

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ
ДЛЯ КЕРУВАННЯ БІЗНЕС-ПРОЦЕСАМИ У САЛОНІ КРАСИ
RESEARCH AND SOFTWARE IMPLEMENTATION OF A WEB APPLICATION
FOR BUSINESS PROCESS MANAGEMENT IN A BEAUTY SALON

Виконав: студент(ка) 2 курсу, групи КНТ-142м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ПАРХОМЕНКО В.В.

(ПРИЗВИЩЕ та ініціали)

Керівник ОЛІЙНИК А.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

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

Інститут, факультет ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
« » 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ПАРХОМЕНКО Вікторії Валеріївни

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація вебзастосунку для керування бізнес-процесами у салоні краси. Research and Software Implementation of a Web Application for Business Process Management in a Beauty Salon

керівник проєкту (роботи) д.т.н., професор, ОЛІЙНИК Андрій Олександрович,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від «14» листопада 2023 року № 437

2. Строк подання студентом проєкту (роботи) 01 грудня 2023 року

3. Вихідні дані до проєкту (роботи) технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Дослідження клієнт-серверних рішень для керування салоном краси. 2. Засоби розробки програмного продукту. 3. Проєктування програмного продукту. 4. Конструювання програмного продукту. 5. Експлуатація та тестування розробленого програмного продукту.

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

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	ОЛІЙНИК А.О., професор		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання «04» вересня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Дослідження предметної області та рішень вирішення завдання.	1 тиждень	Розділ 1
3	Вибір інструментів для розробки програмного продукту.	2 тиждень	Розділ 2
4	Моделювання вимог та проектування архітектури до програмного продукту.	3 тиждень	Розділ 3
5	Розробка програмного продукту.	4-7 тижні	Розділ 4
6	Тестування та експериментальне дослідження розробленого програмного продукту.	8 тиждень	Розділ 5
7	Оформлення пояснювальної записки та документів до неї.	9-10 тижні	Додатки
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

(підпис) Вікторія ПАРХОМЕНКО
(Ім'я ПРІЗВИЩЕ)

Керівник проекту (роботи)

(підпис) Андрій ОЛІЙНИК
(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
136 с., 7 табл., 46 рис., 3 дод., 77 джерел.

БІЗНЕС-ПРОЦЕС, ВЕБЗАСТОСУНОК, КЛІЄНТ, ПОСЛУГА, САЛОН
КРАСИ, СФЕРА ПОСЛУГ КРАСИ, CRM, MERN, MONGODB.

Об'єкт дослідження – бізнес-процеси у сфері надання послуг краси.

Предмет дослідження – вебзастосунок для керування бізнес-процесами у салоні краси.

Мета роботи – дослідження і розробка вебзастосунку, що надає функціональні можливості для керування бізнес-процесами у салоні краси.

Матеріали, методи та технічні засоби: текстовий редактор Visual Studio Code, функціональне програмування, мова розмітки вебдокументів HTML, мова опису стилів оформлення елементів вебдокументів CSS, CSS препроцесор Sass, мова програмування JS, бібліотека елементів інтерфейсу користувача React, бібліотека компонентів інтерфейсу користувача Material UI, бібліотека маршрутизації вебсторінок React Router, JS бібліотека керування станами застосунку Redux, серверна платформа розробки Node.js, серверний фреймворк Express.js, нереляційна СКБД MongoDB, бібліотека роботи з базою даних Mongoose та ноутбук під керуванням операційної системи Microsoft Windows 10 з процесором Intel Core i5-11320H.

Результати. Проведено дослідження програмних рішень, які надають аналогічні функції. Розроблено вебзастосунок, який надає функції керування бізнес-процесами у салоні краси та підвищує ефективність роботи з обслуговування клієнтів для керівників та співробітників.

Висновки. Було виконано проєктування, конструювання та тестування розробленого вебзастосунку для керування бізнес-процесами у салоні краси.

Галузь використання – компанії сфери послуг краси.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 136 pages, 46 figures, 7 tables, 3 appendixes, 77 sources.

BUSINESS PROCESS, WEB APPLICATION, CLIENT, SERVICE, BEAUTY SALON, BEAUTY SERVICES FIELD, CRM, MERN, MONGODB.

The object of research is business processes in the field of beauty services.

The subject of the research is a web application for managing business processes in a beauty salon.

The purpose of the work is to research and develop a web application that provides functionality for managing business processes in a beauty salon.

Materials, methods and tools: Visual Studio Code text editor, functional programming, HTML web document markup language, CSS web document element style description language, Sass CSS preprocessor, JS programming language, React UI elements library, Material UI components library, React Router web page routing library, Redux application state management JS library, Node.js server-side development platform, Express.js server framework, MongoDB non-relational DBMS, Mongoose database library, and a laptop running Microsoft Windows 10 with an Intel Core i5-11320H processor.

Results. A study of software solutions that provide similar functions was conducted. A web application has been developed that provides functions for managing business processes in a beauty salon and increases the efficiency of customer service work for managers and employees.

Conclusions. The design, construction and testing of the developed web application for managing business processes in the beauty salon was performed.

The field of use is companies in the field of beauty services.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	10
1 Дослідження клієнт-серверних рішень для керування салоном краси.....	12
1.1 Аналіз та особливості предметної області.....	12
1.2 Бізнес-процеси у салоні краси.....	21
1.3 Огляд клієнт-серверних застосунків для салонів краси.....	23
1.3.1 Вебплатформа «CleverBOX:CRM».....	23
1.3.2 Вебзастосунок «Wlaunch».....	34
1.3.3 Вебзастосунок «RemOnline».....	39
1.3.4 Порівняння досліджених вебзастосунків.....	43
1.4 Постановка завдань дипломної магістерської роботи.....	45
1.5 Висновки до першого розділу.....	46
2 Засоби розробки програмного продукту.....	47
2.1 Вибір клієнтської та серверної мови програмування.....	47
2.2 Вибір середовища розробки вебзастосунку.....	49
2.3 Вибір засобів розробки користувацької частини вебзастосунку.....	52
2.3.1 Мова розмітки вебдокументів HTML.....	52
2.3.2 Препроцесор Sass.....	52
2.3.3 Бібліотека елементів інтерфейсу користувача React.....	53
2.3.4 Бібліотека компонентів Material UI.....	57
2.3.5 Бібліотека маршрутизації вебсторінок React Router.....	57
2.3.6 Менеджер керування станами Redux.....	58
2.4 Вибір засобів розробки серверної частини вебзастосунку.....	59
2.4.1 Платформа розробки Node.js.....	59
2.4.2 Серверний фреймворк Express.js.....	61
2.5 Вибір СКБД.....	62
2.6 Вибір бібліотеки для роботи з обраною СКБД.....	66
2.7 Висновки до другого розділу.....	66

	7
3 Проєктування програмного продукту	67
3.1 Проєктування сценаріїв використання вебзастосунку	67
3.2 Проєктування структури навігації вебзастосунку	76
3.3 Проєктування архітектури вебзастосунку	77
3.4 Висновки до третього розділу	79
4 Конструювання програмного продукту	80
4.1 Фізична модель реалізованої бази даних	80
4.2 Алгоритм функціонування розробленого вебзастосунку	81
4.3 Розроблені модулі вебзастосунку	82
4.4 Опис розроблених функцій	83
4.5 Розроблена структура проєкту вебзастосунку	84
4.6 Висновки до четвертого розділу	85
5 Експлуатація та тестування розробленого програмного продукту	86
5.1 Опис призначення програмного продукту	86
5.2 Вимоги до програмно-технічних засобів	89
5.3 Характеристики програмного продукту	90
5.4 Виконання програмного продукту	91
5.5 Опис повідомлень користувачу	93
5.6 Опис тестування функцій розроблених модулів	94
5.7 Висновки до п'ятого розділу	96
Висновки	97
Перелік джерел посилання	98
Додаток А Технічне завдання	105
Додаток Б Текст програми	116
Додаток В Слайди презентації	126

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ACID	– Atomicity, Consistency, Isolation, Durability;
API	– Application Programming Interface;
BASE	– Basic availability, Soft state, Eventual consistency;
BPMN	– Business Process Model and Notation;
BSON	– Binary JavaScript Object Notation;
CAP	– Consistency, Availability, Partition tolerance;
CRM	– Customer Relationship Management;
CRUD	– Create, Read, Update, Delete;
CSS	– Cascading Style Sheets;
DBMS	– Database Management System;
DOM	– Document Object Model;
EPC	– Event-Driven Process Chain;
HTML	– HyperText Markup Language;
HTTP	– HyperText Transfer Protocol;
IDEF0	– Integration Definition for Function Modeling;
JSON	– JavaScript Object Notation;
JS	– JavaScript;
JSX	– JavaScript Syntax Extension;
MERN	– MongoDB, Express.js, React, Node.js;
MQL	– MongoDB Query Language;
MVC	– Model, View, Controller;
NoSQL	– Not Only SQL;
NPM	– Node Package Manager;
NVM	– Node Version Manager;
ODM	– Object Data Modelling;
REST	– Representational State Transfer;
Sass	– Syntactically Awesome Stylesheets;
SPA	– Sanus per Aquam;

SQL – Structured Query Language;

UI – User Interface;

URL – Uniform Resource Locator;

СКБД – система керування базами даних.

ВСТУП

В епоху цифрової трансформації з'являються багато різноманітних технологій, які здатні підвищувати якість обслуговування клієнтів та надають інструменти автоматизації робочих процесів. Різноманітні технології роблять процеси більш прозорими та надають можливість здійснення операцій. Для бізнесу головними перевагами цифрової трансформації є скорочення часу на повторювані завдання завдяки автоматизації процесів та підвищення рівня конкурентоспроможності на ринку [1].

Сьогодні існує багато розроблених та удосконалених програмних рішень для керування бізнес-процесами у салоні краси, в яких враховані завдання та особливості галузі. На бізнес-ринку є безліч автоматизованих сервісів з різними функціональними характеристиками, а саме від запису клієнтів на послуги до обліку замовлень та нарахування заробітної плати [2].

Для ведення обліку різних бізнес-процесів у салоні краси використовуються системи управління відносинами з клієнтами або CRM. CRM активно використовується у всіх сферах бізнесу, а головним її принципом є висока клієнтоорієнтованість. Центром бізнесу у філософії створення CRM постає клієнт, а усі бізнес-процеси спрямовані на задоволення клієнтів та підвищення якості обслуговування. Зосередження саме на клієнті є дуже актуальним у сфері надання послуг краси [2].

CRM дозволяють вести облік послуг, замовлень, товарів, клієнтів, косметичних засобів, візитів клієнтів, оплат, а також контролювати залишки товарів на складах, фінансові операції та завантаження спеціалістів салону краси. За допомогою таких систем можна переглядати аналітичні звіти, які відображають ключові показники компанії та статистику записів [2].

Актуальність дипломної магістерської роботи обумовлена у тому, що кожний бізнес-процес включає велику кількість дій, які віднімають безліч робочого часу, і які суттєво збільшують робоче навантаження. В результаті стає важко проконтролювати кожний бізнес-процес та уникнути помилок в

обліку. Рішенням цієї проблеми є впровадження програмного рішення для керування бізнес-процесами у салоні краси, яке полегшить та автоматизує роботу співробітників та керівників салону краси.

Об'єктом дослідження є бізнес-процеси у сфері надання послуг краси.

Предметом дослідження є вебзастосунок для керування бізнес-процесами у салоні краси.

Метою роботи є дослідження і розробка вебзастосунку, що надає функціональні можливості для керування бізнес-процесами у салоні краси.

У дипломній роботі для досягнення поставленої мети роботи необхідно виконати перераховані нижче задачі магістерського дослідження:

- дослідження програмних рішень, які надають аналогічні функції;
- розробка технічного завдання;
- вибір засобів для розробки вебзастосунку;
- моделювання вимог до вебзастосунку для керування бізнес-процесами у салоні краси;
- проєктування структури навігації вебзастосунку для керування бізнес-процесами у салоні краси;
- проєктування архітектури вебзастосунку для керування бізнес-процесами у салоні краси;
- конструювання вебзастосунку для керування бізнес-процесами у салоні краси;
- тестування функціональності розробленого вебзастосунку.

1 ДОСЛІДЖЕННЯ КЛІЄНТ-СЕРВЕРНИХ РІШЕНЬ ДЛЯ КЕРУВАННЯ САЛОНОМ КРАСИ

У розділі проведено дослідження предметної області та існуючих програмних рішень для керування бізнес-процесами у салоні краси. Наведено порівняння досліджених аналогічних програмних рішень та сформовано завдання, які будуть вирішуватися при виконанні дипломної магістерської роботи.

1.1 Аналіз та особливості предметної області

Сфера послуг краси є популярною сферою заробітку для підприємців-початківців. Зацікавленість в естетичних та косметологічних сервісах нині продовжує зростати. Керівнику салону краси важливо обрати правильний підхід до керування таким видом бізнесу. Прибутковість салону краси залежить від унікальності закладу, правильного ведення бухгалтерського та фінансового обліку, обраних інструментів маркетингу та управлінських можливостей [3].

Салон краси – це заклад, який пропонує різноманітні косметичні процедури, такі як відновлення волосся, стрижка, укладка волосся, фарбування, манікюр та педикюр. Основною функцією кожного салону краси є обслуговування та надання якісних послуг клієнтам [4].

Салон краси є найпопулярнішим типом салонів краси та пропонує широкий спектр послуг для чоловіків та жінок. Існують різні типи салонів краси та кожний з таких закладів може надавати від базових послуг до послуг, які існують в декількох різновидах. Від типу салону краси залежить його прибуток, місцезнаходження, способи просування та кількість закупок товарів необхідних для надання послуг, а також кількість власних товарів для продажу клієнтам [4].

Розрізняють наступні класифікації салонів краси:

- за масштабом бізнесу;
- за кількістю послуг;
- за ціновою категорією [4].

За масштабом бізнесу виділяють наступні типи салонів краси:

- кабінет;
- салон краси;
- медичний центр краси та здоров'я;
- мережа підприємств краси [4].

Кабінет є видом малого бізнесу, який надає базові послуги та знаходиться на орендованій площі з мінімальною кількістю обладнання та меблів. Також зазвичай у таких кабінетах обмежена кількість спеціалістів та низька вартість послуг [4].

Підприємством середнього розміру є салон краси. У салонах краси надаються різноманітні послуги. Наприклад, стрижки дитячі, жіночі та чоловічі, класичний педикюр, манікюр комбінований та класичний. Вартість послуг залежить від кваліфікації спеціаліста. У таких салонах краси надаються додаткові сервіси, наприклад, продаж власної косметики [4].

Медичний центр краси та здоров'я – це багатoproфільна клініка, що надає різноманітні послуги косметологічного лікування та хірургії. Послуги варіюються від косметологічних процедур до пластичної хірургії. Дані послуги надають фахівці із вищою медичною освітою. Клініки при роботі застосовують сучасне обладнання та перевірені методики лікування [4].

Мережа підприємств краси складається з головного салону та декількох менших філій, які працюють за однаковими правилами та назвою. Можна додати, що такі мережі підприємств краси зазвичай мають філіальну структуру [4].

За кількістю послуг виділяють наступні типи салонів краси:

- центр косметології;
- SPA-центр;
- студії краси [5].

Центр косметології – це косметологічна клініка, яка надає комплекс лікарських процедур для обличчя та тіла. Фахівці косметологічних клінік зазвичай надають послуги ін'єкційної, апаратної, естетичної та лазерної косметології [5].

SPA-центр представляє собою заклад, який пропонує комплекс водних процедур із використанням корисних натуральних засобів (грязі, солі, трави, ароматичні олії) для відновлення та оздоровлення організму людини [5].

Студії краси – салон, що надає перукарські та нігтьові послуги. У студіях краси працюють спеціалісти з нарощування волосся, відновлення волосся, стрижок, укладки волосся, манікюру та педикюру [5].

За ціновою категорією виділяють наступні типи салонів краси:

- салони бюджетного класу;
- салони бізнес-класу;
- салони преміумкласу;
- салони класу люкс [6].

Салони краси бюджетного класу пропонують стандартний набір послуг, а саме укладка волосся, фарбування волосся, стрижки, манікюр та педикюр за мінімальною ціною. Для підтримування низького рівня цін, власники таких салонів використовують засоби економсегмента та орендовані приміщення. При зручному географічному положенні, гнучкому графіку роботи та потоку клієнтів салон може приносити непоганий прибуток і швидко повернути вкладені кошти. Персонал закладу представлений спеціалістами середнього рівня та початківцями [6].

Салони краси бізнес-класу надають стандартні послуги, а також пропонуються косметологічні процедури, солярій та масаж. Для роботи використовують косметичні засоби класу преміум. Серед персоналу такого салону краси працюють висококваліфіковані спеціалісти. Клієнт може отримати консультацію у спеціаліста з підбору косметичних засобів або засобів для догляду [6].

Салон краси преміумкласу – це заклад, який спеціалізується на процедурах по догляду за тілом та волоссям, пропонують SPA-процедури. Салони краси преміумкласу знаходяться в окремому приміщенні з дорогим ремонтом та продуманими деталями інтер'єру. Фахівцями закладу є відомі та висококваліфіковані спеціалісти. Однією з особливостей даного типу салону краси є можливість роботи зі стилістом, який допоможе зорієнтувати в модних тенденціях та підібрати образ [6].

Салони краси класу люкс обслуговують особливих клієнтів, яким надають косметологічні та преміум послуги. Такі салони мають високі вимоги до інтер'єру, спеціалістів та якості послуг [7].

Штат співробітників салону краси включає наступних фахівців:

- перукар;
- бухгалтер;
- адміністратор салону;
- косметолог;
- майстер перманенту;
- майстер шугарингу;
- візажист;
- керівник салону;
- стиліст;
- бровіст;
- лешмейкер;
- майстер манікюру;
- майстер педикюру [8].

Найнеобхіднішим обладнанням для салону краси є крісла, лампи, дзеркала, умивальниці, інструментальні шафи, обладнання для дезінфекції інструментів та інше. Необхідний мінімум для салонів краси бюджетного класу включає наступне обладнання:

- крісла для надання перукарських та нігтьових послуг;
- крісла та журнальний столик для клієнтів;

- шафи для зберігання інструментів;
- лампи;
- стерилізатори для інструментів;
- лампи для гель-лаків;
- ванночки;
- мийка;
- дзеркало;
- перукарські візки;
- професійні фени [9].

Для салонів краси бізнес-класу список обладнання збільшується, оскільки надаються послуги апаратної косметології, епіляції та працює масажний кабінет. До списку обладнання наведеного вище можна додати:

- фрейзер;
- косметологічна канапа;
- масажні столи;
- апарати для лазерної епіляції;
- апарати для косметологічних процедур [9].

Для салонів краси преміумкласу обладнання повинно бути якісним та дорогим. До послуг такого типу салону краси додаються, наприклад, SPA-процедури, тоді знадобляться SPA-канапи, SPA-комплекси та шафи для роздягалень. Обладнання для лазерної епіляції необхідно обрати найефективніше та дороге, тому якісний лазер стане найкращим рішенням. При купівлі обладнання необхідно звертати увагу на функціональність та естетичну привабливість [9].

Проводячи дослідження предметної області було сформовано перелік послуг, які надаються у салонах краси та наведено результати нижче на рисунку 1.1.

Послуги у салонах краси		
Перукарські послуги	Візаж	Косметологія
1. Відновлююча маска для блиску волосся 2. Зачіска для особливих випадків 3. Зачіска/ складне плетіння / афронакрутка 4. Догляд та лікування волосся (довге) 5. Догляд та лікування волосся (коротке) 6. Догляд та лікування волосся (середнє) 7. Ламінування волосся 8. Стрижка жіноча 9. Стрижка підліткова 10. Стрижка дитяча 11. Стрижка модельними ножицями 12. Стрижка машинкою (для чоловіків) 13. Укладка волосся (для чоловіків) 14. Укладка (локони до пліч) 15. Укладка (на нарощене волосся) 16. Укладка (пряме волосся до пліч) 17. Укладка (пряме волосся нижче пліч) 18. Камуфлювання волосся (для чоловіків) 19. Кератинове випрямлення волосся (довге) 20. Кератинове випрямлення волосся (коротке) 21. Кератинове випрямлення волосся (середнє) 22. Складне фарбування волосся (довге) 23. Складне фарбування волосся (середнє) 24. Тонування волосся (довге) 25. Тонування волосся (коротке) 26. Тонування волосся (середнє) 27. Фарбування волосся 2-х етапне (довге) 28. Фарбування волосся 2-х етапне (коротке) 29. Фарбування волосся 2-х етапне (середнє) 30. Мелірування волосся 31. Оформлення бороди / вусів	1. Зняття нарощених вій 2. Корекція нарощених вій 3. Ламінування вій 4. Зняття нарощених вій 5. Нарощування вій 6. Нарощування вій 2D 7. Нарощування вій 3D 8. Фарбування вій 9. Ламінування брів 10. Біозавивка вій 11. Макіяж вечірній 12. Макіяж для особливих випадків 13. Коктейльний макіяж 14. Макіяж нюдовий 15. Забарвлення брів 16. Забарвлення вій 17. Корекція брів 18. Оформлення брів	1. Комбінована чистка обличчя 2. Біоревіталізація 3. Контурна пластика 4. Мезотерапія 5. Механічна чистка обличчя 6. Мікротоки 7. Обгортання повне 8. Обгортання часткове 9. Ультразвукова чистка обличчя
Нігтьові послуги	Солярій	Масаж
1. Комплексний манікюр 2. Комплексний педикюр 3. Зняття покриття гель-лак 4. Комплексний СПА манікюр 5. Комплексний СПА педикюр 6. Нарощування нігтів 7. Педикюр чоловічий класичний 8. Холодна парафінотерапія для рук 9. Покриття нігтів гель-лаком Френч / простий дизайн (усі пальці) 10. Реконструкція нігтів IBX 11. Складний педикюр (педикюр з усуненням врослих нігтів, мозолів)	1. Солярій 2. Салонна засмага	1. Загальний масаж тіла (60 хв) 2. Загальний масаж тіла (90 хв) 3. Масаж антицелюлітний (60 хв) 4. Масаж антицелюлітний (90 хв) 5. Масаж комірцевої зони (20 хв) 6. Масаж обличчя 7. Масаж спини (30 хв) 8. Масаж стоп 9. Ліфтинг-масаж обличчя 10. Релакс-масаж

Рисунок 1.1 – Перелік послуг у салонах краси

Переваги салонного бізнесу:

- високий попит на послуги у салонах краси завжди залишається;
- можна відкривати додаткові кабінети та розширювати список послуг;
- продаж власних товарів для догляду за обличчям або косметичних засобів;
- постійні клієнти [10].

Недоліки салонного бізнесу:

- необхідно мати досвід для організації бізнесу у сфері послуг краси;
- багато коштів необхідно вкласти у покупку обладнання та ремонт приміщення;

– високий рівень конкуренції [10].

Обов'язки керівника салону краси передбачають складання графіку роботи співробітників, закупівлі необхідних матеріалів для надання послуг, контроль постачання товарів та ремонт обладнання при виникненні такої необхідності. Головним обов'язком керівника салону є забезпечення комерційного успіху закладу. До основних професійних обов'язків керівника салону входить нарахування заробітної плати співробітникам, взаєморозрахунки з постачальниками, повернення грошей клієнтам, забезпечення належного рівня обслуговування, зворотний зв'язок з клієнтами салону та підбір персоналу [11].

Однією з найважливіших завдань керівника салону краси є роз'яснення працівникам їх обов'язків та прав. Знання обов'язків є основним фактором у продуктивній роботі кожного співробітника. Зазвичай для цього використовують посадові інструкції у яких описано посадові обов'язки та права [12].

Посадова інструкція адміністратора салону має включати знання нормативних документів органів вищого рівня, обов'язки співробітників організації, методи організації процесу обслуговування клієнтів, послуги організації, основи маркетингу, принципи оформлення приміщень та вітрин салону краси, основи організації праці та управління, а також правила техніки безпеки [13].

Адміністратор салону краси зобов'язаний виконувати наступні посадові обов'язки:

- підтримання високого рівня обслуговування;
- продаж засобів для догляду, консультування та допомога клієнтам з приводу вибору косметичних засобів, оформлення замовлень;
- вести записи клієнтів на послуги до спеціалістів, консультування з приводу послуг і вибору спеціаліста;
- проводити облік фінансів;
- проводити контроль залишків товарів на складах;

- проводити анкетування клієнтів;
- слідкувати за дотриманням трудової дисципліни співробітниками;
- організовувати та контролювати маркетингові заходи;
- розглядати негативні відгуки клієнтів через незадовільне обслуговування та проводити відповідні організаційно-технічні заходи;
- вирішення конфліктних ситуацій між співробітниками та клієнтами;
- забезпечувати чистоту та порядок в приміщеннях [14].

У посадовій інструкції перукаря салону краси передбачається вміння обслуговування клієнтів, робити якісно та швидко модельні стрижки, зачіски (жіночі, чоловічі, дитячі), фарбування та ряд інших робіт за видами перукарських послуг. Інструкція майстра манікюру та педикюру передбачає вміти обробляти нігті, шкіру рук і ніг, дотримуватися правил гігієни та санітарних норм [15].

Стратегії розвитку бізнесу салону краси зводяться до трьох напрямків:

- збільшення прибутку салону;
- зниження операційних витрат;
- масштабування бізнесу [16].

Для обліку бізнес-процесів у салоні краси використовуються системи управління відносинами з клієнтами або CRM [17].

CRM (Customer Relationship Management) – система, яка допомагає компаніям в управлінні взаємовідносинами з реальними та потенційними клієнтами. Автоматизовані системи для салонів краси забезпечують автоматизацію обслуговування клієнтів, керування складом, продажів власних товарів компанії, замовлень товарів, а також фінансових операцій та ведення обліку заробітної плати співробітників. Метою впровадження CRM є покращення взаємовідносин із клієнтами шляхом кращого розуміння їх потреб та надання якіснішого обслуговування [18].

До функціональних можливостей автоматизованих систем для салонів краси входить збір, систематизація та збереження клієнтських даних, а також автоматизація щоденних завдань співробітників та аналітичні звіти [19].

CRM для салонів краси дозволяють автоматизувати наступні завдання компанії:

- ведення обліку клієнтів;
- ведення електронного журналу записів на послуги;
- створення та контроль виконання завдань співробітниками;
- повідомлення та нагадування клієнтам;
- облік фінансових операцій;
- облік та продаж доглядових засобів;
- розрахунок заробітної плати співробітників;
- збір відгуків від клієнтів;
- керування подарунковими сертифікатами та абонементом;
- керування графіком подій;
- керування знижками на послуги;
- касовий облік;
- складський контроль;
- переглядати ключові показники компанії;
- переглядати аналітичні звіти робочих процесів компанії [20].

Існує дуже багато CRM, які можуть надавати різні функціональні можливості. При виборі CRM необхідно орієнтуватися на наступні характеристики та внутрішні процеси організації:

- розмір бізнесу;
- функціональні характеристики;
- налаштування модулів програми;
- інтеграції з іншими сервісами;
- бізнес-процеси власної компанії [21].

Отже, впровадження та використання CRM для салонів краси забезпечує керівника необхідними інструментами для прийняття правильних управлінських рішень, а працівникам дозволяє якісно виконувати свої посадові обов'язки [22].

