

Хмельницький національний університет  
Факультет програмування  
та комп'ютерних і телекомунікаційних систем  
Кафедра інженерії програмного забезпечення

## ДИПЛОМНИЙ ПРОЕКТ

Довідково-інформаційна система

Назва теми

туристичної фірми

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного  
забезпечення»

Шифр ДППЗ.170122.01.18.ПЗ

Виконав студент IV курсу група ПЗ-17-1

Підпис

Р. В. Федорук

Ініціали, прізвище

Керівник канд. пед. наук, доцент

Науковий ступінь, звання

Підпис

Н. І. Праворська

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

8 червня 2021 р.

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Програмування та комп'ютерних і телекомунікаційних систем

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

05 02 2021 р.

## ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Федоруку Роману Віталійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Довідково-інформаційна система туристичної фірми

Керівник проекту (роботи) Праворська Наталія Іванівна, канд. пед. наук, доцент

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

Затверджена наказом ректора університету від 05.02.2021 р. № 11

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2021 р.

3. Вихідні дані до проекту (роботи) Матеріали переддипломної практики



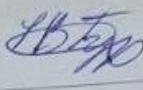
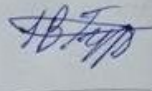
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження предметної області та постановка задачі, проектування програмного забезпечення, програмна реалізація, тестування програмного забезпечення

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

Презентаційні матеріали (слайди)

6. Консультанти розділів дипломного проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання при
Нормоконтроль	Радельчук Г. І., доцент кафедри ІПЗ		
Антиплагіат	Гурман І. В., доцент кафедри ІПЗ		

7. Дата видачі завдання « 05 » лютого 2021р.

КАЛЕНДАРНИЙ ПЛАН

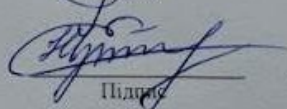
Назва етапів (розділів) дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Приміт
1 Ознайомлення з тематикою дипломного проектування (ДП), визначення та узгодження індивідуальних там ДП	01.12 – 30.12.2020	
2 Дослідження предметної області, в якій планується використання програмного засобу (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2021	
3 Проектування програмного забезпечення	01.02 – 28.02.2021	
4 Програмна реалізація	01.03 – 10.04.2021	
5 Тестування програмного забезпечення	11.04 – 30.04.2021	
6 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки ДП згідно вимог стандартів	01.05 – 25.05.2021	
7 Попередній захист ДП	Травень 2021 (згідно графіка)	
8 Перевірка ДП на плагіат, нормконтроль, отримання відгуків та рецензій. Брошування (зшиття) пояснювальної записки	26.05 – 30.05.2021	
9 Підготовка до захисту та захист ДП	з 01.06.2021	

Студент

  
Підпис

Р. В. Федорук  
Ініціали, прізвище

Керівник проекту (роботи)

  
Підпис

Н. І. Праворська  
Ініціали, прізвище

## АНОТАЦІЯ

Тема дипломного проекту: Довідково-інформаційна система туристичної фірми.

Автор проекту: Федорук Роман Віталійович.

Керівник проекту: Праворська Наталія Іванівна.

Пояснювальна записка: 69 с., 36 рис., 15 табл., 3 дод., 12 джерел.

Графічна частина: 11 слайдів.

ДОВІДКОВО-ІНФОРМАЦІЙНА СИСТЕМА, ТУРИСТИЧНА ФІРМА, ВЕБ-ДОДАТОК, PHP, MY SQL, MY TOUR.

Метою проекту є розробка довідково-інформаційної системи туристичної фірми для автоматизації діяльності малих та середніх підприємств туристичної галузі.

У дипломному проекті проаналізована робота туристичної галузі, детально вивчені та дослідженні основні бізнес-процеси туристичної фірми, проаналізовано наявне на ринку програмна забезпечення для туристичних фірм, сформовані функціональні та нефункціональні вимоги до довідково-інформаційної системи. Спроектвана та розроблена структура програмного забезпечення, логічна та фізична моделі бази даних, визначені програмні модулі системи, здійснена програмна реалізація. Проведене тестування розробленої довідково-інформаційної системи.

Для реалізації довідково-інформаційної системи використано мову програмування PHP; СКБД MySQL; фреймворк Bootstrap; бібліотеку JQuery; каскадні таблиці стилів CSS3; мову гіпертекстової розмітки HTML 5; шаблонізатор Smarty.

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

01.06.2024р.  
Дата

  
Підпис

## ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	ДППЗ.170122.01.18.ВД	Пояснювальна записка	69		
2	A4		Завдання на дипломний проект	1		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A4		Презентаційні матеріали	11		

ДППЗ.170122.01.18.ВД								
Змн.	Арк.	№ докум.	Підпис	Дата	Довідково-інформаційна система туристичної фірми. Відомість документів	Літ.	Арк.	Аркуше
Виконав		Федорук Р.В.		07.06			1	1
Керівник		Праворська Н.І.		08.06				
Н. Контр.		Радельчук Г.І.		08.06				
Зав. Каф.		Бедратюк Л.П.		08.06				ХНУ, ПЗ-17-1

## ЗМІСТ

Вступ.....	5
1 Дослідження предметної області та постановка задачі.....	7
1.1 Змістовний аналіз та опис предметної області .....	7
1.2 Аналіз наявного програмно-технічного забезпечення предметної області.....	12
1.3 Визначення вимог до довідково-інформаційної системи та розробка технічного завдання .....	19
2 Проектування довідково-інформаційної системи туристичної фірми .....	21
2.1 Проектування структури довідково-інформаційної системи.....	21
2.2 Розробка структури довідково-інформаційної системи .....	25
2.3 Проектування логічної моделі бази даних .....	33
3 Програмна реалізація .....	44
3.1 Реалізація бази даних.....	44
3.2 Реалізація модулів системи.....	50
3.3 Інструкція користувача.....	53
3.4 Вимоги до програмно-апаратних засобів .....	61
4 Налагодження і тестування системи .....	63
4.1 Вибір та обґрунтування методів тестування системи.....	63
4.2 Валідація та верифікація системи .....	65
4.3 Аналіз результатів тестування системи.....	68
Висновки.....	70
Перелік джерел посилання .....	71
Додаток А Технічне завдання .....	73
Додаток Б Програмний код .....	78
Додаток В Презентаційні матеріали.....	100

					ДПШЗ.170122.01.18.ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Довідково-інформаційна система туристичної фірми.	Літ.	Арк.	Акрушів
Виконав		Федорук Р.В.	<i>[Підпис]</i>	01.06			4	113
Керівник		Праворська Н.І.	<i>[Підпис]</i>	08.06	Пояснювальна записка	ХНУ, ІПЗ-17-1		
Н. Контр.		Радельчук Г.І.	<i>[Підпис]</i>	08.06				
Зав. Каф.		Бедратюк Л.П.	<i>[Підпис]</i>	08.06				

## ВСТУП

В наш час поняття діджиталізації і цифровізації так швидко увійшли до навколишнього світу, що уявити процес надання послуг без застосування гаджетів не можливо. Тим паче, в такій висококонкурентній галузі як туризм. Також на звичний порядок надання послуг вплинула пандемія коронавірусу, і це ще одна причина з якої власники туристичних фірм мають вдосконалювати процес надання послуг в туристичній галузі, якщо бажають зберегти свій бізнес. В цьому випадку на допомогу приходять інформаційні технології за допомогою яких здійснюється залучення та втримання клієнтів.

Туристична галузь – це галузь, яка найбільше постраждала під час пандемії коронавірусу. Тому актуальним є питання вдосконалення та розробки програмного забезпечення, яке б відповідало викликам часу та надавало конкурентну перевагу за рахунок зручності використання та потрібної функціональності.

Метою дипломного проектування є розробка довідково-інформаційної системи туристичної фірми.

Об'єктом дослідження є довідково-інформаційні системи в галузі туристичної діяльності.

Предметом дослідження є моделі та методи створення програмного забезпечення у вигляді веб-додатку.

Для досягнення поставленої мети були сформовані завдання на дипломне проектування:

- дослідити предметну область туристичного бізнесу та виявити потреби потенційних користувачів програмного продукту;
- проаналізувати наявне програмне забезпечення в цій галузі та сформулювати вимоги на розробку нового програмного забезпечення, що відповідає потребам часу;
- спроектувати структуру довідково-інформаційної системи та розробити

										ДППЗ.170122.01.18.ПЗ	Арк.
											5
Змн.	Арк.	№ докум.	Підпис	Дата							

базу даних;

- вибрати технології для розробки довідково-інформаційної системи;
- розробити та протестувати інформаційно-довідкову систему туристичної фірми.

Розроблюване програмне забезпечення буде корисним для малих та середніх туристичних фірм, які прагнуть автоматизувати бізнес-процеси і перевести взаємодію з клієнтами на новий рівень, використовуючи можливості інформаційних технологій.

					ДППЗ.170122.01.18.ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

# 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Змістовний аналіз та опис предметної області

Темою дипломного проекту є розробка довідково-інформаційної системи туристичної фірми.

Туристичний ринок України представлений фірмами туроператорами і фірмами турагентами.

Виробляють туристичний продукт фірми туроператори. Саме вони через розробку маршрутів формують тури, займаються придбанням туристичних послуг у їх безпосередніх виробників, таких як готельні комплекси, транспортні організації, заклади культури, розваг, організації дозвілля тощо, і збирають їх у пакет, що буде запропонований клієнту – потенційному туристу. Також фірма туроператор займається забезпеченням реалізації туру, а також здійсненням самої подорожі та організацією обслуговування туристів на вибраному маршруті [1].

Найчастіше фірма туроператор виступає оптовим продавцем, який реалізує свій продукт або самостійно, або через торговельних посередників, тобто турагентів.

Фірма туроператор розробляє і пропонує власний туристичний продукт. Зазвичай, це велика фірма, яка має у власності матеріальну базу чи укладає договори з готелями, фірмами перевізниками та іншими організаціями для обслуговування туристів, та реалізує запропонований продукт самостійно або за допомогою фірм турагентів.

На відміну від фірми туроператора, фірми турагенти не займаються виробленням власного туристичного продукту, а виступають у якості посередників, що отримують прибуток через комісійні від продажу або мають за контрактом пільгові умови на продаж туристичного продукту з фірмою туроператором [1].

					ДПШЗ.170122.01.18.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Підставою для продажу турів у фірми турагента є агентська угода, тобто договір укладений між фірмою туроператором і фірмою турагенством. На підставі договору туроператор доручає турагенству реалізацію своїх турів, а саме: він надає йому певну їх кількість під реалізацію або дозволяє користуватися власною офертою (каталогом туристичної пропозиції), доставляє рекламно-інформаційні матеріали, своєчасно попереджає про зміни, а також забезпечує виконання програми туру. У разі виникнення форс-мажорних обставин, що призводять до неможливості здійснення туру туроператор гарантує турагенту повернення вартості туру. Турагент забезпечує:

- рекламу турів туроператора;
- реалізацію турів;
- відправку клієнтів у місце з якого починається подорож;
- своєчасність перерахування грошей за тур туроператору;
- контроль якості обслуговування через збір зауважень та побажань клієнтів системи.

Займатися оформленням туристичної документації та страхуванням туристів можуть як туроператор, так і турагент. Це визначається домовленістю між ними.

Туристична фірма агент працює за агентську винагороду, яка або закладена туроператором як комісійний процент у ціну туру, або формується за рахунок знижки від роздрібною ціни туру. Іноді, фірма туроператор дозволяє робити фірмі турагенту власні нарахування на ціну туру. Але цей підхід застосовується не часто, внаслідок високої конкуренції в галузі і змагання за лояльність клієнта.

Одна й та сама фірма може виступати на туристичному ринку як у ролі туроператора, так і в ролі турагента. Згідно українського законодавства, термін «туроператор» може бути застосований тільки до юридичних осіб, а термін «турагент» і до юридичних, і до фізичних осіб-підприємців.

										ДППЗ.170122.01.18.ПЗ	Арк.
											8
Змн.	Арк.	№ докум.	Підпис	Дата							

Фірми туроператори поділяються на операторів масового ринку та спеціалізовані фірми. Оператори масового ринку, надають послуги широкому колу клієнтів. Спеціалізовані фірми, зосереджені на вибраному сегменті туристичного ринку, пропонують подорожі певного типу або зорієнтовані на певні верстви населення (круїзні компанії, VIP подорожі, дитячі групи тощо). За напрямом діяльності існують:

- фірми, що направляють туристів (ініціативні);
- фірми, що приймають і обслуговують туристів (рецептивні);
- змішані фірми комплексної дії [2].

Саме це визначає організаційну структуру фірми. Ініціативні фірми зосереджуються на комерційній роботі на туристичному ринку, аквізиції туристів. Рецептивні фірми займаються організацією туристичного обслуговування. Відповідно до цього в структурі туристичних фірм визначаються бізнес-процеси та ділянки роботи [2].

Сучасна теорія та практика управління значну увагу приділяє процесу та методам моделювання (опису) бізнес-процесів. Моделювання бізнес-процесів є важливою складовою проектів реінженірінгу бізнес-процесів і створення масштабних систем програмного забезпечення [3].

В моделюванні бізнес-процесів використовується кілька методів, основою яких є структурний та об'єктно-орієнтований підхід.

Діяльність будь-якої туристичної фірми є сукупністю бізнес-процесів різного рівня. Мета діяльності і турагента, і туроператора – це задоволення потреб клієнтів у послугах туристичного, рекреаційного, екскурсійного, анімаційного характеру. Отже, бізнес-процес туристичної фірми – це сукупність специфічних операцій (робіт), що трансформує вхідні ресурси у пропозицію для клієнта для задоволення його потреб.

До основних бізнес-процесів туристичної фірми відносять ті, що спрямовані на створення туристичного продукту. Процес створення туристичного продукту та підготовка до реалізації потребує значних ресурсів

									Арк.
									9
Змн.	Арк.	№ докум.	Підпис	Дата					

(трудо­вих, ма­те­рі­аль­них, фінан­со­вих, ін­фор­ма­цій­них то­що).

Спе­ци­фі­чність бі­з­нес-про­це­су, спря­мо­ва­но­го на ре­алі­за­цію ту­ри­стич­но­го про­дук­ту, по­ля­гає в по­ді­лі ту­ри­стич­них під­при­ємств на фір­ми ту­ро­пе­ра­то­ри і фір­ми ту­ра­ген­ти.

Згідно з ри­сун­ком 1.1 ви­дно, що ту­ро­пе­ра­то­ри – це по­ста­чаль­ни­ки ту­ри­стич­них про­дук­тів для ту­ри­стич­них аген­тів. Ту­ра­ген­ти – це по­се­ред­ни­ки між ту­ро­пе­ра­то­ра­ми та по­тен­цій­ни­ми клієн­та­ми ту­ри­стич­но­го про­дук­ту. По­трі­бно за­зна­чи­ти, що фір­ми ту­ра­ген­ти це по­вно­цін­ні під­при­ємства, які скла­да­ю­ться із взає­мо­за­леж­них бі­з­нес-про­це­сів, що за­без­пе­чують зла­год­же­ну ро­бо­ту цьо­го ти­пу під­при­ємств.



Рисунок 1.1 – Реалізація основних бізнес-процесів туристичних фірм [3].

Розглянемо як відбувається типовий бізнес-процес туристичної фірми – «Реалізація туристичного продукту» (рис.1.2) [3].

Позначення на рисунку:

– ПР – позитивний результат;



## 1.2 Аналіз наявного програмно-технічного забезпечення предметної області

На ринку України працює багато туристичних фірм. Відповідно існує програмне забезпечення для автоматизації роботи подібних фірм. Зазвичай, для автоматизації роботи туристичних фірм використовують CRM системи.

CRM-система (Customer Relationship Management або управління відносинами з клієнтами) – це прикладне програмне забезпечення для організацій, призначене для автоматизації стратегій взаємодії з замовниками (клієнтами), зокрема, для підвищення рівня продажів, оптимізації маркетингу і поліпшення обслуговування клієнтів шляхом збереження інформації про клієнтів і історію взаємин з ними, встановлення і поліпшення бізнес-процесів і подальшого аналізу результатів.

Фактично, CRM-системою можна вважати будь-який варіант контролю і обліку, який допоможе поліпшити взаємодію з клієнтами. Навіть якщо ви ведете історію дзвінків і контактів на папері або в Excel – це можна вважати CRM-системою в тому випадку, якщо розроблена схема обліку і контролю працює і дозволяє контролювати всі варіанти взаємодії з клієнтами. Звичайно, такі методи ведення обліку відходять у минуле, адже в сучасному світі без ефективної автоматизації складно уявити собі роботу будь-якого бізнесу. А тому, коли говорять про CRM-системи, зазвичай мають на увазі спеціальне програмне забезпечення.

Розробники програмного забезпечення, зазвичай, нав'язують своє бачення того, що можна назвати CRM-системою. В більшості випадків, запропонований ними перелік необхідних функцій справжньої CRM збігається з тим варіантом, який вони реалізували в своєму програмному продукті. У деяких випадках подібний маркетинг працює досить явно і нав'язливо, за принципом: CRM повинна бути тільки такою, яку ми Вам пропонуємо. Інші розробники (переважно закордонні компанії) роблять це менш явно, просто демонструючи

						ДППЗ.170122.01.18.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			12

свій продукт та стверджуючи, що це і є CRM система.

Необхідно розуміти що не існує загальних стандартів і чіткого розуміння того, що розуміти під терміном CRM-система. Це може бути будь-яка зручна для бізнесу система управління взаємовідносинами з клієнтами.

Наприклад, для когось CRM – це, перш за все, облік контактів і взаємодії з клієнтами, інші можливості він вважає необов'язковими, оскільки майже не використовує їх. А багато розробників потужних CRM-платформ будуть вважати, що поняття CRM – набагато ширше, з їх точки зору навіть CRM-системи багатьох банків виявляться недостатньо функціональними, щоб називатися справжніми CRM. Насправді, тут все залежить від цілей і масштабів.

Отже, підсумовуючи: CRM-система – це будь-яке програмне забезпечення, яке допомагає бізнесу успішно контролювати роботу з клієнтами, упорядкувати і планувати її.

Визначимо для чого власне потрібна CRM-система.

Щоб не втрачати потенційного клієнта, не пропустити жодного запиту. У туристичному бізнесі у нашій країні високий рівень конкуренції. Туристичні фірми докладають значних зусиль для залучення клієнта, привернення його уваги. У порівнянні з іншими витратами на залучення клієнтів виділяється значний бюджет. І дуже важливо, щоб всі ці кошти і зусилля не пропали даром. CRM-системи дозволяють отримати впевненість, що саме так і будуть працювати співробітники туристичної фірми, що займаються продажами. А власник бізнесу, в свою чергу, отримає впевненість щодо фіксації кожного запиту, кожного ліда.

Контроль роботи співробітників і стандартизація роботи з клієнтами. Без загальної стандартизованої CRM-системи кожен співробітник працює так, як він звик. Одні ведуть облік в електронних таблицях, інші – в записнику або щоденнику, хтось не веде облік взагалі, орієнтується виключно на звіти з ІС або на власну пам'ять. Контакти також відбуваються досить хаотично. Листи клієнтам можуть відправлятися як з корпоративного, так і з особистої поштової

										ДППЗ.170122.01.18.ПЗ	Арк.
											13
Змн.	Арк.	№ докум.	Підпис	Дата							

скриньки, дзвінки відбуватися з будь-якого зручного телефону, контроль якості роботи неможливий. CRM-система може вирішити цю проблему і стандартизує роботу з клієнтами.

Накопичується статистична база, що також дуже важливо для успішного розвитку будь-якого бізнесу. Завдяки використанню CRM-системи вся робоча інформація збирається в одній загальній базі в стандартизованому вигляді. В результаті керівник може аналізувати статистику роботи, складати різні звіти (багато з яких вже в готовому вигляді присутні в CRM-системах), тобто аналізувати роботу і планувати подальшу роботу більш усвідомлено.

Готові рішення, від яких можна відштовхуватися в побудові власної системи роботи. Кожна CRM-система – це втілення бачення розробників того, як потрібно працювати з клієнтом. У ній закладено велику кількість готових інструментів, які дозволяють перевести роботу на якісно новий рівень. У малому і середньому бізнесі роботу з клієнтами направляє найчастіше безпосередньо керівник (власник) бізнесу. У нього немає експертів, а часто немає і напрацювань по організації роботи з клієнтами. Керівнику нема на що спиратися в цьому питанні, а тому продажі товарів чи послуг часто здійснюються далеко не кращим чином. Впровадження CRM-системи дозволяє отримати не тільки інструмент, а й допомогу, погляд розробників на те, як повинен працювати відділ продажів. У свою чергу при розробці CRM-системи, розробники, зазвичай, спираються на кращі практики, на експертів у питаннях роботи з клієнтами. Тому якщо активно використовуються надані CRM-системою інструменти, то і робота відділу продажів також буде оптимізуватися. Різні інструменти системи самі підказують, які кроки варто зробити в процесі оптимізації роботи з клієнтами .

Проаналізуємо наявне програмне забезпечення, що пропонується для використання туристичним фірмам на Україні.

Програмне забезпечення MAGERP CRM (<http://crm-magerp.com/crm-for-tourism>). Вікно веб-сайту, що пропонує дане програмне забезпечення

									ДППЗ.170122.01.18.ПЗ	Арк.
										14
Змн.	Арк.	№ докум.	Підпис	Дата						

представлене на рисунку 1.3.

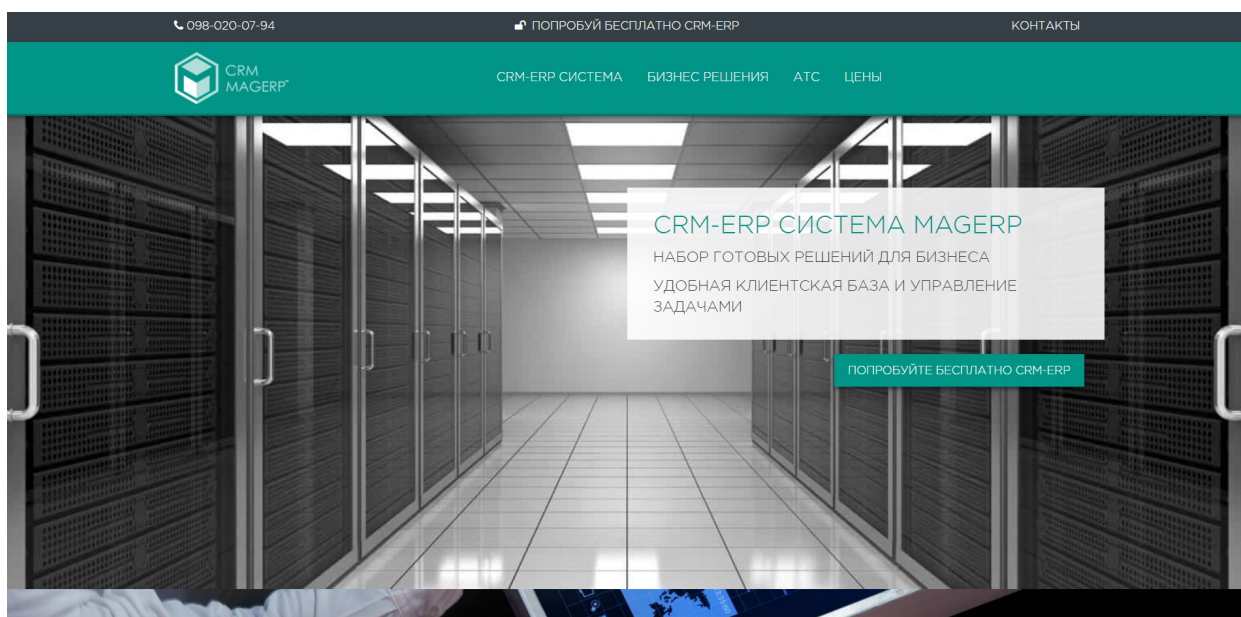


Рисунок 1.3 – Головна сторінка MAGERP CRM

Модуль MAGERP CRM для турагенства – це гнучке і недороге рішення, яке адаптується під бізнес-процеси певної фірми. Так, за допомогою даної конфігурації можна:

- зберігати повну інформацію про кожного клієнта;
- автоматизувати роботу менеджерів;
- реєструвати заявки і формувати пакети туристичних послуг;
- автоматизувати процес формування систем знижок;
- вести облік всіх витрат з клієнтами та операторами;
- інтегрувати CRM з системами бронювання квитків і готелів.

