

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кваліфікаційна робота

перший (бакалаврський)

(рівень вищої освіти)

на тему **Інтелектуальна інформаційна система туристичного**
комплексу

Виконав: студент 5 курсу, групи ПЗ-18-1бз
спеціальності 121 Інженерія програмного
забезпечення

(код і назва спеціальності)

освітньої програми Інженерія програмного
забезпечення

(код і назва освітньої програми)

О. С. Обідін

(ініціали та прізвище)

Керівник доцент, А.І. Безверхий
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Рецензент директор ТОВ «Дісітел»

П.О. Лютий

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

Кафедра електроніки, інформаційних систем та програмного забезпечення

Рівень вищої освіти _____ перший (бакалавський) _____

Спеціальність _____ 121 Інженерія програмного забезпечення _____
(код та назва)

Освітня програма _____ Інженерія програмного забезпечення _____
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ Т. В. Критська
“ 01 ” березня 2023 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

_____ Обідіна Олександра Сергійовича _____

(прізвище, ім'я, по батькові)

1. Тема роботи _____ Інтелектуальна інформаційна система туристичного
комплексу _____

керівник роботи _____ Безверхий Анатолій Ігорович, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від _____ 1894-с від 29.12.2022 _____

2. Строк подання студентом кваліфікаційної роботи _____ 15.05.2023 _____

3. Вихідні дані магістерської роботи

- комплект нормативних документів ;
- технічне завдання до роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- огляд та збір літератури стосовно теми кваліфікаційної роботи;
- огляд та аналіз існуючих рішень та аналогів;
- дослідження проблеми розпізнавання мов та розробка методів її вирішення;
- створення програмного продукту та його опис;
- перелік вимог для роботи програми;
- тестування програмної системи та розробка висновків та пропозицій.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
_____ слайдів презентації _____

6. Консультанти розділів бакалаврської роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата |
|--------|---|------------------|
| | | Завдання прийняв |
| | | |

7. Дата видачі завдання 01.03.2023

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів магістерської роботи | Строк виконання етапів магістерської роботи | Примітка |
|-------|--|---|----------|
| 1 | Аналіз предметної області | 01.03-10.03.23 | виконано |
| 2 | Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником | 11.03-12.03.23 | виконано |
| 3 | Аналіз існуючих методів рішення | 13.03-14.03.23 | виконано |
| 4 | Дослідження засобів виявлення об'єктів на зображеннях | 15.03-20.03.23 | виконано |
| 5 | Дослідження засобів виявлення контурів об'єктів та трансформації зображень | 21.03-26.03.23 | виконано |
| 6 | Узгодження подальших дій з науковим керівником | 27.03-28.03.23 | виконано |
| 7 | Збір тренувального набору для нейронної мережі з відкритих джерел та його попередня підготовка | 29.03-13.04.23 | виконано |
| 8 | Навчання моделі нейронної мережі за допомогою підготовленого тренувального набору | 14.04-16.04.23 | виконано |
| 9 | Представлення отриманих результатів науковому керівнику та узгодження плану подальшого дослідження | 17.04-19.04.23 | виконано |
| 10 | Реалізація функціоналу для виявлення контурів об'єктів на зображенні та трансформації зображень | 20.04-01.05.23 | виконано |
| 11 | Порівняння вихідних характеристик роботи системи за допомогою вхідних тестових відео | 02.05-7.05.23 | виконано |
| 12 | Реалізація користувацького інтерфейсу для комп'ютерної системи | 8.05-12.05.23 | виконано |
| 13 | Оформлення звіту | 13.05-20.05.23 | виконано |

Студент _____ О.С. Обідін
(підпис) (прізвище та ініціали)

Керівник роботи _____ А.І. Безверхий
(підпис) (прізвище та ініціали)

Нормоконтроль пройдено
Нормоконтролер _____ І.А. Скрипник
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Сторінок – 70,

Рисунків – 28,

Джерел – 16.

Обідін О. С. Інтелектуальна інформаційна система туристичного комплексу: кваліфікаційна робота бакалавра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник А. І. Безверхий. Запоріжжя : ЗНУ, 2023. 68 с.

У сучасних умовах управління, побудоване на ефективній взаємодії підрозділів, є цінним ресурсом організації. Отже, підвищення ефективності взаємодії стає одним із напрямків удосконалення діяльності будь-якого підприємства загалом. Найбільш очевидним способом підвищення ефективності перебігу інформаційної взаємодії є його автоматизація.

Мета і завдання дослідження полягають у вивченні особливостей ведення діяльності туристичних організацій, практичних рекомендацій щодо застосування сучасних інформаційних технологій в управлінні підприємствами туристичного бізнесу та аналізі існуючих програмних застосунків. Показано, що автоматизація процесу роботи туристичного комплексу вважається дуже важливим та перспективним процесом.

Проведено порівняльний аналіз існуючих підходів та технологій до розробки складних інформаційних систем. На основі проведеного аналізу розроблено інтелектуальну інформаційну систему керування туристичним комплексом з використанням сучасних методологій. Створений власний застосунок може бути використаний для задоволення потреб туристичних комплексів з метою підвищення ефективності їх роботи.

Ключові слова: *інформаційна система, туристична галузь, MS SQL Server.*

ABSTRACT

Pages – 70,

Drawings – 28,

Sources - 16.

Obidin O. S. Intellectual information system of the tourist complex: qualifying thesis of the bachelor of specialty 121 "Software engineering" / Science. manager A. I. Bezverkhy. Zaporizhzhia: ZNU, 2023. 68 p.

In modern conditions, management built on the effective interaction of units is a valuable resource of the organization. Therefore, increasing the efficiency of interaction becomes one of the directions for improving the activity of any enterprise in general. The most obvious way to increase the efficiency of information interaction is its automation.

The purpose and task of the research is to study the specifics of conducting the activities of tourist organizations, practical recommendations for the use of modern information technologies in the management of tourism business enterprises, and the analysis of existing programmatic relations. It is shown that the automation of the operation process of the tourist complex is considered a very important and promising process.

A comparative analysis of existing approaches and technologies to the development of complex information systems was conducted. Based on the analysis, an intelligent information system for managing the tourist complex was developed using modern methodologies. The created own application can be used to meet the needs of tourist complexes in order to increase the efficiency of their work.

Keywords: *information system, tourism industry, MS SQL Server.*

ЗМІСТ

| | |
|--|----|
| ВСТУП | 7 |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ..... | 12 |
| 1.1 Огляд літературних джерел | 12 |
| 1.2 Аналіз програмних продуктів-аналогів | 17 |
| 1.3 Постановка завдання | 21 |
| 2 ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ..... | 22 |
| 2.1 Аналіз сучасних систем управління базами даних | 22 |
| 2.2 Аналіз розподілених СУБД..... | 27 |
| 2.3 Технологія для доступу до баз даних Entity Framework Core | 33 |
| 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ КЕРУВАННЯ ТУРИСТИЧНИМ КОМПЛЕКСОМ | 35 |
| 3.1 Опис предметної області..... | 35 |
| 3.2 Архітектура системи..... | 36 |
| 3.3 Функціональні вимоги системи..... | 37 |
| 3.4 Проєктування інформаційної системи..... | 38 |
| 3.5 Програмна реалізація застосунку | 50 |
| 3.6 Розробка інтерфейсу інформаційної системи | 63 |
| ВИСНОВКИ..... | 68 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 69 |

ВСТУП

Актуальність теми

Сьогодні у світовому співтоваристві відбуваються глобальні зміни, викликані проникненням у всі сфери життя інформаційних технологій. Не залишився осторонь цієї тенденції також і туризм, як сфера економіки, діяльності та зайнятості людей. Туристична індустрія в даний час широко використовує всі види та форми інформаційного забезпечення своєї діяльності. У туристичній сфері інформаційні потоки забезпечують взаємодію між виробниками послуг, а також зв'язок між споживачами та виробниками даних послуг, при цьому вони здійснюються як у формі потоків даних, а також виступають у вигляді послуг і платежів. Туризм в сучасному світі є глобальним комп'ютеризованим бізнесом, учасниками якого є як туроператори та турагенти, так і авіакомпанії, екскурсійні організації, ресторани та готелі всього світу. Даний сектор економіки включений у провідні галузі економіки з основною часткою реалізації послуг через Інтернет.

Інформаційні технології, проникаючи в бізнес, змінюють технології управління, допомагають отримувати відповіді на будь-які питання про стан справ та приймати оперативні рішення за лічені секунди. Інформатизація – одна з рис системи громадських відносин розвинених країн. Людство досягло такого етапу розвитку цивілізації, у якому інформація (знання) відіграє визначальну роль у всіх сферах діяльності людей. Отже, інформація стає у суспільстві найважливішим чинником економічного зростання.

Оскільки інформаційні технології здатні забезпечити мінімальні витрати при обробці, передачі та зберіганні інформації, виробнича область стає більш рентабельною. Крім цього інформаційні технології, що безперервно прогресують, регулярно пропонують нові товари, наприклад, програмне забезпечення чи допоміжні процеси. За рахунок цього забезпечується як працевлаштування населення так і скорочення вартості допоміжних товарів.

В основі вирішення багатьох завдань лежить обробка інформації. Для спрощення процесу обробки інформації створюються інформаційні системи. Інформаційна система є сукупністю інформації, що міститься в базах даних, інформаційних технологій та технічних засобів, що забезпечують її обробку, це програмний комплекс, функції якого полягають у підтримці надійного зберігання інформації в пам'яті комп'ютера, виконанні специфічних для даної програми перетворень інформації та обчислень, надання користувачам зручного і легко освоюваного інтерфейсу. Зазвичай обсяги інформації, з якими доводиться мати справу з такими системами, досить великі, а сама інформація має досить складну структуру. Класичними прикладами інформаційних систем є банківські та бухгалтерські системи, системи авіаційних чи залізничних квитків, системи податкової служби, статистичні системи, системи резервування місць у готелі тощо.

Залежно від специфіки предметної області, інформаційні системи можуть дуже відрізнятися одна від одної за своїми функціями, архітектурою, реалізацією, але все-таки можна виділити деякі загальні властивості, що характерні для інформаційних систем:

- інформаційні системи призначені для збирання, зберігання та обробки інформації, тому в основі будь-якої з них лежить середовище зберігання та доступу до даних;
- інформаційні системи орієнтуються на кінцевого користувача, що не має високої кваліфікації в галузі застосування обчислювальної техніки, тому клієнтські додатки повинні мати простий, зручний інтерфейс, який надає можливість кінцевому користувачеві виконувати всі необхідні для роботи функції, але в той же час не дає йому виконувати зайві дії.

При розробці будь-якої інформаційної системи доводиться вирішувати наступні задачі:

- завдання розробки бази даних, призначеної для зберігання інформації;
- завдання розробки графічного інтерфейсу користувача.

Туризм є сферою зростаючого застосування інформаційних технологій. Сучасні пакети офісних програм дозволяють туристичному підприємству більш ефективно виконувати рутинні функції, пов'язані з документообігом підприємства. Також, слід зазначити, що основою застосування програмного забезпечення у туристському бізнесі є використання баз даних [2]. У зв'язку з цим, в даний час, аналіз існуючих інформаційних систем у туризмі, вивчення основних галузей застосування інформаційних технологій та розробка інтелектуальної інформаційної системи керування туристичним комплексом для підвищення ефективності його діяльності є актуальною задачею.

Мета дослідження

Розробка інтелектуальної інформаційної системи керування туристичним комплексом з використанням сучасних методологій.

Завдання дослідження

Розглянути особливості та існуючі рішення (аналоги) для ведення діяльності туристичних організацій. Проаналізувати існуючі підходи та технології до розробки складних інформаційних систем та вибрати відповідну для реалізації інтелектуальної інформаційної системи керування туристичним комплексом.

Об'єкт дослідження

Об'єктом дослідження є процес розробки інтелектуальної інформаційної системи туристичного комплексу.

Предмет дослідження

Предметом дослідження є технології створення власного застосунку для задоволення потреб туристичних комплексів з метою підвищення ефективності їх роботи.

Методи дослідження

Теоретичні – аналіз і порівняння технологій створення інформаційних систем; види класифікації та принципів проектування інформаційних систем; порівняльний аналіз для вибору програмного забезпечення.

Практичне значення одержаних результатів

Практичне значення одержаних результатів дослідження полягає у тому, що розробка одного застосунку замість використання великої кількості платних додатків підвищує ефективність діяльності туристичного комплексу. Результати дослідження можуть бути використані для створення туристичної інформаційної системи в регіоні з метою просування місцевих турпродуктів на міжнародний ринок.

Глосарій

Туристичний напрям – напрям міграційного руху населення в оздоровчих, пізнавальних, професійно-ділових чи інших цілях без здійснення оплачуваної діяльності в місці, куди особа від'їжджає.

Туристичний об'єкт – місце, що приваблює туристів завдяки своїм природним і штучно створеним рисам (природа, художні та археологічні цінності, кліматичні умови тощо).

Програми-гиди (англ. automatic tour guide system) – це підклас систем супроводу подорожі.

Ігрові туристичні системи – це системи, які формують завдання користувачеві, які необхідно розв'язати в процесі подорожі, або під час ознайомлення з певним туристичним об'єктом.

Туристичні довідкові інформаційні системи – це системи, які надають користувачеві детальну вичерпну інформацію за різними туристичними напрямками.

Інтерфейс – це сукупність засобів, методів та правил, призначених для взаємодії елементів системи (або цілих систем) між собою.

Архітектура – це базова організація системи, втілена в її компонентах, їхніх відносинах між собою і з оточенням, а також принципи, що визначають проектування та розвиток системи програми, видимі ззовні властивості цих елементів і зв'язки між ними.

Мова – це штучна мова, створена для передачі команд машинам, зокрема комп'ютерам.

Прив'язка даних – процес приєднання даних в одному об'єкті до іншого об'єкту. Він забезпечує зручний шлях передачі даних між різними рівнями застосунку.

Атракція – у сучасній туристичній галузі використовується як симбіоз понять «місце», «краєвид», «пам'ятка культури/архітектури/історії» та принагідна «діяльність», якою турист може займатися у цьому місці.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд літературних джерел

Питанням огляду та аналізу новітніх інформаційних технологій в сфері туризму присвячено достатню кількість наукових праць [1-5]. Зокрема авторами публікації [1] наводяться найважливіші характеристики досліджень провідних спеціалістів галузі електронного туризму. Підкреслюється, що наявні нині інформаційні технології з одного боку зорієнтовані на туристів, а з іншого, не надають їм належної інформаційної підтримки на всіх етапах подорожі, практично не враховують індивідуальні особливості кожного з членів туристичної групи. Результати проведеного аналізу інформаційних джерел підкреслюють, що основною тенденцією розробки сучасних ІТ для потреб туриста є використання мобільних інформаційних технологій. Це забезпечить доступ користувача-туриста до інформації в будь-який момент часу та влюбій точці земної кулі.

Розробленню теоретичних і методологічних засад та практичних рекомендацій щодо застосування сучасних інформаційних технологій в управлінні підприємствами туристичного бізнесу присвячена праця [3]. Автори підкреслюють про особливість створення та реалізації туристичних послуг, що пов'язана з великою кількістю учасників, їх географічним розташуванням, віддаленістю місця реалізації туристичних послуг від місця їх споживання. Зроблено акцент на тому, що рівень використання сучасних інформаційних технологій у туристичній сфері залишається недостатнім, це пов'язано з тим, що розвиток інформаційно-комунікаційної інфраструктури у цій сфері залишається низьким, відсутня база даних туристичного профілю в деяких регіонах нашої країни; обмежена інформація та реклама туристичного продукту, залишається недосконалим механізм інформаційного обміну у сфері туризму за допомогою мережі Інтернет, є низьким рівнем розвитку віртуальних туристичних підприємств і центрів.

З метою виявлення напрямків покращення розвитку туристичних підприємств авторами [4] проаналізовано статистичні дані щодо використання інформаційних технологій в Україні. Для аналізу використовувались ресурси, що є найпопулярнішими за версією дані SimilarWeb Ltd, компанії, яка використовує технології обробки великих даних для збору, вимірювання, аналізу та надання даних про поведінкові моделі та статистику залученості користувачів веб-сайтів і мобільних додатків. Весь їх спектр, що використовується в туристичній галузі, поділено на функціональні групи інформаційних систем. Проведений статистичний аналіз найпопулярніших веб-сайтів та мобільних додатків в Україні, що мають відношення до надання туристичних послуг, дозволив поділити їх на чотири категорії: «розміщення», «подорожі повітрям», «подорожі наземним транспортом» та «туристичні пам'ятки». Серед веб-ресурсів першої з них можна виділити такі, як: Booking.com та Doba.ua (див. Рис.1), що є системами інтернет-бронювання, Otpusk.com та Turpravda.ua (див. Рис.2) є веб-сайтами, функціонал яких дозволяє переглянути відгуки та фото готелів, а веб-сайт Rixos.com є міжнародною мережею п'ятизіркових готелів.

