

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

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

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка
до дипломного проєкту (роботи)

магістр
(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ІНФОРМАЦІЙНИХ МОДЕЛЕЙ РОЗРОБКИ ВЕБСАЙТІВ.
RESEARCH AND SOFTWARE IMPLEMENTATION OF DATA
MODEL FOR WEBSITE DEVELOPMENT

Виконав: студент 2 курсу, групи КНТ-129м
спеціальності

121 Інженерія програмного забезпечення
(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Демічев А. В.

(прізвище та ініціали)

Керівник Скрупський С.Ю., к.т.н., доцент

(прізвище та ініціали)

Рецензент Зеленцова І.Я., к.т.н., доцент

(прізвище та ініціали)

2020

Завідувач кафедри проф., д.т.н.
С. О. Субботін
“ ” 2020 року

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

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1–5	Скрупський С.Ю., доцент		
6	Пожуєва Т.О., професор		
7	Коробко О.В., ст. викладач		
Нормо- контроль	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання 02.10.20

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз предметної області	1 тиждень	
2	Розробка та затвердження технічного завдання	2 тиждень	Технічне завдання
3	Проектування структури системи	3-4 тиждень	Структурна схема
4	Розробка схеми функціонування системи та модулів програми	5-6 тиждень	Функціональна схема. Модулі
5	Створення програмного коду системи	7-8 тиждень	Текст програми
6	Тестування та відлагодження програми	9 тиждень	Працездатна система
7	Розробка розділів пояснювальної записки	10 тиждень	ПЗ
8	Розробка слайдів презентації	11-12 тиждень	Слайди презентації
9	Захист роботи	13 тиждень	

Студент(ка)

(підпис)

Демічев А.В.

(прізвище, ініціали)

Керівник проєкту (роботи)

(підпис)

Скрупський С.Ю.

(прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
121 с., 30 рис., 4 додатки, 50 джерел.

ВЕБСАЙТ, CMS, ASP.NET, UML, СЕРВЕР, ТЕХНІЧНЕ ЗАВДАННЯ,
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес розробки вебсайтів.

Предмет дослідження – методи та засоби розробки вебсайтів.

Мета роботи – дослідження та реалізація інформаційних моделей розробки вебсайтів.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, мова програмування C#, персональний комп'ютер з монітором, клавіатурою та маніпулятором типу «миша», що функціонує під управлінням операційної системи.

Результати. Створено програмні засоби у вигляді вебсайту для обліку онлайн замовлень місць в об'єктах розміщення. Розроблена програмна система може успішно експлуатуватися на персональних комп'ютерах під управлінням операційних систем та використовуватися для обліку онлайн замовлень місць в об'єктах розміщення.

Висновки. Розроблено програмне забезпечення для обліку онлайн замовлень місць в об'єктах розміщення. Проаналізовано предметну область, розроблено технічне завдання для реалізації програми, обрано середовище розробки, розроблено структурну схему програми, виконано конструювання програмного забезпечення для обліку онлайн замовлень місць в об'єктах розміщення, здійснено тестування розробленого програмного забезпечення.

Галузь використання – облік онлайн замовлень в об'єктах розміщення.

ABSTRACT

Explanatory note to the final qualifying work of the magister: 121 pages, 30 figures, 4 appendixes, 47 sources.

WEBSITE, CMS, ASP.NET, UML, SERVER, SPECIFICATIONS, SOFTWARE.

Object of study is a process of website development.

Subject of study are methods and tools of website development.

The purpose of the work is to study and develop a information models of website development.

Materials, methods, and tools: structured and object-oriented programming, programming language C#, computer with a monitor, keyboard and mouse manipulator operating under the control of the operating system.

Results. The software in a form of website for accounting of online orders in accommodation facilities has been created. The developed software system can successfully operate on personal computers running operating system and can be used for online orders in accommodation facilities.

Conclusions. Software for online placements has been developed. The subject area is analyzed, the technical task for program realization is developed, the development environment is chosen, the structural scheme of the program is developed, the software for online placements is executed, the developed software is tested.

The field of use is accounting for online orders in accommodation facilities.

ЗМІСТ

Перелік скорочень та умовних позначок.....	8
Вступ	9
1 Аналіз предметної області	10
1.1 Поняття про вебсайти.....	10
1.2 Дослідження вебсайтів інтернет-магазинів.....	17
1.3 Дослідження засобів розробки вебсайтів інтернет-магазинів.....	20
1.3.1 OpenCart.....	22
1.3.2 WooCommerce	24
1.3.3 Prom.ua	25
1.3.4 Wix Stores	27
1.3.5 MotoCMS	28
1.3.6 Еквід.....	29
1.3.7 Recommerce	30
1.3.8 Висновки з дослідження засобів розробки інтернет-магазинів	31
1.4 Аналіз вебсайтів інтернет-магазинів для бронювання місць у об'єктах розміщення	32
1.5 Висновки за розділом 1	37
2 Вибір та обґрунтування структури системи, яка проєктується.....	39
2.1 Дослідження та аналіз вимог до програмного продукту	39
2.2 Вибір інструментарію розробки програмного забезпечення	42
2.3 Розробка інформаційних моделей створення вебсайтів інтернет-магазинів	43
2.4 Висновки за розділом 2	49
3 Основні рішення щодо реалізації компонентів системи.....	51
3.1 Розробка загальної схеми програми.....	51
3.2 Архітектура веб-застосунку на основі обраного підходу	52
3.3 Розробка структури проєкту	53
3.4 Розробка інтерфейсу програмного забезпечення	56

3.5 Висновки за розділом 3	58
4 Керівництво програміста	59
4.1 Призначення й умови застосування програми.....	59
4.2 Характеристики програми	59
4.3 Звернення до програми	60
4.4 Вхідні і вихідні дані.....	60
4.5 Повідомлення	60
5 Керівництво оператора.....	61
5.1 Призначення програми.....	61
5.2 Умови виконання програми.....	61
5.3 Виконання програми	61
5.5 Повідомлення оператору.....	62
6 Економіко-організаційна частина	63
6.1 Планування розробки програмного продукту	63
6.2 Визначення витрат на розробку програми	64
6.3 Розрахунок економічної ефективності програмного продукту.....	71
7 Охорона праці та безпека в надзвичайних ситуаціях.....	73
7.1 Аналіз потенційних небезпек	73
7.2 Заходи по забезпеченню безпеки	74
7.3 Заходи з виробничої санітарії та гігієни праці.....	78
7.4 Заходи безпеки у надзвичайних ситуаціях.....	80
7.4.1 Заходи з пожежної безпеки.....	80
7.4.2 Заходи з цивільного захисту	81
Висновки.....	85
Перелік джерел посилання.....	87
Додаток А Технічне завдання.....	91
Додаток Б Опис програми.....	96
Додаток В Текст програми.....	99
Додаток Г Слайди презентації.....	113

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CMS – система керування контентом;

MVC – архітектура веб-додатків, заснована на принципі «Модель-Вид-Контролер»;

UML – мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення;

ASP.NET – технологія розробки веб-додатків на мові C#;

ОС – операційна система;

ПЗ – програмне забезпечення;

ПО – предметна область;

ПК – персональний комп'ютер;

ПП – програмний продукт.

ВСТУП

Останнім часом виникають потреби автоматизації процесу продажу готової продукції за допомогою веб-сайтів у вигляді інтернет-магазинів [1]–[5]. Відомі методи створення інтернет-магазинів передбачають, як правило, необхідність використання спеціалізованого програмного забезпечення – платформ-конструкторів, що дозволяють створювати веб-сайти на основі наявних шаблонів [4]–[13]. Такий підхід призводить до високої складності та низької інтерпретовності автоматично згенерованого коду, внаслідок чого суттєво ускладнюється процес доопрацювання та модифікації сайту при необхідності подальшого внесення змін з урахуванням нових потреб користувачів. Крім того, веб-сайти, створені за такою технологією є прив'язаними до певної платформи, що також ускладнює їх подальшу експлуатацію [1]–[43]. Наявність зазначених недоліків обумовлює необхідність створення інформаційних моделей, що спростять розробку спеціалізованих інтернет-магазинів та забезпечать можливість їх модифікації без необхідності внесення суттєвих змін у програмний код .

Роботу присвячено вирішенню актуального завдання дослідження та програмної реалізації інформаційних моделей розробки вебсайтів.

Метою роботи є дослідження та реалізація інформаційних моделей розробки вебсайтів.

Для досягнення поставленої мети необхідно було розв'язати такі задачі:

- аналіз процесу розробки вебсайтів;
- розробка технічного завдання для реалізації програми;
- вибір середовища розробки програмного забезпечення;
- створення інформаційних моделей розробки вебсайтів;
- конструювання програмного забезпечення у вигляді вебсайту з продажу місць в об'єктах розміщення;
- тестування розробленого програмного забезпечення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про вебсайти

«Веб-сайт» – калька з англійського «web-site». Від «web», що означає «мережу», «павутина», і «site», що означає «місце». Таким чином словосполучення «web-site» буквально перекладається як «місце в мережі». Часто замість «веб-сайт» кажуть просто «сайт» для стислості [1].

У комп'ютерній термінології веб-сайтом називається сукупність будь-яких електронних документів, об'єднаних під одним доменним ім'ям в мережі. В якості ім'я може використовуватися і IP-адреса. Зазвичай мається на увазі, що сайт знаходиться в мережі Інтернет. Сукупність усіх веб-сайтів становить Всесвітню павутину.

Для доступу клієнтів до розташованих на серверах веб-сайтів був розроблений спеціальний протокол HTTP. Через проблеми з безпекою протоколу HTTP, з'явився захищений протокол HTTPS, який схожий на HTTP, але має більш високий ступінь безпеки. Крім веб-сайтів в Інтернеті доступні так само WAP-сайти, призначені для мобільних пристроїв [1]-[4].

Сторінками сайтів називаються файли з текстом, що містять розмітку HTML. Ці файли завантажуються на комп'ютер користувача, обробляються браузером і відображаються пристроєм виводу (монітором, принтером, синтезатором мови і т.п.). Мова розмітки HTML дозволяє не тільки форматувати текст, а й позначати в ньому семантичні елементи, створювати гіперпосилання, вставляти в текст зображення та інші мультимедійні матеріали. Крім HTML на сторінках сучасних сайтів застосовуються також CSS, JavaScript та інші технології.

Сторінки сайтів можуть бути простими статичними файлами або створюватися спеціальною програмою на сервері. Деякі з таких програм («двигуни») забезпечують власнику сайту можливість швидкої і простої публікації інформації на сайті, гнучкого налаштування відображення цієї

інформації і зовнішнього вигляду сторінок. Такі двигуни називаються ще «системами керування вмістом».

Перші веб-сайти були наборами статичних документів. Але в даний час переважна більшість сайтів є динамічними і інтерактивними. Фахівці використовують для них термін «веб-додаток», тобто програмний комплекс, призначений для вирішення завдань сайту. Створення оригінальних сайтів перетворилося в особливу професію і спеціалізацію цілих студій, які пропонують створення сайтів на базі найсучасніших технологій.

У більшості випадків одному веб-сайту дається одне доменне ім'я. По доменних іменах веб-сайти ідентифікуються в світовій мережі. Але можливі й інші варіанти: одному сайту можуть бути співставленні кілька доменних імен або кілька сайтів можуть бути доступні під одним ім'ям [1].

Кілька доменів часто використовують великі сайти, щоб логічно відокремити надані послуги по їх виду. Наприклад `news.google.com`, `mail.google.com` і `maps.google.com`. Також нерідкі випадки виділення особливих доменів для окремих країн і мов, як, наприклад, `google.fr` і `google.ru` логічно є регіональними версіями сайту Google, хоча технічно представляють собою різні сайти. Один веб-сайт може бути відкритий для доступу з різних адрес, а його копії можуть зберігатися на різних серверах. В такому випадку копія сайту називається дзеркалом. Існує ще поняття оффлайнової версії сайту - так називається копія сайту, доступна для перегляду на комп'ютері без підключення до мережі і без використання серверного програмного забезпечення.

Об'єднання окремих сайтів під загальним доменом іноді характерно для сервісів безкоштовного розміщення сайтів в Інтернеті. Для ідентифікації таких сайтів після вказівки домена в адресі може стояти тильда і власне ім'я сайту. Наприклад: «`example.com/~site/`». Але частіше для цих цілей використовуються домени третього рівня: «`site.example.com`». Послуга зберігання і забезпечення доступу до сайту на сервері називається веб-хостингом. Колись кожен сайт розташовувався на власному сервері, але в

зв'язку з технологічними покращенням серверів на одному сервері стало можливе розміщення великої кількості сайтів. Розташування сайту на одному сервері з іншими сайтами називається віртуальним хостингом. Існують різні класифікації веб-сайтів.

За доступністю сервісів [49]:

- відкриті – всі сервіси повністю доступні для будь-яких відвідувачів.
- напіввідкриті – для доступу необхідно зареєструватися (зазвичай безкоштовно).
- закриті – повністю закриті службові сайти організацій (в т.ч. корпоративні сайти), особисті сайти приватних осіб. Такі сайти доступні для вузького кола людей. Доступ новим людям зазвичай дається через інвайт (запрошення) [49].

За природою вмісту [49]:

- статичні – весь вміст заздалегідь готується. Користувачеві видаються файли в тому вигляді, в якому вони зберігаються на сервері.
- динамічні – вміст генерується спеціальними скриптами (програмами) на основі інших даних з будь-якого джерела.

За фізичним розташуванням [49]:

- зовнішні сайти мережі Інтернет.
- локальні сайти – доступні тільки в межах локальної мережі. Це можуть як корпоративні сайти організацій, так як і сайти приватних осіб в локальній мережі провайдера.

За схемою подання інформації, її обсягом і категорії вирішуваних завдань можна виділити наступні типи веб-ресурсів:

- сайт візитка;
- корпоративний веб-сайт;
- інтернет магазин;
- інформаційний веб-сайт;
- ігровий портал;
- контент-проект;

- промо-сайт;
- сайт-форум;
- блог.

Сайт-візитка – найпростіша версія сайту. Сайт надає собою набір HTML-сторінок, без засобів інтерактивної взаємодії з користувачем (веб-додатків). Сайт повністю відповідає дизайну компанії. Даний варіант найбільш привабливий для компаній, основний рід діяльності яких не пов'язаний з мережею Інтернет. Компанії, використовують сайт, перш за все лише для надання рекламної інформації для потенційного клієнта. Дві основні функції які виконує сайт візитка - комунікація і інтерактивність [1].

Корпоративний веб-сайт є необхідним атрибутом діяльності будь-якої фірми в даний час. Жодна поважаюча себе організація не обходиться без віртуального представництва в Інтернеті. У кожної з них повинен бути сайт, на якому розповідають про діяльність компанії, будуть дані контакти і основні характеристики. Корпоративний веб-сайт – це представництво організації в Інтернеті, при цьому така організація не завжди є корпорацією. Кількість її співробітників може бути кілька людей [1].

Функції корпоративного сайту – це створення іміджу фірми, просування її на ринку, зберігання інформації, комунікація співробітників один з одним. Високорозвинений корпоративний сайт дозволяє проводити конференції в режимі онлайн, і на таких конференціях вирішувати важливі ділові питання. Крім того, він дозволяє керівництву фірми оперативно відстежувати останні новини компанії і отримувати аналітичні матеріали за різними напрямками діяльності підприємства.

Функції корпоративного сайту вже давно не обмежуються просто ознайомленням зацікавлених осіб з товарами і послугами фірми. Такі сайти допомагають структурувати компанію зсередини, також організують спілкування керівництва компанії з віддаленими філіями, інформують співробітників компанії про її додаткові можливості. У функції корпоративного веб-сайту входить також спілкування співробітників

компанії з клієнтами фірми (для цієї мети часто створюються форуми), розсилка новин (повідомлення партнерів і клієнтів про новини компанії) і багато іншого [1].

Велику роль в розробці корпоративного сайту грає система редагування сайту (cms для корпоративного сайту), яка дозволяє співробітникам корпорації самостійно змінювати інформаційне наповнення сайту і дизайн корпоративного сайту. Така система редагування сайту дуже зручна, тому що в такому випадку клієнт не платить за додавання та оновлення сторінок, а може робити все це сам. Тому cms для корпоративного сайту набирає все більшої популярності. Звичайно, основним для корпоративного веб-сайту є все-таки його взаємодія з клієнтом. Тому навігація сайту повинна бути простою і зрозумілою користувачеві. Користувач не повинен довго займатися пошуком потрібної йому інформації, а знаходити її відразу.

Навігація сайту – це той аспект, якому повинна приділятися особлива увага. Інформаційне наповнення сайту повинно бути простим і зрозумілим, розкривати всі основні сфери діяльності компанії. Дизайн корпоративного сайту повинен залежати від фірмового стилю компанії і орієнтуватися на переваги цільової аудиторії. Корпоративний сайт - це гармонійна цілісність, в якій всі елементи пов'язані між собою [1].

Інтернет-магазин (англ. Online shop або e-shop) – веб-сайт, що рекламує товар або послугу, приймає замовлення на покупку, пропонує користувачеві вибір варіанту розрахунку, способу отримання замовлення і який виписує рахунок на оплату. У наш час широко використовуються веб-сайти для продажу різноманітних товарів через інтернет-магазини.

Інтернет-магазин об'єднує елементи прямого маркетингу з образом відвідування традиційного магазину. Відмінною рисою Інтернет-магазинів в порівнянні зі звичайною формою торгівлі є те, що інтерактивний магазин може запропонувати значно більшу кількість товарів і послуг і забезпечити споживачів значно більший обсяг інформації, необхідним для прийняття

рішення про покупку. Крім того, за рахунок використання комп'ютерних технологій можлива персоналізація підходу до кожного з клієнтів, виходячи з історії його відвідувань магазину і зроблених раніше покупок [1].

Основні проблеми реалізації Інтернет-магазину лежать на стику технологій Інтернету і традиційної комерційної діяльності. У звичайній торгівлі покупець звик до того, що є можливість оцінити товар візуально, визначити його якість і характеристики. В електронній торгівлі він такої можливості позбавлений. Найчастіше візуальної інформації досить, але тут вступають в дію емоційні і психологічні чинники. Часто існують проблеми з доставкою товарів, особливо якщо їх ціна невелика [1].

Як правило інтернет магазин створюється на основі вже наявного бізнесу – тобто реального магазину, фірми, яка вже продає якийсь асортимент продукції, будь то комп'ютери, книги, продукти харчування або щось інше.

Створюючи інтернет сайт з інтернет магазином у компанії з'являється ще одна торгова точка, яка працює цілодобово.

Основні вимоги, що пред'являються покупцями до організації Інтернет-магазинів [1]:

- зрозумілий інтерфейс і зручна навігація по магазину;
- зручна система перехресних посилань і інших елементів навігації, що дозволяє оптимальним чином отримати необхідну інформацію;
- мінімальна кількість дій користувача (клацань миші, переходів і т. д.) для здійснення покупки.

Інформаційний веб-сайт – це дуже великий веб-ресурс, який надає вичерпну інформацію з певної тематики. Інформаційні веб-сайт містять засоби взаємодії з користувачами і дозволяють користувачам спілкуватися в рамках порталу (форуми, чати) – це середовище існування користувача [49]. Інформаційний веб-сайт нагадує енциклопедію або спеціалізований журнал.

Основне завдання такого інформаційного веб-сайту – є безпосередньо інформаційне забезпечення користувачів. Інформаційні веб-сайти –

висвітлюють ті чи інші аспекти людської діяльності, областей науки, техніки, фінансів, соціальних явищ і т.д. Для інформаційних веб-сайтів характерна повнота, об'єктивність і доступність інформації. Найчастіше це, відсутність прямих комерційних інтересів і висока відвідуваність ресурсу.

Основною особливістю інформаційного веб-сайту є те, що він привертає тільки цільову аудиторію.

Контент-проект – це сайт, як правило некомерційної направленості, який являє собою зібрання текстів, статей та інших файлів, основне завдання якого - залучення відвідувачів з певної тематики. Надалі, цей трафік зазвичай перенаправляється на комерційні сайти і, по суті, контент-проект використовується, як рекламний майданчик [50].

Може бути виконаний у вигляді інтернет-бібліотеки, енциклопедії або довідника. Кожна окрема сторінка сайту контент-проекту дає, як правило, незначний трафік, однак загальний трафік такого сайту може бути дуже значним [50].

Контент-проект, за умови розміщення на ньому відповідної за тематикою реклами, дає найякісніший і тематичний трафік на комерційні сайти партнерів (а значить і конвертацію відвідувачів в покупців), тому що ця тема, з дуже високою часткою ймовірності, буде цікава більшості відвідувачів, які прийшли з сайту контент-проекту [50].

Також може являти собою освітній або розважальний ресурс.

Промо-сайт – це сайт створений спеціально для просування будь-якого товару або послуги або основного сайту компанії [50]. Це веб-сайт створений виключно в рекламних цілях, що розробляється для online підтримки нового продукту, послуги, окремого заходу, основного сайту компанії.

Як правило, створення промо-сайтів ведеться в рамках рекламної кампанії і їх існування регламентовано рамками самої рекламної кампанії. Інформаційне наповнення і дизайн промо-сайту повністю підпорядковані рекламним завданням – яскравими і нестандартними засобами формувати

позитивне ставлення цільової аудиторії до нового продукту або послуги компанії. В даному випадку пріоритетна форма, а не зміст – запам'ятовується і ексклюзивний веб дизайн, велика кількість ілюстративного матеріалу. Створення промо-сайту часто використовує flash-технології і містить відеоролики, ігри та презентації.

Ще одне важливе завдання промо-сайту – вибудувати діалог з потенційними споживачами, відповідно обов'язкова наявність інтерактивних елементів: опитування, систем голосування, форумів і т.д. Обов'язкова умова розробки промо сайту - залучення потенційних споживачів в процес взаємодії з продуктом і послугою з отриманням від них зворотного зв'язку.

Сайт-форум – це клас веб-додатків для організації спілкування відвідувачів веб-сайту. Термін відповідає сенсу вихідного поняття «форум». Для роботи форуму часто потрібно база даних, як правило, MySQL.

Форум може бути самостійним сайтом, а може бути розділом сайту. Найчастіше сайт-форум робиться на під домені основного сайту і призначений для організації спілкування відвідувачів сайту між собою і з адміністрацією сайту [50].

Блог – це сайт, який представляє собою інтернет-щоденник, або журнал, який ведеться на зразок стрічки, при цьому на кожному блогу відвідувачі можуть залишати свої коментарі-повідомлення, доступні з головної сторінки відповідної теми. Блог зазвичай дуже часто оновлюється, з цієї причини він може набагато частіше індексуватися пошуковими машинами, ніж звичайні сайти [50].

1.2 Дослідження вебсайтів інтернет-магазинів

Як відзначено вище, у наш час широко використовуються веб-сайти для продажу різноманітних товарів через інтернет-магазини [2].

Інтернет-магазин – сайт, який займається торгівлею і рекламуванням товарів за допомогою мережі Інтернет. Цей торговельний ресурс,

незважаючи на тривалість свого існування, відомий і популярний у всьому світі [2].

На даний момент виділяють наступні види електронних магазинів: електронні вітрини, торгові майданчики і повнофункціональні електронні магазини.

Електронні вітрини вважаються простим способом організації в Інтернеті. У них ми можемо знайти інформацію про пропоновані товари, але відсутні функції автоматичного покупки і можливість прийому електронних платежів [2].

Торгові майданчики об'єднують в собі риси як електронної вітрини, так і повнофункціонального електронного магазину. Але, на відміну від них, тут користувач може додати будь-який товар у віртуальний «кошик», в подальшому редагуючи його вміст. Оформлення і підрахунок вартості замовлення ведеться автоматично [2].

Повнофункціональні електронні магазини мають програмні комплекси, повністю обслуговуючими процес торгівлі, що помітно спрощує покупку товарів.

Структура роботи електронних магазинів досить проста. Схема їх роботи така. Відвідування покупцем сайту інтернет-магазину. Для залучення покупців, електронний магазин володіє величезною кількістю каналів (пошукова оптимізація, реклама в соціальних мережах, сарафанне радіо, сайти-агрегатор товарів, контекстна реклама, природні посилання і т.д.). Оформлення замовлення. Поетапно даний процес виглядає наступним чином: покупець приходить на сайт магазину, бачить, що цікавить його товар, кладе його в кошик, при бажанні покупець доповнює товар аксесуарами, запропонованими в даному інтернет-магазині, покупець оформляє замовлення, співробітник інтернет-магазину зв'язується з покупцем для підтвердження замовлення, а далі віддає його в службу доставки. Доставка товару кур'єром або поштою. Правильна послідовність кроків в даному випадку: пакування товару; відправлення обраним покупцем способом;

очікування отримання товару і оплата. Після продажна робота з покупцем. По-перше, якщо споживачеві сподобався ваш магазин і отриманий товар, то він і далі буде готовий робити замовлення. По-друге, можливо, що при відвідуванні вашого інтернет-магазину, замовник залишив свої дані, через які він зможе дізнаватися про нові акції або спеціальні пропозиції [2].

Насправді, пошук товару по сайту електронного магазину досить спрощений. Можна навести такі способи вибору продукту:

- універсальні пошукові системи товарів. Це сайти, які здатні здійснити пошук відразу по декількох магазинах;
- звичайні пошукові системи;
- великі сайти, на яких продають відразу безліч продавців;
- каталоги інтернет-магазинів.

Вибравши потрібний товар, покупець відправляє його в кошик. Він дає можливість змінювати кількість товарів. Далі необхідно вибрати «оформити замовлення». На даному етапі користувач вводить свої особисті дані і вибирає спосіб оплати і доставки продукту. Після перевірки всієї інформації, можна підтвердити замовлення. Далі з вами зв'яжуться і організовують весь процес доставки товару в потрібний час і потрібне місце [2].

Хоча покупки через Всесвітню мережу набирають популярність, це не заважає покупцям сумніватися в надійності інтернет-магазинів.

Звичайно ж, інтернет-магазини мають велику кількість переваг, таких як:

- економія часу. Для пошуку потрібного товару не доведеться обходити безліч магазинів – досить вбити відповідні дані у відповідний електронний магазин;
- економія грошей. Як правило, ціни в інтернет-магазині значно нижче цін в звичайних магазинах. Також примітно те, що в електронних магазинах регулярно проводяться акції та розпродажі [2];

- широкий асортимент товарів. Найчастіше вибір в Інтернеті багатший, і нерідко деякі продукти можна знайти тільки у Всесвітній павутині;

- повноцінні характеристики товарів. Зазвичай сайти магазинів інформують вас про товар, надають фотографію або відео – огляд;

- зручність оплати. Покупець сам може вибрати спосіб оплати товару - готівкою при доставці, банківською картою або електронними грошима;

- вибір доставки. Пошта, кур'єр або самостійний вивіз - вирішувати вам.

