

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені І.І.МЕЧНИКОВА

(повне найменування вищого навчального закладу)

Факультет математики, фізики та інформаційних технологій

(повне найменування інституту, назва факультету (відділення))

Кафедра механіки, автоматизації та інформаційних технологій

(повна назва кафедри (предметної, циклової комісії))

Дипломна робота

на здобуття ступеня вищої освіти «бакалавр»

(освітньо-кваліфікаційний рівень)

на тему Розробка інформаційної системи керування взаємовідносинами
із клієнтами підприємства

Виконав: студент денної форми навчання

спеціальності 126 – Інформаційні системи та технології

(шифр і назва напрямку підготовки, спеціальності)

Петухов Данило Сергійович

(прізвище, ім'я, по-батькові)

Керівник

Михайло КОСОЙ

(науковий ступінь, вчене звання, прізвище та ініціали, підпис)

Рецензент

Олексій ЦАРЕНКО

(науковий ступінь, вчене звання, прізвище та ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

№ 10 від «10» 06 2022 р.

Завідувач кафедри



Віктор ВОЛКОВ

(підпис)

(прізвище, ініціали)

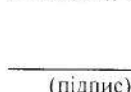
Захищено на засіданні ЕК №

протокол № від « » 2022 р.

Оцінка / /

(за національною шкалою, шкалою ECTS, бали)

Голова ЕК



Володимир ВИЧУЖАНІН

(підпис)

(прізвище, ініціали)

Одеса – 2022

АНОТАЦІЯ

Мета роботи - розробка інформаційної системи для предметної області «Салон краси», що забезпечить комфортне та незалежне розв'язання задач для співробітників, інвесторів та адміністраторів цієї системи. Відмінною особливістю даної системи є організація розкладу відвідування клієнтів, уникнення помилок людського фактора, автоматизація виконання рутинних завдань, покращення якості обслуговування клієнтів. Є деякі переваги, наприклад, нарахування заробітної плати працівникам салону, розрахунок статистики за потрібний період.

Для реалізації обрані такі технології: СУБД PostgreSQL, мову програмування C#, фреймворк для створення інтерфейсу WinForms, для підключення PostgreSQL система управління пакетами Npgsql.

Результатом дипломної роботи є працездатна система, що вирішує задачі користувачів.

ANNOTATION

The purpose of the work is to develop an information system for the subject area "Beauty Salon", which will provide a comfortable and independent solution for employees, investors and administrators of this system. A distinctive feature of this system is the organization of customer visits, avoidance of human error, automation of routine tasks, improving the quality of customer service. There are some advantages, for example, accrual of salaries to employees of salon, calculation of statistics for the necessary period.

The following technologies have been selected for implementation: PostgreSQL database, C # programming language, framework for creating the WinForms interface, Npgsql package management system for PostgreSQL connection.

The result of the thesis is a working system that solves user problems.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	5
ВСТУП	6
1 ОГЛЯД ПРОБЛЕМНОЇ СИТУАЦІЇ ЩОДО КЕРУВАННЯ ВЗАЄМОДІЄЮ ІЗ КЛІЄНТАМИ ПІДПРИЄМСТВ	6
1.1 Огляд існуючих рішень задач, виникаючих з означеної проблеми	5
1.2 Характеристика топ-5 програмних продуктів crm-систем в Україні.....	11
2 АНАЛІЗ РОБОТИ ПІДПРИЄМСТВА ІЗ МЕТОЮ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЮ СИСТЕМОЮ, ВИЗНАЧЕННЯ ЗАДАЧ, ЩО БУДУТЬ РЕАЛІЗОВАНІ CRM-СИСТЕМОЮ	17
3 ПРОЕКТУВАННЯ ТА СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	21
3.1 Інформаційне моделювання предметної області.....	22
3.2 Модель додатку.....	28
3.3 Створення бази даних	31
3.4 Запити до бази даних для вирішення поставлених завдань.....	32
3.5 Програмна реалізація інтерфейсу	34
4 ПИТАННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ СИСТЕМИ. ІНСТРУКЦІЯ ДЛЯ КОРИСТУВАЧА	38
ВИСНОВКИ	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47
ДОДАТОК А (Перелік задач користувачів інформаційної системи)	48
ДОДАТОК Б (Запити на створення таблиць бази даних).....	50
ДОДАТОК В (Запити на створення тригерів та функцій).....	53

ПЕРЕЛІК СКОРОЧЕНЬ

IDE - Інтегроване середовище розробки.

СУБД - Система управління базами даних.

ПрО - Предметна область.

ВСТУП

У сучасному бізнесі необхідність автоматизації різних процесів стала вже звичним явищем. Складно уявити собі складський або бухгалтерський облік без застосування спеціалізованого програмного забезпечення, торгові представники використовують спеціальні додатки для оформлення та відправки замовлення в офіс прямо з планшету або мобільного телефону, досить велика частина замовлень надходить із сайту вже у вигляді готових до обробки документів. Але при цьому взаємини з клієнтами, принаймні в середньому та малому бізнесі, чомусь дуже часто ведуться без впровадження автоматизації та достатньої уваги до обліку.

Вихід із ситуації – автоматизація і стандартизація управління відносин із клієнтами, тобто. впровадження інформаційної системи.

Мета цієї дипломної роботи – розробка інформаційної системи для предметної області «Салон краси», що забезпечить комфортне та незалежне розв’язання задач для співробітників, інвесторів та адміністраторів цієї системи.

Для того, щоб досягти поставленої мети, були сформульовані наступні завдання, які необхідно виконати в процесі проектування:

- 1) виконати аналіз предметної області «Салон краси»;
- 2) визначити категорії користувачів та задачі, які інформаційна система повинна розв’язувати;
- 3) обрати архітектуру та шаблони проектування системи;
- 4) вибрати засоби реалізації;
- 5) спроектувати функціональну модель;
- 6) спроектувати базу даних залежно від сутностей та задач предметної області;
- 7) реалізувати зручний користувацький інтерфейс;

1 ОГЛЯД ПРОБЛЕМНОЇ СИТУАЦІЇ ЩОДО КЕРУВАННЯ ВЗАЄМОДІЄЮ ІЗ КЛІЄНТАМИ ПІДПРИЄМСТВ

Сучасна ринкова ситуація характеризується лавіруванням підприємств між впливом зовнішніх кризових викликів, і проблемами внутрішнього характеру.

У таких умовах формування на підприємстві ефективної системи управління взаємовідносинами з клієнтами є об'єктивною відповіддю на кризові виклики сьогодення, спрямованою на підвищення конкурентоспроможності шляхом збереження існуючих клієнтів, через задоволення їхніх потреб і підвищення лояльності, пошук нових клієнтів та ефективну взаємодію з ними.

Однак впровадження такої системи досить часто є неефективним внаслідок нерозуміння того, що для ефективного управління підприємством необхідно використовувати системний підхід до формування не тільки інформаційного, але й економічного, організаційного та інших видів забезпечення. Незважаючи на те, що основна складова системи взаємовідносин з клієнтами — це технологія CRM (Customer Relationship Management), використання її суто як програмного продукту може призвести до провалу, оскільки успішне впровадження та управління системою вимагає інтегрованого та збалансованого підходу до технології, процесу та людей.

Будь-який бізнес передбачає взаємодію між компанією та її стейкхолдерами. В умовах зростаючої конкуренції проблема управління цією взаємодією набирає особливої актуальності. Зокрема, стосунки з клієнтами перетворюються в один з найважливіших активів компаній, що вимагає зміщення акцентів з їхньої продукції на клієнтів. У розвинених ринкових економіках уся діяльність зі створення і споживання благ наслідує принципи підприємництва. Підприємництво все більше інституціоналізується в інформаційному полі економіки, набуваючи не тільки суто виробничого характеру. Як наслідок, багато організацій відходять від продуктово-

орієнтованого та бренд-орієнтованого маркетингу до клієнторієнтованого підходу. За таких умов і виникла концепція управління взаємовідносинами з клієнтами (Customer Relationship Management) як спосіб вирішення поставлених завдань. Так, Н. С. Краснокутська та Р. С. Тихонченко, виділили два підходи до розуміння змісту управління взаємовідносинами з клієнтами.

1.1 Огляд існуючих рішень задач, виникаючих з означеної проблеми

Перший підхід передбачає розгляд змісту управління взаємовідносинами з клієнтами з точки зору організаційного процесу, спрямованого на усунення бар'єрів між клієнтами і виробниками за рахунок їх взаємної участі у виробництві та споживанні товарів і послуг. Така взаємна участь набуває форми колабораційної діяльності зі створення загальної ціннісної пропозиції. При цьому корисність виробника для суспільства без лояльних клієнтів набагато менша, ніж за умов їх наявності. Другий підхід — розгляд змісту управління взаємовідносинами з клієнтами з позиції здатності до організації діалогу з усіма контактними аудиторіями клієнтів. Мета управління, заснованого на такому діалозі, полягає в інтеграції інформації з різних джерел продажу, обслуговування клієнтів і співробітництво партнерів у ланцюжку створення цінності.

