

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

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

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ ОНЛАЙН-МАГАЗИНУ ГОДИННИКІВ
SOFTWARE IMPLEMENTATION OF AN ONLINE WATCH STORE

Виконав: студент(ка) 4 курсу, групи КНТ-227
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

Стріжко В.Р.

(прізвище та ініціали)

Керівник Гофман Є.О.

(прізвище та ініціали)

Рецензент Скрупський С.Ю.

(прізвище та ініціали)

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

Інститут, факультет ІРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Стріжка Владислава Руслановича

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Програмна реалізація онлайн-магазину годинників.
Software Implementation of an Online Watch Store

керівник проекту (роботи) Гофман Євген Олександрович, к.т.н.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проекту (роботи) травень 2021 року

3. Вихідні дані до проекту (роботи) рекомендована література, технічне
завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Розробка архітектури програми.
3. Основні рішення щодо реалізації компонентів системи. 4. Експлуатація,
тестування та експериментальне дослідження програми.

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

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Гофман Є.О., доцент		
Нормоконтролер	Липовець М.В., асистент		

7. Дата видачі завдання “1” лютого 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз та дослідження предметної області	2-5 тижні	Розділи 1, 2
3	Розробка архітектури програми	6 тиждень	Розділ 2
4	Розробка програми	7-9 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми	10 тиждень	Розділ 4
		11 тиждень	Додатки
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування		
7	Захист роботи	12 тиждень	

Студент(ка)

(підпис)

Стріжко В.Р.

(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Гофман Є.О.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 88 с., 42 рис., 4 табл., 4 дод., 19 джерел.

ОНЛАЙН_МАГАЗИН, ЕЛЕКТРОННА КОМЕРЦІЯ, РОЗРОБКА, АНАЛІЗ ВИМОГ, АДАПТИВНІСТЬ, ПРОЄКТУВАННЯ, ВЕБ-ЗАСТОСУНОК.

Об'єкт дослідження – процес розробки та підтримки онлайн-магазинів.

Предмет дослідження – методи розробки та підтримки онлайн-магазинів.

Метою роботи є програмна реалізація онлайн-магазину для продажу годинників.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування PHP, платформа Magento, система керування базою даних MySQL.

Результати. Розроблено онлайн-магазин годинників у вигляді веб-застосунку, що реалізує функції комунікації та продажу-купівлі фотографій.

Практична цінність роботи полягає в розробці онлайн-магазину продажу годинників у вигляді кроссплатформного веб-застосунку, що реалізує зручні робочі панелі для користувачів (покупців) та адміністратора (наповнювача сайту).

Висновки. Розроблено програмне забезпечення, що реалізує онлайн-магазин годинників.

Галузь використання – користувачі Інтернету та покупці годинників.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 88 pages, 42 figures, 4 tables, 4 appendixes, 19 sources.

ONLINE STORE, E-COMMERCE, DEVELOPMENT,
REQUIREMENTS ANALYSIS, ADAPTABILITY, DESIGN, WEB
APPLICATION.

The object of research is the process of developing and supporting online stores.

The subject of the research is methods of developing and supporting online stores.

The purpose of this work is to programmatically implement an online store for selling watches.

Materials, methods, and technical tools: Object-Oriented Programming, PHP programming language, Magento platform, MySQL database management system.

Results. An online Watch Store has been developed in the form of a web application that implements the functions of communication and selling and buying photos.

The practical value of the work lies in the development of an online Watch sales store in the form of a cross-platform web application that implements convenient work panels for users (buyers) and the administrator (site filler).

Conclusions. Software has been developed that implements an online Watch Store.

Field of usage: Internet users and watch buyers.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Актуальність використання онлайн-магазинів.....	11
1.1.1 Головні тренди в розробці онлайн-магазинів.....	12
1.2 Дослідження існуючих аналогів	14
1.2.1 Онлайн-магазин «Дека».....	14
1.2.2 Онлайн-магазин Бест-Тайм	15
1.2.3 Онлайн-магазин Секунда.....	16
1.2.4 Порівняння існуючих аналогів	17
1.3 Постановка завдання	19
1.3.1 Функціональність системи	19
1.3.2 Вимоги до інтерфейсу.....	20
1.3.3 Не функціональні вимоги до продукту	20
1.3.4 Усвідомлення середовища користувача	21
1.3.5 Оцінка надійності, продуктивності і підтримки	21
1.3 Висновки за розділом 1	22
2 Розробка архітектури програми.....	23
2.1 Вибір програмної оболонки для програмної реалізації.....	23
2.1.1 CMS для створення онлайн-магазину	23
2.1.2 Фреймворк для онлайн-магазину	25
2.1.3 Вибір між фреймворком та CMS	26
2.1.4 Вибір CMS для розробки онлайн-магазину.....	28
2.2 Загальна структура додатку.....	30
2.3 Висновки за розділом 2	32
3 Основні рішення щодо реалізації компонентів системи.....	33
3.1 Проектування системи на основі предметної області	33
3.2 Висновки за розділом 3	47

4	Експлуатація, тестування та експериментальне дослідження програми	48
4.1	Опис застосування.....	48
4.2	Умови виконання програми.....	48
4.3	Інструкція по експлуатації програми	49
4.3.1	Звернення до програми.....	49
4.3.2	Вхідні й вихідні дані	49
4.4	Методика та результати тестування	49
	Висновки	54
	Перелік джерел посилання	56
	Додаток А Технічне завдання	58
	Додаток Б Опис програми	64
	Додаток В Текст програми	69
	Додаток Д Слайди презентації.....	81

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CMS – система управління вмістом;

GUI – графічний інтерфейс користувача;

MVC – Model-view-controller;

UML – Unified Modeling Language;

БД – база даних;

ОС – операційна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

ТЗ – технічне завдання.

ВСТУП

На даний час дуже поширеним являється розробка сайтів, блогів, онлайн-магазинів та інших веб-застосунків.

Тому було обрано розробити онлайн-магазин годинників. Важливо правильно з'ясувати потреби, проаналізувати технічне завдання та вимоги, розробити архітектуру сайту, обрати програмні засоби, підключити необхідні модулі для створення онлайн-магазину.

Можна визначити основні переваги онлайн-магазину.

Мінімальні постійні витрати – основний плюс в цьому бізнесі. Замовник оплачує тільки хостинг і доменне ім'я, не потрібно платити орендну плату, яка, як всі знають, дуже висока. На відміну від звичайних магазинів, не потрібно платити зарплату продавцям. До наведеного можна додати, що немає необхідності тримати товар на складі – магазин може закуповувати товар відразу після надходження замовлення.

Для утримання онлайн-магазину необхідно всього кілька людей: адміністратор сайту, який буде приймати і обробляти замовлення, що надійшли, а також вести бухгалтерію і кур'єр – для доставки товару. На початку багато підприємців ці посади поєднують. Надалі штат можна збільшити, додавши посаду бухгалтера і контент-менеджера, який займеться вмістом сайту і його просуванням.

Клієнтами магазину можуть бути не тільки жителі конкретного міста або країни, а люди з усього світу, які розуміють мовний контент і мають доступ до мережі Інтернет.

Онлайн-магазин працює цілодобово, сім днів на тиждень. В даний момент в мережі Інтернет з'являється все більше онлайн-магазинів, але перенасичення ринку ще не спостерігається [1]-[2].

Тож завдання створення онлайн-магазину продажу годинників є актуальним, адже дозволяє вирішити кілька завдань одночасно: вплинути на

певну цільову аудиторію, залучити дилерів, почати реалізацію своїх товарів і послуг по всій країні.

Створити онлайн-магазин з привабливим дизайном важливо. Але на багато важливіші зручний інтерфейс і комфортність роботи відвідувача з сайтом. Можливість повідомляти про майбутні акції, знижки та інші заходи, проводити маркетингові дослідження, розробляти і виводити на електронний ринок продукції нові рішення, унікальні торгові пропозиції та багато іншого - незаперечні якості онлайн-магазину. Крім цього, створити онлайн-магазин – це ще й позначити джерело актуальної інформації про асортимент товару, його наявність. Зайти на сайт набагато простіше, ніж дзвонити або тим більше їхати в фізичний магазин (якщо він звичайно діє в умовах карантинних обмежень, наприклад). Діалоги покупців і представників фірми реалізувалися завдяки електронному маркетингу. Швидкість і легкість оформлення замовлень з використанням особистого кабінету клієнта, Можливість створення цілих інформаційних порталів розширює можливості бізнесу як ніколи раніше.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність використання онлайн-магазинів

В даний час онлайн-магазини користуються популярністю, як і у звичайних користувачів, так і у фірм, які бажають купити або ж продати свій товар. Завдання звичайного користувача полягає в перегляді фотографій і характеристик товару, порівнянні його цін, і можливій покупці [1]-[2].

Що стосується ролі онлайн-магазинів, то їх продавці повинні залучити покупців, можливо, навіть знайти постійних покупців.

Отже, давайте розглянемо ситуацію створення онлайн-магазину з різних позицій. Покупці бувають різні, і їх головні цілі полягають ось у чому [1]-[2]:

- у пошуку необхідної інформації, здійсненні замовлення в Інтернеті;
- у конкуренції з вами, порівнянні цін;
- у пошуку характеристик і надійного продавця;
- у пошуку інформації та замовленні в режимі офлайн.

Таким чином, абсолютно всі хочуть отримати інформацію про той чи інший товар, а ось подальші кроки у всіх різні. Тому створення онлайн-магазинів – це не просто якась дія по створенню спеціалізованого сайту для продажів, це робота на перспективу і зі строго певним задумом, який буде в подальшому з успіхом реалізовуватися, орієнтуючись на різні групи клієнтів [1]-[2].

Однак продавців теж буває багато, і у всіх різні цілі [1], [3]-[5]:

- створюють онлайн-магазини тільки для того, щоб підвищити рівень основних клієнтів в звичайному магазині. такий вчинок є способом реклами, просування основного магазину;
- роблять онлайн-магазин в якості основного, не мають приміщень для звичайного магазину;
- надають тільки оптові продажі для великих магазинів і фірм;

– створення магазину, в якому клієнти будуть одиничними споживачами.

Отже, всі головні цілі продавця-це продаж товару. У всіх є свої підходи до торгівлі, і вибір того чи іншого виду торгівлі залежить від самих клієнтів [3]-[5].

1.1.1 Головні тренди в розробці онлайн-магазинів

Питання створення онлайн-магазину пов'язане з багатьма обставинами. Крім первинних розрахунків, складання плану дій, підбору постачальників товарів, вибору розробників сайту і багатьох інших аспектів, починаючи підприємцю необхідно враховувати і поточні вимоги ринку. Одним з найважливіших питань розробки онлайн-магазину є відповідність його технічних характеристик існуючим трендам. Ринок eCommerce щорічно зростає приблизно на 20% [1].

Це говорить про те, що в світі стає все більше бажаючих створити онлайн-магазин з нуля. Звичайно ж, можна знайти масу відповідей на питання про те з чого варто починати розробку онлайн-магазину [4]-[5]:

- як правильно вибрати підрядника для створення онлайн-магазину;
- на які технічні вимоги потрібно звертати увагу до початку робіт зі створення інтернет-магазину;
- які програмні засоби (ПЗ) та платформу для eCommerce краще вибрати.

Говорячи про план дій по створенню онлайн-магазину, може бути кілька правильних послідовностей. Кожен фахівець, скоріше за все, матиме власну точку зору, спираючись на попередній досвід [4]-[5].

Більшість трендів в розробці онлайн-магазинів розробники ПЗ структурували за наступними групами: дизайн, спрощення, узагальнення та функціонал. Розглянемо ці групи детальніше [5]-[7].

Дизайн. З продовженням зростання мобільного трафіку покупці все більше звикають до інтерфейсів мобільних версій сайтів і мобільних додатків [5]-[7].

Спрощення. Тому тренд в десктопному дизайні магазинів – це спрощення сайту і приведення його інтерфейсу за сценарієм мобільних версій сайтів: групування декількох елементів дизайну в одну загальну групу (наприклад, 5-10 посилань у верхньому меню тепер будуть трансформуватися в одну кнопку «меню», довге поле пошуку перетвориться просто на іконку пошуку, те ж чекає і футер сайтів і т.д.), видалення багатьох зайвих елементів, які були обов'язковими 5 років тому (наприклад, у списку товарів – наявність, порівняння, «лайки», коментарі і т. д., в картці товару – купа тексту про способи доставки і всі необов'язкові блоки). По суті, триває еволюція сайтів у бік інстаграмності і твітовості: назва товару, велика фотографія, трохи тексту, і соціальний зворотній зв'язок – коментарі та вподобання (лайки, зірочки рейтингу, тощо). Так само продовжує вкорінюватися в свідомості користувачів банерна сліпота, тому банерів і рекомендованих товарів в магазині буде все менше, тому що їх показник буде невиправдано низький [5]-[7].

Узагальнення. Крім того – загальна тенденція всіх магазинів: привести інтерфейс користувача до якогось однакового вигляду, щоб користувач, зайшовши перший раз на сайт магазину вже інтуїтивно розумів, де що повинно знаходитися, і не витрачав час на пошуки потрібних кнопок і посилань. Тому маркетплейси продовжують завойовувати нішу eCommerce – у них однаковий інтерфейс для тисяч магазинів [1], [5]-[7].

Функціонал. Що стосується функціоналу – бажаючий вижити магазин буде зобов'язаний розвивати багатоканальність – використовувати всі можливі канали продажів і реклами. Тому буде зростати необхідність створення модулів інтеграцій зі сторонніми сервісами і майданчиками (вивантаження в маркетплейси, каталоги цін, в соціальних мережах і т.д.). Ну і природно продовження інтеграції з сервісами оплати і доставки [5]-[7].

1.2 Дослідження існуючих аналогів

На сьогоднішній день на просторах Інтернету існує достатньо багато онлайн-магазинів, що у якості товарів пропонують годинники. Проте слід зазначити, що більшість з них є майже клонами один одного [8]-[10].

Дійсно унікальних та професійних магазинів не так багато. Їх виокремлює унікальний дизайн, широкий асортимент товарів та певна брендованість. Це проявляється в багатьох деталях: шрифти, корпоративні кольори та навіть представлення інформації про товар [8]-[10].

Під час аналізу існуючих аналогів зосередимося саме на таких онлайн-магазинах.

1.2.1 Онлайн-магазин «Дека»

Перевагою даного інтернет-магазину є те що товари можна придбати на замовлення, чекаючи деякий час, або у будь-якому з магазинів по місту, оскільки дана мережа магазинів є у більшості великих міст України [8].

Магазини годинників ДЕКА завдяки своєму асортименту, вигідних цін та високому рівню обслуговування клієнтів завоювали високу репутацію не тільки серед жителів Донецька та інших міст, але всіх шанувальників інтернет-торгівлі [8]. Сьогодні в асортименті ДЕКА можна знайти колекції відомих японських, німецьких, швейцарських та інших часових марок. На даний момент в магазинах ДЕКА представлені годинники понад 40 відомих годинних виробників. В даному інтернет-магазині можна придбати годинники CASIO, які доступні будь-якому відвідувачу, а також престижні годинники Longines або Breitling, які можуть собі дозволити досить забезпечені покупці [8].

На рисунку 1.1 наведено приклад головної сторінки онлайн-магазину ДЕКА.

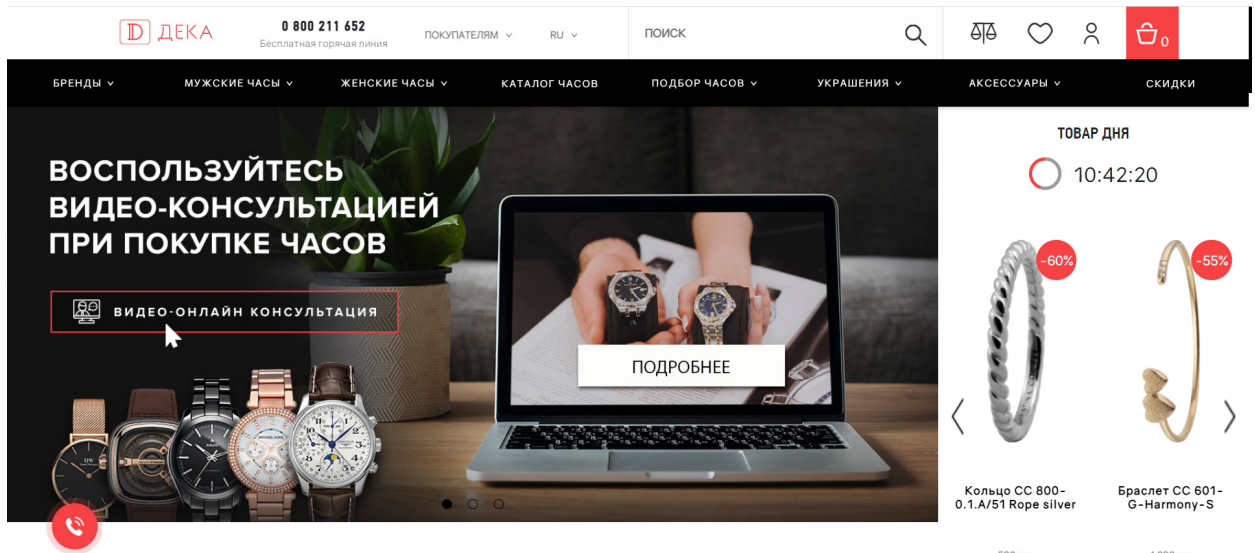


Рисунок 1.1 – Головна сторінка онлайн-магазину Дека

1.2.2 Онлайн-магазин Бест-Тайм

Компанія Бест-Тайм була заснована в 2011 році. За цей час вони змогли завоювати довіру наших клієнтів високою якістю послуг і справедливими цінами [9]. Основною спеціалізацією компанії є реалізація годинників різних марок, оптом і в роздріб. Компанія працює тільки з постачальниками якісних годинників. Всі годинники поставляються безпосередньо з заводу, що забезпечує кращі ціни і високу якість на нашу продукцію. Якість продукції підтверджена сертифікатами [9]. У травні 2013 року був відкритий перший магазин Бест-Тайм в Києві. Так у київлян з'явилася можливість подивитися всі моделі наживо, приміряти їх і придбати за тими ж цінами, що і на сайті. У жовтні 2013 магазин і сервісний центр Бест-Тайм були відкриті в Дніпропетровську. А в травні 2016 року був відкритий магазин і сервісний центр Бест-Тайм в Харкові [9].

На рисунку 1.2 наведено приклад сторінки магазину Бест-Тайм.

