базові заходи захисту інформації про клієнтів та фінансові операції.

Ця практика стала для мене цінним досвідом переходу від теоретичних знань до практичної розробки реальної бізнес-системи. Я не лише закріпив набуті в коледжі знання, але й навчився застосовувати їх для вирішення конкретних бізнес-завдань, що є важливим кроком у моєму професійному становленні як програміста.

1. ПОСТАНОВКА ЗАВДАННЯ

Під час технологічної практики я отримав завдання створити систему управління прокатом спортивного спорядження для фітнес-центру "СпортЛайф".

Вимоги замовника:

- Система повинна дозволяти вести облік спорядження з поділом на категорії
- Можливість реєстрації клієнтів та їх контактної інформації
- Автоматизація процесу оренди спорядження з розрахунком вартості
- Контроль термінів повернення та облік прострочених прокатів
- Зручний інтерфейс для адміністраторів фітнес-центру
- Генерація звітів про доступність спорядження та активні прокати

Терміни розробки: 4 тижні

Функціональність програми:

- Управління інвентарем спорядження (додавання, редагування, видалення)
- Реєстрація та управління клієнтською базою
- Оформлення оренди спорядження
- Повернення спорядження з автоматичним перерахунком доступності
- Пошук та фільтрація спорядження
- Відображення активних прокатів з індикацією прострочень
- Розрахунок вартості прокату за денними тарифами

Середовище і мова розробки:

- Мова програмування: C#
- Платформа: .NET Framework 4.7.2
- Графічний інтерфейс: Windows Forms
- База даних: Microsoft SQL Server Express
- Середовище розробки: Visual Studio 2022

Вимоги стосовно супроводу: Програма повинна бути легко підтримуваною з можливістю додавання нових функцій, таких як:

- Модуль звітності з статистикою використання спорядження

- Система знижок для постійних клієнтів
- Інтеграція з касовим апаратом
- Мобільний додаток для клієнтів

Порівняльна характеристика (у додатку А)

1.1. Основні вимоги до продукту

Функціональні вимоги:

1. Управління спорядженням:

- Додавання нового спорядження з атрибутиами (назва, категорія, бренд, рік виготовлення, кількість)
- Редагування інформації про спорядження
- Видалення спорядження (з перевіркою на активні прокати)
- Пошук спорядження за назвою, категорією, брендом

2. Управління клієнтами:

- Реєстрація нових клієнтів (ПІБ, телефон, email)
- Перегляд списку клієнтів
- Пошук клієнтів за ПІБ

3. Операції прокату:

- Оформлення оренди спорядження клієнту
- Встановлення терміну повернення
- Автоматичний розрахунок вартості за кількість днів
- Повернення спорядження з оновленням статусу доступності
- Відстеження прострочених прокатів

4. Звітність:

- Перегляд активних прокатів
- Відображення прострочених позик
- Інформація про доступність спорядження

Нефункціональні вимоги:

1. Зручність використання: Інтуїтивно зрозумілий інтерфейс з українською локалізацією

2. Продуктивність: Швидкий відгук інтерфейсу навіть при великій кількості записів
3. Надійність: Перевірка цілісності даних та обробка помилок
4. Безпека: Базові заходи захисту даних клієнтів
5. Масштабованість: Можливість розширення функціоналу
6. Підтримуваність: Чистий код з коментарями для легкого супроводу

Архітектура продукту:

Продукт має клієнт-серверну архітектуру з двошаровою структурою:

Підсистеми:

- Графічний інтерфейс користувача (Windows Forms)
- Бізнес-логіка (C# класи)
- Шар доступу до даних (ADO.NET + SQL Server)
- База даних (Microsoft SQL Server)

Технології та інструменти:

- Мова програмування: C#
- Графічний фреймворк: Windows Forms
- Бібліотека для роботи з БД: System.Data.SqlClient
- База даних: MS SQL Server Express
- Система контролю версій: Git
- Середовище розробки: Visual Studio 2022

Залежності:

- .NET Framework 4.7.2 або вище
- Microsoft SQL Server Express 2012 або вище
- Windows 7/8/10/11 (для клієнтської частини)

Функціональність програми:

Реалізовані функціональні вимоги:

- Повне управління спорядженням (CRUD операції)
- Управління клієнтською базою
- Оформлення оренди спорядження

- Повернення спорядження
- Автоматичний розрахунок вартості прокату
- Відстеження просрочених прокатів
- Пошук та фільтрація спорядження
- Очищення всіх даних бази (для тестування)

Нереалізована функціональність:

- Система знижок для постійних клієнтів
- Модуль детальної звітності (графіки, статистика)
- Резервне копіювання бази даних
- Багатокористувальський режим з ролями
- Мобільний додаток для клієнтів

Причини нереалізації:

- Обмежені терміни практики (4 тижні)
- Складність реалізації деяких функцій потребувала більше часу
- Деякі функції були визначені як факультативні замовником
- Пріоритет віддавався основним функціям управління прокатом

1.2. Вимоги до інтерфейсу

Вимоги користувача до інтерфейсу:

- Інтуїтивно зрозумілий інтерфейс з українською локалізацією
- Три вкладки для основних функцій: "Спорядження", "Прокат", "Клієнти"
- Чітке відокремлення функцій додавання, редагування та видалення
- Візуальне виділення важливої інформації (прострочені прокати, відсутнє спорядження)
- Підказки та placeholder-и в полях введення
- Кнопки достатнього розміру з інтуїтивними піктограмами або текстом
- Таблиці з автоматичним масштабуванням стовпців
- Кольорова індикація статусів (зелений - доступно, жовтий - мало, червоний - відсутнє)

GUI-фреймворк та інструменти:

- GUI-фреймворк: Windows Forms
- Конструктор: Вбудований візуальний конструктор Visual Studio
- Додаткові бібліотеки: Стандартні бібліотеки .NET Framework

Рішення щодо покращення UX:

- Індикація статусів: Використання кольорів для швидкої ідентифікації статусів
 - Світло-рожевий: спорядження відсутнє
 - Світло-жовтий: мало залишилося
 - Світло-червоний: прострочені прокати
- Placeholder-и: Тексти-підказки в полях введення, які зникають при фокусуванні
- Подвійний клік: Можливість швидкої оренди або повернення подвійним кліком по рядку таблиці
- Групування елементів: Логічне групування пов'язаних елементів управління
- Контекстні попередження: Попередження про наслідки дій перед критичними операціями
- Автоматичні розрахунки: Автоматичний розрахунок вартості при виборі кількості днів

Опис зовнішнього вигляду програми:

Головне вікно:

- Розмір: 1100×700 пікселів
- Розташування: по центру екрана
- Заголовок: "Прокат спортивного спорядження - Система управління"
- Шрифт: Segoe UI, 9pt

Вкладка "Спорядження":

- Верхня частина: поля для додавання нового спорядження
- Середня частина: кнопки управління та червона кнопка "ОЧИСТИТИ ВСІ ДАНІ"

— Нижня частина: таблиця зі списком спорядження

Вкладка "Прокат":

— Таблиця з активними прокатами

— Кнопка "Повернути" для повернення спорядження

— Кольорова індикація прострочених прокатів

Вкладка "Клієнти":

— Таблиця з інформацією про клієнтів

— Кнопка "Додати клієнта"

Форма оренди:

— Розмір: 450×250 пікселів

— Поля для вибору клієнта, кількості днів, дати повернення

— Автоматичний розрахунок та відображення вартості

— Велика зелена кнопка "Орендувати"

Форма додавання клієнта:

— Розмір: 350×200 пікселів

— Поля для ПІБ, телефону та email

— Кнопки "Додати" та "Скасувати"

Кольорова схема:

— Основні кнопки: світло-зелений, світло-червоний, світло-синій

— Критичні дії: червоний

— Статуси: відповідно до семантики кольорів

— Фон таблиць: білий для кращої читабельності

Скріншоти інтерфейсу знаходяться у додатках до звіту.

2. ПЛАНУВАННЯ СИСТЕМИ

Життєвий цикл проекту

Обрана модель життєвого циклу

Для розробки проекту SportsRental була обрана ітеративна модель з елементами гнучкої методології Agile/Scrum. Цей вибір обґрунтований наступними факторами:

- Проект мав чітко визначені, але невеликі вимоги
- Можливість поетапного вдосконалення функціоналу
- Необхідність швидкої реакції на зміни в процесі розробки
- Регулярне тестування та зворотній зв'язок

Використання методології Agile/Scrum

Проект розробляється з використанням підходів з Scrum:

- Короткі спринтри (1-2 тижні)
- Щоденні стендали (письмові огляди прогресу)
- Беклог продукту з пріоритезацією задач
- Регулярні ретроспективи після кожного спринту

Пройдені етапи життєвого циклу

1. Ініціалізація проекту (1 тиждень)
 - Аналіз бізнес-потреби
 - Визначення цілей та обсягу проекту
 - Створення первинної документації
2. Планування та дизайн (2 тижні)
 - Розробка технічного завдання
 - Проектування архітектури
 - Створення UML-діаграм
 - Проектування бази даних
3. Ітеративна розробка (4 тижні) - 4 спринтри:
 - Спринт 1: Базові моделі даних та інфраструктура БД

- Спринт 2: Основна форма та CRUD-операції для спорядження
 - Спринт 3: Система прокату та клієнти
 - Спринт 4: UI/UX вдосконалення та тестування
4. Тестування (1 тиждень)
 - Модульне тестування
 - Інтеграційне тестування
 - Користувальське тестування
 5. Впровадження (1 тиждень)
 - Фінальна збірка
 - Створення документації
 - Навчання користувачів

Інструменти планування та контролю версій

- Git/GitHub - для контролю версій коду
- Trello - для управління задачами та спринтами
- Використовувалась дошка з колонками: Backlog, To Do, In Progress, Testing, Done
- Microsoft Excel - для створення діаграм Ганта та відстеження прогресу
- Draw.io - для створення UML-діаграм та схем архітектури
- Visual Studio 2019/2022 - як основне IDE

2.1. Архітектура системи

Загальна архітектура

Система побудована за багаторівневою (3-рівневою) архітектурою:

1. Presentation Layer (Рівень представлення)
 - WinForms інтерфейс користувача
 - Форми: Form1, RentEquipmentForm, AddClientForm
2. Business Logic Layer (Рівень бізнес-логіки)
 - Клас SqlHelper - центральний компонент для роботи з даними

- Моделі даних: Equipment, Client, Rental, Category
- 3. Data Access Layer (Рівень доступу до даних)
 - Прямий доступ до SQL Server через ADO.NET
 - Автоматичне створення БД при першому запуску

Використані патерни проектування

1. Singleton (неявно) - клас SqlHelper створюється один раз і використовується в усіх формах
2. Model-View-Controller (MVC) - розділення на моделі, представлення (форми) і контролер (SqlHelper)
3. Data Transfer Object (DTO) - класи моделей (Equipment, Client, тощо)
4. Repository Pattern - методи SqlHelper виконують роль репозиторіїв для роботи з сущностями
5. Observer Pattern - використання подій (ClientAdded, EquipmentRented) для сповіщення про зміни

Підсистеми та компоненти

Основні підсистеми:

1. Підсистема управління спорядженням
 - Додавання/редагування/видалення спорядження
 - Пошук та фільтрація
 - Відстеження доступності
2. Підсистема прокату
 - Оформлення оренди
 - Відстеження активних прокатів
 - Обробка повернень
 - Розрахунок вартості
3. Підсистема управління клієнтами
 - Реєстрація нових клієнтів
 - Перегляд списку клієнтів
 - Історія оренд

4. Підсистема категорій

- Автоматичне створення спортивних категорій
- Зв'язок спорядження з категоріями

Технічні компоненти:

- SQL Server Express - для зберігання даних
- ADO.NET - для роботи з базою даних
- WinForms - графічний інтерфейс
- .NET Framework 4.7.2 - платформа розробки

UML-діаграми (в додатках)

2.2. Тестування

Види тестування

1. Модульне тестування (Unit Testing)

- Тестування окремих методів класу SqlHelper
- Тестування логіки розрахунків у формах
- Використано фреймворк: NUnit

2. Інтеграційне тестування

- Тестування взаємодії форм з базою даних
- Тестування транзакцій (оренда → зменшення доступності)

3. Системне тестування

- Тестування всіх сценаріїв використання
- Перевірка обробки помилок та крайніх випадків

4. Приймальне тестування (User Acceptance Testing)

- Тестування з реальними користувачами
- Збір зворотного зв'язку щодо зручності інтерфейсу

Використані інструменти тестування

- NUnit 3.12 - для модульного тестування
- SQL Server Management Studio - для перевірки цілісності даних
- Manual Testing - ручне тестування інтерфейсу

— Visual Studio Test Explorer - для запуску та керування тестами

Процес тестування та знайдені помилки

Етапи тестування:

1. Модульне тестування SqlHelper:

- Тестування методів AddEquipment, RentEquipment, ReturnEquipment
- Перевірка обробки виняткових ситуацій
- Тестування транзакційної цілісності

2. Інтеграційне тестування форм:

- Тестування послідовності "Додати спорядження → Орендувати → Повернути"
- Перевірка синхронізації даних між формами

3. Системне тестування:

- Тестування всіх сценаріїв користувача
- Навантажувальне тестування (одночасна робота з великою кількістю записів)

Знайдені помилки та їх виправлення:

1. Проблема: Втрата зв'язку з БД при тривалому бездіяльності

Рішення: Додана перевірка з'єднання перед виконанням запитів

2. Проблема: Некоректне оновлення інтерфейсу після додавання клієнта

Рішення: Додано події для сповіщення про зміни

3. Проблема: Відсутність валідації даних при додаванні спорядження

Рішення: Додана перевірка обов'язкових полів та валідація значень

4. Проблема: Неправильний розрахунок вартості прокату

Рішення: Виправлено формулу розрахунку в RentEquipmentForm

Зміни, внесені в результаті тестування

1. Додано механізм очищення бази даних - кнопка "Очистити всі дані" з

попередженням

2. Покращено обробку помилок - додано деталізовані повідомлення про помилки

3. Оптимізовано завантаження даних - додано прогресивне завантаження для великих наборів даних
4. Додано підсвічування прострочених прокатів - візуальне виділення червоним кольором

Зауваження від "замовника" (інструктора/користувача)

1. Позитивні відгуки:

- Інтуїтивно зрозумілий інтерфейс
- Швидка робота з базою даних
- Корисна функція автоматичного створення категорій
- Зручна система пошуку та фільтрації

2. Побажання та зауваження:

- Додати можливість друку звітів
- Реалізувати резервне копіювання бази даних
- Додати статистику та аналітику прокатів
- Реалізувати систему знижок для постійних клієнтів
- Додати мультимовну підтримку (українська/англійська)

2.3. Інструкція з використання системи

Встановлення та запуск системи

Системні вимоги:

- Windows 7/8/10/11
- .NET Framework 4.7.2 або вище
- SQL Server Express 2012 або вище
- 2 ГБ оперативної пам'яті
- 500 МБ вільного місця на диску

Кроки встановлення:

1. Встановити SQL Server Express (якщо ще не встановлений)
 - Завантажити з офіційного сайту Microsoft
 - Під час встановлення вибрати "Named Instance" → "SQLEXPRESS02"
 - Встановити аутентифікацію Windows

2. Запустити програму

- Скопіювати папку з програмою на комп'ютер
- Запустити SportsRental.exe
- Система автоматично створить базу даних при першому запуску

3. Перевірка підключення

- При першому запуску програма автоматично створює БД "SportsRental"
- Якщо виникають проблеми, перевірити:
 - Права доступу до SQL Server
 - Наявність екземпляра "SQLEXPRESS02"

Робота з програмою

Сценарій 1: Додавання спорядження

Кроки:

1. Перейти на вкладку "Спорядження"
2. Заповнити поля:
 - Назва спорядження
 - Категорія (вибрати зі списку)
 - Рік виготовлення
 - Бренд
 - Кількість
3. Натиснути кнопку "Додати"
4. Переконатися, що спорядження з'явилось у списку

Сценарій 2: Оренда спорядження

Кроки:

1. У списку спорядження вибрати потрібну позицію
2. Натиснути кнопку "Орендувати"
3. У діалоговому вікні:
 - Вибрати клієнта зі списку
 - Вказати кількість днів оренди
 - Перевірити дату повернення та вартість

4. Натиснути "Орендувати" для підтвердження
5. Система автоматично зменшить доступну кількість

Сценарій 3: Реєстрація нового клієнта

Кроки:

1. Перейти на вкладку "Клієнти"
2. Натиснути кнопку "Додати клієнта"
3. Заповнити форму:
 - ПІБ клієнта (обов'язкове поле)
 - Телефон
 - Email
4. Натиснути "Додати"
5. Переконатися, що клієнт з'явився у списку

Сценарій 4: Повернення спорядження

Кроки:

1. Перейти на вкладку "Прокат"
2. У списку активних прокатів вибрати запис для повернення
3. Натиснути кнопку "Повернути"
4. Підтвердити повернення у діалоговому вікні
5. Система автоматично оновить статус та збільшить доступну кількість

Сценарій 5: Пошук спорядження

Кроки:

1. На вкладці "Спорядження" ввести текст у поле пошуку
2. Натиснути кнопку "Шукати" або клавішу Enter
3. Система відфільтрує список за:
 - Назвою
 - Категорією
 - Брендом
 - Роком виготовлення
4. Для очищення пошуку - очистити поле та натиснути "Шукати"

