

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Програмно-алгоритмічний комплекс для обліку товарів і товарних операцій перукарського салону

Виконав: студент
спеціальності

IV курсу, групи СНЗс-42
122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис)

Павелець А.Р.

(прізвище та ініціали)

Керівник

(підпис)

Матійчук Л.П.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Жаровський Р.О.

(прізвище та ініціали)

Тернопіль
2022

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра комп'ютерних наук

(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.

(підпис)

(прізвище та ініціали)

«__» _____ 2022 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр

(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки

(шифр і назва спеціальності)

Студенту Павелець Алісі Романівні

(прізвище, ім'я, по батькові)

1. Тема роботи Програмно-алгоритмічний комплекс для обліку товарів і
товарних операцій перукарського салону

Керівник роботи Матійчук Любомир Павлович, к.е.н., доцент кафедри КН

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «23» березня 2022 року № 4/7-172.

2. Термін подання студентом завершеної роботи 13 червня 2022р.

3. Вихідні дані до роботи Наукові публікації, електронні ресурси, підручники, посібники з
тематики дослідження

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. Розділ 1. Аналіз предметної області та проектування. 1.1 Коротка характеристика
об'єкту опис предметної. 1.2 Огляд і аналіз існуючих аналогів специфікація вимог до
системи. 1.3 Розробка архітектури програмної системи. 1.4 Висновок до першого розділу.

Розділ 2. Програмна реалізація, тестування та дослідна експлуатація. 2.1. Програмна
реалізація проекту. 2.2 Програмна реалізація бази даних. 2.3 Тестування. 2.4 Інструкція
користувача. 2.5 Висновок до другого розділу. 3. Безпека життєдіяльності, основи хорони
праці. Висновки. Перелік джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Тема. 2. Актуальність роботи. 3. Організаційна структура салону. 4. Порівняння аналогів.

5. Порівняльні характеристики програмних продуктів. 6. Діаграма варіантів використання.

7. Трирівнева архітектура. 8. UML-діаграма класів. 9. Діаграма активності продажу товарів.

10. Тестування системи. 11.-13. Інструкція користувача. 14. Висновки. 15. Дякую за увагу.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Гурик О.Я. кандидат технічних наук, доцент кафедри МТ		

7. Дата видачі завдання 24 січня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	24.01.2022	<i>Виконано</i>
2.	Підбір джерел про програмно-алгоритмічний комплекс для обліку товарів і товарних операцій перукарського салону	04.01.2022-30.01.2022	<i>Виконано</i>
3.	Переклад та опрацювання джерел про програмно-алгоритмічний комплекс	31.01.2022-06.02.2022	<i>Виконано</i>
4.	Виконання дослідження щодо програмно-алгоритмічний комплексу для обліку товарів і товарних операцій перукарського салону	07.02.2022-13.02.2022	<i>Виконано</i>
5.	Оформлення розділу «Аналіз предметної області та проектування»	14.02.2022-06.03.2022	<i>Виконано</i>
6.	Оформлення розділу «Програмна реалізація, тестування та дослідна експлуатація»	07.03.2022-03.04.2022	<i>Виконано</i>
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	04.04.2022-17.04.2022	<i>Виконано</i>
8.	Виконання завдання до підрозділу «Основи хорони праці»	18.04.2022-01.05.2022	<i>Виконано</i>
9.	Оформлення кваліфікаційної роботи	02.05.2022-15.05.2022	<i>Виконано</i>
10.	Нормоконтроль	16.05.2022-22.05.2022	<i>Виконано</i>
11.	Перевірка на плагіат	01.06.2022	<i>Виконано</i>
12.	Попередній захист кваліфікаційної роботи	07.06.2022	<i>Виконано</i>
13.	Захист кваліфікаційної роботи	13.06.2022	

Студент

_____ (підпис)

Павелець А. Р.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Матійчук Л.П.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Програмно-алгоритмічний комплекс для обліку товарів і товарних операцій перукарського салону // Кваліфікаційна робота освітнього рівня «Бакалавр» // Павелець Аліса Романівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНзс-42 // Тернопіль, 2022 // С.55 , рис. – 33, табл. – 11, бібліогр. – 20.

Ключові слова: проектування та розробка програмного забезпечення, проведення системного тестування, технологія JSP, бази даних.

Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка програмної системи для обліку товарів і товарних операцій перукарського салону.

У першому розділі проведено детальний аналіз предметної області. Розглянуто напрямки діяльності наводяться основні технічні аспекти щодо вирішення поставленої задачі, описуються запропоновані рішення, отримані результати з розробки функціональних та програмних складових кваліфікаційної роботи.

У другому розділі описано програмну реалізацію проекту. Для розробки проекту використано мову програмування Java з технологією JSP та середовище розробки EclipseNeon 2 IDE. В якості СУБД обрано MS SQL Server 2017.

Після процесу проектування бази даних, створено базу даних у СУБД MS SQL за допомогою DDL скрипта та внесені усі необхідні дані.

Обґрунтовано вибір засобів для створення таблиць та формування основних SQL запитів, що відповідають вимогам клієнта.

ANNOTATION

Software and algorithmic complex for product and commodity operations accounting of the hairdressing salon // Qualification work of the educational level "Bachelor"// Pavelets Alisa Romanovna // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Sciences, SNsn-42 group // Ternopil, 2022 // C.55, fig. -33, table. -11, bibliogr. -22.

Key words: software design and development, system testing, JSP technology, databases.

The purpose of this qualification work of the educational level "Bachelor" is to develop a software system for accounting of goods and commodity transactions of the hairdressing salon.

The first section provides a detailed analysis of the subject area. The directions of activity are considered the basic technical aspects concerning the decision of the set task, the offered decisions are described, the received results on development of functional and program components of qualifying work.

The second section describes the program implementation of the project. Java programming language with JSP technology and EclipseNeon 2 IDE development environment were used to develop the project. MS SQL Server 2017 was chosen as the database.

After the database design process, a database was created in the MS SQL database using a DDL script and all the necessary data was entered.

The choice of tools for creating tables and generating basic SQL queries that meet customer requirements is justified.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПРОЕКТУВАННЯ ОБЛАСТІ.....	8
1.1 Коротка характеристика об'єкту опис предметної	8
1.2 Огляд і аналіз існуючих аналогів специфікація вимог до системи...13	
1.3 Розробка архітектури програмної системи	20
1.4 Висновок до першого розділу.....	29
РОЗДІЛ 2. ПРОГРАМНА РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ.....	30
2.1. Програмна реалізація проекту.....	30
2.2 Програмна реалізація бази даних.....	35
2.3 Тестування.....	40
2.4 Інструкція користувача.....	45
2.5 Висновок до другого розділу.....	47
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.....	49
3.1 Методи підвищення мотивації безпеки праці.....	49
3.2. Забезпечення захисту працівників суб'єкта господарювання від іонізуючих випромінювань.....	50
2.5 Висновок до третього розділу	53
ВИСНОВКИ.....	54
ПЕРЕЛІК ДЖЕРЕЛ.....	55

ВСТУП

Актуальність теми. Будні сучасного бізнесу не можливо уявити без прогресивних інформаційних технологій. На сьогоднішній день інформаційні технології в індустріальній красі розглядаються вже не як перспектива, а як необхідна умова успішної діяльності кожного підприємства. Отож, найбільш очевидним способом підвищення ефективності трудового процесу є його автоматизація.

У сучасному суспільстві салони краси розвиваються швидкими темпами, тому важливим постає питання максимально ефективного управління, яке в першу чергу забезпечується швидкою взаємодією з клієнтами, контролем збереження товарів, отриманням актуальних даних про стан товарних запасів. Отже, нам необхідно вести облік всіх товарів що поступають і відображати всі операції, пов'язані з їх вибуттям.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка програмної системи для обліку товарів перукарського салону. Реалізація поставленої мети здійснюється шляхом вирішення наступних задач:

- проведення детального опису проблеми, яка виникла у салоні краси та аналіз існуючих систем для обліку товарів у салонах краси;
- проектування та розробка програмного забезпечення, яке призначене для автоматизації бізнес-процесів з використанням сучасних інформаційних технологій;
 - програмна реалізація проектних рішень
 - проведення системного тестування та представлення результатів
 - провести апробацію розробленої системи.

Об'єкт дослідження – діяльність салону краси.

Предмет дослідження – сукупність теоретико–методологічних, методичних та практичних питань, щодо формування системи для обліку товарів і товарних операцій салону краси.

Методи дослідження. Теоретичною та методологічною основою дослідження є наукові концепції та теоретичні розробки в області аналізу діяльності салонів краси, маркетингу, закони України. Для проектування програмного комплексу для обліку товарів та забезпечення його функціональної придатності використано об'єктно-орієнтований підхід а також – методологію гнучкої розробки програмного забезпечення (англ. Agile software development), а для його створення – мову Java та пакет SWING – для створення графічного інтерфейсу. Також для групування компонентів інтерфейсу використано контейнер JFrame.

Практичне значення одержаних результатів. Практичне значення одержаних результатів полягає у тому, що на основі запропонованих методів вирішення даної проблеми створено програмний комплекс, який застосовується у салоні краси.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПРОЕКТУВАННЯ

1.1 Коротка характеристика об'єкту опис предметної області

Салон краси - місце, де здійснюються найзаповітніші жіночі мрії, де панує атмосфера гармонії та краси. Основним завданням салону краси є надання якісних послуг клієнтам на професійному рівні з використанням професійних матеріалів, обладнання та висококваліфікованим персоналом.

Салон краси надає такі послуги: чоловічі, жіночі та дитячі стрижки, стрижки вусів та борід, вистриження полосок, ламінування, полірування, кератинування волосся, весільні зачіски, догляд за обличчям, корекція і фарбування брів фарбою і хною, фарбування вій, манікюр, педикюр, візаж, лазерна епіляція, вертикальний солярій. Крім того здійснює продаж професійної косметики по догляду за волоссям і тілом, парфумерії, біжутерії, всього для манікюру.

Також у салоні готують майбутніх майстрів. Періодично проводяться курси перукар-універсал, нарощування волосся та вій, нарощування нігтів, манікюру та педикюру.

Салон краси пропонує досить широкий спектр послуг, прагнучи, щоб клієнт отримав всі бажані процедури в одному місці. Асортимент пропонованих послуг постійно розширюється, адже тепер люди не тільки замовляють послуги, але й мають змогу придбати потрібні засоби догляду у салоні.



Рисунок 1.1– Організаційна структура салону краси

Власниця веде організацію всієї роботи у салоні, а також займається підбором необхідного персоналу. Вона здійснює бухгалтерський облік та веде контроль чи ефективно використовуються матеріальні, трудові і фінансові ресурси. А також забезпечує салон косметичними матеріалами та технічним устаткуванням, слідкує за дотримання працівниками правил безпеки; санітарних вимог. В обов'язки директора входить і розгляд скарг та пропозицій та вжиття заходів щодо усунення зазначених недоліків.

Для успішної роботи і заняття впевнених позицій на ринку індустрії краси необхідні не тільки якісні процедури, але і постійне управління процесами, створення умов для залучення нових клієнтів, чіткий облік матеріалів, що витрачаються та управління продажом та списанням товарів. Саме автоматизація дозволяє виконувати всі процеси якісно, щоб клієнтам знову хотілося повернутися в салон. А також це значно полегшить роботу адміністратора, скоротивши час на ведення обліку матеріалів салону, формування списання, тобто товару, який використовується персоналом для надання послуг та продажу товарів.

Розглянемо, які можливості надаватиме система кінцевому користувачеві та яка інформація необхідна для обробки запиту користувача. При цьому механізм функціонування системи від користувача приховано і при визначенні бізнес-процесів не відображається.

В нашому випадку в ролі кінцевого користувача виступає адміністратор салону.

Розглянемо детальніше основні процеси. Розпочнемо огляд з процесу авторизації у систему.

Таблиця 1.1–Характеристика бізнес-процесу авторизація

Назва характеристики	Значення характеристики
Ім'я бізнес процесу	Авторизація
Основні учасники	Адміністратор
Вхідна подія	Запуск програми
Вхідні документи	Відсутні
Вихідна подія	Авторизований користувач
Вихідні документи	Відсутні

Користувач може авторизуватись в систему, ввівши коректний логін і пароль. А також є можливість зміни паролю при умові, що правильно введений старий пароль.

Таблиця 1.2– Характеристика бізнес-процесу Додавання нового товару

Назва характеристики	Значення характеристики
Ім'я бізнес процесу	Додавання нового товару
Основні учасники	Адміністратор
Вхідна подія	Поступив новий товар
Вхідні документи	Накладна
Вихідна подія	Прийнятий товар
Вихідні документи	Журнал доступними товарами

Стандартний сценарій полягає у тому, що авторизований користувач обирає категорію Товари і натискає Додати товар. З'являється вікно з відповідними полями для заповнення. Після введення усіх необхідних даних потрібно натиснути кнопку Зберегти, щоб дані потрапили у базу.

Неможна додати ідентичний товар двічі, можна лише змінити його кількість. Також користувач не може додати товар, якщо незаповнені усі необхідні поля (позначені *).

Після додавання товарів одразу можна переглянути список усіх доступних товарів салону.

Таблиця 1.3– Характеристика бізнес-процесу Формування звіту

Назва характеристики	Значення характеристики
Ім'я бізнес процесу	Формування звіту
Основні учасники	Адміністратор
Вхідна подія	Потреба перегляду звіту
Вхідні документи	Відсутні
Вихідна подія	Отриманий звіт
Вихідні документи	Звіт певної категорії

Оскільки для досягнення прибутку потрібно весь час вести облік продукції у салоні краси, то невід'ємною функцією є формування звітів різного роду.

Стандартний сценарій полягає у тому, що клієнт потрапляє у систему і обирає категорію Звітність. Наступним кроком буде вибір типу звіту (звіт по прийнятих товарах, звіт про залишки тощо) та період часу для якого він формується.

Після створення звіт відкривається у новій вкладці браузера, його можна зберегти локально на комп'ютері або роздрукувати.

Таблиця 1.4 –Характеристика бізнес-процесу Продаж товару

Назва характеристики	Значення характеристики
Ім'я бізнес процесу	Продаж товару
Основні учасники	Адміністратор
Вхідна подія	Потреба клієнта
Вхідні документи	Відсутні
Вихідна подія	Оновлені дані
Вихідні документи	Квитанція на оплату

Стандартний сценарій полягає у тому, що користувач обирає категорію Продаж і формує замовлення клієнта.

Є також певне обмеження для користувача, наприклад неможна продати більшу кількість товару ніж зареєстровано в системі.

