

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

Довідково-інформаційна система комп'ютерного клубу
Назва теми

Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Шифр КвРІПЗ.190143.01.21.ПЗ

Виконав студент IV курсу, група ПЗ-19-1


Підпис

Т. Ф. Шаріфов
Ініціали, прізвище

Керівник канд. техн. наук, доцент


Підпис

Г. І. Радельчук
Ініціали, прізвище

Нормоконтролер канд. техн. наук, доцент


Підпис

Г. І. Радельчук
Ініціали, прізвище

До захисту допускаю:
Завідувач кафедри інженерії
програмного забезпечення


Підпис

Л. П. Бедратюк
Ініціали, прізвище

19 06 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Л. П. Бедратюк

02 01 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Шаріфову Тимуру Фармановичу

Прізвище, ім'я, по батькові студента

1. Тема кваліфікаційної роботи Довідково-інформаційна система комп'ютерного клубу

Керівник кваліфікаційної роботи Радельчук Галина Іванівна, канд. техн. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. № 5

2. Строк подання студентом роботи на кафедру 01.06.2023 р.

3. Вихідні дані до роботи Матеріали переддипломної практики

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

1) Дослідження предметної області та постановка задачі

2) Проектування довідково-інформаційної системи

3) Програмна реалізація та тестування системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Три креслення у ф. А3:

1) UML-діаграма варіантів використання

2) UML-діаграма послідовності реєстрації клієнта клубу

3) Модель бази даних системи

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Радельчук Г. І., доцент кафедри ІПЗ	<i>Г.І.Р.</i> 10.06.23	<i>Г.І.Р.</i> 16.06.23
Антиплагіат	Гурман І. В., доцент кафедри ІПЗ	<i>І.В.Г.</i> 10.06.23	<i>І.В.Г.</i> 16.06.23

7. Дата видачі завдання « 02 » січня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Збір матеріалу за темою кваліфікаційної роботи (КвР); дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання.	02.01 – 31.01.2023	
2 Проектування довідково-інформаційної системи. Розробка графічної частини КвР.	01.02 – 28.02.2023	
3 Програмна реалізація та тестування системи.	01.03 – 30.04.2023	
4 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог стандартів.	01.05 – 25.05.2023	
5 Попередній захист КвР.	Травень 2023 (згідно графіка)	
6 Перевірка КвР на плагіат, нормконтроль, отримання відгуків та рецензій. Брошурування (зшиття) пояснювальної записки.	26.05 – 30.05.2023	
7 Здача КвР на кафедру; підготовка КвР для розміщення у репозитарії ХНУ; підготовка до захисту та захист КвР.	з 01.06.2023	

Студент

Т.Ф.Ш.
Підпис

Т. Ф. Шаріфов

Керівник роботи

Г.І.Р.
Підпис

Г. І. Радельчук

АНОТАЦІЯ

Тема кваліфікаційної роботи: Довідково-інформаційна система комп'ютерного клубу.

Автор роботи: Шаріфов Гімур Фарманович.

Керівник роботи: Радельчук Галина Іванівна.

Пояснювальна записка: 65 с., 26 рис., 11 табл., 3 дод., 41 джерело.

Графічна частина: три креслення у ф. А3.

КОМП'ЮТЕРНИЙ КЛУБ, ДОВІДКОВО-ІНФОРМАЦІЙНА СИСТЕМА,
БАЗА ДАНИХ, ВЕБ-ЗАСТОСУНОК, CSS, PHP, MySQL.

Мета кваліфікаційної роботи: розробка довідково-інформаційної системи комп'ютерного клубу.

У кваліфікаційній роботі проведено аналіз предметної області та її інформаційного забезпечення, визначено вимоги до довідково-інформаційної системи, розроблено її архітектуру та спроектована структура бази даних комп'ютерного клубу і самого додатку.

При розробці системи використано мову програмування PHP, сервер бази даних MySQL та таблиці стилів CSS.

Практична значимість отриманих результатів полягає у розробці готового до використання програмного продукту у вигляді веб-додатку.

Впровадження довідково-інформаційної системи дозволяє оптимізувати роботу комп'ютерного клубу, надає можливість розмежування роботи адміністратора і користувача та значно полегшити використання застосунку.

16.06.2023
Дата


Підпис

ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ.190143.01.21.ПЗ	Пояснювальна записка	65		
2	A4		Завдання на кваліфікаційну роботу	1		
3	A4		Анотація	1		
			<u>Графічна частина</u>			
4	A3	КвРІПЗ.190143.01.21.E8	UML-діаграма варіантів використання	1		
5	A3	КвРІПЗ.190143.01.21.E8	UML-діаграма послідовності реєстрації клієнта клубу	1		
6	A3	КвРІПЗ.190143.01.21.E8	Модель бази даних системи	1		

КвРІПЗ.190143.01.21.ВД

Змн.	Арк.	№ докум.	Грипис	Дата
Виконав		Шаріфове Т. Ф.	<i>[Signature]</i>	16.06
Керівник		Радельчук Г. І.	<i>[Signature]</i>	16.06
Н. контр.		Радельчук Г. І.	<i>[Signature]</i>	16.06
Зав. каф.		Бедратюк Л. П.	<i>[Signature]</i>	19.06

Довідково-інформаційна система комп'ютерного клубу

Відомість документів

Літ.	Арк.	Аркуше
	1	1

ХНУ, ПЗ-19-1

ЗМІСТ

Вступ.....	5
1 Дослідження предметної області та постановка задачі.....	10
1.1 Аналіз змісту і опис предметної області, її функціональних та структурних особливостей.....	10
1.2 Аналіз наявного програмно-технічного забезпечення предметної області	14
1.3 Визначення вимог до довідково-інформаційної системи	17
1.4 Постановка задачі.....	19
2 Проектування довідково-інформаційної системи.....	21
2.1 Проектування архітектури довідково-інформаційної системи	21
2.2 Проектування структури бази даних.....	26
2.3 Аналіз та вибір технологій і засобів реалізації системи	28
2.4 Висновок до розділу 2	33
3 Програмна реалізація та тестування системи.....	34
3.1 Реалізація бази даних.....	34
3.2 Структура та функціональне призначення модулів системи	48
3.3 Інструкція користувача.....	51
3.4 Технічні характеристики системи	52
3.5 Тестування системи	53
3.6 Висновок до розділу 3	60
Висновки	61
Перелік джерел посилання.....	62
Додаток А Технічне завдання	66
Додаток Б Код (лістинг) програми	69
Додаток В Презентаційні матеріали.....	78
Графічна частина.....	86

КвРІПЗ.190143.01.21.ПЗ								
Змн.	Арк.	№ докум.	Підпис	Дата	Довідково-інформаційна система комп'ютерного клубу.	Літ.	Арк.	Акрушів
Розроб.		Шаріфов Т. Ф.	<i>[Підпис]</i>	16.06			4	65
Перевір.		Радельчук Г. І.	<i>[Підпис]</i>	16.06	Пояснювальна записка	ХНУ, ПЗ-19-1		
Реценз.								
Н. Контр.		Радельчук Г. І.	<i>[Підпис]</i>	16.06				
Затверд.		Бедратюк Л.Я.	<i>[Підпис]</i>	19.06				

ВСТУП

Ще донедавна захоплення іграми на комп'ютері викликали невдоволення з боку батьків, проте на сьогодні дохід кіберспортсменів, що займаються ними професійно, може сягати десятків мільйонів доларів. Останнє десятиліття відзначилося появою нового спортивного напрямку, який став популярним і серед українців.

Сьогодні кіберспорт є сегментом індустрії розваг, що набирає обертів у всьому світі. Масштабний ринок eSports, згідно прогнозів міжнародних аналітичних агентств ігрової індустрії, у 2019 році складав 1 млрд доларів, а до 2022 року мав досягнути 1,8 млрд, додаючи, в середньому, по 22% за рік. Прогнози на 2023 рік – 2,3 млрд. доларів [1].

Не просто комп'ютерна гра, а справжні тренування – деякі ігри стали цілими спортивними дисциплінами. Кіберспортивні змагання набули міжнародного масштабу, а у їх гравців спостерігаються такі ж психологічні та фізичні навантаження, як і в їх колег у традиційних видах спорту. А через розповсюдження кіберспорту стають популярними й комп'ютерні клуби, для яких це є важливим напрямком розвитку. Причому це відбувається в усьому світі.

У деяких містах нашої країни, кількість комп'ютерних клубів збільшилася на 145% за останні роки, а в Києві – збільшилася майже в 1,7 разів. Клуби для кіберспорту, як окремий сегмент, виявилися досить гнучким щодо змін. Так, під час пандемії коронавірусу, такі клуби почали активніше проводити саме онлайн-турніри. А відтак, деяким з них вдалося не лише не втратити, а й навіть збільшити їх аудиторію, адже люди, позбавлені звичних для них розваг, в тому числі, класичних видів спорту, могли грати в онлайн-гри. Зараз клуби, що відновлюють свою роботу, мають нову аудиторію, оскільки придбати ігровий ПК і потрібні складові може дозволити далеко не кожен.

Аудиторії комп'ютерних клубів, яка постійно зростає, потрібне підвищення доступності арен та клубів, розвиток інфраструктури ринку,

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

сучасних концепцій, нових форматів для розваг. Також, у просторі комп'ютерних клубів, можуть отримати новий етап у розвитку технології віртуальної, доповненої чи змішаної реальності. Наприклад, один із найбільших виробників ринку віртуальної реальності – компанія Valve, яка відома своїми продуктами – Counter-Strike чи Dota 2, які є найбільш популярними іграми саме серед клієнтів кіберклубів.

Кіберспорт – це змагання з комп'ютерних ігор, де беруть участь професійні гравці [2]. Найбільшим попитом з кіберспортивних предметів користуються: Counter-Strike, Global Offensive та Fortnite, Dota 2, StarCraft II, League of Legends. Найвідомішими турнірами є Evolution Championship Series (файтинг), LoL World Championship, The International (грають в Dota 2), ESL Pro League, Fortnite World Cup Finals, Intel Extreme Masters (Starcraft II, Counter-Strike: Global Offensive), PUBG Global Championship, які організовуються із мільйонними призовими фондами – для переможців і високоякісними відеотрансляціями [3].

Формальною датою появи кіберспорту є 1972 рік. Саме тоді проходили перші у світі змагання із комп'ютерної гри Spacewar (так звані «Міжгалактичні олімпійські ігри»), які згадувалися у пресі та проходили в Стенфордському університеті. Головним призом на цих змаганнях була річна підписка на видання «Rolling Stone».

Виробники цифрової техніки достатньо швидко зорієнтувалися та виявили великий потенціал у цій новій індустрії. Відтак, розпочали спонсорувати великі змагання, із уже більш цінними призами: наприклад, однією зі визначних дат для кіберспорту є червень 1997 – саме тоді американець Денніс «Thresh» Фонг на змаганнях Red Annihilation із Quake-2, в якості головного призу, отримав Ferrari 328 GTS. Машина раніше належала автору першого шутера Doom та головному програмісту id Software –Джону Кармаку. Сам турнір проводила компанія Intergraph, що придбала студію id Software, яка і є розробником Quake.

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

Відтоді грошова винагорода, що могла сягати кількох десятків тисяч доларів, стала звичайним явищем для кібертурнірів. Так, Джонатан «Fatality» Вендал, переможець дуельного турніру по Quake 3 Arena, що проходив у Далласі, отримав винагороду в 40 тис. доларів. У Швеції, на турнірі з цієї ж гри, призовий фонд був 12 тис. доларів. Через кілька місяців після того, в Quake 3 Arena змагалися вже у Великобританії, а переможці отримали приз у 10 тис. доларів.

Весь час, до 2000 року, кіберспорт залишався унікальним явищем. У цей переломний рік, відбулася подія, яку стали називати Олімпійськими іграми цифрового світу, – а саме, перший турнір World Cyber Games (WCG). Турнір дивилися мільйони глядачів, а у відбіркових етапах брало участь кілька тисяч бажаючих. У фіналі ж грало кілька сотень людей, які стали професіональними гравцями. Головною метою заходу було безпосереднє просування комп'ютерних ігор серед ширшого кола.

Кіберспортсмени WCG доводили свої вміння за допомогою Counter Strike, WarCraft 3, Quake III Arena, Starcraft: Brood War та FIFA, а з часом цей список поповнювався іншими. Організатором змагань виступила компанія з Південної Кореї – World Cyber Games Inc., яку фінансувала відома компанія Samsung. Цей фінансовий зв'язок виявився таким міцним, що після закриття Samsung фінансування у лютому 2014, то WCG теж закрили.

Учені із German Sports University старанно вивчили кіберспортсменів та дійшли висновку, що їх можна прирівнювати до справжніх спортсменів, адже представники певних галузей можуть виконувати до 400 рухів на клавіатурі та комп'ютерній миші на хвилину, а це у 4 рази більше, ніж може робити звичайна, не тренувана людина. До того ж, ці рухи асиметричні, тобто вимагають одночасної активності різних відділів мозку – така напруга не спостерігається навіть у інших спортсменів у жодному виді спорту. Рівень кортизолу – гормону стресу, в кіберспортсменів майже такий самий, як у автогонщиків. А у сполученні з швидким пульсом, стан гравців нагадує стан марафонців.

Цей спорт вимагає значної психологічної витривалості. Так, дослідження

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Університету Чичестера показують, що гравці кіберспорту зустрічаються із 51 видом стресових факторів на важливих змаганнях (наприклад, страх виступу перед людьми в реальності чи проблеми із комунікацією у команді). Рівень навантаження на психіку у них такий же, і як в спортсменів (наприклад, регбі чи футболу).

Кіберспорт, як й інші традиційні види спорту, безумовно є комерційним. Інвестори також можуть придбати команди кіберспортсменів – спонсорські контракти на сьогодні є одним із головних джерел для фінансування цієї індустрії. Також організатори можуть отримувати прибуток із прав на трансляцію та продажу квитків на ігри. Так, «Продажі квитків принесли ФІФА у 2018 на Чемпіонаті світу з футболу дохід у 541 млн доларів, а прибутки від спортивних онлайн-турнірів і ліцензування комп'ютерних ігор склали 600 млн доларів за той же період. Говорячи про Dota 2, то кількість його переглядів перевищила рекорди перегляду тенісного Вімблдону, а призовий фонд турніру «The International 2018» переважив 25 млн доларів», – говорить А. Кузьменко – старший юрист ЮФ Etema Law [4].

У кіберспортивні дисципліни в 2018 році грало більше 2 млрд. людей і кількість глядачів зростає щороку. Наприклад, у 2021 році вона зросла на 18% і становила приблизно 380 млн. доларів.

Років 20-30 назад кіберспортсменами називали відвідувачів комп'ютерних клубів: у локальних мережах щодня відбувалися битви армій у іграх-стратегіях та стрілянина в шутерах. На міжнародний рівень наші співвітчизники вийшли у 2001 році під час відбору на World Cyber Games, який був проведений в нашій країні фірмою Samsung. Тоді переможці отримали змогу поїхати на міжнародний турнір із величезним на той час призовим фондом в сотні тисяч доларів.

З 2006 року стала стрімко набирати світову популярність гра Dota (кастомна гра на базі WarCraft 3). У 2011 році на чемпіонаті The International по Dota 2 українська команда з кіберспорту Natus Vincere (NaVi) виграла 1 млн доларів [5]. Це привернуло увагу тисяч гравців та сприяло зростанню

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

популярності кіберспорту.

Загалом, саме гра Dota 2 стала серйозним джерелом заробітків для значної кількості українських кіберспортсменів: саме за рахунок неї досягли успіху 18 із топ-25 найбагатших українських кіберспортсменів. Друга дисципліна, за кількістю зароблених призових коштів, це Counter-Strike: Global Offensive, ну а третя – Hearthstone.

Також в Україні є Федерація кіберспорту, яка є досить активною: в останніх числах 2019 року, під її керівництвом відбувся 1-й офіційний фінал Кубка України із мобільного кіберспорту з PUBG Mobile, де призовий фонд склав 200 000 грн. Також у грудні було проведено гранд-фінал 1-го чемпіонату країни із кіберспорту, у якому призовий фонд складав 1 млн грн.

Наші співвітчизники розвивають цей вид спорту не лише на теренах держави, а й за її межами: керуючі партнери технологічного холдингу Esportainment і TECHIA – компанія WePlay!Esports Олег Крот та Юрій Лазебніков створюють кіберспортивну арену у Лос-Анджелесі.

Загалом, представники цієї індустрії вважають, що на сьогоднішній день кіберспорт в Україні має всі шанси для розвитку, хоча кіберспорт у форматі бізнес досі формується, привертаючи постійну увагу спонсорів.

Враховуючи все сказане, можемо зробити висновок щодо актуальності та доцільності створення довідково-інформаційної системи для комп'ютерного клубу, на меті якої є забезпечення ефективною та злагодженою взаємодією між власниками клубу, адміністраторами та їх клієнтами – відвідувачами.

Мета роботи – розробка довідково-інформаційної системи комп'ютерного клубу. Можна окреслити наступні завдання для досягнення мети:

- дослідити предметну галузь;
- здійснити аналіз існуючих програмних рішень у даній області;
- розробити технічне завдання;
- спроектувати, реалізувати та протестувати систему;
- розробити інструкцію для користувачів.

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз змісту і опис предметної області, її функціональних та структурних особливостей

Для дослідження користувачів комп'ютерних клубів, їх потреб та отримання інформації про те, як надихати людей приходити і, найголовніше, повертатися у ваш заклад, в епоху розповсюдженого інтернету, потрібно знати своїх відвідувачів. Тільки таким чином можна організувати все так, щоб протягом тривалого часу ці люди грали у вас і приносили економічне зростання вашому комп'ютерному клубу.

Що ж потрібно знати про середньостатистичного відвідувача? Це чоловік 20-30 років. Хоча дівчата-геймери теж існують, всупереч стереотипам, але вони частіше уникають комп'ютерних клубів. Не зважаючи на думки про те, що найбільш активними гравцями є хлопці шкільного віку, проте приблизно 75% гравців – це молодь від 20 років. Наприклад, студент, який підробляє у офісі чи на роботі, яка не вимагає певної кваліфікації, хоча при цьому він здобуває вищу освіту і в майбутньому, швидше за все, буде програмістом, офісним робітником чи опанує інший творчий фах.

Хоча денна активність відвідувачів у клубах вища, проте найбільше гравців відвідують комп'ютерні клуби саме ввечері або вночі. Багато власників клубів погоджуються з тим, що часто вночі може не вистачати вільних комп'ютерів, а тому чудовим вирішенням може стати пропонування своїм клієнтам функції попереднього бронювання ПК.

Проте, є й інший тип клієнта. Це може бути людина, що відвідує клуб для роботи, термінової перевірки або відправки пошти тощо. В цей час, такий клієнт може погано розбиратися у комп'ютерах (наприклад, люди літнього віку, яким хочеться зв'язатися зі своїми рідними по Skype). Відмовлятися від таких клієнтів в жодному разі не варто: по-перше, вам потрібна хороша репутація

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

комп'ютерного клубу, яка може створюватися по, так званому, «сарафанному радіо», а по друге, якщо такі клієнти побачать, що їм прагнуть допомогти, то вони будуть із радістю повертатися. А через те, що адміністратори клубів часто лінуються допомагати таким людям, то досить часто може спостерігатися зменшення потоку відвідувачів.

Якщо розглядати відсоткове співвідношення відвідувачів, то 60% клієнтів приходять для того, щоб пограти в різноманітні ігри, а 40% просто потрібен доступ до інтернету чи до певної програми. Враховуючи цю специфіку (адже гравці схильні бути доволі гучними), варто розділити зал комп'ютерного клубу на декілька зон. До того ж, відвідувачі, які прийшли попрацювати, не потребують таких потужних комп'ютерів, як геймери.

Отже, ігрова індустрія – галузь комп'ютерних технологій і, разом з тим, це сектор розваг [6, 7].

У 2020 році аналітична компанія NielsenIQ провела перше в Україні дослідження гравців у відео/комп'ютерні ігри на замовлення Wargaming. Було встановлено, що світовий ринок відеоігор зріс на 9,3% – до \$159,3 млрд. [8].

На рисунку 1.1 представлений профіль гравців у відеоігри, а на рисунку 1.2 – навіщо українці грають у комп'ютерні ігри.



Рисунок 1.1 – Профіль гравців у відеоігри

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

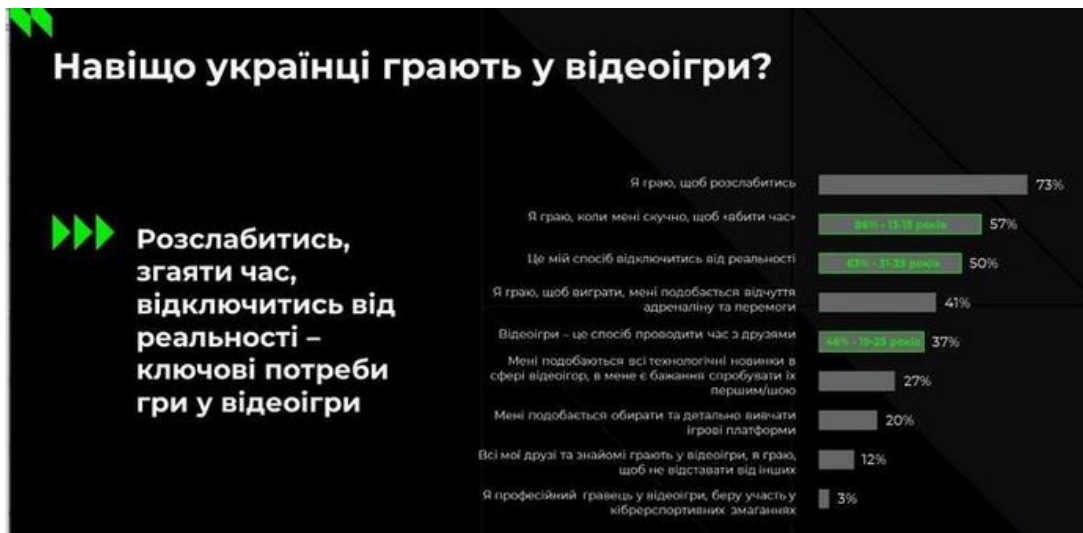


Рисунок 12 – Навіщо українці грають у відеоігри

Свій прогноз щодо відеоігрової індустрії до 2024 року опублікувала і аналітична компанія Newzoo (рисунок 1.3) [9].