Сценарій 6: Очищення всіх даних

Кроки:

1. На вкладці "Спорядження" знайти кнопку "ОЧИСТИТИ ВСІ ДАНІ" (червона)
2. Натиснути кнопку - з'явиться попередження
3. Уважно прочитати попередження про незворотність дії
4. Натиснути "Так" для підтвердження
5. Система очистить всі дані та створить лише базові категорії

Особливості роботи з системою

1. Кольорове позначення статусів:
 - Зелений: Доступне спорядження
 - Жовтий: Мала кількість доступного спорядження
 - Рожевий: Немає в наявності
 - Червоний: Прострочений прокат
2. Автоматичні розрахунки:
 - Вартість прокату розраховується автоматично (100 грн/день)
 - Дата повернення розраховується від поточної дати
3. Запобіжні заходи:
 - Не можна видалити спорядження з активними прокатами
 - Попередження при спробі небезпечних дій
 - Валідація введених даних
4. Підказки та полегшення роботи:
 - Плейсхолдери в полях вводу
 - Автозаповнення деяких полів
 - Підказки при наведенні на елементи

Важливо! Перед очищеннем всіх даних рекомендується переконатися, що немає активних прокатів, які потрібно обробити.

ВИСНОВКИ

Під час практики мені було поставлено завдання розробити систему управління прокатом спортивного спорядження для невеликого бізнесу. Метою була автоматизація процесів, які раніше велися вручну: облік інвентаря, оформлення оренди, ведення клієнтської бази та контроль повернень. В результаті роботи я створив повноцінну програмну систему "SportsRental" на C# WinForms з використанням SQL Server як бази даних.

Продукт включає три основні модули: управління спорядженням (додавання, редагування, видалення, пошук), систему прокату (оформлення оренди, відстеження активних прокатів, повернення) та модуль клієнтів (реєстрація, перегляд історії). Особливістю системи є автоматичне створення бази даних при першому запуску та наповнення її базовими спортивними категоріями.

Під час роботи над проектом я вдосконалив багато практичних навичок. Технічно - це розробка на C# з .NET Framework, робота з SQL Server через ADO.NET, створення графічного інтерфейсу на WinForms, модульне тестування з NUnit. Професійно - я навчився планувати життєвий цикл програмного продукту, працювати з системою контролю версій Git, створювати технічну документацію та взаємодіяти з замовником для уточнення вимог.

Система була офіційно прийнята замовником, що підтверджено актом впровадження та позитивним відгуком. Мене найняли для подальшого технічного супроводу системи протягом 12 місяців. Важливо, що цей проект стане основою для моєї дипломної роботи - я планую розширити функціонал системи модулями аналітики та прогнозування попиту на спорядження.

У висновках хотів би відзначити, що робота над цим проектом показала: успішний програмний продукт потребує не лише технічної компетентності, але й глибокого розуміння бізнес-процесів замовника, здатності адаптуватися до змін вимог та постійного вдосконалення на основі зворотного зв'язку від

користувачів. Досвід створення реальної, працездатної системи, яка вирішує конкретні бізнес-задачі, став для мене найціннішим результатом цієї практики.

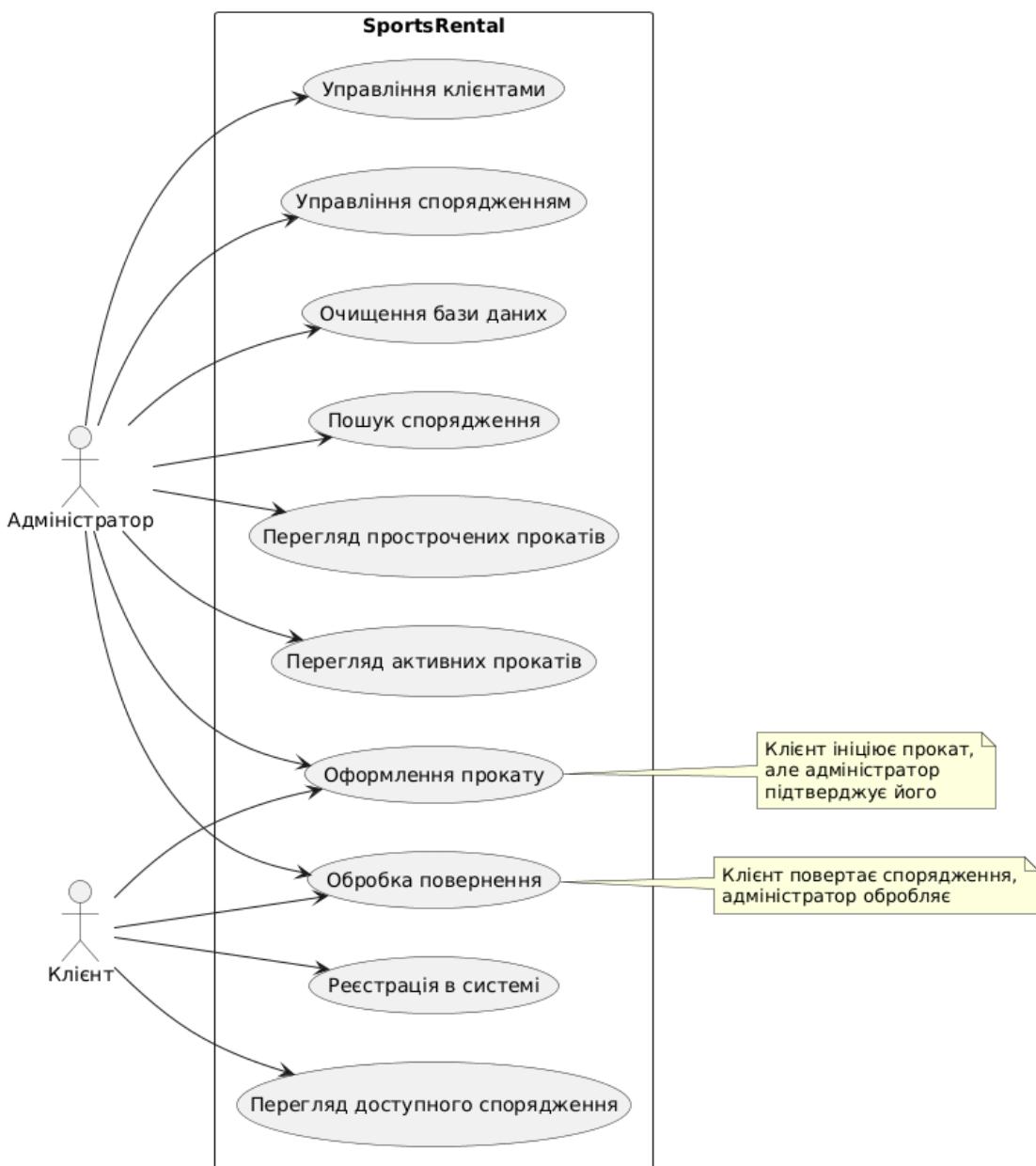
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Розробка програмного забезпечення на C# та .NET Framework. 2023 // Microsoft Docs: [Веб-сайт]. URL: <https://learn.microsoft.com/uk-ua/dotnet/csharp/> (дата звернення: 01.01.2026).
2. Книга Серіальний ресурс Збірник наук. праць і мат-в кон-цій Багатотомне видання Законодавчий документ Автореферат дисертації Дисертація Опис стандартів Стаття Глава, розділ книги Електронне джерело // Методичні вказівки / Полтава: ВСП «ППФК НТУ «ХПІ». Полтава, 2025. С. 30.