Метою будь-якого туристичного агентства є не роздобути клієнта одноразово, а вести його протягом довгого часу, організовуючи для нього кожен відпустку. Для того, щоб MAGERP CRM для туризму сприяла досягненню поставлених цілей, в системі реалізовані наступні функціональні можливості.

Велика клієнтська база, в якій для більшої зручності клієнтів можна розділяти по різних сегментах (нові, постійні, незадоволені тощо).

									Арк.
									15
Змн.	Арк.	№ докум.	Підпис	Дата					

ДПШЗ.170122.01.18.ПЗ

Безпечно зберігання особистих даних клієнтів, щоб під час оформлення кожного наступного замовлення не перепитувати інформацію і оперативно оформляти документи.

Облік всіх путівок і турів, оформлених одним клієнтом. Це дозволить в сезон відпусток оперативно формувати вигідні пропозиції кожного з клієнтів, з огляду на його особисті переваги.

Автоматична розсилка смс і e-mail з палаючими турами, цікавими пропозиціями і навіть привітаннями зі святами.

Нагадування менеджерам про поточні завдання.

Власна аналітика, яка дозволить аналізувати трафік на сайті турагентства.

Програмне забезпечення Бітрікс24.CRM (<https://www.bitrix24.ua/features/>). Вікно веб-сайту, що пропонує дане програмне забезпечення, представлене на рисунку 1.4.

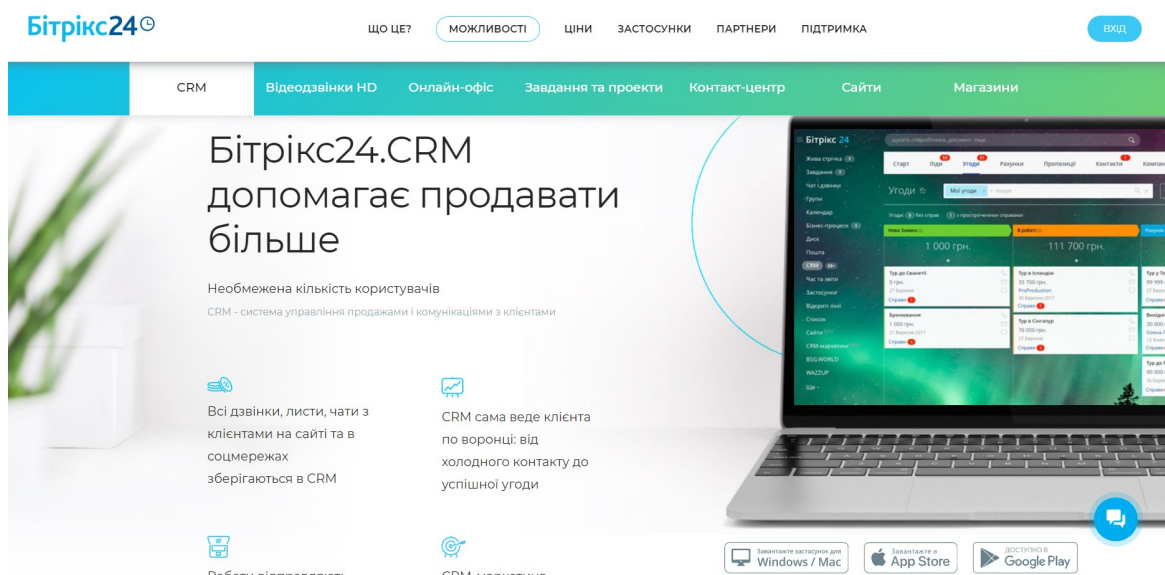


Рисунок 1.4 – Головна сторінка Бітрікс24.CRM

Бітрікс24.CRM для продажів допомагає продавати більше: бере під контроль всі канали комунікацій з клієнтами, підказує, що робити, і автоматизує продажі. Бітрікс24 дозволяє автоматизувати всі процеси у відділі продажів – від внутрішньої роботи до залучення клієнтів та їх обслуговування. Для цього в

									Арк.
									16
Змн.	Арк.	№ докум.	Підпис	Дата					

CRM передбачено цілий комплекс інструментів, наприклад:

- облік звернень клієнтів з усіх каналів (відкриті лінії);
- зручні інструменти для зв'язку з клієнтами (телефонія, пошта, чат); сповіщення та підказки менеджерам;
- можливість ставити завдання і контролювати їх виконання;
- автоматизація типових завдань (бізнес-процеси, роботи і тригери);
- інтеграція зі сторонніми сервісами (інтернет-магазин);
- відстеження замовлень, комерційних пропозицій і оплат (1С);
- аналітичні звіти по кожному співробітнику або угоді (воронки продажів).

Досить один раз налаштувати індивідуальний процес продажів у Бітрікс24, і далі CRM все буде контролювати сама: ставити завдання співробітникам, відправляти листи, запускати рекламу, планувати зустрічі.

Програмне забезпечення МоїТуристи (<https://www.moituristy.ua/ua/>). Вікно веб-сайту представлено на рисунку 1.5.

MoiTuristy.ua – це проста CRM-система для невеликих турагенцій. Облік дзвінків і туристів, нагадування, передача туристів між менеджерами або офісами, гнучке нарахування премій пропорційно внеску, e-mail та смс-маркетинг, аналіз джерел продажів і облік робочого часу. Клієнти задоволені, продажі ростуть, а у вас залишається більше вільного часу.

Програмне забезпечення позиціонується як таке, що не потребує від користувачів високої кваліфікації в галузі інформаційних технологій. Підкреслюється, що програмний продукт належить до покоління хмарних сервісів.

Програмне забезпечення Tourism Creatio галузеве рішення на базі CRM Sales Creatio для туризму ([https://marketplace.terrasoft.ua/app/tourism-powered-creatio?\\_ga=2.132259086.1037236389.1617819119-1764303863.1617819119](https://marketplace.terrasoft.ua/app/tourism-powered-creatio?_ga=2.132259086.1037236389.1617819119-1764303863.1617819119)). Вікно веб-сайту, що пропонує дане програмне забезпечення представлено на рисунку 1.6.

										ДППЗ.170122.01.18.ПЗ	Арк.
											17
Змн.	Арк.	№ докум.	Підпис	Дата							

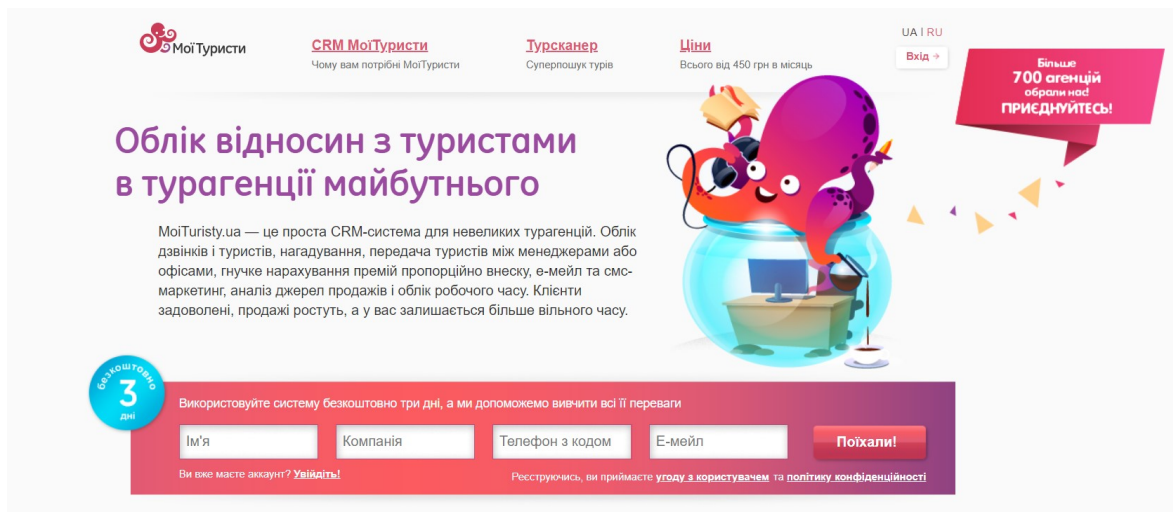


Рисунок 1.5 – Головна сторінка МоїТуристи

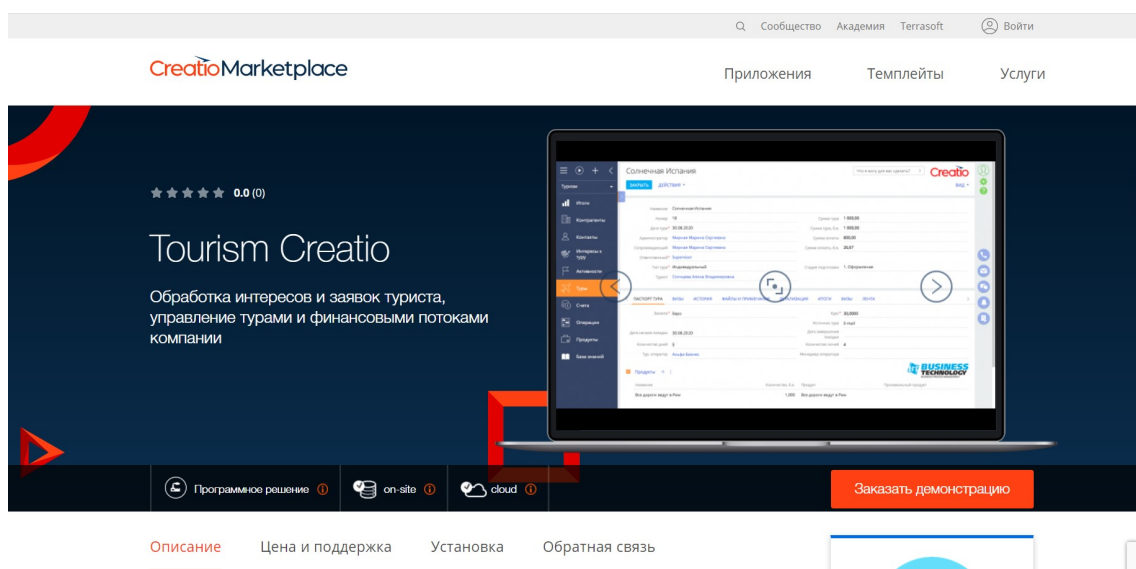


Рисунок 1.6 – Головна сторінка Tourism Creatio

Tourism Creatio – інструмент для автоматизації всього процесу роботи туристичної компанії.

Конфігурація дозволяє закрити всі основні завдання туристичного агентства: обробка потенційних клієнтів, фіксація інтересу, обробка будь-яких типів турів (групові, корпоративні, індивідуальні). Система дозволяє використовувати масове сповіщення клієнтів про гарячі пропозиції або про хід підготовки туру.

Також продукт включає інструменти роботи з фінансовими операціями.

									Арк.
									18
Змн.	Арк.	№ докум.	Підпис	Дата					

ДПШЗ.170122.01.18.ПЗ





## 2 ПРОЕКТУВАННЯ ДОВІДКОВО-ІНФОРМАЦІЙНОЇ СИСТЕМИ ТУРИСТИЧНОЇ ФІРМИ

### 2.1 Проектування структури довідково-інформаційної системи

Проектування структури довідково-інформаційної системи розпочнемо з концептуального моделювання діяльності туристичної фірми. Використовуючи методології сімейства IDEF можна ефективно відображати та аналізувати моделі діяльності широкого спектру складних систем. На сьогоднішній день найбільше розповсюдження та застосування мають методології IDEF0 та IDEF1 (IDEF1X), які отримали в США статус федеральних стандартів [4].

IDEF0 може бути використана для моделювання широкого класу систем. IDEF0 можна використовувати для аналізу функцій, що виконуються системою та відображення механізмів, за допомогою яких ці функції виконуються.

На рисунку 2.1 зображена концептуальна модель туристичної фірми за методологією IDEF0. Фірма позначена прямокутником, стрілками позначені потоки вхідної та вихідної інформації.

Вхідними потоками є:

- рекламні оголошення фірми;
- замовлення клієнтів (контактні дані клієнтів, вид замовлення, опис замовлення);
- оплата замовлення клієнта;
- пропозиції туроператорів;
- пропозиції транспортних компаній (перевізників).

Вихідними потоками є:

- відмова від співробітництва – відмова клієнта від послуг туристичної фірми або неможливість виконання заявки;
- підібрані згідно заявки клієнта тури;
- білети, що забезпечують доставку клієнтів на місце;
- прибуток туристичної фірми;

										ДППЗ.170122.01.18.ПЗ	Арк.
											21
Змн.	Арк.	№ докум.	Підпис	Дата							



безпосередньо займаються обслуговуванням заявок клієнтів;

- клієнти – клієнти туристичної фірми, що роблять замовлення на тури, вибирають транспортні послуги, готель, апартаменти, харчування тощо;
- директор – людина, яка підписує фінансові документи та координує діяльність туристичної фірми;
- бухгалтер, який отримує оплату за виконання послуг від клієнтів, виконує платежі між туристичною фірмою та туроператором тощо.

Виконаємо декомпозицію концептуальної моделі 0-го рівня (рисунок 2.2).

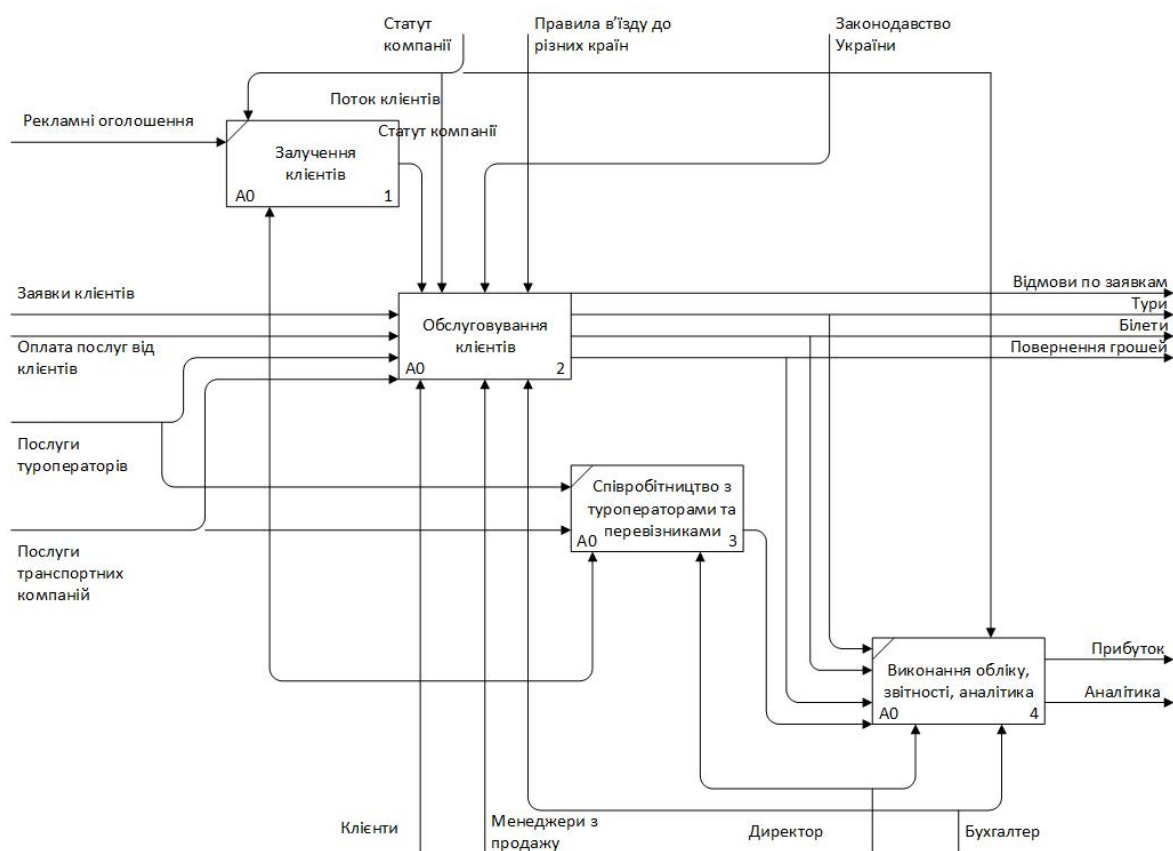


Рисунок 2.2 – Декомпозиція діаграми А0 діяльності турфірми

Діаграма декомпозиції 1-го рівня представлена з 4 блоків:

- залучення клієнтів;
- обслуговування клієнтів;
- співробітництво з туроператорами та транспортними компаніями

(перевізниками);

- складання звітів, аналіз діяльності, аналітика.

Найважливішою функцією туристичної фірми є функція «обслуговування клієнтів», тому виконаємо її деталізацію на діаграмі А1, що представлена на рисунку 2.3.

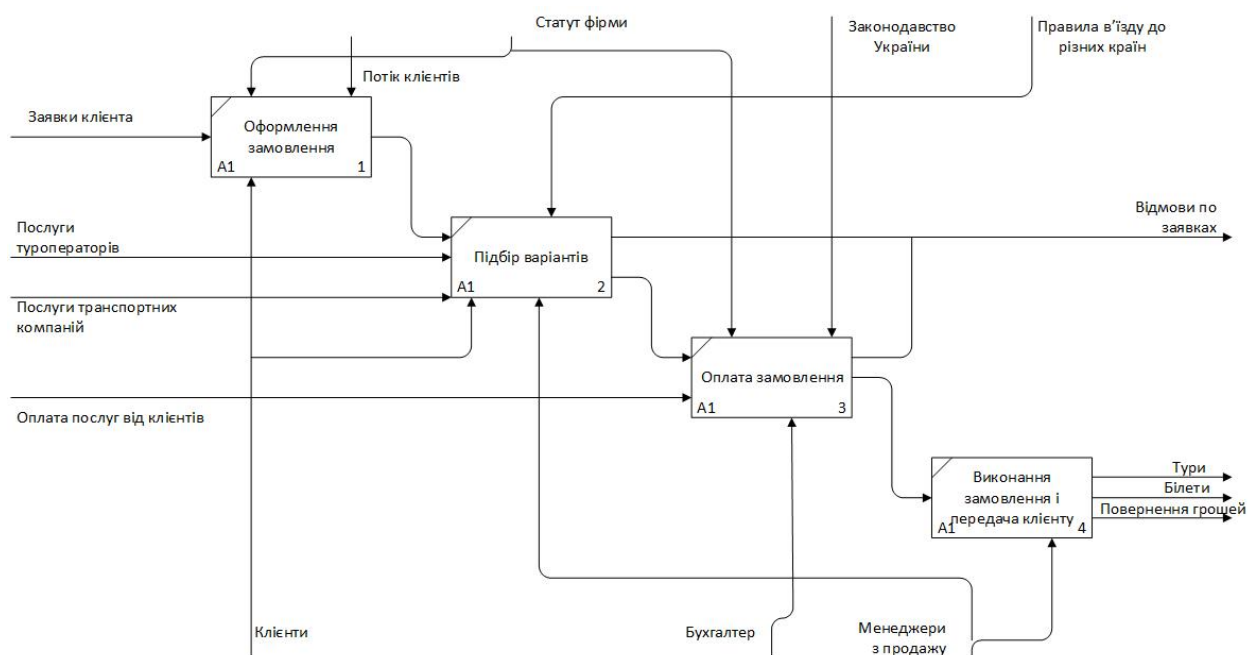


Рисунок 2.3 – Декомпозиція функції «Обслуговування клієнтів»

Складовими функції «Обслуговування клієнтів» є такі підпроцеси:

- оформлення замовлення;
- підбір варіантів виконання замовлення;
- оплата замовлення;
- виконання замовлення та передача документів клієнту.

Взаємодія між менеджером і клієнтом з приводу замовлення послуги починається з реєстрації в реєстрі персональних даних клієнта та вмісту послуги, що потребує клієнт. Основними видами послуг, що надаються туристичною фірмою є продаж туристичного туру разом з перельотом та трансфером, бронювання готелю, бронювання готелю, апартаментів, замовлення харчування, додаткових послуг тощо.

Змн.	Арк.	№ докум.	Підпис	Дата

Після формування вимог замовлення клієнт має мати можливість або власноруч підібрати собі варіант отримання послуги або це може зробити менеджер. Наприклад, вибір кількох варіантів турів у країну, що зацікавила клієнта на вибраний час. При замовленні готелю вибір класу готелю відповідно до зірковості, вибір апартаментів, режиму харчування тощо. Клієнт сам або за допомогою менеджера вибирає те, що його задовільняє найбільше. Менеджер може запропонувати клієнту знижку або клієнт на своєму рахунку має бонуси за попередні замовлення. Можливі випадки коли клієнт не знаходить підходящого для себе варіанту. Якщо варіант підібраний, то менеджер виписує клієнту рахунок, який той має оплати. Після надходження оплати менеджер виконує послугу (бронює тур), оформлює потрібні документи і передає їх конкретному клієнту.

## 2.2 Розробка структури довідково-інформаційної системи

Для проектування структури програмного забезпечення використаємо мову моделювання UML. Уніфікована мова моделювання UML є стандартним інструментом для створення, так званих, креслень програмного забезпечення. За її допомогою можна візуалізувати, специфікувати, конструювати та документувати артефакти будь-яких програмних систем.

UML можна використовувати для моделювання будь-яких систем: від великих інформаційних систем масштабу великого підприємства до розподілених веб-додатків, і, навіть, для вбудованих систем, що працюють в режимі реального часу.

Мова складається зі словника і правил, що створювати комбінації різних слів і отримувати осмислені конструкції. У мові моделювання словник і правила орієнтовані на концептуальне і фізичне уявлення системи. Мова моделювання UML, є стандартним засобом для складання «креслень» довідково-інформаційної системи.

									ДППЗ.170122.01.18.ПЗ	Арк.
										25
Змн.	Арк.	№ докум.	Підпис	Дата						

Моделювання буде нам необхідне для розуміння системи. При цьому єдиною моделі ніколи не буває достатньо. Навпаки, для розуміння будь-якої нетривіальної системи доводиться розробляти велику кількість взаємопов'язаних моделей. У застосуванні до програмних систем це означає, що необхідна мова, за допомогою якого можна з різних точок зору описати уявлення архітектури системи протягом циклу її розробки

Словник і правила такої мови, як UML, пояснює, як створювати і читати добре визначені моделі, але нічого не повідомляють про те, які моделі і в яких випадках потрібно створювати. Це завдання всього процесу розробки програмного забезпечення. Добре організований процес повинен підказати, які потрібні артефакти, які ресурси необхідні для їх створення, як можна використовувати ці артефакти, щоб оцінити виконану роботу і керувати проектом в цілому [5].

Створення моделей на UML переслідує одну мету – полегшення процесу передачі інформації про систему: візуалізація моделі полегшує спілкування.

Розпочинаємо UML моделювання процесу створення інформаційно-довідкової системи туристичної фірми з діаграми варіантів використання (рисунок 2.4).