1.2 Бізнес-процеси у салоні краси

Бізнес-процес – це логічна впорядкована взаємопов'язана послідовність дій або операцій, виконуваних посадовими особами та підрозділами компанії для отримання бажаного кінцевого результату (досягнення мети, вирішення завдання, реалізації програмного продукту, надання послуги). Будь-який процес обов'язково повинен мати вхід і вихід, тобто отримувати будь-яку інформацію або ресурси від зовнішнього об'єкта або іншого процесу, перетворювати їх і передавати іншому об'єкту або процесу [23].

Для малого та середнього бізнесу виокремлюють 4 види бізнес-процесів:

- основні процеси (надання послуг або виробництво товарів);
- допоміжні процеси (комунікація з постачальниками);
- адміністративні процеси (бухгалтерський та кадровий облік);
- керуючі процеси (стратегічне планування) [24].

Без опису дій та їх послідовності, операції не будуть регламентовані, а отже порядок виконання та отриманий кінцевий результат теж буде відрізнятися. Опис бізнес-процесів дозволяє побачити недоліки в організації робочого процесу. Зазвичай описом бізнес-процесів займається керівник салону краси, але краще перед описом проводити збір інформації разом зі співробітниками. Необхідно отримати інформацію про вхідні дані для запуску певного процесу, вихідних даних – результату дій співробітника, а також покроково зафіксувати потрібні дії. Після того, як всю інформацію зібрано, її потрібно перевести в графічну нотацію [25].

Для опису бізнес-процесів використовуються схеми, діаграми, таблиці та графіки. Також використовуються різні нотації, такі як BPMN, EPC та IDEF0. Всі дії, які описуються в бізнес-процесі, повинні бути логічними, їхня послідовність має призводити до певної поставленої мети [25].

Нотація BPMN є одним з найпоширеніших стандартів процесного моделювання. Основною відмінністю процесного моделювання від функціонального є те, що процес представлений послідовністю дій у певних межах, які мають початок та кінець події. Функціональне моделювання засноване на бізнес-моделі з погляду наявних ресурсів та бажаного результату. На підпроцеси можуть розбиватися процеси до деталізації лише на рівні дій, у яких подальша деталізація неможлива [25].

Процесне моделювання зосереджено на тому, що бажаємо отримати та зробити для одержання результату, тобто не результатам, а саме послідовності дій. Однією з головних відмінностей функціональної моделі є те, що її використовують тільки для проєктування. Процесне моделювання дозволяє створювати описи послідовності дій, які використовуються при реалізації якоїсь системи [25].

Функціональне моделювання реалізовано у нотації IDEF0. При функціональному моделюванні бізнес-модель розглядається як функція. Функціональні моделі можна використовувати для демонстрації ідей та варіантів рішень, оскільки в процесі демонстрації можна рухатися від загального до деталей. Також при необхідності розділяти та декомпонувати функції. У функціональній моделі функція має точку входу та точку виходу [25].

Нотація EPC або ланцюжок подій процесів призначена для детального опису процесів нижнього рівня, що виконуються одним підрозділом, декількома підрозділами або конкретними співробітниками. Діаграма в нотації EPC є комбінацією подій та функцій. У моделі EPC для кожної функції можуть бути сформовані початкові та кінцеві події, відповідальні виконавці, матеріальні та документальні потоки. Для будь-якої функції даної моделі може бути проведена декомпозиція на нижчі рівні (підфункції) [25].

У салонах краси бізнес-процесами є облік клієнтів, запис клієнтів, облік співробітників, облік заробітних плат, постачання та надходження товарів, контроль залишків товарів на складах, облік фінансів, маркетингові

процеси, звітність, контроль якості, надання послуг, взаємодія з клієнтом, підбір персоналу та обслуговування обладнання [24].

1.3 Огляд клієнт-серверних застосунків для салонів краси

Нижче наведено результати проведеного дослідження вебзастосунків для керування бізнес-процесами у салоні краси. Для дослідження було обрано системи «CleverBOX:CRM», «Wlaunch» та «RemOnline».

1.3.1 Вебплатформа «CleverBOX:CRM»

Одним з аналогічних програмних продуктів є «CleverBOX:CRM». «CleverBOX:CRM» – це система для автоматизації щоденних завдань співробітників та керування бізнес-процесами у салонах краси, а також косметологічних та медичних клініках (рис. 1.2). Система автоматизує записи, фінанси, зарплати, склад, аналітику та маркетинг [26].

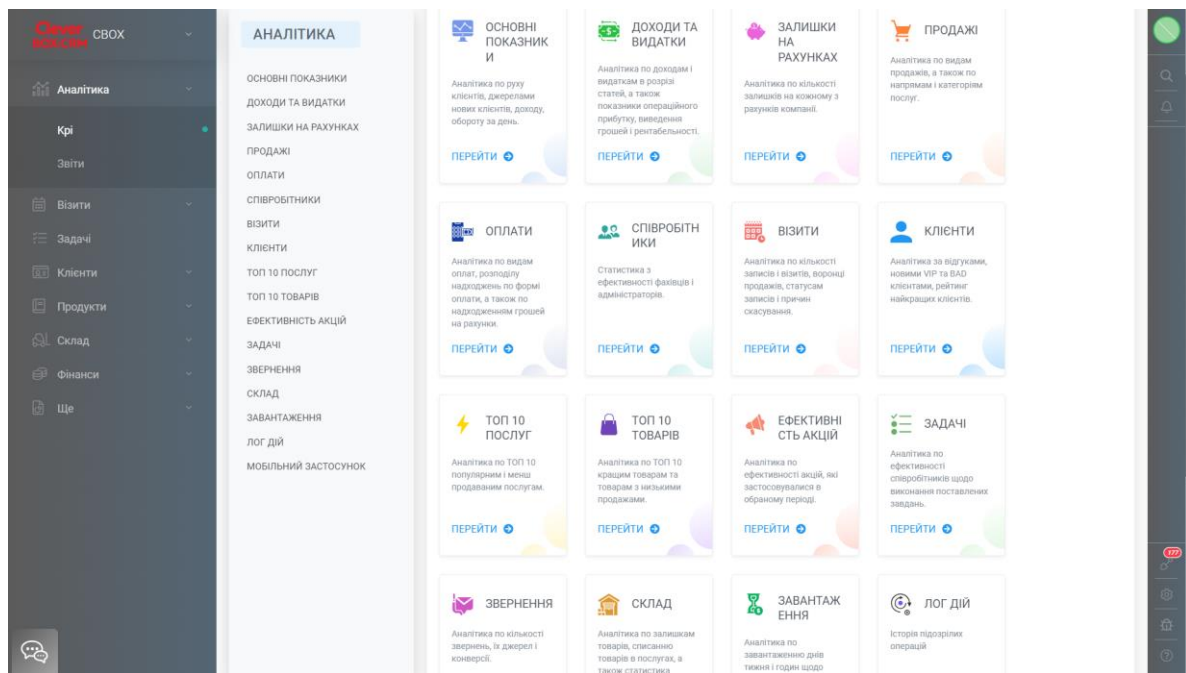


Рисунок 1.2 – Інтерфейс головної сторінки вебзастосунку «CleverBOX:CRM»

Функціональні можливості вебзастосунку «CleverBOX:CRM»:

- перегляд ключових показників компанії;
- перегляд аналітичних звітів;
- налаштування інтерфейсу;
- пошук клієнтських абонементів або сертифікатів;
- налаштування повідомлень;
- запис клієнта на послугу;
- запис кількох клієнтів на однаковий час;
- ведення журналу бронювання записів на послуги;
- створення завдань співробітникам;
- ведення клієнтської бази;
- відображення бази потенційних клієнтів;
- додавання звернень від клієнтів;
- ведення бази послуг або товарів компанії;
- створення техкарт за послугами;
- поповнення балансу клієнта;
- лист очікування клієнтів;
- статистика за день;
- додавання сертифікатів або абонементів;
- додавання типів знижок;
- створення прибуткових накладних;
- ведення бази замовлень постачальникам;
- ведення бази документів переміщення між складами;
- ведення бази документів списання товарів при продажі;
- ведення бази документів повернення товарів постачальнику;
- проведення інвентаризації товарів на складі;
- ведення бази постачальників товарів;
- відображення звіту залишків товарів на складах;
- відображення інформації про залишки товарів;

- ведення журналу фінансових операцій компанії;
- автоматичний розрахунок та формування зарплатної відомості;
- відображення інформації про взаєморозрахунки зі співробітниками;
- відображення звіту продажів за день;
- відображення інформації про рух бонусів клієнтів;
- відображення інформації про всі дзвінки за обраний період;
- відображення графіку роботи співробітників;
- відображення відгуків клієнтів;
- додавання спеціалістів компанії;
- налаштування модулів системи;
- встановлення функціональних розширень [26].

Вебзастосунок «CleverBOX:CRM» дозволяє отримати звіти з усіх напрямків та процесів компанії: сертифікати, продажі, фінанси, склад, візити, співробітники та комунікації з клієнтами. Вебзастосунок надає можливість керівнику салону краси переглядати ключові показники компанії за певний період [27].

До аналітичних звітів входять:

- звіт про прибуток та витрати;
- аналітика кількості залишків на рахунках компанії;
- аналітика продажів за напрямками послуг;
- аналітика за різними видами оплат клієнтів;
- аналітика візитів клієнтів;
- аналітика клієнтів;
- аналітика найбільш та найменш проданих послуг або товарів;
- аналітика ефективності знижок;
- аналітика ефективності виконання завдань співробітниками;
- аналітика за кількістю звернень від клієнтів;
- аналітика постачання товарів;
- аналітика завантаження тижня щодо візитів клієнтів [27].

Журнал запису у вебзастосунку «CleverBOX:CRM» призначений для ведення записів клієнтів на послуги та перегляду співробітниками свого розкладу. Якщо клієнт бажає записатися на час, який вже є зайнятим, його можна внести в лист очікування, щоб у разі звільнення часу швидко зв'язатися та записатися на потрібну послугу. Перегляд візитів можливий колонками або таблицею. У колонках візити розподілені залежно від статусу візиту, а у таблиці є можливість перегляду відфільтрованих візитів [28].

Вебзастосунок «CleverBOX:CRM» надає спеціалістам модуль для коротких швидких записів, які спеціалісти однієї зміни можуть залишити для спеціалістів іншої зміни [28].

У системі є можливість редагування візиту, а саме додавання послуг до різних спеціалістів, зміна спеціаліста, обрання іншої послуги, обрання іншого часу надання послуги, додавання додаткових послуг, товарів або ресурсів. Ресурси у візиті можна змінити, видалити або додати інші ресурси вручну [28].

У вебзастосунку «CleverBOX:CRM» існує три типи задач:

- автоматичні задачі з нагадуванням про повторний візит;
- повторювані задачі;
- звичайні задачі [29].

Модуль автозадачі дозволяє автоматизувати процес нагадування про повторний візит клієнтам. Для коректної роботи модуля необхідно, щоб в налаштуванні послуги була проставлена періодичність повтору днів [29].

Таблиця клієнтів призначена для обліку клієнтів. У таблиці можна додати нового клієнта, редагувати картку клієнта, видалити картку клієнта, відфільтрувати клієнтів, надіслати повідомлення, завантажити базу клієнтів та переглянути статистику клієнтської бази [30].

Функція групової обробки клієнтів дозволяє виконувати дії над обраними клієнтами, а саме привласнити великій кількості клієнтів однакову групу, тег, нарахувати або обнулити бонусний рахунок. Актуальний баланс на рахунку клієнта можна перевірити у звітах балансів клієнтів та карті

клієнта. Система підтримує функцію об'єднання клієнтів, якщо той же самий клієнт внесений в систему під різною інформацією [30].

Звернення передбачає будь-який контакт потенційного або існуючого клієнта з організацією. Найбільш поширеними прикладами звернень є запис на процедуру, перенесення запису на інший час, скасування запису, дізнатися вартість послуг тощо. Джерелами звернень може бути дзвінок або месенджер. Причинами відмов може бути зайнятість обраного спеціаліста, не влаштування ціни або часу тощо. При заповненні звернень клієнтів використовуються статуси та додаткові класифікатори, які також можна налаштовувати вручну [30].

Бонусна система – це система лояльності, що дозволяє клієнтам накопичувати бонуси та розплачуватися ними за послуги. Система лояльності допомагає збільшити суму чека та частоту відвідувань [30].

Функція обліку рекламацій є дуже важливим інструментом для управління лояльністю клієнтів, але зустрічається у системах не так часто. Створюватися рекламації можуть автоматично на основі негативних відгуків клієнтів [30].

У системі надається можливість користувачам працювати із всіма складськими операціями, такими як постачання товару, списання товару, обнулення залишку товару, переміщення товару та інвентаризація [31].

Прибуткова накладна або постачання товару – це документ, який фіксує покупку товарів у конкретного постачальника. Постачання може відбуватися за новим товаром та існуючим товаром для поповнення залишку. Після створення прибуткової накладної змінюються залишки на складі. Прибуткову накладну використовують при купівлі нового товару у постачальника [31].

Інвентаризація у системі передбачає перевірку залишків товарів на складі за обраний період. Система показує результат інвентаризації проводячи порівняння розрахункових та фактичних залишків на складі, тобто результатом є різниця між залишками у кількості та грошах [31].

У вебзастосунку списання товарів бувають ручні та автоматичні. Ручні списання товарів можна формувати самостійно, а автоматичні списання товарів формує система виходячи з надання послуги або продажу товару [31].

Система автоматично підраховує собівартість товарів та послуг. Собівартість товарів та витратних матеріалів розраховується за методом середньозваженої собівартості [31].

Над товарами можна проводити наступні операції:

- додавання товару;
- додавання складового товару;
- копіювання картки товару;
- видалення або переміщення товару до архіву;
- групова обробка товарів;
- зміна одиниць виміру та коефіцієнту у товарі;
- редагування ціни продажу товару у довіднику [31].

Якщо необхідно привласнити великій кількості товарів однаковий напрямок, групу або бренд використовується функція групової обробки [31].

Модуль техкарти використовується для автоматизації списання витратних матеріалів при наданні послуги. Техкарта представляє собою шаблон витратних матеріалів для списання. У візит автоматично підтягуються витратні матеріали з техкарти при додаванні послуги до якої прикріплена техкарта, але за необхідності їх можна змінити в обраному візиті. Для групування техкарт використовуються категорії, що дозволяють прискорити пошук необхідної техкарти у меню [31].

При створенні техкарти у звичайному режимі витратні матеріали списуватимуться зі складу, який зазначений у профілі спеціаліста, що надає послугу. Вказання максимальної собівартості необхідно, щоб собівартість послуги в візиті не перевищувала норму та співробітники не змогли додати більше витратних матеріалів за послугою, ніж заплановано стандартами. Розширений режим техкарт передбачає списання різних витратних матеріалів

з різних складів. Витратні матеріали в техкарті розбиваються за категоріями, оскільки у кожній категорії вказується окремий склад [31].

У розділі «Фінанси» відображається журнал руху грошей по кожному рахунку організації, а саме надходження, витрати, переміщення грошей між рахунками та сторнування. Сторнування – це спосіб редагування даних в обліку. Сторнування дозволяє виправити допущені помилки в обліку щодо залишків на рахунках. Для видалення фінансової операції необхідно ввести пароль [32].

Вебплатформа «CleverBOX:CRM» дозволяє проводити оплату різними видами оплат: готівка, карта, термінал, сертифікат, абонемент, баланс та бонуси. Також можна контролювати прибуток, виручку компанії та залишки по касі [32].

Розділ «Співробітники» надає можливість додавати профілі співробітників та у налаштуваннях для доданих ролей (посад) налаштовувати параметри доступу до різного функціоналу всередині модулів. Після налаштувань спеціаліст користується тими функціями системи, які йому необхідні для роботи враховуючи посадові обов'язки. Коли спеціаліст звільняється з компанії, тоді його необхідно видалити із системи. При видаленні із системи звільненого співробітника його незавершені завдання потрібно передати іншому обраному спеціалісту. Система також надає функцію відновлення співробітника, якщо випадково було звільнено співробітника або необхідно повернути його до списку актуальних співробітників [33].

Посада адміністратор системи є головною у вебзастосунку «CleverBOX:CRM» з найширшими правами доступу до функціонала модулів. Дану посаду необхідно використовувати керівникам або директорам компанії [33].

Вебплатформа «CleverBOX:CRM» є єдиною платформою, яка дозволяє приховати доступ до залишку грошей на рахунку, але відкрити можливість приймати оплати [33].

Розділ «Графік роботи» передбачає створення графіку роботи всіх співробітників організації за днями та місяцями. Якщо спеціаліста не додати в графік співробітників, то він буде відсутній у календарі та до нього неможливо буде записати клієнтів на послуги [26].

Розділ «Послуги» призначений для ведення бази послуг компанії всередині системи та надає можливість виконувати різні операції з ними. У таблиці можна додавати нову послугу, видаляти послугу, редагувати існуючу послугу, завантажувати базу послуг, відфільтровувати пошук послуг за категоріями, оновлювати аналітику, переглядати архів послуг та перераховувати візити за актуальними цінами [26].

У вебзастосунку «CleverBOX:CRM» існує три типи послуг:

- окрема послуга;
- додаткова послуга;
- пакет послуг [26].

Окрема послуга – це тип послуги, яка може бути додана окремо в візит та є самостійною незалежною послугою. Додаткова послуга передбачає послугу, яка може бути додана лише до основної послуги. Пакет послуг представляє декілька послуг, які надаються клієнтам паралельно або по чергово [26].

Система «CleverBOX:CRM» надає можливість створювати додаткові категорії цін, щоб у деяких співробітників ціни на послуги відрізнялися від базової [26].

У системі передбачено облік клієнтських абонементів та сертифікатів. Абонемент містить заздалегідь оплачений набір послуг, які можна відвідати, а сертифікат представляє собою віртуальний рахунок з балансом для оплати товарів або послуг [26].

Існує у системі «CleverBOX:CRM» два типи абонементів:

- роздільний абонемент (індивідуальна кількість відвідувань для кожної обраної послуги);

– універсальний абонемент (загальна кількість відвідувань для різних послуг) [26].

Система надає можливості для комунікації з клієнтами. Наприклад, після створення візиту система автоматично відправить усім клієнтам повідомлення про підтвердження запису з ключовою інформацією щодо візиту [26].

Розділ «Відгуки» автоматично збирає відгуки від клієнтів. Відгуки можна збирати надсилаючи клієнту повідомлення з посиланням на сторінку з питаннями про візит. Зовнішній вигляд сторінки з питаннями можна налаштовувати. Ще одним варіантом збирання відгуків є сторонні сторінки або власний вебсайт [26].

Модуль інтеграції призначений для зв'язку з телефонією. Функціями інтеграції є повідомлення про вхідні дзвінки, відображення інформації про прийняття дзвінка, відображення інформації про пропущений дзвінок, інтеграція статистики телефонних розмов та записів дзвінків, що зв'язується з картками клієнтів та інтегрується до клієнтської бази даних. Система «CleverBOX:CRM» надає можливість інтеграції з такими сервісами телефонії як Binotel, Stream Telecom та Ringostat [34].

Оплатити візит у системі можна через продаж та картку візиту. Система дозволяє продавати товар у візиті та кошику. Ініціатором продажу товару у візиті є співробітник, що виконує послугу. Змінювати ініціатора продажу товару, який закріплений за послугою у візиті неможливо. Лише у кошику можна змінювати ініціатора продажу [32].

У систему «CleverBOX:CRM» інтегрований сервіс Checkbox. Checkbox – це програмний реєстратор розрахункових операцій, що дозволяє фіскалізувати чеки не використовуючи фізичний касовий апарат [34].

Використання знижок є одним із методів збільшення обсягів продажу послуг та товарів. Серед видів знижок система має звичайні знижки для всіх клієнтів та персональні знижки для кожного клієнта [26].

Модуль зарплати у системі дозволяє формувати зарплатну відомість та проводить автоматичний розрахунок. Розділ «Взаєморозрахунки зі співробітниками» відображає інформацію про взаєморозрахунки зі співробітниками. У таблиці по кожному співробітнику відображено інформацію про нарахування, утримання та виплати заробітної плати. Система дозволяє додатково нараховувати премії, відпускні, утримувати штрафи та податки з нарахованої суми заробітної плати. Також у системі присутня кнопка видалення нарахованої заробітної плати конкретному співробітнику, зарплатної відомості та виплати зарплати [35].

У системі можна налаштовувати різні правила нарахування заробітної плати, а саме створювати окреме правило для кожного зі співробітників або використовувати один оклад для декількох співробітників [35].

Вебзастосунок «CleverBOX:CRM» має модуль застосунок для клієнтів та модуль застосунок для спеціалістів, які можна встановити за допомогою магазину функціональних розширень. Після цього необхідно у налаштуваннях модуля налаштувати зовнішній вигляд застосунку та необхідні функціональні можливості [36].

Для спеціаліста мобільний застосунок дозволяє записувати клієнта, редагувати техкарту, переглядати власний розклад, додавати фото до та після послуги, залишати коментарі про візит, продавати товари у візиті, переглядати свою заробітну плату за поточний місяць та переглядати собівартість послуг [37].

Для клієнта мобільний застосунок надає можливість записуватися на послуги, переглядати історію візитів, отримувати push-сповіщення, оплачувати послуги онлайн, залишати відгуки щодо візитів, витрачати бонуси, переглядати новини та актуальні знижки [37].

Переваги вебзастосунку «CleverBOX:CRM»:

- створення власного мобільного застосунку для спеціалістів та клієнтів;
- автоматичні сповіщення та нагадування клієнтам;

- контроль за останніми діями користувачів у системі;
- бронювання візиту;
- звіт про бронювання клієнтами бажаного часу на послуги;
- повідомлення про оновлення у розділах вебзастосунку;
- зберігання файлів;
- інтеграції з іншими сервісами;
- магазин розширень функціональних можливостей вебзастосунку;
- додавання фото до та після візиту клієнта;
- групове нарахування або списання бонусів;
- відкриття та закриття віртуальної каси;
- перенесення візиту на іншу дату;
- редагування у техкартці витратних матеріалів під час візиту;
- списання витратних матеріалів за допомогою технологічних карт;
- створення нового клієнта у вікні запису на послугу;
- запис клієнта на послугу без реєстрації картки клієнта;
- додавання клієнта при записах на візит з історії дзвінків;
- облік реклаमाцій;
- адаптивний інтерфейс;
- інтеграція з сервісом Checkbox для фіскалізації чеків;
- налаштування колонок при відображенні інформації;
- функція групової обробки клієнтів;
- створення вебсайту організації [26].

Недоліки вебзастосунку «CleverBOX:CRM»:

- відсутня можливість додавання подій компанії;
- відсутня можливість додавання форм опитувань та відображення результатів проведення опитувань;
- платність [26].

Вебсторінки створення задач та редагування клієнта вебзастосунку «CleverBOX:CRM» наведено нижче на рисунках 1.3 та 1.4.

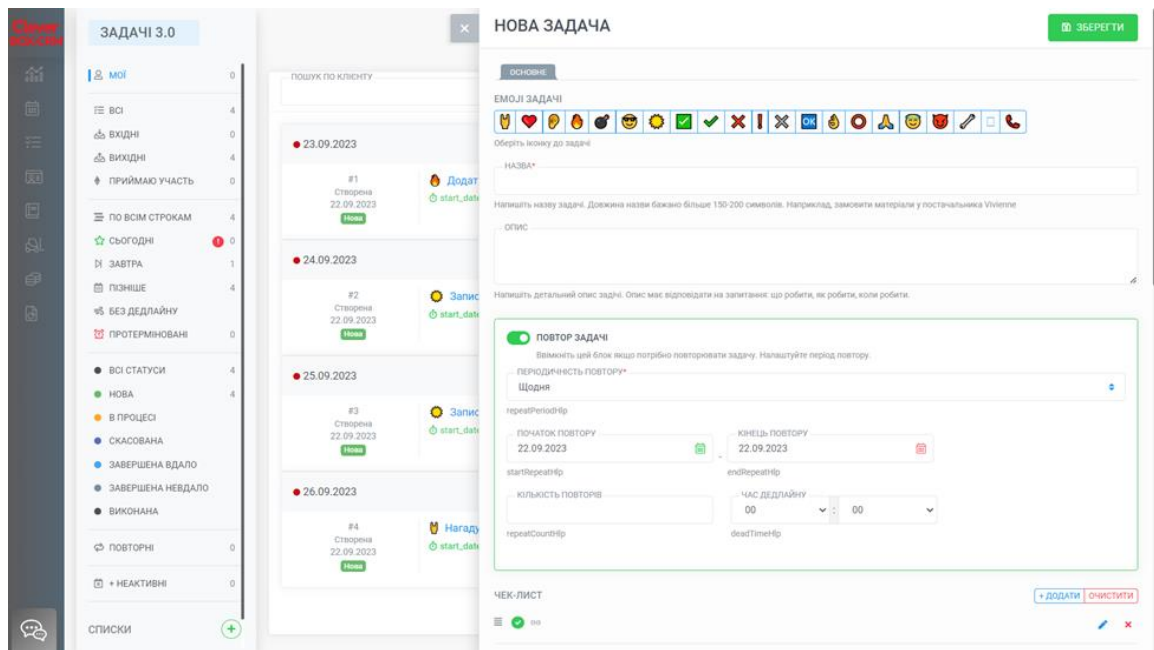


Рисунок 1.3 – Вебсторінка створення задачі системи «CleverBOX:CRM»

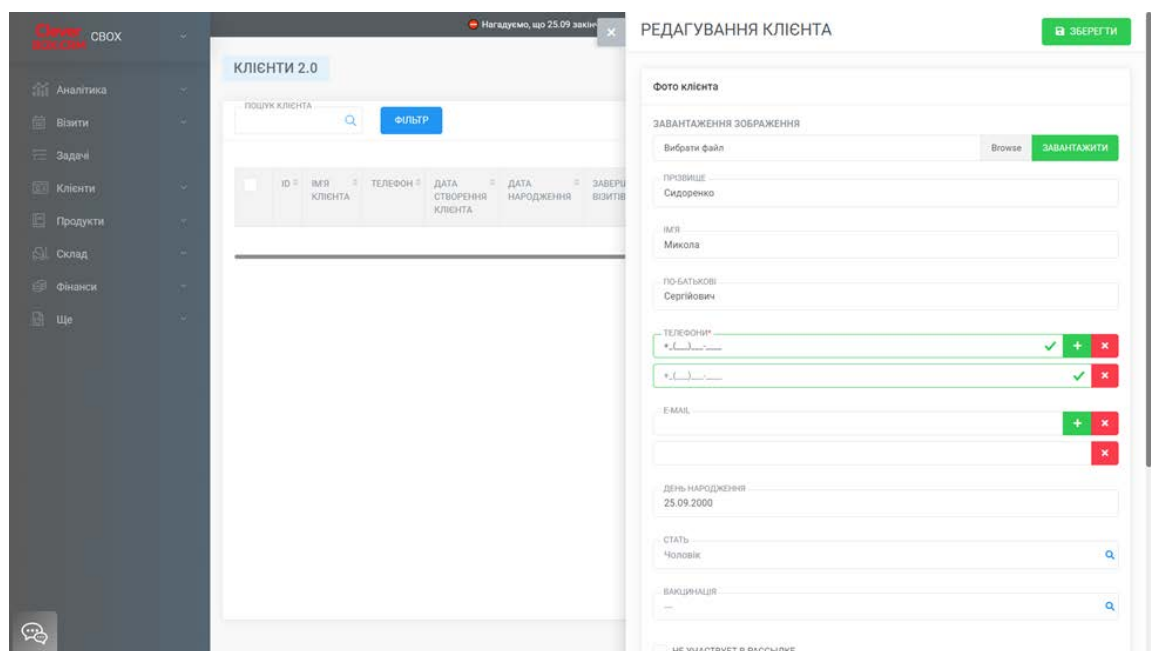


Рисунок 1.4 – Вебсторінка редагування клієнта системи «CleverBOX:CRM»

1.3.2 Вебзастосунок «Wlaunch»

Наступним вебзастосунком, який вирішує аналогічні завдання є «Wlaunch». «Wlaunch» – це система для автоматизації та обліку різних бізнес-процесів у різних сферах діяльності (рис. 1.5). Система має дуже

багато функціональних можливостей та надає можливість керувати філіями компанії [38].