Але у електронних магазинів є ряд недоліків [2]:

- оцінка товару «на око». Неможливо по-справжньому оцінити річ через Інтернет: знайомство з товаром відбувається тільки після його отримання;

- ризик бути обдуреним. Придбаний товар може бути підробкою або зовсім не відповідати опису на сайті;

- можливі труднощі з доставкою. Якщо покупцеві необхідний товар на свято, то краще зробити замовлення заздалегідь [2];

- реєстрація. Інтернет – магазини найчастіше вимагають вашої реєстрації в ньому для інформування вас про акції і т.д. [2].

1.3 Дослідження засобів розробки вебсайтів інтернет-магазинів

Існують декілька способів розробки інтернет-магазинів:

- розробка на основі спеціалізованого програмного забезпечення, тобто програмних двигунів, які надають функціонал для швидкого створення веб-сайту;

- розробка без використання так званих двигунів, але даний підхід набагато збільшує час розробки та задіює більше людських ресурсів, що в результаті призводить до більших фінансових затрат.

Перший підхід передбачає використання платформ для створення веб-сайтів. Платформи для інтернет-магазинів зручно класифікувати за двома ознаками: технічного типу і умов використання.

За технічним типу виділяють власне двигуни або CMS і SaaS-платформи або конструктори.

CMS (від англ. Content Management System) – це система управління контентом сайту, яка включає програмне забезпечення для роботи з вмістом сайту (додавання текстів і мультимедійних файлів, створення нових сторінок і розділів, редагування контенту, зміни дизайну сайту і т.д.). Іноді спрощено CMS ще називають движком сайту. Якщо говорити просто - це основа сайту, яка керує всіма процесами, що відбуваються на веб-майданчику [48]. CMS відрізняється гнучкістю, масштабованістю і необмеженими можливостями кастомізації. Наприклад, користувачі движків можуть довільно міняти дизайн, змінювати технічні параметри роботи сайту, гнучко налаштовувати SEO.

Зворотний бік медалі – власник інтернет-магазину на CMS повністю відповідає за працездатність проекту. Він керує параметрами сервера, стежить за створенням резервних копій, забезпечує відповідність сайту вимогам пошукових систем. Це вимагає від власника сайту спеціальних знань і додаткових витрат.

При використанні CMS власнику магазину доводиться витрачати гроші:

- на платний шаблон або послуги дизайнера;
- на додаткові модулі або послуги програмістів;
- на адаптацію CMS до регіональних вимогам.

Приклади двигунів для інтернет-магазинів – WooCommerce, OpenCart. SaaS-платформи працюють інакше. Вони надають власнику магазину готовий майданчик для торгівлі. Бізнесмену досить вибрати зовнішній вигляд магазину, завантажити товари і залучити покупців. Працездатність проекту, включаючи створення резервних копій, швидкість завантаження сторінок,

дотримання вимог пошукових систем, підключення платіжних агрегаторів і сервісів доставки, бере на себе SaaS-платформа.

В результаті використання платних SaaS-платформ часто обходиться власнику сайту дешевше роботи з умовно безкоштовними движками. А розгорнути магазин можна набагато швидше, так як для роботи з конструктором не потрібні спеціальні технічні знання та послуги програміста.

Зворотний бік медалі – платформи менш гнучкі, їх складніше масштабувати.

У більшості випадків власнику магазину доводиться працювати з шаблонним дизайном. Деякі конструктори мають проблеми з індексацією пошуковими системами.

Далі проаналізуємо існуючі CMS рішення для розробки інтернет магазинів.

1.3.1 OpenCart

За даними builtWith, OpenCart (рис. 1.1) – явний лідер серед CMS для інтернет-магазинів на українському ринку. Частка цього двигуна становить 57%.

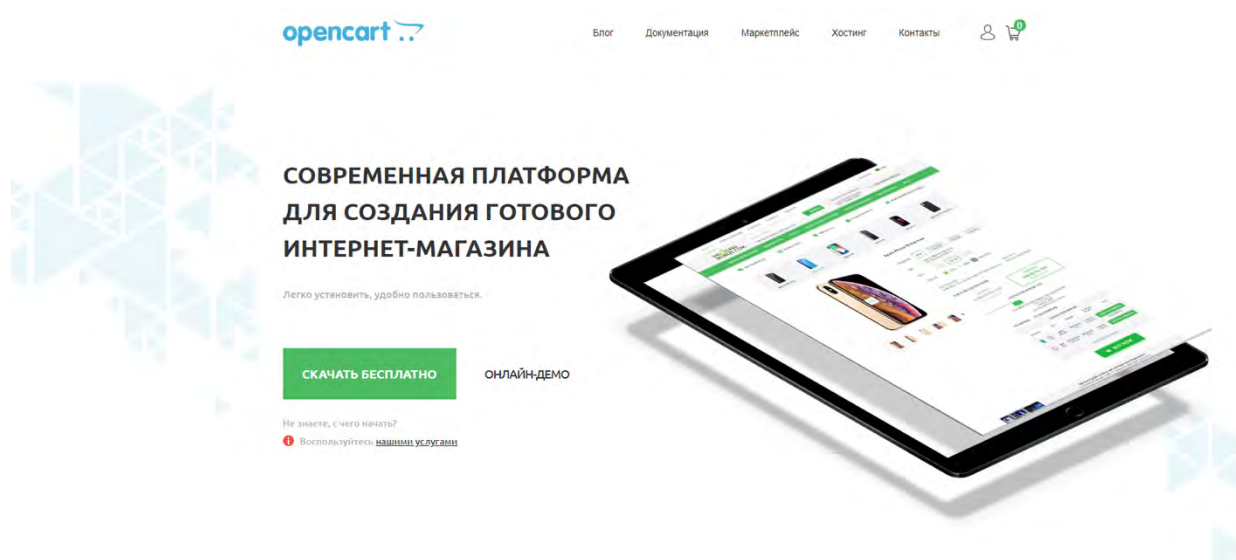


Рисунок 1.1 – Офіційний сайт OpenCart

Основні особливості OpenCart. Це повноцінний движок, який легко масштабувати і кастомізувати. В адмінці OpenCart є готова CRM-система, за допомогою якої можна приймати замовлення і відслідковувати їх виконання.

OpenCart – CMS з відкритим кодом, яка розвивається силами спільноти. У движка є розвинені регіональні клуби. На пострадянському просторі можна використовувати російську або українську збірки OpenCart.

Якщо движок створений на відкритому коді, кожен може його довільно змінювати і безкоштовно використовувати в будь-яких цілях.

Движок після установки має функціональність для онлайн-торгівлі. Однак її потрібно адаптувати до регіональних реалій: підключати платіжні модулі, сервіси доставки.

Власник сайту може розширити функціональність і змінити дизайн за допомогою модулів і шаблонів. Їх можна купити в маркетплейсе або замовити стороннім розробникам.

Умови використання OpenCart. Це CMS з відкритим вихідним кодом. Її можна використовувати безкоштовно для будь-яких проектів. Але власнику сайту доведеться витратити гроші на платні модулі та дизайн-шаблони. Альтернативний варіант - платити розробникам за унікальні модулі та дизайн. Тобто фактично власник магазину змушений купувати готові платні рішення або платити розробникам за розширення функціональності, адаптації сайту до потреб конкретного проекту. Переглянути переваги і недоліки можна на таблиці 1.1.

Таблиця 1.1 – Переваги і недоліки OpenCart

Переваги	Недоліки
Повноцінний движок, дає гнучкість і масштабованість	Потрібно витратити ресурси на обслуговування, безпеку, підтримку актуальності
Є багато модулів для розширення функціональності	Потрібно адаптація до регіональних умов роботи
Є вбудована CRM	За ексклюзивний дизайн і функціональність потрібно платити
Є активна спільнота користувачів	Для невеликих магазинів занадто складний і «неповороткий» движок

1.3.2 WooCommerce

За даними builtWith, WooCommerce (рис. 1.2) - найпопулярніша CMS для інтернет-магазинів в світі без урахування SaaS-платформ.

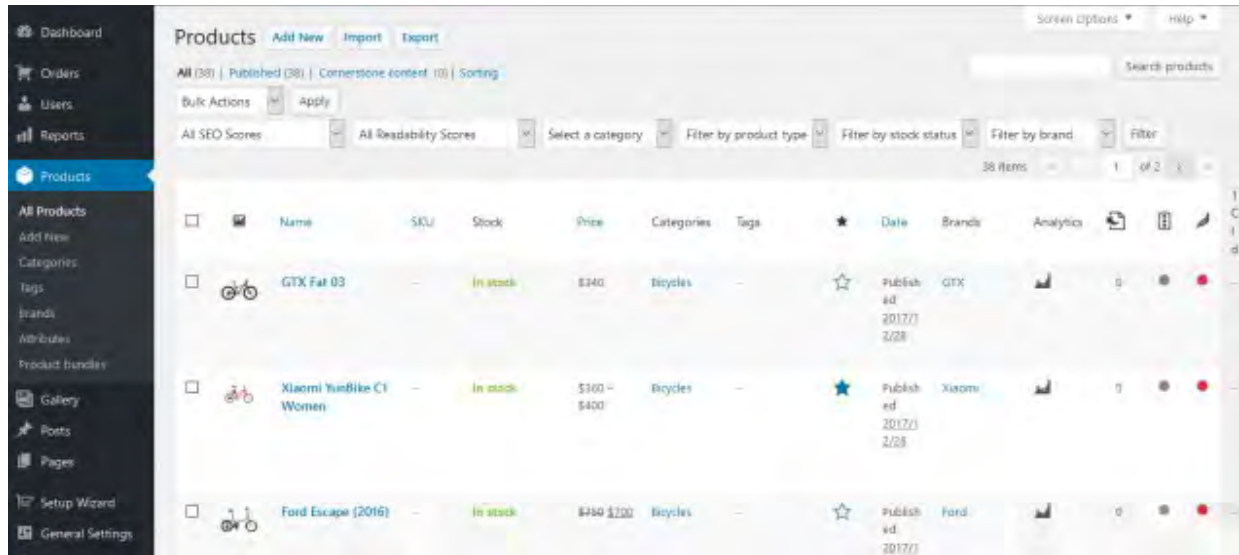


Рисунок 1.2 – Адмін панель WooCommerce

Основні особливості WooCommerce. WooCommerce - надбудова для WordPress. Це найпопулярніший двигун в світі. Зв'язку WordPress і WooCommerce просто встановити і налаштувати. Однак для початку торгівлі її доведеться адаптувати до регіональних умов за допомогою плагінів. Як мінімум знадобляться платіжні агрегатори і віджети сервісів доставки.

WooCommerce відрізняється розвиненою інфраструктурою: в маркетплейсе доступні сотні дизайн-шаблонів і плагінів, за допомогою яких можна розширити функціональність і поліпшити зовнішній вигляд магазину. Також можна звернутися до сторонніх розробників.

Умови використання WooCommerce. Це безкоштовна платформа з відкритим вихідним кодом. Однак за якісні плагіни і дизайн шаблони, необхідні для кастомізації і розширення функціональності, доведеться платити.

WooCommerce і аналогічні двигуни з відкритим кодом безкоштовні

тільки умовно. Якщо власник магазину не знайомий з програмуванням, йому доведеться витратити гроші на послуги розробників або платні плагіни. Засоби знадобляться на унікалізації дизайну, підключення платіжних модулів і сервісів доставки, професійних SEO-плагінів, сервісів зворотного дзвінка і так далі. Переглянути переваги і недоліки можна на таблиці 1.2.

Таблиця 1.2 – Переваги і недоліки WooCommerce

Переваги	Недоліки
Це надбудова для найпопулярнішою в світі CMS, тому в інтернеті є багато інструкцій по роботі з WordPress і WooCommerce	Потрібно адаптувати до локальних умов
Простота установки і управління	За розширення функціональності або унікальний дизайн доведеться платити
Розвинена інфраструктура, є багато розробників	Треба самостійно стежити за технічною відповідністю та безпекою

1.3.3 Prom.ua

Популярна українська платформа (рис. 1.3) для створення інтернет-магазинів і маркетплейсів. Поєднує функції конструктора сайтів і каталогу продуктів.

Основні характеристики Prom.ua. Після реєстрації користувач може вибрати один з декількох сотень дизайн-шаблонів, додати або імпортувати товари і почати торгувати. Конструктор адаптований до українського ринку.

Платформа підтримує функції, необхідні для онлайн-торгівлі. У тому числі на Prom.ua є:

– гнучкий кошик;

- інформування користувачів через sms або email;
- базові функції CRM: управління списками клієнтів і статусами замовлень;
- онлайн-чат;
- навігація по категоріях товарів і брендам.

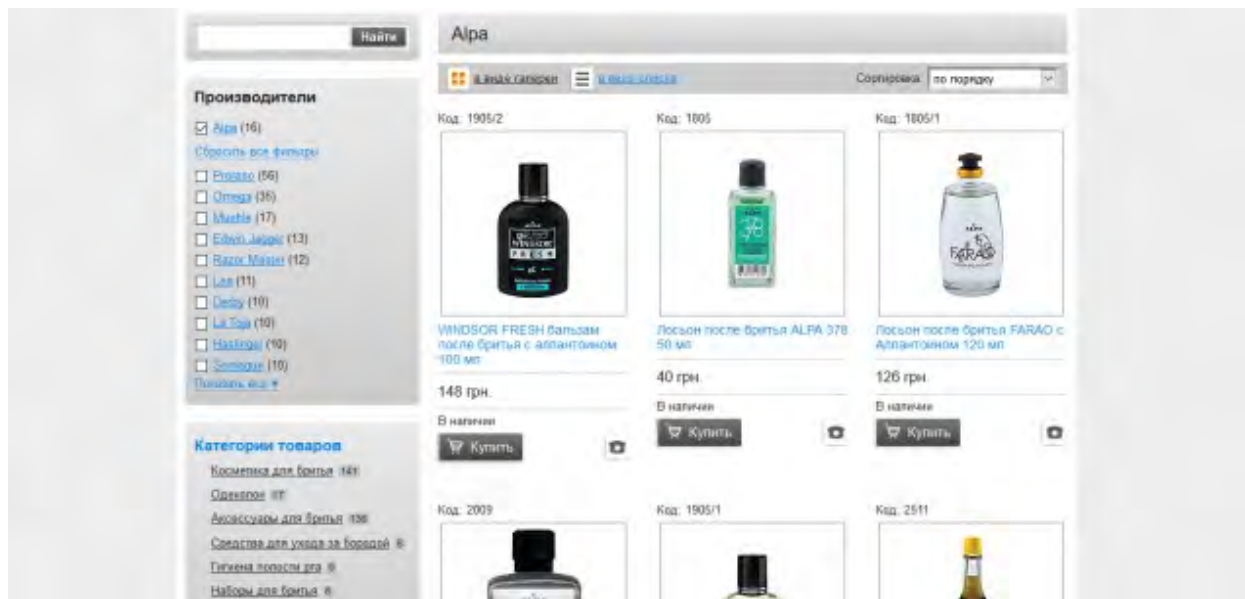


Рисунок 1.3 – Магазин Prom

Платна платформа. Мінімальна вартість використання становить 4900 гривень на рік. Новим користувачам доступний тестовий період – 14 днів. Переглянути переваги і недоліки можна на таблиці 1.3.

Таблиця 1.3 – Переваги і недоліки Prom.ua

Переваги	Недоліки
Простота використання	Платний сервіс
Адаптованість до українського ринку	Магазини виглядають шаблонно
Функції маркетплейса: клієнти можуть переходити з магазину в магазин у пошуках потрібного товару або кращої ціни	Немає можливості впливати на технічні параметри та розширювати функціональність

1.3.4 Wix Stores

За даними builtWith, в серпні 2018 року Wix Stores (рис. 1.4) – найпопулярніша еcommerce-платформа в світі.

Основні особливості Wix Stores. Це SaaS-платформа або конструктор інтернет-магазинів. Після реєстрації користувачеві доступні дизайн-шаблони. Підтримує основні функції інтернет-магазину:

- кошик;
- відстеження статусів замовлень;
- відстеження списків клієнтів;
- фільтрація товарів за категоріями і виробником.

Щоб почати торгувати, користувачеві потрібно підключити платіжні модулі і сервіси доставки.

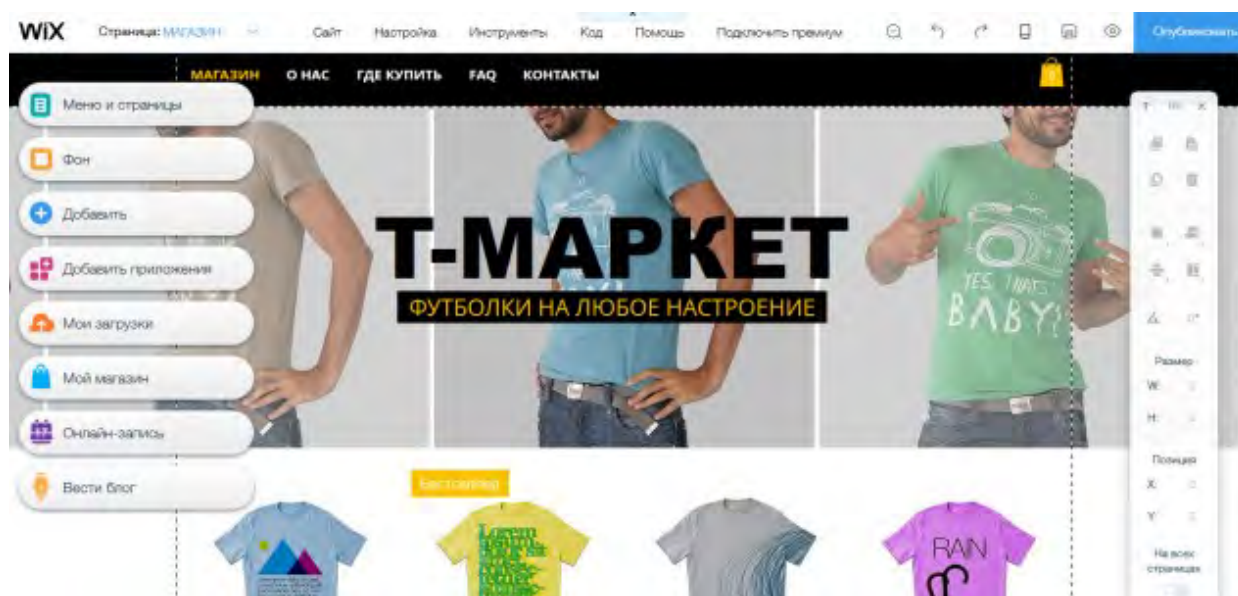


Рисунок 1.4 – Drag-n-drop редактор Wix

Є безкоштовний тариф, на якому можна протестувати функціональність Wix. Тариф з можливостями еcommerce сайту коштує 16 доларів на місяць. Переглянути переваги і недоліки можна на таблиці 1.4.

Таблиця 1.4 – Переваги і недоліки Prom.ua

Переваги	Недоліки
Простий запуск інтернет-магазину без технічних навичок	Платний сервіс
Функціональність дозволяє запускати онлайн-торгівлю за кілька годин	Проблеми з індексацією «Яндекс»
Є дизайнерські шаблони, за допомогою яких можна редагувати зовнішній вигляд	Немає можливості самостійно розширити функціональність

1.3.5 MotoCMS

MotoCMS – SaaS-платформа для створення інтернет-магазинів. Конструктор забезпечує користувачеві гнучкі настройки SEO, включаючи настройку мовних і регіональних атрибутів, вибір пріоритетів індексування URL в карті сайту, можливість підключення зовнішнього файлу sitemap.xml, настройки пермалінков (структури постійних посилань) і редиректів.

Є можливість додати на сайт скрипти зовнішніх сервісів, вибрати шрифти, швидко підключити сервіси аналітики і кабінети веб майстрів. Вбудована CRM з базовими функціями дозволяє відстежувати замовлення і управляти списками користувачів.

Умови використання. Користуватися MotoCMS можна безкоштовно. Але необхідно купити шаблон за ціною від 199 доларів. Підтримка шаблону гарантована протягом року з моменту покупки. Переглянути переваги і недоліки можна на таблиці 1.5.

Таблиця 1.5 – Переваги і недоліки MotoCMS

Переваги	Недоліки
Зручна адміністративна панель з можливостями на рівні простих CMS	Потрібно купувати шаблони
Великий вибір професійних шаблонів	Через рік обраний шаблон може не підтримуватися розробником
Простота настройки і управління	Техпідтримка безкоштовна тільки в перший рік використання платформи

1.3.6 Еквід

За допомогою «Еквід» (рис. 1.5) можна додати функціональність інтернет-магазину на будь-який сайт або в соціальні мережі.

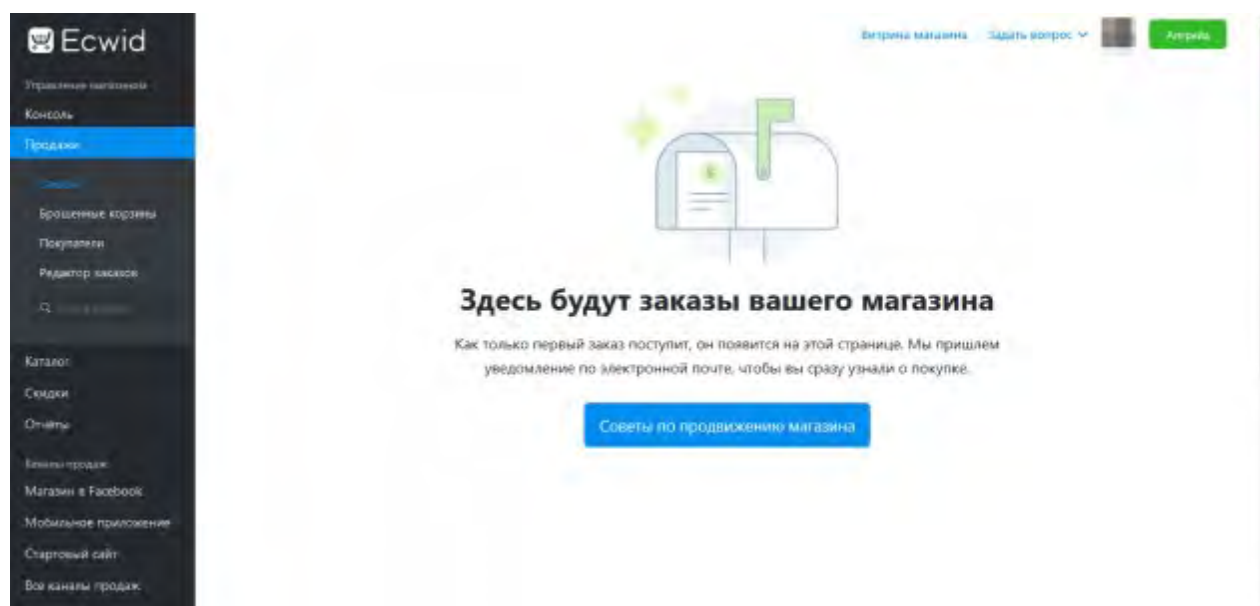


Рисунок 1.5 – Панель управління «Еквід»

Основні особливості «Еквід». «Еквід» являє собою SaaS-платформу. Власник магазину після реєстрації може налаштувати віджет, додати товари, підключити платіжні модулі і встановити віджет на сайт. «Еквід» можна інтегрувати з «ВКонтакте», Facebook, Google Покупки, eBay.

Віджет «Еквіда» легко адаптувати до дизайну сайту. Для цього можна

використовувати шаблонні стилі або самостійно відредагувати CSS в особистому кабінеті на сайті платформи.

За допомогою «Еквід» можна торгувати в інтернеті, якщо у вас немає сайту і відповідних акаунтів в соцмережах. Для цього скористайтеся стартовим сайтом. Його зовнішній вигляд і контент можна змінювати на свій розсуд.

Є безкоштовний тариф, користувачі якого можуть продавати до 10 товарів. Професійні тарифи коштують від 15 доларів в місяць. Переглянути переваги і недоліки можна на таблиці 1.6.

Таблиця 1.6 – Переваги і недоліки «Еквід»

Переваги	Недоліки
Підтримує актуальні для України платіжні сервіси	Сайт і віджет виглядають шаблонно, для редагування дизайну потрібні спеціальні знання
Є безкоштовний тариф до 10 товарів	Висока вартість професійних тарифів для невеликих магазинів
Можна інтегрувати з будь-яким сайтом або соцмережами	Мізерні настройки SEO стартового сайту

1.3.7 Recommerce

Recommerce – SaaS-платформа, на якій можна швидко створити функціональний інтернет-магазин.

Головні особливості Recommerce. Recommerce відрізняється дружньою до користувача і інтуїтивно зрозумілою панеллю керування. Щоб створити інтернет-магазин, не потрібні спеціальні знання.

Власнику доступні гнучкі настройки:

- підключення домену та SSL-сертифіката;
- налаштування кошика;
- функція замовлення зворотного дзвінка;
- підключення зовнішніх сервісів через меню додавання

користувачького JavaScript і HTML;

– повідомлення про події через SMS.

Користувач може вибрати безкоштовний дизайн-шаблон, а також змінити його за допомогою будь-якого фону, кастомного CSS, відповідних налаштувань шаблону.

На відміну від деяких конструкторів, Rescommerce створений з урахуванням вимог пошукових систем. Платформа забезпечує користувачам гнучкі налаштування SEO без додаткових витрат.

Є безкоштовний тестовий період 14 днів. Вартість платних тарифів починається з 8 доларів на місяць. Переглянути переваги і недоліки можна на таблиці 1.7.

Таблиця 1.7 – Переваги і недоліки Rescommerce

Переваги	Недоліки
Цілодобова підтримка, стабільний хостинг	Немає безкоштовного тарифу
Можливість змінювати зовнішній вигляд і функціональність сайту за допомогою HTML, CSS і JavaScript	Без додаткової кастомізації дизайн виглядає шаблонно
Є вигідний тариф для невеликих магазинів	Немає повноцінного drag-n-drop конструктора

1.3.8 Висновки з дослідження засобів розробки інтернет-магазинів

Проаналізувавши двигуни, можна зробити висновок, що найкращим двигуном буде той, який відповідає потребам користувача, масштабу бізнесу і досвіду роботи з веб-сайтами. Якщо присутні глибокі знання в веб-розробці або є можливість платити стороннім розробникам, краще скористуватися CMS з відкритим кодом, наприклад, OpenCart, WooCommerce і їх платними аналогами. Однак, треба пам'ятати, що умовно безкоштовні движки з відкритим кодом зазвичай обходяться власникам дорожче платних SaaS-платформ.