Діаграми варіантів використання описують взаємовідносини і залежності між групами варіантів використання і дійовими особами (акторами), які беруть участь в процесі.

Важливо розуміти, що діаграми варіантів використання не призначені для відображення проекту і не можуть описувати внутрішній устрій системи. Діаграми варіантів використання призначені для спрощення взаємодії з майбутніми користувачами системи, з клієнтами, і особливо знадобляться для визначення необхідних характеристик системи. Іншими словами, діаграми варіантів використання говорять про те, що система повинна робити, не вказуючи самі застосовувані методи.

										ДППЗ.170122.01.18.ПЗ	Арк.
											26
Змн.	Арк.	№ докум.	Підпис	Дата							

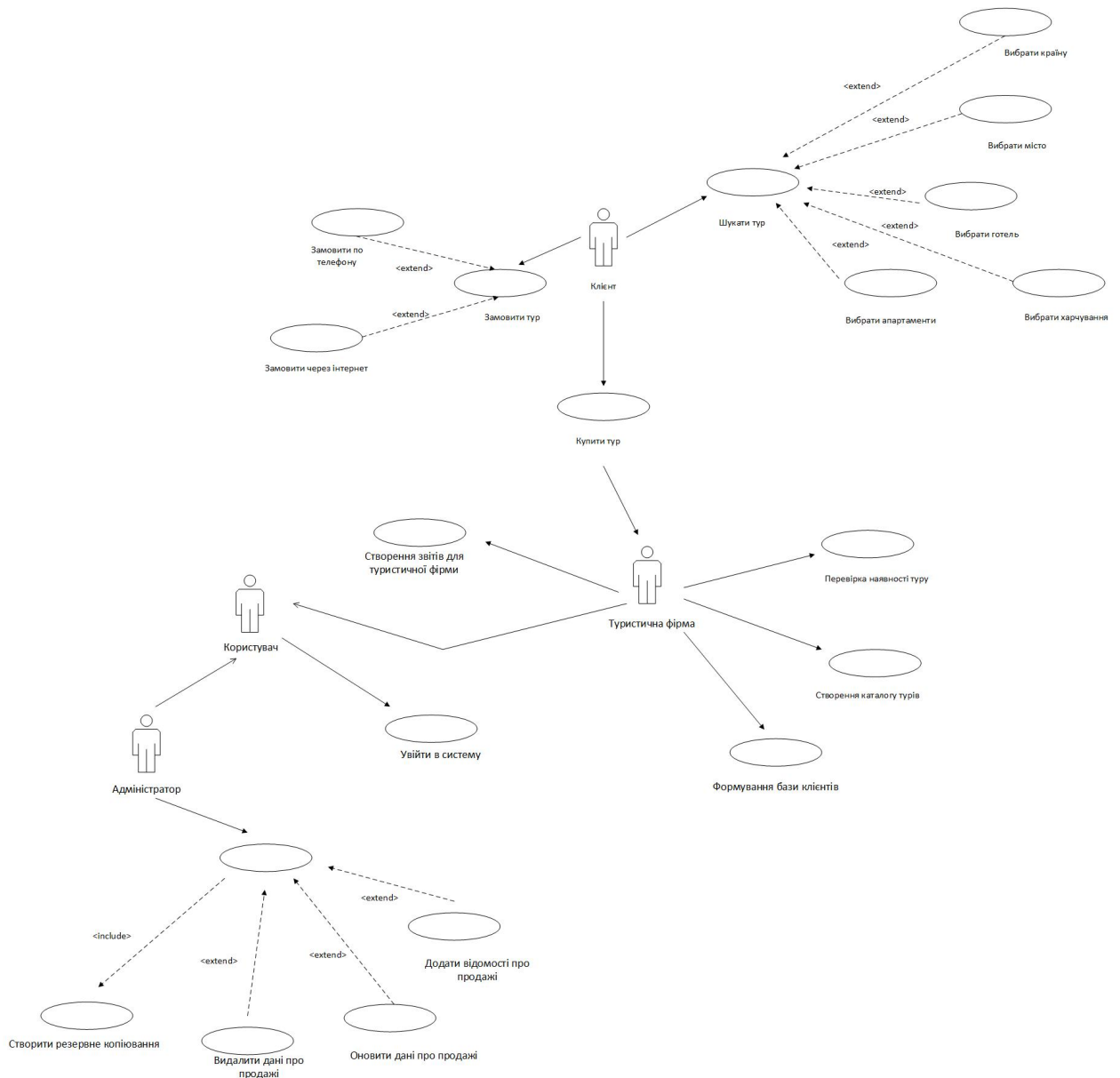


Рисунок 2.4 – Діаграма варіантів використання

На діаграмі представлені наступні актори:

- клієнт;
- туристична фірма;
- користувач;
- адміністратор бази даних.

Між окремими елементами діаграми можемо побачити різні відношення:

- відношення асоціації;
- відношення розширення;
- відношення включення;

Змн.	Арк.	№ докум.	Підпис	Дата

– відношення узагальнення.

Наведемо описи потоків подій до наступних варіантів використання:

- пошук туру;
- купівля туру;
- створення звіту.

Варіант використання «Пошук туру» має на меті дозволити клієнту самому переглянути країни, міста, готелі, апартаменти наявні в готелях, варіанти харчування, тобто всі ті характеристики які він хоче бачити у своєму турі.

Передумова: авторизація в системі.

Основний потік.

1) Клієнт запитує у довідково-інформаційної системи список країн, до яких він хоче потрапити. Система пропонує список країни до яких є тури і клієнт обирає одну чи декілька країн.

2) Клієнт запитує список міст, де розміщуються курорти у країні. Клієнт обирає одне чи декілька міст.

3) Клієнт запитує список готелів, які є в цих містах. Клієнт обирає готель.

4) Клієнт запитує варіанти розміщення в готелі. Клієнт обирає тип апартаментів.

5) Клієнт запитує варіанти харчування в обраному готелі. Клієнт обирає варіант харчування.

Якщо клієнт не зміг нічого підібрати, то виконується альтернативний потік.

б) Варіант використання закінчується.

Альтернативний потік.

1) Клієнт повідомляє довідково-інформаційну систему туристичної фірму, що він не зміг обрати тур.

2) Клієнт повідомляє системі про намір продовжувати пошук турів.

3) Якщо клієнт не має наміру шукати тур, то виконується пункт 5, інакше пункт 4.

										ДППЗ.170122.01.18.ПЗ	Арк.
											28
Змн.	Арк.	№ докум.	Підпис	Дата							

4) Клієнт запускає новий основний потік для підбору тура.

5) Варіант використання завершується.

Післяумова: купити тур.

Варіант використання «Купівля туру» дозволяє клієнту сформувати замовлення і придбати тур в обраний готель з усіма потрібними характеристиками, в тому числі замовлений заздалегідь.

Передумова: авторизація в системі, клієнт вибрав тур.

Основний потік.

1) Система запитує у клієнта кількість осіб для яких він має намір придбати тур, в тому числі дітей. Клієнт повідомляє системі кількість осіб для яких буде придбано тур.

2) Система перевіряє наявність потрібної кількості місць. Система пропонує оплатити тур. Якщо потрібної кількості місць не має в наявності, то виконується альтернативний потік. Клієнт вказує спосіб оплати.

3) При здійсненні оплати, система додає клієнта до бази.

4) Варіант використання завершується.

Альтернативний потік.

1) Система повідомляє клієнту, що потрібної кількості місць не має в наявності. Відбувається перехід на основний потік п.1.

2) Варіант використання завершується.

Після умова: турфірма додає відомості про продаж у базу даних.

Варіант використання «Створення звіту» дозволяє формувати потрібні звітні документи, отримувати аналітику по продажам.

Передумова: авторизація в системі.

Основний потік.

1) Система турфірми запитує з БД усі дані щодо продажу турів, клієнтів.

2) Система турфірми формує потрібний звіт. Якщо помилок не має, то перехід до пункту 4.

3) Адміністратор бази даних виправляє помилку.

										ДПШЗ.170122.01.18.ПЗ	Арк.
											29
Змн.	Арк.	№ докум.	Підпис	Дата							

4) Система відправляє звіт користувачу, який його запитував.

5) Варіант використання завершується.

Післяумова: немає.

Діаграма послідовності являє інтерактивне представлення у вигляді двовимірної діаграми. Діаграма послідовності може бути побудована на декількох рівнях: на рівні бізнес-логіки (рисунок 2.5) та на рівні взаємодії об'єктів (рисунок 2.6). Вертикальна вісь – це вісь часу, а час поширюється вниз по вертикальній лінії. Горизонтальна вісь представляє мета-роль кожного незалежного об'єкта у співпраці. Мета ролі класу представлені лініями життя. Коли об'єкт існує, символ представлений пунктирною лінією, а коли процес об'єкта активований, лінія життя є подвійною лінію.

Повідомлення представлені стрілками від лінії життя одного об'єкта до лінії життя іншого об'єкта. Стрілки розташовані в хронологічному порядку зверху вниз на рисунку.

Діаграма розгортання – діаграма на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Діаграма розгортання в UML моделює фізичне розгортання артефактів на вузлах. Вузли представляються, як прямокутні паралелепіпеди з артефактами, розташованими в них, зображеними у вигляді прямокутників. Вузли можуть мати підвузли, які представляються, як вкладені прямокутні паралелепіпеди. Один вузол діаграми розгортання може концептуально представляти безліч фізичних вузлів, таких, як кластер серверів баз даних.

Діаграму розгортання системи (рисунок 2.7) можна описати наступним чином. Співробітник турфірми взаємодіє із системою шляхом використання інформаційно-довідкової система. Система відправляє запити та отримує відповіді від веб-сервера. Сервер надає користувацький інтерфейс, а також – використовує інтерфейс бази даних для доступу до даних, та їх зміни.

Діаграма компонентів призначена для ілюстрації організації набору компонентів та зв'язків між ними. Зазначена діаграма потрібна для

									Арк.
									30
Змн.	Арк.	№ докум.	Підпис	Дата					

МОДЕЛЮВАННЯ СТАТИЧНОГО ВИДУ СИСТЕМИ З ТОЧКИ ЗОРУ ЇЇ РЕАЛІЗАЦІЇ. (рисунок 2.8).

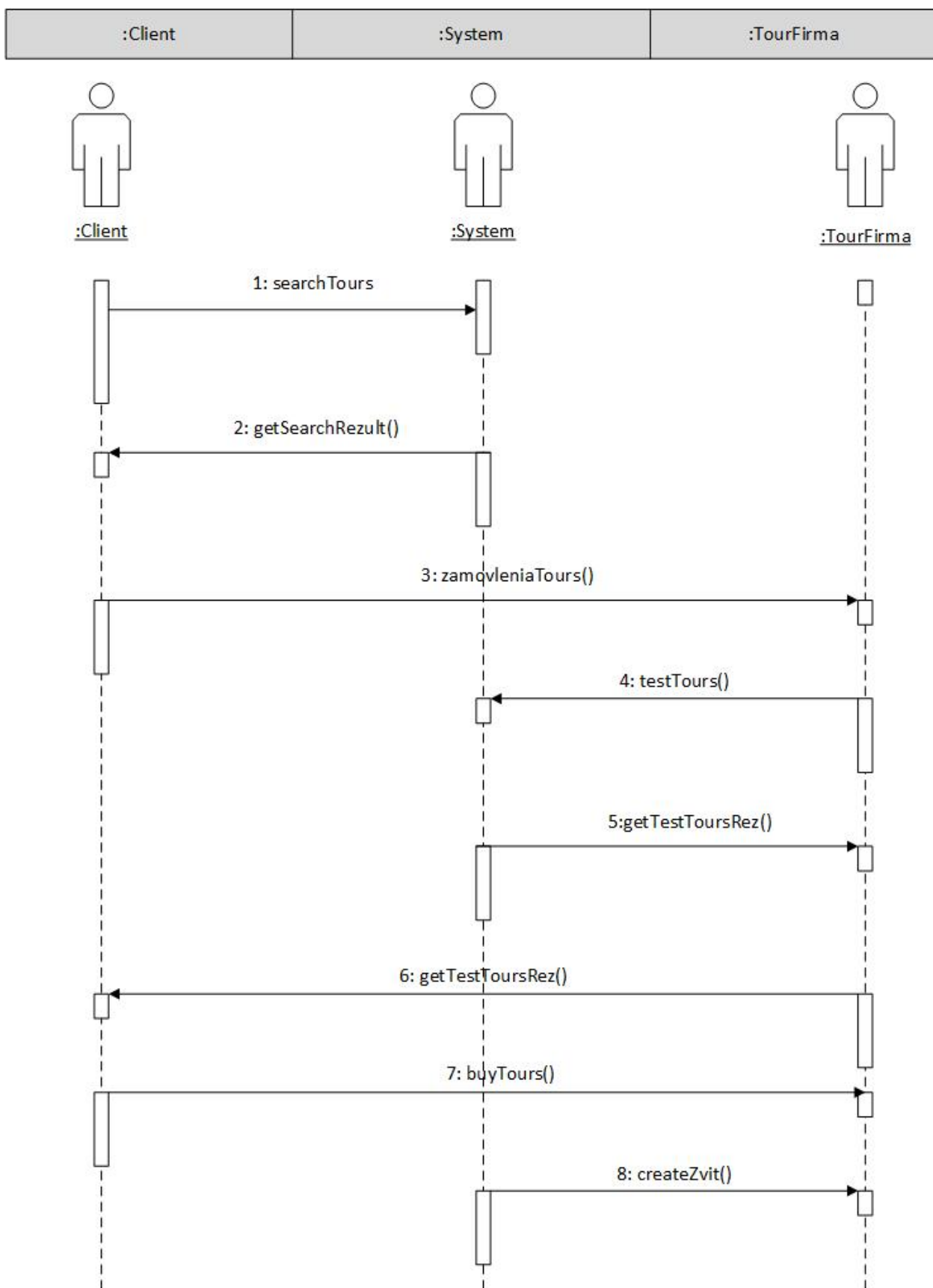


Рисунок 2.5 – Діаграма послідовності на рівні бізнес логіки

Змн.	Арк.	№ докум.	Підпис	Дата

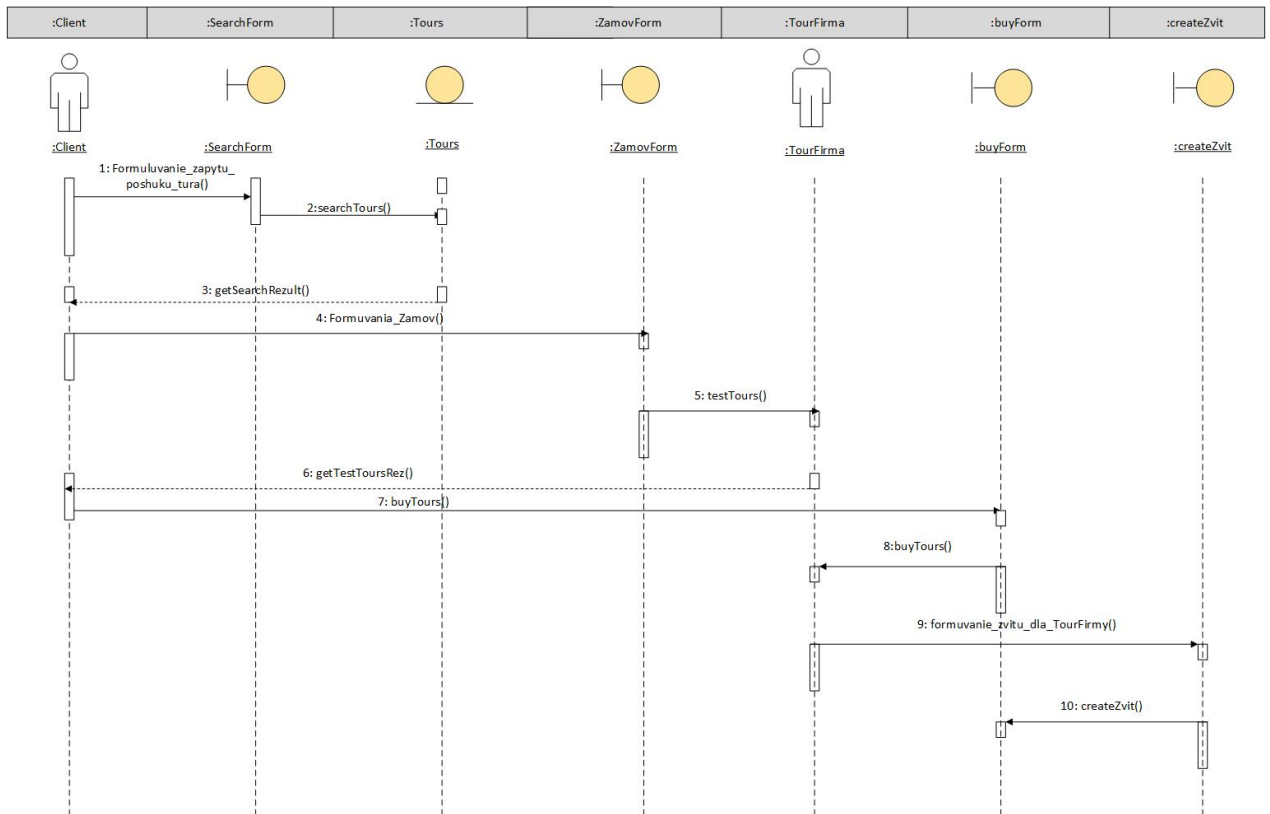


Рисунок 2.6 – Діаграма послідовності на рівні взаємодії об'єктів

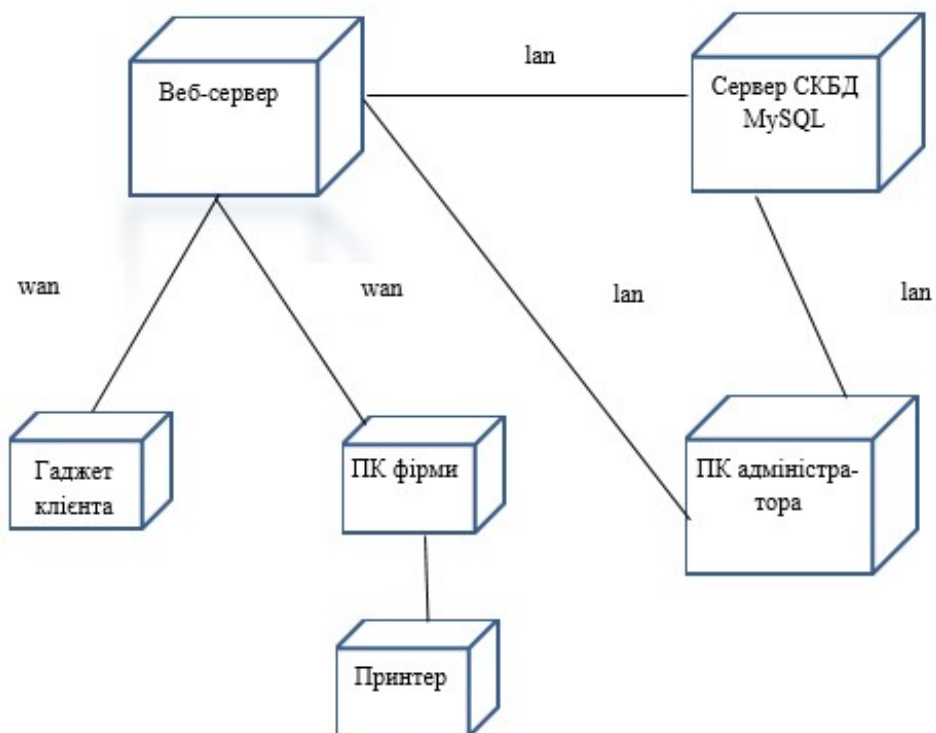


Рисунок 2.7 – Діаграма розгортання

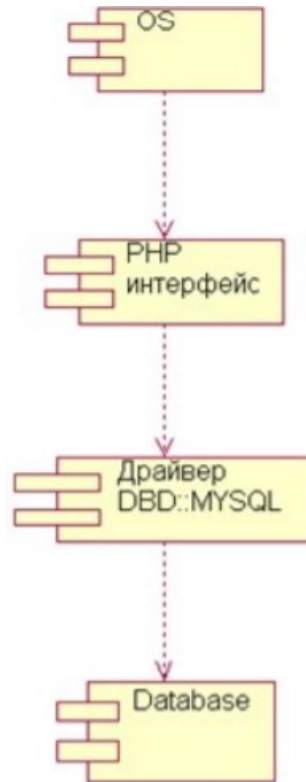


Рисунок 2.8 – Діаграма послідовності

Після виконання uml-моделювання перейдемо до проектування логічної моделі бази даних.

### 2.3 Проектування логічної моделі бази даних

Інфологічна модель будується після аналізу предметної області. Виділяються сутності предметної області, описуються їх атрибути. Між сутностями можуть бути встановлені зв'язки – бінарні асоціації, які показують спосіб співіснування або взаємодії сутностей між собою. Зв'язок може існувати між двома різними сутностями або між сутністю і нею самою (рекурсивний зв'язок). Зв'язки показують, як пов'язані сутності між собою. Якщо зв'язок встановлюється між двома сутностями, то він визначає взаємозв'язок між екземплярами однієї та іншої сутності. Наочне зображення логічної моделі

можливе табличним способом, коли кожному типу запису відповідає таблиця з множиною записів, як показано на рисунку 2.9.

Призначення таблиць наведено в таблиці 2.1.

Таблиця 2.1 – Основні таблиці бази даних

№	Назва таблиці	Призначення
1	Тур	Зберігає дані про тури
2	Готелі	Зберігає дані про готелі, до яких можна придбати тури
3	Апартаменти	Зберігає дані про тип апартаментів, в яких можна проживати під час туру
4	Харчування	Зберігає дані про тип харчування, яке можна вибрати під час вибору туру
5	Користувачі	Зберігає дані про користувачів довідково-інформаційної системи
6	Замовлення	Зберігає дані про замовлення, які були зроблені клієнтами
7	Країни	Зберігає данні про країни, які можна відвідати в рамках туру
8	Регіони	Зберігає дані про регіони і курорти
9	Продажі	Зберігає дані про продаж турів

Після побудови інфологічної моделі бази даних приступимо до вибору технологій для реалізації довідково-інформаційної системи туристичної фірми.