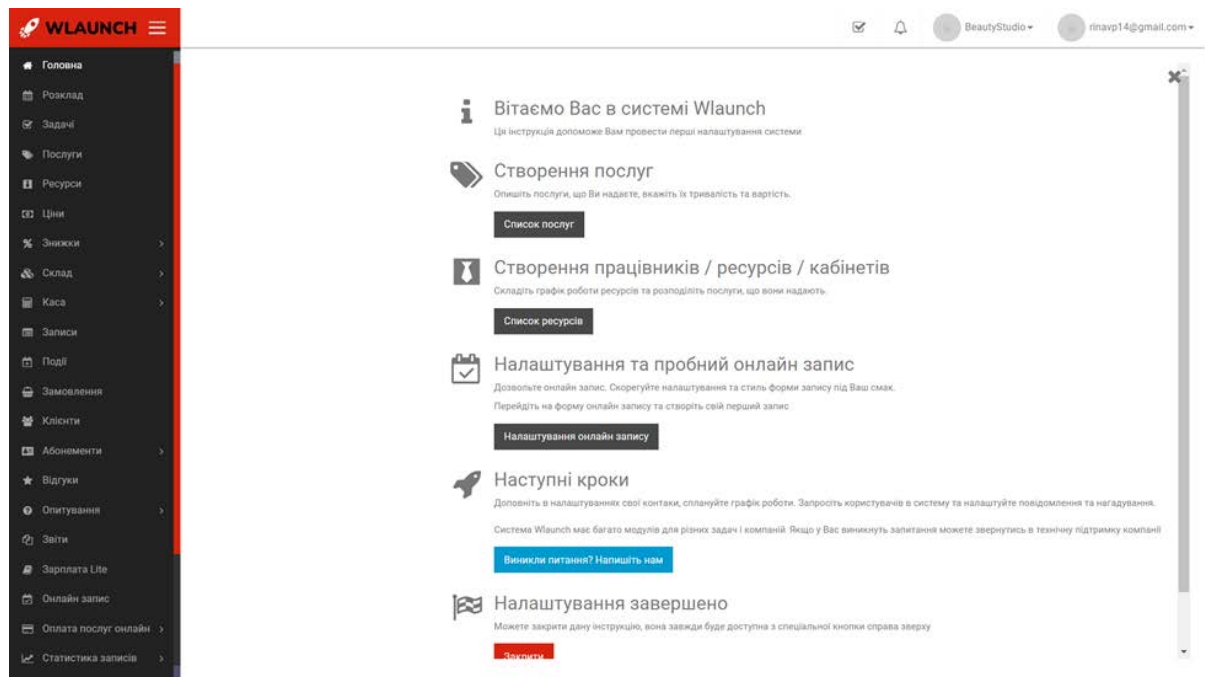


Рисунок 1.5 – Інтерфейс головної сторінки вебзастосунку «Wlaunch»

Функціональні можливості вебзастосунку «Wlaunch»:

- додавання нової філії компанії;
- додавання послуг компанії;
- створення віртуальних тек для групування послуг по типах та категоріям;
- імпорт послуг з інших систем та експорт з системи;
- додавання ресурсів компанії;
- керування цінами на послуги компанії;
- керування цінами на товари компанії;
- додавання знижок на послуги;
- керування складами компанії;
- додавання товару на склад;
- додавання груп товарів;
- обмін даними про товари з зовнішніми системами;

- контроль залишків товарів;
- ведення технологічних карт для послуг компанії;
- додавання одиниць виміру для товарів компанії;
- ведення бази контрагентів компанії;
- додавання складських документів;
- додавання документів операцій переміщення між складами;
- створення нових кас компанії;
- додавання касових документів;
- відображення записів компанії;
- відображення списку замовлень у компанії;
- обмін даними про клієнтів з зовнішніми системами;
- додавання нових клієнтів компанії;
- створення шаблону для нових клієнтських абонементів;
- створення груп шаблонів;
- перегляд відгуків клієнтів;
- створення нових форм опитувань;
- відображення результатів проведених опитувань;
- перегляд фінансової звітності;
- перегляд звітів клієнтської бази;
- перегляд складської звітності;
- ведення планувальника записів клієнтів;
- керування завантаженістю співробітників та ресурсів компанії;
- додавання завдань співробітникам [39].

Філія – це окремий підрозділ організації, який розташований у різних місцях та надає певні послуги або товари. Вебзастосунок «Wlaunch» використовує філії для відстеження зайнятості ресурсів в різних місцях роботи організації [40].

До конкретного ресурсу або філії може бути прикріплена послуга, якщо організація має декілька місць надання послуг. Наприклад, у салоні краси послугами можуть бути відновлення волосся, стрижка, укладка

волосся, фарбування, манікюр та педикюр. Клієнт може записатися на будь-яку з цих послуг в конкретній філії, якщо є така можливість [41].

Контроль залишків дозволяє відстежувати кількість товарів на складі та контролювати їх рух. Використовується для планування закупівель, уникнення надмірних запасів та забезпечення наявності товарів на складі. Система контролю залишків інтегрована зі складським балансом, що забезпечує автоматичне оновлення залишків після кожної операції з товаром [41].

Технологічна картка у контролі залишків може містити перелік товарів, які використовуються для послуги і є складовою її вартості. Картка дозволяє контролювати наявність товарів на складі та автоматично списувати товари зі складу при створенні запису [41].

Вебзастосунок «Wlaunch» надає можливість налаштування кожної філії та переведення спеціалістів між філіями [42].

Ключові можливості для роботи з календарем у вебзастосунку «Wlaunch»:

- перегляд тижневого або денного графіка;
- додавання спеціаліста в розклад;
- налаштування відображення розкладу;
- додавання подій;
- додавання перерви;
- видалення однієї з перерв;
- змінення годин роботи;
- видалення з розкладу;
- перехід на екран керування робочим часом;
- навігація у часі;
- відображення детальної інформації запису на розкладі при наведенні на значок;
- перегляд інформації щодо запису в деталях запису;
- редагування ресурсу та часу початку;

- керування джерелом запису;
- керування статусом запису;
- видалення запису;
- редагування секції клієнта;
- використання абонементів та статистика використаних;
- налаштування повідомлень по окремому запису;
- відображення каси по якій проведено оплату;
- додавання послуг та редагування параметрів [43].

Вебзастосунок «Wlaunch» має опцію підключення форми запису на послуги. Коли у компанії немає власного вебсайту або він тимчасово не працює, можна використовувати форму запису як повноцінний сайт [44].

Для автоматизації та керування складом у сфері надання послуг вебзастосунок «Wlaunch» надає наступні можливості:

- створювати декілька складів для ведення обліку запасів та контролю залишків по певному складу;
- додавання нових товарів;
- додавання складських документів;
- можливість інвентаризації товарів на складі;
- можливість додавати товари групами;
- перегляд актуального балансу в момент створення складського документа;
- створення технологічних карток для списання товарів зі складу в процесі надання послуг [45].

Переваги вебзастосунку «Wlaunch»:

- створення корпоративного мобільного застосунку;
- створення вебсайту компанії;
- пряме посилання на вебсторінку для запису клієнтів;
- платіжний сервіс;
- інтеграція з платіжними системами;
- додавання особистих кастомних полів;

- додавання правил оплати;
- відображення історії платежів клієнтів;
- відображення статистики записів;
- налаштування прав користувачів;
- налаштування модулів системи;
- додавання подій компанії [39].

Недоліки вебзастосунку «Wlaunch»:

- відсутній чат підтримки клієнтів;
- відсутній облік реклаमाцій;
- відсутня можливість ведення обліку звернень від клієнтів [39].

1.3.3 Вебзастосунок «RemOnline»

Третім аналогічним програмним продуктом є вебзастосунок «RemOnline». «RemOnline» – це вебзастосунок, який спрощує облік та керування бізнесом для компаній у різних сферах діяльності (рис. 1.6) [46].

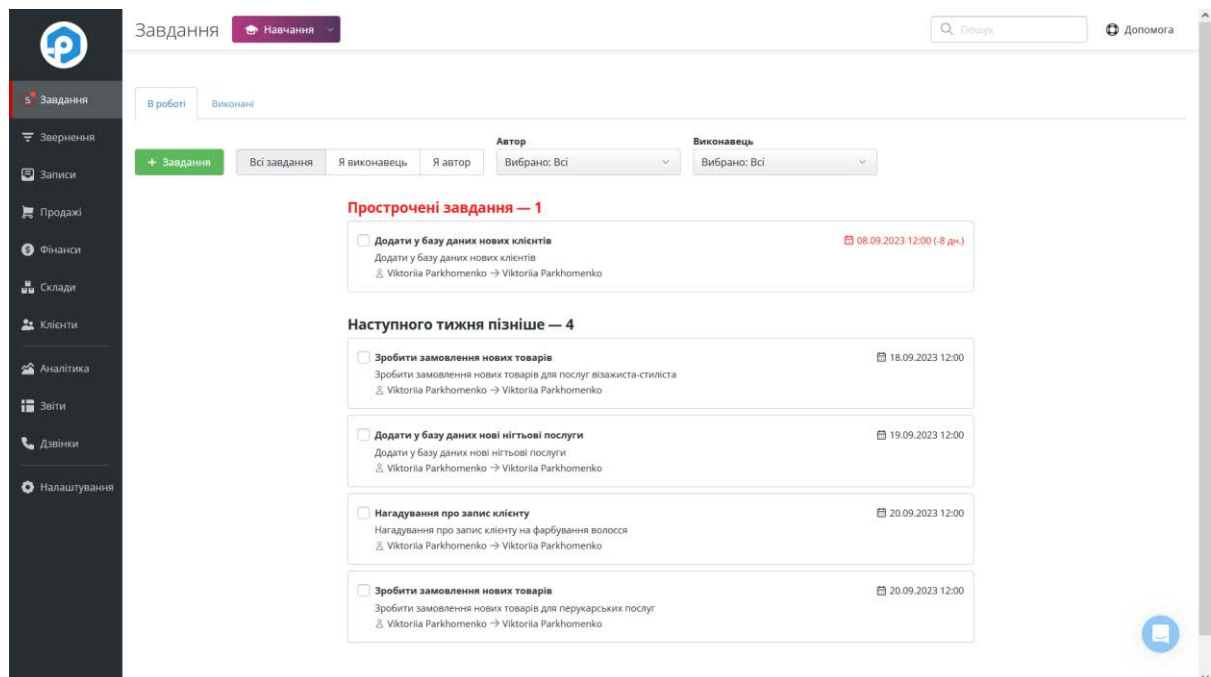


Рисунок 1.6 – Інтерфейс головної сторінки вебзастосунку «RemOnline»

Функціональні можливості вебзастосунку «RemOnline» для салону краси:

- створення нових завдань для співробітників;
- перегляд виконаних завдань;
- відображення стрічки подій завдання;
- додавання документів, зображень або файлу до завдання;
- ведення обліку звернень від клієнтів;
- відображення стрічки подій звернення;
- групові дії з декількома зверненнями;
- бейджі в таблиці звернень;
- додавання записів клієнтів у таблицю;
- відображення стрічки подій запису;
- ведення планувальника;
- можливість застосування фільтрів у розділах вебзастосунку;
- керування продажами товарів компанії;
- ведення обліку руху грошових коштів компанії;
- додавання віртуальних кас;
- автоматичний розрахунок заробітної плати та вартості послуг;
- нарахування заробітної плати по співробітникам;
- редагування касових операцій;
- ведення обліку розрахунків із клієнтами та постачальниками;
- створення рахунків клієнтам;
- оформлення повернень клієнтам із продажу або замовлень;
- можливість автоматизації складських процесів;
- додавання категорій товарів;
- додавання нового товару на склад;
- експорт та імпорт товарів на склад;
- перегляд інформації про товар у діалоговому вікні товару;
- ведення бази клієнтів;
- встановлення персональних знижок у профілі клієнта;

- відображення ключових показників компанії;
- відображення аналітичного звіту діяльності компанії;
- перегляд аналізу асортименту компанії;
- перегляд звітів за розділами вебзастосунку [46].

Вебсторінки додавання нового клієнта та розділу звернень клієнтів вебзастосунку «RemOnline» наведено нижче на рисунках 1.7 та 1.8.

Рисунок 1.7 – Вебсторінка додавання нового клієнта вебзастосунку «RemOnline»

Зверне...	Локація	Статус	Строк статусу	Кінцевий термін	Створено	Менеджер	Клієнт	Рекламна кампанія	Результат
001	Моя локація	Ново записана	5 дн.	22 верес. 2023 12...	16 верес. 2023 20:06	Viktoria Parkhomenko	AnnaCosmetics	Інтернет	
002	Моя локація	В обробі	4 дн.	12 верес. 2023 12...	16 верес. 2023 20:07	Viktoria Parkhomenko	BeautyCompany	Зовнішня реклама	
003	Моя локація	Успішний тегат	—	18 верес. 2023 12...	16 верес. 2023 20:10	Viktoria Parkhomenko	Magia	Знайомі	Замовлення A001
004	Моя локація	Ново записана	2 дн.	19 верес. 2023 12...	16 верес. 2023 20:10	Viktoria Parkhomenko	Ceagri	Соціальні мережі	
005	Моя локація	Анонім	—	26 верес. 2023 12...	16 верес. 2023 20:19	Viktoria Parkhomenko	Юлія	Соціальні мережі	
006	Моя локація	Повторне звернення	—	29 верес. 2023 12...	16 верес. 2023 20:20	Viktoria Parkhomenko	AnnaCosmetics	Соціальні мережі	
007	Моя локація	Ново записана	8 дн.	25 верес. 2023 12...	16 верес. 2023 20:20	Viktoria Parkhomenko	AnnaCosmetics	Інтернет	
008	Моя локація	Успішний тегат	—	19 верес. 2023 12...	16 верес. 2023 21:06	Viktoria Parkhomenko	Юлія	Інтернет	Продаж A001

Рисунок 1.8 – Вебсторінка звернень клієнтів вебзастосунку «RemOnline»

Переваги вебзастосунку «RemOnline»:

- наявність мобільного застосунку для керівника салону краси;
- наявність мобільного застосунку для роботи з замовленнями;
- налаштування розділів вебзастосунку;
- відображення таблиці вхідних та вихідних дзвінків;
- інтеграції зі сторонніми сервісами;
- присутній чат підтримки клієнтів [46].

Недоліки вебзастосунку «RemOnline»:

- платність;
- відсутній облік реклаमाцій;
- відсутні технологічні картки послуг;
- відсутня можливість ведення абонементів та сертифікатів;
- відсутня можливість додавання подій компанії;
- відсутня можливість налаштування форм опитувань [46].

Вебсторінки таблиці записів клієнтів на послуги та планувальника вебзастосунку «RemOnline» наведено нижче на рисунках 1.9 та 1.10.

Запис №	Статус	Створено	Клієнт	Тривалість	Ресурс	Ціна	Тип запису	Реклам...	Найменування посл...	Зона ...
A001	Очікує підтвердження	Viktorlia Parkhomenko 16 верес. 2023 20:19	Марія +38 (095) 678-31-57	1 год 0 хв	Кабінет 2	350 ₴	Перший запис	Знайомі	Стрижка жіноча	Волосся
A002	В роботі	Viktorlia Parkhomenko 16 верес. 2023 21:20	Юлія +38 (095) 385-76-59	1 год 30 хв	Кабінет 1	315 ₴	Перший запис	Знайомі	Вечірній візаж	Обличчя
A003	Новий запис	Viktorlia Parkhomenko 16 верес. 2023 21:21	Юлія +38 (095) 385-76-59	1 год 20 хв	Кабінет 1	250 ₴	Перший запис	Знайомі	Комплексний манікюр	Руки
A004	Скасування запису	Viktorlia Parkhomenko 16 верес. 2023 21:37	Сергій +38 (095) 245-68-76	1 год 0 хв	Кабінет 1	0 ₴	Повторний запис	Знайомі	Стрижка чоловіча	Волосся
A005	Виконано	Viktorlia Parkhomenko 16 верес. 2023 21:38	AnnaCosmetics +38 (095) 375-67-34	1 год 30 хв	Кабінет 2	350 ₴	Повторний запис	Інтернет	Комплексний педикюр	Ноги
A006	Закрито	Viktorlia Parkhomenko 16 верес. 2023 21:43	Марія +38 (095) 678-31-57	1 год 50 хв	Кабінет 2	360 ₴	Повторний запис		Вечірній візаж	Обличчя

Всього — 6

BeautyStudio (ID 126230)
RemOnline © 2012-2023 | support@remonline.ua | Українська

Рисунок 1.9 – Вебсторінка записів клієнтів на послуги вебзастосунку «RemOnline»

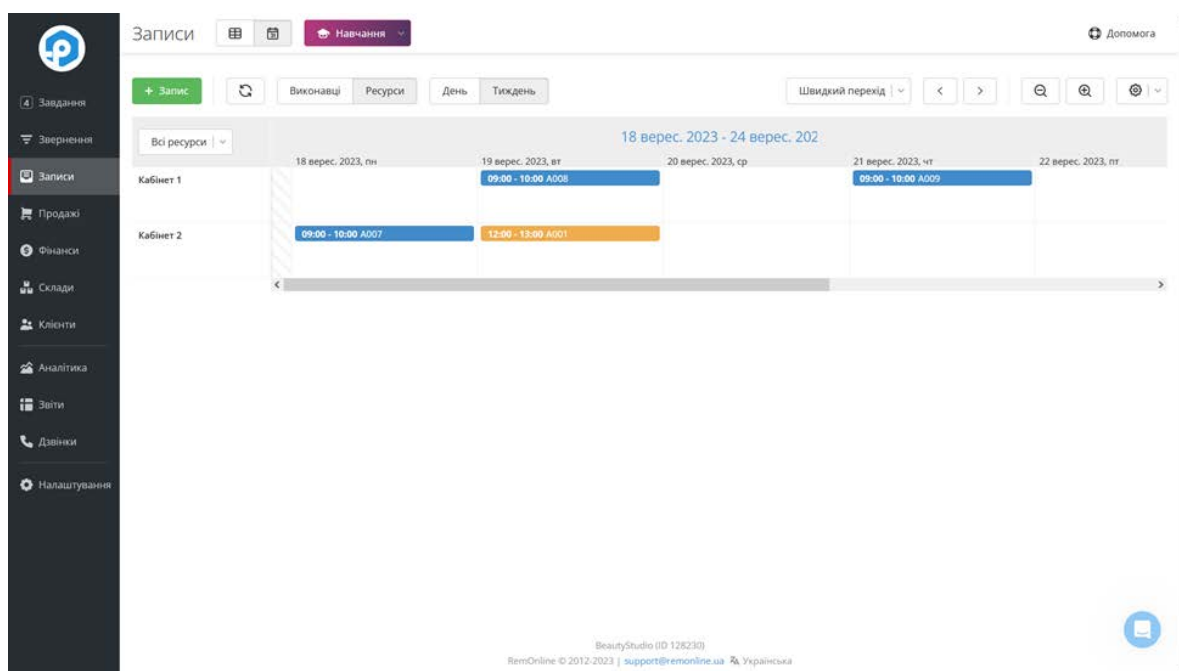


Рисунок 1.10 – Вебсторінка планувальника вебзастосунку «RemOnline»

1.3.4 Порівняння досліджених вебзастосунків

Вище у пунктах були описані та досліджені вебзастосунки, що дозволяють спрощувати облік та автоматизують всі бізнес-процеси у салоні краси. Досліджені вебзастосунки мають дуже багато переваг, але основним недоліком є платність.

Перевагами досліджених вебзастосунків є можливість налаштування розділів системи, інтеграції з іншими сервісами, відображення ключових показників компанії, наявність аналітичних звітів, чат підтримки клієнтів, встановлення персональних знижок у профілі клієнта, створення техкарт за послугами, додавання подій компанії, наявність модуля зарплати, облік рекламацій, додавання віртуальних кас та можливість ведення обліку звернень від клієнтів. Також перевагою усіх досліджених вебзастосунків є адаптивність вебдизайну.

Порівняння досліджених клієнт-серверних застосунків для салонів краси, які були описані та розглянуті вище наведено в таблиці 1.1.

Таблиця 1.1 – Порівняння досліджених клієнт-серверних застосунків для салонів краси

Критерій порівняння	«RemOnline»	«Wlaunch»	«CleverBOX: CRM»
Планувальник	+	+	+
База знань	+	+	+
Менеджер завдань	+	+	+
Чат підтримки клієнтів	+	—	+
Облік клієнтів	+	+	+
Облік товарів	+	+	+
Облік послуг	+	+	+
Складський облік	+	+	+
Контроль залишків товарів	+	+	+
Технологічні картки послуг	—	+	+
Фінансовий облік	+	+	+
База постачальників товарів	+	+	+
Аналітичні звіти	+	+	+
Облік клієнтських абонементів та сертифікатів	—	+—	+
Додавання одиниць виміру для товарів компанії	+	+	+
Облік звернень клієнтів	+	—	+
Додавання подій компанії	—	+	—
Зберігання файлів	—	—	+
Облік рекламацій	—	—	+
Налаштування форм опитувань та відображення результатів проведення опитувань	—	+	—
Модуль зарплати	+	+	+
Повідомлення та нагадування клієнтам	+	+	+
База співробітників	+	+	+
Налаштування розділів системи	+	+	+
Журнал дзвінків	+	—	+
Ключові показники компанії	+	+	+

1.4 Постановка завдань дипломної магістерської роботи

Користувачами вебзастосунку є керівник салону, адміністратор салону та спеціаліст салону краси. Відповідно до ролі користувача у салоні краси керівник або адміністратор при додаванні співробітника вказує розмежування прав доступу. На основі проведеного дослідження аналогічних програмних рішень були сформовані функціональні можливості, які повинен надавати вебзастосунок для керування бізнес-процесами у салоні краси:

– аутентифікація та надання співробітнику після авторизації функцій для роботи відповідно до його посади у компанії;

- ведення клієнтської бази компанії;
- ведення бази завдань співробітників;
- можливість робити позначку, що завдання було виконане;
- ведення бази звернень від клієнтів;
- ведення бази записів клієнтів на послуги;
- ведення бази візитів клієнтів у календарі записів;
- ведення бази співробітників компанії;
- налаштування переліку можливостей співробітників відповідно до посади у компанії;

- ведення бази продажів товарів або послуг компанії;
- ведення бази фінансових операцій компанії;
- ведення бази товарів компанії;
- ведення бази послуг компанії;
- ведення бази списаних витратних матеріалів;
- ведення бази клієнтських сертифікатів та абонементів;
- ведення бази знижок для клієнтів;
- ведення бази поставчань нових товарів;
- ведення бази замовлень поставачальникам товарів;
- ведення бази записів повернень товарів поставачальникам;
- ведення бази поставачальників товарів;

– нарахування заробітної плати по співробітниках.

1.5 Висновки до першого розділу

У першому розділі досліджено предметну область та аналогічні вебзастосунки для керування бізнес-процесами у салоні краси. Було сформовано завдання, які будуть вирішуватися при виконанні дипломної магістерської роботи.

Провівши дослідження аналогічних вебзастосунків можна зробити висновок, що метою кожного вебзастосунку для керування бізнес-процесами у салоні краси є автоматизація щоденних завдань співробітників салону краси та спрощення ведення обліку бізнес-процесів. Користувачами у досліджених вебзастосунків виступають керівник салону, адміністратор салону, візажист, стиліст, перукар тощо.

У ході дослідження аналогічних вебзастосунків було виявлено основні функціональні можливості, які повинен надавати програмний продукт для керування бізнес-процесами у салоні краси:

- ведення обліку клієнтів;
- ведення бази послуг салону краси;
- ведення журналу записів клієнтів на послуги;
- ведення обліку завдань;
- ведення обліку товарів;
- ведення складського обліку;
- проводити контроль залишків товарів;
- ведення фінансового обліку;
- ведення обліку абонементів та сертифікатів;
- ведення обліку звернень від клієнтів;
- ведення обліку продажів власних товарів компанії.

2 ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

У розділі описано обрані засоби для розробки інтерфейсної та серверної частини вебзастосунку, а також засоби взаємодії з базою даних. Наведено порівняння обраної мови з іншими мовами програмування, редакторів коду, засобів для розробки інтерфейсної частини, програмних каркасів для розробки серверної частини та нереляційних документо-орієнтованих СКБД.

2.1 Вибір клієнтської та серверної мови програмування

Для реалізації клієнтської та серверної частини вебзастосунку було обрано мову програмування JS, оскільки обрана мова здатна вирішувати різні завдання та надає велику кількість програмних каркасів для розробки вебзастосунків.

Кожний вебзастосунок складається з клієнтської та серверної частини. Клієнтська частина (frontend) – це частина вебзастосунку, яка завантажується з сервера і виконується на клієнті (веббраузері). Серверна частина (backend) – це частина вебзастосунку, яка працює на сервері та виконує надання даних клієнту, а також обробки запитів, що надходять з боку клієнта [47].

JS (JavaScript) – це мультипарадигмальна інтерпретована сценарна мова програмування, яка використовується для надання вебсторінкам можливості взаємодії з користувачем (інтерактивності) шляхом розробки програмних сценаріїв, а також для динамічного оновлення вмісту вебсторінки. JS є реалізацією специфікації ECMAScript (стандарт ECMA-262). Зазвичай JS використовується при розробці вебзастосунків, тобто для програмування застосунків, які виконуються у веббраузері. Однак JS використовується також для розробки серверних програмних продуктів (платформа Node.js), написання ігор (Unity), створення сценаріїв автоматизації тестування та інше [48].

JS є мовою з динамічною типізацією, тобто під час створення змінної її тип не вказується явно, але змінна зв'язується з типом при присвоєнні значення. При зміні значення змінної може змінитися тип даних змінної. Кожна змінна має тип даних, але в цьому випадку не вказується цей тип явно, як це відбувається в інших мовах зі статичною типізацією [48].

Мова JS є також мовою написання програмних сценаріїв для вебпроектів, що обробляється інтерпретатором окремими інструкціями (командами). Кожна команда JS зазвичай пишеться в окремому рядку та в більшості випадків завершується крапкою з комою [49].