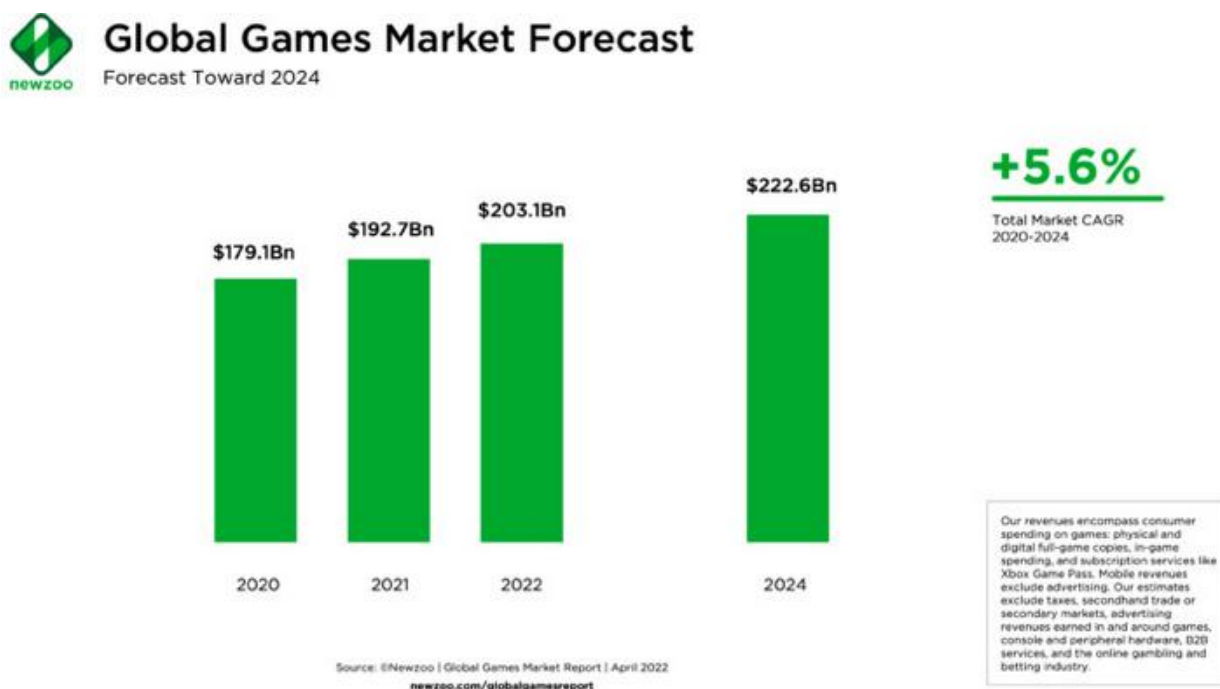


Рисунок 1.3 – Прогноз Newzoo до 2024 року

Необхідно врахувати, що головний дохід для клубу приносять саме постійні клієнти, які, до того ж, досить часто приводять з собою друзів. А тому, завданням комп'ютерного клубу, окрім інформування, є забезпечення лояльності клієнтів. Домогтися цього простіше, купуючи її. Необхідно створити програму

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

лояльності у своєму клубі, яка буде, наприклад, надавати клієнтам накопичувальні знижки, що залежатимуть від кількості його візитів або проведених у клубі годин.

Ще одним корисним моментом, що сприятиме розширенню клієнтської бази є наявність певних унікальних послуг. Тренувальна база для кіберспорту, школа комп'ютерної грамотності дітей та дорослих або ж звичайні послуги із друку – це все допоможе збільшити прибуток комп'ютерного клубу.

Як ще можна збільшити прибуток клубу? Усі власники бізнесу, що пов'язаний із кіберспортом (комп'ютерний клуб, інтернет-кафе чи кібер-арена), хоче зробити його прибутковим та успішним. Для цього власники зазвичай нарощують об'єм прибутку або скорочують витрати, але враховуючи той факт, що відвідувачі клубу приходять, щоб пограти на потужних комп'ютерах, то про економію на якості обладнання не варто й думати. Саме через це власники шукають саме додаткові можливості для збільшення свого прибутку. Це може здійснюватися, наприклад, за рахунок збільшення загальної кількості відвідувачів, але через те, що клуб є офлайн бізнесом, прив'язаним до певного місця, то це робить кількість його відвідувачів доволі обмеженою, і не приводить до очікуваного результату.

Найкращим рішенням щодо збільшення продажів буде розвиток реальної клієнтської бази та збільшення вартості їх середнього чеку. Оскільки погодинний тариф, а також абонементи складають його більшу частину, то підвищення їх вартості може спричинити зниження лояльності клієнтів клубу. А тому кращим способом може стати покращення досвіду відвідувачів, розширюючи свої послуги, наприклад, продаючи додаткові товари – напої та їжу, одяг з клубними фірмовими написами та іншу атрибутику, надаючи послуги з корекції текстів, друку та іншого, надаючи їм можливість витратити більше коштів під час відвідування. Для власника бізнесу це означає необхідність введення фінансової звітності, облік товарів. А для цього необхідним є створення довідково-інформаційної системи.

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Для того, щоб проаналізувати програмно-технічне забезпечення предметної області, розглянуто кілька існуючих довідково-інформаційних систем.

Існує досить широкий спектр програм, які використовуються у комп'ютерних клубах:

- Internet Cafe Software 10.1.0;
- SENET;
- Locker 5.86;
- Network Administrator 6.7.7;
- Smart Club Keeper 2.50;
- GameAdmin 1.0.0.9;
- APTimer 1.2 та ін.

Програму SENET розробила українська софтверна компанія ENESTECH [10]. Одне з вікон SENET показано на рисунку 1.4.



Рисунок 1.4 – SENET: вікно керування комп'ютерами [10]

Аналізуючи цю довідково-інформаційну систему, можна зробити висновок, що їхня сторінка не дуже інформативна: з нею не стає зрозумілою основна суть системи. Інформація тут розташовується у місцях із складним доступом та на нашу думку є недостатньою. Наприклад, якщо зайти до вкладки «Дізнатися вартість», то для отримання інформації потрібно спочатку зареєструватись, визначити точний час сеансу роботи, а лише тоді можна дізнатися вартість послуги. Це займає багато часу і не є зручним, а для людей літнього віку із їхнім рівнем навичок, яких ми теж прагнемо залучати в наш комп'ютерний клуб, це може стати значною проблемою. Тому, на нашу думку, довідково-інформаційна система клубу має бути максимально спрощена та доступна, чого ми не бачимо саме в цій довідково-інформаційній системі.

Наступною є довідково-інформаційна система клубу «It land» [13], що зображується на рисунку 1.7. Вона – більш яскрава, проте, як і попередня, не є інформативною. Одразу видно, що ця система розрахована на молодого користувача та не враховує людей літнього, чи навіть середнього, віку, що також все більше користуються ПК для пошуку необхідної інформації.

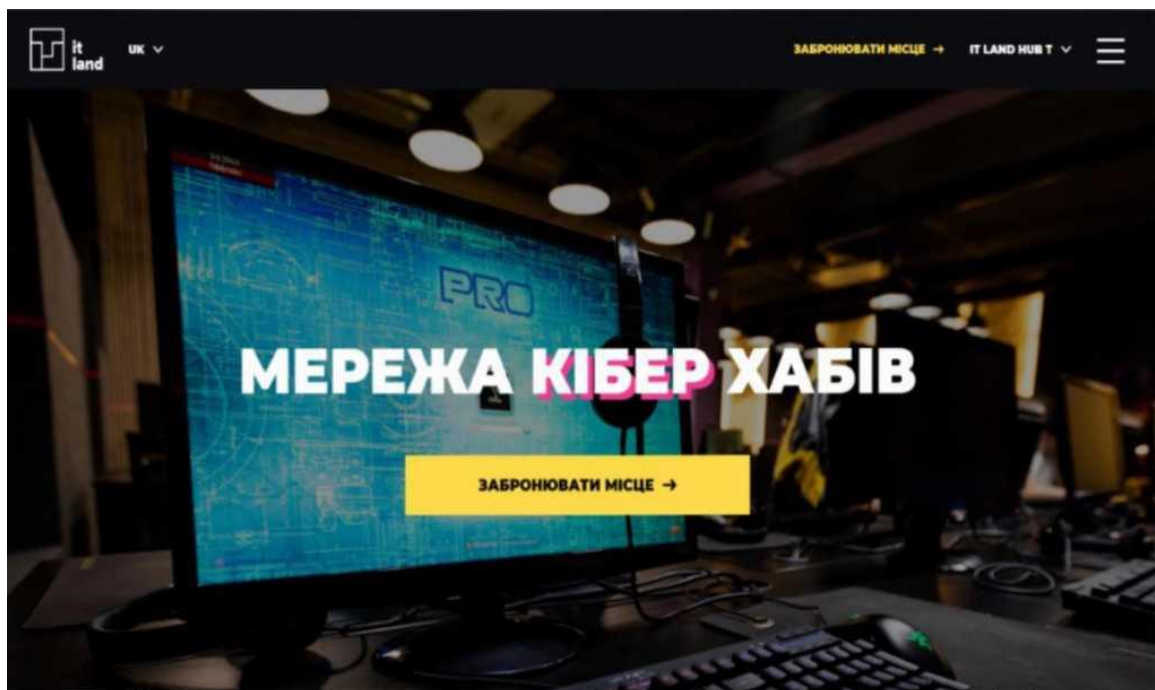


Рисунок 1.7 – Мережа кіберхабів «It land» [13]

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Гарним прикладом цьому є вкладка «Забронювати місце», яка зображена на рисунку 1.8. Дивлячись на неї, стає зрозуміло, що для бронювання місця потрібно бути постійним клієнтом, щоб орієнтуватися в ній, а всю інформацію про роботу може бачити лише адміністратор клубу.



Рисунок 1.8 – Вкладка бронювання місця [13]

1.3 Визначення вимог до довідково-інформаційної системи

Які ж критерії мають бути в нашій довідково-інформаційній системі комп'ютерного клубу? Найголовніше – вона має бути розбита на інтерфейси: для адміністратора клубу та його відвідувачів.

Під час входження в систему, клієнт має зареєструватися і створити власний обліковий запис. Також тут вони мають мати змогу поповнювати баланс свого рахунку, перевіряти новини клубу, обирати різні тарифи та абонементи, мати можливість запускати та переглядати різноманітні ігри, а також купувати певні товари в «Магазині».

Тоді ж адміністратори можуть керувати усіма ПК та їх бронюваннями, оновлювати доступні ігри, обробляти замовлення відвідувачів, розміщувати

					<i>КвРІПЗ.190143.01.21.ПЗ</i>	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

новини клубу, відстежувати доступні комп'ютери, а також створювати звіти. Так можна не лише організувати процес продажів у модулі «Магазин» в інтерфейсі відвідувача, але й управляти клубом. Варто також врахувати, що комп'ютерні клуби повинні дотримуватися за необхідності встановлених епідеміологічних правил та сприяти зручному розміщенню відвідувачів, а програмне забезпечення має бути досить гнучким, щоб бізнес міг адаптуватися до нових норм.

Які задачі має вирішувати наша довідково-інформаційна система?

Окрім стандартних функцій з управління клубом, наша система має призначатися для задоволення потреб продажів. Весь робочий процес має бути автоматизованим: від інформування клієнта про доступні товари і до фінансових звітів. Система має вмщати у себе онлайн-каталог основних і додаткових товарів та послуг. Ця функція також сприятиме зручності для відвідувачів, яким не потрібно буде навіть вставати з-за комп'ютеру, щоб зробити замовлення.

Керування асортиментом та залишками

Як тільки товар почне продаватися через модуль «Магазин», потрібно буде рахувати залишки. Без цієї системи управління, з товарами, що проходять через клуб, може бути плутанина. Тому тут потрібен модуль, що дозволить здійснювати централізований облік кожного з товарів, забезпечуючи при цьому джерело достовірної інформації про продавця, ціни покупки і продажу, специфіку та кількість наявних одиниць товару. Ефективне управління складом виключить можливість замовлень відсутнього товару, своєчасне поповнення запасів, а отже, усуне необхідність для клієнтів виходити за покупками за межі клубу.

Оформлення замовлення

За допомогою нашого онлайн-каталогу відвідувачі клубу матимуть можливість оформити онлайн замовлення та надіслати його прямо до адміністратора. Це позбавить адміністратора можливої паперової тяганини.

Відстеження статусу замовлень та отримання повідомлень

Досить важко весь час відповідати на численні запити відвідувачів стосовно їх замовлень – це відволікає адміністратора від його основної роботи.

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

Тому за допомогою «Магазину» в своєму інтерфейсі відвідувачі матимуть можливість відстежувати стан їхнього замовлення, сидячи за ПК, а адміністратор – обробити це замовлення значно швидше, оскільки він отримує миттєві повідомлення та може, за потреби, зв'язатись з клієнтом онлайн.

Інтегрована каса

Під час продажів додаткових товарів, обов'язковим інструментом є створення каси. З спеціальним інтегрованим касовим модулем стає можливим створення квитанцій в один клік та оброблення операцій за безготівковим чи готівковим розрахунком.

Історія замовлень

Підготовка звітів та управління фінансами є більш болючим питанням, коли все проводиться у паперовому вигляді, а тому цифрове вирішення значно економить час. Тому наша інформаційно-довідкова система має дозволяти співробітникам і клієнтам бачити історію їх покупок.

Виходячи з цього, можна підвести підсумки, що існує багато способів збільшити дохід комп'ютерного клубу, проте тільки деякі з них не зменшують лояльність відвідувачів клубу.

Найкращим способом збільшення величини прибутку, судячи з наших досліджень, може бути створення довідково-інформаційної системи. Вона дозволить збільшити середній чек клієнта клубу за рахунок розширення асортименту послуг та товарів. Більше того, її введення поліпшить і якість обслуговування відвідувачів.

Технічне завдання на розробку подано у додатку А.

1.4 Постановка задачі

Отже, довідково-інформаційна система комп'ютерного клубу має виконувати наступні функції:

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

- надання користувачу можливості створити акаунт, авторизуватись, відновити пароль, керувати власними даними;
- забезпечення охорони даних користувачів від розповсюдження та крадіжки платіжних даних клієнтів;
- надання клієнту можливості оформлювати замовлення та оплачувати замовлення, використовуючи кредитну карту чи інший метод оплати;
- облік клієнтів, відвідувань та ідентифікацію карт;
- контроль за оплатою послуг через клубну карту;
- можливість обліку послуг, що надаються за кількістю відвідувань або на певний період;
- управління тарифами на різні послуги клубу;
- фінансовий аналіз;
- планування графіка роботи співробітників;
- формування звітів про діяльність клубу.

Надійне рішення по управлінню комп'ютерним клубом має включати:

- інтерфейс для адміністратора та відвідувачів;
- можливість створення замовлень;
- карту управління комп'ютером;
- модуль резервування комп'ютеру;
- касову систему;
- систему повідомлень;
- модуль «Магазин»;
- статуси відстеження замовлень;
- цінові пропозиції та програму лояльності;
- онлайн-каталог товарів;
- модуль управління складом;
- можливість перегляду історії замовлень.

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ ДОВІДКОВО-ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Проектування архітектури довідково-інформаційної системи

До початку розробки будь-якої програми варто детально розписати алгоритми її роботи, структуру бази даних, структуру її компонентів тощо. Для цього існують UML-діаграми. UML (англ. Unified Modeling Language) – уніфікована мова моделювання, яка застосовується у прототипі об'єктно-орієнтованого програмування.

Ця мова є однією з частин уніфікованого процесу розробки програмного забезпечення (ПЗ). Вона є мовою широкого профілю, своєрідним відкритим стандартом, який використовує графічні позначки в процесі створення абстрактної моделі системи, що має назву UML-модель. UML була винайдена для візуалізації, визначення, документування та проектування найчастіше саме програмних систем.

Ця мова моделювання може використовуватися на усіх етапах життєвого циклу розробки прикладних програм і аналізу різноманітних бізнес-систем.

UML-модель створювалася для покращення організації перебігу розробки програмних систем, задля покращення ефективності їх реалізації в кілька разів та значно покращити якість безпосередньо кінцевого продукту.

В UML-проектуванні є такі види діаграм:

- діаграми розгортання;
- діаграми комунікації;
- діаграми класів;
- діаграми композитної (складеної) структури;
- діаграми прецедентів;
- діаграми діяльності;
- діаграми послідовностей;
- діаграми компонентів;
- діаграми об'єктів;

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

- діаграми станів;
- діаграми кооперації;
- діаграми пакетів;
- діаграми огляду взаємодії;
- діаграми синхронізації.

Не дивлячись на те, що UML є стандартом мови моделювання, який всебічно визнаний, проте може часто піддаватися критиці через наступне:

- складнощі при застосуванні та вивченні;
- неточність семантики;
- візуальну неоднорідність;
- надмірність мови.

Діаграма прецедентів (варіантів використання, ВВ) в UML – це діаграма, що показує співвідношення між прецедентами і акторами в певній системі. По суті, це граф, до якого входять множини прецедентів і акторів, що є обмеженими межею системи (прямокутник), асоціацій між прецедентами і акторами, а також різноманітних залежностей узагальнення між акторами і залежностей між прецедентами. Діаграми прецедентів відображають відповідні елементи моделі варіантів використання.

Головний зміст цієї діаграми у наступному: спроектована система відображається у вигляді багатьох акторів чи сутностей, які працюють за допомогою варіантів використання з системою. Варіант використання використовують для опису послуг, що система може пропонувати актору. Тобто, кожен варіант використання надає певний набір дій, що опрацьовуються системою під час діалогу із актором. Проте, не визначається саме те, як буде проводитися спільна робота із системою самих акторів.

Діаграма компонентів в UML – це діаграма, у якій видно компоненти, їх впливи і зв'язки. Така діаграма висвітлює залежність між компонентами ПЗ і вміщує бінарні складові, складові вихідних кодів та складові, які можуть виконуватися.

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Модуль ПЗ може бути зображеним як компонент. Деякі компоненти можуть зустрічатися під час компіляції, а інші – під час компонування чи в процесі роботи самої програми. На діаграмі компонентів можна побачити тільки структурні характеристики, а тому для відображення екземплярів компонентів окремо слід застосовувати діаграму розгортання.

Діаграма компонентів може використовуватися для:

- специфікації варіанту програмної системи, що використовується;
- висвітлення початкового коду програмної системи, а саме його загальної структури;
- подачі фізичної та концептуальної схем баз даних;
- забезпечення, за необхідності, використання окремих елементів програмного коду кілька разів.

При складанні діаграм компонентів беруть участь і архітектори, і системні аналітики, і програмісти. Діаграма компонентів створює злагоджений перехід від подання до потрібної реалізації проекту у вигляді програмних кодів. Одні елементи можуть створюватися лише на етапі компіляції програмних кодів, інші ж – в процесі виконання.

Діаграма компонентів висвітлює загальну залежність між компонентами, а також розглядає їх як класифікаторів.

Діаграма класів являється статичним уявленням структури моделі, що представляє статичні елементи, наприклад, типи даних, класи, їх відношення та зміст. Така діаграма може вміщувати позначення для пакетів та вкладень цих пакетів, а також позначення певних елементів поведінки, проте їх розвиток може розкриватися в інших видах діаграм.

Діаграма класів зазвичай застосовується для візуалізації статичної структури моделі системи, із позначенням класів об'єктно-орієнтованого програмування. На цій діаграмі можна відзначати інтерфейси, класи, об'єкти і кооперації, а також співвідношення між ними.

На рисунку 2.1 представлена узагальнена діаграма варіантів використання системи.

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

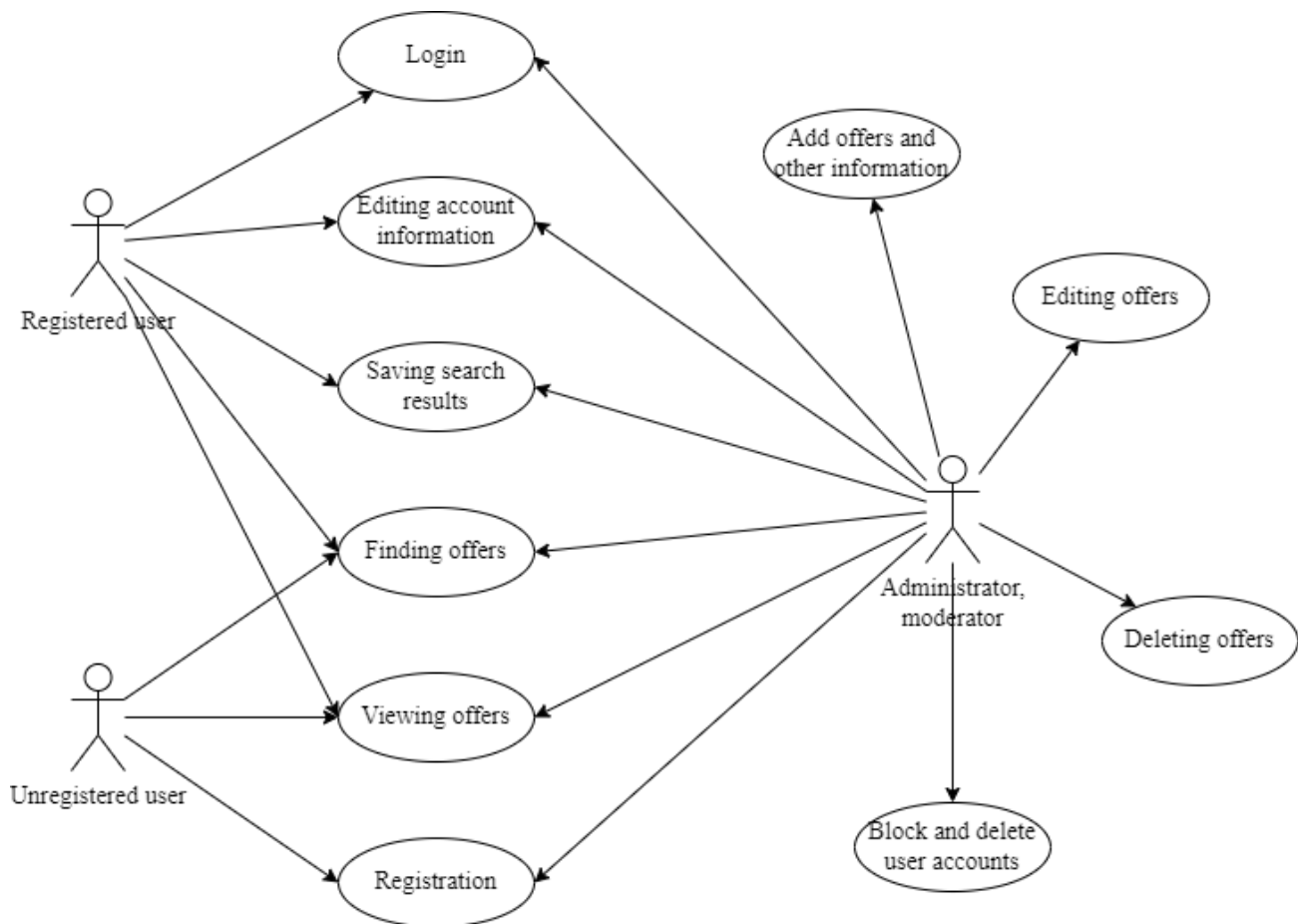


Рисунок 2.1 – Діаграма варіантів використання

Для ефективної роботи довідково-інформаційної системи комп'ютерного клубу важливим є забезпечення для користувачів можливості реєструватися та авторизуватися. Так, на рисунку 2.2 схематично відображено вигляд реєстрації користувача за допомогою UML-діаграми послідовності, а на рисунку 2.3 – діаграма послідовності для ВВ «Авторизація».

Детальний опис ВВ дає змогу програмісту отримати повну інформацію про функціонування кожного з модулів відповідної програми. Це звільняє його від вигадування непередбачених модулів у документації та дозволяє повністю приділити увагу щодо безпосередньої програмної реалізації продукту.

