

Державний торговельно-економічний університет  
Кафедра інженерії програмного забезпечення та кібербезпеки

## ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Програмна компонента інформаційної системи фітнес клубу»**

Студента 4 курсу, 7 групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»,  
освітньої програми  
«Інженерія програмного  
забезпечення»

Потапова Дмитра  
Ростиславовича

*підпис студента*

Науковий керівник  
кандидат технічних наук,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

Рзаєва Світлана  
Леонідівна

*підпис керівника*

Гарант освітньої програми  
кандидат технічних наук,  
доцент кафедри інженерії  
програмного забезпечення  
та кібербезпеки

Рзаєва Світлана  
Леонідівна

*підпис гаранта*

**КИЇВ – 2022**

**Державний торговельно-економічний університет**

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Спеціалізація / освітня програма Інженерія програмного забезпечення

**Затверджую**

Зав. кафедри інженерії програмного  
забезпечення та кібербезпеки

Криворучко О. В.

"09" листопада 2021р.

**Завдання**

**на випускн у кваліфікаційну роботу студентів**

Потанову Дмитру Ростиславовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи Програмна компонента  
інформаційної системи фітнес клубу.

Затверджена наказом ректора від "29" листопада 2021 р. № 12

2. Строк здачі студентом закінченої роботи 06.06.2022

3. Цільова установка та вихідні дані до роботи

Мета роботи: підвищення продуктивності праці співробітників спортивного  
клубу, чого можна досягти за рахунок впровадження системи обліку та  
контролю занять клієнтів клубу.

Об'єкт дослідження: процес обліку та контролю занять клієнтів  
спортивного фітнес-клубу.

Предмет дослідження: методи та засоби, що можуть бути використані для  
обліку та контролю занять клієнтів спортивного фітнес-клубу.

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1			
2			
3			

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

РОЗДІЛ 1 АНАЛІЗ ПРОЦЕСІВ ФУНКЦІОНУВАННЯ СПОРТИВНОГО КЛУБУ

Типові бізнес-процеси, що протікають у фітнес-клубі

Аналіз програмних продуктів, що можуть бути застосовані для автоматизації бізнес-процесів фітнес-клубу

Постановка завдання на розробку

Висновки до розділу 1

РОЗДІЛ 2 ОБГРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ ПРИ СТВОРЕННІ ПРОГРАМНОЇ КОМПОНЕНТИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ФІТНЕС-КЛУБУ

Моделювання предметної галузі, яка розглядається у даній роботі

Вибір технологій і засобів розробки

Обґрунтування вибору архітектури програмної компоненти інформаційної системи фітнес-клубу

Вибір технології розробки програмної компоненти інформаційної системи фітнес-клубу

Вибір мови програмування та короткий опис її особливостей

Обґрунтування вибору системи управління базами даних

Проектування бази даних програмної компоненти інформаційної системи фітнес-клубу

Висновки до розділу 2

РОЗДІЛ 3 ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ КОМПОНЕНТИ  
ІНФОРМАЦІЙНОЇ СИСТЕМИ ФІТНЕС-КЛУБУ

Проектування інтерфейсу користувача програмної компоненти інформаційної  
системи фітнес-клубу

Реалізація бази даних програмної компоненти інформаційної системи фітнес-  
клубу

Реалізація розроблених алгоритмів роботи програмної компоненти  
інформаційної системи фітнес-клубу

Тестування створеної програмної компоненти інформаційної системи фітнес-  
клубу

Розробка керівництва користувача розробленої програмної компоненти  
інформаційної системи фітнес-клубу

Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

## 6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	20.09.2021	20.09.2021
2.	<i>Вступ та список літературних джерел</i>	13.12.2021	13.12.2021
3.	<i>Розділ 1. «Аналіз процесів функціонування спортивного клубу»</i>	18.02.2022	18.02.2022
4.	<i>Розділ 2. «Обґрунтування проектних рішень при створенні програмної компоненти інформаційної системи фітнес-клубу»</i>	04.03.2022	04.03.2022
5.	<i>Розділ 3. «Особливості реалізації програмної компоненти інформаційної системи фітнес-клубу»</i>	11.04.2022	11.04.2022
6.	<i>Висновки та пропозиції</i>	25.04.2022	25.04.2022
7.	<i>Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)</i>	25.05.2022	25.05.2022
8.	<i>Підготовка автореферату та презентації доповіді</i>	26.05.2022	26.05.2022
9.	<i>Попередній захист випускної кваліфікаційної роботи</i>	30.05.2022-03.06.2022	01.06.2022
10.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	04.06.2022	04.06.2022
11.	<i>Здача прошого випускної кваліфікаційної роботи на кафедру</i>	06.06.2022	06.06.2022
12.	<i>Публічний захист випускної кваліфікаційної роботи</i>		

7. Дата видачі завдання «09» листопада 2021 р.

8. Науковий керівник випускної кваліфікаційної роботи

*Рзаєва С.Л.*

(прізвище, ініціали, підпис)

9. Гарант освітньої програми

*Рзаєва С.Л.*

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент

*Потапов Д. Р.*

(прізвище, ініціали, підпис)



## АНОТАЦІЯ

Робота присвячена дослідженню використання сучасних інформаційних технологій у сфері обліку та аудиту, проектування та реалізації основних компонентів програмного забезпечення, його впровадження для контролю занять клієнтів спортивного фітнес-клубу та в свою чергу підвищенню продуктивності праці співробітників спортивного клубу.

Розробка здійснена з використанням мови програмування C#, СУБД MySQL.

Випускна кваліфікаційна робота на тему «Програмна компонента інформаційної системи фітнес клубу» загальним обсягом 59 сторінок містить:

- 29 рисунок
- 1 таблиця

**Ключові слова:** додаток, C#, база даних, облік.

## ANNOTATION

The work is devoted to the research of the use of modern information technologies in the field of accounting and auditing, design and development of key software components, its implementation to control the activities of clients of sports fitness club and in turn increase productivity of sports club staff.

Development was carried out using the C# programming language, MySQL database.

The final qualifying work on the topic "Software component of the information system of the fitness club" with a total volume of 59 pages contains:

- 29 figures
- 1 table

**Keywords:** application, C#, database, accounting.

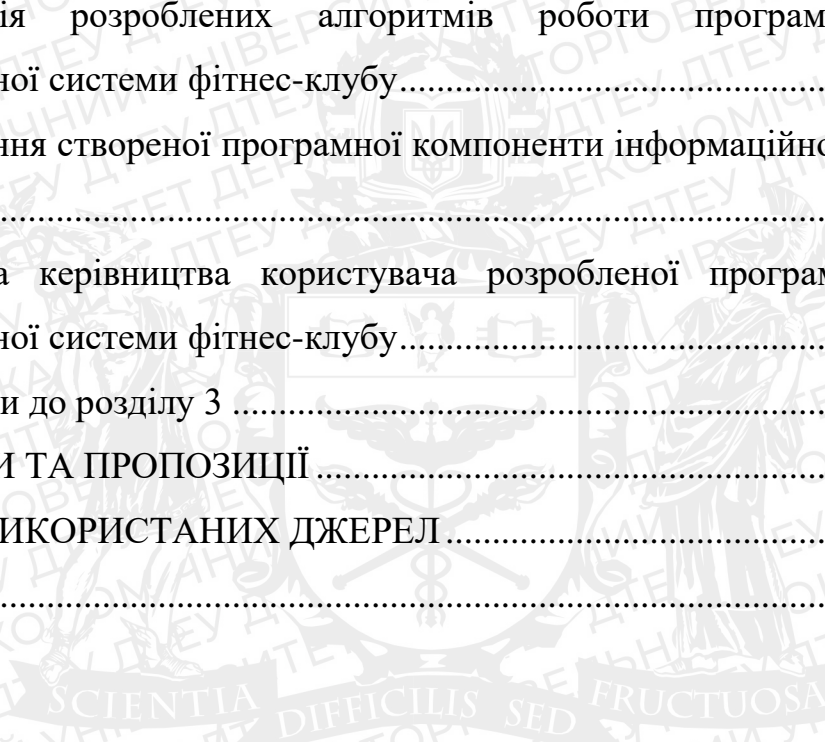
## ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1 .....	6
АНАЛІЗ ПРОЦЕСІВ ФУНКЦІОНУВАННЯ СПОРТИВНОГО КЛУБУ .....	6
1.1 Типові бізнес-процеси, що протікають у фітнес-клубі .....	6
1.2 Аналіз програмних продуктів, що можуть бути застосовані для автоматизації бізнес-процесів фітнес-клубу .....	10
1.3 Постановка завдання на розробку .....	21
1.4 Висновки до розділу 1 .....	23
РОЗДІЛ 2 .....	24
ОБГРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ ПРИ СТВОРЕННІ ПРОГРАМНОЇ КОМПОНЕНТИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ФІТНЕС-КЛУБУ .....	24
2.1 Моделювання предметної галузі, яка розглядається у даній роботі.....	24
2.2 Вибір технологій і засобів розробки .....	28
2.2.1 Обґрунтування вибору архітектури програмної компоненти інформаційної системи фітнес-клубу.....	28
2.2.2 Вибір технології розробки програмної компоненти інформаційної системи фітнес-клубу.....	30
2.2.3 Вибір мови програмування та короткий опис її особливостей .....	33
2.2.4 Обґрунтування вибору системи управління базами даних.....	35
2.3 Проектування бази даних програмної компоненти інформаційної системи фітнес-клубу.....	36
2.4 Висновки до розділу 2 .....	38

					<i>ДТЕУ 121 07-12.БР</i>			
					<i>Програмна компонента інформаційної системи фітнес клубу</i>	<i>Стадія</i>	<i>Аркуші</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуші</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		Зміст	2	59
Зав. каф.		Криворучко О.В.		25.04.22				
Керівник		Рзаєва С. Л.		25.04.22				
Гарант		Рзаєва С. Л.		25.04.22				
Розроб.		Потапов Д. Р.		25.04.22	<i>Зміст</i>	Факультет інформаційних технологій, 4 курс, 7 група		



РОЗДІЛ 3 .....	39
ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ КОМПОНЕНТИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ФІТНЕС-КЛУБУ .....	39
3.1 Проектування інтерфейсу користувача програмної компоненти інформаційної системи фітнес-клубу.....	39
3.2 Реалізація бази даних програмної компоненти інформаційної системи фітнес-клубу .....	43
3.3 Реалізація розроблених алгоритмів роботи програмної компоненти інформаційної системи фітнес-клубу.....	45
3.4 Тестування створеної програмної компоненти інформаційної системи фітнес-клубу .....	47
3.5 Розробка керівництва користувача розробленої програмної компоненти інформаційної системи фітнес-клубу.....	52
3.6 Висновки до розділу 3 .....	55
ВИСНОВКИ ТА ПРОПОЗИЦІЇ .....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	58
ДОДАТКИ.....	60



									Аркуш
									3
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР				

## ВСТУП

**Актуальність дослідження.** У сучасній економіці усе більше підприємств і організацій розуміють переваги автоматизації власних бізнес-процесів та операцій, що здійснюється шляхом використання відповідного програмного забезпечення (ПЗ). Таке ПЗ дозволяє підприємству більш ефективно використовувати власні трудові ресурси, підвищити ефективність управлінської діяльності й більш зручно й легко вести електронний облік.

Сфера фітнес-тренувань не є виключенням. Тренажерні зали є частиною ринку по наданню спортивних і оздоровчих послуг. На сьогоднішній день послуги тренажерних залів знаходять усе більший попит у великої кількості людей і з кожним роком їх кількість зростає. Оскільки попит на фітнес послуги серед населення розвинутих країн поступово зростає, збільшується кількість клієнтів, і, як наслідок, обсяг інформації, оброблюваної співробітниками клубу, стає усе більше. Відповідно, обробляти інформацію вручну стає усе складніше, що позначається на продуктивності працівників фітнес-клубу. Це викликає необхідність автоматизації основних бізнес-процесів, що дозволяє оптимізувати процеси обробки інформації, підвищити якість обслуговування й забезпечити надання інформації для прийняття управлінських рішень.

Таким чином, задача впровадження системи обліку та супроводження занять для фітнес-клубу є вкрай **актуальною**.

**Метою** роботи є підвищення продуктивності праці співробітників спортивного клубу, чого можна досягти за рахунок впровадження системи обліку та контролю занять клієнтів клубу. Відповідно до мети були визначені наступні окремі **завдання роботи**:

					<i>ДТЕУ 121 07-12.БР</i>			
					<i>Програмна компонента інформаційної системи фітнес клубу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		В	4	59
Зав. каф.		Криворучко О.В.		13.12.21	<i>Вступ</i>	Факультет інформаційних технологій, 4 курс, 7 група		
Керівник		Рзаєва С. Л.		13.12.21				
Гарант		Рзаєва С. Л.		13.12.21				
Розроб.		Потапов Д. Р.		13.12.21				

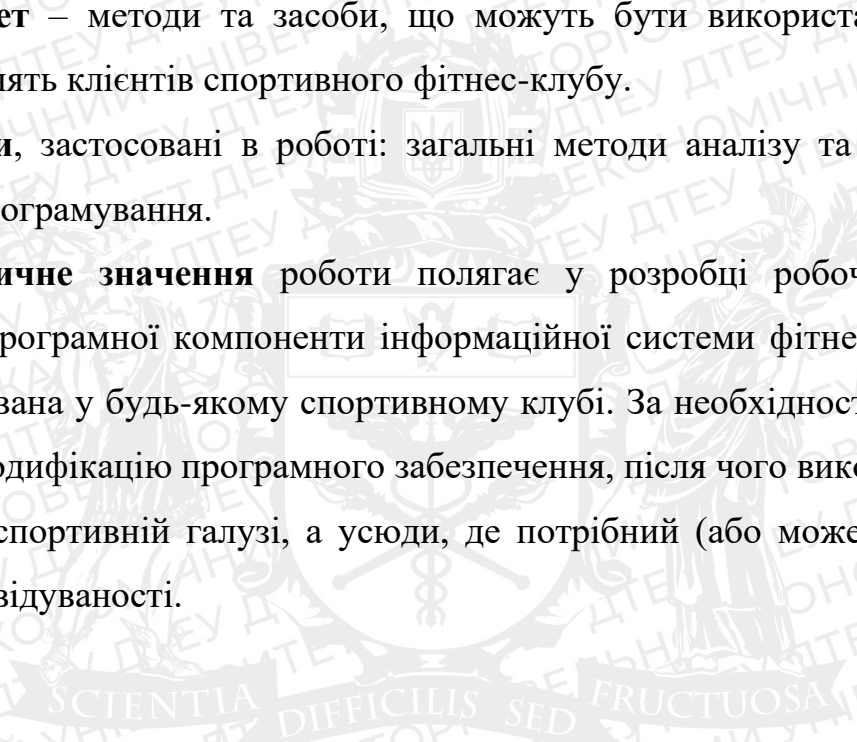
- провести аналіз процесів функціонування фітнес-клубу та встановити вид та структуру інформації, що підлягає обробці та збереженню у проектованій системі;
- здійснити розробку проектних рішень, необхідних для зведення проектованої системи обліку та контролю;
- провести програмну реалізацію, протестувати її та оцінити ефект від її використання.

**Об’єкт дослідження** – процес обліку та контролю занять клієнтів спортивного фітнес-клубу.

**Предмет** – методи та засоби, що можуть бути використані для обліку та контролю занять клієнтів спортивного фітнес-клубу.

**Методи**, застосовані в роботі: загальні методи аналізу та синтезу, а також технології програмування.

**Практичне значення** роботи полягає у розробці робочого програмного продукту - програмної компоненти інформаційної системи фітнес-клубу, що може бути застосована у будь-якому спортивному клубі. За необхідності можна провести невеличку модифікацію програмного забезпечення, після чого використовувати його не тільки у спортивній галузі, а усюди, де потрібний (або може застосовуватися) контроль відвідуваності.



						Аркуш
						5
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР	

## РОЗДІЛ 1

### АНАЛІЗ ПРОЦЕСІВ ФУНКЦІОНУВАННЯ СПОРТИВНОГО КЛУБУ

#### 1.1 Типові бізнес-процеси, що протікають у фітнес-клубі

В першу чергу відмітимо, що фітнес-клуб зазвичай є організацією, яка створюється у формі підприємства, тобто має на меті отримання прибутку (благодійні, комунальні, сімейні та інші нестандартні форми організації фітнес-клубів в подальшому не розглядатимемо через їх надзвичайно малу поширеність). Говорячи про особливості таких підприємств можна сказати, що вони здійснюють господарську діяльність у сфері надання послуг населенню в галузі фізкультури та спорту [1], КВЕД 93.12 «Діяльність спортивних клубів» та 93.13 «Діяльність фітнес-центрів» за класифікатором КВЕД-2010 [2]. Відповідно, всередині фітнес-клубу, як і на будь-якому підприємстві, протікає ряд бізнес-процесів, що можуть бути у більшій, або меншій степені автоматизовані, і які через це слід розглянути докладніше.

В першу чергу автоматизації на підприємстві підлягає облік у самому загальному сенсі даного поняття, що, відповідно до економічної теорії [3] визначається як «віддзеркалення господарської або іншої діяльності підприємства на підставі документів в різних вимірниках (кількісних і (або) якісних)». Таким чином, під «обліком» в даній роботі матимемо на увазі збереження інформації про будь-які аспекти діяльності фітнес-клубу, представленої у кількісних та якісних вимірниках.

					<i>ДТЕУ 121 07-12.БР</i>			
					<i>Програмна компонента інформаційної системи фітнес клубу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		Р1	6	59
Зав. каф.		Криворучко О.В.		18.02.22	<i>Аналіз процесів функціонування спортивного клубу</i>	Факультет інформаційних технологій, 4 курс, 7 група		
Керівник		Рзаєва С. Л.		18.02.22				
Гарант		Рзаєва С. Л.		18.02.22				
Розроб.		Потапов Д. Р.		18.02.22				

Система обліку на сьогоднішній день є одним з найважливіших елементів комплексу бізнес інструментів організації.

Без неї практично неможливо впоратися з обліком на сучасному підприємстві. Дійсно, керівникам різних рівнів важливо знати "картину" щоденного, а буває й щохвилинного стану підприємства без чого ефективно управління буде неможливим. Відповідне програмне забезпечення дозволяє використовувати не тільки облікові методики, але й планувати виробничу й логістичну діяльність, моделювати реальні комерційні й облікові ситуації відповідно до сучасних вимог до подібних завдань.

На сьогоднішній день у світі програмних засобів автоматизації існує величезна кількість систем, що мають у назві слово "облік". Високоякісні системи із широкими можливостями широко використовуються великими компаніями, але їхня вартість для малих підприємств (яким по суті і є довільний фітнес-клуб) зазвичай виявляється дуже високою. Такі системи автоматизації мають великий досвід впровадження, перевірені безліччю практик різних підприємств, налагоджені технології підтримки й довели свою ефективність. Однак, для малих компаній іноді більш доцільним є використання безкоштовних програмних продуктів або навіть розробка власних програмних продуктів, що мають строго визначену, необхідну їм функціональність.

На сьогоднішній день у розвинутих інформаційних системах передбачена автоматизація наступних процесів:

- робота із клієнтами;
- робота з постачальниками;
- робота з фінансами;
- робота зі складом;
- робота зі звітністю;
- робота з персоналом;
- облік маркетингових підприємств (СМС і e-mail розсилання);
- обмін даними з іншими інформаційними базами;
- робота з додатковими послугами;
- можливість підключення необхідного встаткування;
- робота з аналітикою.

						ДТЕУ 121 07-12.БР	Аркуш
Изм.	Аркуш.	№ докум.	Підпис	Дата			7

Важливими умовами популярності інформаційних систем автоматизації діяльності підприємств є забезпечення можливості:

- розширення функціонала інформаційної системи;
- зміни й настроювання інтерфейсу для більш зручного й зрозумілого використання;
- розширення необхідних настроювань програмного забезпечення;
- розвиток інтеграції з різним спеціалізованим устаткуванням (торговельним, охоронним, реєстраційним, тощо).

Якщо говорити про облік саме у спортивній сфері, то по мірі розвитку спортивно-оздоровчої організації, її співробітникам усе складніше бачити глобальну картину, що відбувається у клубі, вести контроль клієнтів, статистику, спостерігати за сучасними тенденціями у спортивно-оздоровчій галузі, оперативно ухвалювати рішення і пропонувати нові послуги та продукти, які пропонує фітнес-клуб власним клієнтам (зокрема, акції), прогнозувати попит на послуги, які надає фітнес-клуб (а, отже, завантаженість приміщень та задіяних тренерів) і планувати нові пропозиції. Комплексна автоматизація процесів роботи фітнес-клубу допоможе впоратися із цими завданнями й забезпечити постійний розвиток підприємства.

Взагалі кажучи, завдання автоматизації того або іншого закладу сфери послуг визначаються його бізнесами–процесами. У випадку автоматизації фітнес-клубів є кілька особливостей, що відрізняють їх від інших спортивних організацій.

Насамперед, такою особливістю є необхідність індивідуального підходу до клієнта. Якщо, приміром, на гірськолижній трасі система може просто регулювати прохід до підйомника, не вимагаючи персоналізації, то у фітнес-клуб клієнт іде за певною процедурою, і навіть до певного тренера. Відвідувач фітнес-клубу оплачує послуги й розраховує одержати за ці гроші індивідуальне обслуговування, розраховане саме на його потреби. Йому важливий час надання послуги, місце, фахівець, що виявляє послугу, а не просто "доступ" до тренажера.

По-друге, важлива в цьому випадку є можливість забронювати послугу. У третій, фітнес-клуби постійно розширюють спектр пропонованих послуг: групові спортивні, силові, танцювальні заняття, водні програми, послуги спортивного клубу,

						Аркуш
					ДТЕУ 121 07-12.БР	8
Изм.	Аркуш.	№ докум.	Підпис	Дата		

персональні тренування та інше. Це вимагає від програми автоматизації великої гнучкості й можливості настроювання в процесі експлуатації.

Нарешті, фітнес-клуби працюють із клієнтами протягом, як правило, тривалого часу. Причому, чим більше період дії клубної карти, і вище відсоток продовження, тим послуга вигідніше для клубу. Тому сьогодні, намагаючись закріпити клієнтів, фітнес-клуби пропонують великий вибір послуг – ранкові річні та піврічні карти, абонементи місячні, кварталні. І ці особливості також повинна враховувати система обліку фітнес-клубу.