Варто відзначити, що мова програмування JS розрізняє великі та малі літери, тобто є реєстрозалежною мовою програмування [49].

У кожній мові програмування існують стандарти кодування, що регламентують особливості оформлення коду, порядок використання синтаксичних конструкцій, що допускають неоднозначності тощо. Найбільш поширеними стандартами кодування для вихідного коду мовою програмування JS є Google JavaScript Style Guide та Airbnb JavaScript Style Guide [49].

JS використовується для роботи зі стилями та елементами вебсторінки, з введеними даними користувача, сервером, базою даних та локальним сховищем [49].

Перелік завдань для яких використовується мова програмування JS:

- створення інтерактивних елементів;
- взаємодія з сервером та базою даних;
- розробка інтерфейсної частини вебзастосунків використовуючи React, Vue та Angular;
- розробка серверних програмних продуктів на платформі Node.js;
- розробка мобільних застосунків використовуючи React Native або Ionic;
- розробка десктопних програм на фреймворку Electron [49].

Переваги мови програмування JS:

- наявність функцій для створення модальних вікон;
- присутній тип даних BigInt для вирішення завдань пов'язаних з BigData або криптографією;
- присутні спеціалізовані об'єкти для роботи з датою, математичними функціями, масивами та інше;
- функціональний підхід розробки;
- менші витрати часу на розробку та налагодження [49].

Порівняння обраної мови з іншими мовами програмування для розробки вебзастосунків наведено нижче в таблиці 2.1.

Таблиця 2.1 – Порівняння мов програмування для розробки вебзастосунків

Критерій порівняння	JS	C#	Python	PHP
Рішення різних завдань	++	++	++	–
Мультипарадигмальна мова	+	++	+	+
Інтерпретована мова	+	–	+	+
Динамічна типізація	+	–	+	+
Зручний синтаксис	+	+	+–	+–
Реєстрозалежна мова	+	+	+	+
Високорівнева мова	+	+	+	+
Наявність менеджера пакетів	+	+	+	+
Прототип-орієнтований стиль	+	–	–	–
Велика кількість програмних каркасів для розробки вебзастосунків	++	+	++	++
Відносно менші витрати часу	+	+–	+	+–
Висока швидкість виконання	+	+	–	+

2.2 Вибір середовища розробки вебзастосунку

Середовищем для розробки вебзастосунку було обрано редактор коду Visual Studio Code, оскільки обране середовище розробки є безкоштовним та дозволяє розроблювати вебзастосунки мовою програмування JS. Також

критерієм вибору даного текстового редактора є те, що він надає можливість встановлювати доповнення, які розширюють можливості середовища та допомагають пришвидшити розробку вебзастосунку.

Редактор коду – це програма, яка має вбудовані функції для пришвидшення розробки програмних продуктів. Метою редакторів коду є створення та зміна файлів із програмним кодом [50].

Основні функції редакторів коду:

- перевірка синтаксису згідно з вимогами мови;
- автоматична генерація програмного коду;
- підтримка різних мов програмування;
- автодоповнення програмного коду;
- діагностика помилок [50].

Visual Studio Code – це редактор вихідного програмного коду, який дозволяє розробляти вебзастосунки та хмарні застосунки. Visual Studio Code підтримує JS, TypeScript та Node.js, а також має багато розширень для інших мов програмування та середовищ виконання (.NET, Unity) [51].

Функціональні можливості редактору коду Visual Studio Code:

- форматування вихідного коду;
- підтримка декількох курсорів для одночасного редагування коду;
- автодоповнення програмного коду;
- встановлення розширень;
- вбудована підтримка рефакторингу;
- вбудований налагоджувач;
- вбудована підтримка Emmet;
- функція Workspace Trust;
- додавання папок до робочої області;
- вбудована підтримка інтеграції з GitHub;
- наявність карти програмного коду;
- можливість розділення робочої області;
- вбудований інтегрований термінал [52].

Порівняння редакторів коду Visual Studio Code та Atom наведено нижче в таблиці 2.2.

Таблиця 2.2 – Порівняння текстових редакторів для розробки вебзастосунку

Критерій порівняння	Visual Studio Code	Atom
Кодування багатьма мовами	+	+
Перевірка правильності написання коду	+	+
Автоматична генерація каркаса проєкту або окремих фрагментів програмного коду	+	+
Вбудована підтримка Git	+	+
Вбудований налагоджувач для Node.js	+	+
Встановлення розширень	+	+
Файловий менеджер	+	+
Діагностика помилок	+	+
Карта програмного коду	+	+
Розділення робочої області	+	+
Підтримка декількох курсорів	+	+
Підтримка IntelliSense	+	+
Вбудований інтерфейс командного рядка	+	+
Підтримка Emmet	+	+
Можливість роботи з базами даних	+	+
Високий рівень безпеки	+	—
Можливість роботи з декількома проєктами одночасно	+	—
Наявність командного рядка для швидкого виконання різних команд	+	+
Рефакторинг	+	+
Інтегрований термінал	+	+
Можливість тестування	+	+

2.3 Вибір засобів розробки користувацької частини вебзастосунку

Для розробки інтерфейсної частини вебзастосунку було обрано використовувати наступні засоби:

- мову розмітки елементів вебдокументів HTML;
- мову опису стилів оформлення елементів вебдокументів CSS;
- CSS препроцесор Sass;
- бібліотеку елементів інтерфейсу користувача React;
- бібліотеку компонентів інтерфейсу користувача Material UI;
- бібліотеку маршрутизації вебсторінок React Router;
- бібліотеку керування станами вебзастосунку Redux.

2.3.1 Мова розмітки вебдокументів HTML

HTML (мова опису гіпертекстових вебдокументів, HyperText Markup Language) – це мова, що визначає структуру та вміст сторінки. Мова розмітки містить набір правил для структурування (розмітки) текстової інформації, додавання зображень, створення таблиць, форм, списків тощо [53].

Основою мови HTML є тег, який виступає елементом гіпертекстової розмітки. Деякі теги використовуються для створення структури HTML документа, а інші для форматування та створення елементів на вебсторінці. Документ, який розмічений за допомогою HTML, інтерпретується браузером, в результаті чого користувач баче не вихідний код з елементами розмітки, а остаточний результат обробки, тобто вебсторінку [53].

2.3.2 Препроцесор Sass

CSS (Cascading Style Sheets) – це мова визначення правил стилізації елементів до вебдокументів HTML. CSS використовується для форматування вмісту вебсторінок та містить загальні властивості зовнішнього вигляду

документа, тобто розмір шрифту, колір тексту, інтервал між рядками, розміщення зображень, окремих блоків та інших елементів. Формат розширення файлу .css призначений для того, щоб розділити опис логічної структури вебсторінки з розміткою від її зовнішнього вигляду. CSS дозволяє значно спростити розробку та підтримку вебсайтів завдяки своїм основним характеристикам [54].

Препроцесор CSS – це в основному мова сценаріїв, яка розширює CSS, а потім компілює його у звичайний CSS. Sass (Syntactically Awesome Style Sheet) – це препроцесор CSS, скомпільований у звичайний CSS, що дозволяє використовувати такі функції, як змінні, вкладені правила та міксини у таблицях стилів. Sass може допомогти зробити CSS більш організованим та легшим для обслуговування, а також полегшити створення складних стилів. Sass дозволяє створювати багато таблиць стилів для різних компонентів вебсторінки та полегшує підтримку стилів [55].

Переваги використання Sass:

- вкладеність стилів;
- використання змінних;
- створення міксинів;
- вбудовані функції для виконання обчислень та маніпулювання значеннями;
- покращена організація коду [55].

2.3.3 Бібліотека елементів інтерфейсу користувача React

React – це бібліотека JS для створення елементів інтерфейсу користувача. Бібліотека є мультиплатформною, тобто розмітку можна рендерити на сервері (Next.js), писати мобільні (React Native) або десктопні (Electron) застосунки. Бібліотека може суттєво пришвидшити процес розробки вебзастосунку шляхом застосування принципу багаторазового використання простих компонентів [56].

Create React App – це інструмент для створення React застосунку останньої версії з автоматичним налаштуванням програмного середовища для використання мови програмування JS та синтаксису JSX [56].

React не вимагає від розробників дотримуватись якоїсь чіткої файлової структури у своїх вебзастосунках, тобто надає можливість використовувати індивідуальну файлову структуру для кожного проєкту. Одним з найбільш популярних способів структуризації файлової системи React застосунків є розбиття проєкту на логічно відокремленні директорії. Файлова структура корпоративних застосунків передбачає окремі директорії для стилізації, компонентів, сторінок, даних, контексту та хуків [56].

JSX – це розширення мови програмування JS, яке дозволяє використовувати синтаксис HTML всередині JS коду. React також дозволяє розробляти застосунки без синтаксису JSX, але його використання суттєво спрощує розробку структури користувацького інтерфейсу [56].

Інтерфейси вебзастосунків на React розробляються за допомогою компонентів. Компонент – це одиниця або будівельний блок для побудови інтерфейсів на React. Використання компонентів при побудові інтерфейсів дозволяє розділити застосунок на незалежні частини, які можна розробляти та проєктувати окремо [56].

Компоненти можуть бути розроблені за допомогою класу або функції. Сьогодні найбільш популярним методом розробки компонентів є функціональний підхід, але в деяких задачах повністю відмовитись від класового підходу неможливо [56].

Методами для роботи зі станами життєвого циклу React всередині функціональних компонентів є хуки. До появи React версії 16.8 у функціональних компонентів не було інструментів для роботи зі станами, рефами та ефектами, що значно зменшувало їх потенціал для використання, в той час, коли у класових компонентів існував власний життєвий цикл, всередині якого можна було контролювати стан елементів та миттєво відображати їхні зміни в інтерфейсі [57].

Основою вебдокументів є теги, тобто кожен тег можна представити об'єктом DOM дерева. В React об'єктами DOM можна вважати елементи, які розробляються за допомогою синтаксису JSX. React використовує віртуальну модель DOM, щоб легше взаємодіяти з елементами DOM. Віртуальна модель DOM представлена у вигляді об'єкта JS, який можна змінювати та часто оновлювати [56].

Процес рендерингу – це внутрішні дії React пов'язані з формуванням віртуального дерева компонентів. Процес рендерингу в React не перерисовує DOM дерево, оскільки його задача сформувати віртуальне DOM дерево, яке потім порівнюється з попереднім та якщо вони нічим не відрізняються, тоді оновлення не відбувається [56].

Переваги бібліотеки React:

- бібліотека використовує компонентну архітектуру;
- декларативний стиль програмування;
- менше шаблонного коду для написання та компактність коду;
- бібліотека забезпечує високу продуктивність завдяки своїй віртуальній моделі DOM;
- надає розширення для перевірки ієрархії компонентів React в інструментах розробника веббраузера Chrome;
- розширена екосистема надає розробникам різноманітні бібліотеки та компоненти;
- використання бібліотек для роботи з формами;
- обробка неконтрольованих та контрольованих елементів форм;
- дозволяє працювати із запитами до сервера використовуючи вбудований об'єкт `fetch` або бібліотеку `axios` для виконання HTTP запитів;
- надає інструмент `React Query` для керування станом та кешування даних;
- має можливість обробки помилок у запитах;
- дозволяє застосовувати принципи повторного використання коду;
- зручність тестування ізоляції компонента;

- має можливість вбудовувати стилі безпосередньо в компоненти використовуючи JS;
- надає можливість використання хуків для оптимізації рендерингу компонентів та зменшення навантаження на процесор;
- надає можливість застосування кастомних хуків для спільного використання логіки між компонентами;
- надає можливість використовувати хуки, які контролюють стан елементів та відображають зміни в інтерфейсі;
- використовується синтаксис JSX для спрощення розробки [58].

Порівняння інструментів для розробки графічного інтерфейсу наведено нижче в таблиці 2.3.

Таблиця 2.3 – Порівняння інструментів для розробки інтерфейсу користувача

Критерій порівняння	React	Vue	Angular
Мова програмування JS	+	+	–
Бібліотека	+	–	–
Віртуальна модель DOM	+	+	–
Двостороннє зв'язування даних	–	+	+
Швидке освоєння інструменту	+	++	–
Компонентний підхід розробки	+	+	+
Наявність вбудованого інструменту для маршрутизації вебсторінок	–	+	+
Наявність вбудованого інструменту для керування станами вебзастосунку	–	+	+
Розробка середніх вебпроектів	+	+–	++
Наявність документації	+	+	+
Вбудовані стилі компонентів	+	–	–
Робота з REST API	+	+	+
Висока швидкість кодування	+	++	–
Розширена екосистема	+	+	+

2.3.4 Бібліотека компонентів Material UI

Material UI – це бібліотека компонентів інтерфейсу користувача, яка реалізує систему дизайну Material Design від Google. Бібліотека містить повну колекцію готових компонентів, які готові до використання при розробці вебзастосунків на бібліотеці React. Бібліотека також дозволяє використовувати готові компоненти при розробці вебзастосунків на інших JS фреймворках [59].

Перевагою бібліотеки Material UI є можливість налаштування компонентів та зручність інтеграції з серверною частиною вебзастосунку, а також бібліотека постійно оновлюється. Альтернативами бібліотеки Material UI є BlueprintJS та React Bootstrap [59].

2.3.5 Бібліотека маршрутизації вебсторінок React Router

Маршрутизація (роутинг) – це зміна URL адреси в момент переходу між розділами у вебзастосунку. Також завдяки маршрутизації у користувача є можливість відстежувати власну історію користування вебзастосунком [60].

Бібліотека React немає вбудованого інструменту маршрутизації, тому використовується бібліотека React Router. React Router надає набір компонентів для створення маршрутизації, можливість керування історією навігації користувача та відображення різних компонентів в залежності від поточного значення URL в адресному рядку браузера [60].

Для створення маршрутизації потрібно створити компонент Root. У компоненті Root буде описана основна структура вебпроєкту та компоненти, які не повинні змінюватись (header, footer, sidebar) [60].

Бібліотека React Router надає можливість детальної обробки помилок та може відображати код помилки. Для уникнення перезавантаження сторінки при переходах між розділами необхідно всі посилання в ньому реалізовувати за допомогою компонентів Link або NavLink [60].

2.3.6 Менеджер керування станами Redux

Redux – це JS бібліотека, яка реалізує функціональні можливості для керування станом вебзастосунку. Бібліотека Redux використовується в поєднанні з React та її ідеєю є створення централізованого місця управління та зберігання глобального стану застосунку. Менеджер Redux використовується при розробці великих вебпроектів, які мають багато функціональних можливостей [61].

Менеджер Redux використовується для наступних завдань:

- підвищення ефективності роботи вебзастосунку;
- керування станами вебзастосунків, які містять великий об'єм даних;
- заміна хука `useState`;
- підвищення масштабованості вебзастосунку;
- унеможливлення виникнення помилок в об'єктах станів;
- контролю історії станів [61].

Шаблон життєвого циклу Redux передбачає сховище, функцію та редуктор (рис. 2.1). Сховище зберігає стани вебзастосунку, надає доступ до стану та дозволяє оновлювати стан. Функція описує дії над даними, які необхідно виконати. Редуктор встановлює стан об'єкта на основі його попереднього значення та функції. Результатом роботи редуктора є новий об'єкт стану з актуальними даними. Даний шаблон впроваджує передбачувану логіку роботи вебзастосунку та сприяє більшій упорядкованості даних [61].

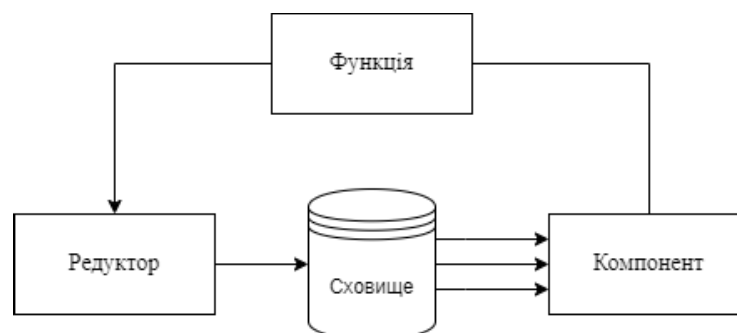


Рисунок 2.1 – Потoki даних в менеджері станів Redux

2.4 Вибір засобів розробки серверної частини вебзастосунку

Для розробки серверної частини вебзастосунку було обрано використовувати наступні інструменти:

- архітектурний стиль для передачі даних між клієнтом та сервером REST API;
- серверне середовище виконання Node.js;
- фреймворк Express.js;
- нереляційну СКБД MongoDB;
- бібліотеку для взаємодії з базою даних Mongoose.

Використання REST API дозволяє взаємодіяти клієнту з сервером використовуючи HTTP методи, тобто CRUD операції. До CRUD операцій належать чотири методи для взаємодії з ресурсами REST API, такі як створення нового ресурсу, отримання певного ресурсу або набору ресурсів, оновлення ресурсу та видалення ресурсу [62].

2.4.1 Платформа розробки Node.js

Node.js – це однопотокове серверне програмне середовище виконання JS, яке дозволяє запускати застосунки написані мовою програмування JS та працює на основі механізму виконання V8. Node.js має подійно-орієнтовану архітектуру та асинхронну взаємодію з неблокуючим введенням та виведенням [63].

Переваги середовища виконання Node.js:

- має широку екосистему пакетів та модулів (NPM);
- обробляє велику кількість запитів одночасно;
- асинхронний та керований подіями;
- висока швидкість виконання коду завдяки механізму Chrome V8;
- використовує однопотокову парадигму з циклом подій;
- відсутність буферизації даних;

- один формат даних для клієнта та сервера (JSON);
- застосування модулів при розробці програм;
- надає можливість масштабованості програм вертикально та горизонтально [63].

Недоліки середовища виконання Node.js:

- значні зміни коду внаслідок нестабільності API;
- багато бібліотек NPM низької якості, неповні та погано описані;
- важко підтримувати код через асинхронний підхід;
- під час виконання важких обчислювальних завдань продуктивність знижується, оскільки Node.js однопоточковий та керований подіями [63].

NPM (Node Package Manager) є менеджером пакетів Node.js, який дозволяє керувати модулями та залежностями при розробці застосунків. Використовуючи даний менеджер можна керувати встановленням, оновленням та видаленням бібліотек або пакетів [64].

NVM (Node Version Manager) дозволяє користувачеві підтримувати кілька версій Node.js на локальній машині та змінювати активну версію залежно від поточних вимог. NVM використовується для розробки застосунків Node.js у кількох версіях [65].

Модулі в Node.js являють собою набір функцій, які можна включити в програму для досягнення певних функцій. Модулі – це інкапсульовані блоки коду, які взаємодіють із зовнішньою програмою на основі відповідних можливостей. Основні модулі Node.js поставляються з набором вбудованих модулів, тобто мають попередньо визначені функції для виконання конкретних завдань. Локальні модулі доступні лише локально та створюються програмістом. Сторонні модулі встановлюються через NPM [66].

Цикл подій – це техніка для розбиття тривалих операцій на керовані блоки. Node.js сканує робочу чергу та запускає нові завдання за мілісекунди. Node.js направляє завдання до стека викликів та виконує завдання. Цикл подій дозволяє Node.js працювати в багатозадачному режимі, тобто додаткові

завдання можуть виконуватися без необхідності очікування завершення першого [67].

У Node.js є вбудований модуль HTTP для створення HTTP запитів та передачі даних через HTTP. Для створення сервера необхідно підключити модуль HTTP та на отриманому об'єкті викликати метод `createServer()` [68].

2.4.2 Серверний фреймворк Express.js

Express.js – це JS фреймворк для розробки серверної частини вебзастосунку, який використовується для створення серверних служб. Фреймворк Express.js побудований на фреймворку Connect. Призначенням фреймворку Express.js є маршрутизація URL адрес та виклики функцій проміжної обробки (middleware) для внесення змін до об'єктів запитів та відповідей або завершення циклу запит-відповідь [69].

Фреймворк Express.js має наступні переваги:

- надає можливість використовувати різні шаблонізатори;
- надає інструменти для маршрутизації та обслуговування статичних файлів поверх Node.js;
- не створює навантаження на продуктивність Node.js;
- використовує архітектурний шаблон MVC для розробки вебзастосунків;
- налаштовує проміжне програмне забезпечення для обробки HTTP запитів та відповідей;
- гнучкість та можливість адаптації до індивідуальних вимог;
- можна розширювати за допомогою сторонніх модулів;
- інтеграція API;
- швидке освоєння фреймворку [70].

Недоліки фреймворку Express.js:

- розробникам доводиться самотійно шукати рішення для налагодження;

– низький рівень безпеки [71].

Порівняння фреймворків для розробки серверної частини вебзастосунку наведено нижче в таблиці 2.4.

Таблиця 2.4 – Порівняння фреймворків для розробки серверної частини

Критерій порівняння	Express.js	Koa.js	Нapi.js
Високий рівень безпеки	–	–	+
Висока швидкість роботи	+	+	+
Розширюваність	+	+	+
Шаблонізація вебзастосунків	+	+	+
Модульність	+	+	+
Кросплатформність	+	+	+
Механізм маршрутизації	+	+–	+
Обробка запитів	+	+	+
Шаблон архітектури MVC	+	–	+
Аутентифікація користувачів	+	+	+
Швидке освоєння фреймворку	++	+–	+
Розробка середніх вебпроектів	+	+	+
Наявність документації	++	+–	+
Масштабованість	+	++	+

2.5 Вибір СКБД

При розробці вебзастосунку для роботи з даними було обрано використовувати нереляційну СКБД MongoDB. Значною перевагою обраної СКБД є те, що обрана СКБД зберігає великі обсяги неструктурованої інформації та має високу швидкість обробки даних. Також бази даних NoSQL можна масштабувати горизонтально, що дозволяє навантаження розподіляти завдяки додаванню більшої кількості серверів або розподілу даних [72].

Бази даних NoSQL мають гнучкі схеми для організації неструктурованих даних у програмах та дані можуть зберігатися у різних форматах, тобто у документах, графах (мережах), стовпчиках та у вигляді

сховища ключ-значення. Гнучкість у базах даних NoSQL дозволяє створювати документи не визначаючи при цьому структуру заздалегідь, документ може мати власну структуру, синтаксис може відрізнятися у різних базах даних та під час роботи можна додавати у документи нові поля. Велика кількість баз даних NoSQL розроблені спеціально для кластерів, оскільки вони базуються на обробці великих обсягів інформації [73].

Однією з основних відмінностей реляційних SQL та NoSQL полягає в тому, що бази даних NoSQL краще піддаються масштабуванню. Реляційні SQL використовують вертикальне масштабування, що має на меті проводити збільшення навантаження на кожному окремому сервері та підвищуючи потужність процесора, а також обсягу оперативної пам'яті та жорсткого диска. Горизонтальне масштабування у базах даних NoSQL використовує велику кількість серверів для розподілу навантаження [73].

Також важливою відмінністю реляційних SQL від NoSQL є те, що при реляційному підході проектування дані зберігаються у заздалегідь спроектованій структурі, а при нереляційному підході відсутня схема та заздалегідь структура. Використовуючи бази даних NoSQL можна легко змінювати сховище даних з розвитком проекту та полегшувати обробку неоднорідних даних, тобто даних з різними наборами полів [73].

MongoDB – це нереляційна масштабована СКБД, яка підтримує документо-орієнтовану модель даних (рис. 2.2) та надає основні функції для роботи з даними, а також багато додаткових функцій. Серед додаткових функцій є такі, як вторинні індекси, запити по діапазону, сортування, агрегація даних, підтримує колекції даних, часткові індекси, геопросторові індекси, підтримує протокол зберігання великих файлів та метаданих файлів. Функціональність СКБД MongoDB дозволяє розташовувати декілька баз даних на декількох фізичних серверах, а також ці бази даних можуть обмінюватися даними та зберігати цілісність [74].

СКБД MongoDB для зберігання даних використовує схожий формат на JSON, який називається BSON. Формат зберігання даних BSON дозволяє

працювати з даними швидше, але має недолік – дані займають більше місця порівняно з форматом JSON [74].

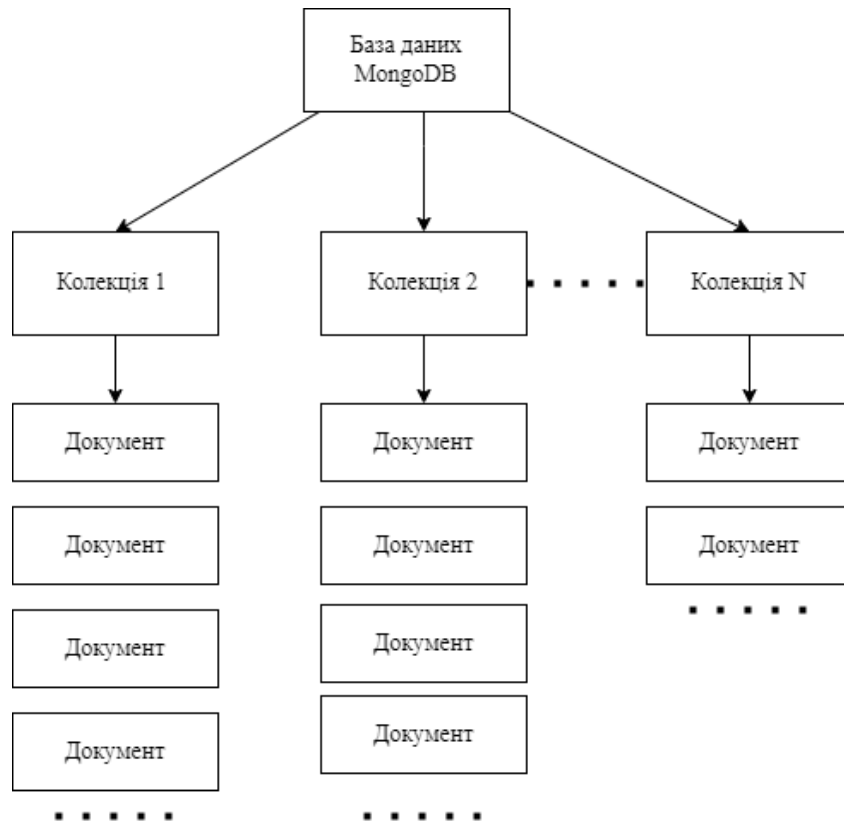


Рисунок 2.2 – Документо-орієнтована модель даних у СКБД MongoDB

СКБД MongoDB використовує власну мову запитів MQL для роботи з документами у колекціях. Мова запитів MQL схожа із мовою програмування JS та надає методи для фільтрування колекції, додавання даних до колекції, оновлення колекції та видалення документів в колекції [74].

У базах даних NoSQL замість вимог ACID розглядається набір властивостей BASE:

- базова доступність – кожен запит гарантовано завершується (успішно або безрезультатно);
- гнучкий стан – стан системи може змінюватися з часом, навіть без введення нових даних, з метою досягнення узгодження даних;
- узгодженість у кінцевому результаті – дані можуть бути деякий час неузгоджені, але приходять в узгодження через деякий час [74].

Набір властивостей BASE був запропонований Еріком Брюєром, автором теореми CAP, згідно з якою в розподілених обчисленнях можна забезпечити лише дві з трьох властивостей: узгодженість даних, доступність або стійкість до розподілу [74].