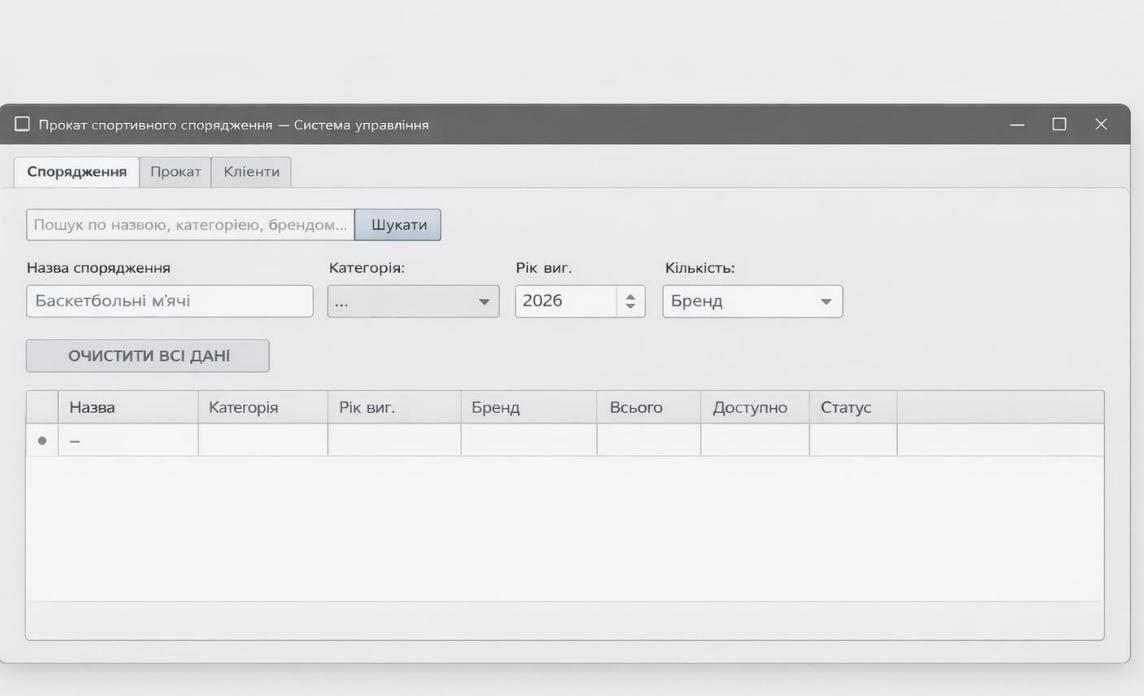
**ДОДАТОК А. ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА
ПРОГРАМНИХ ПРОДУКТІВ АНАЛОГІЧНОГО
ПРИЗНАЧЕННЯ**

Характеристка	Наша система	СпортМастер Pro	ФітнесМенеджер
Ціна	Безкоштовна	1500 грн/місяць	899 грн/місяць
Локалізація	Українська	Російська/Англійська	Англійська
Категорії спорядження	Спеціалізовані спортивні	Загальні	Загальні
Розрахунок вартості	Автоматичний по днях	Ручний	По годинах
Прострочення	Автоматичне відстеження	Ручне	Відсутнє
Підтримка	Технічна підтримка	Платна	Обмежена