В загальному випадку можливості автоматизації фітнес-клубу повинні містити у собі:

- облік клієнтів, ідентифікація клієнтів по клубній карті;
- облік відвідувань та обслуговування клієнтів;
- ведення клубних карт, абонементів і разових відвідувань;
- облік часу дії клубної карти, контроль надання оплачених послуг, оперативний моніторинг завантаження клубу;
- можливість роздільного обліку послуг: календарних – якщо клубна карта розрахована на період, кількісних – по кількості відвідувань, або разових послуг для гостей;
- облік оплати послуг, організація різних форм оплати, облік безкоштовних послуг;
- тарифікація по події: час доби, календарна дата, будень/вихідні;
- керування тарифами, перегляд "історії" тарифних планів;
- «заморозки» клубних карт, блокування клубних карт;
- формування й друк оперативних звітів, автоматизація фінансового аналізу діяльності клубу.

Таким чином, підсумовуючи, можна стверджувати, що задача ведення обліку та контролю занять для фітнес-клубу є надзвичайно актуальною і потребує вирішення за допомогою впровадження відповідних програмних продуктів, питання розробки та використання яких розглядатимуться нижче.

						Аркуш
						9
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР	

## 1.2 Аналіз програмних продуктів, що можуть бути застосовані для автоматизації бізнес-процесів фітнес-клубу

Теоретичний аналіз науково-технічної літератури, що докладніше розглядається нижче, свідчить про наявність різноманітних напрямів впровадження спеціалізованих програмних продуктів у галузь фізичної культури, виховання та спорту. Зокрема, у [4] описуються наступні види програмних продуктів, що можуть бути корисними для галузі фізичної культури та/або спорту, за цілями їх застосування:

- програми, що використовуються, як засіб для навчання;
- програмне забезпечення для інформаційно-методичного супроводу навчально-виховного процесу, а також для управління його проведенням та організацією у навчальних закладах, пов'язаних із галуззю спорту та фізичної культури;
- програмні продукти для автоматизації контролю знань, а також процесів комп'ютерного тестування стану здобувачів освіти (мається на увазі фізичний, розумовий, функціональний і психологічний стан);
- програми для автоматизації обробки результатів спортивних змагань та іншої статистичної інформації із галузі спорту, зокрема, відповідних наукових досліджень;
- програмні продукти для організації інтелектуального дозвілля, пов'язаного із спортивною галуззю;
- програмне забезпечення для рекламної, видавничої, підприємницької діяльності у галузі фізкультури та спорту;
- програмні засоби (можливо, у складі програмно-апаратних комплексів) для моніторингу фізичних кондицій та здоров'я осіб, що займаються фізкультурою (особливо з метою оздоровлення та профілактики захворювань) та спортом.

Аналіз досліджень, що відповідають передовому досвіду впровадження сучасних досягнень сфери ІТ у галузь оздоровчого фітнесу, показує, що закордонні автори також приділяють суттєву увагу створенню та інтенсифікації використання спеціалізованого програмного забезпечення для галузі спорту та фізичної культури.

								Аркуш
								10
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР			



Одним із достатньо якісних програмних продуктів, що доцільно застосовувати для підтримки проведення фітнес-занять, є “Yourself Fitness” від Respondesign inc. Ця програма включає комплекс модулів, які зручно використовувати для проведення самостійних занять з понад 500 різноманітними фізичними вправами, які ілюструються докладними схемами та описами. Важливою можливістю є розробка програми занять та відповідного комплексу вправ відповідно до індивідуальних особливостей організму спортсмена. Під час занять фіксуються показники серцево-судинної системи (ритм серця), вік, характеристики фізичної підготовки, при цьому ураховується мета проведення занять. У даному продукті заняття проводяться у вигляді відеоуроку під керівництвом віртуального тренера. На заняттях ведеться планомірний контроль самопочуття спортсмена, що дає змогу безпечно та ефективно планувати фізичні навантаження. Також програма може надавати рекомендації щодо особливостей харчування спортсмену (знову ж таки з урахуванням мети тренувань та особливостей поточного стану користувача програми).

Схожі функції має і програма Exlib (Norsk Video Digitech AS), що є електронним щоденником, в який спортсмен має фіксувати усі особливості свого тренувального процесу. Даний програмний продукт має чотири модулі: щоденник, програми тренувань, звіти та бібліотека. Тут також наявні комплекси фізичних вправ (яких всього є більше 170), причому у вигляді текстових описів, а також наочних керівництв. Довготривалу динаміку процесів тренувань можна переглядати за допомогою генератора звітів [5]. Недоліком даного рішення є його досить складний інтерфейс із багатьма зайвими функціями.

Нажаль, в Україні (і країнах ближнього зарубіжжя, особливо пострадянського простору) використання здобутків сучасних інформаційних технологій у галузі спорту та фізичної культури, не є таким активним та носить переважно фрагментарний характер. У якості прикладу таких вітчизняних розробок можна назвати програму “Fitness Center”, розроблену О. С. Губоревою. Даний продукт складається з наступних складових: “Медичний кабінет”, “Музей”, “Ресторан”, “Спортивна зала”, “Спортивний магазин”. Перевагами даного продукту є:

- можливості як первинного, так і глибокого тестування позитивних змін фізичного стану, викликаних тренуваннями;
- можливість диференційованого підходу до планування оздоровчих та спортивних занять;
- отримання відеозаписів спортивних занять із музичним супроводом, що носять високо мотивуючий характер.

Недоліком даного продукту є надзвичайно сильна комерційна орієнтація, адже у багатьох місцях клієнту пропонуються платні послуги, а вільні функції продукту є досить обмеженими.

Результатом ще одного дослідження, присвяченого оздоровчому фітнесу з використанням засобів фітболтренингу, зокрема питанням його організаційної підтримки, є продукт О. Ю. Лядської [6] – комп'ютерна програма “Fitball training”. Цей продукт присвячений корекції фігури жінок першого зрілого віку, збільшення рівня їх фізичної підготовки та загального рівня здоров'я. Дана програма включає наступні модулі: “Персональні дані”, “РФП”, “Фізичний розвиток”, “Рівень здоров'я”, “Результати”. Всього в програмі наявні 24 варіанти моделей занять, причому вибір конкретної з них залежить від рівня фізичної підготовки осіб, що займаються (всього запроваджено шість моделей занять для кожного рівня підготовки, яких всього виділено чотири: низький, нижче за середній, середній і вище за середній). Програмне забезпечення дозволяє ведення контролю за ефективністю тренувань, що реалізується шляхом аналізу показників фізичного розвитку людини, що тренується. Недоліком даного продукту також можна назвати складний інтерфейс користувача, що може бути завадою для використання широкими масами користувачів.

Ще одним окремим напрямом застосування програмного забезпечення у сфері спорту та оздоровчого фітнесу, зокрема, є запровадження автоматизації менеджменту компаній, що працюють у даній сфері. Використання відповідних засобів автоматизації та методів аналізу на їх основі, дозволяє вирішувати значний комплекс питань, пов'язаних із функціонуванням, аналізом діяльності підприємства, здійсненням планування його подальшої діяльності.

						Аркуш
						12
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР	

В цілому, дослідження перелічених вище і описаних у науково-методичній літературі [7-9] програмних продуктів для галузі оздоровчого фітнесу, дозволяє сформулювати наступні положення:

- використання інформаційних технологій створює умови для диференціації фізичного навантаження через систему бази даних показників фізичного стану тих, хто займається;
- розроблені програмні продукти дають змогу використовувати систему індивідуальних відеоуроків, комплексів фізичних вправ, та методичних рекомендацій для організації самостійних занять;
- сучасні інформаційні технології створюють умови для оптимізації менеджменту діяльності фітнес-клубів.

Беручи до уваги ці три перелічені тези, можна сформулювати наступний перелік перспективних напрямів реалізації інформаційних технологій:

- надання можливості кожній людині здобути знання про особливості функціонування організму людини, методів і засобів побудови оздоровчих занять, основ здорового способу життя в наочній та доступній для сприйняття формі (відеоролики, малюнки, схеми);
- створення умов для забезпечення комп'ютерної й інформаційної грамотності всіх верств населення;
- створення системи інформаційно-консультативних сайтів, для інформаційного зв'язку з провідними спеціалістами й тренерами.

Проведений аналіз та узагальнення даних доступних джерел свідчить про те, що питання підвищення ефективності занять оздоровчим фітнесом стоїть досить гостро та привертає все більшу увагу науковців. Відповідно, надзвичайно актуальним є впровадження програмних продуктів для технічного обслуговування та супроводу процесу занять фітнесом, і одним із таких продуктів є система обліку та контролю проведення занять.

Наступним кроком є визначення необхідності розробки власного програмного продукту чи використання одного із існуючих рішень.

						Аркуш
						13
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР	

Усі програмні продукти для обліку та контролю спортивних занять можна поділити на дві групи: ті, що працюють на пристроях, що належать користувачу, та ті, що працюють на техніці спортивного клубу. Теоретично можливим є використання програмних продуктів, які встановлені на смартфоні користувача і ведуть облік його спортивної діяльності, а потім в момент виникнення необхідності передадуть інформацію співробітникам клубу (в один із моментів приходу користувача в клуб, наприклад, раз на тиждень, або кожного разу). Перевагою ведення обліку занять на пристроях користувача є те, що співробітникам фітнес-клубу не потрібно встановлювати жодного нового програмного забезпечення або купувати якісь пристрої. В той же час такий підхід є не дуже зручним саме для фітнес-клубу, адже при цьому необхідною є процедура синхронізації, тобто переписування інформації із пристрою користувача у книгу обліку клубу, а крім того, така бізнес-модель не захищає організацію від недобросовісних клієнтів, які можуть щось змінити (зокрема, на свою користь) у інформації, яка зберігається в їхньому пристрої. Відповідно, більш доцільно використовувати ПЗ, яке встановлюється на комп'ютерній техніці фітнес-клубу, варіанти якого розглянемо докладніше.

Найпершим продуктом є конфігурація платформи 1С: Фітнес клуб. Вартість повної версії продукту складає 50 000 грн. (до 5 робочих місць). Технічні вимоги в цілому є незначними і продукт може завантажуватися навіть на слабких ПК. Інтерфейс однієї із часто вживаних сторінок продукту показаний на рис. 1.1.

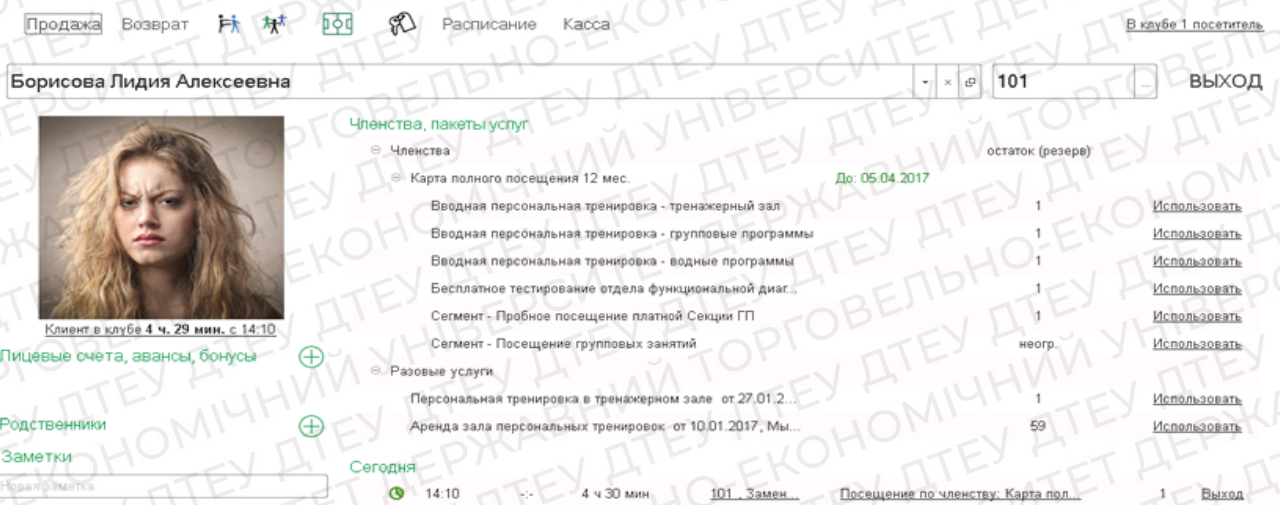


Рис. 1.1. Інтерфейс "1С: Фітнес клуб"

						Аркуш
						14
Изм.	Аркуш.	№ докум.	Подпис	Дата		

Платформа "1С: Фітнес клуб" призначена для ведення автоматизованого управлінського й оперативного обліку. У даній програмі відсутній функціонал для ведення бухгалтерського й податкового обліку. Програма вирішує широке коло різноманітних завдань, пов'язаних з обслуговуванням клієнтів, обліком фінансів і запасів на складі, керуванням персоналом клубу, збором аналітичних даних і формуванням звітів по роботі спорт-клубу.

Функціональні можливості програми:

1. Робота із клієнтами:

- ведення бази клієнтів;
- ведення карти клієнта;
- керування абонементом;
- облік відвідувань клієнтів;
- попередній запис на групові заняття;
- використання пластикових карт;
- облік дисконтних карт;
- облік послуг оренди;
- робота з корпоративними клієнтами.

2. Облік фінансів:

- облік каси;
- облік банку;
- ведення особових рахунків клієнтів;
- облік витрат фітнес клубу.

3. Керування персоналом:

- планування графіка роботи співробітників;
- планування графіка чергувань співробітників;
- облік фактично відпрацьованого часу роботи персоналу;
- розрахунки зарплати;
- аналіз ефективності роботи персоналу.

4. Облік запасів на складі:

								Аркуш
								15
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР			

- облік вступів товарів на склад;
- облік запасів по партіях;
- списання матеріалів по нормах;
- контроль критичних залишків;
- проведення інвентаризації.

5. Проведення маркетингових заходів:

- проведення різних маркетингових опитувань клієнтів з метою поліпшення якості роботи фітнес клубу й залучення нових клієнтів;
- аналіз ефективності джерел залучення клієнтів.

6. Уведення аналітики про роботу фітнес клубу:

- звіти по клієнтах;
- аналітика фінансових результатів;
- звіти по складу;
- аналіз роботи співробітників;
- контроль роботи клубу через Інтернет.

Отже, особливостями даного продукту є дуже широкий функціонал, можливість формування звітів по різних зрізах, порівняно зручний інтерфейс.

Недоліком для використання у середньому за масштабами фітнес-клубі є надлишковий функціонал, а головне, висока вартість самого програмного забезпечення.

Отже, незважаючи на те, що дане програмне забезпечення має ряд переваг, все ж його використання на даному підприємстві недоцільно, в першу чергу через високу вартість.

Іншим аналогічним продуктом є «Проффіт» від розробника Mi-soft. Технічні вимоги програми також не є високими і програма може працювати на слабких ПК. Інтерфейс продукту показано на рис. 1.2.

									Аркуш
									16
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР				

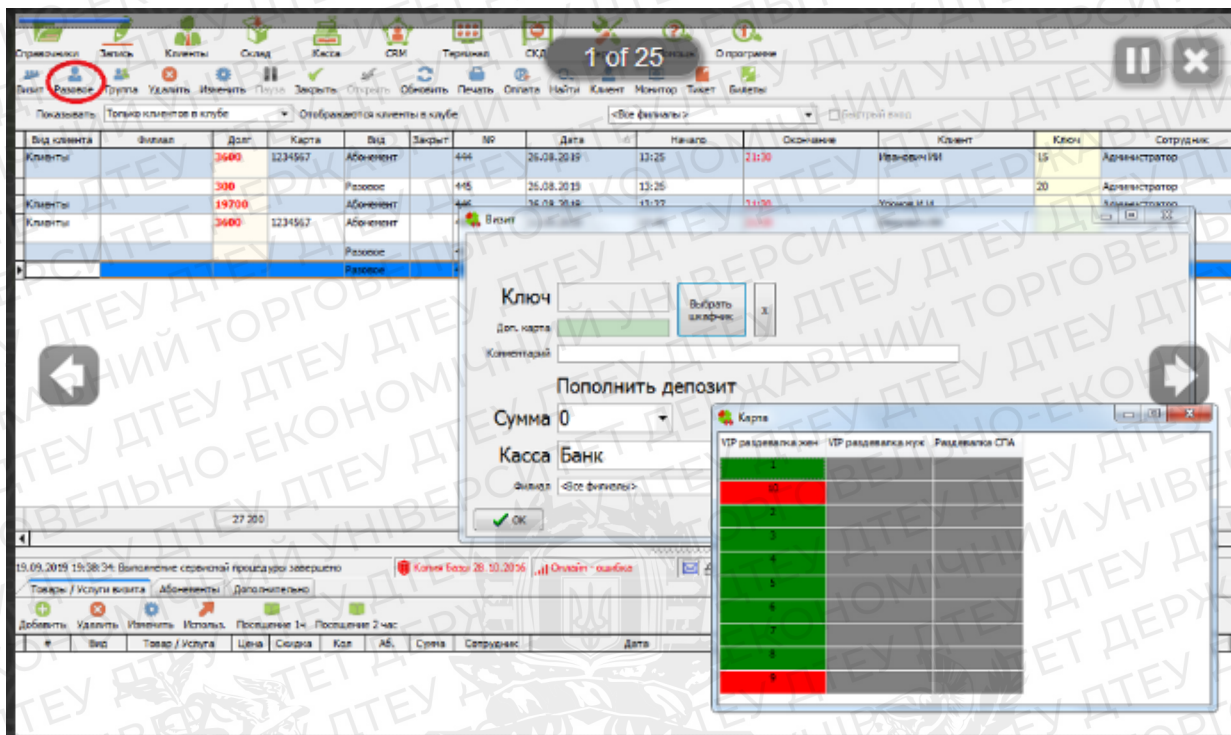


Рис. 1.2. Інтерфейс програми "Профит: Фітнес клуб": екран для введення інформації про разовий візит

Вартість повної версії «Профит: Фітнес клуб» складає 65 000 грн. + 2000 грн. за кожне додаткове місце, що також є досить дорогим рішенням. Програма «Профит: Фітнес клуб» призначена для обслуговування клієнтів, проведення маркетингових заходів, аналізу статистики відвідування клубу, а також допоможе створити "прозорість" у роботі персоналу.

Функціональні можливості продукту є наступними.

1. Адміністратор клубу має можливість здійснювати:

- відкриття/закриття візитів клієнтів по абонементу;
- продаж разових відвідувань;
- продаж і продовження абонементів;
- приймання платежів;
- видачу чеків;
- формування звіту по касі за зміну;
- формування звітності по продажах товарів/послуг і абонементів за зміну.

2. Бармен клубу одержує можливість здійснювати:

Изм.	Аркуш.	№ докум.	Підпис	Дата

- облік продажів клієнтам товарів бару;
- видачу чеків за реалізований товар;
- ведення обліку залишків;
- складання звітності за зміну.

3. Менеджер клубу одержує можливість здійснювати:

- ведення номенклатури послуг/товарів;
- регулювання цін;
- реєстрацію нових клієнтів у системі;
- оформлення й продаж послуг/абонементів клієнтам;
- формування звітності по роботі персоналу;
- "заморозку/разморозка" терміну дії абонементу.

4. Керуючий клубом одержує можливість здійснювати:

- формування оперативної звітності за обсягом виторгу в касах;
- формування звітів по продажах товарів і послуг;
- аналіз доходів.

5. Власник бізнесу одержує можливість здійснювати:

- одержувати оперативну інформацію про роботу клубу;
- облік і контроль роботи;
- оцінку ефективності клубу.

Перевагами продукту є можливість використання багатокористувацького режиму, одержання різномітних форм звітів, зручний інтерфейс при веденні обліку. Недоліками є надлишковий функціонал, платний процес впровадження, додаткова плата за кожне місце і взагалі досить високі ціни на продукцію. Як висновок, можна сказати, що це програмне забезпечення має ряд незаперечних переваг, однак використання на даному підприємстві недоцільно через фінансові питання.

Ще одним аналогічним продуктом є програма fitness365, від «Девпарк: Фітнес». Вартість неповної версії "fitness365" складає 7500 грн. на місяць, а додаткові модулі слід оплачувати окремо.

						ДТЕУ 121 07-12.БР	Аркуш
Изм.	Аркуш.	№ докум.	Підпис	Дата			18



Технічні вимоги є аналогічними до попередніх продуктів, тобто програму можна використовувати на простих ПК. Загальний вигляд інтерфейсу програми представлено на рис. 1.3.

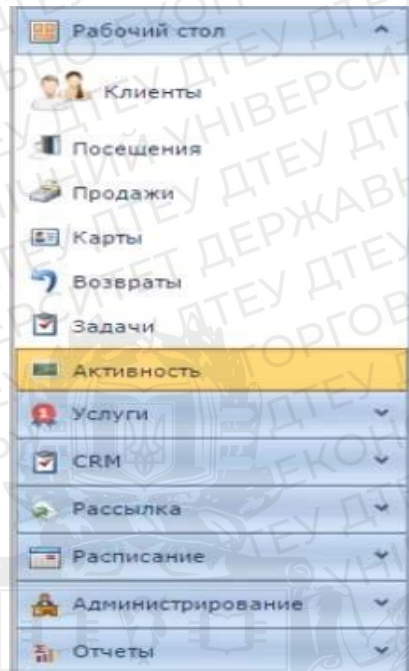


Рис. 1.3. Інтерфейс програми "fitness365"

Програма призначена для керування клієнтською базою фітнес-клубу, обліку відвідувань, клієнтів і продажів, планування розкладу, керування роботою персоналу [13].

Функціональні можливості продукту:

1. Робота із клієнтами:

- реєстрація клієнта;
- ведення бази клієнтів;
- реєстрація клубної карти;
- облік послуг залу;
- облік продажів абонементів;
- заморозку абонементів;
- запис на персональні тренування;
- друк договору із клієнтом.

2. Налаштування послуг, абонементів і карт:

						Аркуш
						19
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР	

- створення послуги;
- створення абонементу;
- створення клубної карти.

3. Робота зі складом:

- облік продажів товарів і послуг;
- складський облік;
- переоцінка вартості;
- друк накладних.

4. Робота з касою:

- виторг за зміну;
- закриття зміни;
- внесення готівки;
- вилучення готівки;
- повернення товару або послуги.

5. Додаткові функції:

- автоматичні нагадування клієнтам і тренерам зі СМС і e-mail;
- планування групових програм;
- звіти по продажах, відвідуваннях, динаміці, підсумках.