Функції адміністратора салону краси є структурним елементом загального процесу функціонування. Тому дуже важливо наскільки якісно і швидко будуть реалізовані бізнес-процеси для адміністратора.

SWOT аналіз інформаційної системи для обліку товарів салону краси представлений у Таблиці 1.5.

Таблиця 1.5 – SWOT аналіз інформаційної системи

Strengths – сильні сторони	Weaknesses – слабкі сторони
<ul style="list-style-type: none"> - Зміна стандартних форм документів відбувається не часто; - Скорочення часу на роботу, пов'язану з обробкою первинної інформації. 	<ul style="list-style-type: none"> - Неповне і неефективне використання технічних засобів – <i>надмірна вартість</i>; - Не спроможність самостійної ресстрації клієнта у системі – <i>відсутня функція; надмірна вартість</i>
Opportunities – можливості	Threats - загрози
<ul style="list-style-type: none"> - Адаптована під потреби салону; - Можливість додання додаткового функціоналу при потребі. 	<ul style="list-style-type: none"> - Втрата інформації, внаслідок непередбачуваних обставин; - Недоліки в процесах збору, занесення, зберігання, передачі і використання інформації

Проаналізувавши ситуацію, яка складається на даний момент у салоні, бачимо, що всі операції по запису клієнтів, послуг, замовлення продукції – відбуваються вручну. Облік зазвичай відбувається за допомогою журналу, що дуже сильно позначається на часі обслуговування. При записи клієнта в салон доводиться враховувати багато факторів, таких як дні роботи майстра, тривалість всіх процедур і наявність вільного часу. Також не завжди коректно вдається записати прихід товару і відповідно до цього зрозуміти наявність необхідної продукції в салоні.

З впровадженням інформаційної системи, процеси роботи салону краси стануть набагато зручнішими і швидшими у виконанні. Багато дій які вимагали уваги і забирали час персоналу будуть автоматизовані. Буде реалізовано вибір найбільш підходящого часу для запису до майстра на основі обраних послуг та днів його роботи, облік закупленої продукції у постачальників, генерація звітів. Все це спростить процес і позбавить адміністратора від непотрібної роботи. Авторизація дозволить запобігти несанкціонованому доступу ззовні.

Для проектування бази даних використовуємо ERD, де кожен рядок кожної таблиці являє собою один екземпляр сутності. Сутність представляє собою існуючий об'єкт або предмет, а також вона має свої атрибути. На ERD сутності зображуються прямокутниками, з'єднаними між собою зв'язками. Таке графічне представлення полегшує сприйняття структури бази даних в порівнянні з текстовим описом.

На даному етапі будемо використовувати логічну модель, адже вона, завдяки своїй абстракції, дозволяє чітко визначити об'єкти, з яких буде складатись система. Об'єкти моделі, які представлені на логічному рівні, називаються сутностями та атрибутами. Логічна модель даних являється універсальною і ніяк не пов'язана із конкретною реалізацією СУБД. На рисунку 1.2 відображено логічну модель даних системи обліку товарів для салону краси.

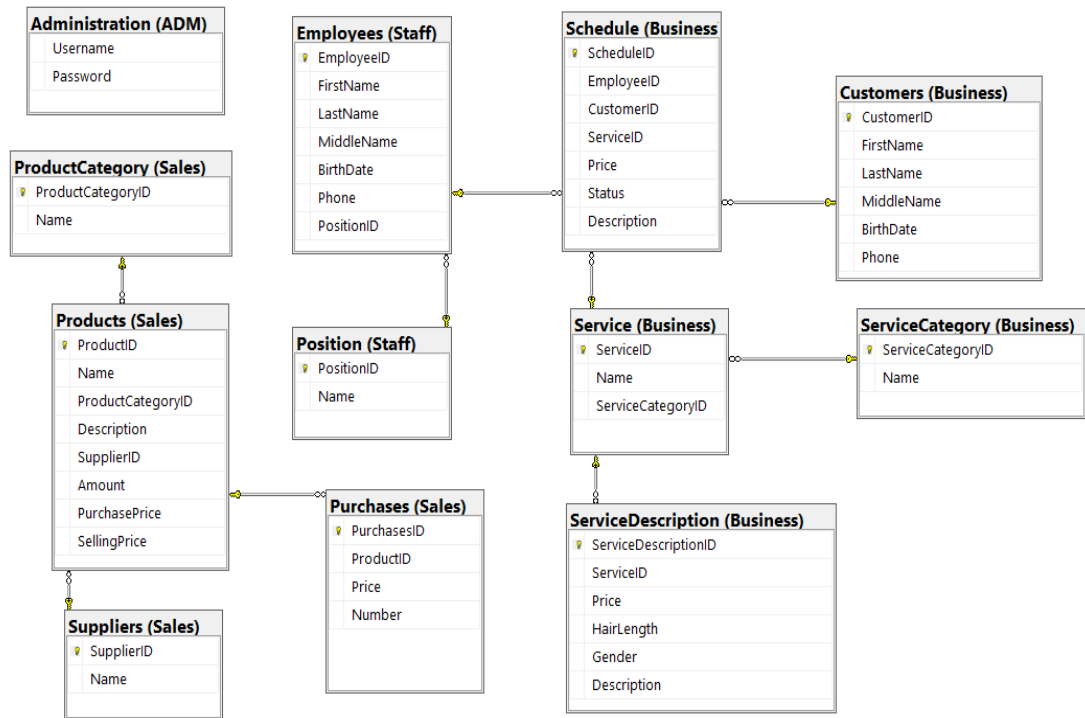


Рисунок 1.2 – Логічна модель даних

Отже нам потрібно розробити комплекс який включатиме: прийом товарів від постачальника та внесення їх у базу даних, можливість продажу та списання товарів, можливість формування звітів відповідно до вимог власниці салону. Провівши детальний аналіз альтернативних рішень для розроблюваного проекту я обрала розробку власного системного програмного комплексу для обліку та товарних операцій у салоні краси.

1.2 Огляд і аналіз існуючих аналогів специфікація вимог до системи

Програмні продукти для салонів краси не є унікальними у своєму роді. Існує безліч аналогів, як платних так і безкоштовних. Та новий програмний продукт створений безпосередньо для вирішення конкретних задач салону краси. Оскільки дуже важко знайти універсальний програмний продукт, який відповідав би конкретним вимогам адміністрації салону. Більшість таких систем мають широкий функціонал та високу вартість, яка не завжди виправдовує покупку. Саме фактор високої вартості є ключовим у даній

ситуації, а більшість функцій таких програмних систем не несуть цінності для конкретного салону краси.

В якості аналогів було розглянуто два програмних продукти: «MySelling_2.0» та «Eserviaforbusiness». Перший із них – безкоштовний з вихідним кодом для обліку продажів і товарів. Призначений для невеликих підприємств та виступає відмінною заміною паперової роботи з калькулятором. А «Eserviaforbusiness» - інноваційний продукт, створений для автоматизації бізнесу. Розглянемо детальніше кожен з них.

MySelling_2.0

Можливості MySelling_2.0:

- Створення декількох користувачів, налаштування прав.
- Можливість перегляду продажів.
- Перегляд і друк списку продажів за будь-який день.
- Перегляд статистики за будь-який період.
- Можливість розширювати можливості програми шляхом додавання плагінів.

Переваги: Відкритий вихідний код; безкоштовно.

Недоліки:

- Не коректна робота з товаром: можна продати будь-що, за будь-яку ціну, тобто немає перевірки наявності товару з певною назвою на складі.
- Вікно програми не масштабується, при зміні розміру
- Неможливо видалити помилковий запис

Початкове вікно програми зображено на рисунку 1.3.

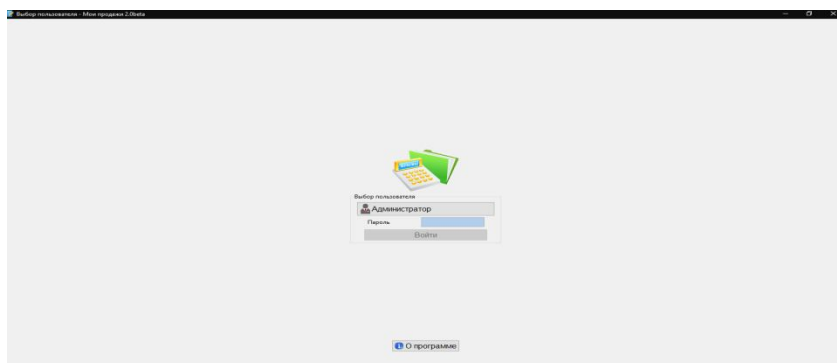


Рисунок 1.3 – MySelling_2.0 – авторизація

На Рис. 1.4. зображено вікно програми MySelling_2.0 після того, як користувач авторизувався. У верхній частині є можливість ввести назву товару, який ми хочемо продати, ціну та кількість. Наступним кроком натискаємо кнопку «Добавить» і вказаний товар з’являється у таблиці. Натиснувши, «Посмотреть продажи» можна побачити продані товари за конкретний період (Див. рисунок 1.5.).

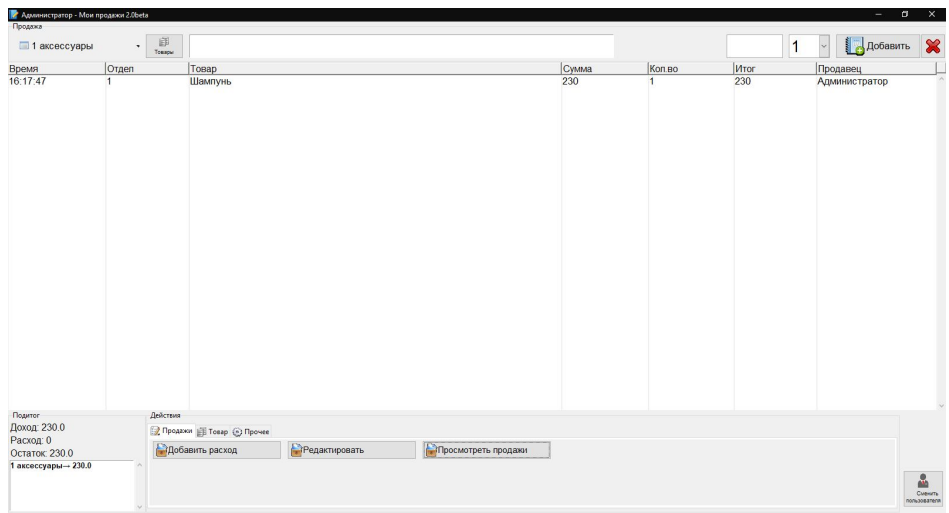


Рисунок 1.4 – MySelling_2.0 – продаж товарів

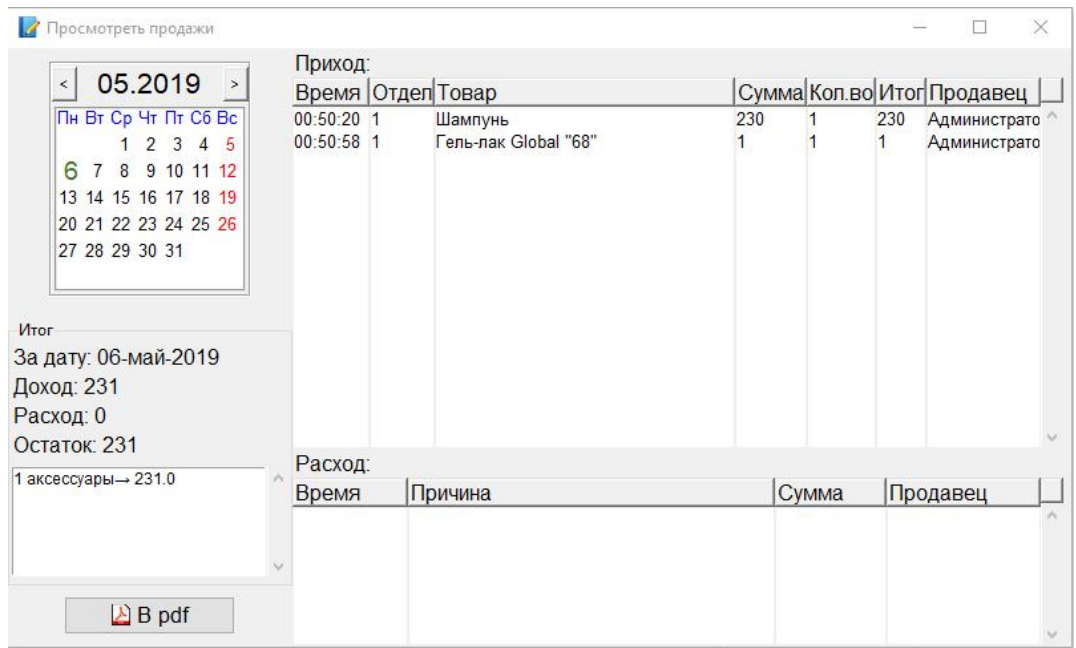


Рисунок 1.5 – MySelling_2.0 – Переглянути продаж

Eserviaforbusiness

Можливості:

- Можливість додавати працівників та створювати їм графік роботи.
- Можливість додавання послуг.
- Бухгалтерія, нарахування зарплат, доходи і витрати.
- Облік клієнтів.
- Календар
- Можливість товарного обліку, додавання постачальників
- Журнал бронювань

Переваги:

- Широкий функціонал для управління.
- 14 днів безкоштовного користування.
- Онлайн бронювання послуг

Недоліки

- Для безкоштовної версії обмежений функціонал, тому складно оцінити всі можливості даного продукту

- Потрібно проводити місячну оплату (від 399 грн до 1199 грн). На

Рис. 1.6. зображено види підписок для користування «Eserviaforbusiness»

	Deluxe 1199 грн/місяць	Premium 799 грн/місяць	Basic 399 грн/місяць
Онлайн бронювання	✓	✓	✓
Розміщення в додатку	✓	✓	✓
Електронний журнал і календар	✓	✓	✓
База клієнтів	✓	✓	✓
Історія візитів клієнтів	Необмежено	Необмежено	До 3 візитів
SMS/Push-сповіщення і нагадування	✓	✓	✓
Кількість послуг	Необмежено	Необмежено	Необмежено
Кількість майстрів	До 25	До 12	До 8
Маркетинг	✓	✓	—
Аналітика	✓	✓	—
Облік товарів	✓	—	—
Розрахунок зарплат і фінанси	✓	—	—
Додаток для майстрів Eservia Works	✓	✓	—
Приєм бронювань наперед	До 361 дня	До 90 днів	До 30 днів
Реакція технічної підтримки	До 4 годин	До 24 годин	До 48 годин
	ВИБРАТИ ПЛАН	ВИБРАТИ ПЛАН	ВИБРАТИ ПЛАН

Рисунок 1.6 – Eservia for business – плани і підписки

Для ідентифікації в системі Eservia, Вам потрібно заповнити профіль користувача. Це особиста інформація користувача бізнес акаунта, яка не буде доступна клієнтам.