Вибудовування власного бачення теорії досліджуваного питання дає змогу стверджувати, що управління взаємовідносинами з клієнтами має розглядатися саме з позиції стратегічного проактивного клієнтоорієнтованого підходу. Оскільки, зважаючи на виклики сьогодення (політичні, суспільні, економічні), клієнти, за умов високої обізнаності, значної диференціації продукції, товарів і послуг, можливості швидко знаходити альтернативу задоволення своїх потреб, стають джерелом грошових потоків компанії. Управлінці, які дозволять собі повільний рух у напрямку розвитку проактивності та клієнтоорієнтованості своїх компаній, ризикують одержати стратегічний розрив між результатами своєї діяльності і

позиціями своїх конкурентів. Основними складовими системи управління взаємовідносинами з клієнтами вважаємо: управління знаннями про клієнта; маркетингову політику просування продукції; систему продажів; створення цінності для клієнта; цінову політику; інформаційне середовище взаємовідносин з клієнтами. Тому, на нашу думку, система управління взаємовідносинами з клієнтами — це комплекс взаємопов'язаних елементів (кількісних і якісних) діяльності підприємства в межах його клієнтоорієнтованої стратегії, інтегрованих в єдине інформаційне середовище, які слугують для автоматизації, оптимізації та підвищення ефективності бізнес-процесів взаємодії з клієнтами та спрямовані на формування клієнтського капіталу підприємства. З огляду на вищезазначене, була визначена й узагальнена така проблемна симптоматика взаємодії підприємств харчової промисловості з клієнтами, що спричиняє падіння прибутковості: - відсутність ефективної системи планування продажів та несформованість системи управління взаємовідносинами з клієнтами, що проявляється в недостатній аналітиці потреб кінцевого клієнта і призводить до неефективного управління життєвим циклом клієнта; - використання застарілих та неорієнтованих на кінцевого клієнта методів продажів, прибутковість яких сумнівна, а вплив на репутацію компанії руйнівний; невідповідність цих методів місії та цінностям компанії; - відсутність маркетингової стратегії в роздрібній мережі; - неефективне використання можливостей продажів у мережі Інтернет; - неефективна система управління змінами, наявність опору змінам збоку менеджменту; затягування процесів прийняття рішень. Усе це призводить до повільного реагування та адаптації компаній на зміни зовнішнього середовища, до зниження кількості постійних клієнтів. Основна мета формування системи управління взаємовідносинами з клієнтами має полягати в тому, щоб, ґрунтуючись на детальних знаннях про клієнтів, допомогти підприємствам краще їх розуміти, будувати відносини з ними, забезпечувати збереження клієнтів, а отже, прибутковість діяльності. Адже залучення нових клієнтів коштує підприємству значно дорожче, ніж

утримання існуючих. Детальні знання про клієнтів дають змогу провести сегментацію залежно від їхніх потреб та майбутньої економічної цінності для підприємства. Враховуючи вищенаведене, в концепції проактивної клієнтоорієнтованої стратегії підприємства можна виділити чотири групи цілей: - встановлення взаємовигідних і ефективних відносин з усіма групами клієнтів; - збільшення й утримання бізнес-партнерів (клієнтів) підприємства шляхом задоволення споживчих переваг на основі ціннісного підходу; - формування позитивної репутації компанії як клієнтоорієнтованої складової її розвитку; - зростання вартості компанії та збалансований її розвиток у довгостроковому періоді.

На нашу думку, систему управління взаємовідносинами з клієнтами слід розглядати з позиції стратегічного проактивного клієнтоорієнтованого підходу. Це сприятиме гнучкому реагуванню на екзогенні та ендогенні фактори впливу на діяльність підприємств, дасть змогу перетворити стосунки з клієнтами в один з найважливіших активів компаній, сформувати клієнтський капітал підприємства та, врешті-решт, забезпечити поліваріантність ефективних управлінських рішень.

1.2 Характеристика топ-5 програмних продуктів crm-систем в Україні

На світовому ринку, а останніми роками і на внутрішньому представлено величезне різноманіття програмних продуктів CRM-систем різних виробників, тому розглянемо найпопулярніші в Україні.

1) “Sales Creatio” – це готовий продукт для прискорення та автоматизації повного циклу продажів, а саме: від ліда до повторних замовлень. Окрім того, є конфігурації для управління сервісом, маркетингом та бізнес-процесами. На онлайн-майданчику Marketplace присутні готові доповнення і галузеві рішення, шаблони та коннектори для розширення можливостей платформи і автоматизації різноманітних бізнес-задач (рис. 1.2.1).

Основні можливості системи:

- Внутрішня телефонія;
- Сегментація контактної бази;
- Управління документообігом;
- Корпоративна соціальна мережа;
- Управління замовленнями, лідами та рахунками;
- Каталогізація товарів;
- Автоматизація бізнес-процесів;
- Пошук профілю контакту в соцмережах.

Вартість: реалізація цієї CRM-системи коштує від \$22 за 1 користувача на місяць. Пробна безкоштовна версія є доступною протягом 14 днів.

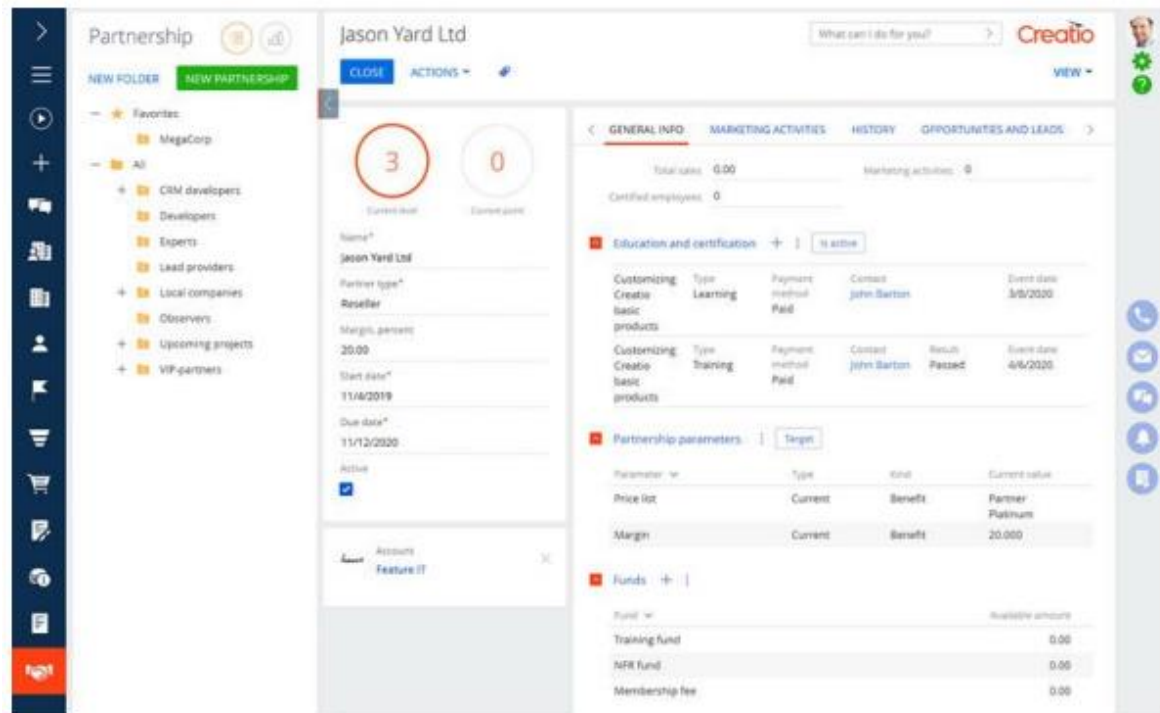


Рисунок 1.2.1 – Інтерфейс CRM-системи “Sales Creatio”

2) “Бітрікс24” – це соціальний інтранет за схемою SaaS. Містить календар, мікроблоги, завдання, месенджер, фотогалереї, CRM, бізнес-процеси, файлоховище, екстранет, профайли, облік робочого часу та звіти. Існує мобільна та безкоштовна версія (рис. 1.2.2).

Основні можливості системи:

- Обмін даними з 1С;
- Управління лідами та контрактами;
- Виставлення рахунків;
- Управління проектами та завданнями;
- Налаштування своїх стадій та воронки продажів;
- Друк та відправлення документів з CRM;
- Створення сайтів, лендінгів із вбудованою CRM;
- Налаштування планів продажів та звітів;
- Налаштування регулярних контрактів;
- Автоматизація бізнес-процесів;

– Налаштування реклами, тригерних листів, смс-повідомлень та робіт безпосередньо із CRM.

Вартість: команда, що складається з 12 чоловік має змогу використовувати базові функції системи абсолютно безкоштовно. Мінімальний платний тариф складає \$16. В свою чергу, для відділу продажів є спеціальний тариф CRM+ вартістю \$50 на місяць.

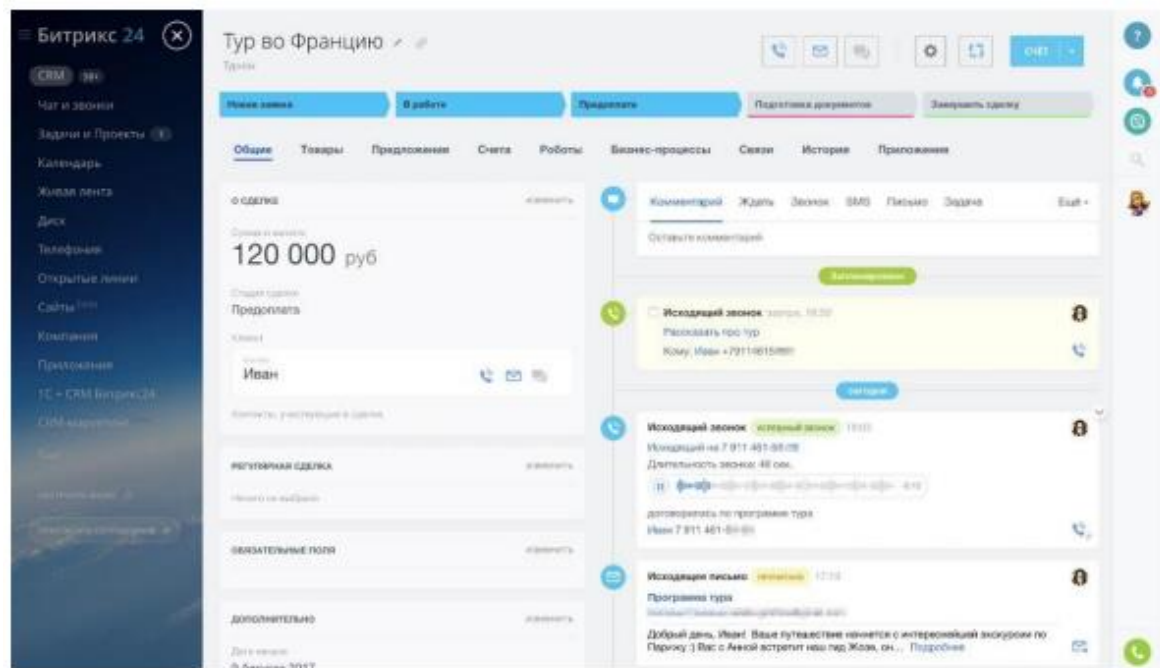


Рисунок 1.2.2 – Інтерфейс CRM-системи “Бітрікс24”

3) Microsoft Dynamics CRM – це CRM-система з впізнаваним майкрософтським інтерфейсом. Доступна повна інтеграція з додатками та платформою Microsoft. Має інструменти для управління бізнес-процесами, маркетингом, продажами та сервісом.