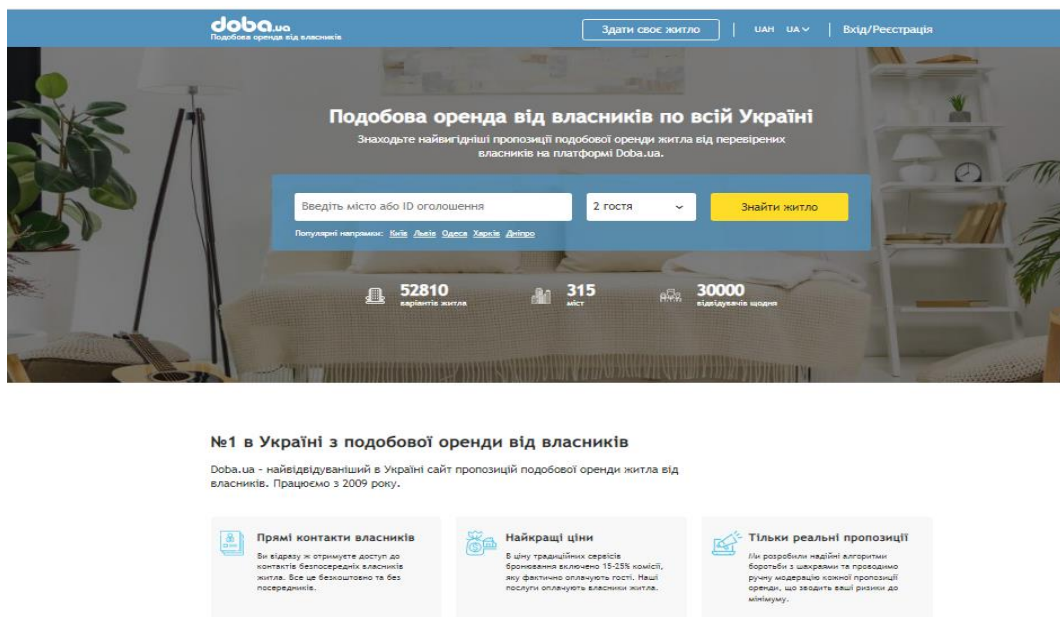


Рис. 1 Інтерфейс системи інтернет-бронювання Doba.ua

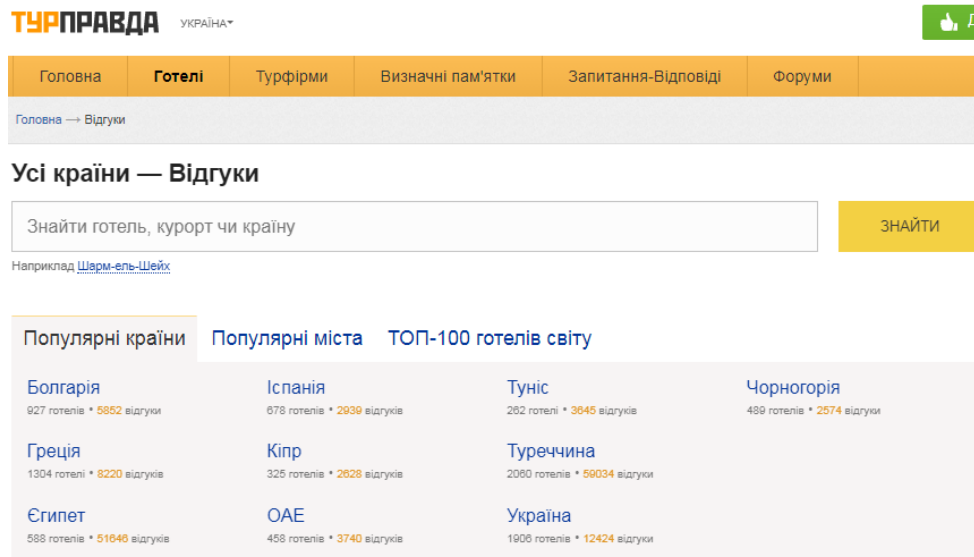


Рис. 2 Сторінка відгуків веб-сайта *Turpravda.ua*

До другої категорії серед популярних вітчизняних сайтів відносяться такі, як: *flyua.com*, *wizzair.com* (див. Рис.3), *ryanair.com*, *kbr.aero* (див. Рис.4), що є офіційними сайтами авіакомпаній, та аеропорту «Бориспіль». Також виділяють ресурс *avianews.com* який містить актуальні новини про рейси, розпродажі квитків, інформацією про аеропорти та авіакомпанії.

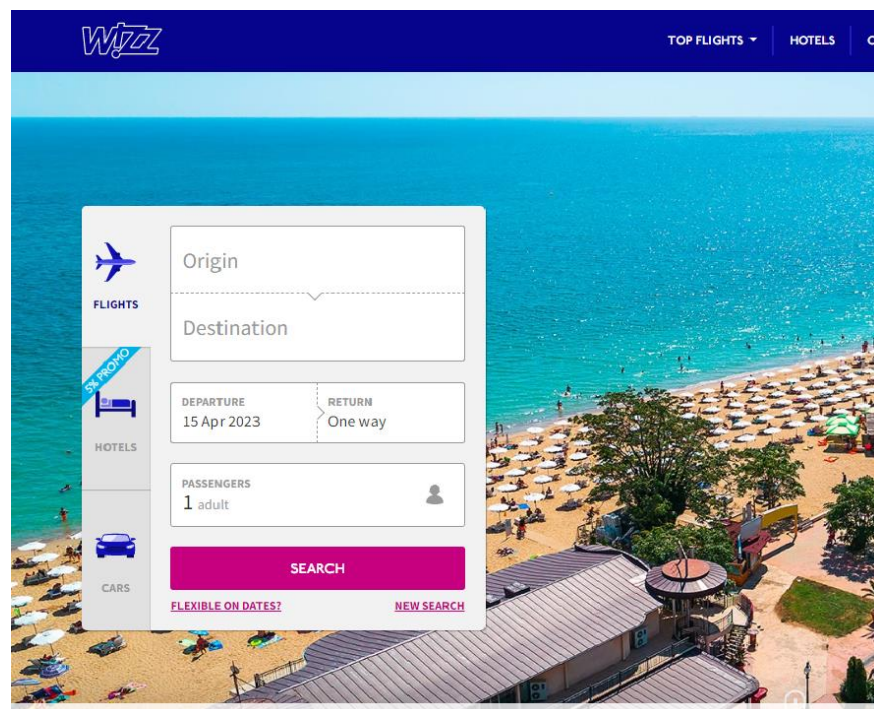


Рис.3 Сторінка пошуку веб-сайта *wizzair.com*

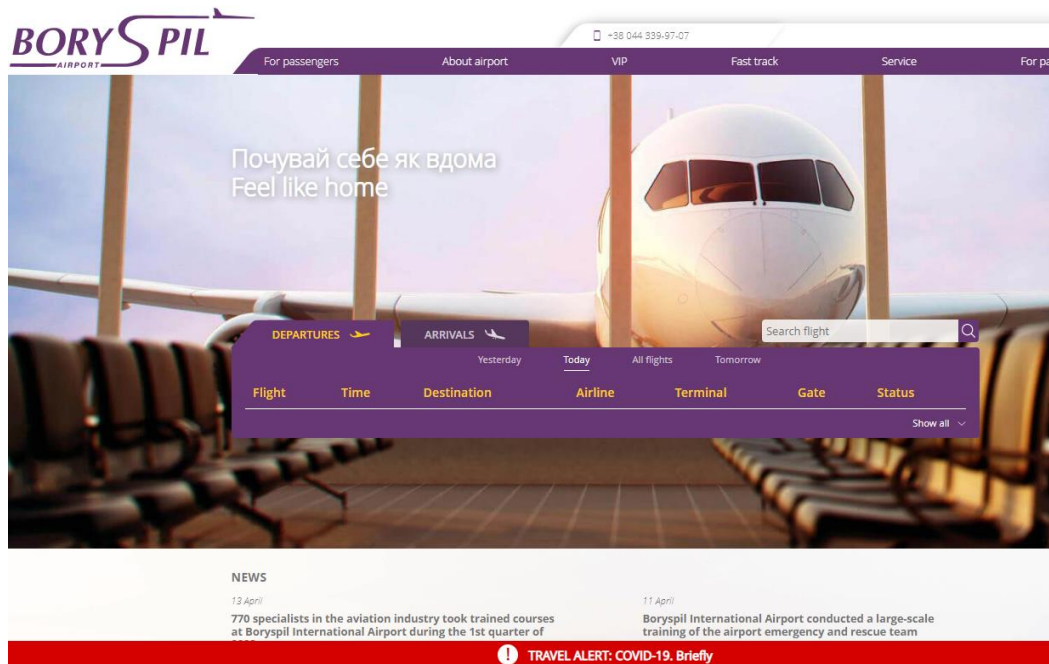


Рис.4 Головна сторінка сайту аеропорту «Бориспіль»

Однак у порівнянні з міжнародними аналогами, відзначено рід недоліків вказаних ресурсів, серед яких: неможливість користування електронним квитком, неможливість зберегти дані карти лояльності, технічні несправності у меню додатку та ін. Значна увага приділена характеристикам третьої категорії, оскільки вона має більш бюджетний напрям та користується попитом у населення. Онлайн-платформа Vlabcacar.com.ua створена для пошуку автомобільних попутників, сайт busfor.ua (див. Рис.5) призначено для продажу міжміських та міжнародних автобусних квитків, туристичне онлайн-агентство tickets.ua надає послуги щодо покупки авіабілетів, квитків на автобус та залізницю, а також оренди автомобілів, сервіси proizd.ua і roizdato.net розроблені для покупки залізничних квитків.

Також в ході порівняльного аналізу розглянуто вузькоспеціалізовані сайти, на яких можна дізнатися інформацію та придбати квитки на до певної туристичної атракції, такі як portaventuraworld.com – парк розваг у Іспанії, disneylandparis.com – парк розваг у Франції та edinburghzoo.org.uk – зоопарк у Шотландії (див. Рис.6).

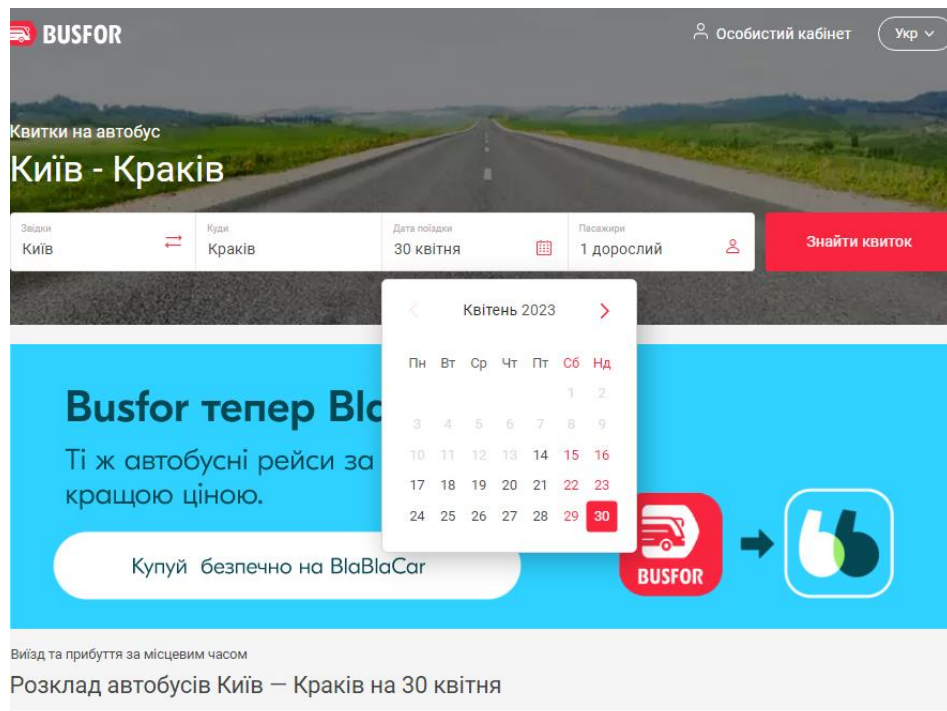


Рис. 5 Головна сторінка сайту busfor.ua

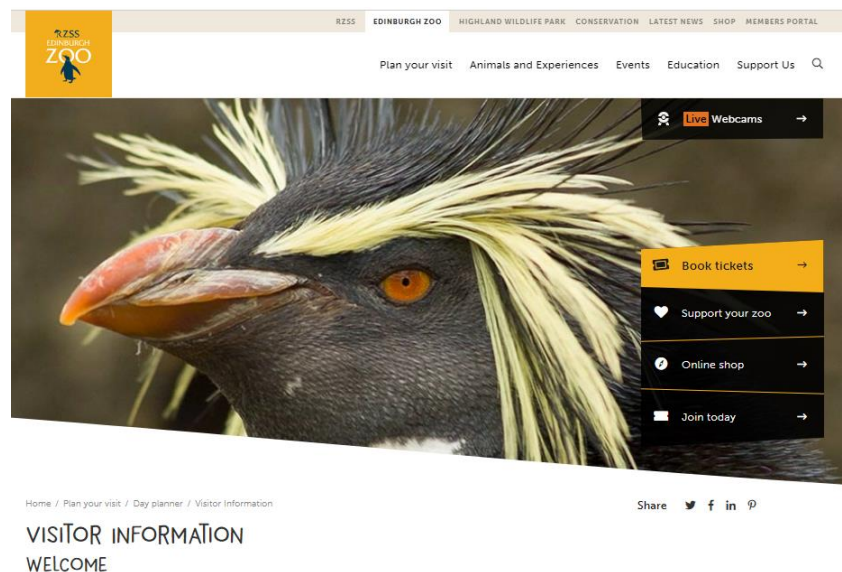


Рис. 6 Сторінка бронювання квитків сайту edinburghzoo.org.uk

В продовженні попередніх досліджень в роботі [2] опрацьовано значний масив інформаційних джерел, внаслідок чого визначено основні напрямки розвитку та вдосконалення туристичних мобільних інформаційних технологій. Акцентуючи увагу на створенні нових засобів і методів подання та опрацювання туристичної інформації та проектуванні інтелектуальних

систем генерування персоналізованих рекомендацій, авторами виділено перспективні напрямки туристичних інформаційних систем у зв'язку з використанням широкого спектра підходів та методів прийняття рішень, такі як: методи аналізу ієрархій та функцій корисності, генетичні алгоритми, нечіткі рішення, семантичні мережі.

1.2 Аналіз програмних продуктів-аналогів

Туристична галузь стрімко розвивалася в Україні до подій 2022 року і є велика надія на те, що вона набуде ще більшої популярності в майбутньому. Аналіз існуючих програмних продуктів засвідчує і суттєве зростання внутрідержавного туризму. Розглянемо деякі з них.

Сервіс пошуку емоцій Flixess (див. Рис.7) допомагає вибрати оптимальний тур зі зручними датами. На сайті доступні програми від топових туроператорів України з багаторічним досвідом. За необхідності є можливість сформувати корпоративний тур під конкретну компанію або надати індивідуальний тур. Сайт має зручну фільтрацію та інші фічі, що допомагає користувачу знайти тур мрії за заданими параметрами.

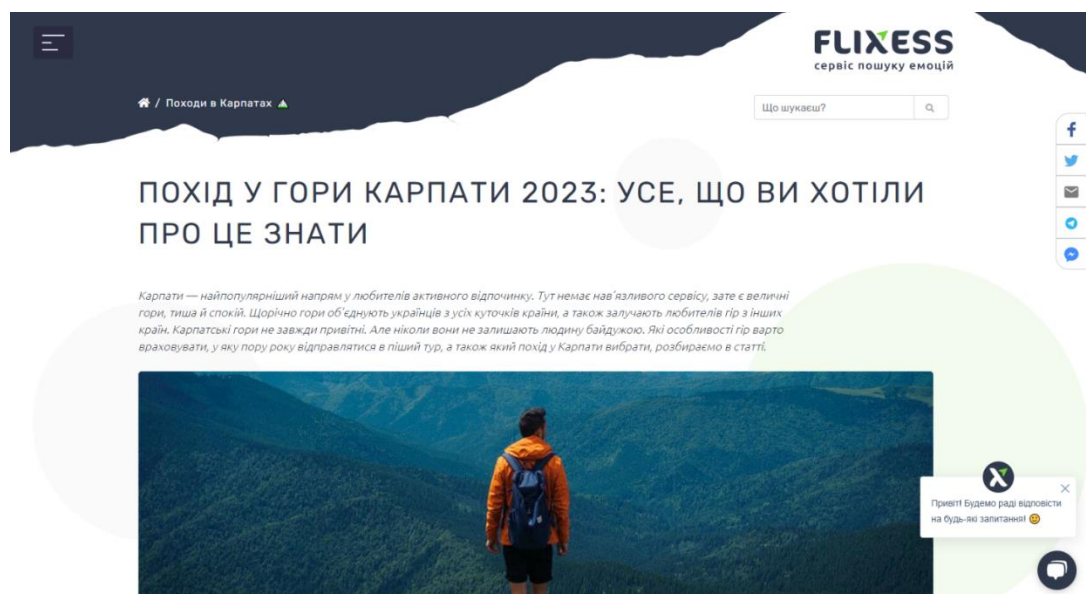


Рис. 7 Сторінка сервісу пошуку емоцій Flixess

Концепцією клубу Pohod V Gory є організація туристичних походів, які є доступні кожному. За допомогою застосунку користувач може підібрати для себе маршрут різного рівня складності й тривалості в різні регіони Карпат на будь-який смак (див. Рис.8).