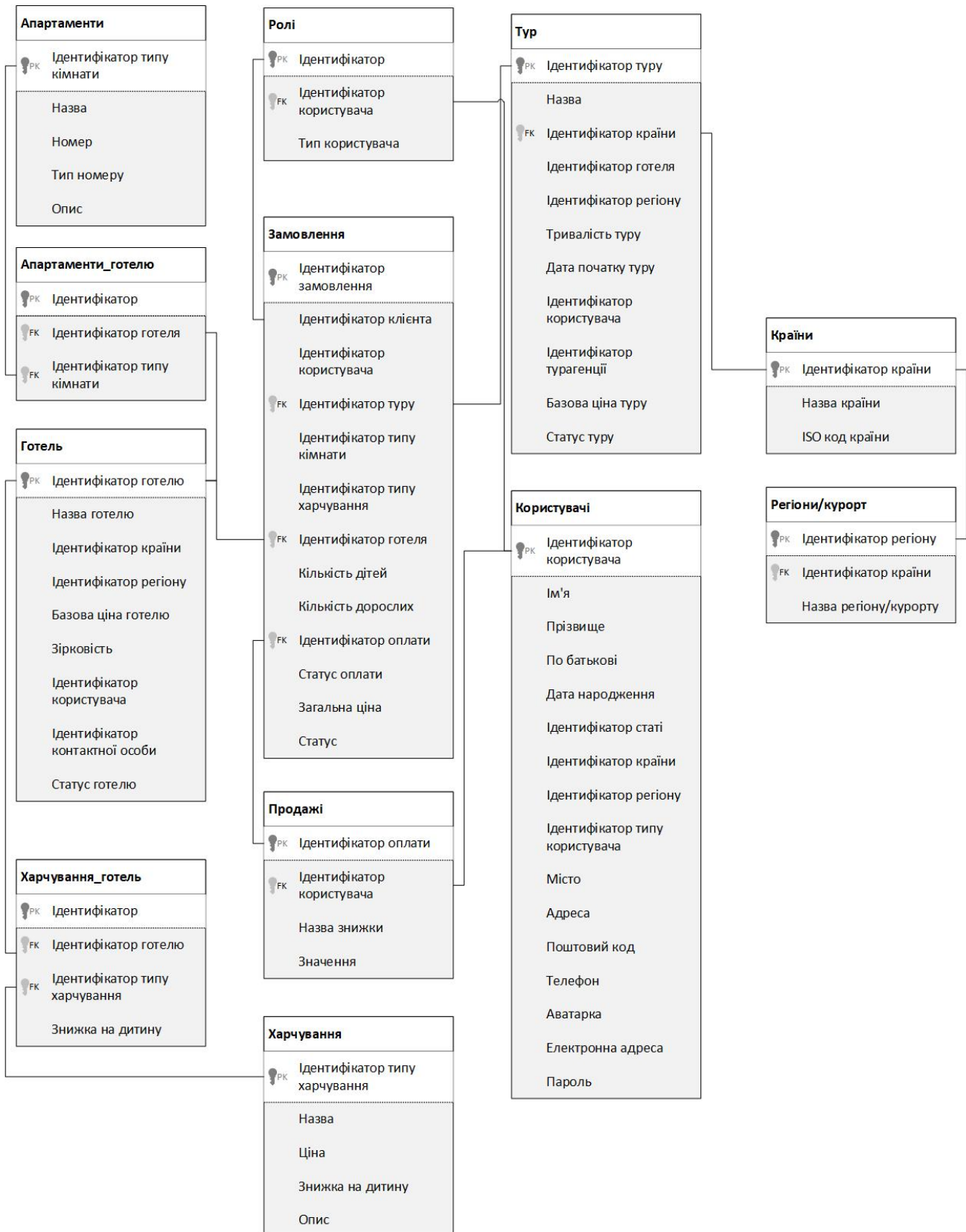


Рисунок 2.9 – Інфологічна модель бази даних

## 2.4 Аналіз та вибір технологій і методів реалізації довідково-інформаційної системи

Для програної реалізації всіх спроектованих функцій довідково-інформаційної системи туристичної фірми були обрані наступні технології:

- мова програмування PHP;
- система управління базами даних MySQL;
- фреймворк Bootstrap;
- бібліотека JQuery;
- каскадні таблиці стилів CSS3;
- мова гіпертекстової розмітки HTML 5;
- шаблонізатор Smarty.

Однією з найпопулярніших мов програмування в середовищі веб-розробників є PHP. Початкову версію PHP розробив Рasmus Лердорф. Основною ідеєю була розробка набору інструментів для спрощення процесу створення динамічних веб-сторінок. Сучасний PHP найчастіше використовується як серверний інструмент генерації HTML-коду, який буде інтерпретовуватися веб-браузером, хоча PHP є мовою загального призначення.

PHP – це мова програмування з відкритими початковими кодами. Розвивають мову PHP програмісти-ентузіасти зі всього світу. Ця мова має простий синтаксис подібний до Java і C++. Мова постійно розвивається і на даний час актуальною є 7-ма версія мови. За статистикою, кожен шостий програмний продукт створений на PHP.

Можна виділити три основні області використання PHP [6].

1) Для написання скриптів і повноцінних веб-додатків, що виконуються на стороні сервера. Оскільки мова була створена саме для веб-розробок, то це власне найпопулярніша сфера застосування PHP. Щоб веб-програма написана на PHP повноцінно працювала необхідними є сервер, парсер (CGI-додаток) і веб-браузер, який відображає результат виконання коду.

									ДППЗ.170122.01.18.ПЗ	Арк.
										36
Змн.	Арк.	№ докум.	Підпис	Дата						

2) Для створення сценаріїв, що будуть виконуватися в командному рядку. Саме такі веб-додатки можуть працювати на будь-якому персональному комп'ютері. Для їх виконання необхідним є тільки парсер. PHP славиться потужними інструментами для роботи з рядками, тому такі сценарії найчастіше створюються для обробки текстових даних.

3) Для створення графічних інтерфейсів. PHP має багато відгалужень, які були створені для реалізації найрізноманітніших завдань. Одним з таких відгалужень є PHP-GTK. Він зазвичай використовується тими програмістами, які звикли до синтаксису PHP.

Використання PHP для розробки довідково-інформаційної системи туристичної фірми обумовлено його такими перевагами.

Простий і інтуїтивно зрозумілий синтаксис. Навіть починаючі програмісти можуть швидко освоїти PHP. Йому притаманні кращі особливості таких популярних мов як C, Java і Perl. Його код легкий для читання незалежно від того, що було створено за його допомогою – невеликі скрипти чи потужні додатки.

Кросплатформеність і гнучкість. PHP можна суміщати з усіма найпопулярнішими операційними системами – Linux, Windows, MacOS. Додатки, що були написані за допомогою PHP можуть працювати на різному серверному програмному забезпеченні – IIS, Nginx, Apache і багатьох інших.

Маштабованість. Із зростанням апаратних ресурсів, що використовуються, PHP дозволяє домогтися максимальної продуктивності додатків, що були написані цією мовою. Веб-додатки, розподілені на кілька серверів можуть справлятися з великим трафіком.

Вбудовуваність в HTML-документи. Просту HTML-сторінку можна легко вдосконалити змінюваним контентом шляхом вставки блоків коду PHP. Ці блоки додаються аналогічно HTML-тегам і не змінюють структуру документа.

Постійне вдосконалення мови. Спільнота розробників активно вдосконалює існуючий функціонал, розширює можливості мови, спрощує

									Арк.
									37
Змн.	Арк.	№ докум.	Підпис	Дата					

синтаксис і покращує захист від можливих атак.

Детальна документація. Офіційний сайт проекту надає повні відомості про кожну функціональну одиницю мови з прикладами використання.

Легкий пошук рішень виникаючих проблем. В мережі існує величезна кількість форумів про програмування на PHP, де можна обговорити виникаючі проблеми з програмістами-практиками.

Перспективи подальшого розвитку. Переважна більшість CMS були створені за допомогою чистого PHP та фреймворку. Завдяки цьому PHP є настільки популярним та затребуваним серед програмістів [7].

Жодну довідково-інформаційну систему не можна розробити без системи управління базами даних. В нашому випадку була обрана MySQL.

MySQL – це система управління базами даних (СКБД), яка поширюється як вільне програмне забезпечення (користувачі мають право на необмежену установку, запуск, вільне використання).

MySQL підтримує SQL (структурована мова запитів) і може застосовуватися в якості SQL-сервера. Це означає, що спілкуватися з сервером можна на мові SQL: клієнт посилає серверу запит, той його обробляє і віддає клієнту тільки ті дані, які були отримані в результаті цього запиту. Тим самим клієнтові не потрібно викачувати дані і робити обчислення, як, наприклад, в Microsoft Access.

Крім того, MySQL – це ПЗ з відкритим кодом, тобто його можна вільно вивчати та змінювати. Пакет розповсюджується на умовах GPL (General Public License), його можна безкоштовно завантажити з Інтернету (<http://www.mysql.com>) для некомерційного використання.

З появою Інтернет-технологій, що дозволяють створювати динамічні веб-сторінки, з'явилася потреба на СУБД, які будуть найбільш повно підходити б для цього за швидкодією, надійністю і стабільністю. MySQL є простою, швидкою і надійною СУБД з достатньою функціональністю для розробки нескладних інформаційних систем.

										Арк.
										38
Змн.	Арк.	№ докум.	Підпис	Дата						

Сьогодні MySQL є найпопулярнішою СУБД на ринку. Области використання MySQL наведені нижче [8].

1) Основна область використання MySQL - це веб. Більшість сайтів, чатів і форумів працюють на даній СУБД. CMS (Content Manage System), яка необхідна для стабільної роботи сайтів, також працює під управлінням MySQL.

У галузі веб-розробки MySQL лідер. Це пов'язано не тільки з тим, що продукт поставляється з відкритим кодом, але і з гнучкістю конфігурації, простотою управління.

2) Корпоративні системи за типом ERP / CRM, особливо в сегменті SMB. Це пов'язано з браком коштів на придбання Microsoft SQL або аналогічних рішень. Тому більшість компаній даного сегмента воліють використовувати безкоштовний продукт, за функціональністю не поступається конкурентам.

MySQL-частина програмного комплексу, який скорочено називають LAMP. Аббревіатура розшифровується як Linux, Apache, MySQL (MariaDB), PHP. Сам комплекс був розроблений як безкоштовна альтернатива аналогічним продуктам з комерційною ліцензією. Частина дистрибутивів Linux поставляється відразу з передвстановленим LAMP (або його аналогами в залежності від типу ОС).

Зв'язка LAMP розроблена для операційних систем сімейства Linux / Unix, тому для стабільної роботи під ОС Windows потрібно замінити компоненти на сумісні, наприклад, WASP (Windows, Apache, SQL, PHP) або WIMP (Windows, IIS, SQL, PHP).

MySQL – відмінна альтернатива комерційним продуктам на ринку СУБД. Відкритий вихідний код дозволяє клієнтам допрацьовувати програмне забезпечення під свої завдання і вимоги [9].

Переваги MySQL.

Крім універсальності і поширеності СУБД MySQL володіє цілим комплексом важливих переваг перед іншими системами. Зокрема слід відзначити наступні якості.

									Арк.
									39
Змн.	Арк.	№ докум.	Підпис	Дата					

ДППЗ.170122.01.18.ПЗ

Простота у використанні. MySQL досить легко інсталується, а наявність безлічі плагінів і допоміжних додатків спрощує роботу з базами даних.

Великий функціонал. Система MySQL має практично всім необхідним інструментарієм, який може знадобитися в реалізації практично будь-якого іншого проекту.

Безпека. Система спочатку створена таким чином, що безліч вбудованих функцій безпеки в ній працюють за замовчуванням.

Масштабованість. Будучи досить універсальною СУБД, MySQL в рівній мірі легко може бути використана для роботи з різними об'ємами даних.

Швидкість. Висока продуктивність системи забезпечується за рахунок спрощення деяких використовуваних в ній стандартів [9].

Фреймворки – це програмні продукти, які спрощують створення і підтримку технічно складних або навантажених проектів. Фреймворк, як правило, містить тільки базові програмні модулі, а все специфічні для проекту компоненти реалізуються розробником на їх основі. Тим самим досягається не тільки висока швидкість розробки, а й велика продуктивність.

Веб-фреймворк – це платформа для створення сайтів і веб-додатків, що полегшує розробку і об'єднання різних компонентів великого програмного проекту. За рахунок широких можливостей в реалізації бізнес-логіки і високої продуктивності ця платформа особливо добре підходить для створення складних сайтів, бізнес-додатків і веб-сервісів.

Для розробки довідково-інформаційної системи туристичної фірми було обрано фреймворк Bootstrap. Він з'явився в далекому 2011 році в компанії Twitter. Спочатку це була невелика бібліотека для внутрішніх потреб компанії, проте він швидко завоював популярність і вийшов далеко за її межі.

На сьогоднішній день Bootstrap є визнаним фаворитом серед розробників адаптивних сайтів. Це фреймворк, що складається з шаблонів HTML, CSS і розширень JavaScript.

Він дотримується принципу «mobile-first», що має на увазі початкову

									Арк.
									40
Змн.	Арк.	№ докум.	Підпис	Дата					

розробку сайту для невеликого екрану мобільного телефону, потім контент розміщується згідно з дозволом планшета і, нарешті, верстальник працює над версією для ноутбука і комп'ютера. Завдяки такому підходу кінцевий користувач може переглядати сайт на будь-яких пристроях, не відчуваючи найменшого незручності.

Переваги розробки додатку на Bootstrap:

- швидкість розробки: завдяки великій кількості встановлених компонентів верстальщику немає потреби починати розробку проекту з нуля;
- адаптивність. Фреймворк Bootstrap включає в себе 12-ти колоночну сітку. При розробці сайту верстальщику залишається тільки прийняти рішення, скільки колонок відвести на той чи інший компонент в залежності від розміру екрана користувача.
- популярність. Мільйони сайтів по всьому світу створені на Bootstrap. Відповідно, розроблено безліч шаблонів і різних варіантів оформлення.
- безкоштовність. Bootstrap можна використовувати безкоштовно як при самостійній розробці сайту, так і при комерційної діяльності.
- гнучке налаштування. Bootstrap підтримує можливість гнучкого налаштування компонентів. Можна вибрати тільки ті компоненти CSS і JavaScript, які знадобляться в поточному проекті.
- документованість. Офіційний сайт Bootstrap володіє вичерпною документацією з великою кількістю прикладів; також багато інформації можна знайти на спеціалізованих форумах.
- підтримка. Bootstrap активно розвивається, виходять нові версії, виправляються помилки і з'являється новий функціонал [10].

Бібліотека JQuery – це фреймворк, який був розроблений на базі технології Java Script. Від спрощує написання великих фрагментів програмного коду. Вказана бібліотека містить велику кількість заготовлених функцій для створення інтерактивних елементів веб-додатку. Ця бібліотека працює з елементами DOM і дозволяє маніпулювати ними, з іншої сторони за її

									ДППЗ.170122.01.18.ПЗ	Арк.
										41
Змн.	Арк.	№ докум.	Підпис	Дата						

допомогою реалізують API модулі для використання AJAX.

Бібліотека JQuery призначена для програмування різноманітних елементів веб-ресурсів та створення веб-додатків. Це не комерційна розробка, яка зв'явилася силами програмістів-ентузіастів.

Каскадні таблиці стилів CSS3 та мова гіпертекстової розмітки HTML 5. CSS (англ. Cascading Style Sheets - каскадні таблиці стилів) – технологія опису зовнішнього вигляду документа, оформленого мовою розмітки.

Переважає використання як засіб оформлення веб-сторінок в форматі HTML і XHTML, але може застосовуватися з будь-якими видами документів в форматі XML, включаючи SVG і XUL.

Каскадні таблиці стилів використовуються розробниками веб-сторінок для завдання кольорів, шрифтів, розташування і інших аспектів представлення веб-документа. Основною метою розробки CSS було розділення вмісту (написаного на HTML або іншій мові розмітки) і оформлення документа (написаного на CSS). Це поділ може збільшити доступність документа, надати йому гнучкість і можливість управління його поданням, а також зменшити складність і повторюваність в структурному вмісті. Крім того, CSS дозволяє представляти один і той же документ в різних стилях або методах виведення, таких як екранне уявлення, друк, читання голосом (спеціальним голосовим браузером або програмою читання з екрану), або при виведенні пристроями, що використовують шрифт Брайля.

Каскадні таблиці стилів (Cascading Style Sheets, CSS) – це стандарт, який визначає подання даних в браузері. Якщо HTML надає інформацію про структуру документа, то таблиці стилів повідомляють як він повинен виглядати.

Стиль – це сукупність правил, що застосовуються до елемента гіпертексту і визначають спосіб його відображення. Стиль включає всі типи елементів дизайну: шрифт, фон, текст, кольори посилань, поля і розташування об'єктів.

Таблиця стилів – це сукупність стилів, які можна застосувати до гіпертекстових документів.

									Арк.
									42
Змн.	Арк.	№ докум.	Підпис	Дата					

Каскадування – це порядок застосування різних стилів до веб-сторінці. Браузер, що підтримує таблиці стилів, буде послідовно застосовувати їх відповідно до пріоритетом: спочатку пов'язані, потім впроваджені і, нарешті, вбудовані стилі. Інший аспект каскадирования – успадкування (inheritance), означає, що якщо не вказано інше, то конкретний стиль буде застосований до всіх дочірнім елементами гіпертекстового документа.

Використання каскадних таблиць дає можливість розділити вміст і його поєднання і гнучко управляти відображенням гіпертекстових документів шляхом зміни стилів.

Шаблонізатор – це програмне забезпечення, яке дозволяє використовувати HTML-шаблони, щоб створювати HTML-сторінок. Основна мета використання шаблонізатору є відділення представлення даних від виконуваного коду. Шаблонізатор використовують для отримання більш красивого і зрозумілого результату. Часто використання шаблонізатору необхідно для забезпечення можливості паралельної роботи програміста і дизайнера-верстальника. Такий підхід значно прискорює час розробки та прототипування додатку, дизайнеру не потрібно розбиратися в програмуванні, а програмісту перейматися інтерфейсом. Використання шаблонізатору покращує читабельність коду і спрощує внесення змін в зовнішній вигляд, коли проект повністю виконує одна людина. Для проекту був обраний шаблонізатор Smarty.

Використання вище описаних технологій надає можливість реалізувати всі спроектовані функції і забезпечує використання ліцензованого програмного забезпечення для розробки власного додатку.

Отже, другий розділ присвячений проектуванню довідкової-інформаційної системи туристичної фірми. Було проведено концептуальне моделювання діяльності туристичної фірми, розроблена структура програмного забезпечення, побудована логічна модель бази даних та вибрані технології та методи реалізації програмного забезпечення.

										ДППЗ.170122.01.18.ПЗ	Арк.
											43
Змн.	Арк.	№ докум.	Підпис	Дата							

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

#### 3.1 Реалізація бази даних

Першим кроком до реалізації баз даних довідково-інформаційної системи туристичної фірми є розробка фізичної моделі даних. Фізична модель даних представляє собою конкретне подання концептуальної та логічної моделі даних в комп'ютері. Ця модель описує певну організаційну структуру даних на фізичному носії даних. Вона пов'язана не тільки з вибраною системою керування бази даних, а також з вибраною операційною системою та певним обладнанням.

Концептуальна модель бази даних описує головні (основні) сутності та відношення між ними. Логічна модель розширює концептуальну шляхом визначення для сутностей їх атрибутів, описів і обмежень, деталізує склад сутностей та взаємозв'язки між ними. Фізична модель описує реалізацію об'єктів логічної моделі на рівні конкретної бази даних.

Іншими словами, на при побудові концептуальної моделі, об'єктами моделі є предметна область (subject area), сутності (entitys) та взаємозв'язки між сутностями (entity relationships); при побудові логічної моделі – сутності (entitys), взаємозв'язки між сутностями (entity relationships), атрибути (attributes), первинні ключі (primary keys), зовнішні ключі (foreign keys); при побудові фізичної моделі – первинні ключі (primary keys), зовнішні ключі (foreign keys), назви таблиць (table names), назви полів (column names), типи даних (data types).

Фізична модель даних будується на базі логічної моделі даних і має всебічно враховувати обмеження різних умов зберігання для проектування бази даних.

Фізична модель даних довідково-інформаційної системи туристичної фірми наведена на рисунку 3.1.

					ДППЗ.170122.01.18.ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		



Таблиця 3.1 – Таблиця foods

Назва поля	Поле	Тип	Розмір	Індекс
Ідентифікатор	id	int	11	Primary
Назва	title	varchar	255	
Код	cod	varchar	3	
Знижка на дитину	sale	float		
Опис	description	text		

Таблиця 3.2 – Таблиця members

Назва поля	Поле	Тип	Розмір	Індекс
Ідентифікатор	id	int	11	Primary
Ім'я	first_name	varchar	255	
Прізвище	last_name	varchar	255	
По-батькові	middle_name	varchar	255	
Дата народження	dob	date		
Ідентифікатор статі	gender_id	int	11	index
Ідентифікатор країни	country_id	int	11	index
Ідентифікатор регіону	region_id	int	11	index
Ідентифікатор типу користувача	role_id	int	11	index
Місто	city	varchar	255	
Адреса	address	varchar	255	
Поштовий код	zip	varchar	255	
Телефон	phone	varchar	255	
Аватарка	photo	varchar	255	
Електронна адреса	email	varchar	255	
Пароль	password	varchar	255	
Дата створення	inserted	datetime		
Дата поновлення	updated	datetime		

Таблиця 3.3 – Таблиця rooms

Назва поля	Поле	Тип	Розмір	Індекс
Ідентифікатор	id	int	11	Primary
Назва	title	varchar	255	
Код	cod	varchar	3	
Тип номеру	type	int	11	index
Опис	description	text		

Таблиця 3.4 – Таблиця hotels

Назва поля	Поле	Тип	Розмір	Індекс
Ідентифікатор	id	int	11	Primary
Назва	title	varchar	255	
Ідентифікатор країни	country_id	int	11	index
Ідентифікатор регіону	region_id	int	11	index
Базова ціна готеля	price	decimal	10,2	
Зірковість	star	int	11	
Ідентифікатор користувача	member_id	int	11	index
Ідентифікатор контактної особи	owner_id	int	11	index
Статус готелю	status	int	11	
Дата створення	inserted	datetime		
Дата поновлення	updated	datetime		

Таблиця 3.5 – Таблиця tours

Назва поля	Поле	Тип	Розмір	Індекс
1	2	3	4	5
Ідентифікатор	id	int	11	Primary
Назва	title	varchar	255	
Ідентифікатор країни	country_id	int	11	index
Ідентифікатор готеля	hotel_id	int	11	index
Ідентифікатор регіону	region_id	int	11	index
Тривалість туру	duration	int	11	index
Дата початку туру	tour_start	date		
Ідентифікатор користувача	member_id	int	11	index
Ідентифікатор турагенції	agency_id	int	11	index
Базова ціна туру	price	decimal	10,2	

### Кінець таблиці 3.5

1	2	3	4	5
Статус туру	status	int	11	
Дата створення	inserted	datetime		
Дата поновлення	updated	datetime		

### Таблиця 3.6 – Таблиця hotel\_rooms

Назва поля	Поле	Тип	Розмір	Індекс
Ідентифікатор	id	int	11	Primary
Ідентифікатор готеля	hotel_id	int	11	Index
Ідентифікатор типу кімнати	room_type_id	int	11	Index
Сортування	sort	int	11	

### Таблиця 3.7 – Таблиця orders

Назва поля	Поле	Тип	Розмір	Індекс
Ідентифікатор	id	int	11	Primary
Назва	title	varchar	255	
Ідентифікатор клієнта	client_id	int	11	index
Ідентифікатор користувача	owner_id	int	11	index
Ідентифікатор туру	tour_id	int	11	index
Ідентифікатор типу кімнати	room_id	int	11	index
Ідентифікатор типу харчування	food_id	int	11	index
Ідентифікатор готеля	hotel_id	int	11	index
Кількість дітей	adults	int	11	
Кількість дорослих	childrens	int	11	
Ідентифікатор оплати	payment_id			
Статус оплати	payment_status			
Загальна ціна	total_price	decimal	10,2	
Статус	status	int	11	
Дата створення	inserted	datetime		
Дата поновлення	updated	datetime		

Змн.	Арк.	№ докум.	Підпис	Дата

ДППЗ.170122.01.18.ПЗ

Арк.

48



Таблиця 3.12– Таблиця roles

Назва поля	Поле	Тип	Розмір	Індекс
Ідентифікатор	id	int	11	Primary
Ідентифікатор користувача	member_id	int	11	Index
Тип користувача	title	text		
Сортування	sort	int	11	

Після побудови фізичної моделі бази даних перейдемо до реалізації програмних модулів системи.

### 3.2 Реалізація модулів системи

Для програмної реалізації довідково-інформаційної системи туристичної фірми було побудовано UML-діаграму класів (рис. 3.2). Кожний клас має свої змінні та методи.