Веб-інтерфейс є доступним через Internet Explorer. Існує мобільна версія. Є можливість роботи з системою прямо з Outlook (рис. 1.2.3).

Основні можливості системи:

- Доступні iOS, Android, Windows та веб-додатки;
- Візуалізація даних на розширеному дашборді;
- Створення шаблонів файлів Word та Excel для швидких звітів;

- Управління контактами, підтримкою, продажами, маркетингом та іншими даними в одній системі;
 - Автоматизація процесів обробки даних за допомогою сценаріїв.
- Вартість: від \$65 на місяць за 1 користувача.

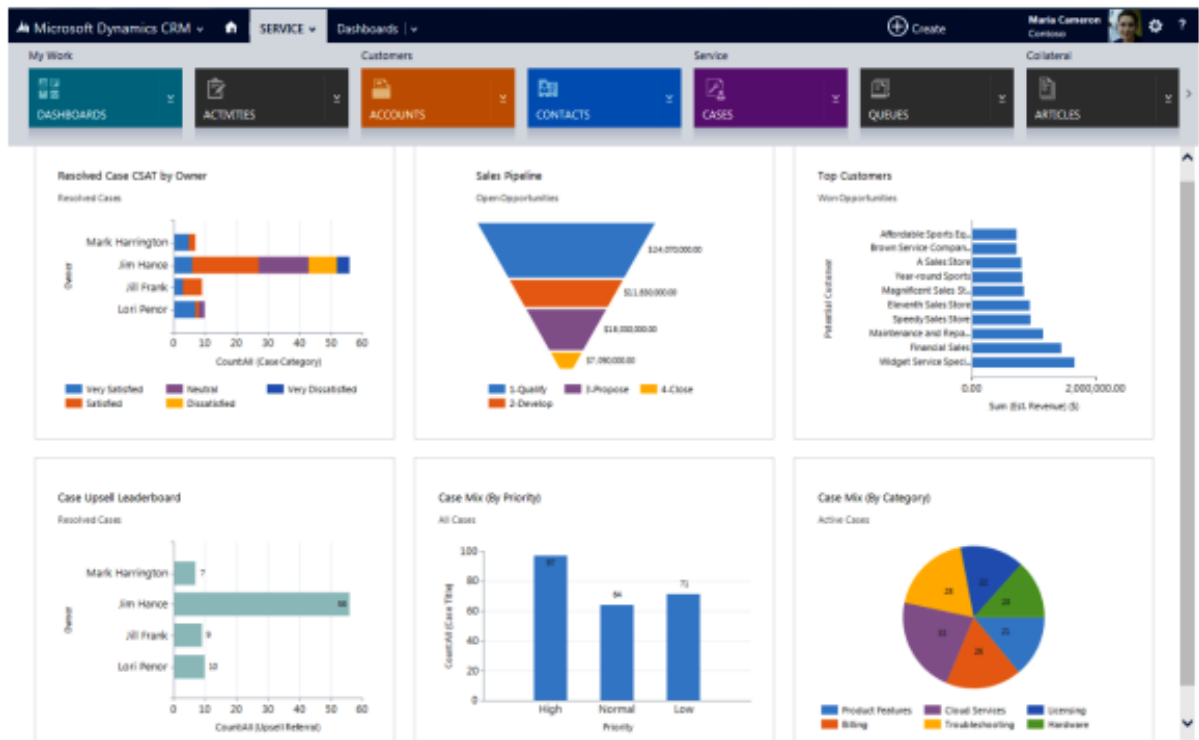


Рисунок 1.2.3 – Інтерфейс CRM-системи “Microsoft Dynamics CRM”

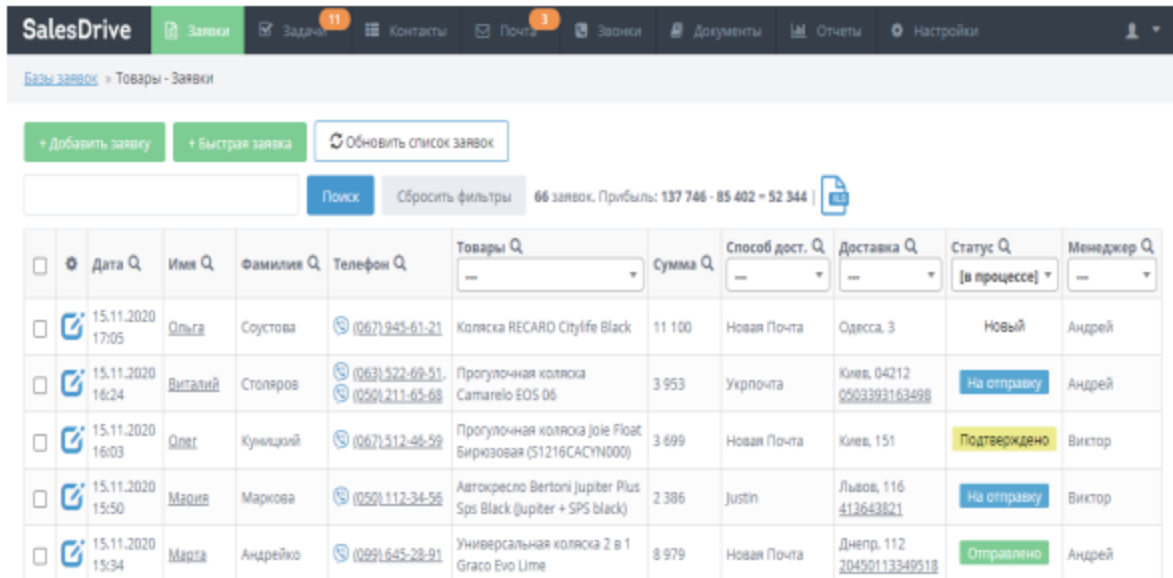
4) SalesDrive – це система управління продажами. Є можливість налаштування під власний бізнес. В наявності готові CRM-рішення для різноманітних галузей.

Повна інтеграція з сайтом, sms-повідомленнями, електронною поштою та телефонією. Дану CRM-систему найкраще використовувати для інтернетмагазинів (рис. 1.2.4).

Основні можливості системи:

- Прогнозування продажів;
- Автоматизація робочих процесів;
- Потужна аналітика;

- Різноманітні налаштовувані інструменти для сумісної роботи та управління проектами;
 - Повноцінне управління контактними даними.
- Вартість: від \$25 на місяць (до 10 користувачів).



	Дата	Ім'я	Фамілія	Телефон	Товари	Сумма	Спосіб дост.	Доставка	Статус	Менеджер
☐	15.11.2020 17:05	Ольга	Соустова	(067) 945-61-21	Коляска RECARO Citylife Black	11 100	Новая Почта	Одесса, 3	Новый	Андрей
☐	15.11.2020 16:24	Виталий	Столяров	(063) 522-69-51 (050) 211-65-68	Прогулочная коляска Camarelo EOS 06	3 953	Укрпочта	Киев, 04212 6503393163498	На отправку	Андрей
☐	15.11.2020 16:03	Олег	Хунищкий	(067) 512-46-59	Прогулочная коляска Jole Float Бирюзовая (51216CACYN000)	3 699	Новая Почта	Киев, 151	Подтверждено	Виктор
☐	15.11.2020 15:50	Мария	Маркова	(050) 112-34-56	Автомобиль Berton Jupiter Plus Sps Black (Jupiter + SPS black)	2 386	Justn	Львов, 116 413643821	На отправку	Виктор
☐	15.11.2020 15:34	Мария	Андрейко	(099) 645-28-91	Универсальная коляска 2 в 1 Graco Evo Lime	8 979	Новая Почта	Днепр, 112 20450113349518	Отправлено	Андрей

Рисунок 1.2.4 – Інтерфейс CRM-системи “SalesDrive”

5) Keeping CRM – це онлайн CRM-система, що призначена для малого та середнього бізнесу. Це достатньо гнучкий, інтуїтивний, зручний та простий у користуванні інструмент для оптимізації роботи з складом, прайс-листом, угодами, клієнтами, завданнями, лідами, введенням фінансів та документів (рис. 1.2.5).

Основні можливості системи:

- Візуалізація продажів;
- Автоматизація роботи зі складом та підрядниками;
- Управління прибутками та витратами;
- Інтеграції;
- Управління поточними завданнями та контроль за їх виконанням;
- Формування картки клієнта в один клік.

Вартість: до двох користувачів – безкоштовно, в подальшому ціна зростає в залежності від кількості додаткових користувачів. Мінімальний тариф складає \$17 за трьох користувачів на рік.