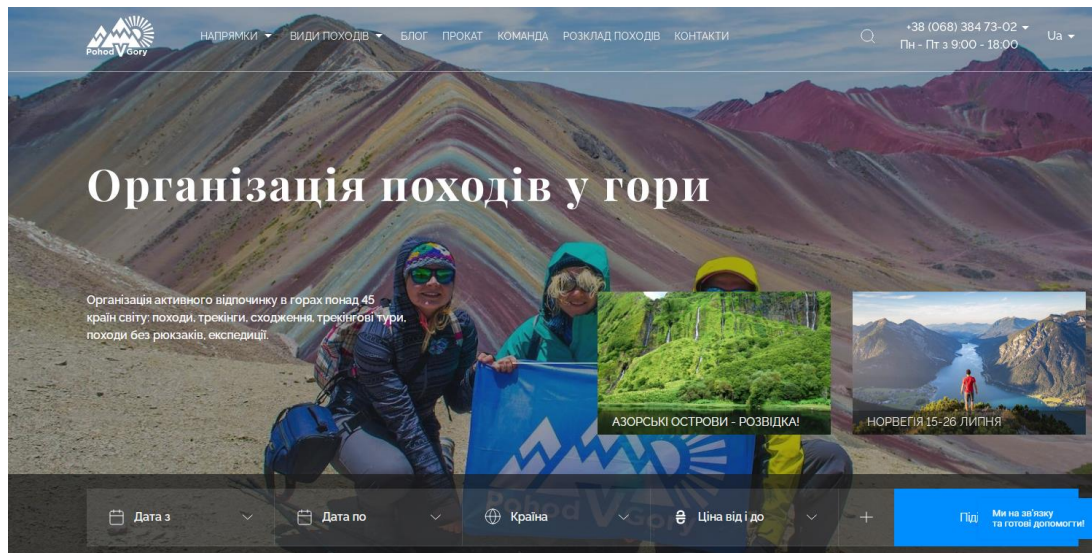


Рис. 8 Сторінка клубу Pohod V Gory

Користувачі сайту Discover мають можливість підібрати реальний перевірений автомобільний маршрут по різним регіонам України (див.Рис.9).

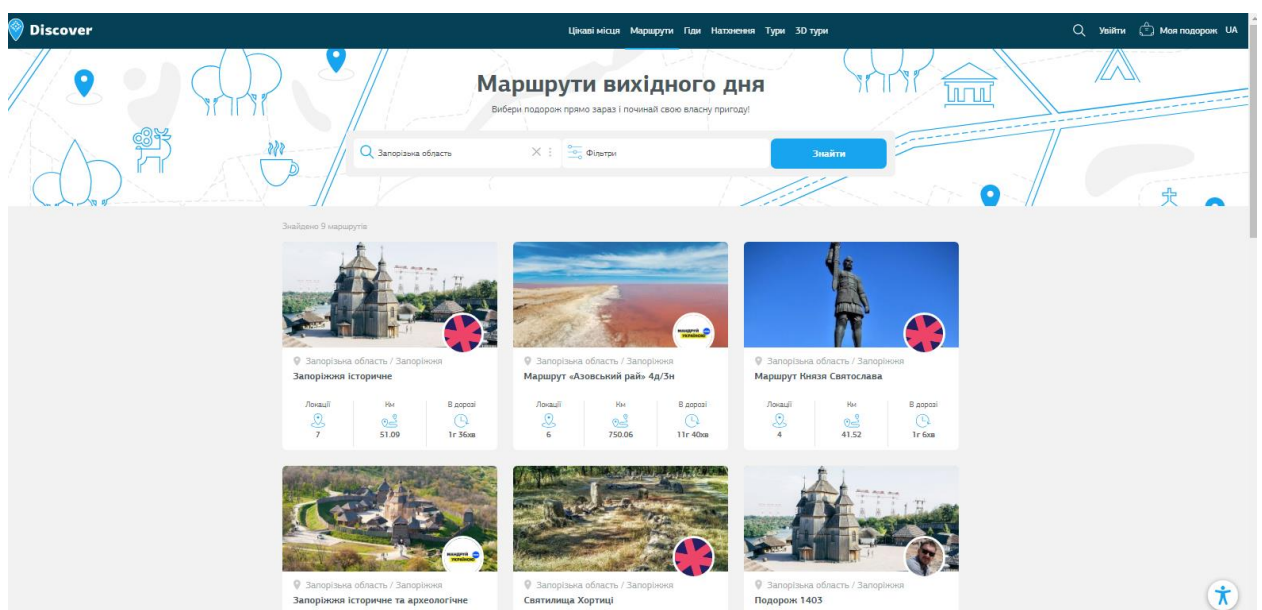


Рис. 9 Сторінка сайту Discover

В роботі [16] авторами проведено аналіз основних проблемних ситуацій, які виникають у процесі створення інтелектуальної інформаційної системи, і визначено суттєві функціональні особливості системи МІАТ. Мобільний інформаційний асистент туриста (МІАТ) – система яка забезпечуватиме комплексний повнофункціональний інформаційно-технологічний супровід туриста на етапах планування та здійснення ним подорожі (див.Рис.10).

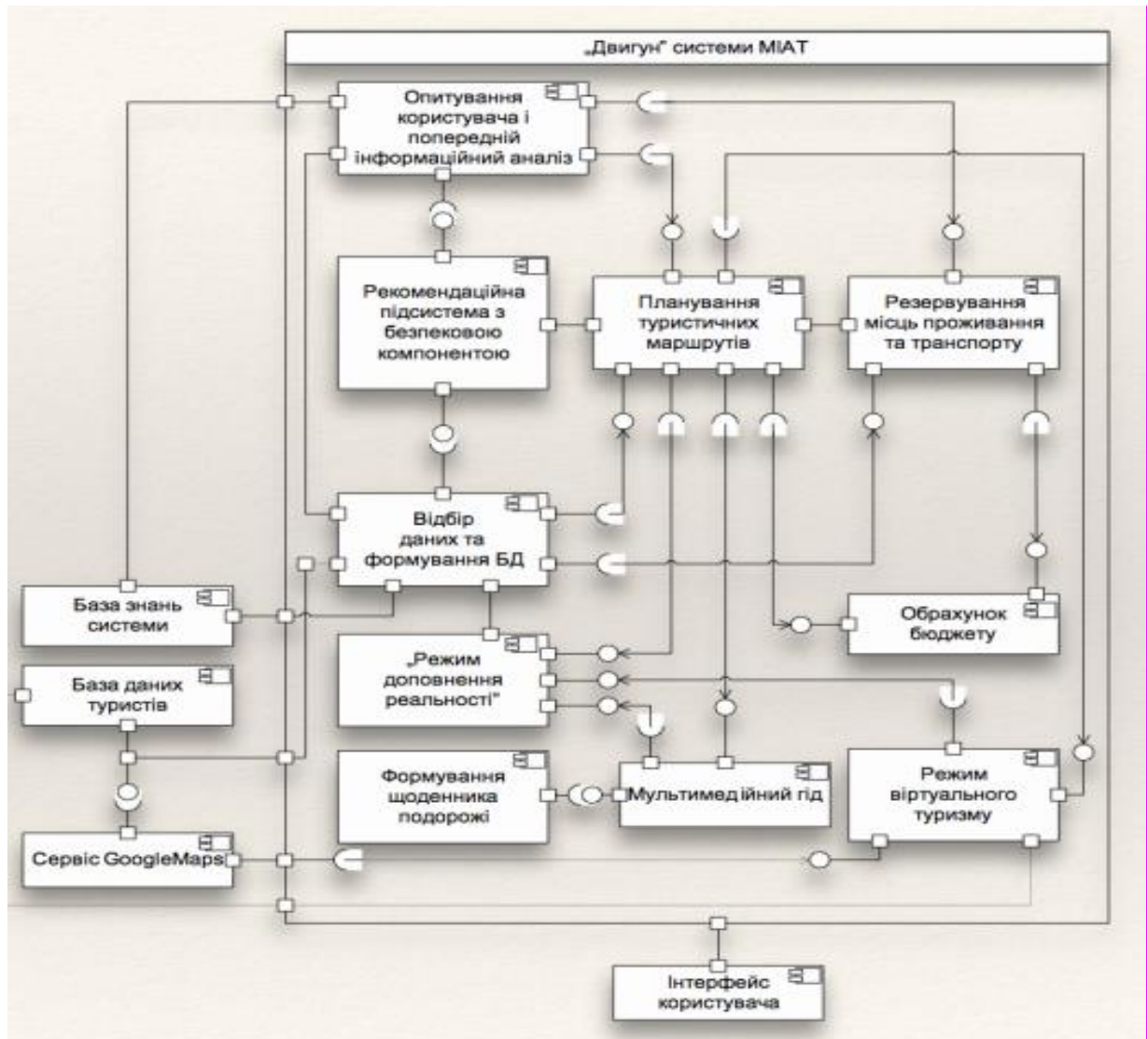


Рис. 10 Діаграма компонентів (Component diagram) системи МІАТ

Система має ряд функціональних можливостей, серед яких слід виділити:

- формування персоналізованих маршрутів з можливістю їх зміни під час подорожі;
- визначення місцезнаходження та навігація користувача під час подорожі;
- формування звіту затрат та щоденника подорожі, наявність режиму офлайн;
- обрахунок бюджету подорожі;
- мультимедійна система доповнення реальності під час відвідуванні конкретного туристичного об'єкта.

Інтерфейс програмного застосунку наведено на рисунку 11.

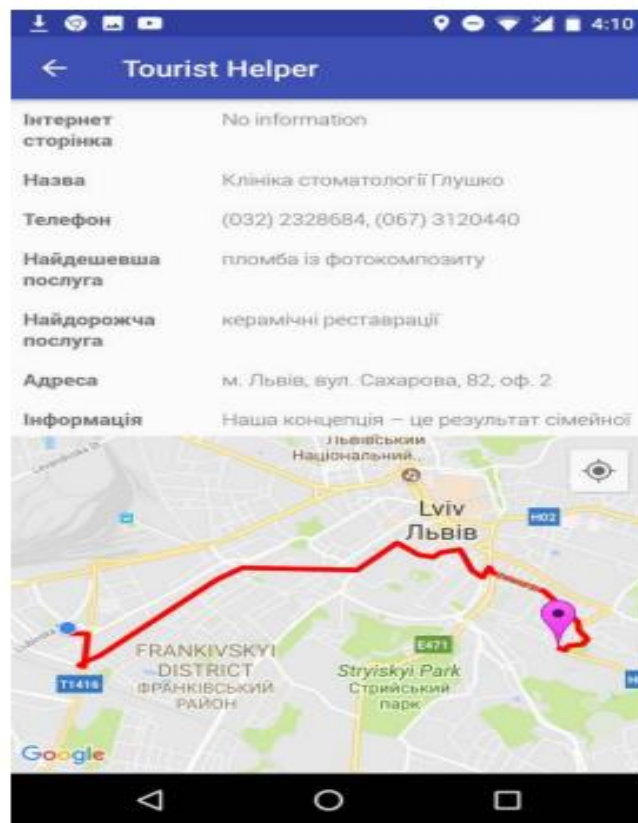


Рис. 11 Мобільний інформаційний асистент туриста

Не менш важливим є створення власної інтелектуальної інформаційної системи організації чи підприємства для підтримки різних процесів і функцій цього підприємства. Аналізуючи робочі процеси всередині туристичного

клубу доцільним буде мати власну базу туристичного клубу з присвоєнням розрядів туристам, з веденням щоденників походів, з графіками маршрутів. Це дасть можливість фіксувати похідну історію туриста, яка дозволяє підібрати похід відповідного рівня складності. Але в походах це ще потрібно для того, щоб не тільки не пропонувати легкі нецікаві походи досвідченим, але й стежити за верхньою планкою, щоб новачки не зійшли з маршруту, який вибрали, а він виявився для них дуже складний.

1.3 Постановка завдання

Мета кваліфікаційної роботи – це створення інтелектуальної інформаційної системи для підтримки діяльності туристичних клубів. Впровадження комп'ютеризованих технологій у сферу діяльності туристичного бізнесу має на меті полегшити організацію роботи співробітників клубів з клієнтами.

Для досягнення поставленої мети потрібно вирішити наступні задачі:

1. Побудувати інфологічну концептуальну модель (ER-модель), для чого:
 - проаналізувавши предметну область, при необхідності уточнивши і доповнивши її, виявити необхідний набір сутностей;
 - визначити необхідний набір атрибутів для кожної сутності, виділивши ідентифікуючі атрибути;
 - визначити зв'язки між сутностями, формалізувати зв'язки між сутностями;
 - описати отриману ER-модель на мові ER-діаграм.
2. Використовуючи СУБД MS SQL Server створити спроектовану базу даних на SQL .
3. Виконати на SQL необхідні дії з додавання, модифікації і видалення даних в таблицях створеної бази даних.
4. Створити інтерфейс користувача.
5. Створити систему безпеки.

2 ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Аналіз сучасних систем управління базами даних

Використання баз даних є однією з характерних більшості сучасних інформаційних систем. Система управління базами даних (СУБД) – це комплекс програмних, лінгвістичних засобів загального чи спеціального призначення, які забезпечують створення чи використання баз даних. СУБД створює на екрані комп'ютера певне середовище для роботи користувача (інтерфейс користувача). Крім того, СУБД має певні режими роботи та систему команд. На основі СУБД створюються та функціонують інформаційні системи. Також, системи управління базами даних – це одна з найуспішніших технологій у всій комп'ютерній галузі. Системи управління базами даних відіграють виняткову роль організації сучасних промислових, інструментальних та дослідних інформаційних систем. Характерними режимами роботи з СУБД є створення, редагування, керування та пошук у базі даних. Для роботи в кожному режимі існує своя система команд СУБД. Будь-яка робота користувача з базою даних будується у вигляді алгоритму, складеного з цих команд. Такі алгоритми можуть виконуватися в режимі прямого виконання (віддана команда відразу виконується) і в режимі автоматичного виконання, тобто у програмному режимі [9]. Сьогодні встановлення та адміністрування баз даних – це набагато менш складний процес, ніж кілька років тому. Проектування та управління базою даних великою мірою автоматизовано. Тому теорії створення та практиці використання баз даних приділяється достатня увага протягом періоду функціонування інформаційних систем.

Отже, основними функціями СУБД є:

- підтримка мов баз даних;

- керування даними на дисках;
 - ведення журналу змін, функції резервного копіювання та відновлення бази даних при збоях;
 - керування даними оперативної пам'яті із застосуванням дискового кеша.
- Зміст сучасної СУБД містить наступні складові:
- сервісну програму, що забезпечує додаткові можливості обслуговування інформаційних систем;
 - процесор мови БД, що оптимізує запити щодо вилучення, зміни та створення машинно-незалежного виконуваного внутрішнього коду;
 - підсистему підтримки періоду виконання, що інтерпретує програму з маніпуляції даними, формує користувальницький інтерфейс із СУБД;
 - ядро, що управляє даними та оперативною пам'яттю.

Досить тривалий час основним типом були реляційні бази даних, які на сьогодні вже вважаються класичними. Проте розвиток інформаційних систем поставив перед сучасними базами даних завдання, вирішення яких неможливе в межах використання тільки реляційних баз даних. Крім класичних завдань, сучасні бази даних повинні забезпечувати багатомашинну обробку та зберігання великих обсягів інформації, оперативний аналіз даних, інтеграцію із мережею Інтернет, розмежування доступу користувачів до інформації, що зберігається, захист інформації під час її передачі по мережі.

За способами доступу до баз даних виділяють наступні СУБД:

1) Файл-серверні СУБД: Microsoft Access, Paradox, Fox Pro, dBase та ін. В файл-серверних СУБД для програм є відкритим загальний доступ до всіх файлів бази даних (які зазвичай зберігаються в якому-небудь файловому сховищі) і можуть спільно обробляти ці дані. Кожний застосунок обробляє дані самостійно. У таких системах, по-перше, відсутні суттєві переваги клієнт-серверних застосунків, такі як паралелізм запитів, кешування даних, висока продуктивність, по-друге, їм властивий ряд недоліків, такі як складності відновленням, блокуваннями, з підтримкою цілісності бази. Це призводить до зниження продуктивності та надійності. Отже, на сьогодні файл-серверна

технологія вважається застарілою, і її використання у великих інформаційних системах є недоліком.

Найбільш поширеною є система управління базами даних від Microsoft, яка поєднує реляційне ядро БД Microsoft Jet з графічним інтерфейсом користувача та інструментами розробки програмного забезпечення. Файл-серверна СУБД *Microsoft Access* є гарним рішенням для початку роботи з даними, але її продуктивність не розрахована на великі проекти. У MS Access можна використовувати C, C#, C++, Java, VBA та Visual Rudimental.NET. Access зберігає всі таблиці баз даних, запити, форми, звіти, макроси та модулі у базі даних Access Jet у вигляді одного файлу. До особливостей можна віднести: можливість використовувати VBA для створення багатофункціональних рішень з розширеними можливостями управління даними та користувальницьким контролем, імпорт та експорт у формати Excel, Outlook, ASCII, dBase, Paradox, FoxPro, SQL Server та Oracle, формат бази даних Jet.

2) Клієнт-серверні СУБД: InterBase, Microsoft SQLServer, Firebird, Oracle, MySQL, PostgreSQL. У клієнт-серверних СУБД вся обробка даних проводиться в одному місці – на сервері, там же, де зазвичай зберігаються дані, доступ до файлів даних має лише один сервер, одна система – сама система управління базами даних. При цьому програми-клієнти лише посилають запити до СУБД на обробку даних та отримують відповіді; але безпосереднього доступу до файлів даних програми-клієнти не мають. На даний момент практично всі промислові СУБД є саме клієнт-серверними.

Клієнт-серверна СУБД *Oracle RDBMS (Oracle Database)* є популярною серед розробників, оскільки вона проста у використанні, має зрозумілу документацію, підтримку довгих найменувань, JSON, покращений тег списку та Oracle Cloud, підтримує SQL, до нього можна отримати доступ із реляційних баз даних Oracle, використовує Oracle NoSQL Database з Java/C API для читання та запису даних.

