

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки
Факультет комп'ютерних наук та технологій
(повне найменування інституту, факультету)

Кафедра комп'ютерних систем та мереж
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(ступінь вищої освіти (освітній ступінь))

на тему РОЗРОБКА ВЕБ СИСТЕМИ
ТОРГОВО-РОЗВАЖАЛЬНОГО ЦЕНТРУ

Виконав: студент 4 курсу, групи КНТ-517
спеціальності _____

123 «Комп'ютерна інженерія»

(код і найменування спеціальності)

Освітня програма (спеціалізація) _____

«Комп'ютерна інженерія»

Куцак Сергій Олександрович

(прізвище та ініціали)

Керівник Киричек Г.Г.

(прізвище та ініціали)

Рецензент Фурманова Н.І.

(прізвище та ініціали)

м. Запоріжжя
2021 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування вищого навчального закладу)

Інститут, факультет інформатики та радіоелектроніки, комп'ютерних наук і технологій
Кафедра «Комп'ютерні системи та мережі»
Ступінь вищої освіти (освітній ступінь) бакалаврський
Спеціальність 123 Комп'ютерна інженерія
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерна інженерія
(назва освітньої програми (спеціалізації))


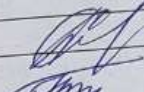
ЗАТВЕРДЖУЮ
Завідувач кафедри Кудерметов Р.К.
« » 2021 року

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА

Куцаку Сергію Олександровичу
(прізвище, ім'я, по батькові)

- Тема проекту (роботи) Розробка веб системи торгово-розважального центру
- керівник проекту (роботи) Киричек Галина Григорівна, к. т. н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від «17» березня 2021 року № 81
- Строк подання студентом проекту (роботи) 06 травня 2021 року
- Вихідні дані до проекту (роботи) опис культурно-масових подій торгово-розважального центру, наявність робочих місць менеджерів, перелік торгових точок.
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Аналіз сучасних веб систем та технічного завдання;
 - Розробка бази даних;
 - Розробка веб сайту підприємства;
 - Оптимізація та налаштування сайту.
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
 - Діаграма використання веб системи;
 - Реляційна схема бази даних;
 - Структурна схема сайту;
 - Інтерфейс інформаційної системи.

6. Консультанти розділів проекту (роботи)

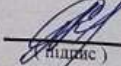
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3	Киричек Г.Г., к. т. н., доцент		
Нормо контроль	Щербак Н.В., ст.викладач		

7. Дата видачі завдання 01.03.2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз наявних методів та алгоритмів побудови веб систем, перегляд отриманих результатів, виділення недоліків та переваг	01.03.2021 р.	
2	Розробка структури системи	15.03.2021 р.	
3	Розробка бази даних	20.03.2021 р.	
4	Розробка веб-сайту	01.04.2021 р.	
5	Оптимізація та налаштування веб-системи	10.04.2021 р.	
6	Оформлення отриманих результатів у ПЗ	15.04.2021 р.	
7	Оформлення графічного матеріалу	20.04.2021 р.	
8	Оформлення допоміжного матеріалу	01.05.2021 р.	

Студент  С.О.Кущак
(підпис) (ініціали та прізвище)

Керівник проекту (роботи)  Г.Г. Киричек
(підпис) (ініціали та прізвище)

РЕФЕРАТ

ПЗ: 60 с., 30 рисунках, 12 таблиця, 12 джерел.

БАЗИ ДАНИХ, ВЕБ-САЙТ, ДІАГРАММА, ІНФОРМАЦІЙНА СИСТЕМА, КОРИСТУВАЧ, ПРОЕКТУВАННЯ, СПЕЦИФІКАЦІЯ ВИМОГ ТОРГОВО-РОЗВАЖАЛЬНИЙ ЦЕНТР

Об'єкт розробки – веб-сайт для торгово-розважального центру.

Мета роботи – розробити веб-систему торгово-розважального центру.

Предмет розробки – моделі, методи, і інструментальні та програмні засоби проектування та розробки веб-системи.

Пояснювальна записка складається з 4 розділів. Перший розділ присвячено аналізу технічного завдання. В даному розділі приведені аналіз вихідних даних веб-системи та розроблена специфікація вимог RSS. Здійснюється детальний розбір об'єкта та порівняння з іншими відомими сайтами. Другий розділ описує розробку бази даних, яка зберігає інформацію про заходи даного центру. База даних розроблена за допомогою MySQL Workbench. Третій розділ присвячено розробці веб сайту торгово-розважального центру. Веб сайт цієї системи має розширене меню і додаткові модулі. При розробці використовувалася CMS WordPress. Наведена структурна схема сайту. Описані керівництва програміста та користувача веб-сайту. Внесени корективи в код сайту. Четвертий розділ проведено оптимізацію та налаштування веб-сайту в мережі Інтернет.

ЗМІСТ

Скорочення та умовні позначки	6
Вступ.....	7
1 Аналіз технічного завдання.....	8
1.1 Аналіз вихідних даних і вимог до веб системі.....	8
1.2 Аналіз сучасних веб систем	10
1.3 Розробка специфікації вимог SRS	12
1.3.1 Вимоги до веб системи	12
1.3.2 Побудова діаграм використання сайту	16
1.3.3 Побудова діаграм послідовності сайту	20
2 Розробка бази даних.....	24
2.1 Опис предметної області	24
2.2 Реляційна модель бази даних.....	27
2.3 Розробка бази даних.....	29
3 Розробка веб сайту торгово-розважального центру	36
3.1 Структурна схема сайту.....	36
3.2 Керівництво програміста.....	38
3.2.1. Налаштування сервера.....	38
3.2.2 CMS WordPress	41
3.3 Керівництво користувача	48
4 оптимізація та налаштування сайту	49
Висновки	53
Перелік джерел посилання	54
Плакати:	
ПЛ1 - Діаграма використання веб системи	
ПЛ2 - Реляційна схема бази даних	
ПЛ3 - Структурна схема сайту	
ПЛ4 - Інтерфейс інформаційної системи.	

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

CMS	-	Content Management System
ERP	-	Enterprise Resource Planning
HTML	-	HyperText Markup Language
LAN	-	Local Area Network
MVC	-	Model View Control
NOS	-	Network Operating System
OSI	-	Open Systems Interconnection
SRS	-	Software Requirement Specification
UML	-	Unified Modeling Language
БД	-	база даних
ВС	-	веб-система
ІС	-	інформаційна система
ПЗ	-	програмне забезпечення
ПК	-	персональний комп'ютер
РА	-	розподілена архітектура
СКБД	-	система керування базами даних

ВСТУП

Електронні комунікації дозволяють спілкуватися і спільно працювати людям, що знаходяться в різних регіонах планети. Єдиний інформаційний простір Інтернет не тільки скорочує величезні відстані, але і розриває національні та класові кордони, забезпечує кожному індивідууму можливість для самовираження і задоволення різних духовних потреб.

Інтернет надає безпрецедентні можливості підвищення продуктивності роботи, допомагає людям з різних куточків планети спілкуватися і обмінюватися досвідом, також реалізує недорогий спосіб глобальних комунікацій, як всередині будь-якої організації, так і поза нею. Технології Інтернет освоюють малі і великі підприємства, комерційні фірми, банки, урядові організації, установи освіти, науки, охорони здоров'я та інших сфер людської діяльності.

Розробка веб-системи (ВС) торгово-розважального центру дозволить автоматизувати процес надання услуги користувачам та вести електронний облік даних. Сайт запропонує багато ресурсів для інформаційної комунікації мешканців міста, ознайомлення з подіями та датами їх проведення, надасть платформу для електронної комерції та інше.

Метою дипломної роботи є розробка веб-системи для торгово-розважального центру, який надає услуги продажу товарів онлайн та має можливість використовуватися, як платформа для проведення масових подій.

Для того щоб створити таку систему необхідно виконати наступні завдання:

- аналіз технічного завдання;
- розробка специфікацій вимог;
- проектування бази даних;
- розробка веб сайту;
- проектування інтерфейсу користувача;
- розробка макетів сторінок.

1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вихідних даних і вимог до веб системи

Завдання даної системи - це веб сайт торгового центру, швидкий пошук, обробка і зберігання інформації про заходи, що проходять на території центру.

Для реалізації даного завдання необхідно розробити ВС. Роль інтерфейсу між прикладною програмою і базою даних грає програмний комплекс - система керування базами даних (СКБД).

База даних буде зберігатися на сервері. Управління над нею буде виконуватися за допомогою СУБД MySQL.

У базі даних міститиметься інформація:

- заходи;
- магазини;
- клієнти;
- новини;
- афіші.

Веб сайт повинен містити інформацію:

- новини;
- магазини;
- заклади та проекти;
- музика;
- користувачі;
- фотозвіти;
- спільноти.

Перегляд інформації було вирішено реалізовувати з використанням HTML-технологій. Як інтерфейс для клієнта буде використовуватися веб-браузер Google Chrome. Причиною даного вибору є наявність наступних переваг:

- простота отримання доступу до інформації через веб-інтерфейс з будь-

якого пристрою (користувачеві досить вказати адресу);

- відсутність необхідності установки додаткового ПЗ.

Розробляється ВС повинна надавати користувачам доступ до інформації бази даних. Вона повинна дозволяти фільтрувати інформацію веб сайту за різними критеріями (час, відвідувач і т.п.). По кожному фільтру потрібно задавати тип фільтрації (містить, починається, так само, закінчується), що необхідно для підвищення швидкості роботи.

Використовувані програмне забезпечення повинно бути ліцензійно чистим, містити вбудовані можливості забезпечення безпеки і надійності зберігання даних. Вхід в систему користувача повинен проходити через реєстрацію і введення пароля.

При розробці ВС необхідно передбачити виконання наступних вимог.

- можливості адаптації. Тут мається на увазі гнучкість настройки веб сайту і адаптивність для різних видів пристроїв;

- відкритість. Ця вимога має на увазі вільну взаємодію веб системи з іншими додатками як на рівні обміну даними, так і на рівні обміну різного роду керуючими впливами (наприклад, запуск додатків і ін.);

- аналітичність. Ця вимога має на увазі надання якісної аналітичної інформації на різних рівнях управління;

- ергономічність інтерфейсу. У веб сайті повинні застосовуватися стандартизовані засоби інтерфейсу користувача, що відповідають характеру виконуваних людиною операцій;

- особливості вирішення проблеми розподіленого бізнесу. Тут маються на увазі конкретні вимоги по можливості роботи з розподіленими базами даних, використання доступу із застосуванням Інтернету, можливостям консолідації звітності при наявності віддалених філій і т.п;

- захищеність даних. Особливу увагу тут слід приділити порядку призначення і реалізації прав доступу до інформації.

1.2 Аналіз сучасних веб систем

В останні десятиліття концепція торгового центру стала широко поширеною сучасної роздрібною тенденцією, яка поширилася по всьому світу. Торгові центри позбавляють від необхідності відвідувати кілька місць, тут можна знайти все під одним дахом.

Робота сучасних торгових центрів заснована на багаторічних дослідженнях поведінки споживачів як в офлайн, так і в онлайн. Це потрібно враховувати при створенні сайту ТРЦ - інтерфейс повинен бути інтуїтивно зрозумілим.

Веб-сайт це інструмент, який дозволяє розширювати відносини з клієнтами навіть після відвідування торгового центру. А це, в свою чергу, допомагає створювати більш глибокі зв'язки за допомогою програм лояльності і просувати нових орендарів.

Фахівці називають 10 критеріїв хорошого сайту [1, 2, 3]:

1) зміст, або наповнення сайту корисною інформацією. Сайт повинен залучати відвідувача якісними статтями, оригінальними фотографіями і зображеннями. Повноцінне зміст формується власними силами, або залученням праці професійних копірайтерів і фотографів;

2) структура, тобто зручне розміщення інформації на сайті. Сайт повинен бути організований так, щоб випадковий відвідувач легко знайшов інформацію на потрібну йому тему по системі вкладок. При створенні сайту слід пам'ятати, що потенційний користувач не готовий довго розбиратися в складній системі переходів від сторінки до сторінки. Користувач воліє прості і зрозумілі рішення;

3) оформлення, або дизайн. Стиль сайту важливий для формування позитивного враження про нього. В оформленні вітається яскрава індивідуальність, здатна викликати довіру до змісту сайту. Важливо, щоб в гонитві за неповторним виглядом сайту, його творець не забував про інтереси відвідувачів, і не відлякав їх занадто оригінальним оформленням;

40 оновлення контенту. Сайт живе, поки на ньому з'являються нові статті, кумедні картинки, відмінні фотографії і відеозвіти про події. Без свіжої інформації, сайт приречений на забуття і загибель;

5) адреса, або розміщення в інтернеті. Щоб сайт викликав довіру, слід подбати про лаконічному його назві та розміщенні на платному хостингу. Серйозні компанії ніколи не користуються послугами "безкоштовних сайтів", це негативно позначається на їх іміджі;

б) швидкість завантаження сайту. Користувач не чекатиме завантаження понад чотири або п'яти секунд. Тому при створенні сайту засікайте період завантаження за допомогою секундоміра;

7) продумана система зв'язку з відвідувачем. Контакти для зв'язку з адміністрацією сайту повинні бути легко обнаружими. Передбачте кілька варіантів: зв'язок через електронного листа, за номером телефону, через онлайн - консультанта. Якісна зворотний зв'язок допомагає створити коло постійних відвідувачів сайту;

8) індексація сайту, його просування. Основний алгоритм пошуку інформації в Мережі - через пошукові системи і ключові запити. Тому контент сайту повинен містити ключові слова і цілі словосполучення, за допомогою яких пошукові системи будуть переадресовувати користувачів на конкретний сайт. Допоможіть користувачам легко знайти ваш сайт;

9) реклама. Сайт без реклами виглядає підозріло. Сайт з надмірною кількістю реклами, з нав'язливими банерами і величезними спливаючими вікнами викликає бажання відразу ж покинути його назавжди. Турбота про додатковий дохід через рекламу не повинна шкодити самому існуванню сайту;

10) інтеграція з соціальними мережами. Практично весь час в інтернеті користувачі проводять в улюблених соціальних мережах.

1.3 Розробка специфікації вимог SRS

1.3.1 Вимоги до веб системи

Вимоги до веб-додатку — набір вимог щодо властивостей, якості та функцій програмного забезпечення, що буде розроблено, або знаходиться у розробці. Вимоги визначаються в процесі аналізу вимог та фіксуються в специфікації вимог, діаграмах прецедентів та інших артефактах процесу аналізу та розробки вимог [4]. Даним сайтом можуть користуватися люди різного віку і різної статі, але якщо робота має дорослий рейтинг то перед тим як почати з нею ознайомлюватися буде виводиться попередження.

Вимоги веб-додатку для звичайних користувачів:

- простий і зрозумілий інтерфейс користувача;
- можливість пошуку інформації за ключовими словами;
- можливість коментування;
- можливість переглядати новини;
- можливість перегляду афіши.

Вимоги веб-системи для адміністраторів:

- можливість редагування бази даних;
- наявність зручного інтерфейсу для взаємодії з базою даних і сервісом.

ВС повинна забезпечувати:

- сортування новин;
- сортування афіши;
- доступ до бази даних;
- можливість зберігати коментарі;
- можливість робити онлайн покупок;
- можливість зберігання фотогалереї;
- можливість оцінювати роботи.

Специфікація вимог веб системи – закінчений опис поведінки програми, яку потрібно розробити. Включає ряд користувальницьких сценаріїв, які описують всі варіанти взаємодії між користувачами і програмним забезпеченням.

Призначені для користувача сценарії є засобом представлення функціональних вимог. В доповнення до призначених для користувача сценаріїв, специфікація також містить нефункціональні вимоги, які накладають обмеження на дизайн або реалізацію (такі як вимоги продуктивності, стандарти якості, або проектні обмеження) [5].