Довідково-інформаційна система туристичної фірми складається з наступних програмних модулів:

- користувачі;
- налаштування;
- курорти;
- готелі;
- тури;
- замовлення;
- модуль інтерфейсу;
- модуль пошуку;
- модуль фільтрів.

Саме такий склад програмних модулів дає можливість реалізувати всі потрібні функції, описані в технічному завданні.

Модуль «Користувачі». Цей модуль призначений для надання прав доступу

										Арк.
										50
Змн.	Арк.	№ докум.	Підпис	Дата						



Рисунок 3.2 – Діаграма класів

до контенту довідково-інформаційної системи. Без використання цього модуля є можливість переглядати тільки загальний контент розміщений на веб-сторінці. Для виконання дій в інформаційно-довідковій системі туристичної фірми необхідна авторизація та автентифікація користувача. Користувачами системи є:

«Клієнт» – це користувач довідково-інформаційної системи, який хоче переглянути всі можливі варіанти турів та сконструювати собі тур у відповідності зі своїми бажаннями. Також, він може звернутися через систему до менеджера, який допоможе йому з замовленням.

«Менеджер» – це співробітник туристичної фірми (менеджер, бухгалтер, директор), який може через довідково-інформаційну систему формувати замовлення для клієнтів та вносити інформацію в систему.

«Адміністратор» – як правило співробітник туристичної або найнятий фахівець, що займається адмініструванням довідково-інформаційної системи та бази даних.

Модулю «Користувачі» властиві такі методи класу:

- реєстрація користувача;
- редагування введених даних користувача;
- видалення даних користувача;
- отримання даних користувача;
- формування форм для введення даних;
- надання прав доступу;
- формування переліку користувачів.

Модуль «Налаштування» призначений для введення загальних налаштувань, таких як мови, валюти, країни тощо, так і для введення налаштувань турів (статуси турів – активний/не активний, категорії, зірковість готелю тощо).

Модуль «Курорти» призначений для опису курорту на який може бути запланований тур.

Модуль «Готелі» призначений для повного опису готелів, який можуть

									ДППЗ.170122.01.18.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата						52

вибрати клієнти.

Модуль «Тури» призначений для опису туру, який можуть вибрати клієнти.

Модуль «Замовлення» призначений для підрахунку суми замовлення на вибраний курорт, готель, вибраний тур.

Модуль інтерфейсу призначений для налаштування інтерфейсу довідково-інформаційної системи.

Модуль пошуку призначений для пошуку інформації в інформаційно-довідковій системі туристичної фірми.

Модуль фільтрів призначений для полегшення пошуку інформації.

Код основних модулів поданий у додатку Б.

### 3.3 Інструкція користувача

Інструкція користувача це документ, який відноситься до пакету експлуатаційної документації. Основна мета цього документу це забезпечення користувача необхідною інформацією для самостійної роботи з програмою або автоматизованою системою.

Головна сторінка довідково-інформаційної системи туристичної фірми, реалізована у вигляді веб-додатку MyTour, наведена на рисунку 3.3 – 3.4. Клієнту туристичної фірми надається абсолютно простий та дружній інтерфейс, робота з яким не потребує будь-яких пояснень. Все інтуїтивно зрозуміло і можна користуватися додатком без жодних інструкцій. При чому користуватися розробленим додатком можна як з персонального комп'ютера, так і з іншого гаджету, наприклад телефону або планшету.

Якщо клієнт знайшов тур, який його зацікавив, надсилається запит до туристичної фірми і клієнт автоматично додається до бази клієнтів фірми.

Адміністративна частина додатку є значно складнішою і потребує інструкцій при використанні. Адміністративна частина додатку складається з двох частин: панелі меню (зліва) і робочої області (справа) (рис.3.5).

									ДППЗ.170122.01.18.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата						53



#	Готель	Тип номеру	Харчування	Заїзд	Вартість
<input type="checkbox"/>	Tirana Dahab Resort 5*	Double	Все включено	з 19.07 по 26.07 (7 ночей)	28 350 грн.

Електронна адреса  Телефон

Рисунок 3.3 – Вигляд головної сторінки довідково-інформаційної системи туристичної фірми MyTour на персональному комп'ютері

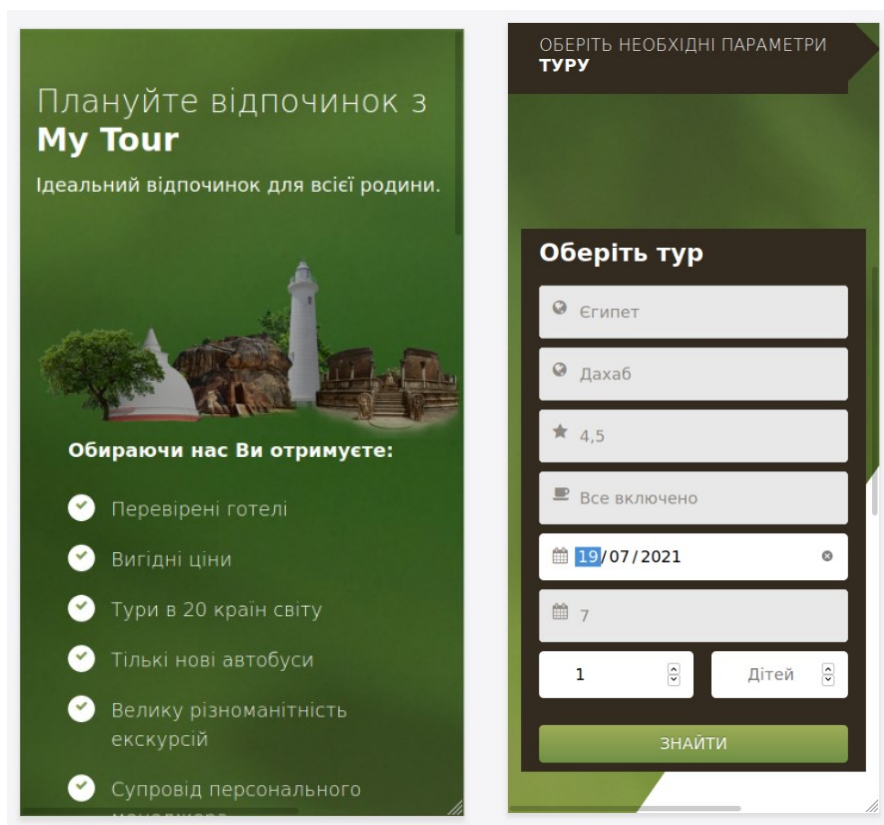


Рисунок 3.4 – Вигляд головної сторінки довідково-інформаційної системи туристичної фірми MyTour на смартфоні

Додати користувача

Статус:  Акт./Не акт.

**Основна інформація**

Електронна пошта\*  Ім'я\*  Прізвище\*  Стать  Фото

Дата народження  Телефон  Адреса  Поштовий код  Місто  Країна

**Доступ**

Логін\*  Пароль\*  Тип

Рисунок 3.5 – Вікно адміністративної частини додатку. Форма додавання нового користувача

На панелі меню, наведені всі функції, які можна виконувати в системі. Панель меню включає три розділи роботи з інформаційно-довідковою системою: Тури, Клієнти і Адміністрування.

Розділ Тури призначений для роботи безпосередньо з турами, які туристична фірма пропонує клієнтам. В розділі Тури можна виконувати наступні дії:

- переглядати існуючі тури;
- додавати нові тури;
- переглядати існуючі готелі;
- додавати нові готелі;
- переглядати сформовані заявки;
- переглядати існуючі курорти;
- додавати нові курорти;
- переглядати типи номерів;
- додавати типи номерів;

- переглядати можливі варіанти організації харчування в турі;
- додавати нові варіанти організації харчування;
- здійснювати повернення коштів для турів, що не відбулися внаслідок певних причин.

На рисунках 3.6-3.12 представлені деякі вікна розділу Тури додатку My Tour. На рисунку 3.6 наведена форма додавання готелю. Через форму вносить наступна інформація: Назва готелю, Адреса, Поштовий код, Місто, Країна, Курорт, Зірковість, Фото готелю, реквізити контактної особи. А також, встановлюється статус готелю: активний/не активний.

**Додати готель**

Статус:  Акт./Не акт.

**Загальна інформація**

Назва\*      Адреса      Поштовий код      Місто      Країна      Курорт      Зірковість      Зображення

**Контактна особа**

Ім'я      Посада      Телефон      Ел. пошта\*

Рисунок 3.6 – Форма додавання готелю

На рисунку 3.7 наведено вікно «Заявки». Тут можна переглядати інформацію про сформовані заявки.

**Заявки**

Дні:      

ID	Назва	Тип	Тип номеру	Вартість	Тур	Клієнт	Осіб	Оплата	Створена
4	Запит 1	Авіатур	lux	37000 грн.	Екскурсійний	Степан Майріяко	4	TAK	March 12, 2021
6	Запит 3	Лайнер	single	19340 грн.	Екскурсійний	Степан Майріяко	1	TAK	March 22, 2021
7	Запит 4	Лайнер	double	27800 грн.	Екскурсійний	Степан Майріяко	2	TAK	March 22, 2021

Рисунок 3.7 – Вікно заявок на тур

На рисунку 3.8 наведено вікно форми додавання курорту. Тут вноситься інформація про курорти і вибирається країна до якої відноситься курорт.

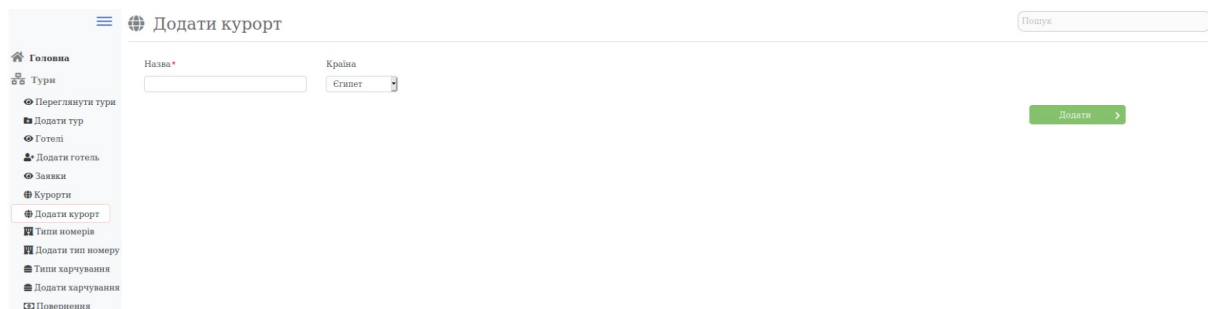


Рисунок 3.8 – Форма додавання курорту

На рисунку 3.9 наведено вікно форми додавання номеру. Через форму заноситься інформація про типи номеру в готелі.

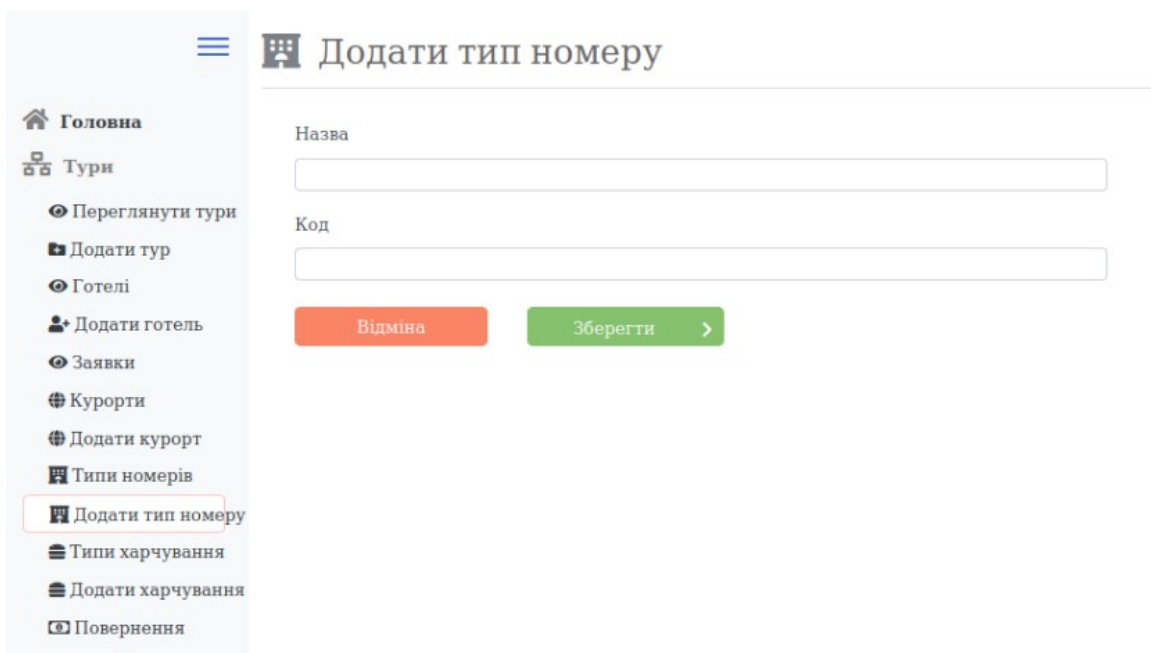


Рисунок 3.9 – Форма додавання типу номеру

На рисунку 3.10 наведено вікно «Типи харчування». Тут можна переглядати інформацію про типи харчування доступні в готелі.

									Арк.
									57
Змн.	Арк.	№ докум.	Підпис	Дата					

ID	Назва	Базова ціна	Знижка на дитину	Тип
1	Тільки сніданок	20.00	15.00%	ВВ
2	Сніданок та вечеря	50.00	17.00%	НВ

Рисунок 3.10 – Вікно перегляду типів харчування

На рисунку 3.11 наведено вікно форми додавання типів харчування. Через зазначену форму додається тип харчування.

Рисунок 3.11 – Форма додавання харчування

На рисунку 3.12 наведено вікно перегляду повернень проплат за тур. Тут можна переглядати інформацію про повернення коштів клієнту, в разі якщо тур не відбувся.

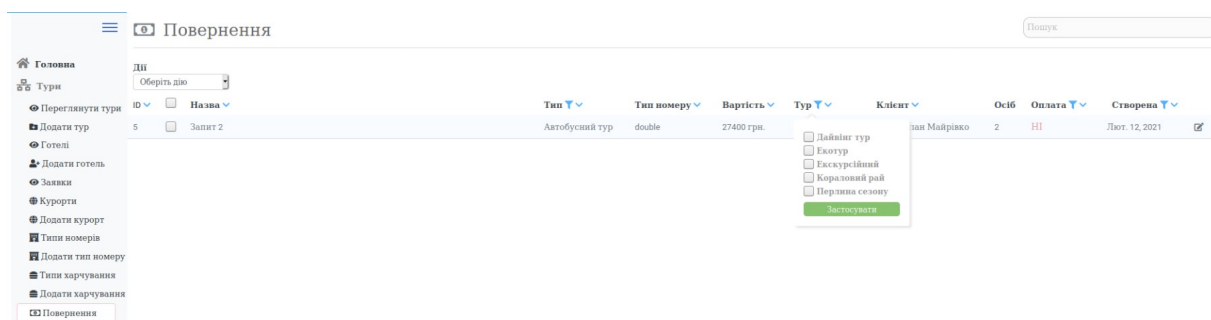


Рисунок 3.12 – Вікно перегляду повернень проплат за тур

Розділ Клієнти призначений для формування бази клієнтів туристичної фірми. В розділі Клієнти можна виконувати наступні дії:

- переглядати зареєстрованих клієнтів;
- додавати нових клієнтів.

При формуванні заявки на тур клієнт потрапляє до бази клієнтів. Вікно перегляду бази клієнтів зображено на рисунку 3.13.

Розділ Налаштування призначений для виконання загальних налаштувань, які не увійшли до попередніх двох розділів.

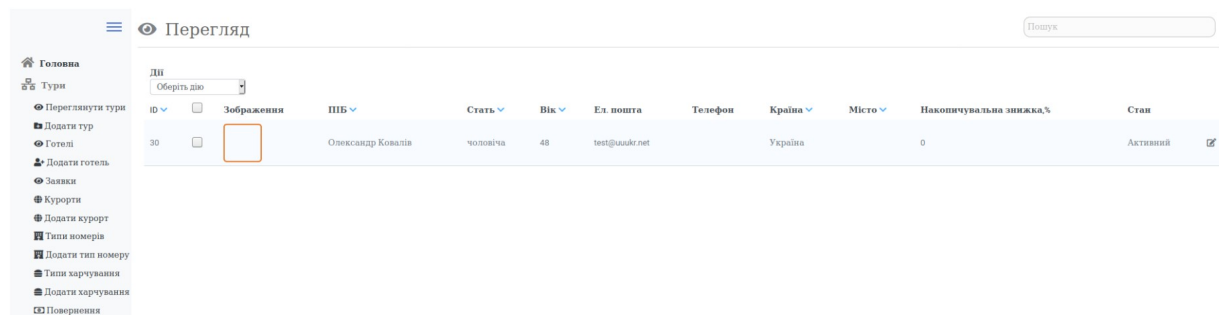


Рисунок 3.13 – Вікно перегляду клієнтів, що знаходяться в базі клієнтів

В розділі Налаштування можна виконувати наступні дії:

- налаштовувати категорії;
- налаштовувати статуси;
- налаштовувати зірковість готелю;
- здійснювати загальні налаштування (мова, валюта, країни тощо);
- переглядати користувачів довідково-інформаційної системи;

– додавати користувачів.

На рисунках 3.14 – 3.17 наведені деякі вікна розділу Налаштування. На рисунку 3.14 – вікно додавання статусів готелів.

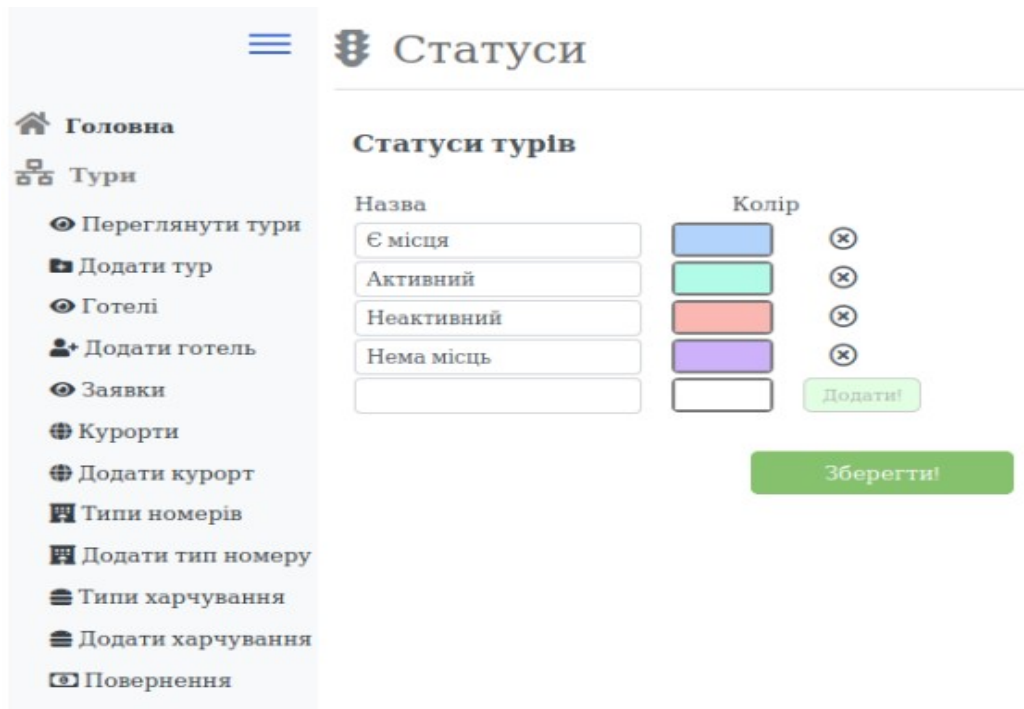


Рисунок 3.14 – Вікно додавання статусів

На рисунку 3.15 представлено вікно редагування класу готелю.

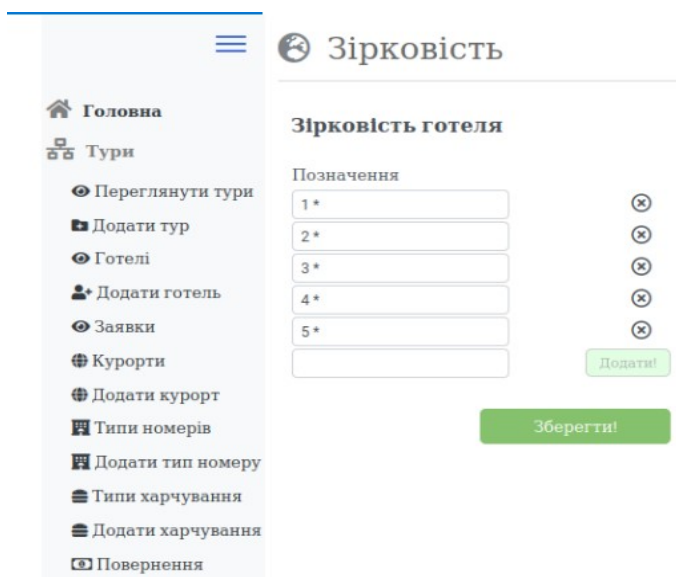


Рисунок 3.15 – Вікно редагування класу готелю

На вкладці Налаштування здійснюються необхідні для користувача додатку налаштування довідково-інформаційної системи (рисунки 3.16).

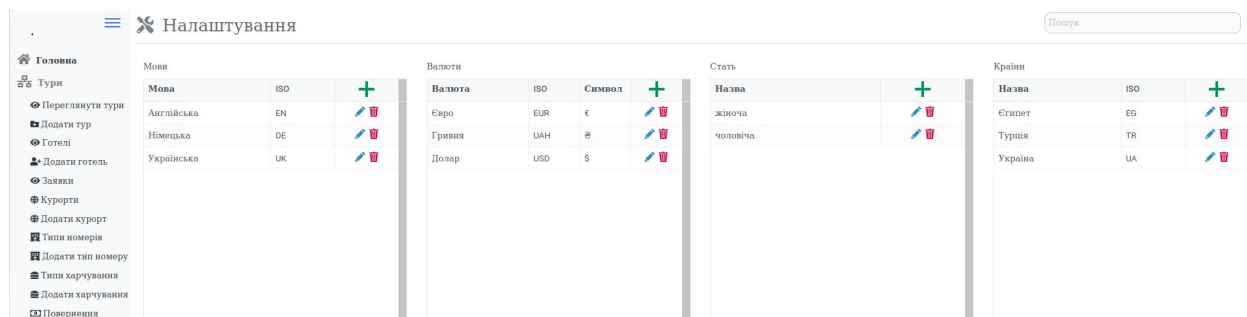


Рисунок 3.16– Вікно Налаштування

Для того, щоб користувач міг працювати з інформаційною-довідковою системою він має бути зареєстрованим. Для цього використовується форма додавання нового користувача (рисунки 3.5). Після реєстрації нового користувача подальша робота з додатком здійснюється тільки після авторизації. Всі зареєстровані в системі користувачі, відображаються у вікні користувачі (рисунки 3.17).

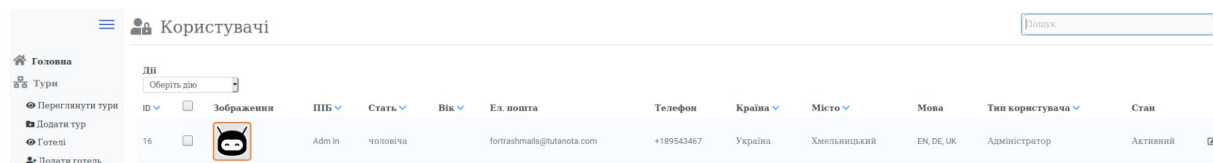


Рисунок 3.17 – Вікно перегляду користувачів додатку

Після розробки інструкції користувача, сформуємо вимоги до програмно-апаратних засобів довідково-інформаційної системи My Tour.