Якщо немає можливості витратити час і гроші на технічні питання, краще обрати одну з SaaS-платформ. Для торгівлі в соцмережах підійде «Еквід». Щоб швидко створити повноцінний якісний сайт, краще скористатися Recommerce, MotoCMS або аналогічної платформою.

У таблиці 1.8 наведено узагальнені результати дослідження засобів розробки вебсайтів інтернет-магазинів [3].

Таблиця 1.8 - Узагальнені результати дослідження засобів розробки вебсайтів інтернет-магазинів

Платформа	SEO підтримка	Кастомізація дизайну	Розширення функціональності	Адаптація до регіонального ринку	Приховані платежі
WooCommerce	+	+	+	—	—
OpenCart	+	+	+	—	—
Еквід	+—	+—	+—	—	—
Recommerce	+	+	+	+	+
MotoCMS	+	+—	+—	+	+
Wix	—	+—	+—	+	—
Prom.ua	—	—	—	+	—

Таким чином, наявність недоліків кожного з засобів обумовлює доцільність створення нових моделей розробки вебсайтів.

1.4 Аналіз вебсайтів інтернет-магазинів для бронювання місць у об'єктах розміщення

Останнім часом широкого використання набули інтернет-магазини для бронювання місць у об'єктах розміщення. Такі інтернет-магазини представляють собою веб-сайти, що надають користувачу можливість пошуку готелів за заданими критеріями (дата, місце розташування, ціна, кількість відпочиваючих та ін.).

Один із таких сайтів – є сайт Hotellook, який пропонує величезну кількість варіантів бронювання - це сотні тисяч готелів, хостелів, гостьових домівок та кілька мільйонів апартаментів. Пропозиція самого вигідного житла ідеально лише після конкурентного порівняння цін за 70 агентствами. Для того, щоб заощадити на бронюванні бажаного готельного закладу, Hotellook пропонує підписати зміни цін, так як ціни щодня змінюються. При плануванні подорожі заздалегідь можна заощадити до 60% від вартості проживання.

Hotellook (рис. 1.6) веде підбори міст, готелей і романтичних місць. На всіх популярних пристроях також діють акції. Клієнт угоди бачить остаточну ціну, без прихованих платежів. Тревел сервіс представляє детальний опис номерів із фотографіями, рейтингами та відзивами реальних жильців.

На сайті є розділ допомоги, де зібрані відповіді на найважливіші та найпопулярніші питання подорожей, а також рекомендації щодо вибору готелів, керівництво користувача Hotellook. Для клієнтів, яким важливо мати доступ до сервісу у будь-якій ситуації, є мобільний додаток для iOS та Android.

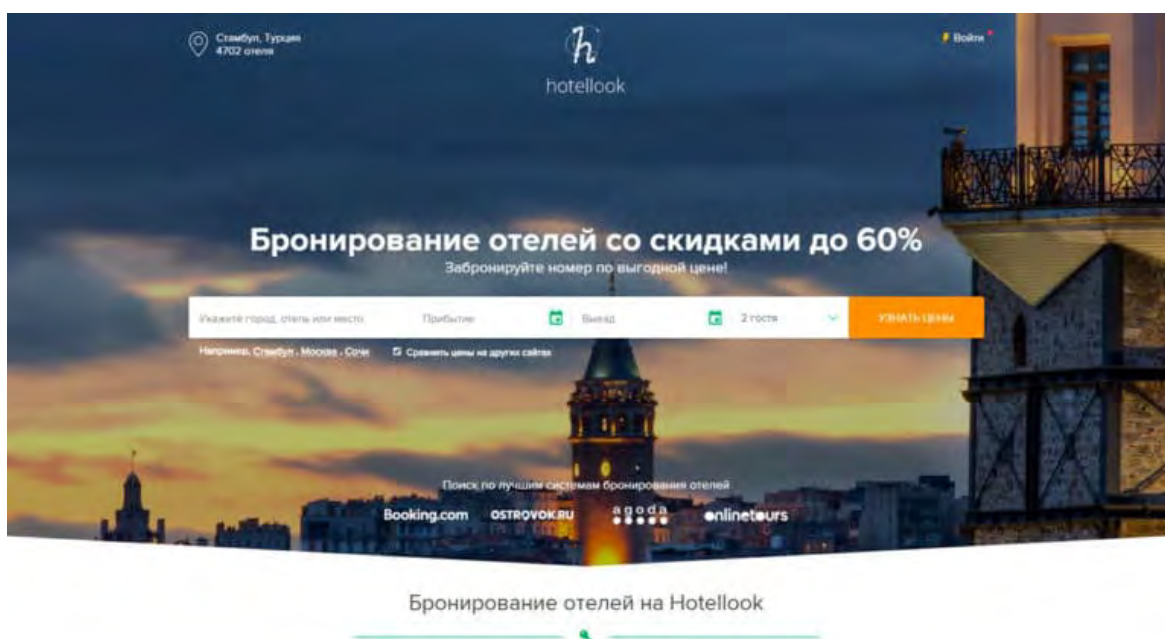


Рисунок 1.6 – Зовнішній вигляд сайту Hotellook.com

Airbnb (рис. 1.7) допоможе знайти кращі пропозиції готелів за допомогою безлічі фільтрів, які дозволяють звузити область пошуку. У каталозі представлені готелі з високими оцінками в центральних районах міста. Це підходить більшості мандрівників, які шукають готель або хостел для відпочинку на кілька днів.

Результати пошуку готелів також включають квартири, окремі кімнати, студії з двома ліжками вартість від 1000 у.о. за одну ніч. Серед популярних напрямків виділяються: Токіо, Париж, Лондон, Лос-Анджелес і Барселона.

Кожен номер має докладний опис: наявність телебачення, Wi-Fi, сніданків, окремої кухні і розташування на карті. За відгуками можна складе повне враження про представлене житло. Виберіть необхідну дату заїзду на календарі і сайт Airbnb автоматично порахує всі витрати на проживання в у різних країнах [7].

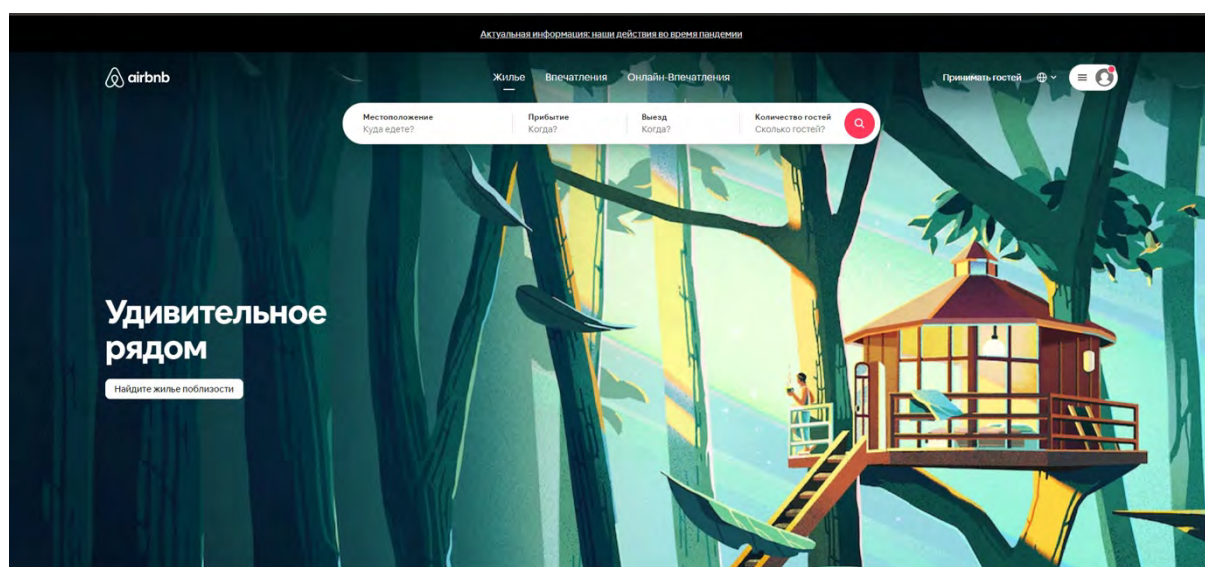


Рисунок 1.7 – Зовнішній вигляд сайту Airbnb.ru

Найбільш популярним і використовуваним у світі веб-сайт, що надає послуги з продажу тарифних планів готельних господарств, є Booking.com – відомий міжнародний сервіс для бронювання номерів (рис. 1.8) [38].

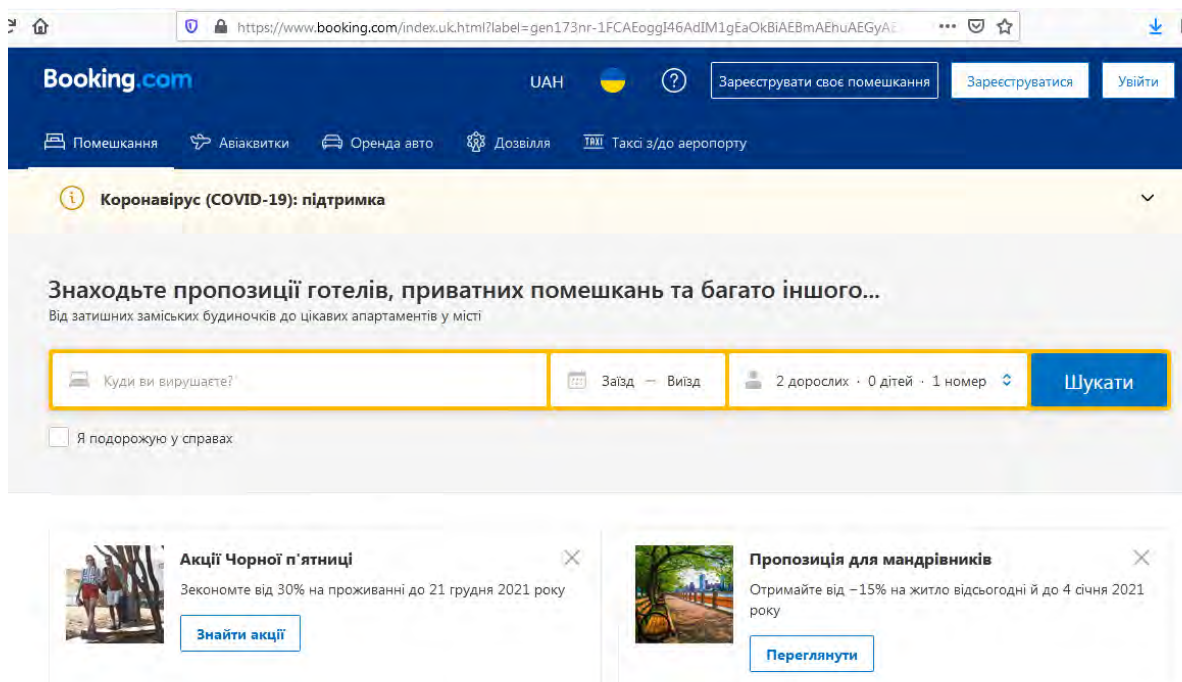


Рисунок 1.8 – Зовнішній вигляд сайту Booking.com

До переваг сервісу варто віднести такі:

- широке використання по усьому світу;
- велика база даних готелів;
- зручний інтерфейс.

Недоліками сайту є:

- сервіс занадто узагальнений;
- орієнтований здебільшого на пошук готелей;
- відсутні індивідуальні налаштування.

Іншим сайтом є Hotels.com – веб-сервіс для бронювання номерів, схожий на Booking.com, але менш популярний (<https://ua.hotels.com/>, рис. 1.9) [39].

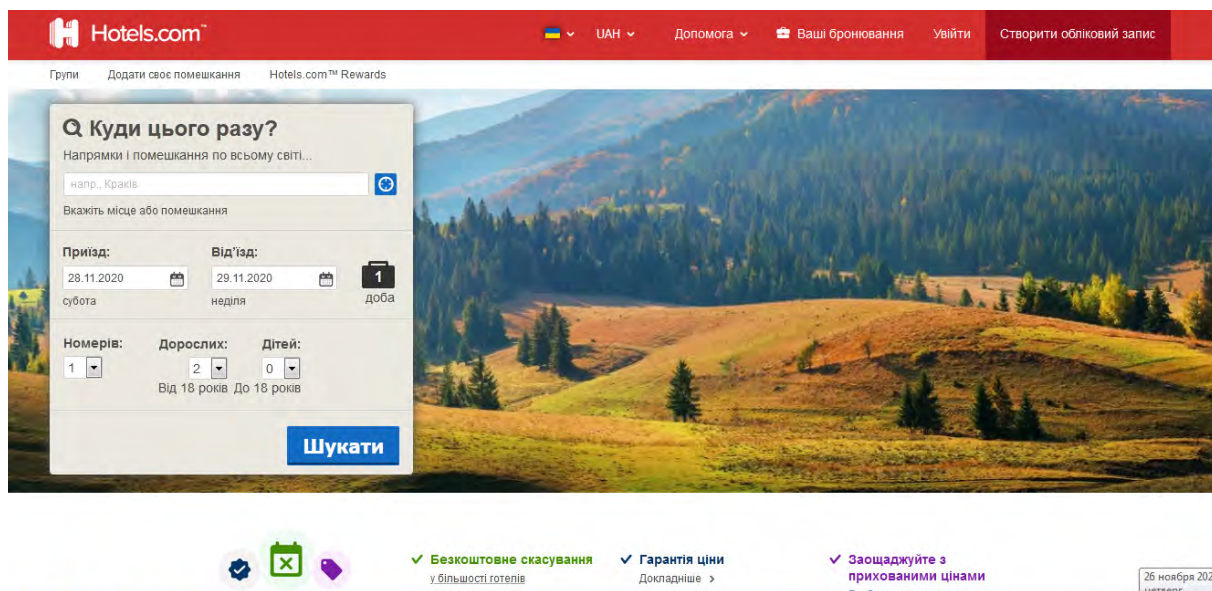


Рисунок 1.9 – Зовнішній вигляд сайту Hotels.com

Переваги:

- система знижок;
- багатомовність;
- підтримка замовника інформацією на кожному етапі, включаючи email-розсилку;

Недоліки:

- занадто узагальнений;
- орієнтований здебільшого на пошук готелей;
- відсутність контролю цінних категорій номерів.

Подібні сайти є досить рентабельними для їх власників, які отримують комісійну винагороду від надавачів готельних послуг. Проте розробка, експлуатація та модифікація таких інтернет-магазинів вимагає суттєвих матеріальних та часових витрат.

Складність розробки і модифікації подібних сайтів під поточні потреби ринку та потреби користувачів обумовлює доцільність створення інформаційних моделей, що спростять розробку таких інтернет-магазинів та забезпечать можливість їх модифікації без необхідності внесення суттєвих змін у програмний код.

1.5 Висновки за розділом 1

Досліджено процес розробки вебсайтів. Визначено, що у наш час, вебсайти є невід’ємною частиною життя майже кожної людини. Особливо популярними стали інтернет магазини, а оскільки зараз ще і пандемія, то їх цінність зростає багатократно. Саме тому багато людей орієнтуються на перенесення свого бізнесу в інтернет, створюючи інтернет магазини, де можуть продавати свою продукцію. Правильний вибір підходу для розробки власного веб-сайту інтернет магазину може зберегти багато часу та фінансів.

Досліджено засоби розробки вебсайтів інтернет-магазинів. Було досліджено, що є два варіанти в розробці власного інтернет магазину:

- використовуючи мови програмування та наймаючи дизайнерів, можна самостійно створити бажаний веб сайт. Але це при наявності необхідних навичок програмування та бюджету, який розрахований на дизайнерські послуги;

- використовуючи готові рішення, двигуни та конструктори для створення того чи іншого інтернет магазину по обраному шаблону. Користувачу не потрібно вміти програмувати, лише мати достатній бюджет на покупку та підтримку того чи іншого дизайнерського шаблону, технічну частину повністю реалізує обрана користувачем технологія (платформа) для створення веб сервісу.

Серед двигунів також існує багато варіантів. Після глибокого дослідження можна зробити висновок, що вибір технології повністю залежить від масштабів та цілей інтернет магазину. Кожна технологія налічує свої переваги та недоліки, саме тому перед тим, як приймати рішення в сторону тієї чи іншої платформи, треба зробити детальний аналіз. Таблиця порівняння, яка була створена під час дослідження, є гарним інструментом для прийняття остаточного рішення.

Проаналізовано вебсайти інтернет-магазинів. Досліджено вебсайти інтернет-магазинів для бронювання місць у об’єктах розміщення.

Визначено, що складність розробки і модифікації подібних сайтів під поточні потреби ринку та потреби користувачів обумовлює доцільність створення інформаційних моделей, що спростять розробку таких інтернет-магазинів та забезпечать можливість їх модифікації без необхідності внесення суттєвих змін у програмний код.

2 ВИБІР ТА ОБҐРУНТУВАННЯ СТРУКТУРИ СИСТЕМИ, ЯКА ПРОЄКТУЄТЬСЯ

2.1 Дослідження та аналіз вимог до програмного продукту

Метою розробки є створення програмного продукту для перегляду інформації про номери об'єкта розміщення, їх бронювання з урахуванням обраного тарифного плану, що надається готельним господарством та може передбачати наявність додаткових послуг, зокрема харчування та ін..

Діловий контекст розробки полягає в тому, що по завершенню реалізації проєкту власник матиме готовий програмний продукт, з якого зможе отримати матеріальну вигоду, автоматизувавши процес продажу та бронювання місць у номерах об'єкта розміщення.

Розроблюване програмне забезпечення (ПЗ) Booking Fast має задовольняти таким вимогам до функціоналу:

- ПЗ «Booking Fast» має складатись з двох основних частин:
 - 1) back end – частина, що надає комплекс можливостей для моніторингу замовлень та налаштувань модулю бронювання;
 - 2) front end – частина, яку бачить звичайний користувач, та яка надає можливості перегляду каталогу номерів та замовлення;
- ПЗ «Booking Fast» має розташовуватись на віддаленому сервері, що витримує одночасне підключення не менше 1000 клієнтів, та здатен розпаралелювати їхні запити;
- ПЗ «Booking Fast» має забезпечувати функцію бронювання номерів через веб-браузер;
- ПЗ «Booking Fast» має бути створеним на базі ASP.NET;
- всі функції по збереженню даних клієнта, даних модулю бронювання, мають бути реалізовані з використанням СКБД SQL Server;
- ПЗ «Booking Fast» має забезпечувати функції прив'язки зображень до таких сутностей: номер, категорія номеру, будинок(корпус).
- ПЗ «Booking Fast» має підтримувати облік планів харчування;

- ПЗ «Booking Fast» має надавати можливість замовляти трансфер разом із бронюванням номеру;
- ПЗ «Booking Fast» має підтримувати функцію мультивалютності, та контролю за документом курсу валют;
- ПЗ «Booking Fast» має надавати можливість перемикання між декількома мовами на вибір клієнта;
- вибір номерів має забезпечуватись через систему фільтрів: кількість відпочиваючих, період відпочинку, готель та місто (для мульти-готельного модулю);
- ПЗ «Booking Fast» має підтримувати наступні налаштування: вибір мов перекладу модулю, кастомізація перекладів, вибір відображуваних фільтрів, відключення та підключення модулів (трансферів, оплати, людей в замовленні, знижок), налаштування розмітки полігонів на карті чи схемі для будинків та номерів, налаштування цін та курсів валют, налаштування періодів зміни цін та сезонів відпочинку;
- в разі системного збою ПЗ «Booking Fast», за допомогою агента контролю якості має відправляти повідомлення про системний збій адміністрації.

Система «BF» має задовольняти таким вимогам до надійності:

- ПЗ “BF” має безперебійно працювати в умовах великих навантажень та бути стійким до помилок;
- ПЗ “BF” має бути захищеним від “SQL-ін’єкцій”, та вставок з JavaScript – коду;
- дані різних клієнтів(замовників) не мають перетинатись між собою.

Вимоги до інтерфейсу: інтерфейс системи повинен бути простим та інтуїтивно-зрозумілим для користувачів системи. Вся інформація сайту повинна бути чітко структурована і поділена на розділи.

Експлуатаційні вимоги: для запуску серверної частини веб-застосунку, сервер має містити сервер IIS, з встановленою СУБД SQL Server. Сервер повинен мати функцію «SMTP». Для прискорення швидкодії, сервер

повинен мати можливість запускати веб-додаток в декількох потоках. Для запуску клієнтської частини веб-застосунку, користувач повинен мати одну з поширених версій веб-браузеру.

Вимоги продуктивності: система повинна підтримувати до 250 завантажень в секунду. Система та всі її компоненти повинні працювати стабільно і з максимально можливою швидкістю, не нижче ніж 3 секунди на завантаження сторінки при нормальних умовах з боку клієнтського термінала. Реєстрація користувача з занесенням його даних в базу даних зареєстрованих користувачів і відправленням йому електронного листа повинна займати не більше 5 хвилин. Система повинна підтримувати до 5 запитів на реєстрацію в секунду.

Вимоги до програмної документації: склад програмної документації повинен включати в себе технічне завдання, текст сайту (код), керівництво програміста, керівництво оператора та при необхідності інші програмні документи.

Розроблений ПП має зберігатися на DVD диску розміром 8ГБ. Данні повинні зчитуватися на платформі Windows.

Для запуску розроблюваного ПП треба:

- процесор частотою 2 ГГц;
- місце на жорсткому диску – 500 Мб;
- відеокарта з пам'яттю 256 Мб;
- оперативна пам'ять – 1Гб;
- маніпулятор миша.

Для функціонування системи необхідною є наявність сервера, на якому встановлені сервер IIS та SQL Server 2012. Для роботи програми зі сторони клієнта необхідна ОС Microsoft Windows зі встановленим середовищем виконання CLR .NET Framework 2.0.

2.2 Вибір інструментарію розробки програмного забезпечення

Для розробки веб-сайту з продажу тарифних планів готельних господарств обрано середовище розробки Microsoft .NET, яке є відкритим мовним середовищем. Це означає, що поряд з мовами програмування, включеними в середовище фірмою Microsoft Visual C++ .NET, Visual C# .NET, J# .NET, Visual Basic .NET, в нього можуть добавлятися будь-які мови програмування, при дотриманні певних умов. Головне обмеження, яке можна вважати і головною достоїнністю, полягає в тому, що всі мови, що включаються в середовище розробки Visual Studio .NET, повинні використовувати єдиний каркас — .NET Framework. Завдяки цьому досягаються багато властивостей:

- легкість використання компонентів, розроблених на різних мовах;
- можливість розробки декількох частин однієї програми на різних мовах;
- можливість безшовного налагодження такого застосування;
- можливість писати клас на одній мові, а його нащадків – на інших мовах.

Єдиний каркас призводить до зближення мов програмування, дозволяючи разом з тим зберігати їх індивідуальність і наявні в них достоїнності.

Платформа Microsoft .NET забезпечує середовище виконання програм і визначає особливості розробки та виконання програмного коду на ПК. Вона має засоби організації взаємодії з операційною системою і прикладними програмами, методи доступу до баз даних, засоби підтримки мережних додатків, мови програмування, а також засоби розробки веб-сайтів.

Середовище розробки Visual Studio .NET підтримує розробку веб-сайтів за допомогою технології ASP.NET (Active Server Pages) – технології створення веб-додатків і веб-сервісів від компанії Майкрософт. Хоча ASP.NET бере свою назву від старої технології Microsoft ASP, вона значно

від неї відрізняється. Microsoft повністю перебудувала ASP.NET, ґрунтуючись на Common Language Runtime (CLR), яка є основою всіх додатків Microsoft .NET. Розробники можуть писати код для ASP.NET, використовуючи практично будь-які мови програмування, що входять в комплект .NET Framework (C #, Visual Basic.NET і JScript .NET). ASP.NET має перевагу у швидкості в порівнянні зі скриптовими технологіями, так як при першому зверненні код компілюється і поміщається в спеціальний кеш, і згодом тільки виконується, не вимагаючи витрат часу на парсинг, оптимізацію, і т.ін.

2.3 Розробка інформаційних моделей створення вебсайтів інтернет-магазинів

Як відзначено вище, існуючі методи і засоби розробки веб-сайтів на основі платформ-конструкторів характеризуються високою складністю автоматично згенерованого коду і складність доопрацювання, а також прив'язка до певної платформи. Створення інтернет-магазинів без застосування платформ-конструкторів пов'язано з високою вартістю розробки, довготривалістю, а також складністю модифікації сайту у майбутньому.

Для усунення зазначених недоліків в цьому розділі розробляється набір інформаційних моделей для створення спеціалізованих веб-сайтів з продажу тарифних планів (зокрема, тарифних планів готельних господарств), що дозволять скоротити час створення інтернет-магазинів без використання платформ-конструкторів, а також забезпечити простоту їх модифікації у майбутньому.

Структурні діаграми системи відображають основні її структурні елементи – системі сутності та відношення між ними. Моделі, подані у вигляді таких діаграм, є базою для створення динамічної поведінки системи. Наявність таких діаграм при розробці системи дозволяє забезпечити у

майбутньому зручність її модифікації. Структурні діаграми для розробки програмних систем у вигляді веб-сайтів інтернет-магазинів бронювання місць у об'єктах розміщення подамо у вигляді рис. 2.1 – 2.3, на яких зображено діаграми класів, пакетів та компонентів, відповідно.