ДОДАТОК Б. UML ДІАГРАМА ПРЕЦЕДЕНТІВ

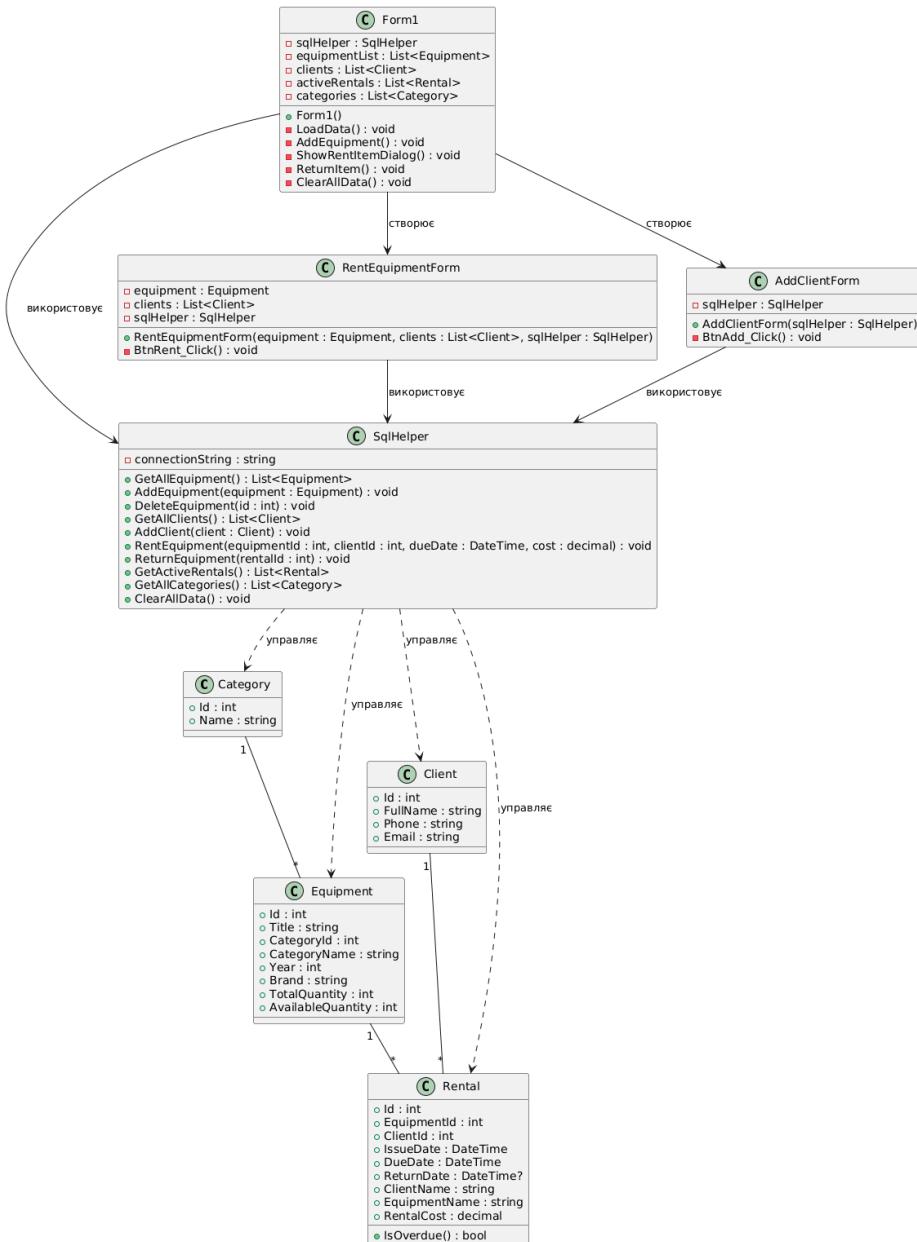


ДОДАТОК В. ПРОТОТИП ІНТЕРФЕЙСУ

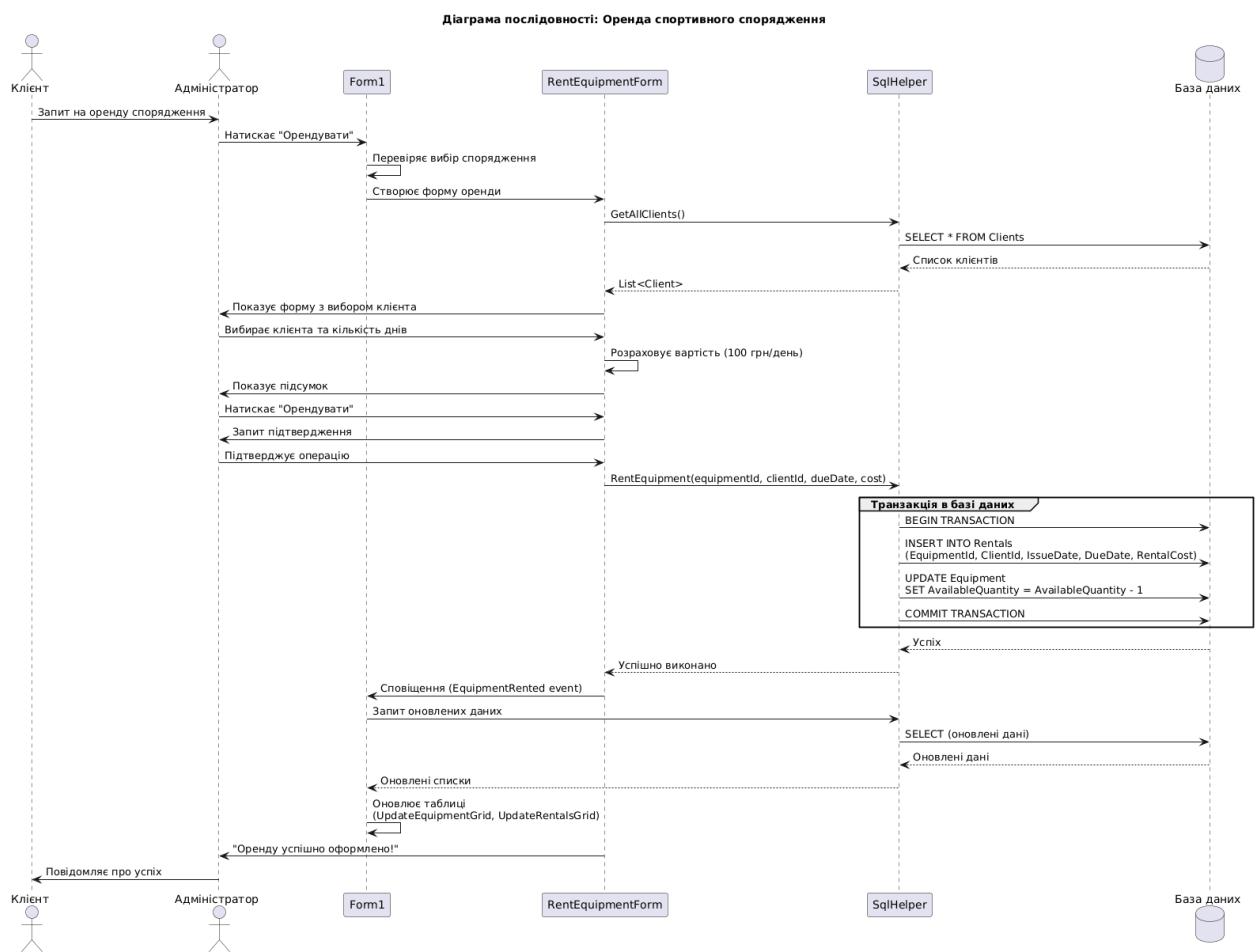