3) Вбудовані СУБД: MicrosoftSQL Server Compact, Firebird Embedded, SQLite. Вони поставляються у складі якогось готового програмного продукту, їм потрібна самостійна установка. Завданням СУБД цього типу є локальне збереження даних застосунку, вони не розраховані на колективне використання в мережі. Наприклад, одна з вбудованих безкоштовних СУБД SQLite широко використовується в мобільній ОС Android і багатьох мобільних додатках

4) NoSQL системи: MongoDB, Apache Cassandra, Redis.

MongoDB – кросплатформена документоорієнтована база даних з відкритим програмним забезпеченням, спирається на концепції колекцій і документів. Будь-яка реляційна база даних має стандартну схему, яка показує кількість таблиць і зв'язок між ними, а в MongoDB такі зв'язки не передбачені. До переваг MongoDB перед реляційними СУБД відносять:

- відсутність схеми. MongoDB – це база даних, в якій одна колекція містить різні документи, кількість полів, вміст і розмір документів можуть відрізнятися;
- зрозуміла структура об'єкта;
- відсутність складних з'єднань Join;
- розширені можливості запитів. MongoDB підтримує динамічні запити документів, використовуючи мову запитів на їх основі, що практично не поступається в ефективності SQL;
- налаштування;
- легкість масштабування;
- не потребує перетворення об'єктів програми на об'єкти бази даних;
- використовує внутрішню пам'ять для зберігання віконного робочого набору, забезпечуючи прискорений доступ до даних.

Apache Cassandra – це не реляційна, стійка до відмови розподілена СУБД, розрахована на створення високомасштабованих і надійних сховищ великих масивів даних, представлених у вигляді хеша. Завдяки розподіленій архітектурі Cassandra надає наступні можливості [4]:

- розподіл даних між вузлами кластера є прозором для користувачів – кожен сервер може приймати будь-який запит (на читання, запис або видалення даних), пересилаючи його на інший вузол, якщо інформація, що запитується, зберігається не в цьому місці;
- користувачі можуть самі визначити необхідну кількість реплік, створення та управління якими забезпечить Cassandra;
- наявність рівня узгодженості даних щодо кожної операції зберігання та зчитування, що настраюється користувачами;
- висока швидкість запису (близько 80-360 МБ/с на вузол) – дані записуються швидше, ніж зчитуються за рахунок того, що їх більша частина зберігається в оперативній пам'яті відповідального вузла, і будь-які оновлення спершу виконуються в пам'яті, а потім – у файловій системі;
- масштабованість і надійність через відсутність центрального сервера (Master Node), відмова якого може стати причиною збою всього кластера.

Redis (Remote Dictionary Server) – СУБД з відкритим початковим кодом. Вона є однією з найпродуктивніших СУБД, оскільки час відгуку сервера становить частки мілісекунд. Система користується попитом у сферах, де потрібно працювати з великими масивами даних, а саме: розробка ігор, аналітика, робота з геоданими, підтримка онлайн-чатів, служби таксі, потокова передача мультимедіа та інші. Сховище використовується не тільки як база даних, а й як засіб кешування та брокер повідомлень. Сервер зберігає дані не строго у схемах (таблицях, рядках і стовпцях), а самих структурах даних, що дозволяє гнучко налаштовувати систему.

Незважаючи на переваги, до яких відноситься: висока продуктивність, масштабованість, асинхронна реплікація, підтримка різних структур даних, функціональність (підтримка мов програмування – Java, PHP, C++, JavaScript, Node.js, Python, Ruby, Go), СУБД Redis не є універсальною і підходить для обмеженого списку завдань і проєктів. До переліку задач, в яких використовують СУБД *Redis* можна віднести:

- Кешування даних основного сховища. Redis cache допомагає знизити навантаження на основне сховище та сам додаток без нарощування потужностей на серверах.
- Створення чатів та черг. Висока продуктивність дозволяє налаштувати «швидкі» стрічки новин у блогах та соціальних мережах, оновлення стрічок із коментарями та чатів у режимі реального часу.
- Брокер повідомлень. Дозволяє розвантажити веб-сервіси у розподіленій системі: під час надсилання повідомлень не потрібен додатковий час на ресурсоємні операції (наприклад, маршрутизацію або пошук приймачів).
- СУБД для великих програм. Remote server дозволяє рівномірно розподілити мережевий трафік.
- Зберігання сесій. Завдяки високій доступності, відмовостійкості та масштабованості в Redis зручно зберігати дані сесій.
- Машинне навчання. Redis зберігає інформацію у швидкісному сховищі, яке працює за принципом ОЗП у комп'ютері. Воно дозволяє швидко обробляти великі обсяги даних та автоматизувати прийняття рішень.
- Зберігання проміжних даних, наприклад форм голосування або таблиць зі збором результатів.
- Зберігання «швидких» даних, якими оперують під час аналітики, аналізу даних, на торгових сервісах.
- Таблиці лідерів у іграх. Структура даних Redis Sorted Set дозволяє створювати ранжовані списки у реальному часі. Щоб внести одну зміну, достатньо оновити результат конкретного користувача, а не всієї таблиці.

2.2 Аналіз розподілених СУБД

За методами зберігання та обробки даних СУБД поділяють на централізовані та розподілені. Централізована СУБД зберігається на одному комп'ютері, але користувач працює з базою даних через віддалений доступ (у режимі клієнт-сервер). Більшість централізованих СУБД виконують лише свої

стандартні функції, які ускладнюються за рахунок одночасного доступу багатьох користувачів до даних.

Розподілені СУБД призначені для того, щоб забезпечити засоби інтеграції локальних баз даних, що знаходяться в деяких вузлах комп'ютерної мережі, для того, щоб користувач, який працює в будь-якому вузлі мережі, мав доступ до всіх цих баз даних як до єдиної.

Розподілені СУБД у порівнянні з централізованими мають ряд переваг, а саме:

- відбивають структуру організації;
- мають локальну автономність;
- забезпечують високу доступність даних;
- мають високу надійність і підвищену продуктивність.

Серед недоліків розподілених СУБД можна виділити:

- розподілені СУБД є складнішими програмними комплексами, ніж централізовані СУБД, що зумовлено розподіленою природою використовуваних ними даних, а також реплікацією даних;
 - збільшення складності означає і збільшення витрат на придбання та супровід розподілених СУБД;
 - у розподілених системах потрібно організувати контроль доступу не тільки до даних, що реплікуються на кілька різних вузлів, але і захист мережових з'єднань самих по собі;
 - у розподілених СУБД підвищена вартість передачі та обробки даних може заважати організації ефективного захисту від порушень цілісності даних;
 - відсутні стандарти на канали зв'язку та протоколи доступу до даних, а також відсутні інструментальні засоби та методології, які здатні допомогти користувачам у перетворенні централізованих систем у розподілені;
 - недостатньо досвіду промислової експлуатації розподілених систем, у порівнянні із досвідом експлуатації централізованих систем;

- розподілені СУБД складні в управлінні, що може привести до потенційної небезпеки втрати цілісності даних.

Зміна типів даних, що зберігаються, вимоги до швидкості і продуктивності привели до триваючого розширення типів баз даних. При цьому кожен з них продовжує бути потрібним у своїй ніші, де взаємозв'язки між даними асоціюються з певною схемою будови бази даних. Вміння вибрати СУБД важливе під час розробки будь-якого програмного забезпечення. З метою обрання бази даних для розробки інформаційної системи керування туристичним комплексом проведемо аналіз найпопулярніших розподілених систем управління базами даних.

PostgreSQL є масштабованою об'єктно-реляційною базою даних з відкритими текстами, що працює на Linux, Windows, OSX та деяких інших системах. У PostgreSQL 10 є такі функції, як логічна реплікація, декларативне розбиття таблиць, покращені паралельні запити, безпечніша аутентифікація по пароллю на основі SCRAM-SHA-256. До особливостей можна віднести: підтримку табличних просторів, а також процедур, об'єднань, уявлень і тригерів, що зберігаються, відновлення на час (PITR), асинхронну реплікацію, вкладені транзакції (точки збереження), має підтримку міжнародних кодувань, у тому числі і багатобаштових. PostgreSQL має API для Java, C/C++, Perl, Ruby, Python, ODBC, Tcl. Існують діючі PostgreSQL системи, які можуть керувати більш ніж 4 терабайтами даних. Отже, PostgreSQL є не тільки потужною системою управління базами даних, яка дозволяє забезпечувати діяльність організації, а й платформою розробки для створення програм, які вимагають використання реляційної СУБД

IBM DB2 є сімейством систем управління реляційними базами даних, які випускає корпорація IBM.

До характерних особливостей DB2 відносять:

- вбудовані засоби контролю та розмежування доступу, які реалізують модель розмежування доступу, а також надають можливості обмеження доступу до інформації у розрізі об'єктів (таблиць, уявлень);

- масштабність, обмежена лише наявними обчислювальними ресурсами, та забезпечує максимально економічне та ефективне використання обчислювальних ресурсів;
- наявність розвиненої інтегрованої системи резервного копіювання та відновлення даних;
- повний набір технологій для побудови «класичних» аналітичних сховищ даних;
- висока відмовостійкість та здійснення лінійного масштабування кластерних конфігурацій DB2 pureScale, з можливістю зберігання даних на дисках із загальним доступом;
- підтримка механізму побудови конфігурацій масивнопаралельної аналітичної обробки даних (MPP) з великої кількості серверів, об'єднаних у комунікаційній мережі, на базі DB2 DPF;
- підтримка технології DB2 BLU, що реалізує підтримку сучасного ін-методу «поколонкового» аналізу без застосування ручної оптимізації структури баз даних.

До перевагами даної СУБД відносять:

- наявність гарної безкоштовної версії без обмежень на розмір бази;
- відсутність безкоштовної версії обмежувача запитів;
- наявність хорошої безкоштовної підтримки;
- на відміну від PostgreSQL, наявність можливості отримання платної підтримки виробника;
- має високу продуктивність;
- менше труднощів, пов'язаних із неунікальністю індексів;
- відсутність обмеження на 256 таблиць, що дозволяє розширювати можливості роботи з RLS.

Серед недоліків виділяють:

- невелику поширеність;
- невелику кількість фахівців та високу вартість їхніх послуг;
- більший розмір баз, ніж у інших СУБД;

- повільну швидкість завантаження dt-файлу.

MySQL є системою, яка дозволяє створювати користувачів бази даних та встановлювати їм права доступу. Для редагування прав доступу в *MySQL*, потрібно спочатку увійти на свій сервер, потім підключитися до клієнта *mysql*. Зазвичай виконується підключення до *root* або до будь-якого облікового запису, який є основним, початковим обліковим записом суперкористувача та має повний доступ протягом усієї роботи з *MySQL* [8]. Для надання привілеїв користувачам використовується команда *GRANT*. У більшості випадків привілеї надаються користувачам *MySQL* на основі конкретної бази даних, до якої обліковий запис повинен мати доступ. Проте в системі передбачена можливість створювати ще одного суперкористувача, який матиме повний доступ до всіх баз даних. Для інтерактивної роботи з базою даних до складу системи *MySQL* входить *MySQL Workbench* – уніфікований візуальний інструмент для архітекторів баз даних, розробників і адміністраторів. *MySQL Workbench* забезпечує моделювання даних, розробку *SQL* та комплексні інструменти адміністрування для конфігурації сервера, адміністрування користувачів, резервного копіювання. *MySQL* працює на *Linux*, *Windows*, *OSX*, *FreeBSD* та *Solaris* [9]. Ця система управління базами даних використовує стандартну форму *SQL*. Утиліти проектування таблиць мають інтуїтивно зрозумілий інтерфейс. *MySQL* підтримує до 50 мільйонів рядків у таблиці. Граничний розмір файлу для таблиці за замовчуванням 4 ГБ, але можна збільшити. Підтримує секціонування та реплікацію, а також *Xpath* та збережені процедури, тригери та уявлення. До особливостей можна віднести: масштабованість, легкість використання, безпеку, підтримку *Novell Cluster*, швидкість, підтримка багатьох операційних систем. Гнучкість забезпечується підтримкою великої кількості таблиць, користувачі можуть вибрати як таблиці, які підтримують транзакції на рівні окремих записів, так і таблиці, які підтримують транзакції на рівні окремих записів. Крім того, СУБД поставляється зі спеціальним типом разом із таблицями *EXAMPLE*, що

демонструє принципи створення нових типів таблиць. Їхній появі сприяє відкрита архітектура та GPL-ліцензування в СУБД MySQL.

Microsoft SQL Server є найпопулярнішою комерційною СУБД. SQL Server виконує обробку запитів від користувачів та надсилає їм лише результати запиту. Таким чином, по мережі передається мінімальна інформація, що прискорює час відповіді та дозволяє усувати вузькі місця у мережі. Це також робить SQL Server ідеальною базою даних для використання в Інтернеті. Технічне обслуговування SQL Server дуже просте і не вимагає великих знань, можливі зміни у структурі даних, а також резервне копіювання під час роботи сервера без зупинки [6]. До особливостей використання цієї СУБД можна віднести: залежність від платформи, базування на командах і графічному інтерфейсу і програмного забезпечення, підтримка SQL, не процедурні і нечутливі до регістру та загальні мови баз даних, висока продуктивність, можливість встановлення різних версій на одному комп'ютері, генерація скриптів для переміщення даних.

Висока надійність забезпечується за рахунок застосування різних базових технологій, таких як створення відмовостійких кластерів, дзеркалювання, надання різноманітних засобів для роботи з журналами.

Поява функції партиціонування дозволяє розбивати великі таблиці на кілька елементів, які прив'язані до різних файл-груп. В результаті дані фізично розміщуються на кількох жорстких дисках, і таким чином операції читання/записи розподіляються. Ця технологія абсолютно багатозначна для користувача додатка, та її робота здійснюється автоматично.

За рахунок того, що мережна безпека пов'язана із самим сервером безпеки – дані максимально захищені від несанкціонованого доступу. Оскільки функції безпеки реалізовані на рівні користувача, їх доступ до запису даних може бути обмеженим, захищаючи їх тим самим від пошуку чи модифікації. Для подальшого підвищення швидкодії використовується технологія стиснення, яка працює як на рівні записів, так і на сторінках. З

метою підвищення безпеки у системі управління базами даних реалізовано підтримку сучасних криптоалгоритмів.

Проведений аналіз підтвердив, що клієнт-серверна СУБД є найменш вимогливою до пропускнуої можливості комп'ютерної мережі, ніж файл-серверна СУБД, оскільки відсутня необхідність отримувати на клієнта повний масив даних: клієнт передає серверу характеристики запиту, а сервер виконує пошук з локальної бази даних. Результат виконання запиту, зазвичай менше обсягу, ніж весь масив даних, повертається клієнту з наступним відображенням результату користувачу.

2.3 Технологія для доступу до баз даних Entity Framework Core

Entity Framework Core (EF Core) являє собою об'єктно-орієнтовану технологію від компанії Microsoft для доступу до даних, є легкою, відкритою, має можливість розширюватися [5]. EF Core є ORM-інструментом (object-relational mapping – відображення даних на реальні об'єкти). Тобто EF Core дозволяє працювати з базами даних, але має більш високий рівень абстракції: EF Core дозволяє абстрагуватися від самої бази даних і її таблиць, а також працювати з даними незалежно від типу сховища. На відміну від фізичного рівні, на якому є можливість оперувати таблицями, індексами та первинними і зовнішніми ключами, на концептуальному рівні, який пропонує даний Framework, працюють саме з об'єктами (див. Рис.12).

Враховуючи той факт, що Entity Framework Core підтримує безліч різних систем баз даних, є можливість через нього працювати з будь-якої СУБД, якщо для неї є відповідний провайдер [6]. За замовчуванням на даний момент Microsoft надає ряд вбудованих провайдерів: для роботи з MS SQL Server, для SQLite, для PostgreSQL, крім того, є провайдери від сторонніх постачальників, наприклад, для MySQL.

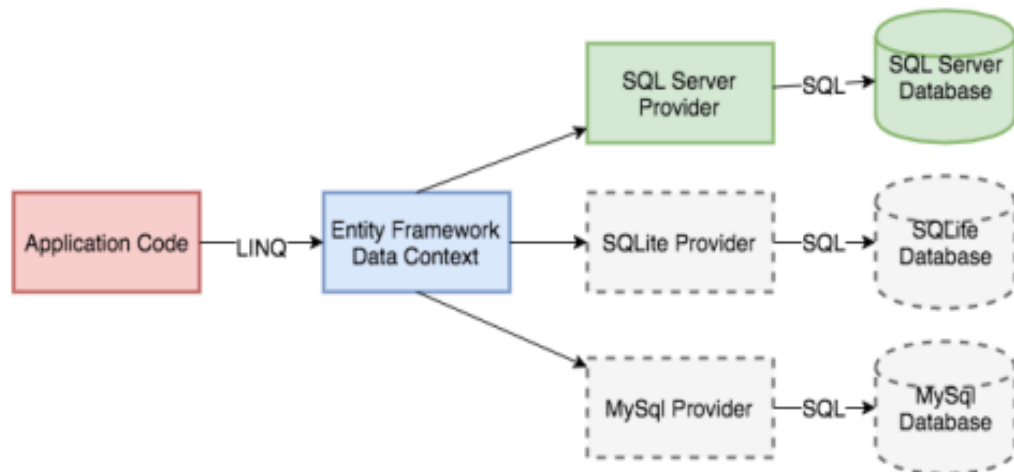


Рис. 12 Структура роботи технології EF Core

Важливим чинником під час розробки програмного продукту є вибір засобів програмної реалізації та технологій. Проаналізувавши поставлену задачу та методи її вирішення, було вирішено розробити програмний продукт у вигляді десктопного додатку з використанням технології Entity Framework Core для роботи з ним. Для створення інформаційної системи керування туристичним комплексом обрано СУБД Microsoft SQL Server 2018.