Переваги СКБД MongoDB:

- вбудована підтримка реплікації даних або копії бази даних;
- вбудована підтримка шардингу або горизонтального розподілення;
- індексація для прискорення операцій вилучення даних із бази даних;
- відсутність складних з'єднань та зрозуміла структура;
- геопросторова індексація;
- наявність спеціальної технології GridFS для зберігання різних документів з різними наборами даних великих розмірів;
- надає фреймворк для агрегації даних на основі концепції конвеєрів обробки даних [74].

Порівняння нереляційних документо-орієнтованих СКБД наведено нижче в таблиці 2.5.

Таблиця 2.5 – Порівняння нереляційних документо-орієнтованих СКБД

Критерій порівняння	MongoDB	CouchDB	Couchbase
Формат зберігання даних	BSON	JSON	JSON
Реплікації	++	+	++
Високий рівень безпеки	++	+	+
Індексування	++	+—	+
Агрегація даних	+	+	+
Мультиплатформність	+	+	+
Швидке освоєння СКБД	+	+	+
Простота використання	++	+	+
Засіб візуалізації даних	+	—	—
Розробка високонавантажених масштабованих систем зберігання даних	++	+—	++

2.6 Вибір бібліотеки для роботи з обраною СКБД

Mongoose є бібліотекою JS для моделювання об'єктів бази даних (ODM), яка дозволяє взаємодіяти між СКБД MongoDB та фреймворком Express.js. ODM – це інструмент, який забезпечує абстракцію для роботи з базами даних. Mongoose надає систему моделювання даних на основі схем, яка дозволяє визначати структуру даних за допомогою схеми [75].

2.7 Висновки до другого розділу

Провівши огляд особливостей обраних інструментів для програмної реалізації вебзастосунку та порівнявши з іншими існуючими засобами, було обрано використовувати при розробці наступні засоби:

- клієнтську та серверну мову програмування JS;
- редактор коду Visual Studio Code;
- мову розмітки елементів вебсайтів HTML;
- мову опису стилів оформлення елементів вебсайтів CSS;
- CSS препроцесор Sass;
- бібліотеку елементів інтерфейсу користувача React;
- бібліотеку компонентів інтерфейсу користувача Material UI;
- бібліотеку маршрутизації вебсторінок React Router;
- бібліотеку керування станами вебзастосунку Redux;
- інструмент для передачі даних між клієнтом та сервером REST API;
- серверне програмне середовище виконання Node.js;
- серверний фреймворк Express.js;
- нереляційну СКБД MongoDB;
- бібліотеку для взаємодії з базою даних Mongoose.

У пунктах розділу було описано особливості та переваги обраних засобів для розробки інтерфейсної та серверної частини вебзастосунку для керування бізнес-процесами у салоні краси.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

У розділі наведені діаграми варіантів використання вебзастосунку, які моделюють бізнес-процеси, тобто описують функціональні вимоги до вебзастосунку. Наведено структуру навігації вебзастосунку та схему обраної програмної архітектури. Також у розділі розглянуто основні архітектурні моменти та описано переваги обраного архітектурного рішення для реалізації вебзастосунку.

3.1 Проєктування сценаріїв використання вебзастосунку

Провівши дослідження аналогічних вебзастосунків для керування бізнес-процесами у салоні краси було сформовано функціональні вимоги, які повинен надавати розроблювальний вебзастосунок. Користувачами у розроблювальному вебзастосунку є керівник салону краси, адміністратор салону краси та спеціаліст салону краси. Нижче перераховані сформовані функціональні вимоги для кожного користувача у вебзастосунку.

Для адміністратора салону краси:

- аутентифікація та надання після авторизації функцій для роботи відповідно до посади у салоні краси;
- пошук існуючого завдання;
- перегляд існуючих завдань;
- додавання нового завдання;
- пошук існуючого клієнта;
- перегляд клієнтів;
- додавання нових клієнтів;
- редагування інформації по обраному клієнту;
- видалення клієнтів;
- пошук звернення від клієнта;
- перегляд звернень клієнтів;

- додавання нового звернення від клієнта;
- перегляд записів клієнтів на послуги;
- пошук записів клієнтів на послуги;
- додавання нового запису клієнта на послугу;
- редагування інформації про запис клієнта на послугу;
- видалення запису клієнта на послугу;
- перегляд візитів клієнтів до салону краси у календарі записів;
- додавання нового візиту клієнта до салону краси у календар записів;
- редагування інформації щодо візиту клієнта до салону краси у календарі записів;
- видалення візиту клієнта до салону краси у календарі записів;
- перегляд співробітників салону краси;
- пошук співробітників салону краси;
- додавання профілю співробітника;
- редагування профілю співробітника;
- видалення профілю співробітника при його звільненні;
- налаштування переліку функціональних можливостей для обраного співробітника;
- перегляд та додавання продажів;
- перегляд фінансових операцій салону краси;
- перегляд товарів;
- пошук товарів;
- додавання нового товару;
- редагування інформації про товар;
- видалення товару;
- перегляд послуг салону краси;
- пошук послуг салону краси;
- додавання нової послуги;
- видалення послуги;
- перегляд постачання нових товарів;

- додавання інформації про постачання нових товарів;
- видалення інформації про постачання нового товару;
- перегляд замовлень постачальникам товарів;
- додавання замовлення товару;
- видалення замовлення товару;
- перегляд постачальників товарів;
- пошук постачальників товарів;
- додавання постачальників товарів;
- видалення постачальника товарів;
- перегляд списаних витратних матеріалів;
- перегляд сертифікатів клієнтів;
- пошук сертифікатів клієнтів;
- додавання сертифіката на послуги або товари;
- видалення сертифіката;
- перегляд абонементів;
- пошук абонементів;
- додавання абонементів;
- видалення абонементів;
- перегляд даних про знижки на послуги.

Для керівника салону краси:

- аутентифікація та надання після авторизації функцій для роботи відповідно до посади у салоні краси;
- пошук існуючого клієнта;
- перегляд клієнтів;
- додавання нових клієнтів;
- редагування клієнтів;
- видалення клієнтів;
- пошук існуючого завдання;
- додавання нового завдання;
- перегляд існуючих завдань;

- редагування існуючих завдань;
- видалення завдань;
- можливість робити позначку, що завдання виконане;
- пошук звернення від клієнта;
- перегляд звернень клієнтів;
- додавання нового звернення від клієнта;
- змінення статусу звернення клієнта;
- редагування звернення клієнта;
- видалення звернення клієнта;
- перегляд записів клієнтів на послуги;
- пошук записів клієнтів на послуги;
- додавання нового запису клієнта на послугу;
- редагування інформації про запис клієнта на послугу;
- видалення запису клієнта на послугу;
- перегляд візитів клієнтів до салону краси у календарі записів;
- додавання нового візиту клієнта до салону краси у календар записів;
- редагування інформації щодо візиту клієнта до салону краси у календарі записів;
- видалення візиту клієнта до салону краси у календарі записів;
- перегляд співробітників салону краси;
- пошук співробітників салону краси;
- додавання профілю співробітника;
- редагування профілю співробітника;
- видалення профілю співробітника при його звільненні;
- налаштування переліку можливостей співробітників відповідно до посади у компанії;
- перегляд продажів клієнтам;
- додавання продажу;
- збереження проведеного продажу клієнту;
- повернення із продажу товару від клієнта на склад;

- редагування продажів;
- видалення продажів;
- перегляд фінансових операцій салону краси;
- додавання таких фінансових операцій, як надходження, переказ коштів між рахунками та витрата коштів;
- видалення внесеної до бази даних фінансової операції;
- редагування фінансових операцій;
- перегляд товарів;
- пошук товарів;
- додавання нового товару;
- редагування інформації про товар;
- видалення товару;
- перегляд послуг салону краси;
- пошук послуг салону краси;
- додавання нової послуги;
- редагування інформації про послугу;
- видалення послуги;
- перегляд списаних витратних матеріалів;
- додавання до бази та видалення із бази списані витратні матеріали;
- перегляд сертифікатів;
- пошук сертифікатів;
- додавання сертифіката на послуги або товари;
- видалення сертифіката;
- редагування інформації про сертифікат;
- перегляд абонементів;
- пошук абонементів;
- додавання абонементів;
- редагування абонементів;
- видалення абонементів;
- перегляд знижок на послуги;

- додавання знижок на послуги;
- видалення знижок на послуги;
- перегляд постачання нових товарів;
- додавання інформації про постачання нових товарів;
- видалення інформації про постачання нового товару;
- перегляд замовлень постачальникам товарів;
- додавання замовлення товару;
- видалення замовлення товару;
- перегляд повернутих товарів постачальникам;
- додавання запису про повернення товару постачальнику;
- видалення запису про повернення товару постачальнику;
- перегляд постачальників товарів;
- пошук постачальників товарів;
- додавання постачальників товарів;
- редагування інформації про постачальника товарів;
- видалення постачальника товарів;
- перегляд нарахованої заробітної плати;
- нарахування заробітної плати по співробітниках.

Для спеціаліста салону краси:

- аутентифікація та надання після авторизації функцій для роботи відповідно до посади у салоні краси;
- пошук клієнта;
- перегляд інформації про клієнта;
- пошук існуючого завдання;
- перегляд існуючих завдань;
- додавання нового завдання;
- перегляд записів клієнтів на послуги;
- пошук записів клієнтів на послуги;
- додавання нового запису клієнта на послугу;
- редагування інформації про запис клієнта на послугу;

- видалення запису клієнта на послугу;
- можливість позначати виконане завдання;
- перегляд візитів клієнтів до салону краси у календарі записів;
- додавання нового візиту клієнта до салону краси у календар записів;
- редагування інформації щодо візиту клієнта до салону краси у календарі записів;
- видалення візиту клієнта до салону краси у календарі записів;
- перегляд та пошук співробітників салону краси;
- пошук та перегляд даних про послугу;
- перегляд даних про сертифікати клієнтів;
- перегляд даних про абонементи клієнтів;
- пошук абонементів;
- перегляд даних про знижки на послуги у салоні краси;
- перегляд нарахованої заробітної плати.

Діаграму взаємодії спеціаліста салону краси із вебзастосунком для керування бізнес-процесами у салоні краси зображено на рисунку 3.1.

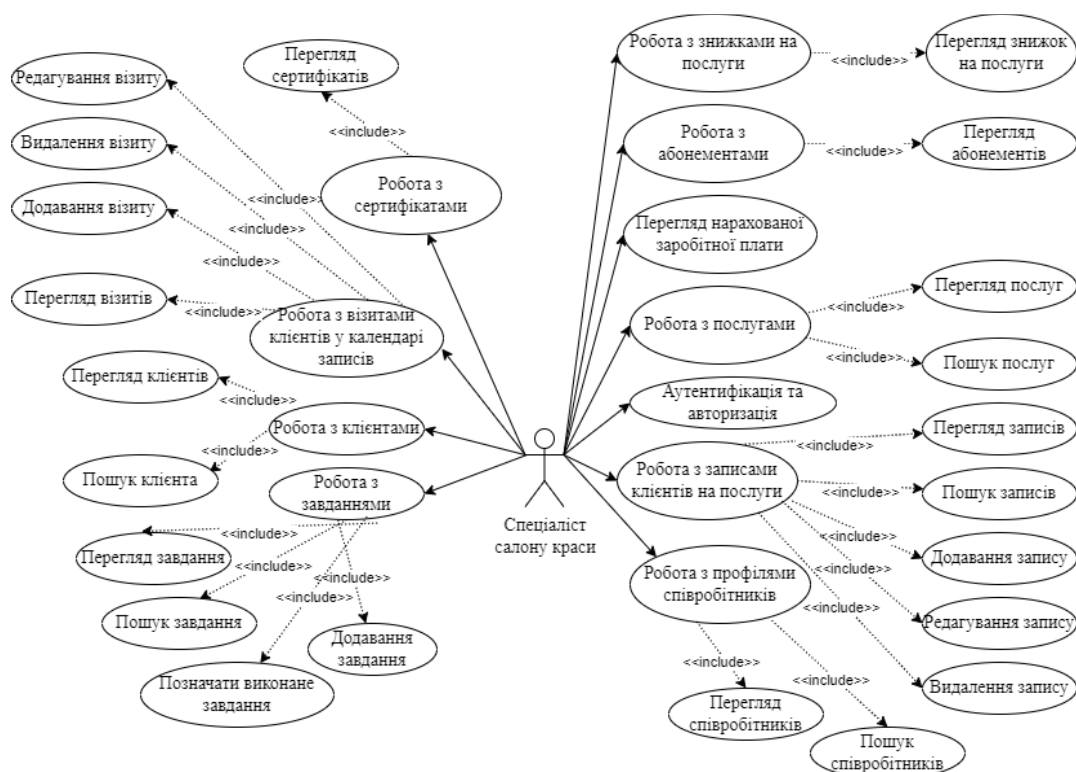


Рисунок 3.1 – Діаграма взаємодії спеціаліста салону краси із вебзастосунком

Діаграму взаємодії адміністратора салону краси із вебзастосунком для керування бізнес-процесами у салоні краси зображено на рисунку 3.2.

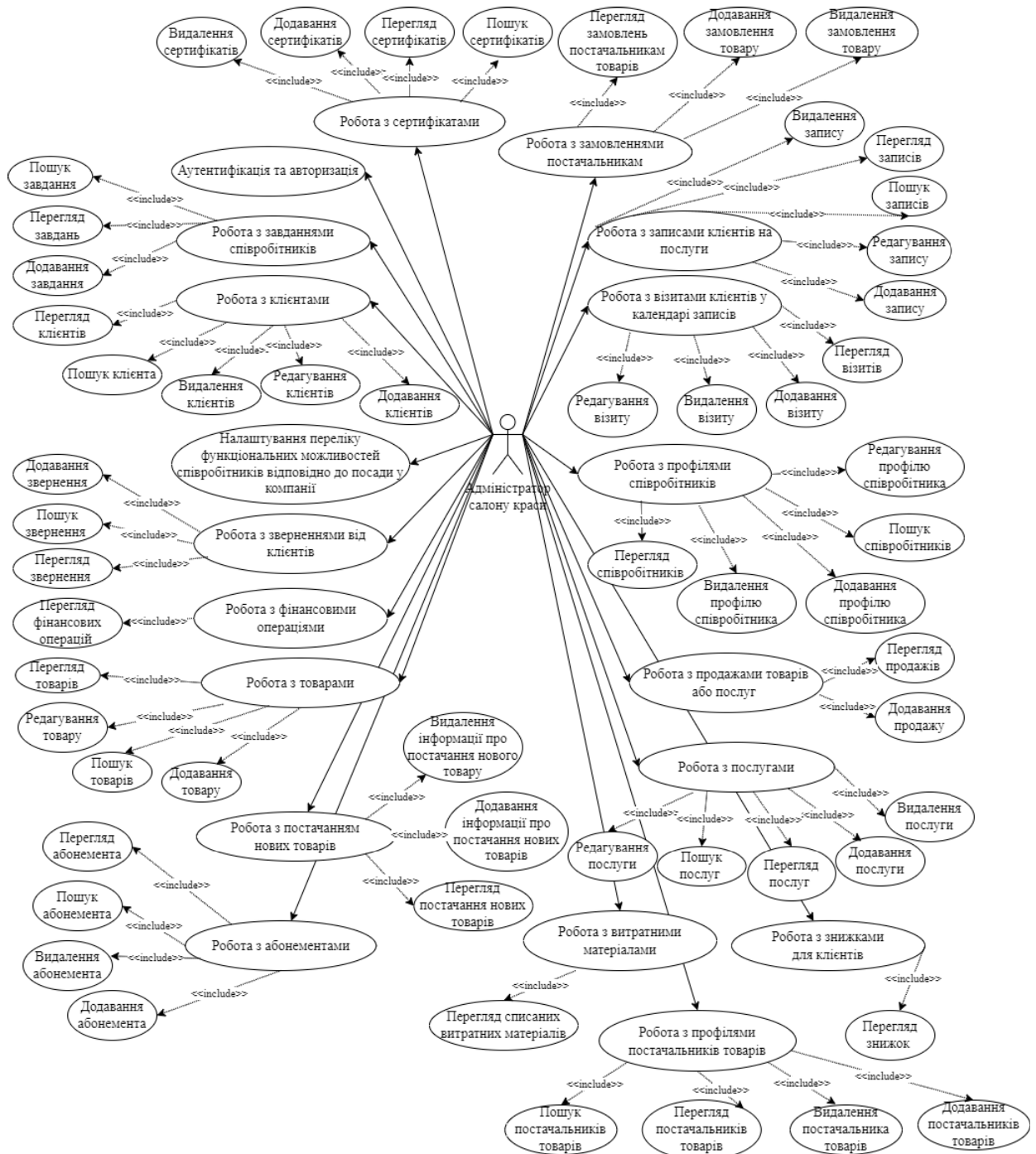


Рисунок 3.2 – Діаграма взаємодії адміністратора салону краси із вебзастосунком

Діаграму взаємодії керівника салону краси із вебзастосунком для керування бізнес-процесами у салоні краси зображено на рисунку 3.3.

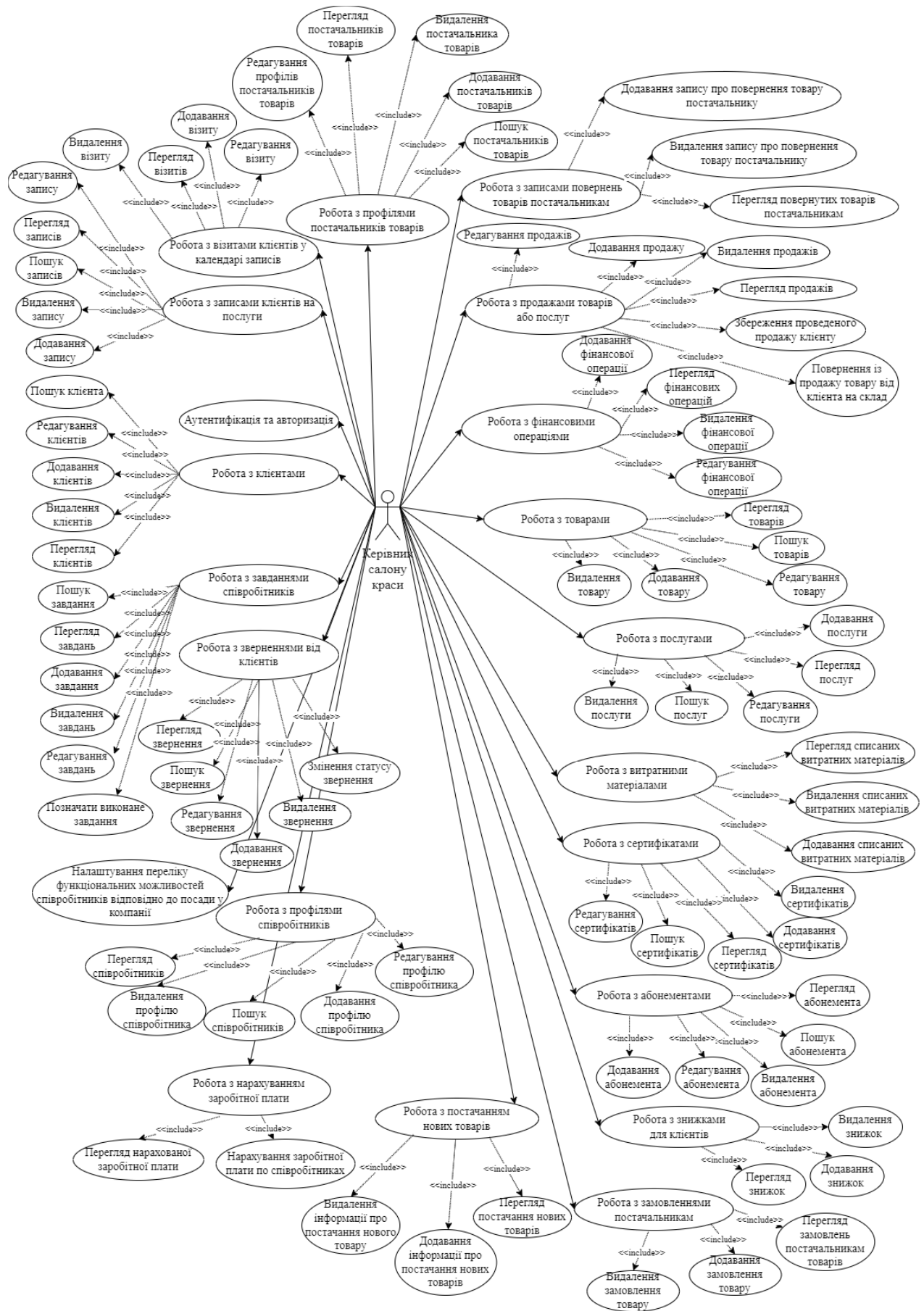


Рисунок 3.3 – Діаграма взаємодії керівника салону краси із вебзастосунком

3.2 Проектування структури навігації вебзастосунку

У вебзастосунку для керування бізнес-процесами у салоні краси буде розроблене навігаційне меню, яке дозволить користувачам переходити між розділами та обирати необхідну вебсторінку. Вебзастосунок для керування бізнес-процесами у салоні краси буде мати такі розділи як «Клієнти», «Завдання співробітників», «Звернення клієнтів», «Записи на послуги», «Календар записів», «Співробітники», «Налаштування переліку функціональних можливостей для співробітників», «Продажі», «Фінансові операції», «Товари», «Послуги», «Списані витратні матеріали», «Сертифікати», «Абонементи», «Знижки для клієнтів», «Постачання товарів», «Замовлення постачальникам», «Повернення товарів», «Постачальники товарів» та «Нарахування заробітної плати». Спроектована структура навігації вебзастосунку зображена на рисунку 3.4.

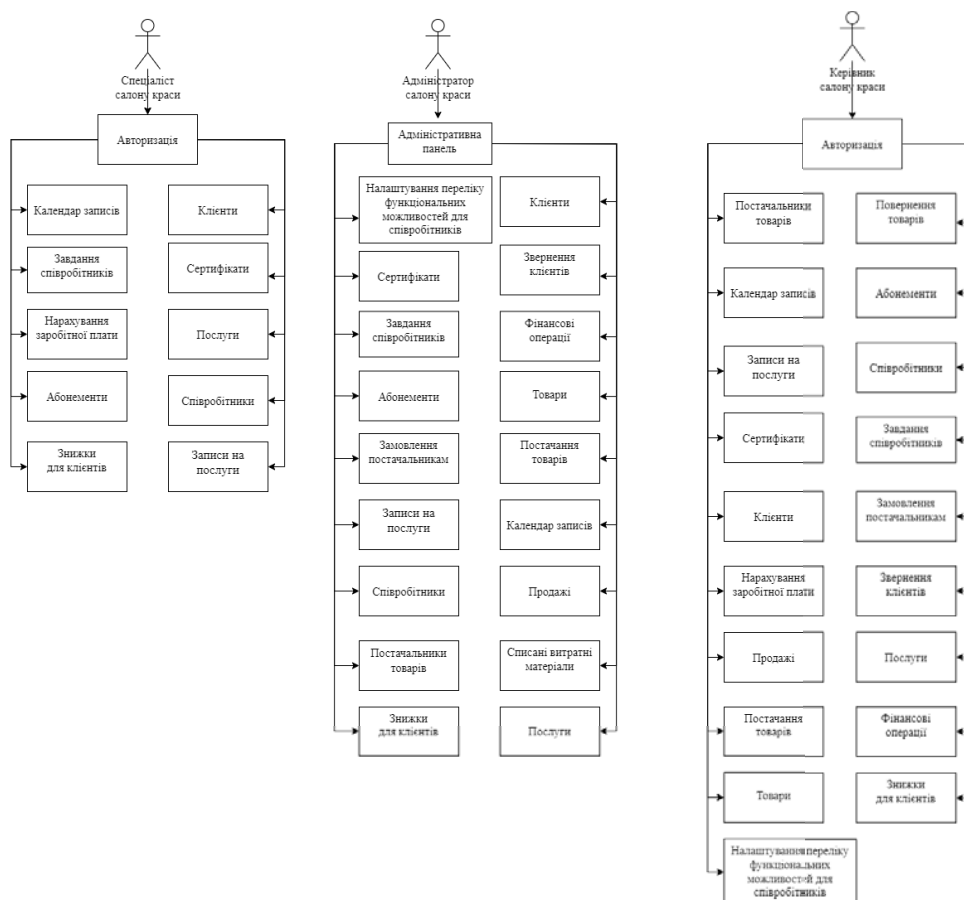


Рисунок 3.4 – Структура навігації вебзастосунку

3.3 Проектування архітектури вебзастосунку

Архітектурним рішенням для розробки вебзастосунку для керування бізнес-процесами у салоні краси було обрано використовувати патерн MVC. Архітектурне рішення MVC дозволяє ділити вебзастосунок на три рівні:

- контролер (сервер);
- представлення (інтерфейс);
- модель (завдання пов'язані з базою даних) [76].

На рисунку 3.5 наведено схему архітектурного рішення MVC.

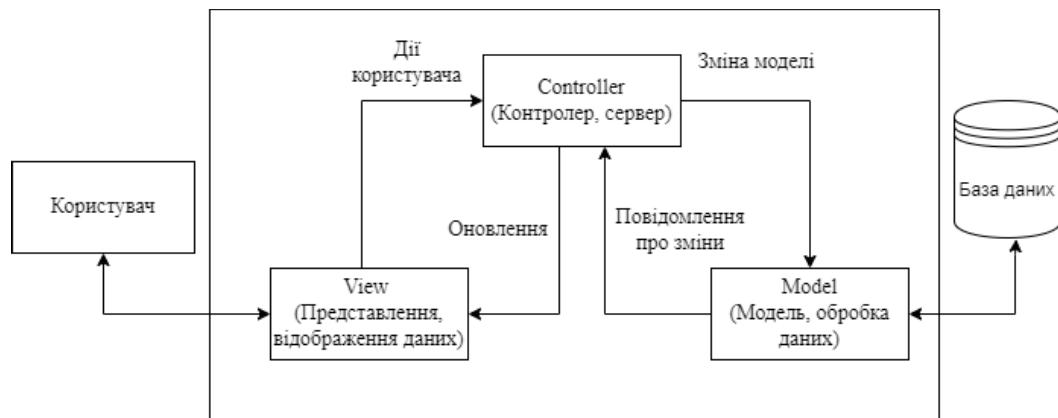


Рисунок 3.5 – Архітектурний шаблон MVC

Рівень контролера відповідає за бізнес-логіку вебзастосунку та роботу з обробки всіх запитів та прийняття рішень, а також контролер є проміжним шаром, який забезпечує зв'язок між моделлю та представленням. Рівень представлення призначений для взаємодії з користувачем, тобто відображає набір графічних елементів, які бачать та з якими взаємодіють користувачі на вебсторінках. Рівень моделі передбачає роботу з даними, які пов'язані з бізнес-логікою, тобто модель взаємодіє з базою даних та відповідає на запит контролера [76].

Переваги архітектурного рішення MVC:

- програмний код можна легко розширювати;
- створення гнучких та масштабованих вебзастосунків;

- можливість відокремлення прикладних завдань, а саме бізнес-логіку, інтерфейс користувача та обробку запитів;
- контролер можна легко ізолювати від інших компонентів, що дозволяє писати тести не маючи жодних залежностей від інших частин системи;
- розробники можуть використовувати повторно моделі, представлення та контролери в інших частинах програми або у різних проєктах;
- модульне тестування окремих компонентів [76].

