

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

_____ проф. Приходько С.Б.

«___» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

**на тему: Розробка програмного забезпечення інтернет-магазину
«Soft IT»**

Виконав: студент групи 4151



_____ Лопата М.М.

(підпис, ПІБ)

Керівник роботи:

_____ доцент, к.ф.-м.н., доцент

(посада, науковий ступень вчене звання)

_____ Латанська Л.О.

(підпис, ПІБ)

Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

_____ доц. Макарова Л.М.

(підпис)

«___» _____ 2022 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття ступеня вищої освіти «бакалавр»

Студенту Лопаті Михайлу Михайловичу

_____ (Прізвище, ім'я, по батькові)

Тема роботи: Розробка програмного забезпечення інтернет-магазину «Soft IT»

Керівник роботи Латанська Людмила Олексіївна

Затверджені наказом ректора №256-уч від «11» травня 2022 р.

2. Термін подання роботи: 10.06.2022 р.

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____

Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Опис предметної галузі та постановка задачі; Проект програмного забезпечення (ескізний, технічний та робочий); Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів 1. Титульний, 2. Мета і завдання дослідження 3. Модель варіантів використання, 4. Діаграми діяльності варіантів використання, 5. Концептуальна модель даних програмного забезпечення, 6. Логічна модель даних програмного забезпечення, 7. Діаграма класів, 8. Діаграми послідовності варіантів використання, 9-11. Результати розробки, 12-13. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 11.05.2022 р. _____

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу Вступ	24.03.2022	ПП*
2.	Опис та аналіз предметної галузі	28.03.2022	ПП*
3.	Постановка задачі	31.03.2022	ПП*
4.	Ескізний проєкт програмного забезпечення	27.04.2022	
5.	Технічний проєкт програмного забезпечення	06.05.2022	
6.	Робочий проєкт програмного забезпечення	13.05.2022	
7.	Підготовка розділу Результати розробки програмного забезпечення	17.05.2022	
8.	Підготовка розділу з охорони праці	18.05.2022	ПП*
9.	Підготовка розділу Висновки	19.05.2022	
10.	Оформлення списку використаних джерел та додатків	20.05.2022	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	10.06.2022	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
12.	Підготовка презентації та доповіді	12.06.2022	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	14.06.2022	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді	24.06.2022	

* - за результатами переддипломної практики (ПП), яка була з 21.03.2022 до 08.05.2022 р.

Студент



(підпис)

М.М. Лопата

(ПІБ)

Керівник роботи

(підпис)

Л.О. Латанська

(ПІБ)

АНОТАЦІЯ

В даній роботі розв'язується практична задача інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробляється програмне забезпечення інтернет-магазину «Soft_IT».

Робота включає в себе: аналіз предметної галузі, постановку задачі, проект програмного забезпечення, розділ з охорони праці, технічне завдання, програму і методику випробувань, інструкцію користувача, опис програми, програмний код.

Робота викладена українською мовою на 95 сторінках друкованого тексту, містить 4 таблиці, 40 рисунків, 5 додатків та список використаних джерел з 6 найменувань.

ABSTRACT

This work solves the practical problem of software engineering, which is characterized by complexity and uncertainty of conditions, namely the development of software online store "Soft_IT".

The work includes: analysis of the subject area, problem statement, software project, section on labor protection, terms of reference, program and test methods, user manual, program description, program code.

The work is set out on the Ukrainian language on 95 typewritten pages, contains 4 tables, 40 figures, 5 appendices and list of references from 6 names.

ЗМІСТ

ВСТУП.....	7
1 ОПИС ІСНУЮЧИХ ТЕХНОЛОГІЙ РОЗРОБКИ WEB-САЙТІВ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНУ «SOFT_IT».....	9
1.1 Опис існуючих технологій розробки Web-сайтів.....	9
1.2 Опис роботи інтернет-магазину	15
1.3 Організація роботи з покупцями в інтернет-магазині.....	16
1.4 Обґрунтування необхідності розробки програмного забезпечення інтернет-магазину «Soft_IT».....	17
1.5 Постановка задачі на розробку програмного забезпечення інтернет-магазину «Soft_IT».....	17
2 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНУ «SOFT_IT».....	20
2.1 Ескізний проект програмного забезпечення інтернет-магазину «Soft_IT».....	20
2.1.1 Побудова моделі варіантів використання ПЗ інтернет-магазину «Soft_IT».....	20
2.1.2 Специфікації варіантів використання ПЗ інтернет-магазину «Soft_IT».....	24
2.1.3 Сценарії основних варіантів використання ПЗ інтернет-магазину «Soft_IT» (діаграми діяльності)	30
2.1.4 Концептуальна модель даних ПЗ інтернет-магазину «Soft_IT».....	32
2.1.5 Розробка інтерфейсу користувача ПЗ інтернет-магазину «Soft_IT».....	33
2.2 Технічний проект програмного забезпечення інтернет-магазину «Soft_IT».....	37
2.2.1 Статична модель програмного забезпечення інтернет-магазину	37
2.2.2 Специфікації класів ПЗ інтернет-магазину «Soft_IT».....	38
2.2.3 Динамічна модель ПЗ інтернет-магазину «Soft_IT».....	42
2.2.4 Логічна модель даних ПЗ інтернет-магазину «Soft_IT».....	45
2.3 Робочий проект програмного забезпечення інтернет-магазину «Soft_IT».....	47
2.3.1 Обґрунтування вибору мови програмування та засобів розробки програмного забезпечення.....	47
2.3.2 Побудова діаграми компонентів ПЗ інтернет-магазину «Soft_IT».....	48

2.3.3 Побудова діаграми розміщення ПЗ інтернет-магазину «Soft_IT».....	48
2.3.4 Реалізація та тестування основних модулів програмного забезпечення інтернет-магазину «Soft_IT».....	49
2.3.5 Випробування ПЗ інтернет-магазину «Soft_IT».....	52
3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНУ «SOFT_IT».....	55
4 ОХОРОНА ПРАЦІ.....	56
4.1 Загальні принципи охорони праці.....	56
4.2 Аналіз небезпечних і шкідливих факторів при роботі на ПК в інтернет-магазині «Soft_IT».....	58
4.3 Розробка заходів щодо зменшення впливу небезпечних і шкідливих факторів ...	61
ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	64
ДОДАТОК А – Технічне завдання на розробку програмного забезпечення інтернет-магазину «Soft_IT».....	65
ДОДАТОК Б – Програма та методика випробувань програмного забезпечення інтернет-магазину «Soft_IT».....	71
ДОДАТОК В – Керівництво користувача програмного забезпечення інтернет-магазину Soft_IT».....	74
ДОДАТОК Г – Опис програмного забезпечення інтернет-магазину «Soft_IT».....	81
ДОДАТОК Д – Текст програмного забезпечення інтернет-магазину «Soft_IT».....	85

ВСТУП

В наш час мережа Internet набирає все більшої популярності, користувачі мережі все частіше використовують її не тільки з ціллю пошуку інформації та спілкування, але й для замовлення потрібних їм товарів. Здійснити замовлення через мережу на даний момент дозволяє спеціальне програмне забезпечення отримання замовлень через мережу Internet (сайти інтернет-магазинів).

Сайти інтернет-магазинів – це добре продумані програмні продукти, які налаштовуються під вподобання окремих покупців, надають їм змогу переглянути каталог продукції та окрему інформацію по кожному товару, залишити відгук, замовити товар і оплатити його. За допомогою таких програмних ресурсів можливо побудувати досить вдалий бізнес в мережі та успішно реалізовувати будь-яку продукцію.

Такий спосіб торгівлі є особливо актуальним для невеликих груп розробників програмного забезпечення, або для розробників початківців. Тому було прийнято рішення про розробку інтернет-магазину з продажу програмного забезпечення. Цей сайт має стати зручною платформою одночасно як для розробників, так і для покупців. Він дозволить розробникам знизити ціну на програмний продукт за рахунок зменшення витрат на рекламу, а отже підніме його конкурентну здатність. Також цей програмний продукт дозволить розробникам початківцям збільшити свою клієнтську базу за рахунок користувачів сайту.

На сьогоднішній день існують готові сайти інтернет-магазинів. Але можна навести досить вагомі аргументи проти їхнього використання: частина програмних продуктів цієї сфери являються ліцензійними, чимало коштують, мають надлишковий функціонал та розроблялися спеціально під окремі підприємства; всі безкоштовні аналоги не відповідають існуючим вимогам та досить невдало побудовані для оптимізації.

Метою даної кваліфікаційної роботи є розв'язання практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробка якісного, зручного, багатофункціонального інтернет-магазину з продажу комп'ютерних програм з найменшими витратами.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- дослідити предметну галузь програмного забезпечення;
- сформулювати вимоги до програмного забезпечення;
- виконати ескізне проектування;
- виконати технічне проектування;
- виконати робоче проектування: кодування, тестування, випробування програмного забезпечення;
- розробити документацію.

1 ОПИС ІСНУЮЧИХ ТЕХНОЛОГІЙ РОЗРОБКИ WEB-САЙТІВ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНУ «SOFT_IT»

1.1 Опис існуючих технологій розробки Web-сайтів

Створення Web-сайту за допомогою вже існуючих шаблонів

Шаблон сайту – це вже готова оболонка, яку потрібно заповнити своєю інформацією (текстами, фото або відео). В Інтернеті представлений досить багатий вибір шаблонів.

Переваги використання шаблонів

- Доступність. Шаблони можна знайти в інтернеті, причому абсолютно безкоштовно і на будь-яку тематику.
- Вони заощадають час і гроші. Користувачі не витрачають ресурсів на розробку і складання власного проекту. Необхідно скачати потрібний макет і наповнити його інформацією про компанію. Не потрібно узгоджувати правки і докладно пояснювати, що хочуть побачити – вибирають вже кінцеву версію.
- Простота використання. Шаблони сайтів не вимагають знань або досвіду в сфері програмування або ІТ: потрібно тільки додати інформацію у відповідне поле.
- Великий вибір. Зрозуміло, що доведеться витратити якийсь час, щоб вивчити запропоновані варіанти і співвіднести їх з існуючими вимогами. Фінальний макет обійдеться без сюрпризів і несподіваних дрібних недоліків.
- Можна самостійно вносити зміни. Шаблони сайтів робляться для загального використання – в них легко вносити правки. Працюючи з персональним дизайнером, маніпуляції з сайтом доведеться виконувати через нього. Це відтягне момент готовності сайту і зв'яже руки при подальшій експлуатації.
- Особливі навички не потрібні. Замовнику не потрібно кодувати інформацію.

- Майже безмежні можливості вибору і використання шаблонів.

Недоліки використання шаблонів:

- Відсутність унікальності.
- Немоżliвість змінити ключові моменти в оформленні.
- В безкоштовних шаблонах відсутність гарантійного обслуговування.
- Висока ймовірність збоїв при неякісній верстці.

Створення сайту за допомогою системи управління контентом (CMS).

CMS – це програма, яка встановлюється на віддалений сервер, під'єднаний до інтернету, і виконує такі основні функції:

- Формує та показує сторінки сайту, користуючись наперед сформованими текстами, рисунками, таблицями та іншими матеріалами, які зберігаються у базі даних.

- Допомагає користувачу без знань програмування керувати сайтом.

Існує багато движків, частина з яких – платні версії, частина – безкоштовні. До платних версій відноситься двигун DLE. До безкоштовної версії відносяться движки WordPress, Drupal, Joomla, Media Wiki. Всі вони дуже функціональні, все залежить від того, що необхідно створити на їх основі.

Недоліки від використання CMS:

- Неекономічність серверних ресурсів.
- Загалом незручна архітектура побудови програми.
- Важкість нарощування функціоналу.

Більша частина *CMS* – комерційні.

Створення Web-сайту за допомогою мов програмування

До числа широко використовуваних в Web-розробках мов відносяться JavaScript, PHP, Perl, Python, Ruby, Java, C#. З них Perl, PHP, Python, JAVA, Ruby, C# – мови, які виконуються на сервері (серверні мови). JavaScript – мова, що виконується в браузері (клієнтська). Комп'ютер, що відкриває доступ до власних ресурсів, носить назву сервера, а отримує такий доступ – клієнт [1].

- *Мова програмування Perl* (Practical Extrifiction and Report Language) ефективна при обробці текстових файлів та має такі можливості: по-перше,

вона здатна обробляти величезну кількість документів на Web-сайті (сервері); по-друге, багато серверів написані саме на мові Perl [1].

Програмісту Perl допоможе перевірити вірність перехресних посилань на сайтах, здійснить перевірку заповнення елементів форми, дозволить приєднувати бази даних з наступною їх обробкою. Perl може працювати і з протоколом FTP. Це дозволяє автоматизувати отримання файлів з інших серверів, а в поєднанні з можливостями мови з обробки текстових файлів надає програмісту можливість створювати складні інформаційні системи.

- *Мова програмування PHP* створювалася спеціально для мережі. PHP найкраще охарактеризувати так: програміст пише сценарії – код; сервер, отримуючи готові вказівки, обробляє інформацію і виводить результат на монітор користувача.

PHP допомагає створювати динамічні WEB-додатки, такі як форуми, лічильники статистики, може працювати з великими обсягами даних – масивами. Дуже добре мова PHP підходить для написання систем управління сайтами (CMS).

Для того щоб сайт швидше завантажувався, варто розбити сторінку на окремі елементи. Мова програмування PHP дозволяє за лічені частки секунди об'єднувати окремі складові в цільну WEB-сторінку. Основна перевага мови PHP – простота вивчення. Мова буде інтуїтивно зрозуміла програмістам, що володіють навиками в таких мовах, як Perl і C.

У PHP немає підтримки посилань на змінні, через що складно обробляти масштабні структури даних, наприклад, вкладені списки. Також немає перевірки типів даних функцій, які передаються, що може спричинити численні помилки.

- *Мова програмування Python* проста для освоєння, а між тим ця мова програмування на рівень вище, ніж PHP і C. Це реалізується за допомогою використання наступних структур даних: списків, масивів, словників.

Головна перевага Python – реалізація інтерпретатора на більшості платформ та операційних систем. Python має можливість розширюватися.

Тобто будь-який програміст може дописати й удосконалити Python під себе. Python – абсолютно безкоштовна і при цьому проста у вивченні мова.

Єдиний недолік Python – це деяка «неквапливість».

- *Мова програмування JAVA* – це повноцінна кросплатформенна мова програмування, яка буде правильно оброблена і комп'ютером з операційною системою Windows, і ПК під управлінням UNIX.

Мовою JAVA для мережі розробляються невеликі програмки, названі аплетами, вони завантажуються з Інтернету і виконуються браузером (прикладом аплету можуть послужити різного роду ігри, наприклад, шашки). Більшість WEB-камер, що передають на сайти зображення в реальному часі, також працюють на базі програм JAVA.

- *Мова програмування Ruby* – одна з наймолодших мов програмування. Своім ім'ям вона зобов'язана дорогоцінному каменю рубіну (за аналогією з іншою широко поширеною мовою програмування Perl – перлина).

Ось як описує Ruby її творець, японський програміст Юкіхіро Мацумото: «Це потужна і динамічна об'єктно-орієнтована мова з відкритим кодом, яку я почав розробляти в 1993 році. Ruby працює на багатьох платформах, включаючи Linux і багато реалізацій Unix, MS-DOS, Windows 9x/2000/NT. Головна мета Ruby – ефективність розробки програм, і користувачі переконуються, що програмування на ній ефективне і навіть забавне».

- *Мова програмування C#*

Альтернативним варіантом для Web розробок являється технологія .NET. Не кажучи про те, що сама платформа є продуктом з закритим кодом, мова програмування C# включає в себе переваги мов C++ і Java.

Розглянемо деякі переваги розробки сайту з використанням мови високого рівня C# [2]:

- Компілювання коду.

Написаний код при першому зверненні компілюється і згодом виконується вже скомпільований код, на відміну від PHP. Це помітно прискорює роботу додатків. Веб-сервер сам виконує компіляцію. Якщо

замінити текст програми, то сервер при першому зверненні до сторінки сам проведе перекомпіляцію.

- Код виконується швидше, займає менше ресурсів, і при цьому процес розробки не ускладнюється. Швидше навпаки, в разі помилки можна отримати повний лістинг компілятора, з докладним описом помилки.

Основні характеристики мови:

- C# – це проміжний варіант між C++ і Visual Basic, якщо розглядати складність мовних конструкцій і можливості мови.

- C# має C стиль синтаксису (для керуючих конструкцій, блоків коду, опису сигнатури методів і ін.). Багато спільного з Java (відсутність множинного спадкоємства і шаблонів, наявність збирача сміття) і Дельфі (орієнтованість на створення компонент), в той же час має і свій колорит.

В основі дизайну мови – легкість використання, яка домінує над швидкістю та потужністю. Звідси і збирач сміття з керованими об'єктними посиланнями, який автоматично звільняє пам'ять. Також користувач отримує безпеку роботи з типами, а це, на думку багатьох, є найважливішим фактором уникнення помилок.

- C# відноситься до об'єктно-орієнтованих мов, що мають можливості розробляти компоненти.

При створенні мови розглядалася не тільки простота створення програмного забезпечення, але і його підтримки – в зв'язку з чим в мову включили підтримку XML коментарів і контроль версій.

Крім збирача сміття і безпеки роботи з типами для уникнення помилок в мові також застосовується автоматична ініціалізація змінних.

- Можливості розширення рішення

Включена підтримка мультипроцесорних і кластерних рішень. Що дозволяє при написанні програми, розраховувати на те, що систему можна буде без зусиль розширювати.

- Обробка помилок.

У зв'язку з новими концепціями додані нові можливості по обробці помилок. На стадії розробки можна отримати повну інформацію про помилку і лістинг потрібної частини коду.

- *Мова програмування JavaScript* – на відміну від перерахованих вище мов, JavaScript працює не на сервері, а на стороні клієнта, тобто підключається безпосередньо в HTML-файл. Програма (сценарій) на мові JavaScript обробляється вбудованим в браузер інтерпретатором.

У JavaScript є свій стандарт: ECMAScript. Хтось каже, що JavaScript схожа на Python, хтось говорить про схожість з мовами Ruby, Self. Правда полягає в тому, що JavaScript сама по собі. Це особлива мова.

JavaScript використовується для:

- Розробки динамічних, інтерактивних сторінок;
- Контролю, перед пересиланням на сервер, коректності занесення інформації в форми;
- Розв'язання «локальних» задач з використанням сценаріїв і т.і.

JavaScript надає наступні можливості:

- Редагування сторінок, написання тексту, додавання та вилучення тег, зміна стилів елементів;
- Реагування на події: скрипт може чекати, коли що-небудь трапиться і в відповідь виконувати функції;
- Виконання серверних запитів і завантаження даних без перезавантаження сторінок;
- Перевірка коректності даних, виведення повідомлення, встановлення та зчитування cookie і т.і.

Унікальність і переваги Javascript полягають всього в декількох пунктах:

- Повна інтеграція з браузером;
- Прості речі робляться просто;
- Підтримується майже скрізь.

Цей набір переваг не можна знайти ні в одній з інших технологій. Наприклад, такі технології як ActiveX, VBScript, XUL – підтримуються не в

кожному браузері. Такі технології Flash, Silverlight, Java – не повністю інтегровані з браузером, працюють у своєму оточенні. Тому Javascript – унікальна технологія, і такою залишиться.