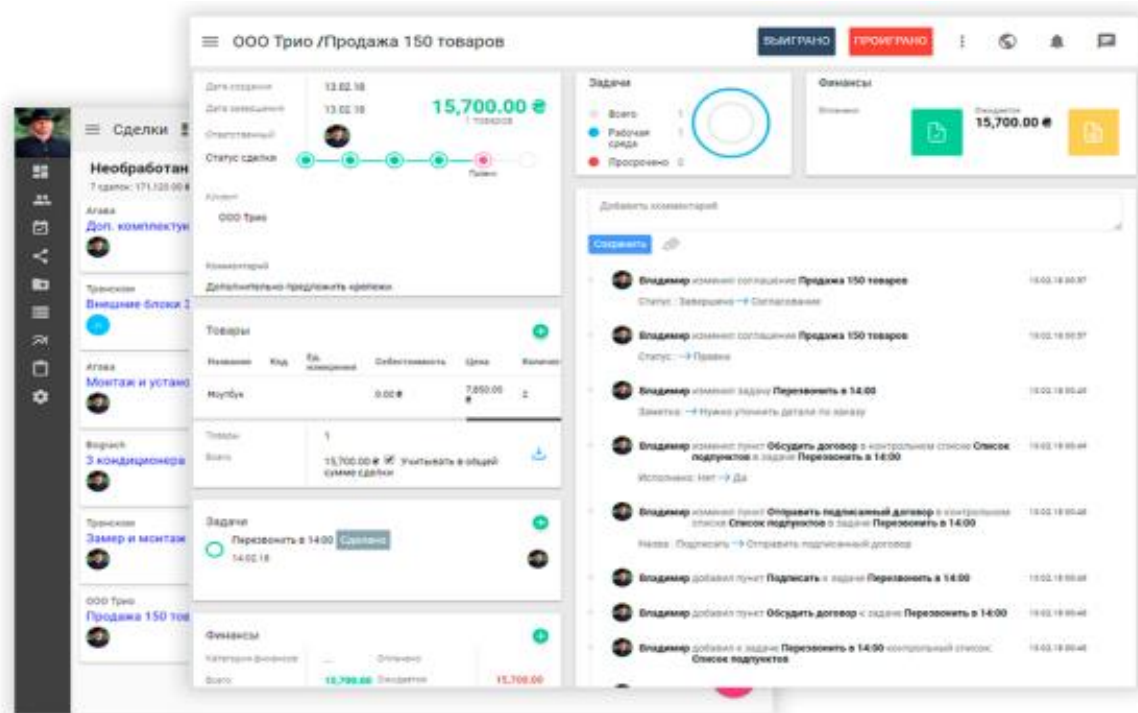


Рисунок 1.2.5 – Интерфейс CRM-системы “Keeping CRM”

2 АНАЛІЗ РОБОТИ ПІДПРИЄМСТВА ІЗ МЕТОЮ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЮ СИСТЕМОЮ, ВИЗНАЧЕННЯ ЗАДАЧ, ЩО БУДУТЬ РЕАЛІЗОВАНІ CRM-СИСТЕМОЮ

Основними функціями та можливостями CRM-системи є:

1) Облік клієнтів – CRM-система формує єдину базу даних для клієнтів та контрагентів компанії, де реєструються всі необхідні реквізити, повна історія взаємодій та покупок, а також канали зв'язку. Наявні контактні дані в базі є можливість сегментувати, фільтрувати та здійснювати управління за різноманітними критеріями.

2) Управління продажами – CRM-система здійснює ведення даних щодо реальних та потенційних угод, аж до поетапного контролю кожної операції, реалізовується автоматизація воронки продажів та частково бізнес-процесів компанії.

3) Аналітичні функції – опираючись на масив зібраної інформації CRM-система формує різноманітні статистичні звіти, відповідно до яких є можливість в повній мірі оцінити ефективність роботи співробітників, маркетингових каналів, заходів, планувати певні зміни та здійснювати прогноз продажів.

Варто також зазначити, що у кожній компанії, яка застосовує CRM, функції та завдання такої системи можуть досить кардинально відрізнятися. Отже, необхідно обов'язково визначити бізнес-вимоги до CRM-системи, цілі для яких планується її використання, оскільки процес вибору для сервісного центру або ж відділу продажів буде різним.

Можливості сучасних CRM-систем (рис. 2.1):

- Контроль за роботою менеджерів;
- Повний доступ до актуальної інформації про клієнтів;
- Оперативне обслуговування клієнтів та здійснення операцій;
- Швидкий доступ до необхідної аналітичної інформації та звітних даних;

- Скоординована взаємодія між підрозділами та співробітниками;
- Скорочення операційних витрат менеджерів;
- Автоматизація документообігу – в системі є можливість ввести шаблони будь-яких документів, що використовуються в компанії і при цьому зникає необхідність у ручному формуванні нового документа;
- Управління бізнес-процесами – дає змогу автоматизувати послідовні операції, що здійснюються співробітниками компанії;
- Управління та планування процесом доставки та закупівлі – в системі менеджери можуть постійно контролювати кількість та наявність товару на складі. В свою чергу, співробітники мають змогу відслідковувати виконання плану закупівель;
- Управління маркетингом – система дає змогу здійснювати управління маркетинговими заходами та визначати їх результативність (пряма розсилка, електронна розсилка та sms-розсилка). Варто також зазначити, що є можливість сегментації клієнтів (наявних та потенційних), які знаходяться в базі, за певними параметрами для реалізації маркетингових заходів;
- Управління та планування продажами – система дає змогу формувати плани за різноманітними показниками (наприклад, дохід по відділах, менеджерах, послугам, продуктам). Таким чином, можна побудувати воронку продажів, яка дозволить визначити проблемні зони в циклах продажів;
- Історія взаємодії з клієнтами та управління контактами – це база даних усіх контрагентів компанії, а саме: конкурентів, постачальників та клієнтів, з внесеною детальною інформацією про них, їх співробітників і т.д.

Варто також зазначити, що система дає змогу здійснювати швидкий пошук важливої інформації щодо контрагентів, а також отримувати всю необхідну інформацію стосовно листування, зустрічей, угод та переговорів.



Рисунок 2.1 – Основні можливості сучасних CRM-систем

Принципи сучасних CRM-систем:

1) Аналіз зібраної інформації щодо клієнтів та прийняття організаційних рішень, наприклад, “розподіл” клієнтів на основі їхньої прибутковості для компанії;

2) Наявність єдиного сховища даних, звідки у будь-який момент доступна інформація щодо усіх випадків взаємодії компанії з контрагентами (клієнтами, постачальниками та конкурентами);

3) Синхронізація управління та взаємодії між підрозділами компанії. Відповідно до функціонального призначення та застосування CRM-систем розрізняють три основні категорії (табл. 1.2):

1) Операційний CRM – відповідає за реєстрацію і оперативний доступ до первинної інформації стосовно контактних даних, проектів, подій тощо.

Основними функціональними можливостями даного типу CRM-систем є: контроль процесу обробки складних та тривалих звернень; формалізація усіх бізнес-процесів взаємодії з клієнтом; контроль та планування

комунікацій з клієнтами; класифікація, сортування та пошук максимальної інформації щодо клієнта.

Варто також зазначити, що даний тип CRM-систем найчастіше використовується в компанії, яка має досить тривалі проекти з багатьма етапами, в яких приймають участь декілька співробітників або навіть відділів.

Можливості аналітики подібних систем є досить обмеженими, адже через рідкісні звернення клієнтів в них просто немає потреби. В даному випадку наявний аналіз щодо кожного етапу роботи, оцінки роботи співробітників та вибірки по клієнтам.

CRM-системи даного типу можуть працювати цілком автономно, при чому інтеграція з іншим програмним забезпеченням є не критичним.

2) Аналітичний CRM – відповідає за звітність та аналіз інформації в різних ракурсах. До функціональних можливостей даного типу системи можна віднести: пошук закономірностей з метою формування найефективнішої стратегії продажів, маркетингу та обслуговування клієнтів; синхронізація розрізнених масивів даних. Аналітичний CRM вимагає чималого об'єму напрацьованих статистичних даних, відмінної інтеграції систем, а також якісного аналітичного інструментарію.

Потенційними користувачами подібних систем є компанії, у яких досить велика кількість звернень в одиницю часу, проте кожна взаємодія з клієнтом є короткою. В даному випадку для організації характерним є широкий асортимент продукції та послуг, а також значне число клієнтів. Найважливішим модулем в системах цього типу є аналітичний інструмент, що дає змогу працювати з чималим обсягом різноманітної інформації.

3) Коллаборативний CRM – передбачає досить тісний контакт із клієнтами та кінцевими споживачами, які мають можливість частково впливати на внутрішні процеси компанії.

Наприклад, це можуть бути опитування, що направлені на зміну якостей продукту (послуги) або ж порядку обслуговування;

3 ПРОЕКТУВАННЯ ТА СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Для реалізації інформаційної системи була обрана дворівнева архітектура клієнт-сервера (рис. 3.1).

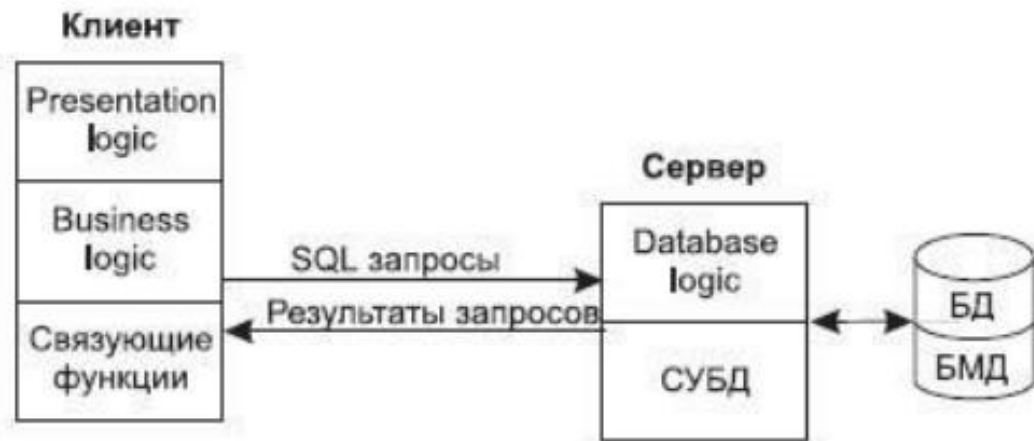


Рисунок 3.1 – Схема роботи двохрівневої архітектури клієнт-сервера

Обчислювальна або мережна архітектура, в якій завдання або мережеве навантаження розподілені між постачальниками послуг, які називаються серверами, та замовниками послуг, які називають клієнтами. Зазвичай ці програми розташовані на різних обчислювальних машинах і взаємодіють між собою через обчислювальну мережу за допомогою мережевих протоколів, але можуть бути розташовані також і на одній машині.