Було обрано використовувати програмну архітектуру для реалізації вебзастосунку на основі стека технологій MERN. MERN є повностековим JS рішенням, який використовує фреймворки та інструменти для створення JS вебзастосунків. Вебзастосунки на основі стека MERN розроблюються на архітектурному рішенні MVC, який підтримує трирівневу архітектуру [77].

Клієнтський рівень передбачає розробку інтерфейсної частини вебзастосунку з використанням таких інструментів як мови розмітки елементів HTML, мови опису стилів оформлення елементів CSS, бібліотеку елементів інтерфейсу користувача React, бібліотеку компонентів Material UI, бібліотеку маршрутизації React Router та менеджер станів вебзастосунку Redux. На цьому рівні користувачі отримують доступ до функцій вебзастосунку [77].

Рівень бізнес-логіки або контролеру передбачає використання серверного фреймворку Express.js на програмному середовищі виконання Node.js. Даний рівень дозволяє взаємодіяти між клієнтом та рівнем бази даних. Бібліотека Mongoose взаємодіє між СКБД MongoDB та серверним фреймворком Express.js за допомогою інструменту ODM, який дозволяє визначати об'єкти за допомогою схеми, які відображаються в документі СКБД MongoDB. Рівень бізнес-логіки приймає запити від клієнта та обслуговує їх шляхом отримання відповідних даних з рівня бази даних. Рівень бази даних зберігає дані вебзастосунку у СКБД MongoDB [77].

На рисунку 3.6 наведено архітектуру вебзастосунку на основі обраного стека для розробки MERN.

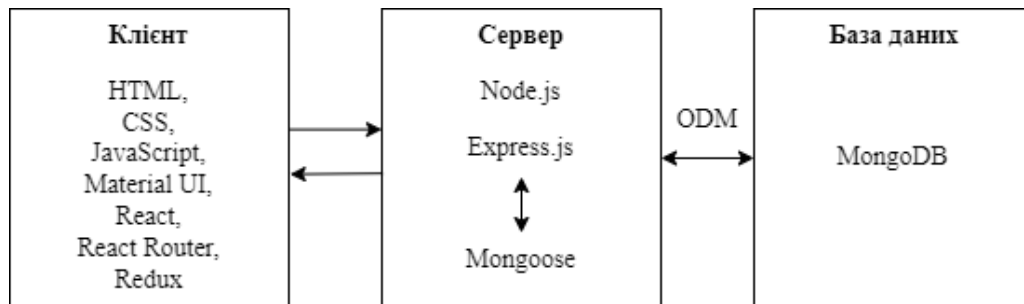


Рисунок 3.6 – Схема архітектури програмного продукту

3.4 Висновки до третього розділу

У пунктах даного розділу було описано функціональні вимоги для кожного користувача та приведено набір діаграм сценаріїв використання вебзастосунку для керування бізнес-процесами у салоні краси. Наведені діаграми відображають бізнес-процеси у розроблювальному вебзастосунку.

Спроектовано структуру навігації розроблювального вебзастосунку, яка дозволить користувачам переходити між розділами вебзастосунку. Також було оглянуто особливості обраного архітектурного рішення та його переваги. Приведено схему обраного архітектурного шаблону для реалізації вебзастосунку та схему обраної програмної архітектури на основі обраного стека MERN.

4 КОНСТРУЮВАННЯ ПРОГРАМНОГО ПРОДУКТУ

У розділі приведено фізичну модель розробленої бази даних у СКБД MongoDB та наведено сутності розроблених колекцій. Наведено розроблений алгоритм функціонування вебзастосунку для керування бізнес-процесами у салоні краси. Описано розроблені модулі, функції розроблених частин вебзастосунку та наведено опис структуризації файлової системи, тобто розроблених директорій вебзастосунку для керування бізнес-процесами у салоні краси.

4.1 Фізична модель реалізованої бази даних

Дані вебзастосунку для керування бізнес-процесами у салоні краси зберігаються у СКБД MongoDB. Виконавши дослідження існуючих вебзастосунків було виокремлено сутності для розробки колекцій:

- календар (calendar);
- завдання (task);
- клієнт (client);
- звернення (request);
- запис (registration);
- співробітник (employee);
- продаж (sale);
- фінансова операція (transaction);
- товар (product);
- послуга (service);
- списані матеріали (materials);
- сертифікат (certificate);
- абонемент (subscription);
- знижка (discount);
- постачання товару (delivery);

- замовлення (order);
- повернення товару (return);
- постачальник (supplier);
- заробітна плата (salary).

Схему фізичної моделі бази даних, яка розроблена за допомогою СКБД MongoDB зображено на рисунку 4.1.

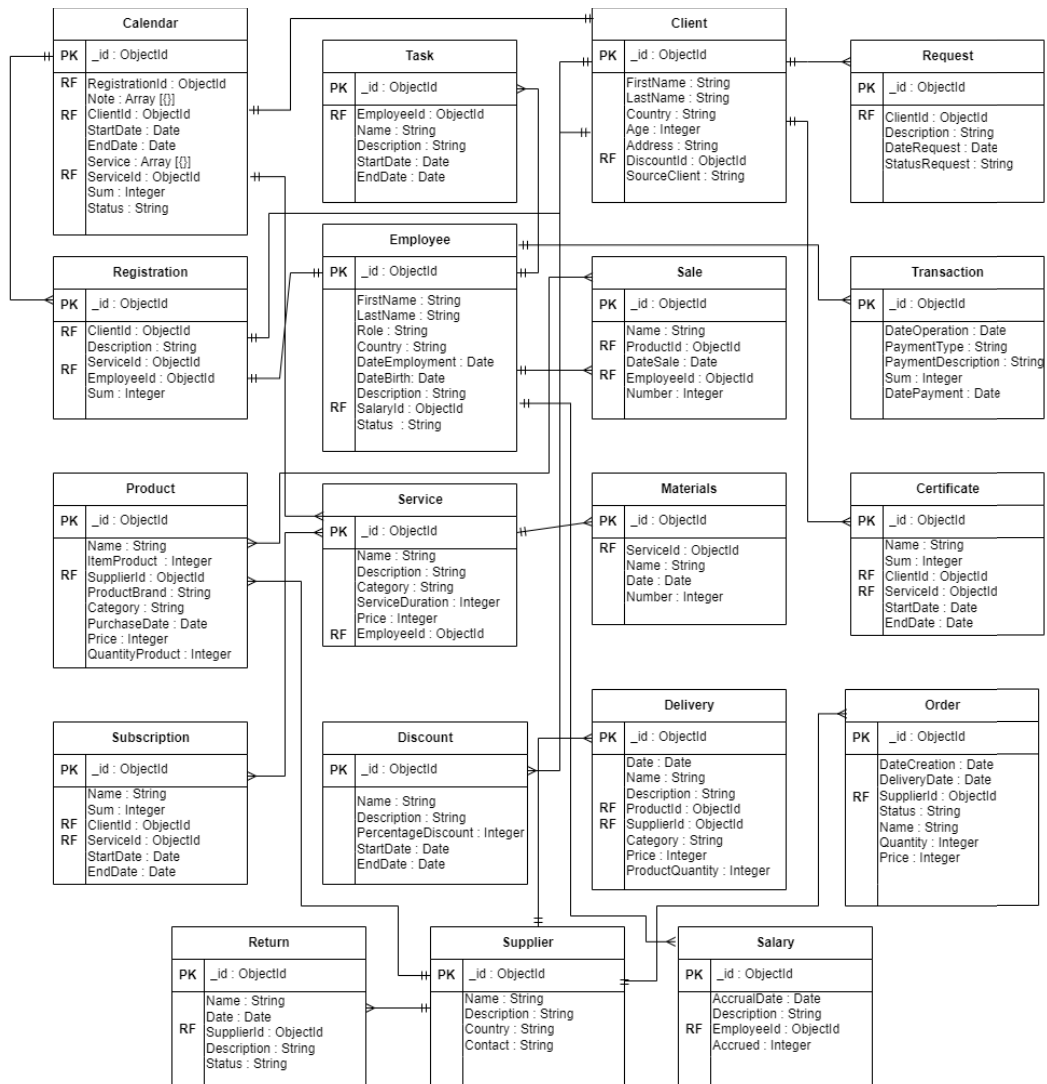


Рисунок 4.1 – Фізична модель бази даних

4.2 Алгоритм функціонування розробленого вебзастосунку

Розроблений алгоритм функціонування вебзастосунку для керування бізнес-процесами у салоні краси наведено на рисунку 4.2.

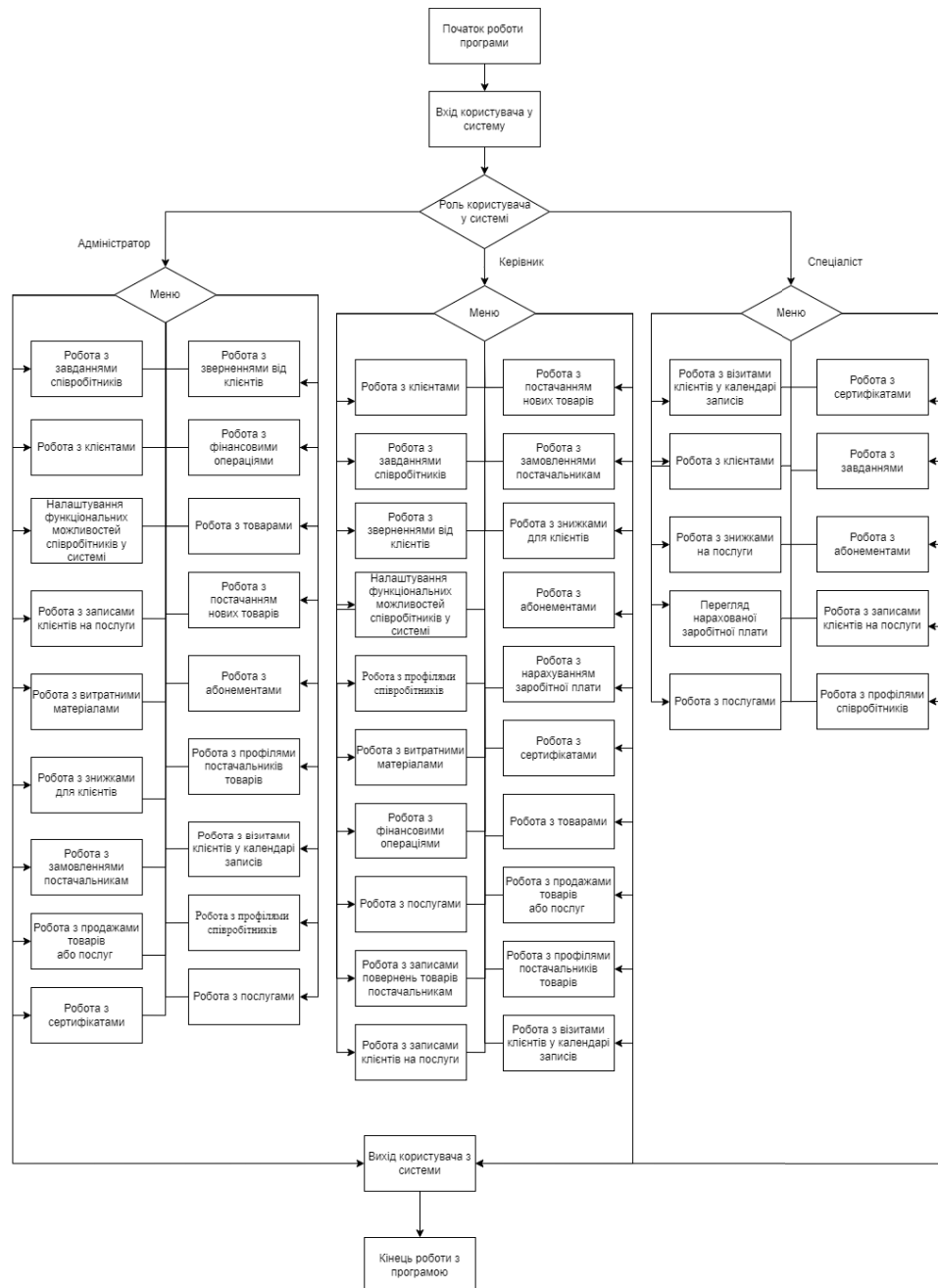


Рисунок 4.2 – Розроблений алгоритм функціонування вебзастосунку

4.3 Розроблені модулі вебзастосунку

Були розроблені наступні модулі у вебзастосунку для керування бізнес-процесами у салоні краси:

- модуль аутентифікації та авторизації;
- модуль відображення та роботи зі списком клієнтів;
- модуль відображення та роботи з завданнями співробітників;

- модуль відображення та роботи зі зверненнями від клієнтів;
- модуль записів клієнтів на послуги;
- модуль відображення та роботи з календарем записів;
- модуль відображення та роботи зі списком співробітників;
- модуль налаштування переліку можливостей співробітників;
- модуль відображення та роботи зі списком продажів;
- модуль відображення та роботи з фінансовими операціями компанії;
- модуль відображення та роботи зі списком товарів;
- модуль відображення та роботи зі списком послуг;
- модуль відображення та роботи з витратними матеріалами;
- модуль відображення та роботи зі списком сертифікатів;
- модуль відображення та роботи зі списком абонементів;
- модуль відображення та роботи зі списком знижок для клієнтів;
- модуль відображення та роботи зі списком постачань товарів;
- модуль відображення та роботи зі списком замовлень;
- модуль відображення та роботи зі списком повернень товарів;
- модуль відображення та роботи зі списком постачальників товарів;
- модуль нарахування заробітної плати по співробітниках.

4.4 Опис розроблених функцій

Розроблені функції інтерфейсної частини вебзастосунку для керування бізнес-процесами у салоні краси:

- `ModalOpen()` – функція для підтвердження дії користувача, що викликає модальне вікно;
- `ModalClose()` – функція для закриття вікна підтвердження дії користувача;
- `UserId()` – функція для визначення id користувача;
- `Auth()` – функція для визначення ролі користувача у системі та отримання даних про користувача;

- SaveData() – функція для передачі та збереження даних;
- AddData() – функція для додавання даних у базу даних;
- DeleteData() – функція для видалення даних з бази даних;
- UpdateData() – функція для редагування та оновлення даних;
- createBrowserRoute() – функція для побудови маршрутизації;
- GetData() – функція для отримання даних з сервера;
- ErrorPage() – функція для створення шаблону сторінки з помилкою;
- reducer() – функція для оновлення стану вебзастосунку.

Розроблені функції серверної частини вебзастосунку для керування бізнес-процесами у салоні краси:

- serverError() – функція для перевірки помилок та повернення статусу помилки;
- post() – функція для операції створення нового запису у базі даних;
- put() – функція для операції оновлення запису у базі даних;
- delete() – функція для операції видалення запису з бази даних;
- get() – функція для передачі та оновлення даних на інтерфейсній частині вебзастосунку;
- listen() – функція для запуску сервера.

4.5 Розроблена структура проєкту вебзастосунку

Інтерфейсна частина вебпроєкту складається з наступних директорій:

- public – директорія, що містить статистичну інформацію вебзастосунку;
- components – директорія, що зберігає компоненти багаторазового використання;
- containers – директорія, що зберігає складні компоненти у яких розроблено функціонал та сторінки вебзастосунку;
- store – директорія, що зберігає дані для початкового стану сховища;
- pages – директорія, що зберігає головні сторінки проєкту;

- `assets` – директорія, що зберігає зображення, глобальні `css` файли для стилів, логотипи, тобто статичні файли;

- `hooks` – директорія, що містить хуки та методи для роботи зі спеціальними компонентами;

- `utils` – директорія, що містить службові функції.

Серверна частина вебпроєкту складається з наступних директорій:

- `config` – директорія, що містить файл з'єднання з СКБД MongoDB за допомогою бібліотеки `Mongoose`;

- `models` – директорія, що зберігає файли з описом схем та створених моделей для СКБД MongoDB за допомогою бібліотеки `Mongoose`;

- `middleware` – директорія, що зберігає файли з логікою обробки запитів.

4.6 Висновки до четвертого розділу

У розділі було наведено структуру розроблених колекцій у СКБД MongoDB та розроблений алгоритм функціонування вебзастосунку для керування бізнес-процесами у салоні краси. Описано розроблені модулі, функції у розроблених частинах вебпроєкту та наведено опис розроблених директорій вебзастосунку для керування бізнес-процесами у салоні краси.

5 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ПРОДУКТУ

У розділі описано призначення вебзастосунку для керування бізнес-процесами у салоні краси. Описано вимоги до апаратної та програмної частини для функціонування вебзастосунку, перелік файлів програми, виконання програми, повідомлення користувачам при роботі з вебзастосунком та результати тестування функцій розроблених модулів у веббраузерах.

5.1 Опис призначення програмного продукту

Вебзастосунок «SalonSystem» призначений для автоматизації щоденних завдань співробітників салону краси та спрощення ведення обліку бізнес-процесів. Вебзастосунок «SalonSystem» використовується для користувачів, яким потрібно додати до бази клієнтів, завдання, звернення клієнтів, запис клієнтів на послуги, новий візит у календар записів, співробітників салону краси, продажі послуг або товарів, фінансові операції (дохід, витрату або перекази між рахунками), товари, послуги, списання витратних матеріалів, сертифікати, абонементи, знижки на послуги, постачання товарів, замовлення постачальникам, повернення товарів, постачальників товарів та нарахування заробітної плати.

Вебзастосунок «SalonSystem» надає користувачам наступні функції:

- аутентифікація та надання співробітнику після авторизації функцій для роботи відповідно до переліку наданих йому прав адміністратором;
- пошук існуючого клієнта;
- перегляд клієнтів;
- додавання нових клієнтів;
- редагування клієнтів;
- видалення клієнтів;

- пошук існуючого завдання;
- додавання нового завдання;
- перегляд існуючих завдань;
- редагування існуючих завдань;
- видалення завдань;
- можливість робити позначку, що завдання виконане;
- пошук звернення від клієнта;
- перегляд звернень клієнтів;
- додавання нового звернення від клієнта;
- змінення статусу звернення клієнта;
- редагування звернення клієнта;
- видалення звернення клієнта;
- перегляд записів клієнтів на послуги;
- пошук записів клієнтів на послуги;
- додавання нового запису клієнта на послугу;
- редагування інформації про запис клієнта на послугу;
- видалення запису клієнта на послугу;
- перегляд візитів клієнтів до салону краси у календарі записів;
- додавання нового візиту клієнта до салону краси у календар записів;
- редагування інформації щодо візиту клієнта до салону краси у календарі записів;
- видалення візиту клієнта до салону краси у календарі записів;
- перегляд співробітників салону краси;
- пошук співробітників салону краси;
- додавання профілю співробітника;
- редагування профілю співробітника;
- видалення профілю співробітника при його звільненні;
- налаштування переліку можливостей співробітників відповідно до посади у компанії;
- перегляд продажів клієнтам;

- додавання продажу;
- збереження проведеного продажу клієнту;
- повернення із продажу товару від клієнта на склад;
- редагування продажів;
- видалення продажів;
- перегляд фінансових операцій салону краси;
- додавання таких фінансових операцій, як надходження, переказ коштів між рахунками та витрата коштів;
- видалення внесеної до бази даних фінансової операції;
- редагування фінансових операцій;
- перегляд товарів;
- пошук товарів;
- додавання нового товару;
- редагування інформації про товар;
- видалення товару;
- перегляд послуг салону краси;
- пошук послуг салону краси;
- додавання нової послуги;
- редагування інформації про послугу;
- видалення послуги;
- перегляд списаних витратних матеріалів;
- додавання до бази та видалення із бази списані витратні матеріали;
- перегляд сертифікатів;
- пошук сертифікатів;
- додавання сертифіката на послуги або товари;
- видалення сертифіката;
- редагування інформації про сертифікат;
- перегляд абонементів;
- пошук абонементів;
- додавання абонементів;

- редагування абонементів;
- видалення абонементів;
- перегляд знижок на послуги;
- додавання знижок на послуги;
- видалення знижок на послуги;
- перегляд постачання нових товарів;
- додавання інформації про постачання нових товарів;
- видалення інформації про постачання нового товару;
- перегляд замовлень постачальникам товарів;
- додавання замовлення товару;
- видалення замовлення товару;
- перегляд повернутих товарів постачальникам;
- додавання запису про повернення товару постачальнику;
- видалення запису про повернення товару постачальнику;
- перегляд постачальників товарів;
- пошук постачальників товарів;
- додавання постачальників товарів;
- редагування інформації про постачальника товарів;
- видалення постачальника товарів;
- перегляд нарахованої заробітної плати;
- нарахування заробітної плати по співробітниках.

5.2 Вимоги до програмно-технічних засобів

Вимоги до апаратного забезпечення вебзастосунку «SalonSystem» включають наступні параметри до апаратних засобів:

- процесор Intel Core i5 або інший сучасний починаючи з 5 покоління;
- оперативна пам'ять обсягом не менше 8 Гб;
- монітор Full HD та типом матриці IPS;
- внутрішній або зовнішній жорсткий диск не менше 512 Гб;

- накопичувач не менше 512 Гб;
- мережева карта;
- відеокарта NVIDIA GeForce GTX 1650;
- бездротова або дротова комп'ютерна миша та клавіатура;
- вільний простір на жорсткому диску.

Для експлуатації вебзастосунку «SalonSystem» необхідна наявність встановленого редактору коду Visual Studio Code, середовища виконання Node.js, СКБД MongoDB Community Server, клієнта MongoDB Shell для взаємодії з сервером та графічного клієнта MongoDB Compass. У редакторі коду необхідно встановити через термінал бібліотеки React, Material UI, React Router, Redux, Mongoose та фреймворк Express.js.

5.3 Характеристики програмного продукту

Найменування вебзастосунку – «SalonSystem».

Перелік файлів клієнтської частини вебзастосунку:

- App.jsx – файл, який містить визначені основні компоненти вебзастосунку;
- App.css – файл для стилізації компонента App;
- index.js – файл, який виконує розгортання усього вебзастосунку;
- Main.js – файл, який містить логіку для зв'язку клієнтської частини з серверною;
- AppHeader.jsx – компонент, що формує верхній розділ сторінки;
- AppFooter.jsx – компонент, що формує нижній розділ сторінки;
- AppContainer.jsx – компонент, що формує основний розділ сторінки;
- AppContainer.css – файл для стилізації компонента AppContainer;
- AppHeader.css – файл для стилізації компонента AppHeader;
- AppFooter.css – файл для стилізації компонента AppFooter;
- apiFunc.js – файл, що містить основні функції, які використовуються у проєкті.

Перелік файлів серверної частини вебзастосунку:

- index.js – файл, що містить основні функції, які використовуються для обробки запитів користувачів;
- db.js – файл, що містить налаштування з'єднання з СКБД MongoDB;
- appModel.js – файл, що містить схему та моделі для структуризації даних у СКБД MongoDB за допомогою бібліотеки Mongoose.

5.4 Виконання програмного продукту

На початку роботи з вебзастосунком «SalonSystem» користувачу необхідно пройти аутентифікацію облікового запису. Для аутентифікації користувачу надається форма для заповнення логіну та паролю (рис. 5.1). Після коректно заповнених даних користувач проходить авторизацію, яка визначає роль користувача у системі та його функціональні можливості для роботи із вебзастосунком.

SalonSystem

ВХІД

Логін

Пароль

УВІЙТИ

Вебзастосунок для керування бізнес-процесами у салоні краси © 2023

Рисунок 5.1 – Вебсторінка аутентифікації вебзастосунку «SalonSystem»

Адміністратор салону має можливість роботи з клієнтами (рис. 5.2), завданнями співробітників, зверненнями клієнтів, календарем записів, базою

співробітників салону краси, продажами послуг або товарів, фінансовими операціями, товарами, послугами, списанням витратних матеріалів, сертифікатами, абонементами, знижками для клієнтів, постачанням товарів та базою постачальників товарів. Також адміністратор салону краси має можливість налаштування функціональних можливостей для кожного співробітника.

The screenshot shows the 'SalonSystem' web application interface. At the top, there is an orange navigation bar with the following menu items: 'Сертифікати', 'Налаштування', 'Абонементи', 'Постачальники', 'Календар', 'Продажі', 'Замовлення', 'Послуги', 'Клієнти', 'Товари', 'Співробітники', and 'Завдання'. Below the navigation bar, the main content area is titled 'Додати нового клієнта:'. It contains a form with the following fields: 'Ім'я:' (text input), 'Прізвище:' (text input), 'Країна:' (text input), 'Адреса:' (text input), 'Вік:' (dropdown menu), and 'Дата народження:' (date picker showing '01.05.2001'). At the bottom of the form is a button labeled 'Додати та зберегти'. Below the form, there is a footer bar with the text 'Вебзастосунок для керування бізнес-процесами у салоні краси © 2023'.

Рисунок 5.2 – Вебсторінка додавання нового клієнта вебзастосунку «SalonSystem»

Керівник салону має можливість роботи з послугами (рис. 5.3), клієнтами, завданнями, зверненнями клієнтів, записами клієнтів на послуги, календарем записів, базою співробітників салону, продажами, фінансовими операціями салону, товарами, списанням витратних матеріалів, сертифікатами, абонементами, знижками для клієнтів на послуги, постачанням товарів, замовленнями товарів, поверненнями товарів, базою постачальників товарів та нарахуванням заробітної плати. Також керівник салону краси має можливість налаштування функціональних можливостей для кожного співробітника.

Рисунок 5.3 – Вебсторінка додавання нової послуги вебзастосунку «SalonSystem»

Спеціаліст салону має можливість роботи з завданнями (рис. 5.4), календарем записів, нарахуванням заробітної плати, абонементами, знижками для клієнтів, клієнтами, сертифікатами, послугами, базою співробітників салону та записами на послуги.

Рисунок 5.4 – Вебсторінка додавання нового завдання вебзастосунку «SalonSystem»

5.5 Опис повідомлень користувачу

Вебзастосунок «SalonSystem» виводить повідомлення користувачу про помилку при створенні запису у базі даних (рис. 5.5).

Помилка при створенні запису



Рисунок 5.5 – Помилка при створенні запису

Вебзастосунок «SalonSystem» виводить повідомлення користувачу про успішність операції (рис. 5.6).

Операція успішно виконана

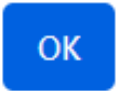


Рисунок 5.6 – Операція успішно виконана

Вебзастосунок «SalonSystem» виводить повідомлення користувачу про некоректність введених даних при вході у систему (рис. 5.7).

Некоректні дані логіну або паролю

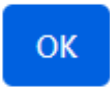


Рисунок 5.7 – Некоректні дані логіну або паролю

5.6 Опис тестування функцій розроблених модулів

Для тестування функцій розроблених модулів було створено чек-лист, який відображає результати проведеного тестування у веббраузерах Firefox та Chrome. Чек-лист тестування розроблених функцій вебзастосунку наведено у таблиці 5.1.