Дана ВС є незалежним від інших програмних виробів і повністю автономним щодо виконання своїх функцій. В контексті даного документу під ВС розуміється взаємопов'язана структура, яка складається з таких компонентів: веб сайту, БД, СУБД.

В таблиці 1.1 наведені характеристики всіх користувачів ВС.

Таблиця 1.1 – Користувачі інформаційної веб-системи

Користувач	Характеристика
Відвідувач	Особа, яка перейшла до сайту з метою ознайомлення з контентом.
Користувач (читач)	Особа, яка зареєструвалася на сайті з метою отримання новин або онлайн покупок.
Користувач (магазин)	Особа, яка зареєструвалася на сайті для продажу товарів.
Адміністратор	Людина, яка може редагувати різноманітну інформацію на сайті.
Тех. підтримка	Людина, яка виконує різноманітні функції, що забезпечують коректну роботу і регулярне оновлення інтернет-ресурсу. Починаючи з технічної підтримки сайту і закінчуючи його візуальним, текстовим наповненням.

Сайт має відповідати таким обмеженням:

- сайт має швидко переходити між сторінками;
- повне і безпомилкове здійснення заявлених функцій;

– видалена інформація має бути знищена з БД.

Функціональні вимоги кожного користувача системи мають власні особливості.

Клас користувачів «Відвідувачі».

Сайт повинен:

- забезпечувати перегляд інформації;
- забезпечити реєстрацію;
- забезпечити зручний та швидкий спосіб пошуку інформації;
- забезпечити зручний спосіб сортування новин;

Клас користувачів «Користувачі(читач)».

Сайт повинен:

- забезпечувати перегляд інформації;
- забезпечити авторизацію;
- забезпечити зручний та швидкий спосіб пошуку;
- забезпечити зручний спосіб сортування;
- забезпечити можливість коментування новини;
- забезпечити зручний сервіс онлайн покупок.

Клас користувачів «Користувачі(магазин)».

Сайт повинен:

- забезпечувати перегляд інформації;
- забезпечити авторизацію;
- забезпечити можливість додавання товарів;
- забезпечити можливість керувати вже доданими товарами (редагування, видалення, доповнення);

- забезпечити зручний та швидкий спосіб пошуку товарів;
- забезпечити зручний спосіб сортування товарів;
- забезпечити можливість коментування новин.

Клас користувачів «Адміністратор».

Сайт повинен:

- надавати різноманітну інформацію ВС;
- забезпечувати контроль за магазинами;
- забезпечувати редагування контенту ВС.

Клас користувачів «Тех. підтримка».

Сайт повинен:

- дозволяти здійснювати розробку нового функціоналу;
- дозволяти вносити зміни у зовнішнього вигляду сайту.

Зазвичай веб-сайти є складні і динамічні структури. На їх розробку виділяють короткий час, щоб сайт як можна швидше запрацював і давав ефект. Часто розробники відразу ж сідають за написання коду, навіть не обдумавши, що вони збираються створювати, і як вони це збираються зробити. Код для сервера часто пишеться з листа, таблиці в базах даних створюються в міру потреби, і в підсумку іноді архітектура системи починає розвиватися в зовсім непередбаченому напрямку. Однак за допомогою простого моделювання і суворого підходу до програмування можна зробити процес розробки в набагато більш гладким і гарантувати, що створену веб-систему буде просто підтримувати в подальшому.

Моделювання дозволяє вирішити чотири різні завдання:

- візуалізувати систему в її поточному або бажаному для замовника стані;
- описати структуру або поведінку системи;
- отримати шаблон, що дозволяє сконструювати систему;
- документувати прийняті рішення, використовуючи отримані моделі;

Чим більша і складніша система, тим більшого значення набуває моделювання при її розробленні. Без моделі складну систему неможливо сприйняти як одне ціле. Людське сприйняття складних сутностей є обмеженим. Моделювання системи чи об'єкта дозволяє звузити проблему, зосередивши увагу в кожен момент тільки на певних її аспектах. Складне завдання завжди легше вирішити, якщо розділити його на менші. Правильно обрана модель дозволяє створювати проекти на вищих рівнях абстракції.

Для швидкої і ефективної розробки програмного продукту з мінімальними витратами потрібно залучити робочу силу, вибрати правильні інструменти і визначити правильний напрям роботи. Щоб впоратися з поставленим завданням, беручи до уваги витрати на життєвий цикл системи, необхідно, щоб процес розробки програми був ретельно продуманий і міг бути адаптований до потреб вашого бізнесу і технології [6].

1.3.2 Побудова діаграм використання сайту

Для роботи веб-сайту на діаграмі варіантів використання були описані наступні блоки:

- Відвідувач – особа, яка перейшла до сайту з метою ознайомлення з контентом (рис.1.1);
- Читач– особа, яка зареєструвалася на сайті з метою отримання новин або онлайн покупок (рис.1.2);
- Магазин – особа, яка зареєструвалася на сайті для продажу товарів (рис.1.3);
- Адміністратор – людина, яка може редагувати різноманітну інформацію на сайті (рис.1 4);
- Тех. підтримка – людина, яка виконує різноманітні функції, що забезпечують коректну роботу і регулярне оновлення інтернет-ресурсу. Починаючи з технічної підтримки сайту і закінчуючи його візуальним, текстовим наповненням (рис.1.5).

На рисунку 1.6 наведена загальна діаграма використання веб-сайту.

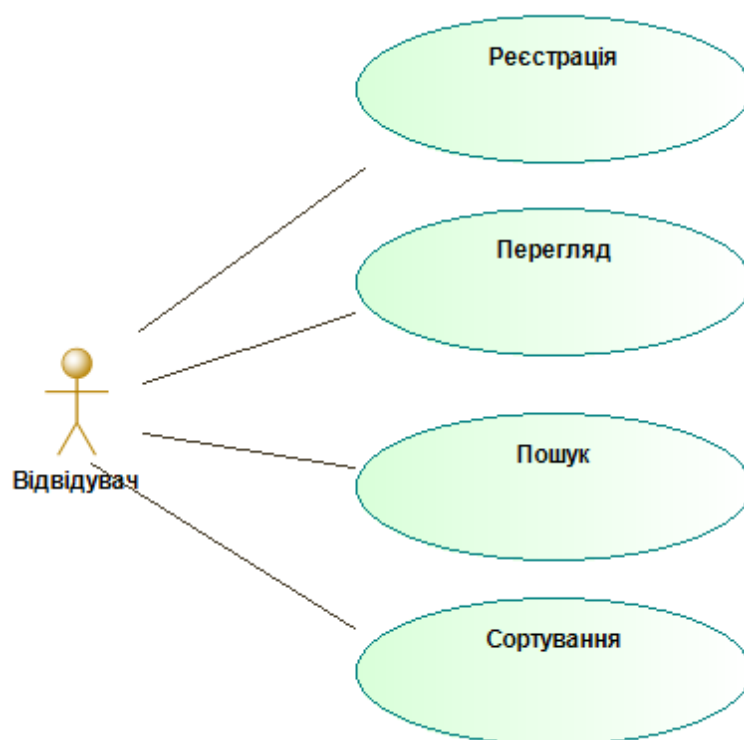


Рисунок 1.1 – Діаграма варіантів використання для актора «Відвідувач»

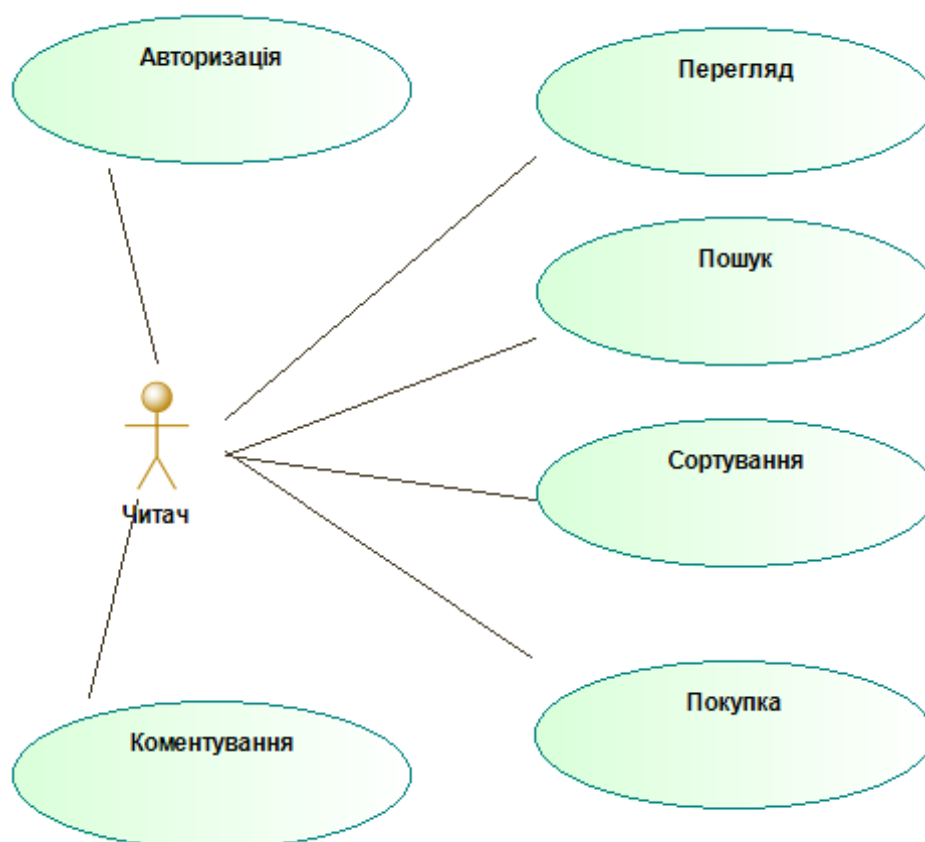


Рисунок 1.2 – Діаграма варіантів використання для актора «Читач»

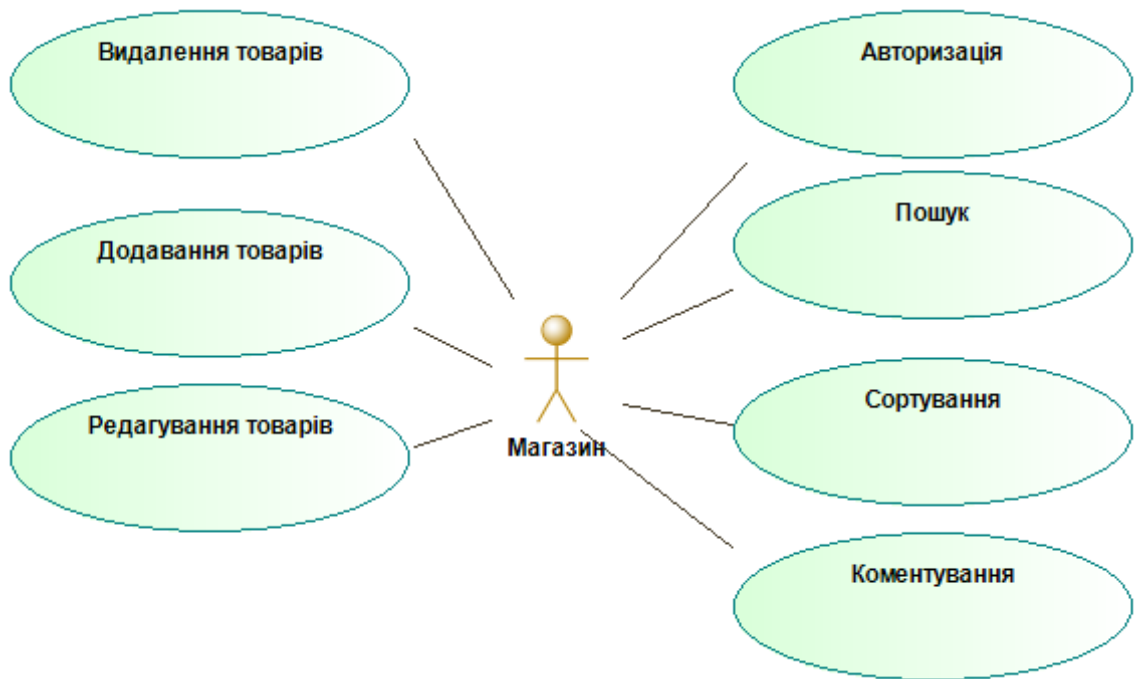


Рисунок 1.3 – Діаграма варіантів використання для актора «Магазин»

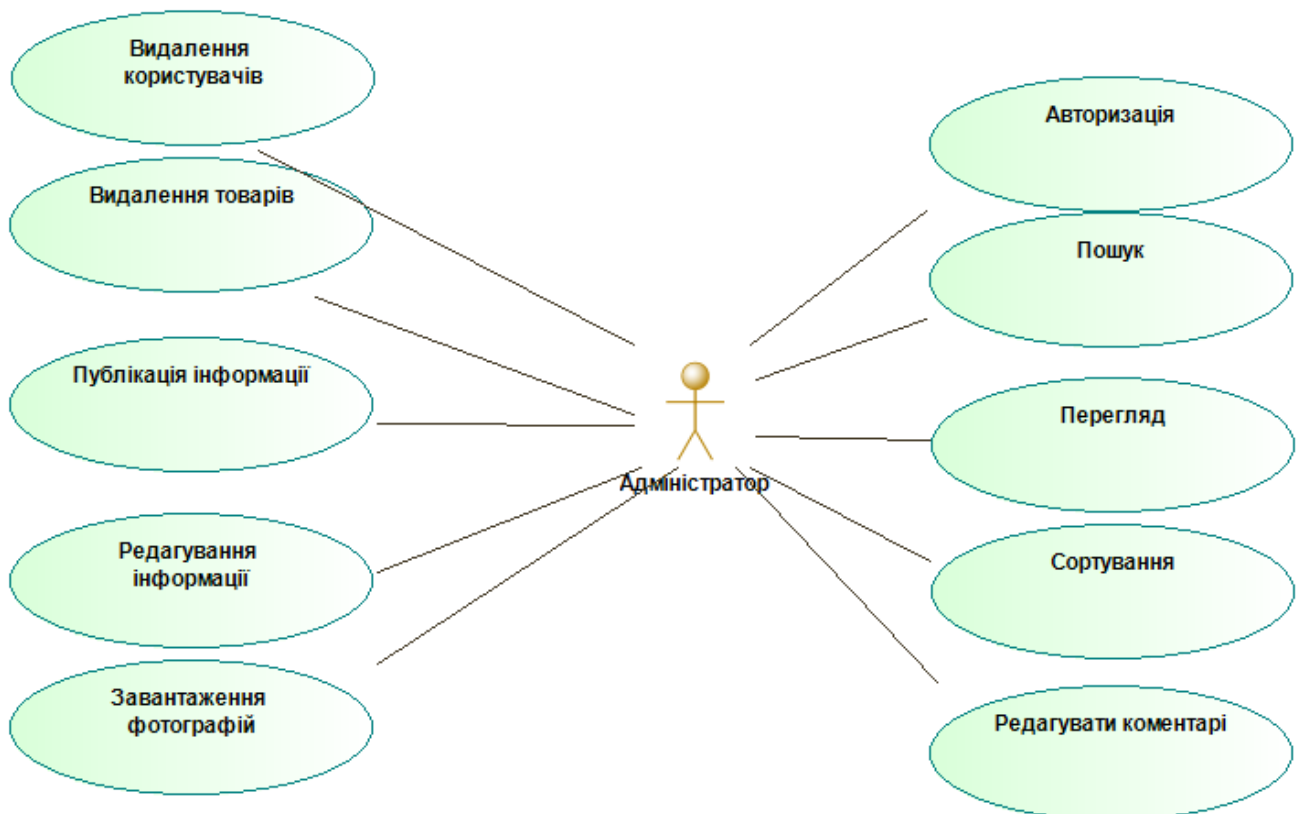


Рисунок 1.4 – Діаграма варіантів використання для актора «Адміністратор»