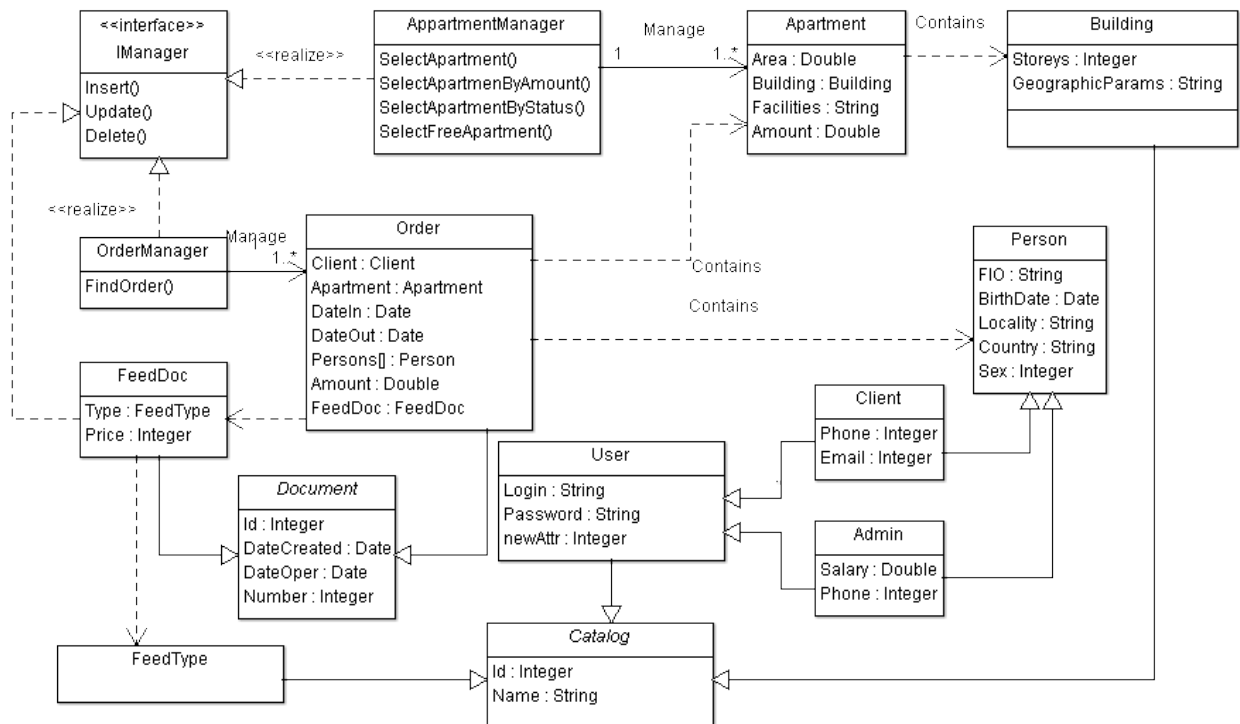


Рисунок 2.1 – Діаграма класів, представлена у нотації UML 2.5

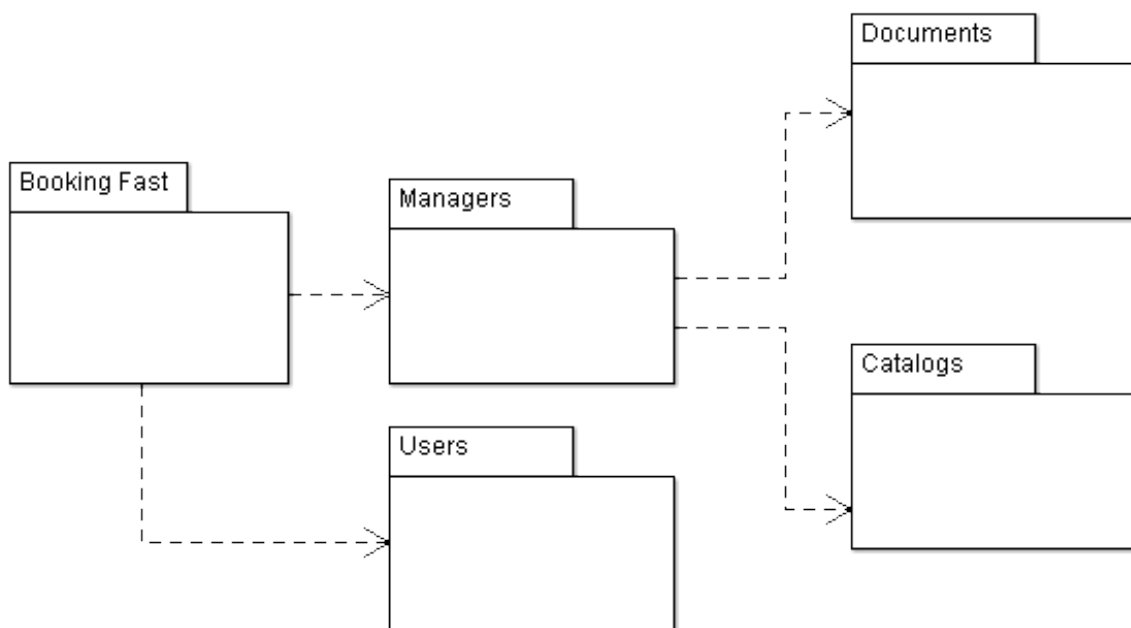


Рисунок 2.2 – Діаграма пакетів

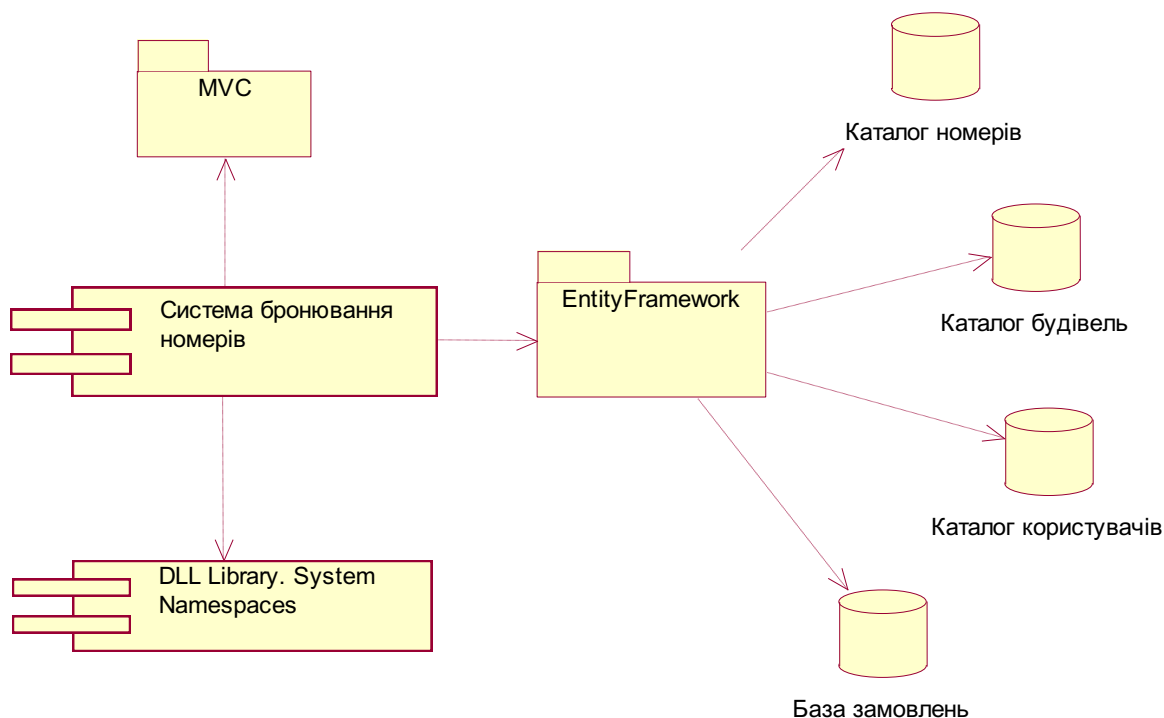


Рисунок 2.3 – Діаграма компонентів

Діаграми поведінки відображають процеси, що відбуваються у модельованому середовищі. Діаграми поведінки у програмних системах веб-сайтів інтернет-магазинів з продажу тарифних планів готельних господарств пропонується подати у вигляді діаграм станів (рис. 2.4, 2.5) та діаграм прецедентів (рис. 2.6, 2.7), представлених у нотації UML 2.5.

Діаграми станів (рис. 2.4 та 2.5) відображають можливі стани, в яких може знаходитися конкретний об'єкт системи, а також зміни стану об'єкту, які відбуваються в результаті впливу деяких подій на цей об'єкт.

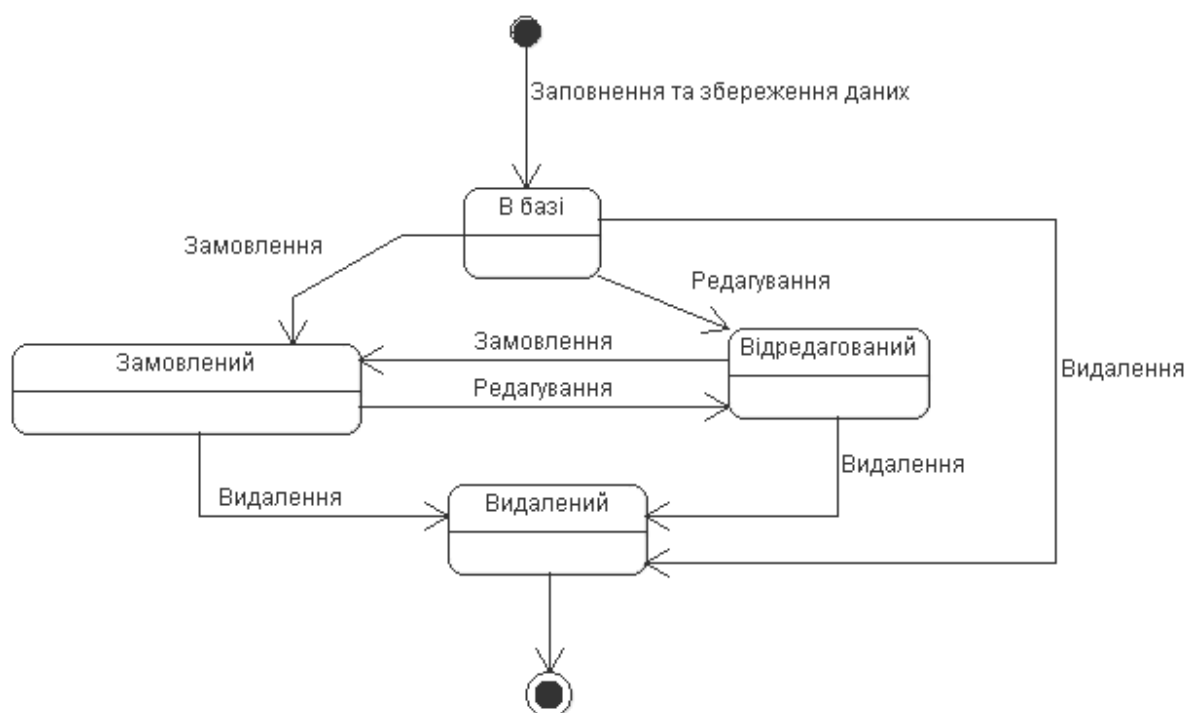


Рисунок 2.4 – Діаграма стану для номера

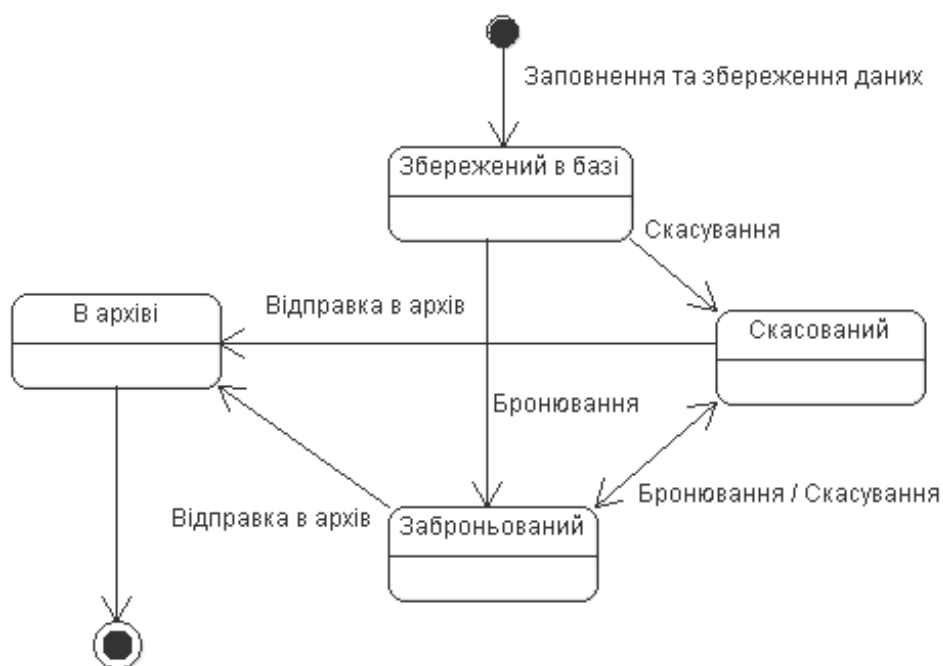


Рисунок 2.5 – Діаграма станів для замовлення

Діаграми прецедентів інтернет-магазинів з продажу тарифних планів готельних господарств (рис. 2.6, 2.7) застосовуються для моделювання вигляду системи з точки зору прецедентів (варіантів використання). Даний

вид діаграм використовується для моделювання динамічних аспектів інтернет-магазинів з продажу тарифних планів.

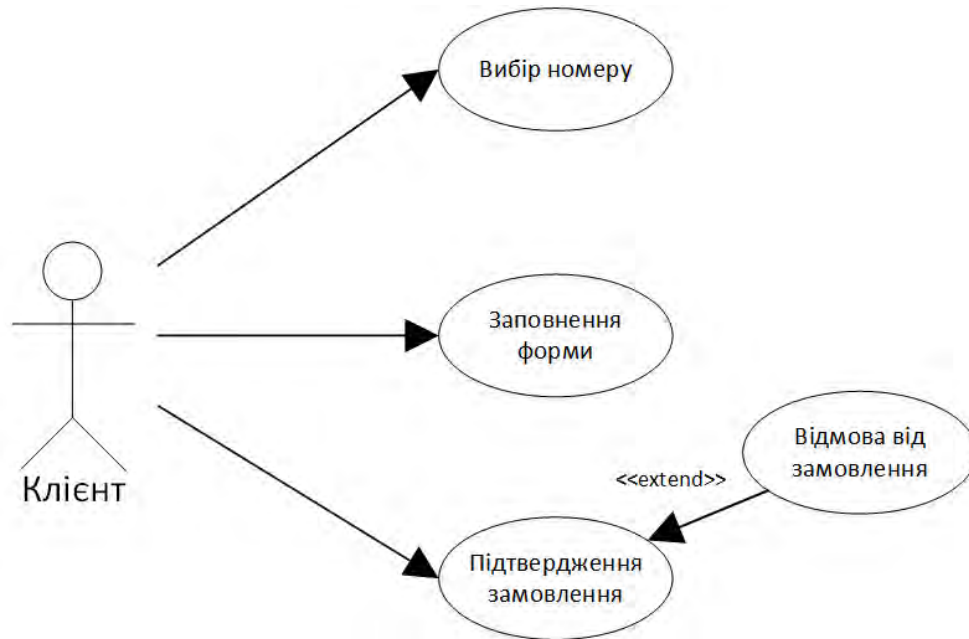


Рисунок 2.6 – Діаграма прецедентів «Замовлення номеру»

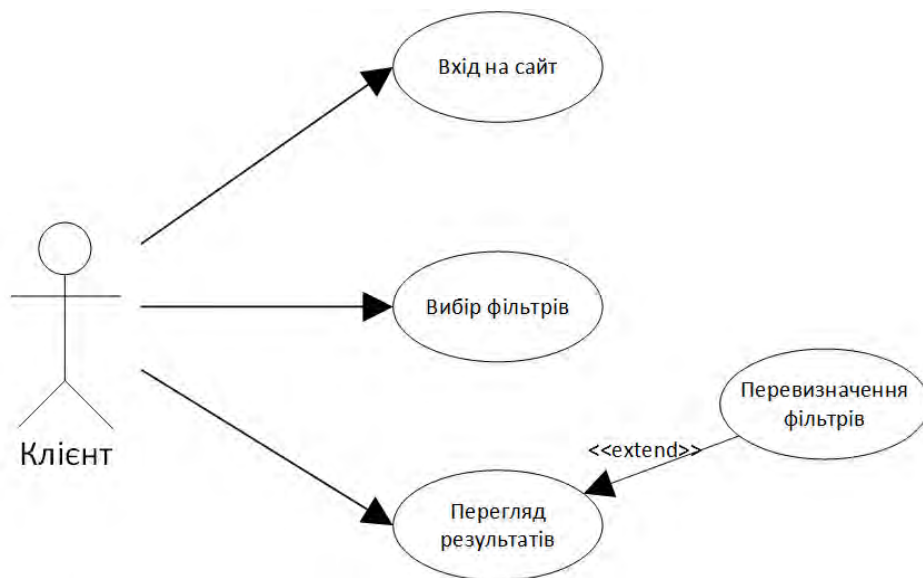


Рисунок 2.7 – Діаграма прецедентів «Пошук номеру»

Інформаційні моделі, представлені за допомогою діаграм взаємодії, відображають структурні аспекти взаємодії об'єктів у програмних системах

веб-сайтів інтернет-магазинів з продажу тарифних планів готельних господарств.

Послідовність взаємодії об'єктів, а також всі структурні відносини між об'єктами, що беруть участь в цій взаємодії, представимо за допомогою діаграм кооперації «Клієнт-система» (рис. 2.8) та «Адміністратор-система» (рис. 2.9).

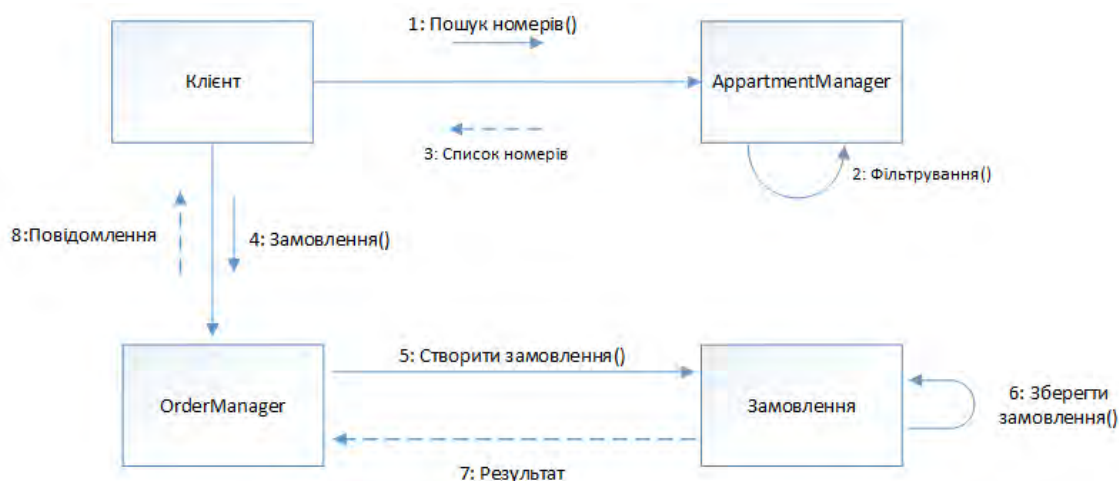


Рисунок 2.8 – Діаграма кооперації Клієнт-система

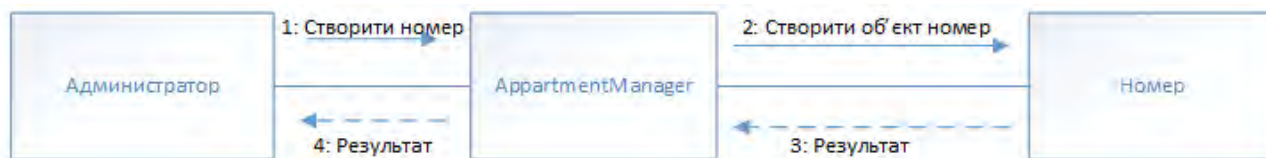


Рисунок 2.9 – Діаграма кооперації Адміністратор-система

Діаграма послідовності показує учасників взаємодій у програмних системах веб-сайтів інтернет-магазинів, а також послідовність повідомлень, якими обмінюються ці учасники. На діаграмах послідовності (рис. 2.10 та 2.11) зображуються виключно ті об'єкти, що безпосередньо беруть участь у взаємодії і не показуються можливі статичні асоціації з іншими об'єктами, при цьому ключовим моментом є саме динаміка взаємодії об'єктів у часі.

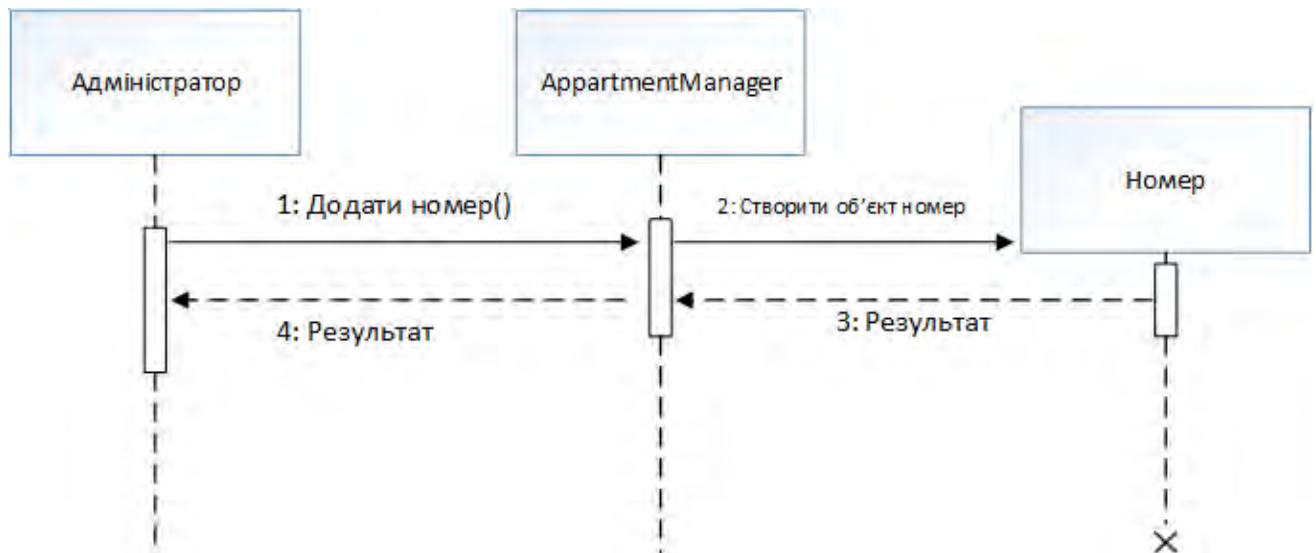


Рисунок 2.10 – Діаграма послідовності Адміністратор-система

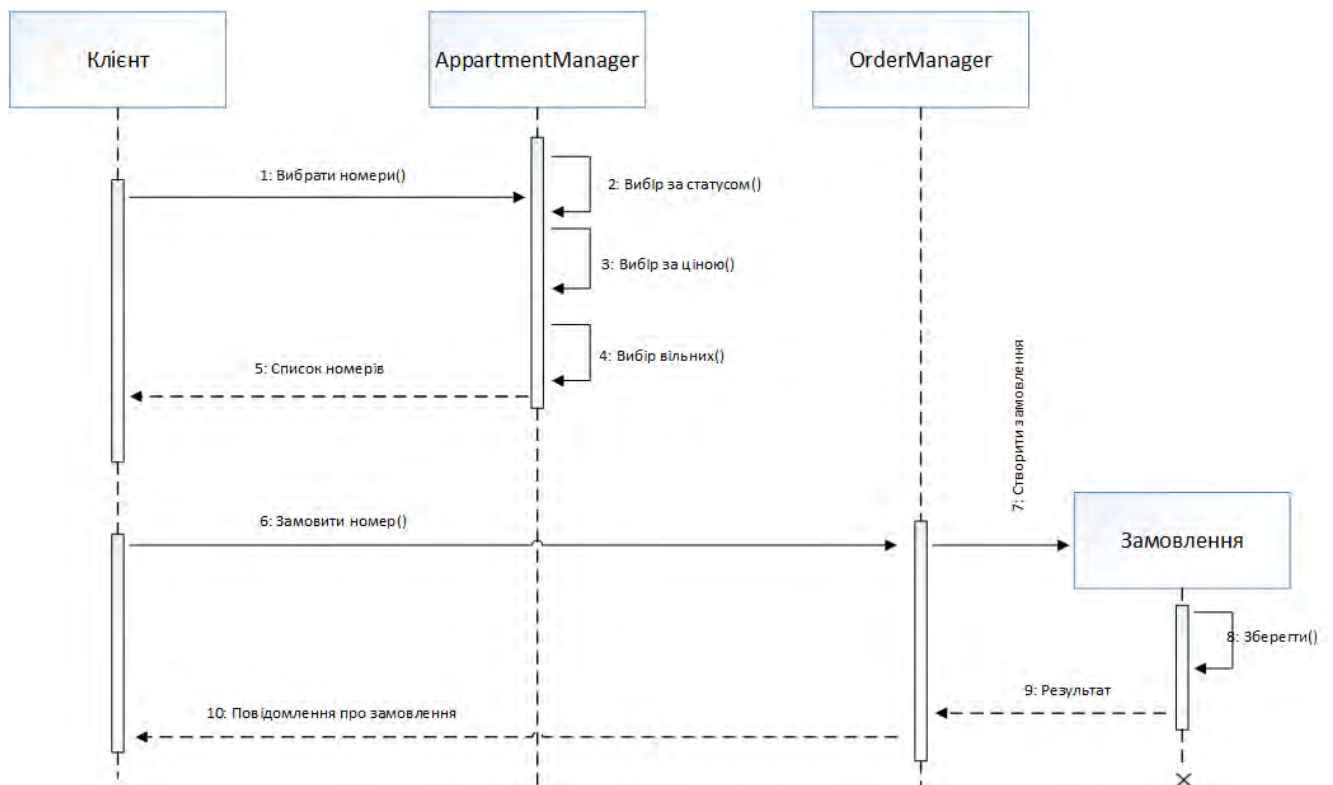


Рисунок 2.11 – Діаграма послідовності Клієнт-система

2.4 Висновки за розділом 2

Досліджено та проаналізовано вимоги до програмного забезпечення вебсайту інтернет-магазину для бронювання місць у об'єктах розміщення. Проаналізовано інструментарії розробки програмного забезпечення.

Порівнявши відомі середовища розробки програмних засобів, оцінивши їх переваги та недоліки, було прийнято рішення про використання для розробки програмного забезпечення інтернет-магазину для бронювання місць у об'єктах розміщення технології ASP.NET, оскільки вона є досить зручною для використання, дозволяє створювати програмні продукти різного призначення, та має багато бібліотек, що спрощують реалізацію поставлених завдань при створенні інтернет-магазину для бронювання місць у об'єктах розміщення.