Перевагами програми є зручний інтерфейс, можливість створення усіляких звітів, обширний функціонал, а також механізм нагадування клієнтам і тренерам з використанням СМС та e-mail. Недоліки аналогічні до попередніх продуктів-аналогів та полягають у високій вартості та зайвому функціоналі.

Провівши аналіз існуючих програмних розробок, можна зробити висновок про те, що на ринку програмного забезпечення на сьогоднішній день існує велика кількість розробок, які спрямовані на розв'язок завдань автоматизації різних бізнес-процесів кампаній, що працюють у сфері надання фітнес-послуг. Однак їх застосування для середнього за масштабами спортивного клубу із різних причин є недоцільним, тому що при виборі програмного забезпечення необхідно враховувати як функціонал, так і його вартість. У зведеному вигляді характеристики програмних продуктів-аналогів представлені в табл. 1.1.

						Аркуш
						20
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР	

Табл. 1.1 Аналіз програмних продуктів-аналогів

	1С: Фітнес клуб	Проффіт. Фітнес клуб	fitness365
Вартість	90 000 грн. (-1)	99 000 грн. + 2000 грн. за кожне додаткове місце(-1)	7500 грн. в місяць + додаткові модулі (-1)
Системні вимоги	Мінімальні (+1)	Мінімальні (+1)	Мінімальні (+1)
Функціонал	Занадто великий (-1)	Занадто великий (-1)	Занадто великий (-1)
Оплата впровадження	Безкоштовно (+1)	Платне (-1)	Платне (-1)
Інтерфейс	Зручний (+1)	Зручний (+1)	Зручний (+1)
Інтеграція	Є (+1)	Є (+1)	Немає (-1)
Загальна оцінка	+2	0	-2

Як видно з таблиці, при прийнятих оцінках кращим виглядає програмне забезпечення від 1С, однак зважаючи на такий важливий показник, як вартість, більш доцільною є розробка власного програмного продукту, який будучи дешевшим, не матиме зайвого функціоналу.

### 1.3 Постановка завдання на розробку

Таким чином, можна сформулювати наступне технічне завдання на розробку програмного забезпечення у даній роботі (ТЗ на ПЗ).

Назва продукту: FitAccounting.

Тип продукту: інформаційна система фітнес-клубу.

Архітектура продукту: настільний додаток.

Призначення продукту: підвищення ефективності діяльності підприємства типу «фітнес клуб».

Цільова програмно-апаратна платформа: настільна операційна система Windows версії не нижче 8.1, тобто такої, що офіційно підтримується розробником – компанією Microsoft.

						Аркуш
						21
Изм.	Аркуш.	№ док.ум.	Підпис	Дата	ДТЕУ 121 07-12.БР	

Вимоги до інтерфейсу користувача: графічний, максимально простий, інтуїтивний.

Мінімальні системні вимоги (співпадають з мінімальними системними вимогами настільної ОС Windows):

- процесор 1 ГГц;
- оперативна пам'ять 1 Гб;
- 100 Мб на диску;
- екран 800x600 пікселів.

Функціональність:

- авторизація користувачів із розділенням функціоналу на два варіанти: клієнтську та адміністраторську;
- можливість запису для клієнта до потрібного тренера;
- можливість перегляду для адміністратора інформації про обраного тренера та автоматичного нарахування йому заробітної платні (залежно від кількості годин, напрацьованих за звітний період);
- перегляд списку клієнтів, з можливістю їх розділення на тих, що мають абонемент, та тих, що віддають перевагу разовим відвідуванням;
- можливість пошуку клієнта за поширеними полями (прізвище, ім'я, телефон, пошта);
- можливість детального перегляду інформації про обраного клієнта;
- можливість зміни інформації про обраного клієнта.

Програмний продукт повинен мати зручний сучасний інтерфейс, стандартний для настільних додатків, що забезпечить можливість використання програми звичайними користувачами, що не потребують спеціальної підготовки.

Розробка програмного продукту ведеться в рамках написання дипломної роботи і буде захищатися під час представлення результатів її створення.

Керуючись цими вимогами можна переходити до процесу проектування кінцевого програмного продукту, який включає наступні етапи:

- вибір засобів розробки;
- проведення програмної реалізації;

Изм.	Аркуш.	№ докум.	Підпис	Дата

- тестування та документування.

#### 1.4 Висновки до розділу 1

Таким чином, розглянуто особливості предметної галузі, а саме – бізнес-процесів, які протікають у фітнес-клубі. Також проаналізовано програмні продукти аналоги та встановлено, що існує доцільність створення власного програмного забезпечення. Також у розділі сформовано технічне завдання на розробку програмної компоненти відповідної інформаційної системи.



									Аркуш
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР				23

## РОЗДІЛ 2

### ОБГРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ ПРИ СТВОРЕННІ ПРОГРАМНОЇ КОМПОНЕНТИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ФІТНЕС-КЛУБУ

#### 2.1 Моделювання предметної галузі, яка розглядається у даній роботі

Відповідно до завдання на розробку, яке наведено у підрозділі 1.3, програмне забезпечення розділяє свою функціональність на дві частини: для клієнтів фітнес-клубу та для адміністраторів стійки фітнес-клубу, що розміщується при його вході і де знаходиться адміністратор. В обох випадках програма має забезпечувати належне виконання комплексу стандартних операцій, притаманних будь-якій роботі із базами даних:

- створення (Create) нового об'єкта у базі даних;
- читання (Read) інформації про певний об'єкт, що вже знаходиться у базі даних;
- зміна (Update) інформації про певний об'єкт, який уже міститься у БД;
- видалення (Delete) об'єктів із БД.

За першими літерами англійських назв відповідних операцій утворюється аббревіатура CRUD, яка і символізує четвірку вказаних у попередньому переліку дій. Також слід звернути увагу, що останні три операції перед тим, як проводити якісь дії над записом, що міститься у базі даних (видавати його на екран, змінювати його або видаляти), вимагають виконання операції пошуку, тобто даний запис треба знайти. Працюючи у термінах реляційних баз даних, слова «знайти якийсь запис» означають,

					<i>ДТЕУ 121 07-12.БР</i>			
					<i>Програмна компонента інформаційної системи фітнес клубу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		P2	24	59
Зав. каф.		Криворучко О.В.		04.03.22	<i>Обґрунтування проектних рішень при створенні програмної компоненти інформаційної системи фітнес-клубу</i>	Факультет інформаційних технологій, 4 курс, 7 група		
Керівник		Рзасва С. Л.		04.03.22				
Гарант		Рзасва С. Л.		04.03.22				
Розроб.		Потапов Д. Р.		04.03.22				

що необхідно встановити значення первинного ключа (ідентифікатора поля), яким характеризується шуканий рядок у таблиці БД.

Відповідно до означеного, можна сформуванати наступну діаграму прецедентів (варіантів використання або Use Case) – рис. 2.1, що відповідає можливостям по роботі з системою і для клієнта фітнес-клубу і для його адміністратора.

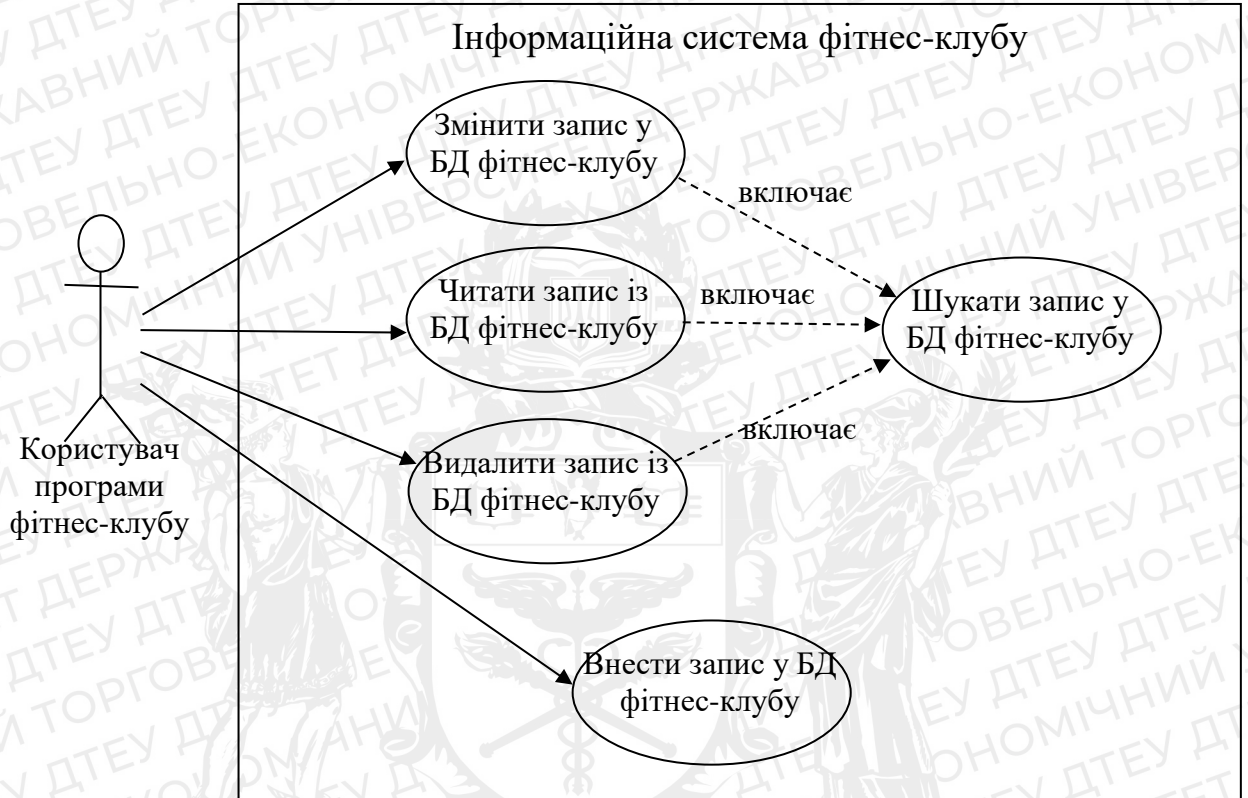


Рис. 2.1. Модель Use Case (прецедентів) для програмного забезпечення, що розробляється

Як відомо, для кожного варіанту використання необхідно створити діаграму активності, що деталізує складові дії та порядок їх виконання при роботі по одному із сценаріїв, показаному на діаграмі прецедентів. Відповідно, розроблено наступні діаграми:

- для прецеденту «Читати запис із БД фітнес-клубу» модель активності наведена на рис. 2.2;
- для прецеденту «Шукати запис у БД фітнес-клубу» модель активності - на рис. 2.3;

- для прецеденту «Видалити запис із БД фітнес-клубу» модель активності - на рис. 2.4;
- для прецеденту «Внести запис у БД фітнес-клубу» модель активності - на рис. 2.5;
- для прецеденту «Змінити запис у БД фітнес-клубу» модель активності - на рис. 2.6.

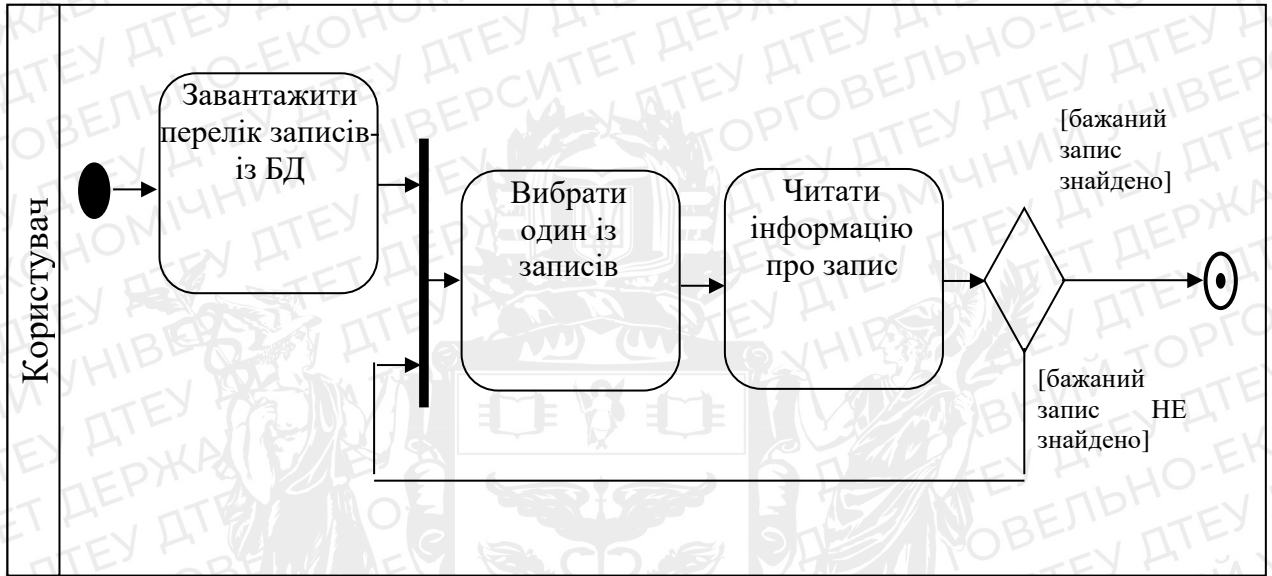


Рис. 2.2. Модель діяльності для прецеденту «Читати запис із БД фітнес-клубу»

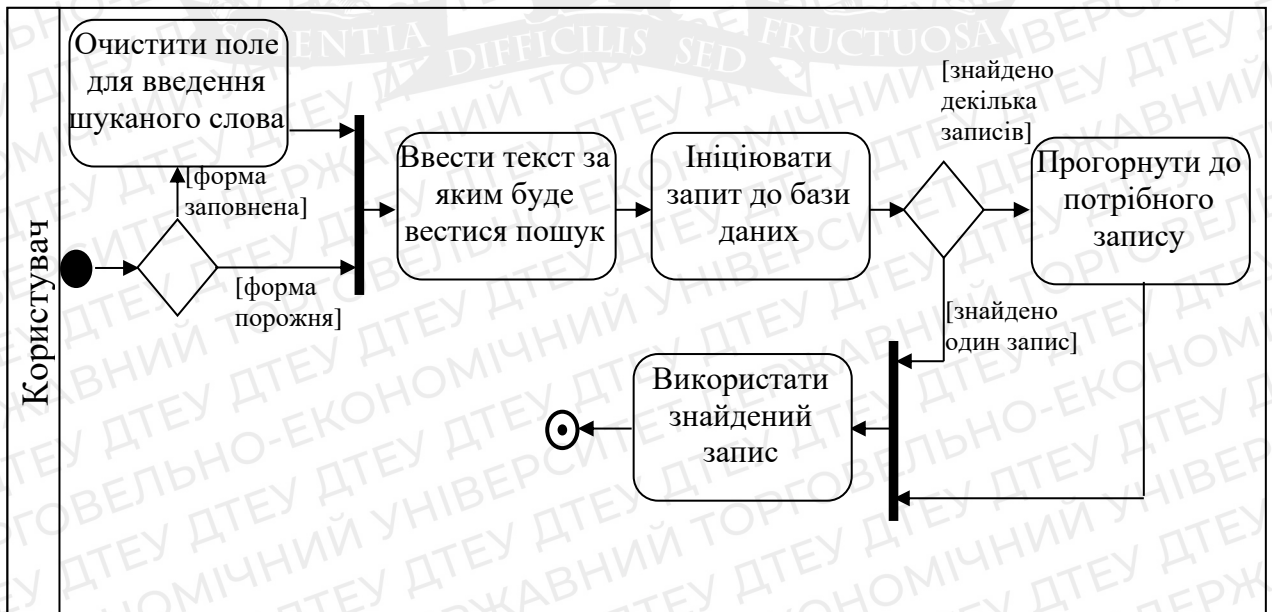


Рис. 2.3. Модель діяльності для прецеденту «Шукати запис у БД фітнес-клубу»



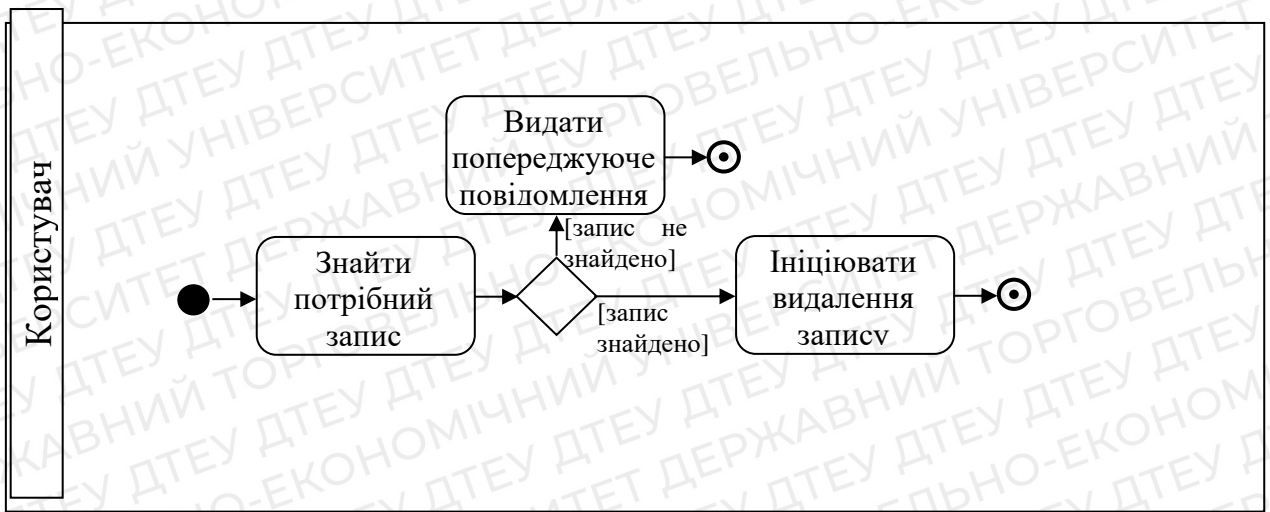


Рис. 2.4 Модель діяльності для прецеденту «Видалити запис із бази даних фітнес-клубу»

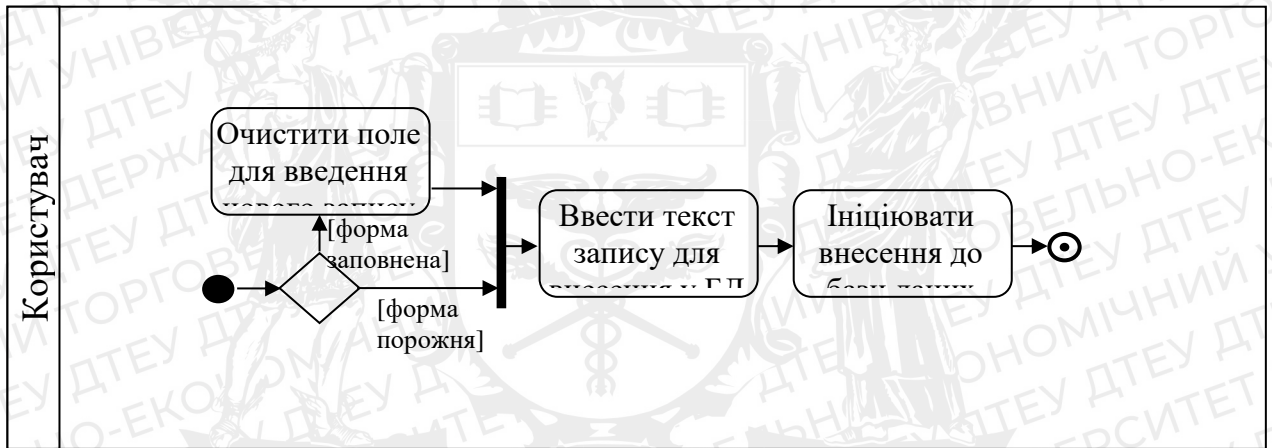


Рис. 2.5 Модель діяльності для прецеденту «Внести запис у БД фітнес-клубу»

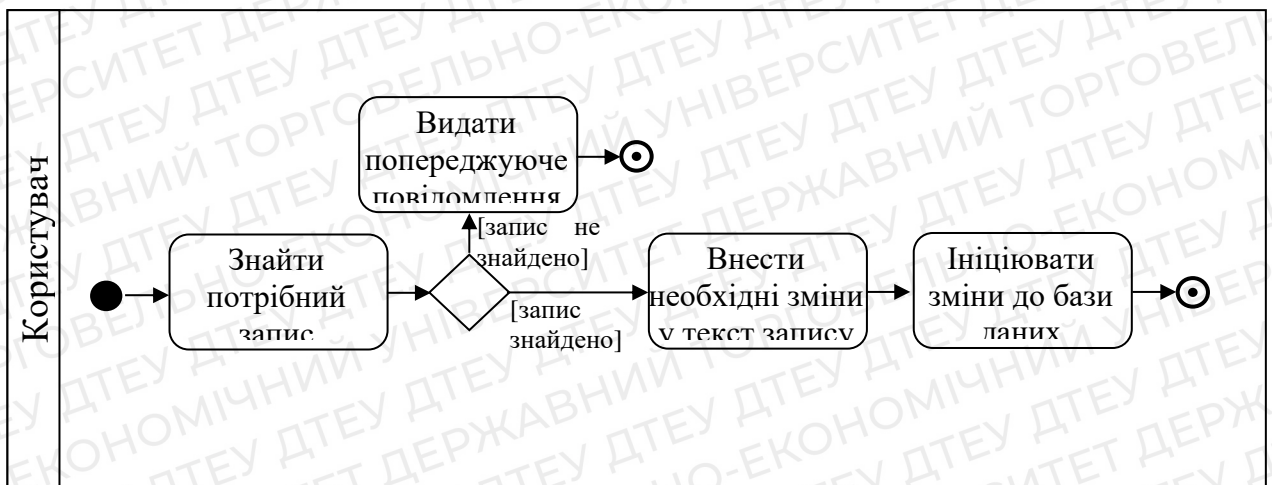


Рис. 2.6 Модель діяльності для прецеденту «Змінити запис у БД фітнес-клубу»

Використовуючи розроблені моделі, можна проводити програмну реалізацію, опис якої приведено нижче.