1.2 Опис роботи інтернет-магазину

Інтернет-магазин являється веб-сайтом та використовується для ознайомлення потенційних покупців з товаром, що реалізується, для залучення покупців, поживлення торгівлі та підвищення прибутку.

За допомогою інтернет-магазину можна ознайомитися з асортиментом товарів, отримати детальну інформацію про товар і, не виходячи з дому, здійснити покупку.

Також в деяких інтернет-магазинах реалізована можливість вивчення вподобань клієнтів через попередні відвідування, проведення анкетування та конференцій. Інтернет-магазини потребують менших затрат на своє відкриття, утримання та функціонування.

Недоліками інтернет-магазинів є неможливість перевірки існування та якості товару, наявність шахраїв, можливість існування проблем з доставкою та оплатою.

Інтернет-магазини повинні задовольняти наступним вимогам:

- Зручний та простий інтерфейс;
- Наявність необхідного функціоналу;
- Затребуваний асортимент товарів;
- Детальна інформація про товари;
- Різноманітні форми доставки та оплати.

В наш час все більше покупців користуються послугами інтернет-магазинів як вітчизняних, так і закордонних.

1.3 Організація роботи з покупцями в інтернет-магазині

Процес обслуговування клієнтів полягає в наступному

- Завантаження сайту магазину та знайомство з представленими в його каталозі товарами: в каталозі повинна бути представлена найточніша інформація про товари та їх характеристики. Також в каталозі часто містяться фото товару (інколи тривимірні зображення).

- Вибір необхідних товарів: клієнт обирає необхідні товари, які попадають в кошик. В кошику міститься інформація про обраний товар, його кількість та загальна вартість. Товар з кошику можна вилучити з перерахунком вартості, або зовсім очистити.

- Підтвердження замовлення: після остаточного вибору товарів необхідно підтвердити замовлення.

- Реєстрація на сайті: для оформлення покупки необхідно зареєструватися на сайті в спеціальній формі.

- Вибір форм оплати та доставки: також необхідно обрати більш зручну форму оплати та доставки.

Замовлення обробляється інтернет-магазином, перевіряється наявність на складі. У випадку відсутності товару повідомляється дата його поставки.

- Оплата замовлення: у випадку наявності товару здійснюється оплата замовлення та його доставка. Оплата може здійснюватися і накладним платежем.

- Доставка замовлення: В залежності від обраної форми доставки, клієнт отримує товар в той чи інший спосіб.

Клієнт може відстежувати статус замовлення, отримувати довідкову інформацію з сайту та відповіді на питання.

Існують також інтернет-магазини, де частину роботи по веденню замовлення виконує менеджер.

1.4 Обґрунтування необхідності розробки програмного забезпечення інтернет-магазину «Soft_IT»

На сьогоднішній день існують готові сайти інтернет-магазинів. Але можна навести досить вагомі аргументи проти їхнього використання: частина програмних продуктів цієї сфери являються ліцензійними, чимало коштують, мають надлишковий функціонал та розроблялися спеціально під окремі підприємства; всі безкоштовні аналоги не відповідають існуючим вимогам та досить невдало побудовані для оптимізації.

Розроблене ж програмне забезпечення буде абсолютно безкоштовним і надасть неоцінений поштовх в розвитку бізнесу. Також, використовуючи готові рішення, не можливо гарантувати абсолютну безпеку даних та захист від дій зловмисників або деяких невидимих на перший погляд недоліків, які в подальшому можуть призвести до неможливості подальшого використання цього сайту або суттєвих фінансових втрат та втрати репутації. При розробці ж власного програмного забезпечення ми можемо бути впевнені в його якості, а також в безпеці його використання.

Також розробка власного сайту допоможе зробити його максимально зручним та функціональним для вирішення саме наших задач і функцій, що в свою чергу зробить його максимально зручним для користувача.

1.5 Постановка задачі на розробку програмного забезпечення інтернет-магазину «Soft_IT»

Виходячи з вище викладеного конкретизуємо постановку задачі. В даній кваліфікаційній роботі необхідно розробити програмне забезпечення інтернет-магазину «Soft_IT», яке має відповідати зазначеним нижче вимогам.

Вимоги до складу виконуваних функцій

Програма повинна забезпечувати можливість виконання наведених нижче функцій:

- Додавання на сайт нових комп'ютерних програм від користувачів (вхідні дані: файл програми, назва програми, опис програми; вихідні дані: відображення на сайті інформації про успішне додання програми);
- Видалення програм з сайту (вхідні дані: id програми; вихідні дані: відображення на сайті інформації про успішне видалення);
- Завантаження програм з сайту (вхідні дані: id програми; вихідні дані: файл з програмою);
- Можливість залишати коментарі до кожної з програм (вхідні дані: текст, введений користувачем у спеціальне текстове поле для коментарів; вихідні дані: відображення на сайті коментаря користувача);
- Реєстрація користувачів (вхідні дані: логін, пароль та електронна пошта користувача; вихідні дані: відображення на сайті інформації про успішну реєстрацію);
- Аутентифікація користувачів (вхідні дані: логін та пароль; вихідні дані: відображення на сайті інформації про успішну авторизацію).

Вимоги до організації вхідних і вихідних даних

Вхідні дані програми повинні вводитися у спеціальні поля на сайті, або бути організовані у вигляді файлу, обраного на жорсткому диску.

Вихідні дані програми мають відображатися на сайті, або зберігатися в файл.

Вимоги до інформаційної та програмної сумісності

Вимоги до вихідних кодів і мов програмування

Вихідні коди програми повинні бути реалізовані на мові C#. В якості інтегрованого середовища розробки програми може бути використане середовище Visual Studio.

Вимоги до програмних засобів, використовуваних програмою

Системні програмні засоби, які використовуються програмою, повинні бути представлені ліцензійною версією операційної системи Windows (версія не нижче Windows 7) та встановленою на комп'ютері платформою .NET не нижче версії 4.5.

Вимоги до складу та параметрів технічних засобів

Для роботи із програмою необхідний наступний склад технічних засобів з мінімальними параметрами:

- Процесор Intel Pentium 1Ghz і вище.
- Оперативна пам'ять 512Мб і вище.
- Мінімум дискового простору 1 Гб.
- Екран з роздільною здатністю не менше ніж 800х600 пікселів.

2 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНУ «SOFT_IT»

В ході розробки проекту програмного забезпечення було виконане ескізне, технічне та робоче проектування. Під час розробки проекту використовувалася об'єктно-орієнтована методологія та уніфікована мова моделювання UML [3]. В якості CASE-засобу був обраний Rational Rous.

2.1 Ескізний проект програмного забезпечення інтернет-магазину «Soft_IT»

Після аналізу вимог до ПЗ інтернет-магазину «Soft_IT» був розроблений ескізний проект даного ПЗ шляхом побудови діаграми варіантів використання і специфікацій варіантів використання, для формалізації представлення сценаріїв варіантів використання були побудовані діаграми діяльності для основних варіантів використання, згідно з якими був спроектований проект користувальницького інтерфейсу, а також за допомогою діаграми класів розроблена інформаційна модель (концептуальна модель даних предметної галузі).

2.1.1 Побудова моделі варіантів використання ПЗ інтернет-магазину «Soft_IT»

Сайт орієнтований на клієнтів інтернет-магазину «Soft_IT», їх названо «Користувачем». Особу, що заповнює інформацією сайт, редагує та видаляє, а також несе відповідальність за працездатність сервера та мережеву безпеку, названо «Адміністратором». Вони є акторами системи. Короткий опис акторів наведено в таблиці 2.1

Таблиця 2.1 – Виявлені актори

Актор	Короткий опис
Користувач	Особа, котра продивляється товари, оформлює замовлення та завантажує власні програми на сайт
Адміністратор	Особа, котра редагує інформацію про товар та видаляє товар з сайту

Розроблюваний сайт має певні функції, які в мові UML відображаються у вигляді варіантів використання. Виявлені варіанти використання зведені до таблиці 2.2.

Таблиця 2.2 – Виявлені варіанти використання ПЗ інтернет-магазину «Soft_IT»

Актор	Назва	Короткий опис
1	2	3
Адміністратор, Користувач	Аутентифікація	Даний варіант використання дозволяє користувачеві та адміністратору пройти аутентифікацію на сайті
Адміністратор	Редагувати інформацію про товар	Даний варіант використання дозволяє адміністратору редагувати інформацію про будь-який товар на сайті
Адміністратор	Видалити товар	Даний варіант використання дозволяє адміністратору видаляти будь-який товар з сайту
Користувач	Реєстрація	Даний варіант використання дозволяє користувачеві зареєструватися на сайті

Продовження таблиці 2.2

1	2	3
Користувач	Залишити коментар	Даний варіант використання дозволяє користувачеві залишати коментарі до кожного з товарів
Користувач	Видалити коментар	Даний варіант використання дозволяє користувачеві видаляти власні коментарі
Користувач	Додати товар до корзини	Даний варіант використання дозволяє користувачеві додавати товар до корзини
Користувач	Видалити товар з корзини	Даний варіант використання дозволяє користувачеві видаляти раніше обраний товар із корзини
Користувач	Оформити замовлення	Даний варіант використання дозволяє користувачеві оформити замовлення на основі обраних раніше в корзині товарів
Користувач	Завантажити товар на сайт	Даний варіант використання дозволяє користувачеві завантажувати власні програми для продажу в інтернет-магазині
Користувач	Завантажити програму з сайту	Даний варіант використання дозволяє користувачеві завантажувати комп'ютерні програми, які є безкоштовними

Варіанти використання керують усім процесом розробки. Під час аналізу і проектування варіанти використання дозволяють зрозуміти як результати, які хоче отримати користувач, впливають на архітектуру системи і як повинні себе

вести компоненти системи, для того щоб реалізувати потрібну для користувача функціональність. Діаграма варіантів використання програмного забезпечення інтернет-магазину «Soft_IT» представлена на рисунку 2.1.

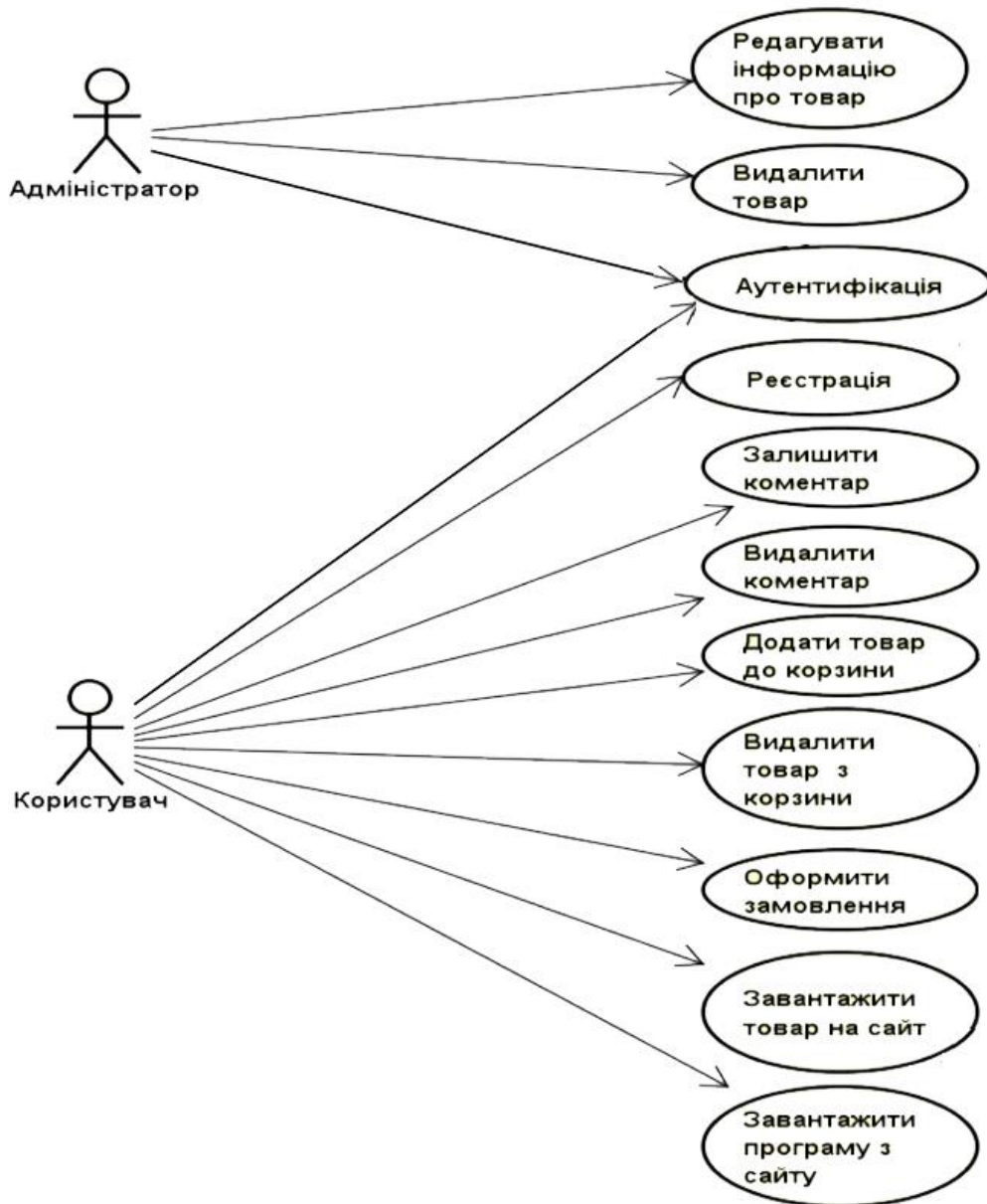


Рисунок 2.1 – Діаграма варіантів використання програмного забезпечення інтернет-магазину «Soft_IT»

2.1.2 Специфікації варіантів використання ПЗ інтернет-магазину «Soft_IT»

Для деталізації варіантів використання, наведених на рисунку 2.1, наведемо їх специфікації.

Специфікація прецеденту «Аутентифікація»

1.0 Найменування прецеденту: «Аутентифікація».

1.1 Короткий опис: Користувач або адміністратор на сторінці реєстрації натискає на кнопку «Ввійти», система перенаправляє його на сторінку авторизації, користувач або адміністратор вводить свої логін та пароль.

2.0 Потоки подій

2.1 Основний потік: Користувач або адміністратор заходить на сайт. Система перенаправляє його на сторінку реєстрації. Користувач або адміністратор натискає на кнопку «Ввійти». Система переадресує його на сторінку авторизації. Користувач або адміністратор вводить свій логін та пароль. Система перевіряє ці дані на збіг з даними у базі даних та аутентифікує користувача.

2.2 Альтернативні потоки: Користувач або адміністратор вводить свої дані. Система не знаходить в базі даних такої комбінації логіну та паролю. Система відображає повідомлення про неправильний логін або пароль.

3.0 Спеціальні вимоги: Не визначені.

4.0 Передумови: Користувач або адміністратор мають попередньо зареєструватися на сайті.

5.0 Післяумови: Якщо користувач чи адміністратор були успішно аутентифіковані, то їх ім'я має відобразитися у спеціально відведеному для цього місці.

Специфікація прецеденту «Видалити товар»

1.0 Найменування прецеденту: «Видалити товар».

1.1 Короткий опис: Адміністратор після аутентифікації потрапляє на сторінку адміністрування. Там він натискає на кнопку «Видалити» навпроти відповідного товару і товар видаляється із сайту.

2.0 Потоки подій

2.1 Основний потік: Адміністратор натискає кнопку «Видалити» навпроти необхідного товару. Система видає повідомлення про те, чи справді він впевнений, що хоче видалити цей товар. Адміністратор підтверджує свій вибір. Система видає вибраний товар.

2.2 Альтернативні потоки: Система після аутентифікації перенаправляє адміністратора на сторінку адміністрування, де відображені усі товари. Адміністратор натискає кнопку «Видалити» навпроти необхідного товару. Система видає повідомлення про те, чи справді він впевнений, що хоче видалити цей товар. Адміністратор натискає на кнопку «Ні». Система не видає вибраний товар.

3.0 Спеціальні вимоги: Не визначені.

4.0 Передумови: Адміністратор заходить на сайт. Адміністратор проходить аутентифікацію.

5.0 Післяумови: Після виконання прецеденту товар видаляється з сайту.

Специфікація прецеденту «Реєстрація»

1.0 Найменування прецеденту: «Реєстрація».

1.1 Короткий опис: Користувач заходить на сайт та вводить свої ім'я, адресу електронної пошти та пароль та реєструється.

2.0 Потоки подій

2.1 Основний потік: Користувач вводить свої дані. Користувач натискає кнопку «Реєструватися». Система зберігає його дані. Система підтверджує реєстрацію.

2.2 Альтернативні потоки: Користувач вводить свої дані. Користувач нажимає кнопку «Реєструватися». Виникає помилка валідації даних, введених користувачем. Система відображає повідомлення про помилку, що виникла.

3.0 Спеціальні вимоги: Не визначені.

4.0 Передумови: Користувач заходить на сайт. Користувач проходить аутентифікацію.

5.0 Післяумови: Якщо користувач був успішно зареєстрований, його ім'я має відобразитися у спеціально відведеному для цього місці.

Специфікація прецеденту «Видалити коментар»

1.0 Найменування прецеденту: «Видалити коментар».

1.1 Короткий опис: Користувач видаляє один із залишених раніше коментарів.

2.0 Потоки подій

2.1 Основний потік: Користувач переходить на сторінку одного з товарів. Користувач натискає кнопку «видалити коментар» навпроти того коментаря, який він хоче видалити. Система видаляє цей коментар. Система відображає користувачеві сторінку товару без його коментарю.

2.2 Альтернативні потоки: не визначені.

3.0 Спеціальні вимоги: Не визначені.

4.0 Передумови: Користувач заходить на сайт. Користувач проходить аутентифікацію.

5.0 Післяумови: відсутні.

Специфікація прецеденту «Залишити коментар»

1.0 Найменування прецеденту: «Залишити коментар».

1.1 Короткий опис: Користувач залишає коментар до одного з товарів.

2.0 Потоки подій

2.1 Основний потік: Користувач переходить на сторінку одного з товарів. Користувач вводить свій коментар у спеціальне поле. Користувач натискає кнопку «Відправити коментар». Система зберігає цей коментар. Система відображає коментар користувача.

2.2 Альтернативні потоки:

3.0 Спеціальні вимоги: Не визначені.

4.0 Передумови: Користувач заходить на сайт. Користувач проходить аутентифікацію.

5.0 Післяумови: Після успішного додавання свого коментаря користувач отримує можливість видалити його.

Специфікація прецеденту «Оформити замовлення»

1.0 Найменування прецеденту: «Оформити замовлення».

1.1 Короткий опис: Користувач у вікні корзини натискає кнопку «Оформити замовлення» і система зберігає інформацію про замовлення.

2.0 Потоки подій

2.1 Основний потік: Користувач відкриває сторінку корзини. Користувач натискає кнопку «Оформити замовлення». Система відображає вікно оформлення замовлення. Клієнт вводить необхідні дані. Система зберігає інформацію про замовлення. Система відображає повідомлення про успішне оформлення замовлення.

2.2 Альтернативні потоки: Користувач відкриває сторінку корзини. Користувач натискає кнопку «Оформити замовлення». Система відображає вікно оформлення замовлення. Клієнт вводить дані. Виникає помилка валідації даних. Система відображає повідомлення про виниклу помилку.