Серед шаблонів проектування був обраний шаблон MVP, який уявляє собою розділення системи на три рівня з метою відокремлення відображення від логіки.

Цей підхід дозволяє створювати абстракцію уявлення. Для цього необхідно виділити інтерфейс уявлення з певним набором властивостей та методів. Презентер, у свою чергу, отримує посилання на реалізацію інтерфейсу, підписується на події подання та на запит змінює модель.

Для проектування такої архітектури був вибраний паттерн MVP (рис. 3.2).

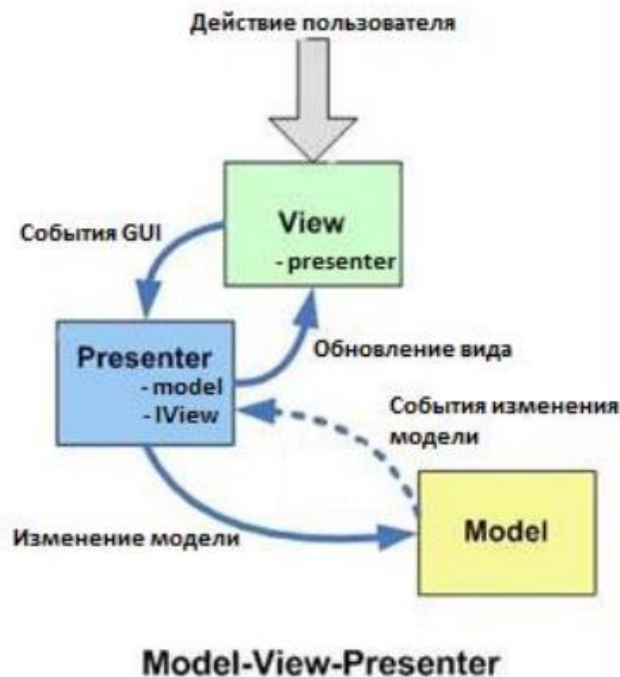


Рисунок 3.2 – Схема роботи паттерна MVP

MVP має низку переваг, власне і чому він був обраний.

Головні переваги такого поділу:

- 1) Чіткий поділ обов'язків між компонентами. Такий поділ дозволяє спростити розуміння та обслуговування бази коду.
- 2) Такий шаблон робить систему доволі стабільною до змін, тому що дозволяє змінювати або навіть видаляти гілки додатку та не заважати роботі інших гілок, які не потребують змін.
- 3) Легше тестування. Оскільки між компонентами є чітко визначені межі, стає набагато простіше тестувати кожен компонент ізольовано.

3.1 ІНФОРМАЦІЙНЕ МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Таблиця 3.1.1 - Опис сутностей ПрО «Салон краси»

Ім'я атрибуту	Призначення атрибуту	Обмеження
Клієнти (client)		
client_id	ідентифікатор клієнта	Первинний ключ
client_name	ім'я	Не порожнє
client_surname	прізвище	Не порожнє
client_patronymic	по батькові	
gender	стать	Не порожнє
phone	телефон	Не порожнє
notice_bool	повідомлення	
birthday	день народження	
num	номер	Не порожнє
status	статус	Не порожнє
note	нотатки	
link_photo	посилання на фото	
Працівник (worker)		
worker_id	ідентифікатор працівника	Первинний ключ
worker_name	ім'я	Не порожнє
worker_surname	прізвище	Не порожнє
worker_patronymic	по батькові	
phone	телефон	Не порожнє
birthday	день народження	
color	персональний колір	Не порожнє
worker_money	гроші	Має бути не менше нуля
worker_positions	Ідентифікатор посади	Зовнішній ключ для зв'язку з сутністю positions (positions_id)

Продовження таблиці 3.1.1

Посада (positions)		
position_id	ідентифікатор посади	Первинний ключ
position_name	назва посади	Не порожнє
rate	ставка	Не порожнє
percentage_of_services	відсоток від послуг	Не порожнє
percentage_of_sales	відсоток від продажу	Не порожнє
Візит(visit)		
visit_id	ідентифікатор візиту	Первинний ключ
cabinet	номер кабінету	Не порожнє
visit_date	дата візиту	Не порожнє
worker_id	ідентифікатор працівника	Зовнішній ключ для зв'язку з сутністю worker (worker_id)
client_id	ідентифікатор клієнта	Зовнішній ключ для зв'язку з сутністю client (client_id)
final_amount	кінцева сума	Не порожнє
time_start	час початку процедури	Не порожнє
time_end	час кінця процедури	Не порожнє
visit_status	при додаванні процедури її присвоюється статус "not confirmed", після оплати він змінюється на "completed"	Не порожнє , можуть бути тільки такі значення: “ completed ”, “ not confirmed ”
notification	показує, чи повідомлений клієнт про процедуру	Не порожнє

Продовження таблиці 3.1.1

Електроепіляція (electro)		
electro_id	ідентифікатор електроепіляції	Первинний ключ
electro_name	статус	Не порожнє
price_per_minute	ціна за хвилину	Не порожнє
Лазерна епіляція (laser)		
laser_id	ідентифікатор лазерної епіляції	Первинний ключ
laser_name	назва процедури	Не порожнє
laser_time	тривалість процедури	Не порожнє
price_woman	вартість для жінок	Не порожнє
price_man	вартість для чоловіків	Не порожнє
bonus	бонус	Не порожнє
Масаж (massage)		
massage_id	ідентифікатор масажу	Первинний ключ
massage_name	назва процедури	Не порожнє
massage_time	тривалість процедури	Не порожнє
price	вартість	Не порожнє
bonus	бонус	Не порожнє
Шугаринг (shugaring)		
shugaring_id	ідентифікатор шугарінгу	Первинний ключ
shugaring_name	назва процедури	Не порожнє
shugaring_time	тривалість процедури	Не порожнє
price	вартість	Не порожнє
Продаж (sell)		
sell_id	ідентифікатор продажу	Первинний ключ
sell_name	назва засобу	Не порожнє

Продовження таблиці 3.1.1

volume	обсяг	
price	вартість	Не порожнє
Візит на електроепіляцію(visit_electro)		
visit_id	ідентифікатор візиту	Первинний ключ
electro_id	ідентифікатор електроепіляції	Зовнішній ключ для зв'язку з сутністю electro (electro _id)
Візит на лазерну епіляцію (visit_laser)		
visit_id	ідентифікатор візиту	Первинний ключ
laser_id	ідентифікатор лазерної епіляції	Зовнішній ключ для зв'язку з сутністю laser (laser _id)
Візит на масаж (visit_massage)		
visit_id	ідентифікатор візиту	Первинний ключ
massage_id	ідентифікатор масажу	Зовнішній ключ для зв'язку з сутністю massage (massage _id)
Візит на шугаринг (visit_shugaring)		
visit_id	ідентифікатор візиту	Первинний ключ
shugaring_id	ідентифікатор шугарінгу	Зовнішній ключ для зв'язку з сутністю shugaring (shugaring _id)

Розглянемо зв'язок між працівником та посадою. На одній посаді може бути кілька працівників, але в конкретного співробітника може бути лише одна посада. Це є безумовний зв'язок один до багатьох між працівником та посадою, тому для формалізації такого зв'язку ідентифікатор посади додано до працівника в якості зовнішнього ключа.

Розглянемо зв'язок між працівником та візитом. До одного співробітника може бути багато візитів, але кожен візит закріплений за певним співробітником. Це є безумовний зв'язок один до багатьох між працівником та візитом, тому для формалізації такого зв'язку ідентифікатор працівника додано до візита в якості зовнішнього ключа.

Розглянемо зв'язок між клієнтом та візитом. Кожен клієнт має можливість здійснювати безліч візитів, але кожен візит закріплений за певним клієнтом. Це є безумовний зв'язок один до багатьох між клієнтом та візитом, тому для формалізації такого зв'язку ідентифікатор клієнта додано до візита в якості зовнішнього ключа.

Розглянемо зв'язок між послугою і візитом. Кожному візиту відповідає одна послуга певної категорії, тому для формалізації зв'язку багато-до-багатьох для кожної категорії послуг було створено свої сутності, такі як (`visit_electro`, `visit_laser`, `visit_massage` та інші) в яких поміщені первинні ключі таблиць `visit` та певної послуги.

Розглянемо діаграму даних предметної області.(Рисунок 3.1.1)