## 2.2 Вибір технологій і засобів розробки

### 2.2.1 Обґрунтування вибору архітектури програмної компоненти інформаційної системи фітнес-клубу

Програмний продукт для обліку та контролю спортивних занять може бути реалізований різними шляхами, зокрема це можливо у вигляді:

а) програмного продукту для персонального комп'ютера для ОС Windows (вибір обумовлений тим, що саме дана ОС є найбільш поширеною для ПК). Цей варіант є найзручнішим, так як дозволяє отримувати доступ до файлової системи комп'ютера, зокрема баз даних та створювати найвищий рівень інтеграції всього додатку;

б) веб-додатку, що забезпечує максимальну гнучкість, кросплатформність, через те, що доступ до нього є можливим із будь-якого клієнтського пристрою, що має підключення до Інтернет. Такий варіант організації програми може застосовуватися і з настільного комп'ютера (ноутбуку), так і зі смартфона, телевізора, чи іншого пристрою із доступом до Інтернет. Втім реалізація системи у вигляді веб-додатку не є доцільною, оскільки при такому способі організації програмного забезпечення більш низьким є загальний рівень безпеки, складність доступу до системних функцій комп'ютера, зайва функціональність у вигляді виходу до мережі Інтернет. Головне: якщо в спортзалі колись буде відсутнє підключення в мережі Інтернет, то весь цей час система обліку буде у повністю неробочому стані. Недоцільно піддавати проектовану систему реальним небезпекам із самого початку (вже на стадії проектування), якщо можна легко цього позбутися (як у попередньому пункті а). Із описаних причин варіант веб-додатку розглядати не будемо;

в) вбудованого програмно-апаратного рішення, такого, як спеціальний пристрій для обліку спортивних занять. Цей підхід потребує значних невиправданих фінансових затрат на проектування та реалізацію даного пристрою, що не є прийнятним для спортивної організації, яка не має особливих вимог до захисту

						ДТЕУ 121 07-12.БР	Аркуш
Изм.	Аркуш.	№ докум.	Підпис	Дата			28

інформації, високої обчислювальної продуктивності проектованого рішення і т.п. Використовувати якісь спеціалізовані пристрої для даних цілей недоцільно, оскільки потрібні функції цілком задовільно можуть забезпечуватися звичайним персональним комп'ютером (ноутбуком);

г) мобільного додатку, що має ті ж самі недоліки, що й веб-додаток (низький рівень захисту інформації, складність використання системних функцій обчислювального пристрою та комп'ютера, на якому встановлена база даних, тощо). Крім того, мобільний додаток зазвичай має малу корисну площу екрану, що є негативним фактором для системи, яка повинна відображувати облікові дані. Також мобільні додатки із очевидних причин мають таку особливість, що встановлюються у переносні пристрої (смартфони), і якщо в інших випадках ця особливість є корисною, то в даному випадку це навпаки недолік. Справа в тому, що доступ до системи обліку повинен забезпечуватися тільки зі стійки рецепції спортивного залу, але, якщо випадково смартфон хтось із співробітників покладе собі до кишені, можуть виникнути небажані заминки із пошуком пристрою, що є цілком неприйнятним для проектованої системи і легко унеможлиблюється шляхом розробки програми для настільного ПК.

Таким чином, використовувати будемо архітектурне рішення у вигляді настільного додатку для операційної системи Windows.

Очевидно, що будь-яка система обліку повинна використовувати базу даних, куди будуть зберігатися відомості про усі візити клієнтів до фітнес-клубу. Теоретично можливо реалізувати збереження даних у власному форматі і писати усі процедури для доступу до них, однак такий підхід не є доцільним оскільки максимально ефективні рішення по збереженню даних уже реалізовані у сучасних системах управління базами даних (СУБД). Таким чином, для того, щоби не винаходити велосипед, слід зберігати інформацію, що обробляється системою, у певній СУБД (вибір конкретної СУБД буде здійснюватися нижче у даному розділі). З архітектурної точки зору це означає, що програмний продукт розділяється на дві частини:

- клієнтську частину, яка збирає інформацію від працівника фітнес-клубу та надає йому необхідні звіти/відомості;

Изм.	Аркуш.	№ докум.	Підпис	Дата

- серверну частину, яка у даному випадку являтиме собою СУБД із реалізованою у ній базою даних, яка розроблена спеціально для обліку відомостей фітнес-клубу.

Таким чином, проєктований програмний продукт для фітнес-клубу зводитиметься за технологією «клієнт-сервер».

### **2.2.2 Вибір технології розробки програмної компоненти інформаційної системи фітнес-клубу**

Першочерговим питанням, яке постає перед розробниками будь-якого програмного забезпечення, є вибір моделі або технології, парадигми його розробки. Такими, що широко використовуються на сьогоднішній день у виробничій практиці, є технології структурного (процедурного) та об'єктно-орієнтованого програмування. Кожна з них має свої особливості, переваги і недоліки, які розглянемо докладніше.

Структурне, або як його ще називають практикуючі програмісти, процедурне програмування засноване на використанні окремих структурних блоків, в першу чергу, підпрограм (процедур і функцій). Структурне програмування має на увазі побудову програми відповідно до трьох основних принципів: слідування, розгалуження, повторення.

Слідування означає, що оператори і блоки програмного коду слідують і виконуються один за іншим. Розгалуження реалізується різними умовними операторами типу if (а також, оператор «?», switch/case), і дозволяє вибрати один з декількох подальших варіантів виконання програми. Повторення зазвичай відносять на рахунок циклів (багаторазових повторів однієї і тієї ж ділянки коду), хоча цей же принцип можна віднести і до підпрограм.

Взагалі ж, структурне програмування є більш старшою методикою програмування, на зміну якій поступово прийшло об'єктно-орієнтоване програмування (деякі сучасні мови програмування загального призначення навіть не дозволяють створити структурну програму, тільки об'єктно-орієнтовану – як, наприклад, Visual C# чи Java). Проте, при створенні невеликих програм (наприклад, до 10000 рядків коду і без передбачуваного розширення) застосування цієї методики

								Аркуш
								30
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР			

програмування все ще виправдано і такий код краще сприймається, ніж його об'єктно-орієнтований варіант.

Суть же методології об'єктно-орієнтованого програмування полягає в тому, що система розглядається, як сукупність окремих активних сутностей - об'єктів, які мають набір якихось своїх внутрішніх параметрів - властивостей, а також можуть взаємодіяти між собою за допомогою деяких дій - викликів методів (або більш непрямим чином - шляхом надсилання повідомлень, оброблюваних методами об'єктів; для цього необхідна присутність активної сутності, яка роздає повідомлення адресатам, як, наприклад, менеджер вікон в ОС Windows, який надсилає вікнам програм адресовані до них повідомлення).

Якщо говорити про програмний код, то для того, щоб оперувати об'єктом, його спочатку потрібно створити. Об'єкти створюються як змінні, у яких типом виступає клас об'єкта. Клас являє собою опис, які властивості можуть мати об'єкти такого типу (тобто яку інформацію вони можуть зберігати), і які у них є методи (тобто які дії вони можуть виконувати). Об'єктом же є набір значень, чому саме рівні властивості даного об'єкта (а свої методи кожен об'єкт отримує від свого класу, тобто методи однакові у всіх об'єктів, що належать даному класу).

Реалізуючи всі сутності, необхідні, згідно з алгоритмом, для роботи програми, у вигляді класів і об'єктів, ми спрощуємо її розуміння для самих себе. Саме тому ОО-підхід рекомендується до застосування для великих проектів (більше десятків тисяч рядків коду), коли утримувати «в голові» всю систему цілком стає важко. Можна сказати, що розбиття програми на об'єкти і проектування їх класів наближає розуміння предметної області до звичного людського образу мислення (в разі «великих» проектів). Людина мислить класами, об'єктами і зв'язками між ними.

Відзначимо, що часто, крім розглянутих міркувань, також на вибір методики програмування впливають інші чинники, наприклад, можливість майбутнього розширення функціональності, створення якомога більш зрозумілого коду (для роботи над проектом цілої команди, а не одного програміста), або просто побажання замовника застосувати найбільш сучасний підхід до програмування та ін.

									Аркуш
									31
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР				

Крім розбиття (декомпозиції) всієї предметної області на об'єкти (класи) і співвідношення між ними, також ОО-підхід має на увазі дотримання трьох основних його принципів: інкапсуляція, наслідування, поліморфізм.

Під інкапсуляцією мається на увазі об'єднання даних (значення властивостей класу у деякого конкретного об'єкта) та засобів їх обробки (методи класу). Це знову ж таки зручно психологічно, так як дозволяє реалізовувати окремі завершені сутності - класи, які самі обробляють свої дані. Звернення до об'єктів цих класів відбувається за допомогою методів, що утворюють інтерфейс класу.

Наслідування є дуже корисною можливістю, тому що дозволяє значно скорочувати обсяги повторюваного коду (до чого потрібно завжди прагнути при розробці будь-якого програмного забезпечення). Згідно з цим принципом виділяється клас, який має спільний набір властивостей і методів для декількох більш розширених класів. Цей клас оголошується батьком, базовим класом для декількох похідних від нього (нащадків, спадкоємців). Всі класи-нащадки успадковують від базового всі його властивості та методи, але до цього ще мають свої власні оригінальні властивості і / або методи.

При наслідуванні іноді методи батьківського і похідного класу мають однакове призначення, але реалізуються по-різному. Такі методи називаються перевантаженими. При цьому ще раз підкреслимо, що призначення методу в обох випадках одне і те саме.

Поліморфізм є можливістю деякої функції (яка належить якомусь класу, тобто функції-методу, або окремих, розміщених поза усіма класами) приймати об'єкти як батьківського, так і похідних класів, і вміти викликати перевантажені методи саме того класу, об'єкт якого був переданий у функцію. Слід сказати, що це досить специфічна можливість і в загальному багато програмістів використовують ОО-підхід і без звернення до поліморфізму.

Важливими поняттями в ООП також є: статичні члени класу, абстрактні методи і класи, дружба функцій і класів, і т.д.

Грунтуючись на перерахованих особливостях двох найпоширеніших методик програмування, вибираємо об'єктно-орієнтований підхід як більш сучасний, гнучкий

						ДТЕУ 121 07-12.БР	Аркуш
Изм.	Аркуш.	№ докум.	Підпис	Дата			32

та прогресивний, а також такий, що відповідає середнім масштабам проектного ПЗ.

### 2.2.3 Вибір мови програмування та короткий опис її особливостей

При розробці програмного забезпечення крім обрання архітектури та технологій програмування слід встановити, які із численних наявних на ринку чи у вільному доступі засобів розробки надають достатні можливості для реалізації алгоритму у програмних кодах. Слід відмітити, що різні засоби розробки, чи навіть цілі мови програмування, з самого початку свого створення мали різне призначення і деякі з них є спеціалізованими, тобто пристосованими до виконання лише певних задач окремих прикладних галузей, а деякі – засобами загального призначення. Серед останніх можна виділити такі, що містять окремі уже готові, більше чи менше пророблені, компоненти для вирішення заданої прикладної задачі, а також такі, в яких таких компонентів немає.

Також важливим фактором при виборі засобів розробки виступає його популярність (зокрема, мови програмування), тобто поширеність фактів застосування цього засобу у професійній діяльності. Не слід недооцінювати цей момент, оскільки в процесі розробки нового програмного забезпечення більш-менш складної функціональності, у будь-якого програміста виникатимуть питання щодо її реалізації, причому якраз за допомогою обраного засобу розробки. У даний час найпершим і найефективнішим місцем, де можна отримати інформацію про питання розробки програмного забезпечення є глобальна мережа Інтернет (на відміну від часів 90-х років, коли найкращим способом вирішення проблеми пошуку відповідей на свої запитання було живе спілкування з експертом, програмістом-старшим товаришем). Сьогоднішні програмісти порівняно легко і швидко знаходять відповіді на практично будь-які питання, пов'язані із професійною діяльністю, в Інтернет (середній час пошуку складає порядку кількох хвилин, максимум – до 10). Така, практично ідеальна, ситуація реалізується тільки за умови поточної популярності мови програмування (або середовища розробки, бібліотеки, фреймворку, і т.п.), що використовується. Наприклад, існують десятки (якщо не сотні) тисяч ресурсів

						Аркуш
					ДТЕУ 121 07-12.БР	33
Изм.	Аркуш.	№ докум.	Підпис	Дата		

присвячених мові програмування РНР, причому деякі з них мають як докладну теоретичну базу у вигляді пакетів документації, інформаційних статей, відеолекцій, тощо, так і засоби для «живого» спілкування програмістів між собою від форумів до онлайн-чатів. В той же час, зважаючи на те, що всього у світі існує вже 500-2000 (за даними різних джерел) більш-менш активно використовуваних мов програмування, очевидним є те, що деякі з них використовуються надзвичайно рідко і знайти необхідну інформацію про них у мережі Інтернет іноді не уявляється можливим.

Відмітимо, що на сьогоднішній день достатньо популярними є такі мови програмування загального призначення як: C++, Java, C#, Delphi.

Перший варіант, мова C++, є надзвичайно потужною, яка має стабільну популярність вище середнього рівня, але і досить складною для сприйняття; фактично, це інструмент для професіоналів найвищого рівня. В той же час серед широких мас на сьогоднішній день більш популярними є мови Java та C#. Мова Delphi (або точніше – середовище розробки, оскільки у ньому використовується мова програмування Object Pascal), не зважаючи на свої зручність та простоту (а, можливо, навпаки, через них), стабільно не має широкої популярності серед професійних розробників програмного забезпечення, а використовується в основному для малих проектів та навчальних цілей.

Таким чином, при реальному процесі вибору мови програмування для професійної розробки програмістом середнього рівня на сьогоднішній день слід розглядати два основних варіанти – Java та C#. Дуже багато полеміки було присвячено порівнянню цих двох мов, але беззаперечним фактом є значна перевага продукту від Microsoft над конкурентом у частині зведення користувацького інтерфейсу. У цьому питанні середовище Visual Studio надає практично той же рівень зручності, що й Delphi, але використовує популярні мови, а не Паскаль, до якого традиційно укорінилося ставлення як до навчальної мови програмування для новачків. Отже, зважаючи на необхідність створення програмного засобу для операційної системи Windows, кінцево обираємо мову програмування C#, що і відповідає завданню на розробку в рамках даної роботи. Коротко розглянемо особливості цієї мови програмування.



## 2.2.4 Обґрунтування вибору системи управління базами даних

Важливою складовою програмного забезпечення, що розробляється, є база даних із обліковими даними клієнтів фітнес-клубу, включаючи відомості про їхні візити та діяльність на території закладу. Для цього на комп'ютері, де запускається розроблюване ПЗ, має бути наявним доступ до серверу баз даних із розгорнутою відповідною базою даних.

Зважаючи на відносну простоту задачі, використання потужних корпоративних продуктів типу Oracle Database, та й PostgreSQL є невиправданим. В той же час найпростіші засоби збереження інформації у БД типу Paradox dBase, Microsoft Access, miniSQL, SQLite або є застарілими, або мають ліцензійні обмеження, або сильно обмежені функціонально. В той же час існує надзвичайно популярна на сьогоднішній день система управління базами даних MySQL, що вже стала стандартом де-факто при зведенні веб-проектів малого та середнього розміру (а може використовуватися і для великих проектів, що обмежується лише психологічними факторами). Не зважаючи на те, що у даному проекті весь процес має виконуватися локально, це не заважає взяти до використання той же сервер баз даних, але підключатися до нього не за допомогою веб-серверу, а безпосередньо, через його програмний інтерфейс (API), чи іншими методами.

Вибір СУБД MySQL обумовлений також наступними її перевагами у порівнянні з конкурентними продуктами:

- це порівняно легкий продукт, що займає малу кількість оперативної пам'яті сервера (а також, що менш критично, і малі обсяги постійної пам'яті жорсткого диску сервера);
- це порівняно функціональний продукт, що включає практично усі основні функції по обробці даних та створенню ефективних БД;
- це програмне забезпечення є вільним для використання (на відміну від деяких корпоративних продуктів, ліцензії на які можуть коштувати тисячі умовних одиниць, тобто сотні тисяч грн.);

- існує значна кількість уже розробленого програмного забезпечення (фрагментів, модулів, і т.п.) для роботи з цією СУБД;

- існує чимало Інтернет-ресурсів, на яких здійснюється технічна підтримка програмістів, що здійснюють розробку для MySQL (тобто висвітлюються особливості роботи із нею).

Зважаючи на усі ці особливості, СУБД MySQL береться за основу для збереження даних, які використовуватимуться у інформаційній системі фітнес-клубу, що розробляється.

### 2.3 Проектування бази даних програмної компоненти інформаційної системи фітнес-клубу

Наступним кроком після обґрунтування вибору СУБД є моделювання бази даних та визначення алгоритмів її взаємодії із прикладним програмним забезпеченням, тобто програмою, що розробляється. Для цього можна використовувати набір функцій, що надаються самою СУБД, тобто її API, однак такий підхід не є дальновидним кроком, оскільки, якщо в подальшому виникне потреба у переході на іншу СУБД, то для цього треба буде переписувати значну кількість рядків коду усієї програми. Іншим шляхом, що робить програмний продукт більш гнучким, менш залежним від конкретних застосованих засобів управління базами даних, є використання універсального механізму доступу до баз даних - ODBC (Open DataBase Connectivity). Це фактично драйвер для уніфікованого доступу до БД будь-якого виробника, причому особливості самої БД прописуються лише один раз – у рядку підключення до БД. Для цього у програмному коді слід вставити наступні рядки:

```
string MyConnectionString = "Driver={MySQL ODBC 8.0 ANSI  
Driver};Server=localhost;Database=fitnessclub;uid=root;pwd=  
zxc123/.,098";  
OdbcConnection MyConnection = new  
OdbcConnection(MyConnectionString);
```

```

MyConnection.Open();
OdbcCommand MyCommand = new
OdbcCommand("ЗАПИТ", MyConnection);
MyCommand.ExecuteNonQuery();
MyConnection.Close();

```

Перший рядок задає драйвер ODBC та особливості підключення до бази даних. Другим рядком створюється змінна `MyConnection`, що уособлює саме з'єднання та включатиме його параметри. Третім рядком підключення відкривається за допомогою метода `Open()`. У четвертому рядку створюється нова змінна, що відповідає команді до бази даних, яка містить як сам SQL-запит (його текст слід підставити замість слова **ЗАПИТ**), так і змінну з'єднання, по якому слід виконати цю команду. Передостанній рядок – це безпосереднє виконання SQL-команди на сервері баз даних, а останній – закриття з'єднання. Відмітимо, що вказаний метод `ExecuteNonQuery()` використовується для виконання SQL-команд, які не повертають жодних даних (директивні команди типу створити/видалити таблицю чи базу даних, введення відомостей у БД, тощо). Якщо необхідно виконати запит типу `SELECT`, який передбачає повернення даних (як у розроблюваному програмному забезпеченні), то слід використовувати замість методу `ExecuteNonQuery()`, метод `ExecuteReader()`.

Відмітимо, що ще одним додатковим програмним засобом, що використовувався в роботі, є `phpMyAdmin` – утиліта, написана мовою PHP спеціально для зручної роботи у СУБД MySQL. З використанням даного програмного засобу створено наступну схему даних для системи, що проектується, що показана на рис. 2.7.

Изм.	Аркуш.	№ докум.	Підпис	Дата

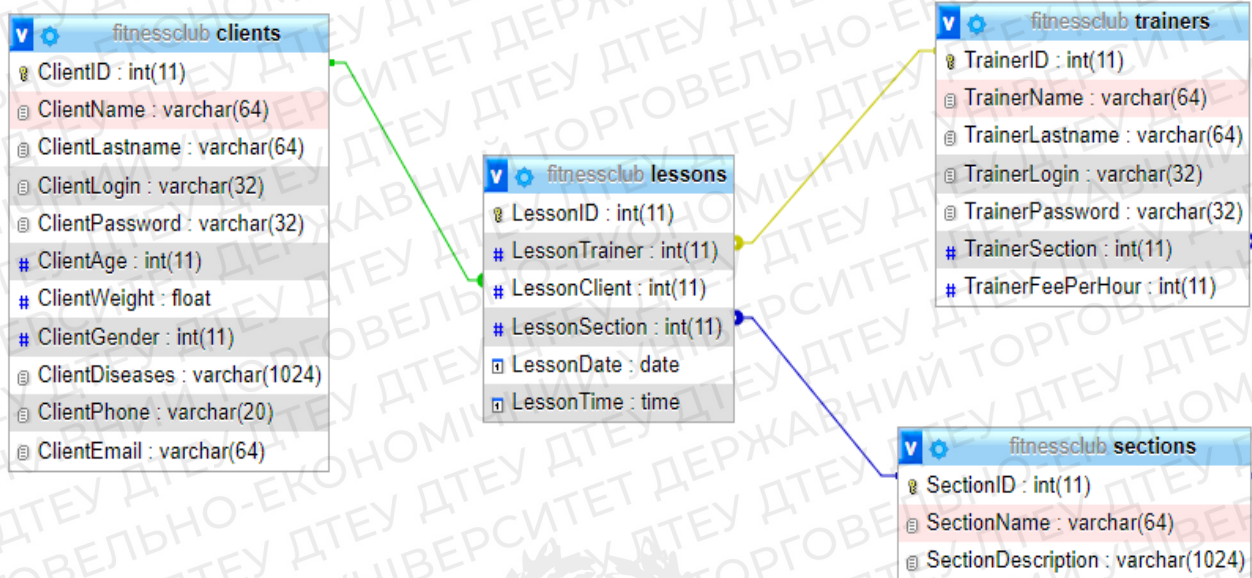


Рис. 2.7. Схема даних розробленої бази даних fitnessclub

Таким чином, структуру бази даних реалізовано, а наповнення виконано на достатньому рівні для того, щоби продемонструвати усі функціональні можливості системи, тобто її відповідність до технічного завдання на розробку.

## 2.4 Висновки до розділу 2

Таким чином, у даному розділі виконано розробку проектних рішень для реалізації програмної компоненти інформаційної системи фітнес-клубу. Виконано моделювання предметної галузі мовою UML, обрано тип додатку (настільне програмне забезпечення), дворівневу архітектуру клієнт-сервер, об'єктно-орієнтовану технологію розробки, та мову програмування C# (у середовищі програмування Microsoft Visual Studio 2019 Community). Обрано СУБД MySQL та розроблено структуру відповідної бази даних інформаційної системи.