3.0 Спеціальні вимоги: Не визначені.

4.0 Передумови: Користувач заходить на сайт. Користувач проходить аутентифікацію.

5.0 Післяумови: Після виконання цього прецеденту до бази даних має додатися інформація про замовлення.

Специфікація прецеденту «Завантажити товар на сайт»

1.0 Найменування прецеденту: «Завантажити товар на сайт».

1.1 Короткий опис: Користувач переходить на сторінку завантаження на сайт і там заповнює форму для відправки товару на сайт.

2.0 Потоки подій

2.1 Основний потік: Користувач переходить на сторінку завантаження на сайт. Користувач заповнює необхідну форму і натискає кнопку «Зберегти». Система зберігає інформацію про товар. Система відображає повідомлення про

успішне додавання товару. Система перенаправляє користувача на головну сторінку.

2.2 Альтернативні потоки: Користувач вводить свої данні. Виникає помилка валідації даних, введених користувачем. Система відображає повідомлення про виниклу помилку.

3.0 Спеціальні вимоги: Не визначені.

4.0 Передумови: Користувач заходить на сайт. Користувач проходить аутентифікацію.

5.0 Післяумови: Доданий таким чином товар буде знаходитися у загальному доступі.

Специфікація прецеденту «Додати товар до корзини»

1.0 Найменування прецеденту: «Додати товар до корзини».

1.1 Короткий опис: Користувач натискає на сторінці одного з товарів кнопку «Додати товар до корзини» і цей товар додається до корзини.

2.0 Потоки подій

2.1 Основний потік: Користувач проходить аутентифікацію. Користувач переходить на сторінку одного з товарів. Користувач натискає кнопку «Додати до корзини». Система додає цей товар до корзини. Система відображає повідомлення про успішне додавання товару до корзини.

2.2 Альтернативні потоки: Відсутні.

3.0 Спеціальні вимоги: Не визначені.

4.0 Передумови: Користувач заходить на сайт. Користувач проходить аутентифікацію.

5.0 Після умови: Додані до корзини товари зберігаються там до завершення сеансу або до того часу, як користувач їх видалить.

Специфікація прецеденту «Видалити товар з корзини»

1.0 Найменування прецеденту: «Видалити товар з корзини».

1.1 Короткий опис: Користувач у вікні корзини натискає кнопку «Видалити» і цей товар видаляється з корзини.

2.0 Потоки подій

2.1 Основний потік: Користувач відкриває сторінку корзини. Користувач натискає кнопку «Видалити» навпроти відповідного товару. Система видаляє цей товар із корзини. Система відображає сторінку корзини без видаленого товару.

2.2 Альтернативні потоки: Відсутні.

3.0 Спеціальні вимоги: Не визначені.

4.0 Передумови: Користувач заходить на сайт. Користувач проходить аутентифікацію.

5.0 Після умови: Не визначені.

Специфікація прецеденту «Завантажити програму з сайту»

1.0 Найменування прецеденту: «Завантажити програму з сайту».

1.1 Короткий опис: Користувач обирає безкоштовну програму, система відправляє йому файл з цією програмою.

2.0 Потоки подій

2.1 Основний потік: Користувач обирає необхідну безкоштовну програму. Користувач натискає кнопку «Завантажити». Система відправляє користувачеві необхідний файл.

2.2 Альтернативні потоки: Відсутні.

3.0 Спеціальні вимоги: Не визначені.

4.0 Передумови: Користувач заходить на сайт. Користувач проходить аутентифікацію.

5.0 Після умови: Не визначені.

Специфікація прецеденту «Редагувати інформацію про товар»

1.0 Найменування прецеденту: «Редагувати інформацію про товар».

1.1 Короткий опис: Адміністратор на сторінці адміністрування натискає на кнопку «Редагувати» навпроти відповідного товару і у вікні, яке відкриється, вводить відповідні зміни і натискає кнопку зберегти.

2.0 Потоки подій

2.1 Основний потік: Система після аутентифікації перенаправляє адміністратора на сторінку адміністрування, де відображені усі товари.

Адміністратор натискає кнопку «Редагувати» навпроти необхідного товару. Система перенаправляє адміністратора на сторінку редагування товару. Адміністратор вводить необхідні зміни і натискає кнопку «Зберегти». Система зберігає внесені зміни.

2.2 Альтернативні потоки: Система після аутентифікації перенаправляє адміністратора на сторінку адміністрування, де відображені усі товари. Адміністратор натискає кнопку «Редагувати» навпроти необхідного товару. Система перенаправляє адміністратора на сторінку редагування товару. Адміністратор вводить зміни і натискає кнопку «Відміна». Система не зберігає внесені зміни. Система перенаправляє користувача на сторінку Адміністрування.

3.0 Спеціальні вимоги: Не визначені.

4.0 Передумови: Адміністратор заходить на сайт. Адміністратор проходить аутентифікацію.

5.0 Післяумови: Після виконання прецеденту до вибраного товару вносяться відповідні зміни.

2.1.3 Сценарії основних варіантів використання ПЗ інтернет-магазину «Soft_IT» (діаграми діяльності)

Діаграма діяльності задає поведінку системи у вигляді певних робіт, які може виконувати система або актор і ці роботи можуть залежати від заданих умов або від обмежень. Ця діаграма відображає функціональну структуру системи і принципи поведінки її окремих елементів під час виконання відповідної діяльності.

Діаграма діяльності для варіанту використання «Видалити товар із кошика» представлена на рисунку 2.2, діаграма діяльності для варіанту використання «Залишити коментар» представлена на рисунку 2.3.



Рисунок 2.2 – Діаграма діяльності для варіанту використання «Видалити товар із корзини»



Рисунок 2.3 – Діаграма діяльності для варіанту використання «Залишити коментар»

Діаграма діяльності для варіанту використання «Реєстрація» представлена на рисунку 2.4.

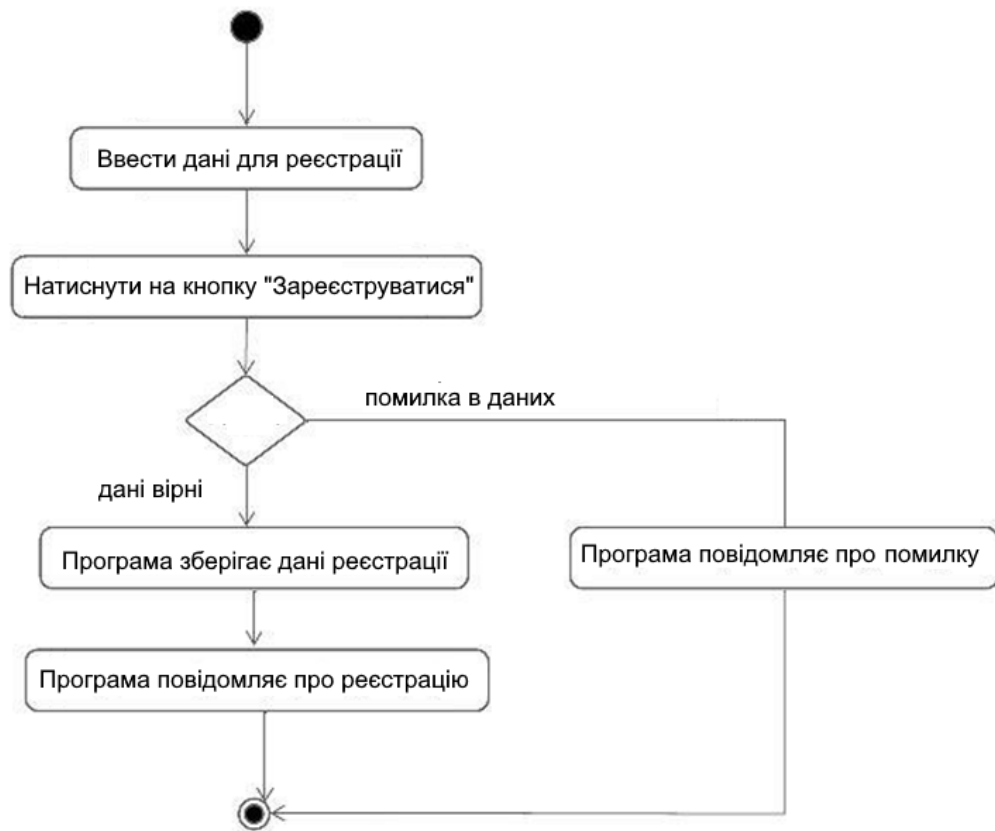


Рисунок 2.4 – Діаграма діяльності для варіанту використання «Реєстрація»

2.1.4 Концептуальна модель даних ПЗ інтернет-магазину «Soft_IT»

Розробимо узагальнену інформаційну модель для ПЗ інтернет-магазину «Soft_IT». З цією метою створимо концептуальну модель даних. Концептуальна модель розроблялася за методологією IDEF1X [4].

На рисунку 2.5 зображено концептуальну модель даних.

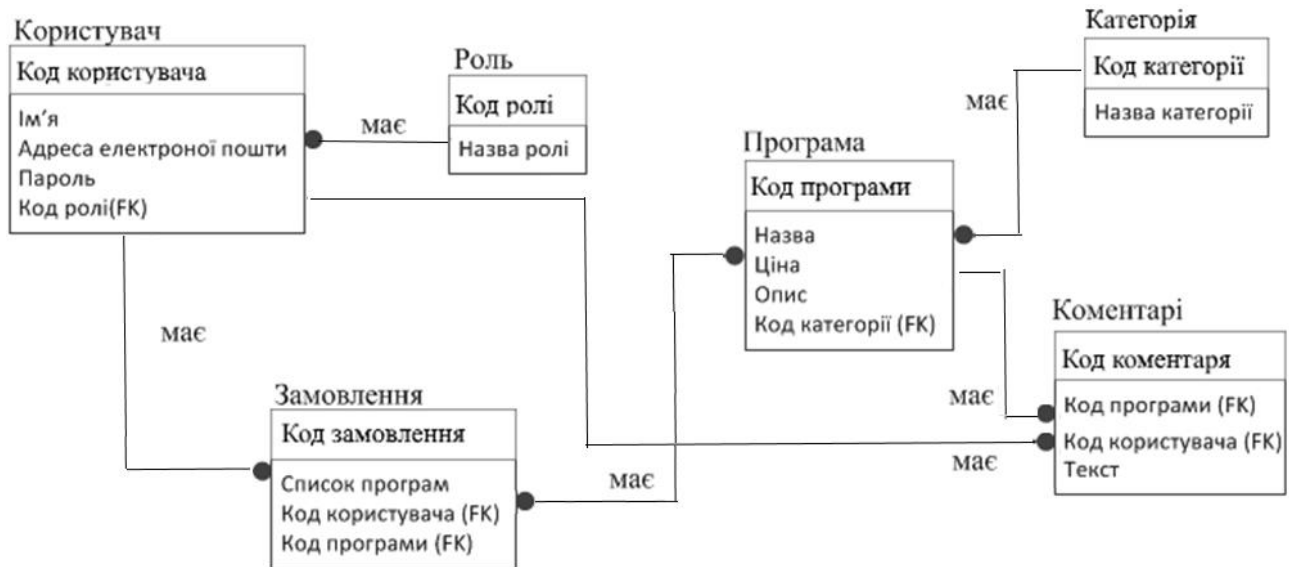


Рисунок 2.5 – Концептуальна модель даних ПЗ інтернет-магазину «Soft_IT»

2.1.5 Розробка інтерфейсу користувача ПЗ інтернет-магазину «Soft_IT»

Важливим етапом при проектуванні програмного забезпечення є розробка інтерфейсу. Беручи до уваги вимоги, що зазначені постановці задачі, визначимо зовнішній інтерфейс. Для забезпечення зручної навігації необхідно розробити інтуїтивно-зрозумілий інтерфейс користувача (клієнта):

На рисунку 2.6 представлено ескіз головного вікна програми.



Рисунок 2.6 – Ескіз головного вікна програми

На рисунку 2.7 Представлено ескіз сторінки товару.

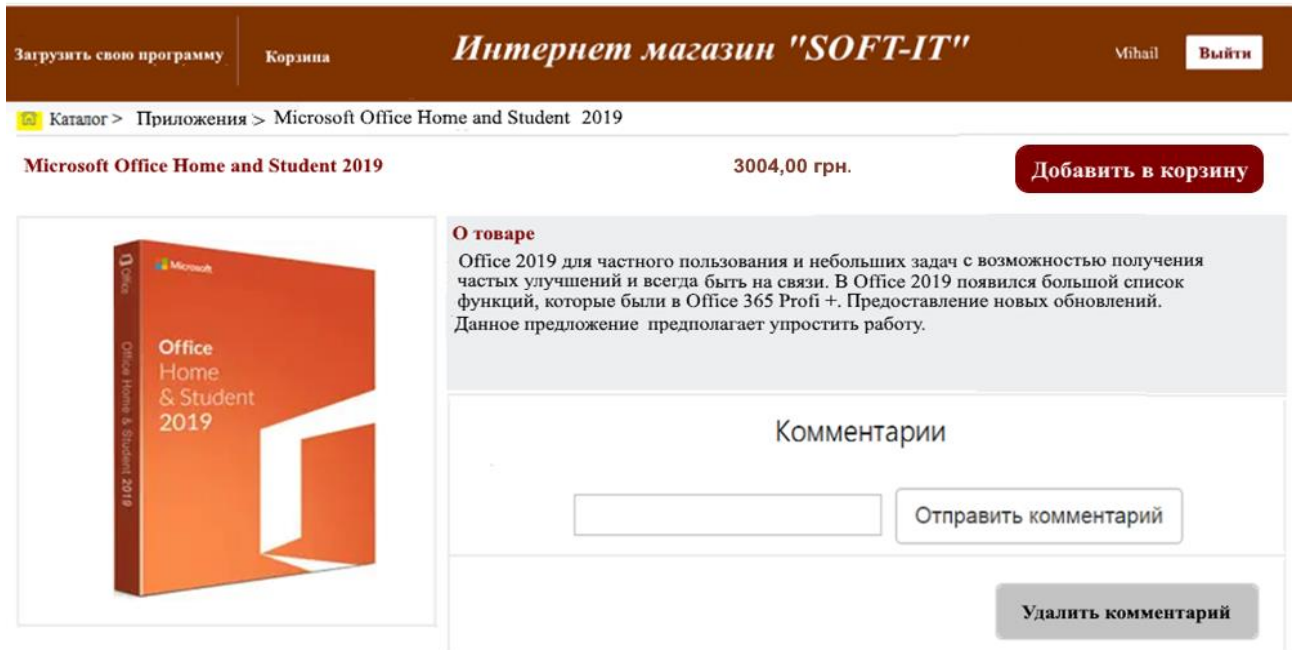


Рисунок 2.7 – Ескіз сторінки товару

На рисунку 2.8 представлено ескіз сторінки корзины.

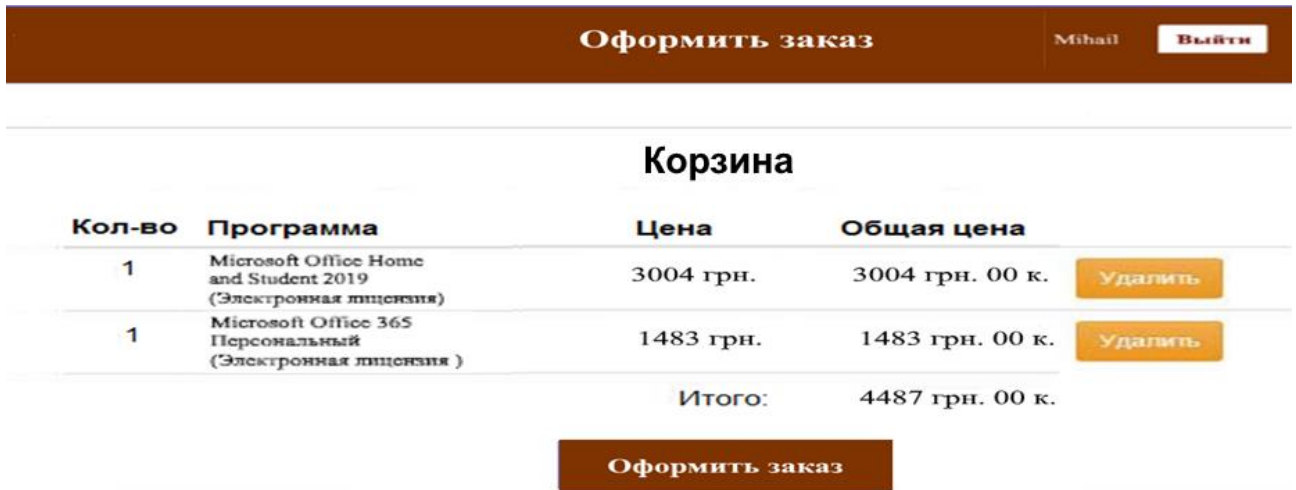


Рисунок 2.8 – Ескіз сторінки корзины

На рисунку 2.9 представлено ескіз панелі адміністрування

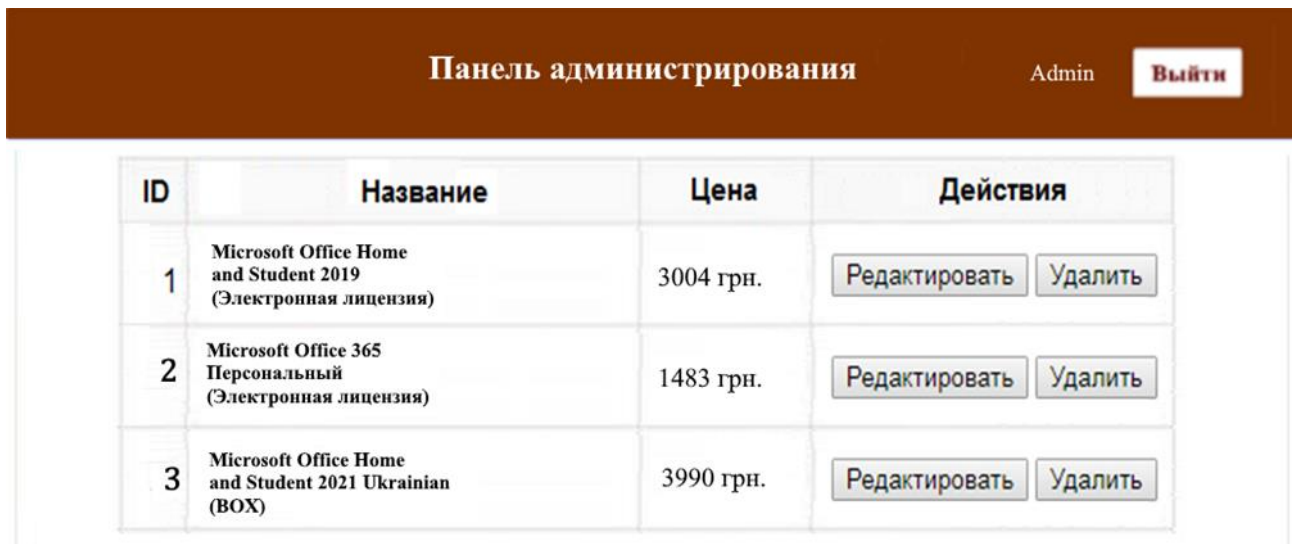


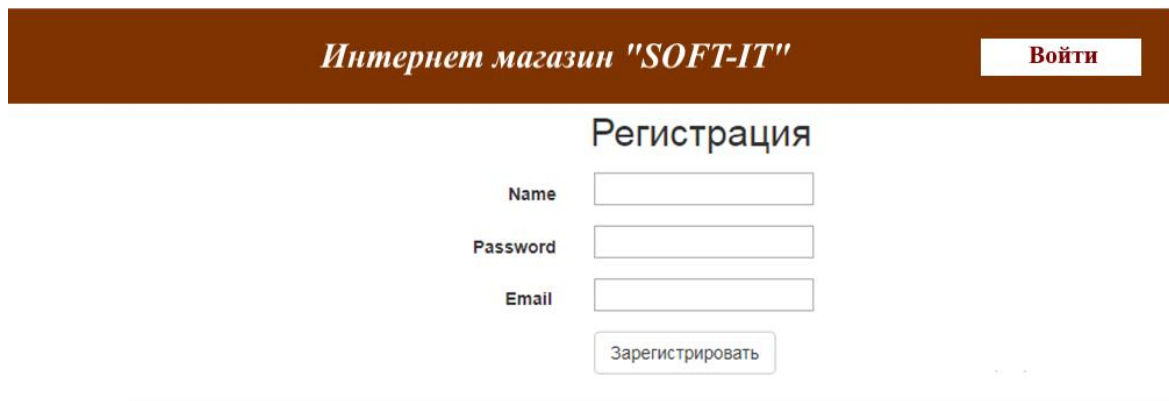
Рисунок 2.9 – Ескіз панелі адміністрування

На рисунку 2.10 представлено ескіз сторінки оформлення замовлення

The screenshot shows a web interface for an internet store named 'Интернет магазин "SOFT-IT"'. The header is dark brown with the store name in white. On the left, there are links 'Загрузить свою программу' (Download your program) and 'Корзина' (Cart). On the right, the name 'Mihail' is displayed, and next to it is a white button with the text 'Выйти' (Logout). Below the header is a form for placing an order. The form has a title 'Оформить заказ сейчас' (Place order now) and a subtitle 'Пожалуйста введите ваши контактные данные, и мы сразу отправим товар!' (Please enter your contact information, and we will immediately ship the goods!). The form is divided into sections: 'Данные' (Data) with a field for 'Ваше имя:' (Your name:); 'Адрес доставки' (Delivery address) with fields for 'Первый адрес:' (First address:), 'Второй адрес:' (Second address:), 'Третий адрес:' (Third address:), 'Город:' (City:), and 'Страна:' (Country:); and 'Опции' (Options) with a checkbox 'Использовать подарочную упаковку?' (Use gift wrapping?). At the bottom of the form is a dark brown button with the text 'Обработать заказ' (Process order).