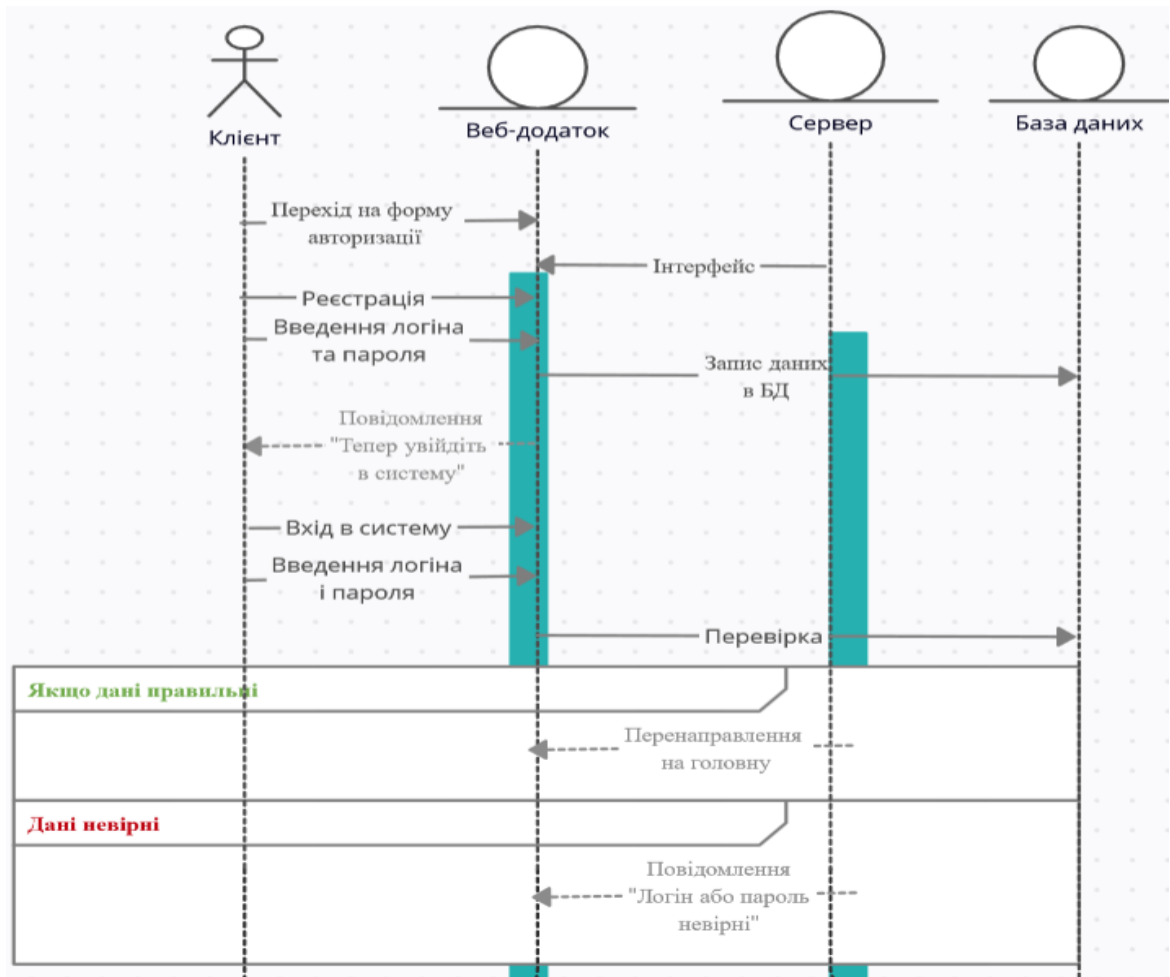


Рисунок 2.2 – Діаграма послідовності реєстрації клієнта клубу

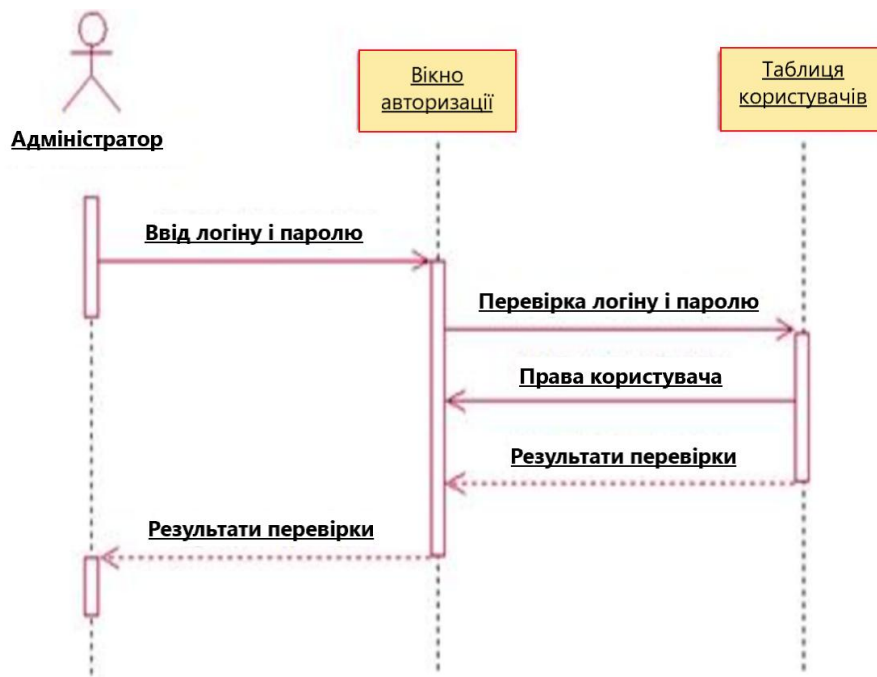


Рисунок 2.3 – Діаграма послідовності для ВВ «Авторизація»

2.2 Проектування структури бази даних

MySQL є досить швидкою та, водночас, стабільною системою. Це є основною причиною її широкого вжитку. В грудні 2002 року, видавництво «eWeek» провело тестування найбільш розповсюджених систем з управління базами даних, таких як MySQL, Microsoft SQL Server, Oracle і DB2, та опублікувало його результати. В процесі дослідження, було виявлено, що MySQL і Oracle 9 посіли перше місце саме за їх продуктивністю.

Система MySQL є доступною в двох версіях: комерційне та безкоштовне програмного забезпечення, що стає можливим завдяки використанню виробником «подвійної» ліцензійної схеми: ПЗ MySQL можна отримати безкоштовно із відкритою ліцензією, проте, якщо потрібна бізнес-версія, то її можна купити.

MySQL може підтримувати більшість функцій, які є важливими для спільноти розробників і користувачів баз даних: зовнішні ключі, транзакції, підзапити, блокування на рівні стрічок та повнотекстовий пошук.

Система MySQL є надійною і багаторазово перевіреною. Нею користується багато компаній, навіть відомих: Yahoo!, Slashdot, Finance. MySQL – це чудовий засіб вивчення баз даних, оскільки вона проста у використанні й інсталяції, має незначні вимоги стосовно дискового простору та оперативної пам'яті.

Наша база даних, що була створена для інформаційно-довідкової системи комп'ютерного клубу, зображається на рисунку 2.4.

Надаємо опис таблиць, які входять в цю базу даних, з метою побудови нашої інформаційно-довідкової системи для комп'ютерного клубу.

Записи

\$wpdb->posts – таблиця, у якій зберігаються постійні сторінки, вкладення, пости тощо.

\$wpdb->postmeta – уточнює таблицю \$wpdb->posts (зберігає додаткові дані для записів).

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

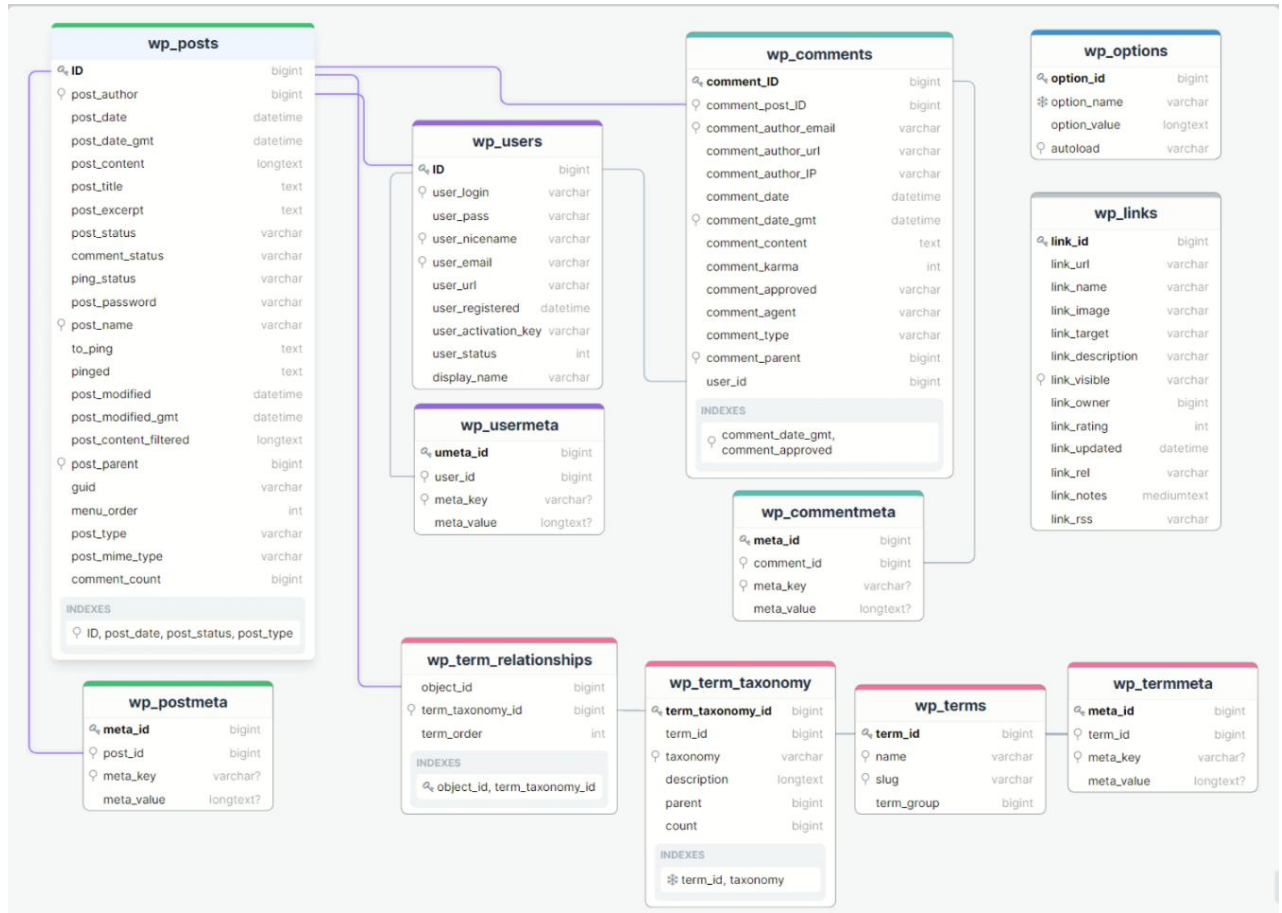


Рисунок 2.4 – Модель бази даних системи

Користувачі

\$wpdb->users - таблиці з усіма даними про користувачів, які є зареєстрованими.

\$wpdb->usermeta - додаткова інформація про користувачів, наприклад, ім'я, нік, його права та інше для таблиці \$wpdb->users.

Коментарі

\$wpdb->comments - таблиця з записами усіх коментарів.

\$wpdb->commentmeta - метаполя для таблиці \$wpdb->comments.

Таксономії

\$wpdb->terms – таблиця, де вказана інформація про назви міток, категорій, категорій посилань тощо.

\$wpdb->termmeta – таблиця із додатковими полями для таблиці \$wpdb->terms.

\$wpdb->term_taxonomy - таблиця з інформацією про різні таксономії, а також їх опис.

Змн.	Арк.	№ докум.	Підпис	Дата

\$wpdb->term_relationships – таблиця, що пов’язує таксономії з контентом.

Інші таблиці

\$wpdb->links – таблиця, де зберігаються записи усіх посилань.

\$wpdb->options – таблиця налаштувань.

2.3 Аналіз та вибір технологій і засобів реалізації системи

Далі надаємо характеристику тим засобам ПЗ, якими будемо користуватись для створення довідково-інформаційної системи комп’ютерного клубу.

До найбільш популярних засобів відносяться PHP та MySQL. Розглянемо їх детальніше та проведемо аналіз, порівнявши з іншими поширеними засобами реалізації проекту, які відомі на даний момент.

Active Server Pages (ASP) є найбільш вживаною «мовою» сценаріїв Microsoft. Взагалі, ASP – це більше не мова, а просто розширення у Visual Basic, яке використовується для створення сценаріїв. Тому кожному, хто знайомий з Visual Basic, стане відносно просто освоїти і ASP. Але тут є і недоліки:

- ASP як правило працює повільніше, ніж PHP;
- основу ASP складає архітектура, яка створена на COM, а тому під час звернення програмою до бази даних чи здійснення виведення певних даних, ASP робить це за допомогою сом-об’єктів інших сервісів NT чи рівнів ОС (такі навантаження накопичуються і можуть призвести до того, що кожного разу, окрім як видачі простих сторінок, в той час, коли трафік матиме середню потужність, його продуктивність буде не дуже високою).

Іншим недоліком є те, що ASP не завжди підходить для перенесення на інші платформи та інтеграції засобами GNU, та з серверами і середовищами open source. Враховуючи те, що ASP - це фірмова система Microsoft, вона зазвичай використовується з її IIS (Internet Information Server), а тому ASP частіше обирають обмежено, а саме для 32-розрядних систем Windows, оскільки для

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

дуже великої кількості серверів ця технологія є безкоштовною. Є також версії ASP і для UNIX – наприклад, ChilliSoft ASP, і для ряду інтерпретаторів ASP для різноманітних систем та веб-серверів, проте для них ціна цієї системи може виявитися досить високою, особливо, враховуючи її продуктивність. Хоча технологія ASP.NET відрізняється досить сильно.

На перспективу, ASP може значно покращити можливість масштабування і свою продуктивність, проте реальних переваг можна буде досягти лише за умови значних витрат на різноманітні додаткові сервери.

PHP може працювати практично на усіх платформах, хоча її версії Cold Fusion підходять тільки для Solaris, Win32, HP/UX та Linux. Для використання PHP потрібні базові навички програмування, на відміну від Cold Fusion з поліпшеним інтегрованим середовищем розробки (IDE) та мовними конструкціями. PHP не потребує вимог до ресурсів.

Оскільки PHP розроблений для Інтернету, то він є кращим в цій області, на відміну від Perl. Останній створювався для різних застосувань, що відобразилося також і на його зовнішньому вигляді. Синтаксис і форма у Perl значно ускладнюють читання сценаріїв та їх модифікації, а це є потрібним. Не дивлячись на те, що Perl використовується тривалий час і є широко підтримуваним, проте він перетворився на доволі складну конструкцію із розширеннями і доповненнями, які часто є недоцільними.

З цієї точки зору, PHP є кращим для сприйняття. До того ж, при збереженні гнучкості, PHP простіше поєднується із вже існуючим HTML і має функції, що схожі із Perl, але із ще більшою елегантністю.

PHP простіший у використанні, ніж Java. Також з його допомогою можна доволі легко будувати веб-сервери-застосування, що володіють подібними перевагами масштабованості та гнучкості. При застосуванні PHP не потрібно володіти значним досвідом розробки ПЗ для створення найпростіших динамічних сторінок – достатньо просто бути спостережливим, навіть, якщо ви маєте незначний досвід у програмуванні. До того ж, Java може обійтися у більшу

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

вартість, адже у багатьох компаніях, зазвичай, можуть встановлювати для Java Enterprise іншу машину та використовувати Oracle чи інше дороге ПЗ.

Проте, PHP також потребує подальшого розвитку, адже воно не є тотожним та не володіє деякими іншими функціями, наприклад, пулом об'єктів чи відображенням баз даних, які присутні в Java.

Не дивлячись на це, мова PHP є кращим варіантом, у нашому випадку, для нашої задачі – розробки програмного коду з взаємодією із базою даних сайту.

Content Management System (CMS) – система керування контентом – це ПЗ, що використовується під час забезпечення процесу та організації створення, редагування і управління текстом, графічних або мультимедійних елементів сайту у всесвітній мережі Інтернет чи у локальних комп'ютерних мережах.

Є багато різних систем по управлінню контентом, що створюються за допомогою різноманітних технологій. Серед них є і безкоштовні, і платні програми.

Основний елемент будь-якої CMS – це сховище інформації. У більшості сучасних CMS-сховищах є реляційна база даних, яка складається із таблиць з визначеними співвідношеннями. Коли CMS потребує зберегти інформацію – просто заносить її до бази даних, де для усіх сутностей створена окрема таблиця.

Інформацію для відтворення CMS позичає з бази даних. Тут використовуються шаблони – тобто файл, що містить в собі дизайн сторінки, створений технологіями спеціальної відповідної мови. Саме з їх допомогою можна побачити певну інформацію у HTML-форматі. Наприклад, HTML-код розмічений певним чином, в середині якого міститься відповідна інформація, що показує місцезнаходження певних елементів, які можна брати з бази даних.

Інша частина CMS – це певна система користувачів і їх ролей, тобто наборів відповідних дій, які користувач може здійснювати в конкретний час.

В нових CMS часто можна визначати декілька ролей користувача: адміністратор, відвідувач, користувач. Конкретному користувачу можна

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

присвоїти певну роль, а процес надання ролей може виконуватися або адміністратором, або ж автоматично.

Те, що CMS так широко використовується і впроваджується на сьогоднішній день, сприяє багато факторів. Один із головних – ускладнення функцій сайтів, які створюються. Розробка кожного разу однієї й тієї ж програмної частини (модулю) не є раціональним, саме тому розпочалось програмування спеціальних бібліотек із корисними функціями, пізніше з'явилися рішення, з допомогою яких і розпочалась розробка універсальних CMS.

Інший дуже важливий аргумент – процес спрощення самих CMS. Достатньо значна кількість хостингів можуть пропонувати можливість завантаження готових платних або ж безкоштовних версій CMS.

Схема роботи CMS наступна: користувач увійшов на сайт, що є під керівництвом CMS, і виконав там якусь дію. Першочергово, CMS має зрозуміти, яка саме реакція має бути на цю саму дію. Потім модуль, який відповідає за обробку запиту користувача, фіксує цю інформацію, яка надійшла до бази даних. Далі система має провести запит до бази даних та отримати інформацію, яку необхідно відобразити, або формує відповідне повідомлення. Інформація з бази даних надсилається до відповідного модулю, що бере цю тему та додає до неї усю потрібну інформацію. У результаті виконання таких дій, утворюється HTML-код, що надсилається назад до користувача, а відповідний результат проведеної роботи користувач може побачити в себе на екрані.

Розглянемо переваги CMS:

- використання CMS економить час на розробку сайтів;
- у більшості CMS є технічна підтримка, переважно це інтернет-спільноти користувачів, які можуть не лише давати поради один одному, а й брати участь у вдосконаленні та розробці CMS, а це гарантує постійне вдосконалення таких CMS;
- зміст CMS відокремлений від дизайну сайту, а це суттєво спрощує процес редагування змісту самого сайту;

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

– використання певних шаблонів дозволяє автоматично, незалежно від наповнення сайту, змінюватимого зовнішній вигляд;

– більшість CMS має модульну архітектуру, отже її функції можна збільшувати, використовуючи плагіни та модулі.

Є платні та безкоштовні системи управління контентом. Зазвичай, більш популярними є, звичайно, безкоштовні. Найбільш використовуваними є Drupal, Joomla та WordPress. Охарактеризуємо їх.

CMS Drupal – один з кращих варіантів, що використовується при створенні інтернет спільнот, блогів, форумів, онлайн енциклопедій тощо. У нього є всі можливості, що потрібні для успішного управління змістом певного сайту. Крім того, тут є багато модулів та шаблонів.

Проте, є й недоліки. Так CMS Drupal має меншу кількість документації, ніж, наприклад, Joomla чи WordPress, а також менш дружній інтерфейс і низьку швидкість завантаження сторінок.

CMS Joomla – це ще одна досить популярна у світі CMS, яка, однак, є більш складнішою у вивченні та використанні, аніж, наприклад, WordPress. Хоча вона має більш ширшу сферу застосування.

Для Joomla розроблена велика кількість різних модулів, що робить її практично універсальною CMS. Крім цього, у цієї CMS існує досить велика кількість шаблонів, що надають величезні можливості для створення дизайну сайту. Також Joomla сумісна із різноманітними серверами, наприклад, MacOSX server, Linux, AIX, Solaris, FreeBSD, завдяки чому нею можна користуватися, не беручи до уваги тип серверу. Недоліком цієї CMS є можливі труднощі із індексацією пошуковими системами сайтів та недостатній захист системи від вторгнень.

Ще одна CMS WordPress – теж досить часто використовується. Вона дозволяє запрограмувати різні сайти, хоча частіше вона є двигом для різноманітних блогів. Ця CMS включає багато локалізацій, а також для неї було розроблено велику кількість додаткових шаблонів і модулів. Щодо недоліків, то

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

можна відмітити можливість збоїв під час надто високої відвідуваності та низьку швидкість роботи сайтів.

Якщо ж порівнювати вживаність вищезгаданих CMS, то можна зробити висновок, що більш використовуваною, станом на липень 2021 року, є саме CMS WordPress [14] – це зображено на діаграмі на рисунку 2.5.



Рисунок 2.5 – Кількість сайтів, створених на різних CMS (на липень 2021 року)

Якщо порівнювати відмінні риси безкоштовних і платних CMS, то можемо відзначити їх універсальність, а також наявність декількох версій – від безкоштовних до більш дорогих, у яких присутні можливості створення будь-яких сайтів та, загалом, більш потужні функції. Що стосується саме платних CMS, то більш популярним є Бітрікс, який застосовується для безлічі різноманітних сайтів. Крім нього, достатньо хорошими (як платними, так і безкоштовними) CMS є Amiro, OpenCart, PrestaShop, CS-cart, Magento, UMI.CMS, NetCat, Webasyst та інші. Хоча часто розробники можуть надавати для сайтів CMS перевагу у використанні власних розробок.

2.4 Висновок до розділу 2

В даному розділі було спроектовано структуру бази даних і архітектуру системи загалом. Також обрано технології та засоби для програмної реалізації довідково-інформаційної системи: мову програмування PHP, сервер бази даних MySQL та таблиці стилів CSS.

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Реалізація бази даних

Будь-які створені дані мають десь зберігатися. В програмуванні для цього використовують різноманітні бази даних. Проаналізуємо деякі їх основні типи.

Типами баз даних, або, так званими, моделями баз даних, є шаблони і структури, що можуть бути використані для організації певних даних в системі управління базами даних (СУБД). Вибір відповідного типу залежить від того, які операції додаток зможе виконувати та як дані будуть представлені.