## РОЗДІЛ 3

### ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ КОМПОНЕНТИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ФІТНЕС-КЛУБУ

#### 3.1 Проектування інтерфейсу користувача програмної компоненти інформаційної системи фітнес-клубу

У програмного продукту, який підлягає реалізації, наявні дві ярко виражені ролі користувачів: клієнт фітнес-клубу та працівник (адміністратор або тренер) фітнес-клубу. Відповідно, можна реалізувати два окремих програмних продукти, що працюватимуть з однією і тією ж базою даних, але матимуть різний інтерфейс користувача. Такий підхід обумовлений тим, що функціональність для цих двох груп користувачів відрізняється кардинальним чином.

Отже, для клієнта слід передбачити елементи управління чи інші функціональні елементи для:

- проведення авторизації;
- перегляду існуючих занять;
- можливості відміни заняття;
- створення нового заняття (планованого);
- перегляду інформації про програму.

Зовнішній вигляд інтерфейсу клієнтської частини показано на рис. 3.1. Оскільки інформації у клієнта менше, аніж у працівника-адміністратора, то усю її компонуємо на одній формі – на головному вікні програми. Довідкова інформація про програму виводиться окремим вікном – рис. 3.2.

					<i>ДТЕУ 121 07-12.БР</i>			
					<i>Програмна компонента інформаційної системи фітнес клубу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		РЗ	39	59
Зав. каф.		Криворучко О.В.		11.04.22	<i>Особливості реалізації програмної компоненти інформаційної системи фітнес-клубу</i>	Факультет інформаційних технологій, 4 курс, 7 група		
Керівник		Рзаєва С. Л.		11.04.22				
Гарант		Рзаєва С. Л.		11.04.22				
Розроб.		Потапов Д. Р.		11.04.22				

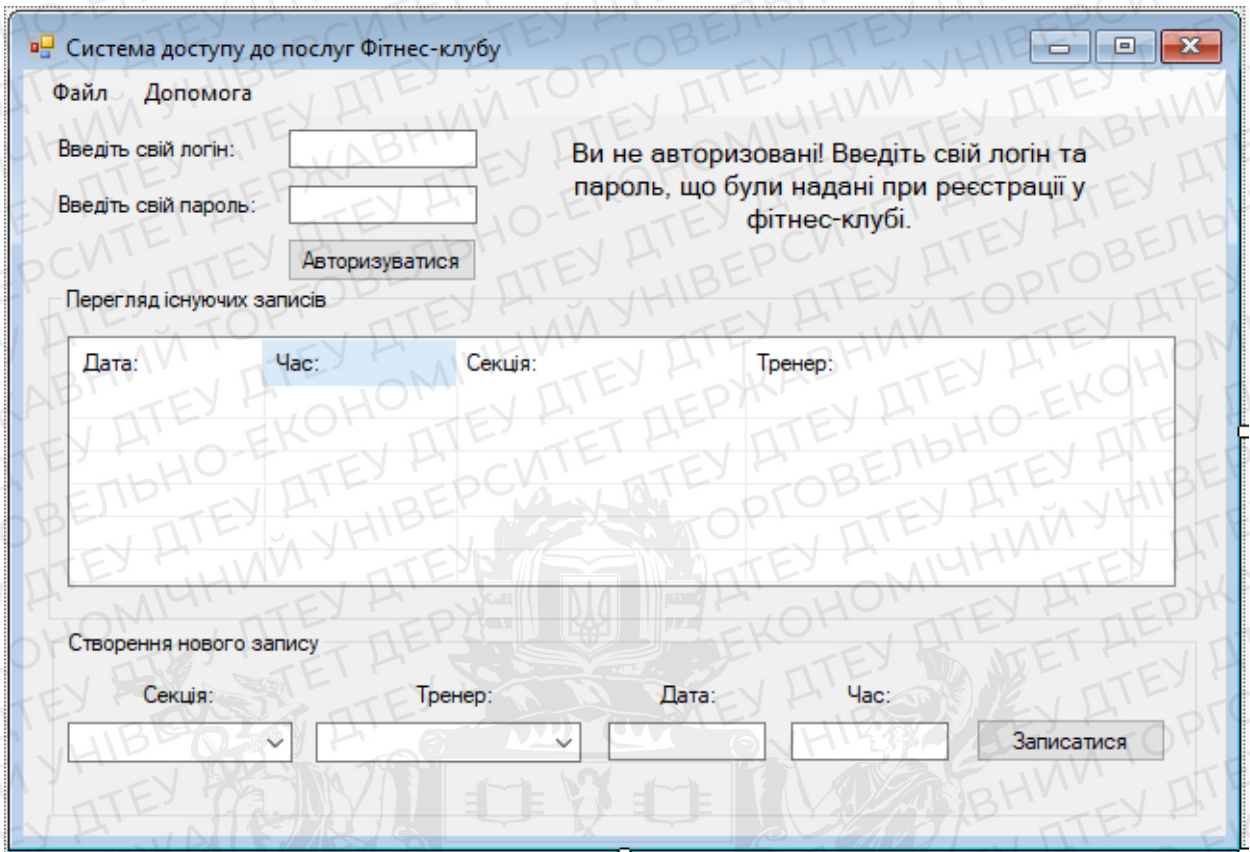


Рис. 3.1. Зовнішній вигляд головного вікна клієнтської частини розробленого програмного забезпечення фітнес-клубу

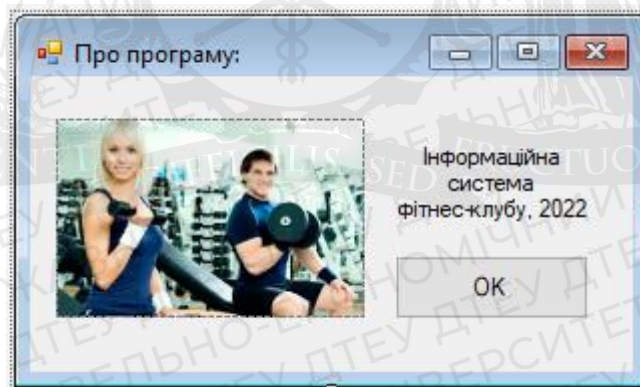


Рис. 3.2. Зовнішній вигляд вікна «Про програму» розробленого програмного забезпечення

Переходячи до частини програмного забезпечення, призначеного для адміністрування, у ній можна виділити елементи управління для:

- проведення авторизації;
- перегляду існуючих секцій;

- перегляду існуючих тренерів;
- перегляду існуючих клієнтів;
- видалення існуючих секцій;
- видалення існуючих тренерів;
- видалення існуючих клієнтів;
- додавання нових секцій;
- додавання нових тренерів;
- додавання нових клієнтів;
- вирахування заробітної платні для тренерів за місяць.

Кількість елементів управління, що необхідні для реалізації вказаних дій є досить великою, тому розбиваємо їх на чотири закладки:

- авторизація;
- клієнти;
- тренери;
- секції.

Зовнішній вигляд відповідних вкладок показаний на рис. 3.3-3.6.

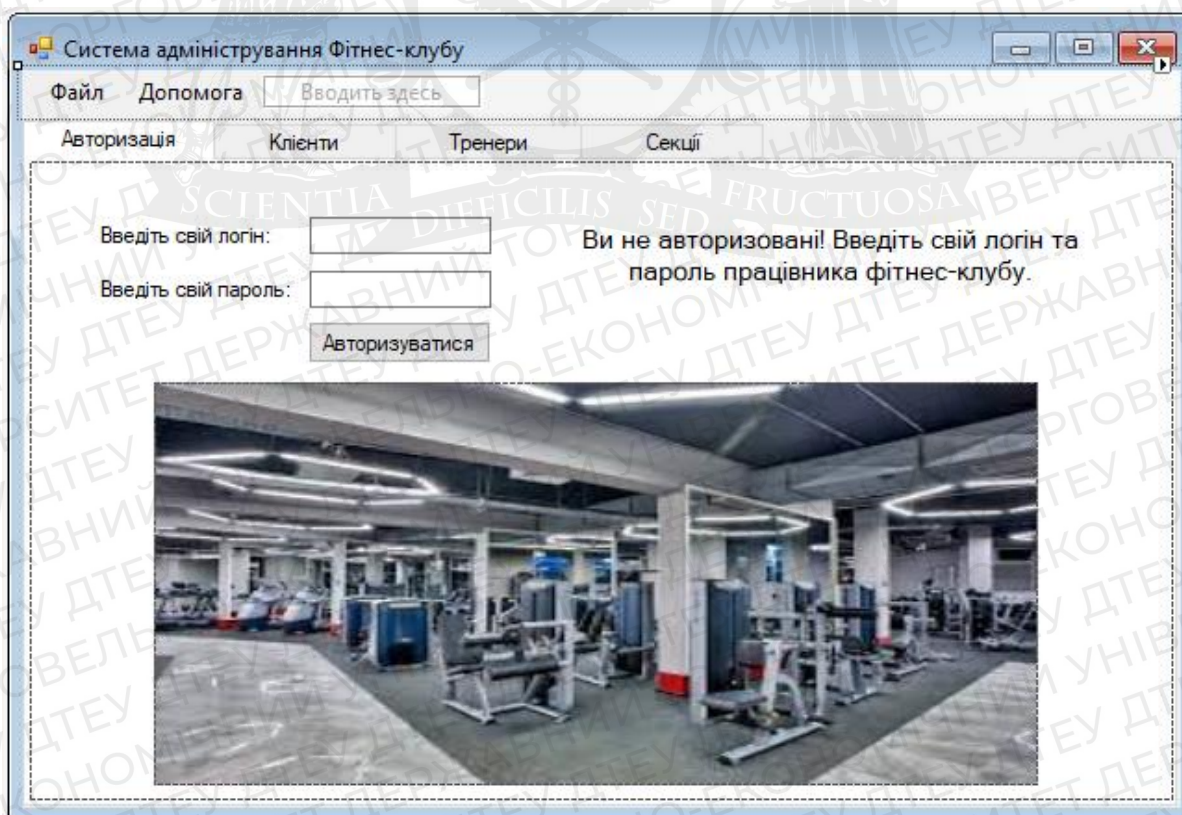


Рис. 3.3. Вкладка «Авторизация» интерфейсу адміністратора

						Аркуш
						41
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР	

Система адміністрування Фітнес-клубу

Файл    Допомога

Авторизація    Клієнти    Тренери    Секції

Перегляд існуючих у БД клієнтів

Прізвище:	Ім'я	Вік:	Вага:	Ста...	Хвороби:	Телефон:	E-mail

Створення нового клієнту

Прізвище:     Ім'я:     Вік:     Вага:     Стать:    

Логін:     Пароль:     Телефон:     E-mail:     Хвороби:

Рис. 3.4. Вкладка «Клієнти» інтерфейсу адміністратора

Система адміністрування Фітнес-клубу

Файл    Допомога

Авторизація    Клієнти    Тренери    Секції

Перегляд існуючих у БД тренерів

Прізвище:	Ім'я:	Секція:	Оплата, грн./год.

Місяць (01, 02, ..., 12):    

Створення нового тренера

Секція:     Прізвище:     Ім'я:     Оплата за годину:    

Логін:     Пароль:

Рис. 3.5. Вкладка «Тренери» інтерфейсу адміністратора



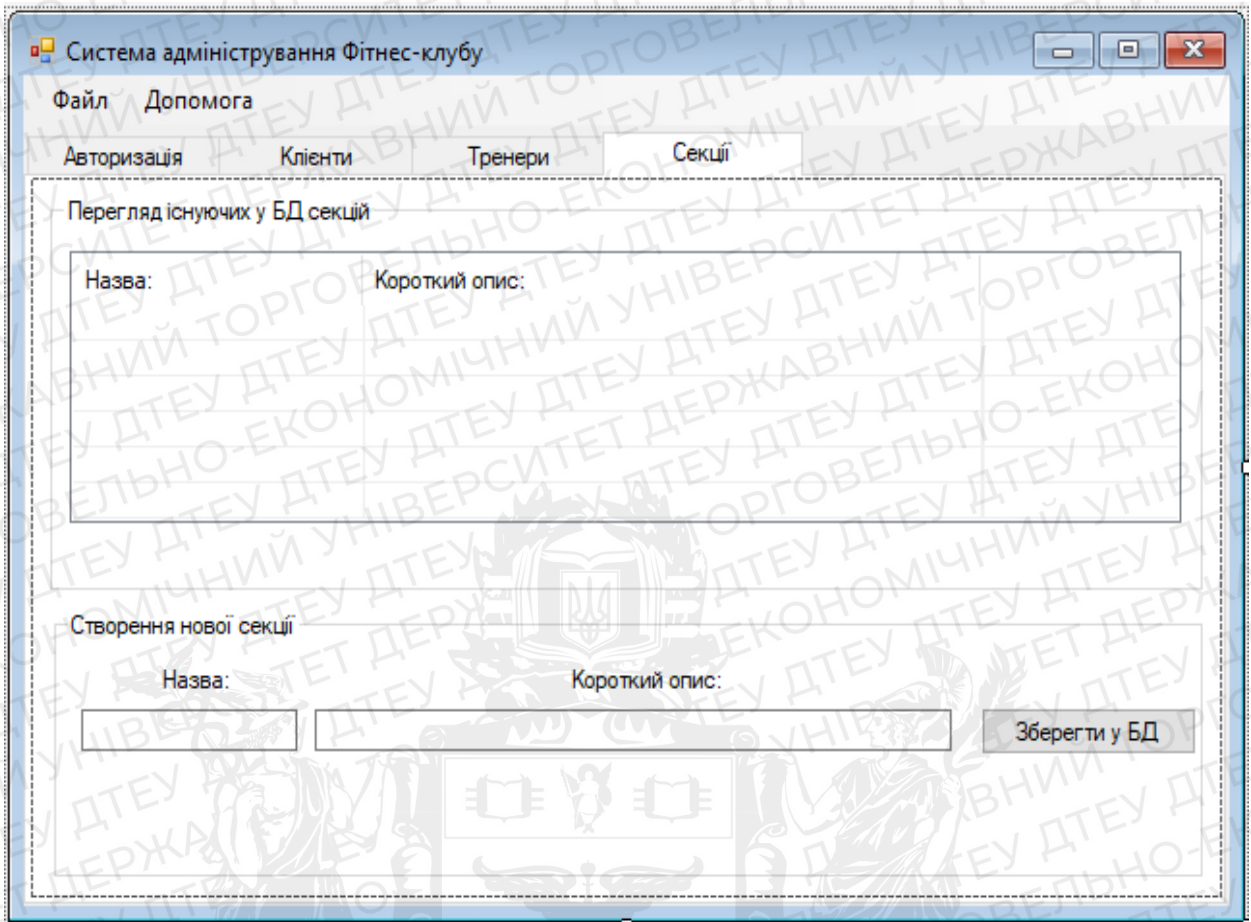


Рис. 3.6. Вкладка «Секції» інтерфейсу адміністратора

### 3.2 Реалізація бази даних програмної компоненти інформаційної системи фітнес-клубу

Базу даних проекту, відповідно до вищенаведеної інформації, будемо реалізовувати у вигляді поєднання чотирьох наступних таблиць:

а) таблиця clients, яка містить усю необхідну для роботи фітнес-клубу інформацію про його клієнтів та має структуру, наведену на рис. 3.7.

						Аркуш
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР	
						43

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1 ClientID 🗝	int(11)			Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2 ClientName	varchar(64)	utf8mb4_general_ci		Нет	Нет		
<input type="checkbox"/>	3 ClientLastname	varchar(64)	utf8mb4_general_ci		Нет	Нет		
<input type="checkbox"/>	4 ClientLogin	varchar(32)	utf8mb4_general_ci		Нет	Нет		
<input type="checkbox"/>	5 ClientPassword	varchar(32)	utf8mb4_general_ci		Нет	Нет		
<input type="checkbox"/>	6 ClientAge	int(11)			Нет	Нет		
<input type="checkbox"/>	7 ClientWeight	float			Нет	Нет		
<input type="checkbox"/>	8 ClientGender	int(11)			Нет	Нет		
<input type="checkbox"/>	9 ClientDiseases	varchar(1024)	utf8mb4_general_ci		Нет	Нет		
<input type="checkbox"/>	10 ClientPhone	varchar(20)	utf8mb4_general_ci		Нет	Нет		
<input type="checkbox"/>	11 ClientEmail	varchar(64)	utf8mb4_general_ci		Нет	Нет		

Рис. 3.7. Структура створеної таблиці clients

б) таблиця trainers, яка містить інформацію про тренерів, які працюють у фітнес-клубі, та має структуру, показану на рис. 3.8.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1 TrainerID 🗝	int(11)			Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2 TrainerName	varchar(64)	utf8mb4_general_ci		Нет	Нет		
<input type="checkbox"/>	3 TrainerLastname	varchar(64)	utf8mb4_general_ci		Нет	Нет		
<input type="checkbox"/>	4 TrainerLogin	varchar(32)	utf8mb4_general_ci		Нет	Нет		
<input type="checkbox"/>	5 TrainerPassword	varchar(32)	utf8mb4_general_ci		Нет	Нет		
<input type="checkbox"/>	6 TrainerSection 🗝	int(11)			Нет	Нет		
<input type="checkbox"/>	7 TrainerFeePerHour	int(11)			Нет	Нет		

Рис. 3.8. Структура розробленої таблиці trainers

в) таблиця sections, яка включає інформацію про секції, за якими можна проходити заняття у даному фітнес-клубі (тобто такі секції, по яким у клубі наявні тренери). Структура таблиці показана на рис. 3.9.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1 SectionID 🗝	int(11)			Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2 SectionName	varchar(64)	utf8mb4_general_ci		Нет	Нет		
<input type="checkbox"/>	3 SectionDescription	varchar(1024)	utf8mb4_general_ci		Нет	Нет		

Рис. 3.9. Структура таблиці sections

г) таблиця lessons містить усю інформацію про заняття, проведені та заплановані у фітнес-клубі. Структура цієї таблиці, яка показана на рис. 3.10, насичена посиланнями на таблиці клієнтів, тренерів та секцій.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/> 1	LessonID	int(11)			Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/> 2	LessonTrainer	int(11)			Нет	Нет		
<input type="checkbox"/> 3	LessonClient	int(11)			Нет	Нет		
<input type="checkbox"/> 4	LessonSection	int(11)			Нет	Нет		
<input type="checkbox"/> 5	LessonDate	date			Нет	Нет		
<input type="checkbox"/> 6	LessonTime	time			Нет	Нет		

Рис. 3.10. Структура таблиці lessons

### 3.3 Реалізація розроблених алгоритмів роботи програмної компоненти інформаційної системи фітнес-клубу

Розглянемо тут самі основні особливості реалізованого програмного коду, а за необхідності більш детально можна ознайомитися із ним у Додатках.

При виконанні операцій додавання до БД інформації попередньо виконуються численні перевірки усіх значень, що вносяться, за допомогою умовних операторів виду:

а) перевірка, що у списку, який випадає, було вибрано хоча б якесь значення, наприклад:

```
if (comboBox1.SelectedIndex<0)
{
    MessageBox.Show(this, "Задайте, будь-ласка, секцію!");
    return;
}
```

б) перевірка, що текстове поле не є порожнім, наприклад:

```
if (textBox14.Text == "")
{
    MessageBox.Show(this, "Задайте, будь-ласка, прізвище!");
    return;
}
```

в) більш інтелектуальні перевірки вмісту текстових полів, що відрізняються, залежно від того, яке саме значення перевіряється, наприклад:

```
if (textBox20.Text.Length!=2)
```

```

    {
        MessageBox.Show(this, "Формат місяця має бути: 01, 02, 03, ...,
11, 12");
        return;
    }

```

Наступною особливістю коду є повний захист від помилок (виключень), які пов'язані із роботою з базою даних, який реалізується вміщенням усього такого коду у секції try-catch виду:

```

try
{
    //робота з базою даних
}
catch (OdbcException MyOdbcException)
{
    MessageBox.Show(this, "Виникла помилка доступу до бази даних! Її
текст: " + MyOdbcException.Errors[0].Message);
}

```

Саме читання із бази даних виконується за допомогою членів класу **OdbcDataReader**, наприклад:

```

querystr = "SELECT * FROM sections;";
OdbcCommand MyCommand = new OdbcCommand(@querystr, MyConnection);
OdbcDataReader MyDataReader;
MyDataReader = MyCommand.ExecuteReader();
if (MyDataReader.HasRows)
{
    if (MyDataReader.HasRows)
    {
        int NumRes = 0;
        while (MyDataReader.Read())
        {
            String sectionname;
            sectionname = MyDataReader.GetString(MyDataReader.
GetOrdinal("SectionName"));
            //робота зі зчитаними записами
            NumRes++;
        }
    }
    else
    {
        MessageBox.Show(this, "У базі даних відсутні секції!");
    }
    //Close all resources
    MyDataReader.Close();
}

```

Інші особливості коду розробленого програмного забезпечення можна встановити безпосередньо переглядаючи цей у код, розміщений у додатках до даної роботи.

### 3.4 Тестування створеної програмної компоненти інформаційної системи фітнес-клубу

Для будь-якого нового програмного забезпечення потрібно провести процедуру його тестування. Застосуємо ручне системне бета-тестування по схемі білого ящика із позитивною перевіркою. Важливим елементом процесу тестування є контрольний приклад виконання програми. Для клієнтської частини контрольний приклад показано на рис. 3.11-3.14. Спершу програма запускається у неавторизованому стані і усі елементи управління не є активними – рис. 3.11.

Система доступу до послуг Фітнес-клубу

Файл    Допомога

Введіть свій логін:

Введіть свій пароль:

Авторизуватися

Ви не авторизовані! Введіть свій логін та пароль, що були надані при реєстрації у фітнес-клубі.

Перегляд існуючих записів

Дата:	Час:	Секція:	Тренер:

Створення нового запису

Секція:  Тренер:  Дата:  Час:

Записатися

Рис. 3.11. Стартовий екран клієнтської частини до виконання успішної авторизації

						Аркуш
						47
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР	

Якщо авторизація неуспішна, то переходу до наступних етапів роботи із програмою не відбувається, а користувачу видається невелике попереджувальне повідомлення. Після успішної авторизації з'являється можливість роботи у програмі і користувачеві видається весь список його занять – рис. 3.12.