Рисунок 2.10 – Ескіз сторінки оформлення замовлення

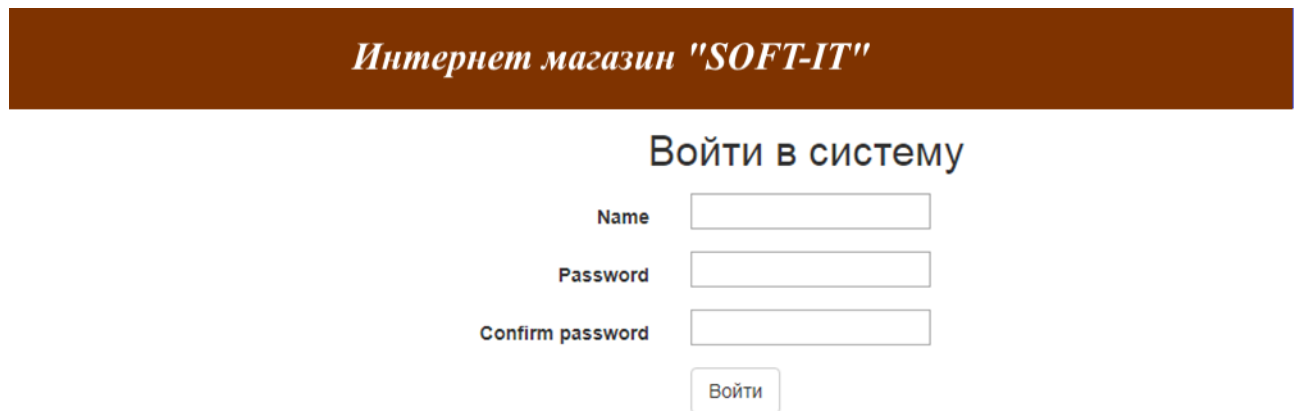
На рисунку 2.11 представлено ескіз сторінки реєстрації



The image shows a registration form for an online store. At the top, there is a dark brown header bar with the text "Интернет магазин 'SOFT-IT'" in white italicized font on the left, and a white button with the text "Войти" in dark brown font on the right. Below the header, the title "Регистрация" is centered. The form consists of three input fields labeled "Name", "Password", and "Email" stacked vertically. Below these fields is a button labeled "Зарегистрировать".

Рисунок 2.11 – Ескіз сторінки реєстрації

На рисунку 2.12 представлено ескіз сторінки аутентифікації



The image shows a login form for the same online store. It has a dark brown header bar with the text "Интернет магазин 'SOFT-IT'" in white italicized font. Below the header, the title "Войти в систему" is centered. The form consists of three input fields labeled "Name", "Password", and "Confirm password" stacked vertically. Below these fields is a button labeled "Войти".

Рисунок 2.12 – Ескіз сторінки аутентифікації

2.2 Технічний проект програмного забезпечення інтернет-магазину «Soft_IT»

Проаналізувавши ескізний проект, був розроблений технічний проект даного ПЗ шляхом побудови статичної і динамічної моделей та логічної моделі даних. Статична модель представлена діаграмою класів, а динамічна – діаграмами послідовності. Логічна модель даних програмного забезпечення представлена таблицями [5].

2.2.1 Статична модель програмного забезпечення інтернет-магазину

На рисунку 2.13 зображено діаграму класів програмного забезпечення

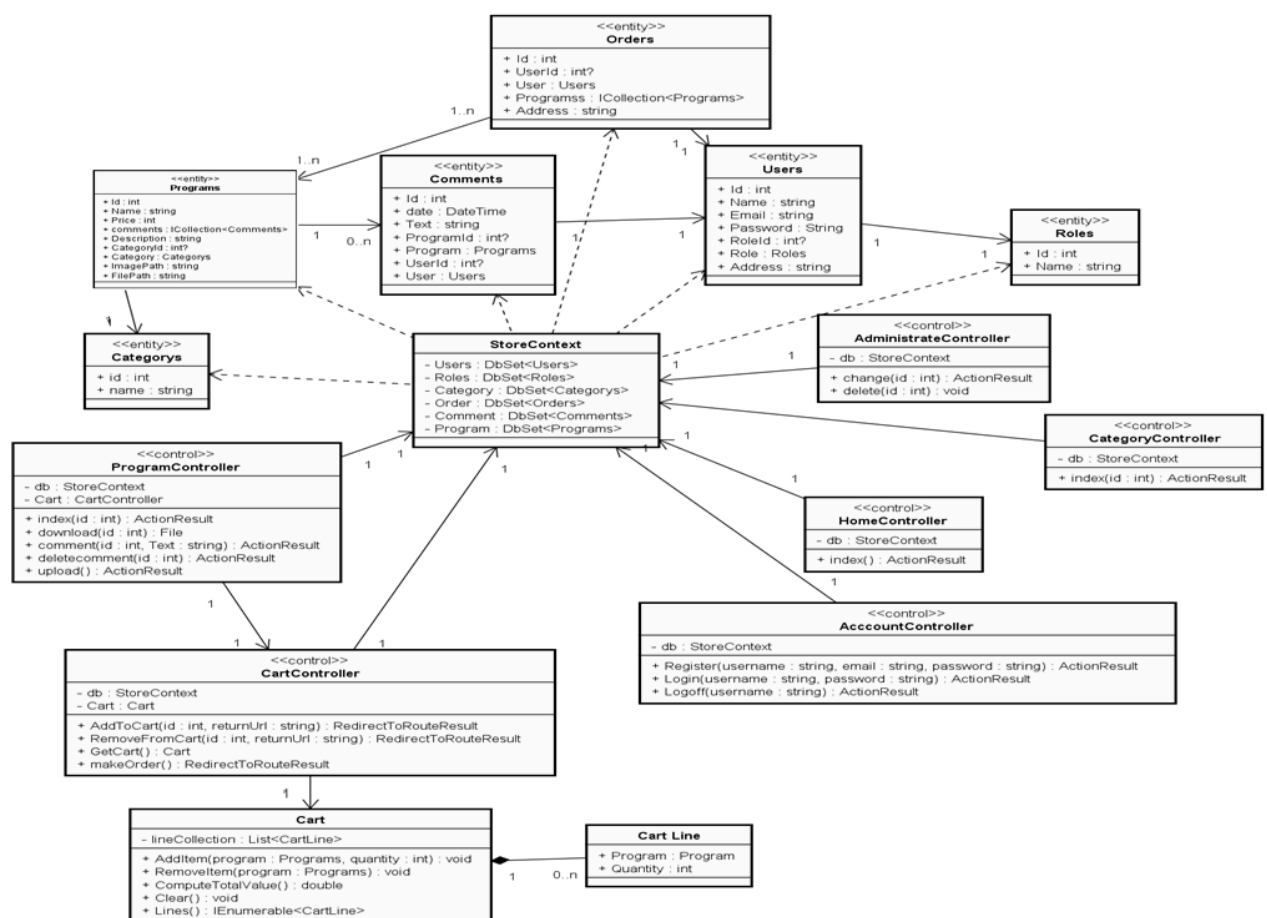


Рисунок 2.13 – Діаграма класів програмного забезпечення

2.2.2 Специфікації класів ПЗ інтернет-магазину «Soft_IT»

Клас Orders

Клас, що призначений для обробки даних, які передаються з таблиці бази даних Orders до інтерфейсу користувача.

Включає в себе наступні атрибути:

Id – призначений для зберігання id даного замовлення

UserId – призначений для зберігання id пов'язаного з цим замовленням клієнта

User – призначений для зберігання екземпляру класу пов'язаного з цим замовленням користувача

Programs – призначений для зберігання колекції екземплярів класу пов'язаних з цим замовленням програм

Address – призначений для зберігання адреси даного замовлення

Клас Categorys

Клас, що призначений для обробки даних, які передаються з таблиці бази даних Categorys до інтерфейсу користувача.

Включає в себе наступні атрибути:

Id – призначений для зберігання id даної категорії

Name – призначений для зберігання імені даної категорії

Клас StoreContext

Клас, що призначений для отримання інформації з бази даних за допомогою Entity Framework.

Включає в себе наступні атрибути:

Users – призначений для доступу до усіх користувачів із бази даних

Roles – призначений для доступу до усіх ролей із бази даних

Category – призначений для доступу до усіх категорій із бази даних

Order – призначений для доступу до усіх замовлень із бази даних

Comment – призначений для доступу до усіх коментарів із бази даних

Program – призначений для доступу до усіх програм із бази даних

Клас ProgramController

Клас, що реалізує різноманітні методи для роботи з товаром.

Включає в себе наступні атрибути:

db – екземпляр класу StoreContext

Cart – екземпляр класу CartController

Включає в себе такі методи:

add() – метод, який призначений для додавання нового товару до бази даних;

delete() – метод, який призначений для видалення товару до бази даних;

index() – метод, який призначений для переходу на сторінку з необхідним товаром;

change() – метод для зміни інформації про товар;

download() – метод для завантаження безкоштовних програм;

upload() – метод для завантаження власних програм на сайт;

comment() – метод, який призначений для коментування товарів;

deleteComment() – метод, який призначений для видалення своїх коментарів.

Клас AccountController

Реалізує методи для аутентифікації та реєстрації користувачів сайту.

Включає в себе наступні атрибути:

db – екземпляр класу StoreContext

Включає в себе такі методи:

Login() – метод для аутентифікації;

Logoff() – метод для виходу зі свого облікового запису;

Register() – метод для реєстрації користувачів;

Клас HomeController

Реалізує методи для головної сторінки сайту.

Включає в себе наступні атрибути:

db – екземпляр класу StoreContext

Включає в себе такі методи:

`index()` – метод для переходу на головну сторінку;

`choose_program()` – метод для переходу на сторінку конкретної програми;

`choose_category()` – метод переходу на сторінку, де відображаються програми даної категорії;

`administrate()` – перехід на сторінку адміністрування;

Клас CartController

Клас призначений для виконання операцій з корзиною.

Включає в себе наступні атрибути:

`db` – екземпляр класу `StoreContext`

`Cart` – екземпляр класу `Cart`

Включає в себе такі методи:

`AddToCart()` – метод для додавання товару до корзини;

`RemoveFromCart()` – метод для видалення товару з корзини;

`GetCart()` – метод для отримання корзини;

`MakeOrder()` – метод для оформлення замовлення.

Клас Roles

Клас призначений для обробки даних, які передаються з таблиці бази даних `Roles` до інтерфейсу користувача.

Включає в себе наступні атрибути:

`Id` – призначений для зберігання `id` даної ролі

`Name` – призначений для зберігання імені даної ролі

Клас Comments

Клас призначений для обробки даних, які передаються з таблиці бази даних `Comments` до інтерфейсу користувача.

Включає в себе наступні атрибути:

`Id` – призначений для зберігання `id` даної ролі

`date` – призначений для зберігання дати коментарю

`Text` – призначений для зберігання тексту коментарю

`ProgramId` – призначений для зберігання `id` пов'язаної з цим коментарем програми

Program – призначений для зберігання екземпляру класу пов’язаної з цим коментарем програми

UserId – призначений для зберігання id пов’язаного з цим коментарем користувача

User – призначений для зберігання класу пов’язаного з цим коментарем користувача

Клас Programs

Клас призначений для обробки даних, які передаються з таблиці бази даних Programs до інтерфейсу користувача.

Включає в себе наступні атрибути:

Id – призначений для зберігання id даної програми

Name – призначений для зберігання імені даної програми

Price – призначений для зберігання ціни даної програми

Comments – призначений для зберігання колекції екземплярів класу пов’язаних з цим коментарем програм

Description – призначений для зберігання опису даної програми

CategoryId – призначений для зберігання id пов’язаної з цією програмою категорії

Category – призначений для зберігання екземпляру класу пов’язаної з цією програмою категорії

FilePath – призначений для зберігання шляху до даної програми

ImagePath – призначений для зберігання шляху до картинки пов’язаної з даною програмою

Клас Users

Клас призначений для обробки даних, які передаються з таблиці бази даних Users до інтерфейсу користувача.

Включає в себе наступні атрибути:

Id – призначений для зберігання id даного користувача

Name – призначений для зберігання імені даного користувача

Password – призначений для зберігання паролю даного користувача

Email – призначений для зберігання електронної адреси даного користувача

RoleId – призначений для зберігання id пов'язаної з цим користувачем ролі

Role – призначений для зберігання екземпляру класу пов'язаної з цим користувачем ролі

Клас Cart

Клас який виконує роль корзини.

Включає в себе наступні атрибути:

linecollection – колекція екземплярів класу CartLine

Включає в себе такі методи:

AddItem() – додати товар до корзини;

RemoveItem() – метод для видалення товару з корзини;

ComputeTotalValue() – метод для підрахунку необхідної до оплати суми;

Clear() – очистити корзину;

Lines() – отримати усі замовлення.

Клас CartLine

Клас який відображає запис у корзині.

Включає в себе наступні атрибути:

Program – призначений для зберігання інформації про програму

Quantity – призначений для зберігання кількості програм у замовленні

2.2.3 Динамічна модель ПЗ інтернет-магазину «Soft_IT»

На основі моделі сценаріїв варіантів використання і статичної моделі програми була створена динамічна модель ПЗ. На рисунках 2.14 – 2.18 наведені діаграми послідовності основних варіантів використання, які реалізують цю модель. На цих діаграмах зображена взаємодія користувача з системою у часі.

Діаграма послідовності для прецеденту «Додати товар до корзини» представлена на рисунку 2.14.

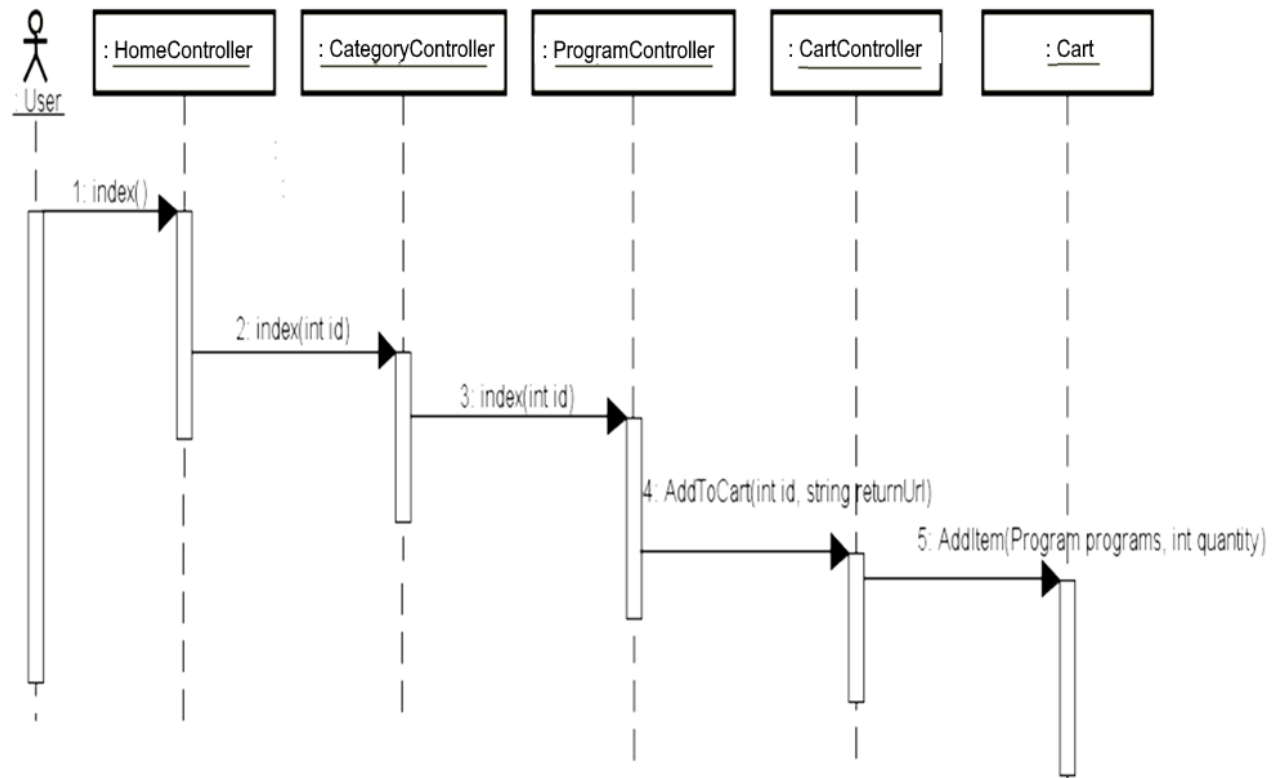


Рисунок 2.14 – Діаграма послідовності для прецеденту «Додати товар до корзини»

Діаграма послідовності для прецеденту «Оформити замовлення» представлена на рисунку 2.15.