Прості структури даних

В таких структурах даних перший і найпростіший спосіб зберігання - текстові файли. Такий метод застосовується навіть в наш час під час роботи з невеликими об'ємами інформації. Розглянемо це на рисунку 3.1.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
bob:x:1000:1000:Bob Smith,,:/home/bob:/bin/bash
```

Рисунок 3.1 – Приклад бази даних у файлах текстового формату [18]

Недоліками простих баз даних є:

- обмежений тип і рівень складності збереженої інформації;
- труднощі у встановленні зв'язків між компонентами даних;
- відсутність функцій паралелізму.

Тому такі бази використовуються лише для систем з невеликими вимогами

									Арк.
									34
Змн.	Арк.	№ докум.	Підпис	Дата					

КвРІПЗ.190143.01.21.ПЗ

до запису та читання, і можуть призначатися для зберігання конфігураційних даних. При їх створенні не потрібне стороннє забезпечення.

Ієрархічні бази даних

На відміну від текстових, в цих базах даних вже з'являються зв'язки між певними об'єктами. Тут кожен запис має «батька», що утворює деревоподібну структуру, в якій записи класифікуються відповідно до їх стосунків із ланцюгом батьківських записів. Це відображається на рисунку 3.2 [18].

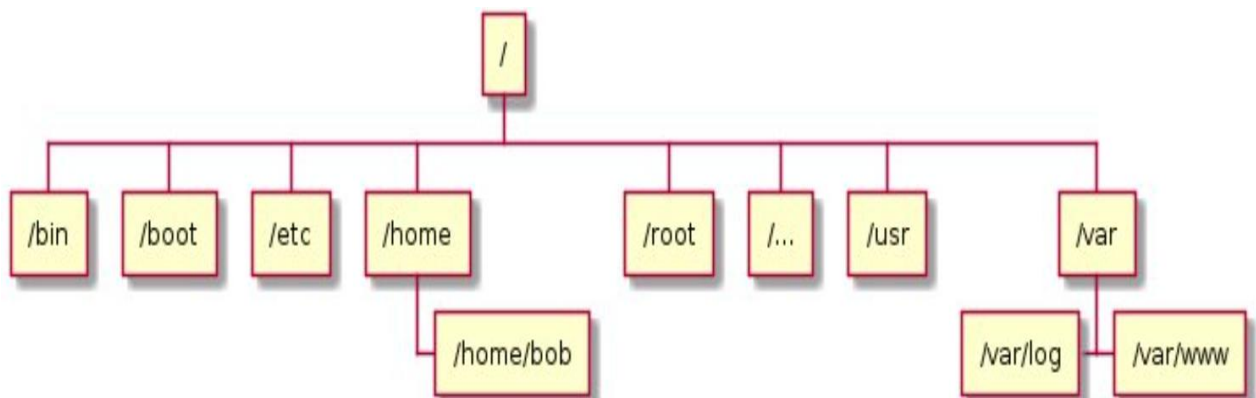


Рисунок 3.2 – Зразок побудови деревоподібних зв'язків в ієрархічній базі даних

Особливості таких баз даних:

- інформація організовується у вигляді дерева «батько-нащадок»;
- кожен запис має лише одного «батька»;
- зв'язки між записами - фізичні покажчики;
- неможливо створити відносини «багатьох до багатьох».

Мережеві бази даних

Ці бази розширюють функціональність ієрархічно. Саме тому записи можуть мати вже більше, ніж одного «батька». Це означає, що можна створювати складні відносини, як показано на рисунку 3.3 [18].

Особливість мережевих баз у тому, що дані в них подаються не деревом, а загальним графом, проте вони обмежені такими самими шаблонами доступу, що й бази ієрархічні.

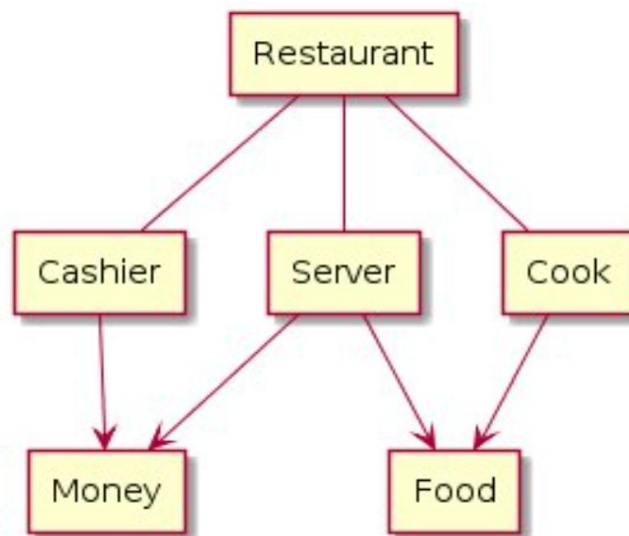


Рисунок 3.3 – Приклад мережевої бази даних та зв'язків в ній

Реляційні бази даних

Це найстаріший тип баз даних загального призначення, який й досі використовується. Дані та їх зв'язки в цих базах організуються за допомогою таблиць, де кожен стовпець у таблиці має свій тип та ім'я, а кожен рядок - це окремий запис або елемент даних із значенням для кожного стовпця. В таких базах даних поле у таблиці (зовнішній ключ) може включати в себе навіть посилання на стовпці з інших таблиць, що дозволяє їх об'єднувати.

Завдяки такій розвиненій структурі та гнучкості, реляційні бази є потужними й адаптивними до різних видів даних. Щоб отримати доступ до даних, тут використовується мова структурованих запитів SQL. Через це ця база є вибором для багатьох додатків (рисунок 3.4) [18].

Документні бази даних

Їх ще називають, документоорієнтовані бази чи сховища документів.

У таких базах спільно використовується базова семантика пошуку сховищ значень і ключів та доступу.

У документних базах може використовуватися ключ для унікального визначення даних. Сховища «ключ-значення» і ці бази даних відрізняються тим, що замість зберігання blob-об'єктів, останні зберігають дані в структурованих форматах – JSON, BSON або XML (як показано на рисунку 3.5).

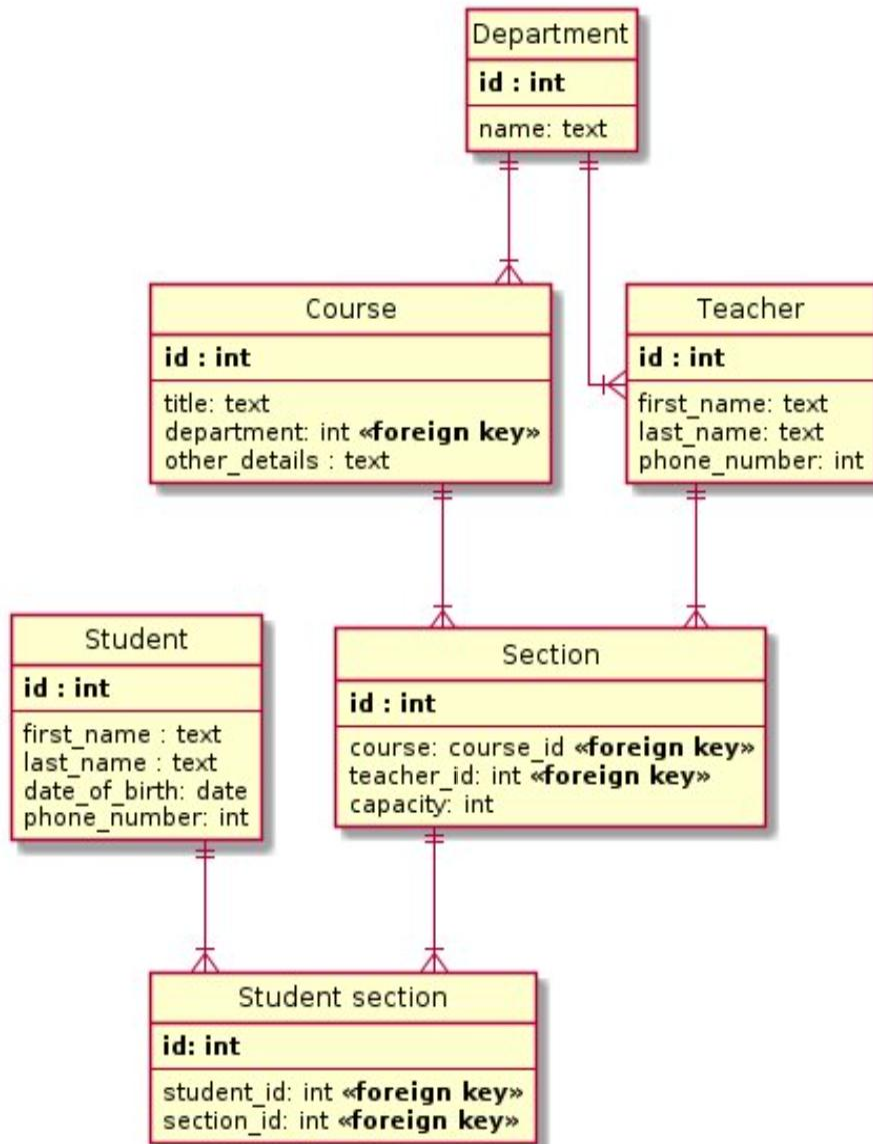


Рисунок 3.4 – Реляційна база даних – приклад

Особливості документоорієнтованих баз даних полягають у тому, що вони не виділяють окрему схему чи формат, а внутрішня структура кожного документу може відрізнятися. Такі бази даних – хороший вибір для швидкої розробки, оскільки будь-коли можна змінювати властивості даних, не змінюючи при цьому структуру чи безпосередньо дані.

Графові бази даних

В цих базах замість зіставлення зв'язків із ключами зовнішніми і таблицями, встановлюють зв'язки, використовуючи при цьому властивості, ребра та вузли.

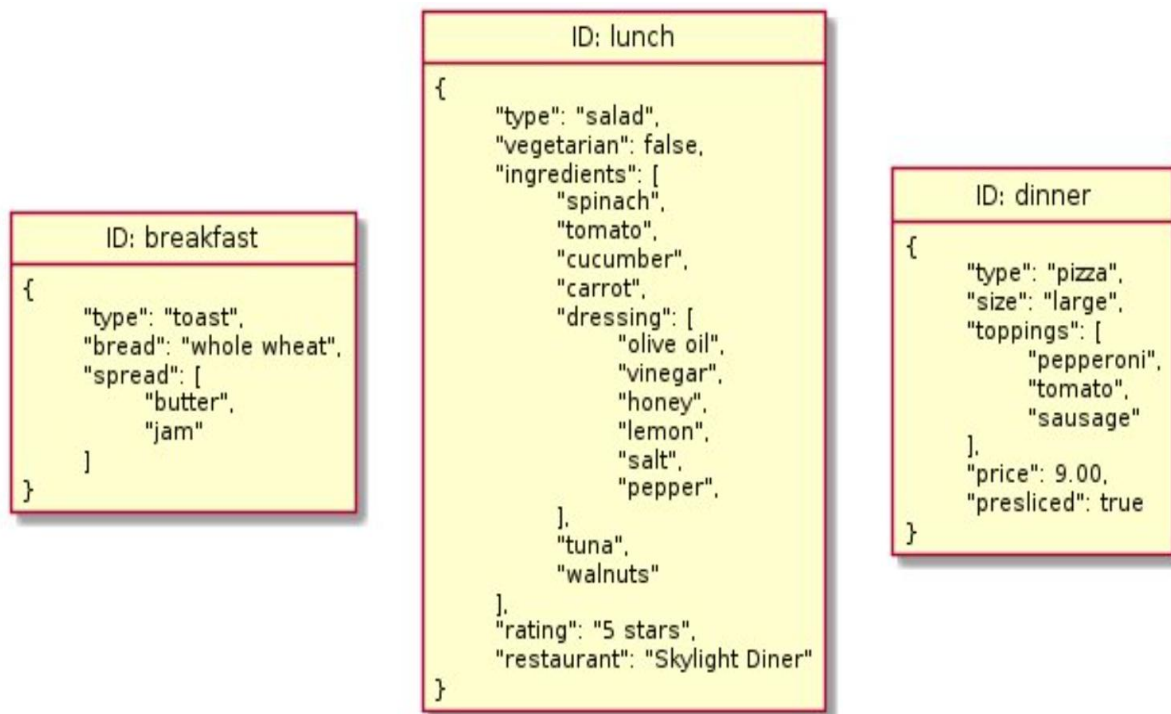


Рисунок 3.5 – Документна база даних – приклад

У графових базах даних дані представлені у вигляді вузлів, що можуть мати певну кількість пов'язаних із ними властивостей. Ці бази виглядають подібно мережевим і акцентуються на зв'язках між відповідними елементами. Графові бази чітко відображають зв'язки між різними типами даних і не потребують покрокового обходу для переміщення між певними елементами. Тут відсутні обмеження в типах зв'язків. Приклад цієї бази наведений на рисунку 3.6.

Проаналізувавши більшість можливих типів баз даних, зроблено висновок, що потрібно використати реляційну базу даних для створення довідково-інформаційної системи.

Дані довідково-інформаційної системи збережені у таблицях цієї бази даних, а її опис представлено у таблиці 3.1.

Розглянемо ці таблиці більш докладніше для аналізу їх вмісту.

Підключимося до потрібної бази: `mysql -u user -h host -p database`

Виберемо кодування для вмісту (UTF-8): `mysql> set names utf8;`

Можна переглядати таблиці за допомогою простих запитів, наприклад:

`mysql> SELECT * FROM posts LIMIT 10;`

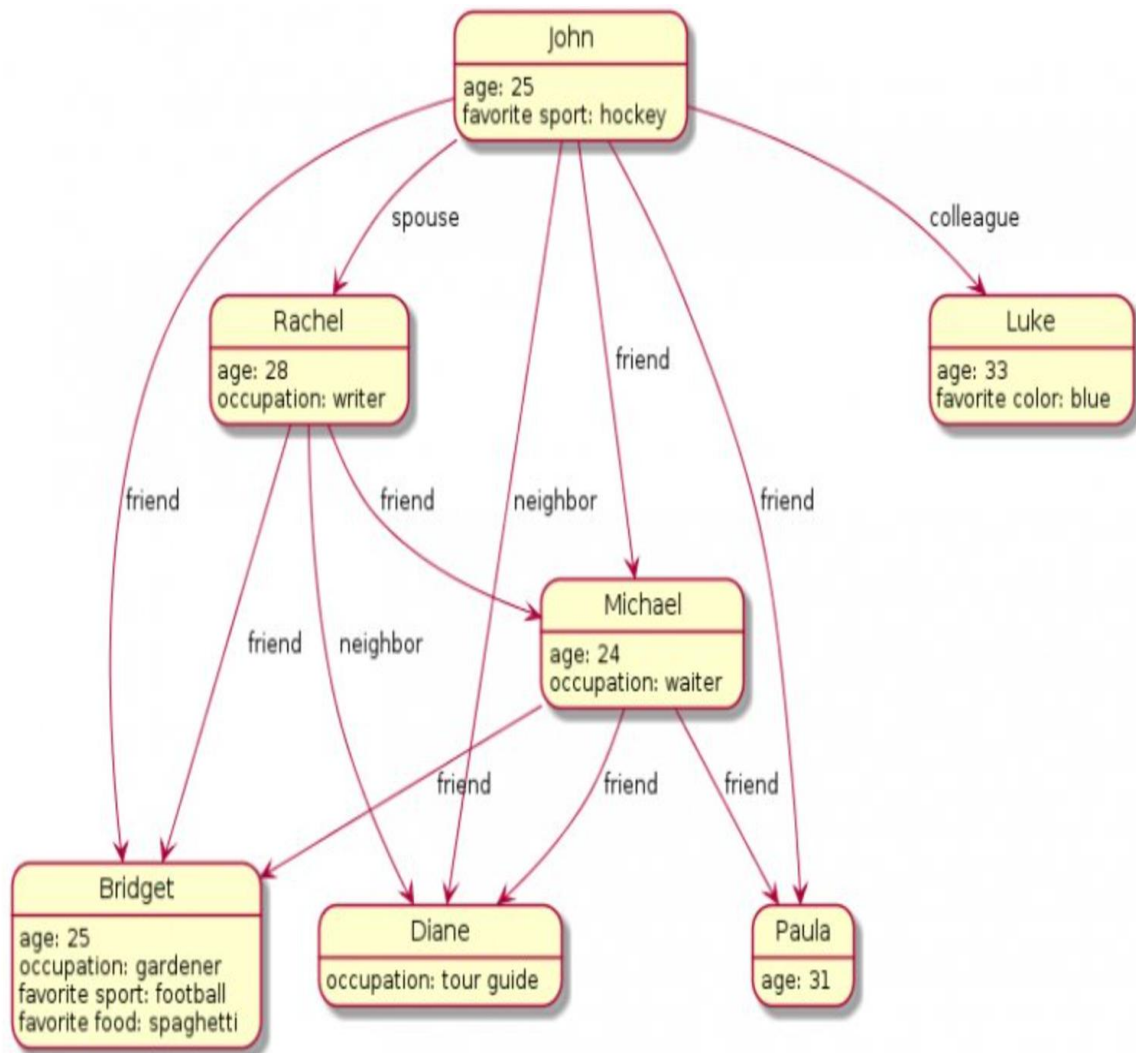


Рисунок 3.6 –Графова база даних – приклад

Таблиця 3.1 – Опис таблиць бази даних

Ім'я таблиці	Опис
1	2
posts	Основна таблиця, у якій зберігаються всі статті (заголовок, зміст) та вся необхідна службова інформація (дата створення, тип, хто автор тощо).
postmeta	Додаткові дані, що належать до публікацій і відсутні у стандартних полях таблиці posts.
users	Відомості про користувачів як авторів статей, з можливістю їх коментувати.
usermeta	Додаткові поля, що необхідні для опису користувачів та доповнення до таблиці users.
comments	Зберігаються усі коментарі для вже наявних статей.
terms	Простий список рубрик, до яких може належати стаття. Частина службової інформації про рубрики.
term_taxonomy	Таблиця слугує для систематизації рубрик, створює так звані «дерева» різних рубрик.

Кінець табл. 3.1

1	2
term_relationships	Тут задається відповідність статті та рубрик, до якої вона належить.
links	Зберігаються посилання та інформація стосовно них.
options	Загальні налаштування «рушія».

Переглянемо таблиці, що були використані для збереження записів, чернеток і бібліотек зображень і медіафайлів.

Код для створення таблиці в MySQL:

```
CREATE TABLE `posts` (
  `ID` bigint(20) unsigned NOT NULL auto_increment,
  `post_author` bigint(20) unsigned NOT NULL default '0',
  `post_date` datetime NOT NULL default '0000-00-00 00:00:00',
  `post_date_gmt` datetime NOT NULL default '0000-00-00 00:00:00',
  `post_content` longtext NOT NULL,
  `post_title` text NOT NULL,
  `post_category` int(4) NOT NULL default '0',
  `post_excerpt` text NOT NULL,
  `post_status` varchar(20) NOT NULL default 'publish',
  `comment_status` varchar(20) NOT NULL default 'open',
  `ping_status` varchar(20) NOT NULL default 'open',
  `post_password` varchar(20) NOT NULL default '',
  `post_name` varchar(200) NOT NULL default '',
  `to_ping` text NOT NULL,
  `pinged` text NOT NULL,
  `post_modified` datetime NOT NULL default '0000-00-00 00:00:00',
  `post_modified_gmt` datetime NOT NULL default '0000-00-00 00:00:00',
  `post_content_filtered` text NOT NULL,
  `post_parent` bigint(20) unsigned NOT NULL default '0',
```

					<i>КвРІПЗ.190143.01.21.ПЗ</i>	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

```

`guid` varchar(255) NOT NULL default "",
`menu_order` int(11) NOT NULL default '0',
`post_type` varchar(20) NOT NULL default 'post',
`post_mime_type` varchar(100) NOT NULL default "",
`comment_count` bigint(20) NOT NULL default '0',
PRIMARY KEY (`ID`),
KEY `post_name` (`post_name`),
KEY `type_status_date` (`post_type`,`post_status`,`post_date`,`ID`),
KEY `post_parent` (`post_parent`)
)

```

Розглянемо детальніше зміст таблиці «Збереження записів, чернеток, бібліотеки медіафайлів».

Таблиця 3.2 – Зміст таблиці «Збереження записів, чернеток, бібліотеки медіафайлів»

Поле	Опис
1	2
ID	Ідентифікатор запису або первинний ключ. Значення – від 1 до 263-1. Максимальне значення комірки – 20-розрядне число.
post_author	Автор статті, який позначається ідентифікатором ID таблиці users.
post_date	Дата публікації, час – місцевий.
post_date_gmt	Дата публікації, час – за Гринвічем.
post_content	Вміст статті. При збереженні та виведенні на сторінку, вміст пропускається через фільтр заміни, тож на екрані можна побачити зовсім не те, що вводилося.
post_title	Заголовок статті чи ім'я для зображення в бібліотеці медіафайлів (посилання на них також мають зберігатися в цій же таблиці).
post_category	Завжди дорівнює 0.
post_excerpt	Коротка витримка, інформація про запис, зображена цитатою. Має вводитися самостійно, за замовчуванням відсутня.
post_status	Статус запису. Publish - опублікована, draft – чернетка, inherit - пов'язана (може бути резервна копія запису чи зображення).
comment_status	Можливість коментувати: open – дозволяється коментувати, closed – не дозволяється.
post_password	Пароль в явному вигляді, якщо стаття з паролем. В іншому випадку – ні.

Кінець таблиці 3.2

1	2
post_name	Назва сторінки для статичного посилання. Записи і зміни задають це ім'я автоматично.
post_parent	Для резервних копій та зображень – ID батьківської запису.
Guid	Повне інтернет-посилання на запис (об'єкт).
menu_order	Дорівнює 0.
post_type	Тип запису: post – звичайний запис, page – запис з постійним посиланням (статичний), revision – автозбереження, attachment – мультимедійний файл (наприклад, зображення).
post_mime_type	Тип вмісту запису. Важливий і потрібний тільки для прикріплень (наприклад, png–зображення проходить як image/png).
comment_count	Кількість коментарів до запису.

Код для створення таблиці в MySQL:

```
CREATE TABLE `postmeta`
(
`meta_id` bigint(20) unsigned NOT NULL auto_increment,
`post_id` bigint(20) unsigned NOT NULL default '0',
`meta_key` varchar(255) default NULL,
`meta_value` longtext,
PRIMARY KEY (`meta_id`),
KEY `post_id` (`post_id`),
KEY `meta_key` (`meta_key`)
)
```

Таблиця 3.3 – Вміст таблиці posts

Поле	Опис
1	2
meta_id	Первинний ключ.
post_id	До якої наші метадані відносяться таблиці posts.
meta_key	Ім'я додаткового параметра. Використовується для прикріплених до статті файлів чи інших метаданих.
meta_value	Значення додаткового параметру. Вміст залежить від імені параметра.

Таблиця 3.4 – Вміст таблиці users

Поле	Опис
ID	Ідентифікатор, первинний ключ.
user_login	Ім'я користувача.
user_pass	Пароль користувача, в закодованому вигляді.
user_nicename	Як звертатись до користувача. Зазвичай збігається з ім'ям для входу.
user_email	Електронна пошта користувача для зв'язку.
user_url	Адреса сайту користувача.
user_registered	Дата реєстрації відповідного користувача.
user_activation_key	Активаційний ключ, який видається після активації.
display_name	Ім'я для відображення.

Ігноруючи існування таблиці users, всі пов'язані з користувачем дані знаходяться тут.

Код для створення таблиці в MySQL:

```
CREATE TABLE `usermeta` (
  `umeta_id` bigint(20) unsigned NOT NULL auto_increment,
  `user_id` bigint(20) unsigned NOT NULL default '0',
  `meta_key` varchar(255) default NULL,
  `meta_value` longtext,
  PRIMARY KEY (`umeta_id`),
  KEY `user_id` (`user_id`),
  KEY `meta_key` (`meta_key`)
)
```

Таблиця 3.5 – Вміст таблиці usermeta

Поле	Опис
umeta_id	Ідентифікатор метазапису, первинний ключ.
user_id	Ідентифікатор користувача, що відноситься до запису.
meta_key	Додатковий параметр.
meta_value	Значення додаткового параметру.

Код для створення таблиці в MySQL:

```
CREATE TABLE `comments` (
  `comment_ID` bigint(20) unsigned NOT NULL auto_increment,
  `comment_post_ID` bigint(20) unsigned NOT NULL default '0',
  `comment_author` tinytext NOT NULL,
  `comment_author_email` varchar(100) NOT NULL default '',
  `comment_author_url` varchar(200) NOT NULL default '',
  `comment_author_IP` varchar(100) NOT NULL default '',
  `comment_date_gmt` datetime NOT NULL default '0000-00-00 00:00:00',
  `comment_content` text NOT NULL,
  `comment_approved` varchar(20) NOT NULL default '1',
  `comment_agent` varchar(255) NOT NULL default '',
  `comment_type` varchar(20) NOT NULL default '',
  `comment_parent` bigint(20) unsigned NOT NULL default '0',
  `user_id` bigint(20) unsigned NOT NULL default '0',
  PRIMARY KEY (`comment_ID`),
  KEY `comment_approved` (`comment_approved`),
  KEY `comment_post_ID` (`comment_post_ID`),
  KEY `comment_approved_date_gmt` (`comment_approved`,`comment_date_gmt`),
  KEY `comment_date_gmt` (`comment_date_gmt`)
)
```

Таблиця 3.6 – Вміст таблиці comments

Поле	Опис
1	2
comment_ID	Ідентифікатор коментарів, первинний ключ.
comment_post_ID	Ідентифікатор заміток, до яких відноситься коментар.
comment_author_email	Електронна пошта.
comment_author	Автор (використовується з даних реєстрації чи вводиться вручну, якщо для коментарів реєстрація не потрібна).

Кінець табл. 3.6

1	2
comment_author_url	Сайт автора коментарів.
comment_author_IP	IP-адреса коментаторів.
comment_date	Дата та час.
comment_content	Самі коментарі.
comment_approved	Чи схвалено коментар до показу.
comment_agent	З якого браузеру або системи заходили.
comment_parent	Батьківський коментар. Потрібний для створення відповідного «дерева».
user_id	Ідентифікатор зареєстрованого користувача.

Код для створення таблиці в MySQL:

```
CREATE TABLE `terms` (
  `term_id` bigint(20) unsigned NOT NULL auto_increment,
  `name` varchar(200) NOT NULL default "",
  `slug` varchar(200) NOT NULL default "",
  `term_group` bigint(10) NOT NULL default '0',
  PRIMARY KEY (`term_id`),
  UNIQUE KEY `slug` (`slug`),
  KEY `name` (`name`)
)
```

Таблиця 3.7 – Вміст таблиці terms

Поле	Опис
term_id	Ідентифікатор.
name	Назва рубрики. Використовується для визначення рубрики майже завжди.
slug	Ярлик рубрики. Зазвичай складається лише з латинських літер в нижньому реєстрі, цифри та тире.
term_taxonomy	Таблиця для об'єднання рубрик в «дерева» та для опису взаємозв'язків рубрик.

Код для створення таблиці в MySQL:

```
REATE TABLE `term_taxonomy` (  
  `term_taxonomy_id` bigint(20) unsigned NOT NULL auto_increment,  
  `term_id` bigint(20) unsigned NOT NULL default '0',  
  `taxonomy` varchar(32) NOT NULL default '',  
  `description` longtext NOT NULL,  
  `parent` bigint(20) unsigned NOT NULL default '0',  
  `count` bigint(20) NOT NULL default '0',  
  PRIMARY KEY (`term_taxonomy_id`),  
  UNIQUE KEY `term_id_taxonomy` (`term_id`,`taxonomy`),  
  KEY `taxonomy` (`taxonomy`)  
)
```

Таблиця 3.8 – Вміст таблиці term_taxonomy

Поле	Опис
term_taxonomy_id	Ідентифікатор, первинний ключ.
term_id	Ідентифікатор рубрики.
taxonomy	category – рубрика, link_category – рубрика посилань.
description	Короткий опис.
parent	Ідентифікатор батьківської рубрики.
count	Кількість асоційованих записів.
term_relationships	Таблиця для зв'язку статті з рубриками.

Код для створення таблиці в MySQL:

```
CREATE TABLE `term_relationships` (  
  `object_id` bigint(20) unsigned NOT NULL default '0',  
  `term_taxonomy_id` bigint(20) unsigned NOT NULL default '0',  
  `term_order` int(11) NOT NULL default '0',  
  PRIMARY KEY (`object_id`,`term_taxonomy_id`),  
  KEY `term_taxonomy_id` (`term_taxonomy_id`)  
)
```

Таблиця 3.9 – Вміст таблиці term_relationships

Поле	Опис
object_id	Який об'єкт описуємо.
term_taxonomy_id	З яким зв'язком рубрик асоціюємо.

Код для створення таблиці в MySQL:

```
CREATE TABLE `links` (
  `link_id` bigint(20) unsigned NOT NULL auto_increment,
  `link_url` varchar(255) NOT NULL default "",
  `link_name` varchar(255) NOT NULL default "",
  `link_image` varchar(255) NOT NULL default "",
  `link_target` varchar(25) NOT NULL default "",
  `link_category` bigint(20) NOT NULL default '0',
  `link_description` varchar(255) NOT NULL default "",
  `link_visible` varchar(20) NOT NULL default 'Y',
  `link_owner` bigint(20) unsigned NOT NULL default '1',
  `link_rating` int(11) NOT NULL default '0',
  `link_updated` datetime NOT NULL default '0000-00-00 00:00:00',
  `link_rel` varchar(255) NOT NULL default "",
  `link_notes` mediumtext NOT NULL,
  `link_rss` varchar(255) NOT NULL default "",
  PRIMARY KEY (`link_id`),
  KEY `link_category` (`link_category`),
  KEY `link_visible` (`link_visible`)
)
```

Загальні налаштування рушія

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

Код для створення таблиці в MySQL:

```
CREATE TABLE `options` (  
  `option_id` bigint(20) unsigned NOT NULL auto_increment,  
  `blog_id` int(11) NOT NULL default '0',  
  `option_name` varchar(64) NOT NULL default "",  
  `option_value` longtext NOT NULL,  
  `autoload` varchar(20) NOT NULL default 'yes',  
  PRIMARY KEY (`option_id`,`blog_id`,`option_name`),  
  KEY `option_name` (`option_name`)  
)
```

Створення сторінок – одна з невід’ємних та найважливіших складових роботи зі створення сайту.

Коригувати сторінку та її зміст можна у вікні текстового редактора, який вбудований у систему, або ж у віконці редагування html-коду. Це дозволяє людям, які не дуже добре розбираються в html, створювати прості сторінки, а решті – збільшити їх функціональність та можливості. Можна також створити материнську, основну, сторінку, в яку помістити підсторінки, або скористатися функцією «Меню» та розмістити каталоги на головній сторінці, до яких розмістити всі інші. Також можна додати аудіо-, фото- і відеофайли.

3.2 Структура та функціональне призначення модулів системи

Якщо розглянути функції, що має виконувати наша система, то можна побачити, що довідково-інформаційна система розділена на два блоки: блок користувача та блок адміністратора.

Блок адміністратора містить наступні вкладки:

- адміністрування послуг;

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

- користувачі;
- керування класом;
- послуги;
- додаткові налаштування.

У вкладці «Адміністрування послуг» знаходяться розділи: «Черга», де адміністратор бачить замовлення та їх виконання, і «Створити», де, власне, формуються замовлення.

У вкладці «Користувачі» містяться розділи: «Переглянути» (зображено на рисунку 3.7), де можна побачити, хто працює на даний час у комп'ютерному клубі, їх особисті дані, можна надіслати їм повідомлення, побачити суму за надані послуги, видалити користувачів, і «Додати», у якій можна додати користувачів та послуги до них або вже наявних.

У вкладці «Керування класом» знаходяться розділи: «Інфраструктура», «Робочі місця», «Типи робочих місць», «Пристрої» (зображено на рисунку 3.8), «Типи пристроїв», «Моделі пристроїв».

Розглянемо розділ «Пристрої» більш детально.

Тут адміністратор може вибрати тип місця та пристрій.

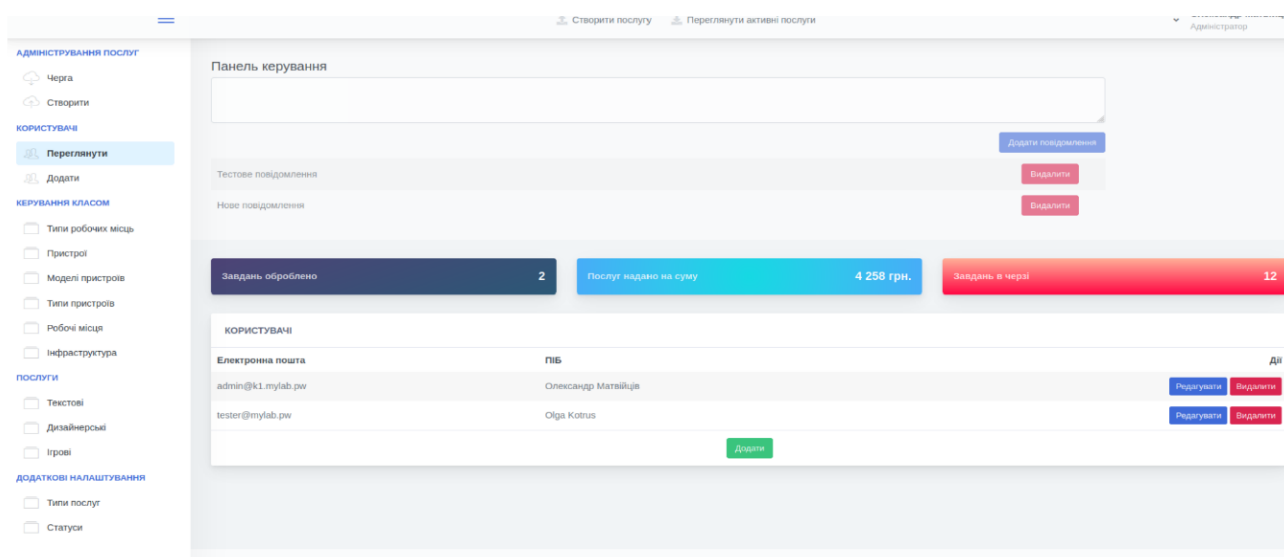


Рисунок 3.7 – Вкладка «Користувачі»(розділ «Переглянути»)

Блок «Користувачі» – для відвідувачів клубу (рисунок 3.10).

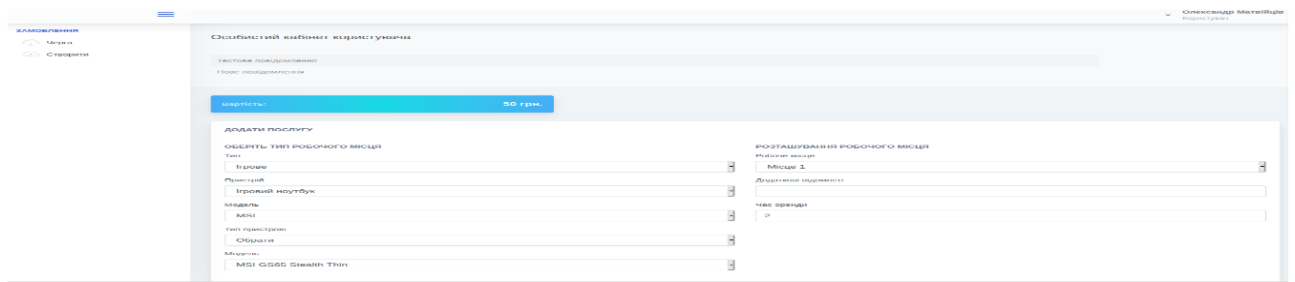


Рисунок 3.10 – Особистий кабінет користувача

Тут користувачі можуть обрати тип бажаного робочого місця, тип комп'ютера, який відповідатиме їхнім потребам, його місцезорозташування, зарезервувати певний час для оренди і одразу побачити вартість обраних ними послуг.

3.3 Інструкція користувача

Наведемо для прикладу інструкцію щодо користування даною довідково-інформаційною системою.

Спочатку потрібно зайти на веб-сайт довідково-інформаційної системи.

Потім варто зареєструватися – для перегляду усіх функцій нашої системи та її кращого розуміння.

Після цього потрібно зайти в особистий кабінет користувача, щоб мати змогу побачити послуги, які використовуються на даний момент.

У віконці «Замовити» можна переглядати можливий час для відвідування, щоб краще розподілити власний час та не створювати незручностей іншим. Тут відвідувач може побачити відповідний номер вільного робочого місця та його вільний час для резерву. Наочно це представлено на рисунку 3.11.

Стосовно адміністратора, то йому так само потрібно зареєструватися, щоб зайти на сторінку, здійснювати адміністрування і побачити роботу даної

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

довідково-інформаційної системи: переглянути усі замовлення користувачів, їх вартість, розташування відвідувачів у залі, послідовність виконання послуг на даний момент та інше.

Замовити

Час	Робоче місце 1	Робоче місце 2	Робоче місце 3	Робоче місце 4	Робоче місце 5
8.00-9.00	вільно	вільно	вільно	вільно	
9.00-10.00		вільно	вільно	вільно	вільно
10.00-10.30		вільно	вільно	вільно	вільно
10.30-11.00	вільно	вільно	вільно	вільно	вільно
11.00-11.30	вільно	вільно	вільно	вільно	вільно
11.30-12.00	вільно	вільно	вільно	вільно	вільно
12.00-12.30	вільно	вільно	вільно	вільно	вільно
12.30-13.00	вільно	вільно		вільно	вільно
13.00-13.30	вільно	вільно		вільно	вільно
13.30-14.00	вільно	вільно		вільно	вільно
14.00-14.30	вільно	вільно	вільно	вільно	вільно
14.30-15.00	вільно	вільно	вільно	вільно	вільно
15.00-15.30	вільно	вільно	вільно	вільно	вільно
15.30-16.00	вільно	вільно	вільно	вільно	вільно

Рисунок 3.11 – Сторінка «Замовити» (фрагмент)

3.4 Технічні характеристики системи

Для якісної роботи потрібно, щоб комп'ютер відповідав таким основним критеріям:

- тактова частота процесора – не менше 1 ГГц;
- обсяг ОП – від 1024 Мбайт;
- вільне місце на жорсткому диску – від 400 Мбайт;
- обсяг відеопам'яті – від 512 Мбайт;
- операційна система – Windows (найкраще – Windows 7 або Windows 10);
- екранне розширення 1024 x 768;
- наявність CD/DVD-rom;
- клавіатура і комп'ютерна миша.

Для запуску на ПК також потрібен Framework 4.5 або вище, для необхідної якісної роботи Windows-додатків, створених на Microsoft Visual Studio. Даний

пакет надається практично з усіма сучасними версіями Windows, але іноді його потрібно довстановлювати.

Код даної програми подано у додатку Б.

3.5 Тестування системи

Після готовності довідково-інформаційної системи комп'ютерного клубу потрібно провести ряд тестів для розуміння її коректної роботи. Розглянемо можливі класифікації та типи тестувань, і оберемо, якими будемо користуватись для нашої перевірки.

Існують наступні види тестувань за рівнем тестування ПЗ:

– модульне тестування, яке перевіряє функціональність і шукає дефекти у доступних частинах додатку, які можуть тестуватися окремими частинами (наприклад, модулі програм, об'єкти, класи, функції тощо);

– інтеграційне тестування, що зазвичай перевіряє взаємодію між компонентами системи і застосовується після перевірки окремих компонентів;

– операційне тестування полягає у перевірці задоволення всіх потреб користувачів і виконання ролі у межах своєї експлуатації, відповідно до визначення у бізнес-моделі конкретної системи. Слід врахувати і можливість помилок у бізнес-моделі, саме тому важливо проводити повне операційне тестування як завершальний крок даної перевірки. До того ж, дане тестування дозволяє виявити і певні нефункціональні труднощі: наприклад, конфлікт з іншими системами, дотичними в області бізнесу або в інших програмних і електронних середовищах, їх системну недостатню продуктивність. Знаходження таких речей на стадії впровадження може стати критичною і дорогою проблемою, тож важливо не лише провести верифікацію, а й перевірка валідації вже на перших етапах розробки ПЗ;

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

– системне тестування, яке перевіряє функціональні та не функціональні вимоги в конкретній системі загалом і виявляє помилки: наприклад, незручність у користуванні, невірне використання ресурсів конкретної системи, непередбачені комбінації даних користувача, відсутня або невірна певна функціональна несумісність з його оточенням, непередбачувані сценарії застосування тощо;

– приймальне тестування, що є формальним процесом тестування, і перевіряє відповідність даної системи відповідним вимогам та проводиться для визначення відповідності системи певним критеріям, а також винесення рішення про прийняття чи неприйняття додатку.

За ознакою позитивності сценаріїв виділяють позитивне та негативне тестування.

За знанням системи виділяють наступні типи:

– тестування «білого ящика», коли відомі вхідні та вихідні дані, а також процес конкретного перетворення;

– тестування «чорного ящика», коли відомі вхідні та вихідні дані, але процес перетворення невідомий;

– тестування «сірого ящика», що об'єднує обидва попередні типи.

За ступенем автоматизації тестування поділяється на:

– автоматизоване тестування;

– ручне тестування;

– напівавтоматизоване тестування.

Види тестування за об'єктом тестування:

– функціональне тестування, що розглядає наперед задану поведінку і ґрунтується на аналізуванні особливостей функцій частин або системи в цілому;

– тестування локалізованої версії програмного продукту;

– тестування безпеки системи;

– тестування продуктивності системи, що має наступні види: навантажувальне тестування; стрес–тестування; тестування стабільності;

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

– тестування зручності використання, спрямоване на встановлення ступеня зручності, зрозумілості та привабливості для користувача, навчання та застосування;

– тестування інтерфейсу користувачів;

– тестування сумісності (тобто перевірка здатності ПЗ працювати у відповідному середовищі).

За ступенем підготовленості до тестування виділяють:

– тестування без документації, яке відбувається на методі передбачення помилки, що ґрунтується на особистому досвіді того, хто тестує продукт;

– тестування по документації;

– дослідницьке тестування, де водночас виконується паралельне навчання, виконання тестування та дизайн тестів.

За ступенем ізольованості компонентів:

– інтеграційне тестування – тобто тестування взаємодії всіх частин програми, які поєднані в одне ціле;

– компонентне тестування – тестування окремо кожної частини системи;

– системне тестування – процес тестування даної інтегрованої системи.

За часом проведення поділяють на такі види:

а) альфа-тестування – коли відбувається імітація реальної роботи системи її розробниками чи справжня робота з системою її замовником або потенційними користувачами; Цей вид має свої підвиди:

– «димове» тестування чи тестування при прийманні готового замовлення;

– тестування при задачі;

– регресивне тестування;

– тестування нової функціональності;

б) бета-тестування, коли можливе поширення версії ПЗ із обмеженими функціями для певної групи осіб, з метою дізнатися, чи готовий продукт містить

					<i>КвРІПЗ.190143.01.21.ПЗ</i>	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

помилки (іноді цей тип тестування виконується для отримання відгуку про продукт від його майбутніх потенційних користувачів).

Сфера веб-застосунків стрімко розвивається. Надійність веб-застосунків є необхідністю, враховуючи розвиток Інтернет-технологій за останні роки, проте вона має свої особливості. Не всі методи та підходи, що застосовуються для того, щоб протестувати стандартні застосунки, можуть бути застосовні для їх тестування.

Веб-застосунок – клієнт-серверний, у якому браузер є клієнтом, а сервером – вебсервер. По суті, це є вже двома окремими програмами, які мають бути протестовані як окремо, так і в сукупності.

Розглянемо структуру веб-додатку на рисунку 3.12.

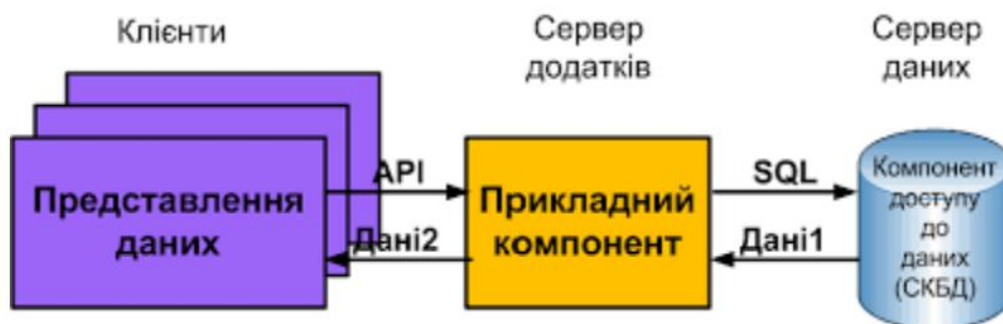


Рисунок 3.12 – Структура веб-застосунку

Більшість сучасних програм орієнтовані на роботу з мережею. Обмін інформацією у веб-застосунках проходить по мережі, тоді як збереження даних відбувається частіше на сервері. Якщо користувач виявляє помилку у середовищі мережі, то іноді буває важко визначити те місце, де вона сталася. Саме тому режим роботи або повідомлення про помилку може бути результатом помилок із різних частин мережевої системи.

Тестування веб-орієнтованих застосунків має багато спільного з тестуванням класичних програм, проте, має особливості, пов'язані, в першу чергу, із середовищем функціонування. Маючи технологічні, структурні та компонентні особливості, у веб-застосунків є характерні особливості режимів інсталяції, запуску, роботи, зупинки та видалення, створення інтерфейсів користувачів.

Працюючи, зазвичай, з мережею та величезною кількістю користувачів, веб-застосунки розмежовують для них права доступу. Логістика веб-застосунку розподілена між сервером та клієнтом, зберігання даних здійснюється на їхньому сервері, а обмін інформацією проходить по певній мережі. Одним з переваг цього підходу є те, що клієнти не залежать від конкретної ОС користувача, а, отже, веб-застосунки є міжплатформними сервісами.

У таблиці 3.10 наведено відмінності тестування веб-застосунків від класичних програм.

Таблиця 3.10 – Відмінності в тестуванні застосунків

Тип відмінності	Класичний застосунок	Веб-застосунок
1	2	3
Технологічні відмінності	Працює з використанням однієї або сімейства подібних технологій.	Працює з використанням різних технологій.
Структурні відмінності	Застосунок є «монолітним» та складається з одного або невеликої кількості частин. Не використовує сервери бази даних та веб-сервери.	«Багатокомпонентний»: складається з багатьох частин. Обов'язково використовує сервери баз даних, веб-сервери та сервери додатків.
Відмінності в режимах роботи	Працює, зазвичай, в режимі реального часу - про дії користувача відомо відразу ж після виконання.	Працює в режимі «запит-відповідь»: про дії стає відомо тільки після запиту на сервер.
Відмінність формування інтерфейсу	Використовує для формування інтерфейсу користувача усталені та стандартизовані технології.	Використовує для формування інтерфейсу технології, що стрімко розвиваються та які конкурують між собою.
Відмінності роботи з мережею	Практично не використовує мережеві канали передачі даних.	Активно використовує мережеві канали для передачі даних.
Відмінності запуску і зупинки	Запускається і зупиняється не часто.	Запускається і зупиняється за фактом надходження кожного запиту, тобто дуже часто.
Різниця в кількості користувачів	Кількість користувачів, що одночасно використовують застосунок, піддається контролю, вона обмежена і є передбачуваною.	Кількість користувачів, що одночасно використовують застосунок, важко прогнозується і вона сильно може змінюватися.
Особливості збоїв і відмов	Вихід з ладу тих чи інших компонентів застосунку відразу стає очевидним.	Вихід з ладу деяких компонентів надає непередбачуваний вплив на працездатність усієї програми в цілому.

Кінець табл. 3.10

1	2	3
Відмінності інсталяції	Процес інсталяції стандартизований та орієнтований на широку аудиторію користувачів. Не вимагає специфічних знань. Додавання компонентів програми виконується стандартним способом та з використанням одного і того ж інсталятора.	Процес інсталяції досить часто недоступний кінцевому користувачеві. Інсталяція вимагає специфічних знань. Процес зміни компонент програми не передбачається або ж потребує специфічної кваліфікації користувачів. Інсталятор відсутній.
Відмінності в деінсталяції	Процес деінсталяції стандартизований та виконується автоматично або ж напівавтоматично.	Процес деінсталяції вимагає втручання адміністратора та пов'язаний із зміною коду в середовищі функціонування програми.

Щоб перевірити стійкість до навантаження системи, можуть бути використані спеціальні програми, що перевіряють можливості системи. За допомогою однієї з них пройшла перевірку наша довідково-інформаційна система. Після проведення перевірки на навантаження, програма видала діаграму (рисунок 3.13), згідно якої ми бачимо, що вона здатна працювати із значною кількістю користувачів одночасно.

На малюнку знаходиться діаграма, де на осі y вказана кількість одночасних користувачів на сайті, а на осі x – час проходження даного тесту.

Тестовий сценарій – це послідовність дій, спрямованих на перевірку системи та отримання відповідних результатів на виході. Ця інформація може подаватися у вигляді таблиць у документі Excel, в якому кожен рядок показує відповідний вплив на систему, її стан та отримані результати. Найбільш неконтрольованим, а тому складним є процес інтерпретації вимог, специфікацій до певних тестових послідовностей.

Показником якості виборів на виробництві може бути визначена така характеристика, як необхідний час для створення програми задля нової технології написання сценаріїв тестування (для прикладу наводимо таблицю 3.11).

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

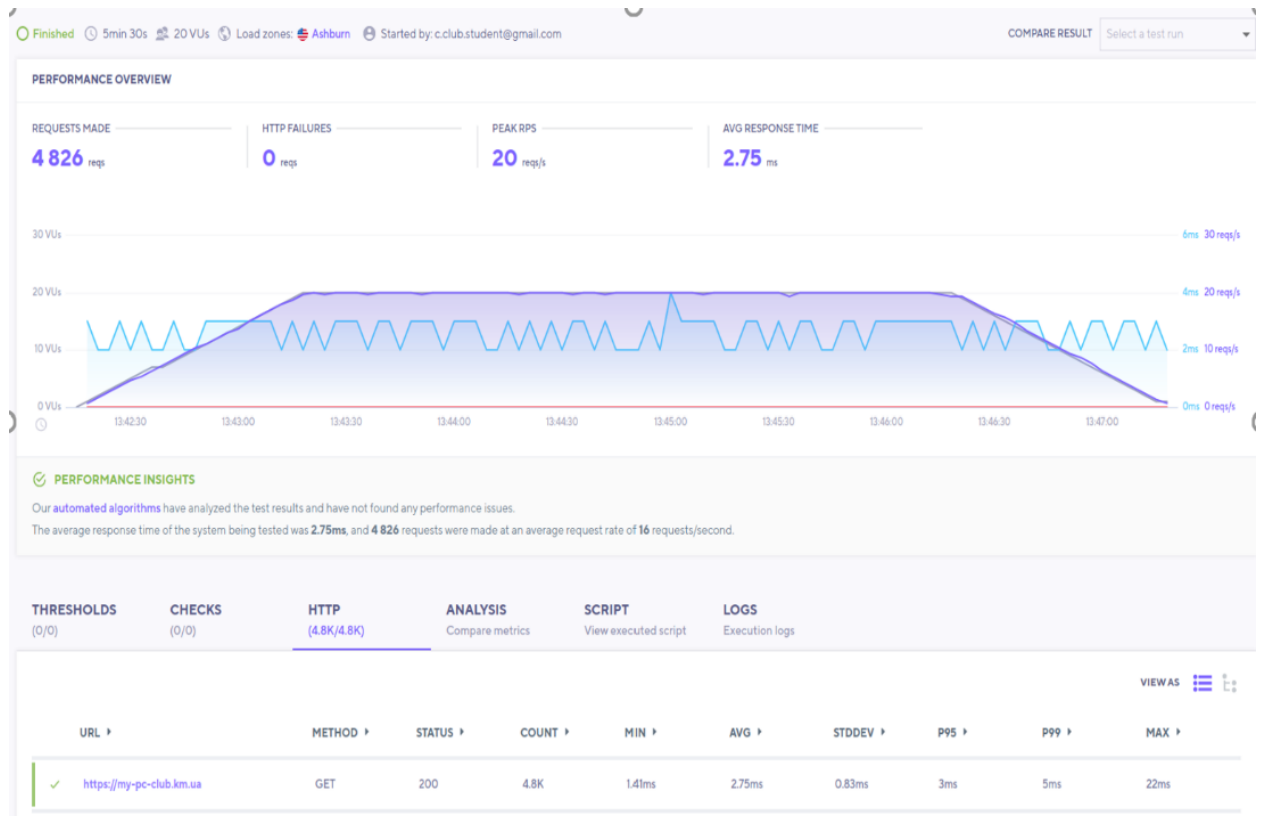


Рисунок 3.13 – Результати навантажувального тестування

Цей критерій визначається як щабель трудовитрат із розроблення програмного забезпечення. Процес створення подібного сценарію має мати чіткі послідовності дій, їх переходів та умов завершення.

Таблиця 3.11 – Сценарій для тестування інтерфейсу користувача

Дія	Реакція програми	Результат
1. Натиснути на кнопку «Реєстрація».	З'являється діалогове вікно, у якому потрібно ввести логін та пароль.	Вірно.
2. У діалоговому вікні ввести ім'я користувача user та пароль 000000. Потім натиснути кнопку «Вхід».	Вікно «Вхід» має закритися.	Невірно. Вікно входу має відкритись.
3. Завершити сеанс роботи з вкладкою, натиснувши кнопку «Замовити».	Вкладка «Замовити» має закритися, і має відкритись наступне вікно.	Вірно.

3.6 Висновок до розділу 3

У цьому розділі здійснено програмну реалізацію системи. Була подана структура і функціональне призначення частин програми, показано розробку та застосування бази даних, створено систему користування, а також проведено тестування системи. Наша довідково-інформаційна система тестування пройшла.

					<i>КвРІПЗ.190143.01.21.ПЗ</i>	Арк.
						60
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ВИСНОВКИ

Під час виконання кваліфікаційної роботи поставлену мету, а саме: розробка довідково-інформаційної системи комп'ютерного клубу, було досягнуто.

У першому розділі роботи здійснено дослідження предметної області та виділення функціональних і нефункціональних вимог для довідково-інформаційної системи, яка розроблялася, та окреслено технічне завдання роботи і постановку її задачі. Також у даному розділі був поданий аналіз існуючих довідково-інформаційних систем цього типу та виявлено їх позитивні та негативні сторони.

У другому розділі проведено проектування довідково-інформаційної системи, зокрема структура, архітектура, база даних та інтерфейс користувачів. Також було проведено аналіз технологій і методів реалізації довідково-інформаційної системи, обрано підходящі для нашої роботи.

У третьому розділі проведена програмна реалізація довідково-інформаційної системи. Окрім цього, подано функціональне призначення модулів програми та її структуру, а також продемонстровано реалізацію бази даних та розроблено відповідний інтерфейс. Також проведено тестування програмного засобу, результатом якого стало підтвердження його можливої реалізації.

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Історія відеоігор. URL: [https://uk.wikipedia.org/wiki/-%D0%86%D1%81%D1%82%D0%BE%D1%80%D1%96%D1%8F_%D0%B2%D1%96%D0%B4%D0%B5%D0%BE%D1%96%D0%B3%D0%BE%D1%80](https://uk.wikipedia.org/wiki/%D0%86%D1%81%D1%82%D0%BE%D1%80%D1%96%D1%8F_%D0%B2%D1%96%D0%B4%D0%B5%D0%BE%D1%96%D0%B3%D0%BE%D1%80) (дата звернення: 21.02.2023)
2. Що таке кіберспорт та як ця культура розвинена в Україні. URL: <https://www.radiosvoboda.org/a/29189982.html> (дата звернення: 21.02.2023)
3. Турніри із кіберспорту 2023. URL: <https://tips.gg/ua/tournaments/> (дата звернення: 27.02.2023)
4. 8 технологічних революцій України. Революція восьма: кіберспорт. URL: <https://techiiia.com/ua/news/8-tehnologichnih-revolyucij-ukrayini-revolyuciya-vosma-kibersport> (дата звернення: 06.03.2022).
5. Natus Vincere (Dota 2). URL: [https://uk.wikipedia.org/wiki/-Natus_Vincere_\(Dota_2\)](https://uk.wikipedia.org/wiki/Natus_Vincere_(Dota_2)) (дата звернення: 10.03.2023).
6. Ігрова індустрія, історії перспективи. Як стати розробником ігор? URL: <https://lviv.logos-academy.com/igrova-industriya-istoriyi-perspektyvy-yak-staty-rozrobnykom-igor> (дата звернення: 10.03.2023).
7. Як керувати комп'ютерним клубом. URL: <https://techiiia.com/ua/news/yak-keruvati-kompyuternim-klubom-istoriya-ukrayinskogo-saas-servisuyakij-pidkoriv-30-krayin> (дата звернення: 12.03.2023).
8. Ігрова індустрія в цифрах: скільки українці витрачають на відеоігри. – URL: <https://mind.ua/publications/20222353-igrova-industriya-v-cifrah-skilki-ukrayinci-vitrachayut-na-videoigri> (дата звернення: 15.03.2023).
9. Ігрова індустрія у 2022 році. URL: <https://itc.ua/ua/novini/igrova-industriya-u-2022-roczy-obyem-200-milyardiv-dolariv-ponad-3-mlrd-gravcziv-i-istorichna-peremoga-ssha-nad-kitayem-prognoz-newzoo/> (дата звернення: 16.03.2023).
10. SENET. URL: <https://senet.cloud/> (дата звернення: 17.03.2023).

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

11. Система керування комп'ютерним клубом Locker URL: <https://androidas.ru/sistema-upravleniya-kompyuternym-klubom-locker-programma-dlya/> (дата звернення: 20.03.2023).
12. Комп'ютерний клуб PlayHub. URL: <https://playhub-internetcafe.business.site> (дата звернення: 25.03.2023).
13. Мережа кібер хабів ItLand. URL: <https://itland.kiev.ua> (дата звернення: 01.04.2023).
14. Переваги та недоліки WordPress. URL: <https://infosoft.ua/ua/perevahy-i-nedoliky-wordpress> (дата звернення: 03.04.2023).
15. UML – мова моделювання та документування складних систем. URL: http://ni.biz.ua/2/2_7/2_74751_UML---yazik-modelirovaniya-i-dokumentirovaniya-slozhnih-sistem.html (дата звернення: 10.04.2023).
16. Уніфікована мова моделювання UML. URL: <http://www.znannya.org/-?view=uml> (дата звернення: 13.04.2023).
17. Як будувати UML-діаграми. URL: <https://dou.ua/forums/topic/40575/> (дата звернення: 13.04.2023).
18. Типи баз даних: особливості, відмінності та приклади. URL: <https://dou.ua/lenta/articles/types-of-databases/> (дата звернення: 16.05.2023).
19. База даних MySQL. URL: <https://promoter.net.ua/articles/baza-danix-mysql.html> (дата звернення: 13.04.2023).
20. SQL Server 2022. URL: <https://www.microsoft.com/en-us/sql-server/sql-server-2022> (дата звернення: 17.04.2023).
21. Вивчення основ PHP. URL: <http://weplib.com.ua/projects/2> (дата звернення: 17.04.2023).
22. Васильєв О. Програмування мовою PHP: навч. посібник. К.: В-тво Ліра-К, 2022. 368 с.
23. Active Server Pages. URL: [https://learn.microsoft.com/en-us/previous-versions/iis/6.0-sdk/ms526064\(v=vs.90\)](https://learn.microsoft.com/en-us/previous-versions/iis/6.0-sdk/ms526064(v=vs.90)) (дата звернення: 17.04.2023).

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

24. Active Server Pages. What Does Active Server Pages Mean? URL: <https://www.techopedia.com/definition/23088/active-server-pages-asp> (дата звернення: 20.04.2023).

25. Чому Java – найпопулярніша мова програмування у світі. URL: <https://java.lviv.ua/chomu-java-najpopulyarnisha-mova-programuvannya-u-sviti> (дата звернення: 20.04.2023).

26. JavaScript. Підручник. Основи веб-програмування. URL: <https://w3schoolsua.github.io/js/index.html#gsc.tab=0> (дата звернення: 20.04.2023).

27. Content Management System. URL: <https://docs.plesk.com/ru-RU/12.5/customer-guide/%D1%81%D0%B0%D0%B9%D1%82%D1%8B-%D0%B8-%D0%B4%D0%BE%D0%BC%D0%B5%D0%BD%D1%8B/%D1%81%D0%BE%D0%B4%D0%B5%D1%80%D0%B6%D0%B8%D0%BC%D0%BE%D0%B5-%D1%81%D0%B0%D0%B9%D1%82%D0%BE%D0%B2/content-management-systems.74103/> (дата звернення: 25.04.2023).

28. What Is a CMS and Why Should You Care? URL: <https://blog.hubspot.com/blog/tabid/6307/bid/7969/what-is-a-cms-and-why-should-you-care.aspx> (дата звернення: 25.04.2023).

29. Drupal. URL: <https://astwellsoft.com/uk/blog/cms/drupal.html> (дата звернення: 25.04.2023).

30. Створіть ваш сайт з Drupal. URL: <https://smile-ukraine.com/ua/-drupal/introduction> (дата звернення: 25.04.2023).

31. Можливості на Joomla. URL: <https://joomla-ua.org/features> (дата звернення: 28.04.2023).

32. Особливості розробки сайту на CMS Joomla. URL: <https://cityhost.ua/uk/-blog/osobennosti-razrobotki-sayta-na-cms-joomla.html> (дата звернення: 17.04.2023).

33. Так що ж таке WordPress? URL: <https://wpfan.com.ua/tak-scho-zh-take-wordpress/> (дата звернення: 28.04.2023).

34. Переваги та недоліки WordPress. URL: <https://esfirum.com/blog/pros-and-cons-cms-wordpress/> (дата звернення: 28.04.2023).

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

35. WordPress – однозначно популярна CMS. URL: <https://hostpro.ua/blog/-ua/wordpress-demanded-cms> (дата звернення: 28.04.2023).

36. CMS Рейтинг: ТОП-22 CMS для інтернет-магазину у 2021 році. URL: <https://rocketmen.com.ua/ua/article/cms2021> (дата звернення: 30.04.2023).

37. 10 популярних CMS для інтернет-магазинів. URL: <https://project-seo.net/uk/blog-uk/10-populyarnyh-cms-dlya-internrt-magazyniv/> (дата звернення: 30.04.2023).

38. Magento. URL: <https://uk.wikipedia.org/wiki/Magento> (дата звернення: 30.04.2023).

39. NetCat (система керування контентом). URL: [https://uk.wikipedia.org/wiki/NetCat_\(%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BA%D0%BE%D0%BD%D1%82%D0%B5%D0%BD%D1%82%D0%BE%D0%BC\)](https://uk.wikipedia.org/wiki/NetCat_(%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BA%D0%BE%D0%BD%D1%82%D0%B5%D0%BD%D1%82%D0%BE%D0%BC)) (дата звернення: 30.04.2023).

40. Що таке CMS і які види CMS для сайтів бувають. URL: <https://cityhost.ua/uk/blog/chto-takoe-cms-i-kakie-vidy-cms-dlya-saytov-byvayut.html> (дата звернення: 30.04.2023).

41. Каскадні таблиці стилів (CSS). URL: http://ni.biz.ua/4/4_8/4_-8446_kaskadnie-tablitsi-stiley-CSS.html (дата звернення: 03.05.2023).

					КвРІПЗ.190143.01.21.ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК А
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

Введення

Робота виконується в рамках кваліфікаційної роботи «Довідково-інформаційна система комп'ютерного клубу».

1 Підстава для розробки

Підставою для розробки є «Завдання на кваліфікаційну роботу», затверджене завідувачем кафедри інженерії програмного забезпечення.

Найменування розробки: «Довідково-інформаційна система комп'ютерного клубу»

2 Призначення розробки

Довідково-інформаційна система комп'ютерного клубу призначена для узагальнення та отримання інформації по роботі даного закладу і можливість вести облік відвідувачів та їх види діяльності з одного боку та надання послуг їм з іншого.

Користувачами програми є адміністратор та відвідувачі клубу.

Адміністратор може контролювати роботу всього закладу. Він може бачити та керувати всіма послугами комп'ютерного клубу, а саме, бачити, що за обладнання замовив відвідувач, на якому місці він сидить, який час він використовує та одразу бачить загальну вартість послуг. Також він може проводити зміни у замовленнях безпосередньо зі свого робочого місця.

Відвідувач може вибирати робоче місце та його тип, бачити графік зайнятості та місцерозташування в комп'ютерному клубі відвідувачів та вільні робочі місця на протязі цілого дня.

3 Вимоги до програми

3.1 Вимоги до функціональних характеристик

Довідково-інформаційна система комп'ютерного клубу має забезпечувати виконання наступних функцій:

- введення, зберігання, пошук та обробку інформації за замовленнями;

- ведення журналу реєстрації;
- своєчасне формування звітів по оплаті послуг;
- формування інформації по наповненості клубу.

У програмі необхідно передбачити: можливість коригування налаштувань системи; резервне збереження даних; можливість зміни пароля входу в систему; наявність вбудованої довідкової системи; швидкий пошук необхідних документів та довідкової інформації тощо.

3.2 Вимоги до надійності

Розроблюване система повинна мати:

- можливість самовідновлення після збоїв (відключення електроживлення, збої в операційній системі тощо);
- парольний захист при запуску програми;
- обмеження несанкціонованого доступу до даних;
- можливість резервного копіювання інформаційної бази;
- розмежування прав користувачів;
- контроль інформації, що вводиться, та блокування некоректних дій користувача при роботі із системою.

3.3 Вимоги до складу та параметрів технічних засобів

Системні вимоги для роботи системи мають бути наступними:

- тактова частота процесора - 1200 Гц;
- обсяг оперативної пам'яті - 64 Мб;
- обсяг вільного дискового простору - 50 Мб;
- роздільна здатність монітора 1024x768.

3.4 Вимоги до інформаційної та програмної сумісності

Програма повинна працювати в операційних системах Windows XP/7/10. Всі звіти, що формуються, повинні мати можливість експортування у редактор електронних таблиць MS Excel.

3.5 Вимоги до транспортування та зберігання

Умови експлуатації програмного забезпечення збігаються з умовами експлуатації ПК.

3.6 Спеціальні вимоги

Програма повинна мати дружній інтерфейс, розрахований на користувача середньої кваліфікації (з точки зору комп'ютерної грамотності).

З огляду на обсяг проекту завдання передбачається вирішувати поетапно. При цьому модулі системи, створені в різний час, повинні передбачати можливість нарощування системи і бути сумісними один з одним; тому документація на прийняте експлуатаційне ПЗ повинна містити повну інформацію, необхідну для роботи з ним користувачів.

Мова програмування визначається вибором виконавця, при цьому він має забезпечувати можливість інтеграції системи з пакетом MS Office.

4 Вимоги до програмної документації

В ході розробки програми мають бути підготовлені:

- текст програми;
- опис програми;
- програма;
- методика випробувань;
- керівництво користувача.

5 Техніко-економічне обґрунтування

Для роботи комп'ютерного клубу, який, завдяки розвитку кіберспорту переживає своє відродження, доцільно мати довідково-інформаційну систему свого клубу з максимально дружнім інтерфейсом для користувача та простотою у роботі. На даний час можна виділити дві категорії людей, які можуть відвідувати комп'ютерний клуб – молодь, яка захоплюється комп'ютерними іграми, та старше покоління, яке користується переважно соціальними мережами.

6 Стадії та етапи розробки

Стадії та етапи розробки відповідні затвердженому графіку.

ДОДАТОК Б (ОБОВ'ЯЗКОВИЙ)

КОД (ЛІСТИНГ) ПРОГРАМИ

```

<?php

class Devices {
    public function construct()
    {
        parent::construct();
        do_action('dashboard.device_controller_init_before');
        if (!current_user_can('access_device|manage_device')) {
            sp_not_permitted();
        }
        breadcrumb_add('dashboard.device',    (Пристрій '),
            url_for('dashboard.device'));
        view_set('device    active', 'active');
        do_action('dashboard.device_controller_init_after');
    }
    public function index() {
        $app = app();
        $clubModel = new ClubModel;
        $currentPage = (int) $app->request->get('page', 1);
        $itemsPerPage = 12;
        $sort = $app->request->get('sort', null);
        if (!$clubModel->isSortAllowed($sort)) { $sort = 'newest';
        }
        $mime = $app->request->get('type', null);
        $sortRules = $clubModel->getAllowedSorting(); $mimeRules = $clubModel->
        >mimeSortType; array_unshift($mimeRules, 'everything');
        $filters = [ 'sort' => e_attr($sort)
        ];
        $filters['where'][] = ['clubjype', '=', 'attachment'];
        if ($mime && $clubModel->isValidSortMimeType($mime)) {
            $mimeRegex = $clubModel->getMimeTypeRegex($mime);
            $filters['where'][] = ['club_mimetype', 'REGEXP', $mime]; } else {
            $mime = 'everything';
        }
        $totalCount = $clubModel->countRows(null, $filters);
        $queryStr = request_build_query(['page', 'sort', 'type']);
        $pagination = new Pagination($totalCount, $currentPage, $itemsPerPage);
        $pagination->setUrl("?page=@id&sort={$sort}&type={$mime}{$queryStr}");
        $paginationHtml = $pagination->renderHtml();
        $offset = $pagination->offset();
        $entries = $clubModel->readMany( ['*'],
        $offset,
        $itemsPerPage, $filters
        foreach ($entries as $key => $entry) { $entry = $entry;
        $entry['club_ext'] = pathinfo($entry['club_path'], PATHINFO_EXTENSION);
        $entry['club_file_type'] = $clubModel->getFiletype($entry['club_ext'],
        $entry['club_minnetype']);
        $entry['club_url'] = uploads_uri($entry['club_path']);
        $entry['club_rel_path'] = trailingslashit(UPLOADS_DIR). $entry['club_path'];
        $entry['club_readable_size'] = format_size_units($entry['club_size']);
        $entry['club_filename'] = $entry['club_title'] .'.'. $entry['club_ext'];
        if (preg_match('#Aimage/#i', $entry['club_mimetype'])) {
            $entry['club_thumbnail'] = sp_thumbnail_uri($entry['club_rel_path']); } else {

```

```

$entry['club_thumbnail'] =
site_uri("assets/img/nmedia/{$entry['club_file_type']}.png");
}
$entries[$key] = Sentry;
}
$data = [ 'list_entries' => $entries,
'max_upload_size' => format_bytes(get_max_upload_size(), 'M', 0),
'allowedjiletypes' => $clubModel->getAllowedFileTypes(),
'totaljitems' => $totalCount,
'offset' => $offset === 0 ? 1 : $offset,
'current_page' => $currentPage,
'items_per_page' => $itemsPerPage,
'current_items' => $itemsPerPage * $currentPage,
'sortjtype' => $sort,
'mimejtype' => $mime,
'pagination_html' => $paginationHtml,
'sorting_rules' => $sortRules,
'mime_rules' => $mimeRules,
'query_str' => $queryStr
];
return view('admin::device/index.php', $data);
public function create() {
if (!current_user_can('manage_device')) { response_status(403);
return response_body( "Доступ запрещено ");
}

$uploadPath = uploadspath(date('Y/M/d'));

if (!is_dir($uploadPath)) { mkdir($uploadPath, 0755, true);
}
$nonAbsoluteUploadPath = date('Y/M/d');
$clubModel = new ClubModel;
$storage = new FileSystem($uploadPath);
$allowedExtensions = $clubModel->getAllowedFileTypes();
$maxSize = format_bytes(get_max_upload_size(), 'M'). 'M';
$file = new File('file', $storage);