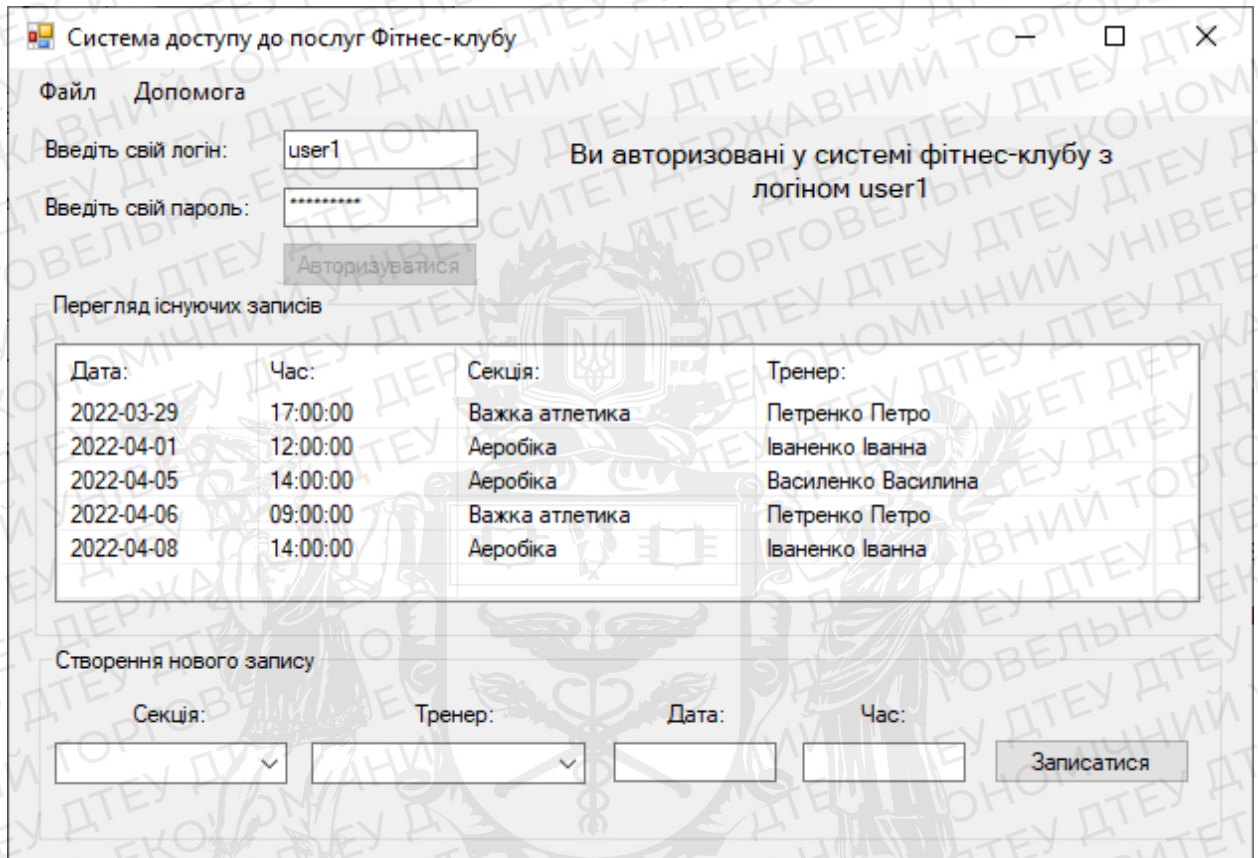


Рис. 3.12. Робоча область вікна клієнтської частини розробленого програмного забезпечення після успішної авторизації

Після заповнення усіх полів, що відображуються у нижній частині вікна програми та натискання на кнопку «Записатися» відбувається успішне внесення даних у базу і видача відповідного інформаційного повідомлення – рис. 3.13. При натисненні на кнопку «ОК» видається оновлений список занять, в якому присутнє нове заняття – рис. 3.14. Якщо двічі клацнути у таблиці занять на одному з рядків, то відбувається його видалення: наприклад, якщо двічі клацнути по новому рядку, що був внесений на рис. 3.14, то програма видає інформаційне повідомлення про успішне видалення і переходить до вигляду екрану рис. 3.12.

Аналогічно успішно виконується і робота частини програмного забезпечення, призначеного для адміністрування – рис. 3.15-3.19.

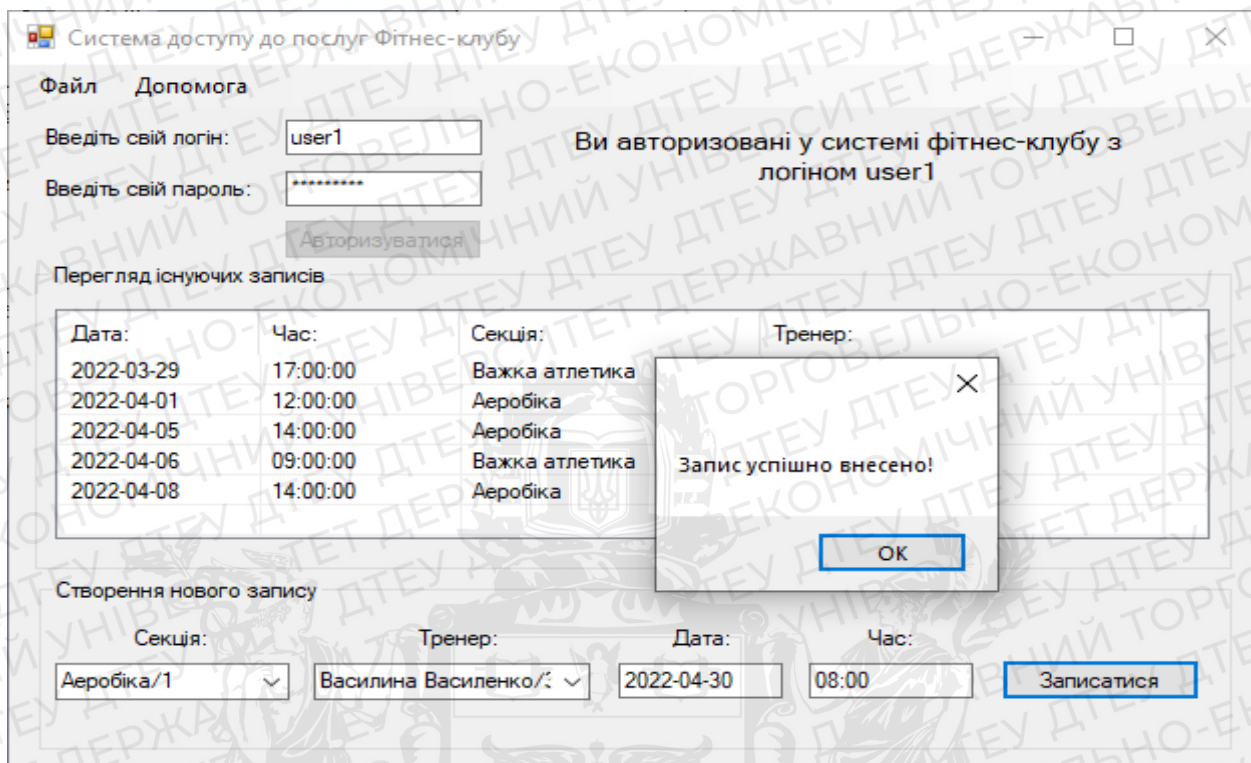


Рис. 3.13. Вікно клієнтської частини при успішному внесенні у БД

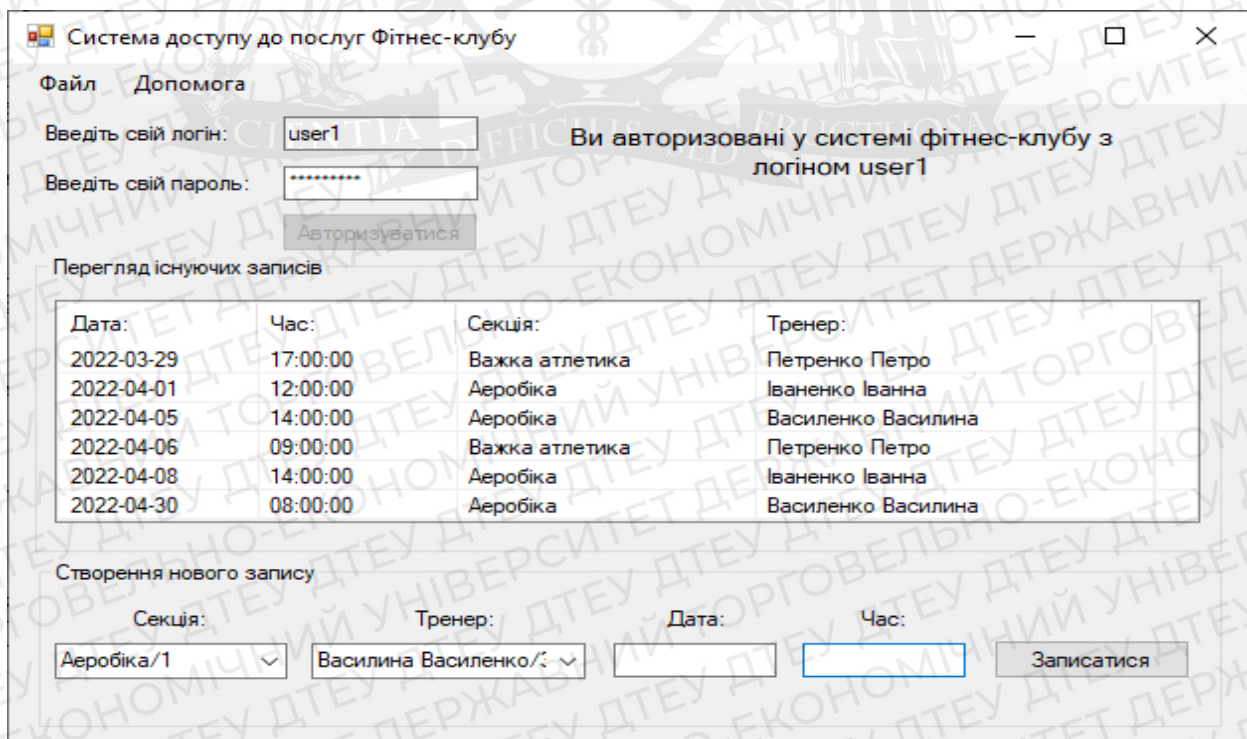


Рис. 3.14. Вікно клієнтської частини при завершенні внесення у БД

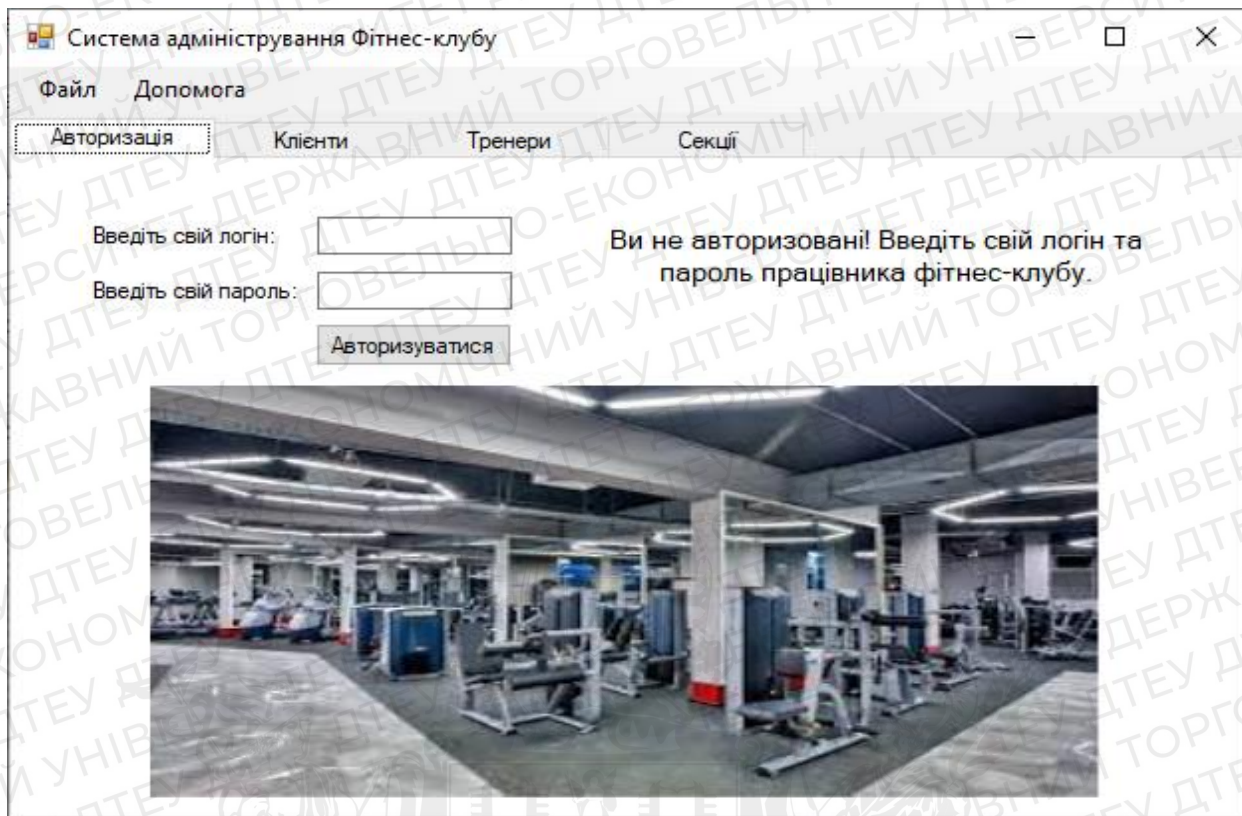


Рис. 3.15. Вікно адмінчастини до виконання успішної авторизації

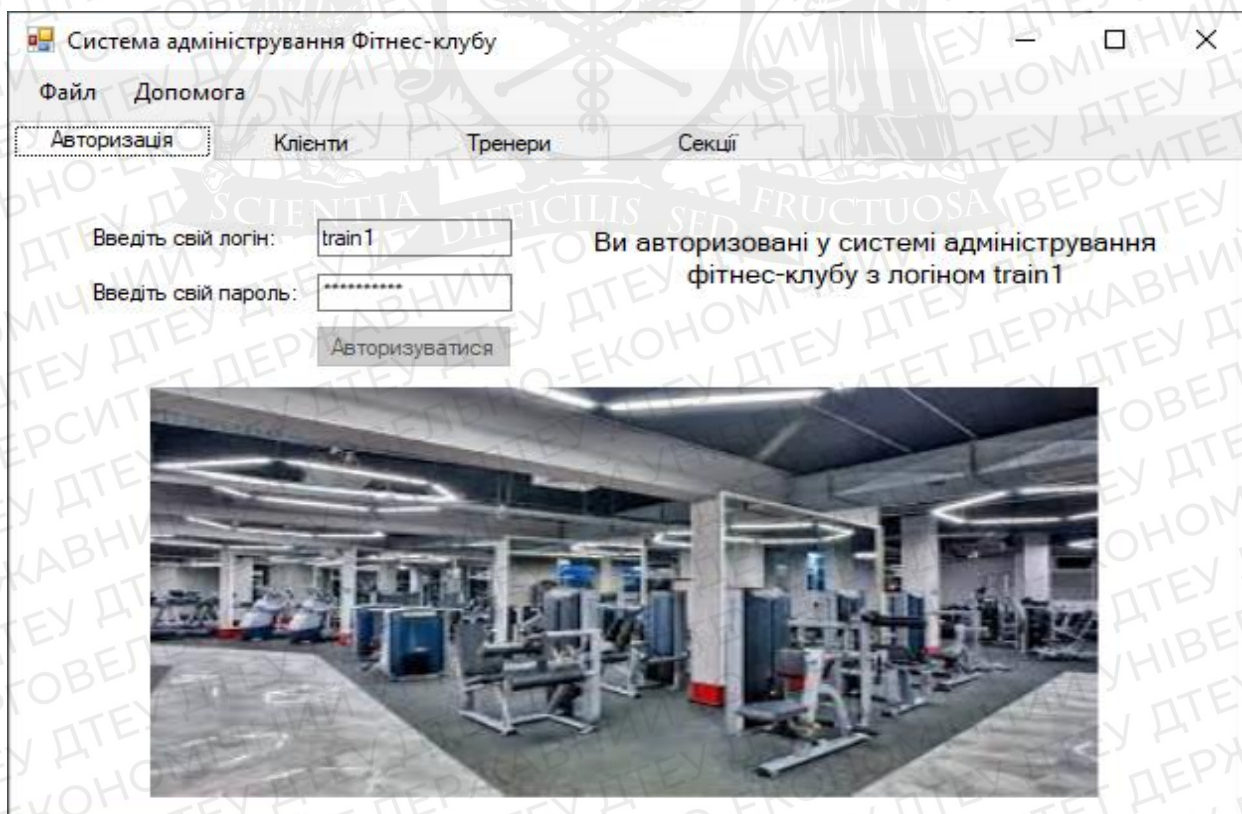


Рис. 3.16. Вікно адмінчастини після виконання успішної авторизації

					Аркуш
					50
Изм.	Аркуш.	№ док.ум.	Підпис	Дата	ДТЕУ 121 07-12.БР



Система адміністрування Фітнес-клубу

Файл    Допомога

Авторизація    **Клієнти**    Тренери    Секції

Перегляд існуючих у БД клієнтів

Прізвище:	Ім'я	Вік:	Вага:	Ста...	Хвороби:	Телефон:	E-mail
Петрова	Ганна	30	90	0	Хвороб не має	+380-99-99...	hannapetrova@...

Створення нового клієнту

Прізвище:     Ім'я:     Вік:     Вага:     Стать:    

Логін:     Пароль:     Телефон:     E-mail:     Хвороби:

Рис. 3.17. Закладка адміністративної частини «Клієнти»

Система адміністрування Фітнес-клубу

Файл    Допомога

Авторизація    Клієнти    **Тренери**    Секції

Перегляд існуючих у БД тренерів

Прізвище:	Ім'я:	Секція:	Оплата, грн./год.
Петренко	Петро	Важка атлетика	200
Іваненко	Іванна	Аеробіка	250
Василенко	Василина	Аеробіка	200

Місяць (01, 02, ..., 12):    

Створення нового тренера

Секція:     Прізвище:     Ім'я:     Оплата за годину:    

Логін:     Пароль:

Рис. 3.18. Закладка адміністративної частини «Тренери»

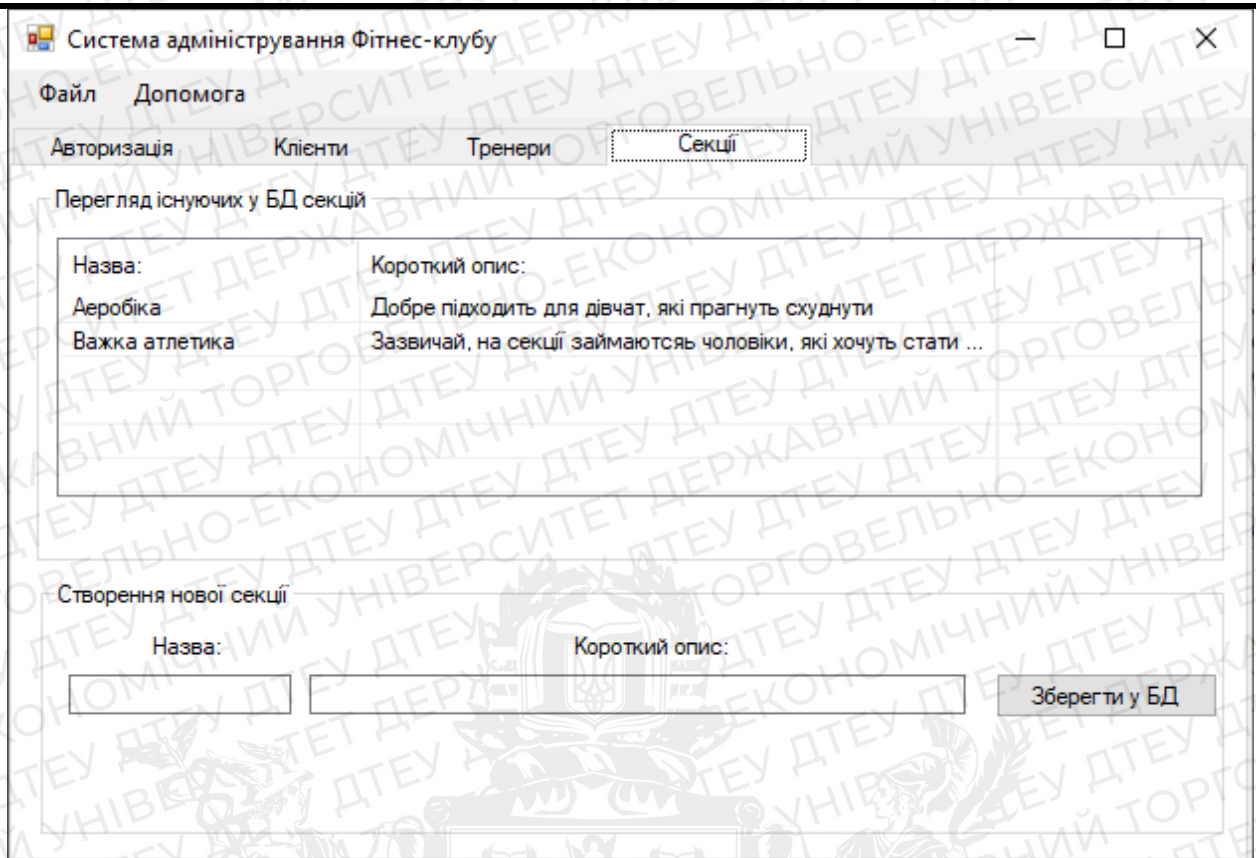


Рис. 3.19. Закладка адміністративної частини «Секції»

На кожній із закладок, показаних на рис. 3.16-3.19 успішно виконуються операції додавання нових записів у базу даних, а також видалення уже існуючих записів, відображених у відповідній таблиці.

Таким чином, тестування програмного продукту можна вважати успішним і переходити до його реального практичного використання.