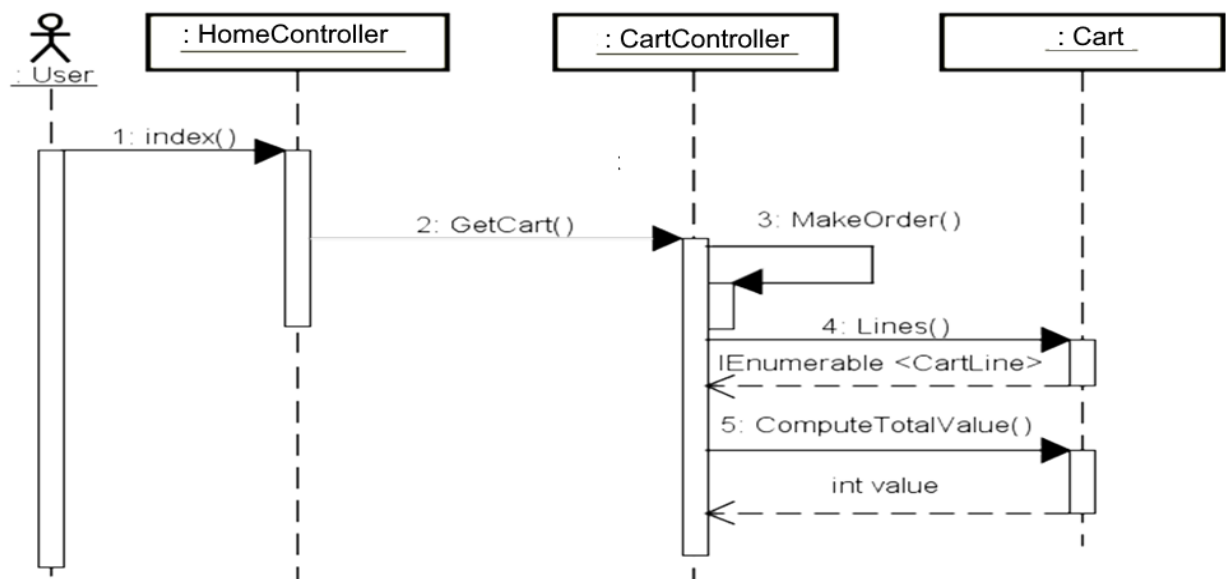


Рисунок 2.15 – Діаграма послідовності для прецеденту «Оформити замовлення»

Діаграма послідовності для прецеденту «Залишити коментар» представлена на рисунку 2.16.

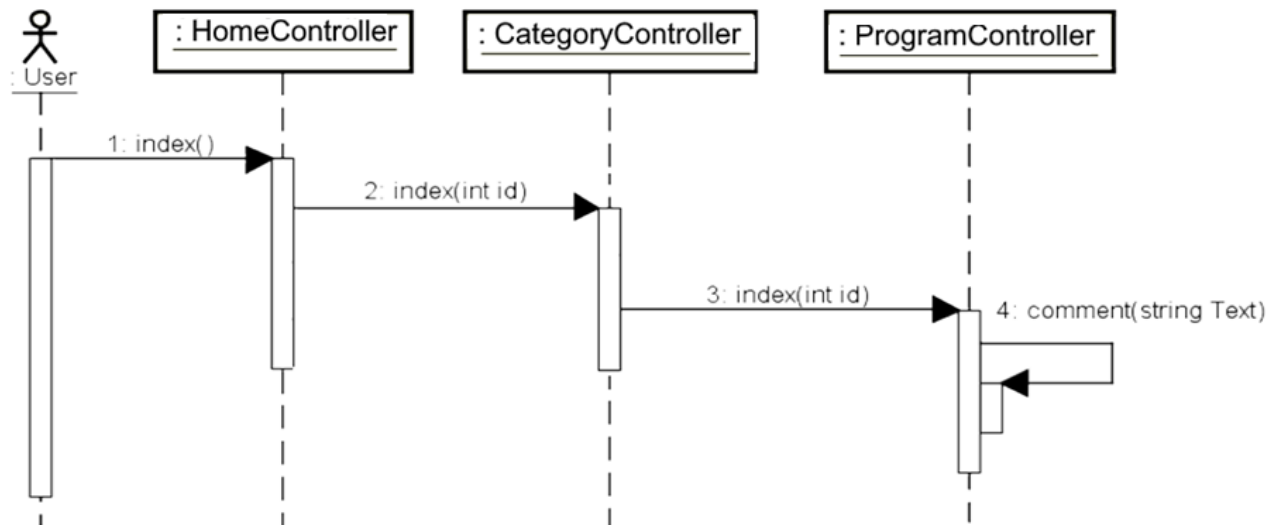


Рисунок 2.16 – Діаграма послідовності для прецеденту «Залишити коментар»

Діаграма послідовності для прецеденту «Редагувати інформацію про товар» представлена на рисунку 2.17.

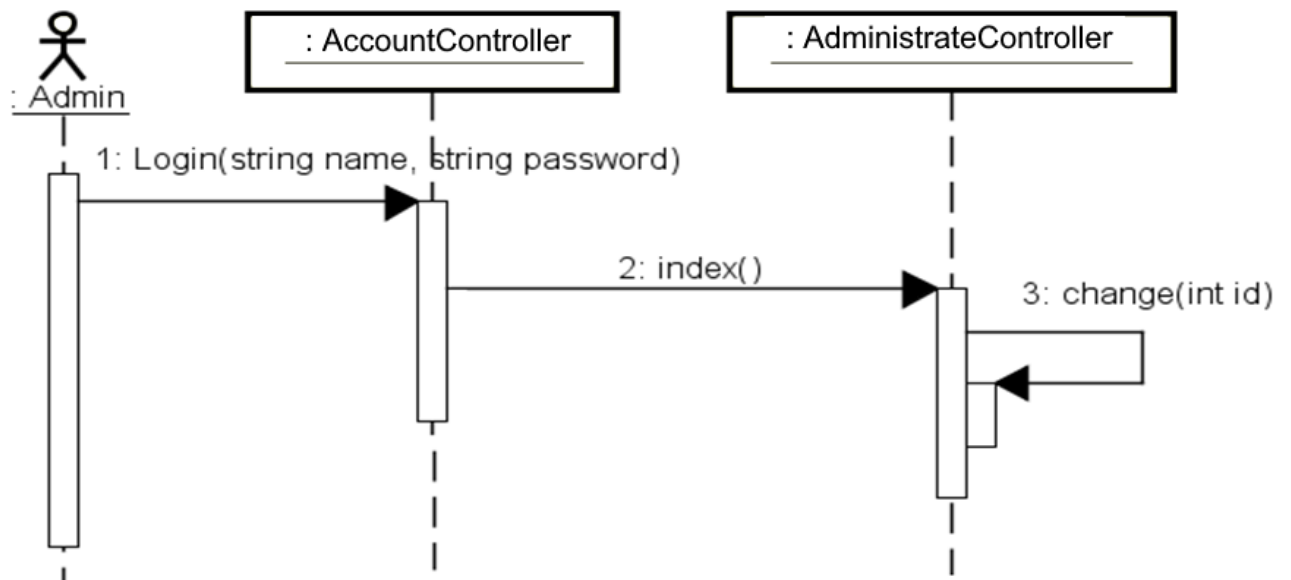


Рисунок 2.17 – Діаграма послідовності для прецеденту «Редагувати інформацію про товар»

Діаграма послідовності для прецеденту «Загрузити програму з сайту» представлена на рисунку 2.18.

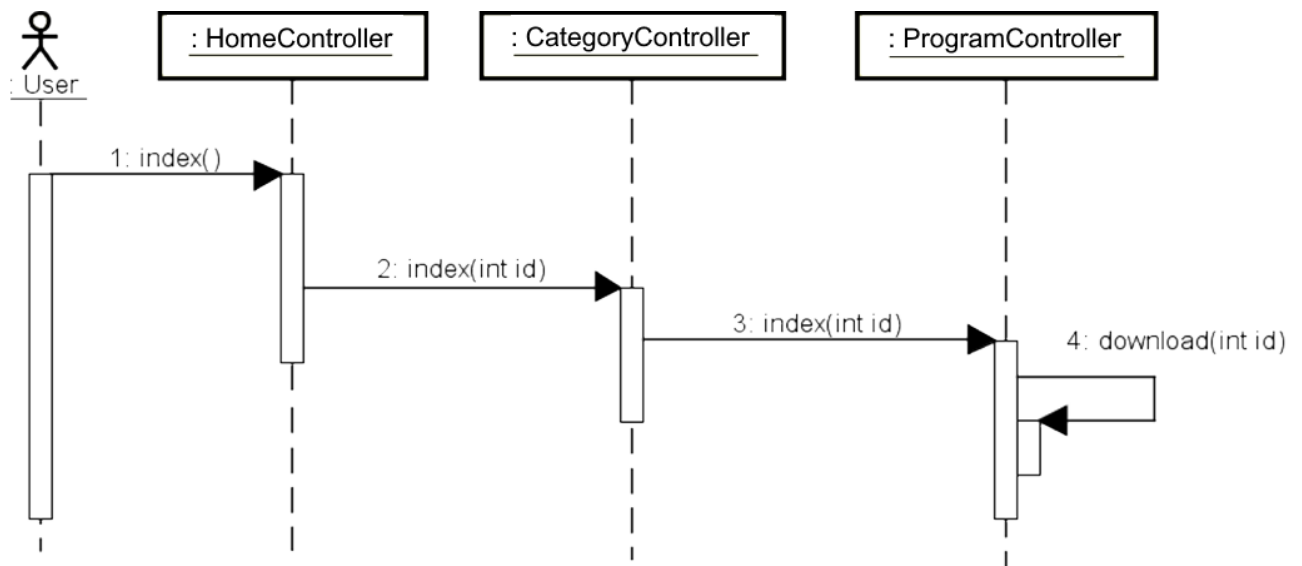


Рисунок 2.18 – Діаграма послідовності для прецеденту «Загрузити програму з сайту»

2.2.4 Логічна модель даних ПЗ інтернет-магазину «Soft_IT»

На основі концептуальної моделі було розроблено логічну модель бази даних програмного забезпечення інтернет-магазину «Soft_IT». У кожному таблицю було вписано атрибути, які будуть характерні для кожної сутності. Кожен атрибут має свій тип, який визначається типом даних, які зберігаються у базі даних. Також було вказано ключові атрибути. Атрибути, які організують зв'язок між таблицями, є зовнішніми ключами (FK) [4].

Логічна модель даних ПЗ інтернет-магазину «Soft_IT» представлена на рисунку 2.19.

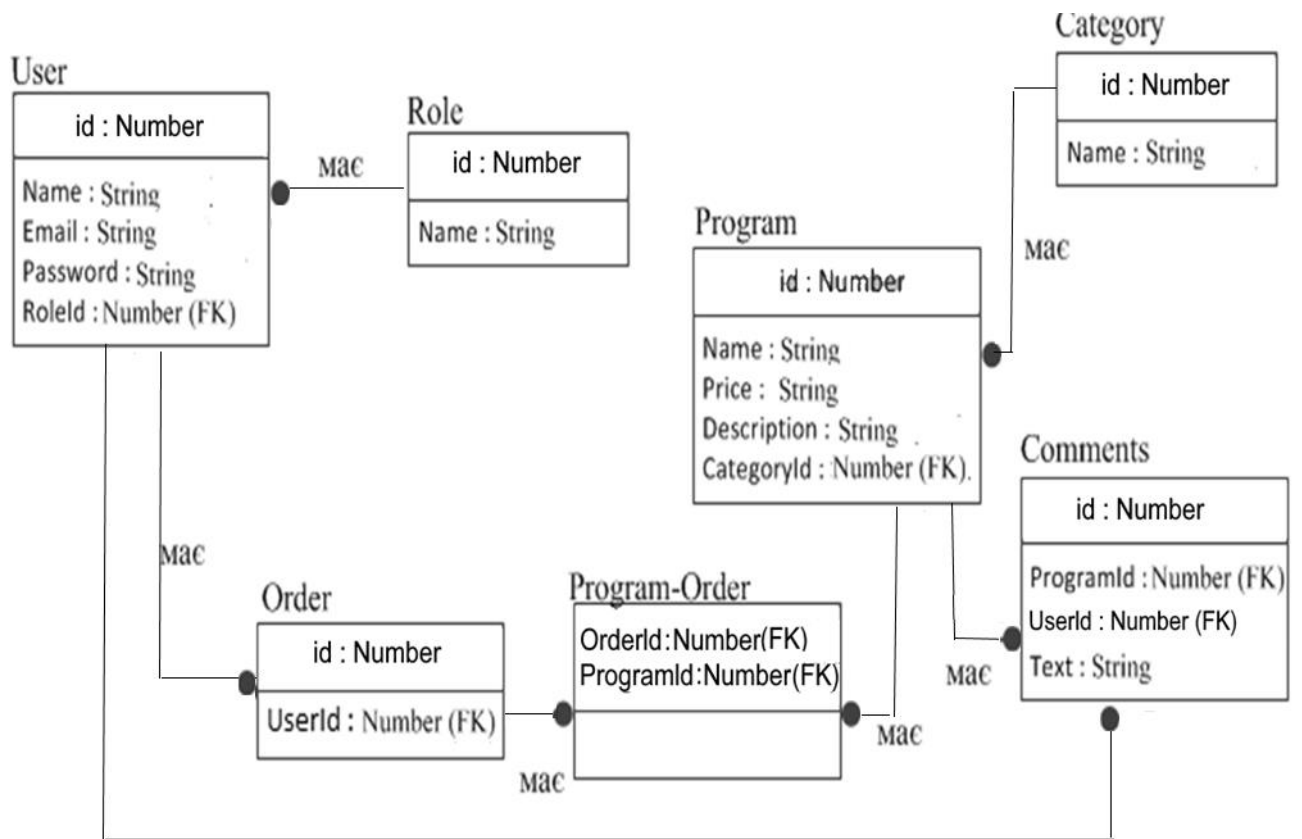


Рисунок 2.19 – Логічна модель бази даних

2.3 Робочий проект програмного забезпечення інтернет-магазину «Soft_IT»

На стадії робочого проекту рішення, прийняті на попередніх етапах, перетворюються у форму, доступну користувачеві [5]. Вибираються засоби розробки програмного забезпечення, створюються діаграма компонентів та діаграма розміщення, виконуються реалізація, тестування та випробування програмного забезпечення.

2.3.1 Обґрунтування вибору мови програмування та засобів розробки програмного забезпечення

Дане програмне забезпечення спроектоване для розробки на об'єктно-орієнтованій мові програмування. До найбільш потужних об'єктно-орієнтованих мов відносяться C# та Java. Так як для Java потрібно окремо встановлювати Java Runtime Environment, а компоненти мови C# одразу інсталиються з ОС Window, яка є найпопулярнішою ОС, то стає очевидним вибір C# в якості мови програмування програмного забезпечення програмного забезпечення інтернет-магазину «Soft_IT».

C# – дана мова програмування є об'єктно-орієнтованою мовою та дозволяє максимально зекономити час при розробці програмного продукту [2]. Швидкодія програм, написаних на C# досить висока. Перевагою даної мови є те, що розмір результуючих програмних продуктів є дуже малий і не потребує додаткових бібліотек. Для виконання програм, написаних на мові C#, необхідно щоб на комп'ютері був встановлений Framework 4.5. Також Перевагою C# програм є виконання на різних платформах, для яких розроблений Net Framework. Програма написана на C# не залежить ні від апаратури комп'ютера, ні від платформи.

2.3.2 Побудова діаграми компонентів ПЗ інтернет-магазину «Soft_IT»

Діаграма компонентів – статична структурна діаграма, яка показує розбиття програмної системи на структурні компоненти та зв'язки (залежності) між компонентами. В якості фізичних компонентів можуть виступати файли, бібліотеки, модулі, виконувані файли, пакети.

На рисунку 2.20 представлено діаграму компонентів ПЗ інтернет-магазину «Soft_IT».

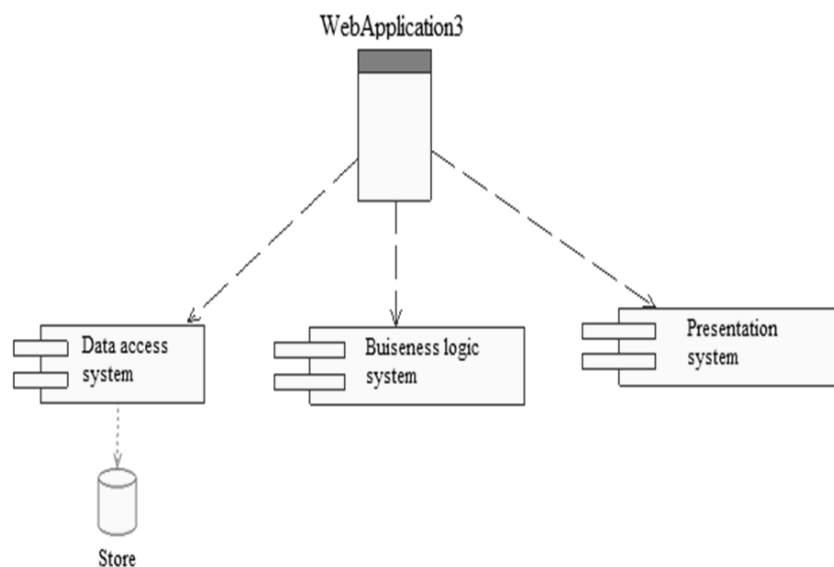


Рисунок 2.20 – Діаграма компонентів ПЗ інтернет-магазину «Soft_IT».

2.3.3 Побудова діаграми розміщення ПЗ інтернет-магазину «Soft_IT»

Діаграма розміщення задає склад ресурсів системи, на яких розміщуються компоненти, і відношення між ними.

На рисунку 2.21 представлено діаграму розміщення ПЗ інтернет-магазину «Soft_IT».

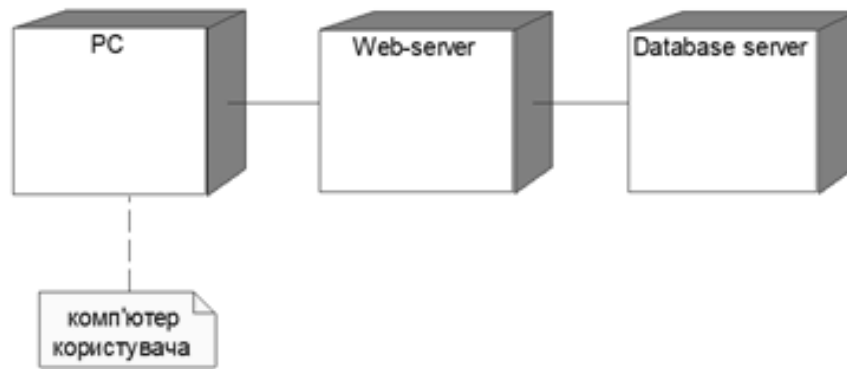


Рисунок 2.21 – Діаграма розміщення ПЗ інтернет-магазину «Soft_IT»

2.3.4 Реалізація та тестування основних модулів програмного забезпечення інтернет-магазину «Soft_IT»

При тестуванні об'єктно-орієнтованого ПЗ найменшим тестованим елементом є клас.