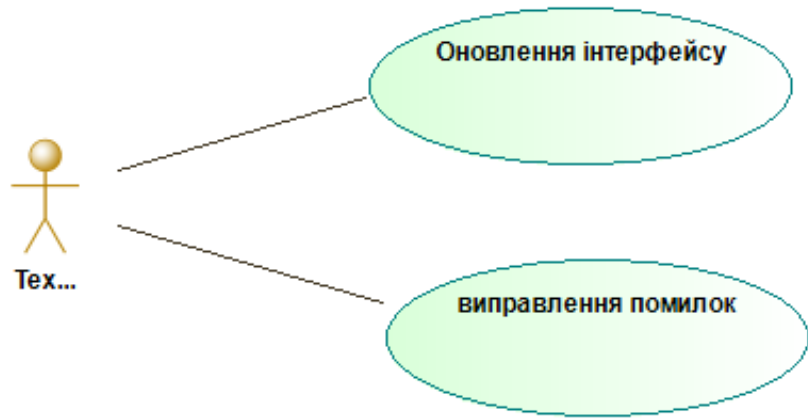


Рисунок 1.5 – Діаграма варіантів використання для актора «Тех. підтримка»

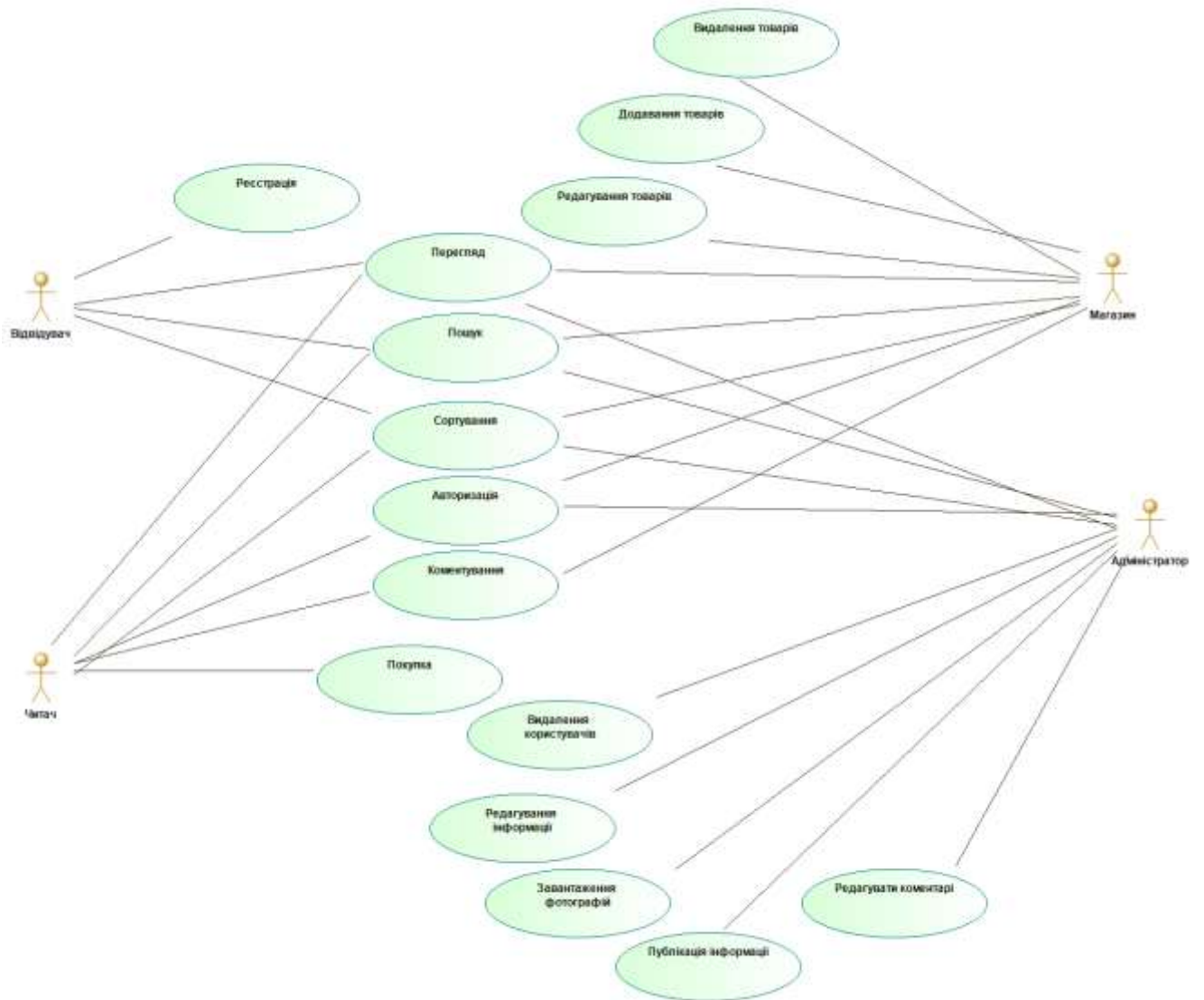


Рисунок 1.6 – Загальна діаграма використання

1.3.3 Побудова діаграм послідовності сайту

Діаграма послідовності - діаграма, на якій для деякого набору об'єктів на єдиній тимчасовій осі показаний життєвий цикл будь-якого певного об'єкта (створення-діяльність-знищення якоїсь сутності) і взаємодія акторів (дійових осіб) ІС в рамках якої-небудь певної прецеденту (відправка запитів і отримання відповідей). Використовується в мові UML.

Основними елементами діаграми послідовності є позначення об'єктів (прямокутники з назвами об'єктів), вертикальні «лінії життя», що відображають плин часу, прямокутники, що відображають діяльність об'єкта або виконання ним певної функції (прямокутники на пунктирною «лінії життя»), і стрілки, що показують обмін сигналами або повідомленнями між об'єктами [7].

Опис прецеденту «читання новин». Основний виконавець: користувач. Передумови немає.

Основний потік:

- 1) користувач обирає тему;
- 2) користувач шукає потрібним йому за темою новину;
- 3) користувач залишає коментар;
- 4) додаток записує результат у базу.

На даній діаграмі об'єкти розташовуються зліва направо (рис.1.7).



Рисунок 1.7 – Діаграма послідовності читання новини

Опис прецеденту редагування бази даних (рис. 1.8). Основний виконавець прецеденту адміністратор. Передумови: адміністратор авторизований.

Основний потік:

- 1) адміністратор редагує базу даних;
- 2) додаток надсилає запит до бази даних;

- 3) додаток обробляє запит, якщо запит невдалий, додаток повертає повідомлення про помилку;
- 4) додаток повертає результат.

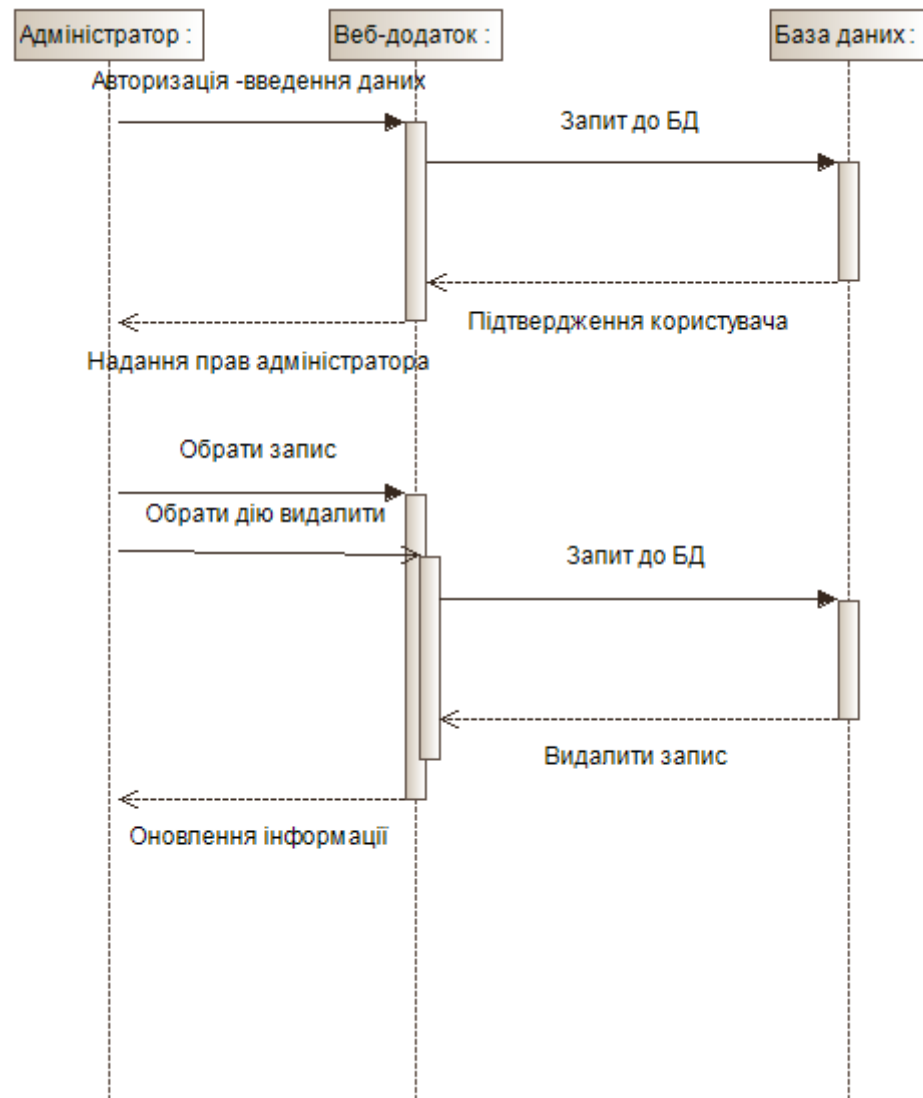


Рисунок 1.8 – Діаграма послідовності видалення записів адміністратором

Опис прецеденту реєстрації користувачів (рис. 1.9). Основний виконавець прецеденту відвідувач. Передумови: відвідувач неавторизований.

Основний потік:

- 1) відвідувач відкриває форму реєстрації;
- 2) відвідувач вводить дані в поля введення;

- 3) додаток проводить валідацію введених даних, якщо дані не є коректними, додаток виведе повідомлення про помилку;
- 4) додаток робить запит до бази даних на перевірку існуючих даних;
- 5) база даних робить перевірку і повертає результат, якщо введені дані вже існують в базі даних, додаток виведе повідомлення про помилку;
- 6) відвідувач підтверджує реєстрацію;
- 7) відвідувач закриває форму реєстрації.

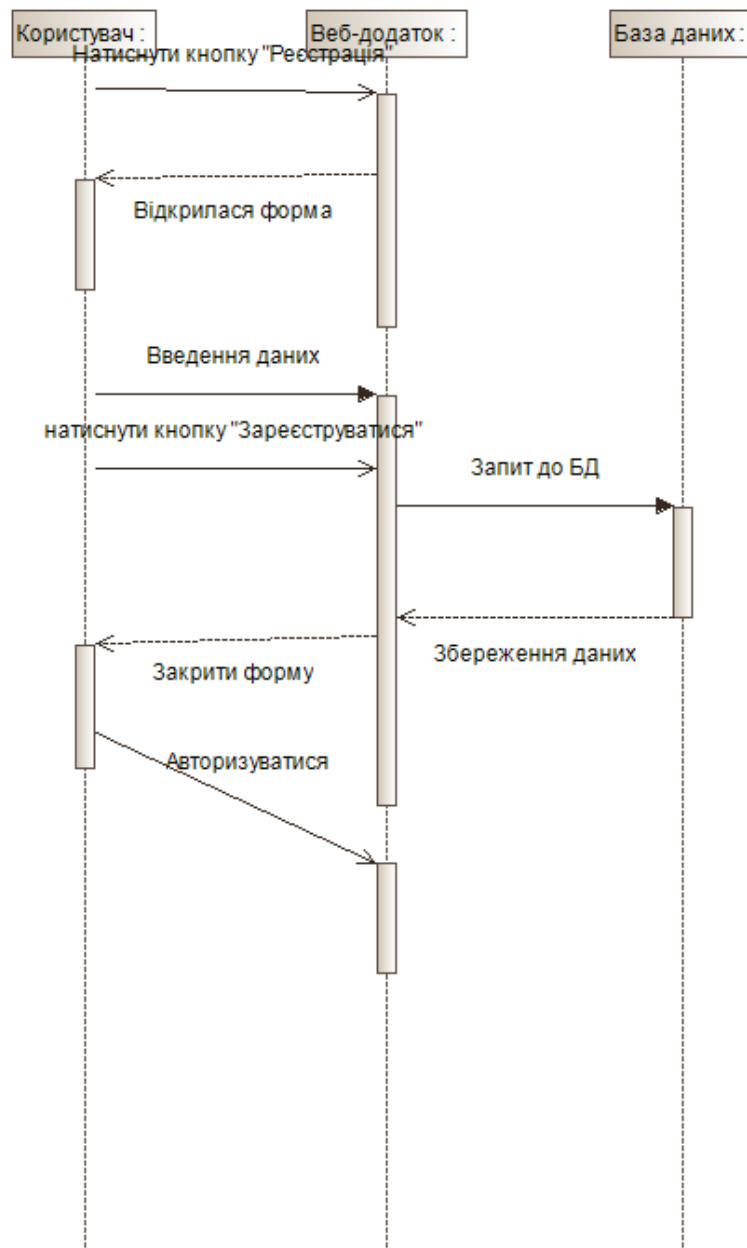


Рисунок 1.9 – Діаграма послідовності реєстрації

2 РОЗРОБКА БАЗИ ДАНИХ

2.1 Опис предметної області

База даних повинна зберігати інформацію торгово-розважального центру. У ній повинні міститися такі дані:

- інформація про замовників заходів;
- інформація про захід;
- інформація про реєстрацію проведеного заходу;
- інформація про місце проведення;
- інформація про співробітників служби підтримки та організації заходів;
- інформація про клієнтів, які реєструються;
- інформація про товари і магазинах.

На даному етапі розробки реалізована підтримка розважальних заходів торгового центру. Побудова інформаційно-логічної (концептуальної) моделі було виконано на основі методики, детально описаної в роботі [8]. При цьому були виділені інформаційні об'єкти бази даних і зв'язку між ними. Відповідність описових і ключових реквізитів представлено в таблиці 2.1.

Таблиця 2.1 – Угрупування об'єктів з інформаційних елементів

Назва	Назва реквізита	Реквізити об'єкта	Признак ключа
Місце	Код місця	КМ	Простий Унікальний
	Назва	НМ	
	Адреса	АМ	
	Кількість квитків	КК	
Замовник	Код замовника	КЗ	Простий Унікальний
	ПІБ	ПІБ	
	Назва компанії	НК	

Продовження таблиці 2.1

Назва	Назва реквізита	Реквізити об'єкта	Признак ключа
	Адреса	А	
	Місто	М	
	Номер телефону	НТ	
	Ел.почта	ЕП	
	Нотатки	Н	
Захід	Код заходу	КЗ	Простий Унікальний
	Код замовника	КЗА	
	Назва	НЗ	
	Тип заходу	ТП	
	Стан	С	
	Код місця	КМ	
	Начальна дата	НД	
	Кінцева дата	КД	
	Час початку	ЧП	
	Час закінчення	ЧЗ	
	Супроводження	С	
	Затверджено	З	
	Опис	О	
	Нотатки	НЗ	
Користувачі	Код користувача	КК	Простий Унікальний
	ПІБ	ПІБК	
	Посада	ПС	
	Ел.почта	ЕПК	
	Телефон	Т	
Афіша	Код афіши	КА	Простий Унікальний
	Код_співробітника	КС	

Продовження таблиця 2.1

Назва	Назва реквізита	Реквізити об'єкта	Признак ключа
	Дата створення	ДС	
	Строк дії	СД	
Реєстрація	Код реєстрації	КР	Простий Унікальний
	Код заходу	КЗ	
	Код користувача	КК	
	Дата реєстрації	ДР	
	Платіж	ПР	

У таблиці 2.2 наведені інформаційні зв'язки між об'єктами предметної області

Таблиця 2.2 - Зв'язки інформаційних об'єктів

Головний об'єкт	Підлеглий об'єкт	Тип зв'язку
Місце	Захід	1:М
Замовник	Захід	1:М
Афіша	Користувач	1:М
Користувач	Реєстрація	М:1
Захід	Реєстрація	1:М

На рис. 2.1 представлена інформаційно-логічна модель розглянутої БД, побудована відповідно до виявлених інформаційними об'єктами і зв'язками між ними.