Таблиця 5.1 – Чек-лист тестування функціоналу вебзастосунку

Вебзастосунок «SalonSystem»	Firefox	Chrome
Модуль аутентифікації та авторизації	+	+
Модуль відображення та роботи зі списком клієнтів	+	+
Модуль відображення та роботи з завданнями співробітників	+	+
Модуль відображення та роботи зі зверненнями від клієнтів	+	+
Модуль записів клієнтів на послуги	+	+
Модуль відображення та роботи з календарем записів	+	+
Модуль відображення та роботи зі списком співробітників	+	+
Модуль налаштування переліку можливостей співробітників	+	+
Модуль відображення та роботи зі списком продажів	+	+
Модуль відображення та роботи з фінансовими операціями компанії	+	+
Модуль відображення та роботи зі списком товарів	+	+
Модуль відображення та роботи зі списком послуг	+	+
Модуль відображення та роботи з витратними матеріалами	+	+
Модуль відображення та роботи зі списком сертифікатів	+	+
Модуль роботи зі списком абонементів	+	+

Продовження таблиці 5.1

Вебзастосунок «SalonSystem»	Firefox	Chrome
Модуль відображення та роботи зі списком знижок для клієнтів	+	+
Модуль відображення та роботи зі списком поставачань товарів	+	+
Модуль відображення та роботи зі списком замовлень	+	+
Модуль відображення та роботи зі списком повернень товарів	+	+
Модуль відображення та роботи зі списком поставачальників товарів	+	+
Модуль нарахування заробітної плати по співробітниках	+	+

5.7 Висновки до п'ятого розділу

У розділі були описані вимоги до апаратних та програмних засобів при експлуатації вебзастосунку, призначення розробленого продукту, функціональні можливості, перелік файлів розробленого вебпроєкту, розроблені повідомлення користувачам та наведено таблицю результатів тестування вебзастосунку у різних веббраузерах.

ВИСНОВКИ

При виконанні дипломної магістерської роботи було досліджено та розроблено вебзастосунок для керування бізнес-процесами у салоні краси. Було визначено функції, якими володіють вебзастосунки для керування бізнес-процесами у салоні краси. Метою вебзастосунків для керування бізнес-процесами у салоні краси є автоматизація щоденних завдань співробітників салону краси та спрощення ведення обліку бізнес-процесів.

Для реалізації інтерфейсної частини вебзастосунку було обрано використовувати мову програмування JS, бібліотеку елементів React, бібліотеку компонентів Material UI, бібліотеку маршрутизації вебсторінок React Router та менеджер станів Redux. Для реалізації серверної частини вебзастосунку було обрано використовувати мову програмування JS, програмне середовище виконання Node.js, фреймворк Express.js, нереляційну СКБД MongoDB та бібліотеку для взаємодії з базою даних Mongoose.

Приведено набір діаграм сценаріїв використання вебзастосунку для керування бізнес-процесами у салоні краси, які моделюють бізнес-процеси. Виконано проектування структури навігації вебзастосунку для керування бізнес-процесами у салоні краси. Оглянуто та виокремлено переваги обраного архітектурного рішення та програмної архітектури, а також побудовано відповідні схеми. Наведено структуру розроблених колекцій у СКБД MongoDB та розроблений алгоритм функціонування вебзастосунку для керування бізнес-процесами у салоні краси. Описано розроблені модулі, функції та наведено опис розроблених директорій вебзастосунку для керування бізнес-процесами у салоні краси.

У роботі науковою новизною є розроблені моделі взаємодії користувачів з вебзастосунком для керування бізнес-процесами у салоні краси, які наведені у вигляді діаграм сценаріїв поведінки користувачів із вебзастосунком. Тестування розроблених функцій вебзастосунку довело успішність виконання їх при роботі у різних веббраузерах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Принцип еталона: як поліпшити якість сервісу в салоні краси [Електрон. ресурс]. – Режим доступу: <https://beautyprosoftware.com/uk/blog/kak-uluchshit-kachestvo-servisa/>.
2. CRM для салонів краси [Електрон. ресурс]. – Режим доступу: <https://easyweek.com.ua/crm-dlya-saloniv-kрасy.html>.
3. Салон краси [Електрон. ресурс]. – Режим доступу: <https://business.diia.gov.ua/idea/kрасa-ta-zdorova/salon-kрасi>.
4. Класифікація салонів краси: знайдіть свою нішу на б'юті-ринку [Електрон. ресурс]. – Режим доступу: <https://beautyprosoftware.com/uk/blog/klassifikatsia-salonov-kрасoty/>.
5. Б'юті-підприємства: особливості та класифікація [Електрон. ресурс]. – Режим доступу: <https://marketer.ua/ua/beauty-enterprises-features-and-classification/>.
6. Різновиди салонів краси в залежності від цінової політики [Електрон. ресурс]. – Режим доступу: <https://krasunia.com.ua/a401892-raznovidnosti-salonov-kрасoty.html>.
7. Бізнес-план салону краси [Електрон. ресурс]. – Режим доступу: <https://easyweek.com.ua/biznes-plan-salonu-kрасi-z-rozrahunkami.html>.
8. Як відкрити свій салон краси з нуля [Електрон. ресурс]. – Режим доступу: <https://appointer.ua/blog/yak-vidkryty-salon-kрасy/>.
9. Відкриваємо салон краси. Необхідне обладнання для старту салону [Електрон. ресурс]. – Режим доступу: <https://medicalaser.com.ua/ua/blog-ua/lazerna-epiljatsija/vidkrivajemo-salon-kрасi-neobhidne-obladnannja-dlja-startu-salonu/>.
10. Салонний бізнес: рентабельність, переваги і недоліки [Електрон. ресурс]. – Режим доступу: https://ukrsalon.com.ua/blogs/cat_dlya-rukovoditeley-salonov/salonnyu-biznes-rentabelnost-preimuschestva-i-nedostatki?lang=ua.

11. Посадові обов'язки керуючого салоном краси: особисті якості і функції співробітника [Електрон. ресурс]. – Режим доступу: <https://what.com.ua/posadovi-oboviazki-keryuchogo/>.
12. Посадові інструкції в салоні краси [Електрон. ресурс]. – Режим доступу: <https://duso.ua/ua/blog/dolzhnostnye-instrukcii-v-salone-krasoty>.
13. Обов'язки адміністратора салону краси: завдання та інструменти для їх виконання [Електрон. ресурс]. – Режим доступу: <https://beautyprosoftware.com/uk/blog/obiazannosti-administratora-salona-krasoty/>.
14. Обов'язки адміністратора салону краси [Електрон. ресурс]. – Режим доступу: <https://appointer.ua/blog/obovyazki-administratora-salonu-krasy/>.
15. Посадова інструкція парикмахера [Електрон. ресурс]. – Режим доступу: <https://cosmotrade.com.ua/uk/dolzhnostnaja-instrukcija-parikmahera/>.
16. Розвиток салону краси. Стратегії та практичні рекомендації [Електрон. ресурс]. – Режим доступу: <https://beautyprosoftware.com/uk/blog/razvitie-salona-krasoty-strategii-i-prakticheskie-retomendatsii/>.
17. Впровадження CRM: завдання, способи, вартість [Електрон. ресурс]. – Режим доступу: <https://remonline.ua/blog/crm-implementation/>.
18. CRM-система – це ключ до розвитку бізнесу [Електрон. ресурс]. – Режим доступу: <https://remonline.ua/blog/what-is-crm/>.
19. CRM для салонів краси [Електрон. ресурс]. – Режим доступу: <https://wezom.com.ua/ua/blog/crm-dlya-salonov-krasoty>.
20. Програми для салону краси, перукарні, манікюрного салону [Електрон. ресурс]. – Режим доступу: <https://1b.app/ua/software-for/hairdresser-and-beauty-salons/>.
21. 3 причини, чому ви більше не можете вести облік в таблицях [Електрон. ресурс]. – Режим доступу: <https://remonline.ua/blog/3-reasons-dont-keep-management-accounting-spreadsheets/>.
22. CRM для салону краси: як не втрачати клієнтів з месенджерів

[Електрон. ресурс]. – Режим доступу: <https://blog.keycrm.app/uk/crm-dlya-salonu-krasi-yak-ne-vtrachati-kliientiv-z-mesendzheriv/>.

23. Навіщо потрібен опис бізнес-процесів [Електрон. ресурс]. – Режим доступу: <https://remonline.ua/blog/business-process/>.

24. Автоматизація бізнесу [Електрон. ресурс]. – Режим доступу: <https://remonline.ua/blog/business-process-automation/>.

25. Business processes and business process problems: definition and categories [Electronic resource] / – Access mode: <https://www.cardanit.com/blog/what-are-business-processes/>.

26. Більше, ніж просто CRM. Одна програма для управління салоном краси [Електрон. ресурс]. – Режим доступу: <https://cleverbox-crm.com/>.

27. Глибока аналітика для повного контролю роботи в режимі онлайн [Електрон. ресурс]. – Режим доступу: <https://cleverbox-crm.com/zvity.html>.

28. Журнал запису [Електрон. ресурс]. – Режим доступу: <https://cleverbox-crm.com/kalendar.html>.

29. Задачі CleverBOX:CRM [Електрон. ресурс]. – Режим доступу: <https://cleverbox-crm.com/zadachi-nagaduvanya.html>.

30. Клієнтська база CleverBOX:CRM [Електрон. ресурс]. – Режим доступу: <https://cleverbox-crm.com/klienty.html>.

31. Складський облік [Електрон. ресурс]. – Режим доступу: <https://cleverbox-crm.com/sklad.html>.

32. Облік фінансів [Електрон. ресурс]. – Режим доступу: <https://cleverbox-crm.com/finance.html>.

33. Доступ до системи. Ролі [Електрон. ресурс]. – Режим доступу: <https://cleverbox-crm.com/dostup.html>.

34. Інтеграція CRM + IP-телефонія [Електрон. ресурс]. – Режим доступу: <https://cleverbox-crm.com/telephonija.html>.

35. Розраховуйте зарплати швидко, легко та безпомилково [Електрон. ресурс]. – Режим доступу: <https://cleverbox-crm.com/zarplata.html>.

36. Власний МОБІЛЬНИЙ ЗАСТОСУНОК для вашого салону краси

чи клініки [Електрон. ресурс]. – Режим доступу: <https://cleverbox-crm.com/aplication.html>.

37. Як налаштувати застосунок для клієнтів [Електрон. ресурс]. – Режим доступу: <http://support.cleverbox-crm.com/uk/zastosunok-dlia-kliientiv/nalashtuvati-zastosunok>.

38. Салон краси [Електрон. ресурс]. – Режим доступу: https://wlaunch.net/uk/companies/beauty_salon.

39. Ключові можливості системи [Електрон. ресурс]. – Режим доступу: <https://wlaunch.net/uk/features>.

40. Краса [Електрон. ресурс]. – Режим доступу: <https://wlaunch.net/uk/companies/beauty/>.

41. Термінологія [Електрон. ресурс]. – Режим доступу: <https://wlaunch.net/uk/articles/article/terminology>.

42. Філії компанії [Електрон. ресурс]. – Режим доступу: <https://wlaunch.net/uk/articles/article/branches>.

43. Робота з календарем [Електрон. ресурс]. – Режим доступу: https://wlaunch.net/uk/articles/article/scheduler_actions.

44. Форма для сайту [Електрон. ресурс]. – Режим доступу: https://wlaunch.net/uk/articles/article/website_booking_form.

45. Мій склад [Електрон. ресурс]. – Режим доступу: https://wlaunch.net/uk/articles/article/warehouse_bootstrap.

46. CRM для салону краси [Електрон. ресурс]. – Режим доступу: <https://remonline.ua/beauty-salon-and-spa-center/>.

47. Вступ до JavaScript [Електрон. ресурс]. – Режим доступу: <https://uk.javascript.info/intro>.

48. JavaScript - Overview [Electronic resource] / – Access mode: https://www.tutorialspoint.com/javascript/javascript_overview.htm.

49. Що таке JavaScript [Електрон. ресурс]. – Режим доступу: <https://cases.media/creativepractice/course/javascript-basics/module/js-intro/show-take-javascript?list=additionalArticles>.

50. Редактори коду [Електрон. ресурс]. – Режим доступу: <https://uk.javascript.info/code-editors>.

51. Getting Started [Electronic resource] / – Access mode: <https://code.visualstudio.com/docs>.

52. Visual Studio Code Vs Atom: Which Code Editor Is Better [Electronic resource] / – Access mode: <https://www.softwaretestinghelp.com/visual-studio-code-vs-atom/>.

53. HTML [Electronic resource] / – Access mode: <https://influencemarketinghub.com/glossary/html/>.

54. CSS : Cascading Style Sheets [Electronic resource] / – Access mode: <https://medium.com/educipher/css-cascading-style-sheets-b9706e77224>.

55. Why to use SASS instead of CSS ? [Electronic resource] / – Access mode: <https://www.geeksforgeeks.org/why-to-use-sass-instead-of-css/>.

56. The Best Guide to Know What Is React [Electronic resource] / – Access mode: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>.

57. React Hooks — огляд можливостей нового API [Електрон. ресурс]. – Режим доступу: <https://dou.ua/lenta/articles/react-hooks-guide/>.

58. Building Your Own Hooks [Electronic resource] / – Access mode: <https://legacy.reactjs.org/docs/hooks-custom.html>.

59. React Material UI [Electronic resource] / – Access mode: <https://www.geeksforgeeks.org/react-material-ui/>.

60. Feature Overview [Electronic resource] / – Access mode: <https://reactrouter.com/en/main/start/overview>.

61. React Redux [Electronic resource] / – Access mode: <https://www.javatpoint.com/react-redux>.

62. What Is a REST API? [Electronic resource] / – Access mode: <https://www.sitepoint.com/rest-api/>.

63. What Is Node.js and Why You Should Use It [Electronic resource] / – Access mode: <https://kinsta.com/knowledgebase/what-is-node-js/>.

64. Що таке npm (Node Package Manager)? Як встановити та розмістити пакети [Електрон. ресурс]. – Режим доступу: <https://highload.today/uk/npm/>.

65. Node Version Manager Setup Guide [2023] [Electronic resource] / – Access mode: <https://codedamn.com/news/nodejs/nvm-setup-guide>.

66. Node.js Modules [Electronic resource] / – Access mode: <https://www.codingninjas.com/studio/library/node-js-modules>.

67. Node.js Event Loop [Electronic resource] / – Access mode: <https://www.geeksforgeeks.org/node-js-event-loop/>.

68. Node.js HTTP module is beautiful [Electronic resource] / – Access mode: <https://medium.com/@roushanverma123/working-with-node-js-http-module-a8723a87af92>.

69. Розробка вебсайтів [Електрон. ресурс]. – Режим доступу: <https://brander.ua/technologies/expressjs>.

70. Найкращі фреймворки Node.js: розбираємо їх особливості [Електрон. ресурс]. – Режим доступу: <https://wezom.com.ua/ua/blog/luchshie-freymvorki-nodejs>.

71. Top 6 Node JS Frameworks: Which One to Choose in 2023 [Electronic resource] / – Access mode: <https://anywhere.epam.com/business/best-node-js-frameworks>.

72. Kaufmann M. SQL and NoSQL Databases: Modeling, Languages, Security and Architectures for Big Data Management [Text] / M. Kaufmann, A. Meier. – Rotkreuz: Springer Cham, 2023. – 19 p.

73. What is NoSQL? [Electronic resource] / – Access mode: <https://aws.amazon.com/nosql/>.

74. MongoDB Features [Electronic resource] / – Access mode: <https://www.mongodb.com/features>.

75. Introduction to Mongoose [Electronic resource] / – Access mode: <https://www.codingninjas.com/studio/library/introduction-to-mongoose>.

76. What Is The MERN Stack and Why It Is The Best to Fit for Web

Development [Electronic resource] / – Access mode:
<https://www.groovyweb.co/blog/what-is-the-mern-stack-and-why-it-is-the-best-to-fit-for-web-development/>.

77. The Ultimate Guide to MERN Stack [Electronic resource] / – Access mode: <https://altcampus.com/guides/the-ultimate-guide-to-MERN-stack>.

ДОДАТОК А
Технічне завдання

Вступ

Сфера надання послуг краси стрімко розвивається, що обумовлено високим попитом на послуги. Відкриття салону краси є перспективною ідеєю, яка може стати джерелом стабільного прибутку. Однак лише за умови обліку бізнес-процесів у закладі та ефективної роботи з обслуговування клієнтів.

Програмні рішення для обліку бізнес-процесів у салоні краси дозволяють зручно працювати з клієнтською базою, вести запис клієнтів на послуги, контролювати витрату потрібних для роботи матеріалів, керувати продажем власних товарів, формувати статистику та аналітику компанії, а також планувати роботу співробітників. Крім того, такі програмні рішення надають можливості автоматичного розрахунку вартості послуг, що дозволяє уникати помилок в обліку та сприяє економії часових ресурсів закладу.

A.1 Найменування програмного продукту

Назва вебзастосунку – «SalonSystem».

A.2 Підстави для розробки

Випускна кваліфікаційна робота виконується за темою «Дослідження та програмна реалізація вебзастосунку для керування бізнес-процесами у салоні краси» відповідно до наказу № 437 від 14 листопада 2023 р. затвердженим Національним університетом «Запорізька політехніка».

A.3 Призначення та цілі реалізації програмного продукту

Призначенням розроблювального вебзастосунку є керування бізнес-процесами компаніями у сфері послуг краси.

Програмний продукт надає функціональні можливості в залежності від посади співробітника у компанії.

Вебзастосунок «SalonSystem» використовується для користувачів, яким потрібно додати до бази клієнтів, завдання, звернення клієнтів, запис клієнтів на послуги, новий візит у календар записів, співробітників салону краси, продажі послуг або товарів, фінансові операції (дохід, витрату або перекази між рахунками), товари, послуги, списання витратних матеріалів, сертифікати, абонементи, знижки на послуги, постачання товарів, замовлення постачальникам, повернення товарів, постачальників товарів та нарахування заробітної плати.

У базі даних має зберігатися необхідна інформація та надавати користувачу можливість перегляду даних, редагування та видалення зайвої інформації.

Основною ціллю реалізації вебзастосунку «SalonSystem» є покращення обслуговування клієнтів та прискорення виконання щоденних бізнес-процесів у салоні краси.

A.4 Перелік вимог до програмного продукту

A.4.1 Функціональні вимоги

У вебзастосунку «SalonSystem» повинні бути реалізовані функції перераховані нижче відповідно до посади співробітника у компанії. Можливість взаємодії із вебзастосунком буде для наступних ролей співробітників у системі:

- керівник салону краси;
- адміністратор салону краси;
- спеціаліст салону краси.

Вебзастосунок «SalonSystem» повинен надавати користувачам наступні функціональні можливості:

– аутентифікація та надання співробітнику після авторизації функцій для роботи відповідно до переліку наданих йому прав адміністратором;

- пошук існуючого клієнта;
- перегляд клієнтів;
- додавання нових клієнтів;
- редагування клієнтів;
- видалення клієнтів;
- пошук існуючого завдання;
- додавання нового завдання;
- перегляд існуючих завдань;
- редагування існуючих завдань;
- видалення завдань;
- можливість робити позначку, що завдання виконане;
- пошук звернення від клієнта;
- перегляд звернень клієнтів;
- додавання нового звернення від клієнта;
- змінення статусу звернення клієнта;
- редагування звернення клієнта;
- видалення звернення клієнта;
- перегляд записів клієнтів на послуги;
- пошук записів клієнтів на послуги;
- додавання нового запису клієнта на послугу;
- редагування інформації про запис клієнта на послугу;
- видалення запису клієнта на послугу;
- перегляд візитів клієнтів до салону краси у календарі записів;
- додавання нового візиту клієнта до салону краси у календар записів;
- редагування інформації щодо візиту клієнта до салону краси у календарі записів;
- видалення візиту клієнта до салону краси у календарі записів;
- перегляд співробітників салону краси;

- пошук співробітників салону краси;
- додавання профілю співробітника;
- редагування профілю співробітника;
- видалення профілю співробітника при його звільненні;
- налаштування переліку можливостей співробітників відповідно до посади у компанії;
- перегляд продажів клієнтам;
- додавання продажу;
- збереження проведеного продажу клієнту;
- повернення із продажу товару від клієнта на склад;
- редагування продажів;
- видалення продажів;
- перегляд фінансових операцій салону краси;
- додавання таких фінансових операцій, як надходження, переказ коштів між рахунками та витрата коштів;
- видалення внесеної до бази даних фінансової операції;
- редагування фінансових операцій;
- перегляд товарів;
- пошук товарів;
- додавання нового товару;
- редагування інформації про товар;
- видалення товару;
- перегляд послуг салону краси;
- пошук послуг салону краси;
- додавання нової послуги;
- редагування інформації про послугу;
- видалення послуги;
- перегляд списаних витратних матеріалів;
- додавання до бази та видалення із бази списані витратні матеріали;
- перегляд сертифікатів;

- пошук сертифікатів;
- додавання сертифіката на послуги або товари;
- видалення сертифіката;
- редагування інформації про сертифікат;
- перегляд абонементів;
- пошук абонементів;
- додавання абонементів;
- редагування абонементів;
- видалення абонементів;
- перегляд знижок на послуги;
- додавання знижок на послуги;
- видалення знижок на послуги;
- перегляд постачання нових товарів;
- додавання інформації про постачання нових товарів;
- видалення інформації про постачання нового товару;
- перегляд замовлень постачальникам товарів;
- додавання замовлення товару;
- видалення замовлення товару;
- перегляд повернутих товарів постачальникам;
- додавання запису про повернення товару постачальнику;
- видалення запису про повернення товару постачальнику;
- перегляд постачальників товарів;
- пошук постачальників товарів;
- додавання постачальників товарів;
- редагування інформації про постачальника товарів;
- видалення постачальника товарів;
- перегляд нарахованої заробітної плати;
- нарахування заробітної плати по співробітниках.

Керівнику салону краси вебзастосунок «SalonSystem» повинен надавати повний доступ до усіх функцій.

Вебзастосунок «SalonSystem» має забезпечувати наступні функції для роботи адміністратора салону краси:

- пошук, перегляд та додавання нового завдання;
- пошук, перегляд, додавання, редагування інформації та видалення клієнтів;
- пошук, перегляд та додавання нового звернення від клієнта;
- пошук, перегляд, додавання, редагування та видалення інформації про запис клієнта на послугу;
- перегляд, додавання, редагування та видалення інформації щодо візиту клієнта у календарі записів;
- пошук, перегляд, додавання, редагування інформації та видалення профілю співробітника;
- налаштування переліку функціональних можливостей для конкретного співробітника;
- перегляд та додавання продажів;
- перегляд фінансових операцій салону краси;
- пошук, перегляд, додавання, редагування інформації про товар та видалення товарів;
- пошук, перегляд, додавання та видалення послуг салону краси;
- перегляд, додавання та видалення інформації про постачання нових товарів;
- перегляд, додавання та видалення замовлень постачальникам товарів;
- пошук, перегляд, додавання та видалення постачальників товарів;
- перегляд списаних витратних матеріалів;
- пошук, перегляд, додавання та видалення сертифікатів клієнтів;
- пошук, перегляд, додавання та видалення абонементів клієнтів;
- перегляд даних про знижки на послуги.

Наступні функціональні можливості повинен забезпечувати вебзастосунок «SalonSystem» для спеціаліста салону краси:

- пошук та перегляд інформації про клієнта;
- пошук, перегляд та додавання завдання;
- можливість робити позначку, що завдання виконане;
- пошук, перегляд, додавання, редагування та видалення даних про запис клієнта на послугу;
- перегляд, додавання, редагування та видалення інформації про візит клієнта у календарі записів;
- перегляд та пошук співробітників салону краси;
- пошук та перегляд даних про послугу;
- перегляд даних про сертифікати клієнтів;
- перегляд даних про абонементи клієнтів;
- пошук абонементів;
- перегляд даних про знижки на послуги у салоні краси;
- перегляд нарахованої заробітної плати.

Початковими даними у вебзастосунку «SalonSystem» є введені дані користувачем при додаванні клієнтів, завдань, звернень клієнтів, записів клієнтів на послуги, нових візитів до салону краси у календар записів, співробітників, продажів, фінансових операцій, товарів, послуг, списаних витратних матеріалів, сертифікатів, абонементів, знижок на послуги, інформації про постачання нових товарів, замовлень постачальнику товарів, запису про повернення товару постачальнику, постачальників товарів та нарахування заробітної плати по співробітнику.

Вихідними даними у вебзастосунку «SalonSystem» є відображення доданої інформації користувачами при переході на різні розділи.

A.4.2 Вимоги до продуктивності

Вебзастосунок «SalonSystem» має забезпечувати час швидкості відповіді на запити користувачів не більше 3 секунд.

А.4.3 Вимоги до користувацького інтерфейсу

При розробці інтерфейсу користувача необхідно проаналізувати різні програмні продукти, які вже існують і мають аналогічні функції. Виходячи з цього ретельно підібрати колірну гамму і зовнішній вигляд інтерфейсу. Кольори слід підбирати за допомогою кола Іттена та бажано використовувати 3 монохромні кольори.

Основні вимоги до інтерфейсу вебзастосунку:

- дотримання однієї стилістики в елементах інтерфейсу;
- адаптивність вебдизайну до різних розмірів екранів;
- посилання повинні мати шрифт достатнього розміру;
- навігаційні елементи повинні бути зручно розташовані.

А.4.4 Експлуатаційні вимоги програмного продукту

Вебзастосунок не вимагає встановлення якоїсь конкретної операційної системи, але бажано Windows або Linux. Перед роботою із вебзастосунком потрібно організувати можливість запуску програми на будь-якому пристрої, що має можливість взаємодії з веббраузером. Перелік факторів без яких неможливе повноцінне функціонування вебзастосунку перераховано нижче:

- контроль вхідних та вихідних даних;
- підключення пристрою до мережі Інтернет;
- повідомлення користувачу про помилки, які виникають внаслідок взаємодії користувача з вебзастосунком;
- безперебійне живлення апаратного забезпечення.

В системі повинно бути встановлено Visual Studio Code для реалізації інтерфейсної та серверної частини під React та Express.js. У редакторі коду Visual Studio Code необхідно встановити розширення та використовуючи команди через термінал програмні бібліотеки. Для роботи з пакетами використовується пакетний менеджер Node.js.

Вебзастосунок повинен мати доступ до потрібної СКБД, щоб забезпечити ефективне збереження та обробку даних. Для сервера необхідна наявність встановленої СКБД MongoDB Community Server, клієнтської програми MongoDB Shell для взаємодії з сервером та графічного клієнта MongoDB Compass.

A.4.5 Вимоги до апаратного забезпечення

Вимоги до апаратного забезпечення вебзастосунку «SalonSystem» включають наступні параметри до апаратних засобів:

- процесор Intel Core i5 або інший сучасний починаючи з 5 покоління;
- оперативна пам'ять обсягом не менше 8 Гб;
- монітор Full HD та типом матриці IPS;
- внутрішній або зовнішній жорсткий диск не менше 512 Гб;
- накопичувач не менше 512 Гб;
- мережева карта;
- відеокарта NVIDIA GeForce GTX 1650;
- бездротова або дротова комп'ютерна миша та клавіатура;
- вільний простір на жорсткому диску.