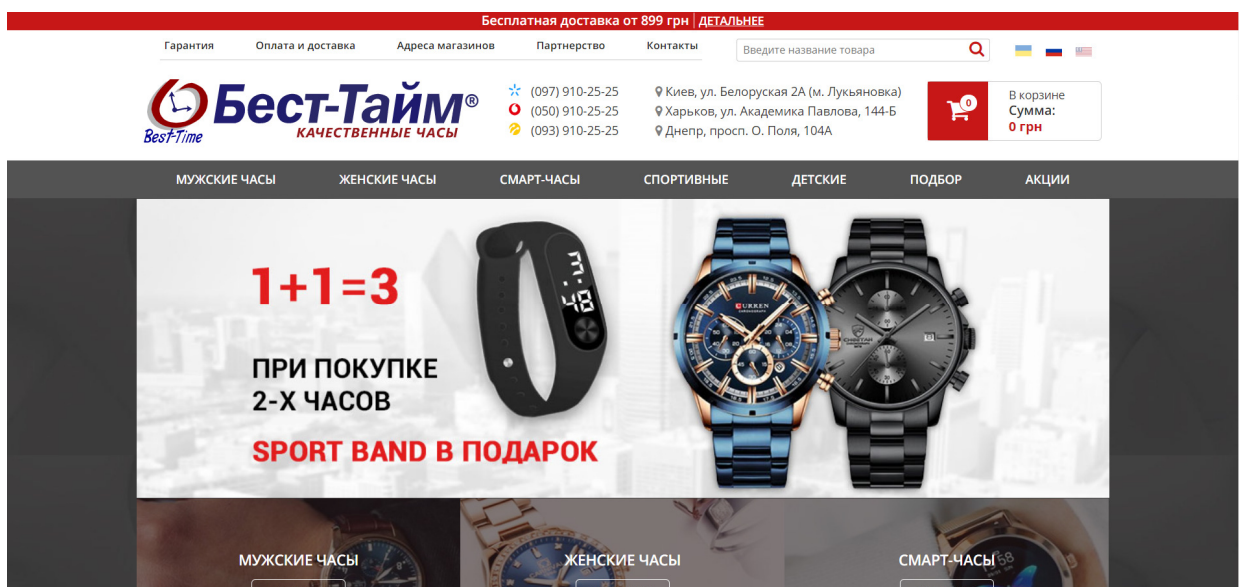


Рисунок 1.2 – Приклад сторінки Бест-Тайм

1.2.3 Онлайн-магазин Секунда

На сьогоднішній день мережа годинникових магазинів «Секунда» об'єднує 32 магазини в 14 містах України [10]. У вітринах магазинів представлені більше 30-ти швейцарських, японських, американських, французьких, австрійських, німецьких та інших європейських годинникових брендів. Свою історію компанія «Мікроприлад» починає з 1992 року. 15 березня 1992 року в Києві було відкрито перший магазин годинників «Casio». У 1996 році компанія домоглася права на дистрибуцію всесвітньо відомого бренду «Casio». Перші роздрібні магазини працювали і стали відомі всім під вивіскою «Casio». У тому ж, 1996, році відкрився авторизований Сервісний центр [10]. У 1997 році компанія отримує дистрибуцію по брендам Citizen і Hermle. У 2000 році вперше звучить нова назва торгової мережі – «Секунда» – компанія реєструє торговий знак і відкриває шість годинних магазинів у Києві [10]. З 2001 року почалася експансія компанії в регіони. Торгова мережа «Секунда» успішно розвивається по всій території України – магазини відкриваються в Черкасах, Львові, Миколаєві, Одесі. Одночасно розширюється брендовий портфель, до звичного і знайомого Casio приєднуються нові японські бренди Seiko і Q&Q. В 2002 році, компанія

отримує дистрибуцію по ще ряду брендів. До 2006 року компанія «Мікроприлад» відкриває 20 годинних магазинів. У листопаді 2007 року відривається інтернет-магазин «Секунда» [10]. У 2007-2008 роках компанія активно розвивається: торгова мережа «Секунда» складається вже з 55 магазинів. У 2009 році компанія проводить оптимізацію бізнесу, що дозволило вистояти в умовах кризи [10].

На рисунку 1.3 наведено приклад сторінки магазину Секунда.

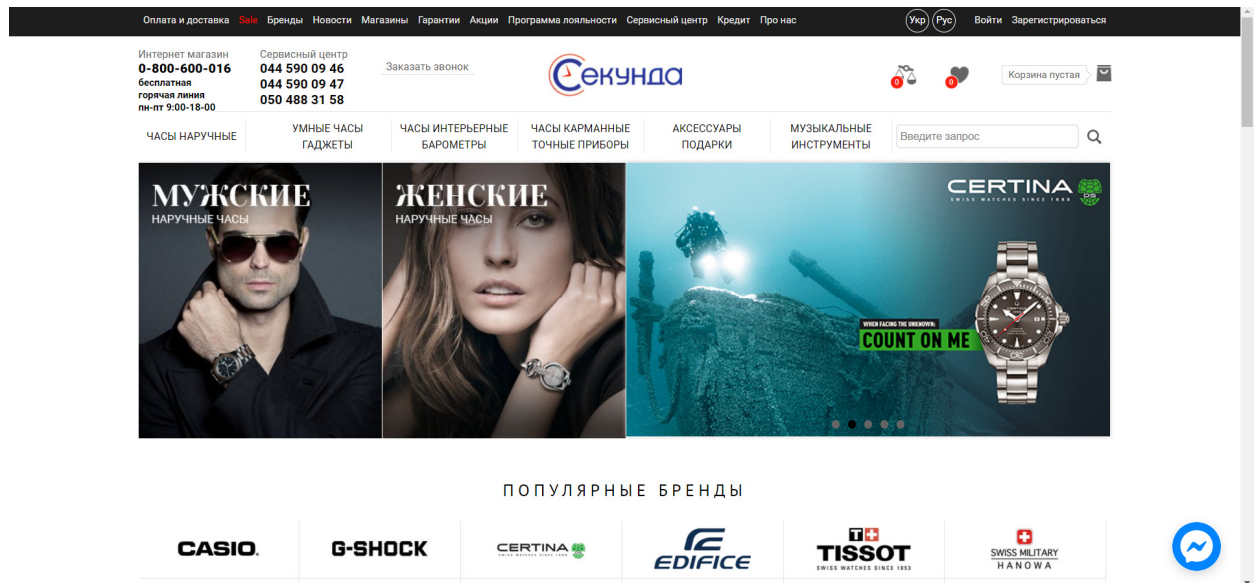


Рисунок 1.3 – Приклад сторінки магазину Секунда

1.2.4 Порівняння існуючих аналогів

Проведемо порівняння за певними критеріями, хід якого представимо у вигляді таблиці (табл. 1.1). У таблиці будуть порівнюватися попередні аналоги із соціальною мережею, що розробляється – Focus. Серед критеріїв виділимо наступні:

- брендованість онлайн-магазину – загальне враження від онлайн-магазину (його оформлення та представлення товарів). Чи має воно завершений вигляд;
- каталог магазину – зрозумілість розділів каталогу та їх представлення;

- оптимізація онлайн-магазину – як швидко система реагує на дії користувача, на скільки вона оптимізована під різні браузері та апаратні платформи;
- зрозумілість дизайну та верстки – загальне враження від логіки дизайну веб-застосунку та його верстки.

Таблиця 1.1 – Порівняння існуючих аналогів

Критерій порівняння	ДЕКА	Бест-Тайм	Секунда	Хронос
Брендованість онлайн-магазину	Високий рівень	Низький рівень	Високий рівень	Високий рівень
Каталог магазину	Високий рівень	Низький рівень	Високий рівень	Високий рівень
Оптимізація онлайн-магазину	Низький рівень	Середній рівень	Високий рівень	Високий рівень
Зрозумілість дизайну та верстки	Середній рівень	Низький рівень	Середній рівень	Високий рівень

З результатів порівняння можна зробити наступні висновки. По-перше, певний рівень брендованості, що визначається і преміальним дизайном, і використанням корпоративного підходу в оформленні та представленні інформації відрізняються виключно онлайн-магазини, що є в певній мірі представниками звичайних магазинів та використовують Інтернет-середовище виключно як додаткову платформу. Також є ряд проблем із дизайном: так оформлення магазину «Бест-Тайм» в деяких розділах має темний фон та темний колір текстової інформації, що значно ускладнює її сприйняття. Також Магазин Секунда має занадто багато інформації в шапці сайту (що є статичною на всіх сторінках та розділах) та на мініатюрах в розділі «Каталог» – що значно відволікає при здійсненні вибору. ДЕКА вигідно відрізняється від інших аналогів раціональним мінімалізмом та дійсно брендовим підходом, проте має значні проблеми саме

з оптимізацією. Так навіть десктопна версія сайту працює повільно, а от мобільна взагалі має проблеми із адаптивністю та часом відклику.

Тож актуальною залишається задача розробки та підтримки нового онлайн-магазину продажу годинників, який би відрізнявся інтуїтивно-зрозумілим інтерфейсом користувача, високим рівнем адаптивності та зручною навігацією і інструментарієм як для користувача, так і для адміністратора.

1.3 Постановка завдання

За результатами аналізу проблемної області та дослідження існуючих аналогів для вирішення проблемних питань, сформулюємо постановку задачі.

Метою є розробка нового онлайн-магазину годинників «Хронос». Система повинна забезпечувати можливості зручних:

- перегляду каталогу;
- вибору товарів;
- оформлення заказів;
- поповнення асортименту;
- опрацювання заказів.

Повинна бути кроссплатформною. Система також повинна бути простою й зрозумілою користувачу.

Розглянемо вимоги до розроблюваної системи за їх типом [11]-[12].

1.3.1 Функціональність системи

Межі системи: система повинна працювати постійно без збою. У системі повинні бути реалізовані всі функції зазначені нижче.

Функціональні вимоги:

- перегляд товарів;

- реєстрація користувачів;
- додавання товарів до кошику;
- видалення товарів з кошика;
- оформлення замовлення;
- фільтрація товарів за певним критерієм;
- додавання коментарів до товарів;
- вибір способу доставки та оплати товарів;
- вибір товарів за категоріями;
- пошук товарів;
- порівняння товарів.

1.3.2 Вимоги до інтерфейсу

Для забезпечення ергономічності, зручності та зрозумілості інтерфейсу розроблюваного веб-застосунку повинен бути зручним та інтуїтивно-зрозумілим.

1.3.3 Не функціональні вимоги до продукту

Мінімальні системні вимоги та програмне забезпечення, що необхідні для функціонування серверної частини мобільного застосунку:

- серверна операційна системи (ОС) – будь-яка з різновидів операційної системи Unix (включаючи Mac OS X) або Windows;
- веб-сервер – HTTP-сервер Apache для Unix і Windows, сервер на кшталт XAMPP;
- інтерпретатор PHP5 та PHP7;
- MySQL-сервер із правами створення та вилучення користувачів бази даних (БД) (або база SQLite з аналогічними правами).

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- персональний комп'ютер (ПК);
- процесор із тактовою частотою від 1,5 GHz;
- обсяг оперативної пам'яті – не менше 3GB;
- підключення до мережі Internet;
- встановлений веб-оглядач (браузер) останньої версії на рушії Chromium (Google Chrome), Gecko (Mozilla Firefox) або інших (Opera, Microsoft Edge).

1.3.4 Усвідомлення середовища користувача

Середовище користувача, що використовуватиме дану систему можна описати такими вимогами:

- систему будуть використовувати люди, що цікавляться купівлею-продажем годинників з використанням Інтернету;
- рівень освіти може бути будь-яким, як мінімум, початкова шкільна програма;
- звичайні користувачі, що не мають спеціальної комп'ютерної освіти, володіють комп'ютером на початковому рівні;
- користувачі не обов'язково повинні мати досвід роботи з даними програмами;
- система планує бути кроссплатформною;
- продукт повинен підняти рейтинг інтернет-магазину, та забезпечити йому прибуток.

1.3.5 Оцінка надійності, продуктивності і підтримки

Для забезпечення достатньої надійності та продуктивності система має відповідати певним вимогам:

- система повинна працювати постійно без збою;
- займатися підтримкою системи повинен адміністратор та модератор;
- кожен користувач системи повинен мати ідентифікований логін та пароль;
- бажано щоб програма працювала без додаткового налаштування;
- дана система безкоштовна в межах компанії.

1.3 Висновки за розділом 1

Виконано огляд проблеми розробки та підтримки онлайн-магазинів. Виділено основні шляхи для направи для розроблення нового ПЗ у вигляді веб-застосунку.

Розробка та підтримка онлайн-магазину годинників «Хронос» має надати систему, що забезпечує можливості зручних перегляду каталогу, вибору товарів, оформлення заказів, поповнення асортименту, опрацювання заказів. Система повинна бути кроссплатформною. Система також повинна бути простою й зрозумілою користувачу.

2 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

У попередньому розділі було проведено аналіз та дослідження існуючих аналогів онлайн-магазинів для продажу годинників. Виходячи з проведеного аналізу, було сформульовано основні можливості, якими повинна оперувати система, що проєктується.

Що стосується технічних та програмних можливостей, то система повинна забезпечувати можливість швидко здійснювати замовлення необхідного товару, з урахуванням всіх нюансів.

Даний розділ містить опис питань, які стосуються безпосередньо структури системи, що розробляється, вибору середовища проєктування, а також визначаються основні та специфічні архітектурні рішення, які має система.

2.1 Вибір програмної оболонки для програмної реалізації

Реалізація проєктів, що використовуються для комерції вкрай складна, адже необхідно враховувати масу технічних нюансів, адаптуватися під умови ринку, постійно взаємодіяти з аудиторією потенційних клієнтів і т.д. у числі ключових моментів при створенні онлайн-магазину вибір між системами керування вмістом (CMS) і фреймворками [13]-[14]. Необхідно своєчасно визначитися з програмною платформою, яка буде використовуватися для забезпечення стабільної та ефективної роботи комерційного сайту [13]-[14]. Розглянемо вибір в деталях.

2.1.1 CMS для створення онлайн-магазину

CMS – це програмне забезпечення, яке дозволяє створювати, редагувати і управляти контентом сайту без спеціальних знань в програмуванні [13]-[14].

Прикладами популярних CMS є: Bitrix, Magento, Wordpress і д.р. Ці системи являють собою готові шаблони з певним набором функцій для майбутніх онлайн-магазинів [13]-[14].

Кожна система спочатку пропонує набір функцій, необхідних для створення хорошого онлайн-магазину. Якщо щодо проєкту висуваються стандартні вимоги, то найчастіше вони задовольняються можливостями використовуваної CMS. Такими проєктами можуть стати онлайн-магазини одягу, квітів, доставки їжі та інші [13]-[14].

Таке рішення підійде для сайту зі стандартними вимогами, які покриваються можливостями CMS. Наприклад, онлайн-магазин жіночого одягу, квітковий бутік, доставка піци [13]-[14].

Переваги використання CMS [13]-[14]:

- структура онлайн-магазину вже створена, що економить час і бюджет на основний функціонал вже передбачений «з коробки»;
- при наявності стандартних вимог результат легко передбачити і прорахувати;
- у будь-який момент можна купити новий шаблон і швидко змінити дизайн сайту;
- технічна підтримка від компанії-розробника CMS (передбачена для деяких CMS, як Bitrix, Magento).

Недоліки використання CMS [13]-[14]:

- відсутня унікальність. Кожен може взяти за основу таку ж тему CMS;
- функціонал обмежений, обраної CMS. Створення Екстра функцій тягне додаткові витрати і не завжди можливо;
- необхідно щорічно продовжувати ліцензію;
- стандартне рішення може містити непотрібний функціонал, який буде обтяжувати роботу ресурсу.

Однак якщо розробкою сайту онлайн-магазину займаються кваліфіковані фахівці, то знання програмного коду і нюансів експлуатації

конкретної CMS компенсує описані недоліки. На місце шаблону може бути розроблений індивідуальний дизайн, весь непотрібний функціонал буде відключений, а необхідні модулі – вбудовані в роботу проєкту. Природно, вартість створення онлайн-магазину під ключ збільшиться в порівнянні з запуском шаблонного ресурсу, але і комерційна перспектива покращиться [13]-[14].

2.1.2 Фреймворк для онлайн-магазину

Фреймворк – це ПЗ, що полегшує розробку та об'єднання різних модулів програмного проєкту [13]-[14].

У них немає готового функціоналу, та він і не потрібен, тому що призначені такі платформи для індивідуального створення онлайн-магазину з нуля. Унікальні проєкти мають свої нетипові структурні елементи. Найбільш актуальне використання фреймворку для компаній, що продають специфічну продукцію, реалізують нестандартні пропозиції, здійснюють облік і формують схеми ведення бізнесу [13]-[14].

Популярні фреймворки: Yii, Laravel, Symfony і ін. Вони не містять готового функціоналу сайту [13]-[14].

Таке рішення підходить для компаній, що займаються специфічними товарами або послугами, з нестандартними торговими пропозиціями, системами обліку та схемами ведення бізнесу. Для сайтів з високою відвідуваністю і навантаженням. Для проєкту з нетиповим рішенням [13]-[14].

Переваги використання фреймворку [13]-[14]:

- дозволяє реалізувати всі задумки замовника без будь-яких обмежень;
- дозволяє створити унікальний за функціоналом проєкт;
- додаючи новий функціонал, будь-який проєкт можна масштабувати з мінімальними втратами продуктивності;

- сайти, розроблені на фреймворках, працюють швидше своїх аналогів на CMS;

- можна повністю виключити невикористовуваний функціонал;

Недоліки використання фреймворків [13]-[14]:

- етап розробки як правило займає більше часу, ніж у випадку з CMS, так як технологія має на увазі створення як клієнтської частини, так і повної адміністративної частини;

- необхідні висококваліфіковані фахівці, тому що розробка складніше, ніж створення онлайн-магазину на CMS;

- всі доопрацювання необхідно замовляти у розробників, не можна впровадити самостійно;

- розробка функціоналу, стандартного для сайтів на CMS, обійдеться дорожче і займе більше часу.

Фактично, якщо виконувати розробку онлайн-магазину під ключ, то проєкт на базі фреймворку буде саме таким, яким його задумав замовник. Однак для цього доведеться звернутися до послуг професійної web-студії. Створення онлайн-магазину, згідно з усіма правилами, буде виконано в оптимальні терміни і на вигідних умовах [13]-[14].

Однак при створенні онлайн-магазину ціна зросте, а працювати над будь-якими правками і доопрацюваннями доведеться з кваліфікованим фахівцем, бажано тим же, який спочатку розробляв весь проєкт. Перш ніж думати, де краще замовити онлайн-магазин, визначитеся з особливостями і потребами вашого бізнесу [13]-[14].

2.1.3 Вибір між фреймворком та CMS

Для полегшення вибору проведемо порівняння використання фреймворку та CMS під потреби замовника.

Результати порівняння представимо в табличному вигляді – таблиця 2.1. У якості критеріїв виділемо:

- зрозумілий та простий графічного інтерфейсу користувача (GUI) – можливість та швидкість розроблення простого GUI;
- підключення платіжних систем – простота та можливість автоматизованого підключення платіжних систем до застосунку;
- адміністративна панель – наявність механізмів та простота розробки адміністративної панелі;
- простота та швидкість розробки – загальна швидкість розробки платформи для онлайн комерції;
- простота підтримки – простота підтримки системи після розгортання.

Таблиця 2.1 – Порівняння використання фреймворку та CMS під завдання замовника

	Фреймворк	CMS
Зрозумілий та простий GUI	++	+
Підключення платіжних систем	++	++
Адміністративна панель	+	++
Простота та швидкість розробки	+	++
Простота підтримки	+	++

Виходячи з порівняння розробка з використанням CMS буде більш швидкою, не буде потребувати додаткових витрат ресурсів часу на розробку адміністративної панелі. Більш того така розробка буде більш простою в подальшій підтримці, а за рахунок професіонального підходу GUI вдасться наблизити до унікального та корпоративного брендингу [13]-[14].

2.1.4 Вибір CMS для розробки онлайн-магазину

CMS Magento – це рушій для електронної комерції з відкритим вихідним кодом. Дуже популярний на західному ринку, де використовується в тому числі для створення сайтів міжнародних брендів [15]-[17].

Magento – це рішення для створення великих проєктів. Використовувати рушій для запуску магазину з сотнею товарів неефективно: його важко освоїти, а вартість розробки помітно вище в порівнянні з іншими популярними CMS. Особливо це відноситься до українського Інтернет-простору, де до всього іншого додаються проблеми з документацією та інтеграціями з сервісами електронної комерції [15]-[17].

Сильною стороною системи є велика кількість вбудованих інструментів для управління онлайн-торгівлею. Однак недоліки в деяких випадках виявляються настільки значними, що повністю нівелюють гідності Magento. Але щоб зрозуміти, для чого CMS можна використовувати, спочатку необхідно познайомитися з нею ближче [15]-[17].

Після встановлення Magento доступна тема за замовченням. Проте, в решті решт, головна функція стандартного шаблону-демонстрація можливостей рушія, а не постійна робота в режимі продажів [15]-[17].

Тема Magento, що визначає зовнішній вигляд сайту, включає три компоненти [15]-[17]:

- макети, що визначають конструкцію;
- шаблони, що складаються з суміші PHP і HTML, що відповідають за вигляд окремих елементів;
- скіни, які забезпечують відображення кольорів, шрифтів, зображень, встановлюють правила розташування елементів — простіше кажучи, формують дизайн сторінки.

Magento – гнучка система, яка в базовій комплектації пропонує інструменти, доступні на інших CMS тільки після установки плагінів. Тому її вибір для створення магазину виправданий. На одному движку можна

запустити кілька сайтів. Вони будуть мати загальну базу даних і управлятися з єдиного інтерфейсу [15]-[17].

Magento пропонує всі необхідні інструменти для управління магазином. Не будемо зупинятися на базових можливостях на кшталт додавання товарів і редагування їх описів – це можна робити на будь-який CMS. Куди цікавіше подивитися на менш поширені функції [15]-[17]:

- зручна робота з цінами. Можна тимчасово знизити вартість товарів, запустити акцію, налаштувати знижкову або бонусну систему;
- розмежування прав користувачів, що дозволяє надати співробітникам магазину тільки ті можливості, які їм потрібні для роботи.

Вбудована система оцінювання товарів і відправки відгуків. Це дуже цінують відвідувачі сайту, які на основі рейтингів і коментарів інших покупців роблять вибір [15]-[17].

Широкі можливості для аналізу діяльності магазину і надання самих різних звітів про поведінку покупців, популярності товарів, привабливості сторінок і т.п. [15]-[17].

Magento за замовчуванням оптимізований під вимоги пошукових систем і в базовій функціональності пропонує інструменти для додавання мета-тегів, налаштування параметрів пошукової оптимізації (SEO) [15]-[17]. Для кожного товару доступні два види опису, які ґрунтуються на його унікальних характеристиках. Мета-теги автоматично не генеруються – виправити це можна за допомогою стороннього модуля або шаблону [15]-[17].

У таблиці 2.2 наведено порівняння CMS за критеріями. Серед характеристик для порівняння виділімо:

- вбудована система кешування – для пришвидшення роботи застосунку;
- плагіни та розширення для електронної комерції – кількість та простота встановлення плагінів для електронної комерції;

- система керування правами користувачів – механізми розгалуження прав та рівнів доступу користувачів;
- адаптивність – можливість динамічно адаптувати сторінки до екранів різних приладів;
- SEO-оптимізація – інструменти для налаштування пошукової оптимізації.

Таблиця 2.2 – Порівняння CMS

	OpenCart	Magento
Вбудована система кешування	+	++
Плагіни та розширення для електронної комерції	+	++
Система керування правами користувачів	+	++
Адаптивність	+	++
SEO-оптимізація	+	++

2.2 Загальна структура додатку

Для побудови архітектурного каркасу ПЗ буде використано шаблон проєктування Model-view-controller (MVC) [18]-[19]. Ця схема використання декількох шаблонів проєктування, за допомогою яких модель даних програми, користувацький інтерфейс і взаємодія з користувачем розділені на три окремих компоненти.

View – відповідає за збір даних від користувачів в різних формах та обробку їх у відповідності зі специфікацією.

Controller – обробляє дані, отримані від користувача для передачі на наступний рівень.

Model – сукупність правил, принципів, залежностей поведінки об'єктів предметної області. Це реалізація предметної області в інформаційній системі.

На рисунку 2.1 показано, який вигляд має архітектура ПЗ на основі предметної області з використанням шаблону проектування MVC.

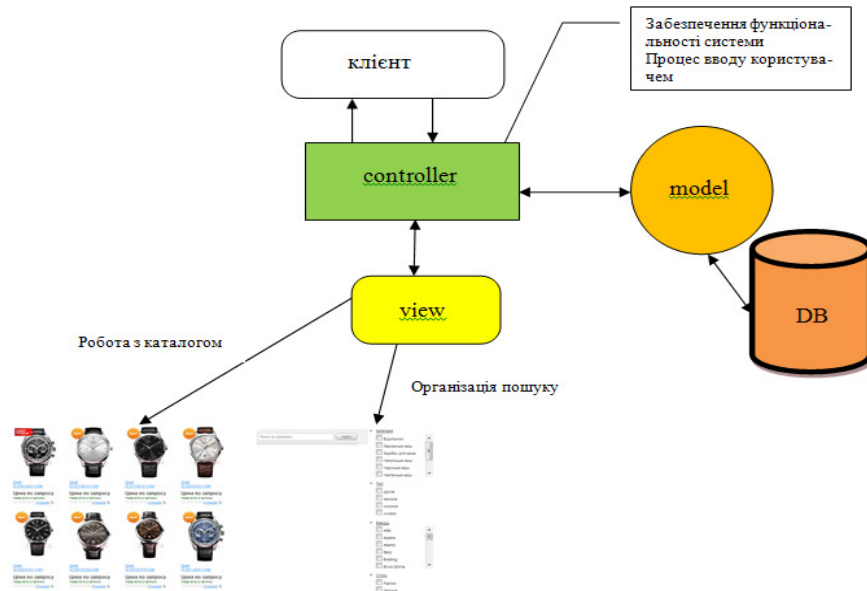


Рисунок 2.1 – MVC схема для користувача

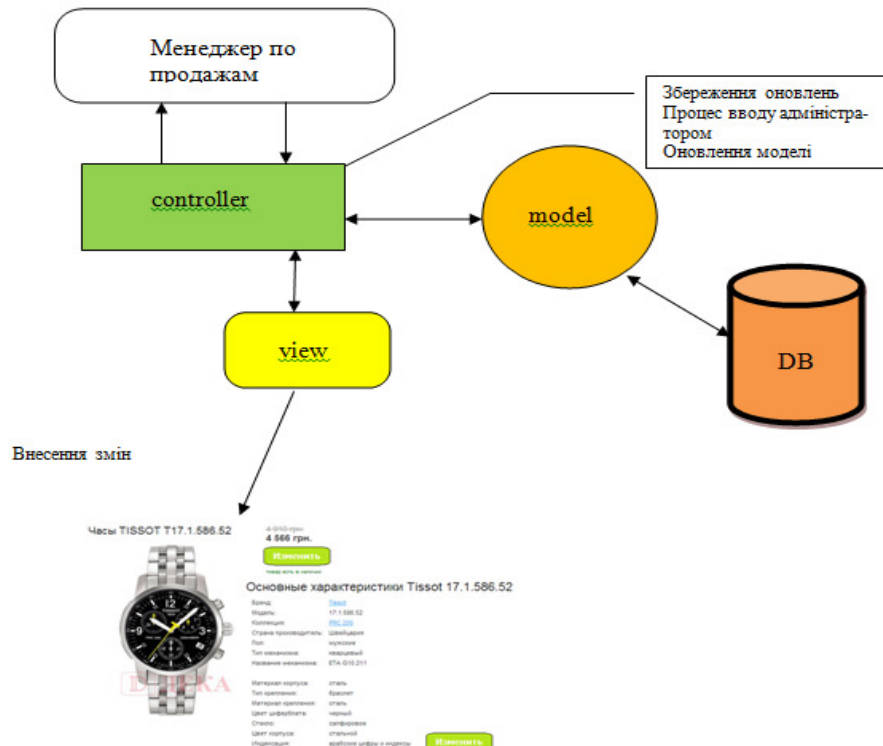


Рисунок 2.2 – MVC схема для менеджера по продажам

2.3 Висновки за розділом 2

Таким чином, архітектура програмного засобу дозволяє вирішувати задачі, що були поставлені перед системою, пов'язані із електронною комерцією. Засоби, що використовуються, забезпечують реалізацію функцій програми.

Для програмної реалізації можливостей ПЗ, що онлайн-магазину продажу годинників, виділених під час аналізу у попередньому розділі, проведено вибір засобів та обґрунтовано структуру системи, що проєктується.

Аналізуючи обрану структуру ПЗ, можна зробити висновок, що вона є лаконічною, але, разом з тим, досить працездатною, використовує ресурси ПК раціонально.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Проектування системи на основі предметної області

Unified Modeling Language (UML) – уніфікована мова моделювання, що використовується у парадигмі об'єктно-орієнтованого програмування та є невід'ємною частиною уніфікованого процесу розроблення ПЗ [18]-[19]. UML заснована на діаграмах, що дають можливість представити систему у такому вигляді, щоб її можна було легко перевести у програмний код.

Діаграми класів – статичні структурні діаграми, що описують структуру системи, демонструють класи систем, їх атрибути, методи і залежності між класами [18]-[19].

Діаграма класів, розроблена для онлайн-магазину «Хронос» має вигляд представлений на рисунку 3.1.

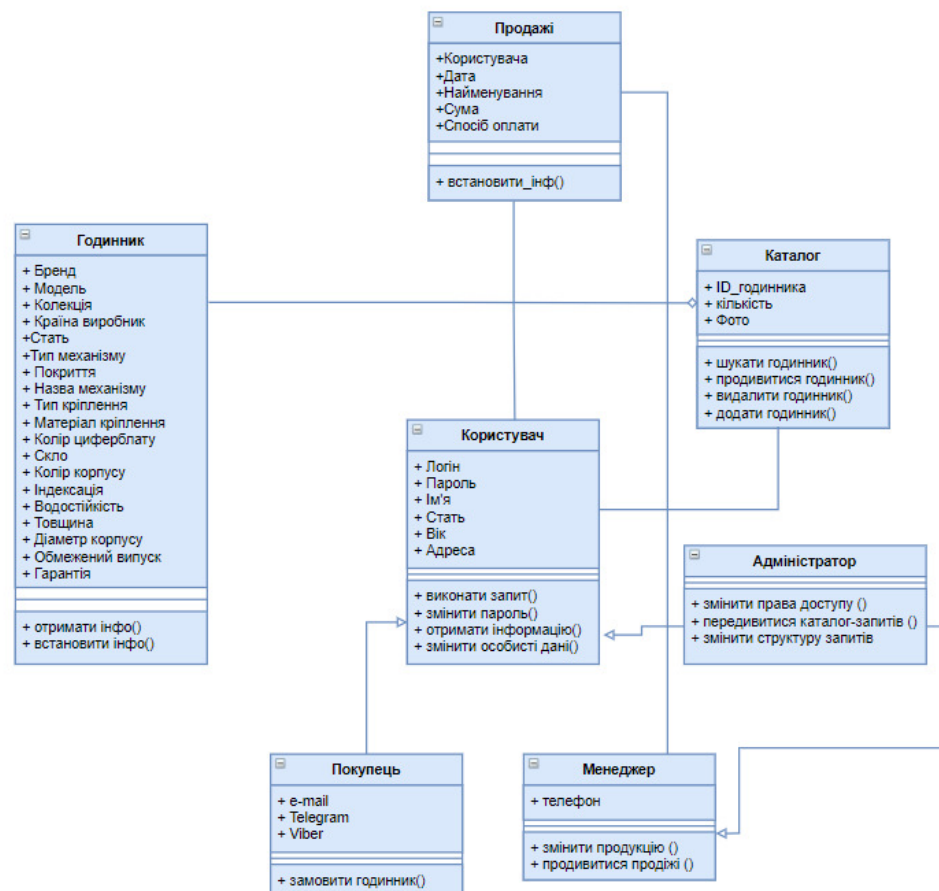


Рисунок 3.1 – Діаграма класів системи

Діаграми пакетів – структурні діаграми, основним вмістом яких є пакети і відношення між ними [18]-[19]. Ці діаграми слугують, в першу чергу, для організації елементів в групі по яким-небудь ознакам з метою спрощення структури і організації роботи з моделлю системи.

Онлайн магазин «Хронос» складається з чотирьох пакетів, для роботи з каталогом, базою даних, адміністративною панеллю, а також для пошуку, що представлені на рисунку 3.2.

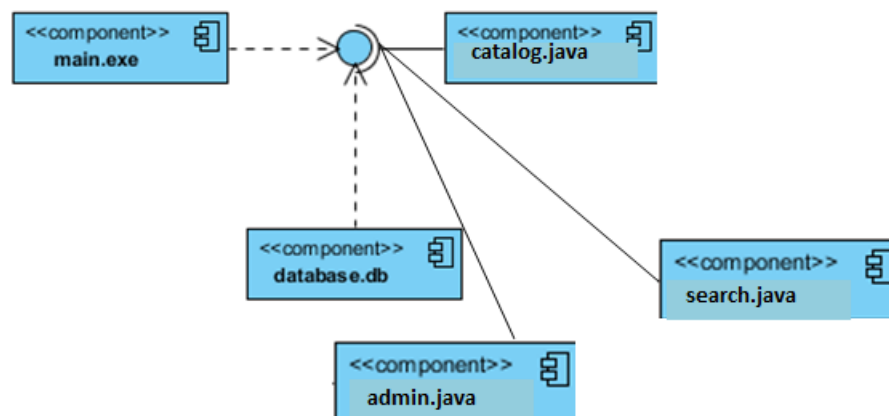


Рисунок 3.2 – Діаграма компонентів системи

Діаграма компонентів – статична структурна діаграма, що показує розбиття програмної системи на структурні компоненти і зв'язки (залежності) між компонентами. В якості фізичних компонент можуть виступати файли, бібліотеки, модулі, виконавчі файли, пакети і т.п. [18]-[19].

На рисунку 3.3 показано, що онлайн-магазин складається з трьох пакетів.

Перший пакет є основним в цій діаграмі, а два інших структурно від нього залежать.

Діаграми прецедентів застосовуються для моделювання вигляду системи з точки зору прецедентів (або варіантів використання). Даний вид діаграм використовується для моделювання динамічних аспектів системи, так само, як і діаграми станів та діяльності [18]-[19].

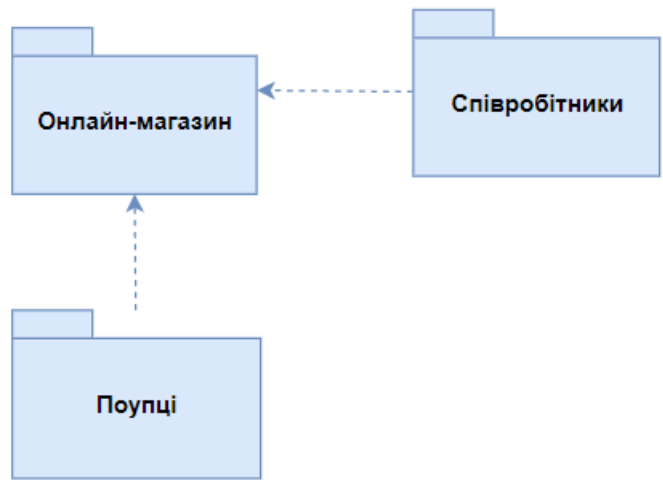


Рисунок 3.3 – Діаграма пакетів системи

На рисунках 3.4 – 3.11 зображено діаграми прецедентів для соціальної мережі «Focus». Для них приведено базові та альтернативні потоки виконання.

Робота з системою

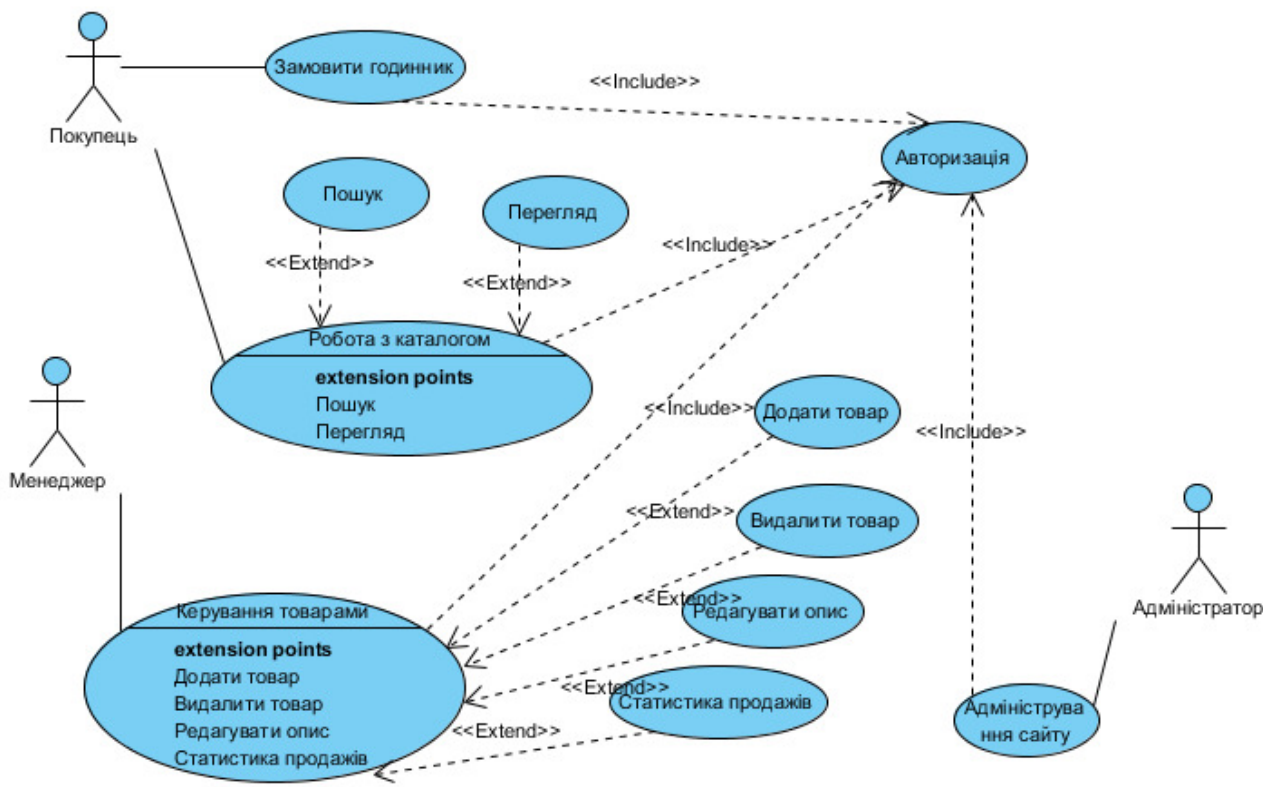


Рисунок 3.4 – Діаграма прецедентів «Робота з системою»

Зареєструвати клієнта – базовий потік:

- активувати систему;
- ввести ім'я;
- ввести прізвище;
- ввести e-mail;
- ввести логін;
- ввести пароль;
- ввести підтвердження пароля;
- підтвердження реєстрації;
- занесення даних у БД.

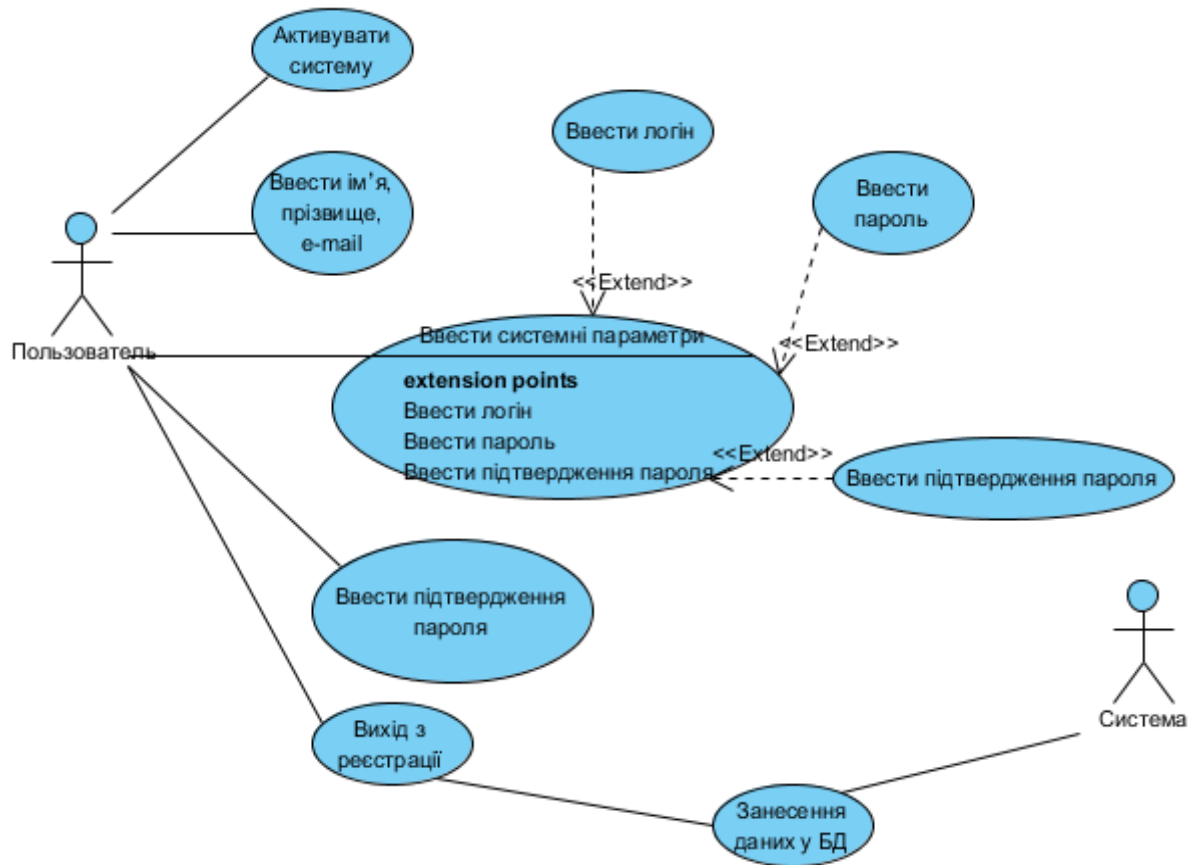


Рисунок 3.5 – Діаграма прецедентів «Зареєструвати клієнта»

Альтернативні потоки:

- якщо введений e-mail не існує;
- вивести повідомлення про помилку у введеному e-mail;

- якщо введений пароль не співпадає з підтвердженням;
- вивести повідомлення про не співпадання паролю;
- якщо введений логін існує;
- вивести повідомлення про помилку введення у полі «логін».

Авторизувати клієнта – базовий потік:

- активувати систему;
- ввести логін;
- ввести пароль;
- підтвердити рішення про авторизацію;
- перевірка наявності даних у БД.

На рисунках 3.6 – 3.11 зображено діаграми прецедентів для соціальної мережі «Focus».

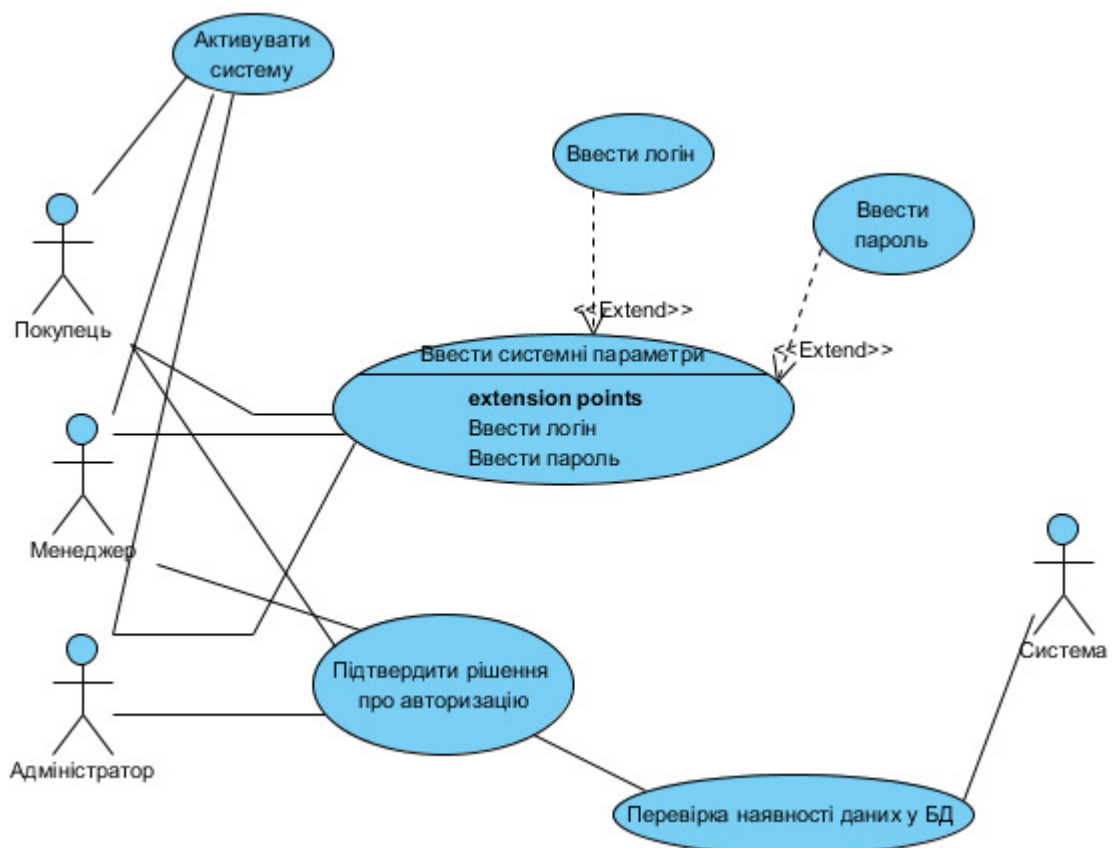


Рисунок 3.6 – Діаграма прецедентів «Авторизувати клієнта»

Альтернативні потоки:

- якщо введений логін не існує;
- вивести повідомлення про помилку у введеному логін;
- якщо введений пароль не існує;
- вивести повідомлення про помилку вводу паролю;
- а також (та) у разі необхідності;
- запропонувати відправити пароль на e-mail.

Переглянути товари – базовий потік:

- активувати систему;
- вибір категорії;
- перегляд.

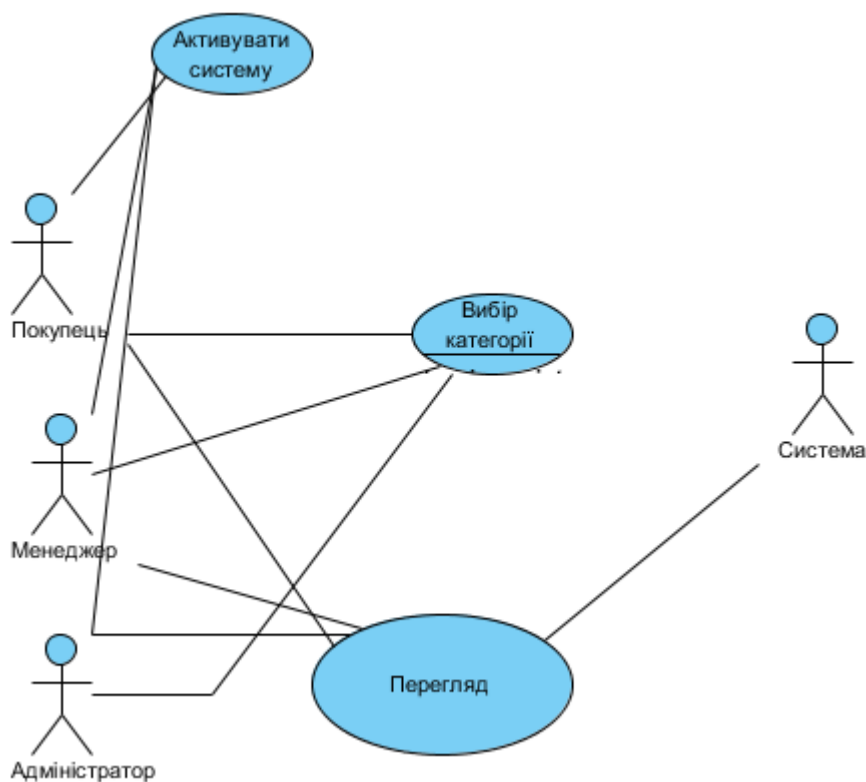


Рисунок 3.7 – Діаграма прецедентів «Переглянути товари»

Знайти товари – базовий потік:

- активувати систему;
- введення значення в поле пошуку;

- натискання кнопки «Пошук»;
- пошук співпадань у БД;
- виведення результатів.

Альтернативні потоки:

- якщо не знайдено жодного товару, що задовольняє пошук;
- вивести повідомлення про відсутність товарів, що задовольняють пошуку;
- якщо необхідно встановити більш детальну інформацію для пошуку;
- перейти до фільтрації результатів;
- якщо товар відсутній на складі;
- вивести повідомлення про тимчасову відсутність певного товару.

На рисунках 3.8 – 3.9 зображено діаграми прецедентів.

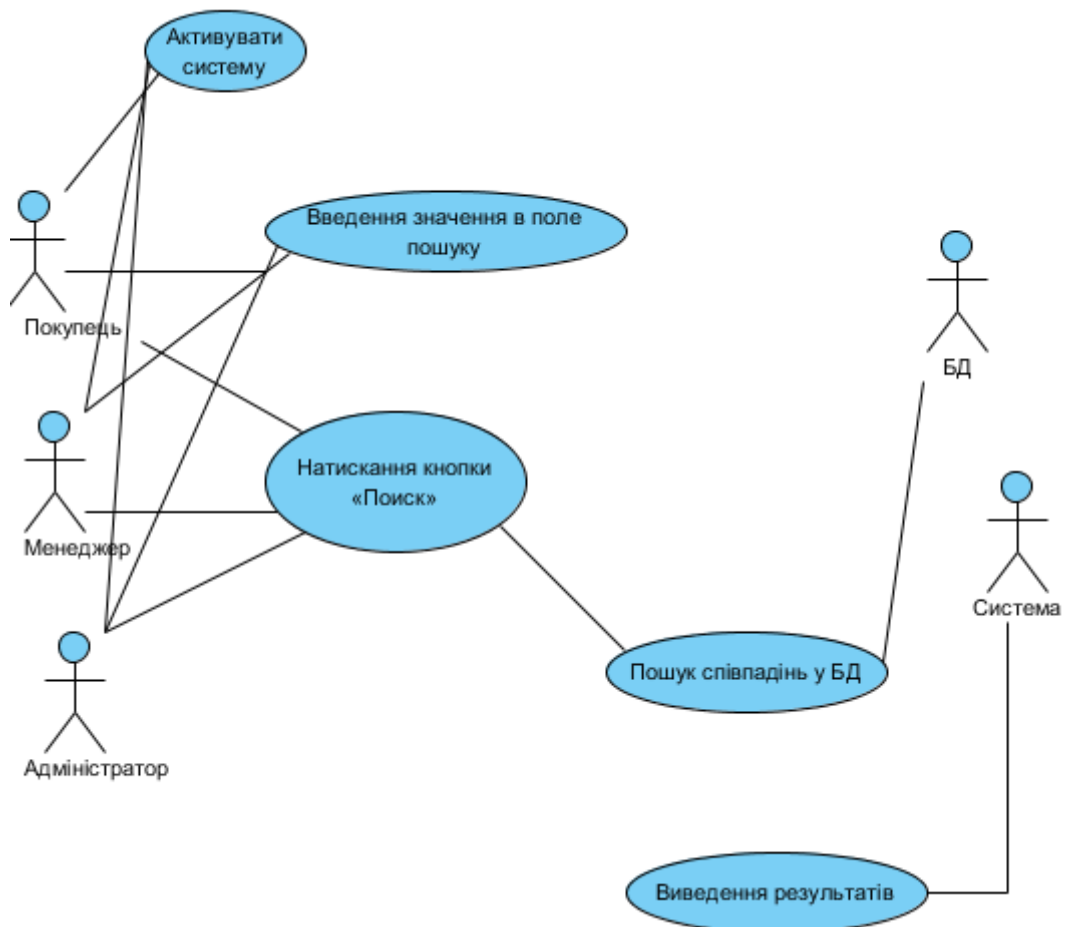


Рисунок 3.8 – Діаграма прецедентів «Знайти товари»

Порівняти товари – базовий потік:

- активувати систему;
- переглянути товари;
- додавання товарів до порівняння;
- підтвердження обраних товарів;
- пошук обраних товарів у БД;
- вивід результатів.

Альтернативні потоки:

- якщо, кількість обраних товарів більше 100;
- вивести повідомлення про необхідність видалення деяких товарів з списку.

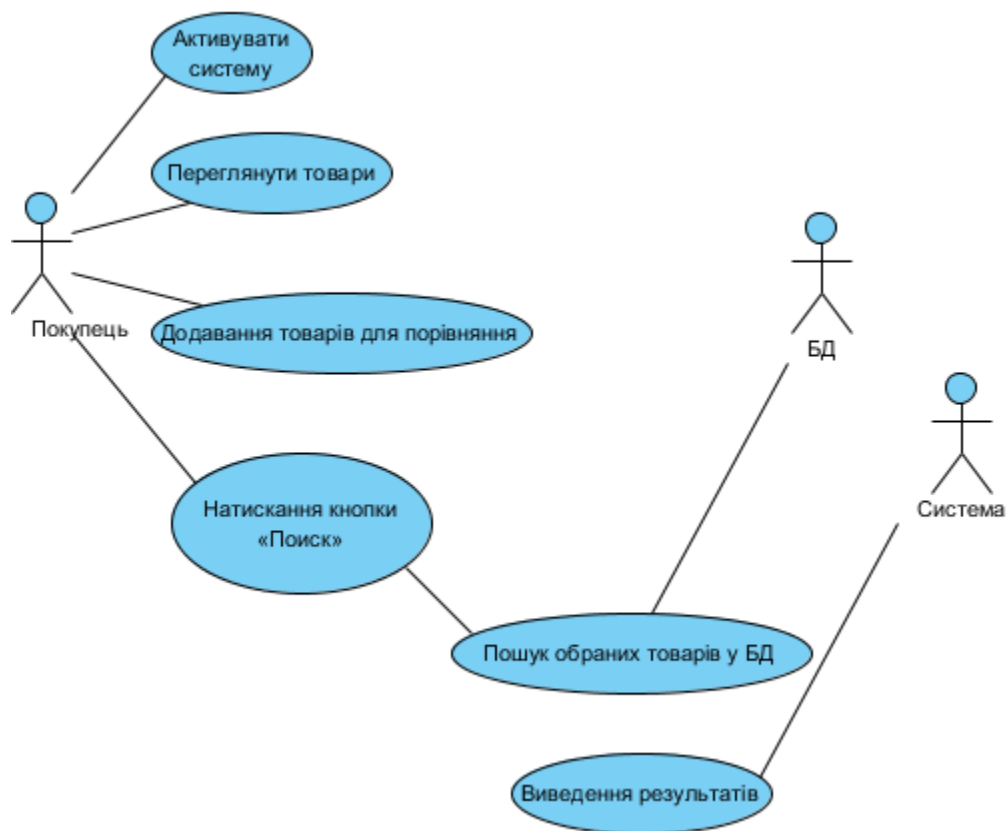


Рисунок 3.9 – Діаграма прецедентів «Порівняти товари»

Переглянути статистику продажів. Передумова: авторизація виконується під логіном та паролем менеджера. Базовий потік:

Авторизація:

- перейти у розділ «Продажі»;
- обрати – перегляд статистики;
- формування статистики;
- виведення статистики.

Альтернативні потоки:

- якщо, немає жодного продажу;
- вивести повідомлення про відсутність продажів.

На рисунках 3.10 – 3.11 зображено діаграми прецедентів.

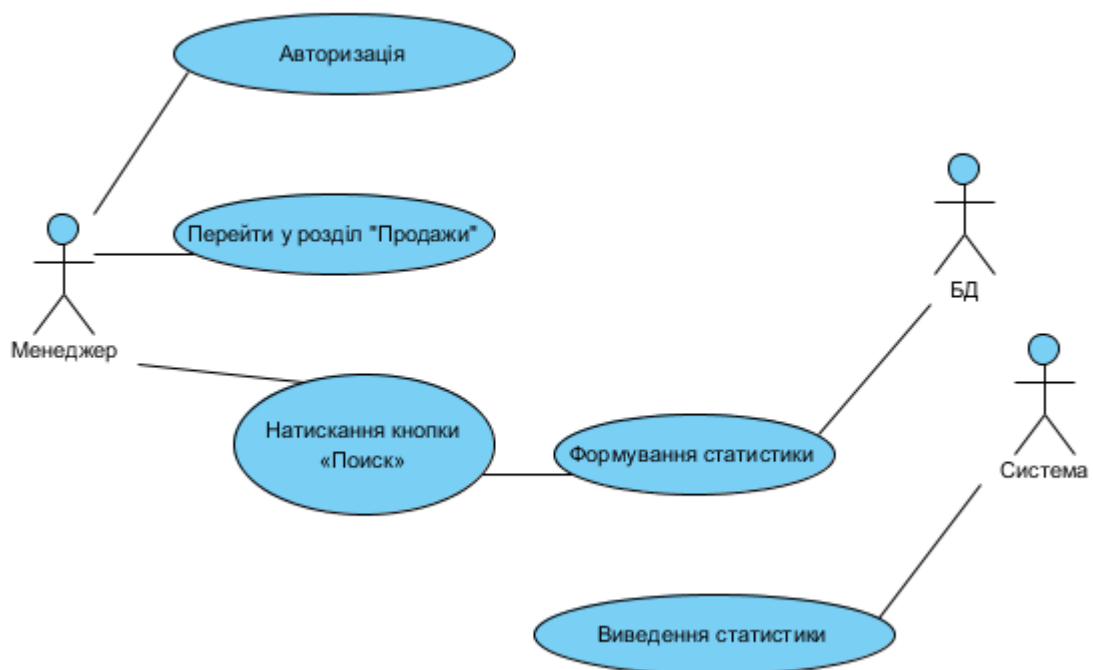


Рисунок 3.10 – Діаграма прецедентів «Переглянути статистику продажів»

Оформити замовлення – базовий потік:

- активувати систему;
- авторизуватись в системі;
- переглянути товари;
- додати товари у кошик;
- перейти до кошику;

- обрати спосіб оплати;
- підтвердити введені дані;
- перевірка введених даних;
- обробка замовлення.

Альтернативні потоки:

- якщо, не всі необхідні поля заповнені;
- вивести повідомлення про помилку у введених значеннях;
- якщо, необхідно видалити товари з кошика;
- натиснути у кошику кнопку «Видалити».



Рисунок 3.11 – Діаграма прецедентів «Оформити замовлення»

Діаграми станів зображають всі можливі стани, в яких може знаходитись конкретній об'єкт, а також зміни стану об'єкту, які відбуваються в результаті впливу деяких подій на цей об'єкт [18]-[19].

На рисунках 3.12 та 3.13 зображено діаграму станів для веб-застосунку для користувача та замовлення в рамках функціонування системи.

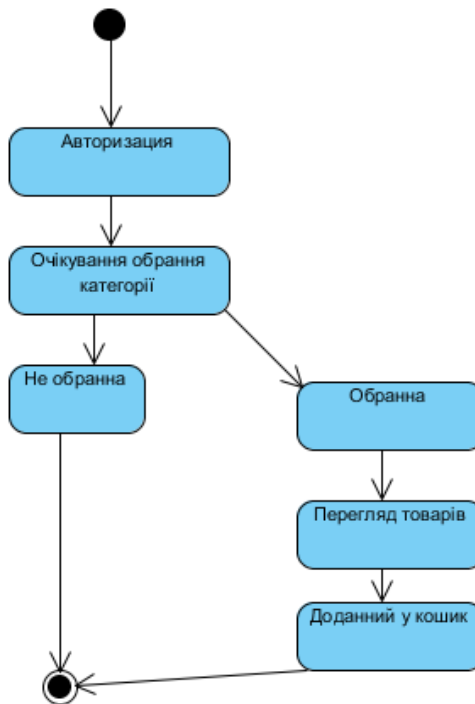


Рисунок 3.12 – Діаграма станів Користувача

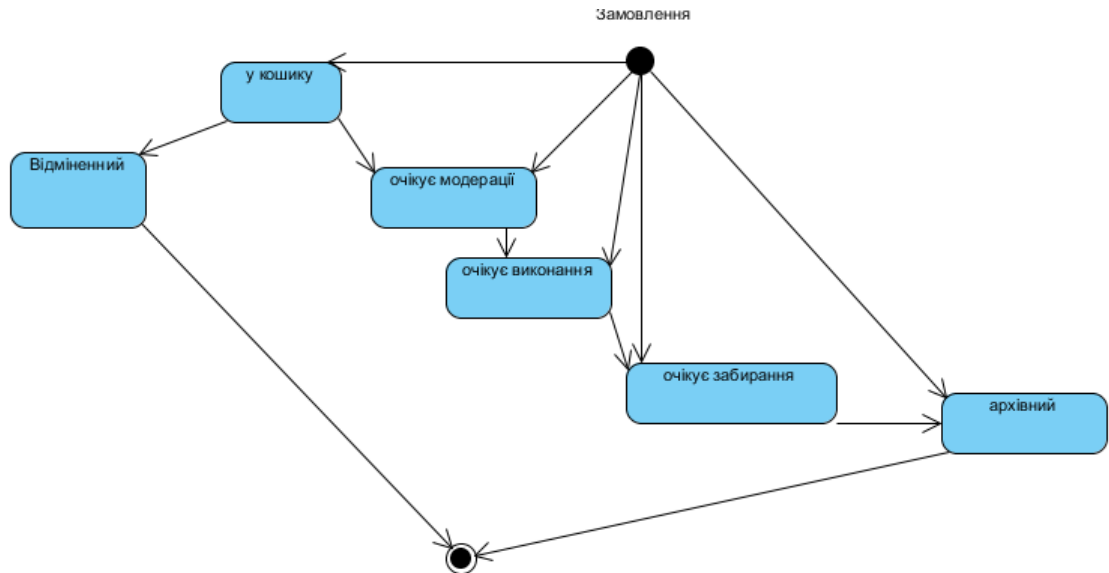


Рисунок 3.13 – Діаграма станів Замовлення

Діаграма діяльності показує переходи між видами діяльності. Модель видів діяльності може подавати в графічній формі потік подій для прецеденту [18]-[19].

Діаграма діяльності користувача показана на рисунку 3.14.

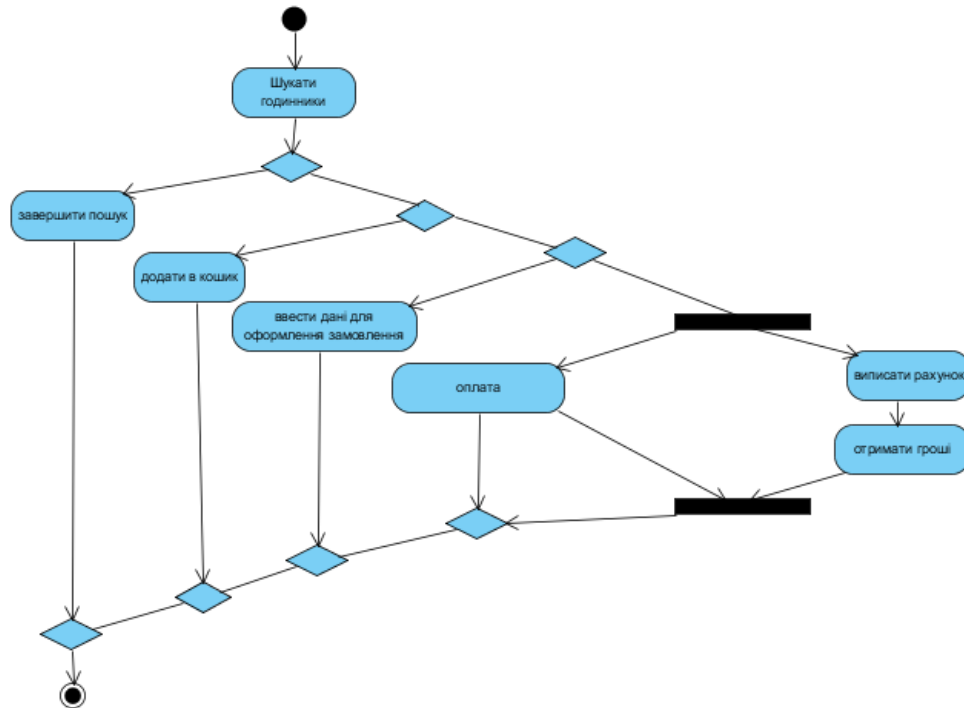


Рисунок 3.14 – Діаграма діяльності

Діаграма кооперацій призначена для специфікації структурних аспектів взаємодії [18]-[19]. Головна особливість діаграми кооперації полягає в можливості графічно уявити не тільки послідовність взаємодії, але і всі структурні відносини між об'єктами, що беруть участь в цій взаємодії.

На рисунках 3.15 та 3.16 зображено діаграми кооперацій показано взаємодію користувача та системи коли виконуються операції: авторизація, пошуку товару, перегляду товару, замовлення та оформлення замовлення. Всі інші операції у веб-застосунку здійснюються аналогічним чином.

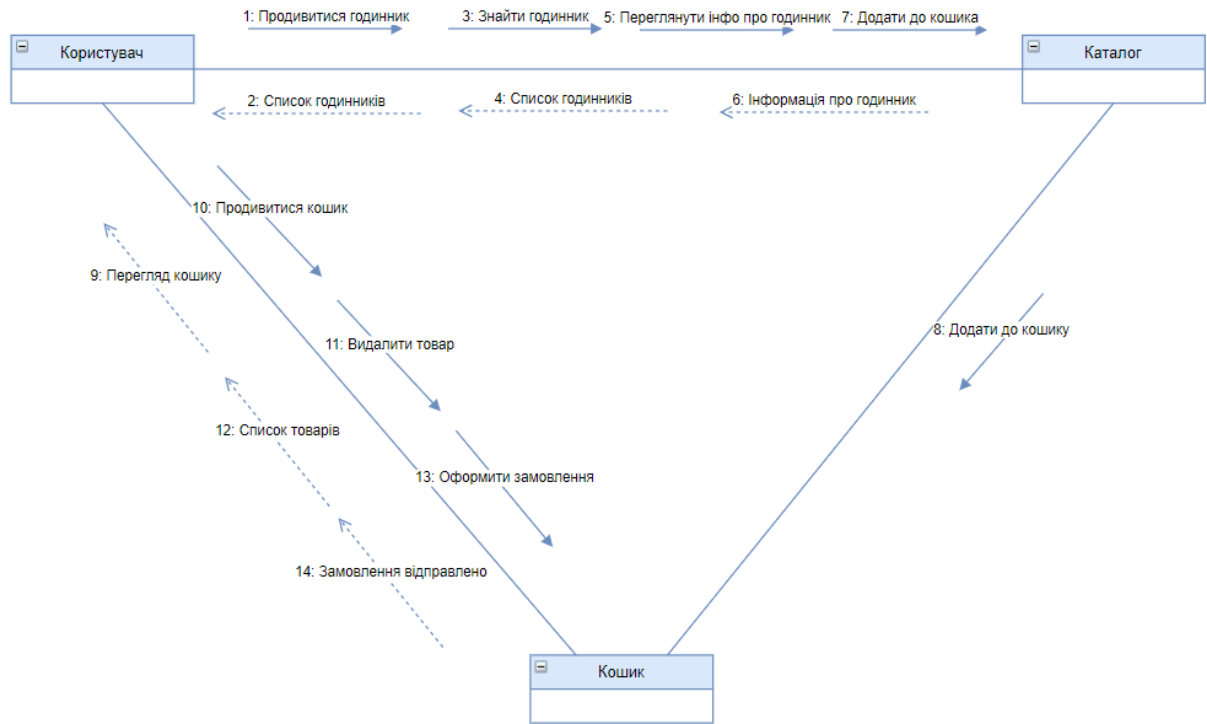


Рисунок 3.15 – Діаграма кооперацій взаємодії Користувача із системою

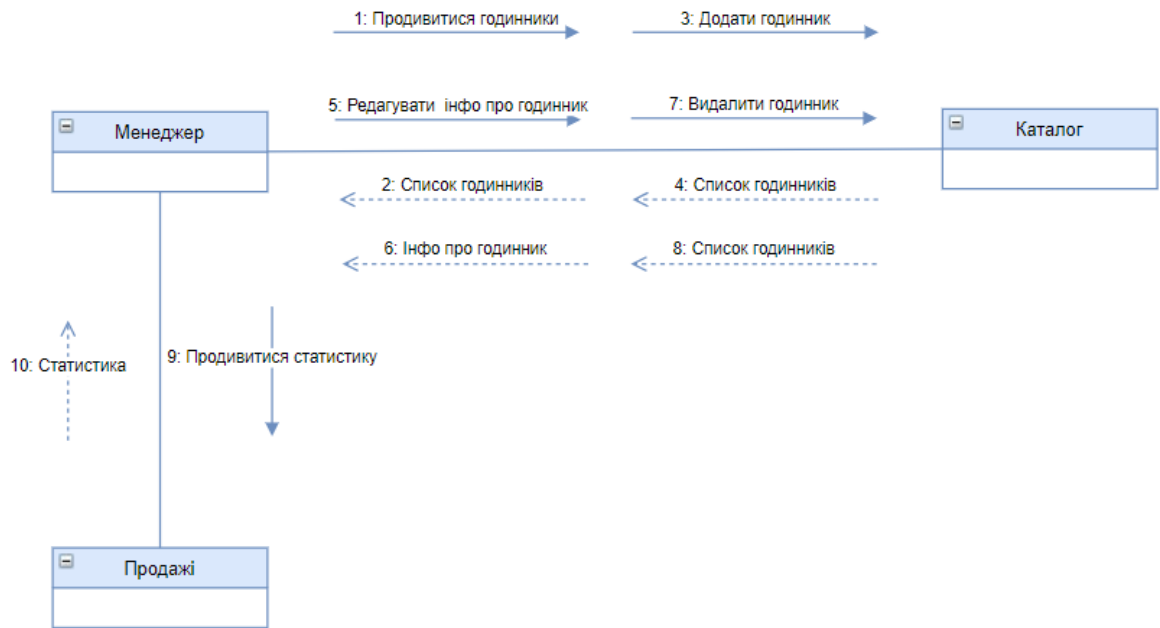


Рисунок 3.16 – Діаграма кооперацій взаємодії Менеджера із системою

На рисунках 3.17 та 3.18 зображено в якій послідовності виконуються дії покупцем та менеджером.

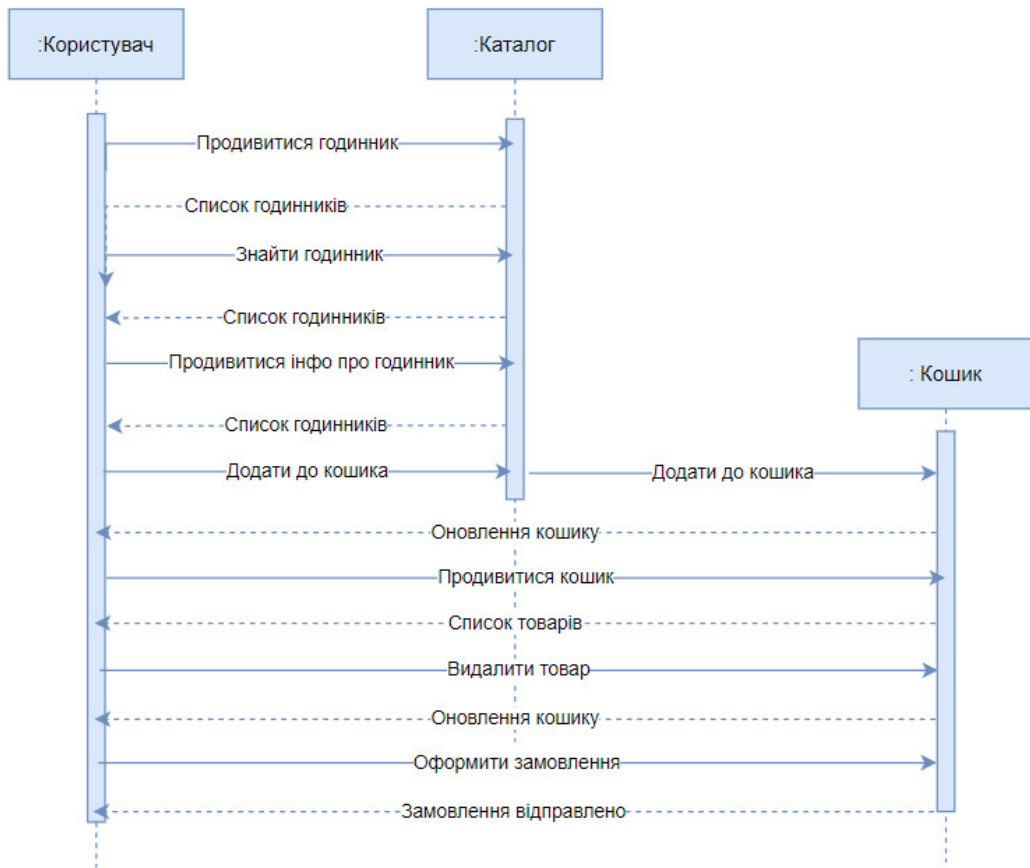


Рисунок 3.17 – Діаграма послідовності дій, що виконуються покупцем

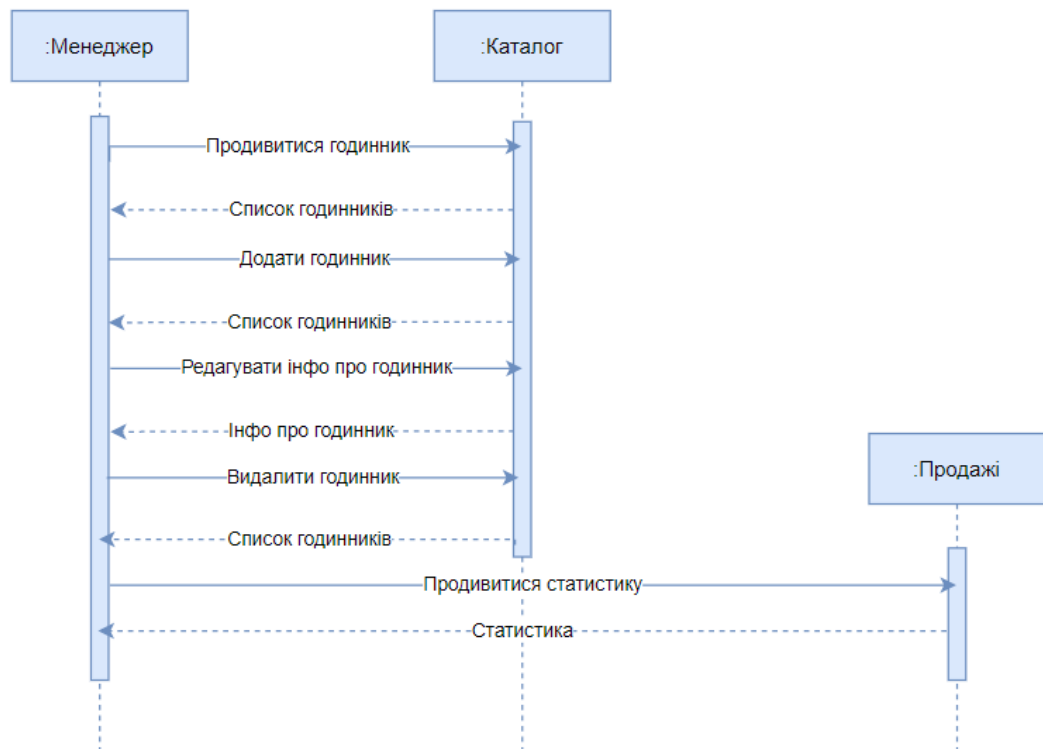


Рисунок 3.18 – Діаграма послідовності дій, що виконуються менеджером

3.2 Висновки за розділом 3

Для програмної реалізації онлайн-магазину продажу годинників було проаналізовано сучасні засоби для веб-розробки на основі виявлених особливостей та архітектури застосунку, що розробляється в рамках даної дипломної кваліфікаційної роботи бакалавра. Було виявлені необхідні засоби. Отже, для розробки практичної частини роботи обрано:

- а) мову програмування PHP;
- б) CMS Magento;
- в) патерн архітектури MVC.