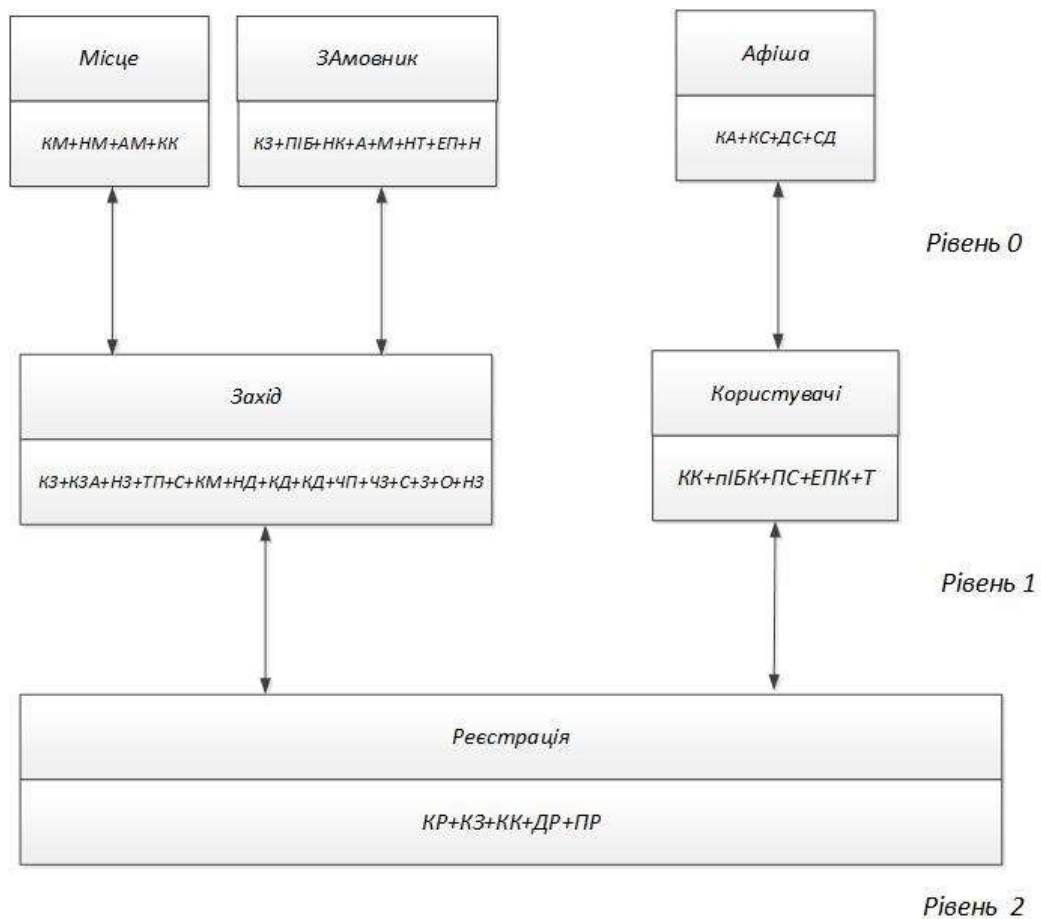


Рисунок 2.1 – Концептуальна модель предметної області

2.2 Реляційна модель бази даних

Логічна (реляційна) модель наведена на рисунку 2.2. При цьому в табл.2.3 наведено опис таблиць.

Фізичні моделі баз даних визначають способи розміщення даних в середовищі зберігання і способи доступу до цих даних, які підтримуються на фізичному рівні. Історично першими системами зберігання і доступу були файлові структури і системи управління файлами (СУФ), які фактично були частиною операційних систем. СКБД створювала над цими файловими моделями свою надбудову, яка дозволяла організувати всю сукупність файлів таким чином, щоб вона працювала як єдине ціле і отримувала централізоване управління від СКБД.

Однак безпосередній доступ здійснювався на рівні файлових команд, які СКБД використовувала при маніпулюванні всіма файлами, складовими збережені дані однієї або декількох баз даних.

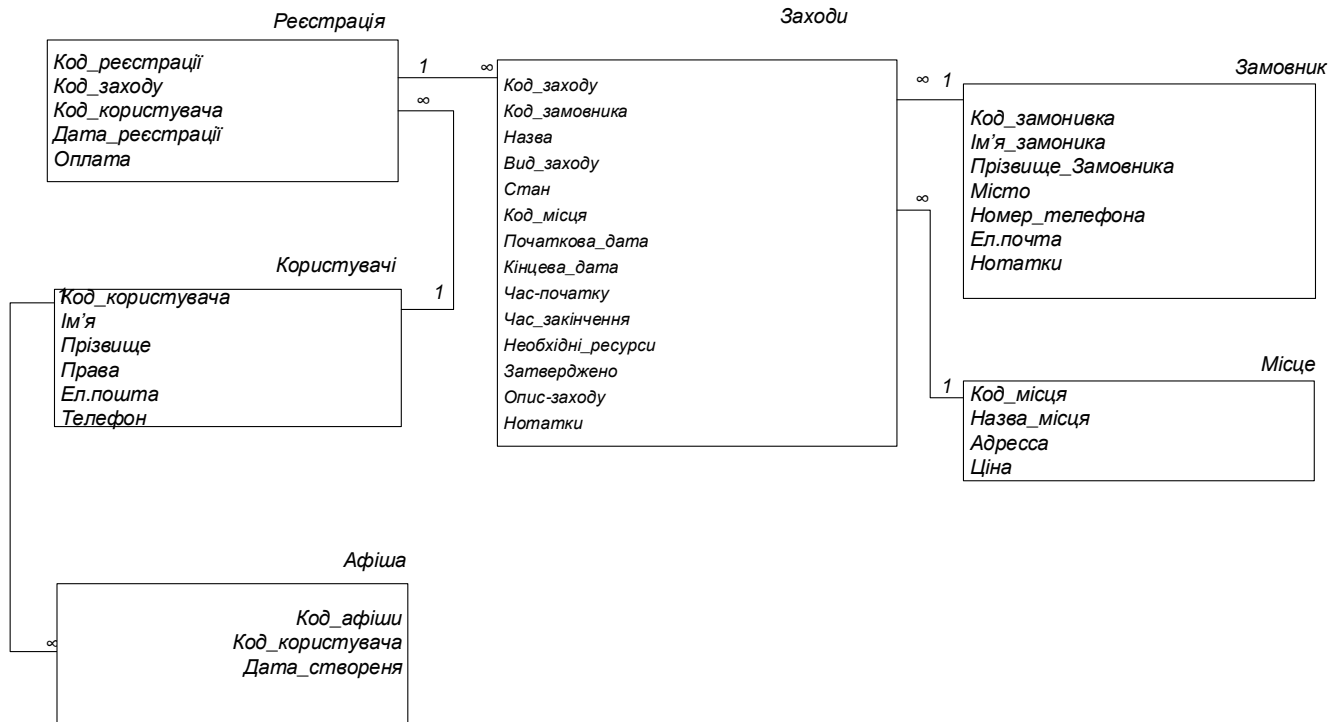


Рисунок 2.2 - Логічна (реляційна) модель бази даних

Таблиця 2.3 - Опис таблиць бази даних

№ п.п.	Назва	Ідентифікатор	Назва	Тип зв'язку	Атрибути
1.	Місце	place	Перелік місць	1:М Захід	Код місця
2.	Замовник	client	Список замовників	1:М Захід	Код замовника
3.	Захід	party	Список заходів	1:М Реєстрація	Код заходу
4.	Афіша	afiha	Список афіш	1:М Користувач	Код користувача
5.	Користувач	worker	Список скористувачів	М:1 Реєстрація	Код користувача
6.	Реєстрація	register	Список реєстрація	1:М Користувач	Код користувача
				М:1 Захід	Код заходу

2.3 Розробка бази даних

Фізична база даних, відповідна логічній моделі предметної області, була створена на основі СУБД MySQL засобами MySQL Workbench.

Для початку необхідно створити таблиці, для цього натиснемо на кнопку Add Table, з'явиться наступна форма (рис.2.3).

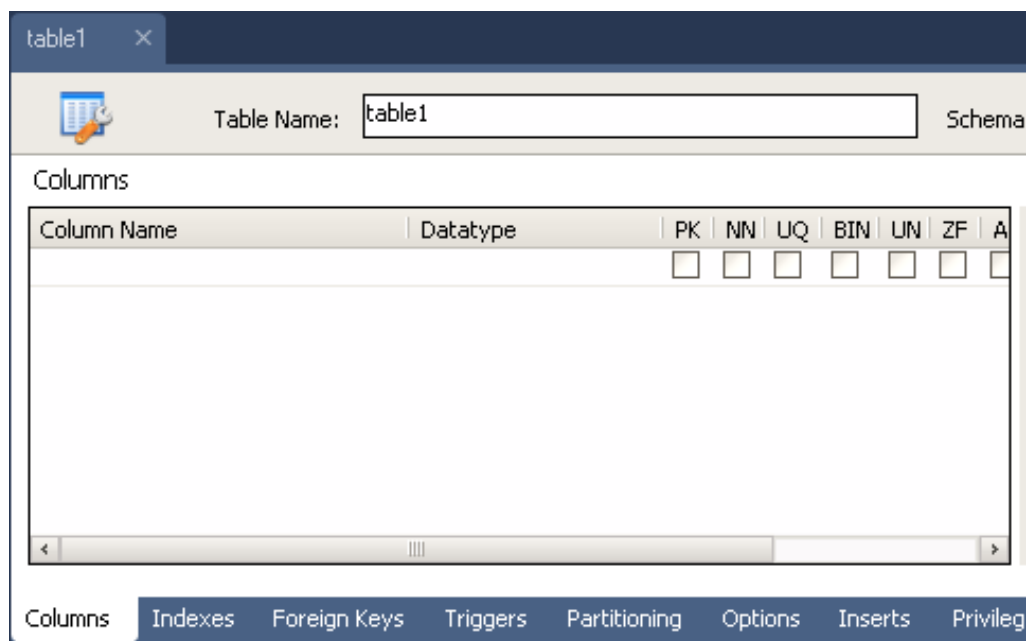


Рисунок 2.3 – Створення нової таблиці

Спочатку створюємо таблицю client, яка буде зберігати дані про клієнтів організації, в поле table Name впишемо ім'я таблиці, в розділі форми Columns створюємо поля таблиці:

- перше поле id_client містить унікальний номер клієнта, задаємо йому властивості: Auto Increment, Not Null, Primary key і Unique, в розділі Data type вибираємо цілочисельний тип integer;

- друге і третє поля зберігають інформацію про П.І.Б. клієнта, встановлюємо полю властивості: Not Null, Primary key ,, в розділі Data type вибираємо строковий тип VARCHAR і задаємо кількість символів в 255;

- четверте поле name_org, містить інформацію про назву організації та

задаємо кількість символів в 255;

- наступні поля: address, city містить адресу, e_mail містить адресу електронної пошти, comments коментарі і поле telefon містить телефон, зі строковим типом VARCHAR довжиною в 255 символів, за винятком останнього поля type якому вистачить з 45 символів.

Після виконаних маніпуляцій форма з ім'ям таблиці users виглядає як на рисунку 2.4.

The screenshot shows a window titled 'client - Table' with a table structure for the 'client' table. The table has the following columns and properties:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_client	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name_client	VARCHAR(255)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
surname_client	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name_org	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
address	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
city	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
telefon	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
comments	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 2.4 – Форма таблиці «client»

На діаграмі з'явиться таблиця client с полями і індексами (рис.2.5).

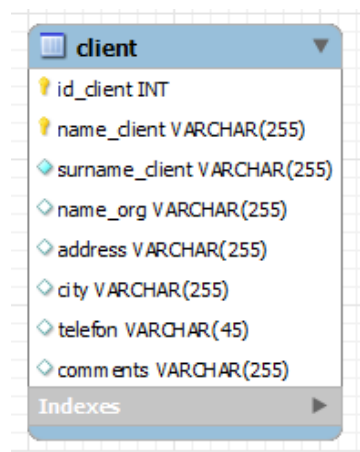


Рисунок 2.5 – Таблиця «client»

Аналогічним способом створюємо таблицю `place` з даними про місце проведення заходу, що містить поля `id`, `name_place` назва місця (закладу), `address`-адреса, `ticket` кількість місць в даному закладі (рис. 2.6).

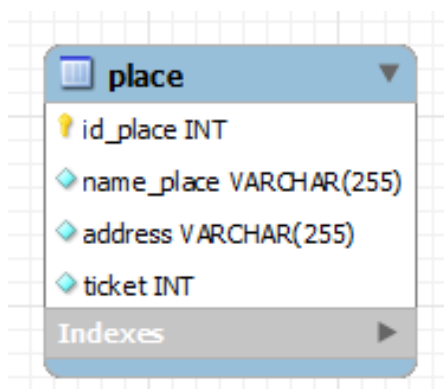


Рисунок 2.6 – Таблиця «**place**»

Далі по вже відомим методом створюємо таблицю `register`, яка зберігає інформацію про реєстрацію заходів. `id_register` - поле коду реєстрації, `id_party` - код заходу, `id_worker` - код співробітника, який відповідальний за захід, `date` - дата реєстрації, `pay` - інформація про оплату (рис.2.7).

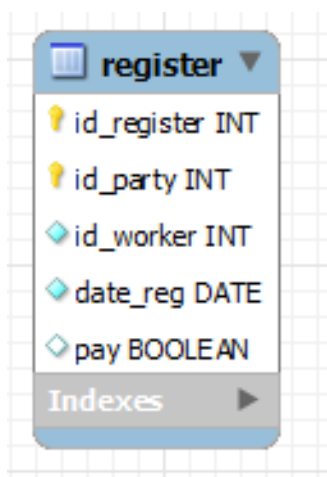


Рисунок 2.7 – Таблиця «**register**»

Потім створюємо таблицю `worker` для зберігання інформації про співробітників фірми. `id_worker` - поле для унікального номера співробітника, `name`

містить ім'я співробітника, surname - прізвище, post - посаду співробітників, e-mail
telefon - поля містять контактну інформацію зі співробітником (рис.2.8).

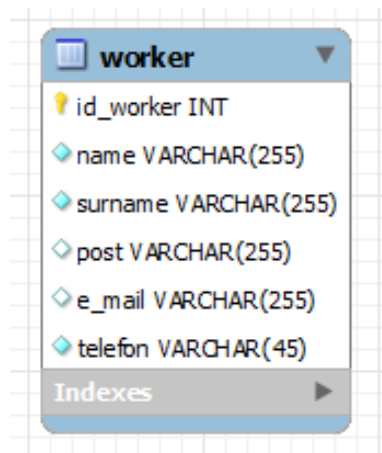


Рисунок 2.8 – Таблиця «**worker**»

Далі створюємо таблицю, що зберігає інформацію про розроблених афішах afiha, вона складається з унікального, ненульового, ключового поля id_afiha з автоінкрементом цілочисленного типу, і коду співробітника id_worker, який відповідальний за неї, а також інформації про час її створення і актуальності date, time. (рис.2.9).