### 3.4 Вимоги до програмно-апаратних засобів

Довідково-інформаційна система туристичної фірми реалізована у вигляді

веб-додатку і має наступні вимоги до програмно-апаратних засобів.

Програмне забезпечення на стороні серверу:

- Apache 1.3 / 2 або IIS 6 та вище;
- PHP 5.6 та вище;
- MySQL 5.0 та вище .

Апаратні вимоги до серверу:

- ширина каналу Internet від 256 кб/сек;
- розмір оперативної пам'яті від 1 Гб;
- мінімальних обсяг вільного місця на жорсткому диску – 400 Мб.

Вимоги до програмного забезпечення клієнтської частини довідково-інформаційної системи туристичної фірми «MyTour»:

- наявність веб-браузера (Google Chrome, Mozilla Firefox, Opera тощо);
- підтримка відображення за стандартом HTML5;
- підтримка каскадних таблиць стилів стандарту CSS3.

Отже, враховуючи вимоги вибраної СКБД, була побудована фізична модель бази даних, визначені та реалізовані програмні модулі довідково-інформаційної системи, розроблена інструкція користувача для майбутніх користувачів додатку.

									Арк.
									62
Змн.	Арк.	№ докум.	Підпис	Дата					

ДППЗ.170122.01.18.ПЗ

## 4 НАЛАГОДЖЕННЯ І ТЕСТУВАННЯ СИСТЕМИ

### 4.1 Вибір та обґрунтування методів тестування системи

Тестування – це процес оцінки системи або її компонентів з метою з'ясування чи задовольняє вона визначеним вимогам. Простіше кажучи, тестування – це запуск системи та виконання її функцій з метою виявлення прогалин, помилок або відсутніх вимог, що суперечать фактичним вимогам.

Відповідно до стандарту ANSI / IEEE 1059 тестування можна визначити як – процес аналізу елемента програмного забезпечення для виявлення відмінностей між існуючими і необхідними умовами (тобто дефектами / помилками) і для оцінки характеристик елемента програмного забезпечення.

Зазвичай в ІТ-фірмі, тестування проводить команда, яка відповідає за оцінку розробленого програмного забезпечення в контексті заданих вимог. Більш того, розробники також проводять тестування, яке називається Unit Testing. У більшості випадків такі фахівці беруть участь в тестуванні системи в рамках своїх відповідних можливостей. Це і тестер програмного забезпечення, і розробник програмного забезпечення, і керівник проекту/менеджер, і кінцевий користувач [11].

Різні компанії по різному називають людей, які тестують програмне забезпечення на основі свого досвіду і знань це може бути Software Tester, Software Quality Assurance Engineer, QA Analyst тощо.

В процесі тестування відбувається валідація та верифікація довідково-інформаційної системи. Різниця між цими двома термінами наведена в таблиці 4.1.

Типи тестування. Розрізняють ручне та автоматизоване тестування. Ручне тестування включає в себе тестування програмного забезпечення вручну, тобто без використання будь-якого автоматизованого інструменту або будь-якого сценарію. При використанні такого типу тестування, тестер виконує роль кінцевого користувача і тестує програмне забезпечення, щоб виявити непередбачувану поведінку системи або помилки. До етапів ручного

									Арк.
									63
Змн.	Арк.	№ докум.	Підпис	Дата					

тестування відноситься до тестування окремих модулів, інтеграційне тестування, тестування системи та тестування для прийому користувачем.

Таблиця 4.1 – Відмінності між верифікацією та валідацією

Верифікація	Валідація
Визначає чи правильно розробляється система	Визначає чи розробляється правильна система
Гарантує, що система програмного забезпечення відповідає всім функціональним можливостям.	Гарантує, що функціональні можливості відповідають передбачуваній поведінці
Перевірка виконується в першу чергу і включає перевірку документації, коду тощо	Перевірка відбувається після розробки і в основному включає перевірку всієї системи
Виконується розробниками	Виконується тестувальниками
Це статична активність, оскільки включає в себе збирання відгуків, покрокове керівництво, перевірку програмного забезпечення	Це динамічна активність, оскільки включає в себе виконання програмного забезпечення у відповідності до вимог
Це об'єктивний процес і для перевірки програмного забезпечення не вимагаються суб'єктивні рішення	Це суб'єктивний процес, який включає суб'єктивні рішення

Автоматизоване тестування, також відоме як Test Automation – це коли тестувальник пише сценарії і використовує інше програмне забезпечення для тестування продукту. Цей процес включає в себе автоматизацію ручного процесу. Автоматизоване тестування використовується для повторного запуску тестових сценаріїв, які виконувалися вручну, швидко і багаторазово.

Методи тестування. Розрізняють методи тестування чорного ящика, білого ящика і сірого ящика.

Тестування чорного ящика це техніка тестування, яка не вимагає будь-яких знань про внутрішню роботу додатка. Тестер не звертає уваги на

архітектуру системи і не має доступу до вихідного коду. Як правило, при виконанні тесту чорного ящика тестер взаємодіє з призначеним для користувача інтерфейсом системи, надаючи вхідні дані і аналізуючи вихідні дані, не знаючи, як і де обробляються вхідні дані.

Тестування білого ящика – це детальне дослідження внутрішньої логіки програми і структури коду. Тестування білого ящика також називається тестуванням скла або тестуванням відкритого ящика. Щоб виконати тестування «білого ящика» додатка, тестувальник повинен знати внутрішню роботу коду. Тестувальник повинен зазирнути в вихідний код і з'ясувати, який блок / блок коду поводить неадекватно.

Тестування «сірого ящика» – це метод тестування програми з обмеженими знаннями про внутрішню роботу додатка. На відміну від тестування «чорного ящика», коли тестувальник перевіряє тільки призначений для користувача інтерфейс програми; при тестуванні у вигляді сірого ящика тестер має доступ до проектної документації та бази даних. Володіючи цими знаннями, тестувальник може підготувати кращі тестові дані і тестові сценарії при складанні плану тестування [11].

Для тестування довідково-інформаційної системи туристичної фірми «MyTour» будемо застосовувати типи ручного та автоматизованого тестування, та метод тестування «Чорного ящика».

#### 4.2 Валідація та верифікація системи

Проведемо функціональне тестування довідково-інформаційної системи туристичної фірми MyTour. Для цього входимо на головну сторінку додатку MyTour, призначеного для клієнтів довідково-інформаційної системи туристичної фірми.

На головній сторінці додатку MyTour обираємо країну подорожі, курорт,

										Арк.
										65
Змн.	Арк.	№ докум.	Підпис	Дата						

готель потрібного рівня зірковості, тип харчування, дату початку відпочинку, тривалість туру, кількість дорослих та дітей.

При формування запиту були введені наступні дані (рис.4.1):

- країна – Єгипет;
- курорт – Дахаб;
- зірковість готелю – 4 або 5 зірок;
- тип харчування – все включено;
- дата початку відпочинку – 20.07.2021;
- тривалість туру – 7 ночей;
- кількість дорослих – 1 особа;
- кількість дітей – 1 особа.

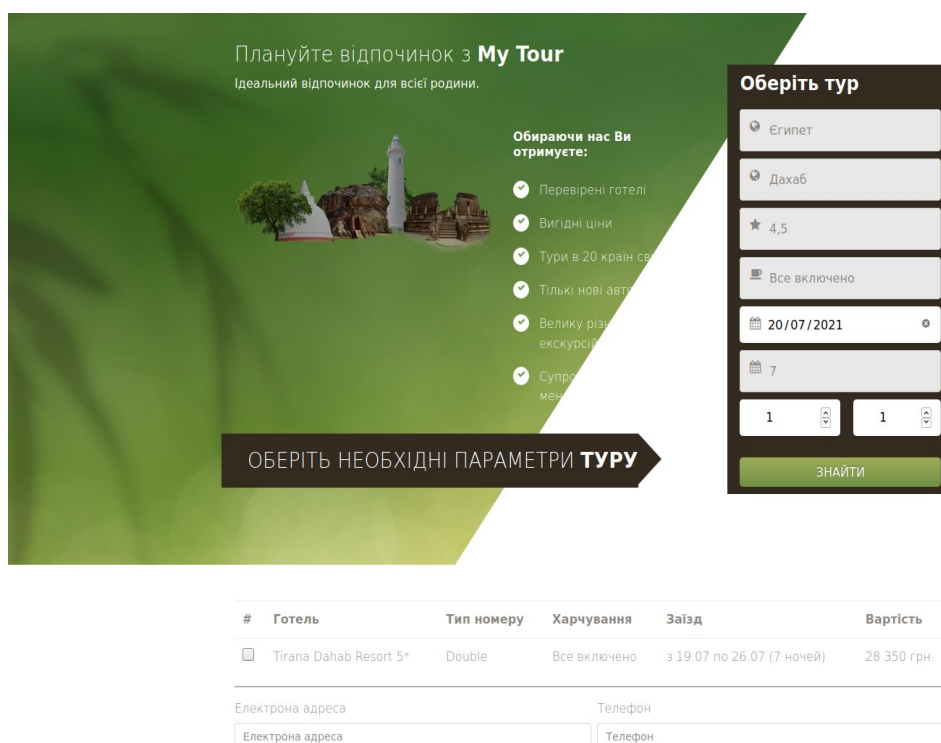


Рисунок 4.1 – Валідація роботи клієнтської частини довідково-інформаційної системи туристичної фірми MyTour

Довідково-інформаційна система MyTour запропонувала тур в готель Tirana Dahab Resort 5\*, двомісний номер типу Double з типом харчування все

									ДПШЗ.170122.01.18.ПЗ	Арк.
										66
Змн.	Арк.	№ докум.	Підпис	Дата						



На наведених рисунках адміністративної частини довідково-інформаційної системи MyTour ми бачимо, що вартість номеру типу Double з типом харчування все включено в готелі Tirana Dahab Resort 5\* складає 4500 грн. за добу на двох осіб. Знижка на дитину складає 20 %.

Порахуємо вартість туру. Номер типу Double для туру тривалістю з харчуванням все включено 7 днів буде коштувати  $4500 \cdot 7 = 31\,500$  грн. Це ціна на двох осіб. Якщо є дитина, то до дитини застосовується знижка 20%. Тоді, вартість туру буде рахуватися наступним способом:  $7 \cdot 2250 + 7 \cdot 2250 \cdot 0,8 = 15750 + 12600 = 28350$  грн.

Отже, дані видані інформаційно-довідковою системою у відповідь на запит клієнта та результати розрахунку згідно даних наведених в адміністративній частині додатку співпадають. Тому, можна стверджувати, що довідково-інформаційна система туристичної фірми MyTour працює коректно.

#### 4.3 Аналіз результатів тестування системи

Тестування навантаження довідково-інформаційної системи, яка реалізована у вигляді веб-додатку, є частиною тестування продуктивності. Його мета – перевірити надійність та продуктивність системи. Вузькі місця визначаються за допомогою важких сценаріїв стресового навантаження до того, як вони виникають у виробничому середовищі. Таким чином, випробування на навантаження не тільки забезпечує чудовий захист від низької продуктивності, але також враховує додаткові стратегії. Тестування навантаження додатку проводиться у вигляді автоматизованого тестування.

Це тип нефункціонального тестування в галузі програмної інженерії, який визначає, як поводить веб-додаток або веб-сайт при одночасному доступі кількох користувачів.

За допомогою тестування навантаження веб-додатку можна отримати

										ДППЗ.170122.01.18.ПЗ	Арк.
											68
Змн.	Арк.	№ докум.	Підпис	Дата							

наступні показники: максимальну працездатність веб-сайту; поточну інфраструктуру веб-сайтів достатню для роботи в реальному середовищі; стійкість веб-сайту або веб-програми щодо пікового навантаження користувачів; кількість користувачів, які одночасно можуть користуватися програмою чи сайтом без втрати продуктивності; функцію масштабованості, щоб забезпечити більшу доступність для користувачів.

Проведення тестування навантаження веб-додатку під час його розробки має на меті виявити можливі проблеми до введення додатку в експлуатацію. Під час тестування навантаження виконуються наступні заміри: вимірювання часу відгуку для кожної транзакції; аудит продуктивності компонентів системи під різними навантаженнями; оцінка продуктивності компонентів бази даних при різних навантаженнях; обчислення часу відгуку між клієнтом та сервером; перевірка проблем з конфігурацією сервера на веб-сервері, сервері додатків, сервері баз даних тощо; визначення проблем з апаратними обмеженнями, такими як максимізація процесора, обмеження пам'яті, вузькі місця в мережі тощо. Результати тестування навантаження довідково-інформаційної системи туристичної фірми MyTour наведені на рисунку 4.4.

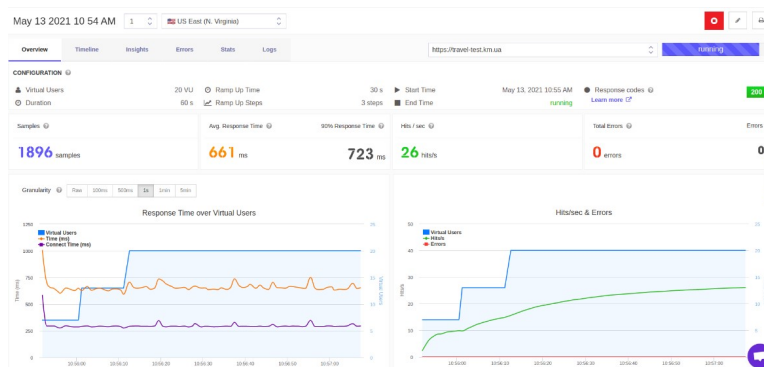


Рисунок 4.4 – Тестування навантаження довідково-інформаційної системи туристичної фірми MyTour

Як видно з наведених даних, результати тестування навантаження є задовільними і додаток готовий до використання.

										Арк.
										69
Змн.	Арк.	№ докум.	Підпис	Дата						

## ВИСНОВКИ

В результаті виконаної під час дипломного проектування роботи можна зробити наступні висновки.

У першому розділі був зроблений детальний аналіз предметної області, визначені її структурні та функціональні особливості. Було досліджено роботу суб'єктів ринку туристичних послуг, проаналізовані бізнес-процеси туристичних фірм. Вивчені потреби туристичних фірм в програмному забезпеченні, яке б надавало можливість автоматизувати бізнес процеси та будувати відношення з клієнтами. На основі проведених досліджень були сформовані функціональні та нефункціональні вимоги до програмного забезпечення та підготовлене технічне завдання на розробку довідково-інформаційної системи «My Tour».

Другий розділ дипломного проекту присвячений проектуванню довідкової-інформаційної системи туристичної фірми. Було проведено концептуальне моделювання діяльності туристичної фірми, розроблена структура програмного забезпечення, побудована логічна модель бази даних та вибрані технології та методи реалізації програмного забезпечення.

Третій розділ дипломного проекту присвячений безпосередньо програмній реалізації. Враховуючи вимоги вибраної СКБД, була побудована фізична модель бази даних, визначені та реалізовані програмні модулі довідково-інформаційної системи, розроблена інструкція користувача для майбутніх користувачів додатку.

В четвертому розділі дипломного проекту здійснювалася валідація та верифікація роботи довідково-інформаційної системи. Проведене функціональне та нефункціональне тестування підтвердило достатню якість розробленого програмного забезпечення. Розроблена довідково-інформаційна система My Tour може бути рекомендована до використання на малих та середніх підприємствах туристичної галузі.

										Арк.
										70
Змн.	Арк.	№ докум.	Підпис	Дата	ДППЗ.170122.01.18.ПЗ					

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гаврилов В.П. Інформаційні системи і технології в туризмі : навчальний посібник для студентів напряму підготовки 6.140103 "Туризм" / В. П. Гаврилов. – Харків : ХНЕУ ім. С. Кузнеця, 2016. – 168 с.
2. Сокол Т. Г. Основи туристичної діяльності: Підручник / Під ред. доктора пед. наук, проф. Орлова В. Ф. – К.: Грамота, 2004. – 327 с.
3. Мельниченко С. В. Управління бізнес-процесами в туризмі : монографія / С.В. Мельниченко, К.А. Шеєнкова. – К. : Київ. нац. торг.- екон. ун-т, 2015. – 264 с.
4. Цуканова О. А. Методология и инструментарий моделирования бизнес-процессов: учебное пособие / О. А. Цуканова – СПб.: Университет ИТМО, 2015. – 100 с.
5. Буч Г. Мова UML: Керівництво користувача / Г. Буч, Д. Рамбо, А. Джекобсон : пер.с англ. - М.: ДМК, 2000 - 432 с.: іл.
6. Лукьянов М. Ю. PHP. Полное руководство и справочник функций / М. Ю. Лукьянов. – Наука и Техника, 2020. – 432 с.
7. Что такое PHP? [Електронний ресурс] / Портал «Freehost.ua» – Режим доступу: <https://freehost.com.ua/faq/wiki/chto-takoe-php/>.
8. Артеменко Ю. Н. MySQL. Руководство администратора. / Под редакцией Артеменко Ю. Н.: Пер. с англ. — М. : Издательский дом "Вильямс", 2005. – 624 с.
9. Достоинства и недостатки MySQL [Електронний ресурс] / Портал «Все о гаджетах». – Режим доступу: <https://mod-book.ru/info/mysql-dostoinstva-i-nedostatki/>.
10. Bootstrap – фреймворк для адаптивной разработки [Електронний ресурс] / Портал «Redline» – Режим доступу: <https://redline.by/novosti/bootstrap.html>.

										ДППЗ.170122.01.18.ПЗ	Арк.
											71
Змн.	Арк.	№ докум.	Підпис	Дата							

11. Тестирование программного обеспечения (Обзор) [Электронный ресурс] / Портал «CoderLessons.com – уроки по программированию, DevOps и другим ИТ-технологиям» – Режим доступа: <https://coderlessons.com/tutorials/kachestvo-programmnogo-obespecheniia/izuchite-testirovanie-programmnogo-obespecheniia/testirovanie-programmnogo-obespecheniia-kratkoe-rukovodstvo> – Назва з екрану.

12. Бедратюк Л. П. Дипломний проект : методичні вказівки щодо його виконання для студентів-бакалаврів спеціальності «Інженерія програмного забезпечення» / Л. П. Бедратюк, Г. І. Радельчук, Ю. В. Форкун, О. М. Ящина. – Хмельницький : ХНУ, 2020. – 79 с.

									ДПШЗ.170122.01.18.ПЗ	Арк.
										72
Змн.	Арк.	№ докум.	Підпис	Дата						

ДОДАТОК А  
(обов'язковий)

**ТЕХНІЧНЕ ЗАВДАННЯ**

## **Введення**

Робота виконується в рамках проекту розробки довідково-інформаційної системи туристичної фірми

### **1 Підстава для розробки**

Підстави для виконання робіт, терміни виконання:

Розробка програмного забезпечення ведеться на підставі виконання дипломного проекту. Довідково-інформаційна система туристичної фірми «My Tour» повинна бути розроблена до 01.05.2021 року.

Повне найменування програмного забезпечення – «Довідково-інформаційна система туристичної фірми «My Tour».

### **2 Призначення розробки**

Призначення довідково-інформаційної системи туристичної фірми «My Tour»:

- програма для конструювання турів та прорахунку вартості відпочинку;
- відображення облікової інформації щодо клієнтів, замовлень, продажів, повернень турів.

Завдання, які вирішуються:

- пошук турів, готелів, номерів, типів харчування, додаткових послуг;
  - облік клієнтів, замовлень, продажів, повернень турів;
- ведення аналітики, статистичний аналіз.

### **3 Вимоги до програми**

#### **3.1 Вимоги до функціональних характеристик**

Сайт повинен забезпечувати наступні функції:

- пошук турів;
- пошук готелів;
- вибір номеру;
- вибір режиму харчування;
- підбір додаткових послуг;
- створення турпакетів з різних набрів послуг, розрахування ціни;

- облік замовлень;
- облік клієнтів;
- облік продаж;
- облік повернень;
- визначення продуктивності співробітників туристичної фірми;
- формування підсумкових звітів і списків;
- статистичний аналіз.

### **3.2 Вимоги до надійності**

Надійність системи повинна бути забезпечена за такими напрямками:

- забезпечення працездатності компонентів програмно-технічної платформи;
- збереження даних.

### **3.3 Умови експлуатації**

Для введення в експлуатацію довідково-інформаційної системи потрібно: здійснити інсталяцію програмного забезпечення, провести тестування готового продукту за різними критеріями користувачами.

При наявності зауважень або недопрацювань, складається документ «Виправлення і зауваження», де вказуються недоліки програмного забезпечення. При задовільному результаті тестування програмне забезпечення буде випущено у реліз, та опубліковано на маркеті.

### **3.4 Вимоги до інформаційної та програмної сумісності**

Парольні політики для адміністраторів мають визначатись у вигляді налаштувань і автоматично контролюватись системою керування контенту. Система має бути захищена від найбільш поширених типів атак, наприклад SQL injection, XSS, отримання доступу методом перебирання паролів, тощо.

### **3.5 Спеціальні вимоги**

Веб-додаток має бути адаптивним і коректно масштабуватися на екранах різного розміру. Система має бути адаптованою для використання в усіх сучасних браузерах.

#### 4 Вимоги до програмної документації

Комплект супроводжувальної документації включає:

- загальний опис довідково-інформаційної системи;
- інструкція користувача.

#### 5 Стадії та етапи розробки

Стадії та етапи розробки проекту Довідково-інформаційної системи туристичної фірми наведено в таблиці А.1.

Таблиця А.1 – Стадії та етапи розробки проекту

Стадія розробки	Етапи робіт	Зміст робіт
Технічне завдання 02.01.21 – 31.01.21	Обґрунтування необхідності розробки програми	Коротка характеристика програмного забезпечення; підстава і призначення розробки; вимоги до програмної системи і документація; стадії і етапи розробки програми; порядок контролю і приймання
Ескізний проект 01.02.21 – 14.02.21	Розробка ескізного проекту	Попередня розробка структури вхідних і вихідних даних; уточнення середовища програмування; розробка і опис загальної алгоритмічної структури системи, що буде розроблюватися
Технічний проект 15.02.21 – 28.02.21	Розробка технічного проекту	Уточнення структури вхідних і вихідних даних; розробка докладного алгоритму; розробка структури програми; остаточне визначення конфігурації технічних засобів
Робочий проект 01.03.21 – 10.04.21	Розробка програмного забезпечення	Реалізація програмного забезпечення; відладка; проведення попереднього тестування
Розробка програмної документації 11.04.21 – 20.04.21	Розробка документації до програмного забезпечення	Розробка необхідної документації, передбаченої технічним завданням
Тестування системи 21.04.21 – 30.04.21	Проведення тестування програмного забезпечення	Розробка методики тестування; проведення основних тестів; коректування програмного забезпечення
Впровадження	Підготовка і передача програми	Підготовка і передача програмного забезпечення; навчання персоналу використуванню програмного забезпечення; внесення коректувань в програмне забезпечення і документацію

## **6 Порядок контролю та приймання**

Контроль виконання робіт та роботи довідково-інформаційної системи туристичної фірми здійснюватиметься розробником на основі тестування програмного забезпечення. При виявленні недоліків інформаційно-довідкова системи буде допрацьована. Прийняття довідково-інформаційної системи туристичної фірми «My Tour» буде здійснюватися державною екзаменаційною комісією.