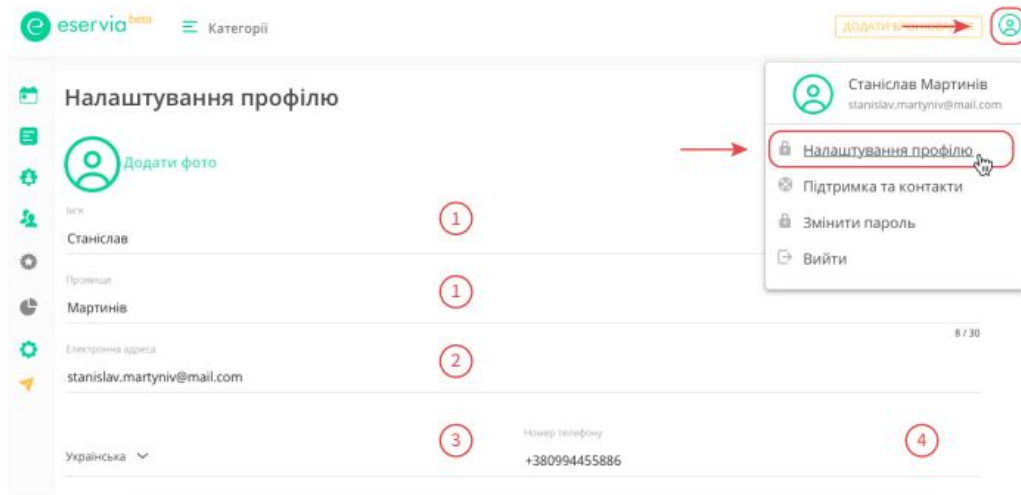


Рисунок 1.7 – Eservia for business – профіль користувача

На Рис. 1.8. зображено головне вікно програми *Eserviaforbusiness*.

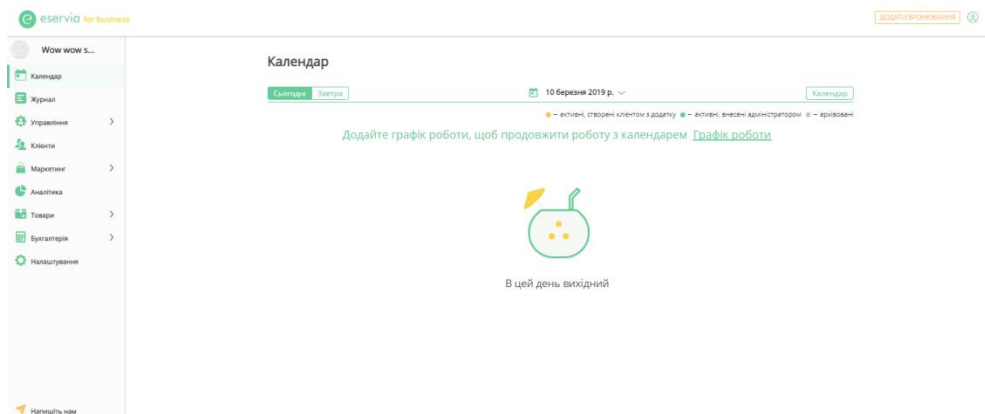


Рисунок 1.8 – Eserviaforbusiness – головне вікно

На рисунку 1.9. виділено вкладку на яку потрібно перейти, щоб почати роботу з товарами.

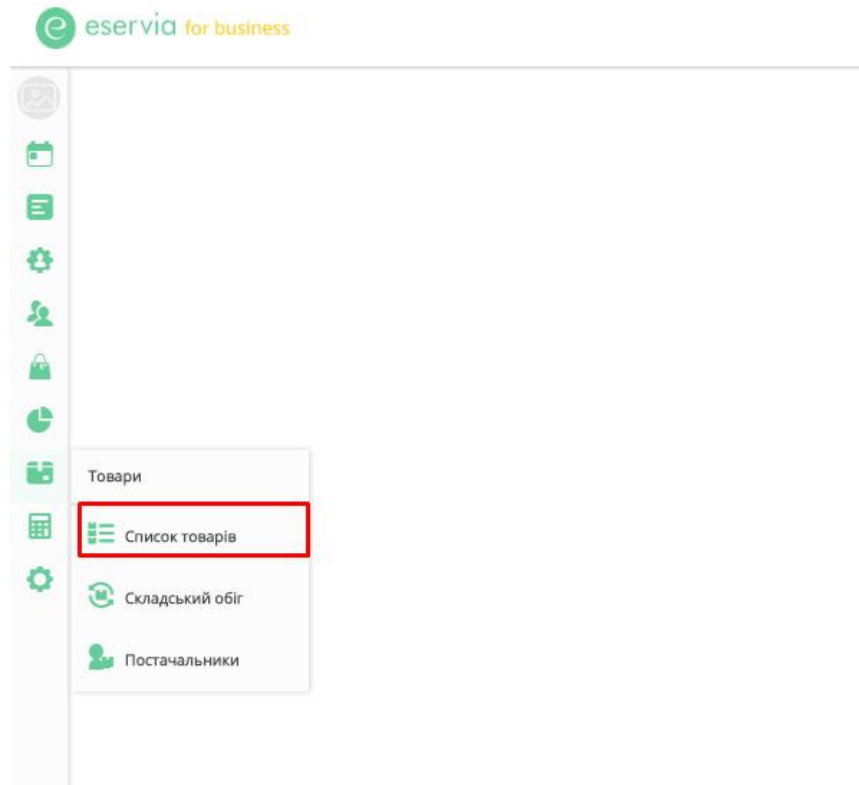


Рисунок 1.9 – Eserviaforbusiness – робота з товарами

Порівняльна характеристика програмних продуктів наведена у таблиці 1.6.

Таблиця 1.6 – Порівняльна характеристика програмних продуктів

Фірма-розробник	KAIST	Eservia LLC
Версія	2.0	For business
Назва	MySelling_2.0	Eservia for business
Оплата	Безкоштовно	Від 399 до 1199/місяць
Інтерфейс	Простий, однотипний	Гнучкий, адаптивний
Допомога користувачу	Екранні повідомлення про помилку	Онлайн підтримка, екранні повідомлення

Проаналізувавши вищевказані програмні продукти можна прийти до висновку, якщо безкоштовно – то не якісно, якщо якісно – то дорого. Таким чином, можна прийти до висновку, що набагато дешевше і швидше написати окремий програмний продукт, який вирішує конкретну проблему даного салону. А також є простим у встановленні та використанні – дешевому та зручному використанні.

Система представляє собою веб-орієнтований додаток, написаний на мові Java з використанням технології JSP. В якості СУБД обрано MSSQLServer 2017. Оскільки продукт є кросплатформним рішенням, його використання не прив'язується до конкретної операційної системи. Потрібно налаштувати локальну мережу в офісі салону, віртуальну машину JAVA і контейнер для сервлетів Apache Tomcat 9.

При реалізації поставлених завдань буде досягнута мета ПП, що розробляється: підвищити якість послуг, що надаються салоном краси, пришвидшити і полегшити роботу адміністратора.

Уніфікована мова моделювання UML (Unified Modeling Language) є мовою для визначення, подання, проектування і документування програмних систем, організаційно-економічних систем, технічних систем та інших систем різної природи. UML містить стандартний набір діаграм і нотацій найрізноманітніших видів.

Діаграма варіантів використання (use case diagrams) відображає функціональність, яка реалізована в програмному продукті (Див. рисунок 1.10). Кожен варіант використання це функція у програмному продукті.

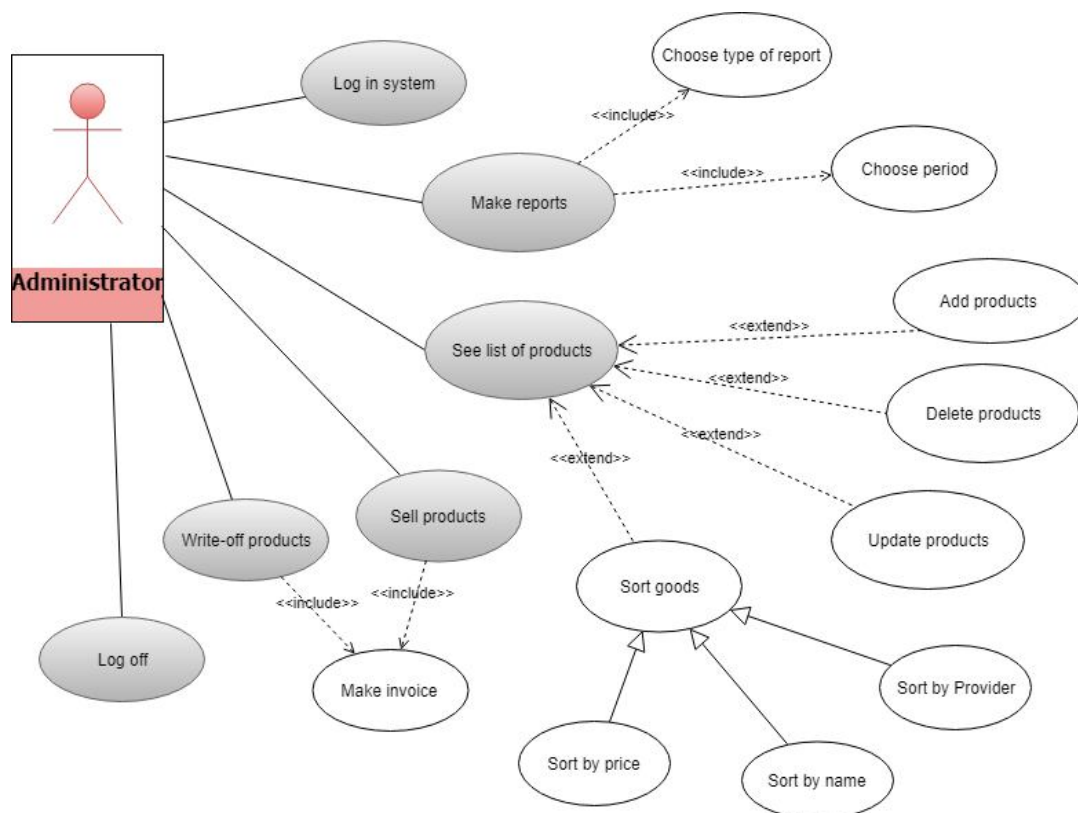


Рис. 1.10. – Діаграма варіантів використання

1.3 Розробка архітектури програмної системи.

Для розробки програмної системи для салону краси використано 3-ох рівневу архітектуру. Такий тип часто використовується у програмних продуктах як специфічний тип клієнт-серверної системи. Трирівнева архітектура має багато переваг для розроблюваного середовища шляхом модулювання користувацького інтерфейсу, бізнес-логіки та шарів зберігання даних. Це дає більшу гнучкість команді розробників, дозволяючи оновлювати певну частину програми незалежно від інших частин. Така гнучкість може поліпшити загальний час на розробку та тестування системи.

Наприклад, користувацький інтерфейс системи може бути перероблений або модернізований, не впливаючи на функціональність та логіку доступу до даних. Ця архітектурна система ідеально підходить для вбудовування та інтеграції нового програмного забезпечення в існуючу програму. Таким чином трирівневі архітектури часто використовуються у локальних додатках.

На рисунку 1.11. зображено загальний вигляд трирівневої архітектури.



Рисунок 1.11 –Трирівнева архітектура

Тепер розглянемо детальніше кожен шар трирівневої архітектури.

Презентаційний рівень (PresentationLayer) – це фронтальний рівень в трирівневій архітектурі, який складається з інтерфейсу користувача. Інтерфейс користувача є графічним і доступним через веб-браузер або веб-додаток, та відображає дані корисні для кінцевого користувача. Цей рівень часто побудований на веб-технологіях, таких як HTML5, JavaScript, CSS і зв'язується з іншими шарами через ви-клик API.

Рівень додатку (ApplicationLayer) – містить функціональну бізнес-логіку, яка керує основними можливостями програми. Зазвичай цей рівень написаний на Java, .NET, C #, Python, C ++ і т.д.

Рівень даних (DataLayer) – складається з бази даних / системи зберігання даних і рівня доступу до даних. Прикладами таких систем є MySQL, Oracle,

PostgreSQL, Microsoft SQL Server, MongoDB тощо. Доступ до даних здійснюється прикладним рівнем через виклики API.

Також для проектування додатку використано Java ServerPages, які є частиною тривірневої архітектури. Сервер (як правило, Application або Web Server) підтримує серверні сторінки Java. Цей сервер буде діяти як посередник між клієнтським браузером і базою даних. На діаграмі(Див. рисунок 1.12) зображено архітектуру JSP.

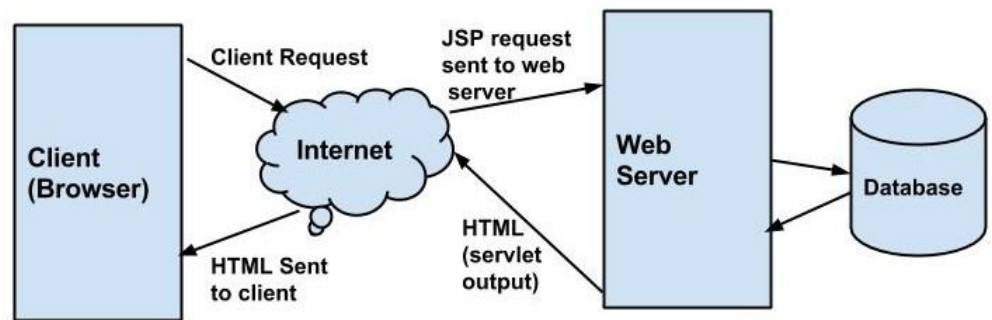


Рисунок 1.12 – JSP архітектура

Тепер детальніше розглянемо як відбувається потік JSP:

1. Користувач переходить на сторінку JSP і робить запит через у веб-браузері.
2. Запит JSP надсилається веб-серверу.
3. Веб-сервер приймає запит .jsp і передає JSP-файл JSP Servlet Engine.
4. Якщо файл JSP був викликаний вперше, то його буде проаналізовано. Наступним кроком є створення сервлету з JSP-файлу. Згенерований сервлет надсилається через Інтернет-форму веб-сервера користувачам веб-браузера.
5. Останнім кроком є відображення HTML результату у веб-браузері користувача.