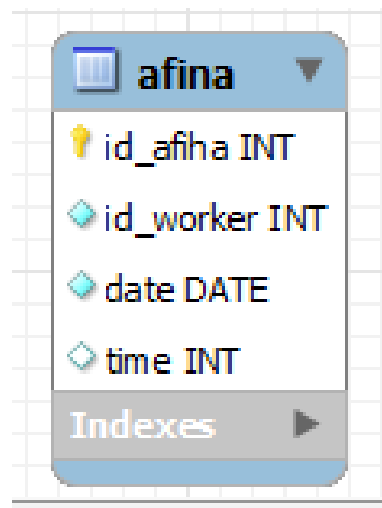


Рисунок 2.9 – Таблиця **afiha**

Остання таблиця party заходу містить інформацію про заходи міста.

Поля таблиці party: id ключове, нульове, унікальне поле цілочисленного типу

з автоінкрементом, поле `id_client` містить інформацію про кзаказчіке заходи, `name` назва заходу, `type` - поле відповідає за тип заходу (тобто концерт, прем'єра, вечірка і т.д.), `state` - поле вказує на інформацію про стан даного заходу, `id_place` - код місця проведення, `first_date`, `end_date` - дата початку і закінчення заходів, `time_start`, `time_end` - час початку і завершення заходу, `resource` - додаткові ресурси для проведення заходу `approved` - інформація про затвердження та погодження з міською адміністрацією, `description` - опис, `comments` - замітки (рис.2.10).

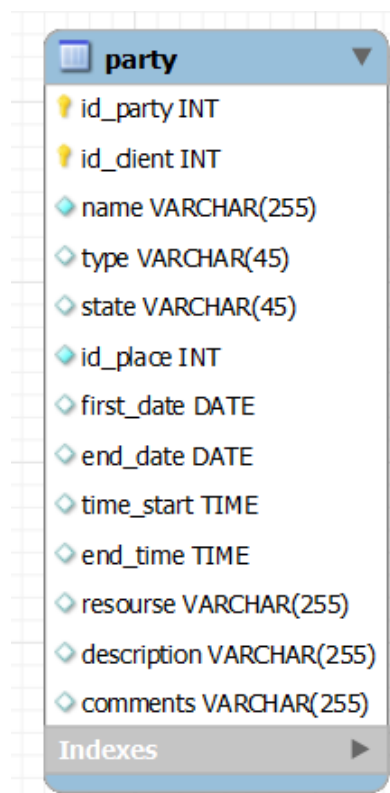


Рисунок 2.10 - Таблиця **party**

Розроблену базу даних, що складається з шести таблиць, тепер необхідно пов'язати, ми вже практично розробили основні поля целочисленного типу, вони і стануть основою для зв'язування.

Для того щоб зв'язати дві таблиці необхідно двічі клацнути лівою кнопкою миші на діаграму з таблицею і вибрати вкладку Foreign keys (зовнішні ключі), далі в поле Foreign key name ввести унікальне ім'я зовнішнього ключа, двічі клацнути по вкладці Referenced table і вибрати другу таблицю, потім в формі розташованій

правіше вибрати посилається поле `type_id` і вибрати в спливаючому списку поле `id`.

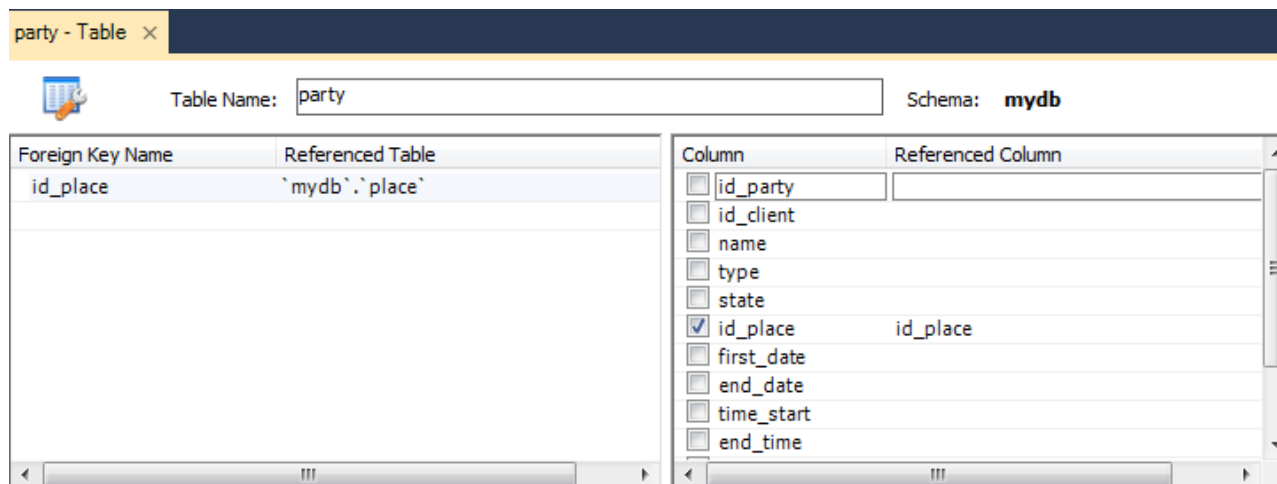


Рисунок 2.11 – Властивості зв'язку таблиць

Таким чином, обидва поля таблиці виявляються пов'язані, потім задаємо тип відносин зв'язку між таблицями, відкриваємо вікно, клікнувши на інформацію, що з'явилася зв'язок між таблицями, і виберемо вкладку Foreign Key і в розділі Cardinality виберемо тип зв'язку один до багатьох, і закриємо вікно. На діаграмі відобразиться зв'язок таблиць (рис.2.12).

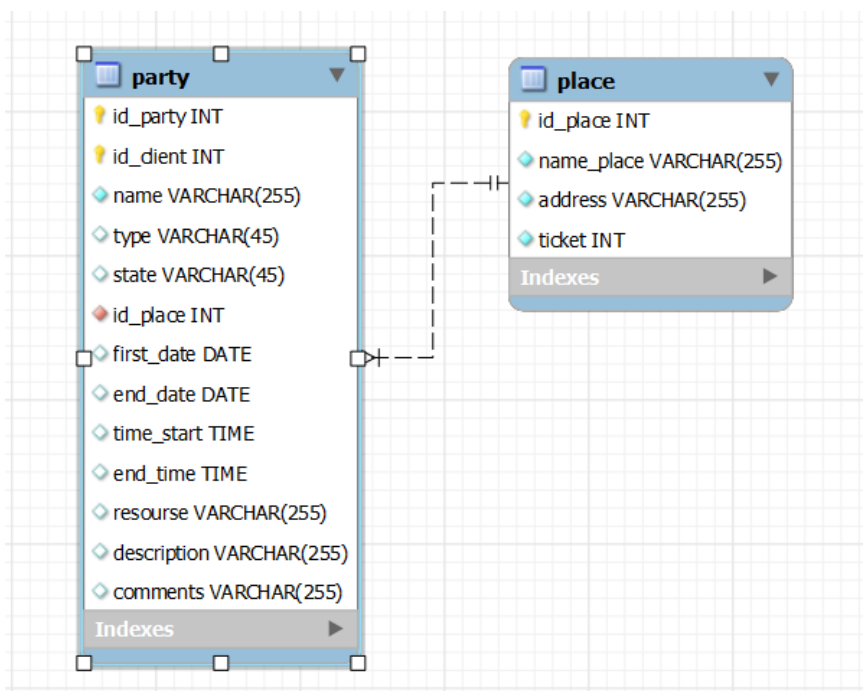


Рисунок 2.12 – Зв'язок таблиць

Аналогічним чином пов'язуємо всі ключові поля в таблицях, щоб вони були логічно пов'язані між собою, далі нам необхідно переконається, що спроектована база даних відповідає третій нормальній формі.

Нормальна форма - якість зв'язку в реляційної моделі даних, що характеризує його з точки зору надмірності, яка потенційно може привести до логічно помилкових результатів вибірки або зміни даних. Нормальна форма визначається як сукупність вимог, яким має задовольняти відношення.

У реляційної моделі відношення завжди знаходиться в першій нормальній формі за визначенням поняття відношення. Що ж стосується різних таблиць, то вони можуть не бути правильними уявленнями відносин і, відповідно, можуть не перебувати в першій нормальній формі. Мінлива відносини знаходиться в другій нормальній формі тоді і тільки тоді, коли вона знаходиться в першій нормальній формі і кожен не ключовий атрибут неприводимого (функціонально повно) залежить від її потенційного ключа. База даних буде знаходитися в третій нормальній формі, якщо вона приведена до другої нормальної форми і кожен не ключовий стовпець незалежний один від одного.

Таким чином, наша база знаходиться в третій нормальній формі, тому що кожен не ключовий стовпець незалежний один від одного. Це наочно видно на діаграмі нашої бази даних на рисунку 2.13.

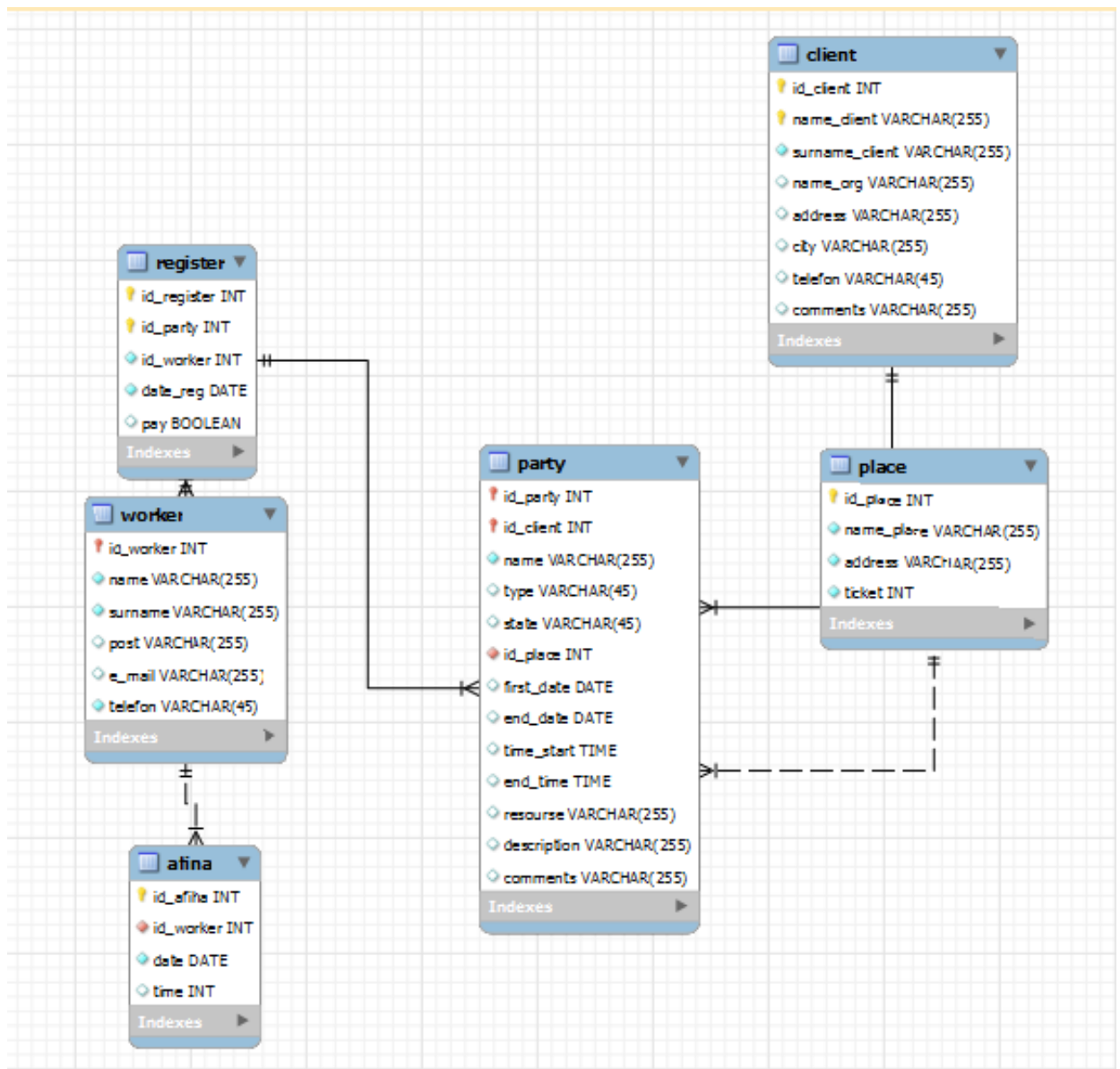


Рисунок 2.13 – Структура бази даних

3 РОЗРОБКА ВЕБ САЙТУ ТОРГОВО-РОЗВАЖАЛЬНОГО ЦЕНТРУ

3.1 Структурна схема сайту

Перш ніж приступити до написання коду і створення веб-сторінок сайту, необхідно розробити його структуру, кількість сторінок, які буде містити сайт, систему посилань між ними. Для цього необхідно скласти схему сайту, його структурну діаграму, на якій показані станиці і посилання між ними. У структурі

також необхідно передбачити можливість подальшого розширення сайту, додавання розділів і посилань.

Структурна діаграма сайту визначає також методи навігації, які будуть використовуватися. Існує три типи навігації:

- переміщення по посиланнях, які є на сайті;
- використання форми пошуку по сайту;
- доступ через портал до його вмісту.

Зовсім не обов'язково, щоб на сайті були присутні всі три методи. Найчастіше використовується перший спосіб.

Структурна діаграма веб сайту торгово-розважального центру показана на рисунку 3.1. Розглянемо більш детально елементи сайту першого рівня.

Головна сторінка - найперша сторінка, яка з'являється при заході на сайт. Головна сторінка є у всіх сайтів, від солідних інтернет-магазинів до популярних форумів. Основна функція головної сторінки - це привернення уваги цільової аудиторії і конвертації користувачів в реальних клієнтів.

Меню - найбільш поширена навігаційна система, вона схожа на зміст до книги. Меню складається з коротких текстових активних посилань на певні сторінки сайту. Загалом, справа краще розташовувати найважливіші посилання на розділи, оскільки дана зона найбільш активна на сайті (на цей кут відвідувач найчастіше дивиться, напевно, пристосовуючись натиснути на хрестик «закрити» в правому верхньому куті браузера).

Для великих сайтів, де просто фізично неможливо вмістити все назви розділів в меню, найкраще додати пошукову форму. Переваги «пошуку» в тому, тому, що завдяки йому користувач знаходить кілька текстів, в яких зустрічалося шукане слово, тому має можливість найбільш повно ознайомитися з цікавою інформацією.

Недолік пошуку в тому, що часто відбувається неякісна сортування і у відповідь на запит видається багато документів, що містять шукане слово (або словосполучення) в іншому контексті - це заважає правильній фільтрації інформації та ускладнює її вивчення [9].

Рубрики - це важливий елемент навігації будь-якого сайту. Дуже часто меню рубрик стає де-факто головним меню, а як би основне меню відходить в тінь. Хороший приклад - даний сайт: головне меню, яке складається зі сторінок, займає другорядну роль, тоді як меню рубрик забезпечує основну навігацію по змісту сайту.

Мітки класифікують зміст сайту по частоті використання певних тем. Вага мітки тим більше, чим більшої кількості записів ця мітка присвоєна.



Рисунок 3.1 – Структурна схема веб сайту торгово-розважального центру

3.2 Керівництво програміста

3.2.1. Налаштування сервера

В якості дистрибутива для установки, що включає до свого складу веб сервер ми вибрали XAMPP, саме даний дистрибутив включає в себе MySQL, Apache,

FileZilla [10]. Після завантаження дистрибутива XAMPP запускаємо установку (рис.3.2).