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ КЕРУВАННЯ ТУРИСТИЧНИМ КОМПЛЕКСОМ

3.1 Опис предметної області

Зазвичай, автоматизація роботи туристичних комплексів здійснюється на основі стандартних офісних процедур (Excel, Word, Outlook), а компанія створює власний сайт в Інтернеті. Програмне забезпечення для автоматизації всієї роботи може відрізнятися від потреб туристичної компанії. З метою підвищення ефективності туристичного комплексу створюються типові програмні комплекси, які є універсальними (незалежно від характеру туристичної діяльності, ними можуть користуватися як невеликі туристичні фірми, так і потужні турфірми), мають інтуїтивно зрозумілий інтерфейс та не вимагають спеціальної комп'ютерної підготовки, прості у використанні, що запобігає помилковій роботі користувачів.

Туристи, що приходять в туристичний клуб, можуть не лише ходити в планові походи, але й займатися у різних секціях протягом усього року. Для цього вони записуються в групи, що відносяться до певних секцій. Туристів можна умовно по всій базі розділити на аматорів, спортсменів і тренерів. Кожна з перерахованих категорій може мати свій набір характеристик-атрибутів. Секції клубу очолюються керівниками, у функції яких входить контроль за роботою секції. В роботі секції беруть участь тренери, які адміністративно відносяться до однієї із секцій. Керівник секції призначає кожній групі тренера. Тренер може тренувати декілька груп, при чому вони не обов'язково належать до його секції. Спортсмени і тренери можуть брати участь в різних змаганнях. Кожен рік складається розклад роботи секцій. У ньому вказується, які тренування будуть проводитися і в яких секціях: їх кількість, місце, час і т.д. У відповідності з цим керівники секцій здійснюють розподіл навантаження для тренерів (з урахуванням їх спеціальності). Відомості про проведені тренування та про відвідування тренувань

збираються керівниками. Протягом року клуб організовує різноманітні походи. Кожен похід має свій маршрут, на який відводиться певна кількість днів. За маршрутом і кількості днів визначається категорія складності даного походу. Похід очолює інструктор, котрим може бути будь-який тренер або спортсмен. Він набирає групу в кількості 5-15 чоловік для свого походу, виходячи з типу походу (піший, кінний, водний, гірський) і фізичних даних туристів (за їх занять у секціях: водники, спелеологи, альпіністи та інші, з урахуванням специфіки занять – не вміє плавати ніколи не візьмуть на сплав, а в піший похід невеликої категорії складності можуть взяти будь-якого туриста). Інструктор може водити в походи даної категорії складності, якщо він сам її раніше вже пройшов. Походи можуть бути плановими і непланових. Для кожного планового походу існує точний план, в якому вказується маршрут, розклад привалів та стоянок на кожен день. Під час планового походу ведеться щоденник. Непланові походи мають тільки маршрут і повний час його проходження. Неплановий похід може бути переведено в категорію планових. Кожному туристу присвоюється категорія максимально складного з пройдених ним планових походів.

3.2 Архітектура системи

Інформаційна система ведення діяльності туристичного клубу, представлена в даній роботі, підрозділяється на дві частини:

- серверна база даних;
- клієнт – підсистема, яка безпосередньо відповідає за ведення діяльності туристичного клубу, що дозволяє документувати дані про туристів та їх підготовку, тренерів та їх кваліфікацію, походи, маршрути та їх рівень складності.

На діаграмі розгортання (див. Рис. 13) показано фізичне розташування системи на сервері і клієнтських машинах туристичного клубу. Також там

показана локальна обчислювальна мережа (ЛОМ) як засіб обміну даними між сервером і клієнтами.

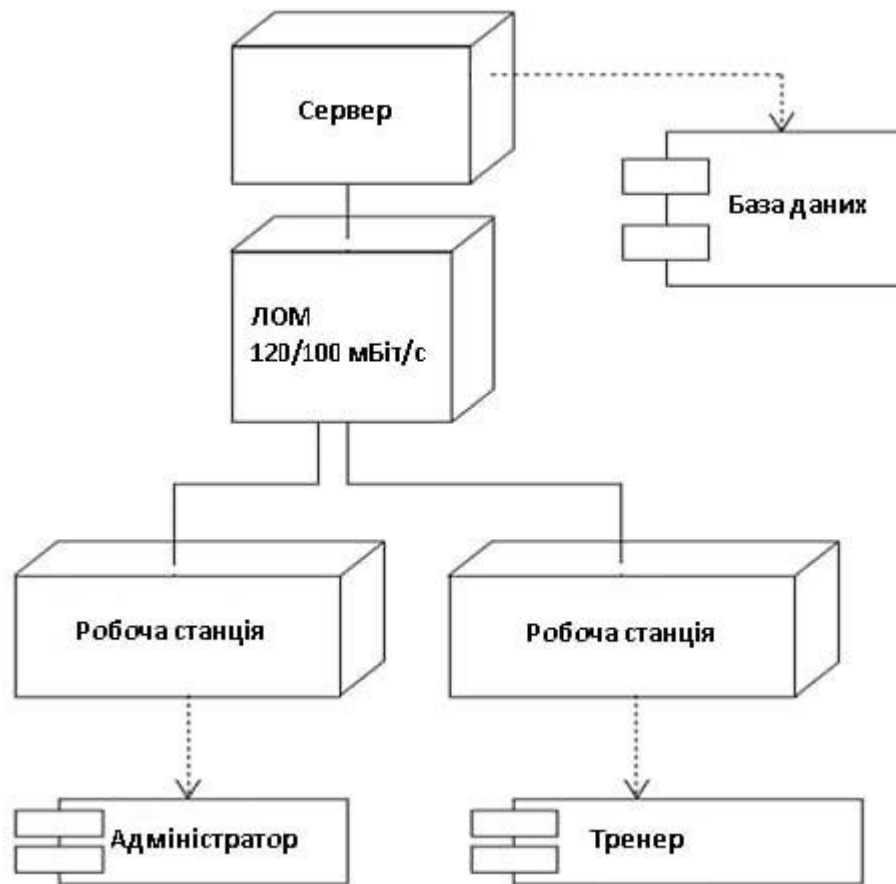


Рис. 13 Діаграма розгортання системи

3.3 Функціональні вимоги системи

У відповідності з архітектурою системи вона поділяється на дві частини, тому необхідно виробити функціональні вимоги як для системи в цілому, так і для кожної з частин окремо.

Система повинна забезпечувати:

1. Ведення даних – систематизованого переліку маршрутів, туристичних напрямків, походів по цих маршрутах, рівня кваліфікації

туристів, що мають відповідний досвід у проходженні маршрутів, які пропонує туристичний клуб.

2. Оперативне оформлення відповідних документів (щоденник, розклади секцій, тренувань, наявні маршрути, результати проведених змагань).

3. Ведення кількісного обліку туристів, що займаються в клубі, в зазначеній секції, групі, за статевою ознакою, роком народження, віком, та тренерів, що провели тренування у зазначеній групі, за вказаний період часу.

4. Ведення кількісного обліку туристів з вказаної секції, групи, які ходили в задану кількість походів, в зазначений похід, у зазначений час, по заданому маршруту, були в деякій точці, мають відповідну категорію.

5. Ведення кількісного обліку навантаження тренерів (вид занять, кількість годин), його обсяг із конкретних видів занять і загальне навантаження за вказаний період часу для даного тренера або зазначеної секції.

6. Складання всіх необхідних звітів по діяльності туристичного клубу.

Серверна база даних повинна забезпечувати:

1. Адекватне відображення даних предметної області.
2. Надійне зберігання даних системи.
3. Зручне використання даних (перегляд та редагування).

3.4 Проектування інформаційної системи

Під час етапу проектування було проведено структурно-функціональне моделювання, котре дозволило прослідити виконання всіх бізнес-процесів інформаційної системи, проаналізувати напрямки удосконалення роботи [10].

Головною метою проектування є забезпечення ефективного функціонування інформаційної системи у відповідності до поставлених вимог та при наявності заданих обмежень та технологій [12]. Для виділення основних

можливостей системи також було створено діаграми варіантів використання для акторів. Діаграми варіантів використання потрібні для відображення дій, які зовнішній користувач (актор) може виконувати у системі [14]. Діаграми варіантів використання для акторів адміністратора та тренера наведені на рисунках 14 та 15 відповідно.

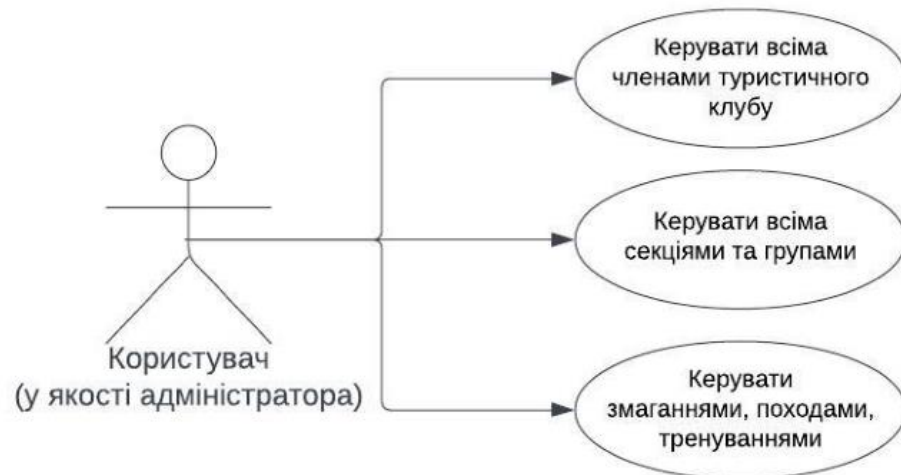


Рис. 14 Діаграма варіантів використання для актора адміністратора

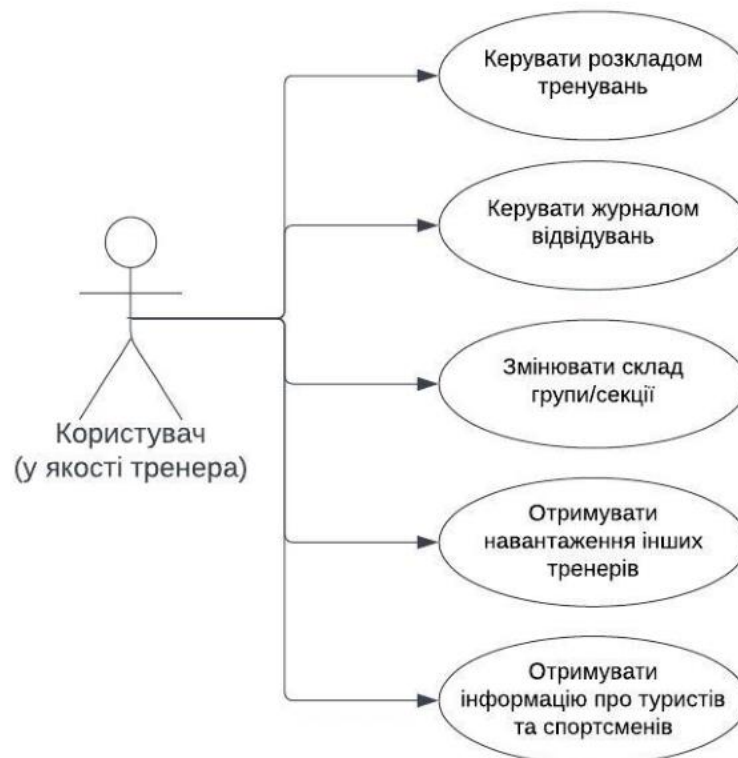


Рис. 15 Діаграма варіантів використання для актора тренера

Для опису моделі предметної області наведено ER-діаграму бази даних туристичного комплексу (див. Рис.16), а також структуру таблиць реляційної бази даних. На наведеній діаграмі зображені зв'язки між виділеними сутностями.

Структура бази даних відображена на схемі, яка наведена у двох частинах. Перша частина – робота секцій туристичного клубу (див. Рис.17).

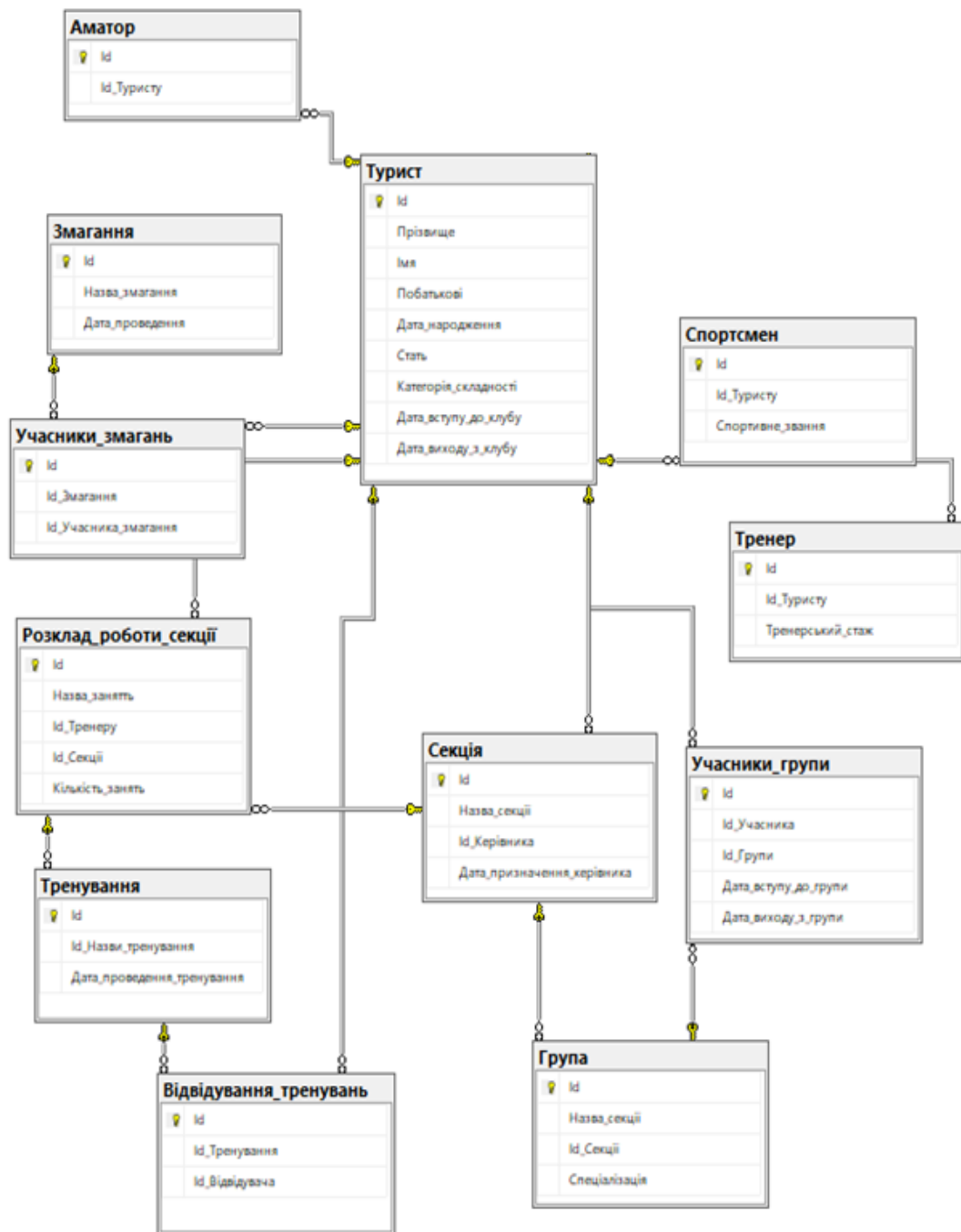


Рис. 17 Схема роботи секцій туристичного клубу

Окрім тренувань, учасники туристичного клубу можуть ходити у походи, що продемонстровано другою частиною схеми бази даних (див. Рис. 18).