ДОДАТОК Б  
(обов'язковий)

**ПРОГРАМНИЙ КОД ОСНОВНИХ МОДУЛІВ ДОВІДКОВО-  
ІНФОРМАЦІЙНОЇ СИСТЕМИ**

```
<?php

function section($name, $emptyFallback = '')
{
    echo app()->view()->getInstance()->section($name, $emptyFallback);
}
function extend($template, array $data = [])
{
    echo app()->view()->getInstance()->extend($template, $data);
}
function block($name)
{
    app()->view()->getInstance()->start($name);
}
function endblock($appendMode = 1)
{
    app()->view()->getInstance()->end($appendMode);
}
function insert($template, array $data = [])
{
    $view = app()->view->getInstance();
    echo $view->render($template, $data);
}
function has_template($template)
{
    $view = app()->view->getInstance();
    return $view->exists($template);
}
function register_templates($namespace, $directory)
{
    $view = app()->view->getInstance();
    return $view->addFolder($namespace, $directory);
}
function sp_template_meta($key, $fallback = false)
{
    return app()->template->meta($key, $fallback);
}
function svg_icon($id, $class = '', array $attrs = [])
{
    $id = e_attr($id);
    $class = 'svg-icon ' . e_attr($class);
    $attrTxt = '';

    foreach ($attrs as $key => $value) {
        if ($key === 'class') {
            continue;
        }

        $attrTxt .= ' ' . e_attr($key) . '=' . e_attr($value) . ' ';
    }
}
```

```

    }
    return '<svg class="' . $class . '" . $attrTxt . '><use xlink:href="#"' . $id
. '"/></svg>';
}
function sp_footer()
{
    do_action('sp.footer_before');
    echo app()->asset->renderHtml(2);
    if (is_frontend()) {
        echo get_option('footer_scripts');
    }

    do_action('sp.footer_after');
}
function sp_register_style($id, $url, array $args = [])
{
    return app()->asset->registerStyle($id, $url, $args);
}
function sp_register_script($id, $url, array $args = [])
{
    return app()->asset->registerScript($id, $url, $args);
}
function sp_enqueue_style($id, array $dependencies = [])
{
    return app()->asset->enQueue($id, null, $dependencies);
}
function sp_enqueue_script($id, $position = null, array $dependencies = [])
{
    return app()->asset->enQueue($id, $position, $dependencies);
}

function sp_dequeue_asset($id)
{
    return app()->asset->deQueue($id);
}
function sp_is_enqueued($id)
{
    return app()->asset->isEnqueued($id);
}
function sp_add_sidebar_menu($id, array $args, $parent = false, $insertAfter =
null)
{
    return MenuStorage::addSidebarMenu($id, $args, $parent, $insertAfter);
}
function sp_add_navbar_menu($id, array $args, $parent = false, $insertAfter =
null)
{
    return MenuStorage::addNavbarMenu($id, $args, $parent, $insertAfter);
}

function sp_render_sidebar_menu()
{
    $menu = new SidebarGenerator(MenuStorage::getSidebarMenus());
    return $menu->renderHtml();
}
function sp_render_tabs($parent, $classes = '')
{
    $menu = new SidebarGenerator(MenuStorage::getSidebarMenus());
    return $menu->renderTabs($parent, $classes);
}
function sp_render_navbar_menu()

```

```

{
    $menu = new NavbarGenerator(MenuStorage::getNavbarMenus());
    return $menu->renderHtml();
}
function sp_alert_flashes($prefix, $dismissable = true, $withIcon = true)
{
    $types = ['danger', 'warning', 'success', 'info', 'primary', 'secondary',
'light', 'dark'];
    $flash = app()->environment['slim.flash'];
    $dismiss = '';
    if ($dismissable) {
        $dismiss = "\n" . '<button type="button" class="close" data-
dismiss="alert" aria-label="Закрити">
<span aria-hidden="true">&times;</span>
</button>';
    }
    $html = '';
    $extraClass = '';

    if ($withIcon) {
        $extraClass = 'alert-icon';
    }
    foreach ($types as $type) {
        $key = $prefix . '-' . $type;
        $icon = '';

        if ($withIcon) {
            $icon = sp_svg_icon_for_alert($type);
        }
        if ($flash[$key]) {
            $html .= sp_bootstrap_alert($flash[$key], $type, $icon,
$dismissable) . "\n";
        }
    }

    return $html;
}
function sp_svg_icon_for_alert($type)
{
    switch ($type) {
        case 'danger':
        case 'warning':
            $svgID = 'warning';
            break;

        case 'success':
            $svgID = 'checkmark';
            break;
        default:
            $svgID = 'notifications';
            break;
    }
    return svg_icon($svgID, 'mr-2');
}
function sp_bootstrap_alert($message, $type = 'info', $iconHtml = null,
$dismissable = true)
{
    $class = '';

    if ($iconHtml) {
        $class = 'alert-icon';
    }
}

```

```

    }

    $dismiss = '';
    if ($dismissable) {
        $dismiss = "\n" . '<button type="button" class="close" data-
dismiss="alert" aria-label="' . __('Закрити') . '">
    <span aria-hidden="true">&times;</span>
    </button>';
    }

    return '<div class="alert alert-' . e_attr($type) . ' ' . $class . '"
role="alert">' . $dismiss . ' ' . $iconHtml . ' ' . $message . ' </div>';
}
function sp_die($msg, $httpStatus = 500)
{
    $app = app();
    $body = $app->view->fetch('admin::white_screen_of_dead.php', ['error' =>
$msg]);
    $app->halt($httpStatus, $body);
}
function sp_current_form_uri(array $ignore = [])
{
    return '?' . e_attr(request_build_query($ignore, null));
}
function register_template_options($TemplateName, $handlerCallback, $label =
null)
{
    $template = get_option('active_template');
    registry_store("template.{$template}__options_template", $TemplateName,
true);
    add_action("template.{$template}OnOptionsSubmit", $handlerCallback);
    if (!$label) {
        return true;
    }

    add_action('dashboard.controller_init_after', function () use ($template,
$label) {
        sp_add_sidebar_menu(
            "template-{$template}-options",
            [
                'type' => 'link',
                'url' => url_for('dashboard.settings.template', ['template' =>
$template]),
                'label' => $label,
            ],
            'settings'
        );
    });

    return true;
}
function template_has_options()
{
    $template = get_option('active_template');

    if (registry_read("template.{$template}__options_template")) {
        return true;
    }

    return false;
}

```

```

function get_template_locales($template = null, $forceScan = false)
{
    return app()->locale->getThemeLanguages($template, $forceScan);
}
function get_template_active_locale()
{
    return registry_read('_spark.template.locale', Locale::DEFAULT_LOCALE);
}

function get_template_active_locale_item()
{
    $locale = get_template_active_locale();

    return app()->locale->getThemeLocaleInfo($locale);
}
function current_template_uri($path = '')
{
    $templatePath = trailingslashit(get_option('active_template')) .
    untrailingslashit($path);
    return template_uri($templatePath);
}
function is_template_active_locale($locale)
{
    return $locale === registry_read('_spark.template.locale', 'en_US');
}
function has_custom_template($slug)
{
    $customTemplate = trailingslashit(config('custom_page_template_path',
    'pages')) . $slug . '.php';

    if (has_template($customTemplate)) {
        return $customTemplate;
    }

    return false;
}

function has_items($item)
{
    if (!is_array($item)) {
        return false;
    }

    return !empty($item);
}
<?php
class Tours Controller
{

    public function home()
    {
        $customerModel = new EngineModel;
        $customers = $customerModel->readMany(['customer_id', 'customer_name']);

        if ((int) get_option('use_search_as_default', 0)) {
            $data = [
                'default_customer'          => (int)
get_option('default_customer'),
                'customers'                  => $customers,
            ];

```

```

        return view('search-index.php', $data);
    }

    $app = app();
    $tourModel = new TourModel;
    $hotelModel = new HotelModel;

    $data = [];

    $hotelsTable = $hotelModel->getTable();
    $toursTable = $tourModel->getTable();
    $currentPage = (int) $app->request->get('page', 1);

    $filters['sort'] = 'tour-publish-date';

    $itemsPerPage = (int) get_option('latest_tours_count', 10);
    $totalCount = $tourModel->countRows(null, $filters);
    $pagination = new Pagination($totalCount, $currentPage, $itemsPerPage);
    $pagination->setUrl("?page=@id");

    $paginationHtml = $pagination->renderHtml();

    $offset = $pagination->offset();

    $fields[] = "{$toursTable}.tour_id, {$toursTable}.tour_title,
{$toursTable}.tour_excerpt, {$toursTable}.tour_featured_image,
{$toursTable}.tour_type, {$toursTable}.tour_source, {$toursTable}.tour_pubdate,
{$toursTable}.created_at, {$toursTable}.tour_hits";
    $fields[] = "{$hotelsTable}.hotel_name, {$hotelsTable}.hotel_id,
{$hotelsTable}.hotel_logo_url";

    $sql = $tourModel->select($fields)
->leftJoin(
    $hotelsTable,
    "{$toursTable}.tour_hotel_id",
    '=',
    "{$hotelsTable}.hotel_id"
)->limit($itemsPerPage, $offset);

    $sql = $tourModel->applyModelFilters($sql, $filters);
    $stmt = $sql->execute();

    $latestTours = $stmt->fetchAll();

    $maxSliderItems = get_option('max_slider_items', 10);
    $sliderTours = array_slice($latestTours, 0, $maxSliderItems);

    $data['is_home'] = true;
    $data['default_customer'] = (int)
get_option('default_customer');
    $data['customers'] = $customers;
    $data['home_active'] = 'active';

```

```

    $data['title_append_site_name'] = false;
    $data['latest_tours']           = $latestTours;
    $data['slider_tours']           = $sliderTours;
    $data['pagination_html']        = $paginationHtml;

    return view('home.php', $data);
}

public function list()
{
    $app = app();
    $tourModel = new TourModel;
    $hotelModel = new HotelModel;
    $customerModel = new EngineModel;

    $data = [
        'title' => __('list', _T),
        'meta.description' => __('list-description', _T),
        'body_class' => 'list',
        'searching_site' => false,
        'by_hotel' => false,
    ];

    $query = sp_strip_tags(trim($app->request->get('s')), true);
    $hotelID = (int) $app->request->get('hotel', 0);

    $hotelsTable = $hotelModel->getTable();
    $toursTable = $tourModel->getTable();

    $currentPage = (int) $app->request->get('page', 1);

    $filters['sort'] = 'tour-publish-date';

    if (mb_strlen($query) > 3) {
        $filters['where'][] = ["{$toursTable}.tour_title", 'LIKE',
"%$query%"];
        $filters['where'][] = ["{$toursTable}.tour_content", 'LIKE',
"%$query%", 'OR'];

        $data['site_search_query'] = $query;
        $data['searching_site'] = true;

        $data['page_heading'] = sprintf(__('search-results-for', _T),
$query);
        $data['title'] = sprintf(__('search-results-for-title', _T),
$query);
        $data['meta.description'] = sprintf(__('search-results-for-desc',
_T), $query);
        $data['body_class'] .= ' search-results';
    }

    if ($hotelID) {
        $hotel = $hotelModel->read($hotelID, ['hotel_id', 'hotel_name']);
        if ($hotel) {
            $filters['where'][] = ["{$toursTable}.tour_hotel_id", '=',
$hotel['hotel_id']];
            $data['by_hotel'] = true;
            $data['hotel'] = $hotel;
        }
    }
}

```

```

        $data['title'] = $hotel['hotel_name'];
        $data['page_heading'] = $hotel['hotel_name'];
    }
}

$itemsPerPage = (int) get_option('latest_tours_count', 10);

$totalCount = $tourModel->countRows(null, $filters);

$pageQuery = [];

if ($data['searching_site']) {
    $pageQuery['s'] = $query;
}

if ($data['by_hotel']) {
    $pageQuery['hotel'] = $hotelID;
}

$queryStr = http_build_query($pageQuery);

if (!empty($queryStr)) {
    $queryStr = "&{$queryStr}";
}

$pageination = new Pagination($totalCount, $currentPage, $itemsPerPage);
$pageination->setUrl("?page=@id@{$queryStr}");

$pageinationHtml = $pageination->renderHtml();

$offset = $pageination->offset();

$fields[] = "{$toursTable}.tour_id, {$toursTable}.tour_title,
{$toursTable}.tour_excerpt, {$toursTable}.tour_featured_image,
{$toursTable}.tour_type, {$toursTable}.tour_source, {$toursTable}.tour_pubdate,
{$toursTable}.created_at, {$toursTable}.tour_hits";
$fields[] = "{$hotelsTable}.hotel_name, {$hotelsTable}.hotel_id,
{$hotelsTable}.hotel_logo_url";

$sql = $tourModel->select($fields)
->leftJoin(
    $hotelsTable,
    "{$toursTable}.tour_hotel_id",
    '=',
    "{$hotelsTable}.hotel_id"
)->limit($itemsPerPage, $offset);

$sql = $tourModel->applyModelFilters($sql, $filters);
$stmt = $sql->execute();

$latestTours = $stmt->fetchAll();

$maxSliderItems = get_option('max_slider_items', 10);

```

```

$sliderTours = array_slice($latestTours, 0, $maxSliderItems);

$customers = $customerModel->readMany(['customer_id', 'customer_name']);

$data['default_customer'] = (int)
get_option('default_customer');
$data['customers'] = $customers;
$data['list_active'] = 'active';
$data['slider_tours'] = $sliderTours;
$data['latest_tours'] = $latestTours;
$data['pagination_html'] = $paginationHtml;

return view('list.php', $data);
}

public function category($slug)
{
    $app = app();
    $categoryModel = new CategoryModel;
    $tourModel = new TourModel;
    $hotelModel = new HotelModel;

    $hotelsTable = $hotelModel->getTable();
    $toursTable = $tourModel->getTable();

    $category = $categoryModel->select(['*'])
        ->where('category_slug', '=', $slug)
        ->execute()
        ->fetch();

    if (!$category) {
        return $app->notFound();
    }

    $currentPage = (int) $app->request->get('page', 1);

    $filters['sort'] = 'tour-publish-date';
    $filters['where'][] = ["{$toursTable}.tour_category_id", '=',
$category['category_id']];

    $totalCount = $tourModel->countRows(null, $filters);

    $itemsPerPage = (int) get_option('category_tours_count', 20);

    $pagination = new Pagination($totalCount, $currentPage, $itemsPerPage);
    $pagination->setUrl("?page=@id@");

    $paginationHtml = $pagination->renderHtml();

    $offset = $pagination->offset();

```

```

        $fields[] = "{$toursTable}.tour_id, {$toursTable}.tour_title,
{$toursTable}.tour_excerpt, {$toursTable}.tour_featured_image,
{$toursTable}.tour_type, {$toursTable}.tour_source, {$toursTable}.tour_pubdate,
{$toursTable}.created_at, {$toursTable}.tour_hits";
        $fields[] = "{$hotelsTable}.hotel_name, {$hotelsTable}.hotel_id,
{$hotelsTable}.hotel_logo_url";

        $sql = $tourModel->select($fields)
->leftJoin(
            $hotelsTable,
            "{$toursTable}.tour_hotel_id",
            '=',
            "{$hotelsTable}.hotel_id"
        )->limit($itemsPerPage, $offset);

        $sql = $tourModel->applyModelFilters($sql, $filters);
        $stmt = $sql->execute();

        $latestTours = $stmt->fetchAll();

        $title = __(
            "category-label-{$category['category_slug']}",
            _T,
            ['defaultValue' => $category['category_name']]
        );

        $data = [
            'title' => $title,
            'meta.description' => sprintf(
                __('category-meta-desc', _T),
                $category['category_name']
            ),
            "{$category['category_id']}_active" => 'active',
            'category' => $category,
            'latest_tours' => $latestTours,
            'pagination_html' => $paginationHtml,
        ];

        return view('category.php', $data);
    }

    public function readTourDetails($slug, $id)
    {
        $app = app();

        $categoryModel = new CategoryModel;
        $tourModel = new TourModel;
        $hotelModel = new HotelModel;

        $hotelsTable = $hotelModel->getTable();
        $toursTable = $tourModel->getTable();
        $categoriesTable = $categoryModel->getTable();

```

```

    $fields[] = "{$toursTable}.tour_id, {$toursTable}.tour_title,
{$toursTable}.tour_content, {$toursTable}.tour_excerpt,
{$toursTable}.tour_featured_image, {$toursTable}.tour_author,
{$toursTable}.tour_type, {$toursTable}.tour_source, {$toursTable}.tour_pubdate,
{$toursTable}.created_at, {$toursTable}.tour_category_id,
{$toursTable}.tour_hits";
    $fields[] = "{$hotelsTable}.hotel_name, {$hotelsTable}.hotel_id,
{$hotelsTable}.hotel_logo_url";
    $fields[] = "{$categoriesTable}.category_name,
{$categoriesTable}.category_id, {$categoriesTable}.category_slug";

    $tour = $tourModel->select($fields)
->leftJoin(
    $hotelsTable,
    "{$toursTable}.tour_hotel_id",
    '=',
    "{$hotelsTable}.hotel_id"
)
->leftJoin(
    $categoriesTable,
    "{$toursTable}.tour_category_id",
    '=',
    "{$categoriesTable}.category_id"
)
->where("{$toursTable}.tour_id", '=', $id)
->execute()
->fetch();

if (!$tour) {
    return $app->notFound();
}

$tourModel->db()->query("UPDATE {$toursTable}
SET tour_hits = tour_hits + 1
WHERE tour_id = {$id}");

$tour['tour_hits']++;

breadcrumb_add(
    'category',
    __(
        "category-label-{$tour['category_slug']}",
        _T,
        ['defaultValue' => $tour['category_name']]
    ),
    url_for('site.category', ['slug' => $tour['category_slug']])
);

breadcrumb_add('tour', $tour['tour_title']);

$relatedToursCount = (int) get_option('related_tours_count', 3);
$relatedTours = [];

```

```

if ($relatedToursCount) {
    $filters = [];
    $filters['sort'] = 'tour-publish-date';

    $fields = [];

    $fields[] = "{$toursTable}.tour_id, {$toursTable}.tour_title,
{$toursTable}.tour_excerpt, {$toursTable}.tour_featured_image,
{$toursTable}.tour_type, {$toursTable}.tour_hits, {$toursTable}.tour_source,
{$toursTable}.tour_pubdate, {$toursTable}.created_at";
    $fields[] = "{$hotelsTable}.hotel_name, {$hotelsTable}.hotel_id,
{$hotelsTable}.hotel_logo_url";

    $sql = $tourModel->select($fields)
->where('tour_category_id', '=', $tour['tour_category_id'])
->where('tour_id', '!=', $tour['tour_id'])
->leftJoin(
    $hotelsTable,
    "{$toursTable}.tour_hotel_id",
    '=',
    "{$hotelsTable}.hotel_id"
)->limit($relatedToursCount, 0);

    $sql = $tourModel->applyModelFilters($sql, $filters);
    $stmt = $sql->execute();

    $relatedTours = $stmt->fetchAll();
}

$data = [
    'title' => $tour['tour_title'],
    'meta.description' => $tour['tour_excerpt'],
    'meta.image' => feat_img_url($tour['tour_featured_image']),
    'tour' => $tour,
    "{$tour['tour_category_id']}_active" => 'active',
    'related_tours' => $relatedTours,
];
return view('single.php', $data);
}

public function thumbnail()
{
    load_functions(['thumb.php']);
}

public function notFound()
{
    $app = app();

    $app->response->setStatus(404);

    $data = [
        'title' => __('404 Сторінку Не знайдено', _T),
    ];
    return view('404.php', $data);
}

```

```

}

public function page($slugOrID)
{
    $app = app();
    $contentModel = new ContentModel;

    $filters = [];
    $filters['where'][] = ['content_slug', '=', $slugOrID, 'OR'];
    $filters['where'][] = ['content_type', '!=', ContentModel::TYPE_PAGE];

    $page = $contentModel->db()->select()
        ->from($contentModel->getTable())
        ->where('content_slug', '=', $slugOrID)
        ->orWhere('content_id', '=', $slugOrID)
        ->where('content_type', '=', 'page')
        ->limit(1, 0)
        ->execute()
        ->fetch();

    if (!$page) {
        return $app->notFound();
    }

    $customTemplate = has_custom_template($page['content_slug']);

    $template = 'page.php';

    if ($customTemplate) {
        $template = $customTemplate;
    }

    $method = strtolower($app->request->getMethod());

    breadcrumb_add('page', $page['content_title']);

    do_action("page.render_before", $page, $customTemplate);

    do_action("page.{$page['content_slug']}", $page, $customTemplate);

    $description = limit_string(sp_strip_tags($page['content_body'], true),
300, '');

    $data = [
        'title' => $page['content_title'],
        'meta.description' => $description,
        'page' => $page,
    ];

    return $app->render($template, $data);
}

```

```

public function handleContactForm()
{
    $app = app();
    $req = $app->request;

    $data = [
        'name' => $req->tour('name'),
        'email' => $req->tour('email'),
        'subject' => sp_strip_tags($req->tour('subject'), true),
        'message' => sp_strip_tags($req->tour('message')),
    ];

    $v = new Validator($data);

    $v->labels([
        'email' => __('E-Mail', _T),
        'subject' => __('Subject', _T),
        'name' => __('Name', _T),
        'message' => __('Message', _T),
    ])->rule('required', ['email', 'name', 'message'])
    ->rule('email', 'email')
    ->rule('lengthBetween', 'subject', 10, 200)
    ->rule('lengthBetween', 'message', 200, 5000);

    if (!$v->validate()) {
        $errors = sp_valitron_errors($v->errors());
        flash('pages-danger', $errors);
        sp_store_tour($data);
        return follow_referer_uri();
    }

    return follow_referer_uri();
}

public function runTasks()
{
    $app = app();
    $req = $app->request;
    $token = $req->get('token');

    if ($token !== get_option('spark_cron_job_token')) {
        return sp_not_permitted();
    }

    @ignore_user_abort(1);
    @set_time_limit(0);

    $hotelModel = new HotelModel;
    $tourModel = new TourModel;
    $rssImporter = new RssImporter;

    $hotels = $hotelModel->select()->where('hotel_auto_update', '=', 1)
    ->orderBy('hotel_priority', 'ASC')

```

```

->execute()
->fetchAll();

foreach ($hotels as $hotel) {

    DashboardHotelsController::importHotelTours($hotel, true);
}

$days = (int) get_option('auto_delete_tours_after');

if ($days > 0) {
    $olderThan = strtotime("-{$days} days");
    $tourModel->purgeOldTours($olderThan, [TourModel::TYPE_IMPORTED]);
}

do_action('cron_tasks');

logger()->info(': ' . date('d-m-Y h:i:s A'));
}

public function logOut()
{
    $auth = new Auth;
    $auth->logOut();

    return follow_redirect_to_uri(url_for('dashboard.account.signin'));
}

public function search()
{
    $app = app();
    $q = trim($app->request->get('q'));

    $customerModel = new EngineModel;

    $customers = $customerModel->readMany(['customer_id', 'customer_name'],
0, 100, ['sort' => 'oldest']);

    if (empty($q)) {
        if (has_template('search-index.php')) {
            $data = [
                'default_customer' => (int)
get_option('default_customer'),
                'customers' => $customers,
            ];
            return view('search-index.php', $data);
        }

        return redirect_to('site.home');
    }

    $customerID = (int) $app->request->get('customer');

    if (!$customerID) {
        $customerID = get_option('default_customer');
    }
}