Сторінка JSP - це ще один спосіб написати сервлет без необхідності бути фахівцем з програмування Java. Сторінка JSP обробляється так само, як звичайний сервлет. Тепер, коли ми побачили загальну картину архітектури JSP, можна зануритися більш глибоко і зрозуміти, як обробляти JSP файл в контейнері і які фази він проходить.

Життєвий цикл JSP схожий на життєвий цикл сервлету з додатковим кроком, який компілює JSP в сервлет. Сторінки JSP зазвичай керуються веб-контейнером.

У таблиці 1.7. наведено етапи життєвого циклу JSP та їх опис.

Таблиця 1.7 – Життєвий цикл JSP

Етап	Опис
Translation	Контейнер JSP парсить сторінки JSP. Потім він переводить сторінки JSP для створення відповідного вихідного коду сервлету.
Page Compilation	Якщо переведення успішне, то сформований java-файл компілюється контейнером.
Class Loading	Коли JSP буде зкомпільовано як сервлет він завантажиться в пам'ять.
Instance Creation	Коли клас JSP завантажується в пам'ять, його об'єкт створюється контейнером
Call jspInit()	Під час цієї фази клас JSP ініціалізується з нормального класу в сервлет. Після завершення ініціалізації об'єкти ServletConfig і ServletContext стають доступними для класу JSP. Метод jspInit () викликається один раз у життєвому циклі JSP. Він ініціалізує параметри конфігурації.
Call _jspService or Request Processing	Цей метод викликається для кожного запиту клієнта.
Call jspDestroy or Destroy	Це останній етап, який викликається, коли контейнер вирішує вивантажити JSP з пам'яті.

Типовий сценарій використання застосунку представлений на рисунку 1.13. Як ми бачимо з використанням обраного типу архітектури ми підвищуємо надійність і доступність програмного продукту.

1. Користувач вводить адресу в веб-браузері і в браузері URL-адреса декодується, тобто ім'я хоста перетворюється в IP-адресу. Потім запит надсилається на віддалений сервер з використанням протоколу HTTP.

2. Повертається відповідна HTML-форма.

3. На наступному рівні функціональність програми забезпечується шляхом виконання детальної обробки даних з рівня презентації. Для підтримки цього використовується сервер ApacheTomcat.

4. З сервером (Apache) ідентифікуються відповідні дії, такі як вибірка файлу або передача запиту інтерпретатору.

5. Скрипт взаємодіє з сервером бази даних, таким чином забезпечується доступ до вхідних даних або генерування вхідних даних. Він інтерпретує запити відповідно до бізнес вимог і минулих транзакцій від цього клієнта. А також генерує відображення актуальних даних на HTML формі .

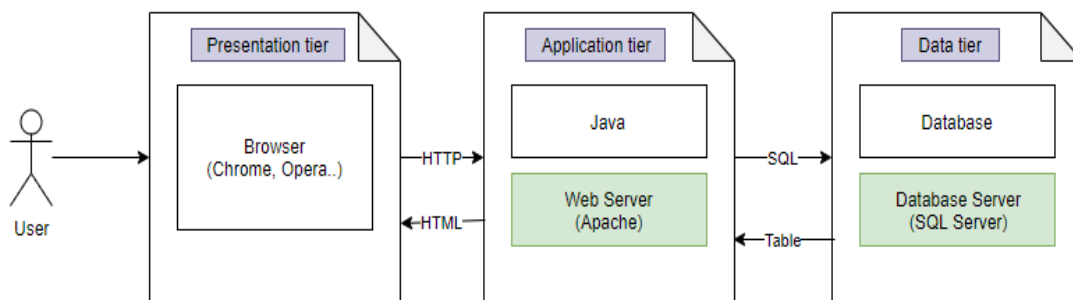


Рисунок 1.13 – Сценарій використання

Для опису основної бізнес-логіки програмного продукту та його короткого опису використано діаграму класів (Див. рисунок 1.14). Також наведено призначення окремих класів розроблюваної системи.

- classIndexController дозволяє користувачу зареєструватися в систему; викликає метод з класу LoginDao, якому передає два параметри введені користувачем: логін і пароль;
- classLoginDao виконує перевірку наявності певного акаунту в системі, звертаючись до бази даних;
- classReport представляє усі можливі звіти, які можна зробити в системі;
- classProduct представляє товари, які наявні у салоні краси, а також посилається на клас ProductCategory та Supplier; наявні методи для додавання, редагування та видалення товарів.

- classPurchase представляє куплені клієнтами товари у салоні; клас пов'язаний з Products та ProductCategory для того, щоб при продажі товару можна було швидко обрати потрібний.
- classWriteoff представляє списані адміністрацією салону товари; клас пов'язаний з Products для того, щоб списання відбувалось лише по наявних товарах;
- classProductCategory представляє категорії товарів, що існують у салоні краси;
- classSupplier представляє постачальників, які безпосередньо здійснюють доставку товарів для салону; пов'язаний з Products та ProductCategory.

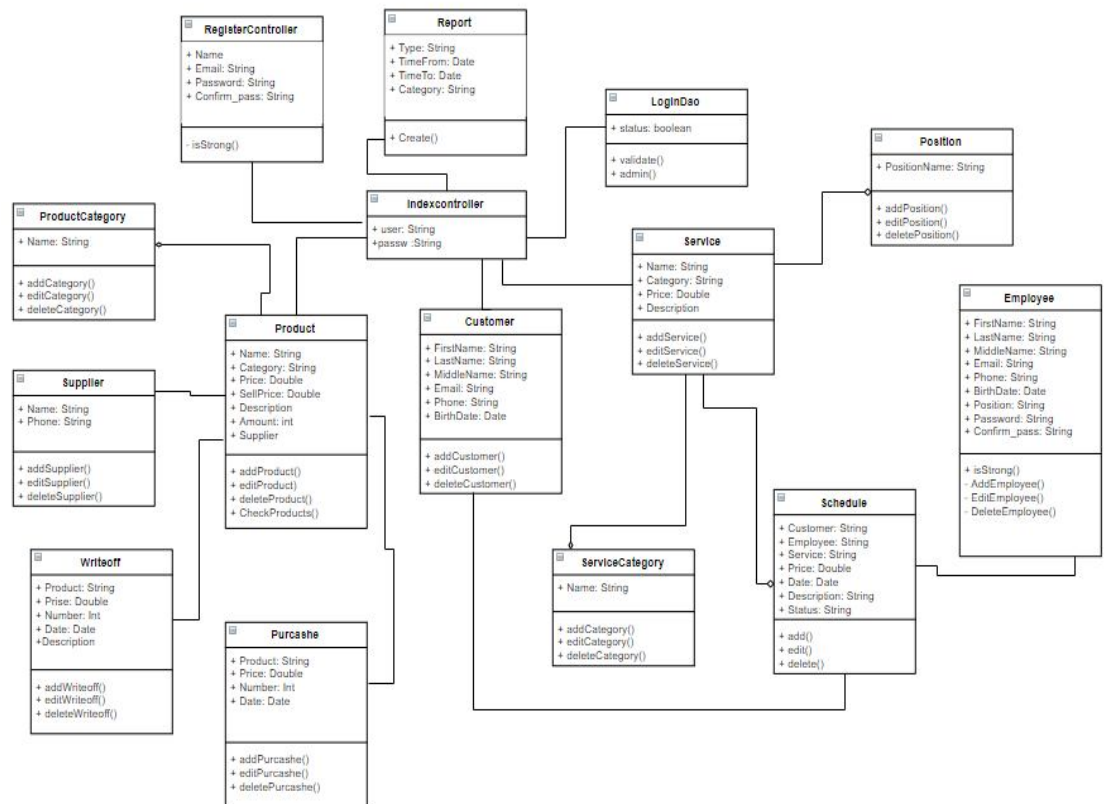


Рисунок 1.14 – Діаграма класів

Отож взаємодія і зв'язки, які наявні між вищепереліченими класами інтерпретують логіку для обліку товарів і товарних операцій для салону краси.

Розглянемо детальніше основні бізнес-процеси пов'язані з обліком товарів. Для більш наглядного представлення та кращого сприйняття зобразимо ці процеси на діаграмі активності.

Одним із головних бізнес-процесів є продаж товару. При програмній реалізації слід врахувати, щоб дані про товари були весь час актуальними. Спочатку користувач потрапляє на вкладку продаж і йому відображається журнал останніх транзакцій. Далі сценарій розгортається наступним чином, користувач обирає кнопку Продати і перенаправляється на відповідну форму. Щоб сформувати замовлення клієнта потрібно заповнити необхідні поля: назву товару або просканувати штрих-код товару і вказати кількість одиниць. Решта потрібної інформації, наприклад сума до оплати підтягнеться з бази даних і також з'явиться на формі. Користувач не може виконати продаж товару якого немає у БД. На рисунку 1.15 продемонстровано сценарій, коли авторизований користувач потрапляє на сторінку Продаж та оформляє покупку певного товару, який в достатній кількості наявний у салоні. Якщо немає необхідної кількості товару, з'являється повідомлення.

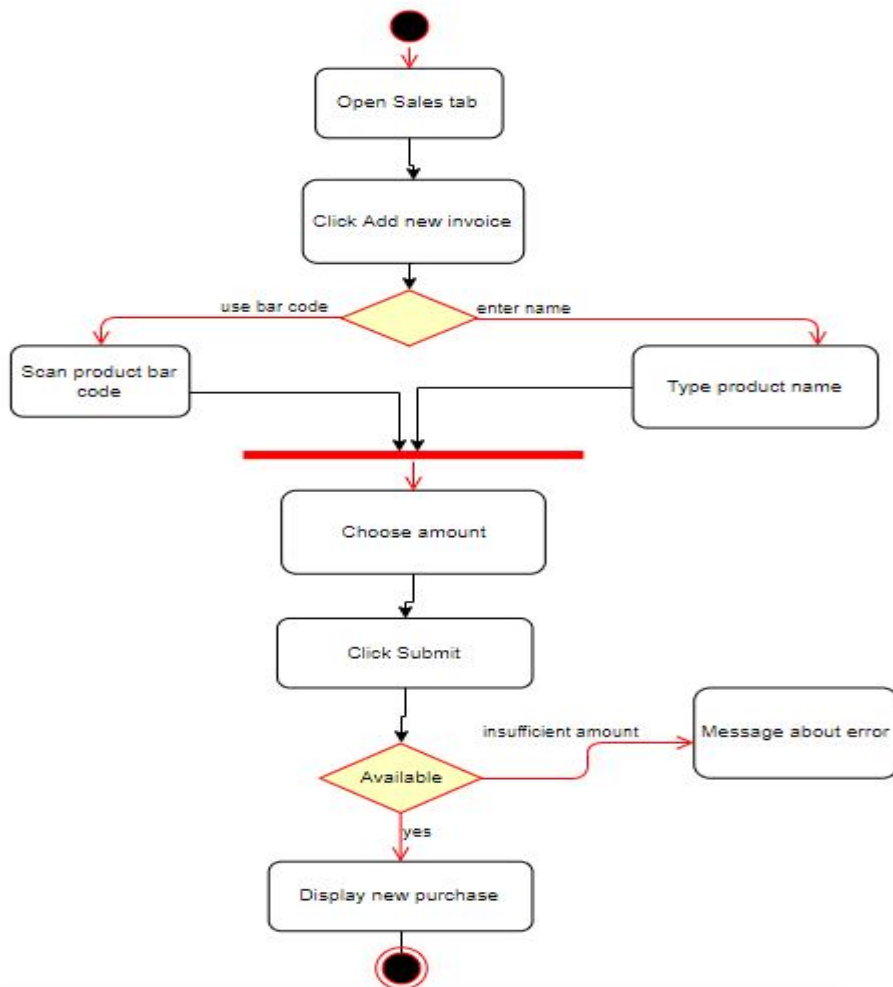


Рисунок 1.15 – Діаграма активності процесу продаж товару

Графічне представлення заповнення форми для процесу продажу товару зображено на рисунку 1.16.

Повторюємо кроки, що зображені на діаграмі активності в програмній системі:

1. Відкриваємо вкладку Продаж.
2. Натискаємо Продати.
3. Обираємо товар та кількість.
4. Натискаємо Підтвердити.
5. Отримуємо відповідь.

В результаті успішного виконання нова покупка буде збережена в базу даних та відображена у журналі.

Рисунок 1.16 – Форма продажу товару

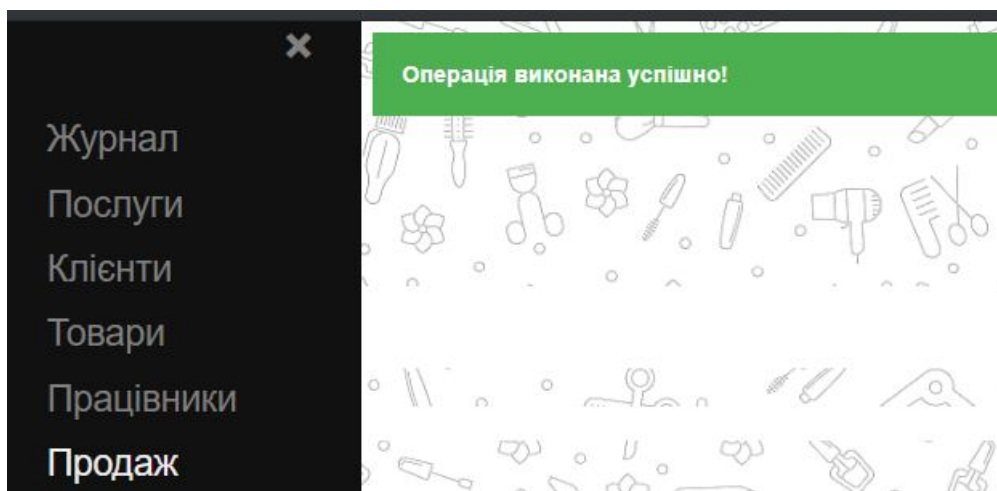


Рис 1.17 – Операція продажу виконана успішно

Наступним розглянемо процес додавання нового товару за допомогою сканування штрих-коду (Див. рисунок 1.18). Та оскільки бувають випадки, що штрих-код не зчитується це все можна ввести і вручну.

Щоб додати товар у базу даних потрібно перейти на вкладку Товари та обрати функцію Додати новий товар. Потім з'являється форма з відповідними полями. Щоб просканувати штрих-код з товару потрібно поставити курсор на поле Штрих-код та натиснути кнопку на сканері. На першому розгалуженні виконується перевірка чи є даний товар у базі.