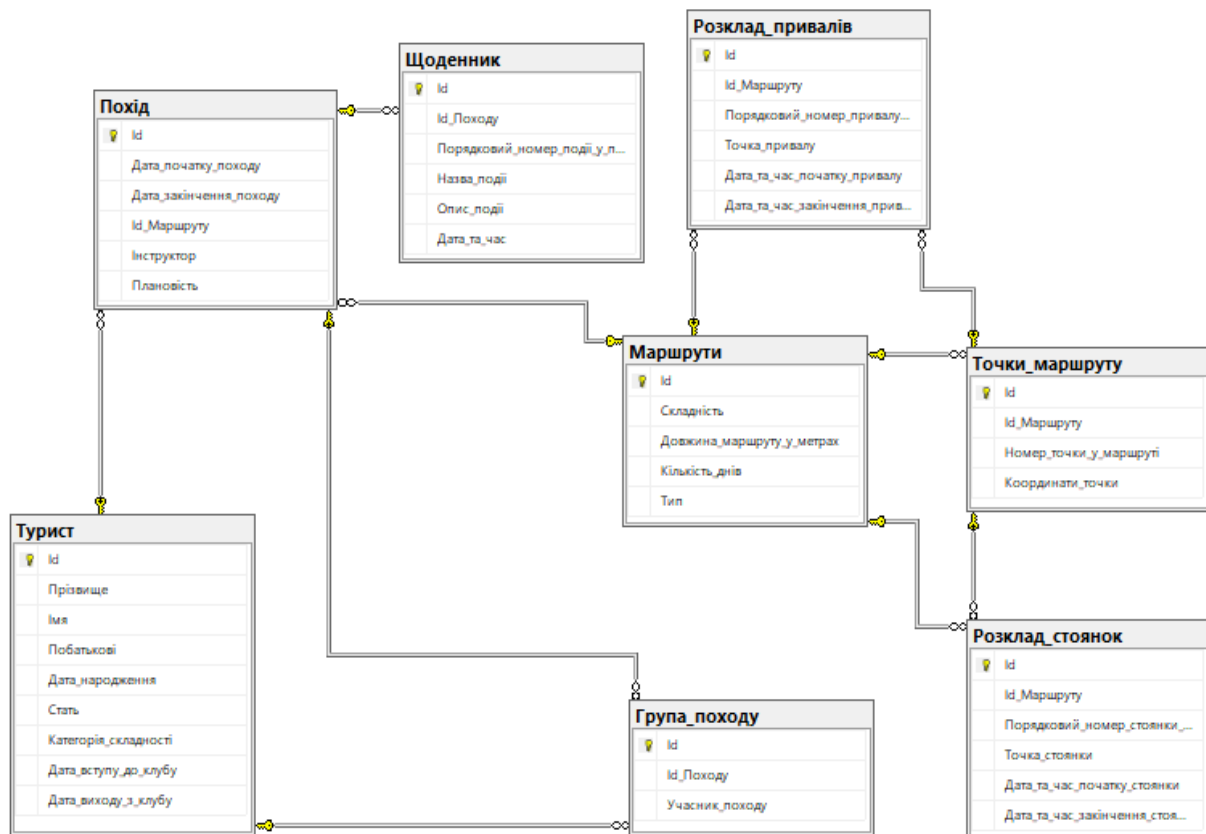


Рис. 18 Схема походів туристичного клубу

Для створення бази даних необхідно виконати відповідний скрипт (див. Лістинг 1).

Лістинг 1 Створення бази даних

```
use master
go
drop database if exists TouristDatabase
go
create database TouristDatabase
go
```

База даних складається з 19 таблиць, які містять в собі відповідні сутності. Кожна таблиця має первинний ключ.

Сутність «Турист» містить інформацію про туристів, що входять до туристичного клубу, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 1.

Таблиця 1. Атрибути сутності «Турист»

| Назва | Тип | Зв'язок | Опис |
|----------------------|---------------|---------|----------------------|
| Id | int | - | Первинний ключ |
| Прізвище | nvarchar(255) | - | Прізвище |
| Імя | nvarchar(255) | - | Ім'я |
| Побатькові | nvarchar(255) | - | По батькові |
| Дата_народження | date | - | Дата народження |
| Стать | nvarchar(255) | - | Стать |
| Категорія_складності | nvarchar(255) | - | Категорія складності |
| Дата_вступу_до_клубу | date | - | Дата вступу до клубу |
| Дата_виходу_з_клубу | date | - | Дата виходу з клубу |

Сутність «Тренер» містить інформацію про досвідчених туристів, які вже набули статусу тренерів туристичного клубу, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 2.

Таблиця 2. Атрибути сутності «Тренер»

| Назва | Тип | Зв'язок | Опис |
|------------------|-----|---------|------------------|
| Id | int | - | Первинний ключ |
| Id_Туристу | int | Турист | Турист |
| Тренерський_стаж | int | - | Тренерський стаж |

Сутність «Спортсмен» призначена для зберігання інформації про набути туристами спортивні звання, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 3.

Таблиця 3. Атрибути сутності «Спортсмен»

| Назва | Тип | Зв'язок | Опис |
|------------------|---------------|---------|------------------|
| Id | int | - | Первинний ключ |
| Id_Туристу | int | Турист | Турист |
| Спортивне_звання | nvarchar(255) | - | Спортивне звання |

Сутність «Аматор» містить інформацію про не професіональних спортсменів туристичного клубу, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 4.

Таблиця 4. Атрибути сутності «Аматор»

| Назва | Тип | Зв'язок | Опис |
|------------|-----|---------|----------------|
| Id | int | - | Первинний ключ |
| Id_Туристу | int | Турист | Турист |

Сутність «Секція» містить інформацію про структуру секцій, що є в туристичному клубі, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 5.

Таблиця 5. Атрибути сутності «Секція»

| Назва | Тип | Зв'язок | Опис |
|----------------------------|---------------|---------|----------------|
| Id | int | - | Первинний ключ |
| Назва_секції | nvarchar(255) | - | Назва секції |
| Id_Керівника | int | Турист | Керівник |
| Дата_призначення_керівника | date | - | Спеціалізація |

Сутність «Група» містить інформацію про групи, що входять до різних секцій з метою організації різних типів походів, що організуються в туристичному клубі, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 6.

Таблиця 6. Атрибути сутності «Група»

| Назва | Тип | Зв'язок | Опис |
|---------------|---------------|---------|----------------|
| Id | int | - | Первинний ключ |
| Назва_секції | nvarchar(255) | - | Назва секції |
| Id_Секції | int | Секція | Секція |
| Спеціалізація | date | - | Спеціалізація |

Сутність «Розклад_роботи_секції» містить інформацію про розклад роботи секцій, в якому вказується, які тренування будуть проводитися і в яких секціях: їх кількість, місце, час і т.д. У відповідності з цим керівники секцій здійснюють розподіл навантаження для тренерів (з урахуванням їх спеціальності). Відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 7.

Таблиця 7. Атрибути сутності «Розклад_роботи_секції»

| Назва | Тип | Зв'язок | Опис |
|-------------------|---------------|---------|-------------------|
| Id | int | - | Первинний ключ |
| Назва_занягть | nvarchar(255) | - | Назва секції |
| Id_Тренеру | int | Турист | Тренер |
| Id_Секції | int | Секція | Секція |
| Кількість_занягть | int | - | Кількість занягть |

Сутність «Тренування» містить інформацію про тренування, які будуть проводитися в секціях туристичного клубу та в який час, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 8.

Таблиця 8. Атрибути сутності «Тренування»

| Назва | Тип | Зв'язок | Опис |
|----------------------------|------|-----------------------|-------------------------|
| Id | int | - | Первинний ключ |
| Id_Назви_тренування | int | Розклад_роботи_секції | Тренування |
| Дата_проведення_тренування | date | - | Дата проведення занягтя |

Сутність «Відвідування_тренувань» містить інформацію про відвідування проведених тренувань, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 9.

Таблиця 9. Атрибути сутності «Відвідування_тренувань»

| Назва | Тип | Зв'язок | Опис |
|----------------|-----|------------|----------------|
| Id | int | - | Первинний ключ |
| Id_Тренування | int | Тренування | Тренування |
| Id_Відвідувача | int | Турист | Відвідувач |

Сутність «Учасники_групи» містить інформацію про час вступу та перебування учасниками груп в секціях туристичного комплексу, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 10.

Таблиця 10. Атрибути сутності «Учасники_групи»

| Назва | Тип | Зв'язок | Опис |
|----------------------|------|---------|----------------------|
| Id | int | - | Первинний ключ |
| Id_Учасника | int | Турист | Учасник групи |
| Id_Групи | int | Група | Група |
| Дата_вступу_до_групи | date | - | Дата вступу до групи |
| Дата_виходу_з_групи | date | - | Дата виходу з групи |

Сутність «Змагання» містить інформацію про всі змагання, що організовуються туристичним клубом, відображені в базі даних відповідною таблицею, містять атрибути, зазначені в таблиці 11.

Таблиця 11. Атрибути сутності «Змагання»

| Назва | Тип | Зв'язок | Опис |
|-----------------|---------------|---------|-----------------|
| Id | Int | - | Первинний ключ |
| Назва_змагання | nvarchar(255) | - | Назва змагання |
| Дата_проведення | Date | - | Дата проведення |

Сутність «Учасники_змагання» містить інформацію про тих туристів, що приймають участь в усіх змагання, які організуються туристичним клубом, відображені в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 12.

Таблиця 12. Атрибути сутності «Учасники_змагання»

| Назва | Тип | Зв'язок | Опис |
|----------------------|-----|----------|-----------------|
| Id | int | - | Первинний ключ |
| Id_Змагання | int | Змагання | Змагання |
| Id_Учасника_змагання | int | Турист | Учасник змагань |

Сутності «Маршрути» та «Точки_маршруту» містять інформацію про тривалість туристичних маршрутів, що пропонує туристична компанія, та дані про стоянки, що є на протязі туристичного маршруту, відображені в базі даних відповідними таблицями, містять атрибути, зазначені в таблицях 13 та 14 відповідно.

Таблиця 13. Атрибути сутності «Маршрути»

| Назва | Тип | Зв'язок | Опис |
|---------------------------|---------------|---------|----------------------|
| Id | int | - | Первинний ключ |
| Складність | nvarchar(255) | - | Складність |
| Довжина_маршруту_у_метрах | int | - | Довжина маршруту (м) |
| Кількість_днів | int | - | Кількість днів |
| Тип | nvarchar(255) | - | Тип |

Таблиця 14. Атрибути сутності «Точки_маршруту»

| Назва | Тип | Зв'язок | Опис |
|------------------------|---------------|----------|------------------------|
| Id | int | - | Первинний ключ |
| Id_Маршруту | int | Маршрути | Маршрут |
| Номер_точки_у_маршруті | int | - | Номер точки у маршруті |
| Координати_точки | nvarchar(255) | - | Координати точки |

Сутність «Щоденник» містить повну інформацію про планові походи, яка відображається у відповідних щоденниках, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 15.

Таблиця 15. Атрибути сутності «Щоденник»

| Назва | Тип | Зв'язок | Опис |
|---------------------------------|---------------|---------|---------------------------------|
| Id | int | - | Первинний ключ |
| Id_ Походу | int | Похід | Похід |
| Порядковий_номер_події_у_поході | int | - | Порядковий номер події у поході |
| Назва_події | nvarchar(255) | - | Назва події |
| Опис_події | nvarchar(255) | - | Опис події |
| Дата_та_час | datetime | - | Дата та час події |

Сутність «Похід» містить інформацію про походи, що протягом року організовує туристичний клуб, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 16.

Таблиця 16. Атрибути сутності «Похід»

| Назва | Тип | Зв'язок | Опис |
|------------------------|----------|----------|------------------------|
| Id | int | - | Первинний ключ |
| Дата_початку_походу | datetime | - | Дата початку походу |
| Дата_закінчення_походу | datetime | - | Дата закінчення походу |
| Id_Маршруту | int | Маршрути | Маршрут |
| Інструктор | int | Турист | Інструктор |
| Плановість | bit | - | Плановість |

Сутність «Група_походу» містить інформацію про склад кожної групи, що сформована для походу, виходячи з типу походу і фізичних даних туристів, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 17.

Таблиця 17. Атрибути сутності «Група_походу»

| Назва | Тип | Зв'язок | Опис |
|----------------|-----|---------|----------------|
| Id | int | - | Первинний ключ |
| Id_Походу | int | Похід | Похід |
| Учасник_походу | int | Турист | Учасник походу |

Окремо виділена детальна інформація про всі привали та стоянки протягом побудованого маршруту, інформація відображена в базі даних відповідними таблицями, містять атрибути, зазначені в таблицях 18 та 19 відповідно.

Таблиця 18. Атрибути сутності «Розклад_привалів»

| Назва | Тип | Зв'язок | Опис |
|-----------------------------------|----------|----------------|-----------------------------------|
| Id | int | - | Первинний ключ |
| Id_Маршруту | int | Маршрути | Маршрут |
| Порядковий_номер_привалу_у_поході | int | - | Порядковий номер привалу у поході |
| Точка_привалу | int | Точки_маршруту | Точка привалу |
| Дата_та_час_початку_привалу | datetime | - | Дата та час початку привалу |
| Дата_та_час_закінчення_привалу | datetime | - | Дата та час закінчення привалу |

Таблиця 19. Атрибути сутності «Розклад_стоянок»

| Назва | Тип | Зв'язок | Опис |
|-----------------------------------|----------|----------------|-----------------------------------|
| Id | int | - | Первинний ключ |
| Id_Маршруту | int | Маршрути | Маршрут |
| Порядковий_номер_стоянки_у_поході | int | - | Порядковий номер стоянки у поході |
| Точка_стоянки | int | Точки_маршруту | Точка стоянки |
| Дата_та_час_початку_стоянки | datetime | - | Дата та час початку стоянки |
| Дата_та_час_закінчення_стоянки | datetime | - | Дата та час закінчення стоянки |

3.5 Програмна реалізація застосунку

Розглянемо структуру розробленої інформаційної системи. У підпапці *Forms* лежать класи, які використовуються для додавання запису до таблиці або до зміни існуючої (логіка всіх класів ідентична). Відповідно, у кожному класі два конструктори: конструктор із параметрами використовується для зміни запису та конструктор без параметрів, використовується для створення нового запису. З бази даних копіюється модель поточного запису і виводиться на форму (для створення нового запису всі поля моделі створюються порожніми). Далі користувач може внести зміни. Кнопка *Save* перевіряє їхню наявність. Якщо зміни є – він оновлює ці дані в моделі, після чого передає інформацію до класу *TouristDB* (описаний нижче) для зміни БД.

У підпапці *Procedures* лежать класи до виконання запитів (логіка роботи всіх класів ідентична). Створюються порожні поля. Після натискання на кнопку *Save* відбувається аналіз наявності порожніх полів (вони не беруть участь у запиті), після чого виконується пошук відповідних записів у БД.

Підпапка *HeadDBClasses* містить три файли.

ProceduresEnum – зберігає у собі числові відповідності запитам.

TablesTypesEnum – зберігає числові відповідності таблицям.

TouristDB – є найбільшим основним класом. Всі методи можна поділити на унікальні та повторні. До унікальних відносяться: *DeleteRows* (відповідає за видалення запису з таблиці) та *LoadTable* (відповідає за завантаження обраної користувачем таблиці).

Повторні методи діляться на чотири підтипи, у якості приклада розглянемо *Аматора*).

- *GetАматори()* – необхідний для отримання таблиці з бази даних (саме ці методи викликаються у функції *LoadTable*).
- *GetАматори(int id)* – необхідний для отримання конкретної записи з таблиці (пошук по *id*).

- *AddАматор(Аматор amator)* – отримує інформацію з вікна додавання (клас, який лежить у підпапці Forms і описаний вище) і додає до БД, виконуючи перевірку на зв'язаність таблиць, якщо це необхідно (наприклад, що Аматору відповідає існуючий запис у таблиці Турист).

- *UpdateАматору(Аматор amator)* – принцип дії аналогічний додаванню з тією різницею, що ми вказуємо id записи зміни.

Клас *Auth* відповідає за авторизацію користувача у системі. Конструктор виконує зв'язування програми з БД, а натискання на кнопку входу запускає перевірку відповідності введених користувачем логіну та паролю із записом у таблиці.

Клас *Form1* відповідає за основну форму. При подвійному кліку запису в таблиці відкривається вікно для зміни (метод *dataGridView1_CellDoubleClick*). При натисканні кнопки додавання запису викликається відповідна форма для створення нового запису (метод *buttonAdd_Click*). При натисканні кнопки видалення виконується видалення запису (*buttonDelete_Click*). Перелічені методи виконують лише виклик необхідного методу, пов'язаного з активною таблицею (тобто, викликає необхідний саме цій таблиці метод класу *TouristDB*).

Крім цього є методи завантаження таблиць (*LoadTables (int currentItem)*) і виконання запитів (*LoadProcedure (int currentItem)*), які викликаються за допомогою вибору необхідного пункту в списку (кожен пункт випадаючого меню має своє число, яке відповідає номеру таблиці або запиту з класів *TablesTypesEnum* і *ProceduresEnum*, що зазначені вище).

В відповідності з архітектурою система підрозділяється на дві частини. Серверна частина являє собою базу даних під керуванням СУБД MS SQL.

Після створення бази даних необхідно створити таблиці для зазначених вище сутностей за допомогою відповідного скрипта (див. Лістинг 2).