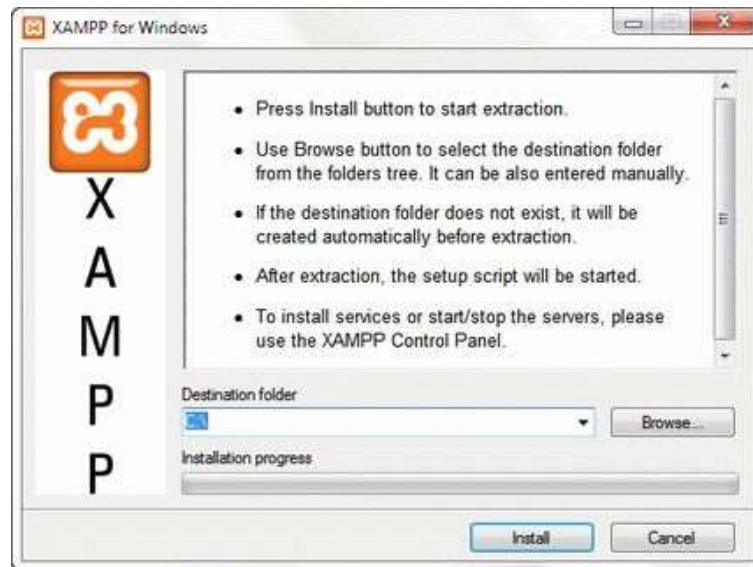


Рисунок 3.2 – Установка веб-сервера

За замовчуванням шлях для установки веб сервера вказано диск C: \. Як видно на рисунку 3.3 відбувається вилучення файлів.



Рисунок 3.3 – Вилучення файлів

Після завершення установки, дивимося в трей, там знаходиться значок XAMPP натискаємо на нього і перед нами з'являється панель управління XAMPP

(рис. 3.4). Встановлюємо галочки на модулях Apache, MySQL і звертаємо вікно управління XAMPP (рис.3.5).

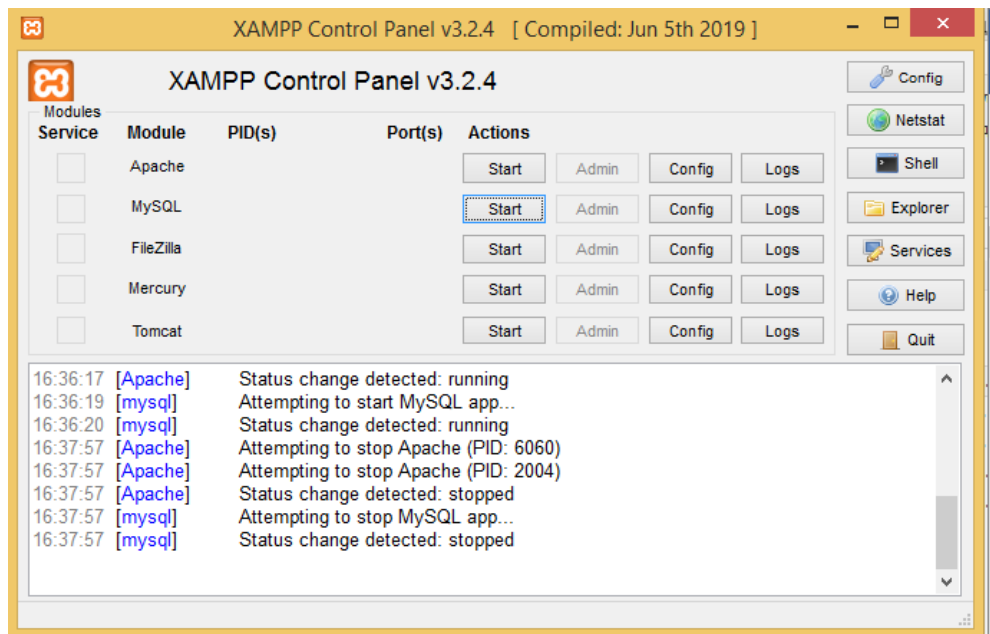


Рисунок 3.4 – Панель управління веб-сервера

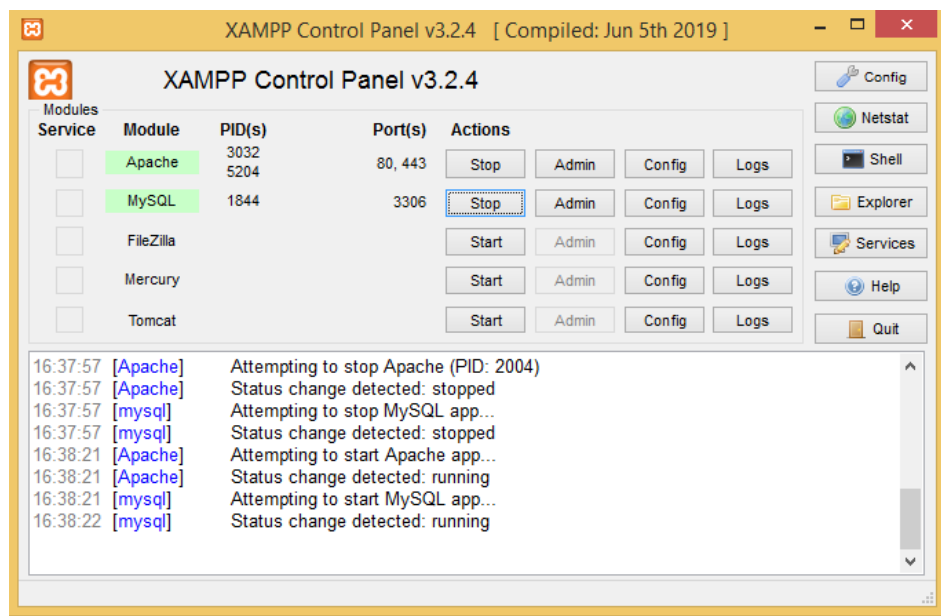


Рисунок 3.5 - Запуск веб-сервера

Перевірка установки XAMPP. Запускаємо браузер і вводим наступний URL-адресу: <http://localhost>. Відкриється сторінка вітання XAMPP (рис.3.6).



Рисунок 3.6 - Сторінка вітання XAMPP

3.2.2 CMS WordPress

WordPress - одна з найпопулярніших в світі інформаційних систем для забезпечення і організації процесу створення, редагування і управління вмістом (CMS). Вона надає всі необхідні функції для створення блогу, новинного сайту (і, зрозуміло, для сателітів). Ця сучасна платформа зорієнтована на зручність використання, мережеві стандарти і, природно, естетичну привабливість. WordPress блог безкоштовний і вільний до розповсюдження. Мова написання - PHP, в якості бази даних використовується MySQL, поширюється під універсальної загальнодоступній ліцензією GNU [11].

Процес установки wordpress, дуже простий і зводиться до того, що після скачування бажаної версії wordpress, потрібно розпакувати і скопіювати пакет в потрібну директорію на сервер і тут не потрібно додаткову освіту у Великій Британії. Це можна робити навіть не зі свого компа, а за допомогою спеціальної програми віддаленого управління - установка TightVNC. Потім слід звернутися до цієї директорії з браузера за потрібною адресою і дати відповіді на кілька запитань.

Можливості WordPress.

- інформаційна система дозволяє створювати публікації за допомогою сторонніх програм;
- установка і настройка не представляють ніякої складності;
- підтримує функції створення автоматично оброблюваних списків оновлень RSS, Atom, trackback, pingback;
- унікальна простота системи взаємодії з кодом модулів (плагінів);
- реалізація «тем», як наборів файлів - шаблонів (на PHP), досить помітно збільшує швидкість і гнучкість;
- існує можливість легко змінювати зовнішній вигляд і способи виведення даних за рахунок підтримки «тем»;
- невичерпні обсяги бібліотек «тем» і «плагінів»;
- доступна реалізація складних рішень за рахунок закладеного потенціалу архітектури;
- пропонується використання людино-зрозумілих URL;
- присутній система російського перекладу.

Після установки Wordpress копіюємо його в C:\ WebServers \ home \ localhost \ www. У файлі wp-config.php пишемо назву бази даних, користувача і пароль, в результаті чого отримуємо:

Ім'я бази даних для WordPress:

```
define('DB_NAME', 'database_wordpress');
```

MySQL - имя пользователя:

```
define('DB_USER', 'root');
```

MySQL - пароль базы данных

```
define('DB_PASSWORD', '');
```

MySQL сервер - иногда тАнонребуется изменять это значение.

```
define('DB_HOST', 'localhost').
```

Далі необхідно створити з такою ж назвою базу даних в phpmyadmin.

У директорії з темами wordpress:

```
C:\WebServers\home\localhost\www\wordpress\wp-content\themes
```

Копіюємо папку з вихідним шаблоном. Для того, щоб wordpress міг

використовувати дану тему обов'язково необхідний файл `index.php`, який є шаблоном майже для всіх сторінок, тобто всі його частини будуть виводиться практично на всіх сторінках за винятком текстів.

Створюємо шаблон в папці:

`C:\WebServers\home\localhost\www\wordpress\wp-content\themes\template`, в якому використовуються функції для виведення title десятків заголовків футера.

Наведемо основні функції:

Рядок: `<? Php get_template_part ('content', 'page'); ?>` Є php-скриптом, в якому є функція `get_template_part`, що виводить вміст сторінок з адмінки wordpress по її назві. Тобто, переходячи, наприклад, на сторінку `http://localhost/wordpress/main-4/` береться вміст сторінки `main-4`.

Для відображення зображення і флеш необхідно прописати url-адреса теми, наприклад, для css стилів:

```
<Link href = "http://localhost/wordpress/wp-content/themes/template4/style.css" rel = "stylesheet" type = "text / css">.
```

Файл `content-page.php` здійснює вибірку текстів по сторінці.

Віджети в WordPress грають роль модулів. За допомогою віджетів можна вивести інформацію на сайт. Наприклад: сторінки, категорії (рубрики), хмара тегів, архів записів і т.д. Встановлюються вони разом з плагінами.

Віджети - це блоки, які можна розміщувати в колонках сайту. Вбудовані в WordPress віджети дозволяють додати наступну інформацію в колонках блогу:

- список категорій;
- список сторінок;
- архіви;
- посилання;
- останні повідомлення;
- останні коментарі;
- календар;
- RSS потоки.

І найкорисніше: текстові блоки. У текстовому віджеті можна розмістити html-код, JavaScript або простий текст. Цей віджет можна використовувати для розміщення коду різних лічильників, рекламних блоків і тому подібного. Кількість текстових віджетів можна варіювати від 1 до 9 (рис.3.7).

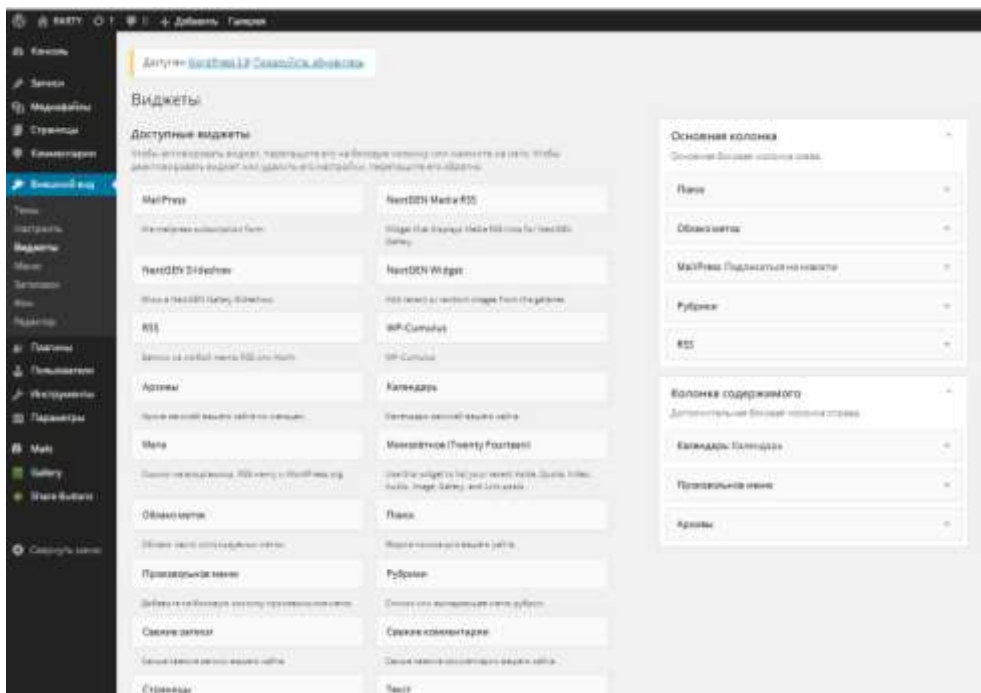


Рисунок 3.7 – Віджети сайту

Теми, які мають великий вибір місць для розміщення різних віджетів, подобаються користувачам і зручні для настройки. Перший крок на шляху до використання віджетів на створюваному сайті під керуванням WordPress - зробити тему сайту доступною для застосування віджетів. Завдання досить просто вирішується, якщо помістити правильний код в правильне місце.

Наступним етапом ми внесли власні корективи в стилі веб-сайту.

Лістинг 3.1 – Стилль стартової панелі

```
.welcome-panel .welcome-icon:before,
#dashboard_right_now li a:before,
#dashboard_right_now li span:before,
#dashboard_right_now .search-engines-info:before {
    color: #646970;
    font: normal 20px/1 dashicons;
    speak: never;
```

```

display: inline-block;
padding: 0 10px 0 0;
position: relative;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
text-decoration: none !important;
vertical-align: top;
}

```

Функції для роботи з категоріями товарів

Лістинг 3.2 – Функції для роботи з категоріями товарів

```

function get_categories( $args = '' ) {
    $defaults = array( 'taxonomy' => 'category' );
    $args     = wp_parse_args( $args, $defaults );
    $args['taxonomy'] = apply_filters(
'get_categories_taxonomy', $args['taxonomy'], $args );
    if ( isset( $args['type'] ) && 'link' === $args['type'] ) {
        _deprecated_argument(
            __FUNCTION__,
            '3.0.0',
            sprintf(
                /* translators: 1: "type => link", 2:
"taxonomy => link_category" */
                __( '%1$s is deprecated. Use %2$s instead.' ),
                '<code>type => link</code>',
                '<code>taxonomy => link_category</code>'
            )
        );
        $args['taxonomy'] = 'link_category';
    }

    $categories = get_terms( $args );

    if ( is_wp_error( $categories ) ) {
        $categories = array();
    } else {
        $categories = (array) $categories;
        foreach ( array_keys( $categories ) as $k ) {
            _make_cat_compat( $categories[ $k ] );
        }
    }
}

```

Клас `php` для збереження коментарів, сутності класу приведені на лістингу 3.3. Для даного класу важливо берегти інформацію про автора коментаря, його персональні данні, час, коли залиши коментар, ідентифікатор посту.

Лістинг 3.3 – Робота з коментарями

```

public $comment_ID;
public $comment_post_ID = 0;
public $comment_author = '';
public $comment_author_email = '';
public $comment_author_url = '';
public $comment_author_IP = '';
public $comment_date = '0000-00-00 00:00:00';
public $comment_date_gmt = '0000-00-00 00:00:00';
public $comment_content;
public $comment_karma = 0;
public $comment_approved = '1';
public $comment_agent = '';
public $comment_type = 'comment';
public $comment_parent = 0;
public $user_id = 0;
protected $children;
protected $populated_children = false;
protected $post_fields = array( 'post_author', 'post_date',
'post_date_gmt', 'post_content', 'post_title', 'post_excerpt',
'post_status', 'comment_status', 'ping_status', 'post_name',
'to_ping', 'pinged', 'post_modified', 'post_modified_gmt',
'post_content_filtered', 'post_parent', 'guid', 'menu_order',
'post_type', 'post_mime_type', 'comment_count' );

```

Функції класу коментарів наведено на лістингу 3.4.