```

```

    }

    $customer = $customerModel->read($customerID, ['customer_is_image',
'customer_cse_id', 'customer_id', 'customer_show_thumb']);

    if (!$customer) {
        return $app->notFound();
    }

    do_action('search.discovered', $q, $customer);

    $cseID = js_string($customer['customer_cse_id']);

    $locale = get_theme_active_locale();

    $script = <<<SCRIPT

    (function() {
        var cx = '{$cseID}';
        var gcse = document.createElement('script');
        gcse.type = 'text/javascript';

        gcse.dataset.cfasync = false;
        gcse.async = false;
        gcse.src = 'https:
var s = document.getElementsByTagName('script')[0];
s.parentNode.insertBefore(gcse, s);
    })();
SCRIPT;

    $isDefaultEngine = (int) get_option('default_customer') === (int)
$customerID;

    $instantAnswer = [];

    if ($isDefaultEngine) {
        $instantAnswer = (new InstantAnswer)->getAnswer($q);
    }

    $safeSearch = get_option('safesearch_status', 'off');
    $newWindow = (int) get_option('search_links_newwindow', 1);

    $target = '_self';

    if ($newWindow) {
        $target = '_blank';
    }

    $isImage = (int) $customer['customer_is_image'];

    $attrs = [
        'enableHistory' => true,
        'safeSearch' => $safeSearch,
        'linkTarget' => $target,
        'noResultsString' => sprintf(__('no-search-results', _T), $q),

```

```

];

$attrText = '';

if ($isImage) {
    $attrs['disableWebSearch'] = true;
    $attrs['enableImageSearch'] = true;
    $attrs['defaultToImageSearch'] = true;

    $attrs['mobileLayout'] = 'enabled';
    $attrs['imageSearchResultSetSize'] = (int)
get_option('search_items_count', 10);
} else {

    $attrs['mobileLayout'] = 'forced';
    $attrs['webSearchResultSetSize'] = (int)
get_option('search_items_count', 10);
}

foreach ($attrs as $key => $value) {
    $attrText .= " data-{$key}=\"{$value}\"";
}

$cseElement = '<div class="gcse-searchresults-only" ' . $attrText .
'></div>';

$data = [
    'title' => $q,
    'body_class' => 'search',
    'cse_script' => $script,
    'answer' => $instantAnswer,
    'q' => $q,
    'items_per_page' => (int) get_option('search_items_count', 10),
    'customers' => $customers,
    'customer' => $customer,
    'is_image' => $isImage,
    'cse_element' => $cseElement,
];
return view('search.php', $data);
}

public function hotelRedirect($id)
{
    $app = app();
    $tourModel = new TourModel;
    $table = $tourModel->getTable();
    $id = (int) $id;

    $tour = $tourModel->read($id, ['tour_type', 'tour_source']);

    if (empty($tour['tour_source'])) {
        return redirect_to('site.home');
    }
    $tourModel->db()->query("UPDATE {$table}
    SET tour_hits = tour_hits + 1
    WHERE tour_id = {$id}");
    return redirect($tour['tour_source']);
}

```

```

    }
}

function section($name, $emptyFallback = '')
{
    echo app()->view()->getInstance()->section($name, $emptyFallback);
}
function extend($template, array $data = [])
{
    echo app()->view()->getInstance()->extend($template, $data);
}
function block($name)
{
    app()->view()->getInstance()->start($name);
}
function endblock($appendMode = 1)
{
    app()->view()->getInstance()->end($appendMode);
}
function insert($template, array $data = [])
{
    $view = app()->view->getInstance();
    echo $view->render($template, $data);
}
function has_template($template)
{
    $view = app()->view->getInstance();
    return $view->exists($template);
}
function register_templates($namespace, $directory)
{
    $view = app()->view->getInstance();
    return $view->addFolder($namespace, $directory);
}
function sp_template_meta($key, $fallback = false)
{
    return app()->template->meta($key, $fallback);
}
function svg_icon($id, $class = '', array $attrs = [])
{
    $id = e_attr($id);
    $class = 'svg-icon ' . e_attr($class);
    $attrTxt = '';

    foreach ($attrs as $key => $value) {
        if ($key === 'class') {
            continue;
        }

        $attrTxt .= ' ' . e_attr($key) . '=' . e_attr($value) . ' ';
    }
    return '<svg class="' . $class . '" ' . $attrTxt . '><use xlink:href="#" ' . $id
    . '"/></svg>';
}
function sp_footer()
{
    do_action('sp.footer_before');
    echo app()->asset->renderHtml(2);
    if (is_frontend()) {

```

```

        echo get_option('footer_scripts');
    }
    do_action('sp.footer_after');
}
function sp_register_style($id, $url, array $args = [])
{
    return app()->asset->registerStyle($id, $url, $args);
}
function sp_register_script($id, $url, array $args = [])
{
    return app()->asset->registerScript($id, $url, $args);
}
function sp_enqueue_style($id, array $dependencies = [])
{
    return app()->asset->enqueue($id, null, $dependencies);
}
function sp_enqueue_script($id, $position = null, array $dependencies = [])
{
    return app()->asset->enqueue($id, $position, $dependencies);
}

function sp_dequeue_asset($id)
{
    return app()->asset->dequeue($id);
}
function sp_is_enqueued($id)
{
    return app()->asset->isEnqueued($id);
}
function sp_add_sidebar_menu($id, array $args, $parent = false, $insertAfter = null)
{
    return MenuStorage::addSidebarMenu($id, $args, $parent, $insertAfter);
}
function sp_add_navbar_menu($id, array $args, $parent = false, $insertAfter = null)
{
    return MenuStorage::addNavbarMenu($id, $args, $parent, $insertAfter);
}

function sp_render_sidebar_menu()
{
    $menu = new SidebarGenerator(MenuStorage::getSidebarMenus());
    return $menu->renderHtml();
}
function sp_render_tabs($parent, $classes = '')
{
    $menu = new SidebarGenerator(MenuStorage::getSidebarMenus());
    return $menu->renderTabs($parent, $classes);
}
function sp_render_navbar_menu()
{
    $menu = new NavbarGenerator(MenuStorage::getNavbarMenus());
    return $menu->renderHtml();
}
function sp_alert_flashes($prefix, $dismissable = true, $withIcon = true)
{
    $types = ['danger', 'warning', 'success', 'info', 'primary', 'secondary', 'light', 'dark'];
    $flash = app()->environment['slim.flash'];
    $dismiss = '';

```

```

    if ($dismissable) {
        $dismiss = "\n" . '<button type="button" class="close" data-
dismiss="alert" aria-label="Закрити">
        <span aria-hidden="true">&times;</span>
    </button>';
    }
    $html = '';
    $extraClass = '';

    if ($withIcon) {
        $extraClass = 'alert-icon';
    }
    foreach ($types as $type) {
        $key = $prefix . '-' . $type;
        $icon = '';

        if ($withIcon) {
            $icon = sp_svg_icon_for_alert($type);
        }
        if ($flash[$key]) {
            $html .= sp_bootstrap_alert($flash[$key], $type, $icon,
$dismissable) . "\n";
        }
    }

    return $html;
}
function sp_svg_icon_for_alert($type)
{
    switch ($type) {
        case 'danger':
        case 'warning':
            $svgID = 'warning';
            break;

        case 'success':
            $svgID = 'checkmark';
            break;
        default:
            $svgID = 'notifications';
            break;
    }
    return svg_icon($svgID, 'mr-2');
}
function sp_bootstrap_alert($message, $type = 'info', $iconHtml = null,
$dismissable = true)
{
    $class = '';

    if ($iconHtml) {
        $class = 'alert-icon';
    }

    $dismiss = '';
    if ($dismissable) {
        $dismiss = "\n" . '<button type="button" class="close" data-
dismiss="alert" aria-label="" . __('Закрити') . "'>
        <span aria-hidden="true">&times;</span>
    </button>';
    }
}

```

```

        return '<div class="alert alert-' . e_attr($type) . ' ' . $class . '"
role="alert">' . $dismiss . ' ' . $iconHtml . ' ' . $message . ' </div>';
    }
function sp_die($msg, $httpStatus = 500)
{
    $app = app();
    $body = $app->view->fetch('admin::white_screen_of_dead.php', ['error' =>
$msg]);
    $app->halt($httpStatus, $body);
}
function sp_current_form_uri(array $ignore = [])
{
    return '?' . e_attr(request_build_query($ignore, null));
}
function register_template_options($templateName, $handlerCallback, $label =
null)
{
    $template = get_option('active_template');
    registry_store("template.{$template}__options_template", $templateName,
true);
    add_action("template.{$template}OnOptionsSubmit", $handlerCallback);
    if (!$label) {
        return true;
    }

    add_action('dashboard.controller_init_after', function () use ($template,
$label) {
        sp_add_sidebar_menu(
            "template-{$template}-options",
            [
                'type' => 'link',
                'url' => url_for('dashboard.settings.template', ['template' =>
$template]),
                'label' => $label,
            ],
            'settings'
        );
    });

    return true;
}
function template_has_options()
{
    $template = get_option('active_template');

    if (registry_read("template.{$template}__options_template")) {
        return true;
    }

    return false;
}
function get_template_locales($template = null, $forceScan = false)
{
    return app()->locale->getThemeLanguages($template, $forceScan);
}
function get_template_active_locale()
{
    return registry_read('_spark.template.locale', Locale::DEFAULT_LOCALE);
}

function get_template_active_locale_item()

```

```

{
    $locale = get_template_active_locale();

    return app()->locale->getThemeLocaleInfo($locale);
}
function current_template_uri($path = '')
{
    $templatePath = trailingslashit(get_option('active_template')) .
    untrailingslashit($path);
    return template_uri($templatePath);
}
function is_template_active_locale($locale)
{
    return $locale === registry_read('_spark.template.locale', 'en_US');
}
function has_custom_template($slug)
{
    $customTemplate = trailingslashit(config('custom_page_template_path',
    'pages')) . $slug . '.php';

    if (has_template($customTemplate)) {
        return $customTemplate;
    }

    return false;
}

function has_items($item)
{
    if (!is_array($item)) {
        return false;
    }

    return !empty($item);
}

```

ДОДАТОК В  
(обов'язковий)

**ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ**

Хмельницький Національний Університет  
Факультет програмування та комп'ютерних  
і телекомунікаційних систем  
Кафедра інженерії програмного забезпечення

Дипломний проект, на тему:  
**«Довідниково-інформаційна система  
туристичної фірми»**

Студент: Федорук Роман Віталійович

Керівник: канд. пед. наук, доцент Праворська Н.І.

## Вступ

В наш час поняття діджиталізації і цифровізації так швидко увійшли до навколишнього світу, що уявити процес надання послуг без застосування гаджетів не можливо. Тим паче, в такій висококонкурентній галузі як туризм. Також на звичний порядок надання послуг вплинула пандемія коронавірусу, і це ще одна причина з якої власники туристичних фірм мають вдосконалювати процес надання послуг в туристичній галузі, якщо бажають зберегти свій бізнес. В цьому випадку на допомогу приходять інформаційні технології за допомогою яких здійснюється залучення та втримання клієнтів.

Туристична галузь – це галузь, яка найбільше постраждала під час пандемії коронавірусу. Тому актуальним є питання вдосконалення та розробки програмного забезпечення, яке б відповідало викликам часу та надавало конкурентну перевагу за рахунок зручності використання та потрібної функціональності.

## Мета та завдання проекту

Розробити інформаційно-довідкову систему туристичної фірми, що буде однаково зручною і функціонально достатньою для всіх типів користувачів. Потенційні туристи за допомогою додатку «MyTrip» повинні мати можливість самостійно вибрати і прорахувати бажаний тур, а туристична фірма повинна мати можливість фіксувати кожного потенційного туриста та формувати клієнтську базу, формувати аналітику.

**Для досягнення поставленої мети були сформовані такі завдання:**

1. дослідити предметну область туристичного бізнесу та виявити потреби потенційних користувачів програмного продукту;
2. проаналізувати наявне програмне забезпечення в цій галузі та сформулювати вимоги на розробку нового програмного забезпечення, що відповідає потребам часу;
3. спроектувати структуру інформаційно-довідкової системи та розробити базу даних;
4. вибрати технології для розробки інформаційно-довідкової системи;
5. розробити та протестувати інформаційно-довідкову систему туристичної фірми.

## Актуальність

Не дивлячись на широке представлення на ринку різноманітних CRM-систем, які можна налаштувати під вимоги конкретної фірми існує потреба в розробці більш досконалого програмного забезпечення. Пандемія коронавірусу, періодичні локдауни перевели всю ділову активність в цифровий світ. І сьогодні недостатньо мати CRM-систему, за допомогою якою власник бізнесу буде відслідковувати контакти з клієнтами та розсилати пропозиції. В сьогоднішніх реаліях, власнику бізнесу потрібно мати інструмент за допомогою якого клієнт сам зможе сформувати той продукт, який потрібен саме цьому клієнту без додаткових контактів з співробітниками туристичної фірми. А вже після формування клієнтом потрібної пропозиції, або для надання консультації, до процесу купівлі туру долучаються співробітники туристичної фірми.

## Призначення

Призначення інформаційно-довідкової системи туристичної фірми «MyTour»:

- програма для конструювання турів та прорахунку вартості відпочинку;
- відображення облікової інформації щодо клієнтів, замовлень, продажів, повернень турів.

Завдання, які вирішуються:

- пошук турів, готелів, номерів, типів харчування, додаткових послуг;
- облік клієнтів, замовлень, продажів, повернень турів;
- ведення аналітики, статистичний аналіз.

## Результат проекту

Плануйте відпочинок з **Глобал Турс**  
Ідеальної відпочинку для всієї родини.

Обираючи нас Ви отримуєте:

- Переглянути
- Замовити
- Тур
- ...

**ОБЕРІТЬ НЕОБХІДНІ ПАРАМЕТРИ ТУРУ**

**Оберіть тур**

#	Готель	Тип номеру	Харчування	Заяд	Вартість
1	Tigana Dahab Resort 5*	Double	Все включено	з 19.07 по 26.07 (7 ночей)	32 300 грн
2	Sams 5*	Double	Все включено	з 23.07 по 30.07 (7 ночей)	28 700 грн
3	Torres 4*	Double	Все включено	з 21.07 по 28.07 (7 ночей)	26 300 грн

Електронна адреса:   
 Електронна адреса:   
 Контактна особа (ІМЯ):

Телефон:   
 Телефон:

# Результат проекту

### Додати користувача

Статус: Активний

**Основна інформація**

Ім'я:  Прізвище:  Стать:  Вік:

Електронна пошта:  Діагональ:  Мобільний номер:  Адреса:  Місто:  Країна:

Дата народження:  Телефон:

**Додатково**

Діагональ:  Прізвище:  Ім'я:

### Користувачі

№	Ім'я	Прізвище	Стать	Вік	Е-пошта	Телефон	Країна	Місто	Мова	Тип харчування	Статус
1	Іван	Іванов	М	35	ivan@ukr.com	+380981234567	Україна	Київ	Українська	Звичайний	Активний

### Зірковість

**Зірковість готеля**

Позначення

1\*

2\*

3\*

4\*

5\*

# Результат проекту

### Статуси

**Статуси турів**

Назва:  Колір:

Є місяць:

Активний:

Неактивний:

Нема місяць:

### Налаштування

Мова	Валюта	Стать	Місце	Країна
Українська	USD	М	Київ	Україна
Українська	USD	Ж	Київ	Україна
Українська	USD	М	Київ	Україна

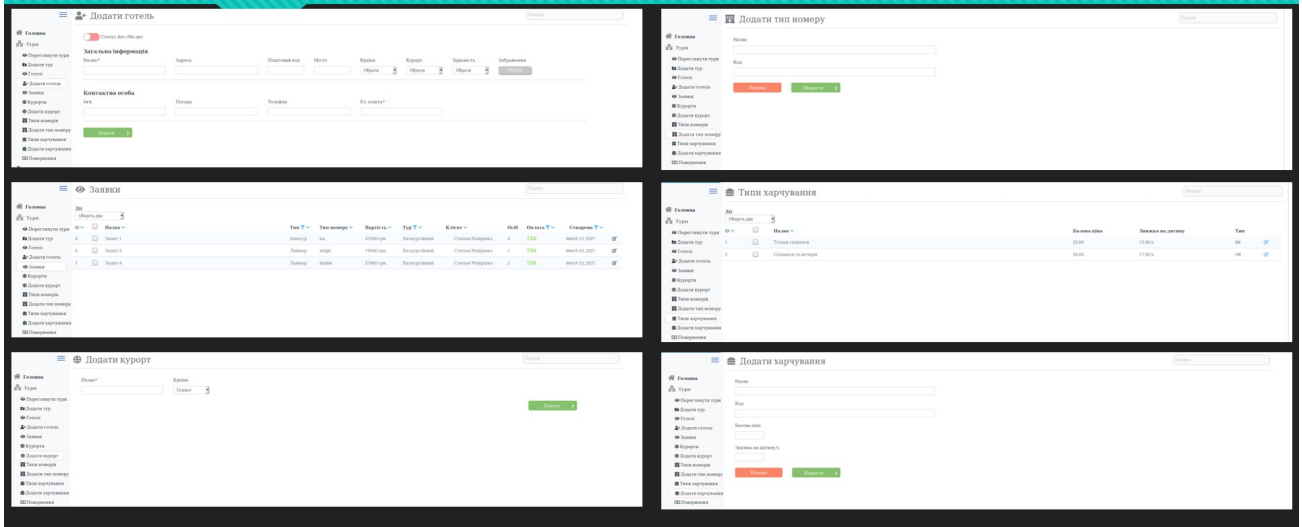
### Повернення

№	Дата повернення	Тип номеру	Вартість	Тип Т	Кількість	№	Дата повернення	№	Дата повернення
1	2023-10-15	Стандарт	100	1	2	101	2023-10-15	102	2023-10-15

### Перегляд

№	Ім'я	Прізвище	Стать	Вік	Е-пошта	Телефон	Країна	Місто	Ваша персональна інформація	Статус
1	Іван	Іванов	М	35	ivan@ukr.com	+380981234567	Україна	Київ	Українська	Активний

## Результат проекту



## Висновок

Кращим способом збільшення доходу, виходячи з досліджень, є створення такої довідниково-інформаційної системи, яка дозволить збільшення середнього чека за рахунок розширення асортименту товарів і послуг. Щоб уникнути паперової тяганини, людських помилок і зосередитися на пріоритетах, потрібно власнику туристичної фірми використовувати довідниково-інформаційну систему.

Всі вимоги до реалізації та рекомендації по вдосконаленню проекту враховані в процесі реалізації. Було проведено функціональне тестування та тестування зручності користування. Таким чином, завдання вирішені в повному обсязі, мета досягнута.

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

здобувача вищої освіти

Федорука Р. В.

Прізвище, ініціали

факультет ПКТС, 4 курс, група ПЗ-17-1

### ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та 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 Anti-Plagiarism) та використання роботи для виявлення 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.

01.06.2021р.

дата

  
підпис

## Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 8.0%**

Словники перевірки: en\_US, pl\_RU, ua\_UA. Помилки в документах: 17%

ID: 92297 Назва: Довідково-інформаційна система туристичної фірми Додано в БД: 2021-06-04 Автора: Р. В. Федорук Керівники: Н. І. Праворська Консультанти: Опоненти:		Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми	
	116844	1167	13574 (12%)	209 (18%)	

Джерело платігу

ID	Опис	Навність платігу в документі	
		Символи	Лексеми
92249	Назва: Довідково-інформаційна система туристичної фірми Додано в БД: 2021-06-04 Автора: Р. В. Федорук Керівники: Н. І. Праворська Консультанти: Опоненти:	9583 (8.0%)	153 (13.0%)



Ім'я користувача:  
Кафедра ІПЗ

ID перевірки:  
1008200055

Дата перевірки:  
06.06.2021 20:53:35 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
06.06.2021 20:59:03 EEST

ID користувача:  
100005589

Назва документа: **Дипломний проект Федорук Р.В**

Кількість сторінок: 112 Кількість слів: 16646 Кількість символів: 135755 Розмір файлу: 7.82 MB ID файлу: 1008275342

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

**12.3%**  
**Схожість**

Найбільша схожість: 3.7% з джерелом з Бібліотеки (ID файлу: 1008265193)

8.29% Джерела з Інтернету

279

Сторінка 114

5.23% Джерела з Бібліотеки

74

Сторінка 116

**0% Цитат**

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

**0%**  
**Вилучень**

Немає вилучених джерел

**Модифікації**

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

Замінені символи

6

Підозріле форматування

18  
сторінок

## ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНИЙ ПРОЕКТ  
освітнього ступеня «Бакалавр»Дипломник Федорук Роман ВіталійовичТема Довідниково-інформаційна система туристичної фірмиСпеціальність 121 – Інженерія програмного забезпечення**Обсяг дипломного проекту:**Кількість листів креслень \_\_\_\_\_; кількість сторінок записки 69

1. Короткий зміст пояснювальної записки та прийнятих рішень У бакалаврській роботі був проведений аналіз предметної області. Був проведений аналіз наявного програмного забезпечення, чи є достатнім їх функціонал. На основі отриманих даних були розроблені функціональні вимоги до дипломного проекту. Були розглянуті деталі реалізації дипломного проектування. Було проведено тестування системи, і за результатами якого, система працює коректно.

2. Висновок про відповідність проекту поставленому завданню Дипломна робота повністю відповідає поставленому завданню та виконана із дотриманням всіх вимог

3. Характеристика виконання кожного розділу проекту, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі обґрунтовується актуальність обраної теми дипломного проектування, формуються цілі і завдання дослідження, описується практична значимість дипломного проекту. У першому розділі охарактеризовано існуючі рішення для вирішення поставленої задачі, викладено опис функціональних вимог до дипломного проекту. В наступних розділах розкриваються деталі реалізації довідково-інформаційної системи, моделювання бази даних та системи

4. Позитивні сторони проекту Тематика дипломного проекту є актуальною, визначено існуючі проблеми та незручності, завдяки чому було модернізовано й усучаснено підхід до роботи з клієнтами туристичної фірми, застосували при цьому існуючі необхідні технології.

5. Негативні сторони проекту Система розроблена таким чином, що клієнти зможуть вказати довідку про вакцинацію тільки тоді, коли з ним зв'яжеться менеджер.

6. Оцінка графічного оформлення та пояснювальної записки проекту Графічне оформлення виконане відповідно до теми дипломного проекту. Графічне представлення виконано на задовільному рівні рівні. Пояснювальна записка відповідає стандартам оформлення.

7. Відгук про дипломний проект в цілому В цілому дипломна робота заслуговує скоріше позитивної оцінки. Матеріал пояснювальної записки добре структурований та добре розписаний, що дозволяє чітко зрозуміти викладений матеріал у рамках тематики дипломної роботи. Графічний матеріал підкріплений інформацією та дозволяє наочно побачити деталі проектування системи.

8. Інші зауваження

9. Оцінка дипломного проекту Робота виконана в повному обсязі. Розглянувши позитивні та негативні сторони представленої дипломної роботи, можна зробити висновок, що вона заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи) Лисенко Сергій Миколайович, доктор технічних наук, доцент кафедри комп'ютерної інженерії та системного програмування (КІСП) ХНУ.

"01" Сервія 2021 р.

(підпис)

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

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: «Довідково-інформаційна система туристичної фірми»

Автор: Федорук Роман Віталійович

Спеціальність: 121 – Інженерія програмного забезпечення

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

Науковий керівник: Праворська Наталія Іванівна, канд. пед. 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 та 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 в роботі, є законними і не є плагіатом, оскільки:

- 1) у тексті дипломного проекту системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень в бланках (титулка, бланк завдання, в структурі підрозділів ВСТУПУ) та в назвах публікацій джерел;
- 2) В якості запозичень системою було зафіксовано послідовність вихідного коду, які є спільними для великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 3) усі запозичення фрагментарні, або мають належним чином оформленні посилання.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності / схожості, складає 12,3% і адресується до 353 першоджерел, що з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь дипломного проекту.

Керівник

Гарант ОП

Завідувач кафедри

Н.І. Праворська

Л.П. Бедратюк

Л.П. Бедратюк