Лістинг 2 – Створення таблиць

```
create table Турист (
```

```
    Id int identity(1,1) primary key,  
    Прізвище nvarchar(255),  
    Імя nvarchar(255),  
    Побатькові nvarchar(255),  
    Дата_народження date,  
    Стать nvarchar(255),  
    Категорія_складності nvarchar(255),  
    Дата_вступу_до_клубу date,  
    Дата_виходу_з_клубу date  
)  
create table Секція (  
    Id int identity(1,1) primary key,  
    Назва_секції nvarchar(255),  
    Id_Керівника int foreign key references Турист,  
    Дата_призначення_керівника date  
)  
create table Група (  
    Id int identity(1,1) primary key,  
    Назва_секції nvarchar(255),  
    Id_Секції int foreign key references Секція,  
    Спеціалізація nvarchar(255)  
)  
create table Розклад_роботи_секції (  
    Id int identity(1,1) primary key,  
    Назва_занять nvarchar(255),  
    Id_Тренеру int foreign key references Турист,  
    Id_Секції int foreign key references Секція,  
    Кількість_занять int  
)  
create table Змагання (  
    Id int identity(1,1) primary key,  
    Назва_змагання nvarchar(255),
```

```

        Дата_проведення date
    )
create table Учасники_змагань (
    Id int identity(1,1) primary key,
    Id_Змагання int foreign key references Змагання,
    Id_Учасника_змагання int foreign key references
Турист
    )
create table Маршрути (
    Id int identity(1,1) primary key,
    Складність nvarchar(255),
    Довжина_маршруту_у_метрах int,
    Кількість_днів int,
    Тип nvarchar(255)
    )
create table Точки_маршруту (
    Id int identity(1,1) primary key,
    Id_Маршруту int foreign key references Маршрути,
    Номер_точки_у_маршруті int,
    Координати_точки nvarchar(255)
    )
create table Похід (
    Id int identity(1,1) primary key,
    Дата_початку_походу datetime,
    Дата_закінчення_походу datetime,
    Id_Маршруту int foreign key references Маршрути,
    Інструктор int foreign key references Турист,
    Плановість bit
    )
create table Щоденник (
    Id int identity(1,1) primary key,
    Id_Походу int foreign key references Похід,

```

```

        Порядковий_номер_події_у_поході int,
        Назва_події nvarchar(255),
        Опис_події nvarchar(255),
        Дата_та_час datetime
    )
create table Розклад_привалів (
    Id int identity(1,1) primary key,
    Id_Маршруту int foreign key references Маршрути,
    Порядковий_номер_привалу_у_поході int,
    Точка_привалу int foreign key references
Точки_маршруту,
    Дата_та_час_початку_привалу datetime,
    Дата_та_час_закінчення_привалу datetime
)

```

Після створення таблиць їх необхідно заповнити за допомогою відповідного скрипта, частина якого наведена у Лістингу 3.

Лістинг 3 – Заповнення таблиць

```

insert into Секція
values
    (N'Плавання', 15, '2020-02-01'),
    (N'Альпінізм', 16, '2020-02-01'),
    (N'Спелеологія', 17, '2020-02-01');

insert into Група
values
    (N'20.1-плв', 1, N'плавання'),
    (N'20.1-алп', 2, N'альпінізм'),
    (N'20.1-спл', 3, N'спеліологія');

insert into Розклад_роботи_секції

```

values

```
(N'Плавання "Кролем"', 1, 1, 3),
(N'Плавання "Брасом"', 1, 1, 2),
(N'Техніка подолання скелі', 6, 2, 3),
(N'Техніка подолання снігових барханів', 7, 2, 2),
(N'Вивчення техніки для спуску/підйому', 11, 3, 5),
(N'Вивчення методіки страхування', 11, 3, 4);
```

Коректність наведеного коду перевіряється за допомогою select-запиту до відповідної таблиці (див. Рис. 19).

| Id | Прізвище | Імя | Побачкові | Дата_народження | Стать | Категорія_складності | Дата_вступу_до_клубу | Дата_виходу_з_клубу |
|----|--------------|-------------|---------------|-----------------|----------|----------------------|----------------------|---------------------|
| 1 | Волкова | Мілана | Андріана | 1991-10-21 | жіноча | вища | 2020-10-10 | NULL |
| 2 | Корольова | Аліса | Микитіна | 2000-11-27 | жіноча | початкова | 2020-10-20 | NULL |
| 3 | Ситніков | Іван | Олександрович | 2001-01-28 | чоловіча | початкова | 2020-11-20 | NULL |
| 4 | Васильєва | Анастасія | Тимофеевна | 2000-10-16 | жіноча | початкова | 2020-08-20 | NULL |
| 5 | Федосєєва | Микита | Єгорович | 2001-04-11 | чоловіча | початкова | 2021-03-14 | NULL |
| 6 | Іванов | Іван | Владимирович | 1976-08-04 | чоловіча | вища | 2020-08-20 | NULL |
| 7 | Васильєв | Олег | Алійович | 2000-03-10 | чоловіча | початкова | 2020-08-20 | NULL |
| 8 | Андрєєв | Дмитро | Олександрович | 2001-08-23 | чоловіча | початкова | 2019-01-23 | NULL |
| 9 | Сидоренко | Андрій | Владиславович | 2001-01-18 | чоловіча | початкова | 2020-09-20 | NULL |
| 10 | Касаткіна | Данііл | Костянтинівич | 2001-01-21 | жіноча | початкова | 2020-08-20 | NULL |
| 11 | Ларісова | Євеліна | Андріана | 1985-05-03 | жіноча | вища | 2020-11-13 | NULL |
| 12 | Сміров | Владислав | Сергієвич | 2001-07-06 | чоловіча | початкова | 2020-02-20 | NULL |
| 13 | Карєніна | Олена | Анатоліівна | 2000-08-10 | жіноча | початкова | 2020-04-19 | NULL |
| 14 | Пірожніченко | Сергій | Матиславович | 2001-01-03 | чоловіча | початкова | 2020-06-10 | NULL |
| 15 | Симоненко | Пантелеймон | Сергієвич | 1972-07-31 | чоловіча | вища | 2020-01-19 | NULL |
| 16 | Тухачєв | Спіридон | Андрійович | 1976-09-17 | чоловіча | початкова | 2020-01-20 | NULL |
| 17 | Полєхасєв | Андрій | Володимирович | 1978-06-30 | чоловіча | початкова | 2020-01-21 | NULL |
| 18 | Лежньова | Дарина | Владиславівна | 2000-10-14 | жіноча | початкова | 2021-10-01 | NULL |

Рис. 19 Результат запиту до таблиці «Турист»

Основною частиною програмної реалізації за стосунку є створення збережених процедур і табличних функцій. Наведемо у якості приклада деякі з них (див. Лістинги 4–8).

Лістинг 4 – Список туристів із зазначеної секції, групи, що ходили в походи зі своїм тренером в якості інструктора

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE FUNCTION twelfth_getTrainerByParams (
```

```

        @SectionId int= null, @GroupId int= null,
        @TrainerInstructor int = null
    )
    RETURNS TABLE
    AS
    RETURN (
        SELECT [Турист].Id, Прізвище, Імя, Побатькові,
        Дата_народження, Стать, Категорія_складності,
        Дата_вступу_до_клубу, Дата_виходу_з_клубу
        FROM [Турист]
        JOIN [Учасники_групи] ON [Турист].Id =
        [Учасники_групи].Id_Учасника
        JOIN [Група] ON [Учасники_групи].Id_Групи =
        Група.Id
        JOIN Група_походу ON Турист.Id =
        Група_походу.Учасник_походу
        JOIN Похід ON Група_походу.Id_Походу =
        Похід.Id
        WHERE ((@SectionId is null) or ((@SectionId is
        not null) and (@SectionId > 0) and @SectionId =
        [Група].Id_Секції))
        AND ((@GroupId is null) or ((@GroupId is not
        null) and (@GroupId > 0) and [Учасники_групи].Id_Групи =
        @GroupId))
        AND ((@TrainerInstructor is null) or
        ((@TrainerInstructor is not null) and EXISTS (
            SELECT
            Відвідування_тренувань.Id_Відвідувача
            FROM Відвідування_тренувань
            JOIN Тренування ON
            Відвідування_тренувань.Id_Тренування = Тренування.Id

```



```

        JOIN Розклад_роботи_секції ON
Тренування.Id_Назви_тренування = Розклад_роботи_секції.Id
        WHERE Похід.Інструктор =
Розклад_роботи_секції.Id_Тренеру)))
GO

SELECT * FROM twelfth_getTrainerByParams(null, null,
null);
GO

```

**Лістинг 5 – Список туристів з деякою секції, групи,
що ходили по всіх маршрутах, по зазначених маршрутах**

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE FUNCTION thirteenth_getTrainerByParams (
    @SectionId int= null, @GroupId int= null,
    @RouteId int = null
)
RETURNS TABLE
AS
RETURN (
    SELECT DISTINCT [Турист].Id, Прізвище, Імя,
Побатькові, Дата_народження, Стать, Категорія_складності,
Дата_вступу_до_клубу, Дата_виходу_з_клубу
    FROM [Турист]
    JOIN [Учасники_групи] ON [Турист].Id =
[Учасники_групи].Id_Учасника
    JOIN [Група] ON [Учасники_групи].Id_Групи =
Група.Id

```

```

        JOIN Група_походу ON Турист.Id =
Група_походу.Учасник_походу
        JOIN Похід ON Група_походу.Id_Походу =
Похід.Id
        JOIN Маршрути ON Похід.Id_Маршруту =
Маршрути.Id
        WHERE ((@SectionId is null) or ((@SectionId is
not null) and (@SectionId > 0) and @SectionId =
[Група].Id_Секції))
        AND ((@GroupId is null) or ((@GroupId is not
null) and (@GroupId > 0) and [Учасники_групи].Id_Групи =
@GroupId))
        AND (((@RouteId is not null) and (@RouteId >
0) and Похід.Id_Маршруту = @RouteId) or
        ((@RouteId is null) and Турист.Id = ANY(

        SELECT [Турист].Id
        FROM [Турист]
        JOIN Група_походу ON Турист.Id =
Група_походу.Учасник_походу
        JOIN Похід ON Група_походу.Id_Походу =
Похід.Id
        JOIN Маршрути ON Похід.Id_Маршруту =
Маршрути.Id
        GROUP BY [Турист].Id
        HAVING COUNT([Турист].Id) = (select
count(*) from Маршрути))))
    )
GO

SELECT * FROM thirteenth_getTrainerByParams(null, null,
2);

```

GO

Лістинг 6 – Перелік і загальна кількість інструкторів, інструкторів-спортсменів, інструкторів-тренерів, що мають певну категорію, що ходили в зазначену кількість походів, ходили в певний похід, ходили по деякому маршруту, були у зазначеній точці

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE FUNCTION eleventh_getTrainerByParams (
    @SportsmanInstructor int = null,
    @TrainerInstructor int = null, @Category nvarchar(255) =
    null,
    @QuantitiOfTrips int = null, @TripId int= null,
    @RouteId int = null, @Point nvarchar(255) = null
)
RETURNS TABLE
AS
RETURN (
    SELECT DISTINCT [Турист].Id, Прізвище, Імя,
    Побатькові, Дата_народження, Стать, Категорія_складності,
    Дата_вступу_до_клубу, Дата_виходу_з_клубу
    FROM [Турист]
    JOIN Похід ON [Турист].Id = Похід.Інструктор
    JOIN Точки_маршруту ON Похід.Id_Маршруту =
    Точки_маршруту.Id_Маршруту
    WHERE ((@SportsmanInstructor is null) or
    ((@SportsmanInstructor is not null) and EXISTS (
        SELECT Спортсмен.Id_Туристу
        FROM Спортсмен

```

```

WHERE Спортсмен.Id_Туристу = Турист.Id))
AND ((@TrainerInstructor is null) or
((@TrainerInstructor is not null) and EXISTS (
SELECT Тренер.Id_Туристу
FROM Тренер
WHERE Тренер.Id_Туристу = Турист.Id))
AND ((@Category is null) or ((@Category is not
null) and [Турист].Категорія_складності = @Category))
AND ((@QuantitiOfTrips is null) or
((@QuantitiOfTrips is not null) and (@QuantitiOfTrips > 0)
and EXISTS (
SELECT Похід.Інструктор,
COUNT(Похід.Інструктор)
FROM Похід
WHERE Похід.Інструктор = Турист.Id
GROUP BY Похід.Інструктор
HAVING COUNT(Похід.Інструктор) =
@QuantitiOfTrips)))
AND ((@TripId is null) or ((@TripId is not null)
and (@TripId > 0) and Похід.Id = @TripId))
AND ((@RouteId is null) or ((@RouteId is not
null) and (@RouteId > 0) and Похід.Id_Маршруту = @RouteId))

AND ((@Point is null) or ((@Point is not null)
and Точки_маршруту.Координати_точки = @Point))
GO

SELECT * FROM eleventh_getTrainerByParams(null, null,
null, null, null, null, null);
GO

```

Лістинг 7 – Перелік і загальну кількість маршрутів, що проходять через деяку точку, мають довжину більше вказаної, можуть задовольняти заданій категорії складності

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE FUNCTION ninth_getTrainerByParams (
    @Point nvarchar(255) = null, @Length int = null,
@Category nvarchar(255) = null
)
RETURNS TABLE
AS
RETURN (
    SELECT DISTINCT Маршрути.Id,
Маршрути.Кількість_днів, Маршрути.Складність, Маршрути.Тип,
Маршрути.Довжина_маршруту_у_метрах
    FROM Маршрути
    JOIN Точки_маршруту ON Маршрути.Id =
Точки_маршруту.Id_Маршруту
    WHERE ((@Length is null) or ((@Length is not
null) and (@Length > 0) and @Length <
Маршрути.Довжина_маршруту_у_метрах))
    AND ((@Category is null) or ((@Category is not
null) and Маршрути.Складність = @Category))
    AND ((@Point is null) or ((@Point is not null)
and Точки_маршруту.Координати_точки = @Point))
)
GO

```

```

SELECT * FROM ninth_getTrainerByParams(null, null,
null);
GO

```

Лістинг 8 – Навантаження тренерів, його обсяг із конкретних видів занять і загальне навантаження за вказаний період часу для даного тренера або зазначеної секції

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE seventh_getTrainerByParams
    @SectionId int= null, @TrainerId int= null,
    @FromDate date= null, @UntilDate date= null
AS
BEGIN
    SET NOCOUNT ON;
    SELECT DISTINCT [Турист].Id,
Розклад_роботи_секції.Id_Секції,
Розклад_роботи_секції.Кількість_занять,
Розклад_роботи_секції.Назва_занять, Прізвище, Імя,
Побатькові, Дата_народження, Стать, Категорія_складності,
Дата_вступу_до_клубу, Дата_виходу_з_клубу
    FROM [Турист]
    JOIN Розклад_роботи_секції ON Турист.Id =
Розклад_роботи_секції.Id_Тренеру
    JOIN Тренування ON Розклад_роботи_секції.Id =
Тренування.Id_Назви_тренування
    WHERE ((@SectionId is null) or ((@SectionId is
not null) and (@SectionId > 0) and @SectionId =
Розклад_роботи_секції.Id_Секції))

```