Розроблено модель програмного забезпечення інтернет-магазину для бронювання місць у об'єктах розміщення, що представляє собою набір діаграм у нотації UML, які дозволяють розробнику створювати програмне забезпечення відповідно до вимог замовника та спрощують розробку інтернет-магазинів без необхідності використання спеціалізованих платформ-конструкторів та забезпечують можливість модифікації розроблених сайтів у майбутньому при зміні вимог користувачів.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Розробка загальної схеми програми

Загальну схему програми наведено на рис. 3.1.

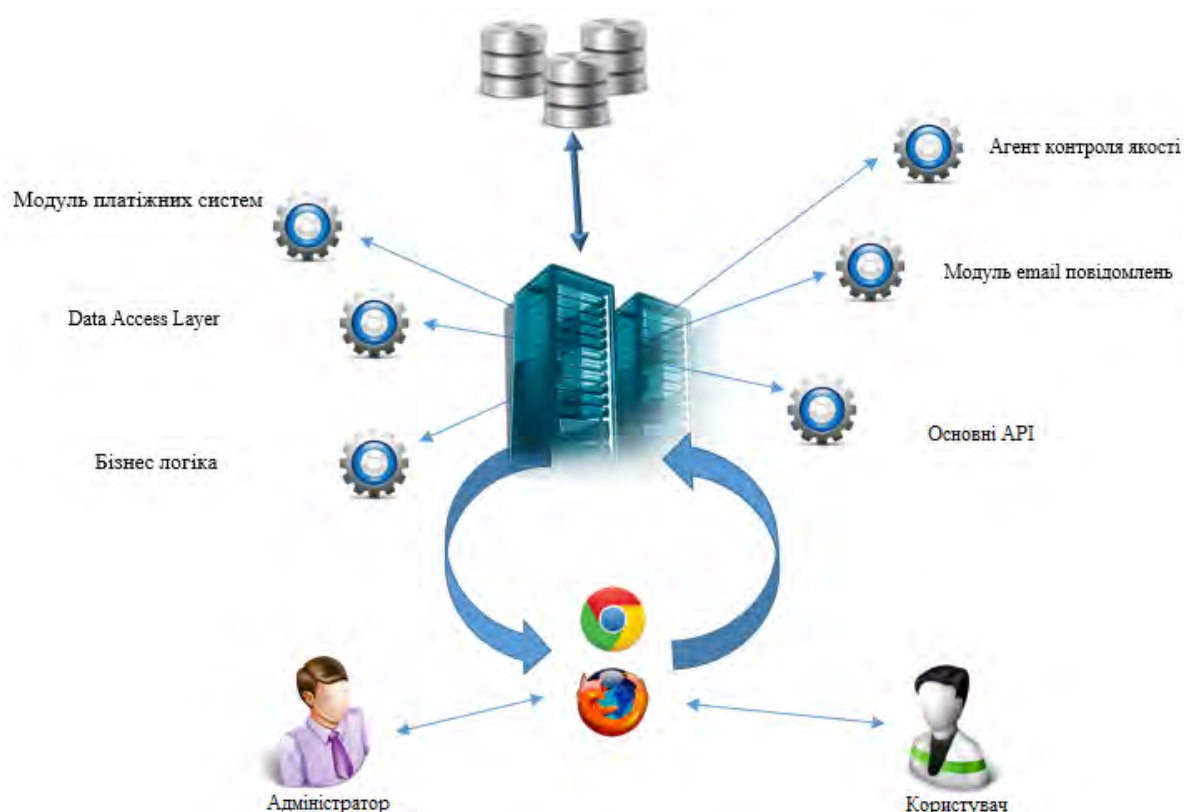


Рисунок 3.1 – Загальна схема системи

Як видно з рис.3.1, користувачі або адміністратор системи отримують доступ до програмного забезпечення через веб-браузер (Mozilla Firefox, Google Chrome або інші). Основний функціонал системи (бізнес-логіка, програмні модулі системи та ін.) та набори користувальницьких даних зберігаються на сервері. Організація взаємодії користувачів з системою відбувається за допомогою глобальної мережі Internet.

3.2 Архітектура веб-застосунку на основі обраного підходу

Як відзначено вище, розроблене програмне забезпечення подано у вигляді веб-сайту. Для побудови архітектурного каркасу програми використовується схема проектування Model-view-controller (MVC) (рис. 3.2).

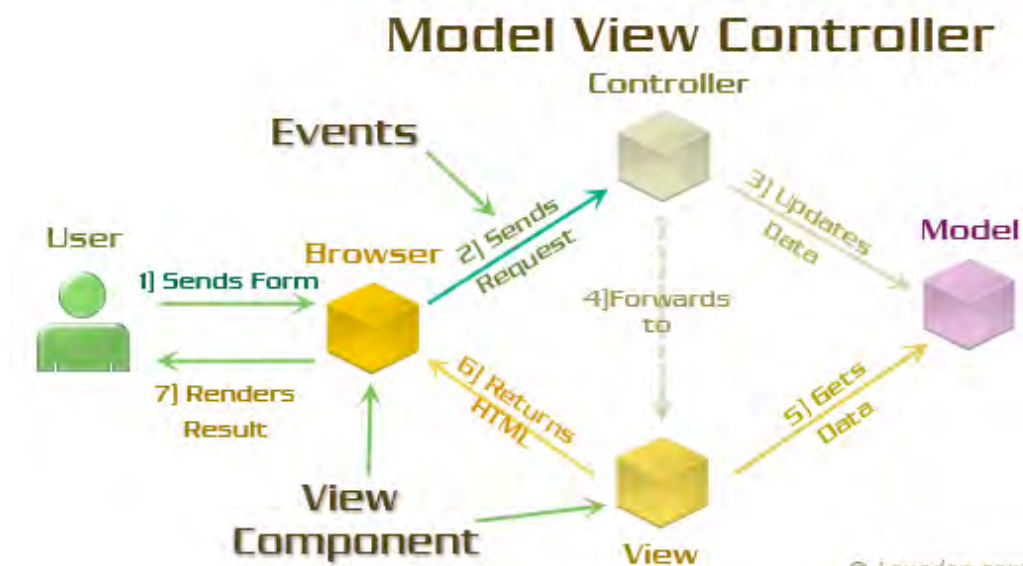


Рисунок 3.2 – Схема проектування Model-view-controller

У веб-застосунку, що розроблюється, роль model містить, так названу, «бізнес-логіку» – обробку і верифікацію даних, представляє внутрішнє уявлення системи. Не повинна напряду взаємодіяти з користувачем.

View – це графічна оболонка веб-додатку, що реалізується через веб-сторінки.

А controller – виконує роль зв'язної ланки між Моделлю і Видом. Отримує дані від користувача, передає їх Моделі, після чого виводить користувачу у Виді отримані результати.

3.3 Розробка структури проекту

Проект має наступну файлову структуру (рис 3.3).

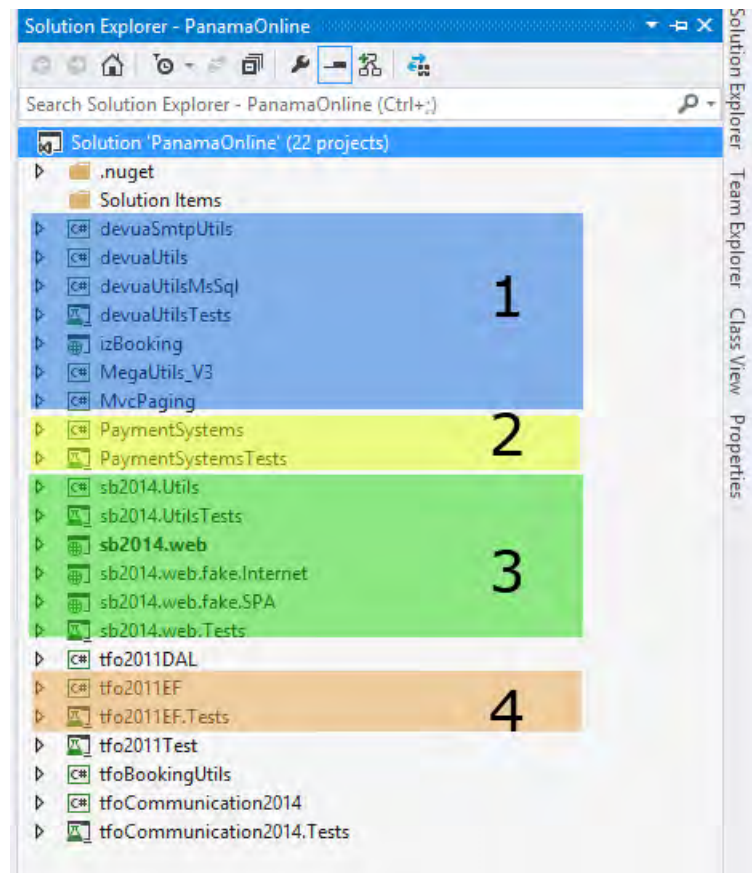


Рисунок 3.3 – Файлова структура проекту

На рис. 3.3 файли проектів утиліт загального призначення згруповано під цифрою «1», проекти для проведення онлайн платежів – під цифрою «2», проекти, що безпосередньо відносяться до веб- застосунку (утиліти, веб-додаток, тести) – під цифрою «3», проекти для взаємодії з базою даних – під цифрою «4».

Структура файлів та папок безпосередньо веб- застосунку зображена на рис. 3.4. Середовище Visual Studio автоматично генерує базову структуру

проекту MVC. Visual Studio підтримує зручну навігацію по коду, розширений пошук файлів та виразів.

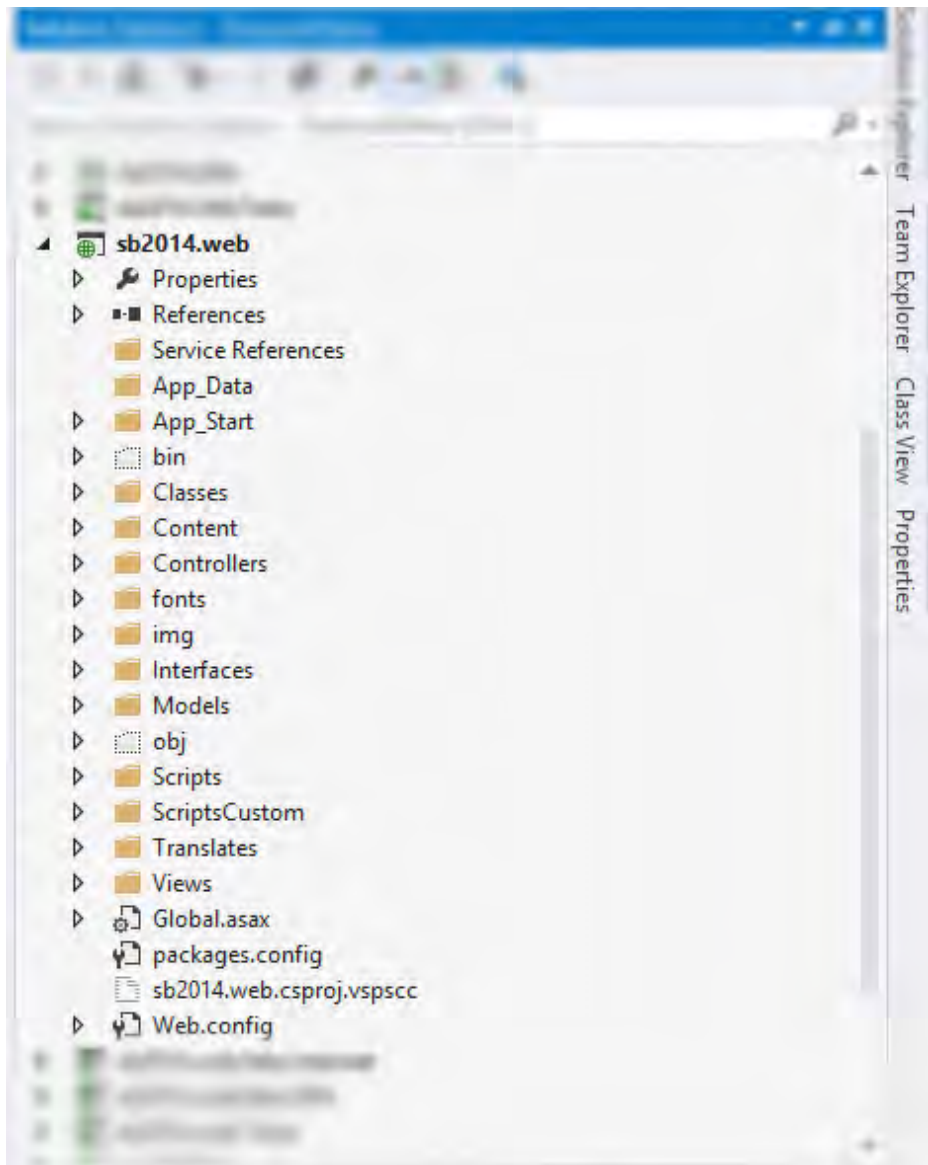


Рисунок 3.4 – Структура веб-застосунку

Розподілення файлів проекту на логічні структури по каталогах є запорукою гарної структуризації та ізольованості підсистем. Проект ASP.NET складається з багатьох типів каталогів та файлів. Призначення файлів та каталогів наведені в табл. 3.1.

Таблиця 3.1 – Опис структури проекту

Назва елементу	Призначення
Properties	Конфігурація проекту
References	Посилання на зовнішні компоненти
Service References	Посилання на зовнішні служби
AppData	Дані додатку
AppStart	Папка для класів, що конфігурують додаток при запуску
bin	Папка для скомпільованих файлів
Classes	Класи додатку
Content	Папка для css та лог-файлів
Controllers	Контролери (MVC)
Fonts	Шрифти
Img	Зображення
Interfaces	Інтерфейси класів
Models	Моделі (MVC)
Scripts	Зовнішні js-бібліотеки
ScriptsCustom	Власний js-код
Translates	Переклади сайту
Views	Види (MVC)
Global.asax	Клас для обробки глобальних подій
Packages.config	Налаштування підключених пакетів
Web.config	Налаштування веб-додатку

3.4 Розробка інтерфейсу програмного забезпечення

Основні інтерфейсні форми, що відображають роботу програми, наведено на рис 3.5-3.10.

Рисунок 3.5 – Панель фільтрів

Категория	Стоимость (За весь период)
Single	231.00 eur

Просторный номер с изысканным дизайном, продуманным до мельчайших деталей. Зеркальный шкаф, столик и тумбочки в викторианском стиле – все это создает незабываемую, легкую и ненавязчивую, атмосферу прошлых веков. Тем не менее, современная комплектация комнаты не даст Вам забыть, что все удобства XXI века к Вашим услугам.

[Заказать](#)

Рисунок 3.6 – Результати пошуку

Suite (3x местные)

406.00 eur



Торжественный, элегантный номер (площадью от 14 до 20 м.кв) является воплощением пышности и, в то же время, утонченности. Мебель в стиле викторианской эпохи, огромная кровать стандарта King, балдахины, изысканные гардины, зеркальный шкаф, изящные абажуры – все это создает неповторимую атмосферу роскоши, но без лишней помпезности.

[Заказать](#)

Рисунок 3.7 – Інформація про номер Подтверждение заказа

[Назад](#)

Проживание

Цена может измениться в зависимости от курса и избранных дополнительных услуг. Refundable - в случае аннуляции за 1 сутки до даты заезда штраф не взимается. В случае аннуляции позднее этого срока или в случае не-шоу взимается 100 процентов от стоимости первой ночи проживания.

Город	Киев
Отель	Royal City Hotel
Категория	Junior Suite
Состав	Взрослых: 2
Стоимость	350.00 eur

Контактные данные

Фамилия Имя Отчество *		
☐ Платежные не совпадают с заказчиком		
Мобильный телефон *		
Email *		
Страна	[Выберите страну] ▼	
Населенный пункт		

* - обязательные для заполнения поля

Рисунок 3.8 – Бронювання номеру та заповнення контактних даних

Авторизация

имя

пароль

☐ Оставаться в системе

Войти

Рисунок 3.9 – Форма авторизації

The screenshot shows a web application interface for configuring the booking module. On the left is a sidebar with a menu containing items like 'Общие параметры модуля', 'Платежные системы', 'Курсы валют', 'Страны', 'Города', and 'Настройки для администратора'. The main area is titled 'Общие параметры модуля бронирования' and includes a 'Сохранить' (Save) button and a 'Параметры клиентов' (Client parameters) link. Under the 'Общие' (General) section, there are input fields for 'Ширина модуля' (Module width) set to '765px' and a checkbox for 'Скрывать заголовок' (Hide header). The 'Период' (Period) section includes checkboxes for 'Показывать' (Show) and 'Редактировать' (Edit), and input fields for 'Минимальное кол-во дней в заказе' (1), 'Начать период через N дней с текущей даты' (7), and 'Длина периода по умолчанию, дней' (7). At the bottom, there is a field for 'Список Email-ов, на которые дублируются письма' (List of email addresses to which emails are duplicated) with an '(Изменить)' (Change) link.

Рисунок 3.10 – Загальні налаштування модуля бронювання

3.5 Висновки за розділом 3

Розроблено вебсайт інтернет-магазину для бронювання місць у об'єктах розміщення на основі інформаційних моделей у вигляді UML-діаграм, запропонованих у попередньому розділі. При цьому розроблено архітектуру програмного забезпечення інтернет-магазину для бронювання місць у об'єктах розміщення, обрано інструментальні засоби розробки. Наведено файлову структуру та інтерфейс розробленого програмного забезпечення інтернет-магазину для бронювання місць у об'єктах розміщення.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення й умови застосування програми

Програмний продукт призначений для автоматизованого бронювання місць у об'єктах розміщення.

Для коректної роботи програму потрібно запускати в операційній системі Windows 7 або більш пізнішій версії операційної системи.

Для експлуатації програмного комплексу необхідні такі програмно-технічні засоби:

- процесор частотою 2 ГГц;
- місце на жорсткому диску – 500 Мб;
- відеокарта з пам'яттю 256 Мб;
- оперативна пам'ять – 1Гб;
- маніпулятор миша.

Для функціонування системи необхідною є наявність сервера, на якому встановлені сервер IIS та SQL Server 2012. Для роботи програми зі сторони клієнта необхідна ОС Microsoft Windows зі встановленим середовищем виконання CLR .NET Framework 2.0.

4.2 Характеристики програми

Дане програмне забезпечення являється веб-додатком, розміщеним на сервері, тобто працює цілодобово, поки сервер запущений та знаходиться в працездатному стані. Час роботи програми залежить від конфігурації комп'ютера, на якому вона виконується.

4.3 Звернення до програми

Для виклику додатку слід у будь-якому браузері необхідно перейти за веб-адресою, де встановлено програмне забезпечення. Для керування роботою додатку використовуються посилання, випадаючі списки, перемикачі та інші активні елементи.

4.4 Вхідні і вихідні дані

Вхідними даними програми є дані інформація про готель, номери, ціновий фонд, посилання на зображення, дані замовника. Вихідними даними є: повідомлення про замовлення та лист на поштову скриньку.

4.5 Повідомлення

Програма може виводити наступні повідомлення:

- повідомлення про некоректні дані фільтрів;
- повідомлення про відсутні обов'язкові значення полів;
- повідомлення про помилку під час бронювання;
- повідомлення про помилку під час відправлення email-повідомлення;
- повідомлення агента контролю якості, які надходять email;
- повідомлення про успішне завершення бронювання;
- повідомлення про успішне завершення оплати.

5 КЕРІВНИЦТВО ОПЕРАТОРА

5.1 Призначення програми

Програмний продукт призначений для автоматизованого бронювання місць у об'єктах розміщення.

Для коректної роботи програму потрібно запускати в операційній системі Windows 7 або більш пізнішій версії операційної системи.

Інтерфейс користувача забезпечує просту і ефективну взаємодію користувача з програмним продуктом.

5.2 Умови виконання програми

Для експлуатації програмного комплексу необхідні такі програмно-технічні засоби:

- процесор частотою 2 ГГц;
- місце на жорсткому диску – 500 Мб;
- відеокарта з пам'яттю 256 Мб;
- оперативна пам'ять – 1Гб;
- маніпулятор миша.

Час роботи програми залежить від конфігурації комп'ютера, на якому вона виконується. Для функціонування системи необхідною є наявність сервера, на якому встановлені сервер IIS та SQL Server 2012. Для роботи програми зі сторони клієнта необхідна ОС Microsoft Windows зі встановленим середовищем виконання CLR .NET Framework 2.0.

5.3 Виконання програми

Для запуску додатку потрібно виконати наступні дії:

- запустити встановлений веб-браузер;

– у адресному рядку набрати адресу сайту, після чого з'явиться головна сторінка додатку.

Для замовлення номеру слід:

- запустити веб-додаток;
- вибрати на панелі фільтрів потрібні критерії номеру;
- переглянути доступні варіанти та натиснути кнопку «Замовити»;
- ввести дані та забронювати номер.

Для редагування налаштувань необхідно:

- авторизуватись в адмін-частині сайту;
- перейти до потрібного розділу та змінити налаштування;
- зберегти налаштування та очистити кеш.

5.5 Повідомлення оператору

Як Веб-додаток може виводити наступні повідомлення:

- повідомлення про некоректні дані фільтрів;
- повідомлення про відсутні обов'язкові значення полів;
- повідомлення про помилку під час бронювання;
- повідомлення про помилку під час відправлення email-повідомлення;
- повідомлення агента контролю якості, які надходять email;
- повідомлення про успішне завершення бронювання;
- повідомлення про успішне завершення оплати.

6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

У дипломній кваліфікаційній роботі магістра розроблено програмне забезпечення у вигляді веб-сайту інтернет-магазину.

Очікується, що впровадження в експлуатацію розробленого програмного забезпечення на підприємстві дозволить скоротити 0,5 ставки одного адміністратора, з заробітною платою 12500 гривень на місяць.

У розробці беруть участь програміст із заробітною платою 13500 грн. в місяць і консультант з заробітною платою 9500 грн. у місяць. На розробку проєкту було витрачено 2 місяці. Розробка програми розпочалася першого вересня і повинна бути виконана до кінця жовтня 2020 року.

6.1 Планування розробки програмного продукту

Весь комплекс розробки програмного продукту можна розділити на етапи [44]. Для кожного етапу вказуються трудомісткість, кількість виконавців і тривалість робіт. Інформація з характеристики робіт по розробці програмного забезпечення зведена у таблицю 6.1.

Таблиця 6.1 – Характеристика робіт по розробці програми

Найменування робіт	Трудомісткість		Виконавці	Тривалість, днів
	люд-дн	%		
Аналіз ПЗ	7	10,4	Програміст Консультант	4 3
Визначення вимог до ПП	9	13,4	Програміст Консультант	4 5
Проектування структури програми	30	44,8	Програміст Консультант	16 14
Розробка схеми функціонування програми	3	4,5	Програміст	3
Створення програмного коду	10	14,9	Програміст	10
Тестування і налагодження програмного продукту	5	7,5	Програміст	5
Складання програмної документації	3	4,5	Програміст	3
Разом	67	100	Програміст Консультант	45 22

6.2 Визначення витрат на розробку програми

Для визначення витрат на розробку програми складають калькуляцію кошторисної вартості робіт, що включає статті:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок (ЄСВ);
- матеріали та комплектуючі вироби;
- витрати на спеціальне обладнання (на оплату машинного часу ЕОМ для написання і налагодження програмних засобів);
- накладні витрати.

Витрати за статтею «Основна заробітна плата» складаються з планового фонду зарплати всіх категорій працівників, зайнятих в розробці програми. Розрахунок зарплати на підставі даних про трудомісткість, представлений в таблиці 6.2.

Таблиця 6.2 – Основна заробітна плата

Посада виконавця	Чисельність, чол	місячний оклад	Кількість місяців роботи	Сума ЗП, грн
Програміст	1	13500	2	27000
Консультант	1	9500	1	9500
Разом	2			36500

Додаткову заробітну плату приймають рівною 10% від основної зарплати. Вона розраховується за формулою (6.1):

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} \cdot 10\% . \quad (6.1)$$

Підставивши величину основної заробітної плати в дану формулу отримаємо: $ЗП_{\text{дод}} = 36500 \cdot 0,1 = 3650$ грн.

Відрахування на єдиний соціальний внесок (ЄСВ) визначають у відсотковому відношенні від сум основної та додаткової зарплати з урахуванням премій і доплат. Відрахування вираховуються за формулою (6.2):

$$ОТ = (ЗП_{\text{осн}} + ЗП_{\text{дод}}) \cdot 0,22 . \quad (6.2)$$

$$ОТ = (36500 + 3650) \cdot 0,22 = 8833 \text{ грн.}$$

Використовується три найменування матеріалів: папір – 200 грн., картридж – 800 грн, флеш-носій – 500 грн.. Витрати на матеріали розраховують за формулою (6.3):

$$M = \sum_{i=1}^n (C_i \cdot N_i \cdot (1 + K_{m.з.}) - C_{io} \cdot N_{io}), \quad (6.3)$$

де M – витрати на матеріали, покупні напівфабрикати і комплектуючі вироби, грн .;

$K_{m.з.}$ – коефіцієнт, що враховує транспортно-заготівельні витрати;

C_i – ціна і-го найменування матеріалу, напівфабрикату і комплектуючого, грн .;

N_i – потреба в і-му матеріалі, напівфабрикаті і комплектуючому;

C_{io} – ціна зворотних відходів і-го найменування матеріалу, грн;

N_{io} – кількість зворотних відходів і-го найменування;

n – кількість найменувань матеріалів, напівфабрикатів і комплектуючих.

$$C_{io} = 0; N_{io} = 0; K_{m.з.} = 0,05;$$

$$M = (1 + 0,05) \cdot (200 + 800 + 500) = 1575 \text{ грн.}$$

Таким чином, разом витрати на матеріали та комплектуючі вироби становлять 1575 гривень.

У статтю «Витрати на спеціальне обладнання» входять витрати на придбання, транспортування, монтаж і налагодження нестандартного обладнання. Практично, в даному випадку, в цій статті враховуються витрати на оплату машинного часу ЕОМ для написання і налагодження програмних засобів. Для цього необхідно скласти кошторис «витрат на утримання і експлуатацію устаткування», виходячи з якого визначиться вартість одного машино-години роботи ПК, після множення якої на машинний час пішло на написання і налагодження програми отримаємо витрати на оплату машинного часу.