$originalFileName = $file->getName(); $fileName = $originalFileName;

$i = 1;
while (file_exists($uploadPath . '/' . $fileName . '.' . $file->getExtension())) {
$i++;
$fileName = $originalFileName . " ({$i})";
}

$file->setName($fileName);
$fileData = [ 'name' => $file->getNameWithExtension(), 'rawName' => $file-
>getName(), 'extension' => $file->getExtension(), 'mime' => $file-
>getMimeType(), 'size' => $file->getSize()
];

$file->addValidations([ new Extension($allowedExtensions), new Size($maxSize)
]);

try{
$file->upload(); } catch (\Exception $e) { $errors = $file->getErrors();
response_status(403); return response_body(join($errors, "\n"));
}

$filePath = trailingslashit($nonAbsoluteUploadPath) . $fileData['name'];
$data = [ 'clubjtitle' => $fileData['rawName'], 'club_size' => $fileData['size'],
'club_mimejtype' => $fileData['mime'],
];

```

```

$clubModel- >addAttachment($filePath, $data);

$jsonData = $data; $jsonData['success'] = true; $jsonData['club_url'] =
uploads_uri($filePath);
$jsonData['club_relative_url'] = leadingslashit(UPLOADS_DIR).
leadingslashit($filePath);
return json($jsonData);
public function update($id) {
breadcnjmb_add('dashboard.device.update', ('Поновити '));
$clubModel = new ClubModel; $device = $clubModel- >read($id); if (!$device) {
flash ('device- danger1,      ('Пристроїв не знайдено '));
return redirect_to('dashboard.device');
}

$data = [];
return view('admin::device/update.php', $data);
}

public function deletePOST($id) {
if (!$current_user_can('manage_device')) { return sp_not_permitted();
}
$clubModel = new ClubModel; $filters = [];
$filters['where'][] = ['clubjype', '=', 'attachment']; $deviceItem = $clubModel-
>read($id, ['club_path'], $filters);

if (!$deviceItem) {
flash ('device- danger1,      ('Пристроїв не знайдено '));
return;
}

$clubModel- >deleteAttachment($id);

flash ('device- success',      ('Пристрій видалено '));
}
}
<?php

class DashboardUsersController {
public function  construct()
{
parent::  construct);

do_action('dashboard.users_controller_init_before');
breadcrumb_add('dashboard.users',      ('Користувачі'), url_for('dashboard.users'));
view_set('users  active', 'active');
do_action('dashboard.users_controller_init_after');
}

public function index() {
if (!$current_user_can('add_user|edit_user|delete_user')) { sp_not_permitted();
}
$app = app();
$userModel = new UserModel;
$roleModel = new RoleModel;

$usersTable = $userModel- >getTable(); $rolesTable = RoleModel::getTable();
$currentPage = (int) $app- >request- >get('page', 1);
$search = $app- >request- >get('s', null);
$itemsPerPage = (int) $app- >config('dashboard.items_per_page'); $sort = $app-
>request- >get('sort', null);
if (!$userModel- >isSortAllowed($sort)) { $sort = 'oldest'; }
$sortRules = $userModel- >getAllowedSorting();
$filters = [ 'sort' => e_attr($sort)

```

```

];

$roleId = (int) $app->request->get('role_id', 0);

$role = [ 'role_id' => $roleId, 'role_name' => null, 'users_count' => 0
];
if ($roleId) {
$roleQuery = $roleModel->read($roleId, ['role_name']);
if (!$roleQuery) {

$roleId = 0; } else {
$role['role_name'] = $roleQuery['role_name'];
$filters['where'][] = ["{$usersTable}.role_id", '=', $roleId];
}
}
if ($filter) {
$filters['where'][] = ["{$usersTable}.full_name", 'LIKE', "%$search%"];
$filters['where'][] = ["{$usersTable}.email", 'LIKE', "%$search%", 'OR'];
$filters['where'][] = ["{$usersTable}.username", 'LIKE', "%$search%", 'OR'];
$filters['where'][] = ["{$usersTable}.user_ip", 'LIKE', "%$search%", 'OR'];
}
$totalCount = $userModel->countRows(null, $filters);
$role['users_count'] = $totalCount;
$queryStr = request_build_query(['page', 'sort']);
$pagination = new Pagination($totalCount, $currentPage, $itemsPerPage); $pagination->setUrl("?page=@id&sort={$sort}{$queryStr}");
$paginationHtml = $pagination->renderHtml();
$offset = $pagination->offset();
$fields[] = "{$usersTable}.*"; $fields[] = "{$rolesTable}.role_name";
$sql = $userModel->select($fields) ->leftJoin(
$rolesTable,
 "{$usersTable}.role_id",
 "{$rolesTable}.role_id"
);
$sql = $sql->limit($itemsPerPage, $offset);
$sql = $userModel->applyModelFilters($sql, $filters); $stmt = $sql->execute();
$entries = $stmt->fetchAII();
$roleList = [];
$roleList[] = [ 'role_id' => 0,
'users_count' => $userModel->countRows(), 'role_name' => _('AII'),
];

$protectedRoles = $roleModel
->select(['role_id', 'role_name'])
->where('is_protected', '=', 1)
->execute()
->fetchAII();
$currentRolesProtected = false;
foreach ($protectedRoles as $key => $_role) {
$_role['users_count'] = $roleModel->getUsersCountUnderRole($_role['role_id']);
$roleList[] = $_role;
if ($roleId === $_role['role_id']) { $currentRolesProtected = true;
}
}
if (!$currentRolesProtected && $roleId) { $roleList[] = $role;
}
$data = [
'page_subheading' => ('Користувачі'),
'list_entries' => $entries,
'totalItems' => $totalCount,
'offset' => $offset === 0 ? 1 : $offset,
'current_page' => $currentPage,
'items_per_page' => $itemsPerPage,

```



```

return redirect_to_current_route(); break; case 'verify':
if (!current_user_can('edit_user')) { sp_not_permitted();
}
$i = 0;
foreach ($userIDs as $id) {
$userModel->update($id, ['is_verified' => 1]); $i++;
}
flash('users- success', sprintf(('%d %s був доданий', $i)));
return redirect_to_current_route(); case 'unverify':
if (!current_user_can('edit_user')) { sp_not_permitted();
}
$i = 0;
foreach ($userIDs as $id) {
$userModel->update($id, ['is_verified' => 0]); $i++;
}
flash('users- success', sprintf(, $i)); return redirect_to_current_route();
case 'block':
if (!current_user_can('change_user_status')) { sp_not_permitted();
}
$i = 0;
foreach ($userIDs as $id) {
$userModel->update($id, ['is_blocked' => 1]); $i++;
}
flash('users- success', sprintf(('%d'), $i));
return redirect_to_current_route(); case 'unblock':
if (!current_user_can('change_user_status')) { sp_not_permitted();
}
$i = 0;
foreach ($userIDs as $id) {
$userModel->update($id, ['is_blocked' => 0]); $i++;
}
flash('users- success', sprintf(('%d'), $i));
return redirect_to_current_route(); default:
flash('users- warning', ('Помилка, спробуйте пізніше '));
return redirect_to_current_route();
}
}
public function create() {
if (!current_user_can('add_user')) { sp_not_permitted();
}
public function create() {
if (!current_user_can('add_user')) { sp_not_permitted();
}
breadcrumb_add('dashboard.users.create', ('Додати '));
$roles = [];
if (current_user_can('change_user_role')) { $roleModel = new RoleModel;
$roles = $roleModel->readMany(['role_id', 'role_name'], 0, 50);
}
$data = [ 'rolejist' => $roles
];
return view('admin::users/create.php', $data);
}
public function createUser() {
if (!current_user_can('add_user')) { sp_not_permitted();
}
$app = app();
$req = $app->request;
$data = [
'email' => $req->post('email'),
'password' => $req->post('password'),
'username' => $req->post('username'),
'role_id' => $req->post('role_id'),
'full_name' => $req->post('full_name'),

```

```

    'user_ip' => $req->post('user_ip'),
];
$userModel = new UserModel;
$roleModel = new RoleModel;
$v = new Validator($data);
$v->labels([
    'email' => __('Електронна пошта'),
    'password' => __('Пароль'),
    'username' => __('Користувач'),
    'full_name' => __('ПІБ'),
])->rule('required', ['email', 'password'])
->rule('email', 'email')
->rule('uniqueEmail', 'email')
->rule('uniqueUsername', 'username')
->rule('lengthMin', 'password', (int) config('internal.password_minlength'))
->rule('lengthMax', 'full_name', 200)
->rule('ip', 'user_ip')
->rule('username', 'username');
if (!$v->validate()) {
    $errors = sp_valitron_errors($v->errors());
    flash('users-danger', $errors);
    sp_store_post($data);
    return redirect_to_current_route();
}
if (!$current_user_can('change_user_role') || !$roleModel->exists($data['role_id']))
{ $data['role_id'] = RoleModel::TYPE_USER;
}
$data['full_name'] = sp_strip_tags($data['full_name'], true);
$data['is_verified'] = 1;
if (!$id = $userModel->addUser(
    $data['email'],
    $data['password'],
    $data)) {
    flash('users-danger', UserModel::DB_ERROR);
    return redirect_to_current_route();
}
flash('users-success', ("Користувача додано"));
return redirect_to('dashboard.users');
}
public function update($id) {
    if (!$current_user_can('edit_user')) {
        sp_not_permitted();
    }
    breadcrumb_add('dashboard.users.update', ('Update User1'));
    $userModel = new UserModel;
    $user = $userModel->read($id);
    if (!$user) {
        flash('users-danger', ('Користувачі не знайдено'));
        return redirect_to('dashboard.users');
    }
    if ((int) $user['user_id'] === current_user_ID()) {
        flash('users-danger', ('Редагування заборонено'));
        return redirect_to('dashboard.users');
    }
    $roleModel = new RoleModel;
    $roleName = $roleModel->read($user['role_id'], ['role_name'])['role_name'];
    $user['role_name'] = $roleName;
    $roles = [];
    if (current_user_can('change_user_role')) { $roleModel = new RoleModel;
    $roles = $roleModel->readMany(['role_id', 'role_name'], 0, 50);
    }
    $data = [
        'rolejlist' => $roles,

```

```

'user" => $user
];
return view('admin::users/update.php', $data);
}
public function updateUser($id) {
if (!current_user_can('edit_user')) {
sp_not_permitted();
}
$userModel = new UserModel;
$roleModel = new RoleModel;
$app = app();
$req = $app->request;
$user = $userModel->read($id, ['user_id', 'username', 'email', 'role_id']);
if (!$user) {
flash('users- danger', ('К о р и с т у в а ч і в н е з н а й д е н о '));
return redirect_to('dashboard.users');
}
if ((int) $user['user_id'] === current_user_ID()) {
flash('users- danger', ('Р е д а г у в а н н я з а б о р о н е н о '));
return redirect_to('dashboard.users');
}
$data = [
'email' => $req->post('email'),
'password' => $req->post('password'),
'username' => $req->post('username'),
'role_id' => $req->post('role_id', 0),
'full_name' => $req->post('full_name'),
'user_ip' => $req->post('user_ip'),
'is_blocked' => sp_int_bool($req->post('is_blocked')),
'is_verified' => sp_int_bool($req->post('is_verified')),
];
$v = new Validator($data);
$v->labels([
'email' => ('Е л е к т р о н н а п о ш т а ' ),
'password' => _ ('П а р о л ь ' ),
'username' => _ ('К о р и с т у в а ч ' ),
'full_name' => _ ('П І Б ' ), ])->rule('required', ['email', 'user_ip'])
->rule('email', 'email')
->rule('uniqueEmail', 'email', $user['email'])
->rule('uniqueUsername', 'username', $user['username'])
->rule('lengthMin', 'password', (int) config('internal.password_minlength'))
->rule('lengthMax', 'full_name', 200)
->rule('ip', 'user_ip')
->rule('username', 'username');
if (!$v->validate()) {
$errors = sp_valitron_errors($v->errors());
flash('users- danger', $errors);
sp_store_post($data);
return redirect_to_current_route ();
}
if (current_user_can('change_user_role')) {
if (!$roleModel->exists($data['role_id'])) {
unset($data['role_id']);
}
} else {
unset($data['role_id']);
}
if (!current_user_can('change_user_status')) {
unset($data['is_blocked']);
}
if (empty($data['password'])) {
unset($data['password']);
}
}

```