### 3.5 Розробка керівництва користувача розробленої програмної компоненти інформаційної системи фітнес-клубу

У програмному продукті виділяється дві частини: для клієнта фітнес-клубу керівництво користувача може бути представлено у вигляді наступного списку можливих операцій:

- а) Завантажити програму стандартним для ОС Windows способом (наприклад, клацнувши двічі мишею по її піктограмі);

б) Ввести логін та пароль і натиснути на кнопку «Авторизуватися», після чого зачекати успішної авторизації, а якщо виникла помилка, то перевірити введений логін та пароль і повторити спробу;

в) Продивитися перелік усіх існуючих на даний момент часу у базі даних занять;

г) За необхідності видалення якогось запису (якщо плани змінилися і дане тренування уже не потрібно) – клацнути мишею два рази по відповідному рядку, після чого даний рядок повинен зникнути, а програма має видати інформаційне повідомлення про успішність операції видалення;

д) За необхідності внесення інформації про нове плановане тренування – слід заповнити чотири поля в нижній частині вікна програми і натиснути кнопку «Записатися». При цьому поле «Дата» повинно мати формат YYYY-MM-DD (наприклад, 2022-01-31), а поле «Час» повинно мати формат HH:MM (наприклад, 11:00 або 17:30). Після натиснення на кнопку «Записатися» програма має видати інформаційне повідомлення про успішність операції додавання нового тренування, а у списку (який відсортований за датами та часом) повинен з'явитися новий рядок із тільки що введеною інформацією. Якщо при введенні інформації була допущена помилка, то запис можна видалити (пункт 2 цієї інструкції) і створити новий;

е) За бажанням можна прочитати інформацію про програму, для чого слід у головному меню вибрати пункт «Допомога» - «Про програму...».

є) Коли усі необхідні дії у програмі вже виконано її можна закрити будь-яким стандартним для ОС Windows способом (зокрема через головне меню «Файл» - «Вихід»).

Для працівника фітнес-центру, який займається питаннями адміністрування створеного програмного забезпечення, інструкція щодо його використання також може бути представлена по пунктах:

а) Завантажити програму;

б) Авторизуватися за допомогою робочого логіну та паролю;

в) Перейти на одну із трьох закладок, залежно від того, із чим саме слід працювати у даний момент: Клієнти, Тренери, Секції.

						Аркуш
					ДТЕУ 121 07-12.БР	53
Изм.	Аркуш.	№ докум.	Підпис	Дата		

г) Якщо необхідно додати нового клієнта, то на відповідній закладці слід ввести з клавіатури усі його персональні дані, створити йому логін (наприклад, у вигляді імені і паролю, записаних транслітом і розділених крапкою, або у вигляді user87, де 87 – номер даного клієнта, або будь-яким іншим способом) та вигадати випадковий пароль. Коли усі поля будуть заповнені слід натиснути кнопку «Зберегти в БД» та пересвідчитися, що цей клієнт успішно внесений у базу даних (наприклад, це можна зробити впевнившись, що він з'явився у загальному списку клієнтів).

д) Якщо необхідно додати нового тренера, то на закладці «Тренери» слід ввести з клавіатури його прізвище та ім'я, створити логін та пароль, вказати величину погодинної оплати тренера і секцію, за якою він працює (якщо секції немає, то попередньо слід створити її – див. пункт е). Коли усі поля будуть заповнені слід натиснути кнопку «Зберегти в БД» та пересвідчитися, що цей тренер успішно внесений у базу даних (наприклад, це можна зробити впевнившись, що він з'явився у загальному списку тренерів).

е) Якщо необхідно додати нову секцію, то слід перейти на закладку «Секції», ввести назву та короткий опис і натиснути кнопку «Зберегти в БД». Після цього слід пересвідчитися, що нова секція додана у БД.

є) У будь-якій з трьох таблиць можна видаляти позиції, для чого слід двічі клацнути по відповідному рядку, але потрібно звернути увагу, що через необхідність дотримання цілісності бази даних, видалятися можуть лише ті записи, що не задіяні в записах інших таблиць (тобто видалити можна тільки тренера, який не провів жодного заняття, або секцію, по якій не було занять і немає жодного тренера. Наприклад, легко можна видалити записи, що були щойно внесені з помилкою).

ж) За необхідності програма дозволяє порахувати заробітну платню за місяць для усіх тренерів. Для цього слід перейти на закладку «Тренери» та заповнити поле «Рік» (обов'язково повних чотири цифри) та «Місяць» (обов'язково дві повних цифри, наприклад, 01, 02), після чого натиснути кнопку «Розрахувати ЗП». Слід відмітити, що зарплата нараховується по простій формулі як добуток кількості відпрацьованих за місяць годин на величину ставки погодинної оплати праці, встановленої для кожного тренера. В результаті в таблиці з інформацією про тренерів з'явиться нова

						ДТЕУ 121 07-12.БР	Аркуш
Изм.	Аркуш.	№ докум.	Підпис	Дата			54

колонка із сумами, що їх напрацював кожен тренер за місяць. За необхідності можна поміняти місяць і здійснити новий розрахунок без перезавантаження програми.

з) Коли робота з програмою завершена, слід закрити її будь-яким зручним для користувача способом (комбінацією Alt+F4, натискаючи хрестик у правому верхньому куті вікна програми, через головне меню програми – «Файл» – «Вихід».

Таким чином, використовуючи два наведених списки, можна керувати роботою обох частин розробленого програмного забезпечення.

### 3.6 Висновки до розділу 3

Таким чином, у даному розділі описано реалізацію програмної компоненти інформаційної системи фітнес-клубу. Описується розробка інтерфейсу користувача програмного забезпечення, причому для двох частин програмного продукту: для клієнта фітнес-клубу та для його адміністратора. У розділі розроблена база даних інформаційної системи фітнес-клубу, відповідно до проекту, створеного у попередньому розділі 2. Наведено основні особливості виконання реалізації у початкових кодах створеного програмного забезпечення. Тестування, проведене для усіх варіантів використання програмної компоненти інформаційної системи фітнес-клубу, показало належне функціонування програмного продукту та відповідність технічному завданню на розробку, наведеному у підрозділі 1.3. Дана розробка може бути рекомендована до реального практичного використання у фітнес-клубі.

						Аркуш
					ДТЕУ 121 07-12.БР	55
Изм.	Аркуш.	№ докум.	Підпис	Дата		

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Отже, у даній роботі проведено проектування архітектури програмної компоненти інформаційної системи фітнес-клубу та виконано її реалізацію у початкових кодах. Використання даної системи дозволяє підвищити продуктивність праці співробітників фітнес-клубу, а також підвищити зручність запису на заняття для клієнтів фітнес-клубу.

В роботі першочергово було проведено аналіз предметної галузі, а саме типових бізнес-процесів, що відбуваються у фітнес-клубі; встановлено вид та структуру інформації, що є необхідною для роботи даної організації та має оброблятися інформаційною системою фітнес-клубу. Проведений аналіз аналогічних програмних продуктів показав, що у переважній більшості вони мають перевантажений інтерфейс користувача, що може заважати ефективному використанню цих продуктів окремими групами осіб (користувачами із низьким рівнем комп'ютерної грамотності), а також часто у аналогічних програмах є присутньою платна функціональність. Відповідно встановлено доцільність проведення власної розробки та оформлено технічне завдання на розробку програмної компоненти інформаційної системи фітнес-клубу.

По-друге, розроблено комплекс проектних рішень, необхідних для подальшої програмної реалізації, а саме: виконано моделювання предметної галузі мовою UML, обрано тип додатку (настільне програмне забезпечення) та запропоновано для нього дворівневу архітектуру типу «клієнт-сервер» (настільний програмний продукт виконує роль клієнту, а СУБД – роль серверу). Обґрунтовано використання об'єктно-орієнтованої технології розробки та мови програмування C# (у середовищі програмування Microsoft Visual Studio 2019 Community), розглянуто основні їх можливості. Обрано СУБД MySQL та розроблено структуру відповідної бази даних

					<i>ДТЕУ 121 07-12.БР</i>			
					<i>Програмна компонента інформаційної системи фітнес клубу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		ВП	56	59
Зав. каф.		Криворучко О.В.		25.04.22				
Керівник		Рзасва С. Л.		25.04.22				
Гарант		Рзасва С. Л.		25.04.22				
Розроб.		Потапов Д. Р.		25.04.22	<i>Висновки та пропозиції</i>	Факультет інформаційних технологій, 4 курс, 7 група		

програмної компоненти інформаційної системи фітнес-клубу.

По-третє, спроектовано та реалізовано інтерфейс користувача програмної компоненти, причому у двох варіантах: для клієнта клубу та для його працівника (адміністратора клубу, що працює за стійкою при вході). Описується реалізація таблиць спроектованої бази даних, а також наводяться особливості програмної реалізації кодів інформаційної системи мовою C#.

Для зведеної програмної реалізації проведено тестування, яке показало цілком задовільний рівень її роботи та повну відповідність завданню на розробку. Таким чином, ціль роботи можна вважати досягнутою, а саму розробку можна рекомендувати для практичного використання у фітнес-клубах.

У перспективі можливе розширення спектру даних, що підлягають обліку в діяльності фітнес-клубу, та включення відповідних функцій по їх обробці у створений програмний продукт.



									Аркуш
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР				57

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Господарський кодекс України: Закон України № 436-IV від 16.01.2003 р. [Електронний ресурс] // Відомості Верховної Ради України (ВВР), – 2003. – №18-22 – Режим доступу: <http://zakon3.rada.gov.ua/laws/show/436-15>.

2. Національний Класифікатор України. Класифікація видів економічної діяльності ДК 009:2010 // Державний Комітет України з питань технічного регулювання та споживчої політики, К.: 2010.

3. В. П. Нагребельний. Облік // Юридична енциклопедія : [у 6 т.] / ред. кол.: Ю. С. Шемшученко (відп. ред.) [та ін.]. — К. : Українська енциклопедія ім. М. П. Бажана, 2002. — Т. 4 : Н — П. — 720 с. — ISBN 966-7492-04-4.

4. Грибан Г. П. Методична система фізичного виховання студентів : навчальний посібник. Житомир : Рута, 2014. – 305 с.

5. Костюкевич В.М. та ін. Основи науково-дослідної роботи здобувачів вищої освіти зі спеціальності "Фізична культура і спорт" : навчальний посібник / В. М. Костюкевич, О. А. Шинкарук, В. І. Воронова, О. В. Борисова ; Національний університет фізичного виховання і спорту. Київ : Олімпійська література, 2018. – 526 с.

6. Лядська О. Ю. Застосування комп'ютерної програми "Fitball training" для удосконалення організації фізкультурно-оздоровчих занять з жінками першого зрілого віку із застосуванням футболу / О. Ю. Лядська // Педагогіка, психологія та медико-біологічні проблеми фізичного виховання і спорту. – 2010. – № 12. – С. 76-80.

7. Каплінський В. В., Асаулюк І. О.. Основи виховної діяльності вчителя фізичної культури : навч. посіб. Вінниця : Едельвейс і К, 2014. – 291 с.

					<i>ДТЕУ 121 07-12.БР</i>			
					<i>Програмна компонента інформаційної системи фітнес клубу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		СВД	58	59
Зав. каф.		Криворучко О.В.		13.12.21	<i>Список використаних джерел</i>	Факультет інформаційних технологій, 4 курс, 7 група		
Керівник		Рзаєва С. Л.		13.12.21				
Гарант		Рзаєва С. Л.		13.12.21				
Розроб.		Потапов Д. Р.		13.12.21				



8. Вознюк Т. В. Основи теорії та методики спортивного тренування : навчальний посібник. Вінниця : ФОП Корзун Д. Ю., 2016. – 236 с.

9. Швець О.П. Курс лекцій та матеріал для самостійної роботи студентів вищих навчальних закладів з дисципліни фізичне виховання : навчальний посібник / О. П. Швець ; Вінницький державний педагогічний університет ім. М. Коцюбинського. Вінниця : Планер, 2016. – 251 с.



									Аркуш
									59
Изм.	Аркуш.	№ докум.	Підпис	Дата	ДТЕУ 121 07-12.БР				

## ДОДАТКИ

### Додаток А

#### КОД РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МОВОЮ С#

Файл Form1.cs адміністраторської частини розробленого програмного забезпечення

```
using System;
using System.Data.Odbc;
using System.Windows.Forms;

namespace FitAccounting
{
    public partial class Form1 : Form
    {
        public
            String TrainerID;
        public Form1()
        {
            InitializeComponent();
            try
            {
                string MyConnectionString =
                    "DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
                    ..,098"; //Connect to MySQL using Connector/ODBC
                OdbcConnection MyConnection = new
                    OdbcConnection(MyConnectionString);
                MyConnection.Open();

                string querystr = "SELECT * FROM sections;";
                OdbcCommand MyCommand = new OdbcCommand(@querystr,
                    MyConnection);
                OdbcDataReader MyDataReader;
                MyDataReader = MyCommand.ExecuteReader();
                if (MyDataReader.HasRows)
                {
                    if (MyDataReader.HasRows)
                    {
                        int NumRes = 0;
                        while (MyDataReader.Read())
                        {
                            String sectionname, sectionid;
```

```

        sectionname =
MyDataReader.GetString(MyDataReader.GetOrdinal("SectionName"));
        sectionid =
MyDataReader.GetInt32(MyDataReader.GetOrdinal("SectionID")).ToString()
;
        comboBox4.Items.Add(sectionname + "/" +
sectionid);
        NumRes++;
    }
}
else
{
    MessageBox.Show(this, "У базі даних відсутні
секції!");
}
//Close all resources
MyDataReader.Close();
//Close all resources
MyConnection.Close();
}
catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
{
    MessageBox.Show(this, "Виникла помилка доступу до бази
даних! Її текст: " + MyOdbcException.Errors[0].Message);
    Environment.Exit(0);
}
}

private void вихідToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Close();
}

private void проПрограмуToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Form2 about = new Form2();
    about.ShowDialog();
}

private void button3_Click(object sender, EventArgs e)
{
    String login, password;
    if (textBox1.Text != "")
        login = textBox1.Text.Replace("[^\\da-zA-Za-яёА-ЯЁ]",
""");
    else
    {
        MessageBox.Show("Введіть логін!");
        return;
    }
}

```

```

    }
    if (textBox3.Text != "")
        password = textBox3.Text.Replace("[^\\da-zA-Za-яёА-
яЁ]", "");
    else
    {
        MessageBox.Show("Введіть пароль!");
        return;
    }
    try
    {
        string MyConnectionString =
        "DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
        .,098"; //Connect to MySQL using Connector/ODBC
        OdbcConnection MyConnection = new
        OdbcConnection(MyConnectionString);
        MyConnection.Open();

        string querystr = "SELECT * FROM trainers WHERE
        TrainerLogin='" + login + "' AND TrainerPassword='" + password + "';";
        OdbcCommand MyCommand = new OdbcCommand(@querystr,
        MyConnection);
        OdbcDataReader MyDataReader;
        MyDataReader = MyCommand.ExecuteReader();
        if (MyDataReader.HasRows)
        {
            groupBox1.Enabled = true;
            groupBox2.Enabled = true;
            groupBox3.Enabled = true;
            groupBox4.Enabled = true;
            groupBox5.Enabled = true;
            groupBox6.Enabled = true;
            label4.Text = "Ви авторизовані у системі
            адміністрування фітнес-клубу з логіном " + login;
            button3.Enabled = false;
            MyDataReader.Read();
            TrainerID =
            MyDataReader.GetString(MyDataReader.GetOrdinal("TrainerID"));
        }
        else
        {
            MessageBox.Show(this, "Перевірте логін та пароль!
            Вказана комбінація у базі даних відсутня!");
        }
        //Close all resources
        MyDataReader.Close();
        //Close all resources
        MyConnection.Close();
    }
    catch (OdbcException MyOdbcException) //Catch any ODBC
    exception ..
    {

```

```

        MessageBox.Show(this, "Виникла помилка доступу до бази
даних! Її текст: " + MyOdbcException.Errors[0].Message);
        Environment.Exit(0);
    }
    FillTables();
}

private void FillTables()
{
    string querystr;
    listView1.Items.Clear();
    listView2.Items.Clear();
    listView3.Items.Clear();

    try
    {
        string MyConnectionString =
"DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
.,098"; //Connect to MySQL using Connector/ODBC
        OdbcConnection MyConnection = new
OdbcConnection(MyConnectionString);
        MyConnection.Open();

        querystr = "SELECT * FROM sections;";
        OdbcCommand MyCommand = new OdbcCommand(@querystr,
MyConnection);
        OdbcDataReader MyDataReader;
        MyDataReader = MyCommand.ExecuteReader();
        if (MyDataReader.HasRows)
        {
            if (MyDataReader.HasRows)
            {
                int NumRes = 0;
                while (MyDataReader.Read())
                {
                    String sectionname, sectionid,
sectiondescription;
                    sectionname =
MyDataReader.GetString(MyDataReader.GetOrdinal("SectionName"));
                    sectiondescription =
MyDataReader.GetString(MyDataReader.GetOrdinal("SectionDescription"));
                    sectionid =
MyDataReader.GetInt32(MyDataReader.GetOrdinal("SectionID")).ToString();

                    ListViewItem lvi = new ListViewItem(new
string[] { sectionname, sectiondescription, sectionid });
                    // добавляем элемент в ListView
                    lvi.UseItemStyleForSubItems = false;
                    listView3.Items.Add(lvi);
                    NumRes++;
                }
            }
        }
    }
}

```

```

else
{
    MessageBox.Show(this, "У бази даних відсутні
секції!");
}
//Close all resources
MyDataReader.Close();
//Close all resources
MyConnection.Close();
}
catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
{
    MessageBox.Show(this, "Виникла помилка доступу до бази
даних! Її текст: " + MyOdbcException.Errors[0].Message);
    Environment.Exit(0);
}
try
{
    string MyConString =
"DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
.,098"; //Connect to MySQL using Connector/ODBC
OdbcConnection MyConnection = new
OdbcConnection(MyConString);
MyConnection.Open();

    querystr = "SELECT * FROM trainers,sections WHERE
TrainerSection=SectionID;";
    OdbcCommand MyCommand = new OdbcCommand(@querystr,
MyConnection);
    OdbcDataReader MyDataReader;
    MyDataReader = MyCommand.ExecuteReader();
    if (MyDataReader.HasRows)
    {
        if (MyDataReader.HasRows)
        {
            int NumRes = 0;
            while (MyDataReader.Read())
            {
                String trainername, trainerlastname,
trainerid, trainersection, trainerfee, trainermonthfee;
                trainername =
MyDataReader.GetString(MyDataReader.GetOrdinal("TrainerName"));
                trainerlastname =
MyDataReader.GetString(MyDataReader.GetOrdinal("TrainerLastname"));
                trainerid =
MyDataReader.GetInt32(MyDataReader.GetOrdinal("TrainerID")).ToString()
;
                trainersection =
MyDataReader.GetString(MyDataReader.GetOrdinal("SectionName"));

```

```

trainerfee =
MyDataReader.GetInt32(MyDataReader.GetOrdinal("TrainerFeePerHour")).ToString();

trainermonthfee = "";
ListViewItem lvi = new ListViewItem(new
string[] { trainerlastname, trainername, trainersection, trainerfee,
trainerid, trainermonthfee});
// добавляем элемент в ListView
lvi.UseItemStyleForSubItems = false;
listView1.Items.Add(lvi);
NumRes++;
}
}
else
{
MessageBox.Show(this, "У бази даних відсутні тренери!");
}
//Close all resources
MyDataReader.Close();
//Close all resources
MyConnection.Close();
}
catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
{
MessageBox.Show(this, "Виникла помилка доступу до бази
даних! її текст: " + MyOdbcException.Errors[0].Message);
Environment.Exit(0);
}
try
{
string MyConString =
"DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
.,098"; //Connect to MySQL using Connector/ODBC
OdbcConnection MyConnection = new
OdbcConnection(MyConString);
MyConnection.Open();

querystr = "SELECT * FROM clients;";
OdbcCommand MyCommand = new OdbcCommand(@querystr,
MyConnection);
OdbcDataReader MyDataReader;
MyDataReader = MyCommand.ExecuteReader();
if (MyDataReader.HasRows)
{
if (MyDataReader.HasRows)
{
int NumRes = 0;
while (MyDataReader.Read())
{

```

```

        String clientname, clientlastname,
clientage, clientweight, clientgender, clientdiseases, clientphone,
clientemail, clientid;
        clientname =
MyDataReader.GetString(MyDataReader.GetOrdinal("ClientName"));
        clientlastname =
MyDataReader.GetString(MyDataReader.GetOrdinal("ClientLastname"));
        clientid =
MyDataReader.GetInt32(MyDataReader.GetOrdinal("ClientID")).ToString();
        clientage =
MyDataReader.GetInt32(MyDataReader.GetOrdinal("ClientAge")).ToString();
;
        clientweight =
MyDataReader.GetInt32(MyDataReader.GetOrdinal("ClientWeight")).ToString();
        clientgender =
MyDataReader.GetInt32(MyDataReader.GetOrdinal("ClientGender")).ToString();
        clientdiseases =
MyDataReader.GetString(MyDataReader.GetOrdinal("ClientDiseases"));
        clientphone =
MyDataReader.GetString(MyDataReader.GetOrdinal("ClientPhone"));
        clientemail =
MyDataReader.GetString(MyDataReader.GetOrdinal("ClientEmail"));

        ListViewItem lvi = new ListViewItem(new
string[] { clientlastname, clientname, clientage, clientweight,
clientgender, clientdiseases, clientphone, clientemail, clientid });
        // добавляем элемент в ListView
lvi.UseItemStyleForSubItems = false;
        listView2.Items.Add(lvi);
        NumRes++;
    }
}
else
{
    MessageBox.Show(this, "У базі даних відсутні
тренери!");
}
//Close all resources
MyDataReader.Close();
//Close all resources
MyConnection.Close();
}
catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
{
    MessageBox.Show(this, "Виникла помилка доступу до бази
даних! ïï текст: " + MyOdbcException.Errors[0].Message);
    Environment.Exit(0);
}
}
}

```



```

private void listView2_DoubleClick(object sender, EventArgs e)
{
    int
    lessonid=Int32.Parse(listView2.Items[listView2.SelectedIndices[0]].Sub
    Items[4].Text);
    try
    {
        string MyConnectionString =
        "DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
        .,098"; //Connect to MySQL using Connector/ODBC
        //Connect to MySQL using Connector/ODBC
        OdbcConnection MyConnection = new
        OdbcConnection(MyConnectionString);
        MyConnection.Open();
        OdbcCommand MyCommand = new OdbcCommand("DELETE FROM
        lessons WHERE LessonID=" + lessonid + ";", MyConnection);
        MyCommand.ExecuteNonQuery();
        //Close all resources
        MyConnection.Close();
        MessageBox.Show(this, "Запис видалено!");
        FillTables();
    }
    catch (OdbcException MyOdbcException) //Catch any ODBC
    exception ..
    {
        MessageBox.Show(this, "Виникла помилка доступу до бази
        даних! Її текст: " + MyOdbcException.Errors[0].Message);
    }
}

private void button4_Click(object sender, EventArgs e)
{
    if (textBox8.Text == "")
    {
        MessageBox.Show(this, "Задайте, будь-ласка, назву
        секції!");
        return;
    }
    if (textBox7.Text == "")
    {
        MessageBox.Show(this, "Задайте, будь-ласка, опис
        секції!");
        return;
    }
    try
    {
        string name = textBox8.Text;
        string description = textBox7.Text;
        string MyConnectionString =
        "DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
        .,098"; //Connect to MySQL using Connector/ODBC