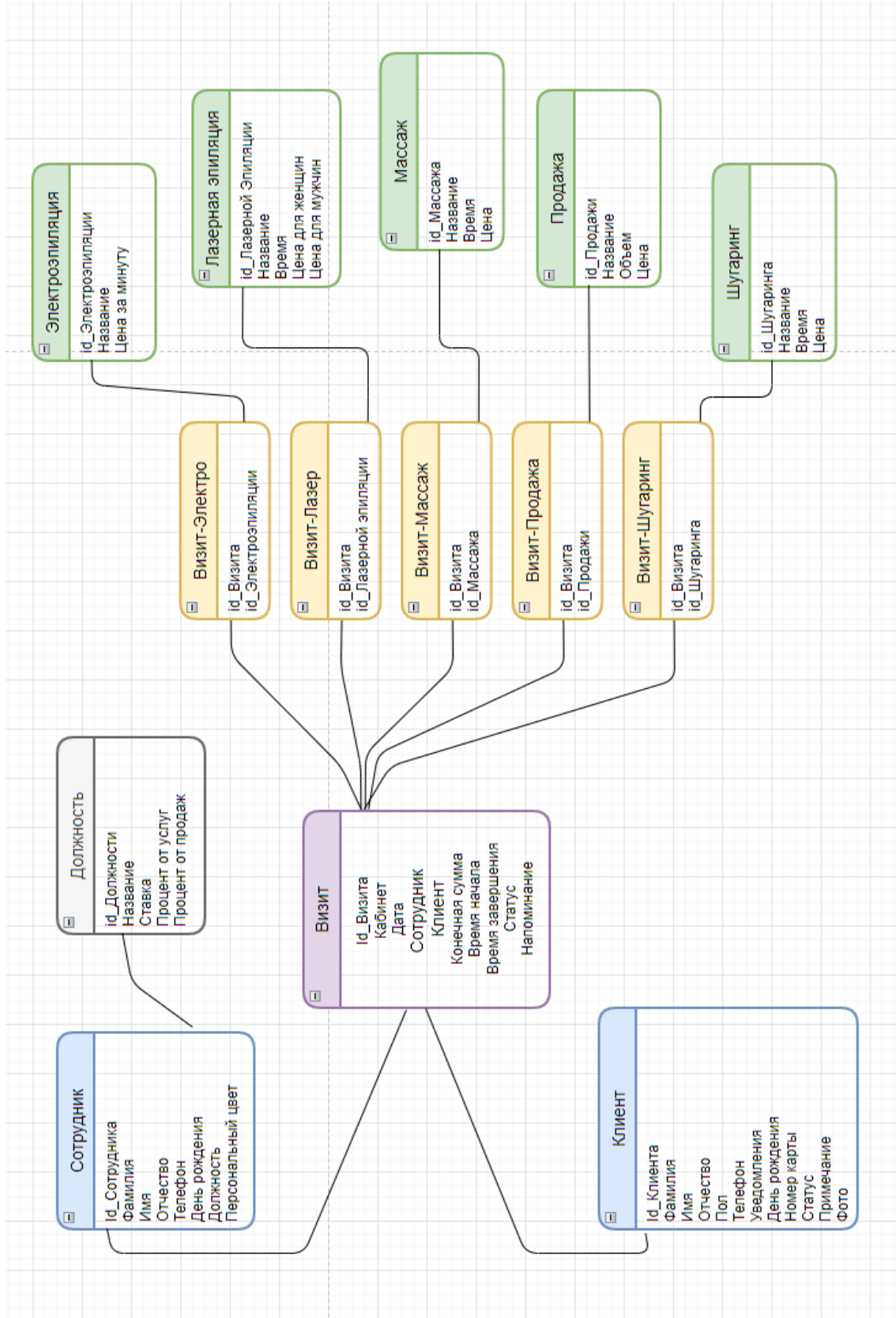


Рисунок 3.1.1 – Диаграмма ПрО салона краси

3.2 МОДЕЛЬ ДОДАТКУ

У Main реалізовані події які викликаються при певних діях користувача. У Model реалізовані основні функції для роботи програми. Для Model та Main реалізовані інтерфейси при передачі яких у презентер відбувається підписка функцій Model на події Main. Ще у моделі є об'єкт SqlData, цей клас потрібен для роботи з базою даних, завдяки йому, Model отримує дані з сервера. Є ряд допоміжних класів, за допомогою яких, дані передаються структуровано.

На рисунку 3.2.1 показані зв'язки компонентів головної форми з презентером, моделлю та допоміжними класами.

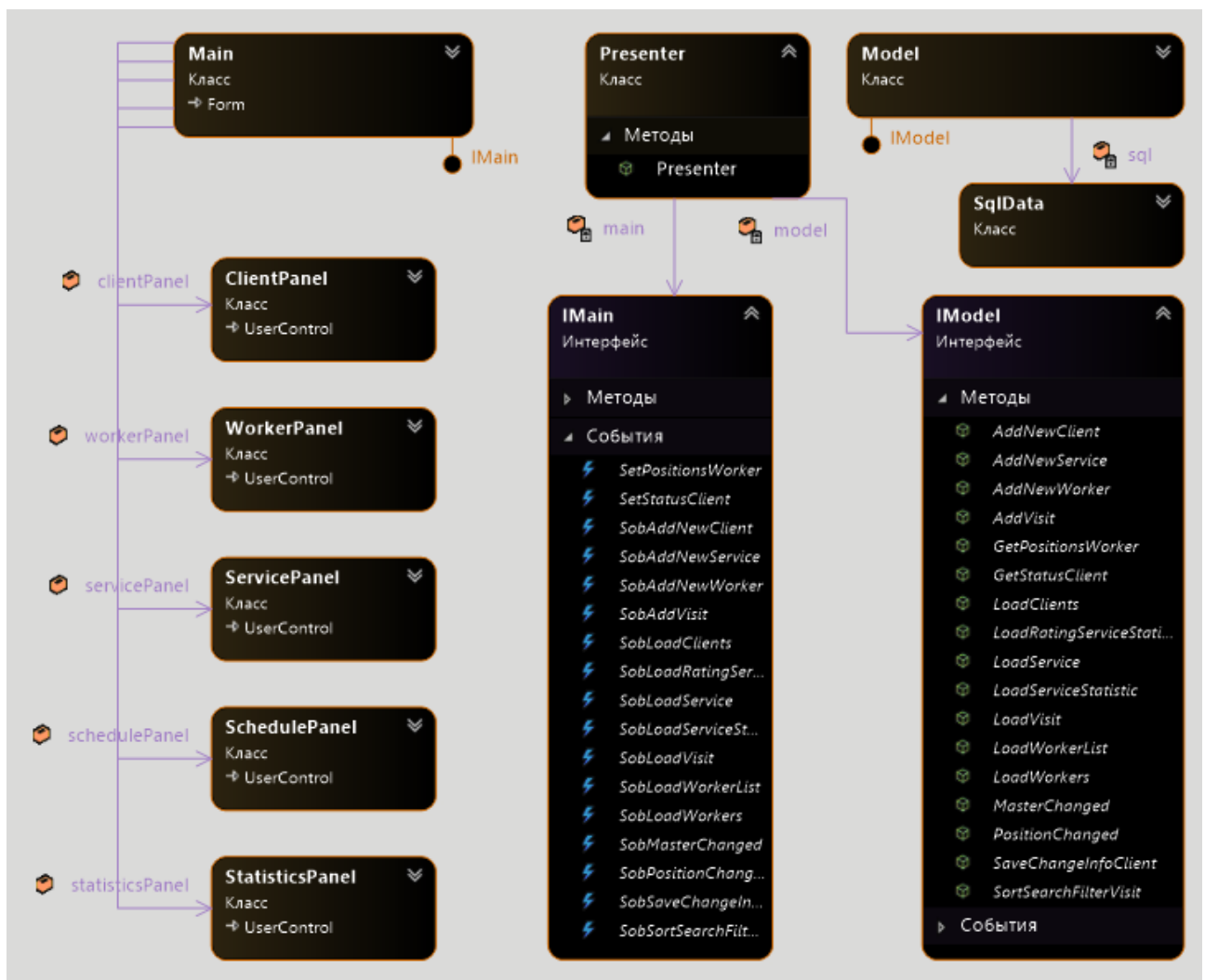


Рисунок 3.2.1 – Діаграма класів

Опис методів класу Model:

- 1) AddNewClient - Додавання нового клієнта до бази.
- 2) AddNewService - Додавання нової послуги до бази.
- 3) AddNewWorker - Додавання нового співробітника до бази.
- 4) AddVisit - Додавання візиту до розкладу.
- 5) GetPositionsWorker - Повертає список посад, використовується при додаванні нового співробітника.
- 6) GetStatusClient - Повертає список статусів клієнтів, використовується при додаванні нового клієнта.
- 7) LoadClients - Завантажує всіх клієнтів із бази.
- 8) LoadRatingServiceStatistic - Обчислює найпопулярніші послуги за певний період.
- 9) LoadService - Завантажує всі послуги з бази.
- 10) LoadServiceStatistic - Обчислює загальну кількість грошей зароблених за певний період поділяючи їх за категоріями (електроепіляція, шугарінг та ін.).
- 11) LoadVisit - Завантажує візити на певну дату до зазначеного кабінету.
- 12) LoadWorkerList - Повертає список імен та прізвищ співробітників, використовується при записі клієнта на процедуру.
- 13) LoadWorkers - Завантажує всіх співробітників з бази.
- 14) MasterChanged - Відбувається при зміні майстра, в момент додавання клієнта на процедуру, залежно від обраного майстра змінюються послуги, що надаються, наприклад майстер лазерної епіляції не може надавати послуги шугарингу.
- 15) PositionChanged - Відбувається при зміні посади, у момент додавання нового майстра, показується її характеристики.
- 16) SaveChangeInfoClient - Функція зміни даних у існуючого клієнта.
- 17) SortSearchFilterVisit - Відбувається при сортуванні та пошуку клієнта (на ім'я або номер).