Лістинг 3.4 – Функції роботи з коментарями

```

public static function get_instance( $id ) {
    global $wpdb;

    $comment_id = (int) $id;
    if ( ! $comment_id ) {
        return false;
    }

    $_comment = wp_cache_get( $comment_id, 'comment' );

    if ( ! $_comment ) {
        $_comment = $wpdb->get_row( $wpdb->prepare(
"SELECT * FROM $wpdb->comments WHERE comment_ID = %d LIMIT 1",
$comment_id ) );
        if ( ! $_comment ) {
            return false;
        }

        wp_cache_add( $_comment->comment_ID, $_comment,
'comment' );
    }

    return new WP_Comment( $_comment );
}

```

Функції класу створення нового допису наведено на лістингу 3.5.

Лістинг 3.5 – Файл створення ного допису

```

$content = apply_filters( 'wp_mail_original_content',
$content );
if ( false !== stripos( $content_transfer_encoding,
'quoted-printable' ) ) {
    $content = quoted_printable_decode( $content ); }

if ( function_exists( 'iconv' ) && ! empty( $charset ) ) {
    $content = iconv( $charset, get_option( 'blog_charset'
), $content );}
$content = explode( $phone_delim, $content );
$content = empty( $content[1] ) ? $content[0] :
$content[1];
$content = trim( $content );
$post_content = apply_filters( 'phone_content', $content );
$post_title = xmlrpc_getposttitle( $content );

if ( '' === trim( $post_title ) ) {
    $post_title = $subject; }
$post_category = array( get_option(
'default_email_category' ) );
$post_data = compact( 'post_content', 'post_title',
'post_date', 'post_date_gmt', 'post_author', 'post_category',
'post_status' );
$post_data = wp_slash( $post_data );
$post_ID = wp_insert_post( $post_data );
if ( is_wp_error( $post_ID ) ) {
    echo "\n" . $post_ID->get_error_message(); }
if ( empty( $post_ID ) ) {
    continue; }
do_action( 'publish_phone', $post_ID );
echo "\n<p><strong>" . __( 'Author:' ) . '</strong> ' .
esc_html( $post_author ) . '</p>';
echo "\n<p><strong>" . __( 'Posted title:' ) . '</strong> '
. esc_html( $post_title ) . '</p>';
if ( ! $pop3->delete( $i ) ) {
    echo '<p>' . sprintf(
        /* translators: %s: POP3 error. */
        __( 'Oops: %s' ),
        esc_html( $pop3->ERROR )
    ) . '</p>';
    $pop3->reset();
    exit;
} else {
    echo '<p>' . sprintf(
        /* translators: %s: The message ID. */
        __( 'Mission complete. Message %s deleted.' ),
        '<strong>' . $i . '</strong>'
    ) . '</p>'; }}

$pop3->quit();

```

3.3 Керівництво користувача

Веб сайт має дружній інтерфейс, побудований на активному діалозі з користувачем. Він являє собою графічний інтерфейс користувача, розташований на php - сторінках.

Частина графічного інтерфейсу користувача призначена для здійснення вибірки даних про заходи відповідно до параметрів запити, розміщених в СУБД MySQL. Вона представляла собою набір форм, що складається з текстового поля і кнопки "вибрати". У текстове поле потрібно вводити параметри запити в БД.

Для успішної роботи з графічним інтерфейсом, користувач повинен використовувати маніпулятор- «миша» і клавіатуру персонального комп'ютера.

Після набору в командному рядку браузера Internet - адреси, <http://localhost/party/index.php> перед користувачем з'являється вікно, зображене на рисунку 3.9 (тут ім'я_сервера - назва сервера в файлі конфігурації операційної системи, party - назва директорії з файлами комп'ютерної системи).

На рис. 3.10- 3.11 наведені сторінка редагування новини та сторінка створення нового користувача.



Рисунок 3.9 – Головна сторінка

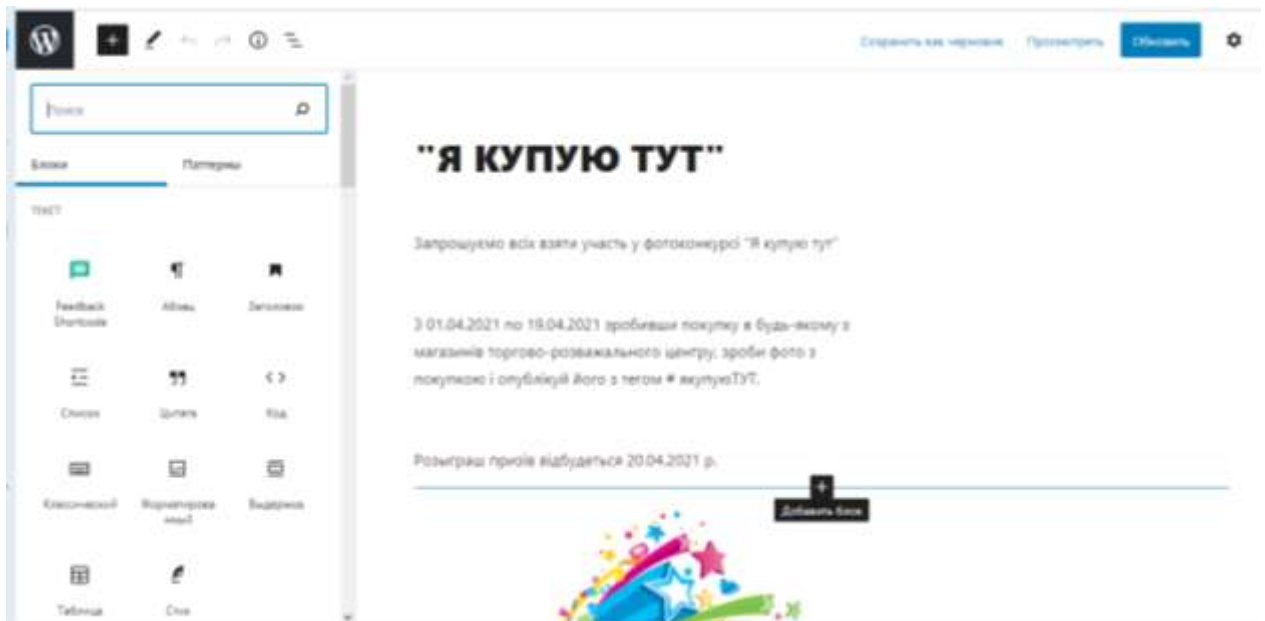


Рисунок 3.10-Редагування новини

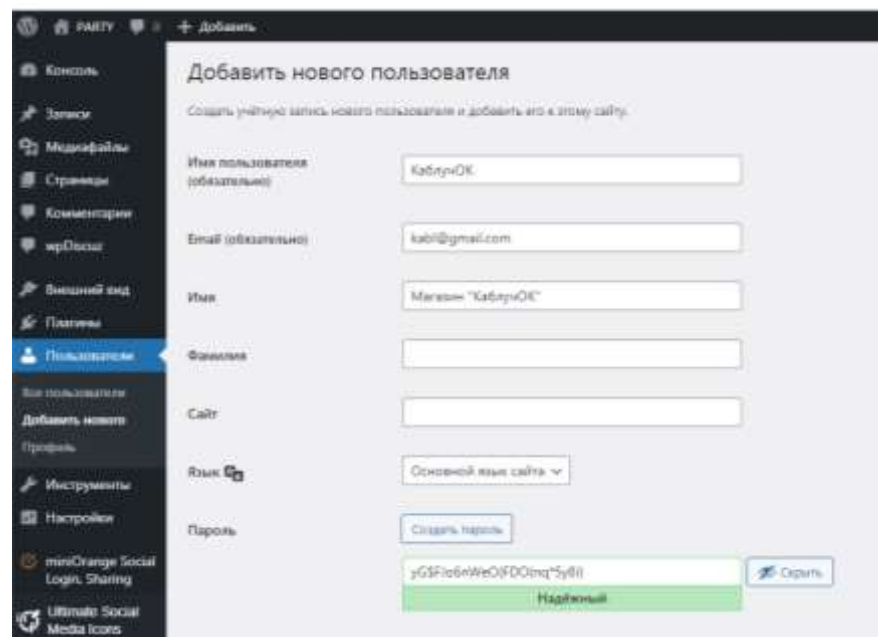


Рисунок 3.11 – Створення користувача

4 ОПТИМІЗАЦІЯ ТА НАЛАШТУВАННЯ САЙТУ

Для оптимізації знаходження сайту потенційними покупцями у мережі Internet було розроблене семантично ядро, що є впорядкованим списком запитів

користувачів на вибрану тему відсортовані від найбільш частого запиту до найменшого (табл. 4.1).

Таблиця 4.1 – Ключові слова за запитом «торговий центр»

Ключові слова	
торговий центр київ	
торговий центр гулівер	
торговий центр україна	
торговий центр проспект	
торговий центр олімпійський	
торговий центр блокбастер	
торговий центр львів	
торговий центр аркадія	
торговий центр аладдін	
торговий центр аракс	
торговий центр англійською	
торговий центр апетит	
торговий центр арсен	
торговий центр ашан львів	
торговий центр арсен львів	
торговий центр а	
торговий центр більшовик	
торговий центр бровари	
торговий центр большевик	
торговий центр біла церква	
торговий центр боярка	
торговий центр бориспіль	
торговий центр біла церква графік роботи	
городоцька 286-б торговий центр	
торговий центр б-52	
торговий центр вінниця	
торговий центр варшавський	
торговий центр в тернополі	
торговий центр варшава	
торговий центр в луцьку	
торговий центр варшавський луцьк	
торговий центр вернісаж львів	
торговий центр в чернівцях	
торговий центр в києві	
торговий центр в львові	

Проаналізувавши перелік найбільш поширених ключових фраз можна сказати, що для веб-сайту торгово-розважального центру обов'язково треба вказати

місто та адресу розташування.

Далі на кожній сторінці були додані заголовки сторінки, опис та ключові слова. Наприклад для сторінки мистецтво ці пункти будуть такими:

Заголовок сторінки: Інформація про мистецькі події, які проходять на території торгово-розважального центру;

Опис сторінки: Сайт для пошуку та придбання різних товарів побуду та одягу для всієї сім'ї. Платформа для проведення масових заходів;

Ключові слова: торговий центр, розважальний центр, мистецтво, події міста, афіша, онлайн покупки, акції, нова колекція, квитки, товари для дітей, дитячий день народження.

Внутрішня структура сайту була оптимізована для пошуку пошукових систем за допомогою розміщення тегів різних рівнів від h1 до h4 включно. h1 використано у хедери та застосовано стиль `display: none`. Людина його не побачить за ненадібністю, а пошукові системи проаналізують.

У тегу h2 розміщені всі назви категорій та розділів. У тегу h3 – назви типів за категоріями. Теги іншого рівня впроваджені за потребою.

Для підвищення існуючої ступені оптимізації для пошукових машин у кожний тег зображення було прописано атрибут `alt`, який застосовується браузером при слабкому підключенні до мережі або якщо користувач відключив завантаження зображень.

Крім усього іншого хедер є доступним на будь-якій сторінці та містить гіперпосилання на усі існуючі сторінки.

Для успішного просування сайту у глобальній мережі Internet сайт був зареєстрований в пошуковій системі `google.com`, де його було проіндексовано.

Індексація - це процес збору інформації пошуковою системою про вміст вашого сайту. Індексація проводиться за допомогою повного сканування усіх сторінок сайту на предмет тексту, мета тегів, зображень, відео контенту. Ця інформація буде занесена у базу даних пошукової системи та використана під час пошуку.

Для проведення успішної індексації потрібно перейти до пошукової консолі

обраної пошукової системи та завантажити файл sitemap.xml, що містить усі посилання. Результат успішної індексації можна побачити на рис. 4.1.



Рисунок 4.1 – Результат готовності до індексації сайту в Google Search Console

Крім того для успішної індексації потрібно долучити файл robots.txt, який містить технічну інформацію для пошукових машин та введений у лістингу

Лістинг – Файл robox.txt

```
# Group 1
User-agent: Googlebot
Disallow: /photo/

# Group 2
User-agent: *
Allow: /

Sitemap: http://www.party.su/sitemap.xml
```

Файл robots.txt або індексний файл - звичайний текстовий документ в кодуванні UTF-8, діє для протоколів http, https, а також FTP. Файл дає пошуковим роботам рекомендації: які сторінки / файли варто сканувати. Якщо файл буде містити символи не в UTF-8, а в іншому кодуванні, пошукові роботи можуть неправильно їх обробити. Правила, перераховані у файлі robots.txt, дійсні лише щодо того хоста, протоколу і номера порту, де розміщений файл.

Файл повинен розташовуватися в кореневому каталозі в вигляді звичайного текстового документа і бути доступний за основною адресою сайту.

ВИСНОВКИ

В бакалаврській роботі була розроблена веб-система торгово-розважального центру, яка реалізує послуги продажу товарів і публікації актуальних новин. Кожен модуль має своє функціональне призначення, яке дозволить значно скоротити процес доступу та систематизації необхідної інформації, а так само процес роботи центру.

В ході роботи було проведено аналіз вихідних даних і вимог до системи, на основі якого в подальшому будувався вибір апаратного та програмного забезпечення системи. Розроблена база даних, яка зберігає інформацію про заходи даного центру. База даних розроблена за допомогою MySQL Workbench. Веб сайт цієї системи має розширене меню і додаткові модулі. При розробці використовувалася CMS WordPress. Сайт ефективний для комунікації та є вдалим інструментом для тих, прагнем дізнатися про актуальні події чи акції. Це ідеальне місце для онлайн шопінгу.

При створенні веб-системи вирішені наступні завдання:

- проаналізовано технічне завдання;
- розроблена специфікацій вимог, UML діаграми;
- розроблена база даних;
- розроблені макети сторінок;
- розроблено веб сайту за допомогою CMS.