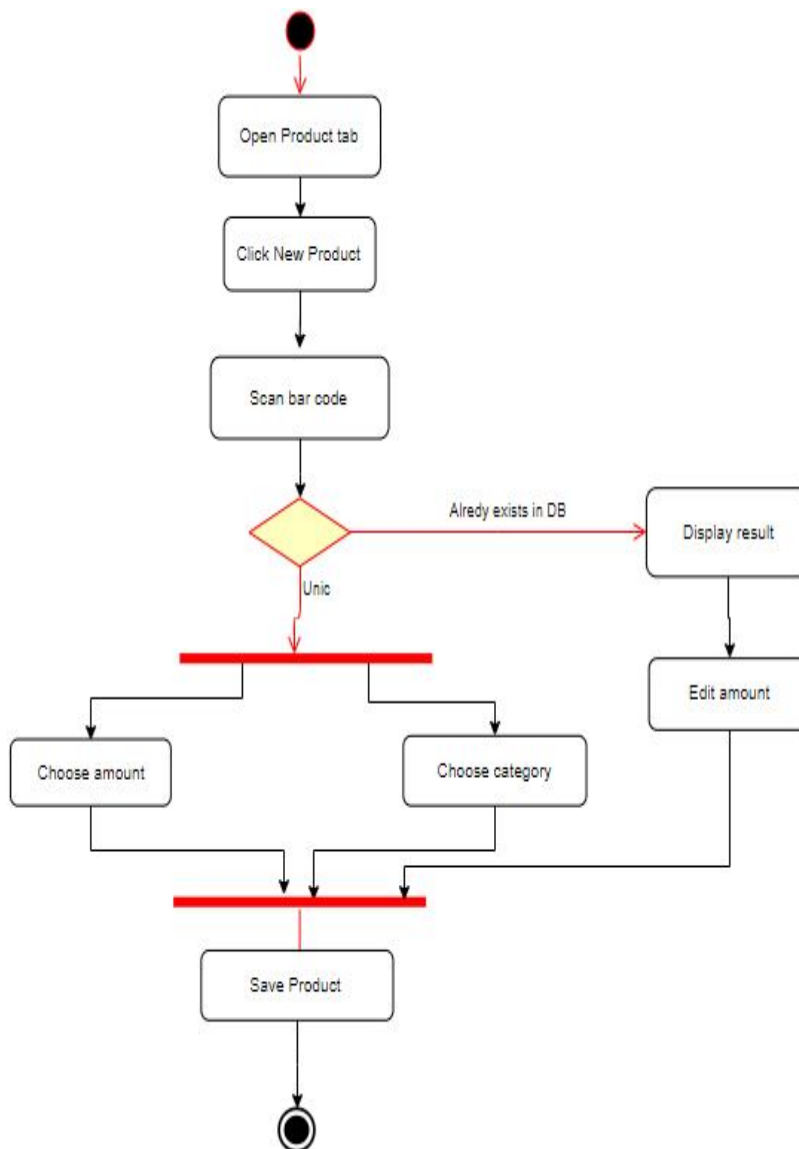


Рис 1.18 – Діаграма активності для додавання товару

Якщо є, то оновлюємо кількість товару та зберігаємо. Якщо ні, то заповнюємо необхідну інформацію про товар та зберігаємо.

1.4 Висновок до першого розділу

У даному розділі проведено детальний аналіз предметної області.

Спроектовано детальну модель програмного забезпечення, що розробляється. Визначено архітектуру системи, спроектовано базу даних, інтерфейс користувача та об'єктну модель програмного засобу. Для опису архітектури використано UML діаграму класів. Логічна архітектура представлена за допомогою діаграми активності.

РОЗДІЛ 2. ПРОГРАМНА РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ

2.1. Програмна реалізація проекту

Програмний продукт для салону краси є веб-орієнтованим додатком. Java Web Application використовується для створення динамічних веб-сайтів. Java надає підтримку для веб-додатків через Servlets і JSP. Можна створити веб-сайт із статичними HTML-сторінками, але коли потрібно, щоб інформація була динамічною, кращим варіантом буде веб-додаток.

Веб-сервер - це програмне забезпечення, яке може обробляти запит клієнта і відправляти відповідь клієнту. Apache є одним з найбільш широко використовуваних веб-серверів. Веб-сервер може працювати на локальній машині і слухати запити клієнта на конкретному порті.

Веб-клієнт - це програмне забезпечення, яке допомагає в спілкуванні з сервером. Найбільш широко використовувані веб-клієнти: Firefox, Google Chrome, Safari і т.д. Коли ми хочемо отримати інформацію із сервера (через URL), веб-клієнт надсилає його на сервер, а потім аналізує відповідь сервера і відображає користувачеві.

Веб-сервер і веб-клієнт є двома окремими програмними засобами, тому для встановлення зв'язку між ними має існувати певна спільна мова. HTML є спільною мовою між сервером і клієнтом. Протокол HTTP (HyperText Transfer Protocol) - це протокол зв'язку між сервером і клієнтом. На рисунку 2.1 зображено приклад запиту до сервера.

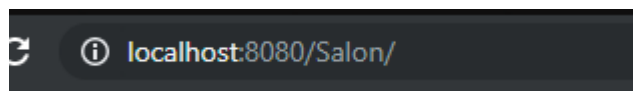


Рисунок 2.1 – Приклад запиту до сервера

URL - аббревіатура від Universal Resource Locator і використовується для пошуку сервера та ресурсу. Кожен ресурс у мережі має свою унікальну адресу.

Розберемо частину нашої URL-адреси. Localhost - унікальна адреса сервера, в більшості випадків це ім'я хоста сервера, яке відображається на унікальній IP-адресі. 8080 - це порт, на якому прослуховується сервер, він є необов'язковим, і якщо ми не надаємо його в URL, запит надходить до порту за замовчуванням протоколу. Номери портів від 0 до 1023 є зарезервованими портами для добре відомих служб, наприклад 80 для HTTP, 443 для HTTPS, 21 для FTP тощо. Salon - ресурс, запитаний з сервера. Це може бути статичний html, pdf, JSP, сервлети.

Веб-сервери є корисними для статичних сторінок HTML, але вони не знають, як генерувати динамічний вміст або як зберігати дані в базах даних, тому нам потрібен інший інструмент, який можна використовувати для створення динамічного вмісту. Для формування динамічного контенту використано Java Servlets і JSP. Java Servlet і JSP є серверними технологіями для розширення можливостей веб-серверів, забезпечуючи підтримку динамічних відповідей і збереження даних.

Для розробки програмної системи використано Eclipse Neon 2 IDE для розробників Java EE. Оскільки сервлет є технологією на стороні сервера, нам знадобиться веб-контейнер, який підтримує технологію Servlet, тому ми будемо використовувати сервер Apache Tomcat. Додамо конфігурацію Eclipse з Tomcat, вона допомагає легко запускати та тестувати наш додаток (Див рисунок 2.2).

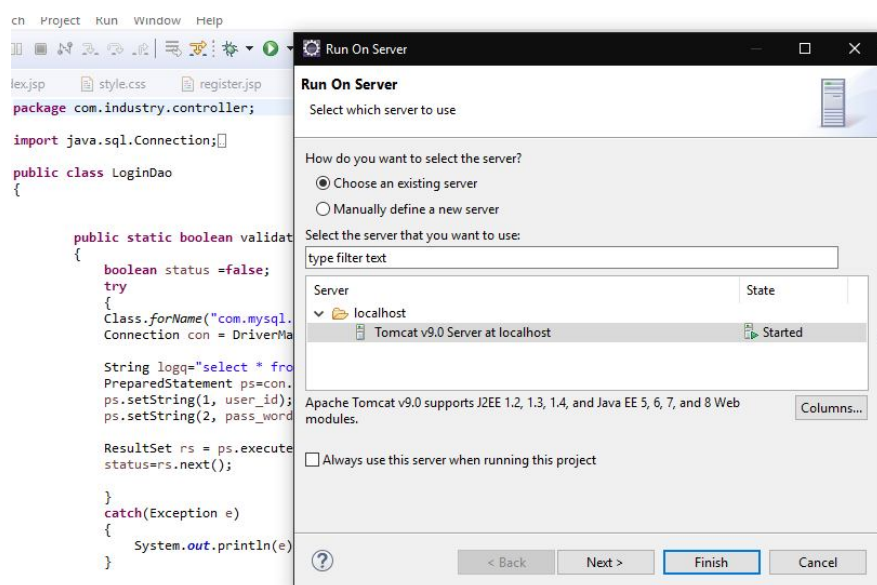


Рисунок 2.2 – Running program on Tomcat server

JSP також є технологією на стороні сервера, і вона схожа на HTML з додатковими функціями для додавання динамічного вмісту, де нам це потрібно. JSP корисні для презентації даних, оскільки їх легко написати, тому що вони схожі на HTML.

Якщо ми запусимо наш index.JSP, то отримаємо результат, що на рисунку 2.3.

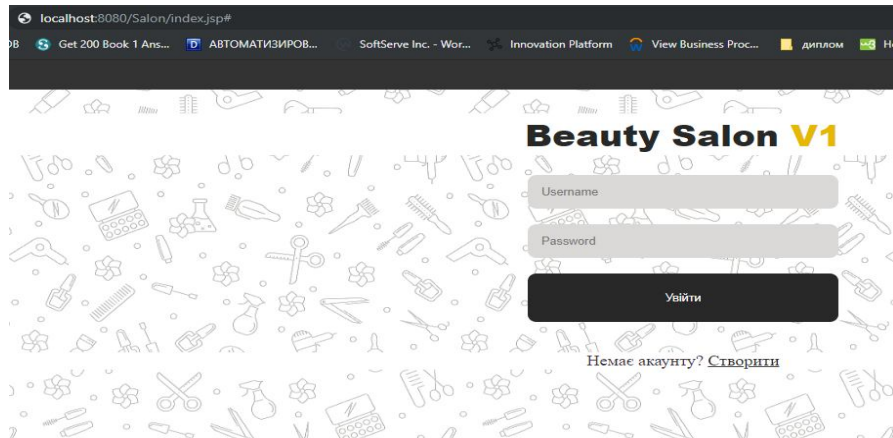


Рисунок 2.3– Результат виконання

Коли веб-контейнер отримує запит для сервлета, то контейнер створює два об'єкти `HttpServletRequest` і `HttpServletResponse`. Далі він знаходить правильний сервлет на основі URL і створює потік для запиту. Потім викликає метод сервлету `service()` і на основі методу `HttpService` () викликає методи `doGet()` або `doPost()`. Методи сервлетів генерують динамічну сторінку і записують її у відповідь. Приклад коду методу `doPost` наведено нижче. Він приймає два параметри і передає їх у метод `validate()`, якщо авторизація пройшла успішно ми потрапляємо на стартову сторінку. У іншому випадку на сторінку з повторним введенням даних.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String user= request.getParameter("uid");
    String passw = request.getParameter("pass");

    if(LoginDao.validate(user, passw))
    {
        if(LoginDao.admin(user)){
            RequestDispatcher rd=
request.getRequestDispatcher("admin.jsp");
            rd.forward(request, response);
        } else{
            RequestDispatcher rd=
request.getRequestDispatcher("successful.jsp");
            rd.forward(request, response);}
    }
}
```

```

    }
    else
    {

        out.println("<script type=\"text/javascript\">");
        out.println("alert('Sorry, Password or Username Error');");
        out.println("</script>");

        RequestDispatcher rd =
request.getRequestDispatcher("unsuccessful.jsp");
        rd.include(request, response);
    }

    out.close();
}

```

Метод `validate()` виконує перевірку на правильність введення даних для авторизації користувача.

```

public static boolean validate(String user_id, String pass_word)
{
    boolean status = false;
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mssql://MOON:1433/databaseName=Salon;user=<user>;pass
word=<password>");

        String logq="select * from administration where username =?
and password =?";

        PreparedStatement ps=con.prepareStatement(logq);
        ps.setString(1, user_id);
        ps.setString(2, pass_word);

        ResultSet rs = ps.executeQuery();
        status=rs.next();

    }
    catch(Exception e)
    {
        System.out.println(e);
    }

    return status;
}

```

Веб-додатки Javaзберігаються як WebArchive (WAR) та мають певну структуру (Див. рисунок 2.4).

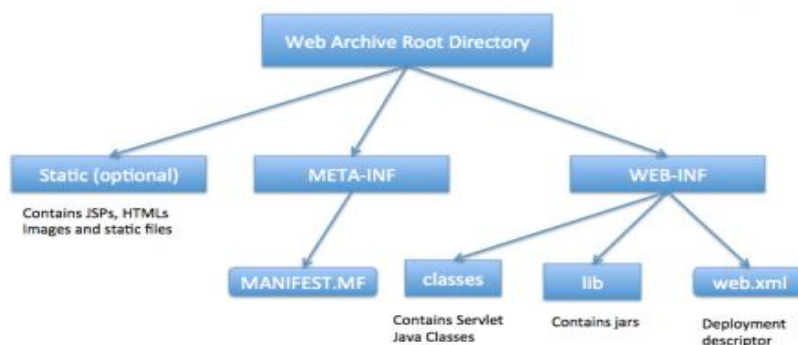


Рисунок 2.4 – Структура додатку

Файл web.xml є дескриптором розгортання веб-програми і містить відображення для сервлетів (до 3.0), початкові сторінки, конфігурації безпеки, налаштування таймауту сеансу тощо. Для коректного відображення кирилиці на формах програми додаємо наступний елемент у web.xml:

```
<jsp-property-group>
<url-pattern>*.jsp</url-pattern>
<page-encoding>UTF-8</page-encoding>
</jsp-property-group>
</jsp-config>
```

Розглянемо клас об'єктних компонентів ProductController, який має такі поля для зберігання даних: ProductID, Name, ProductCategory, Description, PurchasePrice, SellPrice, Amount. Крім того, він має реалізацію наступних бізнес-методів getProductID, getName, getProductCategory, getDescription, getPurchasePrice, getSellPrice, getAmount:

```
public int getProductID() {
    System.out.println("getProductID");
    return ProductID; }

public String getProductCategory() {
    System.out.println("getProductCategory");
    return ProductName; }

public String getName() {
    System.out.println("getName");
    return Name; }

public String getDescription() {
    System.out.println("getDescription");
    return Description;
}

public double getPurchasePrice() {
    System.out.println("getPurchasePrice");
    return PurchasePrice; }
}
```

Для того, щоб підключитись до бази даних необхідно передати URL бази, ім'я користувача і пароль для методу getConnection класу java.sql.DriverManager.

У методі getConnection визначаємо змінну для ConnectionString та створюємо простий запит, щоб перевірити з'єднання.

```
String connectionStr =
"jdbc:sqlserver://MOON:1433;databaseName=Salon;user=<user>;password=<password>";

try (Connection con = DriverManager.getConnection(connectionUrl);
```

```
Statement stmt = con.createStatement();) {
    String SQL = "SELECT TOP 10 * FROM Sales.Products";
    ResultSet rs = stmt.executeQuery(SQL);
```

Для того, аби забезпечити надійну роботу програмного продукту, необхідне наступне програмне забезпечення:

- Операційна система Windows 8.1 або 10,
- Windows Server 2016 або Windows Server 2012 R2 RTM
- SQL Server 2017
- .NET 4.6.1 і вище. Інсталяція SQL Server 2017 встановить необхідні файли перед своєю установкою.

- SQL Server Core engine is 64-bit only and does not support the 32-bit editions of Windows 8 or Windows 10.

- браузер Chrome (version 72)

Щодо апаратного забезпечення необхідно як мінімум (Див. таблиця 2.1):

Таблиця 2.1 – Вимоги до апаратного забезпечення

Оперативна пам'ять	2 GB for Standard, Developer & Enterprise
Файлова система	NTFS or ReFS
Дисковий простір	Minimum of 6 GB
Швидкість процесора	Clockspeed of 2 GHz or more. 1.4 GHz minimum
Ядра процесора	2 cores (Virtual or Physical)
Тип процесора	64 bit x64-compatible AMD or Intel CPU only

2.2 Програмна реалізація бази даних

Для розробки програмного продукту обрано SQLServer 2017. Система управління реляційними базами даних (RDBMS) в даному випадку SQLServerManagementStudio забезпечує менеджмент базою даних. Дані користувача і системи зберігаються в таблицях. У MicrosoftSQLServerManagementStudio таблиці містяться в схемах, які дуже подібні до папок, що зберігають файли в операційній системі. На рисунку 3.5 зображено структуру бази у SSMS. Для бази даних Salon створено 4 схеми:

1. Sales – у схемі зберігаються таблиці, дані яких складають інформацію про проданий та наявний товар, категорії товарів, постачальників та списання.

2. Bussines – у цій схемі знаходяться таблиці, дані яких складають інформацію про клієнтів салону, журнал записів на послуги, види послуг та їх категорії.

3. ADM – у цій схемі знаходяться таблиці, які відносяться до адміністрування базою, наприклад таблиця логів та таблиця із логінами і паролями користувачів.

4. Staff – у цій схемі зберігаються таблиці, в яких зберігаються дані про посади працівників салону та самих працівників

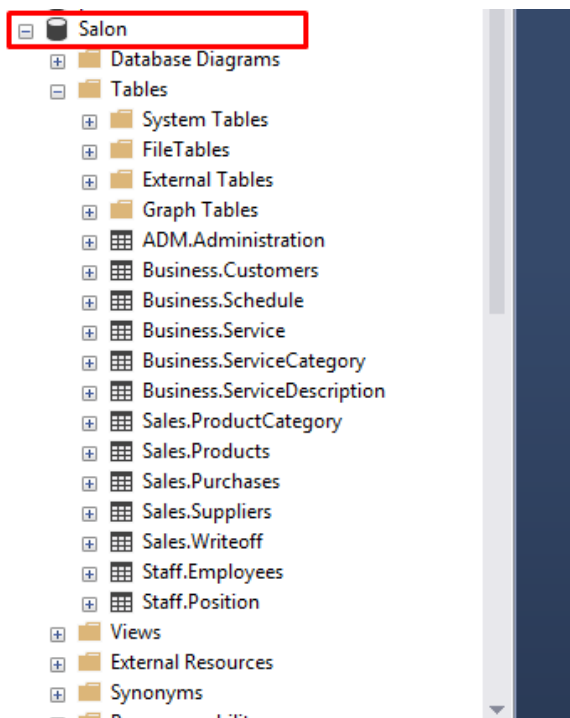


Рисунок 2.5 – Структура бази у SSMS

Тепер детальніше розглянемо скрипти створення таблиць, які призначені для обліку товару у салоні краси. Нижче наведений DDLскрипт для створення таблиць Products, ProductCategory, Suppliers, WriteoffiSales. На рисунку 2.6 представлено назви атрибутів та типи даних таблиці Productsy SQLServerManagementStudio.

```
CREATE TABLE [Sales].ProductCategory (
    ProductCategoryID int not null identity(1,1),
    [Name] NVARCHAR(50) NOT NULL,
    CONSTRAINT PK_ProductCategoryID PRIMARY KEY (ProductCategoryID)
)
GO
```

```

CREATE TABLE [Sales].Suppliers (
    SupplierID int not null identity(1,1),
    [Name] NVARCHAR(50) NOT NULL,
    [Phone] VARCHAR(50) NOT NULL,
    CONSTRAINT PK_SupplierID PRIMARY KEY (SupplierID)
)
GO

CREATE TABLE [Sales].Products (
    ProductID int not null,
    [Name] NVARCHAR(50) NOT NULL,
    ProductCategoryID INT NOT NULL,
    Description NVARCHAR(255) NULL,
    SupplierID INT NOT NULL,
    Amount INT NOT NULL,
    PurchasePrice DECIMAL(19,2) NOT NULL,
    SellingPrice DECIMAL(19,2) NOT NULL,
    CONSTRAINT PK_ProductID PRIMARY KEY (ProductID),
    CONSTRAINT FK_Product_ProductCategory FOREIGN KEY (ProductCategoryID)
REFERENCES [Sales].ProductCategory (ProductCategoryID),
    CONSTRAINT FK_Product_Supplier FOREIGN KEY (SupplierID) REFERENCES
[Sales].Suppliers (SupplierID)
)
GO

CREATE TABLE [Sales].Sales(
    SalesID int identity(1,1) not null,
    ProductID INT NOT NULL,
    Price DECIMAL(19,2) NOT NULL ,
    Number INT NOT NULL,
    [Date] DATE NOT NULL DEFAULT (getdate()),
    CONSTRAINT PK_SalesID PRIMARY KEY (SalesID),
    CONSTRAINT FK_Sales_Product FOREIGN KEY (ProductID) REFERENCES Sales.Products
(ProductID),
)
GO

CREATE TABLE [Sales].Writeoff(
    WriteoffID int identity(1,1) not null,
    ProductID INT NOT NULL,
    Price INT NOT NULL DEFAULT 0 ,
    Number INT NOT NULL,
    [Date] DATE NOT NULL DEFAULT (getdate()),
    [Description] VARCHAR(255),
    CONSTRAINT PK_WriteoffID PRIMARY KEY (WriteoffID),
    CONSTRAINT FK_Writeoff_Product FOREIGN KEY (ProductID) REFERENCES
Sales.Products (ProductID),
)
GO

```

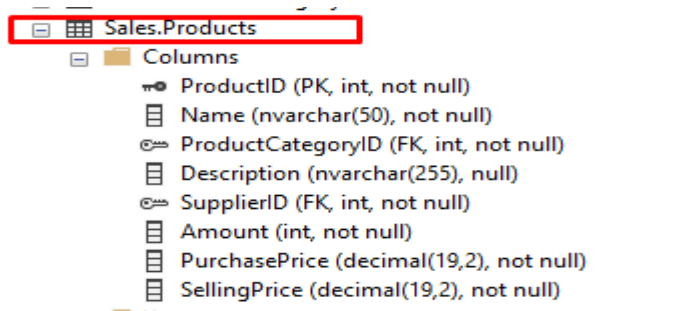


Рисунок 2.6 –Вигляд таблиці Products у SSMS

Під час додавання даних у базу важливо забезпечити їх цілісність. Перш за все дані, які додаються у кожну таблицю перевіряються на цілісність домену – це відбувається на рівні перевірки типу даних та допустимості NULL значень. Наприклад, якщо у таблиці Sales атрибут Date має тип даних Date, то користувач не зможе помилково внести у це поле ніщо, окрім дати.

Цілісність відношень забезпечується завдяки створенню Primarykey для кожної таблиці бази даних Salon, тобто атрибути відношень повністю залежать від первинного ключа.

Цілісність за посиланням реалізовано за допомогою ForeignKey, що визначає допустимі значення для певних атрибутів. Для прикладу на рисунку 2.7 Категорія продукту у таблиці Products може бути обрана лише з тих, які представлені у таблиці ProductCategory. Це запобігає внесенню у базу некоректних даних.

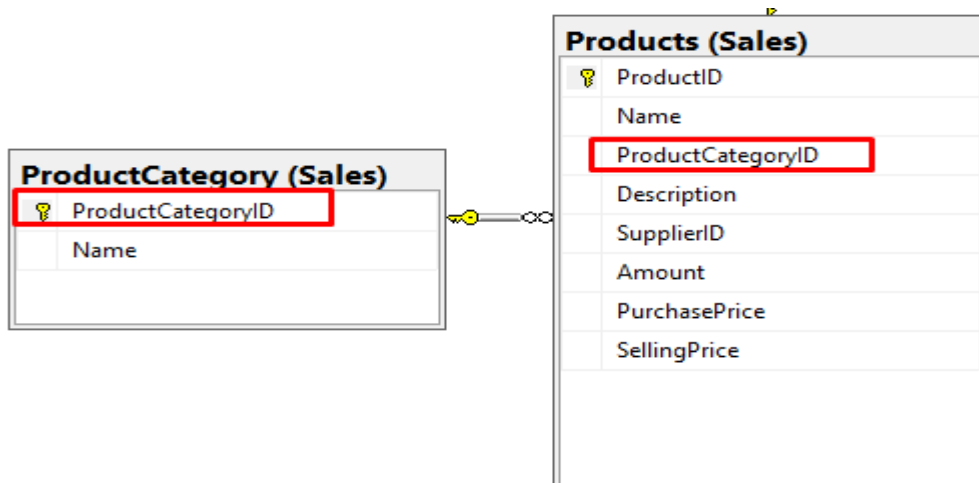


Рисунок 2.7 – Цілісність за посиланням

Також на поля таблиць накладається визначена користувачем цілісність даних. Вона реалізована з врахуванням бізнес-правил салону краси. Наприклад користувач не може продати більшу кількість товару, ніж є збережено у базі. Визначена користувачем цілісність даних програмно реалізована за допомогою збережених процедур та тригерів.

```
CREATE PROCEDURE Sales.STP_UpdateSupplier
(
  @SupplierID INT,
  @Name VARCHAR(20) = NULL,
  @Phone CHAR(17) = NULL
)
AS
```

```

BEGIN
    DECLARE @ErrorMessage VARCHAR(1000),
            @UserName SYSNAME = SYSTEM_USER,
            @ServerName NVARCHAR(50) = @@SERVERNAME,
            @DatabaseName NVARCHAR(50) = DB_NAME(),
            @ProcessName NVARCHAR(50) = OBJECT_SCHEMA_NAME(@@PROCID) + '.' +
OBJECT_NAME(@@PROCID),
            @StartTime DATETIME = GETDATE(),
            @EndTime DATETIME,
            @Status NVARCHAR(20) = 'SUCCESS',
            @Message NVARCHAR (MAX) = NULL,
            @TransactionID BIGINT = (SELECT TOP(1) transaction_id FROM
sys.dm_tran_current_transaction)
    BEGIN TRANSACTION
    BEGIN TRY

        IF SupplierID NOT IN (SELECT SupplierID FROM Sales.Suppliers)

        BEGIN
            SET @ErrorMessage = CONCAT('Supplier with ID ', @SupplierID, '
does not exist')
            RAISERROR(@ErrorMessage,16,1)
        END

        ELSE IF @Phone IN (SELECT Phone FROM Sales.Suppliers WHERE Phone =
@Phone)

        BEGIN
            SET @ErrorMessage = CONCAT('Supplier with Phone', @Phone, '
already exists')
            RAISERROR(@ErrorMessage,16,1)
        END

        ELSE IF @Name IN (SELECT Name FROM Sales.Suppliers WHERE Name = @Name)

        BEGIN
            SET @ErrorMessage = CONCAT('Supplier with Name', @Name, '
already exists')
            RAISERROR(@ErrorMessage,16,1)
        END

        UPDATE Sales.Suppliers
        SET
            Name = COALESCE (@Name, Name),
            Phone = COALESCE (@Phone, Phone),
        WHERE SupplierID = @SupplierID

    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION
        PRINT @ErrorMessage
        SET @Status = 'FAIL'
        SET @Message = @ErrorMessage
    END CATCH
    SET @EndTime = GETDATE()
    EXEC Administration.STP_InsertLogs @UserName,
        @ServerName,
        @DatabaseName,
        @ProcessName,
        @StartTime,
        @EndTime,
        @Status,
        @Message,
        @TransactionID
    IF @@TRANCOUNT > 0
    COMMIT TRANSACTION
END

```


2.3 Тестування

Веб-тестування дозволяє перевірити веб-додаток на потенційні помилки до моменту передачі замовнику.

Під час тестування перевірено наступне: безпеку веб-додатку, функціонування сайту, доступ до даних, а також дизайн та функціональність інтерфейсу користувача – що є головними моментами для замовника.

Було проведено наступні види тестування:

- 1) Тестування зручності використання та інтерфейсу
- 2) Тестування функціональності
- 3) Тестування БД

Тестування зручності використання та інтерфейсу користувача (Див. таблицю 2.2) — це не що інше, як перевірка навігації по системі (меню, кнопки, посилання на інші сторінки) та вмісту. Вміст має бути чітким, без орфографічних та граматичних помилок.

Таблиця 2.2 – Тестування зручності використання та інтерфейсу

№ п/п	Темтовий випадок	Очікуваний результат	Фактичний результат	Пройдено / Не пройдено
1	Перевірка відображуваних даних на наявність орфографічних помилок	Вміст веб-сторінки має бути правильним та без будь-яких орфографічних та граматичних помилок	Наявність орфографічних помилок не виявлено	Пройдено
2	Перевірка простоти навігації	Навігація для кожної операції повинна бути простою і легкою	Навігація для кожної операції проста і легка	Пройдено
3	Перевірка відповідності шрифтів згідно вимог	Усі шрифти повинні відповідати вимогам	Усі шрифти відповідають вимогам	Пройдено
4	Перевірка вирівнювання полів	Усі поля повинні бути належним чином вирівняні	Знайдено не вирівняні належним чином поля	Не пройдено
5	Перевірка відображення повідомлень про успішну чи не успішну операцію	Повідомлення про підтвердження має відображатися для будь-якої операції оновлення та видалення	Повідомлення про підтвердження відображається для будь-якої операції додавання оновлення	Пройдено

			та видалення	
6	Перевірка сайту на масштабованість	Масштабованість повинна змінюватись із збереженням пропорцій	Знайдено елемент із фіксованим розміром	Не пройдено
7	Перевірка відображення даних випадючих списків	Значення випадючих списків повинні відображатися у визначеному порядку сортування	Значення випадючих списків відображаються у визначеному порядку сортування	Пройдено
8	Перевірка на наявність інтервалів між мітками полів, стовпцями, рядками, повідомленнями про помилки	Повинно бути виділено достатньо місця між мітками полів, стовпцями, рядками, повідомленнями про помилки	Виділено достатньо місця між мітками полів, стовпцями, рядками, повідомленнями про помилки	Пройдено

2) Функціональне тестування (Див. таблицю 2.3) перевіряє, що кожна функція розроблюваної програми працює у відповідності зі специфікацією вимог.

Таблиця 2.3– Функціональне тестування

№ п/п	Тестовий випадок	Очікуваний результат	Фактичний результат	Пройдено / Не пройдено
1	Перевірка елементів на кожній сторінці	Функціонування перевірених елементів мають відповідають вимогам	Функціонування перевірених елементів відповідають вимогам	Пройдено
2	Перевірка елементів, об'єктів, полів усіх наявних форм	Всі елементи, об'єкти, поля мають функціонувати згідно вимог	Всі елементи, об'єкти, поля функціонують згідно вимог	Пройдено
3	Перевірка виведення запиту для підтвердження видалення запису	При видалення запису має виводитись повідомлення для підтвердження операції	Повідомлення не виводиться	Не пройдено
4	Перевірка функціональності входу	Має відбутися вхід у систему після введення логіну та пароля	Відбувається вхід у систему після введення логіну та пароля	Пройдено
5	Перевірка функції блокування входу	При неправильному введенні логіну і паролю, не має допускатись вхід у систему	При неправильно введенні логіну і паролю, не допускається вхід у систему	Пройдено

3) Тестування бази даних (Див. таблицю 2.4) — є складовою функціонального тестування, в якому виконується перевірка даних, маніпуляції з даними (додавання, оновлення та видалення), цілісності даних.

Таблиця 2.4 –Тестування бази даних

№ п/п	Тестовий випадок	Очікуваний результат	Фактичний результат	Пройдено / Не пройдено
1	Перевірка правильності зберігання даних в базі даних після успішного підтвердження введення на формі	Після підтвердження введення на формі, у БД мають зберігатися коректні дані	у БД зберігаються коректні дані	Пройдено
2	Перевірка значення для стовпців, які не приймають значення null	Не має відбуватись збереження даних	Неможливість збереження даних	Пройдено
3	Перевірка цілісності даних	Дані повинні зберігатися в одній або декількох таблицях відповідно до дизайну	Дані відповідно зберігаються в одній або декількох таблицях відповідно до дизайну	Пройдено
4	Перевірка наявності первинного ключа	Таблиці повинні мати атрибут первинного ключа	Таблиці мають атрибут первинного ключа	Пройдено
5	Перевірка виконання транзакцій	У разі невдалої транзакції зміни будуть відхилені	У разі невдалої транзакції зміни не відбулися	Пройдено
6	Перевірка зашифрованих даних в БД	Не має бути змоги перегляду зашифрованих даних	Неможливо переглянути зашифровані дані	Пройдено

Розгортання програмного забезпечення включає в себе весь процес, необхідний для встановлення та роботи в певному середовищі. Воно передбачає встановлення, налаштування, тестування та внесення змін для оптимізації продуктивності програмного забезпечення. Розгортання програмної системи здійснюється наступним чином.

Щоб відкрити додаток потрібно встановити веб-сервер Apache Tomcat. Оскільки Tomcat написаний на Java перед цим встановлюємо JDK. Tomcat працює з JDK 1.2 і вище. Веб-сервер можна завантажити з сайту Apache Tomcat (Див. рисунок 2.8). Налаштовуємо JAVA_HOME або і CATALINA_HOME, щоб

можна було виконувати деякі операції з обслуговування безпосередньо на сервері Tomcat, наприклад Starting and Stopping сервера.

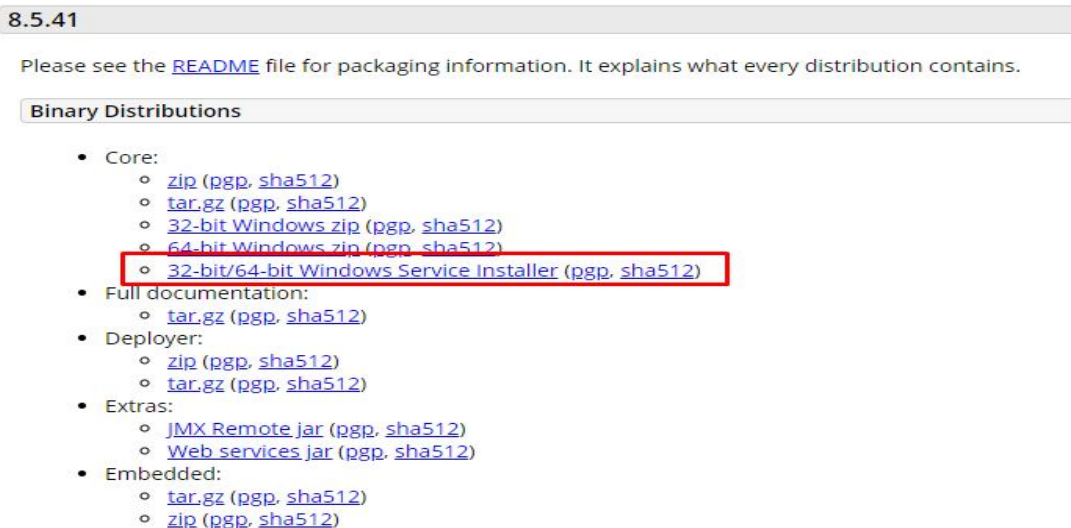


Рисунок 2.8 – ApacheTomcatInstaller

Далі потрібно експортувати створений у EclipseIDEпроект у WARфайл, назвемо його Salon.war (Див. рисунок 2.9). WAR (Web ARchive) є стандартним форматом контейнерних файлів для упаковки додатків Java EE, який може бути розгорнутий на Java EE-сумісних серверах додатків, в нашому випадку Tomcat.

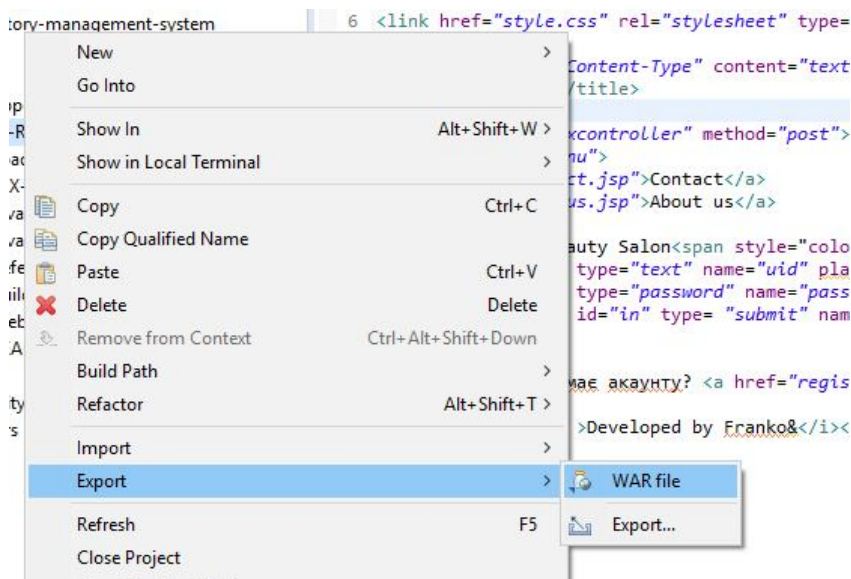


Рисунок 2.9 – Експорт проекту у WARфайл

Для роботи з базою даних також потрібно встановити SQLServer 2017 Developer edition. Зробити це можна, перейшовши за посиланням <https://www.microsoft.com/en-au/sql-server/sql-server-downloads>.



Рисунок 2.10 – Скачування SQLServer 2017

Після установки всіх необхідних компонентів системи можемо приступати до розгортання середовища. Спочатку з бекапу бази даних Salon.bak робимо restore у нашу продакшн базу. Наступним кроком створюємо bat-файл (Див. рисунок 2.11), який відповідає за запуск веб сервера та відкриває програму у браузері.



Рисунок 2.11 – Batфайл для запуску програми.

В результаті виконання попередніх кроків програмний продукт запускається у браузері і є готовим до взаємодії з користувачем. На рисунку

2.12 відображено вікно авторизації. Для того, щоб почати роботу у системі потрібно залогуватись з отриманими логіном та паролем.

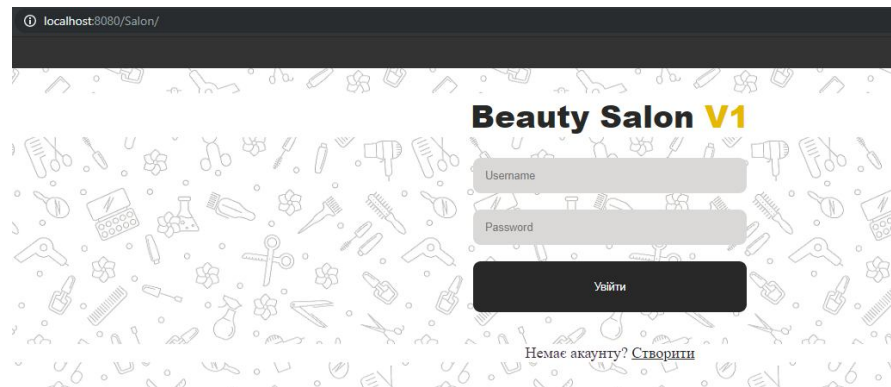


Рисунок 2.12 – Вікно авторизації програми

Пароль для входу в систему можна змінити у будь-який час, ввівши правильно старий пароль.

2.4. Інструкція користувача

1. Компоненти ПЗ

Програмна система складається з наступних компонентів:

- веб-браузер (клієнт) – презентаційний рівень програми корисний для кінцевого користувача – Chrome
- веб-сервер – його завдання встановити з'єднання між сервером і браузером користувача. Він відповідає за обмін даними в обидві сторони (клієнт-серверна архітектура) – Apache Tomcat 9.0
- сервер баз даних – забезпечує можливість управління базою даних – MS SQL Server 2017

2. Встановлення ПЗ :

Для роботи програмної системи повинно бути встановлене наступне ПЗ:

- Браузер Chrome
- Apache Tomcat 9
- MS SQL Server 2017
- JVM 10.0.2 і вище

3. Базові функції ПЗ

Для авторизації користувача в системі необхідно ввести логін та пароль у відповідні поля та натиснути кнопку «Увійти» (Див. рисунок 2.13). Для тестування створено користувача з логіном та паролем admin.

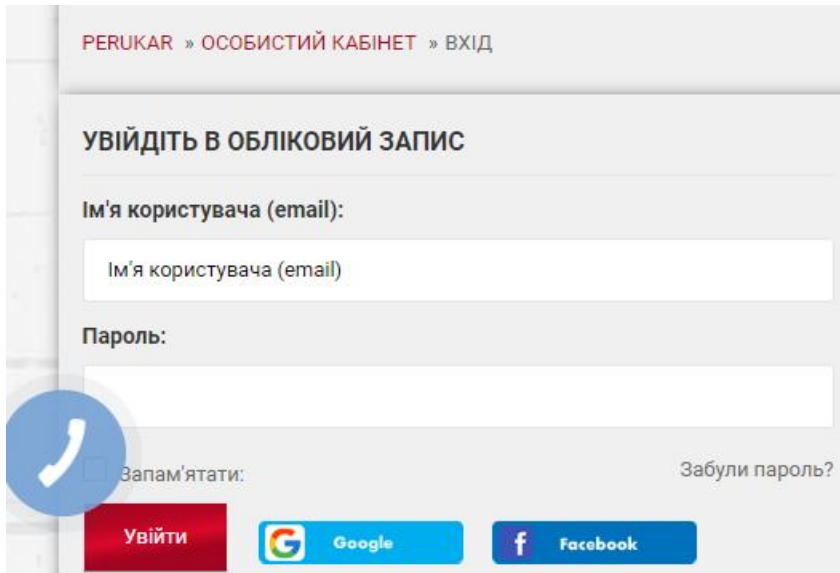


Рисунок 2.13 – Введення логіну та паролю

Якщо логін або пароль введено невірно користувач отримає відповідне повідомлення (Див. рисунок 2.14). В такому разі він знову повернеться на форму авторизації для повторної авторизації.

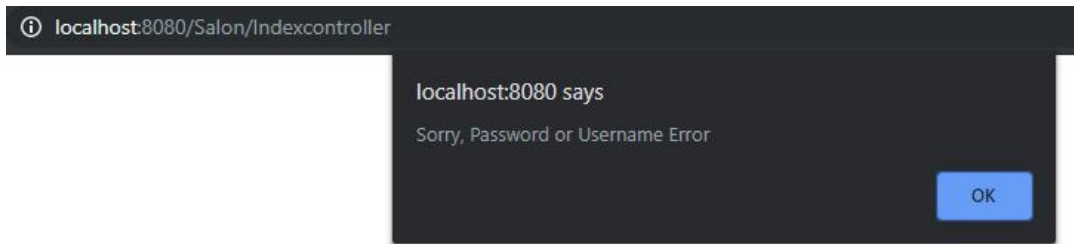


Рисунок 2.14 – Помилка при неправильному введенні логіна або пароля

При необхідності змінити пароль потрібно натиснути на кнопку Змінити пароль. Далі на формі потрібно ввести старий пароль та новий пароль (Див. рисунок 2.15). Якщо старий пароль введено правильно і новий пароль пройшов перевірку на складність після натиснення кнопки Підтвердити зміни будуть збережені.

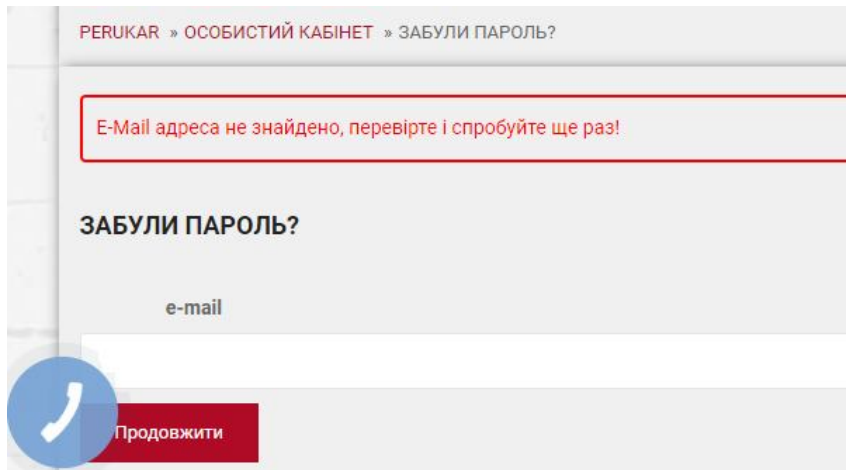


Рисунок 2.15 – Зміна паролю

Для того, щоб додати новий товар в систему потрібно виконати наступні кроки:

1. Переходимо на вкладку «Товари»
2. Натискаємо кнопку «Додати»
3. Заповнюємо відповідні поля у формі для додавання нового товару.
4. Натискаємо кнопку «Зберегти»

Для того, щоб продати товар систему потрібно виконати наступні кроки:

1. Переходимо на вкладку «Продаж»
2. Натискаємо кнопку «Продати»
3. Скануємо штрих-код товару/ обираємо категорію і бажаний товар.
4. Обираємо кількість одиниць.
5. Натискаємо кнопку «Підтвердити»

2.5 Висновок до другого розділу

Описано програмну реалізацію проекту. Для розробки проекту використано мову програмування Java з технологією JSP та середовище розробки Eclipse Neon 2 IDE. В якості СУБД обрано MS SQL Server 2017. Оскільки продукт є кросплатформним рішенням, його використання не прив'язується до конкретної операційної системи

Після процесу проектування бази даних, створено базу даних у СУБД MS SQL за допомогою DDL скрипта та внесені усі необхідні дані. Дане сховище забезпечує доступ до даних за допомогою програмної системи, а також збереження даних.

Обґрунтовано вибір засобів для створення таблиць та формування основних SQL запитів, що відповідають вимогам клієнта.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Методи підвищення мотивації безпеки праці

Потрібно знайти такі способи впливу на людей, щоб вони усвідомили необхідність працювати безпечно, створити такі «правила гри», в межах яких людині було б вигідно дотримуватися встановлених норм.

Безпечна поведінка на виробництві залежить не тільки від професійних знань, навичок і здібностей, а й значною мірою від мотивів поведінки працівника. Відповідно управляти діями людини можна тільки за допомогою управління її мотивами.

В обмін за свій труд працівники очікують не тільки високої оплати, а й створення умов для особистісного росту, отримання задоволення від власної роботи, інших компенсацій, які адекватні професійному рівню та відповідають особистим інтересам. Ефективна праця допомагає швидшому розвиванню підприємства.

Для заохочення працівників потрібно підбадьорення та підтримка з сторони начальства. Стимулювати ефективну роботу можуть матеріальні методи, наприклад премії, винагороди, безкоштовне харчування, додатковий дохід та інше. Мотиваційний комплекс взагалі й безпечної поведінки людини зокрема носить полімотивований характер, містить у собі широкий спектр мотиваційних регуляторів як матеріального, так і нематеріального характеру та має певну ієрархічність. На особистісному рівні працювати продуктивно та безпечно вигідно самій людині; від цього залежить успіх роботи її підрозділу; і нарешті - це потрібно підприємству (компанії). Тобто для вирішення охоронних проблем у праці потрібно зацікавити працівників трудитися безпечно не тільки для себе, а й для оточуючих.

Практично будь-якого працівника можна зацікавити будь-яким мотивом, оскільки абсолютно ні на що не мотивованих людей немає! Очевидно, тільки

закликами, зверненнями, деклараціями, пропагандою ці проблеми навряд чи вдасться вирішити.

Потрібно знайти такі способи впливу на людей, щоб вони усвідомили необхідність працювати безпечно, створити такі «правила гри», в межах яких людині було б вигідно дотримуватися встановлених регламентів. І цей вплив вона повинна відчувати безпосередньо в процесі всієї трудової діяльності.

Проаналізувавши загальні методи мотивації для підвищення роботи працівників можемо охарактеризувати методи які потрібні для підвищення мотивації безпеки праці осіб на різних підприємствах. Найголовнішим чинником для будь-якого підприємства має бути на першому місці створення безпечних умов праці та дотримання всіх необхідних безпечних заходів для своїх працівників. Важливим показником охорони праці на підприємстві є внутрішнє стимулювання для безпечного ведення робіт. При можливості на об'єкті можуть працівники долучитись до охорони праці та запропонувати свої варіанти, підвівши підсумки можуть скласти договір. У цій угоді, яку склали колективно можуть вказати свої матеріальні та нематеріальні вимоги. При цьому не може бути системи стимулювання, яка мотивує всіх співробітників однаково.

Система стимулів має бути персоніфікованою, ретельно дозованою та розроблятися для кожної людини або певної групи людей з подібними домінуючими потребами, або загальна система має індивідуалізуватися. Тому моніторинг домінуючих потреб персоналу - необхідна умова функціонування.

3.2 Забезпечення захисту працівників суб'єкта господарювання від іонізуючих випромінювань

Іонізуюче випромінювання або радіоактивність є небезпечним явищем для людського організму. При взаємодії впливу іонізаційних випромінювань у навколишнє середовище можуть відбутись різні утворення зарядів . Існують два різновиди випромінювання – «альфа» та «бета».

В залежності від носія та енергії, вони мають різну проникаючу здатність. Альфа це випромінювання яке проявляється важкими частинами складеними з протонів і нейтронів.

В свою чергу бета випромінювання являє собою ланцюг електронів та позитронів які є більшу здатність проникати у середовище. Працюючи на таких територіях, де існує радіаційна атмосфера можуть виникнути різні випадки.

На підприємстві можуть виникнути інциденти при користуванні ядерними матеріалами , зберіганні радіоактивних відходів в наслідок чого працівники можуть отримати травму у вигляді дози опромінення, використання іонізуючих джерел випромінювання.

Також у випадку такої радіаційної аварії забруднюється навколишнє середовище, люди можуть отримати травму у вигляді потужної дози опромінення. Призвести аварію на підприємстві може також якщо активна реакційна речовина знаходиться у роботі та це відбувається незаконно.

Це може привезти до опромінення жителів та перевищити межу дози опромінення. Частинки з цього випромінювання можуть залишати сліди на дихальній системі на травній системі людського організму. Також ці елементи можуть бути у водних каналах, які постачають питну воду людям.

На підприємстві де проводяться роботи з радіаційними речовинами обов'язково мають вживатись заходи проти радіації. Протирадіаційні захисти це така система правових, організаційних норм та санітарної гігієни.

До переліку таких захистів можна включити медичні заходи для забезпечення радіаційної безпеки персоналу та проектно-конструкторські. Для організації заходів проти іонізації опромінювання підприємство має ввести обов'язкові методи щоб подбати про безпеку працюючого персоналу. До таких методів можуть належати заходи які обмежують допуск працівників до джерел які випромінюють радіацію.

До таких працівників можемо віднести таких, які не підходять за віком, за статтю та працівники які вже отримали дозу випромінювання. Підприємство мусить створити сприятливі умови що дотримуються встановлених норм та

вимог для працівників та застосовувати індивідуальні засоби для захисту працівника цього підприємства.

Організація повинна контролювати рівні опромінювання та вести інформаційну систему про стан радіації на підприємстві та призначених місць для праці.

На підприємстві повинні бути проведені заходи щодо організації безпеки для робіт які проводяться у радіаційних ділянках а саме:

- організація роботи нарядів та розпоряджень;
- організація та перевірка пропусків до робочих місць;
- оформлення контролю за процесом виконання роботи;
- введення примусового часу на перерву та вчасне закінчення робочого процесу.

До фізичних норм захисту проти радіації існують перешкоди поширення іонізації опроміненень. Для поширення дози випромінювання може бути ряд перешкод, залежать вони від кількості годин, перешкоджати може дистанція , також перешкодою може бути чисельність.

Реалізувати заходи проти радіації за певний відрізок часу можливо, тим що працівники , які працюють з іонізованими випромінюваннями можуть виконувати вчасно свою роботу ,відповідно керівництво може за якісну роботу зменшити кількість робочих днів у тижні.

Цим самим вони застереженням вони зменшать знаходження працівників у зоні випромінювання та відповідно буде менше контактування з радіаційними приладами. Захистити працівників за допомогою відстані підприємство може шляхом доцільного розміщення приміщення, правильно розставити та розрахувати робочі місця для працівників а також забезпечити приладами, які зможуть контактувати, керувати робочим процесом з технікою яка має радіаційний вплив на відстані.

Слугувати захистом може покриття свинцем меблів які присутні у приміщенні (двері, вікна, робочі столи), створення перекриття між поверхами

та перегородки. Працівникам обов'язково має бути виданий спеціальний одяг, такі як фартухи, шапочки та рукавиці зшиті з просвинцевої тканини.

Розміщення робочих місць повинно мати правильний розрахунок на загальну кімнату, не робити перенабір та забезпечити відповідним та необхідним обладнанням робочі кабінети. При користуванні відкритими приладами іонізованого опромінення провести герметизації цих систем, при можливості використовувати роботу техніки. Підприємство повинне вжити усіх санітарно-гігієнічних заходів та соціальних, а також важливо необхідний є медичний захист робочих на об'єкті.

3.3 Висновок до третього розділу

В даному розділі описано заходи та методи із забезпечення радіаційних впливів та іонізації опромінювання на підприємствах. Описані вимоги для керівництва та підлеглих працюючих на об'єктах щодо їхніх дій в разі виникнення радіації.

Також описані вимоги для мотивації робітників щодо дотримання правил охорони праці на підприємстві. Мотивація - одна з центральних функцій управління як персоналом, так і охороною праці. Вона може відігравати важливу роль як фактор спонукання персоналу діяти адекватним способом у власних і корпоративних інтересах.

Для цього потрібно, щоб мета підприємства збігалася з метою працівників. Однак мотивація одночасно є не тільки рушійним механізмом, а й фактором залучення, наприклад, до охорони праці, високопрофесійних спеціалістів. Це механізм, що спонукає вдосконалювати систему управління. Крім того, рівень мотивації працівників відіграє важливу роль у загальному успіху підприємства (компанії).

ВИСНОВКИ

Проведено детальний аналіз предметної області, висвітлено проблему, яка виникла у салоні краси при веденні бізнесу та наведено шляхи її вирішення. Розглянуто напрямки діяльності вище зазначеного салону. Продемонстровано схему організаційної структури управління салоном. Визначено бізнес-процеси, які пов'язані з проблемою обліку товарів та товарних операцій адміністратором салону.

Розглянуто аналоги схожих програмних продуктів, наведені їх основні функції, висвітлено позитивні та негативні сторони. Також розроблено специфікацію вимог для створюваної програмної системи, враховуючи аналіз існуючих аналогів. Наведено діаграму варіантів використання.

Здійснено реалізацію програмної системи для обліку товарів і товарних операцій перукарського салону тестуванню. Під час розробки програмного продукту було проведено ряд тестувань, а саме функціональне, бази даних та користувацького інтерфейсу. Для опису процедур тестування було складено тест-кейси.

Наведено опис апаратних та програмних засобів, які необхідні для функціонування програмного продукту та описано інструкцію для інсталяції на комп'ютері користувача. Для забезпечення ефективного використання розробленого програмного продукту створено інструкцію користувача, яка містить:

1. Компоненти ПЗ
2. Встановлення ПЗ
3. Базові функції ПЗ

ПЕРЕЛІК ДЖЕРЕЛ

1. Web Application benefits [Електронний ресурс]. – Режим доступу до ресурсу: <https://searchsoftwarequality.techtarget.com/definition/Web-application-Webapp>
2. Дрожнікова Ю.О. Автоматизоване тестування мобільного додатку з Indoor-навігацією та навігацією з використанням доповненої реальності AR всередині торгових центрів на операційній системі IOS / Ю.О. Дрожнікова // XLIX Науково-технічна конференція факультету комп'ютерних систем і автоматики (2020)Вінницького національного технічного університету, – Вінниця : ВНТУ, 2020.
3. Конспект лекцій з дисципліни «Програмування для мобільних пристроїв» для студентів денної форми навчання спеціальності 126 «Інформаційні системи та технології» / Укладачі: Готович В.А., Михайлович Т.В. – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2020. – 216 с.
4. DIKIDI [Електронний ресурс]. – Режим доступу до ресурсу: <https://play.google.com/store/apps/details?id=ru.dikidi&hl=uk>– 14.04.2020р.
5. DIKIDI [Електронний ресурс]. – Режим доступу до ресурсу: <https://beauty.dikidi.ru/uk/> – 13.04.2020р. – Назва з титулу екрана.
6. Велюр [Електронний ресурс]. – Режим доступу до ресурсу: <http://velurspa.com.ua/> – 21.04.2020р. – Назва з титулу екрана
7. MySQL [Електронний ресурс]. – Режим доступу до ресурсу: <https://metanit.com/sql/mysql/1.1.php>
8. Відгуки та презентації інструментів розробки програмного забезпечення [Електронний ресурс]. Режим доступу: <http://www.methodsandtools.com/tools/tools.php>. – Назва з екрану.
9. HTML [Електронний ресурс]. – Режим доступу до ресурсу: <https://metanit.com/web/html5/1.1.php>

10. Коли тестування повинно бути автоматизованим? [Електронний ресурс]. Режим доступу: <https://www.stickyminds.com/article/when-should-testbe-automated>– Назва з екрану.

11. JavaScript [Електронний ресурс]. – Режим доступу до ресурсу: https://developer.mozilla.org/ru/docs/Web/JavaScript/%D0%9E_JavaScript

12. Контекстна діаграма [Електронний ресурс]. – Режим доступу до ресурсу: <https://nsu.ru/smk/files/idef.pdf> 57

13. Understanding and Working With Data in WordPress [Електронний ресурс]. – Режим доступу до ресурсу: <https://code.tutsplus.com/tutorials/understanding-andworking-with-data-in-wordpress--cms-20567>

14. Коноваленко І. В. Платформа .NET та мова програмування С# 8.0 : навчальний посібник / І. В. Коноваленко, П. О. Марущак. – Тернопіль : ФОП Паляниця В. А., 2020. – 320 с.

15. Комп'ютерні мережі: [навчальний посібник] / А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, В. В. Пасічник. — Львів: «Магнолія 2006», 2013. — 256 с. ISBN 978-617-574-087-3

16. Петрик М.Р. Моделювання програмного забезпечення : науково методичний посібник / М.Р. Петрик, О.Ю. Петрик – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 200 с.

17. APACHE [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.atlantic.net/what-is-an-apache-server/>

18. « \$_SERVER['HTTP_REFERER'] [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.php.net/manual/en/reserved.variables.server.php>

19. mysqli_query [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.php.net/manual/en/mysqli.query.php>

20. mysqli_fetch_array [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.php.net/manual/en/function.mysql-fetch-array.php>