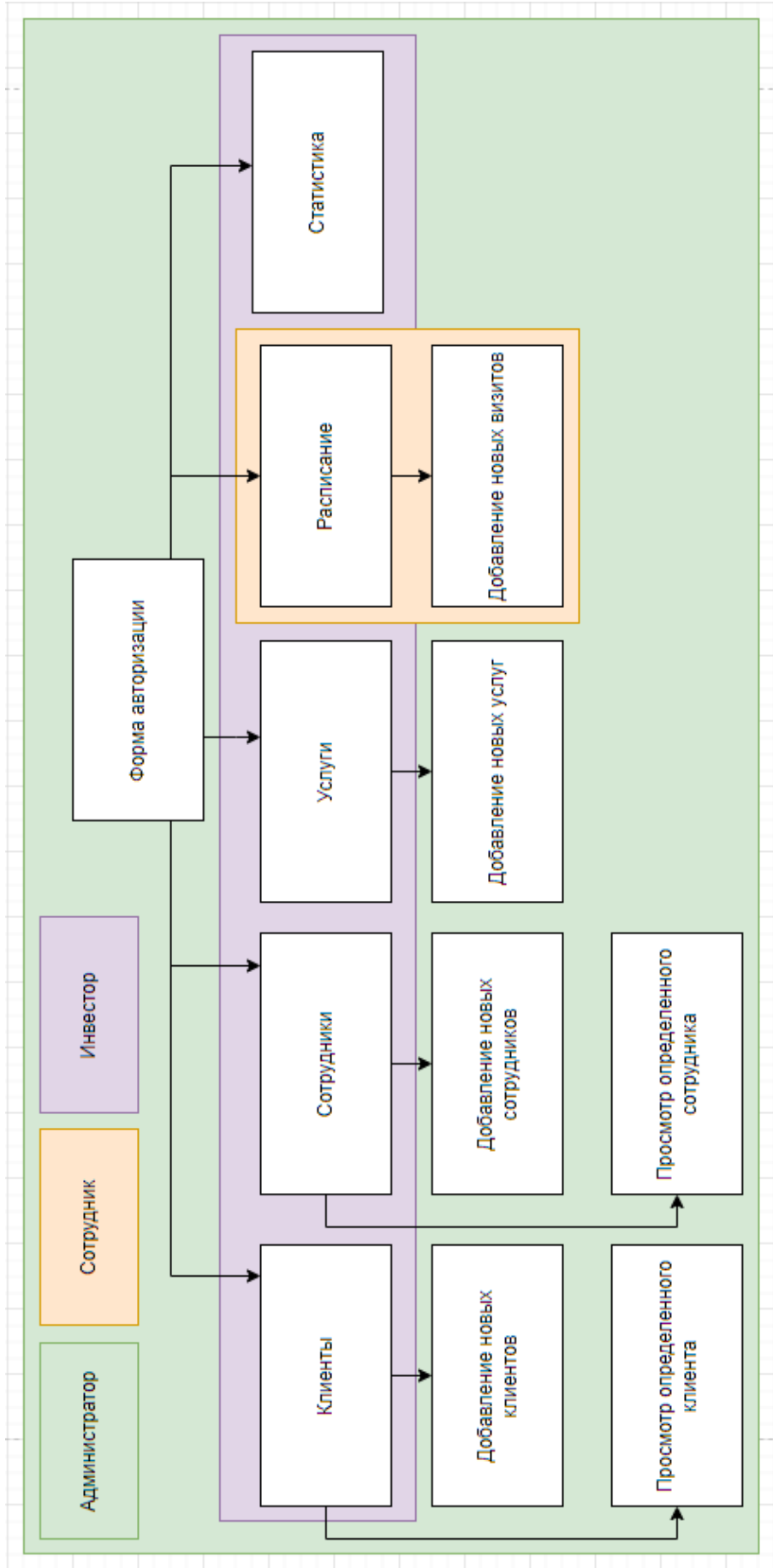


Рисунок 3.2.2 – Иерархия форм пользователей (Администратор, Инвестор, Сотрудник)

3.3 СТВОРЕННЯ БАЗИ ДАНИХ

Розглянемо створення таблиці `worker`.

Створення таблиці відбувається за допомогою команди `CREATE TABLE` далі йде назва таблиці. Кожному полю визначається певний тип даних СУБД та обмеження цілісності.

`PRIMARY KEY (worker_id)` означає, що поле `worker_id` – це первинний ключ цієї таблиці.

`CHECK (worker_money >= 0)` при визначенні поля `worker_money` вказує, що значення має бути не менше нуля.

`FOREIGN KEY (worker_positions)` означає, що поле `worker_positions` – це зовнішній ключ.

Після ключового слова `REFERENCES` вказується ім'я пов'язаної таблиці, а потім у дужках ім'я зв'язаного стовпця, на який вказуватиме зовнішній ключ.

Для створення таблиці `worker` використовується наступний SQL-запит:

```
CREATE TABLE worker
(
  worker_id int PRIMARY KEY,
  worker_name varchar(50) NOT NULL,
  worker_surname varchar(50) NOT NULL,
  worker_patronymic varchar(50),
  phone varchar(50) NOT NULL,
  birthday date,
  color int NOT NULL,
  worker_money numeric CHECK (worker_money >= 0),
  worker_positions int NOT NULL,
  FOREIGN KEY (worker_positions) REFERENCES
positions(positions_id)
);
```

Повний перелік запитів для створення таблиць бази даних представлено у Додатку Б.

3.4 ЗАПИТИ ДО БАЗИ ДАНИХ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНИХ ЗАВДАНЬ

Для вирішення задач A1, A3 та A4 використовуються запити типу *INSERT INTO таблиця (стовбець1, стовбець2) VALUES (значення1, значення2)*. В них потрібно вставити потрібні назви стовбців, таблиць.

Наприклад, для виконання A3 використовується така команда:

```
INSERT INTO worker (worker_name, worker_surname,
worker_patronymic, phone, birthday, color, worker_money,
worker_positions) VALUES (@worker_name, @worker_surname,
@worker_patronymic, @phone, @birthday, @color, @worker_money,
@worker_positions)
```

Для вирішення задачі A2 використовується така команда:

```
UPDATE client
SET client_name = @client_name, client_surname =
@client_surname, client_patronymic = @client_patronymic, gender
= @gender, phone = @phone, notice_bool = @notice_bool, birthday =
@birthday, num = @num, status = @status, note = @note,
link_photo = @link_photo
WHERE client_id = " + client.id
```

Для вирішення задачі A5 використовується така команда:

```
UPDATE worker "
SET worker_money = {worker_money}
WHERE worker_id = {worker_id};
```

При виклику команди UPDATE спрацьовує тригер `pay_out_trigger` який наведено у додатку В.

Для вирішення задач A6-A10 використовуються запити типу *INSERT INTO таблиця (стовбець1, стовбець2) VALUES (значення1, значення2)*. В них потрібно вставити потрібні назви стовбців, таблиць.

Наприклад, для виконання A6 використовується така команда:

```
INSERT INTO electro (electro_name, price_per_minute) VALUES
(@electro_name, @price_per_minute);
```

Для вирішення задач C1 та C2 спочатку візивається функція запитом:

```
SELECT collision('{date}', '{cab}', '{tStart}', '{tEnd}',
{workerId})
```

Тіло функції наведено у додатку В.

Якщо вона поверне значення типу: (True) використовуються запити типу *INSERT INTO таблиця (стовбець1, стовбець2) VALUES (значення1, значення2)*. В них потрібно вставити потрібні назви стовбців, таблиць.

Наприклад, для виконання C1 використовується така команда:

```
INSERT INTO visit (cabinet, visit_date, worker_id, client_id,
final_amount, time_start, time_end, visit_status, notification)
VALUES (@cabinet, @visit_date, @worker_id, @client_id,
@final_amount, @time_start, @time_end, @visit_status,
@notification) RETURNING visit_id;
```

Потім команда:

```
INSERT INTO visit_{service} (visit_id, {service}_id) VALUES
(@visit_id, @{service}_id)
```

Для вирішення задачі C3 використовуються такі команди:

```
DELETE FROM visit_{service} WHERE visit_id = {id};
DELETE FROM visit WHERE visit_id = {id};
```

Для вирішення задачі I1 використовується така команда:

```
SELECT positions.positions_name, sum(final_amount)
FROM visit
INNER JOIN worker ON visit.worker_id = worker.worker_id
INNER JOIN positions ON worker.worker_positions =
positions.positions_id
WHERE visit_status = 'completed' AND visit_date =
'{DateTime.Now}'
GROUP BY positions.positions_name
```

Для вирішення задачі I2 використовується така команда:

```
SELECT visit_id, final_amount, time_start, time_end,
visit_status, notification, client_surname, client_name,
worker_surname, worker_name, color
FROM visit
INNER JOIN client ON visit.client_id = client.client_id
INNER JOIN worker ON visit.worker_id = worker.worker_id
WHERE visit_date = '{date}' AND cabinet = '{cabinet}'
ORDER BY time_start
```

3.5 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРФЕЙСУ

Для прикладу роботи системи розглянемо завантаження клієнтів, співробітників та послуг.

```
private void InitialDataLoad()
{
    if (SobLoadClients != null)
        SobLoadClients("", "", clientPanel);

    if (SobLoadWorkers != null)
        SobLoadWorkers(workerPanel);

    if (SobLoadService != null)
        SobLoadService(servicePanel);
}
```

Функція завантаження клієнтів:

```
public void LoadClients(string searchClient, string
sortClient, ClientPanel clientPanel)
{
    this.searchClient = searchClient;
    this.sortClient = sortClient;

    List<MyClient> myClients =
sql.GetClientList(searchClient, sortClient);

    clientPanel.ListPanel.Controls.Clear();
    ClientList clientList = null;
    bool color = false;

    foreach (MyClient client in myClients)
    {
        clientList = new ClientList(client);
        clientList.Tag = client.id;
        clientList.OpenClient += new
EventHandler(OpenClient);

        if (color) { clientList.BackColor =
Color.FromArgb(35, 35, 37); }
        color = !color;

    }
    clientPanel.ListPanel.Controls.Add(clientList);
}
```

Функція завантаження працівників

```
public void LoadWorkers(WorkerPanel workerPanel)
{
    List<MyWorker> myWorker = sql.GetWorkerList();

    workerPanel.ListPanel.Controls.Clear();
    WorkerList workerList = null;

    foreach (MyWorker worker in myWorker)
    {
        workerList = new WorkerList(worker);
        workerList.Tag = worker.id;
        workerList.Worker_PayOut += new Action<int,
decimal>(Worker_PayOut);

workerPanel.ListPanel.Controls.Add(workerList);
    }
}
```

Функція завантаження послуг, поділена на категорії послуг

```
public void LoadService(ServicePanel servicePanel)
{
    LoadElectroService(servicePanel);

    LoadMassageService(servicePanel);

    LoadLaserService(servicePanel);

    LoadShugaringService(servicePanel);

    LoadSellService(servicePanel);
}
```

Завантаження послуг електроєпіляції

```
public void LoadElectroService(ServicePanel servicePanel)
{
    List<sElectro> ElectroService =
sql.GetElectroService();

    servicePanel.Elektro_panel.Controls.Clear();
    ServiceItem2 serviceItem = null;

    foreach (sElectro electro in ElectroService)
    {
        serviceItem = new ServiceItem2(electro.name,
electro.price_per_minute.ToString() + " грн");
        serviceItem.Tag = electro.id;

servicePanel.Elektro_panel.Controls.Add(serviceItem);}}
```