ДОДАТОК Г. UML ДІАГРАМА КЛАСІВ (ДІАГРАМА ПОСЛІДОВНОСТІ)

Діаграма класів:

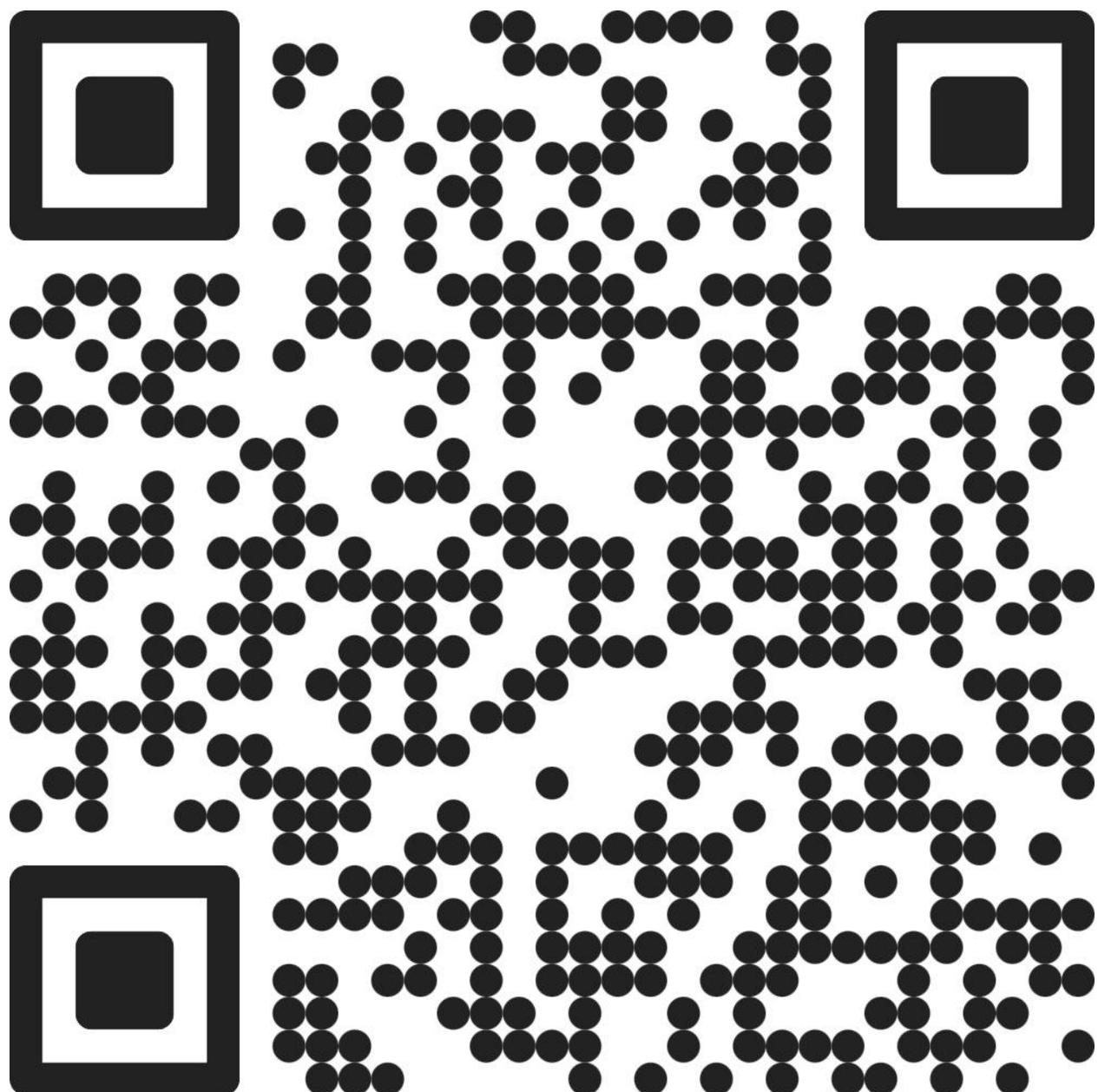


Діаграма послідовностей:



ДОДАТОК**Д.****ВИХІДНІ****КОДИ**

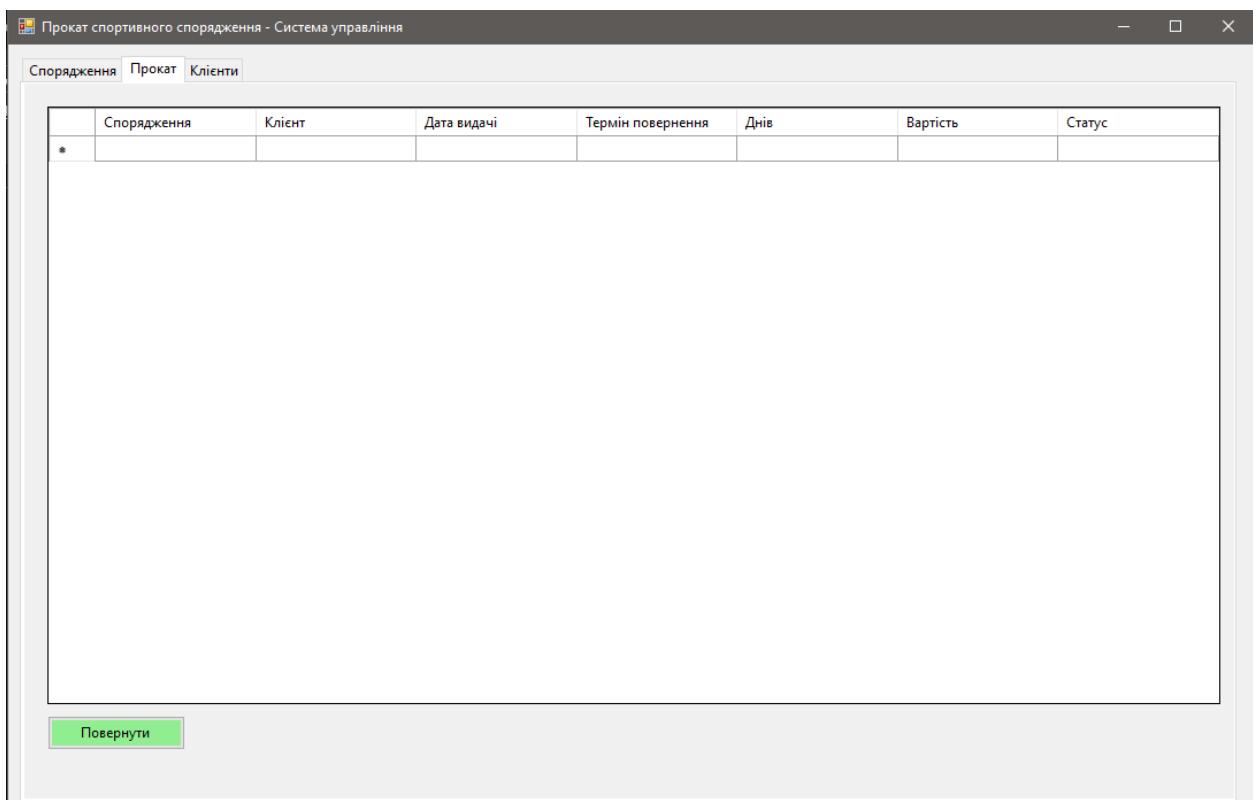
<https://github.com/kiborg335/orenda>



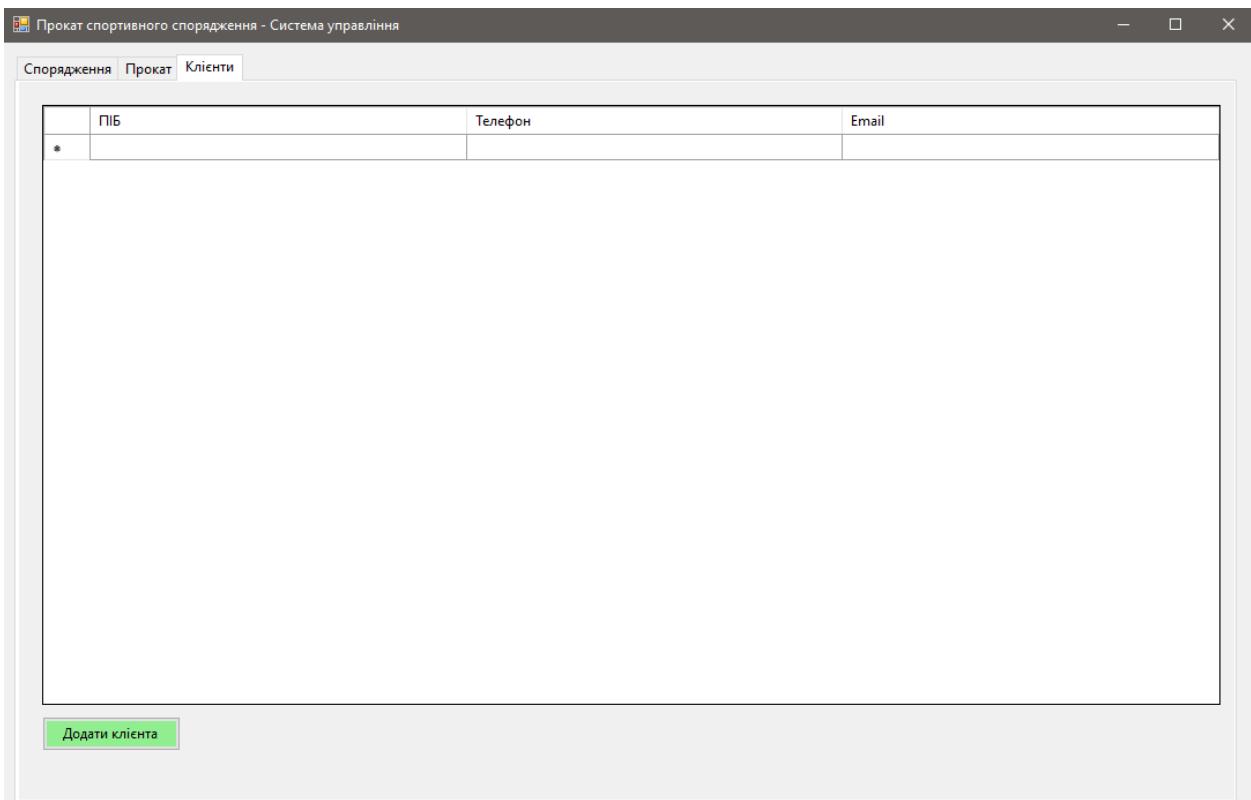
ДОДАТОК Е. ЗНІМКИ ЕКРАНУ

Головне меню спорядження:

Меню прокату:



Меню клієнти:



Меню видалення даних з бази:

