

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Розробка інформаційної системи товарообігу для електронної комерції»

Студента 2м курсу, бз групи,
спеціальності 121 «Інженерія
програмного забезпечення»
спеціалізації «Інженерія
програмного забезпечення»

Мороза Богдана
Валерійовича

підпис студента

Науковий керівник
кандидат педагогічних наук,
викладач кафедри інженерії
програмного забезпечення та
кібербезпеки

Котенко Наталія
Олексіївна

підпис керівника

Гарант освітньої програми
доктор технічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

Криворучко Олена
Володимирівна

підпис гаранта

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

8 листопада 2019 р.

Завдання
на випускний кваліфікаційний проект студенту

Морозу Богдану Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту «Розробка інформаційної
системи товарообігу для електронної
комерції»

Затверджена наказом ректора від 24.12.2019 №4440

2. Строк здачі студентом закінченої проекту 01 грудня 2020

3. Цільова установка та вихідні дані до проекту

Мета дослідження: проектування архітектури, розробка алгоритмів роботи
та реалізація інформаційної системи товарообігу для електронної
комерції.

Об'єкт дослідження: інформаційні технології автоматизації ведення обліку
товарів на складі, надходжень і витрат ресурсів а також інформаційної
системи товарообігу для електронної
комерції.

Предмет дослідження: використання веб-орієнтованих інформаційної системи товарообігу для електронної комерції.

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ВИЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТОВАРООБІГУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

1.1. Аналіз існуючих інформаційних систем товарообігу для електронної комерції

1.2. Порівняння аналогів програмного продукту

1.3. Висновки до розділу 1

РОЗДІЛ 2. ПРОБЛЕМИ ТА ШЛЯХИ ЇХ ВИРІШЕННЯ В ІНФОРМАЦІЙНИХ СИСТЕМАХ ТОВАРООБІГУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

2.1 Аналіз проблематики інформаційних систем товаробігу та взаємодії з клієнтами в електронній комерції

2.2 Розробка алгоритмів роботи інформаційної системи товарообігу в електронній комерції

2.3. Висновки до розділу 2

РОЗДІЛ 3. РОЗРОБКА АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТОВАРООБІГУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

3.1 Вибір архітектури інформаційної системи для електронної комерції

3.2 Проектування бази даних інформаційної системи товарообігу для електронної комерції

3.3. Висновок до розділу 3

РОЗДІЛ 4. ВИБІР ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ТА ОПИС ПРОЦЕСУ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

4.1 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

4.2 Створення структури проекту

4.3. Налаштування проекту

4.4 Створення програми

4.5 Висновки до розділу 4

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання проекту

№ пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів проекту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проекту</i>	20.09.2019	20.09.2019
2.	<i>Розробка та затвердження завдання на проект магістра</i>	08.11.2019	08.11.2019
3.	<i>Вступ та перелік літературних джерел</i>	24.01.2020	24.01.2020
4.	<i>Наукова стаття</i>	01.09.2020	01.09.2020
5.	<i>Розділ 1. Теоретичні аспекти визначення інформаційної системи товарообігу для електронної комерції</i>	25.06.2020	25.06.2020
6.	<i>Розділ 2. Проблеми та шляхи їх вирішення в інформаційних системах товарообігу для електронної комерції</i>	07.09.2020	07.09.2020
7.	<i>Розділ 3. Розробка архітектури інформаційної системи товарообігу для електронної комерції</i>	19.10.2020	19.10.2020
8.	<i>Розділ 4. Вибір інструментальних засобів та опис процесу розробки програмного продукту</i>	30.10.2020	30.10.2020
9.	<i>Висновки та пропозиції</i>	30.10.2020	30.10.2020
10.	<i>Здача випускного кваліфікаційного проекту на кафедрі (перша перевірка)</i>	05.11.2020	05.11.2020
11.	<i>Підготовка автореферату та презентації доповіді</i>	05.11.2020	05.11.2020
12.	<i>Попередній захист випускного кваліфікаційного проекту</i>	25.11.2020- 27.11.2020	25.11.2020 -27.11.2020

13.	<i>Зовнішнє рецензування випускного кваліфікаційного проекту</i>	01.12.2020	01.12.2020
14.	<i>Підготовка до публічного захисту випускного кваліфікаційного проекту</i>	10.12.2020- 11.12.2020	

7. Дата видачі завдання «8» листопада 2019 р.

8. Науковий керівник випускного кваліфікаційного проекту

Котенко Н.О

(прізвище, ініціали, підпис)

9. Гарант освітньої програми

Криворучко О.В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент

Мороз Б.В.

(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

Відмітка про попередній захист _____

(підпис, дата)
Котенко Н.О.
(ПІБ, підпис, дата)

Випускний кваліфікаційний проект студента Мороз Б.В.
(прізвище, ініціали)

Гарант освітньої програми _____ Криворучко О. В.
(прізвище, ініціали, підпис)

Завідувач кафедри _____ Криворучко О. В.
(підпис, прізвище, ініціали)

« _____ » 20 ____ г.

АНОТАЦІЯ

Випускна робота магістра складається з інформаційної системи товарообігу для електронної комерції та пояснювальної записки.

Метою роботи є розробка даної системи для електронної комерції, що містить модулі збереження та перегляду товарів у базі даних, пошуку товарів за назвою та модуль проведення дослідження прибутковості товару.

В роботі визначено основні завдання на розробку системи, проаналізовано аналоги розробленої системи, обґрунтовано вибір архітектурного патерна для реалізації системи – MVC. Наведено сценарії роботи програмного комплексу, загальна структура програмного комплексу, опис бази даних системи (СУБД MySQL), опис алгоритмів взаємодії окремих модулів системи, опис об'єктної структури системи (діаграма класів), алгоритми роботи основних модулів, реалізованих з використанням мови програмування Java.

Ключові слова: база даних, інформаційна система, електронна комерція, товарообіг

ABSTRACT

The final master's degree consists of a software package of information system of ware audit for e-commerce and an explanatory note.

The goal of the work is to develop this system for e-commerce containing modules for storing and reviewing goods in a database, searching for products by name, and a module for conducting a profitability study for a product.

In this work the main tasks for the development of the system are determined, analogs of the developed system are analyzed, the choice of the architectural paternity for the implementation of the system - MVC is substantiated. The scenarios of the program complex work, the general structure of the software complex, the description of the database of the system (MySQL database), description of the algorithms of interaction of the individual modules of the system, the description of the object structure of the system (class diagram), algorithms of the basic modules implemented with the use of the Java programming language.

Keywords: data base, information system, e-commerce, commodity circulation.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CRUD – create, read, update, delete (з *англ.* створити, прочитати, оновити, видалити)

ERM – entity-relationship model (з *англ.* модель сутність — зв'язок)

MVC – Model–View–Controller (з *англ.* Модель–Представлення–Контролер)

IDE – Intelligent Development Environment (з *англ.* Інтелектуальне середовище розробки)

JVM – Java Virtual Machine (з *англ.* Віртуальна машина Java)

CRM – Customer relationship management

ООП – об'єктно-орієнтоване програмування

ПЗ – програмне забезпечення

БД – база даних

					КНТЕУ 121 06з-06.МР			
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка інформаційної системи товарообігу для електронної комерції	Стадія	Арку	Аркушів
Зав. каф.		Криворучко О.В.		19.10.20		ПС	2	53
Керівник		Котенко Н. О.		19.10.20		Факультет інформаційних технологій 2м курс, бз група		
Гарант		Криворучко О.В.		19.10.20				
Розробив		Мороз Б. В.		19.10.20				
					Перелік умовних скорочень			

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ВИЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТОВАРООБІГУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ.....	7
1.1. Аналіз існуючих інформаційних систем товарообігу для електронної комерції.....	7
1.2. Порівняння аналогів програмного продукту.....	9
1.3. Висновки до розділу 1.....	16
РОЗДІЛ 2. ПРОБЛЕМИ ТА ШЛЯХИ ЇХ ВИРІШЕННЯ В ІНФОРМАЦІЙНИХ СИСТЕМАХ ТОВАРООБІГУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ.....	17
2.1. Аналіз проблематики інформаційних систем товарообігу та взаємодії з клієнтами в електронній комерції.....	17
2.2. Розробка алгоритмів роботи інформаційної системи товарообігу в електронній комерції.....	19
2.3. Висновки до розділу 2.....	23
РОЗДІЛ 3. РОЗРОБКА АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТОВАРООБІГУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ.....	24
3.1. Вибір архітектури інформаційної системи для електронної комерції.....	24
3.2. Проектування бази даних інформаційної системи товарообігу для електронної комерції.....	26
3.3. Висновок до розділу 3.....	32
РОЗДІЛ 4. ВИБІР ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ТА ОПИС ПРОЦЕСУ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ.....	33
4.1. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення.....	33
4.2. Створення структури проекту.....	41
4.3. Налаштування проекту.....	43
4.4. Створення програми.....	44

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		4

4.5. Висновки до розділу 4.....	50
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	52

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		4

ВСТУП

Актуальність. Облік товарів, підрахунок витрат та надходжень займають дуже багато часу. Адже якщо це все робити в ручну на папері, чи навіть на комп'ютері, не маючи для цього спеціального програмного забезпечення, людина фізично не зможе швидко обробити велику кількість однотипної інформації. А так, як час є одним із найважливіших і найдорожчих людських ресурсів виникає велика необхідність у спрощенні та автоматизації цих процесів. Використання якісного програмного забезпечення запобігає виникненню помилки спричиненою людським фактором.

Мета дослідження: проектування архітектури, розробка алгоритмів роботи та реалізація інформаційної системи товарообігу для електронної комерції.

Об'єкт дослідження: інформаційні технології автоматизації ведення обліку товарів на складі, надходжень і витрат ресурсів а також інформаційної системи товарообігу для електронної комерції.

Предмет дослідження: використання веб-орієнтованих інформаційної системи товарообігу для електронної комерції.

У відповідності з метою дослідження поставлені наступні завдання:

- Провести аналіз аналогів програмного продукту
- Визначити інструменти необхідні для розробки
- Розробити інформаційну систему товарообігу

Методи дослідження: аналітичний, теоретичний, практичний.

Наукова новизна дослідження полягає в удосконаленій інформаційній системі товарообігу.

					<i>КНТЕУ 121 06з-06.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			24.01.20	Розробка інформаційної системи товарообігу для електронної комерції	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Котенко Н.О.			24.01.20		<i>В</i>	5	53
Гарант	Криворучко О.В.			24.01.20		Факультет інформаційних технологій 2м курс, 6з група		
Розробив	Мороз Б.В.			24.01.20				
					<i>Вступ</i>			

Практичне значення дослідження: спрощення та пришвидшення операцій пов'язаних з товарообігом для електронної комерції в малому та середньому бізнесі.

					<i>КНТЕУ 121 063-06.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		6

РОЗДІЛ 1.

ТЕОРЕТИЧНІ АСПЕКТИ ВИЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТОВАРООБІГУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

1.1. Аналіз існуючих інформаційних систем товарообігу для електронної комерції

Поширеною версією інформаційної системи товарообігу для електронної комерції є CRM системи.

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

					<i>КНТЕУ 121 06з-06.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка інформаційної системи товарообігу для електронної комерції	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко О.В.		25.06.20		РІ	7	53
Керівник		Котенко Н.О.		25.06.20		Факультет інформаційних технологій 2м курс, бз група		
Гарант		Криворучко О.В.		25.06.20				
Розробив		Мороз Б.В.		25.06.20	Теоретичні аспекти визначення інформаційної системи товарообігу для електронної комерції			



Рис. 1.1. Типи компонентів CRM системи

Головною причиною впровадження CRM системи на підприємствах, котрі займаються електронною комерцією, є економія часу працівників на ведення обліку товарів, надходжень, витрат та збереження інформації про клієнтів що може допомогти у подальшій економічній діяльності компанії. Керівництво, в свою чергу, матиме можливість швидше отримувати точні звіти про діяльність підприємства.

Такий підхід може допомогти досягти великих прибутків за більш короткий проміжок часу.

Багато людей різних професій цікавляться тим, як зробити власний бізнес, а зокрема продаючи товари через мережу інтернет. Такий підхід є одним із найменш ризикованих способів розпочати власну справу. Адже для цього не обов'язково мати приміщення і наймати персонал – це все можна

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

робити самому. Достатньо комп'ютера, місця для збереження товару і бажання продавати.

Але з часом успішне підприємство починає зростати і таким чином може виникнути потреба у програмному забезпеченні котре дало б можливість пришвидшити процес ведення інформації про товари, витрати, прибутки, клієнтів тощо.

Загалом робота з інформаційною системою товарообігу для електронної комерції виглядає наступним чином:

- 1 Внесення товарів, що є в наявності на складі, та їх характеристик у базу даних.
- 2 Підтримка роботи існуючої системи управління.

Нами було проведено аналіз аналогів програмного продукту наведеного в даній роботі і було виявлено що існує чимало схожих систем, хоча кожна з них виконує дещо різні функції та має свої відмінності, а зокрема переваги та недоліки, котрі роблять їх унікальними. Таким чином у ході проведення аналізу нам вдалося виявити такі системи як:

- 1C:CRM;
- Бітрікс24;
- Microsoft Dynamics CRM;
- OneBox;
- KeepinCRM.

Цей список не є остаточним, чи вичерпним. Він базується на результатах пошукової системи Google за запитом «CRM система».

1.2. Порівняння аналогів програмного продукту

Для більш точного і якісного порівняння продуктів нам довелося визначити їхні сильні та слабкі сторони. Таким чином це дасть змогу краще

					КНТЕУ 121 063-06.МР	Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата		

розуміти теперішній ринок CRM систем для електронної комерції, вимоги користувачів до програмного забезпечення та рівень конкуренції серед аналогічних продуктів.

Програмне забезпечення 1С:CRM

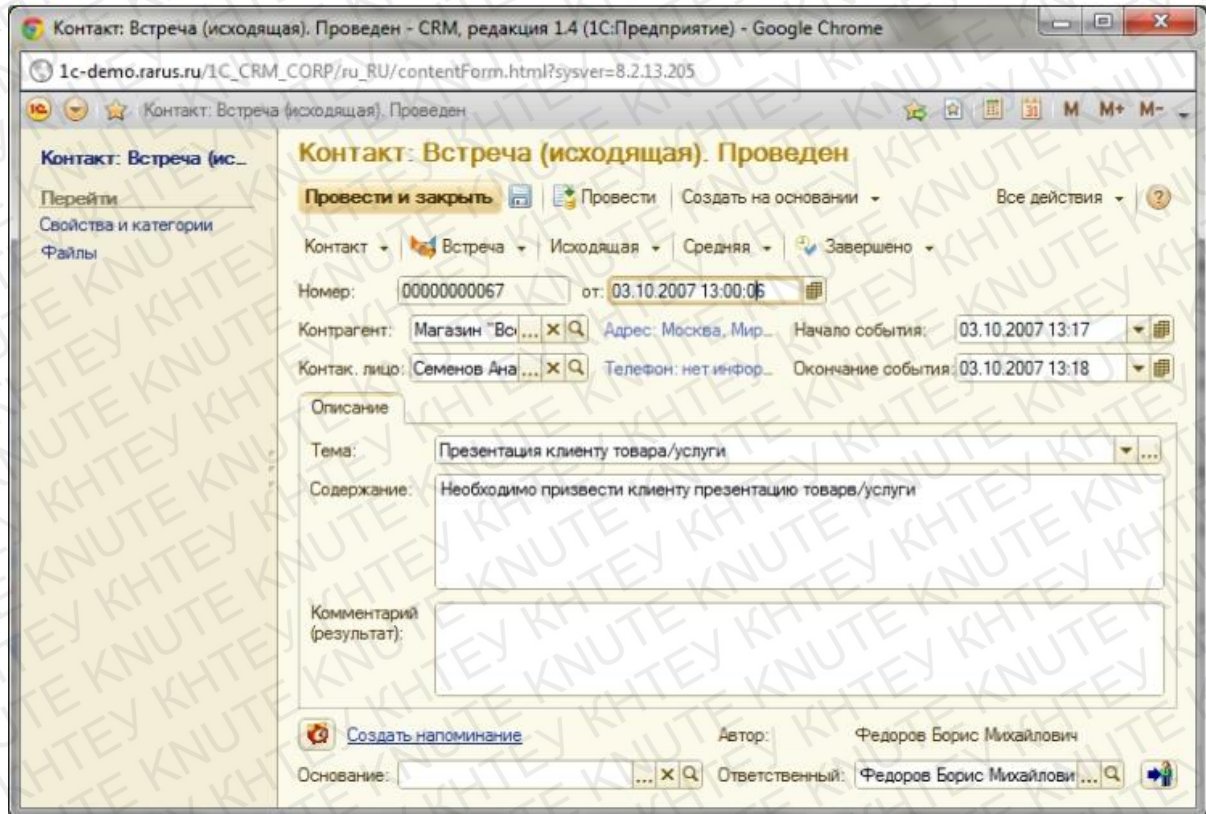


Рис. 1.2. Приклад застосування системи «1С:CRM»

«1С:CRM» — це додатковий модуль програмного забезпечення 1С. Він забезпечує управління базою контактів, ведення історії взаємин з клієнтами, управління продажами, маркетингом, бізнес-процесами, має вбудований Email-клієнт і інтегрується з Outlook. У даному модулі можливий онлайн доступ через термінальний клієнт, web-інтерфейс або мобільний клієнт.

Після огляду даного ПЗ, можемо відмітити приємний дизайн і конкретизованість інформації.

До переваг відносяться наступні пункти:

- 1 Поширеність у країнах СНД.

						Аркуш
						10
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 063-06.МР	

- 2 Можливість ведення історії взаємин з клієнтами.
- 3 Легкий доступ з будь-якого сучасного пристрою.
- 4 Власна мова програмування.

До недоліків відносяться наступні пункти:

- 1 Інтерфейс ПЗ є дещо нагромадженим.
- 2 Складність в освоєнні.
- 3 Платна ліцензія.

Отже, можемо зробити висновок, що система «1С:CRM» є доволі популярною у країнах, що входять до співдружності незалежних держав. Це означає що вона з самого початку була орієнтована на даний ринок, і в ній добре враховані особливості ведення бізнесу притаманні людям що проживають в країнах СНД.

Власна мова програмування дає можливість гнучко підлаштовувати програмне забезпечення під різні потреби користувачів, додавати нові модулі, покращувати існуючі тощо.

Програмне забезпечення Бітрікс24

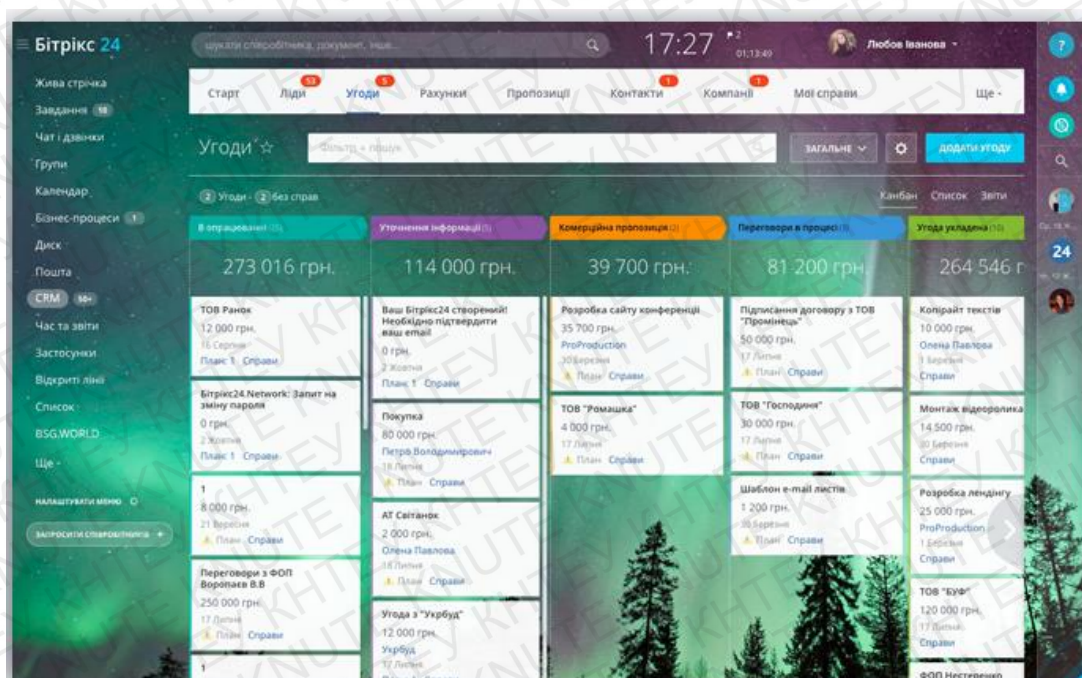


Рис. 1.3. Скріншот програми «Бітрікс24»

«Бітрікс24» – це система яка допомагає вести відносини з клієнтами відслідковувати продажі та вести певні підрахунки, яка застосовується по всьому світі. Програма містить у собі мікроблоги, завдання, файлоховище (з контролем версій), календар, фотогалереї, месенджер, CRM, бізнес-процеси, облік робочого часу, профайли, звіти та інше.

Після огляду даного ПЗ, можемо відмітити приємний дизайн і конкретизованість інформації.

До переваг відносяться наступні пункти.

- 1 Поширеність в країнах СНД.
- 2 Потужний функціонал.
- 3 Наявність мобільної версії.

До недоліків відносяться наступні пункти.

- 1 Громіздкість системи.
- 2 Бажання поєднати в собі купу різних інструментів.

Можемо помітити, що «Бітрікс24», як і попередня система, поширена в країнах що входять в склад СНД, але має дещо приємніший дизайн, та більш інтуїтивну навігацію.

Програмне забезпечення Microsoft Dynamics CRM



Рис. 1.4. Скріншот програми «Microsoft Dynamics CRM»

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		12

«Microsoft Dynamics CRM» — це CRM система зі "знайомим" інтерфейсом продуктів корпорації Microsoft. Містить інструменти для управління продажами, маркетингом, сервісом і бізнес-процесами. Веб-інтерфейс доступний через Internet Explorer. Можлива робота з системою прямо через Outlook.

Після огляду даного ПЗ, можемо відмітити приємний дизайн велику кількість графіків.

До переваг відносяться наступні пункти.

- 1 Повна інтеграція з платформою і додатками Microsoft.
- 2 Потужний функціонал.
- 3 Наявність мобільної версії.

До недоліків відносяться наступні пункти.

- 1 Складність в освоєнні.
- 2 Мала поширеність в країнах СНД.

Дана система має суттєву перевагу, адже вона легко інтегрується з одним із найпопулярніших пакетів офісного програмного забезпечення, що робить її легкою в освоєнні. Адже продукти корпорації Microsoft мають схожий дизайн, і добре оптимізовані під операційну систему Windows.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

Програмне забезпечення OneBox

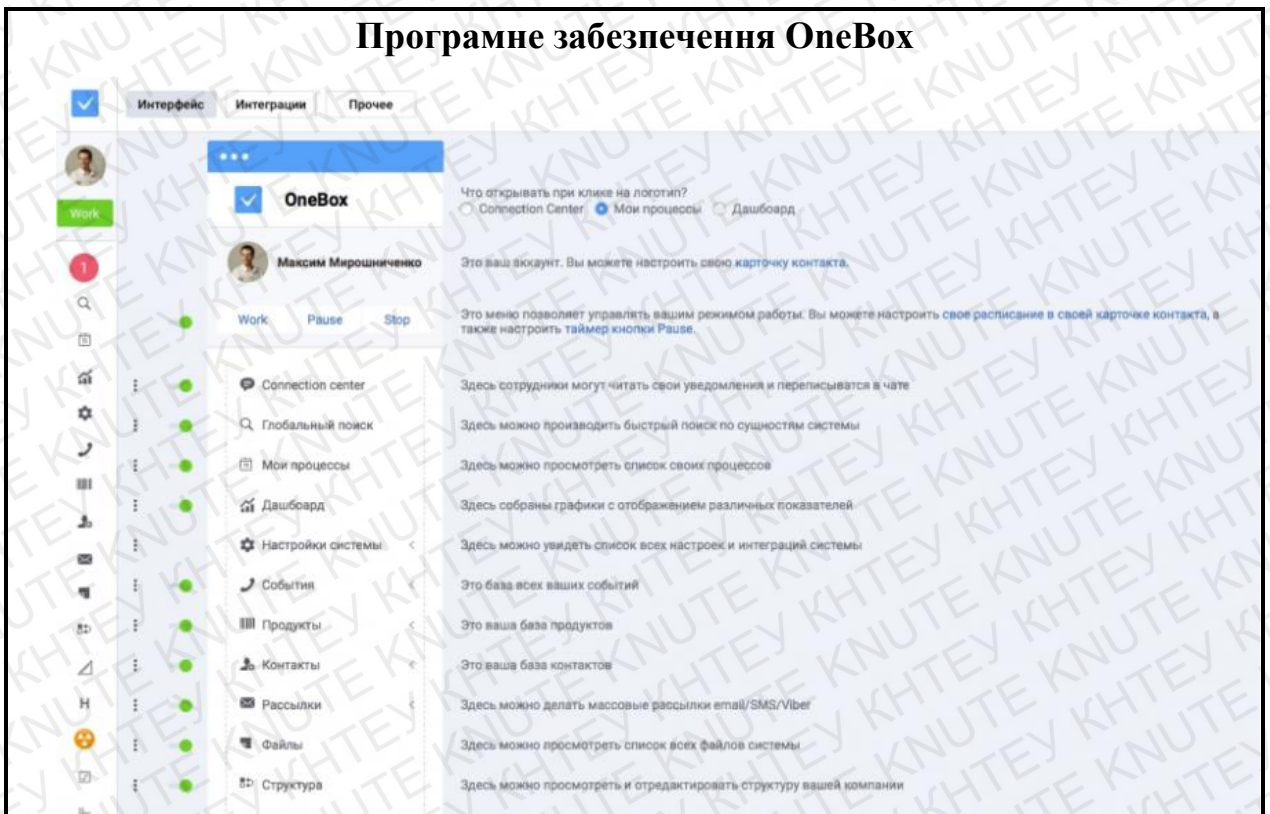


Рис. 1.5. Скріншот програми «OneBox»

«OneBox» — це CRM програма, яка може працювати сама, у фоновому режимі, не вимагаючи уваги людини. Це робот на сторожі вашого бізнесу, який допомагає робити рутинні або складні операції, які вам набридли, але робити їх потрібно.

Після огляду даного ПЗ, можемо відмітити сучасний дизайн та простий інтерфейс.

До переваг відносяться наступні пункти.

- 1 Можливість додавати власні функції.
- 2 Приємний дизайн.

До недоліків відносяться наступні пункти.

- 1 Складність в освоєнні.

Головною відмінністю системи «OneBox», є її безмежність. Користувач може отримати ідеально налаштовану CRM систему з унікальними функціями.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 063-06.МР	14

Дане програмне забезпечення є дуже поширеним в Україні, а отже добре підходить для ведення бізнесу в країна СНД.

Програмне забезпечення KeerpinCRM

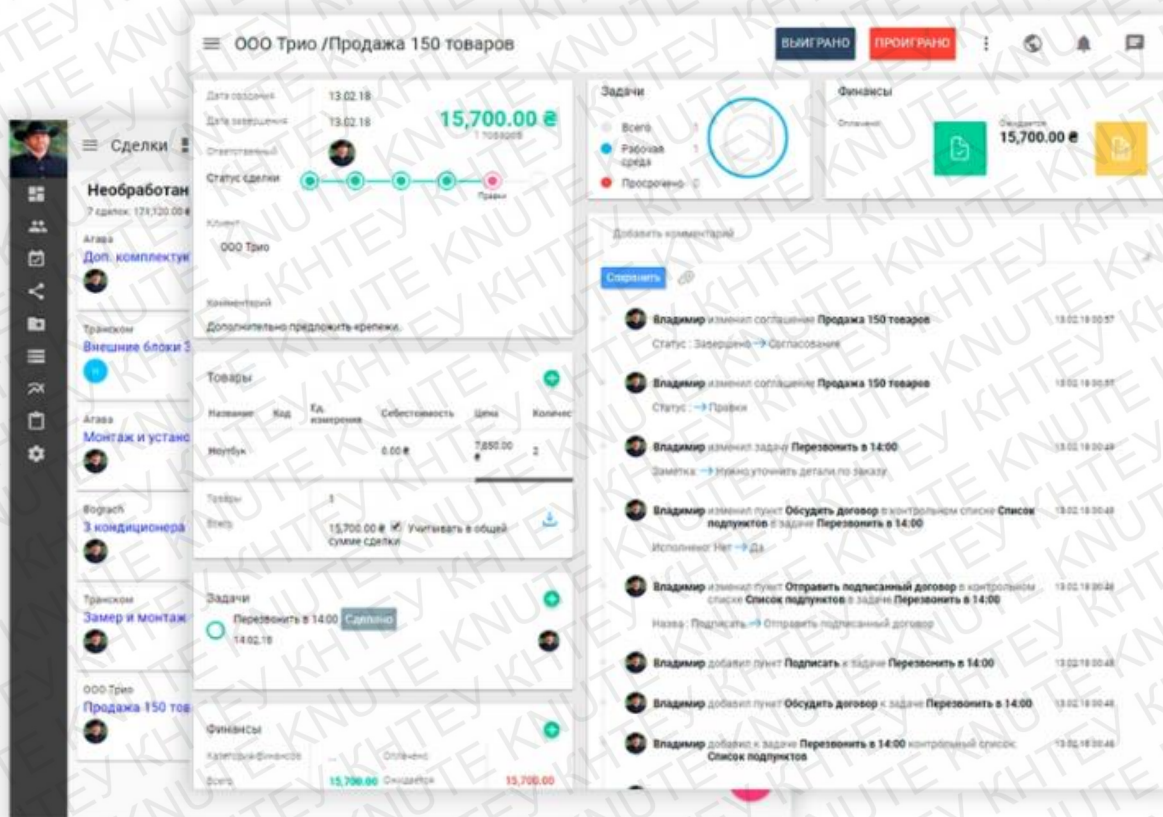


Рис. 1.6. Приклад застосування системи «KeerpinCRM»

«KeerpinCRM» – це онлайн CRM система для малого та середнього бізнесу, котра має зручний та інтуїтивно зрозумілий інструмент для оптимізації роботи з клієнтами, завданнями, угодами, складом, прайс-листом, введенням фінансів і документів.

Після огляду даного ПЗ, можемо відмітити приємний дизайн і конкретизованість інформації.

До переваг відносяться наступні пункти:

- 1 Поширеність у країнах СНД.
- 2 Легка в освоєнні.
- 3 Зрозумілий інтерфейс.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		15

До недоліків відносяться наступні пункти.

1 Платна ліцензія.

Система «КееріпCRM» є досить простою в освоєнні, що добре підходить для початку малого чи середнього бізнесу, без певного підґрунтя в даній сфері. Проте вона намагається поєднати в собі варіанти підприємств, що робить менш зосередженою на чомусь одному.

1.3. Висновки до розділу 1

Отже, у даному розділі було наведено визначення CRM системи, як прикладу інформаційної системи товарообігу. Проведено аналіз доступних на ринку аналогів програмного забезпечення інформаційної системи товарообігу для електронної комерції.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16

РОЗДІЛ 2.

ПРОБЛЕМИ ТА ШЛЯХИ ЇХ ВИРІШЕННЯ В ІНФОРМАЦІЙНИХ СИСТЕМАХ ТОВАРООБІГУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

2.1. Аналіз проблематики інформаційних систем товаробігу та взаємодії з клієнтами в електронній комерції

У ході аналізу та порівняння аналогів нами було виявлено що на ринку є достойні конкуренти, правда вони мають ряд недоліків таких, як:

1. нагромадженість інтерфейсу;
2. платна ліцензія;
3. непоширеність в країнах СНД;
4. складність в освоєнні;
5. зайвий функціонал;
6. відстуність масштабування.

Серед наведених вище проблем, хотілося б виділити: нагромадженість інтерфейсу, платну ліцензію та відсутність масштабування, так як на мою думку вони є наріжними каміннями для тих хто починає бізнес і тільки набуває знань як вести бізнес так і для тих хто хоче додатковий функціонал. До таких категорій підприємців хотілося б віднести власників малого бізнесу, оскільки в умовах карантину більшість з них змушені перенести власний бізнес в мережу Інтернет.

					<i>КНТЕУ 121 06з-06.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.	Криворучко О.В.			07.09.20	Розробка інформаційної системи товарообігу для електронної комерції	Стадія	Аркуш	Аркушів
Керівник	Котенко Н.О.			07.09.20		P2	17	53
Гарант	Криворучко О.В.			07.09.20		Факультет інформаційних технологій 2м курс, бз група		
Розробив	Мороз Б.В.			07.09.20				
					Проблеми та шляхи їх вирішення в інформаційних системах товарообігу для електронної комерції			

Мета створення малого бізнесу – це отримання прибутку та/або самореалізація. Для малого бізнесу характерні відносно невеликі розміри, обмежена кількість людей, обмежений обіг капіталу, локальний район операцій та конкретний напрям роботи. Він відіграє значну роль у забезпеченні зайнятості більшості розвинених країн та країн, які активно розвиваються (в США майже третина робочих місць припадає на фірми, де працює менше 100 осіб), певною мірою забезпечує насичення ринку споживчими товарами та послугами, сприяє послабленню монополізму, розвитку конкуренції, структурній перебудові економіки, є засобом досягнення особистого успіху та активно формує середній клас людей. Малий бізнес — важливе джерело інновації.

Для вирішення перелічених вище проблем потрібно реалізувати інформаційну систему товарообігу електронної комерції котра буде працювати незалежно від реалізації фронт-енду (сайту, частини з якою працює користувач) та буде проектом з відкритим кодом, що дасть можливість будь-кому ознайомитися з структурою програмного забезпечення, модернізувати його і правильно налаштовувати клієнтську частину відповідно до документації серверної частини.

Проте навіть з таким способом цей проект може бути прибутковим завдяки:

- курсу відео уроків стосовно як правильно інтегрувати дану систему свій бізнес та які кроки краще зробити на старті
- можливості зробити систему “під ключ” відповідно до побажань клієнта
- укласти договір, що на сайті користувача обов’язково має бути медійна реклама котра буде монетизуватися

					КНТЕУ 121 063-06.МР	Аркуш
						18
Зм.	Аркуш	№ докум	Підпис	Дата		

2.2. Розробка алгоритмів роботи інформаційної системи товарообігу в електронній комерції

Для успішного підприємця важливо знати своїх клієнтів, свій продукт і вчасно реагувати на будь-які зміни. А для того щоб ці зміни відслідкувати потрібна аналітика. Отже у даній системі товарообігу буде три основних функції:

- 1 Збереження інформації в базі даних.
- 2 Отримання інформації з бази даних.
- 3 Аналіз інформації та виведення результату.

Збереження інформації в базі даних є одним з ключових факторів взаємодії з інформаційною системою, адже якщо не буде інформації для обробки то вся система стає непотрібною. Таким чином, перш за все, для початку роботи потрібно хоча б мінімально заповнити базу даних інформацією про товари, їх закупівельну вартість, кількість, фактичну ціну тощо. А коли вже з'являться перші клієнти їх потрібно додавати у список контактів, адже якщо вони будуть задоволені сервісом, то є велика ймовірність що вони прийдуть ще раз.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		19

Алгоритм роботи модуля збереження даних наведений на рисунку 2.1.

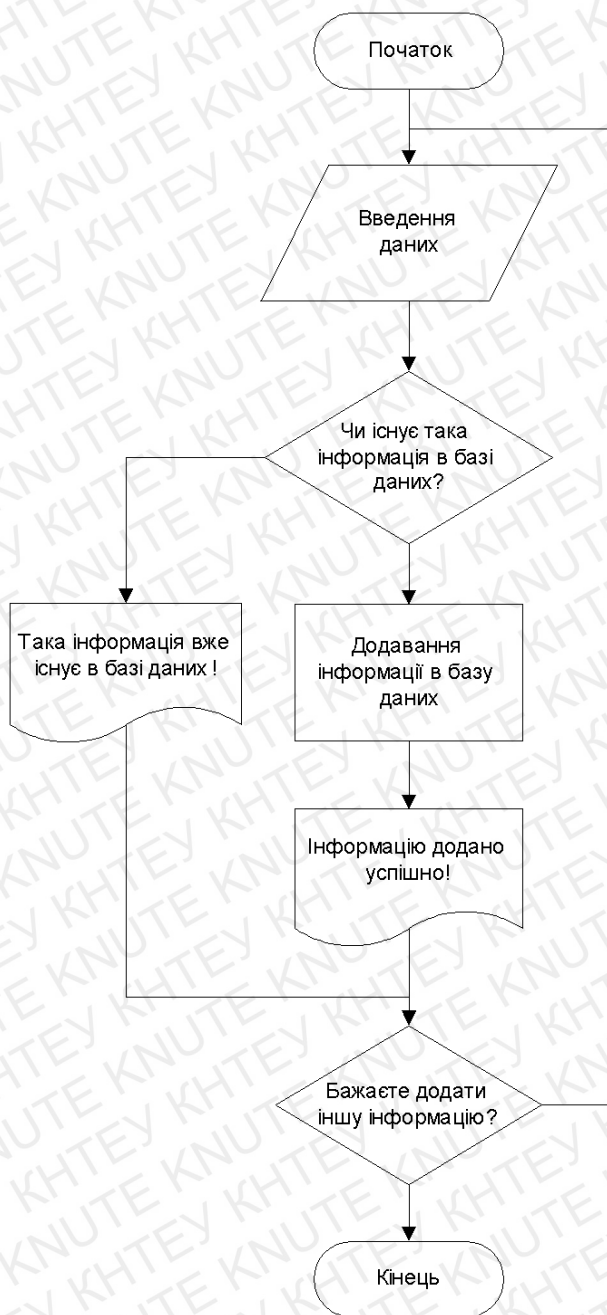


Рис. 2.1. Блок-схема, алгоритм роботи модуля збереження інформації в базу даних

Разом зі збереженням інформації до бази даних, виникає потреба у її відтворенні. Тому нами було спроектовано модуль відтворення інформації бази даних. Алгоритм роботи цього модуля наведений на рисунку 2.2.



Рис. 2.2. Блок-схема, алгоритм роботи модуля відтворення інформації з бази даних

Можливість запису та читання інформації з бази даних, значно спрощує та пришвидшує процес обробки даних. За допомогою вибірки, можна фільтрувати отриманий результат з бази даних, щоб відсіяти непотрібні товари, контакти тощо. Проте зі збільшенням обсягів даних з'являється можливість обробки цих даних, знаходження закономірностей котрі могли б

допомогти у прийнятті певних рішень таких, як визначення цільової аудиторії, її інтересів, фаз активності (день, ніч; зима, літо, тощо), розрахунок прибутків, витрат, надходжень товару. На постійне проведення такого аналізу витрачається дуже багато часу.

Таким чином ми вирішили створити аналітичний модуль (Рисунок 2.3), котрий робитиме ці всі рутинні справи автоматично, що дасть змогу працівникам займатися іншими важливими справами.



Рис. 2.3. Блок-схема, алгоритм роботи аналітичного модуля

Розроблені вище алгоритми дають змогу краще зрозуміти як правильно проектувати систему, так як мають працювати її компоненти.

2.3. Висновки до розділу 2

Отже, у даному розділі було розглянуто основні проблеми та недоліки інформаційних системи на ринку. Визначено конкретні вимоги до інформаційної системи. Розроблено та спроектовано блок-схеми для основних модулів програмного забезпечення.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		23

РОЗДІЛ 3.

РОЗРОБКА АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТОВАРООБГУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

3.1. Вибір архітектури інформаційної системи для електронної комерції

Для реалізації веб-орієнтованої інформаційної системи товарообігу для електронної комерції нами було обрано архітектурний шаблон проектування MVC, адже він забезпечує гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах сприяє впорядкованості їхньої структури і робить їх більш зрозумілими за рахунок зменшення складності.

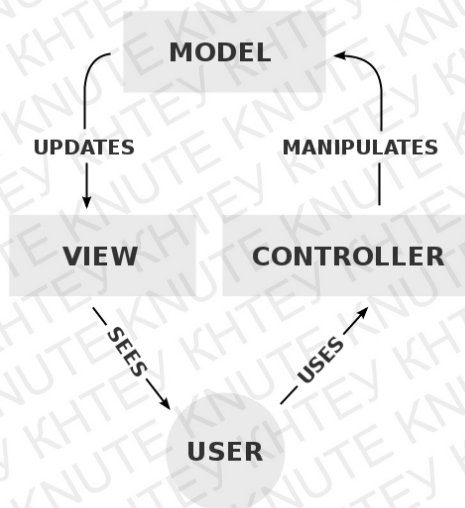


Рис. 3.1. Схема реалізації MVC шаблону

					<i>КНТЕУ 121 06з-06.МР</i>		
Зм.	Аркуш	№ докум.	Підпис	Дата			
Зав. каф.	Криворучко О.В.			19.10.20	Розробка інформаційної системи товарообігу для електронної комерції	Стадія	Аркуш
Керівник	Котенко Н.О.			19.10.20		РЗ	24
Гарант	Криворучко О.В.			19.10.20		Факультет інформаційних технологій 2м курс, 6з група	
Розробив	Мороз Б.В.			19.10.20			
					Розробка архітектури інформаційної системи товарообігу для електронної комерції	Аркуші	53

У рамках архітектурного шаблону модель–вигляд–контролер (MVC) програма поділяється на три окремі, але взаємопов'язані частини з розподілом функцій між компонентами. Модель (Model) відповідає за зберігання даних та їх структуру. Вигляд (View) відповідальний за представлення цих даних користувачеві, тобто інтерфейс програми. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача (зміна положення курсора миші, натискання кнопки, ввід даних в текстове поле) і передає дані у модель.

- **Модель** є центральним компонентом шаблону MVC і відображає поведінку додатку, незалежну від інтерфейсу користувача. Модель стосується прямого керування даними, логікою та правилами застосунку.
- **Вигляд** може являти собою будь-яке представлення інформації, одержуване на виході, наприклад графік чи діаграму. Одночасно можуть співіснувати кілька виглядів (представлень) однієї і тієї ж інформації, наприклад гістограма для керівництва компанії й таблиці для бухгалтерії.
- **Контролер** одержує вхідні дані й перетворює їх на команди для моделі чи вигляду.

Модель інкапсулює (приховує) ядро даних в основний функціонал їхньої обробки і не залежить від процесу вводу чи виводу даних.

Вигляд може мати декілька взаємопов'язаних областей, наприклад різні таблиці і поля форм, в яких відображаються дані.

У функції контролера входить відстеження визначених подій, що виникають в результаті дій користувача. Контролер дозволяє структурувати код шляхом групування пов'язаних дій в окремий клас. Наприклад у типовому MVC-проекті може бути користувацький контролер, що містить групу

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		25

методів, пов'язаних з управлінням обліковим записом користувача, таких як реєстрація, авторизація, редагування профілю та зміна пароля.

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через контролер внесе зміни до моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.

Таким чином, можна зробити висновок, що даний архітектурний шаблон забезпечує всі технологічні можливості для побудови гнучкого додатку із дотриманням принципів SOLID. В нашому випадку такий підхід забезпечить не тільки успішну реалізацію проекту, а й подальшу його модифікацію та розвиток у відповідності до зміни вимог користувача та процедур обробки інформації.

3.2. Проектування бази даних інформаційної системи товарообігу для електронної комерції

Для того щоб побудувати ефективну систему, котра буде швидко працювати потрібно правильно спроектувати базу даних, адже саме база даних є ядром інформаційної системи.

					<i>КНТЕУ 121 063-06.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		26

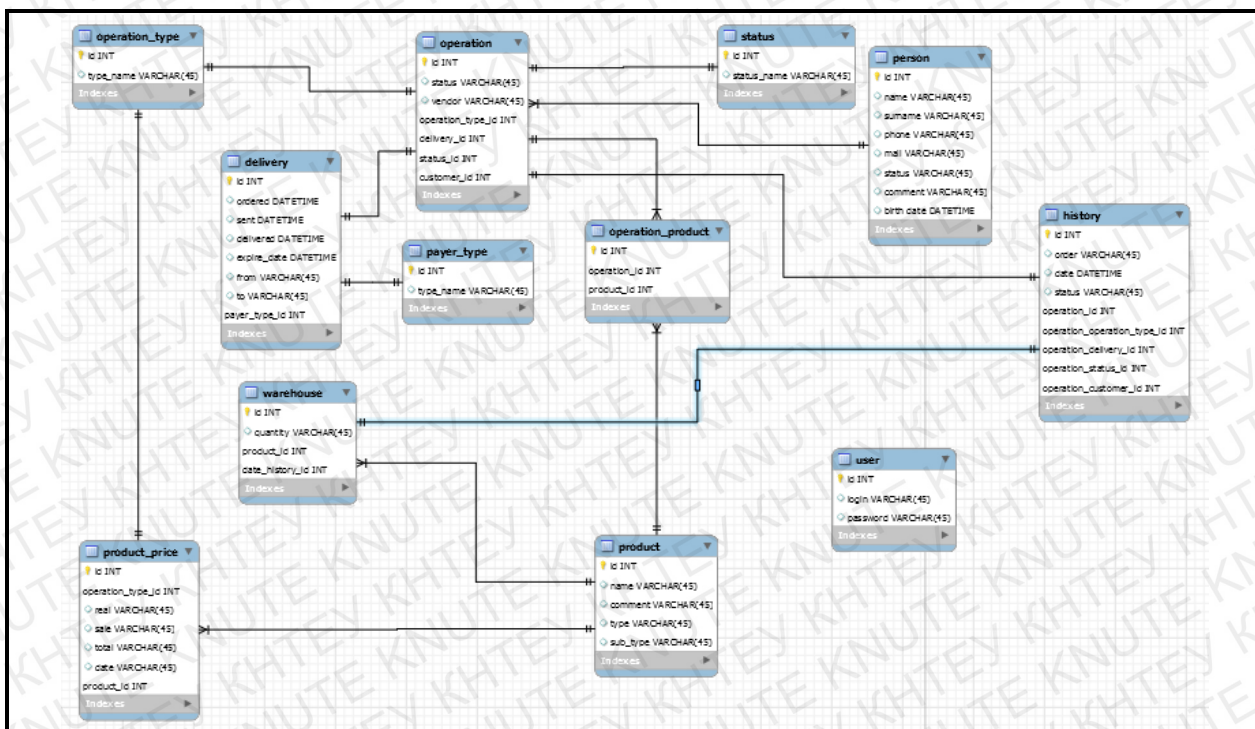


Рис. 3.2. Структура бази даних інформаційної системи товарообігу для електронної комерції

Розроблена структура бази даних є оптимізованою під головні модулі системи, та вміщає в собі такі таблиці:

- warehouse (вміщає в собі інформацію стосовно наявності товарів, їх кількості, ціни, дати поставки тощо);
- person (вміщає в собі інформацію стосовно продавця або клієнта, його прізвище, ім'я, телефон, пошту, адресу, статус а також коментар);
- user (вміщає в собі логін та пароль, котрі використовуються для валідації користувача);
- product (вміщає в собі інформацію стосовно назви продукту, типу, підтипу та коментаря);
- product_price (вміщає в собі інформацію стосовно ціни товару, котра залежить від типу операції: продаж або купівля. Від цього залежить заявлена ціна, фактична ціна та знижка);

- operation (вміщає в собі інформацію стосовно виконуваної операції: продаж або купівля. Від цього буде залежати тип доставки та деякі інші частини програми. Детальний розгляд нижче);
- operation_type (вміщає в собі інформацію про тип виконуваної операції);
- operation_product (слугує проміжною ланкою між операцією та продуктом);
- delivery (вміщає в собі інформацію стосовно орієнтованої та фактичної дати доставки, замовлення, відправки а також місце відправлення та прибуття замовлення);
- delivery_type (вміщає в собі інформацію про тип доставки: купівля або продаж);
- status (вміщає в собі інформацію стосовно статусу операції: виконано, очікується тощо);
- history (вміщає в собі історію всіх транзакцій).

Для кращого розуміння системи хотілося б детальніше розглянути головні таблиці.

Таблиця 3.1 *warehouse* є ключовою, адже поєднує в собі інформацію про товари та ціноутворення. Це дає змогу краще розраховувати обсяги закупівель, приймати рішення щодо маркетингових акцій та прогнозувати прибутки.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		28

Таблиця 3.1.

Приклад розгорнутої таблиці warehouse

id	Назва товару	Тип	Підтип	Дата поставчання	Кількість, од.	Вартість			Знижка, %
						Фактична закупівельна, UAH	Заявлена, UAH	Фактична, UAH	
0	Xiaomi Redmi 4x	Електроніка	Моб. телефон	04.05.2019	17	3128	3565,92	3416,15	4,2
1	Xiaomi Redmi 5	Електроніка	Моб. телефон	12.12.2018	14	4220	4810,8	4483,67	6,8
2	Xiaomi Redmi 5 Plus	Електроніка	Моб. телефон	26.03.2019	8	5605	6389,7	6051,05	5,3

Спираючись на дану таблицю, аналітичний модуль може автоматично підрахувати що потенційний прибуток становить 169254,26 грн. Розрахунок буде відбуватися за формулою:

Кількість од. * Фактична ціна = Потенційний прибуток

Ще однією таблицею, що складає ядро системи є *person*. Адже дуже важливо знати свою аудиторію, її вподобання, тощо. Це дає змогу ефективніше продумувати способи реалізації товару, та побудови бренду компанії.

					КНТЕУ 121 063-06.МР				Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата					29

Таблиця 3.2.

Приклад розгорнутої таблиці person

id	Ім'я	Прізвище	Дата народження	Телефон	Електронна пошта	Адреса	Коментар
0	Андрій	Карпенко	04.11.2002	2938529643	andrey2k2@example.com	Львів	Телефони, смартфони
1	Марина	Жайворон ок	27.06.1999	7242916793	ptashka@example.com	Київ	Косметика
2	Дарина	Іваненко	16.12.1998	4958034891	divan@example.com	Черкаси	Автомобілі, спорт

Спираючись на інформацію в даній таблиці, ми можемо, наприклад, створити нагадування про надання знижки клієнту у його або її день народження. Таким чином ми отримуємо шанс стимулювати людину купити товар у нашому інтернет-магазині.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		30

Для повноцінної роботи аналітичного модуля необхідно зберігати інформацію про транзакції котрі вже відбулися (продано, доставлено тощо). Саме для цього і існує таблиця *history*.

Таблиця 3.3.

Приклад розгорнутої таблиці *history*

id	Тип операції	Статус	Дата	Назва товару	Кількість, од.	Вартість од., UAH	Продавець, id	Покупець, id
0	Закупка	Доставлено	09.07.2018	Xiaomi Redmi 4x	12	3128	218	0
1	Продаж	Продано	03.05.2019	Xiaomi Redmi 5 Plus	2	6051,05	0	32
2	Продаж	Відхилено	05.05.2019	Xiaomi Redmi 5	1	4483,67	0	152
3	Продаж	Продано	05.05.2019	Xiaomi Redmi 5	3	4483,67	0	74
4	Продаж	Продано	05.06.2019	Xiaomi Redmi 5 Plus	1	6051,05	0	128
5	Продаж	Продано	07.06.2019	Xiaomi Redmi 5	1	4483,67	0	92

Спираючись на інформацію в даній таблиці, можна помітити що попит на телефони в травні був більший, ніж у червні. Це може бути спричиненим виходом нових моделей на ринок. Отже можемо припустити що варто розглянути інші моделі товарів для закупки, щоб залишатися конкурентоспроможними.

					КНТЕУ 121 063-06.МР			Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата				31

3.3. Висновок до розділу 3

Отже, у даному розділі було вибрано та обгрунтовано вибір архітектурного паттерна для побудова інформаційної системи товарообгу. Спроектровано базу даних та наведено приклади схем основних модулів програми та зв'язків між ними.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		32

РОЗДІЛ 4.

ВИБІР ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ТА ОПИС ПРОЦЕСУ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

4.1. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Під час реалізації інформаційної системи товаробігу для електронної комерції було вирішено обрати об'єктно-орієнтовану мову програмування Java, фреймворк Spring Framework, веб-сервер Apache Tomcat, інструмент управління залежностями за зібранням проектів Apache Maven базу даних MySQL, мову розмітки гіпертекстових документів HTML, каскадні таблиці стилів CSS, браузерну мову програмування JavaScript, веб-фреймворк Bootstrap, Java-шаблон Thymeleaf, графічний редактор Adobe Photoshop та інтегроване середовище розробки програмного забезпечення IntelliJ IDEA.

Java - це об'єктно-орієнтована мова програмування загального призначення, що базується на класах і розроблена для того, щоб мати якомога менше залежностей реалізації. Вона призначена для того, щоб розробники програм "писали один раз, запускали де-небудь", а це означає, що скомпільований Java код може працювати на всіх платформах, які її підтримують без необхідності перекомпіляції. Програми Java зазвичай компілюються до "байт-коду", який може працювати на будь-якій віртуальній машині Java (JVM) незалежно від базової архітектури комп'ютера. Синтаксис

					<i>КНТЕУ 121 06з-06.МР</i>		
Зм.	Аркуш	№ докум.	Підпис	Дата			
Зав. каф.	Криворучко О.В.			27.10.20	Розробка інформаційної системи товаробігу для електронної комерції	Стадія	Аркуш
Керівник	Котенко Н.О.			27.10.20		Р4	33
Гарант	Криворучко О.В.			27.10.20		Факультет інформаційних технологій 2м курс, бз група	
Розробив	Мороз Б.В.			27.10.20			
					Вибір інструментальних засобів та опис процесу розробки програмного продукту		

Java схожий на C і C ++, але він має менше можливостей для низькорівневого програмування, ніж будь-який з них.

Станом на 2018 рік, Java була однією з найпопулярніших мов програмування, про що свідчить використання на GitHub[9].

За допомогою Java можна:

- розробляти програмне забезпечення на одній платформі і запускати його практично на будь-якій іншій платформі;
- створювати програми, що працюють у веб-браузері і мають доступ до веб-сервісів;
- розробляти додатки на стороні сервера для форумів в Інтернеті, магазинів, опитувань, обробки форм HTML і багато іншого;
- об'єднувати додатки або служби з використанням мови Java для створення високоспеціалізованих додатків або служб;
- створювати багатофункціональні та ефективні додатки для мобільних телефонів, віддалених процесорів, мікроконтролерів, бездротових модулів, датчиків, шлюзів, споживчих продуктів і практично будь-яких інших категорій електронних пристроїв[14].

Spring Framework - це програмний каркас (фреймворк) з відкритим кодом та контейнери з підтримкою інверсії управління для платформи Java. Основні особливості Spring Framework можуть бути використані будь-яким додатком Java, але є розширення для створення веб-додатків на платформі Java EE. Незважаючи на це, Spring Framework не нав'язує якоїсь конкретної моделі програмування, Spring Framework став популярним в спільноті Java як альтернатива, або навіть доповнення моделі Enterprise JavaBean (EJB)[8].

Spring Framework складається з кількох модулів, які надають широкий спектр послуг:

- контейнер інверсії управління: Конфігурація компонентів додатків і управління життєвим циклом об'єктів Java, здійснюється головним

					КНТЕУ 121 063-06.МР	Аркуш
						34
Зм.	Аркуш	№ докум	Підпис	Дата		

- чином через інверсію управління;
- аспектно-орієнтоване програмування: дозволяє реалізувати наскрізні процедури;
- доступ до даних: робота з реляційною системою управління базами даних на платформі Java з використанням JDBC і об'єктно-реляційні відображення та інструменти з NoSQL базами даних;
- управління транзакціями: об'єднує кілька API, керує транзакціями та координує операції для Java-об'єктів;
- Модель-Вигляд-Управління (Model-View-Controller): програмний каркас на основі HTTP сервлета, що забезпечує створення веб-додатків і веб-служб RESTful;
- аутентифікація і авторизація: налаштовувані процеси безпеки, які підтримують цілий ряд стандартів, протоколів, інструментів і практик за допомогою підпроєкту Spring Security;
- віддалене керування: конфігураційний вплив і управління Java-об'єктами для місцевої (локальної) або віддаленої конфігурації через JMX;
- тестування: підтримка класів для написання юніт-тестів та інтеграційних тестів.

Apache Tomcat – це контейнер сервлетів, розроблений Apache Software Foundation. Повністю написаний мовою програмування Java та реалізує специфікацію сервлетів і Java Server Pages від Sun Microsystems, що є стандартами для розробки веб-застосунків на Java [15].

Apache Maven - це фреймворк для автоматизації збирання проєктів на основі опису їх структури в файлах на мові POM (англ. Project Object Model), що є підмножиною XML [16].

Maven звертається до двох аспектів побудови програмного забезпечення: по-перше, він описує, як створюється програмне забезпечення,

					KHTEU 121 063-06.MP	Аркуш
						35
Зм.	Аркуш	№ докум	Підпис	Дата		

і, по-друге, описує його залежності. На відміну від попередніх інструментів, таких як Apache Ant, він використовує умови для процедури збирання, і тільки винятки повинні бути записані. XML-файл описує проект проекту, який будується, його залежність від інших зовнішніх модулів і компонентів, порядок складання, каталоги і необхідні плагіни. Він поставляється з заздалегідь визначеними цілями для виконання певних чітко визначених завдань, таких як складання коду та його упаковка.

Maven динамічно завантажує бібліотеки Java і власні модулі з одного або декількох сховищ, таких як центральне сховище Maven 2, і зберігає їх у локальному кеші. Цей локальний кеш завантажених артефактів також може бути оновлений артефактами, створеними локальними проектами. Також можна оновити загальнодоступні сховища.

MySQL - це одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

HTML - це стандартна мова розмітки для створення веб-сторінок і веб-додатків [11].

Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки.

CSS - це спеціальна мова, що використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних[10].

Каскадні таблиці стилів використовувати створені веб-сторінки для задання кольорів, шрифтів, розкладок окремих блоків і інших аспектів. Основна ціль розробки CSS полягала в розділенні опису логічної структури веб-сторінок, що виражається з використанням HTML або інших форматів даних з опису зовнішнього вигляду цієї веб-сторінки (що тепер є результатом використання формальної мови CSS). Таке розділення може збільшити

					<i>КНТЕУ 121 063-06.МР</i>	Аркуш
						36
Зм.	Аркуш	№ докум	Підпис	Дата		

доступність документа, надаючи велику гнучкість і можливість керування його представленням, а також зменшити складність і повторення в структурному вмісті. Крім того, CSS дозволяє представити один і той же документ у різних стилях або методах вибору, таких як екранне представлення, печатне уявлення, читання голосу (спеціальні голосові браузері або програми з екранами), або при виборі пристроїв, що використовують шрифт Брайля.

JavaScript – об'єктно-орієнтована, скриптова мова програмування, розроблена компанією Netscape. JavaScript, зазвичай використовується, як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке вживання знаходить в браузерах, як мова сценаріїв для додання інтерактивності веб-сторінкам.

Thymeleaf - це сучасний серверний Java-шаблон для обох веб- і автономних середовищ. Він дає змогу розширити функції стандартного HTML коду, і зробити розробку зручнішою та швидшою[18].

Bootstrap - це безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і веб-додатків.

Adobe Photoshop - це багатофункціональний графічний редактор, розроблений і поширюваний фірмою Adobe Systems. В основному працює з растровими зображеннями, проте має деякі векторні інструменти. Продукт є лідером ринку в області комерційних засобів редагування растрових зображень і найбільш відомим продуктом компанії Adobe[17].

IntelliJ IDEA - комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP та ін.) від компанії JetBrains. Система поставляється у вигляді урізаної по функціональності безкоштовної

					<i>KHTEU 121 063-06.MP</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		37

версії «Community Edition» і повнофункціональної комерційної версії «Ultimate Edition»[20].

Порівняльний аналіз середовищ розробки програмного забезпечення дозволяє раціонально вибрати найбільш доцільну систему для створення додатку, що буде відповідати вищевказаним вимогам та стеку технологій. У якості альтернатив будуть розглянуті два варіанти середовищ розробки програмного забезпечення від компанії JetBrains: Community та Ultimate. Інші середовища не бралися до уваги через їхню застарілість, складність в освоєнні і малу поширеність в порівнянні з продуктом компанії JetBrains.

У ході огляду можливостей обох систем було виявлено що Community версія середовища IntelliJ IDEA підтримує інструменти (у вигляді плагінів) для проведення тестування TestNG і JUnit, системи контролю версій CVS, Subversion, Mercurial і Git, засоби складання Maven, Ant, Gradle, мови програмування Java, Scala, Clojure, Groovy і Dart.

Підтримується розробка застосунків для мобільної платформи Android. До складу входить модуль візуального проектування GUI-інтерфейсу Swing UI Designer, XML-редактор, редактор регулярних виразів, система перевірки коректності коду, система контролю за виконанням завдань і доповнення для імпорту та експорту проектів з Eclipse. Доступні засоби інтеграції з системами відстеження помилок JIRA, Trac, Redmine, Pivotal Tracker, GitHub, YouTrack, Lighthouse.

Комерційна версія «Ultimate Edition» відрізняється наявністю підтримки додаткових мов програмування (наприклад, PHP, Ruby, Python, JavaScript, CoffeeScript, HTML, CSS, SQL), підтримкою технологій Java EE, UML-діаграм, підрахунок покриття коду, можливістю роботи з фреймворками (Rails, Grails, Google Web Toolkit, Spring, Play Framework і Hibernate), засобами інтеграції з Perforce, Microsoft Team Foundation Server і Rational ClearCase.

					<i>КНТЕУ 121 063-06.МР</i>	Аркуш
						38
Зм.	Аркуш	№ докум	Підпис	Дата		

Таблиця 4.1.

Порівняння версій IntelliJ IDEA

Мова програмування	IntelliJ IDEA Community Edition	IntelliJ IDEA Ultimate Edition
Java	Так	Так
Clojure (через окремий плагін)	Так	Так
Dart (через окремий плагін)	Так	Так
Erlang (через окремий плагін)	Так	Так
Go (через окремий плагін)	Так	Так
Groovy	Так	Так
Haxe (через окремий плагін)	Так	Так
Perl (via separate plugin)	Так	Так
Scala (через окремий плагін)	Так	Так
XML/XSL	Так	Так
Kotlin	Так	Так
ActionScript/MXML	Ні	Так
CoffeeScript	Ні	Так
Haskell (через окремий плагін)	Так	Так
HTML/XHTML/CSS	Ні	Так
JavaScript	Ні	Так
Lua (через окремий плагін)	Так	Так
PHP (через окремий плагін)	Ні	Так
Python (через окремий плагін)	Так	Так
Ruby/JRuby	Ні	Так
SQL	Ні	Так
TypeScript (через окремий плагін)	Ні	Так

Проведений аналіз таких інтегрованих середовищ розробки програмного забезпечення, як IntelliJ IDEA Community Edition та IntelliJ IDEA Ultimate Edition показав, що для виконання поставлених задач найбільш підходящими є IntelliJ IDEA Ultimate Edition. Комерційна версія має ряд суттєвих переваг над безкоштовною, зокрема кількість технологій котрі підтримує комерційна версія набагато перевищує безкоштовну версію. Крім

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		39

того, в Ultimate Edition реалізована підтримка мов HTML та CSS, що дає змогу створювати повноцінні клієнт-серверні додатки. До того ж всі студенти мають змогу отримати безкоштовно ліцензію на використання IntelliJ IDEA Ultimate Edition.

Проведений аналіз предметної області дозволив визначити основні напрямки автоматизації обліку товарів для електронної комерції. Необхідно реалізувати збереження даних про товари, надходження, продажі, а також про клієнтів підприємства та їх інтереси за категоріями.

В роботі обґрунтовано вибрати наступні засоби розробки:

- мову програмування Java 8;
- Java-шаблон Thymeleaf 3.0.11;
- фреймворк Spring Boot 2.1.5;
- фреймворк Bootstrap 4.3.1;
- мову розмітки гіпертекстових документів HTML 5;
- каскадні таблиці стилів CSS 3;
- мову програмування JavaScript 1.5;
- середовище керування базами даних MySQL 8.0;
- веб-сервер Apache Tomcat 8.5.42;
- менеджер проектів Apache Maven 3.6.1;

Буде використано такі програмні продукти та інтегровані середовища розробки: графічний редактор Adobe Photoshop CS6; середовище розробки IntelliJ IDEA 2019.1.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		40

4.2. Створення структури проекту

При створенні проекту ми використали менеджер керування залежностями Apache Maven. З його допомогою можна автоматично створити початкову структуру веб проекту. Для цього потрібно, при створенні нового проекту, на лівій панелі вибрати пункт Maven, потім помітити птичкою «create from archetype» після чого ми отримаємо змогу вибрати архетип (Рис. 4.1).

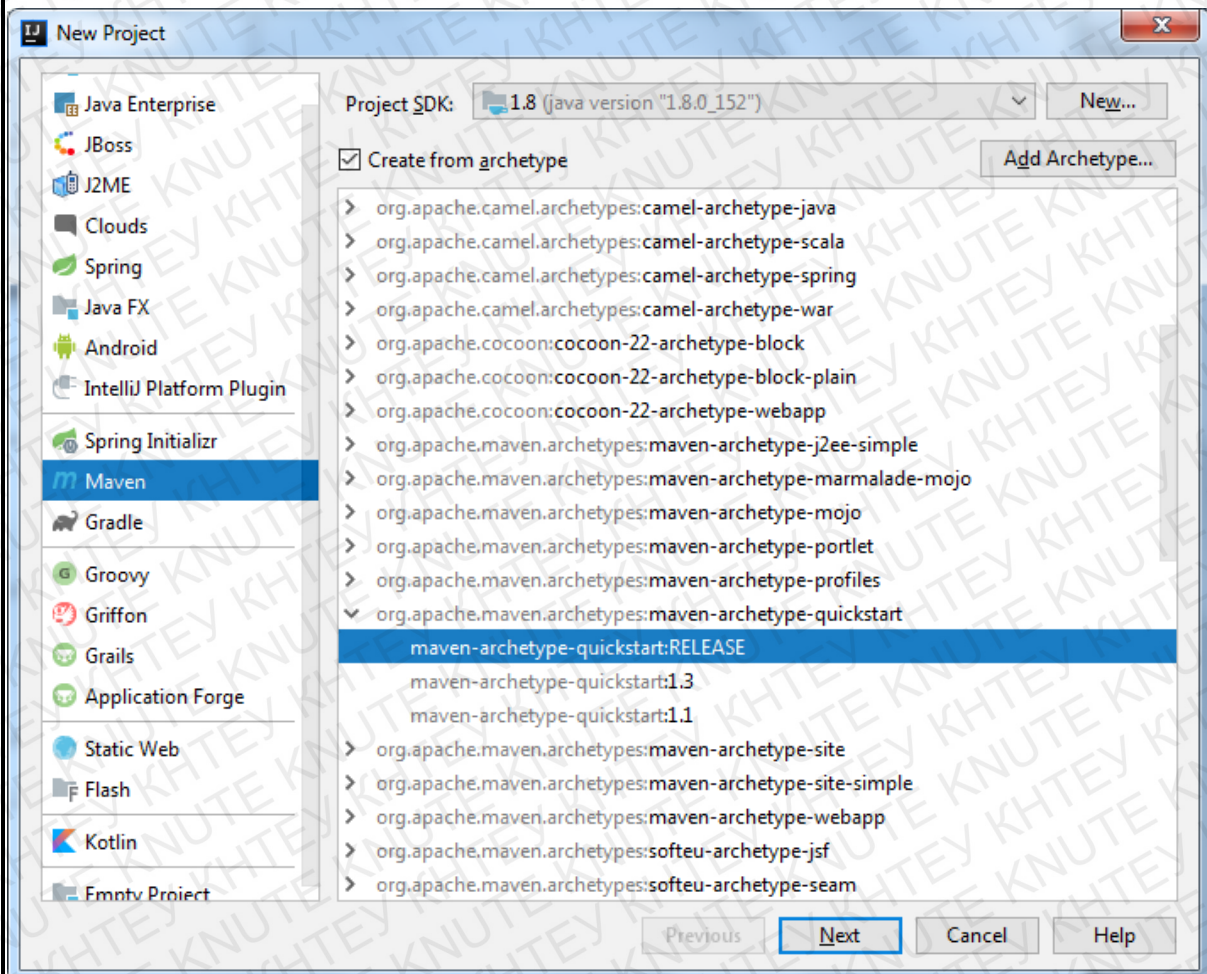


Рис. 4.1. Знімок віна вибору архетипу Maven

Архетипом називається шаблон нового проекту, котрий має певну структуру з заготовками вихідних та конфігураційних файлів.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		41

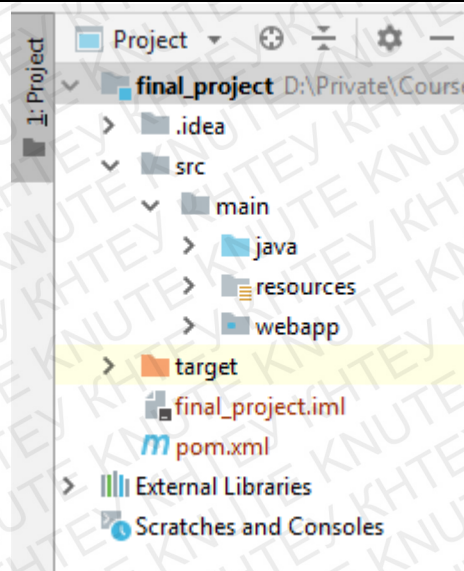


Рис. 4.2. Приклад початкової структури веб-додатку, побудованої за допомогою архетипу

Ключовими елементами будь-якого проекту є:

- Папка *.idea* (містить в собі налаштування конкретного середовища програмного забезпечення, компілятора, системи керування версіями та ін.);
- папка *src* (містить в собі весь код програми, ресурси та налаштування);
- папка *java* (зберігає в собі класи інтерфейси та пакети програми написані на мові Java, такі як: *annotation*, *controller*, *command*, *filter*, *listener*, *exception*, *model*, *DAO*, *entity*, *enumeration*, *service*, *util* та інше);
- папка *resources* (містить в собі налаштування підключення до бази даних, локалізації, а також загальні часто використовувані значення для кращої читабельності коду та уникнення помилок людського характеру);
- папка *webapp* (містить в собі папку *WEB-INF* для збереження веб-сторінок написаних на HTML, каскадних таблиць стилів CSS та налаштувань відображення сторінок; Головну сторінку *index.html*, з котрої починається робота з системою);

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		42

- папка target (містить в собі згенеровані ресурси, і зібраний проект, готовий для запуску);
- файл з розширенням *.iml (зберігає інформацію про модуль розробки Java або Maven)
- pom.xml (зберігає інформацію про проект та його конфігурації, таким чином, щоб Maven міг з ними взаємодіяти. До конфігурацій відносяться: залежності проекту, завдання, версії проекту, плагіни та інше);
- External Libraries (містить у собі всі завантажені бібліотеки, котрі використовуються в проекті).

4.3. Налаштування проекту

Для побудови інформаційної системи товарообігу по шаблону MVC, нам потрібно підключити певні бібліотеки:

- spring-boot-starter-parent (базовий набір інструментів для побудови програми з використанням технології Spring Boot);
- spring-boot-starter-web (набір інструментів для побудови програми з використанням технології Spring Boot MVC);
- spring-boot-starter-thymeleaf (набір інструментів для побудови програми з використанням технології Thymeleaf);
- spring-boot-devtools (набір інструментів для полегшення розробки програми з використанням технології Spring Boot);
- spring-boot-starter-data-jpa (набір інструментів для побудови програми з використанням технології JPA);
- mysql-connector-java (драйвер та набір інструментів для підключення бази даних).

Повний перелік залежностей наведено у додатку.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		43

4.4. Створення програми

При проектуванні системи було прийнято рішення використати шаблон проектування MVC. Отже ключовими елементами програми буде модель, представлення та контролер.

Процес роботи з системою відбувається через взаємодію користувача з представленням (View), яке насправді є звичайною HTML сторінкою. Для оптимізації сайту, відповідності його сучасним вимогам і пришвидшення створення системи ми використали шаблони Bootstrap, які є у вільному доступі на офіційному сайті фреймворка.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		44

Отже давайте розглянемо основні елементи інтерфейсу, які буде бачити користувач, а насамперед сторінку реєстрації (рис. 4.3)

Рис. 4.3. Сторінка реєстрації нового користувача

Після того як користувач введе всю інформацію, вона відправляється на сервер і зберігається в базі даних для подальшого використання під час роботи.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		45

Якщо нового користувача створено успішно, можна увійти в систему
(Рис. 4.4)

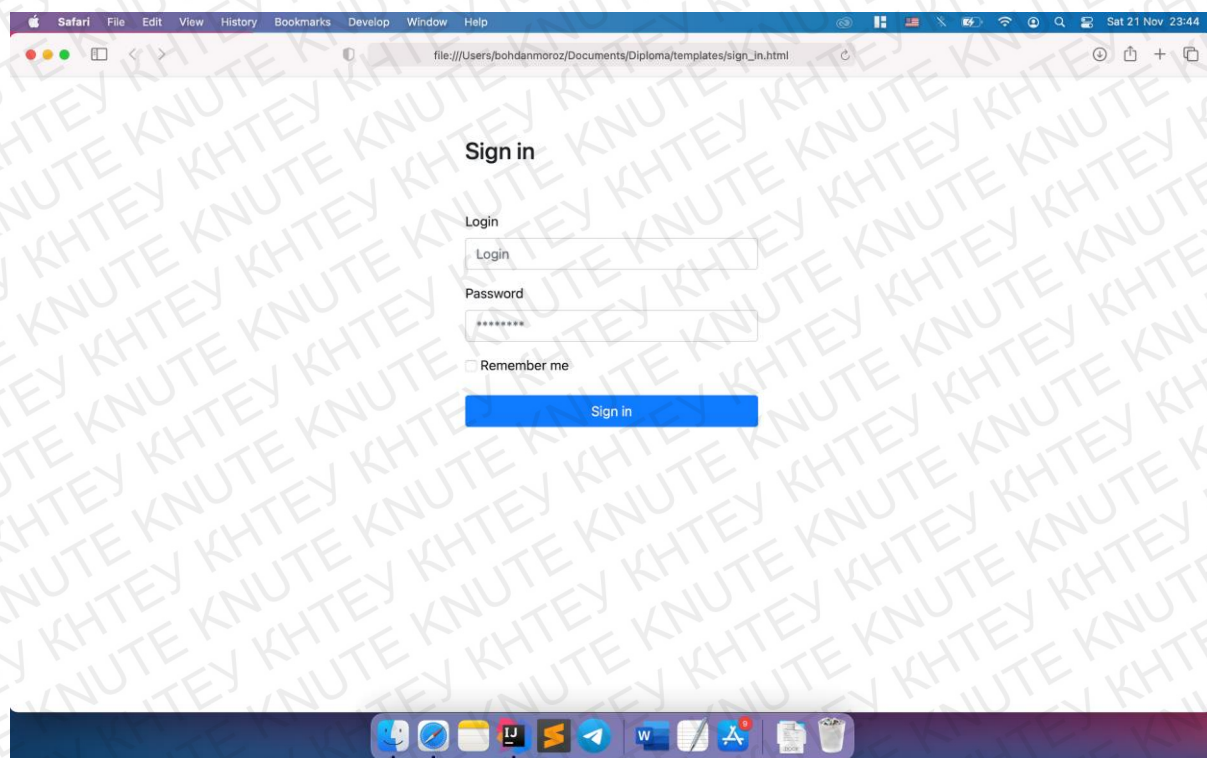


Рис. 4.4. Сторінка авторизації користувача в системі

Даний етап дуже важливий, адже за допомогою авторизації користувач може бути впевнений, що тільки він має доступ до інформації стосовно власного підприємства.

Після введення всіх даних і натиснення кнопки Sign in (укр. - вхід), введена інформація відправляється на сервер і попадає в контролер (Рисунок 4.5).

Для реалізації контролера з використанням технології Spring потрібно перед оголошенням класу поставити анотацію `@Controller`, таким чином ми даємо компілятору знати що цей клас буде виконувати функції контролера. Ще однією важливою анотацією перед оголошенням класу є `@RequestMapping`, вона вказує початок URL для звернення до контролера.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		46

```

@Controller
@RequestMapping(path="/diploma")
public class GreetingController {

    @GetMapping(path="/validate_user")
    public @ResponseBody String validateUser (@RequestParam String login, @RequestParam String password) {
        if (isValidUser(login, password)) {
            return "MainPage";
        }
        return "Error";
    }
}

```

Рис. 4.5. Приклад оголошення контролера. Модуль авторизації користувача

Анотація `@GetMapping` означає що метод контролера обробляє тільки Get запити котрі починаються з `/diploma/validate_user/`. Також можна налаштувати щоб метод обробляв Post запити, за допомогою анотації `@PostMapping`. Якщо дані введено вірно то користувач побачить головну сторінку.

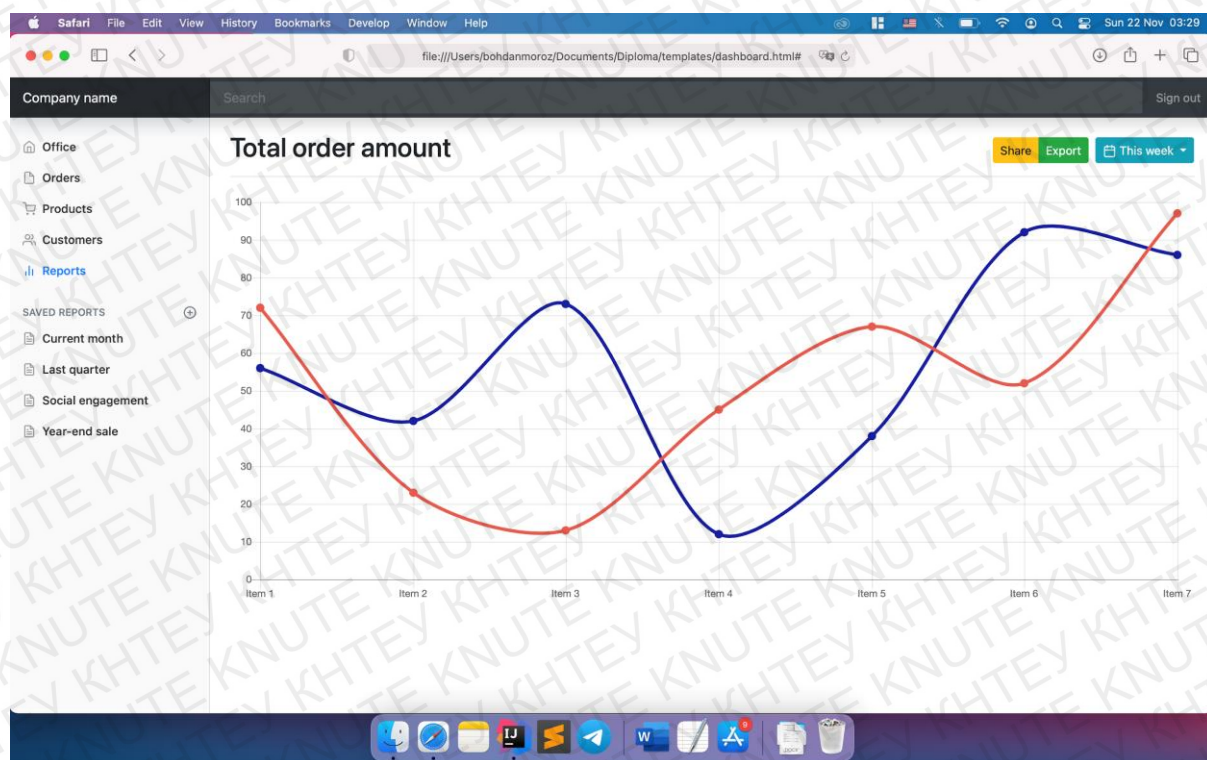


Рис. 4.6. Головна сторінка

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		47

На головній сторінці наведено приклад графіку, котрий відображає прибуток за поточний тиждень.

Також користувачу доступний пошук, перехід на інші внутрішні сторінки системи та переглядати інформацію отриману з бази даних (Рисунок 4.7).

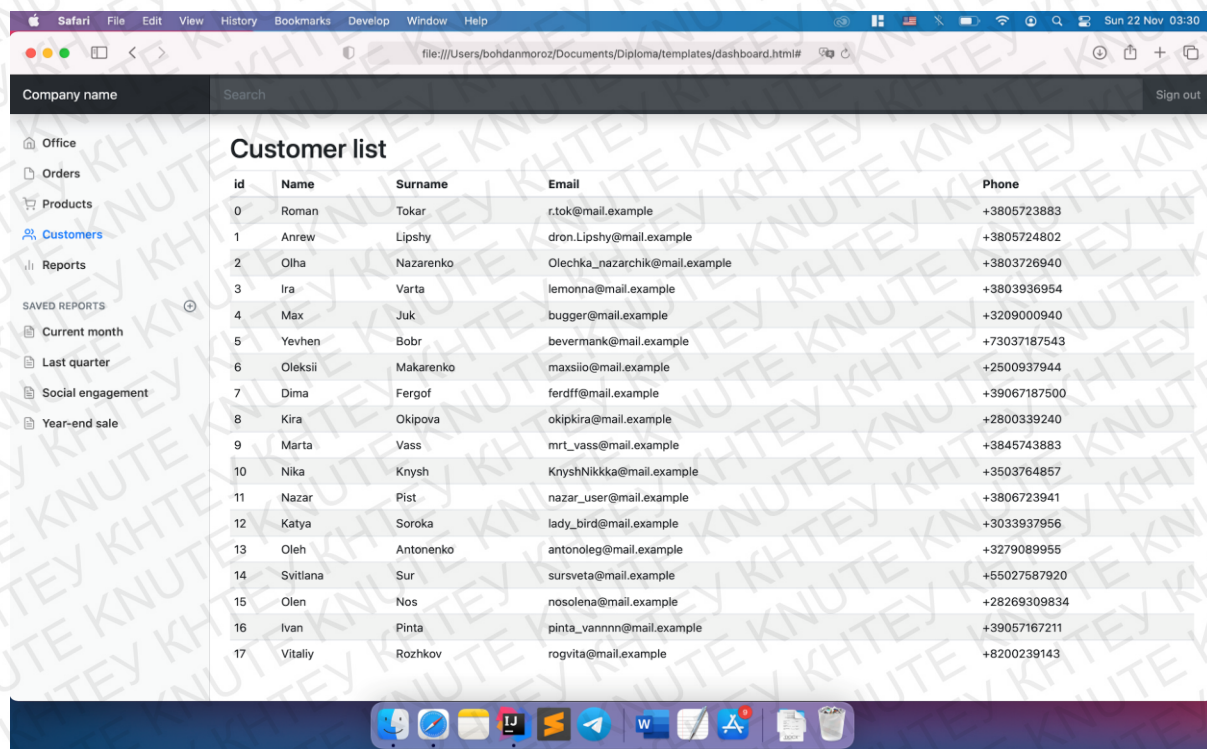


Рис. 4.7. Перегляд інформації про клієнтів внесених в базу інформаційної системи

Для отримання інформації з БД нам потрібно перш за все налаштувати підключення: ввести адресу бази даних, логін та пароль адміністратора (Рис. 4.8), і Spring Framework автоматично зробить все інше.



Рис. 4.8. Налаштування підключення до бази даних

Після підключення потрібно створити модель, у даному випадку сутність, котра буде зберігати в собі тимчасову інформацію (Рис. 4.9).

```

7
8  @Entity
9  public class User {
10     @Id
11     @GeneratedValue(strategy=GenerationType.AUTO)
12     private Integer id;
13
14     private String login;
15
16     private String password;
17
18     public Integer getId() { return id; }
19
20     public void setId(Integer id) { this.id = id; }
21
22     public String getLogin() { return login; }
23
24     public void setLogin(String login) { this.login = login; }
25
26     public String getPassword() { return password; }
27
28     public void setPassword(String password) { this.password = password; }
29
30     @Override
31     public String toString() {
32         return "User{" +
33             "id=" + id +
34             ", login='" + login + '\'' +
35             ", password='" + password + '\'' +
36             '}';
37     }
38 }

```

Рис. 4.9. Сутність користувача

Анотація @Entity позначає даний клас як сутність, котра зберігає в собі інформацію про користувача і дає змогу її читати та змінювати. Іншими словами – це об’єктне представлення запису з таблиці реляційної бази даних. Для того щоб інформація з бази даних попала у сутність потрібно створити репозиторій, котрий буде розширювати інтерфейс CrudRepository. З його допомогою ми зможемо виконувати CRUD операції з базою даних, а саме: створювати нові записи, та читати, редагувати і видаляти існуючі.

					КНТЕУ 121 063-06.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		49

```

public interface UserRepository extends CrudRepository<User, Integer> {
    @Override
    Iterable<User> findAll();

    @Override
    Optional<User> findById(Integer integer);

    @Override
    <S extends User> Iterable<S> saveAll(Iterable<S> iterable);
}

```

Рис. 4.10. Створення репозиторія для сутності User

На даному рисунку наведено приклад наслідування методів findAll (знайти всю інформацію), findById (знайти інформацію про користувача з конкретним ідентифікатором), SaveAll (зберегти всю введену інформацію), оголошених в інтерфейсі CrudRepository<User, Integer>.

Таким чином реалізовані й інші компоненти взаємодії користувача з системою.

У даному розділі було проведено аналіз потрібних інструментів для створення інформаційної системи товарообігу для електронної комерції, та обгрунтовано їх вибір. Наведено процес розробки програмного забезпечення, лістинг основних модулів та приклади із кінцевої реалізації.

4.5. Висновки до розділу 4

У даному розділі було проведено аналіз потрібних інструментів для створення інформаційної системи товарообігу для електронної комерції, та обгрунтовано їх вибір. Наведено процес розробки програмного забезпечення, лістинг основних модулів та приклади із кінцевої реалізації.

					КНТЕУ 121 063-06.МР	Аркуш
						50
Зм.	Аркуш	№ докум	Підпис	Дата		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

З проведених досліджень можна зробити наступні висновки:

1. У ході виконання даної випускної кваліфікаційної роботи було розроблено інформаційну систему товарообігу для електронної комерції. Результати розробки наведені в додатку А.

2. Під час реалізації інформаційної системи товарообігу були закріплені навички здобуті протягом навчання на 2-х курсах у Київському національному торговельно-економічному університеті, а саме: побудова алгоритмів та блок-схем виконання програмного забезпечення, проектування архітектури, навички аналізу даних, програмування на мові Java та JavaScript, написання сайтів на мові HTML з використанням CSS, читання документації на англійській мові.

3. Було реалізовано:

- авторизацію користувача в систему;
- створення контактів;
- додавання товарів на склад;
- відслідковування відправлень та доставки;
- створювати нагадування для певних подій;
- аналіз діяльності та отримання порад щодо покращення.

5. У майбутньому дана система може бути розширеною, і використовуватися на підприємствах що займаються електронною комерцією. Можуть бути додані модулі автоматичного створення накладних, чеків та гарантій, функції генерації та автоматичної відправки електронних листів клієнтам та партнерам.

					<i>КНТЕУ 121 06з-06.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			30.10.20	Розробка інформаційної системи товарообігу для електронної комерції	Стадія	Аркуш	Аркушів
Керівник	Котенко Н.О.			30.10.20		ВП	51	53
Гарант	Криворучко О.В.			30.10.20		Факультет інформаційних технологій 2м курс, бз група		
Розробив	Мороз Б.В.			30.10.20				
					<i>Висновки та пропозиції</i>			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Joshua Bloch. Effective Java. – [3 видання]. - Addison-Wesley Professional, 2018. – 412 с.
2. Robert C. Martin. lean Code: A Handbook of Agile Software Craftsmanship. – [1 видання]. - Prentice Hall, 2008. – 464 с.
3. Prentice Hall. Code Complete: A Practical Handbook of Software Construction. - [2 видання]. - Microsoft Press, 2004. – 960 с.
4. Bruce Eckel. Thinking in Java. - [4 видання]. - Prentice Hall, 2006. – 1150 с.
5. Craig Walls. Spring in Action. - [5 видання]. - Manning Publications, 2018. – 520 с.
6. Craig Walls. Spring Boot in Action. - [1 видання]. - Manning Publications, 2016. – 264 с.
7. Синицын С. Верификация программного обеспечения. / Синицын С. В., Налютин Н. Ю.— М.: БИНОМ, 2008. — 368 с.
8. Офіційна документація Spring Framework [Електронний ресурс] – Режим доступу до ресурсу: <https://spring.io/>
9. Офіційна документація Java SE 8 [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.oracle.com/javase/8/docs/api/>
10. Довідниково-інформаційний портал htmlbook [Електронний ресурс] – Режим доступу до ресурсу: <http://htmlbook.ru/>
11. Довідниково-інформаційний портал w3schools [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/>

					<i>КНТЕУ 121 06з-06.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			24.01.20	Розробка інформаційної системи товарообігу для електронної комерції	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Котенко Н.О.			24.01.20		<i>СВД</i>	52	53
Гарант	Криворучко О.В.			24.01.20		Факультет інформаційних технологій 2м курс, 6з група		
Розробив	Мороз Б.В.			24.01.20				
					<i>Список використаних джерел</i>			

12. Довідниково-інформаційний портал o7planning [Електронний ресурс] –
Режим доступу до ресурсу: <https://o7planning.org/>
13. Інформаційний бізнес портал livebusiness [Електронний ресурс] –
Режим доступу до ресурсу: <https://www.livebusiness.com.ua/tools/crm/>
14. Офіційний портал мови програмування Java [Електронний ресурс] –
Режим доступу до ресурсу: <https://www.java.com/>
15. Офіційний портал веб-сервера Apache Tomcat [Електронний ресурс] –
Режим доступу до ресурсу: <http://tomcat.apache.org/>
16. Офіційний портал веб-сервера Apache Maven [Електронний ресурс] –
Режим доступу до ресурсу: <https://maven.apache.org/>
17. Офіційний портал програми Adobe Photoshop [Електронний ресурс] –
Режим доступу до ресурсу: <https://www.adobe.com/ua/products/photoshop.html>
18. Офіційний портал технології Thymeleaf [Електронний ресурс] – Режим
доступу до ресурсу: <https://www.thymeleaf.org/>
19. Офіційний портал технології Bootstrap [Електронний ресурс] – Режим
доступу до ресурсу: <https://getbootstrap.com/>
20. Офіційний портал інтегрованого середовища розробки програмного
забезпечення IntelliJ IDEA [Електронний ресурс] – Режим доступу до
ресурсу: <https://www.jetbrains.com/idea/>
21. Довідниково-інформаційний портал baeldung [Електронний ресурс] –
Режим доступу до ресурсу: <https://www.baeldung.com>

ДОДАТОК А

Клас Application.java

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Application {
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

    public enum CommandEnum {
        LOGIN {
            {
                this.command = new LoginCommand();
            }
        },
        LOGOUT {
            {
                this.command = new LogoutCommand();
            }
        };
        ActionCommand command;
        public ActionCommand getCurrentCommand() {
            return command;
        }
    }
}
```

Клас LoginCommand.java

```
import javax.servlet.http.HttpServletRequest;

public class LoginCommand implements ActionCommand {
    private static final String PARAM_NAME_LOGIN = "Login";
    private static final String PARAM_NAME_PASSWORD = "Password";

    @Override
    public String execute(HttpServletRequest request) {
        String page = null;

        String login = request.getParameter(PARAM_NAME_LOGIN);
        String password = request.getParameter(PARAM_NAME_PASSWORD);

        if(LoginLogic.checkLogin(login, password)) {
            request.setAttribute("user", login);

            page = ConfigurationManager.getProperty("path.page.main");
        } else {
            request.setAttribute("errorLoginPassMessage",
                ConfigurationManager.getProperty("message.loginerror"));
            page = ConfigurationManager.getProperty("path.page.login");
        }

        return page;
    }
}
```

Клас ActionFactory.java

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ActionFactory {
    public ActionCommand defineCommand(HttpServletRequest request) {
        ActionCommand current = new EmptyCommand();

        String action = request.getParameter("command");
        if (action == null || action.isEmpty()) {
            return current;
        }

        try {
            CommandEnum currentEnum =
                CommandEnum.valueOf(action.toUpperCase());
            current = currentEnum.getCurrentCommand();
        } catch (IllegalArgumentException e) {
            request.setAttribute("wrongAction", action
                + MessageManager.getProperty("message.wrongaction"));
        }

        return current;
    }
}
```

Клас LogoutCommand.java

```
import javax.servlet.http.HttpServletRequest;

public class LogoutCommand implements ActionCommand {

    @Override
    public String execute(HttpServletRequest request) {
        String page = ConfigurationManager.getProperty("path.page.index");

        request.getSession().invalidate();
        return page;
    }
}
```

Клас User.java

```
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class User {
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Integer id;

    private String login;

    private String password;

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getLogin() {
        return login;
    }

    public void setLogin(String login) {
        this.login = login;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    @Override
    public String toString() {
        return "User{" +
            "id=" + id +
            ", login='" + login + '\'' +
            ", password='" + password + '\'' +
            '}';
    }
}
```

Интерфейс ActionCommand.java

```
import javax.servlet.http.HttpServletRequest;  
  
public interface ActionCommand {  
    public String execute(HttpServletRequest request);  
}
```

Інтерфейс UserRepository.java

```
import org.springframework.data.repository.CrudRepository;
import java.util.Optional;

public interface UserRepository extends CrudRepository<User, Integer> {
    @Override
    Iterable<User> findAll();

    @Override
    Optional<User> findById(Integer integer);

    @Override
    <S extends User> Iterable<S> saveAll(Iterable<S> iterable);

    @Override
    <S extends User> S save(S s);

    @Override
    boolean existsById(Integer integer);

    @Override
    Iterable<User> findAllById(Iterable<Integer> iterable);

    @Override
    long count();

    @Override
    void deleteById(Integer integer);

    @Override
    void delete(User user);

    @Override
    void deleteAll(Iterable<? extends User> iterable);

    @Override
    void deleteAll();
}
```

Клас PersonController.java

```
import net.proselyte.springmvc.model.Person;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class PersonController {
    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String index() {
        return "/index";
    }
    @RequestMapping(value = "person", method = RequestMethod.GET)
    public ModelAndView developer() {
        return new ModelAndView("person", "command", new Person());
    }

    @RequestMapping(value = "/addPerson", method = RequestMethod.POST)
    public String addStudent(@ModelAttribute("mvc-dispatcher") Person person,
                             ModelMap model) {
        model.addAttribute("id", person.getId());
        model.addAttribute("name", person.getName());
        model.addAttribute("adress", person.getAdress());
        model.addAttribute("phone", person.getPhone());
        model.addAttribute("mail", person.getMail());
        model.addAttribute("comment", person.getComment());

        return "result";
    }
}
```

Клас ConnectionPool.java

```

import javax.sql.DataSource;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.ResourceBundle;

public class ConnectionPool {

    private static BasicDataSource ds = new BasicDataSource();
    private static ResourceBundle resource =
ResourceBundle.getBundle("database.database");
    private static String URL = resource.getString("db.url");
    private static String USERNAME = resource.getString("db.username");
    private static String PASSWORD = resource.getString("db.password");
    private static int MIN_IDLE =
Integer.valueOf(resource.getString("db.minIdle"));
    private static int MAX_IDLE =
Integer.valueOf(resource.getString("db.maxIdle"));
    private static int MAX_OPEN_PREPARED_STATEMENTS =
Integer.valueOf(resource.getString("db.maxOpenPreparedStatements"
));
    private static String DRIVER_CLASS_NAME =
resource.getString("db.driver");

    static {
        ds.setUrl(URL);
        ds.setUsername(USERNAME);
        ds.setPassword(PASSWORD);
        ds.setMinIdle(MIN_IDLE);
        ds.setMaxIdle(MAX_IDLE);
        ds.setMaxOpenPreparedStatements(MAX_OPEN_PREPARED_STATEMENTS);
        ds.setDriverClassName(DRIVER_CLASS_NAME);
    }

    public static Connection getConnection() throws SQLException {
        return ds.getConnection();
    }

    private ConnectionPool() {}

    private static volatile DataSource dataSource;
    public static Connection getDataSource() throws SQLException{

        if (dataSource == null){
            synchronized (ConnectionPool.class) {
                if (dataSource == null) {
                    dataSource = ds;
                }
            }
        }
        return dataSource.getConnection();
    }
}

```

Налаштування `messages.properties`

`message.loginerror` = Incorrect login or password

`message.nullpage` = Page not found. Business logic error

`message.wrongaction` = : command not found or wrong!

ДОДАТОК І

Налаштування db.properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/diploma?useUnicode=true&useJDBCCompliantTimezoneShift=\  
true&useLegacyDatetimeCode=false&serverTimezone=UTC  
spring.datasource.username=root  
spring.datasource.password=root
```

Конфігурація pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>final_project</groupId>
  <artifactId>diploma</artifactId>
  <version>1.0-SNAPSHOT</version>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.4.RELEASE</version>
  </parent>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <optional>true</optional>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
    </dependency>
  </dependencies>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>
```

Сторінка error.html

```
<%@ page isErrorPage="true" contentType="text/html; charset=UTF-8"
language="java" %>
<html>
<head>
  <title>Error 404</title>
</head>
<body>
  404 - page not found
</body>
</html>
```