A.4.6 Вимоги до безпеки

Кожний співробітник для входу має мати власний обліковий запис. При вході у вебзастосунок «SalonSystem» співробітник проходить аутентифікацію. Адміністратор у відповідному розділі може додати співробітника із зазначенням логіну та паролю, які необхідні при проходженні аутентифікації.

При додаванні співробітника у систему адміністратор вказує перелік функціональних можливостей відповідно до ролі співробітника у компанії. Коли користувач авторизований йому надаються необхідні для роботи

функції. Наприклад, керівник салону краси матиме доступ до усієї функціональності вебзастосунку.

A.5 Етапи виконання магістерської роботи

Виконання дипломної роботи складається з наступних етапів:

- дослідження клієнт-серверних рішень для керування салоном краси;
- обрання засобів розробки вебзастосунку;
- проєктування вебзастосунку;
- конструювання вебзастосунку;
- експлуатація та тестування розробленого вебзастосунку;
- оформлення документації за результатами проведеної роботи.

A.6 Послідовність контролю виконання та приймання роботи

Дипломна магістерська робота виконується згідно з графіком календарного плану. Пояснювальна записка приймається та підписується керівником диплома, після чого подається нормоконтролеру на перевірку. Потім робота подається призначеному рецензенту на отримання рецензії.

Наступним етапом є перевірка дипломної магістерської роботи на плагіат. Якщо дипломна робота має достатній відсоток унікальності після перевірки на плагіат, тоді завідувач кафедри приймає рішення щодо допуску роботи до захисту та відмічає у таблиці з дипломування. Останнім етапом є призначення дати захисту дипломної роботи та захист в присутності визначеної екзаменаційної комісії.

ДОДАТОК Б
Текст програми

Б.1 Маршрутизація сторінок вебзастосунку

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import Calendar from './Pages/Calendar/Calendar';
import Task from './Pages/Task/Task';
import Client from './Pages/Client/Client';
import Request from './Pages/Request/Request';
import Registration from './Pages/Registration/Registration';
import Employee from './Pages/Employee/Employee';
import Sale from './Pages/Sale/Sale';
import Transaction from './Pages/Transaction/Transaction';
import Product from './Pages/Product/Product';
import Service from './Pages/Service/Service';
import Materials from './Pages/Materials/Materials';
import Certificate from './Pages/Certificate/Certificate';
import Subscription from './Pages/Subscription/Subscription';
import Discount from './Pages/Discount/Discount';
import Delivery from './Pages/Delivery/Delivery';
import Order from './Pages/Order/Order';
import Return from './Pages/Return/Return';
import Supplier from './Pages/Supplier/Supplier';
import Salary from './Pages/Salary/Salary';

import {
  createBrowserRouter,
  createRoutesFromElements,
  RouterProvider,
  Route,
  Outlet
} from 'react-router-dom';

const Root = () => {
  return (
    <div className='router'>
```

```

        <Header />
        <Sidebar />
        <Outlet />
    </div>
);
}

const router = createBrowserRouter(
    createRoutesFromElements(
        <Route path="/" element={ <Root /> } >
            <Route index element ={ <App /> } />
            <Route path='calendar' element={ <Calendar /> } />
            <Route path='task' element={ <Task /> } />
            <Route path='client' element={ <Client /> } />
            <Route path='request' element={ <Request /> } />
            <Route path='registration' element={ <Registration />
        } />

        <Route path='employee' element={ <Employee /> } />
        <Route path='sale' element={ <Sale /> } />
        <Route path='transaction' element={ <Transaction /> }
    />

    <Route path='product' element={ <Product /> } />
    <Route path='service' element={ <Service /> } />
    <Route path='materials' element={ <Materials /> } />
    <Route path='certificate' element={ <Certificate /> }
    />

    <Route path='subscription' element={ <Subscription />
    } />

    <Route path='discount' element={ <Discount /> } />
    <Route path='delivery' element={ <Delivery /> } />
    <Route path='order' element={ <Order /> } />
    <Route path='return' element={ <Return /> } />
    <Route path='supplier' element={ <Supplier /> } />
    <Route path='salary' element={ <Salary /> } />
    <Route path='users/:userId' loader={loader}
    element={<UserPage />} errorElement={<ErrorPage/>} />

```

```

        <Route path='*' element={ <ErrorPage /> } />
      </Route>
    )
  );

const root =
ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <RouterProvider router={router} />
);

```

Б.2 Код файла appModel.js

```

const mongoose = require('mongoose')

const calendarSchema = new mongoose Schema({
  registration: string,
  note: string,
  client: string,
  startdate: date,
  enddate: date,
  service: string,
  sum: integer,
  status: string,
})

const taskSchema = new mongoose Schema({
  employee: string,
  name: string,
  description: string,
  startdate: date,
  enddate: date,
})

const clientSchema = new mongoose Schema({

```

```
  firstname: string,  
  lastname: string,  
  country: string,  
  age: integer,  
  address: string,  
  discount: integer,  
  sourceclient: string,  
})
```

```
const requestSchema = new mongoose Schema({  
  client: string,  
  description: string,  
  daterequest: date,  
  statusrequest: string,  
})
```

```
const registrationSchema = new mongoose Schema({  
  client: string,  
  description: string,  
  service: string,  
  employee: string,  
  sum: integer,  
})
```

```
const employeeSchema = new mongoose Schema({  
  firstname: string,  
  lastname: string,  
  role: string  
  country: string,  
  dateemployment: date,  
  datebirth: date,  
  description: string,  
  salary: integer,  
  status: string,  
})
```

```
const saleSchema = new mongoose Schema({
  name : string,
  product: string,
  datesale: date,
  employee: string,
  number: integer,
})
```

```
const transactionSchema = new mongoose Schema({
  dateoperation: date,
  paymenttype: string,
  paymentdescription: string,
  sum: integer,
  datepayment: date,
})
```

```
const productSchema = new mongoose Schema({
  name: string,
  itemproduct: integer,
  supplier: string,
  productbrand: string,
  category: string,
  purchasedate: date,
  price: integer,
  quantityproduct: integer,
})
```

```
const serviceSchema = new mongoose Schema({
  name: string,
  description: string,
  category: string,
  serviceduration: integer,
  price: integer,
  employee: string,
})
```

```
const materialsSchema = new mongoose Schema({
  service: string,
  name: string,
  date: date,
  number: integer,
})
```

```
const certificateSchema = new mongoose Schema({
  name: string,
  sum: integer,
  client: string,
  service: string,
  startdate: date,
  enddate: date,
})
```

```
const subscriptionSchema = new mongoose Schema({
  name: string,
  sum: integer,
  client: string,
  service: string,
  startdate: date,
  enddate: date,
})
```

```
const discountSchema = new mongoose Schema({
  name: string,
  description: string,
  percentagediscount: integer,
  startdate: date,
  enddate: date,
})
```

```
const deliverySchema = new mongoose Schema({
  date: date,
  name: string,
```

```
description: string,  
product: string,  
supplier: string,  
category: string,  
price: integer,  
productquantity: integer,  
})
```

```
const orderSchema = new mongoose Schema({  
  datecreation: date  
  deliverydate: date  
  supplier: string,  
  status: string,  
  name: string,  
  quantity: integer,  
  price: integer,  
})
```

```
const returnSchema = new mongoose Schema({  
  name: string  
  date: date  
  supplier: string,  
  description: string,  
  status: string,  
})
```

```
const supplierSchema = new mongoose Schema({  
  name: string,  
  description: string,  
  country: string,  
  contact: string,  
})
```

```
const salarySchema = new mongoose Schema({  
  accrualdate: date,  
  description: string,
```

```
employee: string,  
accrued: integer,  
})
```

```
const Calendar = model('Calendar ', calendarSchema)  
const Task = model('Task', taskSchema)  
const Client = model('Client', clientSchema)  
const Request = model('Request', requestSchema)  
const Registration = model('Registration', registrationSchema)  
const Employee = model('Employee', employeeSchema)  
const Sale = model('Sale', saleSchema)  
const Transaction = model('Transaction', transactionSchema)  
const Product = model('Product', productSchema)  
const Service = model('Service', serviceSchema)  
const Materials = model('Materials', materialsSchema)  
const Certificate = model('Certificate', certificateSchema)  
const Subscription = model('Subscription', subscriptionSchema)  
const Discount = model('Discount', discountSchema)  
const Delivery = model('Delivery', deliverySchema)  
const Order = model('Order', orderSchema)  
const Return = model('Return', returnSchema)  
const Supplier = model('Supplier', supplierSchema)  
const Salary = model('Salary', salarySchema)
```

```
module.exports = {  
  Calendar,  
  Task,  
  Client,  
  Request,  
  Registration,  
  Employee,  
  Sale,  
  Transaction,  
  Product,  
  Service,  
  Materials,
```

Certificate,
Subscription,
Discount,
Delivery,
Order,
Return,
Supplier,
Salary,
};

ДОДАТОК В
Слайди презентації

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЗАПОРІЗЬКА ПОЛІТЕХНІКА”
КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ

Дипломна кваліфікаційна робота магістра
на тему «Дослідження та програмна реалізація
вебзастосунку для керування бізнес-процесами у
салоні краси»

Виконала: Пархоменко Вікторія Валеріївна, ст. гр. КНТ-142м
Керівник: Олійник Андрій Олександрович, д.т.н., професор НУ“ЗП”

2023

Рисунок В.1 – Титульний слайд

Об’єкт дослідження – бізнес-процеси у сфері надання послуг краси.

Предмет дослідження – вебзастосунок для керування бізнес-процесами у салоні краси.

Мета роботи – дослідження і розробка вебзастосунку, що надає функціональні можливості для керування бізнес-процесами у салоні краси.

Задачі магістерського дослідження:

- дослідження програмних рішень, які надають аналогічні функції;
- розробка технічного завдання;
- вибір засобів для розробки вебзастосунку;
- моделювання вимог до вебзастосунку для керування бізнес-процесами у салоні краси;
- проєктування структури навігації вебзастосунку для керування бізнес-процесами у салоні краси;
- проєктування архітектури вебзастосунку для керування бізнес-процесами у салоні краси;
- конструювання вебзастосунку для керування бізнес-процесами у салоні краси;
- тестування функціональності розробленого вебзастосунку.

2

Рисунок В.2 – Мета та задачі магістерського дослідження

Постановка завдань дипломної магістерської роботи

На основі проведеного дослідження аналогічних програмних рішень були сформовані функціональні можливості, які повинен надавати вебзастосунок для керування бізнес-процесами у салоні краси:

- аутентифікація та надання співробітнику після авторизації функцій для роботи відповідно до його посади у компанії;
- ведення клієнтської бази компанії;
- ведення бази завдань співробітників;
- можливість робити позначку, що завдання було виконане;
- ведення бази звернень від клієнтів;
- ведення бази записів клієнтів на послуги;
- ведення бази візитів клієнтів у календарі записів;
- ведення бази співробітників компанії;
- налаштування переліку можливостей співробітників відповідно до посади у компанії;
- ведення бази продажів товарів або послуг компанії;
- ведення бази фінансових операцій компанії;
- ведення бази товарів компанії;
- ведення бази послуг компанії;
- ведення бази списаних витратних матеріалів;
- ведення бази клієнтських сертифікатів та абонементів;
- ведення бази знижок для клієнтів;
- ведення бази постачань нових товарів;
- ведення бази замовлень постачальникам товарів;
- ведення бази записів повернень товарів постачальникам;
- ведення бази постачальників товарів;
- нарахування заробітної плати по співробітниках.

3

Рисунок В.3 – Постановка завдань дипломної магістерської роботи

Порівняння досліджених клієнт-серверних застосунків для салонів краси

Критерій порівняння	«RemOnline»	«Wlaunch»	«CleverBOX CRM»
Планувальник	+	+	+
База знань	+	+	+
Менеджер завдань	+	+	+
Чат підтримки клієнтів	+	–	+
Облік клієнтів	+	+	+
Облік товарів	+	+	+
Облік послуг	+	+	+
Складський облік	+	+	+
Контроль залишків товарів	+	+	+
Технологічні картки послуг	–	+	+
Фінансовий облік	+	+	+
База постачальників товарів	+	+	+
Аналітичні звіти	+	+	+
Облік клієнтських абонементів та сертифікатів	–	+	+
Додавання одиниць виміру для товарів компанії	+	+	+
Облік ліцензій клієнтів	+	–	+
Додавання полів компанії	–	+	–
Зберігання файлів	–	–	+
Облік рекламних	–	–	+
Налаштування форми опитувань та відображення результатів проведення опитувань	–	+	–
Модуль зарплати	+	+	+
Повідомлення та нагадування клієнтам	+	+	+
База співробітників	+	+	+
Налаштування розділів системи	+	+	+
Журнал дивників	+	–	+
Ключові показники компанії	+	+	+

4

Рисунок В.4 – Порівняння досліджених клієнт-серверних застосунків для салонів краси

Порівняння мов програмування для розробки вебзастосунків

Критерій порівняння	JS	C#	Python	PHP
Рішення різних завдань	++	++	++	—
Мультипарадигмальна мова	+	++	+	+
Інтерпретована мова	+	—	+	+
Динамічна типізація	+	—	+	+
Зручний синтаксис	+	+	+—	+—
Реєстрозалежна мова	+	+	+	+
Високорівнева мова	+	+	+	+
Наявність менеджера пакетів	+	+	+	+
Прототип-орієнтований стиль	+	—	—	—
Велика кількість програмних каркасів для розробки вебзастосунків	++	+	++	++
Відносно менші витрати часу	+	+—	+	+—
Висока швидкість виконання	+	+	—	+

5

Рисунок В.5 – Порівняння мов програмування для розробки вебзастосунків

Порівняння інструментів для розробки інтерфейсу користувача

Критерій порівняння	React	Vue	Angular
Мова програмування JS	+	+	—
Бібліотека	+	—	—
Віртуальна модель DOM	+	+	—
Двостороннє зв'язування даних	—	+	+
Швидке освоєння інструменту	+	++	—
Компонентний підхід розробки	+	+	+
Наявність вбудованого інструменту для маршрутизації вебсторінок	—	+	+
Наявність вбудованого інструменту для керування станами вебзастосунку	—	+	+
Розробка середніх вебпроектів	+	+—	++
Наявність документації	+	+	+
Вбудовані стилі компонентів	+	—	—
Робота з REST API	+	+	+
Висока швидкість кодування	+	++	—
Розширена екосистема	+	+	+

6

Рисунок В.6 – Порівняння інструментів для розробки інтерфейсу користувача

Порівняння фреймворків для розробки серверної частини

Критерій порівняння	Express.js	Koa.js	Napi.js
Високий рівень безпеки	—	—	+
Висока швидкість роботи	+	+	+
Розширюваність	+	+	+
Шаблонізація вебзастосунків	+	+	+
Модульність	+	+	+
Кросплатформність	+	+	+
Механізм маршрутизації	+	+—	+
Обробка запитів	+	+	+
Шаблон архітектури MVC	+	—	+
Аутентифікація користувачів	+	+	+
Швидке освоєння фреймворку	++	+—	+
Розробка середніх вебпроектів	+	+	+
Наявність документації	++	+—	+
Масштабованість	+	++	+

7

Рисунок В.7 – Порівняння фреймворків для розробки серверної частини

Порівняння текстових редакторів для розробки вебзастосунку

Критерій порівняння	Visual Studio Code	Atom
Кодування багатьма мовами	+	+
Перевірка правильності написання коду	+	+
Автоматична генерація каркаса проекту або окремих фрагментів програмного коду	+	+
Вбудована підтримка Git	+	+
Вбудований налагоджувач для Node.js	+	+
Встановлення розширень	+	+
Файловий менеджер	+	+
Діагностика помилок	+	+
Карта програмного коду	+	+
Розділення робочої області	+	+
Підтримка декількох курсорів	+	+
Підтримка IntelliSense	+	+
Вбудований інтерфейс командного рядка	+	+
Підтримка Emmet	+	+
Можливість роботи з базами даних	+	+
Високий рівень безпеки	+	—
Можливість роботи з декількома проектами одночасно	+	—
Наявність командного рядка для швидкого виконання різних команд	+	+
Рефакторинг	+	+
Інтегрований термінал	+	+
Можливість тестування	+	+

8

Рисунок В.8 – Порівняння текстових редакторів для розробки вебзастосунку

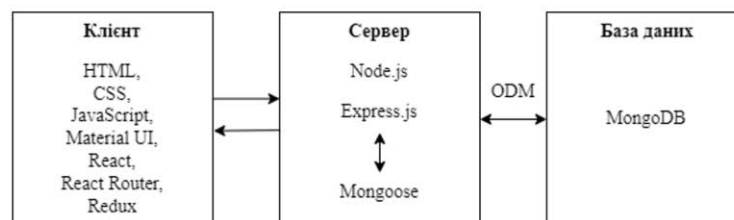
Порівняння нереляційних документо-орієнтованих СКБД

Критерій порівняння	MongoDB	CouchDB	Couchbase
Формат зберігання даних	BSON	JSON	JSON
Реплікації	++	+	++
Високий рівень безпеки	++	+	+
Індексування	++	+—	+
Агрегація даних	+	+	+
Мультиплатформність	+	+	+
Швидке освоєння СКБД	+	+	+
Простота використання	++	+	+
Засіб візуалізації даних	+	—	—
Розробка високонавантажених масштабованих систем зберігання даних	++	+—	++

9

Рисунок В.9 – Порівняння нереляційних документо-орієнтованих СКБД

Схема архітектури програмного продукту



10

Рисунок В.10 – Схема архітектури програмного продукту



Рисунок В.11 – Діаграми сценаріїв поведінки користувачів



Рисунок В.12 – Структура навігації вебзастосунку

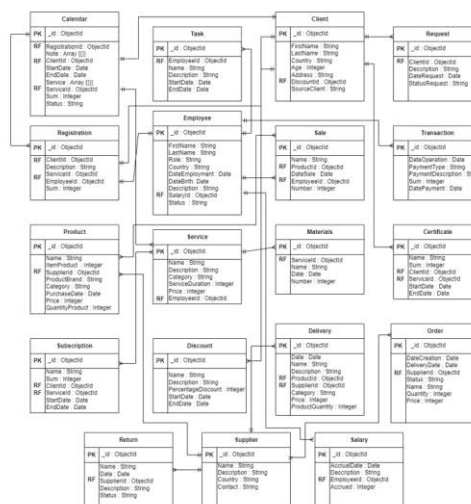
Розроблений алгоритм функціонування вебзастосунку



13

Рисунок В.13 – Розроблений алгоритм функціонування вебзастосунку

Фізична модель бази даних



14

Рисунок В.14 – Фізична модель бази даних

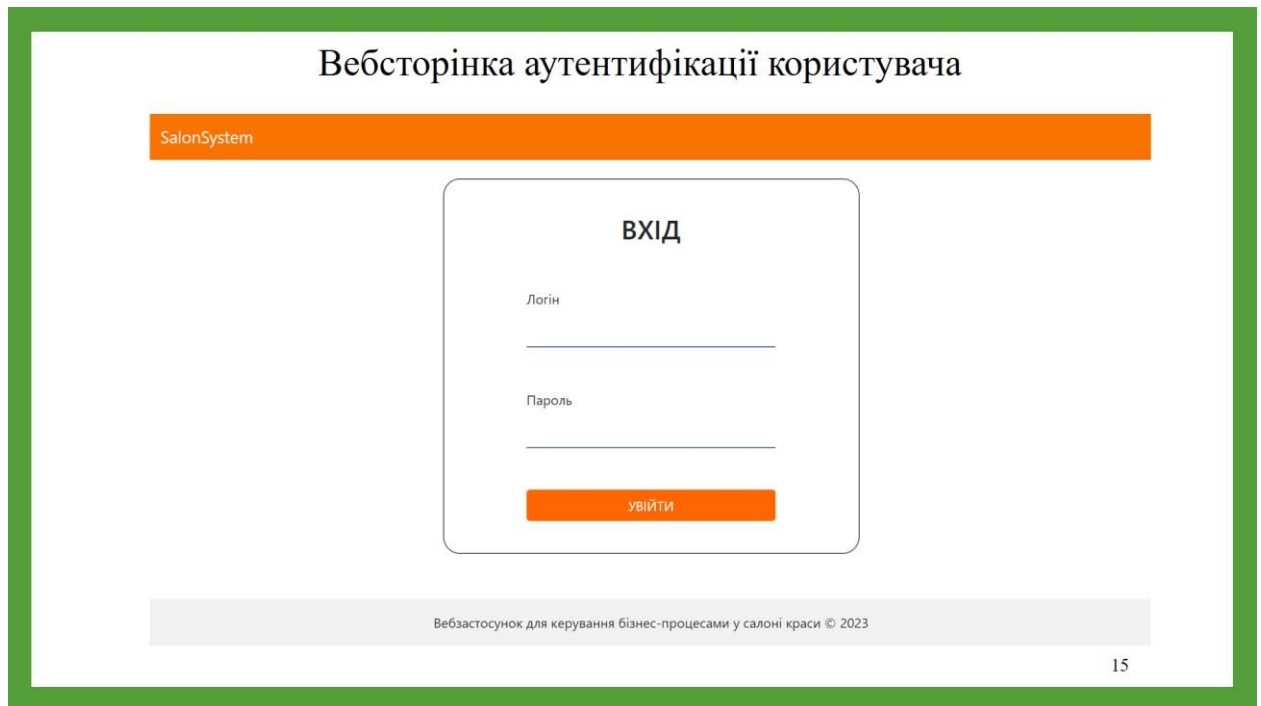


Рисунок В.15 – Вебсторінка аутентифікації користувача

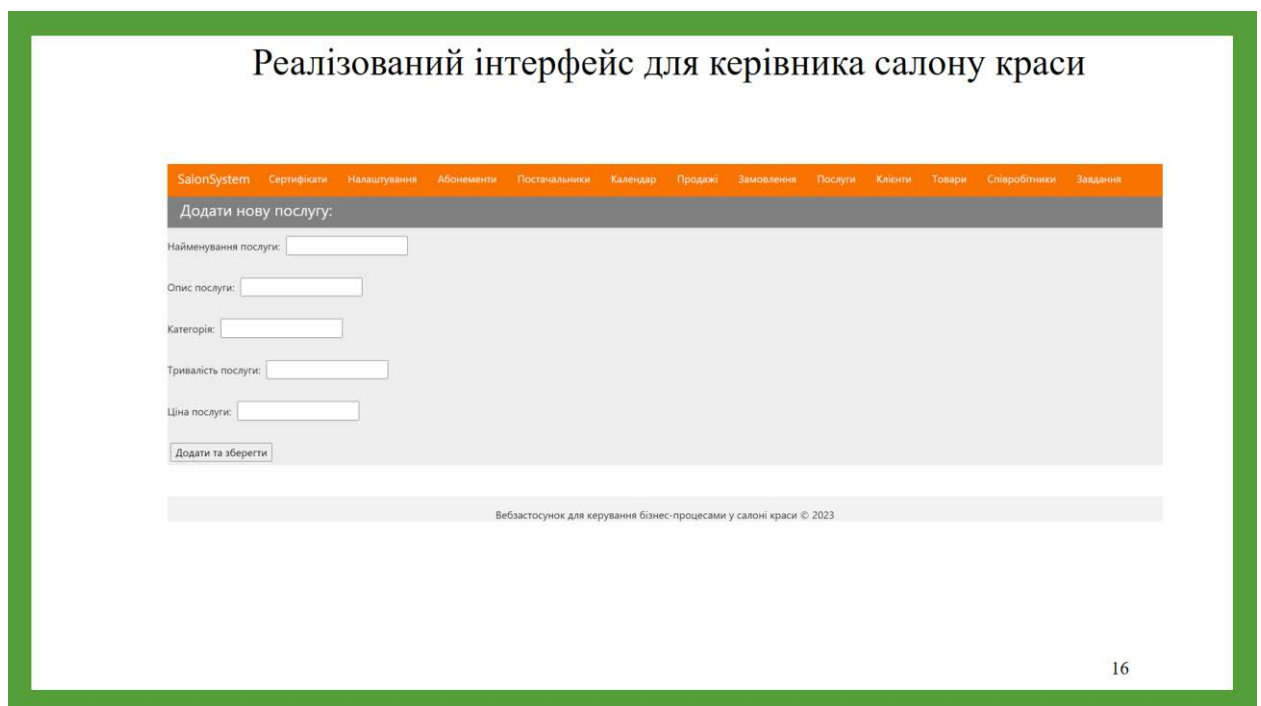


Рисунок В.16 – Реалізований інтерфейс для керівника салону краси

Реалізований інтерфейс для адміністратора салону краси

SalonSystem Сертифікати Налаштування Абонементи Постачальники Календар Продажі Замовлення Послуги Клієнти Товари Співробітники Завдання

Додати нового клієнта:

Ім'я:

Прізвище:

Країна:

Адреса:

Вік:

Дата народження: 01.05.2001

Додати та зберегти

Вебастосунк для керування бізнес-процесами у салоні краси © 2023

17

Рисунок В.17 – Реалізований інтерфейс для адміністратора салону краси

Реалізований інтерфейс для спеціаліста салону краси

SalonSystem Сертифікати Абонементи Календар Нарахування зарплатної плати Послуги Клієнти Знижки Співробітники Завдання

Додати нове завдання:

Найменування завдання:

Опис завдання:

Дата початку: 13.11.2023

Крайній срок: 02.12.2023

Додати та зберегти

Вебастосунк для керування бізнес-процесами у салоні краси © 2023

18

Рисунок В.18 – Реалізований інтерфейс для спеціаліста салону краси

Висновки

При виконанні дипломної магістерської роботи було досліджено та розроблено вебзастосунок для керування бізнес-процесами у салоні краси. Було визначено функції, якими володіють вебзастосунки для керування бізнес-процесами у салоні краси. Метою вебзастосунків для керування бізнес-процесами у салоні краси є автоматизація щоденних завдань співробітників салону краси та спрощення ведення обліку бізнес-процесів.

Для реалізації інтерфейсної частини вебзастосунку було обрано використовувати мову програмування JS, бібліотеку елементів React, бібліотеку компонентів Material UI, бібліотеку маршрутизації вебсторінок React Router та менеджер станів Redux. Для реалізації серверної частини вебзастосунку було обрано використовувати мову програмування JS, програмне середовище виконання Node.js, фреймворк Express.js, нереляційну СКБД MongoDB та бібліотеку для взаємодії з базою даних Mongoose.

Приведено набір діаграм сценаріїв використання вебзастосунку для керування бізнес-процесами у салоні краси, які моделюють бізнес-процеси. Виконано проєктування структури навігації вебзастосунку для керування бізнес-процесами у салоні краси. Оглянуто та виокремлено переваги обраного архітектурного рішення та програмної архітектури, а також побудовано відповідні схеми. Наведено структуру розроблених колекцій у СКБД MongoDB та розроблений алгоритм функціонування вебзастосунку для керування бізнес-процесами у салоні краси. Описано розроблені модулі, функції та наведено опис розроблених директорій вебзастосунку для керування бізнес-процесами у салоні краси.

У роботі науковою новизною є розроблені моделі взаємодії користувачів з вебзастосунком для керування бізнес-процесами у салоні краси, які наведені у вигляді діаграм сценаріїв поведінки користувачів із вебзастосунком. Тестування розроблених функцій вебзастосунку довело успішність виконання їх при роботі у різних веббраузерах.

Рисунок В.19 – Висновки