Амортизаційні відрахування визначають за формулою (6.4):

$$A = BB \cdot \frac{H_a}{100}, \quad (6.4)$$

де BB – балансова вартість обчислювальної техніки, грн;

H_a – норма амортизаційних відрахувань на повне оновлення обчислювальної техніки,%. Приймаємо, що техніка повинна працювати 4 роки, тому $H_a=25\%$.

Балансова вартість обчислювальної техніки становить 29000 грн. Тому амортизаційні відрахування становлять:

$$A = 38000 \cdot 0,25 = 7250 \text{ грн}$$

Статтю "Експлуатація обладнання" розраховують підсумовуванням витрат на силову електроенергію і допоміжні матеріали. Витрати на електроенергію визначають за формулою (6.5):

$$C_{\varepsilon} = N_n \cdot \Phi_{\varepsilon\phi} \cdot K_{\varepsilon\phi} \cdot K_{\varepsilon m} \cdot C_{\varepsilon}, \quad (6.5)$$

де N_n – номінальна потужність ЕОМ, кВт;

$\Phi_{\varepsilon\phi}$ – річний ефективний фонд часу роботи ЕОМ, машино–год;

$K_{\varepsilon\phi}$ – середній коефіцієнт завантаження за часом;

$K_{\varepsilon m}$ – коефіцієнт завантаження по потужності;

C_{ε} – ціна одного кВт · год електроенергії, грн ./ (кВт · год).

Номінальна Потужність ЕОМ – 0,2 кВт. Річний ефективний фонд часу роботи ЕОМ ставити 1800 годин. Середні коефіцієнти завантаження за часом та за потужністю рівні відповідно 0,9 та 0,6. Ціна однієї кіловат–години електроенергії ставить 2,11 грн. Отримуємо:

$$C_e = 0,2 \cdot 1800 \cdot 0,9 \cdot 0,6 \cdot 2,11 = 410,18 \text{ грн.}$$

Приймаємо, що $C_e = 410$ грн.

Витрати за статтею "Заробітна плата обслуговуючих робітників та відрахування на соціальне страхування в інші фонди" визначають за формулою (6.6):

$$ЗП_{обсл} = \Phi ЗП_z \cdot (1 + K_{відрах.}) \cdot \frac{t_{обсл}}{\Phi_{эф.обсл}}, \quad (6.6)$$

де $\Phi ЗП_z$ – річний фонд заробітної плати (основної та додаткової) обслуговуючих робітників, грн.;

$K_{відрах.}$ – коефіцієнт, що враховує відрахування на соціальне страхування і в інші фонди;

$t_{обсл}$ – час протягом року, необхідне технічне обслуговування ЕОМ, годин / рік. Приймаємо $t_{обсл} = 36$ год.;

$\Phi_{эф.обсл}$ – річний ефективний фонд часу обслуговуючого персоналу, годин / рік. Приймаємо, що річний ефективний фонд часу обслуговуючого персоналу становить 1800 годин

Заробітна плата обслуговуючого персоналу = 10400 грн/місяць.

$$\Phi ЗП_z = 10400 \cdot 12 = 124800 \text{ грн.}$$

$$ЗП_{обсл} = 124800 \cdot (1 + 0,22) \cdot 36 / 1800 = 3045 \text{ грн.}$$

Суму витрат за статтею "Поточний ремонт обладнання" обчислимо як 3% від балансової вартості обладнання, оскільки неможливо заздалегідь визначити, скільки коштів необхідно витратити на придбання запасних частин наявного обладнання. Балансова вартість обчислювальної техніки становить 29000 грн. Тому сума витрат за статтею "Поточний ремонт обладнання" становить: $29000 \cdot 0,03 = 870$ грн.

Інших витрат за проектом не передбачено, тому сума витрат за цією статтею дорівнює нуль гривень.

Кошторис витрат на утримання і експлуатацію устаткування наведений в таблиці 6.3.

Таблиця 6.3 – Кошторис витрат на утримання і експлуатацію устаткування

Найменування статей витрат	Сума, грн.
Амортизація обладнання	7250
Заробітна плата основна і додаткова обслуговуючих робітників з відрахуваннями на соціальні заходи	3045
Експлуатація обладнання (крім витрат на поточний ремонт)	410
Поточний ремонт обладнання	870
Інші витрати	0
Разом	11575

Витрати на оплату машинного часу ЕОМ для написання і налагодження програмних засобів визначаються за формулою (6.7):

$$C_{mo} = P_{екс} \cdot t_{mo}, \quad (6.7)$$

де C_{mo} – витрати на оплату машинного часу, грн;

$P_{екс}$ – експлуатаційні витрати на одну годину машинного часу цієї цифрової ЕОМ, грн. / машино–год.;

t_{mo} – машинний час цифрової ЕОМ для написання і налагодження даного програмного продукту, машино–год.

$$P_{екс} = 11575/1800 = 6,43 \text{ грн. / машино–год.}$$

Для написання і налагодження даного програмного продукту ЕОМ експлуатується протягом 45 днів в одну зміну по 8 годин, що складає в сумі 360 годин. Таким чином отримуємо суму за статтею «Витрати на спеціальне обладнання»:

$$C_{mo} = 6,43 \cdot 360 = 2315 \text{ грн.}$$

До накладних витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати і т.д.), утримання та експлуатацію будівель. Накладні витрати включаються до вартості розробки програми непрямым шляхом – у відсотках до основної заробітної плати розробників. Для обчислення цієї статті видатків у дипломному проєкті приймемо, що накладні видатки становлять 25% від основної заробітної плати розробників, що складає 36500 грн. Тому сума накладних видатків становить: $36500 \cdot 0,25 = 9125$ грн.

Калькуляція кошторисної вартості робіт з розробки програми наведена в таблиці 6.4.

Таблиця 6.4 – Калькуляція кошторисної вартості робіт з розробки програми

Найменування статей витрат	Сума, грн.
Основна заробітна плата	36500
Додаткова заробітна плата	3650
ЄСВ	8833
Матеріали і комплектуючі	1575
Витрати на спеціальне обладнання	2315
Накладні витрати	9125
Разом	61998

6.3 Розрахунок економічної ефективності програмного продукту

Річну економію від впровадження програми, що дозволяє знизити витрати машинного часу ЕОМ для рішення певного завдання в порівнянні із програмою, що застосовувалася раніше, визначають за формулою:

$$E = (T_{M_1} - T_{M_2}) \cdot B_{\text{екс.КТЗ}} + E_{\text{н}}^a \cdot \frac{\Phi_{\text{б.КТЗ}} \cdot (T_{M_1} - T_{M_2})}{\Phi_{\text{еф.КТЗ}}} - \frac{S_{\text{рп}}}{T_{\text{с}}} + E_{\text{зп}}, \quad (6.8)$$

де T_{M_1}, T_{M_2} – машинний час, необхідний для розв’язання поставлених завдань, відповідно в старому та у новому варіантах, машино–год./рік;

$B_{\text{екс}}$ – експлуатаційні витрати, що доводяться на 1 год машинного часу ЕОМ, грн/ машино–год.;

$E_{\text{н}}^a$ – нормативний коефіцієнт ефективності капітальних вкладень у засоби автоматизації;

$\Phi_{\text{б.КТЗ}}$ – балансова вартість ЕОМ (КТЗ), грн.;

$\Phi_{\text{еф.КТЗ}}$ – річний ефективний фонд часу роботи ЕОМ, машино–год.;

$S_{\text{рп}}$ – сумарні витрати на розробку програми, грн.;

$T_{\text{с}}$ – термін служби впроваджуваної програми до її морального зношування, років. Приймаємо цей термін 5 років;

$E_{\text{зп}}$ – економія на заробітній платі.

До впровадження даного ПЗ дані функції виконувались співробітником, що відповідає за обробку рукописних символів, 1800 годин на рік, а після 900 годин на рік (економія складає 0,5 ставки одного адміністратора, з заробітною платою 12500 гривень на місяць).

$$E = (1800 - 900) \cdot 6,43 + 0,2 \cdot (29000 \cdot (1800 - 900)) / 1800 - 61998/5 + \\ + 12500 \cdot 0,5(1 + 0,22) \cdot 12 = 87787 \text{ грн.}$$

Коефіцієнт ефективності капітальних вкладень E и строк їхньої окупності $T_{\text{ок}}$ розраховуються за формулами (6.9) і (6.10):

$$Ef = \frac{E}{S_{\text{рп}}}, \quad (6.9)$$

$$T_{\text{ок}} = \frac{S_{\text{рп}}}{E}, \quad (6.10)$$

$$Ef = 87787 / 61998 = 1,42,$$

$$T_{\text{ок}} = 61998 / 87787 = 0,71 \text{ року.}$$

Зведемо отримані техніко–економічні показники в таблицю 6.5.

Таблиця 6.5 – Техніко–економічні показники

Показники	Сума	Одиниця виміру
Витрати на розробку програми	61998	грн.
Річна економія	87787	грн.
Строк окупності програми	0,71	рік

Таким чином, впровадження в експлуатацію розробленого програмного забезпечення на виробничому підприємстві дозволить скоротити 0,5 ставки одного адміністратора, з заробітною платою 12500 гривень на місяць. Економічні розрахунки показали, що строк окупності програми 0,71 року, що значно менше часу її нормативного строку окупності (5 років). Крім того, річна економія при використанні програми становить 87787 грн. Виходячи з виконаних розрахунків, можна зробити висновок, що розробка даного програмного продукту є економічно доцільною.

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Аналіз потенційних небезпек

В дипломному проекті проводиться дослідження та програмна реалізація інформаційних моделей розробки вебсайтів. При роботі на розробників постійно або періодично діють наступні небезпечні і шкідливі чинники.

Невідповідність ергономічних характеристик устаткування нормованим величинам внаслідок нераціональної організації робочого місця.

Негативний вплив випромінювань відео дисплейних терміналів на органи зору людини має основною причиною перенапруження м'язового апарату ока, що пов'язане з недостатньою контрастністю та мерехтінням зображення, відблисками та відбитим світлом від моніторів. Крім того, втома очей настає при частій зміні огляду більш освітленого монітора та менш освітленої клавіатури, що призводить до «астенопії» (відчуття дискомфорту та біль в очах) [45]-[47].

Негативний вплив при роботі з відео дисплейними терміналами на функціональний стан людини який обумовлений електростатичними полями, які створюються моніторами з електронно-променевою трубкою, що може призвести до зниження імунітету, захворювань периферичної нервової системи або бронхо-легеневих та шкіряних захворювань.

Можливість ураження електрострумом внаслідок небезпечного рівня напруги в електричній мережі, тому у приміщенні можуть виникнути аварійні ситуації: замикання, загоряння проводки, ураження електрострумом, що може призвести до електротравм або летальних наслідків.

Негативний вплив статичної електрики.

Невідповідність нормам параметрів мікроклімату.

Незадовільні умови освітлення робочого місця, в наслідок не рівномірного розподілу освітлювальних приладів.

Небезпеки, пов'язані з підвищеним рівнем шуму, який створюється устаткуванням та працюючими у приміщенні, що призводить до погіршення слуху.

Можливість загорянь внаслідок порушень правил пожежної безпеки.

Неправильні дії персоналу під час НС внаслідок відсутності управління та паніки, що може призвести до травм та летальних наслідків.

7.2 Заходи по забезпеченню безпеки

Забезпечення безпеки праці, організації оптимальних умов праці, а також захисту навколишнього середовища досягається шляхом дотримання норм і проведення заходів, передбачених законодавчими актами України.

У НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [19] передбачені гігієнічні вимоги до організації устаткування робочих місць ВДТ ЕОМ і ПЭВМ. Конструкція робочого стола відповідає сучасним вимогам ергономіки й забезпечує оптимальне розміщення на робочій поверхні використовуваного устаткування (дисплея, клавіатури, принтера) і документів. Висота робочої поверхні робочого стола із ВДТ регулюється в межах 680...800 мм, а ширина й глибина - забезпечує можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600...1400 мм, глибина - 800..1000 мм). Приймаємо робочу поверхню з розмірами 1400x800. Щоб уникнути відблисків на екрані вікон робоче місце розміщене торцем до вікна не ближче чим на 1 м. Площа дозволяє нам розмістити додаткові меблі для зберігання документів, полки, стелажі, тумби й т.п.

Велике значення надається ергономічним та естетичним вимогам. Цього домоглися за допомогою колірного рішення приміщення й устаткування з урахуванням вимог ДСТУ 8604:2015 «Дизайн і ергономіка.

Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги» [20] - у приміщеннях стіни пофарбовані фарбами пастельних тонів.

Під час організації робочого простору необхідно враховувати індивідуальні антропометричні параметри користувача з відповідними допусками на можливі зміни робочих поз та потребу у переміщеннях.

Заходи мінімізації негативного впливу випромінювань від відео дисплейних терміналів на органи зору [6].

Передбачено такі внутрішньозмінні режими праці та відпочинку при роботі з ЕОМ при 8-годинній денній робочій зміні в залежності від характеру праці [6]:

- для розробників програм із застосуванням ЕОМ слід призначати регламентовану перерву для відпочинку тривалістю 15 хвилин через кожну годину роботи за ВДТ;

- для операторів із застосування ЕОМ слід призначати регламентовані перерви для відпочинку тривалістю 15 хвилин через кожні дві години;

- для операторів комп'ютерного набору слід призначати регламентовані перерви для відпочинку тривалістю 10 хвилин після кожною години роботи за ВДТ.

У всіх випадках, коли виробничі обставини не дозволяють застосувати регламентовані перерви, тривалість безперервної роботи з ВДТ не повинна перевищувати 4 години [6].

При 12-годинній робочій зміні регламентовані перерви повинні встановлюватися в перші 8 годин роботи аналогічно перервам при 8-годинній робочій зміні, а протягом останніх 4-х годин роботи, незалежно від характеру трудової діяльності, через кожну годину тривалістю 15 хвилин. [6]

З метою зменшення негативного впливу монотонності застосовуються чергування операцій усвідомленого тексту і числових даних (зміна змісту роботи) [6].

Для зниження нервово-емоційного напруження, втомлення зорового аналізатору, поліпшення мозкового кровообігу, подолання несприятливих

наслідків гіподинамії, запобігання втомі доцільні деякі перерви використовувати для виконання комплексу вправ.

В окремих випадках - при хронічних скаргах працюючих з ВДТ ЕОМ і ПЕОМ на зорове втомлення, незважаючи на дотримання санітарно-гігієнічних вимог до режимів праці і відпочинку, а також застосування засобів локального захисту очей - допускаються індивідуальних підхід до обмеження часу робіт з ВДТ, зміни характеру праці, чергування з іншими видами діяльності, не пов'язаними з ВДТ [19].

Активний відпочинок має полягати у виконанні комплексу гімнастичних вправ, спрямованих на зняття нервового напруження, м'язове розслаблення, відновлення функцій фізіологічних систем, що порушуються протягом трудового процесу, зняття втоми очей, поліпшення мозкового кровообігу і працездатності.

Згідно ПУЕ приміщення виробництва відноситься до категорії «Приміщення без підвищеної небезпеки» [21].

Апаратура, що перебуває під небезпечною для життя напругою, припускає забезпечення електробезпечності. Всі металеві частини заземлені на третій провід спеціальних триконтактних розеток. Струмопровідні кабелі розташовані під дерев'яною підлогою, зверху покритою лінолеумом, або у вигляді схованої проводки в стінах, таким чином, контакт із кабелем виключається. Над кожною розеткою попереджуючий надпис.

Електрична ізоляція забезпечується конструкціями пристроїв і способами їхнього підключення. У процесі експлуатації періодично проводиться перевірка якості ізоляції, шляхом виміру її опору. Відповідно до ПУЕ в електроустановках до 1000В мінімальне значення опору ізоляції силових і освітлювальних електропроводок на ділянці між суміжними запобіжниками або за останніми запобіжниками між будь-яким проведенням і землею становить не менше 0,5МОм [6].

Відповідно до ПУЕ захисне заземлення не перевищує 4 Ом у всіх електроустановках змінного струму напругою 220В і вище й постійного

струму 440В і вище. У приміщеннях з підвищеною небезпекою заземлення використовується в електроустановках змінного струму напругою 42В та постійного струму 110В. Захисному заземленню підлягають корпуси трансформаторів, оболонки кабелів, металеві огороження електроустановок [22].

Вирівнювання потенціалів досягаються шляхом устрою контурних заземлювачів. У приміщеннях вирівнювання потенціалів здійснюється з'єднанням металевих стоек ЕОМ, корпусів апаратів світильників і іншого устаткування з усіма доступними для дотику металевими конструкціями будинку, спорудження, а також з контуром шиною захисного заземлення.

Електричний поділ мережі припускає розділ мережі на окремі електрично непов'язані між собою ділянки за допомогою розділових трансформаторів. Підвищення електробезпеки відбувається за рахунок зменшення ємкості мережі й підвищення опору ізоляції фаз мережі щодо землі.

Подвійна ізоляція забезпечує застосування крім основної, робочої ізоляції струмоведучих частин ще одного шару додаткової ізоляції, що ізолює людину від металевих неструмоведучих частин, які можуть випадково виявитися під напругою. При застосуванні електроустаткування з подвійною ізоляцією не потрібно ні заземлення, ні занулення їхніх корпусів [21].

Накопичена статична електрика, зокрема, на екрані монітора притягує пил, бруд та інші частини присутні в повітрі. Причому електризується не тільки екран, а і повітря на робочому місці, а також одяг користувача, особливо якщо він з синтетичного та шерстяного матеріалу [6].

В якості заходів захисту запропоновано:

- кілька разів на протязі робочого дня мити руки і обличчя водою, а після закінчення роботи вимити руки й лице з милом;
- щоденно протирати екран монітора, клавіатуру, пристрій «миша», а якщо є приєднаний фільтр, то і його антистатичною серветкою;
- щоденно в приміщенні з ПК проводити вологе прибирання;

- установити нейтралізатори статичної електрики;
- підтримувати у приміщенні вологість повітря зазначену в нормативних документах;
- користувачу ПК бажано носити одяг з природних (льняних) волокон, в т.ч. і шкарпетки [6].

7.3 Заходи з виробничої санітарії та гігієни праці

Оптимальні норми температури повітря в робочій зоні виробничих приміщень у холодний і перехідний періоди року при II категорії робіт згідно ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень [23] – 22...24оС, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек. У теплий період року температура – 23...25оС, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек.

Для забезпечення вищевказаних параметрів передбачене застосування вентиляції - природної (аерації й провітрювання), механічної (загальобмінної, місцевої приточної і витяжної), кондиціювання повітря.

У приміщенні забезпечений приплив свіжого повітря, кількість якого визначається техніко-економічним розрахунком і вибором схеми системи вентиляції. Мінімальна витрата повітря визначається з розрахунку 50...60 м3/год на одну людину, але не менш дворазового повітрообміну в годину.

При невеликому забрудненні зовнішнього повітря кондиціювання приміщень здійснюється із перемінними витратами зовнішнього повітря й рециркуляційного.

Всі норми по освітленості беруться згідно ДБН В.2.5-28:2018. Природне і штучне освітлення [24]. Штучне освітлення в приміщеннях здійснюється системою загального рівномірного освітлення. При цьому штучне освітлення застосовують не тільки в темний, але й у світлий час доби.

Найбільш прийнятними для приміщень є люмінесцентні лампи ЛБ (білого світла) потужністю 20, 40 або 80 Вт, які встановлюються в

світильники типу ЛСП. Допускається застосування ламп накаливання у світильниках місцевого освітлення.

Загальне освітлення варто виконувати у виді суцільних чи переривчастих ліній світильників, розташованих збоку від робочих місць.

Розрахуємо систему загального рівномірного штучного освітлення в обчислювальному центрі. Приймаємо розміри приміщення: довжина $A = 14$ м, ширина $B = 7$ м, висота $H = 5$ м. Висота робочої поверхні складає $h = 0,7$ м.

Коефіцієнти відбиття поверхонь приміщення: колір стелі, стін - сірий, незначне пиловиділення, $\rho_{\text{с}}=50\%$; $\rho_{\text{ст}}=30\%$; $\rho_{\text{п}}=10\%$,

Згідно з ДБН В.2.5-28:2018 «Природне і штучне освітлення» рівень нормованого загального освітлення в приміщенні нормуються як $E_{\text{н}} = 400$ лк.

Обираємо світильник типу ЛПП, призначений для приміщень висотою до 6 м. $IP=65$ – ступінь захисту світильника, $[L/h] = 1,4$ – числове значення коефіцієнта світильника.

Коефіцієнт запасу K_z враховує зниження рівня освітленості з часом в результаті забруднення та старіння ламп, світильників і поверхонь приміщення, приймається в залежності від виробничих умов. Приймаємо $K_z = 1,6$.

Коефіцієнт нерівномірності (мінімального) освітлення z (відношення середньої освітленості до мінімальної освітленості), для люмінесцентних ламп низького тиску, як правило дорівнює $z = 1,1$.

Згідно з ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» [26] зниження шуму, створюваного на робочих місцях внутрішніми джерелами, а також шуму проникаючого ззовні, є дуже важливим завданням. Зниження шуму в джерелі виникнення забезпечується застосуванням пружних прокладок між основою машини, приладу й опорною поверхнею. Як прокладки використовуються гума, повсть, пробка, різної конструкції амортизатори. Під настільні шумливі апарати можна підкладати

м'які килимки із синтетичних матеріалів, а під ніжки столів, на яких вони встановлені, - прокладки з м'якої гуми, повсті, товщиною 6 - 8 мм.

7.4 Заходи безпеки у надзвичайних ситуаціях

7.4.1 Заходи з пожежної безпеки

Заходи з протипожежної безпеки розроблені для виробничого приміщення обладнаного ПК з ВДТ розроблені згідно вимог НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні».

Виходячи з аналізу речовин та матеріалів, які використовуються при роботі у приміщенні, обладнаному ПК з ВДТ:

- згідно ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» [27] у приміщенні, обладнаному ПК з ВДТ можлива пожежа класу – А та Е;

- відповідно до вимог ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою» [28], воно належить до категорії «Д» з пожежної небезпеки – простір у приміщенні, у якому перебувають тверді горючі речовини та матеріали.

Оскільки приміщення належить до категорії «Д» з пожежної небезпеки, тому відповідно до вимог ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» [27] воно має II ступінь вогнестійкості.

У разі виникнення пожежі у приміщенні обладнаному ПК з ВДТ для евакуації персоналу відповідно до вимог ДБН В.1.1-7:2016 передбачені виходи, по обидві сторони приміщення, з одного боку вікно (на пожежні сходи), а з іншого – вхідні двері. Відстань від найбільш віддаленого робочого місця до найближчого евакуаційного виходу не обмежується.

Обладнання, силові та освітленні мережі приміщення, обладнаного ПК з ВДТ відповідають вимогам пожежної безпеки, оскільки виконані

відповідно до вимог НПАОП 40.1-1.32-01 «Правила будови електроустановок. Електрообладнання спеціальних установок» [29], та мають ступінь захисту ізоляції обладнання IP44 яка відповідає класу пожежанебезпечної зони П-Па до якої належить приміщення.

Для виявлення початкової стадії загоряння, швидкого й точного оповіщення служби пожежної охорони про час і місце виникнення пожежі встановлені системи автоматичної пожежної сигналізації (АПС). Вони самостійно пускають в хід установки пожежогасіння, коли пожежа не досягла великих розмірів. Системи АПС складаються з пожежних сповіщувачів, ліній зв'язку й прийомних пультів (станцій). Згідно вимог ДБН В.2.5-56:2014 «Системи протипожежного захисту» [29], в приміщенні обладнаному ПК з ВДТ встановлена система пожежної й охоронної сигналізації «Фотон-ОФІС», яка забезпечує виявлення теплових і димових ознак пожежі і місця виникнення пожежі з точністю до місця розміщення датчика.

Оскільки приміщення має площу 98 м², передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 5 штук (з розрахунку один вогнегасник с величиною заряду вогнегасної речовини 3 кг. і більше, на 20 м² площі приміщення). Відстань між вогнегасниками та місцями можливих загорянь не перевищує 10 м.

7.4.2 Заходи з цивільного захисту

Оцінка надійності системи матеріально-технічного постачання і виробничих зв'язків.

У своїй діяльності підприємство використовує різноманітні матеріально-технічні ресурси (сировину, матеріали, паливо, енергію, комплектуючі вироби тощо). Вони в процесі виробництва перетворюються на продукцію (послуги) і підлягають постійному поповненню. Для цього організується матеріально-технічне забезпечення, яке включає: визначення

потреби в матеріально-технічних ресурсах, пошук і купівлю ресурсів, організацію доставки, зберігання й видачі окремим споживачам на підприємстві. Постачання матеріально-технічних ресурсів має бути своєчасним, комплектним і з мінімальними витратами. Виконує цю роботу відділ матеріально-технічного постачання [6].

Матеріально-технічне забезпечення — це система принципів, форм, методів, важелів та структур, які спрямовані на виробництво і постачання технічних засобів, запасних частин, паливно-мастильних матеріалів, інших енергоносіїв, пестицидів, техніко-технологічне обслуговування та надання виробничих послуг.

Матеріально-технічні ресурси являють собою сукупність матеріально-речових засобів і предметів праці, що постачаються промисловістю, за допомогою яких виробляється продукція або надаються відповідні послуги.

Система матеріально-технічного забезпечення — це сукупність підприємств-виробників матеріально-технічних ресурсів, торгово-посередницьких організацій, сервісних підрозділів, формувань з машиновикористання та державних органів, що регулюють відносини у сфері техніко-технологічного забезпечення. Діяльність цієї системи має бути спрямована на відтворення і розвиток матеріально-технічної бази, комплексну механізацію і автоматизацію технологічних процесів, відновлення і збереження родючості ґрунту в сільському господарстві на основі вітчизняного машинобудування та виробництва матеріальних ресурсів.

Виробничими структурами у системі матеріально-технічного забезпечення є заводи сільськогосподарського, продовольчого і торгового машинобудування, хімічні підприємства, торговельно-посередницькі організації, підприємства з ремонту і технічного обслуговування машин і обладнання, машинно-технологічні формування з надання виробничих послуг, центри випробування техніки, конструкторські та науково-дослідні установи. У межах свого регіону і сфери діяльності вони вивчають попит,

купівельну спроможність споживачів, умови реалізації товарів, терміни поставок і на основі зібраної інформації налагоджують виробництво та реалізацію продукції.