```
AND ((@TrainerId is null) or ((@TrainerId is
not null) and (@TrainerId > 0) and
Розклад_роботи_секції.Id_Тренеру = @TrainerId))
AND ((@FromDate is null) or ((@FromDate is not
null) and Тренування.Дата_проведення_тренування >=
@FromDate))
AND ((@UntilDate is null) or ((@UntilDate is
not null) and Тренування.Дата_проведення_тренування <=
@UntilDate))
END
GO
exec seventh_getTrainerByParams @FromDate = '2021-10-
02'
GO
```

3.6 Розробка інтерфейсу інформаційної системи

Відповідно до функціональних вимог була розроблена клієнтська програма, де реалізовані всі зазначені функції.

Авторизація контролює доступ легальних користувачів до ресурсів системи після успішного проходження ними аутентифікації (див. Рис. 20).

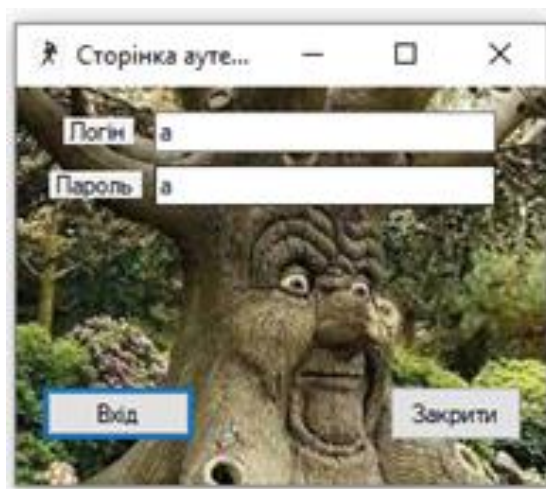


Рис. 20 Вікно аутентифікації

Після проходження якої користувачу системи відкривається вікно інформаційної системи туристичного клубу (див. Рис. 21).

| Id | Прізвище | Імя | Побатькові | Дата_народж_ | Стать | Категорія_скл_ | Дата_вступу_ | Дата_виходу_ |
|----|-----------|-----------|---------------|--------------|----------|----------------|--------------|--------------|
| 1 | Волкова | Мілана | Андріївна | 21.10.1991 | жіноча | вища | 10.10.2020 | |
| 2 | Корольова | Аліса | Микитічна | 27.11.2000 | жіноча | початкова | 20.10.2020 | |
| 3 | Ситніков | Іван | Олександро... | 28.01.2001 | чоловіча | початкова | 20.11.2020 | |
| 4 | Васильєва | Анастасія | Тимофеевна | 16.10.2000 | жіноча | початкова | 20.08.2020 | |
| 5 | Федосєєв | Микита | Єгорович | 11.04.2001 | чоловіча | початкова | 14.03.2021 | |
| 6 | Іванов | Іван | Владимиро... | 04.08.1976 | чоловіча | вища | 20.08.2020 | |
| 7 | Васильєв | Олег | Алійович | 10.03.2000 | чоловіча | початкова | 20.08.2020 | |

Рис. 21 Вікно з записами таблиці

Користувач, залежно від прав доступу, може вибрати таблиці, які відкриються у поточному вікні (див. Рис. 22).

| Імя | Побатькові | Дата_народж_ | Стать | Категорія_скл_ | Дата_вступу_ | Дата_виходу_ |
|-----------|---------------|--------------|----------|----------------|--------------|--------------|
| Мілана | Андріївна | 21.10.1991 | жіноча | вища | 10.10.2020 | |
| Аліса | Микитічна | 27.11.2000 | жіноча | початкова | 20.10.2020 | |
| Іван | Олександро... | 28.01.2001 | чоловіча | початкова | 20.11.2020 | |
| Анастасія | Тимофеевна | 16.10.2000 | жіноча | початкова | 20.08.2020 | |
| Микита | Єгорович | 11.04.2001 | чоловіча | початкова | 14.03.2021 | |
| Іван | Владимиро... | 04.08.1976 | чоловіча | вища | 20.08.2020 | |
| Олег | Алійович | 10.03.2000 | чоловіча | початкова | 20.08.2020 | |

Рис. 22 Вікно вибору таблиць

Користувач має можливість змінити запис через подвійний клік по запису (див. Рис. 23), а також додати новий або видалити існуючий (див. Рис. 24).

Турист

Id: 1

Прізвище: Волкова

Ім'я: Мілана

По батькові: Андріївна

Дата народження: 21 октября 1991 г.

Стать: жіноча

Категорія складності: вища

Дата вступу до клубу: 10 октября 2020 г.

Дата виходу з клубу: 31 декабря 9998 г.

Іконки: Зберегти, Оновити

Рис. 23 Вікно зі змінами запису таблиці

Інформаційна система туристичного клубу (Турист)

Обрати іншу таблицю Виконати Запит Вийти з профілю

Додати запис до таблиці Видалити обраний запис

| Id | Прізвище | Ім'я | Побатькові | Дата_народж | Стать | Категорія_скл | Дата_вступу_ | Дата_виходу_ |
|----|--------------|-------------|---------------|-------------|----------|---------------|--------------|--------------|
| 13 | Кареніна | Олена | Анатоліївна | 10.08.2000 | жіноча | початкова | 19.04.2020 | |
| 14 | Пирожниче... | Сергій | Мстиславов... | 03.01.2001 | чоловіча | початкова | 10.06.2020 | |
| 15 | Симоненко | Пантелеймон | Сергійович | 31.07.1972 | чоловіча | вища | 19.01.2020 | |
| 16 | Тухачев | Спіридон | Андрійович | 17.09.1976 | чоловіча | початкова | 20.01.2020 | |
| 17 | Полежаєв | Андрій | Володимир... | 30.06.1978 | чоловіча | початкова | 21.01.2020 | |
| 18 | Пежньова | Дарина | Владислав... | 14.10.2000 | жіноча | початкова | 01.10.2021 | |

Видалення

⚠ Ви дійсно бажаєте видалити обраний запис?

Да Нет

Рис. 24 Вікно з видаленням запису з таблиці

Але, якщо запис у таблиці пов'язаний із чимось – видалення буде неможливим (див. Рис. 25).

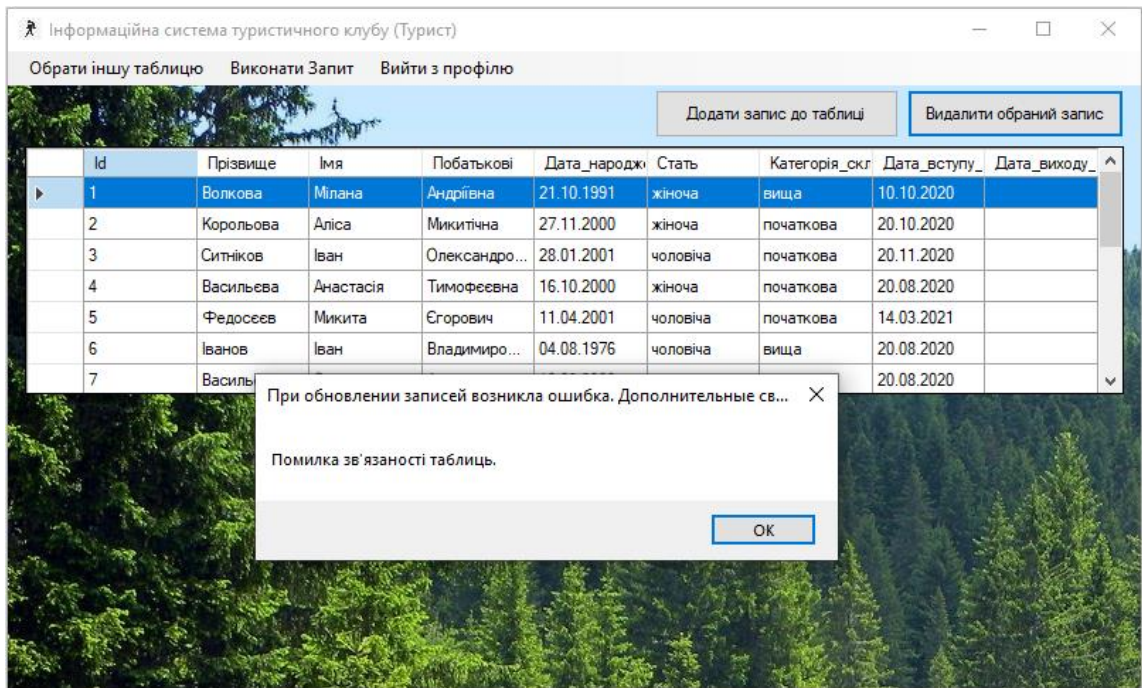


Рис. 25 Вікно з помилкою видалення запису

Крім таблиць, користувач може ознайомитись з інформацією за допомогою запитів (див. Рис. 26).

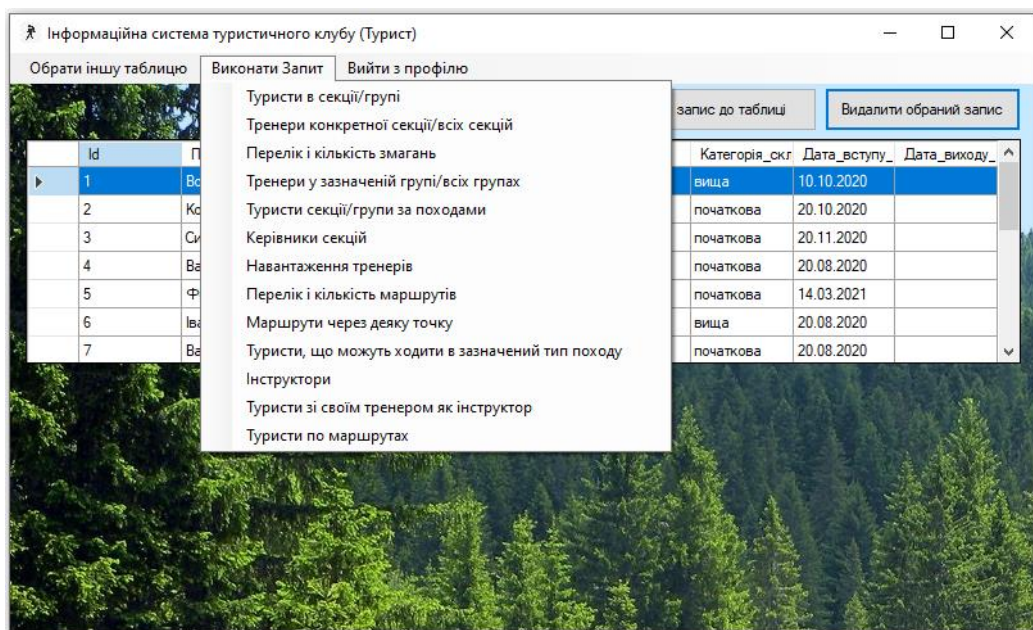


Рис. 26 Вікно вибору запиту

При цьому кожний запит має свій набір параметрів пошуку (див. Рис. 27).

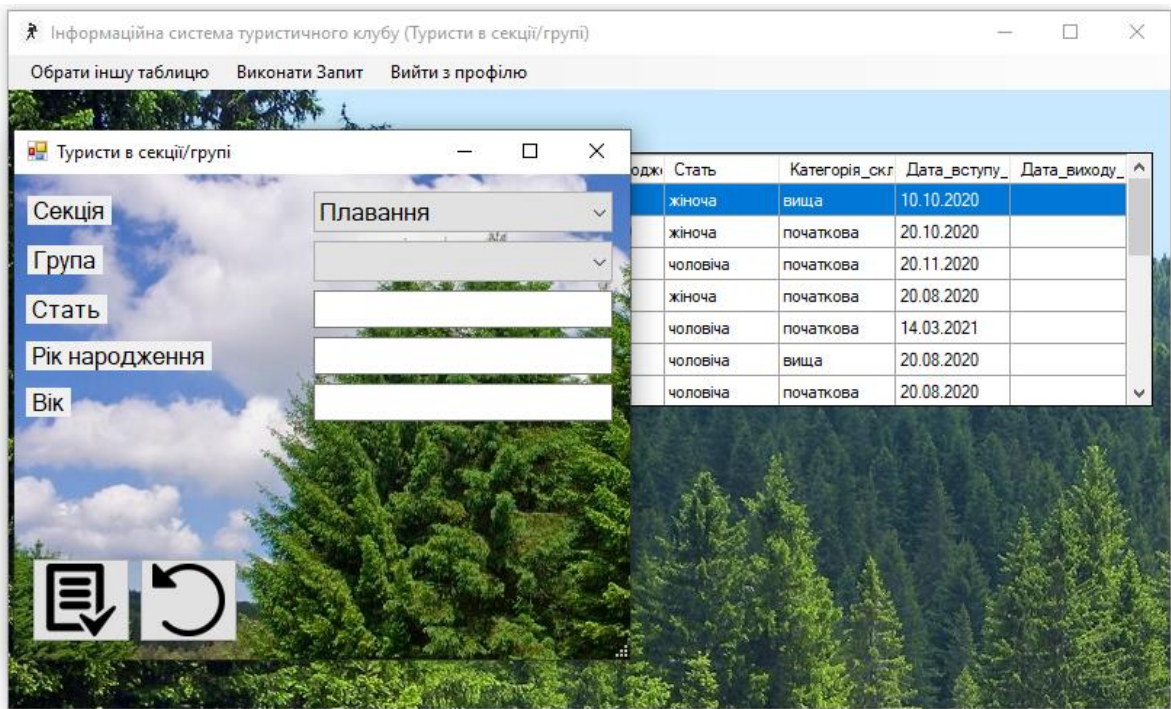


Рис. 27 Вікно виконання запиту

Після виконання запиту відобразяться записи, які відповідають обраному критерію (див. Рис. 28).

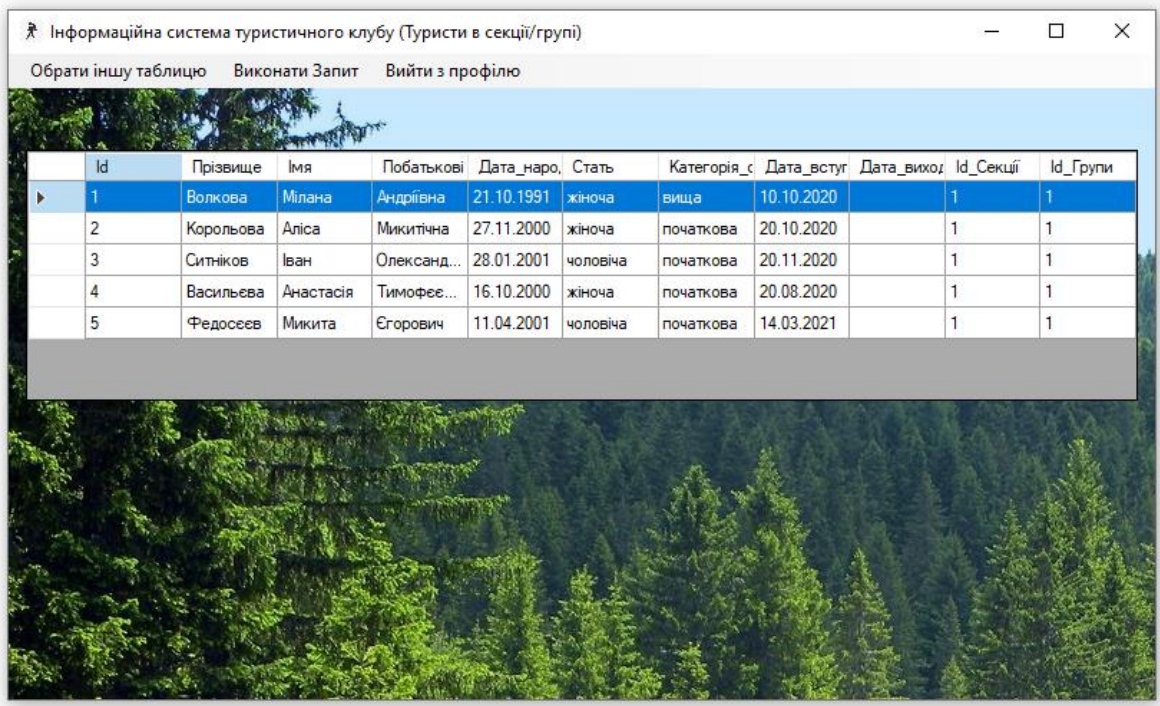


Рис. 28 Вікно інформації із запиту

ВИСНОВКИ

1. У ході реалізації кваліфікаційної роботи було опрацьовано літературні джерела за темою розробки інтелектуальних інформаційних систем та проаналізовано продукти аналогів, які надають підтримку процесів діяльності туристичних комплексів.
2. Аналіз доступних наявних інформаційних матеріалів та результатів наукових досліджень, показав, що найбільша увага розробників та дослідників приділяється ІТ технологіям, зорієтованим на обслуговування туристичних напрямків та об'єктів тому, метою роботи стала розробка власного застосунку для задоволення потреб туристичного комплексу з метою підвищення ефективності його роботи.
3. Після аналізу сучасного стану питання було розроблено технічне завдання та виконане планування робіт. Перед початком реалізації було проведено проектування архітектури системи, бази даних та виділення варіантів використання. Розроблена система відповідає усім поставленим функціональним вимогам.
4. Спроектвана та створена база даних інформаційної системи туристичного клубу за допомогою MS SQL Server, а також розроблено десктопний застосунок з використанням технології Entity Framework Core для роботи з ним.
5. Оскільки, актуальність використання інформаційних технологій у галузі туризму стала предметом досліджень теоретиків і практиків тих країн, ринки яких активно розвиваються, подальшим розширенням роботи може бути створення сучасного сайту з унікальною системою підбору турів та корисною інформацією для туристів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Артеменко О. І., Пасічник В. В., Єгорова В. В. Інформаційні технології в галузі туризму. Аналіз застосувань та результатів досліджень : Вісник Національного університету "Львівська політехніка". Серія : Інформаційні системи та мережі, 2015. № 814. С. 3-22.
2. Артеменко О. І., Кунанець Н.Е., Пасічник В.В., Савчук В.В. Системні особливості сучасних ІТ в галузі туризму : Науковий вісник НЛТУ України, 2015. Вип. 25.9. С. 268-277.
3. Мельниченко С. В. Інформаційні технології в туризмі: теоретичні та практичні аспекти : Вісник Запорізького національного університету, 2010. №2(6). С. 129-138.
4. Гапоненко Г. І., Шульга Н. В. Сучасні тенденції та перспективи впровадження інформаційних технологій в туристичній галузі України : Вісник ХНУ імені В.Н. Каразіна. Серія "Міжнародні відносини. Економіка. Країнознавство. Туризм". Вип. 11, 2020. С. 111-120.
5. Jon P. Smith. Entity Framework Core in Action : Packt, 2018. 520с.
6. Entity Framework Core documentation : веб-сайт. URL: <https://docs.microsoft.com/en-us/ef/core> (дата звернення: 07.03.2023).
7. Виклюк Я. І., Гаць Б. М. Огляд сучасного стану інформаційного забезпечення функціонування туристичної галузі : Вісник Національного університету "Львівська політехніка". Серія: Інформаційні системи та мережі, №715, 2011. С. 59-68.
8. Paul DuBois. MySQL Cookbook. 3rd edition. O'Reilly, 2014. 837 p.
9. MySQL Workbench : веб-сайт. URL: <https://www.mysql.com/products/workbench> (дата звернення: 10.03.2023).
10. Метод функціонального моделювання : веб-сайт. URL: http://ni.biz.ua/3/3_20/3_20501_metod-funktsionalnogo-modelirovaniya-SADT-IDEF.html (дата звернення: 10.03.2023).

11. What is an Entity Diagram(ERD)? веб-сайт. URL: <https://medium.com/@soni.dumitru/what-is-an-entity-relationship-diagram-erd-13daee5b2a>.

12. Мулеса О. Ю. Інформаційні системи та реляційні бази даних : Ужгород, 2018. 118 с.

13. Top 10 Programming Languages for Desktop Apps In 2021. Decipherzone : веб-сайт. URL: <https://www.decipherzone.com/blog-detail/top-programming-languages-for-desktop-apps-in-2021> (дата звернення: 07.04.2023).

14. Карпенко М. Ю., Манакова Н. О., Гавриленко І. О. Технології створення програмних продуктів та інформаційних систем : навч. посіб. Харків : ХНУМГ ім. О. М. Бекетова, 2017. 93 с.

15. UML Use Case Diagrams : веб-сайт. URL: <https://www.uml-diagrams.org/use-case-diagrams.html> (дата звернення: 07.04.2023).

16. Савчук В. В., Пасічник В. В Інтелектуальна система «Мобільний інформаційний асистент туриста»: функціональні та технологічні особливості : Вісник Національного університету "Львівська політехніка" : Інформаційні системи та мережі. № 832, 2015. С. 225-241.