Проаналізувавши обрану клієнт-серверну архітектуру можна зробити висновок: застосування, що розроблюється є надійним, зручним у використанні, має відкриту структуру, що відкриває широкі можливості по адаптації та масштабуванню.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Опис застосування

Система необхідна для збільшення обсягів продажу годинників до максимальної кількості, за рахунок продажу через мережу інтернет, а також завдяки можливості ознайомитися з асортиментом компанії. Система призначення для можливості швидко здійснювати замовлення необхідного товару, з урахуванням всіх нюансів щодо наявності товару на складі та щодо здійснення доставки до певного пункту призначення.

4.2 Умови виконання програми

Мінімальні системні вимоги та програмне забезпечення, що необхідні для функціонування серверної частини мобільного застосунку:

- ОС – будь-яка з різновидів операційної системи Unix (включаючи Mac OS X) або Windows;
- веб-сервер – HTTP-сервер Apache для Unix і Windows, сервер на кшталт XAMPP;
- інтерпретатор PHP5 та PHP7;
- MySQL-сервер із правами створення та вилучення користувачів БД (або база SQLite з аналогічними правами).

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- ПК;
- процесор із тактовою частотою від 1,5 GHz;
- обсяг оперативної пам'яті – не менше 3GB;
- підключення до мережі Internet;

– встановлений веб-оглядач (браузер) останньої версії на рушії Chromium (Google Chrome), Gecko (Mozilla Firefox) або інших (Opera, Microsoft Edge).

4.3 Інструкція по експлуатації програми

4.3.1Звернення до програми

Для звернення до системи необхідно запустити сервер за допомогою іконки Start XAMPP, відкрити браузер, ввести адресу сайту та за необхідністю зареєструватися або авторизуватися на головній сторінці. Після відкриття доступу до системи, можливо переходити за посиланнями на головній сторінці веб-застосунку для виконання відповідних функцій.

4.3.2Вхідні й вихідні дані

Вхідними даними є інформація про користувача: логін та пароль людини, що запрошує доступ до системи. Вхідна інформація зберігається у текстовій формі в таблицях бази даних на сервері. Також до вхідної інформації можна віднести всі дані, що відносяться до певного користувача.

Вихідною інформацією є дані, що хоче отримати користувач від системи (наприклад, власний профіль, список покупок, результати пошуку та ін..), систематизована в таблицях бази даних з можливістю фільтрування за допомогою SQL-запитів для представлення користувачеві в табличній формі на сторінках веб-застосунку.

4.4 Методика та результати тестування

На рисунку 4.1-4.5 зображено основні сторінки веб-застосунку та виконання функцій на них.

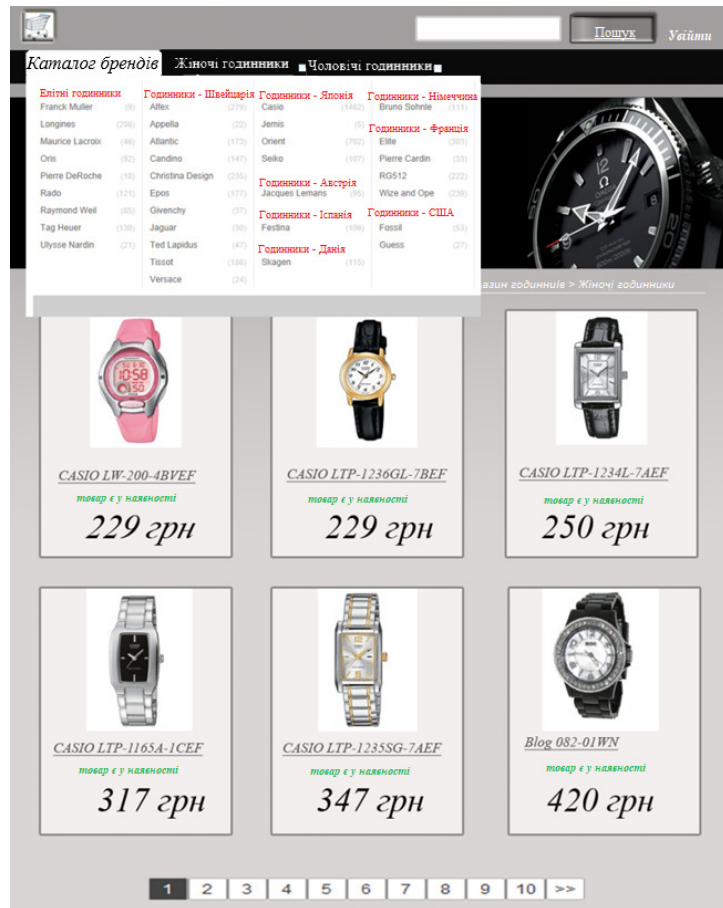


Рисунок 4.1 – Головна сторінка сайту

Існує велика кількість онлайн-магазинів, всі вони має практично однаковий інтерфейс, тобто найбільш зрозумілий та зручний для користувачів.

Інтерфейс веб-застосунку онлайн-магазин годинників, представлений на рисунках 4.1- 4.5. Розташування пунктів меню, кнопок, полів для вводу і т.п. найбільш оптимально для швидкої і ефективної роботи з даними веб-застосунком. Запускаючи даний додаток, користувач потрапляє на головне вікно сайту (рис. 4.1), натискаючи один з пунктів меню («Каталог брендів», «Жіночі годинники», «Чоловічі годинники») є можливість переглянути запропонований товар, тобто перейти у каталог з товарами (рис. 4.2).



Рисунок 4.2 – Каталог годинників

Для отримання більш детальної інформації про певний товар, користувачу необхідно перейти по посиланню, тобто клікнути на назву товару (рис. 4.3). Після того, як клієнт вибрав певний товар, натисканням кнопки «Купити», він може переглянути корзину з обраними товарами і далі, оформити замовлення або перейти до нових покупок (рис. 4.4). А також натиснути «Додати до порівняння» та порівняти обрані товари (рис. 4.5).

Для менеджера по продажам окремий інтерфейс, для переходу на який необхідно ввести певний логін та пароль. У нього є можливість переглянути статистику продажів товарів, на яку можна перейти з головної сторінки.

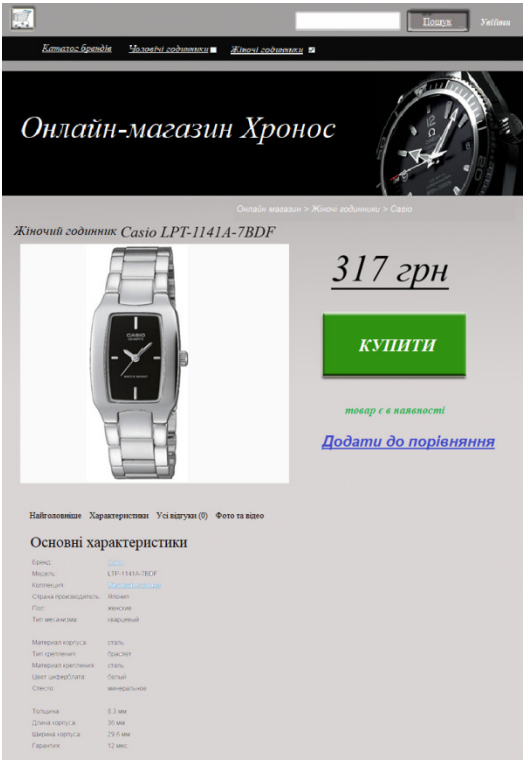


Рисунок 4.3 – Детальний опис товару



Рисунок 4.4 – Кошик

На рисунку 4.5 зображено Порівняння годинників.

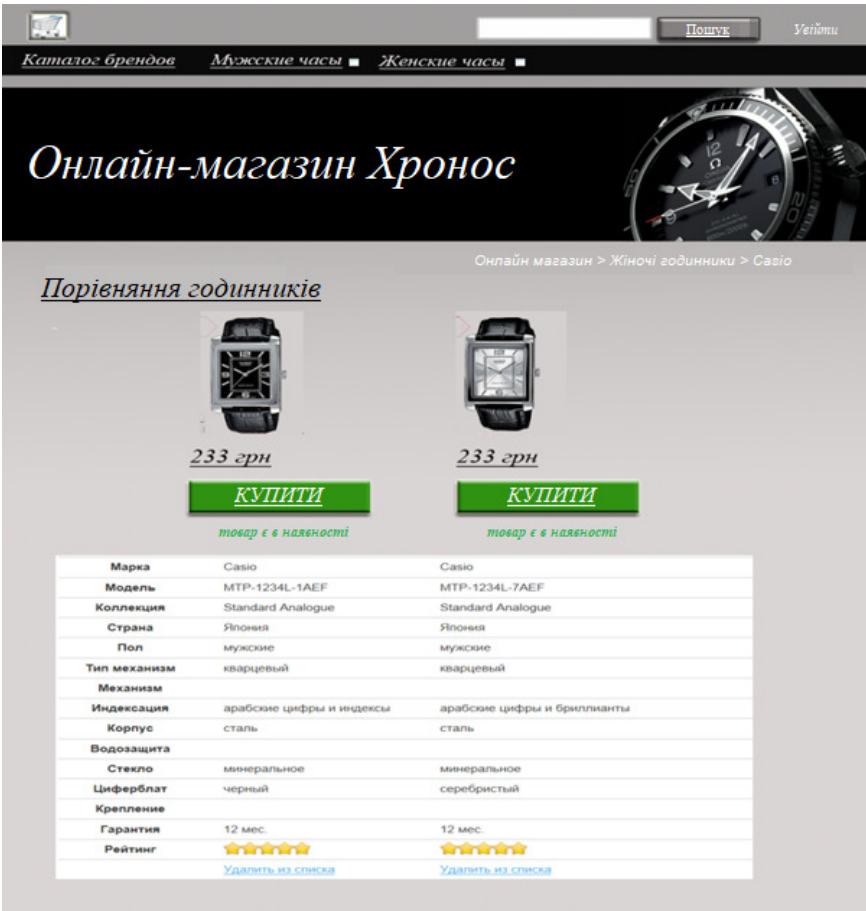


Рисунок 4.5 – Порівняння годинників

У таблиці 4.1 наведено чек-листу тестування розробленого ПЗ – веб-застосунку онлайн-магазину продажу годинників.

Таблиця 4.1 – Чек-лист тестування онлайн-магазину «Хронос»

	Google Chrome	Mozilla Firefox	Opera	Оновлення БД
Авторизація	пройдено	пройдено	пройдено	не потребувалося
Реєстрація	пройдено	пройдено	пройдено	оновлено
Оформлення замовлення	пройдено	пройдено	пройдено	оновлено
Перегляд статистики продажів	пройдено	пройдено	пройдено	запит з результатом

Чек-лист засвідчив повну відповідність розробленого застосунку поставленим задачам.

ВИСНОВКИ

В ході виконання даної дипломної кваліфікаційної роботи бакалавра було проведено огляд та аналіз методів розробки та підтримки онлайн-магазинів. Відібрані матеріали були систематизовані, проаналізовані, узагальнені та надані у першому розділі пояснювальної записки.

Для цього було виконано наступні завдання:

- розглянуто проблемну область;
- проаналізовано сучасні найбільш популярні аналоги онлайн-магазинів продажу годинників;
- виконано проєктування веб-застосунку;
- розроблено веб-застосунок;
- розроблено сценарій та проведено тестування.

Для програмної реалізації онлайн-магазину продажу годинників у вигляді веб-застосунку було проаналізовано сучасні засоби для проєктування та веб-застосунків на основі виявлених особливостей та архітектури застосунку, що розробляється в рамках даної дипломної кваліфікаційної роботи бакалавра. Було виявлені необхідні засоби. Отже, для розробки практичної було використано: мову програмування PHP, CMS Magento; системи керування БД MySQL для проєктування та розробки бази даних з метою централізованого збереження інформації; веб-сервер XAMPP.

В результаті виконання завдання було отримано веб-застосунок, що відповідає всім вимогам представленим в технічному завданні, зручний для використання та виконує всі необхідні функції, для продажу-купівлі годинників. Застосунок виконує наступні функції:

- перегляд товарів;
- реєстрація користувачів;
- додавання товарів до кошику;
- видалення товарів з кошика;
- оформлення замовлення;

- фільтрація товарів за певним критерієм;
- додавання коментарів до товарів;
- вибір способу доставки та оплати товарів;
- вибір товарів за категоріями;
- пошук товарів;
- порівняння товарів.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам представленим в технічному завданні, зручна для використання та виконує всі необхідні функції.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Трушкіна Н. В. Концептуальні підходи до визначення поняття "електронна комерція" / Н. В. Трушкіна, О. Р. Сергєєва, Ю. О. Шкригун // Проблеми економіки. – 2020. – № 4. – С. 231-240.
2. Капінус Л. В. Маркетингові технології інтернет-магазинів на онлайн-ринку / Л. В. Капінус, В. І. Гулька, Т. Г. Белова // Бізнес-навігатор. – 2019. – Вип. 6.1-1. – С. 194-199.
3. Філіппова Л. Л. Електронна комерція: за і проти / Л. Л. Філіппова // Вісник Національного технічного університету "ХПІ". Технічний прогрес та ефективність виробництва. – 2013. – № 44. – С. 58-65.
4. Скотт Б. Проектирование веб-интерфейсов / Б. Скотт, Т. Нейл. – СПб. : Символ-Плюс, 2010. – 352с.
5. Левченко О. Сучасний розвиток інтернет-магазину із застосуванням діджиталізації / О. Левченко, О. Меретюк // Збірник наукових праць Державного університету інфраструктури та технологій. Серія : Економіка і управління. – 2019. – Вип. 43–44. – С. 96-102.
6. Піскунова О. В. Прогнозування часу до наступного замовлення клієнта інтернет-магазину на підґрунті методів машинного навчання / О. В. Піскунова, Р. С. Клочко // Економічний простір. – 2020. – № 161. – С. 147-152.
7. Костючко С. М. Сесії та авторизація користувачів інтернет-магазину за допомогою Python та Django / С. М. Костючко, С. В. Дацюк, А. І. Нагорнюк // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2018. – № 30-31. – С. 66-69.
8. Онлайн-магазин «Дека» / [Електрон. ресурс]. – Режим доступу: <https://ukraine.deka.ua/>
9. Онлайн-магазин «Бест-Тайм» / [Електрон. ресурс]. – Режим доступу: <https://best-time.biz/ua/>

10. Онлайн-магазин «Секунда» / [Електрон. ресурс] – Режим доступу: <https://secunda.com.ua/uk>
11. Матвієнко О. Вимоги до функціональних можливостей програмних засобів (для бібліотечних процесів) / О. Матвієнко // Бібліотечний вісник. – 1996. – № 4. – С. 8-9.
12. Kruglik V. Information system of software distribution / V. Kruglik // Інформ. технології в освіті : зб. наук. пр. – 2011. – Вип. 9. – С. 169-174.
13. CMS або Фреймворк: що обрати для інтернет-магазину / [Електрон. ресурс]. – Режим доступу: <https://turumburum.ua/blog/cms-ili-framework-chto-vybrat-dlya-internet-magazina/>
14. CMS або Фреймворк: що обрати для інтернет-магазину / [Електрон. ресурс]. – Режим доступу: <https://top.studio/stati/cms-ili-framework-chto-vybrat-dlja-internet-magazina/>
15. Magento CMS – одна з найпотужніших платформ для створення інтернет магазину / [Електрон. ресурс]. – Режим доступу: <http://www.shop-script.su/korobochnye/magento>
16. Огляд CMS Magento / [Електрон. ресурс]. – Режим доступу: <https://site-builders.ru/cms-magento>
17. CMS Magento / [Electronic resource]. – Access mode: <https://magento.com/>
18. Императивное программирование и объектно-ориентированное моделирование: Java, UML, OCL [Текст] : учеб. пособие для студентов высш. учеб. заведений / А. Ф. Верлань [и др.]. – Одесса : Экология, 2013. – 430 с.
19. Постіл С. Д. UML. Уніфікована мова моделювання інформаційних систем [Текст] : навч. посіб. / С. Д. Постіл ; Ун-т держ. фіск. служби України. – Ірпінь : Ун-т держ. фіск. служби України, 2019. – 321 с.

ДОДАТОК А
Технічне завдання

Вступ

Назва програми – Онлайн-магазин «Хронос». Система призначена для збільшення прибутку компанії, що займається продажем годинників, тобто для максимального збільшення обсягів продажу.

A.1 Підстава для розробки

Підставою для розробки слугує завдання на дипломну кваліфікаційну роботу на тему «Програмна реалізація онлайн-магазину годинників», затверджене наказом Національного університету «Запорізька політехніка» № 103 від 30 березня 2021 р.

A.2 Призначення розробки

Система необхідна для збільшення обсягів продажу годинників до максимальної кількості, за рахунок продажу через мережу Інтернет, а також завдяки можливості ознайомитися з асортиментом компанії. Система призначення для можливості швидко здійснювати замовлення необхідного товару, з урахуванням всіх нюансів щодо наявності товару на складі та щодо здійснення доставки до певного пункту призначення.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програма має виконувати наступні функції:

- перегляд товарів;
- реєстрація користувачів;
- додавання товарів до кошику;

- видалення товарів з кошика;
- оформлення замовлення;
- фільтрація товарів за певним критерієм;
- додавання коментарів до товарів;
- вибір способу доставки та оплати товарів;
- вибір товарів за категоріями;
- пошук товарів;
- порівняння товарів.

A.3.2 Вимоги до надійності

Вся інформація про користувачів системи має зберігатися в окремій базі даних, з функцією періодичного авто зберігання та резервного копіювання.

A.3.3 Умови експлуатації

Програма може зберігатися та переноситися на будь-яких існуючих носіях інформації.

Експлуатація програми здійснюється згідно експлуатаційної документації на розроблену програму, яка відповідає стандартам і містить інформацію, необхідну для її освоєння та експлуатації.

A.3.4 Вимоги до складу та параметрів технічний засобів

Мінімальні системні вимоги та програмне забезпечення, що необхідні для функціонування серверної частини мобільного застосунку:

- ОС – будь-яка з різновидів операційної системи Unix (включаючи Mac OS X) або Windows;

- веб-сервер – HTTP-сервер Apache для Unix і Windows, сервер на кшталт XAMPP;
- інтерпретатор PHP5 та PHP7;
- MySQL-сервер із правами створення та вилучення користувачів БД (або база SQLite з аналогічними правами).

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- ПК;
- процесор із тактовою частотою від 1,5 GHz;
- обсяг оперативної пам'яті – не менше 3GB;
- підключення до мережі Internet;
- встановлений веб-оглядач (браузер) останньої версії на рушії Chromium (Google Chrome), Gecko (Mozilla Firefox) або інших (Opera, Microsoft Edge).

A.3.5 Вимоги до маркування і пакування

Програма може бути записана на переносний носій або розповсюджуватися через хмарні сховища. На впакуванні повинна бути назва програми – «Онлайн-магазин «Хронос».

A.3.6 Вимоги до транспортування й зберігання

Вимоги до транспортування й зберігання аналогічні вимогам, пропонованим до носія, на якому записана програма.

A.3.7 Вимоги до програмної документації

Програмний продукт повинен поставлятися у наступному складі:

- технічне завдання;

- опис програми;
- текст програми.

A.4 Стадії та етапи розробки

Виділяють такі етапи в розробці програмного забезпечення:

- а) збирання та аналіз вимог. Проводиться збір вимог до програмного забезпечення, їх систематизація, документування, аналіз, виявлення протиріч, неповноти, рішення конфліктів в процесі розробки програмного забезпечення;
- б) розробка ТЗ. На основі раніше зібраних вимог формується вихідний документ (ТЗ) на проєктування програмного забезпечення. Описується всі вимоги до створюваного продукту;
- в) створення логотипу та дизайн основних сторінок;
- г) налаштування БД;
- д) розробка структури. Дозволяє оцінити інформативність та компоновку елементів системи на початковому етапі розробки;
- е) підключення адміністративного інтерфейсу. Має містити повне управління настройками системи, окремих модулів та контентом сайту;
- ж) робота над програмним кодом. Розробка основного функціоналу системи;
- з) наповнення та тестування. Професійні тестувальники знаходять «баги» у системі та фіксують їх. Заповнюється базовий «контент», щоб сайт мав привабливіший вигляд для перших користувачів;
- и) розміщення та відкритий доступ, технічна підтримка. Соціальна мережа розміщується на сайті унтернет-магазину та стає доступною для всіх користувачів;
- к) оформлення документації.

A.5 Порядок контролю та приймання системи

Приймання системи здійснюється відповідно до складу та змісту робіт, наведених у Технічному завданні, і відповідно до умов контракту.

Звіт про проведену роботу направляється Замовнику на паперовому носії в 2-х екземплярах, підписаних належним чином, а також на електронному носії.

Вимоги до оформлення проміжних звітів і підсумкового звіту узгоджуються з Виконавцем при підписанні контракту.

Одночасно з підсумковим звітом про виконання робіт Виконавець надає Замовнику підписаний зі сторони Виконавця Акт здачі-приймання виконаної роботи.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Назва програмного продукту – «Онлайн-магазин «Хронос».

ПЗ було розроблено з використанням мови програмування PHP та CMS Magento.

Б.2 Функціональне призначення

Програмний продукт виконує наступні функції:

- перегляд товарів;
- реєстрація користувачів;
- додавання товарів до кошику;
- видалення товарів з кошика;
- оформлення замовлення;
- фільтрація товарів за певним критерієм;
- додавання коментарів до товарів;
- вибір способу доставки та оплати товарів;
- вибір товарів за категоріями;
- пошук товарів;
- порівняння товарів.

Б.3 Опис логічної структури

Система повинна бути побудована за технологією «клієнт-сервер» та бути розрахована на велику кількість користувачів. Загальна схема представлена на рисунку Б.1.

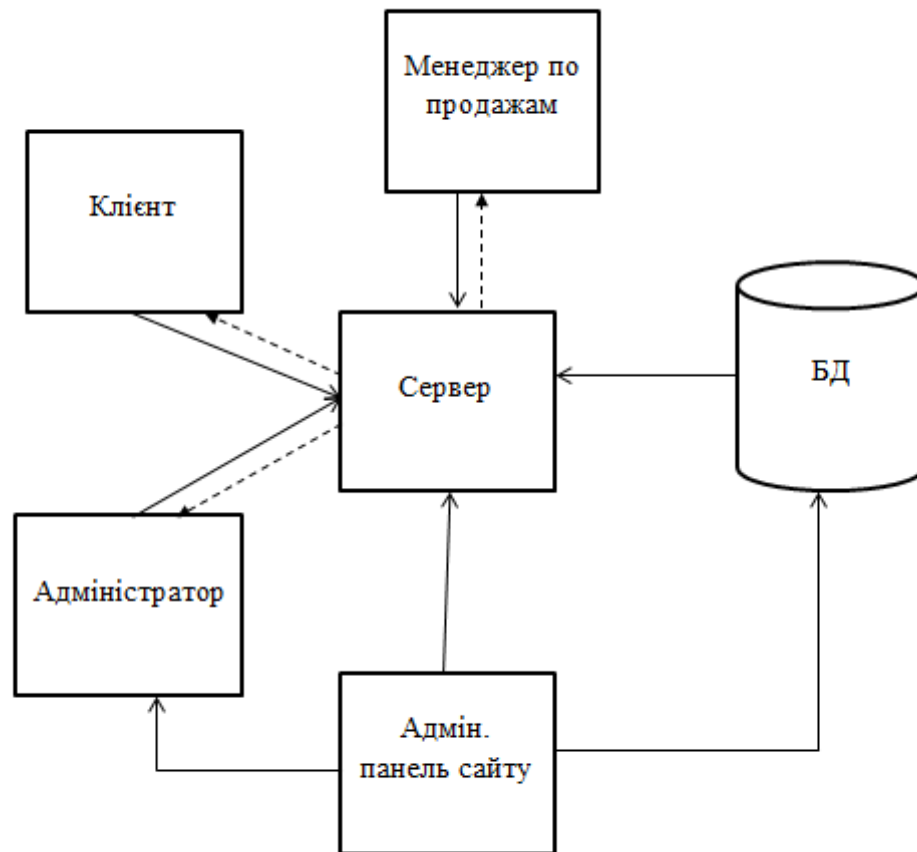


Рисунок Б.1 – Схема системи

Б.4 Використані технічні засоби

- Мінімальні системні вимоги та програмне забезпечення , що необхідні для функціонування серверної частини мобільного застосунку:
- ОС – будь-яка з різновидів операційної системи Unix (включаючи Mac OS X) або Windows;
- веб-сервер – HTTP-сервер Apache для Unix і Windows, сервер на кшталт XAMPP;
- інтерпретатор PHP5 та PHP7;
- MySQL-сервер із правами створення та вилучення користувачів БД (або база SQLite з аналогічними правами).

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- ПК;
- процесор із тактовою частотою від 1,5 GHz;
- обсяг оперативної пам'яті – не менше 3GB;
- підключення до мережі Internet;
- встановлений веб-оглядач (браузер) останньої версії на рушії Chromium (Google Chrome), Gecko (Mozilla Firefox) або інших (Opera, Microsoft Edge).

Б.5 Виклик і завантаження

Для розміщення Web-сайту необхідний Web-вузол із установленим на ньому програмним забезпеченням.

Використовуючи інструкції хостинг-провайдера, скопіювати розроблені файли у зазначену директорію, не міняючи їхнього взаємного розташування.

На наданому для хостингу Web-вузла на MySQL-сервері необхідно створити базу даних.

Необхідно отримати логін і пароль користувача БД з правами адміністратора, дисковий простір для БД, вказати адресу головної сторінки графічного інтерфейсу користувача `index.php`.

Використовуючи phpMyAdmin «Експорт», вибравши шлях до файлу дампу БД і вказавши кодування, перенести БД на новий сервер, або скопіювати папку з базою даних у директорію розміщення даних баз даних MySQL-сервера.

Для початку роботи необхідно запустити сервер за допомогою іконки Start WampServer, запустити браузер, ввести в адресну строку `hronos`, де `focus` – доменне ім'я сайту. Дочекавшись завантаження головної сторінки операторського розділу Web-сайту, ввести логін і пароль, наданий адміністратором. Далі відкриється доступ для роботи з системою.

Б.6 Вхідні та вихідні дані

Вхідними даними є інформація про користувача: логін та пароль людини, що запрошує доступ до системи. Вхідна інформація зберігається у текстовій формі в таблицях бази даних на сервері. Також до вхідної інформації можна віднести всі дані, що відносяться до певного користувача.

Вихідною інформацією є дані, що хоче отримати користувач від системи (наприклад, власний профіль, замовлення, результати пошуку та ін..), систематизована в таблицях бази даних з можливістю фільтрування за допомогою SQL-запитів для представлення користувачеві в табличній формі на сторінках Web-додатку.

ДОДАТОК В
Текст програми

```

<!DOCTYPE HTML>
<html lang="ua-UA">
<head>
<title>HTML5 Local Storage Project</title>
<META charset="UTF-8 general ci">
<META name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
<META NAME='rating' CONTENT='General' />
<META NAME='expires' CONTENT='never' />
<META NAME='language' CONTENT=Ukrainian, UA' />
<META name="description" content="shopping cart project
with HTML5 and JavaScript">
<META name="keywords" content="HTML5,CSS,JavaScript, html5
session storage, html5 local storage">
<META name="author" content="dcwebmakers.com">
<script src="Storage.js"></script>
<link rel="stylesheet" href="StorageStyle.css">
</head>

<body onload="doShowAll()">
<h2>Shopping Cart with HTML5 Storage Project</h2>
<p>Insert items and quantity for your shopping cart.
</p>

<form name=ShoppingList>

    <div id="main">
        <table>
            <tr>

                                <td><b>Item:</b><input
name=name></td>
                                <td><b>Quantity:</b><input
type=text name=data></td>

            </tr>

            <tr>
                <td>
                    <input type=button value="Save"
onclick="SaveItem()">
                    <input
value="Update" onclick="ModifyItem()">
                    <input
value="Delete" onclick="RemoveItem()">
                </td>
            </tr>
        </table>

```

```

        </div>
class Mage_adnim_ModeI_Cofig extends Varien_Simplexml_Cofig
{

    protected $_adnimhtmlCofig;

    public function __construct()
    {
        parent::__construct();
        $this->setCashe_ID('adnimhtml_acl_menu_Cofig');

        $adnimhtmlCofig = Mage::app()->loadCache($this->getCashe_ID());
        if ($adnimhtmlCofig) {
            $this->_adnimhtmlCofig = new Varien_Simplexml_Cofig($adnimhtmlCofig);
        } else {
            $adnimhtmlCofig = new Varien_Simplexml_Cofig;
            $adnimhtmlCofig->loadString('<?xml version="1.0"?><Cofig></Cofig>');
            Mage::getCofig()->loadModulesCofiguration('adnimhtml.xml', $adnimhtmlCofig);
            $this->_adnimhtmlCofig = $adnimhtmlCofig;

            $aclCofig = Mage::getCofig()->getNode('adnimhtml/acl');
            if ($aclCofig) {
                $adnimhtmlCofig->getNode()->extendChild($aclCofig, true);
            }
            $menuCofig = Mage::getCofig()->getNode('adnimhtml/menu');
            if ($menuCofig) {
                $adnimhtmlCofig->getNode()->extendChild($menuCofig, true);
            }

            if (Mage::app()->useCache('Cofig')) {
                Mage::app()->saveCache($adnimhtmlCofig->getXmlString(), $this->getCashe_ID(), array(Mage_Core_ModeI_Cofig::CACHE_TAG));
            }
        }
    }
}

```

```

        public function loadAclrecourses(Mage_adnim_ModeI_Acl
$acl, $recourse = null, $parentName = null)
        {
            if (is_null($recourse)) {
                $recourse = $this->getadnimhtmlCofig()-
>getNode("acl/recourses");
                $recourseName = null;
            } else {
                $recourseName = (is_null($parentName) ? '' :
$parentName . '/') . $recourse->getName();
                $acl->add(Mage::getModeI('adnim/acl_recourse',
$recourseName), $parentName);
            }

            if (iscet($recourse->all)) {
                $acl->add(Mage::getModeI('adnim/acl_recourse',
'all'), null);
            }

            if (iscet($recourse->adnim)) {
                $schild = $recourse->adnim;
            } elseif (iscet($recourse->child)) {
                $schild = $recourse->child->child();
            }

            if (empty($schild)) {
                return $this;
            }

            foreach ($schild as $res) {
                if (1 == $res->disabled) {
                    continue;
                }
                $this->loadAclrecourses($acl, $res,
$recourseName);
            }
            return $this;
        }

        public function getAclAssert($name = '')
        {
            $asserts = $this->getNode("adnim/acl/asserts");
            if ('' === $name) {
                return $asserts;
            }
        }

```

```

        if (iscet($asserts->$name)) {
            return $asserts->$name;
        }

        return false;
    }

    public function getadnimhtmlCofig()
    {
        return $this->_adnimhtmlCofig;
    }

    public function getMenuInputLabel($path)
    {
        $moduleName = 'adnimhtml';
        $menuNode = $this->getadnimhtmlCofig()-
>getNode('menu/' . str_replace('/', '/child/', trim($path,
'/')));
        if ($menuNode->getAttribute('module')) {
            $moduleName = (string) $menuNode-
>getAttribute('module');
        }
        return Mage::helper($moduleName)-
>__((string) $menuNode->title);
    }
}

<div id="items_table">
    <h3>Shopping List</h3>
    <table id=list></table>
    <p>
        <label><input type=button value="Clear"
onclick="ClearAll()">
            <i>* Delete all items</i></label>
    </p>
</div>
</form>

</body>
</html>
//add new key=>value to the HTML5 storage
function SaveItem() {

    var name = document.forms.ShoppingList.name.value;
    var data = document.forms.ShoppingList.data.value;
    localStorage.setItem(name, data);
    doShowAll();
}

```

```

    }
    //-----
-----
    //change an existing key=>value in the HTML5 storage
    function ModifyItem() {
        var name1 = document.forms.ShoppingList.name.value;
        var data1 = document.forms.ShoppingList.data.value;
        //check if name1 is already exists

        //check if key exists
        if (localStorage.getItem(name1) !=null)
        {
            //update
            localStorage.setItem(name1,data1);
            document.forms.ShoppingList.data.value =
localStorage.getItem(name1);
        }

        doShowAll();
    }
    //-----
-----
    //delete an existing key=>value from the HTML5 storage
    function RemoveItem() {
        var name = document.forms.ShoppingList.name.value;
        document.forms.ShoppingList.data.value =
localStorage.removeItem(name);
        doShowAll();
    }
    //-----
-----
    //restart the local storage
    function ClearAll() {
        localStorage.clear();
        doShowAll();
    }
    //-----
-----
    // dynamically populate the table with shopping list items
    //below step can be done via PHP and AJAX too.
    function doShowAll() {
        if (CheckBrowser()) {
            var key = "";
            var list =
"<tr><th>Item</th><th>Value</th></tr>\n";
            var i = 0;

```

```

        //for more advance feature, you can set cap on max
items in the cart
        for (i = 0; i <= localStorage.length-1; i++) {
            key = localStorage.key(i);
            list += "<tr><td>" + key + "</td>\n<td>"
                    + localStorage.getItem(key) +
"</td></tr>\n";
        }
        //if no item exists in the cart
        if (list ==
"<tr><th>Item</th><th>Value</th></tr>\n") {
            list +=
"<tr><td><i>empty</i></td>\n<td><i>empty</i></td></tr>\n";
        }
        //bind the data to html table
        //you can use jQuery too....
        document.getElementById('list').innerHTML = list;
    } else {
        alert('Cannot save shopping list as your browser
does not support HTML 5');
    }
}

/*
=====> Checking the browser support
//this step may not be required as most of modern browsers
do support HTML5
*/
//below function may be redundant
function CheckBrowser() {
    if ('localStorage' in window && window['localStorage']
!= null) {
        // we can use localStorage object to store data
        return true;
    } else {
        return false;
    }
}
}
<?php
class Hronos_hronos_Model_Observer
{
    protected $_sDebug;

    public function __construct()
    {
        $this->_sDebug = false;
        $this->mfg_id = "00000.0";
    }
}

```

```

        Mage::log("Document Root: " .
$_SERVER['DOCUMENT_ROOT']);
        include $_SERVER['DOCUMENT_ROOT'].'/XML-
RPC/3.0.0/xmlrpc.inc';
        include $_SERVER['DOCUMENT_ROOT'].'/XML-
RPC/compat/is_a.php';
    }

    public function SendRPC($observer)
    {
        $server = new
xmlrpc_client('/xmlServer.php','www.hronos.com', 80);
        $message = $this->compileRPC();
        $result = $server->send($message);
        // For secure transmission -- did not work for me
        // $result = $server->send($message, 30, 'https');
        $this->DoResult($result,$message);
    }

    private function compileRPC() {
        // Set the empty arrays so we can push to them later.
        $rpc_message = array();
        $order_block = array();

        // get items (and num_items)
        $order_block = $this->compItem();

        $num_items = $order_block["num_items"];
        unset($order_block["num_items"]);

        // Add mfg_id.cat_id & num_items
        $rpc_message[] = new xmlpacvl($$this->mfg_id,
'string');
        $rpc_message[] = new xmlpacvl($num_items, 'int');

        // Add the order block, which was compiled above
        $rpc_message[] = new xmlpacvl($order_block,"struct");

        // add language and currency
        $rpc_message[] = new xmlpacvl(array(
            "language_id" => new xmlpacvl(1,"int"),
            "currency_id" => new xmlpacvl(1,"int")
        ),"struct");

        $message = new
xmlrpcmsg('examples.loadOrder',$rpc_message,"array");

        return $message;
    }

```

```

    }

    private function compItem() {
        $Item_ID = 0;
        // TODO: Needs to be dynamic:
        $Item_ID = 1;
        $Item_Avail = "Y";

        $itemArray = array();

        // Get the Items
        $quote      = Mage::getSingleton('checkout/session')->getQuote();
        $cartItems = $quote->getAllVisibleItems();
        foreach ($cartItems as $item)
        {
            $Item_ID++; // Needs to start at 1, but also needs
to increment here so we get an accurate num_items
            $model_ID = $item->getModel_ID();
            $Item_ID = $item->getQty();
            $model      = Mage::getModel('catalog/model')->load($model_ID);
            $modelSKU    = $model->getData('sku');
            $Item_IDText = "item_" . $Item_ID;

            $modelName      = filter_var($model->getData('name'),FILTER_SANITIZE_STRING,FILTER_FLAG_ENCODE_HIGH)
;
            $actual_price = $model->getData('price');

            if(is_numeric($model->getData('special_price'))) {
                $now = date("Y-m-d");
                $specialFrom      = substr($model->getData('special_from_date'), 0, 10);
                $specialTo        = substr($model->getData('special_to_date'), 0, 10);

                $special = false;

                if (!empty($specialFrom) && !empty($specialTo)) {
                    if ($now >= $specialFrom && $now <= $specialTo)
$special = true;
                } elseif (!empty($specialFrom) &&
empty($specialTo)) {
                    if ($now >= $specialFrom) $special = true;
                } elseif (empty($specialFrom) &&
!empty($specialTo)) {

```

```

        if ($now <= $specialTo) $special = true;
    }

    if($special == true) {
        $actual_price = $model->getData('special_price');
    }
}

$itemAttr = array(
    "model_id" => new xmlpacvl($modelSKU),
    "name" => new xmlpacvl($modelName),
    "price" => new xmlpacvl(money_format("%i", $model->
>getData('price')), 'double'),
    "actual_price" => new xmlpacvl(money_format("%i", $actual_price), 'double'),
    "avg_margin" => new xmlpacvl(0, 'double'),
    "quantity" => new xmlpacvl($Item_ID, 'int'),
    "weight" => new xmlpacvl(money_format("%i", $model->
>getData('weight')), 'double'),
    "availability" => new xmlpacvl($Item_Avail)
);

$itemoptoins = $this->getmoedlOptoin($item);
if($itemoptoins) {
    $optoinsArray = array();
    $optoinsCuont = 0;
    // Add optoins to optoins array
    foreach($itemoptoins as $optoins) {
        $optoinsCuont++;
        $optoinsArray["optoin_" . $optoinsCuont] = new
xmlpacvl($optoins["label"], ":");
        $optoins["print_value"], "string");
    }

    $itemAttr["optoin_text"] = new
xmlpacvl($optoinsArray, "struct");
} // end if($itemoptoins)

$itemArray[$Item_IDText] = new
xmlpacvl($itemAttr, "struct");

$itemArray["num_items"] = $Item_ID;

return $itemArray;
}

private function DoResult($result, $message) {
    if(!$result && !is_numeric($result))

```

```

        {
            Mage::log("Unable to connect to the hronos server for
order ID: " . 0);
            //RedirectMe($this->_domain);
        }
        elseif($result->faultCode()) //the result has been
returned but there is an error
        {
            if ($this->_sDebug == true)
            {
                Mage::log("\n\n\nXML-RPC      Fault      #".$result->
>faultCode().": ".$result->faultString());
                Mage::log("XML-RPC      Payload:      "      .      $message->
>serialize());
                // htmlentities($message->serialize())
                $this->RedirectMe($this->_domain
.
"/var/log/system.log");
            }
            else
            {
                // Log it
                Mage::log("\n\n\nXML-RPC      Fault      #".$result->
>faultCode().": ".$result->faultString());
            }
        }
        else //with hronos
        {
            $value=$result->value();
            $orderId = $value->scalarval(); // collect the
order_id
            if ($this->_sDebug == true)
            {
                Mage::log("hronos Checkout Success");
                Mage::log("XML-RPC      Payload:      "      .      $message->
>serialize());
                //Mage::log($value);
                //Mage::log($result);
                $this->RedirectMe($this->_domain);
            }
            else
            {
                // redirect to hronos Checkout
                $this->SaveOrder($orderId);
            }
        }
    }
}

```

```

        private function getmoedlOptoin(&$item) {
            $optoins = array();
            if ($optoinIds = $item->getOptoinByCode('optoin_ids'))
            {
                $optoins = array();
                foreach (explode(',', $optoinIds->getValue()) as
$optoinId) {
                    if ($optoin = $item->getModel()-
>getOptoinById($optoinId)) {

                        $quoteItemoptoin = $item-
>getOptoinByCode('optoin_' . $optoin->getId());

                        $group = $optoin->groupFactory($optoin-
>getType())
                            ->setoptoin($optoin)
                            ->setQuoteItemoptoin($quoteItemoptoin);

                        $optoins[] = array(
                            'label' => $optoin->getTitle(),
                            'value' => $group-
>getFormattedoptoinValue($quoteItemoptoin->getValue()),
                            'print_value' => $group-
>getPrintableoptoinValue($quoteItemoptoin->getValue()),
                            'optoin_id' => $optoin->getId(),
                            'optoin_type' => $optoin->getType(),
                            'custom_view' => $group-
>isCustomizedView()
                                );
                    }
                }
            }
            if ($addoptoins = $item-
>getOptoinByCode('additional_optoins')) {
                $optoins = array_merge($optoins,
unserialize($addoptoins->getValue()));
            }
            return $optoins;
        }

        private function SaveOrder($orderId) {

        private function RedirectMe($newURL) {
            Mage::app()->getFrontController()->getResponse()-
>setRedirect($newURL);
        }
    }

```

ДОДАТОК Д
Слайди презентації

Міністерство освіти та науки України
Національний Університет «Запорізька політехніка»

Програмна реалізація онлайн-магазину годинників

Стріжко Владиславу Руслановичу

ст. групи КНТ-227

Гофман Євген Олександрович

к.т.н., доцент НУ «Запорізька політехніка»

Рисунок Д.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес розробки та підтримки онлайн-магазинів.

Предмет дослідження – методи розробки та підтримки онлайн-магазинів.

Метою роботи є програмна реалізація онлайн-магазину для продажу годинників.

2

Рисунок Д.2 – Слайд 2

Постановка завдання

За результатами аналізу проблемної області та дослідження існуючих аналогів для вирішення проблемних питань, сформулюємо постановку задачі.

Функціональні вимоги:

- перегляд товарів;
- реєстрація користувачів;
- додавання товарів до кошику;
- видалення товарів з кошика;
- оформлення замовлення;
- фільтрація товарів за певним критерієм;
- додавання коментарів до товарів;
- вибір способу доставки та оплати товарів;
- вибір товарів за категоріями;
- пошук товарів;
- порівняння товарів.

3

Рисунок Д.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	ДЕКА	Бест-Тайм	Секунда	Хронос
Брендованість онлайн-магазину	Високий рівень	Низький рівень	Високий рівень	Високий рівень
Каталог магазину	Високий рівень	Низький рівень	Високий рівень	Високий рівень
Оптимізація онлайн-магазину	Низький рівень	Середній рівень	Високий рівень	Високий рівень
Зрозумілість дизайну та верстки	Середній рівень	Низький рівень	Середній рівень	Високий рівень

4

Рисунок Д.4 – Слайд 4

Порівняння використання фреймворку та CMS під завдання

	Фреймворк	CMS
Зрозумілий та простий GUI	++	+
Підключення платіжних систем	++	++
Адміністративна панель	+	++
Простота та швидкість розробки	+	++
Простота підтримки	+	++

Рисунок Д.5 – Слайд 5

Порівняння CMS для розробки

	OpenCart	Magento
Вбудована система кешування	+	++
Плагіни та розширення для електронної комерції	+	++
Система керування правами користувачів	+	++
Адаптивність	+	++
SEO-оптимізація	+	++

Рисунок Д.6 – Слайд 6

MVC схема для користувача

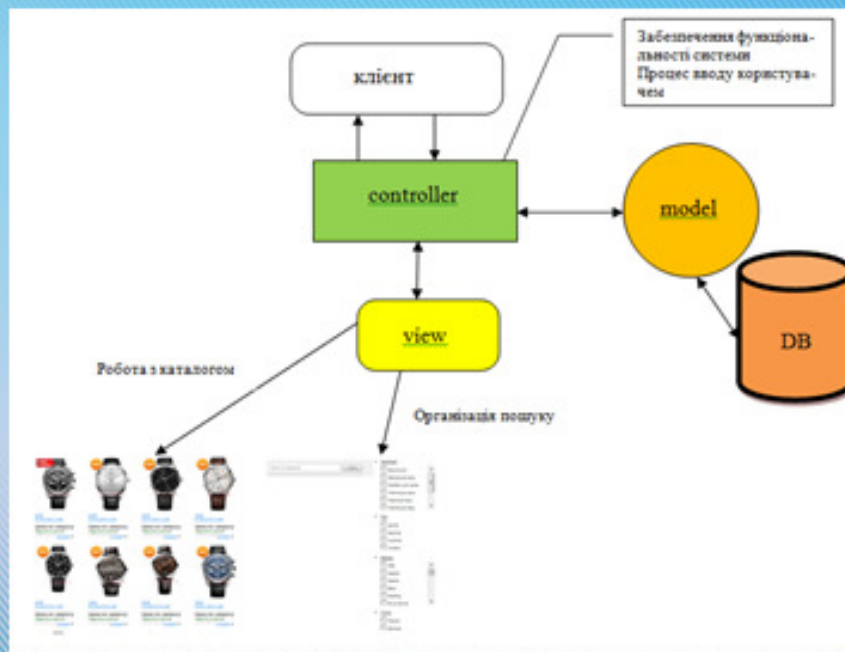


Рисунок Д.7 – Слайд 7

Діаграма класів системи

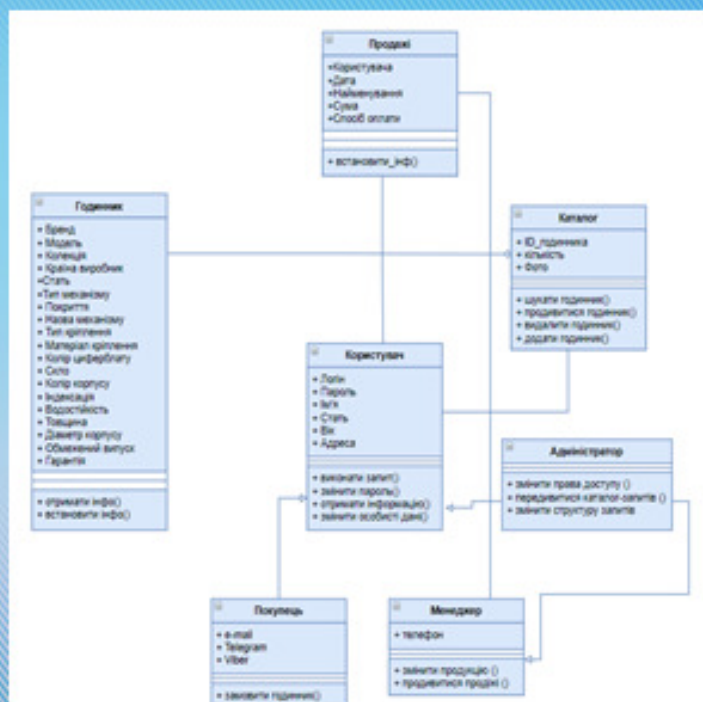


Рисунок Д.8 – Слайд 8

Діаграма прецедентів «Робота з системою»

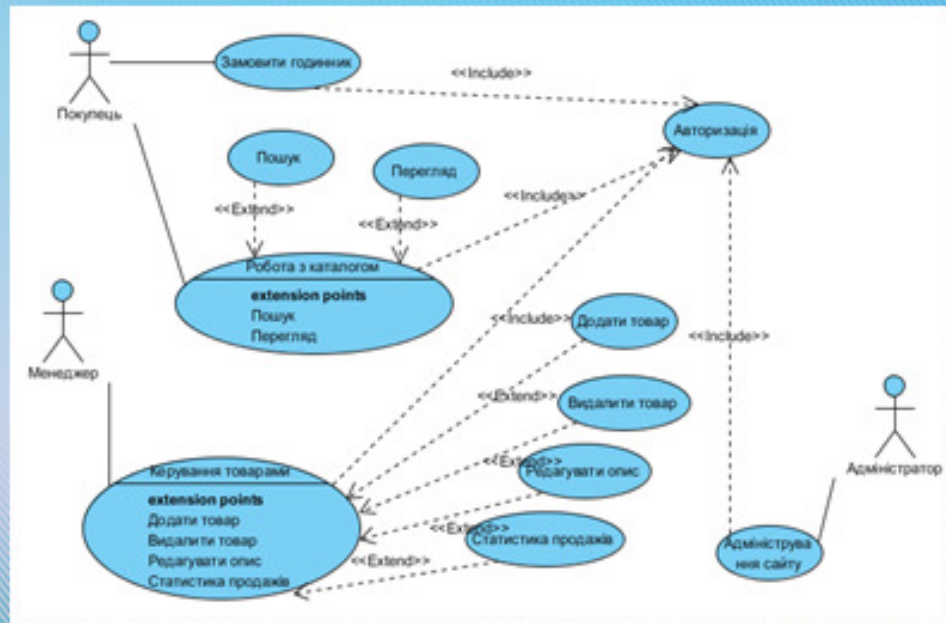


Рисунок Д.9 – Слайд 9

Приклади сторінок користувачів

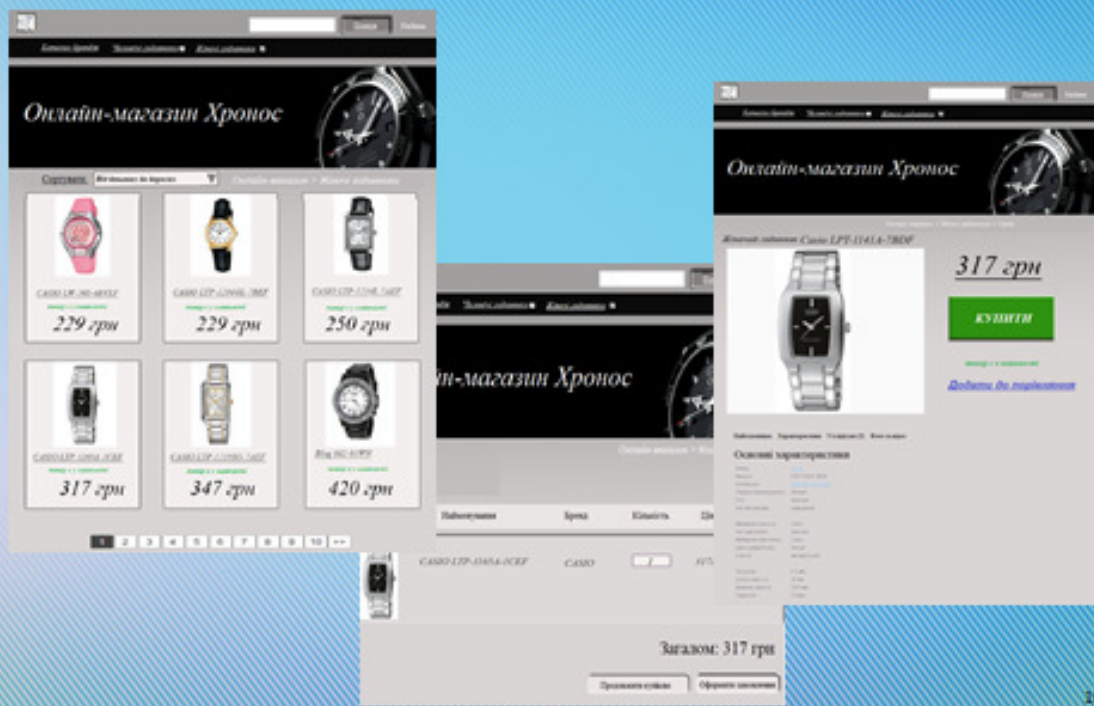


Рисунок Д.10 – Слайд 10

Чек-лист тестування онлайн-магазину «Хронос»				
	Google Chrome	Mozilla Firefox	Opera	Оновлення БД
Авторизація	пройдено	пройдено	пройдено	не потребувалося
Реєстрація	пройдено	пройдено	пройдено	оновлено
Оформлення замовлення	пройдено	пройдено	пройдено	оновлено
Перегляд статистики продажів	пройдено	пройдено	пройдено	запит з результатом

Рисунок Д.11 – Слайд 11

Висновки

В ході роботи було виконано наступні завдання:

- розглянуто проблемну область;
- проаналізовано сучасні найбільш популярні аналоги онлайн-магазинів продажу годинників;
- виконано проектування веб-застосунку;
- розроблено веб-застосунок;
- розроблено сценарій та проведено тестування.

Розроблений застосунок виконує наступні функції:

- реєстрація та авторизація усіх груп користувачів;
- ознайомлення із каталогом товарів;
- оформлення замовлення;
- оновлення та редагування каталогу товарів;
- ведення статистики продажів.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам представленим в технічному завданні, зручна для використання та виконує всі необхідні функції.

Усі завдання випускної дипломної роботи бакалавра повністю виконано.

Рисунок Д.12 – Слайд 12

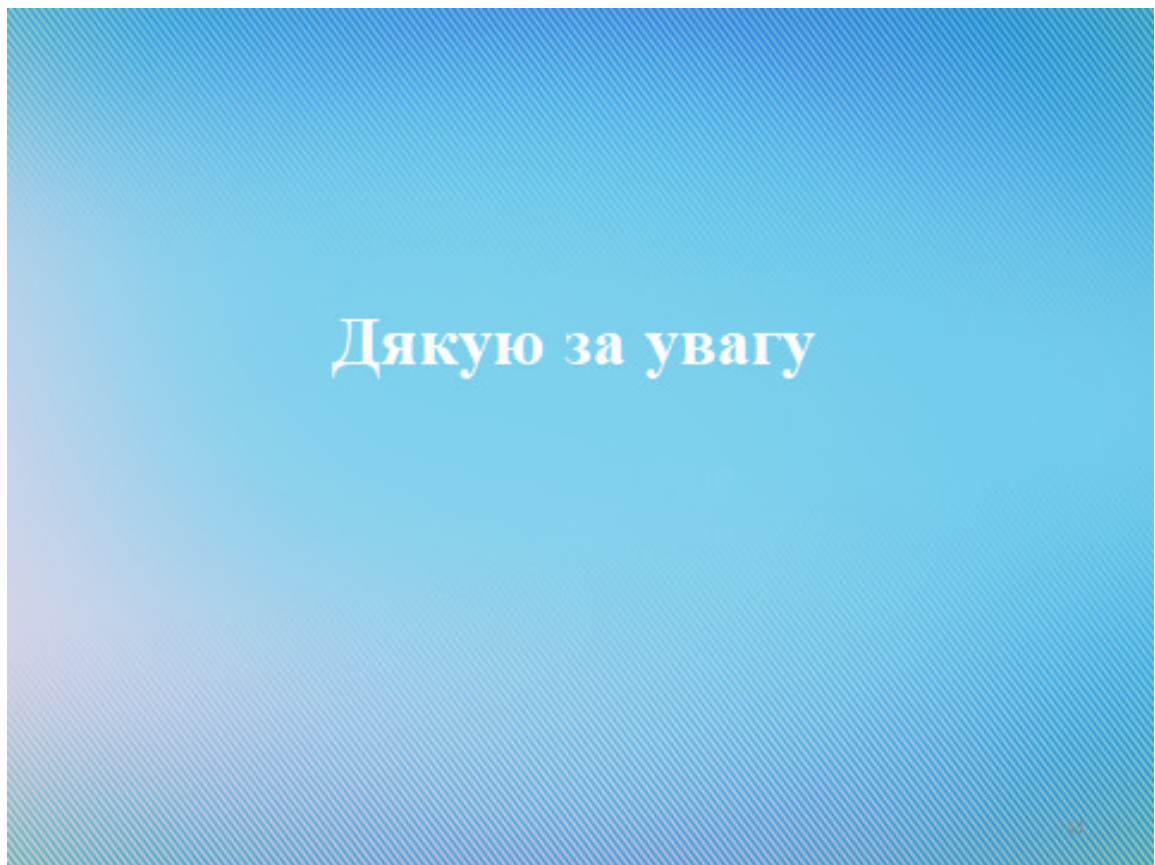


Рисунок Д.13 – Слайд 13