Виробництво та реалізація матеріально-технічних засобів, технічне обслуговування і надання виробничих послуг набувають характеру цілісного технологічного процесу. Управління та координацію цим процесом здійснюють органи, діяльність яких визначається особливостями функціонування агропромислового комплексу, загальнодержавними і регіональними програмами соціально-економічного розвитку, що розроблені Кабінетом Міністрів України.

На ці органи покладаються такі функції:

- формування і проведення єдиної технічної політики виробництва та постачання матеріальних ресурсів, сервісного забезпечення підприємств; закупівля критичних видів ресурсів;
- координація діяльності державних і приватних структур для поліпшення матеріально-технічного забезпечення;
- державна підтримка пріоритетних напрямів технічного і технологічного переоснащення сільськогосподарського, продовольчого та торгового машинобудування і виробництва продукції, придбання ліцензій та освоєння наукоємних технологій;
- реалізація державних і регіональних програм розвитку сільськогосподарського, продовольчого і торгового машинобудування, технічного сервісу і виробничого обслуговування;
- координація замовлень на виконання та освоєння науково-дослідних, дослідно-конструкторських і проектно-технологічних розробок для підвищення ефективності матеріально-технічного та сервісного забезпечення;
- формування єдиного інформаційного простору та координація маркетингу в системі матеріально-технічного забезпечення;

- проведення протекціоністської політики щодо вітчизняного товаровиробника матеріально-технічних ресурсів [30].

ВИСНОВКИ

У ході виконання дипломної кваліфікаційної роботи магістра було розв'язано актуальне завдання дослідження та програмної реалізації інформаційних моделей розробки вебсайтів. Досліджено методи і засоби створення веб-сайтів інтернет-магазинів. Показано, що програмну реалізацію інтернет-магазинів можна здійснити як з використанням спеціалізованого програмного забезпечення, так і без нього. Проаналізовано спеціальні платформи-конструктори для створення сайтів та інтернет-магазинів, які отримали широке розповсюдження за останні роки. В результаті проведеного аналізу виявлено, що суттєвими недоліками використання готових шаблонів при створенні інтернет-магазинів на основі платформ-конструкторів є такі: складність автоматично згенерованого коду і складність доопрацювання, а також прив'язка до певної платформи. Наявність зазначених недоліків обумовлює неможливість використання платформ-конструкторів при розробці вузькоспеціалізованих веб-сайтів інтернет-магазинів, пов'язаних з продажем тарифних планів від імені великої кількості різних надавачів послуг, зокрема у готельному бізнесі. Показано, що створення інтернет-магазинів без застосування платформ-конструкторів пов'язано з високою вартістю розробки, довготривалістю, обумовленою необхідністю створення сайту «з нуля», а також складністю модифікації сайту у майбутньому при зміні кон'юнктури ринку або вимог користувачів.

Наукова новизна роботи полягає у тому, що розроблено модель програмного забезпечення інтернет-магазину для бронювання місць у об'єктах розміщення, що представляє собою набір діаграм у нотації UML, які дозволяють розробнику створювати програмне забезпечення відповідно до вимог замовника та спрощують розробку інтернет-магазинів без необхідності використання спеціалізованих платформ-конструкторів та забезпечують можливість модифікації розроблених сайтів у майбутньому при зміні вимог користувачів.

Практичне значення отриманих результатів полягає в тому, що розроблено веб-сайт інтернет-магазину для бронювання місць у об'єктах розміщення на основі запропонованих інформаційних моделей у вигляді UML-діаграм. При цьому поставлено та проаналізовано вимоги до системи, розроблено архітектуру програмного забезпечення, обрано інструментальні засоби розробки, розроблено інтерфейс веб-сайту.

Виконано тестування розробленого програмного забезпечення. Результати тестування підтвердили, що програма може використовуватися за призначенням.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Виды и функции сайтов [Электронный ресурс]. – Режим доступа: <https://works.doklad.ru/view/g28mdR1Udk8.html>.
2. Сущность интернет магазина [Электронный ресурс]. – Режим доступа: https://knowledge.allbest.ru/marketing/2c0b65625a2bd78a4d53b98521316d27_0.html.
3. Какой движок выбрать для интернет магазина [Электронный ресурс]. – Режим доступа: <https://recommerce.com.ua/blog/pomosh/kakoi-dvizhok-vybrat-dlya-internet-magazina>.
4. CMS для интернет магазина [Электронный ресурс]. – Режим доступа: <https://wezom.com.ua/blog/luchshaja-cms-dlja-internet-magazina>.
5. Рейтинг «CMS для интернет магазина» [Электронный ресурс]. – Режим доступа: <https://uguide.ru/rejting-cms-dlja-internet-magazina-vybiraem-luchshij-dvizhok-dlja-onlajn-torgovli>.
6. Охрана труда [Электронный ресурс]. – Режим доступа: <https://lektsii.org/11-86333.html>
7. Топ 8 лучших сайтов для бронирования отелей [Электронный ресурс]. – Режим доступа: <https://epicris.ru/top-8-luchshix-sajtov-dlya-bronirovaniya-otelej-rejting-onlajn-servisov>.
8. 8 лучших сервисов для бронирования отелей [Электронный ресурс]. – Режим доступа: <https://bronirovanie-otelej-onlajn.ru/>.
9. Сайты покупки билетов [Электронный ресурс]. – Режим доступа: <https://2btraveler.ru/saity-pokupki-biletov/>.
10. Blogpascher [Electronic resources]. – Access mode: <https://ru.blogpascher.com>.
11. Системы онлайн бронирования отелей [Электронный ресурс]. – Режим доступа: <http://tursite.org/modules/sistemy-bronirovaniya/sistemy-on-line-bronirovaniya-otelej>.
12. Онлайн бронирование [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Онлайн-бронирование>.

13. Сайт [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Сайт>.
14. Веб-страница [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Веб-страница>.
15. Сайт [Электронный ресурс]. – Режим доступа: <https://hostiq.ua/wiki/site/>.
16. Сайт что это такое и какие они бывают [Электронный ресурс]. – Режим доступа: <https://ktonanovenkogo.ru/voprosy-i-otvety/sajt-cto-eto-takoe.html>
17. Интернет-сайт [Электронный ресурс]. – Режим доступа: <https://www.seonews.ru/glossary/internet-site/>.
18. Что такое сайт [Электронный ресурс]. – Режим доступа: <http://moolkin.ru/cto-takoe-sajt-internet-sajt-vidy-sajtov/>.
19. Сайт что же это такое [Электронный ресурс]. – Режим доступа: <http://moolkin.ru/cto-takoe-sajt-internet-sajt-vidy-sajtov/>.
20. Типы сайтов [Электронный ресурс]. – Режим доступа: <https://avada-media.ua/tipy-sajtov/>.
21. Скотт Б. Проектирование веб-интерфейсов : Пер. з англ. [Текст] / Б. Скотт, Т. Нейл. – СПб. : Символ-Плюс, 2010. – 352с. : ил.
22. Гото К. Веб-редизайн [Текст] / Келли Гото, Эмили Котлер. – 2-е изд. – СПб: Символ-Плюс, 2006. – 416 с.
23. Clarke A. Hardboiled Web Design [Текст] / Andy Clarke. – Smashing Magazine, 2015. — 441 p.
24. Что такое веб-сайт? [Электронный ресурс]. – Режим доступа : <http://ru.smart-ip.net/what-is-a-website>.
25. Что такое веб-сайт, зачем он нужен и из чего состоит [Электронный ресурс]. – Режим доступа : <http://www.diatekc.net/blog/about-site.html>.
26. Что такое интернет-магазин? (обзорная статья) [Электронный ресурс]. – Режим доступа : <http://www.uzhgorod.ws/articles-trade-eshops.htm>.

27. Создание сайта интернет магазина [Электронный ресурс]. – Режим доступа : <http://webstudio2u.net/ru/webdesign/130-online-shop.html>.
28. Создание интернет магазина. Часть 1 [Электронный ресурс]. – Режим доступа : <https://habrahabr.ru/company/trinion/blog/286188/>.
29. Рейтинг: «Лучший конструктор для создания интернет-магазина» [Электронный ресурс]. – Режим доступа : <https://uguide.ru/kak-otkryt-internet-magazin>.
30. МТЗ [Электронный ресурс]. – Режим доступа : https://studopedia.com.ua/1_191342_sistemi-materialno-tehnichnogo-zabezpechennya-v-ukraini-ta-formuvannya-rinku-zasobiv-virobnitstva.html.
31. Конструктор сайтов для бизнеса [Электронный ресурс]. – Режим доступа : <https://ukit.com/ru>.
32. Все начинается с великолепного сайта [Электронный ресурс]. – Режим доступа : <https://ru.wix.com/>.
33. Пора управлять всем через интернет! [Электронный ресурс]. – Режим доступа : <https://www.webasyst.ru/>.
34. Все для справжнього сайту [Электронный ресурс]. – Режим доступа : <http://nethouse.ua/>.
35. The ecommerce platform made for you [Electronic resources]. – Access mode : <https://www.shopify.com/>.
36. Конструктор сайтов weebly [Электронный resources]. – Режим доступа: <https://www.weebly.com/>.
37. Тарифный план [Электронный ресурс]. – Режим доступа : https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D1%80%D0%B8%D1%84%D0%BD%D1%8B%D0%B9_%D0%BF%D0%BB%D0%B0%D0%BD.
38. Лучшие отели и другое жилье booking.com [Электронный ресурс]. – Режим доступа : <https://www.booking.com>.
39. Пропозиції та знижки для бронювання готелів [Электронный ресурс]. – Режим доступа : <https://ua.hotels.com/>.

40. Платформа ASP.NET – Вікіпедія [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ASP.NET>
41. Модель MVC – Хабрахабр [Електронний ресурс]. – Режим доступу : <http://habrahabr.ru/post/215605/>
42. Документація з AngularJS [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/AngularJS>
43. Дронов В.А. JavaScript и AJAX в Web-дизайне [Текст] / В.А. Дронов. – СПб.: БХВ-Петербург, 2012. – 736 р.
44. Экономика и организация производства программных продуктов [Електронний ресурс]. – 2019. – Режим доступа: <http://megaobzor.net/e-book/book.html>.
45. Бедрій Я.І. Основи охорони праці: Навчальний посібник для студентів вищих навчальних закладів [Текст] / Я.І. Бедрій. – Тернопіль: Навчальна книга – Богдан, 2014. – 240 с.
46. Голінько В.І. Основи охорони праці [Текст] / В.І. Голінько. – Дніпропетровськ: Національний гірничий університет, 2010. – 271с.
47. Сусло С.Т. Цивільний захист: Навч. посібник [Текст] / С.Т. Сусло, В.М. Заплатинський, Г.М. Харамда.– К.: Арістей, 2007.– 386 с.
48. CMS [Електронний ресурс]. – Режим доступу : <https://astwellsoft.com/uk/blog/cms.html>
49. Веб-сайт [Електронний ресурс]. – Режим доступу : <http://wiki.fizmat.tnpu.edu.ua/index.php>
50. Класифікація сайтів [Електронний ресурс]. – Режим доступу : <http://estdomain.com.ua/klasifikatsiya-sajtiv-sajtiv>

ДОДАТОК А
Технічне завдання

Вступ

Програма призначена для автоматизованого бронювання місць у об'єктах розміщення.

A.1 Підстави для розробки

Підставою для розробки є завдання на дипломну роботу на тему «Дослідження та програмна реалізація інформаційних моделей розробки вебсайтів», затверджене наказом по Національному університету «Запорізька політехніка».

A.2 Призначення розробки

Програма призначена для автоматизованого бронювання місць у об'єктах розміщення.

A.3 Вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Програма має задовольняти таким вимогам до функціоналу:

– ПЗ «Booking Fast» має складатись з двох основних частин:

1) back end – частина, що надає комплекс можливостей для моніторингу замовлень та налаштувань модулю бронювання;

2) front end – частина, яку бачить звичайний користувач, та яка надає можливості перегляду каталогу номерів та замовлення;

– ПЗ «Booking Fast» має розташовуватись на віддаленому сервері, що витримує одночасне підключення не менше 1000 клієнтів, та здатен розпаралелювати їхні запити;

- ПЗ «Booking Fast» має забезпечувати функцію бронювання номерів через веб-браузер;
- ПЗ «Booking Fast» має бути створеним на базі ASP.NET;
- всі функції по збереженню даних клієнта, даних модулю бронювання, мають бути реалізовані з використанням СКБД SQL Server;
- ПЗ «Booking Fast» має забезпечувати функції прив'язки зображень до таких сутностей: номер, категорія номеру, будинок(корпус).
- ПЗ «Booking Fast» має підтримувати облік планів харчування;
- ПЗ «Booking Fast» має надавати можливість замовляти трансфер разом із бронюванням номеру;
- ПЗ «Booking Fast» має підтримувати функцію мультивалютності, та контролю за документом курсу валют;
- ПЗ «Booking Fast» має надавати можливість перемикання між декількома мовами на вибір клієнта;
- вибір номерів має забезпечуватись через систему фільтрів: кількість відпочиваючих, період відпочинку, готель та місто (для мульти-готельного модулю);
- ПЗ «Booking Fast» має підтримувати наступні налаштування: вибір мов перекладу модулю, кастомізація перекладів, вибір відображуваних фільтрів, відключення та підключення модулів (трансферів, оплати, людей в замовленні, знижок), налаштування розмітки полігонів на карті чи схемі для будинків та номерів, налаштування цін та курсів валют, налаштування періодів зміни цін та сезонів відпочинку;
- в разі системного збою ПЗ «Booking Fast», за допомогою агента контролю якості має відправляти повідомлення про системний збій адміністрації.

А.3.2 Вимоги до надійності

Система «BF» має задовольняти таким вимогам до надійності:

- ПЗ “BF” має безперебійно працювати в умовах великих навантажень та бути стійким до помилок;
- ПЗ “BF” має бути захищеним від “SQL-ін’єкцій”, та вставок з JavaScript – коду;
- дані різних клієнтів(замовників) не мають перетинатись між собою.

А.3.3 Вимоги до експлуатації

Для запуску серверної частини веб-застосунку, сервер має містити сервер ІІS, з встановленою СУБД SQL Server. Сервер повинен мати функцію «SMTP». Для прискорення швидкодії, сервер повинен мати можливість запускати веб-додаток в декількох потоках. Для запуску клієнтської частини веб-застосунку, користувач повинен мати одну з поширених версій веб-браузеру.

А.3.4 Вимоги до складу й параметрів технічних засобів

Для роботи з програмним продуктом рекомендується використовувати наступне апаратне забезпечення:

- процесор частотою 2 ГГц;
- місце на жорсткому диску – 500 Мб;
- відеокарта з пам’яттю 256 Мб;
- оперативна пам’ять – 1Гб;
- маніпулятор миша.

A.3.5 Вимоги до інформаційної й програмної сумісності

Для функціонування системи необхідною є наявність сервера, на якому встановлені сервер IIS та SQL Server 2012. Для роботи програми зі сторони клієнта необхідна ОС Microsoft Windows зі встановленим середовищем виконання CLR .NET Framework 2.0.

A.3.6 Вимоги до транспортування і зберігання

Програма повинна зберігатися на жорсткому або гнучкому магнітному диску і транспортуватися в спеціальних боксах.

A.3.7 Вимоги до програмної документації

Склад програмної документації повинен включати в себе технічне завдання, текст сайту (код), керівництво програміста, керівництво оператора та при необхідності інші програмні документи.

A.4 Стадії та етапи розробки

Програмний продукт повинен бути виконаний у 16-ти тижневий строк з моменту отримання завдання.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Програма розроблена за допомогою середовища Visual Studio. Використано мову програмування C#.

Б.2 Функціональне призначення

Програма призначена для автоматизованого бронювання місць у об'єктах розміщення.

Б.3 Використані технічні засоби

Для роботи з програмою рекомендується використовувати наступне апаратне забезпечення:

- процесор частотою 2 Гц;
- місце на жорсткому диску – 500 Мб;
- відеокарта з пам'яттю 256 Мб;
- оперативна пам'ять – 1Гб;
- маніпулятор миша.

Б.4 Виклик і завантаження

Для виклику додатку слід у будь-якому браузері необхідно перейти за веб-адресою, де встановлено програмне забезпечення. Для керування роботою додатку використовуються посилання, випадаючі списки, перемикачі та інші активні елементи.

Б.5 Вхідні дані

Вхідними даними для програми є дані інформація про готель, номери, ціновий фонд, посилання на зображення, дані замовника.

Б.6 Вихідні дані

Вихідними даними для програми є інформація щодо бажаних строків та місця замовлення готелю, а також інші додаткові характеристики, а також повідомлення про замовлення та лист на поштову скриньку.

ДОДАТОК В
Текст програми

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web.Mvc;
using MegaUtils_V3;
using Newtonsoft.Json;
using PaymentSystems.LiqPayECommerce;
using sb2014.Utils;
using sb2014.web.Classes;
using sb2014.web.Controllers.Base;
using sb2014.web.Models;
using sb2014.web.Models.Rhg;
using tfo2011EF;
using UtilsConvertTypes = sb2014.web.Classes.UtilsConvertTypes;

namespace sb2014.web.Controllers
{
    public class MultiHotelBookingController : BaseBookingModuleController
    {
        //
        // GET: /MultiHotelBooking/

        private const int DefaultAdultCount = 2;

        private const int DefaultLocalityId = 1; //КМБ

        #region CurrentBookingModuleParams

        private ObjectBookingModuleParams _currentSpravBookingModuleParams;
        public override ObjectBookingModuleParams CurrentBookingModuleParams
        {
            get
            {
                var p = _currentSpravBookingModuleParams ??
                    (_currentSpravBookingModuleParams =
DataManager.GetBookingModuleParamsByClientId(RepSysClient.DefaultClientId));
                return p;
            }
        }

        #endregion

        public ActionResult Index(ModelSelectMultiHotel model)
        {
            InitModel(model);

            return View(model);
        }
    }
}

```

```
}
```

```
private void InitModel(
    ModelSelectMultiHotel model,
    bool fullHotelCompoundInitializaton = false,
    bool initPeriod = true)
{
    model.LocalityId = model.LocalityId == 0 ? DefaultLocalityId :
model.LocalityId;

    model.Hotels =
DataManager.SelectHotels(CurrentBookingModuleParams.ClientId ?? 0);

    model.Localities =
DataManager.SelectLocalitiesByIdList(model.Hotels.Select(i => i.LocalityId ??
0), CurrentLanguageId);

    model.AdultCount = model.AdultCount ?? DefaultAdultCount;

    model.HotelId = model.HotelId == 0 ? model.Hotels.First(i =>
i.LocalityId == model.LocalityId).Id : model.HotelId;

    if (initPeriod)
    {
        if (Request != null && Request.Params["DateFromStr"] != null &&
Request.Params["DateTillStr"] != null)
        {
            model.DateFrom =
Convert.ToDateTime(Request.Params["DateFromStr"]);
            model.DateTill =
Convert.ToDateTime(Request.Params["DateTillStr"]);
        }
        model.Filters.Period =
DataManager.GetCurrentOrDefaultPeriodIfInvalid(
UtilsConvertTypes.SpravHotelToObjectHotelModel(RepProvider.CatalogHotel.GetById(
model.HotelId), CurrentLanguageId),
            CurrentBookingModuleParams,
            model.Filters.Period
        );
    }

    var hotelIdListForSearchCompound = fullHotelCompoundInitializaton ?
model.Hotels.Select(i => i.Id).ToList() : new List<int> { model.HotelId };

    model.HotelPeopleCompoundsList =
DataManager.SelectHotelsPeopleCompounds(hotelIdListForSearchCompound);

    if (model.ImmediateSearch)
    {
```

```

        model.CurrentAppCategories =
DataManager.SelectCategoriesByHotel(model.HotelId, CurrentLanguageId);
    }
}
public JsonResult Search(SearchObj obj)
{
    var res = DataManager.Search(obj);
    return Json(res);
}

public string GetCurrencyRates(int hotelId)
{
    var list = DataManager.SelectCurrencyList(hotelId,
CurrentLanguageId);
    return JsonConvert.SerializeObject(list);
}

public string GetCategories(int hotelId)
{
    try
    {
        var categories = DataManager.SelectCategoriesByHotel(hotelId,
CurrentLanguageId);
        return JsonConvert.SerializeObject(categories);
    }
    catch (Exception e)
    {
        Logger.Error("Ошибка при получении категорий: " +
MegaExceptionHandling.Instance.GetFullExceptionMessage(e));
        throw;
    }
}

public string SearchPlacement(int hotelId)
{
    var list =
RepProvider.CatalogAppCategoryPlacement.SelectAvailablePeopleCountByHotel(hotelI
d);

    return JsonConvert.SerializeObject(list);
}

private void InitViewBagsForMakeOrder(OrderModel model)
{
    var category = UtilsConvertTypes
        .SpravAppCategoryToObjectAppCategory(
            RepProvider.CatalogAppCategory.GetById(model.CategoryId),
CurrentLanguageId);

    ViewBag.Category = category;
    ViewBag.CurrencyList =
DataManager.SelectCurrencyList(model.Hotel.Id, CurrentLanguageId);
    ViewBag.CountryInfoCodeList = new SelectList(

```

```

        Classes.Utills.GetCountiesListForDropDownSelect(RepProvider,
category.HotelId, CurrentLanguageId, Translator),
        "Value",
        "Name",
        0);

    ViewBag.ParamsClientMain =
        CurrentBookingModuleParams.ClientParams.First(
            i => i.ClientType ==
RepCatalogBookingModuleParams.ClientParams.ConstClientTypeMainClient);
        /*Classes.Utills.GetBookingClientParams(
            RepProvider,
            CurrentBookingModuleParams.Id,
RepCatalogBookingModuleParams.ClientParams.ConstClientTypeMainClient);*/
    }

    public ActionResult MakeOrder(OrderModel model)
    {
        try
        {
            model.DateFrom = Convert.ToDateTime(model.DateFromStr);
            model.DateTill = Convert.ToDateTime(model.DateTillStr);

            InitOrderModel(model);

            InitViewBagsForMakeOrder(model);

            // ReSharper disable once Mvc.ViewNotResolved
            return View(model);
        }
        catch (Exception exc)
        {
            UtillsEmail.SendMessageToLutsenkoDmitry(
                "RHG" + ": Ошибка при подтверждении заказа",
                exc.Message +
                "\r\n " +
                " Модель:\r\n" + model,
                MailSvc
            );
            throw;
        }
    }

    private void InitOrderModel(OrderModel model)
    {
        //var category =
RepProvider.CatalogAppCategory.GetById(model.CategoryId);
        model.AppCategory =
            UtillsConvertTypes.SpravAppCategoryToObjectAppCategory(
                RepProvider.CatalogAppCategory.GetById(model.CategoryId));
    }

```

```

        model.Hotel = UtilsConvertTypes.SpravHotelToObjectHotelModel(
            RepProvider.CatalogHotel.GetById(model.AppCategory.HotelId),
            CurrentLanguageId);

        var categoryInfo = RepProvider.CatalogAppCategory.GetCategotyInfo(
            DateTime.Today,
            model.DateFrom,
            model.DateTill,
            model.CategoryId,
            model.AdultCount,
            model.ChildCount,
            model.ChildOlderCount,
            RepCatalogCurrency.CodeEur,
            false,
            false
        );
        model.Amount = categoryInfo.AMOUNT ?? 0;
        model.FreeAppartmentsCount = categoryInfo.FREE_APPARTMENT_COUNT ??
0;
        model.CurrencyCode = model.Hotel.MainCurrenceCode;
//RepCatalogCurrency.CodeEur;
    }

    private void InsertOrderAndSendNotification(OrderModel model)
    {
        var converter = new ConvertOrderModelToDocOrderRhg(RepProvider,
Logger);
        var order = converter.GetDocOrder(model,
CurrentBookingModuleParams);
        model.OrderId = Classes.Utils.SaveAndRecalculateOrder(RepProvider,
order);
        Classes.Utils.SendBookingNotification(
            model,
            RepProvider,
            CurrentBookingModuleParams,
            CurrentQualityAgent,
            IsRequestLocal(),
            Logger,
            MailSvc
        );
    }

    [ActionName("MakeOrder")]
    [HttpPost]
    public ActionResult MakeOrderConfirm(
        OrderModel model
    )
    {
        try
        {
            Logger.Info("Вызов MakeOrderConfirm");
            Logger.Info(JsonConvert.SerializeObject(model));
        }
    }

```

```

        InitOrderModel(model);
        UtilsValidation.ContactsValidation(
            model,
            RepProvider,
            CurrentBookingModuleParams,
            ModelState, IsRequestLocal());
        if (ModelState.IsValid)
        {
            InsertOrderAndSendNotification(model);

            if
(UtilsRhg.FindPrivat24MerchInfoByHotelId(RepProvider.CatalogAppCategory.GetById(
model.CategoryId).HOTEL_RECID, Logger, RepProvider) != null)
            {
                return
RedirectToRoute("UniversalMultiHotelBookingWithLanguage",
                    new
                    {
                        OrderId = model.OrderId,
                        //model,
                        language = CurrentLanguageCode,
                        action = "Payment"
                    });
            }
            return
RedirectToRoute("UniversalMultiHotelBookingWithLanguage",
                    new
                    {
                        language = CurrentLanguageCode,
                        action = "ThankYou"
                    });
        }

        InitViewBagsForMakeOrder(model);
        return View(model);
    }
    catch (Exception exc)
    {
        UtilsEmail.SendMessageToLutsenkoDmitry(
            "RHG" + ": Ошибка при подтверждении заказа",
            MegaExceptionHandling.Instance.GetFullExceptionMessage(exc)
+
            " \r\n " +
            " Модель:\r\n" + model,
            MailSvc
        );
        throw;
    }
}

public ActionResult Payment(
    PaymentModel model

```

```

    )
{
    try
    {

        var order = RepProvider.DocOrder.GetById(model.OrderId ?? 0);

        model.Order = order;

        var hotelId = order.HOTEL_RECID;

        var mngr = new
LiqPayECommerceManager(UtilsRhg.FindLiqPayMerchInfoByHotelId(hotelId, Logger,
RepProvider));

        const int paymentCurrency = RepCatalogCurrency.CodeUah;

        var prepaymentValueSettings =
RepProvider.CatalogPrepaymentValue.FindByHotel(1);

        var prepaymentAmountInOrderCurrency =
            RepProvider.DocOrder.GetOrderPrepaymentValue(
                order.ORDER_RECID,
                prepaymentValueSettings
                //prepaymentValueSettings.VALUE_KIND ?? 0,
                //prepaymentValueSettings.VALUE ?? 0
            );

        var orderCurrency = order.ORDER_CURRENCY_CODE;

        var rateFromOrderCurrencyToPaymentCurrency =
RepProvider.DocCurrencyRate.GetCurrencyRate(orderCurrency??0, paymentCurrency,
DateTime.Today, true);

        var prepaymentAmountInPaymentCurrency =
(prepaymentAmountInOrderCurrency * rateFromOrderCurrencyToPaymentCurrency);

        var description = "Предоплата за проживание в отеле " +
order.SPRAV_HOTEL.HOTEL_NAME + " по заказу "+order.NUMBER;

        var paymentForm = mngr.GetPaymentForm(
            model.OrderId ?? 0,
            prepaymentAmountInPaymentCurrency,
            paymentCurrency,
            description);

        model.PaymentForm = paymentForm;

        var totalPriceWithDiscountInPaymentCurrency =
(order.ORDER_TOTAL_PRICE_WITH_DISCOUNT ??
0)*rateFromOrderCurrencyToPaymentCurrency;

```