Для спрощення розробки тестів використовуються елементи Use Case, що є частиною моделі вимог. Кожен елемент Use Case задає сценарій, який дозволяє виявляти помилки при взаємодії користувача з системою. Для підтвердження правильності може приводитися звичайне тестування «чорного ящика».

Реалізація основних модулів програми представлена в Додатку Д.

Тести «Чорної скриньки»

Тест 1 – Перевірити можливість аутентифікації

Сценарій:

1. Користувач на сторінці аутентифікації вводить правильні логін та пароль.
2. Програма перенаправляє користувача на головну сторінку і замість кнопки «Ввійти» відображає ім'я користувача.

Після натиснення кнопки «Ввійти» користувач перенаправляється на головну сторінку і його ім'я відображається у спеціально відведеному місці, що представлено на рисунку 2.22.



Рисунок 2.22 – Відображення імені користувача на головній сторінці

Тест 2 – Перевірити неможливість аутентифікації за неіснуючою комбінацією логіна та пароллю

Сценарій:

1. Користувач на сторінці аутентифікації вводить неправильні логін та пароль.

2. Програма виводить повідомлення про те, що така комбінація логіну та пароллю не існує.

Після натиснення кнопки «Ввійти» відображається, повідомлення про неправильну комбінацію логіну та пароллю, що представлено на рисунку 2.23

 A login form titled 'Ввойти в систему' in bold. It contains three input fields labeled 'Name', 'Email', and 'Password'. Below the 'Password' field is a button labeled 'Войти'. To the right of the button, there is a red error message: 'Неправильная комбинация логина и пароля'.

Рисунок 2.23 – Відображення повідомлення про неправильну комбінацію логіну та пароллю

Тест 3 – Перевірити можливість додати новий товар до корзини

Сценарій:

1. Користувач авторизується.
2. Користувач на сторінці товару натискає кнопку «Додати до корзини».
3. Програма додає товар до корзини і виводить повідомлення про успішне додання товару до корзини.

Після натиснення кнопки «Додати до корзини» виводиться повідомлення, яке представлене на рисунку 2.24.

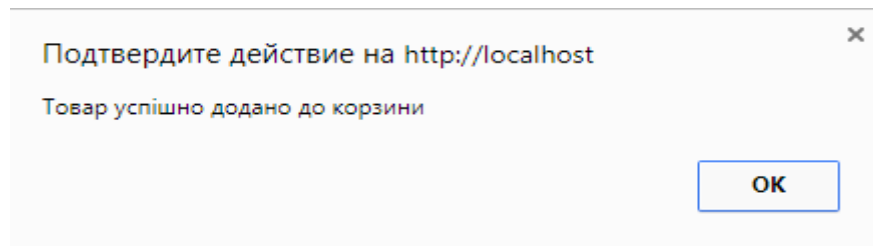


Рисунок 2.24 – Повідомлення про додавання товару до корзини

Тест 4 – Перевірити можливість видалити товар із корзини

Сценарій:

1. Користувач авторизується.
2. Користувач на сторінці корзини навпроти необхідного товару нажимає кнопку «Видалити».
3. Програма видаляє товар з корзини і виводить повідомлення про успішне видалення товару до корзини.

Після натискання кнопки «Видалити» навпроти відповідного товару у корзині виводиться повідомлення представлене на рисунку 2.25.

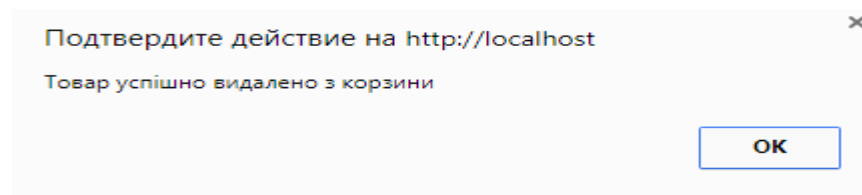


Рисунок 2.25 – Повідомлення про видалення з корзини

Тест 5 – Перевірити неможливість оформити замовлення при помилці валідації

Сценарій:

1. Користувач авторизується.
2. Користувач переходить на сторінку оформлення замовлення.

3. Користувач некоректно заповнює поля і натискає кнопку «Оформити замовлення».

4. Програма не зберігає замовлення і виводить повідомлення про те, які помилки були допущені при заповненні.

Після натискання кнопки «Оформити замовлення» виводиться інформація про те, які саме поля були заповнені невірно, результат представлений на рисунку 2.26.

Оформить заказ сейчас

Пожалуйста введи ваши контактные данные, и мы сразу отправим товар!

Данные

Ваше имя:

Требуется поле имя

Адрес доставки

Первый адрес:

Требуется поле Первый адрес

Второй адрес:

Третий адрес:

Город:

Требуется поле город

Страна:

Требуется поле страна

Опции

☐ Использовать подарочную упаковку?

Обработать заказ

Рисунок 2.26 – Результат натиснення кнопки «Оформити замовлення»

2.3.5 Випробування ПЗ інтернет-магазину «Soft_IT»

Програма і методика випробувань розробленого програмного забезпечення приводиться в Додатку Б.

Тести «чорної скриньки» пишуться із використанням специфікації тестового програмного забезпечення. У результаті опрацювання специфікації до програмного забезпечення, що підлягає тестуванню, було отримано ряд тестів «чорної скриньки» (по одному набору для кожної підзадачі).

Результати тестування за принципом "чорного ящика" наведені в таблиці 2.3.

Таблиця 2.3 – Результати тестування за принципом "чорного ящика"

Тест	Сценарій	Результат
1	2	3
Видалити товар з корзини	Користувач на сторінці корзини навпроти необхідного товару натискає кнопку «Видалити»	Вибраний товар зникає із корзини, а всі інші товари залишаються на місці
Аутентифікація	Користувач вводить правильну комбінацію логіну та паролю	Відбувається перенаправлення користувача на головну сторінку і на ній у спеціальному місці відображається ім'я користувача
	Користувач вводить неправильну комбінацію логіну та паролю	З'являється повідомлення про те, що дана комбінація логіну та паролю не існує
Залишити коментар	Користувач ввів коментар у спеціальне поле і натиснув кнопку «Відправити коментар»	Коментар з'являється на сторінці разом з іншими коментарями
Видалити коментар	Користувач навпроти одного зі своїх коментарів натиснув кнопку «Видалити коментар»	Коментар зникає зі сторінки, всі інші коментарі залишаються на місці
Додати товар до корзини	Користувач на сторінці товару натискає на кнопку «Додати до корзини»	Відображається повідомлення про успішне додання до корзини

Продовження табл. 2.3

1	2	3
Оформити замовлення	Користувач правильно заповнює всі поля та натискає кнопку «Оформити замовлення»	З'являється повідомлення про успішне оформлення замовлення
	Користувач неправильно заповнює поля та натискає кнопку «Оформити замовлення»	Відображаються усі помилки при заповненні полів
Реєстрація	Користувач вірно заповнив усі поля	Відбувається перенаправлення користувача на головну сторінку і на ній у спеціальному місці відображається ім'я користувача
	Користувач не вірно заповнив поля	Відображаються усі помилки при заповненні полів

Дані випробування показали, що розроблене програмне забезпечення функціонує нормально.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНУ «SOFT_IT»

У результаті виконання кваліфікаційної роботи було розроблене програмне забезпечення інтернет-магазину «Soft_IT».

Програма повністю відповідає вимогам до функціональних характеристик, вимогам до організації вхідних і вихідних даних та іншим вимогам, зазначеним у постановці задачі.

Розроблене програмне забезпечення реалізує наступні основні функції:

- Додавання на сайт нових комп'ютерних програм від користувачів;
- Видалення програм з сайту;
- Завантаження комп'ютерних програм з сайту;
- Можливість залишати коментарі до кожної з програм;
- Реєстрація користувачів;
- Аутентифікація користувачів.

Інструкція користувача наведена в Додатку В. Лістинг коду представлено у Додатку Д.

ПЗ призначене для використання на персональних комп'ютерах, що працюють під управлінням наступних операційних систем: ОС MS Windows XP/7/8/8.1/10 та вище.

Для ПЗ пред'являються такі апаратні вимоги до ПЕОМ:

- Процесор Intel Pentium 1Ghz і вище.
- Оперативна пам'ять 512Мб і вище.
- Мінімум дискового простору 1 Гб.
- Екран з роздільною здатністю не менше ніж 800x600 пікселів.

Під час розробки програмного забезпечення було проведене його тестування та випробування (згідно програми та методики випробувань, наведеної в Додатку Б), у ході яких була підтверджена повна працездатність, а також високий рівень надійності програмного продукту.

4 ОХОРОНА ПРАЦІ

4.1 Загальні принципи охорони праці

Згідно до ст.1 Закону "Про охорону праці", охорона праці є системою правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людей в процесі праці. Як правовий інститут, охорона праці визначає цілий комплекс правових норм.

Ст.4 Закону "Про охорону праці" формулює принципи державної політики у сфері охорони праці. Ці принципи в силу вказаного їх визначення, слід кваліфікувати і як принципи інституту охорони праці, які підлягають застосуванню при відсутності відповідних конкретних правових норм, і як такі, що визначають напрямки подальшого розвитку інституту охорони праці, і обов'язки держави, які вона взяла на себе і має додержувати при здійсненні своєї політики у майбутньому. З урахуванням викладеного, принципи, зазначені ст. 4 Закону "Про охорону праці", мають певне регулятивне значення. У названій статті закріплені такі принципи:

- пріоритет життя і здоров'я працівників відносно результатів виробничої діяльності підприємства, повна відповідальність власника за створення безпечних і нешкідливих умов праці;
- комплексне вирішення завдань охорони праці на засадах національних програм з цих питань, а також з урахуванням інших напрямків економічної і соціальної політики, досягнень в галузі науки, техніки та охорони здоров'я;
- соціальний захист працівників, повне відшкодування шкоди особам, потерпілим від нещасних випадків на виробництві та професійних захворювань;
- установлення єдиних нормативів з охорони праці для всіх підприємств

незалежно від форм власності і видів їх діяльності;

- використання економічних методів управління охороною праці, проведення політики пільгового оподаткування, що сприяє створенню безпечних та нешкідливих умов праці, участь держави в фінансуванні заходів з охорони праці;

- здійснення навчання населення, професійної підготовки та підвищення кваліфікації працівників з питань охорони праці;

- забезпечення координації діяльності державних органів, установ, організацій та об'єднань громадян, що вирішують різні проблеми охорони здоров'я, гігієни та безпеки праці.

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. Власник або уповноважений ним орган повинен впроваджувати сучасні засоби техніки безпеки, які запобігають виробничому травматизмові, і забезпечувати санітарно-гігієнічні умови, що запобігають виникненню професійних захворювань працівників [6].

На власника або уповноважений ним орган покладається систематичне проведення інструктажу (навчання) працівників з питань охорони праці, протипожежної охорони.

Охорона праці – один з центральних інститутів трудового права. Він має виключно практичне значення. Недотримання вимог охорони праці створює небезпеку для здоров'я і життя працівників. У свою чергу і ті, кого законодавець називає власником або уповноваженим ним органом, несуть сувору, у тому числі й кримінальну, відповідальність за порушення правил охорони праці.

4.2 Аналіз небезпечних і шкідливих факторів при роботі на ПК в інтернет-магазині «Soft_IT»

З точки зору впливу небезпечних і шкідливих факторів при роботі на ПК в інтернет-магазині «Soft_IT» можна виділити наступні:

- вплив електромагнітного та електростатичного випромінювань;
- вплив недостатньої освітленості робочого місця;
- вплив електричного струму;
- вплив шуму;
- вплив несприятливих умов мікроклімату;
- вплив нераціонального розташування устаткування і неправильної організації робочого місця.

Тривале перебування людини в зоні комбінованого впливу різних несприятливих факторів може привести до професійного захворювання. Розглянемо більш детально вплив означених факторів на користувача ПК.

Вплив електромагнітного та електростатичного випромінювань.

При роботі ПК виникають два типи випромінювань: електростатичне та електромагнітне. Навколо електростатично зарядженого монітора підвищується концентрація пилу. Електромагнітне випромінювання створюється магнітними котушками, що знаходяться біля цокольної частини ЕПТ. Результати медичних досліджень показують, що електризований пил викликає запалення шкіри, а електромагнітне випромінювання приводить до порушення біологічних процесів (порушення сну, зниження кров'яного тиску, уповільнення серцевої діяльності, зміни складу крові).

Вплив недостатньої освітленості робочого місця [6].

Напружена зорова робота внаслідок нераціонального освітлення може бути причиною функціональних порушень у зоровому аналізаторі і привести до розладу зору, а у важких випадках – і до повної втрати. Втома органів зору залежить від ступеня напруженості процесів, що супроводжують зорове сприйняття. При роботі оператора ПК одним з основних навантажень на

організм є навантаження на зір, тому дуже важливо приділити особливу увагу освітленості робочого місця.

У залежності від джерела світла виробниче освітлення може бути трьох видів: природним, штучним і сполученим. При роботі з ПК вдень у ясну погоду використовується природне освітлення, яке здійснюється через світлові віконні прорізи. При нестачі природного світла, а також при роботі у вечірній час використовується штучне освітлення, джерелами якого є газорозрядні лампи. При такому освітленні величезне значення має правильне розташування джерел світла, їхній тип і кількість.

Правильна освітленість робочого місця сприяє високій працездатності користувача ПК і зводить до мінімуму шкідливий вплив на органи зору. Освітленість робочого місця, що рекомендується для користувача ПК складає 400–750 лк.

Вплив електричного струму.

Робота комп'ютерів і периферійних пристроїв можлива завдяки електричному струмові, до джерел якого вони підключаються, тому існує небезпека враження користувача струмом. Щоб уникнути нещасних випадків варто строго дотримуватися правил електробезпеки.

Небезпека електричного струму на відміну від інших небезпек збільшується тим, що людина не в змозі без спеціальних приладів виявити напругу дистанційно, а також швидкоплинністю поразки – небезпека виявляється, коли людина вже вражена. Враження струмом приводить до різних порушень в організмі, викликаючи як місцеву поразку тканин і органів, так і загальне враження організму. Людина відчуває дратівну дію змінного струму промислової частоти силою 0,6–1,5 мА і постійного струму 5–7 мА. Ці струми не представляють серйозної небезпеки для організму людини, тому що при такій силі струму можливе самостійне звільнення людини від контакту зі струмоведучими частинами. Для змінного струму промислової частоти сила невідпускаючого струму знаходиться в межах 6–20 мА. Постійний струм не

викликає невідпускаючого ефекту, але приводить до сильних болючих відчуттів, сила такого струму 15–80мА і більше.

Вплив шуму.

Шумом прийнято називати всякий небажаний для людини звук, що заважає сприйняттю корисних сигналів. Шум являє собою безладне сполучення звуків різної інтенсивності і частоти. Він викликає труднощі з розпізнанням кольорових сигналів, знижує швидкість сприйняття кольору, гостроту зору, зорову адаптацію, порушує сприйняття візуальної інформації, зменшує на 5–12% продуктивність праці. Тривалий вплив шуму з рівнем звукового тиску 90 дБ знижує продуктивність праці на 30–60%.

Робота ПК може супроводжуватися шумами, викликаними деякими рухливими пристроями усередині комп'ютера, а також підключеними до нього. До таких пристроїв можна віднести вентилятори, пристрої читання дисків, принтери. Виникаючий шум, у залежності від інтенсивності, може привести до підвищення нервової напруги і зниження працездатності.

Вплив несприятливих умов мікроклімату.

Мікрокліматичні параметри (температура, відносна вологість і швидкість руху повітря) впливають на самопочуття і здоров'я людини, а також на надійність роботи засобів обчислювальної техніки.

При підвищеній температурі і вологості в приміщенні відбувається перегрівання тіла, що погіршує самопочуття і працездатність. Варто враховувати, що ПК, які знаходяться в приміщенні, сприяють збільшенню температури.

Для комфортної роботи приймається температура $23 (\pm 2)^{\circ}\text{C}$ й $18 (\pm 2)^{\circ}\text{C}$ в теплий і холодний час року відповідно, при відносній вологості $55 (\pm 5)\%$.

Вплив нераціонального розташування устаткування і неправильної організації робочого місця.

Робоче місце – це частина простору, у якому користувач ПК здійснює трудову діяльність і проводить велику частину робочого часу. Важливу роль грає планування робочого місця, яке повинне задовольняти вимогам зручності

виконання робіт і економії енергії і часу користувача, зручності обслуговувань ЕОМ, дотримання правил охорони праці.

При плануванні робочого місця необхідно враховувати зони досяжності при розташуванні моніторів, клавіатури, маніпулятора «миша». Ці зони встановлюються на підставі антропологічних даних людського тіла. Під час роботи користувач ПК не повинен піддаватися впливові сторонніх подразників, якими можуть бути неадекватне фарбування пристроїв ЕОМ або приміщення.

4.3 Розробка заходів щодо зменшення впливу небезпечних і шкідливих факторів

Наявний у даний час комплекс розроблених організаційних заходів і технічних засобів захисту, накопичений досвід показує, що існує можливість домогтися значно більших успіхів у справі усунення впливу на працюючих небезпечних і шкідливих виробничих факторів.

Нижче приведені найбільш розповсюджені організаційні заходи і технічні засоби захисту людей, що працюють з комп'ютерами, від шкідливих виробничих факторів.

Колірне оформлення і світлове відображення

Рациональне колірне оформлення приміщення поліпшує санітарно-гігієнічні умови праці, підвищує його продуктивність і безпеку. Фарбування приміщень впливає на нервову систему людини, його настрій і в кінці кінців на продуктивність праці. Основні виробничі приміщення доцільно фарбувати відповідно до кольору технічних засобів. Освітлення приміщення й устаткування повинне бути м'яким, без блиску.

Відображення, включаючи відображення від вторинних джерел світла, повинне бути зведене до мінімуму. Для захисту від надлишкової яскравості вікон можуть бути застосовані штори й екрани.

У приміщеннях, де знаходиться комп'ютер, необхідно забезпечити наступні величини коефіцієнта відображення: для стелі – 60...70%, для стін –

40...50%, для підлоги – близько 30%. Для інших поверхонь і робочих меблів – 30...40%.

Шум і вібрація

Зниження шуму, створюваного на робочих місцях внутрішніми джерелами, а також шуму проникаючого ззовні, є дуже важливою задачею. Зниження шуму в джерелі надходження можна забезпечити застосуванням пружних прокладок між підставою машини, приладу й опорною поверхнею. В якості прокладки використовуються гума, повсть, пробка, різної конструкції амортизатори. Під настільні шумливі апарати можна підкладати м'які коврики із синтетичних матеріалів, а під ніжки столів, на яких вони встановлені, – прокладки з м'якої гуми, повсті, товщиною 6-8 мм. Кріплення прокладок можливе шляхом приклейки їх до опорних частин.

Рациональне планування приміщення і розміщення устаткування є важливим чинником, що дозволяє знизити шум при існуючому устаткуванні ЕОМ. Зниження рівня шуму, що проникає у виробниче приміщення ззовні, може бути досягнуте збільшенням звукоізоляції обгороджуваних конструкцій, ущільненням по периметру притворів вікон, дверей.

Таким чином для зниження шуму, створюваного на робочих місцях внутрішніми джерелами, а також шуму, що проникає з поза, слід:

- послабити шум самих джерел (застосування екранів, звукоізолюючих кожухів);
- знизити ефект сумарного впливу відбитих звукових хвиль (звуковбирні поверхні конструкцій);
- застосовувати рациональне розташування устаткування;
- використовувати архітектурно-планувальні і технологічні рішення ізоляції джерел шуму.