```

$data['full_name'] = sp_strip_tags($data['full_name'], true);
$userModel->updateUser($id, $data);
flash('users- success', ('Д а н і   к о р и с т у в а ч а   п о н о в л е н о   '));
return redirect_to_current_route();
}
public function delete($id) {
if (!current_user_can('delete_user')) {
sp_not_permitted();
}
sp_enqueue_script('parsley', 2, ['dashboard- core- js']);
breadcrumb_add('dashboard.users.delete', ('В и д а л и т и   к о р и с т у в а ч а   '));
$userModel = new UserModel;
$user = $userModel->read($id, ['full_name', 'email', 'avatar', 'last_seen',
'user_id', 'created_at']);
if (!$user) {
flash('users- danger', ('К о р и с т у в а ч а   н е   з н а й д е н о   '));
return redirect_to('dashboard.users');
}
if ((int) $user['user_id'] === current_user_ID()) {
flash('users- danger', ("В и д а л е н н я   з а б о р о н е н о   "));
if (is_ajax()) {
return;
}
return redirect_to('dashboard.users');
}
$data = [
'user' => $user
];
return view('admin::users/delete.php', $data);
}
public function deleteUser($id) {
if (!current_user_can('delete_user')) {
sp_not_permitted();
}
$userModel = new UserModel;
$user = $userModel->read($id, ['user_id']);
if (!$user) {
flash('users- danger', ('В и д а л и т и   к о р и с т у в а ч а   '));
if (is_ajax()) {
return;
}
return redirect_to('dashboard.users');
}
if ((int) $user['user_id'] === current_user_ID()) {
flash('users- danger', ___ ("В и н е   м о ж е т е   в и д а л и т и   в л а с н и й
а к к а у н т "));
if (is_ajax()) {
return;
}
return redirect_to('dashboard.users');
}
$userModel->deleteUser($id);
flash('users- success', ('К о р и с т у в а ч а   в и д а л е н о   '));
if (is_ajax()) {
return;
}
return redirect_to('dashboard.users');
}

```

ДОДАТОК В
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

Кафедра інженерії програмного забезпечення

ДОВІДКОВО-ІНФОРМАЦІЙНА СИСТЕМА КОМП'ЮТЕРНОГО КЛУБУ

Виконав:
студент IV курсу, група ІПЗ-19-1,
Шаріфов Тимур Фарманович

Керівник:
доцент, кандидат технічних наук,
Радельчук Галина Іванівна

Актуальність теми роботи

Комп'ютерні клуби надають клієнтам доступ до комп'ютерів для різних цілей. Різновидом комп'ютерних клубів є Інтернет-клуби та Інтернет-кафе.

На основі сьогоdnішнього попиту та перспектив розвитку комп'ютерних клубів завданням компанії-власника комп'ютерного клубу є окупити свої витрати у максимально короткі терміни та досягти стабільності, а також рентабельності клубу надалі. Одним із основних способів досягнути успіху є максимальне залучення клієнтів, для чого і потрібні відповідні інформаційні системи.

Також у зв'язку з розширенням сфер послуг у комп'ютерних клубах та збільшення їх кількості зростає конкуренція між ними. Жоден комп'ютерний клуб нині не може обійтися без інформаційної підтримки, яка здійснюється у вигляді відповідних програмних продуктів, щоб бути конкурентоспроможним у сфері даних послуг.

Тому проектування за тематикою кваліфікаційної роботи є актуальним.




Мета та завдання роботи

Мета роботи: розробка довідково-інформаційної системи комп'ютерного клубу.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- а) визначити користувачів розроблюваної системи;
- б) визначити вимоги до розроблюваної системи;
- в) виконати архітектурне та детальне проектування системи;
- г) спроектувати інтерфейси користувачів;
- д) виконати проектування бази даних;
- е) на основі проектних рішень виконати програмну реалізацію системи та реалізацію бази даних;
- є) провести емпіричне дослідження розробленої системи на працездатність, відповідність вимогам та функціональну придатність.



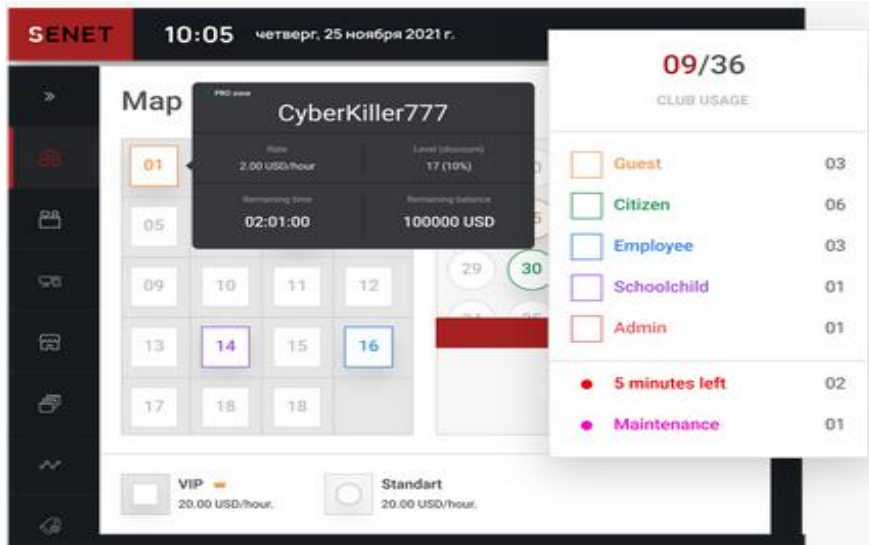
Аналіз наявного програмного забезпечення галузі

Існує досить широкий спектр програм, які використовуються у комп'ютерних клубах:

- Internet Cafe Software 10.1.0;
- SENET;
- Locker 5.86;
- Network Administrator 6.7.7;
- Smart Club Keeper 2.50;
- GameAdmin 1.0.0.9;
- APTimer 1.2 та ін.

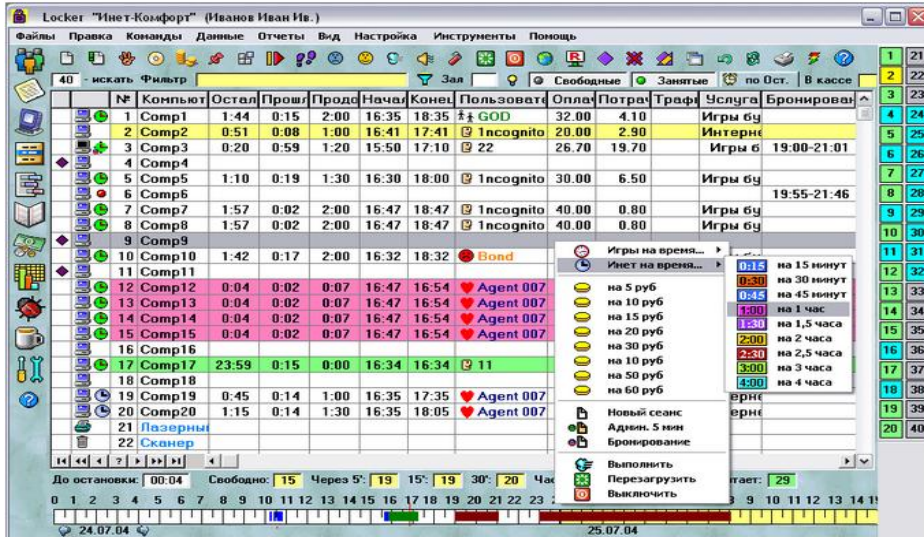
Система SENET

Одне з вікон SENET: керування комп'ютерами через хмару



Система Locker

Одне з вікон програми Locker



Визначення вимог до системи

Що ж повинно бути в нашій довідково-інформаційній системі?

- при входженні в систему відвідувач повинен зареєструватися та створити обліковий запис;
- в системі користувачі можуть поповнювати свій баланс, вибирати тарифи і абонементи, переглядати та запускати різні ігри, перевіряти новини клубу тощо.

Для управління комп'ютерним клубом довідково-інформаційна система має включати в себе наступні можливості:

- інтерфейс для клієнта і адміністратора,
- карту управління ПК,
- модуль резервування ПК,
- цінні пропозиції та програму лояльності;
- модуль «Магазин» та онлайн-каталог товарів;
- створення замовлень;
- систему повідомлень і статуси відстеження замовлень;
- вбудовану касову систему.

Проектування системи - структура

Довідково-інформаційна система розбита на два блоки: **блок адміністратора** і **блок користувача**.

Блок адміністратора включає такі вкладки:

Адміністрування послуг;

Користувачі;

Керування класом;

Послуги;

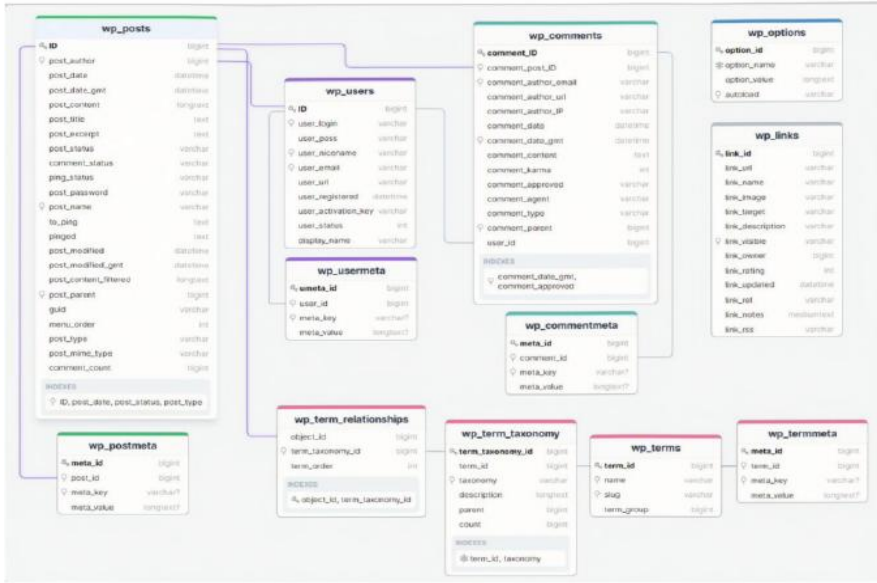
Додаткові налаштування.

У вкладці **Адміністрування послуг** знаходяться такі розділи: **Черга** (де Адміністратор бачить послідовність замовлень та їх виконання), та **Створити** (де формується саме замовлення). Вкладка **Користувачі** розбита на **Переглянути** (де можна переглянути всіх, хто працює в даний час у клубі, їх електронну пошту, ім'я, по-батькові, надіслати їх повідомлення, побачити загальну суму по послугах, видалити користувача) і **Додати** (де додаються послуги та користувачі).

Блок Користувача призначений для відвідувачів клубу.

Проектування системи – база даних

Для довідково-інформаційної системи було створено базу даних:



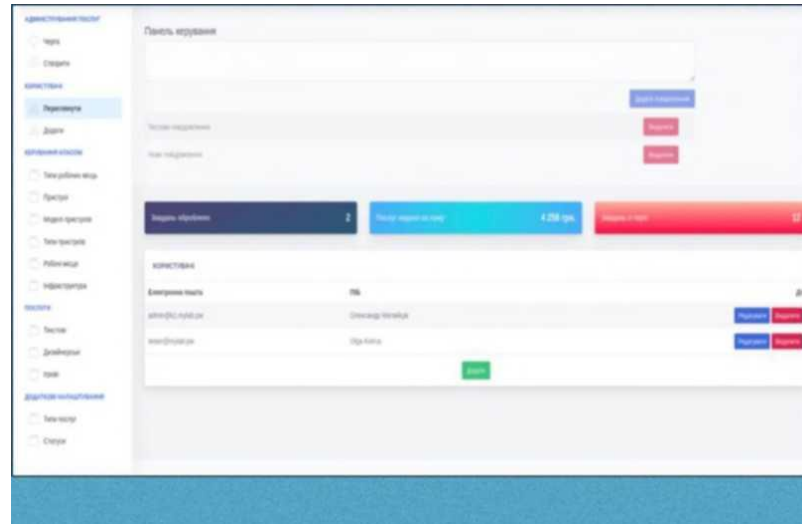
Проектування системи –діаграми UML



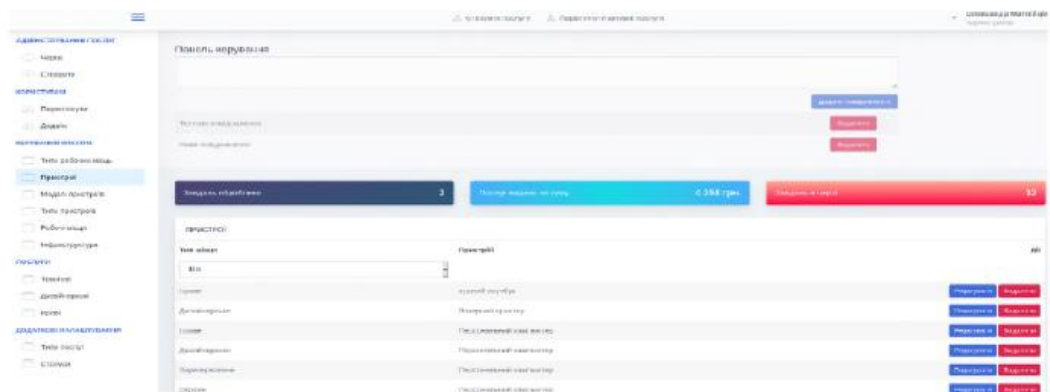
Діаграма послідовності "Авторизація"

Програмна реалізація – блок адміністратора

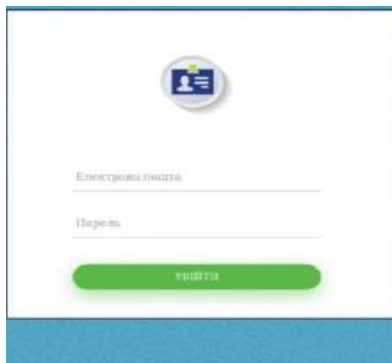
В доступі для адміністратора відкривається вікно з такими можливостями як адміністрування, користувачі, керування класом, послуги, додаткові налаштування. Видно завдання до обробки, рахунок за послуги, кількість завдань в черзі та ін.



Програмна реалізація – сторінка “Пристрої”

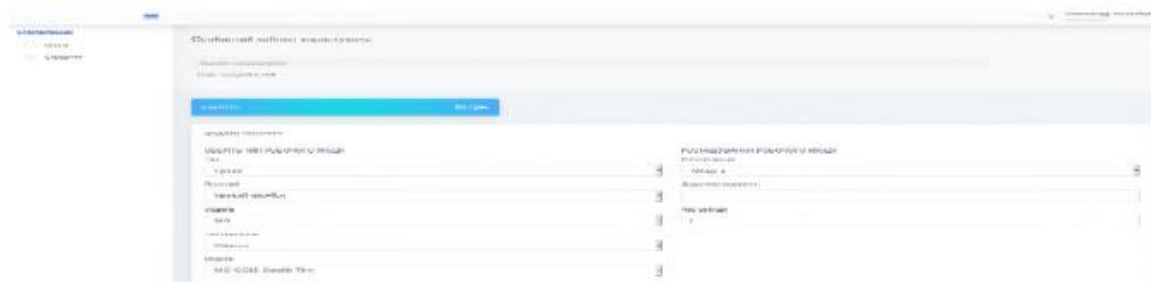


Програмна реалізація – авторизація



The image shows a login interface. At the top center is a circular icon containing a person silhouette and a document. Below the icon are two input fields: the first is labeled 'Електронна пошта' (Email) and the second is labeled 'Пароль' (Password). Below these fields is a prominent green button with the text 'Увійти' (Login) in white. The entire form is set against a white background with a blue footer bar.

Програмна реалізація – особистий кабінет користувача




The image displays a user profile page titled 'Особистий кабінет користувача' (User Profile). The page is divided into two main sections. The left section, titled 'Особисті дані' (Personal Data), includes fields for 'Ім'я' (Name), 'Прізвище' (Surname), 'Повне ім'я' (Full Name), 'Стать' (Gender), 'Дата народження' (Date of Birth), 'Місце народження' (Place of Birth), and 'МАС-CDM Email To:'. The right section, titled 'Контактні дані користувача' (User Contact Data), includes fields for 'Мобільний номер' (Mobile Number), 'Електронна пошта' (Email), and 'Телефон' (Phone). The page features a navigation menu on the left and a search bar at the top.



Висновки

- В результаті виконання кваліфікаційної роботи була спроектована та програмно реалізована довідково-інформаційна система комп'ютерного клубу.
- У першому розділі роботи здійснено дослідження предметної області та визначення вимог для розроблюваної системи.
- У другому розділі проведено проектування системи, зокрема її структура, база даних та інтерфейс користувача.
- У третьому розділі здійснена програмна реалізація довідково-інформаційної системи. Для програмної реалізації системи використані наступні засоби: мова програмування PHP, сервер бази даних MySQL та таблиці стилів CSS.
- У четвертому розділі було проведено тестування створеного програмного засобу в можливому діапазоні.

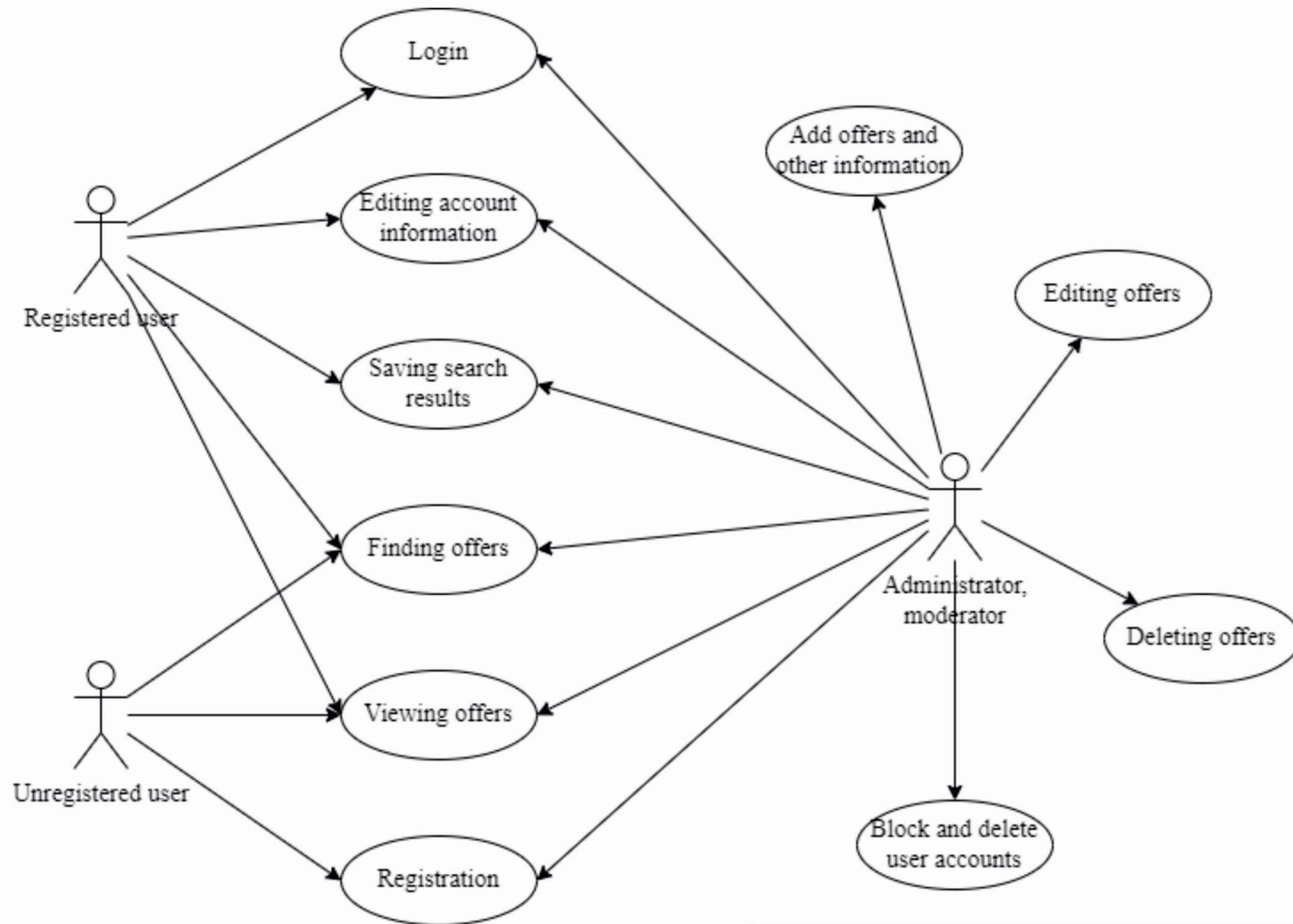


**Доповідь закінчена.
Дякую за увагу!**

ГРАФІЧНА ЧАСТИНА

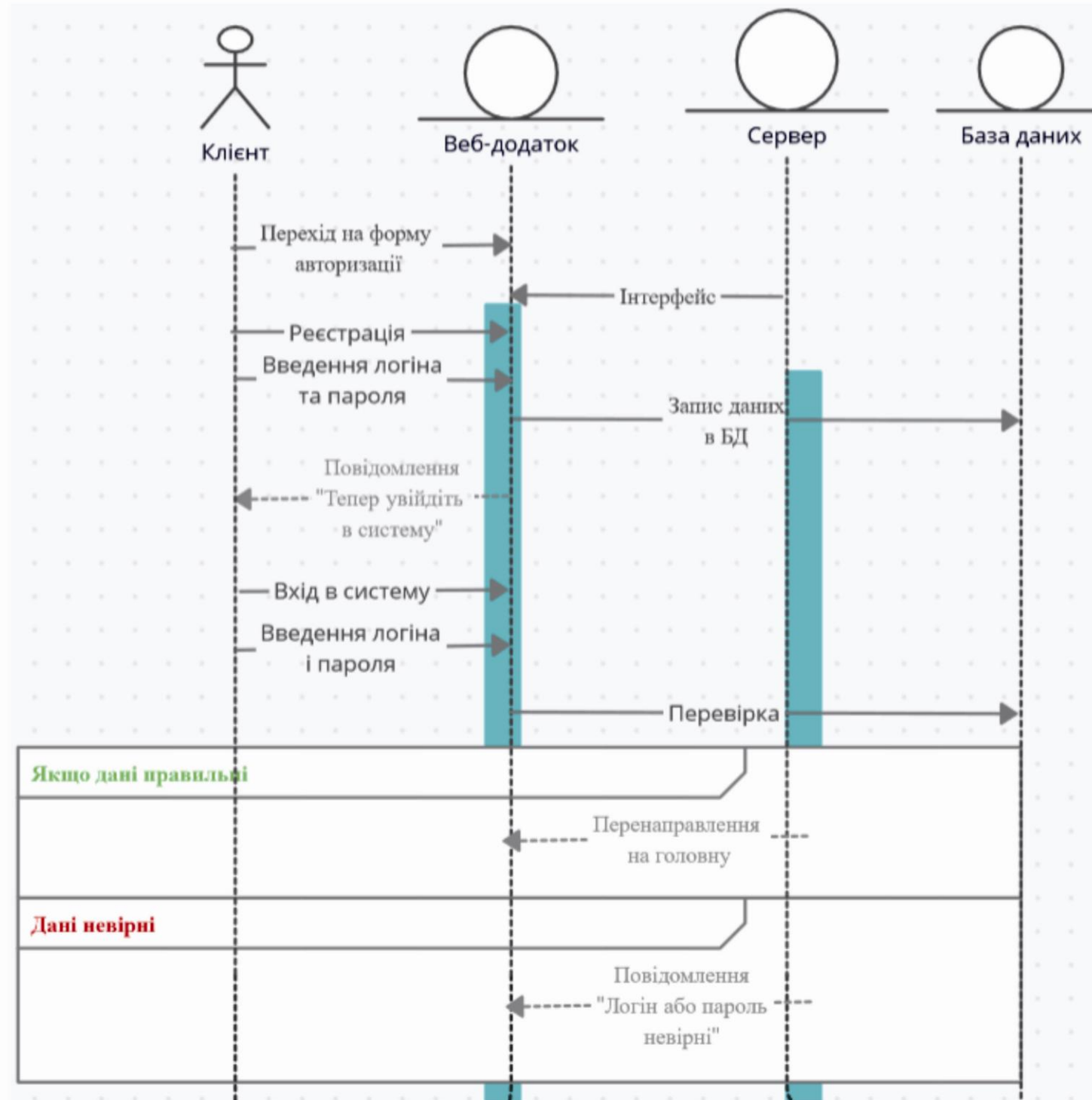
КВРІПЗ.190143.01.21.Е8

Діаграма варіантів використання



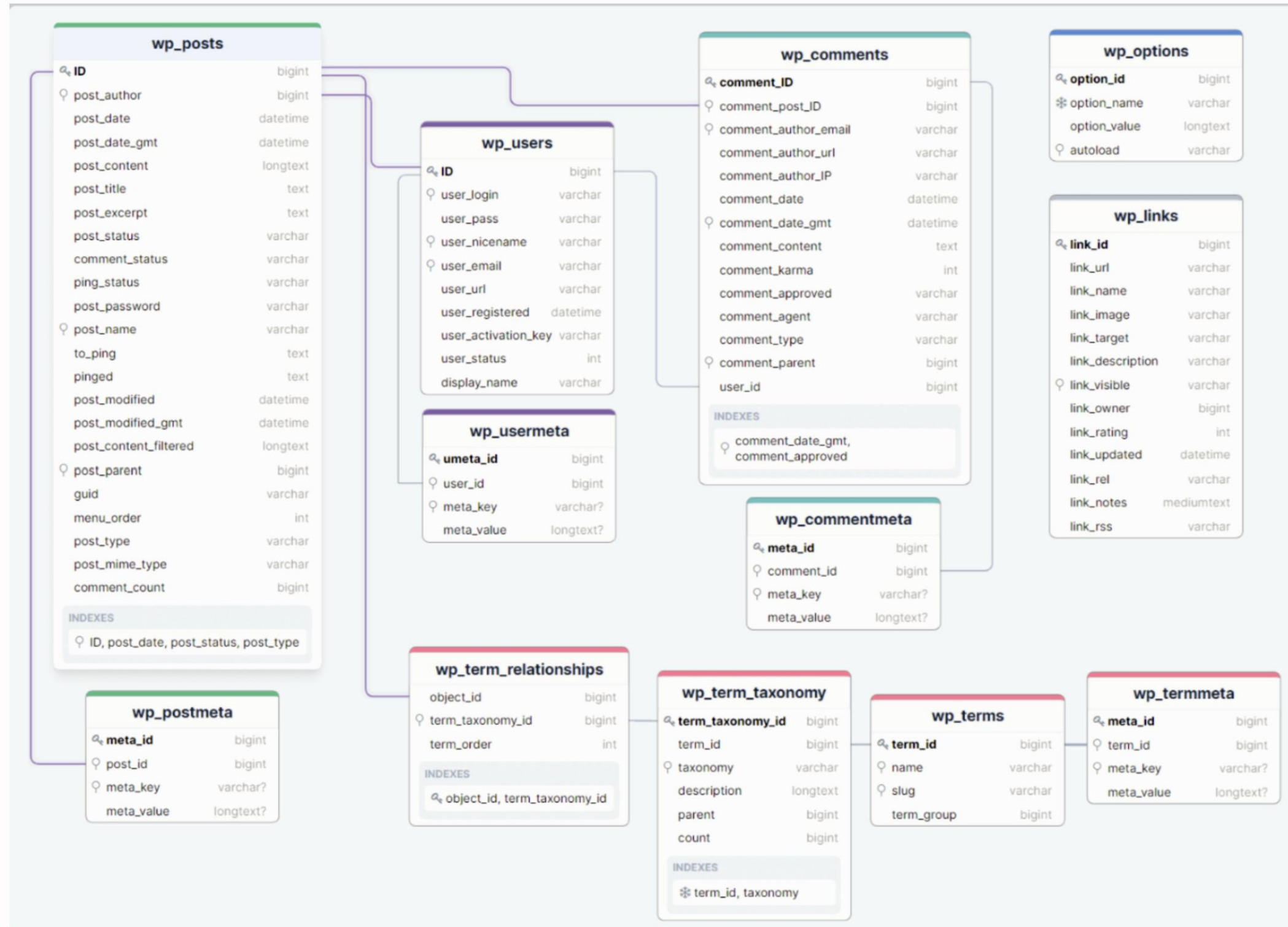
					КВРІПЗ.190143.01.21.Е8		
Зм.	Арк.	№ докум.	Підпис	Дата	Довідково-інформаційна система комп'ютерного клубу.		
Розробив		Шаріфов Т.Ф.			Літера	Маса	Масштаб
Керівник		Радельчук Г.І.			Н		
Консульт.					Аркуш 1	Аркушів 3	
Н. Контр.		Радельчук Г.І.			ХНУ, ІПЗ-19-1		
Зав. каф.		Бедратюк Л.П.					

Діаграма послідовності реєстрації клієнта клубу



					КВРІПЗ.190143.01.21.Е8		
					Довідково-інформаційна система комп'ютерного клубу. Діаграма послідовності реєстрації клієнта клубу		
Зм.	Арк.	№ докум.	Підпис	Дата	Н		
Розробив		Шаріфов Т.Ф.					
Керівник		Радельчук Г.І.					
Консульт.					Аркуш 1	Аркушів 3	
Н. Контр.		Радельчук Г.І.			ХНУ, ІПЗ-19-1		
Зав. каф.		Бедратюк Л.П.					

Модель бази даних системи



					КВРІПЗ.190143.01.21.Е8		
					Довідково-інформаційна система комп'ютерного клубу.		
					Модель бази даних системи		
					ХНУ, ІПЗ-19-1		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив		Шаріфов Т.Ф.					
Керівник		Радельчук Г.І.					
Консульт.							
Н. Контр.		Радельчук Г.І.					
Зав. каф.		Бедратюк Л.П.					
					Літера	Маса	Масштаб
					Н		
					Аркуш 3	Аркушів 3	

СУПРОВІДНІ ДОКУМЕНТИ

Завідувачу кафедри інженерії програмного
забезпечення проф. Бедратюку Л. П.

здобувача вищої освіти

Шаріфова Т. Ф.

Прізвище, ініціали

факультет ІТ, 4 курс, група ПЗ-19-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності в Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та/або Anti-Plagiarism) і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

15.06.2023 р.

дата


підпис

Sat Jun 17 16:20:42 EEST 2023, Хіврич Володимир Русланович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 5.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 19%

ID: 116761 Назва: БКР Довідково-інформаційна система комп'ютерного клубу Додано в БД: 2023-06-17 Автора: Шаріфов Т.Ф. Керівники: Радельчук Г.І. к.т.н. доц. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	73417	635	4544 (6%)	64 (10%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми



Ім'я користувача:
Кафедра ІПЗ

ID перевірки:
1015633556

Дата перевірки:
17.06.2023 17:18:48 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
17.06.2023 17:21:24 EEST

ID користувача:
100005589

Назва документа: КвР_Шаріфов_на_плагіат

Кількість сторінок: 66 Кількість слів: 11310 Кількість символів: 85267 Розмір файлу: 9.54 MB ID файлу: 1015280040

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

16.4%
Схожість

Найбільша схожість: 7.72% з джерелом з Бібліотеки (ID файлу: 1008327152)

8.05% Джерела з Інтернету 594

Сторінка 68

13.4% Джерела з Бібліотеки 183

Сторінка 75

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 5

Підозріле форматування 22 сторінки

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатами звіту/звітів подібності щодо роботи, продуктованими програмно-технічним засобом (ами) перевірки текстів на плагіат:

Назва: «Довідково-інформаційна система комп'ютерного клубу»

Автор: Шаріфов Тимур Фарманович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Радельчук Галина Іванівна, кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені у роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системою перевірки на плагіат Unichesk виявлено схожість з деякими документами у частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів, у рамках основних написів, у назвах публікацій переліку джерел посилання;

2) усі запозичення, виявлені у тексті роботи, є фрагментарними або мають належним чином оформленні посилання;

3) виявлені модифікації тексту не впливають на відсоток схожості.

Максимальний обсяг запозичень, визначений системою Anti-Plagiarism, складає 5.0%. Обсяг запозичень, визначений системою Unichesk виявлення збігів ідентичності/схожості, складає 16.4% і адресується до 594 джерел з інтернету і 183 джерел з бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 19.06.2023 р.

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи

Леонід БЕДРАТЮК

Леонід БЕДРАТЮК

Галина РАДЕЛЬЧУК

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
освітнього ступеня «бакалавр»Дипломник Шаріфов Тимур ФармановичТема Довідково-інформаційна система комп'ютерного клубуСпеціальність 121 – Інженерія програмного забезпечення**Обсяг кваліфікаційної роботи:**Кількість листів креслень 3; кількість сторінок записки 65

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі досліджено і проаналізовано предметну область, визначено функціональні та нефункціональні вимоги до довідково-інформаційної системи Був проведений аналіз існуючих систем на ринку, розглянуто їх переваги і недоліки, та доведено актуальність розробки нової системи. Спроектвані нові рішення, розглянуто інструменти для реалізації спроектованих рішень, в результаті чого створена довідково-інформаційна система комп'ютерного клубу. Також проведено тестування системи, за результатами якого доведено, що розроблена система працює коректно.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням вимог стандартів

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі доведено актуальність теми, визначено мету та завдання дипломного проектування. У першому розділі проведено аналіз предметної області, розглянуто існуючі рішення та визначені вимоги до розроблюваної довідково-інформаційної системи комп'ютерного клубу, виконана постановка задачі. У другому розділі спроектовано структуру системи, проведено аналіз типів сучасних баз даних та розроблена реляційна база даних системи, а також проведено аналіз та вибір технологій і засобів реалізації системи. У третьому розділі описана реалізація системи та її тестування.

4. Позитивні сторони роботи Тематика кваліфікаційної роботи є актуальною. Також для проектування та реалізації довідково-інформаційної системи застосовано актуальні рішення та сучасні технології

5. Негативні сторони роботи Текст пояснювальної записки переобтяжений загальновідомим описовим матеріалом.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення виконано відповідно до теми кваліфікаційної роботи та подано у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Матеріал пояснювальної записки структурований, послідовний та простий, що дозволяє легко зрозуміти викладений матеріал у рамках тематики проектування. Графічний матеріал унаочнює бачення деталей проектування системи.

8. Інші зауваження _____

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана на достатньому рівні, відповідає поставленій задачі та заслуговує оцінки «задовільно».

РЕЦЕНЗЕНТ

Лисенко Сергій Миколайович, професор
кафедри комп'ютерної інженерії та
інформаційних систем ХНУ

19.06.2023 р.


(Підпис)