```

        ViewBag.PaymentCurrencyShortName =
RepProvider.CatalogCurrency.GetShortNameByCode(paymentCurrency);
        ViewBag.PaymentCurrencyTotalPriceWithDiscout =
totalPriceWithDiscountInPaymentCurrency;

        return View(model);

        /*
        var order = RepProvider.DocOrder.GetById(model.OrderId ?? 0);

        var hotelId = order.HOTEL_RECID;

        var p24Manager =
            new
Privat24Manager(UtilsRhg.FindPrivat24MerchInfoByHotelId(hotelId, Logger,
RepProvider));

        //var amount = 3.25;

        var prep = RepProvider.CatalogPrepaymentValue.FindByHotel(1);

        var amount =
RepProvider.DocOrder.GetOrderPrepaymentValue(order.ORDER_RECID, prep.VALUE_KIND
?? 0,
            prep.VALUE ?? 0);

        var obj = p24Manager.GetPaymentForm(model.OrderId ?? 0, amount,
RepCatalogCurrency.CodeUah);

        return View(obj);

        */
    }
    catch (Exception exc)
    {
        UtilsEmail.SendMessageToLutsenkoDmitry(
            "RHG" + ": Ошибка при генерации страницы оплаты",
            MegaExceptionHandler.Instance.GetFullExceptionMessage(exc)
+
            " \r\n " +
            " Модель:\r\n" + model,
            MailSvc
        );
        throw;
    }
}

public ActionResult ThankYou()
{
    //InitViewBagCurrentModuleParams();
    return View();
}

```

```
}
```

```
public ActionResult LiqPayECommerceRespClient()
{
```

```
    Logger.Info("Возврат значения из ECommerce - клиентская страница");
```

```
    var dic = Request.Form.AllKeys.ToDictionary(key => key, key =>
Request.Form[key]);
```

```
    Logger.Info(JsonConvert.SerializeObject(dic));
```

```
    ViewBag.Result = dic;
```

```
    return View();
```

```
}
```

```
public ActionResult LiqPayECommerceRespServer()
{
```

```
    DOC_ORDER order = null;
```

```
    try
```

```
    {
```

```
        Logger.Trace("=====");
```

```
        Logger.Trace("Поступила оплата LiqPayECommerce");
```

```
        Logger.Trace("Разбор результата оплаты LiqPayECommerce -
серверная обработка");
```

```
        var dic = Request.Form.AllKeys.ToDictionary(key => key, key =>
Request.Form[key]);
```

```
        Logger.Trace(JsonConvert.SerializeObject(dic));
```

```
        order = UtilsRhg.ProcessPayment(dic, Logger, RepProvider,
IsRequestLocal(), MailSvc);
```

```
        Logger.Trace("Обработка платежа завершена. Возвращаем статус
OK");
```

```
        return new HttpStatusCodeResult(HttpStatusCode.OK, "The request
was fulfilled");
```

```
    }
```

```
    catch (Exception exc)
```

```
    {
```

```
        var hotelinfo = "";
```

```
        if (order != null)
```

```
        {
```

```
            hotelinfo =
```

```
Utils.UtilsNames.GetHotelTypeAndName(order.SPRAV_HOTEL);
```

```
        }
```

```

        var msg = hotelinfo + ": Ошибка при разборе оплаты
LiqPayECommerce: " + "\r\n" +
MegaExceptionHandler.Instance.GetFullExceptionMessage(exc);
        Logger.Error(msg);
        Logger.Info(msg);
        Logger.Trace(msg);
        UtilsEmail.SendMessageToLutsenkoDmitry(
            hotelinfo + ": Ошибка при разборе оплаты Privat24",
            msg,
            MailSvc);
        return Content(msg);
    }
}

public ActionResult Privat24RespClient()
{
    Logger.Info("Возврат значения из ECommerce - клиентская страница");

    var dic = Request.Form.AllKeys.ToDictionary(key => key, key =>
Request.Form[key]);

    Logger.Info(JsonConvert.SerializeObject(dic));

    ViewBag.Result = dic;

    return View();
}

public ActionResult Privat24RespServer()
{
    DOC_ORDER order = null;
    try
    {
        Logger.Trace("=====");
        Logger.Trace("Поступила оплата Privat24");
        Logger.Trace("Разбор результата оплаты Приват24 - серверная
обработка");

        var dic = Request.Form.AllKeys.ToDictionary(key => key, key =>
Request.Form[key]);
        Logger.Trace(JsonConvert.SerializeObject(dic));

        order = UtilsRhg.ProcessPayment(dic, Logger, RepProvider,
IsRequestLocal(), MailSvc);

        Logger.Trace("Обработка платежа завершена. Возвращаем статус
OK");
    }
}

```

```

        return new HttpStatusCodeResult(HttpStatusCode.OK, "The request
was fulfilled");
    }
    catch (Exception exc)
    {
        var hotelinfo = "";
        if (order != null)
        {
            hotelinfo =
Utils.UtilsNames.GetHotelTypeAndName(order.SPRAV_HOTEL);
        }

        var msg = hotelinfo + ": Ошибка при разборе оплаты Privat24: " +
"\r\n" + MegaExceptionHandling.Instance.GetFullExceptionMessage(exc);
        Logger.Error(msg);
        Logger.Info(msg);
        Logger.Trace(msg);
        UtilsEmail.SendMessageToLutsenkoDmitry(
            hotelinfo + ": Ошибка при разборе оплаты Privat24",
            msg,
            MailSvc);
        return Content(msg);
    }
}

public ActionResult MakeOrderDebug()
{
    return View();
}

[ActionName("MakeOrderDebug")]
[HttpPost]
public ActionResult MakeOrderDebugConfirm(string modelJson)
{
    var model = JsonConvert.DeserializeObject<OrderModel>(modelJson);
    try
    {
        return MakeOrderConfirm(model);
    }
    catch
    {
        return View(modelJson);
    }
}

public PartialViewResult MiniPanel(ModelSelectMultiHotel model)
{
    InitModel(model, true);
    ViewBag.IsEmbedded = true;
    return PartialView("_PartMiniPanel", model);
}

```

```

    }

    public PartialViewResult MiniPanelLocal(ModelSelectMultiHotel model)
    {
        InitModel(model, true);
        ViewBag.IsEmbedded = false;
        return PartialView("_PartMiniPanel", model);
    }

    public ActionResult PriceTable(int hotelId, int? currencyCode=null)
    {
        if (DataManager.SelectHotels(CurrentBookingModuleParams.ClientId ??
0).All(i => i.Id != hotelId))
        {
            return Content("Не найдено отеля с Id " + hotelId);
        }
        var model = new PriceTableModel();
        InitPriceTableModel(model, hotelId, currencyCode);
        return PartialView("PriceTable/_PartPriceTable", model);
    }

    private void InitPriceTableModel(PriceTableModel model, int hotelId,
int? currencyCode)
    {
        var categoryList = DataManager.SelectCategoriesByHotel(hotelId,
CurrentLanguageId);
        var priceDocId =
RepProvider.DocAppCategoryPrice.FindCurrentOrFirstFutureDocId(hotelId,
currencyCode);
        var categoryPriceModels = RepProvider
            .DocAppCategoryPricePos
            .SelectByDocId(priceDocId ??
0).ToList().Select(UtilsConvertTypes.DocAppCategoryPricePosToCategoryPrice);

        model.CategoryPrices = categoryList
            .Select(i =>
                new ObjectFullCategoryPrice
                {
                    Category = i,
                    PriceList = new List<ObjectCategoryPrice>()
                }).ToList();

        foreach (var price in categoryPriceModels)
        {
            model.CategoryPrices.First(i => i.Category.Id ==
price.CategoryId).PriceList.Add(price);
        }

        model.IsDisplayBuilding =
UtilsRhg.IsShowBuildingInPriceTable(hotelId);
        if (model.IsDisplayBuilding)

```

```

        {
            model.BuildingCategoriesDictionary = new
Dictionary<ObjectBuilding, List<int>>();
            var appList = DataManager.SelectApartments(hotelId);
            var buildingList =
DataManager.SelectBuildingsByIdList(appList.Select(i => i.BuildingId),
CurrentLanguageId);
            foreach (var building in buildingList)
            {
                model.BuildingCategoriesDictionary.Add(
                    building,
                    appList.Where(i=>i.BuildingId ==
building.Id).Select(i=>i.CategoryId).Distinct().ToList()
                );
            }
        }
    }
}

```

ДОДАТОК Г
Слайди презентації

Дослідження та програмна реалізація інформаційних моделей розробки вебсайтів

Виконав: ст. гр. КНТ-129м А. В. Демічев

Керівник: доцент, к.т.н. С.Ю.Скрупський

Рисунок Г.1 – Слайд № 1

Об'єктом дослідження є процес розробки вебсайтів.

Предмет дослідження — методи та засоби розробки вебсайтів.

Мета роботи — дослідження та реалізація інформаційних моделей розробки вебсайтів.

Рисунок Г.2 – Слайд № 2

Задачі роботи

- аналіз процесу розробки вебсайтів;
- розробка технічного завдання для реалізації програми;
- вибір середовища розробки програмного забезпечення;
- створення інформаційних моделей розробки вебсайтів;
- конструювання програмного забезпечення у вигляді вебсайту з продажу місць в об'єктах розміщення;
- тестування розробленого програмного забезпечення.

3

Рисунок Г.3 – Слайд № 3

Види веб-сайтів

- комерційний веб-сайт;
- некомерційний веб-сайт;
- інформаційний портал;
- сервісний веб-сайт;
- **інтернет магазин** — засіб електронної торгівлі.

Будь-який веб-сайт, що забезпечує можливість замовлення товару / послуги з вибором з каталогу прийнято вважати **Інтернет-магазином**

4

Рисунок Г.4 – Слайд № 4

Програмна реалізація інтернет-магазинів

- розробка на основі спеціалізованого програмного забезпечення — використання систем керування вмістом сайту CMS (content management system) та спеціальних платформ-конструкторів для створення інтернет-магазинів;
- розробка без використання спеціалізованого програмного забезпечення.

5

Рисунок Г.5 – Слайд № 5

Аналіз засобів розробки вебсайтів інтернет-магазинів

Платформа	SEO підтримка	Кастомізація дизайну	Розширення функціональності	Адаптація до регіонального ринку	Приховані платежі
WooCommerce	+	+	+	—	—
OpenCart	+	+	+	—	—
Еквід	+—	+—	+—	—	—
Recommerce	+	+	+	+	+
MotoCMS	+	+—	+—	+	+
Wix	—	+—	+—	+	—
Prom.ua	—	—	—	+	—

6

Рисунок Г.6 – Слайд № 6

Модель системи у вигляді діаграми класів

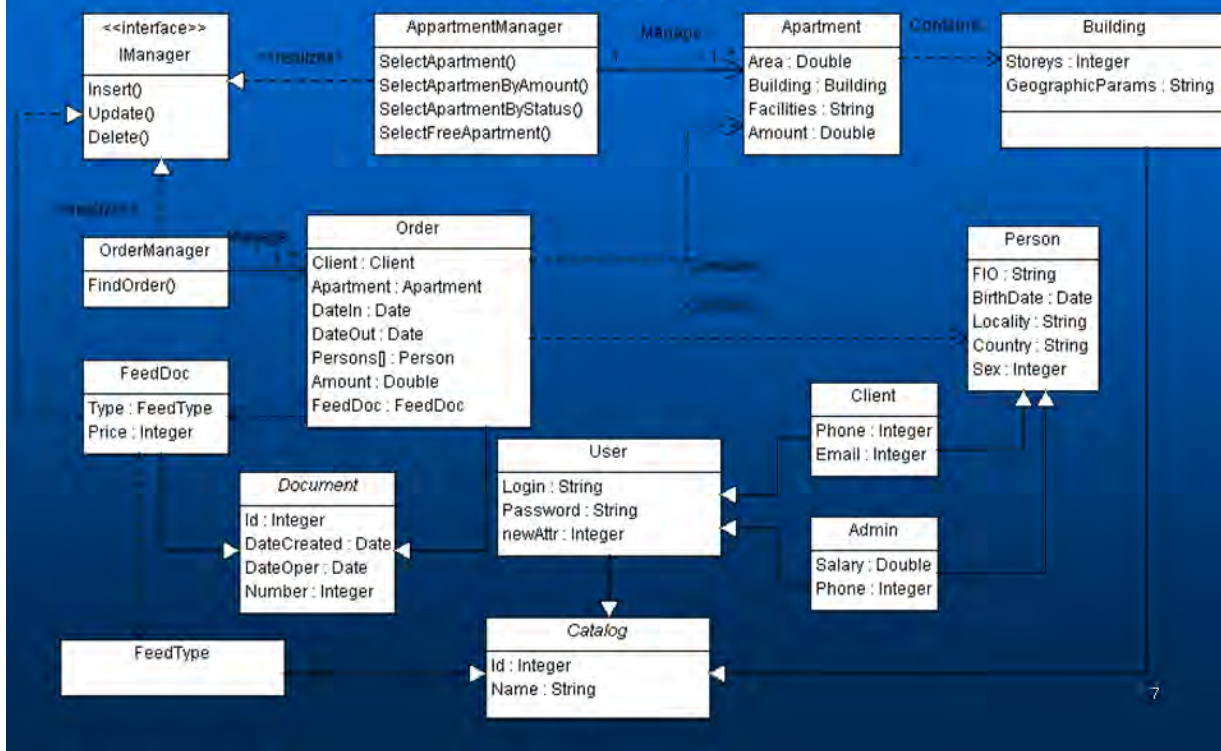


Рисунок Г.7 – Слайд № 7

Моделі системи у вигляді діаграм прецедентів (варіантів використання)

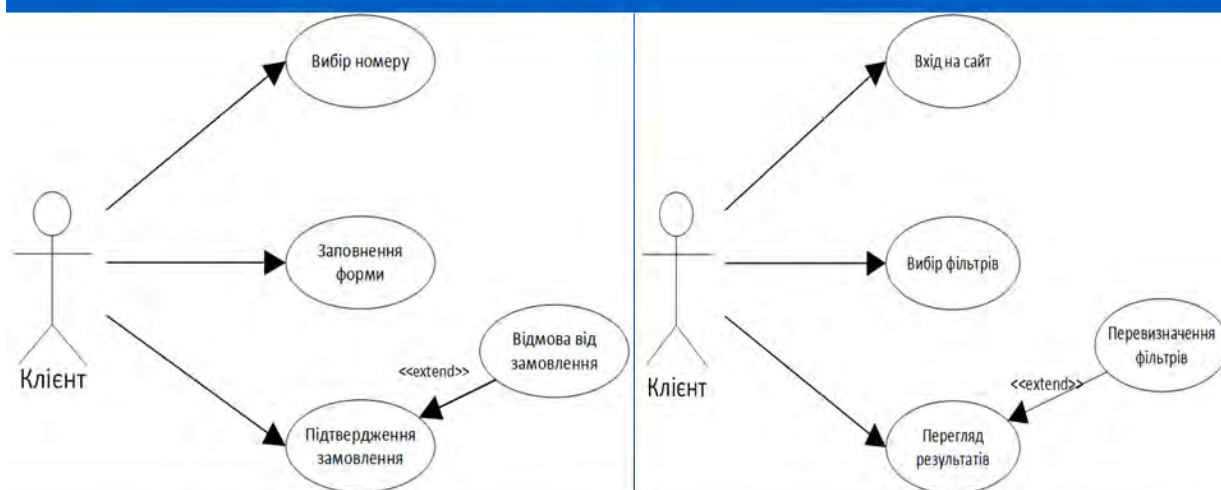
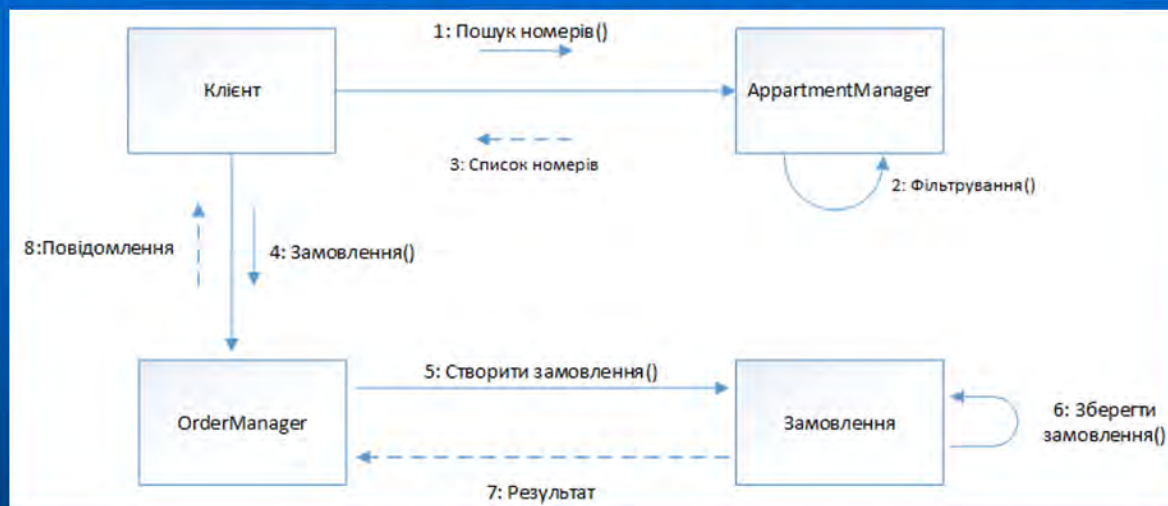
Діаграма прецедентів
«Замовлення номеру»Діаграма прецедентів
«Пошук номеру»

Рисунок Г.8 – Слайд № 8

Моделі системи у вигляді діаграм кооперації



Діаграма кооперації Клієнт-система



Діаграма кооперації Адміністратор-система

9

Рисунок Г.9 – Слайд № 9

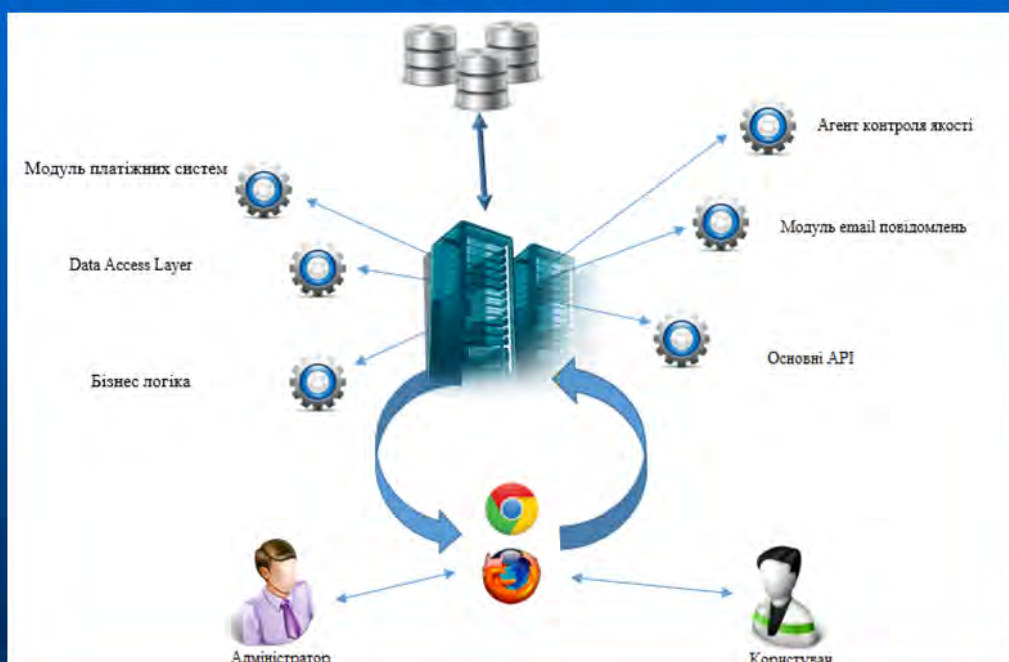
Вибір мови програмування

Критерій порівняння	C#	VB.NET	Java
Підтримка об'єктно-орієнтованої парадигми програмування	+	+	+
Підтримка імперативної парадигми програмування	+	+	+
Наявність open-source компілятора	+	+	+
Зручність використання для створення інтернет-сайтів	+/-	+/-	+
Зручність використання для створення windows-застосунків	+	+/-	+/-
Значення параметрів та замочення	+	+	-
Кросплатформність	+	+	+

10

Рисунок Г.10 – Слайд № 10

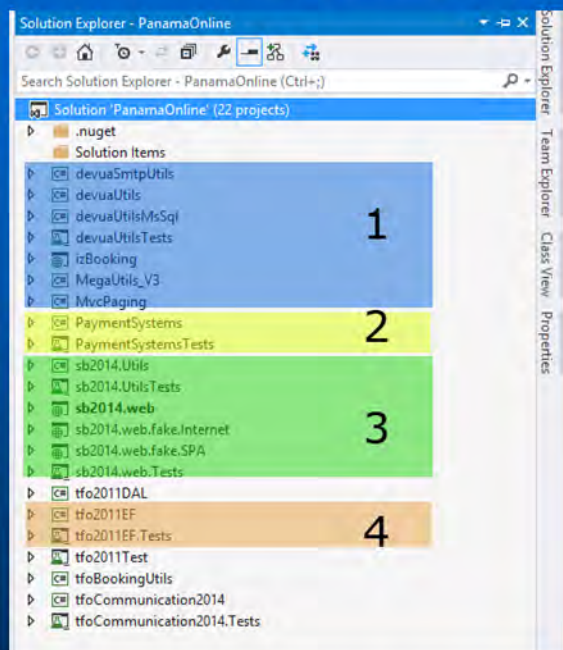
Загальна схема системи



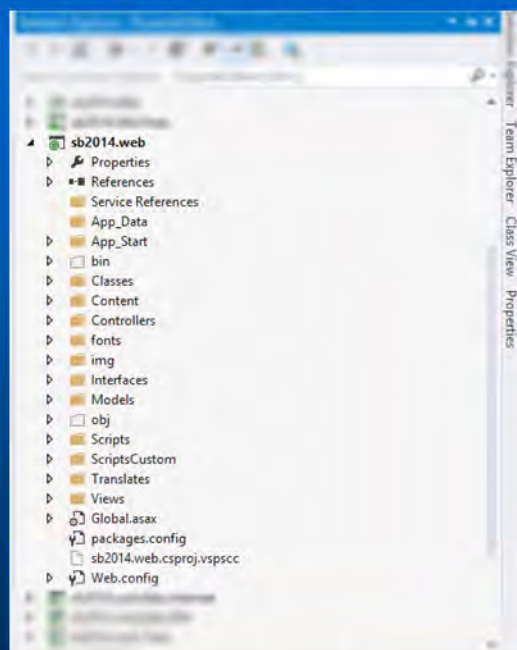
11

Рисунок Г.11 – Слайд № 11

Файлова структура системи



Файлова структура проекту



Структура веб-застосування

12

Рисунок Г.12 – Слайд № 12

Интерфейс системы. Пошук за заданими критеріями

Language: Ru Ua En

Город: Киев Отель: Royal City Hotel

Период: 06.12.2020 13.12.2020 Взрослых: 2 Детей до 12: 0 до 2: 0

Выбрать

Категория	Стоимость (За весь период)
Single	231.00 eur

Просторный номер с изысканным дизайном, продуманным до мельчайших деталей. Зеркальный шкаф, столик и тумбочки в викторианском стиле – все это создает незабываемую, легкую и ненавязчивую, атмосферу прошлых веков. Тем не менее, современная комплектация комнаты не даст Вам забыть, что все удобства XXI века к Вашим услугам.

Заказать

13

Рисунок Г.13 – Слайд № 13

Интерфейс системы. Бронювання номеру та заповнення контактних даних

Подтверждение заказа

[Назад](#)

Проживание

Цена может измениться в зависимости от курса и избранных дополнительных услуг. Refundable - в случае аннуляции за 1 сутки до даты заезда штраф не взимается. В случае аннуляции позднее этого срока или в случае не-шоу взимается 100 процентов от стоимости первой ночи проживания.

Город	Киев
Отель	Royal City Hotel
Категория	Junior Suite
Состав	Взрослых: 2
Стоимость	350.00 eur

Контактные данные

Фамилия Имя Отчество *

☐ Платежщик не совпадает с заказчиком

Мобильный телефон *

Email *

Страна [Выберите страну]

Населенный пункт

* - обязательные для заполнения поля

14

Рисунок Г.14 – Слайд № 14

Висновки

У результаті виконання роботи було розв'язано актуальне завдання дослідження та програмної реалізації інформаційних моделей розробки вебсайтів.

Проаналізовано процес розробки веб-сайтів інтернет-магазинів. Досліджено методи і засоби створення веб-сайтів інтернет-магазинів.

Наукова новизна роботи полягає у тому, що розроблено модель програмного забезпечення інтернет-магазину для бронювання місць у об'єктах розміщення, що представляє собою набір діаграм у нотації UML, які дозволяють розробнику створювати програмне забезпечення відповідно до вимог замовника та спрощують розробку інтернет-магазинів без необхідності використання спеціалізованих платформ-конструкторів та забезпечують можливість модифікації розроблених сайтів у майбутньому при зміні вимог користувачів.

Практичне значення отриманих результатів полягає в тому, що розроблено веб-сайт інтернет-магазину для бронювання місць у об'єктах розміщення на основі запропонованих інформаційних моделей у вигляді UML-діаграм.

Проведені експерименти з тестування розробленого програмного забезпечення засвідчили, що розроблена програма має достатню працездатність і може використовуватися за призначенням.

Рисунок Г.15 – Слайд № 15