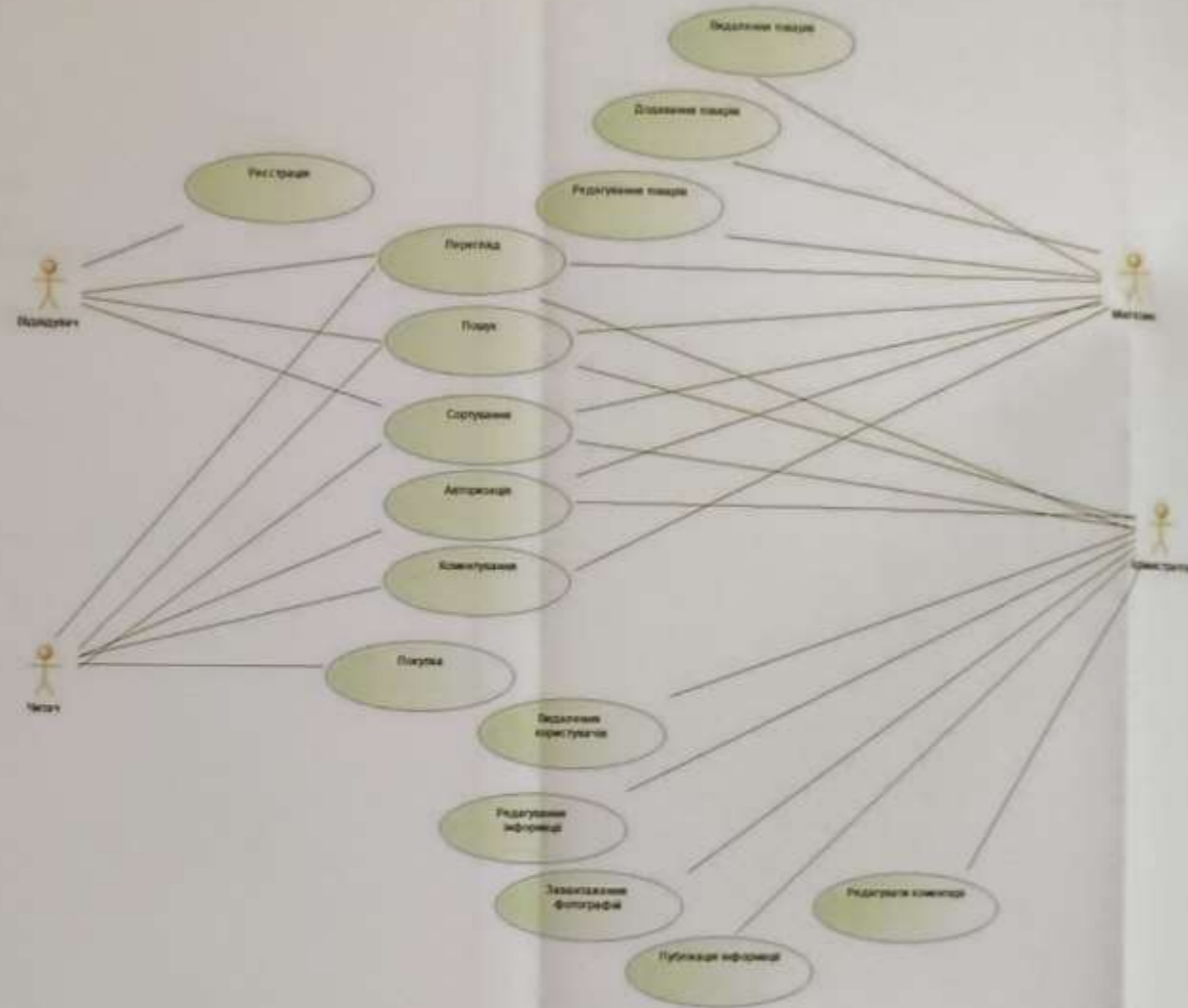
Таким чином, можна констатувати, що поставлена мета бакалаврської роботи досягнута.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Mario A. Making Isometric Social Real–Time Games: With HTML5, CSS3 and JavaScript. - М.: O'Reilly, 2011. - 156 с.
2. Васильев А. Н. Самоучитель Java с примерами и программами. - М.: Наука и техника, 2013. - 368 с.
3. Гонсалвес Э. Изучаем Java EE 7. - М.: "Издательство "Питер", 2016. - 640 с.
4. Дронов В. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. – М.: БХВ-Петербург, 2011. - 416 с.
5. Скотт М. 5 проектов Web-сайтов от фотоальбома до магазина. - М.: ИТ Пресс – Москва, 2013. - 224 с.
6. Сравнение фронтенд-фреймворков в реальных условиях [Электронный ресурс]. – М.: Редакция techrocks.ru, 2017.- Режим доступа: <https://techrocks.ru/2017/12/24/front-end-frameworks-comparison/>
7. Котеров Д.В. PHP7: Полное руководство / Д.В. Котеров, И.В.Симдянов.- СПб: Питер, 2016.- 1088 с.
8. Форсье Д. Django: Разработка веб-приложений на Python.- СПб.: Питер, 2009.- 458 с.
9. Стаффер М. Laravel. Полное руководство 2-е издание.- СПб.: Питер, 2020.- 512 с.
10. Файни, К. SQL: Руководство по изучению языка. – М: ДМК, 2019.- 456 с.
11. Пономаренко, Н.Н. Введение в веб-программирование. HTML-разметка. - ХАИ, 2013. – 49 с.
12. Фридман В. и др. Строительство Web-сайтов / В.А. Фридман, А.В. Александров, Г.Г. Сергеев, С.П. Костин. М.: Триумф, 2011. - 288 с.

13. Энж Э., Спенсер С., Фишкин Р., Стрикчиола Д. SEO. Искусство раскрутки сайтов / Э. Энж, С. Спенсер, Р. Фишкин, Д. Стрикчиола. – М.: БХВ-Петербург, 2014. - 668 с.
14. Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов. - М.: Эксмо, 2018. - 208 с.
15. Макнейл П. Веб-дизайн. Книга идей веб-разработчика. - СПб.: Питер, 2017. - 480 с.
16. Гаевский А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. - М.: Триумф, 2015. - 464 с.
17. Хассей Т. WordPress. Создание сайтов для начинающих. - М.: Эксмо, 2016. - 538 с.
18. Кузнецов РНР. Практика создания Web-сайтов. - М.: БХВ-Петербург, 2015. - 102 с.
19. Сырых Ю.А. Современный веб-дизайн. Настольный и мобильный. - М.: Вильямс, 2014. - 384 с.
20. Голомбински К. Добавь воздуха! Основы визуального дизайна для графики веб и мультимедиа / К. Голомбински, Р. Хаген; Пер. с англ. Н.А. Римицан.. - СПб.: Питер, 2013. - 272 с.

Діаграма варіантів використання веб-системи торгово-розважального центру



Тим.	Лист	№ докум.	Підп.	Дата
Розроб.		Кудак С.О.	<i>[Signature]</i>	
Перев.		Киричка Г.Г.	<i>[Signature]</i>	
Т.контр.				
Н.контр.		Шербан М.В.	<i>[Signature]</i>	
Затв.		Кудак С.О.	<i>[Signature]</i>	

Розробка веб-системи торгово-розважального центру

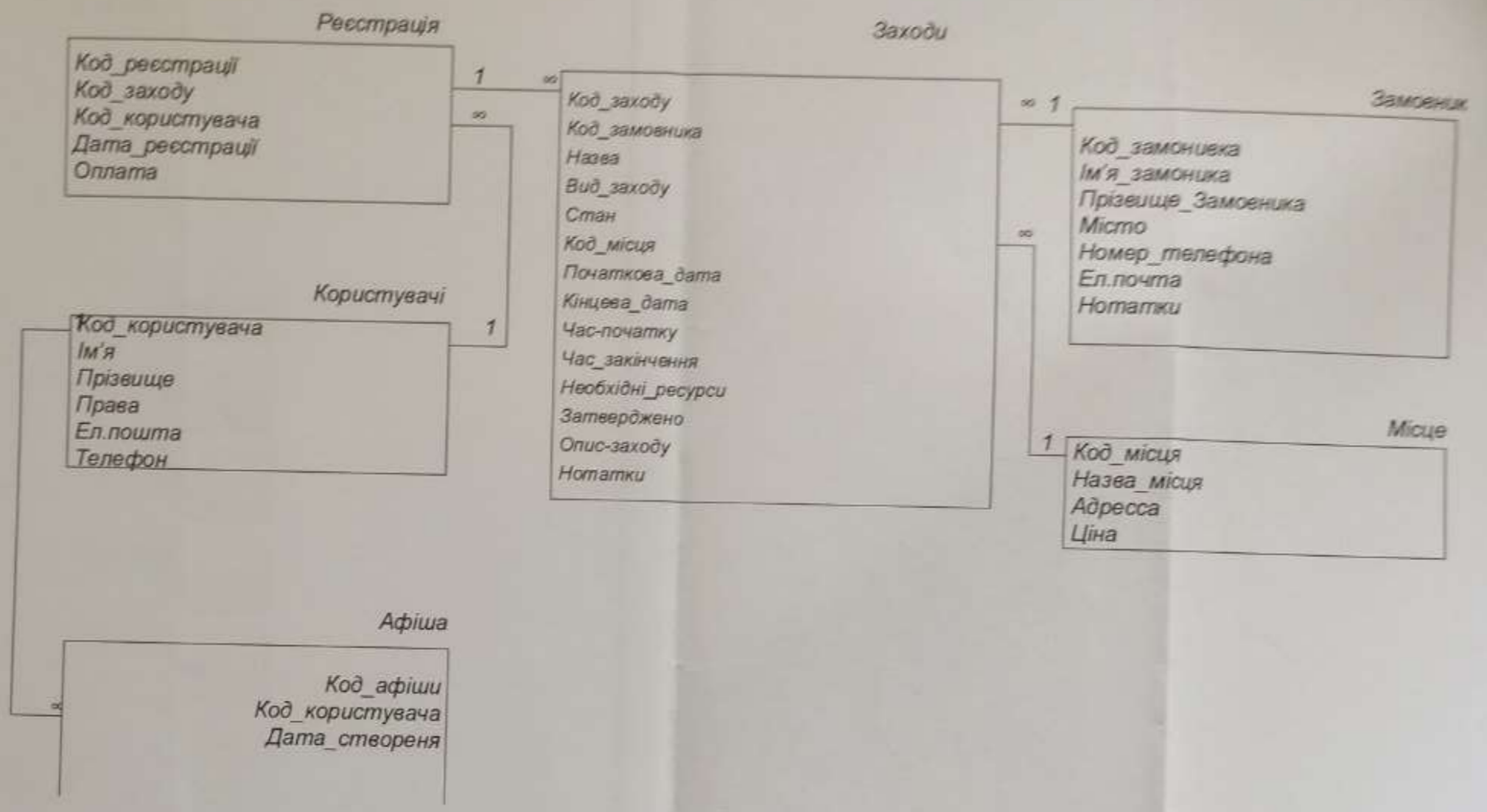
Діаграма варіантів використання веб-системи торгово-розважального центру

Лист	Маса	Масштаб
Лист 1		
Листів 1		

ІНУ «Запорізька політехніка»
КНТ-517

13.02070849.00012 ПЛІ2

Реляційна модель бази даних



Зам.	Лист	№ докум.	Підп.	Дата
Розроб.		Куцак С.О.	<i>[Signature]</i>	
Перев.		Киричак Г.Г.	<i>[Signature]</i>	
Т.контр.				
Н.контр.		Щербак Н.В.	<i>[Signature]</i>	
Затв.		Кудеряшова Р.К.	<i>[Signature]</i>	

13.02070849.00012 ПЛІ2

Розробка веб системи торгово-розважального центру

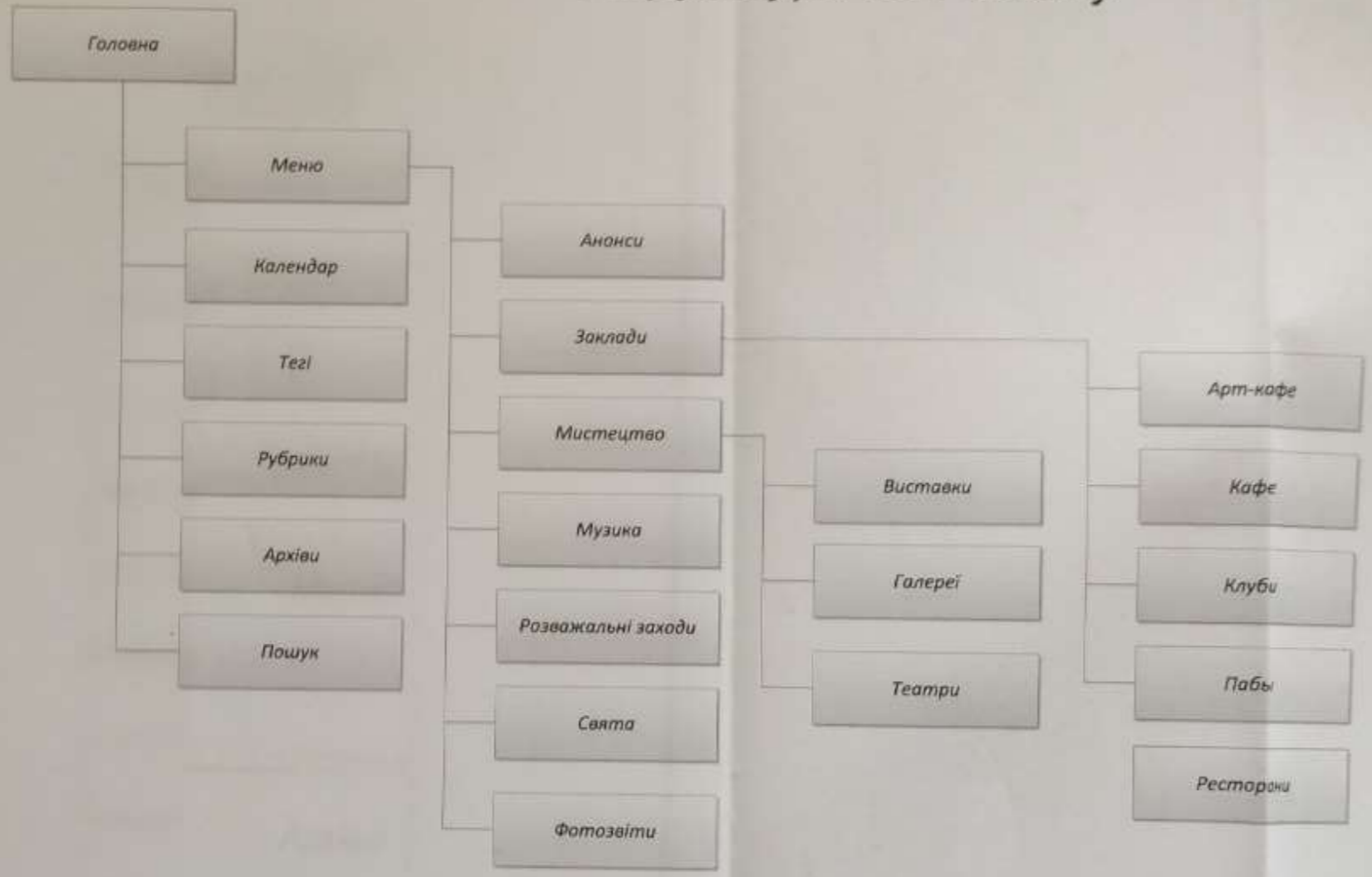
Реляційна модель бази даних

Лист	Маса	Масшт.
Лист 1		Листів 1

НУ «Запорізька політехніка»
КНТ-517

13.02070849.00012 ПЛЗ

Структура веб-сайту



					13.02070849.00012 ПЛЗ			
Зам.	Лист.	№ докум.	Підп.	Зам.	Лист.	Маса	Масшт.	
Розроб.		Кудак С.О.	<i>[Signature]</i>					
Перев.		Киричев Г.Г.	<i>[Signature]</i>					
Уконтр.								
Н контр.		Шербак Н.В.	<i>[Signature]</i>					
Зам.		Кудеримов Р.А.	<i>[Signature]</i>					
					Розробка веб системи торгово-розважального центру		Лист 1	Листів 1
					Структура веб-сайту		ІТД «Харківська політехнічна» КНТ-517	

13.02070849.00012 ПЛ4

Інтерфейс веб-сайту

The screenshot shows a website interface for 'PARTY'. The main content area features a news article titled '« Я КУПУЮ ТУТ »' (I Buy Here) with a colorful graphic and the word 'конкурс' (contest). The interface includes a top navigation bar with 'Меню1' and 'Пошук' (Search), a left sidebar with 'Пошук', 'Мітки' (Tags), 'Мітки' (Tags), 'Користувачі' (Users), 'Рубрики' (Categories), and 'Підписка RSS', and a right sidebar with 'Календар' (Calendar), 'Меню2', and 'Архіви' (Archives).

					13.02070849.00012 ПЛ4		
Зам.	Лист.	№ докум.	Підп.	Дата	Лист	Маса	Масшт.
Розроб.		Кудак С.О.	<i>[Signature]</i>				
Перев.		Киричук Г.Г.	<i>[Signature]</i>				
Т.контр.							
Н.контр.		Шербак Н.В.	<i>[Signature]</i>				
Затв.		Кудеримов Р.К.	<i>[Signature]</i>				
Розробка веб системи торгово-розважального центру					Лист 1	Листів 1	
					Інтерфейс веб-сайту		НУ «Запорізька політехніка» КНТ-517