Мікроклімат

Нормалізація мікроклімату приміщення може бути досягнута поліпшенням вентиляції приміщення за рахунок витяжних вентиляторів і нормального функціонування вбудованої вентиляції будинку.

ВИСНОВКИ

Мета даної кваліфікаційної роботи полягала в розв'язанні практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме в розробці якісного, зручного, багатофункціонального інтернет-магазину з продажу комп'ютерних програм з найменшими витратами.

Для досягнення поставленої мети були вирішені наступні задачі:

- проведено дослідження предметної галузі;
- сформульовані вимоги до програмного забезпечення;
- виконане ескізне проектування;
- виконане технічне проектування;
- виконане робоче проектування: кодування, тестування, випробування програмного забезпечення;
- розроблена документація.

Програма написана з використанням мови програмування C#.

В розробленому програмному забезпеченні реалізували наступні функції:

- додавання на сайт нових комп'ютерних програм від користувачів;
- видалення програм з сайту;
- завантаження програм з сайту;
- можливість залишати коментарі до кожної з програм;
- реєстрація користувачів;
- аутентифікація користувачів.

Крім того, можна зазначити, що даний програмний продукт має можливість подальшого розширення, його можна доповнювати вирішенням додаткових задач, що робить вищевказану систему гнучкою та в певній мірі універсальною.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Крамер Э. HTML: наглядный курс Web-дизайна [Текст] / Э. Крамер – М.: Издательский дом “Вильямс”, 2001. – 304 с.
2. Пахомов Б.И. С# для начинающих. [Текст] / Б.И. Пахомов. – СПб.: БХВ-Петербург, 2014. – 432 с.
3. Квартрани Т. Rational Rose и визуальное моделирование [Текст] / Т. Квартрани – М.: ДМК Пресс, 2001. – 176с.
4. Ульман Джеффри Д. Введение в систему баз данных [Текст] / Джеффри Д. Ульман, Дженнифер Уидом. М.: Издательство “Лори”, 2000. – 375с.
5. Вендров А.М. Проектирование программного обеспечения экономических информационных систем [Текст] / А.М. Вендров – М.: Финансы и статистика, 2006. – 544с.
6. Геврик Є.О. Охорона праці: Навчальний посібник для студентів вищих навчальних закладів [Текст] / Є.О. Геврик. – К.: Ельга, Ніка-Центр, 2003. – 280 с.

ДОДАТОК А – Технічне завдання на розробку програмного забезпечення інтернет-магазину «Soft_IT»

Вступ

Повна назва програмного продукту: програмне забезпечення інтернет-магазину «Soft_IT».

Сфера застосування: інтернет торгівля.

1 Підстави для розробки

Підставою для розробки даного програмного забезпечення є завдання на кваліфікаційну роботу.

2 Призначення розробки

2.1 Функціональне призначення

Функціональним призначенням системи є полегшення процесу купівлі програм, а також процесу продажу власного програмного забезпечення.

2.2 Експлуатаційне призначення

Програма повинна використовуватися людьми, які ознайомлені з основами роботи з комп'ютером.

3 Вимоги до програми або програмного виробу

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

Програма повинна забезпечувати можливість виконання нижче перерахованих функцій:

- Реєстрація користувачів (вхідні дані: логін, пароль та електронна пошта користувача; вихідні дані: відображення на сайті інформації про успішну реєстрацію);
- Аутентифікація користувачів (вхідні дані: логін та пароль; вихідні дані: відображення на сайті інформації про успішну авторизацію);

- Додавання на сайт нових програм від користувачів (вхідні дані: файл програми, назва програми, опис програми; вихідні дані: відображення на сайті інформації про успішне додання програми);
- Видалення програм з сайту (вхідні дані: id програми; вихідні дані: відображення на сайті інформації про успішне видалення);
- Завантаження програм з сайту (вхідні дані: id програми; вихідні дані: файл з програмою);
- Можливість залишати коментарі до кожної з програм (вхідні дані: текст, введений користувачем у спеціальне текстове поле для коментарів; вихідні дані: відображення на сайті коментаря користувача);

3.1.2 Вимоги до організації вхідних та вихідних даних

Вхідні дані програми повинні вводитися у спеціальні поля на сайті, або бути організовані у вигляді файлу, обраного на жорсткому диску.

Вихідні дані програми мають відображатися на сайті, або зберігатися в файл.

3.1.3 Вимоги до часових характеристик

Вимог до часових характеристик не висувається.

3.2 Вимоги до надійності

3.2.1 Вимоги до забезпечення надійного (сталого) функціонування програми

Надійне (стійке) функціонування програми має бути забезпечене виконанням сукупності організаційно-технічних заходів, перелік яких подано нижче:

- захист від некоректних дій користувача(під час розпізнавання);
- встановлення платформи .net не нижче версії 4.5.

3.2.2 Час відновлення після відмови

Вимоги до часу відновлення після відмови не пред'являються.

3.2.3 Відмови через некоректні дії оператора

Відмова можлива при наступних умовах:

- внаслідок закриття програми під час розпізнавання, в цьому випадку користувачеві показується додаткове вікно, чи впевнений він що хоче закрити програму;
- якщо розмір файлу перевищує розмір оперативної пам'яті, то з'являється вікно про нестачу оперативної пам'яті для розпізнавання файлу.

3.3 Умови експлуатації

3.3.1 Кліматичні умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні відповідати вимогам, що пред'являються до технічних засобів в частині умов їх експлуатації.

3.3.3 Вимоги до чисельності та кваліфікації користувачів

Користувач повинен мати базові навички роботи з комп'ютером.

3.4 Вимоги до складу і параметрів технічних засобів

Для роботи із програмою необхідний наступний склад технічних засобів з мінімальними параметрами:

- Процесор Intel Pentium 1Ghz і вище.
- Оперативна пам'ять 512Мб і вище.
- Мінімум дискового простору 1 Гб.
- Екран з роздільною здатністю не менше ніж 800x600 пікселів.

3.5 Вимоги до інформаційної та програмної сумісності

3.5.1 Вимоги до вихідних кодів і мов програмування

Вихідні коди програми повинні бути реалізовані на мові C#. В якості інтегрованого середовища розробки програми може бути використане середовище Visual Studio.

3.5.2 Вимоги до програмних засобів, використовуваних програмою

Системні програмні засоби, які використовуються програмою, повинні бути представлені ліцензійною версією операційної системи Windows (версія не нижче Windows 7) та встановленою на комп'ютері платформи .NET не нижче версії 4.5.

3.5.3 Вимоги до захисту інформації та програм

Вимоги до захисту інформації та програм не пред'являються.

3.6 Вимоги до маркування та упаковки

3.6.1 Вимоги до маркування

Програма поставляється вже встановленою на спеціальному обладнанні та готовою к використанню.

3.6.2 Вимоги до упаковки

Упаковка програмного виробу повинна здійснюватися в пакувальну тару підприємства-виробника.

3.7 Вимоги до транспортування і зберігання

Допускається транспортування програмного виробу в транспортній тарі всіма видами транспорту (у тому числі опалювальних герметизованих відсіках літаків без обмеження відстаней). При перевезенні в залізничних вагонах вид відправки - невеликий малотоннажний.

Кліматичні умови транспортування наведені нижче:

- температура навколишнього повітря, ° С - від +5 до +35;
- атмосферний тиск, 101,3 кПа;
- відносна вологість повітря при 25 ° С – 30%.

3.8 Спеціальні вимоги

Програма повинна забезпечувати взаємодію з користувачем за допомогою графічного інтерфейсу користувача, розробленого відповідно до рекомендацій компанії-виробника операційної системи.

4 Вимоги до програмної документації

Склад програмної документації повинен включати в себе:

- технічне завдання;
- програма і методика випробувань;
- керівництво користувача
- опис програми;
- код програми.

5 Техніко-економічні показники

У даній кваліфікаційній роботі техніко-економічні показники не розраховуються.

6 Стадії та етапи розробки

Стадії та етапи розробки представлені у таблиці 1

Таблиця 3.1 – Стадії та етапи розробки

Стадії розробки	Етапи робіт	Вміст робіт	Контрольні точки
1	2	3	4
Технічне завдання	Обумовлення необхідності розробки	Постановка задачі. Збір початкових матеріалів. Вибір і обумовлення критеріїв ефективності та якості розроблюваного ПЗ.	З 24.03.2022 до 30.03.2022
	Розробка та затвердження технічного завдання	Визначення вимог до ПЗ. Обговорення та затвердження технічного завдання	З 31.03.2022 до 06.04.2022
Ескізний проект	Розробка ескізного проекту	Попередня розробка структури вхідних та вихідних даних, уточнення методів рішення завдань, загальний опис алгоритму	З 07.04.2022 до 17.04.2022
	Затвердження ескізного проекту	Розробка пояснювальної записки. Узгодження і затвердження ескізного проекту	З 18.04.2022 до 20.04.2022

Продовження табл. 3.1

1	2	3	4
Технічний проект	Розробка технічного проекту	Уточнення структури вхідних та вихідних даних, розробка алгоритму рішення задачі, розробка структури ПЗ	З 21.04.2022 до 28.04.2022
	Затвердження технічного проекту	Розробка пояснювальної записки. Узгодження і затвердження технічного проекту	З 29.04.2022 до 31.05.2022
Робочий проект	Розробка ПЗ	Програмування та налагодження програми	З 01.05.2022 до 10.05.2022
	Розробка програмної документації	Розробка програмної документації відповідно до вимог	З 11.05.2022 до 13.05.2022
	Випробування програми	Розробка, узгодження та затвердження програми і методики випробувань, коригування програми і програмної документації за результатами випробувань	З 14.05.2022 до 16.05.2022

7 Порядок контролю та приймання

Для контролю приймання має бути надано опис програми, а також програма і методика випробувань.

Якщо програма не пройшла тестування, виконавець зобов'язаний виправити помилки і недоліки в строк, не більш ніж 1 неділя від дня випробування.

ДОДАТОК Б – Програма та методика випробувань програмного забезпечення інтернет-магазину «Soft_IT»

1 Об'єкт випробувань

Об'єктом випробувань є програмне забезпечення інтернет-магазину «Soft_IT».

2 Мета випробувань

Метою випробувань являється перевірка відповідності функцій програми вимогам, визначеним у постановці задачі.

3 Вимоги до програми

Дане програмне забезпечення повинно виконувати наступні функції:

- Додавання на сайт нових комп'ютерних програм від користувачів;
- Видалення програм з сайту;
- Завантаження програм з сайту;
- Можливість залишати коментарі до кожної з програм;
- Реєстрація користувачів;
- Аутентифікація користувачів.

4 Вимоги до програмної документації

До складу програмної документації повинні входити наступні документи:

- технічне завдання;
- програма та методика випробувань;
- код програми;
- опис програми;
- керівництво користувача.

5 Засоби та порядок випробувань

Вимоги до складу і параметрів технічних засобів:

- Процесор Intel Pentium 1Ghz і вище.
- Оперативна пам'ять 512Мб і вище.
- Мінімум дискового простору 1 Гб.
- Екран з роздільною здатністю не менше ніж 800х600 пікселів.

Вимоги програмних засобів:

Системні програмні засоби, які використовуються програмою, повинні бути представлені ліцензійною версією операційної системи Windows (версія не нижче Windows 7) та встановленою на комп'ютері платформою .NET не нижче версії 4.5.

Порядок проведення випробувань.

Випробування програмного забезпечення повинні виконуватися в наступному порядку:

Порядок проведення випробувань:

- виконуються інструкції до кожної функції;
- робиться аналіз отриманих результатів і встановлюється відповідність програмного продукту вимогам і системним документам.

При виявленні помилок у роботі системи складається їх перелік і обговорюється термін їх виправлення розробником. Після цього замовник проводить повторне тестування в повному обсязі (можливе використання нових чи додаткових тестів).

6 Методи випробувань

Випробування будемо проводити за стратегією «чорного ящика». За рахунок дуже великої кількості функцій, які потрібно випробовувати, представимо тільки декілька результатів випробувань функцій програми. Всі функції програми будуть проходити випробовування.

Складемо програму випробувань:

1) Перевірити можливість додати новий товар до корзини

Натиснемо на сторінці товару кнопку «Додати до корзини», якщо після

натиснення кнопки «Додати до корзини» товар додається до корзини і відображається повідомлення про додавання товару до корзини, то випробування пройшло успішно.

2) Випробування можливості видалити товар із корзини

Натиснемо кнопку «Видалити» навпроти відповідного товару у корзині, якщо товар зникає із корзини і виводиться повідомлення про успішне видалення з корзини, то випробування пройшло успішно.

3) Випробування можливості видалення коментаря

Натиснемо кнопку «Видалити коментар», якщо коментар зникає зі сторінки товару, то випробування пройшло успішно.

4) Випробування неможливості оформити замовлення при помилці валідації

Неправильного заповнемо поля на сторінці оформлення замовлення і натиснемо кнопку «Оформити замовлення», якщо виводиться інформацію про те, які саме поля були заповнені невірно, то випробування пройшло успішно.

5) Випробування можливості залишити коментар

Введемо свій коментар у спеціальну форму на сторінці товару і натиснемо кнопку «Відправити коментар», якщо на сторінці товару з'являється цей коментар, то випробування пройшло успішно.

6) Випробування можливості аутентифікації

Правильного заповнимо всіх поля і натиснемо кнопку «Ввійти», якщо програма перенаправить нас на головну сторінку і ім'я користувача відображається у спеціально відведеному місці, то випробування пройшло успішно.

7) Випробування неможливості аутентифікації за неіснуючою комбінацією логіна та пароллю

Введемо неіснуючу комбінацію логіна та пароллю і натиснемо кнопку «Ввійти». Відображається повідомлення про неправильну комбінацію логіну та пароллю.

ДОДАТОК В – Керівництво користувача програмного забезпечення інтернет-магазину «Soft_IT»

Для встановлення програми необхідно створити папку на жорсткому диску, куди скопіювати вказані нижче файли:

– WebApplication3.exe

Щоб запустити програму треба двічі натиснути на скопійований файл WebApplication3.exe та у браузері перейти за адресою: localhost, після чого з'явиться головна сторінка сайту.

Головна сторінка сайту наведена на рисунку 1.



Рисунок 1 – Головна сторінка сайту

Після цього для доступу до всіх функцій сайту користувач має пройти аутентифікацію(якщо він раніше пройшов реєстрацію) або зареєструватися. Для цього йому необхідно на головній сторінці натиснути кнопку «Реєстрація» і у вікні, що відкрилося, ввести всі необхідні дані та натиснути кнопку «Зареєструвати».

На рисунках 2 – 3 представлено процес реєстрації



Рисунок 2 – Кнопка «Реестрація»

Рисунок 3 – Сторінка реєстрації

Якщо користувач вже був зареєстрований, то на сторінці реєстрації він має натиснути кнопку «Ввійти». У вікні, що відкриється, користувач має ввести всі необхідні дані та натиснути кнопку «Ввійти».

На рисунках 4 – 5 представлено процес реєстрації



Рисунок 4 – Кнопка «Ввійти»

Рисунок 5 – Сторінка аутентифікації

Користувач має можливість додати товар до корзини. Для цього йому необхідно авторизуватися, вибрати одну з категорій, а потім серед програм цієї категорії вибрати необхідну йому і натиснути кнопку «Детальніше». Далі користувач на сторінці програми, яка відкрилася, натискає кнопку «Додати до корзини». Після того як товар був доданий до корзини, програма оповістить користувача про це.

На рисунках 6 – 7 представлено процес додавання товару до корзини

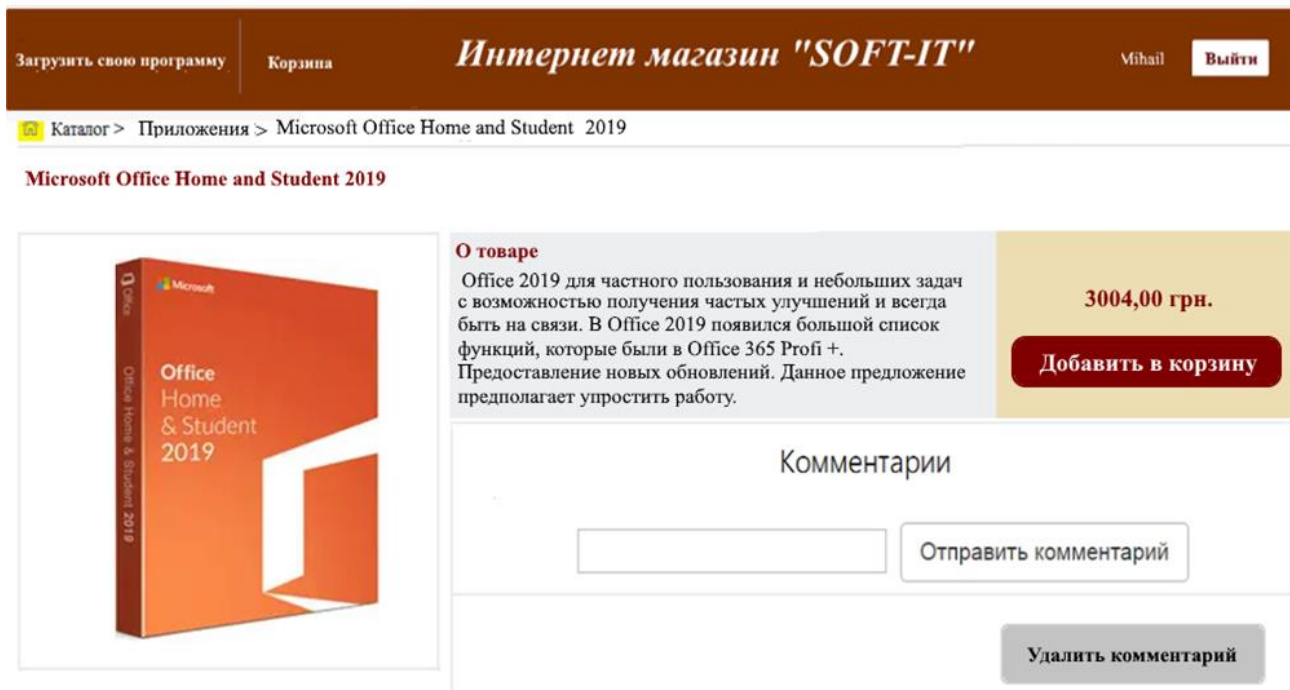


Рисунок 6 – Сторінка товару

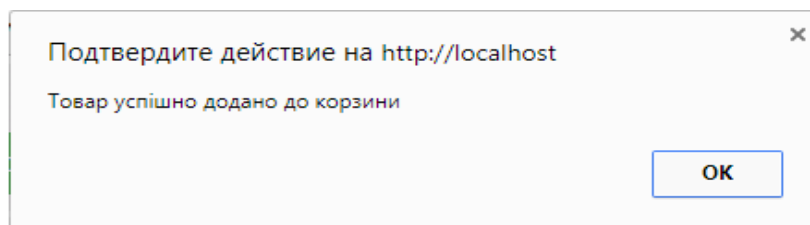


Рисунок 7 – Повідомлення про успішне додавання до корзини

Далі користувач може перейти на сторінку корзини та оформити замовлення або видалити товар із корзини. Для видалення товару із корзини

користувач має натиснути кнопку «Видалити» навпроти необхідного товару. Після цього товар видаляється з корзини, а система виводить повідомлення про успішне видалення товару.