```

```

//Connect to MySQL using Connector/ODBC
OdbcConnection MyConnection = new
OdbcConnection(MyConnectionString);
MyConnection.Open();
OdbcCommand MyCommand = new OdbcCommand("INSERT INTO
sections (SectionName, SectionDescription) Values('" +
name + "', '" + description + "')", MyConnection);
MyCommand.ExecuteNonQuery();
//Close all resources
MyConnection.Close();
MessageBox.Show(this, "Запис успішно внесено!");
FillTables();
textBox7.Text = "";
textBox8.Text = "";
}
catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
{
MessageBox.Show(this, "Виникла помилка доступу до бази
даних! ïï текст: " + MyOdbcException.Errors[0].Message);
}
}
private void listView3_DoubleClick(object sender, EventArgs e)
{
int sectionid =
Int32.Parse(listView3.Items[listView3.SelectedIndices[0]].SubItems[2].
Text);
try
{
string MyConnectionString =
"DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
.,098";
//Connect to MySQL using Connector/ODBC
OdbcConnection MyConnection = new
OdbcConnection(MyConnectionString);
MyConnection.Open();
OdbcCommand MyCommand = new OdbcCommand("DELETE FROM
sections WHERE SectionID=" + sectionid + ";", MyConnection);
MyCommand.ExecuteNonQuery();
//Close all resources
MyConnection.Close();
MessageBox.Show(this, "Запис видалено!");
FillTables();
}
catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
{
MessageBox.Show(this, "Виникла помилка доступу до бази
даних! ïï текст: " + MyOdbcException.Errors[0].Message);
}
}
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    if (textBox6.Text == "")
    {
        MessageBox.Show(this, "Задайте, будь-ласка,
    прізвище!");
        return;
    }
    if (textBox2.Text == "")
    {
        MessageBox.Show(this, "Задайте, будь-ласка, ім'я!");
        return;
    }
    if (textBox11.Text == "")
    {
        MessageBox.Show(this, "Задайте, будь-ласка, оплату
    праці за годину!");
        return;
    }
    if (textBox10.Text == "")
    {
        MessageBox.Show(this, "Задайте, будь-ласка, логін!");
        return;
    }
    if (textBox9.Text == "")
    {
        MessageBox.Show(this, "Задайте, будь-ласка, пароль!");
        return;
    }
    if (comboBox4.SelectedIndex < 0)
    {
        MessageBox.Show(this, "Задайте, будь-ласка, секцію!");
        return;
    }
    try
    {
        string name = textBox2.Text;
        string lastname = textBox6.Text;
        string login = textBox10.Text;
        string password = textBox9.Text;
        string fee = textBox11.Text;
        string section =
    comboBox4.Items[comboBox4.SelectedIndex].ToString().Split('/')[1];

        string MyConnectionString =
    "DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
    .,098";
        //Connect to MySQL using Connector/ODBC
        OdbcConnection MyConnection = new
    OdbcConnection(MyConnectionString);
        MyConnection.Open();
    }
}

```

```

        OdbcCommand MyCommand = new OdbcCommand("INSERT INTO
trainers (TrainerName, TrainerLastname, TrainerLogin, TrainerPassword,
TrainerFeePerHour, TrainerSection) Values('" +
        name + "', '" + lastname + "', '" + login + "', '"
+password + "', '" + fee + "', '" + section + "');", MyConnection);
        MyCommand.ExecuteNonQuery();
        //Close all resources
        MyConnection.Close();
        MessageBox.Show(this, "Запис успішно внесено!");
        FillTables();
        textBox2.Text="";
        textBox6.Text = "";
        textBox10.Text = "";
        textBox9.Text = "";
        textBox11.Text = "";
    }
    catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
    {
        MessageBox.Show(this, "Виникла помилка доступу до бази
даних! ïï текст: " + MyOdbcException.Errors[0].Message);
    }
}

private void listView1_DoubleClick(object sender, EventArgs e)
{
    int trainerid =
Int32.Parse(listView1.Items[listView1.SelectedIndices[0]].SubItems[4].
Text);
    try
    {
        string MyConString =
"DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
..,098"; //Connect to MySQL using Connector/ODBC
        OdbcConnection MyConnection = new
OdbcConnection(MyConString);
        MyConnection.Open();
        OdbcCommand MyCommand = new OdbcCommand("DELETE FROM
trainers WHERE TrainerID=" + trainerid + ";", MyConnection);
        MyCommand.ExecuteNonQuery();
        //Close all resources
        MyConnection.Close();
        MessageBox.Show(this, "Запис видалено!");
        FillTables();
    }
    catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
    {
        MessageBox.Show(this, "Виникла помилка доступу до бази
даних! ïï текст: " + MyOdbcException.Errors[0].Message);
    }
}

```

```
}  
  
private void button2_Click(object sender, EventArgs e)  
{  
    if (textBox14.Text == "")  
    {  
        MessageBox.Show(this, "Задайте, будь-ласка, прізвище!");  
        return;  
    }  
    if (textBox13.Text == "")  
    {  
        MessageBox.Show(this, "Задайте, будь-ласка, ім'я!");  
        return;  
    }  
    if (textBox4.Text == "")  
    {  
        MessageBox.Show(this, "Задайте, будь-ласка, вік!");  
        return;  
    }  
    if (textBox18.Text == "")  
    {  
        MessageBox.Show(this, "Задайте, будь-ласка, вагу!");  
        return;  
    }  
    if (textBox19.Text == "")  
    {  
        MessageBox.Show(this, "Задайте, будь-ласка, стать!");  
        return;  
    }  
    if (textBox12.Text == "")  
    {  
        MessageBox.Show(this, "Задайте, будь-ласка, логін!");  
        return;  
    }  
    if (textBox5.Text == "")  
    {  
        MessageBox.Show(this, "Задайте, будь-ласка, пароль!");  
        return;  
    }  
    if (textBox17.Text == "")  
    {  
        MessageBox.Show(this, "Задайте, будь-ласка, телефон!");  
        return;  
    }  
    if (textBox16.Text == "")  
    {  
        MessageBox.Show(this, "Задайте, будь-ласка, електронну  
пошту!");  
        return;  
    }  
    if (textBox15.Text == "")  
    {
```

```

        MessageBox.Show(this, "Задайте, будь-ласка, наявні у користувача хвороби (або напишіть слово немає)!");
        return;
    }

    try
    {
        string name = textBox13.Text;
        string lastname = textBox14.Text;
        string login = textBox12.Text;
        string password = textBox5.Text;
        string age = textBox4.Text;
        string weight = textBox18.Text;
        string gender = textBox19.Text;
        string phone = textBox17.Text;
        string email = textBox16.Text;
        string diseases = textBox15.Text;

        string MyConnectionString =
        "DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
        .,098"; //Connect to MySQL using Connector/ODBC
        //Connect to MySQL using Connector/ODBC
        OdbcConnection MyConnection = new
        OdbcConnection(MyConnectionString);
        MyConnection.Open();
        OdbcCommand MyCommand = new OdbcCommand("INSERT INTO
        clients (ClientName, ClientLastname, ClientLogin, ClientPassword,
        ClientAge, ClientWeight, ClientGender, ClientPhone, ClientEmail,
        ClientDiseases) Values('" +
            name + "', '" + lastname + "', '" + login + "', '" +
            password + "', '" + age + "', '" + weight + "', '" + gender + "', '" + phone +
            "', '" + email + "', '" + diseases + "')");
        MyCommand.ExecuteNonQuery();
        //Close all resources
        MyConnection.Close();
        MessageBox.Show(this, "Запис успішно внесено!");
        FillTables();
        textBox13.Text = "";
        textBox14.Text = "";
        textBox12.Text = "";
        textBox5.Text = "";
        textBox4.Text = "";
        textBox18.Text = "";
        textBox19.Text = "";
        textBox17.Text = "";
        textBox16.Text = "";
        textBox15.Text = "";
    }
    catch (OdbcException MyOdbcException) //Catch any ODBC
    exception ..
    {
        MessageBox.Show(this, "Виникла помилка доступу до бази
        даних! Її текст: " + MyOdbcException.Errors[0].Message);
    }
}

```

```

    }
}

private void listView2_DoubleClick_1(object sender, EventArgs e)
{
    int clientid =
    Int32.Parse(listView2.Items[listView2.SelectedIndices[0]].SubItems[8].
    Text);
    try
    {
        string MyConnectionString =
        "DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
        .,098"; //Connect to MySQL using Connector/ODBC
        OdbcConnection MyConnection = new
        OdbcConnection(MyConnectionString);
        MyConnection.Open();
        OdbcCommand MyCommand = new OdbcCommand("DELETE FROM
        clients WHERE ClientID=" + clientid + ";", MyConnection);
        MyCommand.ExecuteNonQuery();
        //Close all resources
        MyConnection.Close();
        MessageBox.Show(this, "Запис видалено!");
        FillTables();
    }
    catch (OdbcException MyOdbcException) //Catch any ODBC
    exception ..
    {
        MessageBox.Show(this, "Виникла помилка доступу до бази
        даних! Ії текст: " + MyOdbcException.Errors[0].Message);
    }
}

private void button5_Click(object sender, EventArgs e)
{
    string querystr, year, month;
    if (textBox20.Text.Length != 2)
    {
        MessageBox.Show(this, "Формат місяця має бути: 01, 02,
        03, ..., 11, 12");
        return;
    }
    if (textBox21.Text.Length != 4)
    {
        MessageBox.Show(this, "Формат року має бути: 2022, 2023,
        і т.д.");
        return;
    }
    year = textBox21.Text;
    month = textBox20.Text;
    try
    {
        int[] trainerslessons = new int[1000];

```

```

        for (int i = 0; i < 1000; i++)
            trainerslessons[i] = 0;
        int trainerid, fee;
        string MyConnectionString =
        "DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
        .,098"; //Connect to MySQL using Connector/ODBC
        OdbcConnection MyConnection = new
        OdbcConnection(MyConnectionString);
        MyConnection.Open();

        querystr = "SELECT LessonTrainer FROM lessons WHERE
        LessonDate LIKE '"+year+"-"+month+"-__'";
        OdbcCommand MyCommand = new OdbcCommand(@querystr,
        MyConnection);
        OdbcDataReader MyDataReader;
        MyDataReader = MyCommand.ExecuteReader();
        if (MyDataReader.HasRows)
        {
            int NumRes = 0;
            while (MyDataReader.Read())
            {
                trainerid =
                MyDataReader.GetInt32(MyDataReader.GetOrdinal("LessonTrainer"));
                trainerslessons[trainerid]++;
                NumRes++;
            }

            for (int i = 0; i < listView1.Items.Count; i++)
            {
                fee =
                Int32.Parse(listView1.Items[i].SubItems[3].Text);
                fee =
                trainerslessons[Int32.Parse(listView1.Items[i].SubItems[4].Text)] *
                fee;
                listView1.Items[i].SubItems[5].Text =
                fee.ToString();
            }
        }
        else
        {
            MessageBox.Show(this, "У базі даних взагалі відсутні
            заняття за цей місяць!");
            for (int i = 0; i < listView1.Items.Count; i++)
            {
                listView1.Items[i].SubItems[5].Text = "";
            }
        }
        //Close all resources
        MyDataReader.Close();
        //Close all resources
        MyConnection.Close();
    }
}

```



```
catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
{
    MessageBox.Show(this, "Виникла помилка доступу до бази
даних! Її текст: " + MyOdbcException.Errors[0].Message);
    Environment.Exit(0);
}
}
```



## Файл Form1.cs клієнтської частини розробленого програмного забезпечення

```

using System;
using System.Data.Odbc;
using System.Windows.Forms;

namespace FitAccounting
{
    public partial class Form1 : Form
    {
        public
        String ClientID;
        public Form1()
        {
            InitializeComponent();

            try
            {
                string MyConnectionString =
                "DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
                .,098"; //Connect to MySQL using Connector/ODBC
                OdbcConnection MyConnection = new
                OdbcConnection(MyConnectionString);
                MyConnection.Open();

                string querystr = "SELECT * FROM sections;";
                OdbcCommand MyCommand = new OdbcCommand(@querystr,
                MyConnection);
                OdbcDataReader MyDataReader;
                MyDataReader = MyCommand.ExecuteReader();
                if (MyDataReader.HasRows)
                {
                    if (MyDataReader.HasRows)
                    {
                        int NumRes = 0;
                        while (MyDataReader.Read())
                        {
                            String sectionname, sectionid;
                            sectionname
                            =
                            MyDataReader.GetString(MyDataReader.GetOrdinal("SectionName"));
                            sectionid
                            =
                            MyDataReader.GetInt32(MyDataReader.GetOrdinal("SectionID")).ToString();
                            ;
                            comboBox1.Items.Add(sectionname+"/"+sectionid);
                            NumRes++;
                        }
                    }
                }
            }
        }
    }
}

```

```

else
{
    MessageBox.Show(this, "У бази даних відсутні
секції!");
}
//Close all resources
MyDataReader.Close();
//Close all resources
MyConnection.Close();
}
catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
{
    MessageBox.Show(this, "Виникла помилка доступу до бази
даних! Її текст: " + MyOdbcException.Errors[0].Message);
    Environment.Exit(0);
}
}

private void вихідToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Close();
}

private void проПрограмуToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Form2 about = new Form2();
    about.ShowDialog();
}

private void button3_Click_1(object sender, EventArgs e)
{
    String login, password;
    if (textBox1.Text != "")
        login = textBox1.Text.Replace("[^\\da-zA-Za-яёА-ЯЁ]",
"");
    else
    {
        MessageBox.Show("Введіть логін!");
        return;
    }
    if (textBox3.Text != "")
        password = textBox3.Text.Replace("[^\\da-zA-Za-яёА-
ЯЁ]", "");
    else
    {
        MessageBox.Show("Введіть пароль!");
        return;
    }
    try
    {

```

```

        string MyConnectionString =
"DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
.,098"; //Connect to MySQL using Connector/ODBC
        OdbcConnection MyConnection = new
OdbcConnection(MyConnectionString);
        MyConnection.Open();

        string querystr = "SELECT * FROM clients WHERE
ClientLogin='" + login + "' AND ClientPassword='" + password + "';";
        OdbcCommand MyCommand = new OdbcCommand(@querystr,
MyConnection);
        OdbcDataReader MyDataReader;
        MyDataReader = MyCommand.ExecuteReader();
        if (MyDataReader.HasRows)
        {
            groupBox1.Enabled = true;
            groupBox2.Enabled = true;
            label4.Text = "Ви авторизовані у системі фітнес-
клубу з логіном " + login;
            button3.Enabled = false;
            MyDataReader.Read();
            ClientID =
MyDataReader.GetString(MyDataReader.GetOrdinal("ClientID"));
        }
        else
        {
            MessageBox.Show(this, "Перевірте логін та пароль!
Вказана комбінація у базі даних відсутня!");
        }
        //Close all resources
        MyDataReader.Close();
        //Close all resources
        MyConnection.Close();
    }
    catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
    {
        MessageBox.Show(this, "Виникла помилка доступу до бази
даних! ïï текст: " + MyOdbcException.Errors[0].Message);
        Environment.Exit(0);
    }
    FillTable();
}

private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
{
    comboBox2.Items.Clear();
    int SectionID =
Int32.Parse(comboBox1.Items[comboBox1.SelectedIndex].ToString().Split(
'/')[1]);
    try
    {

```

```

        string MyConnectionString =
"DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
.,098"; //Connect to MySQL using Connector/ODBC
        OdbcConnection MyConnection = new
OdbcConnection(MyConnectionString);
        MyConnection.Open();

        string querystr = "SELECT * FROM trainers WHERE
TrainerSection='"+SectionID+"'";
        OdbcCommand MyCommand = new OdbcCommand(@querystr,
MyConnection);
        OdbcDataReader MyDataReader;
        MyDataReader = MyCommand.ExecuteReader();
        if (MyDataReader.HasRows)
        {
            if (MyDataReader.HasRows)
            {
                int NumRes = 0;
                while (MyDataReader.Read())
                {
                    String teachername, teacherid;
                    teachername =
MyDataReader.GetString(MyDataReader.GetOrdinal("TrainerName"))+"
"+MyDataReader.GetString(MyDataReader.GetOrdinal("TrainerLastname"));
                    teacherid =
MyDataReader.GetInt32(MyDataReader.GetOrdinal("TrainerID")).ToString()
;
                    comboBox2.Items.Add(teachername + "/" +
teacherid);
                    NumRes++;
                }
            }
            else
            {
                MessageBox.Show(this, "У базі даних відсутні
секції!");
            }
            //Close all resources
            MyDataReader.Close();
            //Close all resources
            MyConnection.Close();
        }
        catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
        {
            MessageBox.Show(this, "Виникла помилка доступу до бази
даних! Її текст: " + MyOdbcException.Errors[0].Message);
            Environment.Exit(0);
        }
    }

    private void FillTable()

```

```

    {
        listView2.Items.Clear();
        string querystr;
        try
        {
            string MyConnectionString =
"DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
.,098"; //Connect to MySQL using Connector/ODBC
            OdbcConnection MyConnection = new
OdbcConnection(MyConnectionString);
            MyConnection.Open();
            querystr = "SELECT
SectionName,TrainerName,TrainerLastname,LessonID,LessonDate,LessonTime
FROM lessons,sections,trainers WHERE LessonClient='" + ClientID + "' AND
lessons.LessonSection=sections.SectionID AND
lessons.LessonTrainer=trainers.TrainerID ORDER BY LessonDate,
LessonTime;";
            OdbcCommand MyCommand1 = new OdbcCommand(@querystr,
MyConnection);
            OdbcDataReader MyDataReader1;
            MyDataReader1 = MyCommand1.ExecuteReader();
            if (MyDataReader1.HasRows)
            {
                int NumRes = 0;
                while (MyDataReader1.Read())
                {
                    String section, trainer, date, time,
lessonid;
                    section
MyDataReader1.GetString(MyDataReader1.GetOrdinal("SectionName"));
                    trainer
MyDataReader1.GetString(MyDataReader1.GetOrdinal("TrainerLastname")) +
" " + MyDataReader1.GetString(MyDataReader1.GetOrdinal("TrainerName"));
                    date
MyDataReader1.GetString(MyDataReader1.GetOrdinal("LessonDate"));
                    time
MyDataReader1.GetString(MyDataReader1.GetOrdinal("LessonTime"));
                    lessonid
MyDataReader1.GetInt32(MyDataReader1.GetOrdinal("LessonID")).ToString(
);
                    ListViewItem lvi = new ListViewItem(new
string[] { date, time, section, trainer, lessonid });
                    // добавляем элемент в ListView
                    lvi.UseItemStyleForSubItems = false;
                    listView2.Items.Add(lvi);
                    NumRes++;
                }
            }
            //Close all resources
            MyDataReader1.Close();
            //Close all resources
            MyConnection.Close();
        }
    }

```

```

catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
{
    MessageBox.Show(this, "Виникла помилка доступу до бази
даних! Її текст: " + MyOdbcException.Errors[0].Message);
    Environment.Exit(0);
}
}

private void button2_Click_1(object sender, EventArgs e)
{
    if(comboBox1.SelectedIndex<0)
    {
        MessageBox.Show(this, "Задайте, будь-ласка, секцію!");
        return;
    }
    if (comboBox2.SelectedIndex<0)
    {
        MessageBox.Show(this, "Задайте, будь-ласка, тренера!");
        return;
    }
    string[] dates = textBox4.Text.Split('-');
    if (dates.Length!=3 || dates[0].Length!=4 ||
dates[1].Length!=2 || dates[2].Length != 2)
    {
        MessageBox.Show(this, "Формат дати має бути YYYY-MM-DD,
наприклад 2022-01-31");
        return;
    }
    string[] times = textBox5.Text.Split(':');
    if (times.Length != 2 || times[0].Length != 2 ||
times[1].Length != 2)
    {
        MessageBox.Show(this, "Формат часу має бути HH:MM,
наприклад 14:00");
        return;
    }
    try
    {
        string trainer =
comboBox2.Items[comboBox2.SelectedIndex].ToString().Split('/')[1];
        string section =
comboBox1.Items[comboBox1.SelectedIndex].ToString().Split('/')[1];
        string date = textBox4.Text;
        string time = textBox5.Text;
        string MyConnectionString =
"DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxc123/
.,098";
        //Connect to MySQL using Connector/ODBC
        //Connect to MySQL using Connector/ODBC
        OdbcConnection MyConnection = new
OdbcConnection(MyConnectionString);
        MyConnection.Open();
    }
}

```

```

        OdbcCommand MyCommand = new OdbcCommand("INSERT INTO
lessons (LessonTrainer, LessonClient, LessonSection,
LessonDate,LessonTime) Values ('" +
trainer + "', '" + ClientID + "', '" + section + "',
'" + date + "', '" + time + "')"); MyConnection);
MyCommand.ExecuteNonQuery();
//Close all resources
MyConnection.Close();
MessageBox.Show(this, "Запис успішно внесено!");
FillTable();
}
catch (OdbcException MyOdbcException) //Catch any ODBC
exception ..
{
    MessageBox.Show(this, "Виникла помилка доступу до бази
даних! ïï текст: " + MyOdbcException.Errors[0].Message);
}
}

private void listView2_DoubleClick(object sender, EventArgs e)
{
    int
lessonid=Int32.Parse(listView2.Items[listView2.SelectedIndices[0]].Sub
Items[4].Text);
    try
    {
        string MyConString =
"DSN=MySQL8;Server=localhost;Database=fitnessclub;uid=root;pwd=zxcl23/
.,098"; //Connect to MySQL using Connector/ODBC
//Connect to MySQL using Connector/ODBC
OdbcConnection MyConnection = new
OdbcConnection(MyConString);
MyConnection.Open();
OdbcCommand MyCommand = new OdbcCommand("DELETE FROM
lessons WHERE LessonID=" + lessonid + " "); MyConnection);
MyCommand.ExecuteNonQuery();
//Close all resources
MyConnection.Close();
}
}
}

```