На рисунках 8 – 9 представлено процес видалення товару з корзини



Рисунок 8 – Сторінка корзини

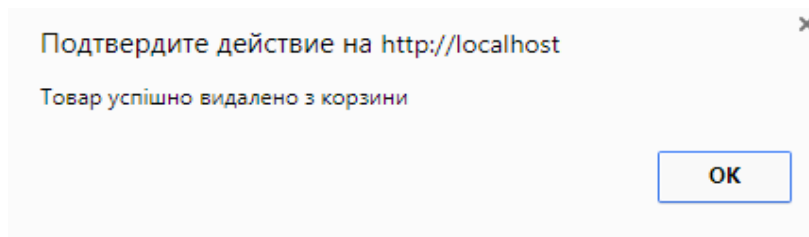


Рисунок 9 – Повідомлення про успішне видалення з корзини

Якщо ж користувач хоче оформити своє замовлення, то він має натиснути кнопку «Оформити замовлення» і у вікні, що відкрилося після цього, заповнити всі обов'язкові поля (ім'я, перша адреса, місто та країна) та всі необхідні йому необов'язкові поля та натиснути кнопку «Опрацювати замовлення».

На рисунках 10 – 11 представлено процес оформлення замовлення

Оформить заказ				Mihail	Выйти
Корзина					
Кол-во	Программа	Цена	Общая цена		
1	Microsoft Office Home and Student 2019 (Электронная лицензия)	3004 грн.	3004 грн. 00 к.	Удалить	
1	Microsoft Office 365 Персональный (Электронная лицензия)	1483 грн.	1483 грн. 00 к.	Удалить	
Итого:			4487 грн. 00 к.		
Оформить заказ					

Рисунок 10 – Сторінка корзини

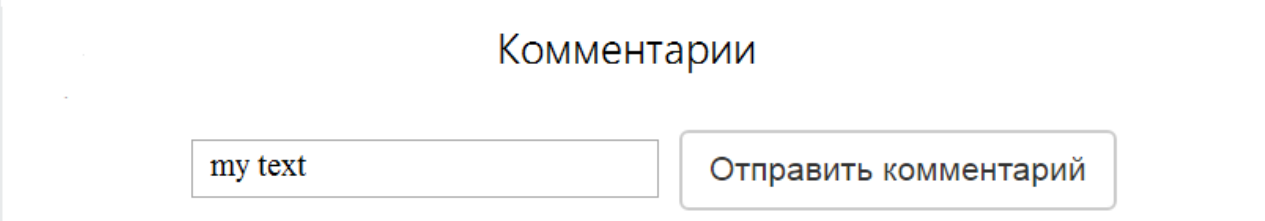
Загрузить свою программу	Корзина	Интернет магазин "SOFT-IT"	Mihail	Выйти
Оформить заказ сейчас Пожалуйста введи ваши контактные данные, и мы сразу отправим товар!				
Данные				
Ваше имя:				
Адрес доставки				
Первый адрес:				
Второй адрес:				
Третий адрес:				
Город:				
Страна:				
Опции				
☐ Использовать подарочную упаковку?				
Обработать заказ				

Рисунок 11 – Сторінка оформлення замовлення

Користувач також може залишати коментарі до будь-якого товару, а також видаляти власні коментарі. Для того щоб відправити коментар користувач на сторінці товару має у спеціальне поле ввести свій текст та

натиснути кнопку «Відправити коментар». Після цього коментар буде відображений на сторінці товару.

Форма відправки коментарю представлена на рисунку 12



The image shows a web form titled "Комментарии" (Comments). It contains a text input field with the placeholder text "my text" and a button labeled "Отправить комментарий" (Submit comment).

Рисунок 12 – Форма відправки коментарю

Якщо ж користувач хоче видалити свій власний коментар, то він має навпроти відповідного коментарю натиснути кнопку «Видалити коментар». Після цього коментар буде видалений і більше не буде відображатись на сторінці товару. Форма видалення коментарю представлена на рисунку 13



The image shows a web form for deleting a comment. It displays the comment ID "comment1" and the text "my text". To the right of the text is a button labeled "Удалить комментарий" (Delete comment).

Рисунок 13 – Форма видалення коментарю

Якщо користувач увійшов в систему як адміністратор, то ПЗ його відразу перенаправляє на сторінку адміністрування, де він може редагувати інформацію про товар або видаляти товар.

Сторінка адміністрування представлена на рисунку 14

Панель администрирования			
Admin			Выйти
ID	Название	Цена	Действия
1	Microsoft Office Home and Student 2019 (Электронная лицензия)	3004 грн.	Редактировать Удалить
2	Microsoft Office 365 Персональный (Электронная лицензия)	1483 грн.	Редактировать Удалить
3	Microsoft Office Home and Student 2021 Ukrainian (BOX)	3990 грн.	Редактировать Удалить

Рисунок 14 – Сторінка адміністрування

ДОДАТОК Г – Опис програмного забезпечення інтернет-магазину «Soft_IT»

1 Загальні відомості

1.1 Позначення і найменування програми

Найменування: Програмне забезпечення інтернет-магазину «Soft_IT»

Позначення: «Soft_IT»

1.2 Програмне забезпечення, необхідне для функціонування програми

Програмне забезпечення інтернет-магазину «Soft_IT» є крос-платформним і основною необхідною вимогою для функціонування виробу є наявність .NET Framework (4.5 і вище версії).

1.3 Мови програмування

В якості інтегрованого середовища розробки програми використане середовище Visual Studio.

ПЗ «Soft_IT» реалізовано на мові високого рівня C#. Модулі, реалізовані на інших мовах програмування не використовуються.

2 Функціональне призначення

2.1 Класи вирішуваних завдань і призначення програми

ПЗ «Soft_IT» повинно забезпечувати можливість виконання нижче перерахованих функцій:

- реєстрація користувачів ;
- аутентифікація користувачів ;
- додавання на сайт нових програм від користувачів ;
- видалення програм з сайту ;
- завантаження програм з сайту ;
- можливість залишати коментарі до кожної з програм ;
- можливість редагувати інформацію про товар.

3 Опис логічної структури

3.1 Загальна структура

3.1.1 Логічна структура програми

В ПЗ всі класи знаходяться в головному пакеті «WebApplication3».

Клас Orders

Клас призначений для обробки даних, які передаються з таблиці бази даних Orders до інтерфейсу користувача.

Клас Categorys

Клас призначений для обробки даних, які передаються з таблиці бази даних Categorys до інтерфейсу користувача.

Клас StoreContext

Клас призначений для отримання інформації з бази даних за допомогою Entity Framework.

Клас ProgramController

Клас реалізуючий різноманітні методи для роботи з товаром.

Клас AccountController

Реалізує методи для аутентифікації та реєстрації користувачів сайту.

Клас HomeController

Реалізує методи для головної сторінки сайту.

Клас CartController

Клас призначений для виконання операцій з корзиною.

Клас Roles

Клас призначений для обробки даних, які передаються з таблиці бази даних Roles до інтерфейсу користувача.

Клас Comments

Клас призначений для обробки даних, які передаються з таблиці бази даних Comments до інтерфейсу користувача.

Клас Programs

Клас призначений для обробки даних, які передаються з таблиці бази даних Programs до інтерфейсу користувача.

Клас Users

Клас призначений для обробки даних, які передаються з таблиці бази даних Users до інтерфейсу користувача.

Клас Cart

Клас який виконує роль корзини.

Клас CartLine

Клас який відображає запис у корзині.

3.2 Алгоритми роботи програми

3.2.1 Запуск ПЗ «Soft_IT»

«Soft_IT» – запускається при подвійному кліку на файл “WebApplication3.exe”.

3.2.2 Графічна оболонка

Роботу з програмою описана в Додатку В

3.3 Використовувана методологія

ПЗ «Soft_IT» розроблено на мові програмування C#, яка є об'єктно-орієнтованою. В основу розробки ПЗ «Soft_IT» покладено об'єктно-орієнтовану методологію.

3.4 Зв'язок із стороннім ПЗ

ПЗ «Soft_IT» взаємодіє з наступним стороннім ПЗ, яким є .NET Framework (версія не нижче 4.5) та операційна система Windows (версія не нижче Windows 7).

4 Використовувані технічні засоби

Для роботи із програмою необхідний наступний склад технічних засобів з мінімальними параметрами:

- Процесор Intel Pentium 1Ghz і вище.
- Оперативна пам'ять 512Мб і вище.

- Мінімум дискового простору 1 Гб.
- Екран з роздільною здатністю не менше ніж 800x600 пікселів.

Швидкість роботи ПЗ «Soft_IT» на конкретному комп'ютері залежить також від характеристик окремих його комплектуючих (процесора, оперативної пам'яті і ін.).

5 Виклик і завантаження

Запуск ПЗ «Soft_IT» в графічному режимі здійснюється двома способами:

- 1) за допомогою файлу запуску: WebApplication3.exe;
- 2) з використанням командного рядку:

WebApplication3.exe

6 Вхідні і вихідні дані

Вхідні дані програми повинні вводитися у спеціальні поля на сайті, або бути організовані у вигляді файлу, обраного на жорсткому диску.

Вихідні дані програми мають відображатися на сайті, або зберігатися в файл.

ДОДАТОК Д – Текст програмного забезпечення інтернет-магазину «Soft_IT»

Нижче представлений текст програми, розробленої на мові програмування C#

Текст класу CartController

```
using System.Linq;
using System.Web.Mvc;
using ProgramStore.Domain.Entities;
using ProgramStore.Domain.Abstract;
using ProgramStore.WebUI.Models;

namespace ProgramStore.WebUI.Controllers
{
    public class CartController : Controller
    {
        StoreContext db = new StoreContext();
        public CartController(IProgramRepository repo)
        {
            repository = repo;
        }

        public RedirectToRouteResult AddToCart(int ProgramId, string returnUrl)
        {
            Program Program = repository.Programs
                .FirstOrDefault(g => g.ProgramId == ProgramId);

            if (Program != null)
            {
                GetCart().AddItem(Program, 1);
            }
            return RedirectToAction("Index", new { returnUrl });
        }

        public RedirectToRouteResult RemoveFromCart(int ProgramId, string returnUrl)
        {
            Program Program = repository.Programs
                .FirstOrDefault(g => g.ProgramId == ProgramId);
```

```

        if (Program != null)
        {
            GetCart().RemoveLine(Program);
        }
        return RedirectToAction("Index", new { returnUrl });
    }

    public Cart GetCart()
    {
        Cart cart = (Cart)Session["Cart"];
        if (cart == null)
        {
            cart = new Cart();
            Session["Cart"] = cart;
        }
        return cart;
    }
}

```

Код класу AccountController

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;
using WebApplication3.Models;

namespace WebApplication3.Controllers
{
    public class AccountController : Controller
    {
        public ActionResult Login()
        {
            return View();
        }
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Login(LoginModel model)
        {
            if (ModelState.IsValid)
            {
                // поиск пользователя в бд
            }
        }
    }
}

```

```

        User user = null;
        using (StoreContext db = new StoreContext())
        {
            user = db.Users.FirstOrDefault(u => u.Name == model.Name &&
u.Password == model.Password);

        }
        if (user != null)
        {
            FormsAuthentication.SetAuthCookie(model.Name, true);
            return RedirectToAction("Index", "Home");
        }
        else
        {
            ModelState.AddModelError("", "Пользователя с таким логином и
паролем нет");
        }
    }
    return View(model);
}

public ActionResult Register()
{
    return View();
}
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Register(RegisterModel model)
{
    if (ModelState.IsValid)
    {
        User user = null;
        using (StoreContext db = new StoreContext())
        {
            user = db.Users.FirstOrDefault(u => u.Name == model.Name &&
u.Password == model.Password);
        }
        if (user == null)
        {
            using (StoreContext db = new StoreContext())
            {
                db.Users.Add(new User { Name = model.Name, Password =
model.Password, RoleId = 2 });
                db.SaveChanges();
            }
        }
    }
}

```

```

        user = db.Users.Where(u => u.Name == model.Name &&
u.Password == model.Password).FirstOrDefault();
    }
    if (user != null)
    {
        FormsAuthentication.SetAuthCookie(model.Name, true);
        return RedirectToAction("Index", "Home");
    }
}
else
{
    ModelState.AddModelError("", "Пользователь с таким логином
уже существует");
}
}
return View(model);
}
public ActionResult Logoff()
{
    FormsAuthentication.SignOut();
    return RedirectToAction("Register", "Account");
}
}
}

```

Текст класу Cart

```

using System.Collections.Generic;
using System.Linq;

namespace ProgramStore.Domain.Entities
{
    public class Cart
    {
        private List<CartLine> lineCollection = new List<CartLine>();

        public void AddItem(Program Program, int quantity)
        {
            CartLine line = lineCollection
                .Where(g => g.Program.ProgramId == Program.ProgramId)
                .FirstOrDefault();

            if (line == null)
            {
                lineCollection.Add(new CartLine

```

```

        {
            Program = Program,
            Quantity = quantity
        });
    }
    else
    {
        line.Quantity += quantity;
    }
}

public void RemoveLine(Program Program)
{
    lineCollection.RemoveAll(l => l.Program.ProgramId ==
Program.ProgramId);
}

public decimal ComputeTotalValue()
{
    return lineCollection.Sum(e => e.Program.Price * e.Quantity);
}

public void Clear()
{
    lineCollection.Clear();
}

public IEnumerable<CartLine> Lines
{
    get { return lineCollection; }
}
}

```

Тест класу CartLine

```

public class CartLine
{
    public Program Program { get; set; }
    public int Quantity { get; set; }
}

```

Тест класу CustomRoleProvider

```

using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Web;
using System.Web.Security;
using WebApplication3.Models;

namespace WebApplication3.Providers
{
    public class CustomRoleProvider : RoleProvider
    {
        public override string[] GetRolesForUser(string username)
        {
            string[] role = new string[] { };
            using (StoreContext db = new StoreContext())
            {
                // Получаем пользователя
                User user = db.Users.FirstOrDefault(u => u.Name == username);
                if (user != null)
                {
                    // получаем роль
                    Role userRole = db.Roles.Find(user.RoleId);
                    if (userRole != null)
                        role = new string[] { userRole.Name };
                }
            }
            return role;
        }
        public override void CreateRole(string roleName)
        {
            Role newRole = new Role() { Name = roleName };
            StoreContext db = new StoreContext();
            db.Roles.Add(newRole);
            db.SaveChanges();
        }
        public override bool IsUserInRole(string username, string roleName)
        {
            bool outputResult = false;
            // Находим пользователя
            using (StoreContext db = new StoreContext())
            {
                // Получаем пользователя
                User user = db.Users.FirstOrDefault(u => u.Name == username);
                if (user != null)
                {
                    // получаем роль
                    Role userRole = db.Roles.Find(user.RoleId);

```

```

        //сравниваем
        if (userRole != null && userRole.Name == roleName)
            outputResult = true;
    }
}
return outputResult;
}
}

```

Текст класу PagingHelper

```

using System;
using System.Text;
using System.Web.Mvc;
using WebApplication3.Models;

namespace WebApplication3.HtmlHelpers
{
    public static class PagingHelper
    {
        public static MvcHtmlString PageLinks(this HtmlHelper html,
            PagingInfo pagingInfo,
            Func<int, string> pageUrl)
        {
            StringBuilder result = new StringBuilder();
            for (int i = 1; i <= pagingInfo.TotalPages; i++)
            {
                TagBuilder tag = new TagBuilder("a");
                tag.MergeAttribute("href", pageUrl(i));
                tag.InnerHtml = i.ToString();
                if (i == pagingInfo.CurrentPage)
                {
                    tag.AddCssClass("selected");
                    tag.AddCssClass("btn-primary");
                }
                tag.AddCssClass("btn btn-default");
                result.Append(tag.ToString());
            }
            return MvcHtmlString.Create(result.ToString());
        }
    }
}

```

Текст класу ProgramController

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using WebApplication3.Models;
using System.Data.Entity;
using Newtonsoft.Json;

namespace WebApplication3.Controllers
{
    public class ProgramController : Controller
    {
        StoreContext db = new StoreContext();

        public ActionResult Comment(int? id)
        {
            var comments = db.Comments.Where(a => a.ProgramId==id).Include(a
=> a.User);
            return PartialView ("Edit",comments);
        }

        public ActionResult Index(int? id)
        {
            ViewBag.id = id;
            return View();
        }

        public ActionResult Comment(int? id, String text)
        {
            Comments com = new Comments();
            com.ProgramId = id;
            com.Text = text;
            com.UserId = 1;
            com.date = DateTime.Now;
            db.Comments.Add(com);
            db.SaveChanges();
            return RedirectToAction("Edit",new {id = id });
        }

        public ActionResult Delete(int? id,int? id2)
        {

```

```

        Comments com = db.Comments.Find(id);
        db.Comments.Remove(com);
        db.SaveChanges();
        return RedirectToAction("Edit", new { id = id2 });
    }

```

```
[HttpPost]
```

```
public ActionResult Upload(HttpPostedFileBase file)
{
```

```
    if (file != null && file.ContentLength > 0)
    {
```

```
        var fileName = Path.GetFileName(file.FileName);
        var path = Path.Combine(Server.MapPath("~/Images/"), fileName);
        file.SaveAs(path);
    }
```

```
        return RedirectToAction("UploadDocument");
    }
```

```
}

```

```
public FileResult Download()
```

```
{
    byte[] fileBytes = System.IO.File.ReadAllBytes(@"c:\folder\myfile.ext");
    string fileName = "myfile.ext";
```

```
    return File(fileBytes,
System.Net.Mime.MediaTypeNames.Application.Octet, fileName);
}
```

```
}
```

Текст класу StoreContext

```
public class StoreContext : DbContext
{
```

```
    public StoreContext() :
        base("DefaultConnection")
```

```
{ }
```

```
    public DbSet<User> Users { get; set; }
```

```
    public DbSet<Role> Roles { get; set; }
```

```
    public DbSet<Programs> Program { get; set; }
```

```
    public DbSet<Comments> Comments { get; set; }
```

```
    public DbSet<Categorys> Categorys { get; set; }
```

```
    public DbSet<Orders> Orders { get; set; }
```

```
}
```

Текст класу Users

```
public class Users
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Email { get; set; }
    public string Password { get; set; }
    public int RoleId { get; set; }
    public Role Role { get; set; }
}
```

Текст класу Roles

```
public class Roles
{
    public int Id { get; set; }
    public string Name { get; set; }
}
```

Текст класу Programs

```
public class Programs
{
    [Key]
    public int Id { get; set; }
    public string Name { get; set; }
    public int Price { get; set; }
    public int CategoryId { get; set; }
    public Category Category { get; set; }

    public ICollection<Comments> comment { get; set; }
    public Programs()
    {
        comment = new List<Comments>();
    }
}
```

Текст класу Comments

```
using System;

namespace WebApplication3.Models
{
    public class Comments
```

```

{
    public int Id { get; set; }

    public DateTime date { get; set; }

    public string Text { get; set; }

    public int? ProgramId { get; set; }
    public Programs Program { get; set; }

    public int? UserId { get; set; }
    public User User { get; set; }
}
}

```

Текст класу Categorys

```

public class Categorys
{
    public int Id { get; set; }
    public string Name { get; set; }
}

```