Завантаження послуг масажу

```

        public void LoadMassageService(ServicePanel
servicePanel)
        {
            List<sMassage> MassageService =
sql.GetMassageService();

servicePanel.MessageBonus_panel.Controls.Clear();
servicePanel.Message_panel.Controls.Clear();
ServiceItem3 serviceItem = null;

        foreach (sMassage message in MassageService)
        {
            serviceItem = new ServiceItem3(message.name,
message.time.ToString() + " мин", message.price.ToString() + "
грн");

            serviceItem.Tag = message.id;

            if (message.bonus)
            {
                serviceItem.BackColor =
Color.FromArgb(0, 139, 220);

servicePanel.MessageBonus_panel.Controls.Add(serviceItem);
            }
            else

servicePanel.Message_panel.Controls.Add(serviceItem);
        }
    }

```

Завантаження послуг лазерної епіляції

```

        public void LoadLaserService(ServicePanel servicePanel)
        {
            List<sLaser> LaserService =
sql.GetLaserService();

servicePanel.Laser_panel.Controls.Clear();
servicePanel.LaserBonus_panel.Controls.Clear();
ServiceItem4 serviceItem = null;

        foreach (sLaser laser in LaserService)
        {
            serviceItem = new ServiceItem4(laser.name,
laser.time.ToString() + " мин", laser.price_woman.ToString() + "
грн", laser.price_man.ToString() + " грн");
            serviceItem.Tag = laser.id;

            if (laser.bonus)
            {

```

```

        serviceItem.BackColor =
Color.FromArgb(204, 57, 123);

servicePanel.LaserBonus_panel.Controls.Add(serviceItem);
    }
    else

servicePanel.Laser_panel.Controls.Add(serviceItem);
    }
}

```

Завантаження послуг шугарінгу

```

    public void LoadShugaringService(ServicePanel
servicePanel)
    {
        List<sShugaring> ShugaringService =
sql.GetShugaringService();

        servicePanel.shugaring_panel.Controls.Clear();
        ServiceItem3 serviceItem = null;

        foreach (sShugaring shugaring in
ShugaringService)
        {
            serviceItem = new
ServiceItem3(shugaring.name, shugaring.time.ToString() + " мин",
shugaring.price.ToString() + " грн");
            serviceItem.Tag = shugaring.id;

servicePanel.shugaring_panel.Controls.Add(serviceItem);
        }
    }

```

Завантаження продажів

```

    public void LoadSellService(ServicePanel
servicePanel)
    {
        List<sSell> SellService = sql.GetSellService();

        servicePanel.sell_panel.Controls.Clear();
        ServiceItem3 serviceItem = null;

        foreach (sSell sell in SellService)
        {
            serviceItem = new ServiceItem3(sell.name,
sell.volume.ToString() + " мл", sell.price.ToString() + " грн");
            serviceItem.Tag = sell.id;

servicePanel.sell_panel.Controls.Add(serviceItem);
        }
    }

```

4 ПИТАННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ СИСТЕМИ. ІНСТРУКЦІЯ ДЛЯ КОРИСТУВАЧА

Для забезпечення безпеки на рівні БД використовується система ролей, яка передбачає розмежування дозволів на роботу із даними (табл. 4.1). В проектній системі може працювати три види користувачів: адміністратор, співробітник, та інвестор. Для розмежування дозволів використовується система CRUDE:

- 1) C – Create (створення нових даних);
- 2) R – Read (перегляд даних);
- 3) U – Update (оновлення даних);
- 4) D – Delete (видалення даних);

Таблиця 4.1 – Привілеї ролей на об'єкти бази даних

Об'єкти БД	Ролі		
	Адміністратор	Співробітник	Інвестор
Таблиці			
client	CRU	R	R
worker	CRUD	R	R
positions	CRUD	-	R
visit	CRUD	CRUD	R
electro	CRUD	R	R
laser	CRUD	R	R
massage	CRUD	R	R
shugaring	CRUD	R	R
sell	CRUD	R	R

При відкритті програми користувач бачить перед собою меню з пунктами: Клієнти, Співробітники, Послуги, Розклад, Статистика. (Рисунок 4.1)

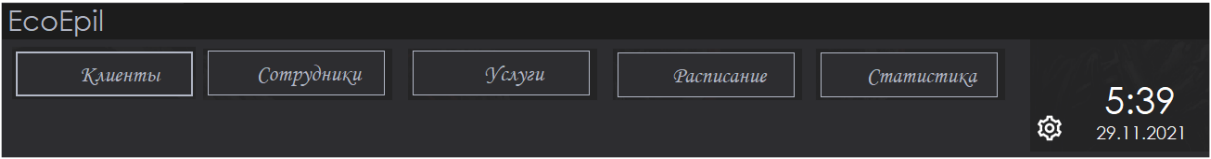


Рисунок 4.1 – Головне меню

При відкритті вкладки клієнти користувач бачить перед собою всіх клієнтів та невелику інформацію про них. (Рисунок 4.2)

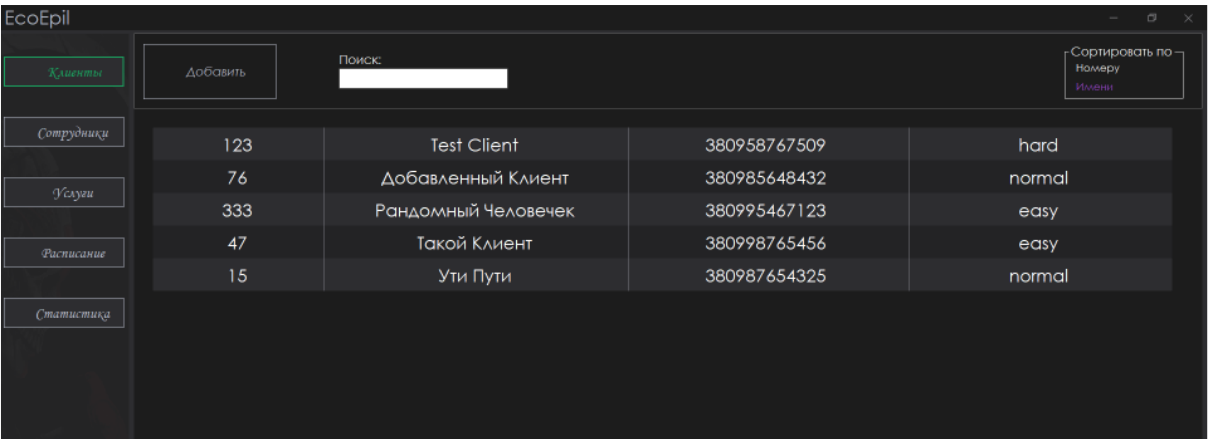


Рисунок 4.2 – Клієнти

Також можна зробити пошук та сортування. (Рисунок 4.3)

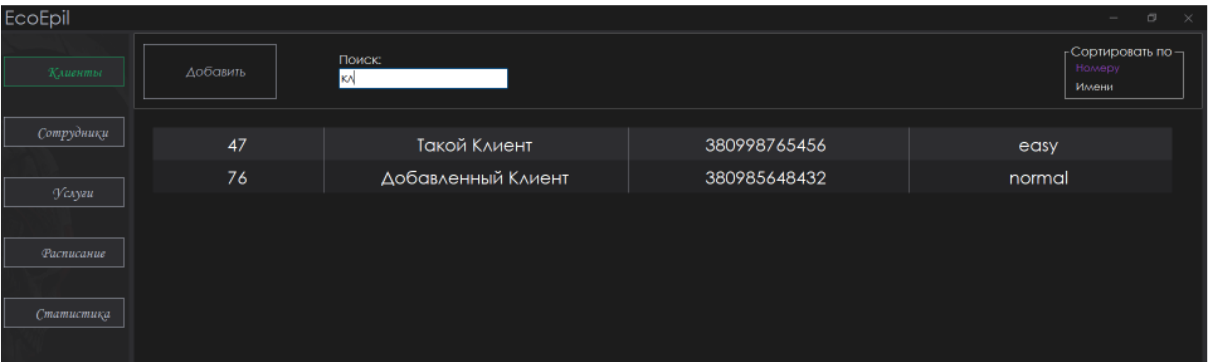


Рисунок 4.3 – Пошук та сортування

При відкритті певного клієнта показується вся інформація. (Рисунок 4.4)

ЕсоЕріл

Клиенты

Сотрудники

Услуги

Расписание

Статистика

Информация

Нет фото

Примечание:

Изменить данные

Фамилия: Такой

Имя: Клиент

Отчество:

Телефон: 380998765456

☐ Не отправлять уведомления

День рождения: 6 июня 1996 г.

Пол: Женщина

Статус: easy

Номер карты: 47

Рисунок 4.4 – Інформація про клієнта

Цю інформацію можна відредагувати. (Рисунок 4.5)

ЕсоЕріл

Клиенты

Сотрудники

Услуги

Расписание

Статистика

Редактирование

Нет фото

Примечание:

Сохранить

Отменить

Фамилия: Такой

Имя: Клиент

Отчество:

Телефон: 380998765456

☒ Не отправлять уведомления

День рождения: 6 июня 1996 г.

Пол: Женщина

Статус: easy

Номер карты: 47

Рисунок 4.5 – Редагування інформації

Також можна додати нового клієнта (Рисунок 4.6)

The screenshot shows the 'EcoEpiL' application window. On the left is a sidebar with buttons: 'Клієнти' (Clients), 'Сотрудники' (Employees), 'Услуги' (Services), 'Расписание' (Schedule), and 'Статистика' (Statistics). The main area is the 'Add Client' form. It features a 'Нет фото' (No photo) placeholder with a camera icon. The form fields are arranged in a grid:

- Row 1: Фамилия: (Family Name), Имя: (First Name), Отчество: (Patronymic)
- Row 2: Телефон: (Phone), Пол: (Gender), День рождения: (Date of Birth) - 24 ноября 2021 г.
- Row 3: Номер карты: (Card Number), Статус: (Status), Примечание: (Remarks)

At the bottom right are 'Отменить' (Cancel) and 'Сохранить' (Save) buttons. The bottom left shows a clock (8:22) and date (24.11.2021).

Рисунок 4.6 – Додавання клієнта

При додаванні клієнтів, без заповнення обов'язкових полів, користувач зустрине попередження (Рисунок 4.7)

This screenshot shows the same 'Add Client' form as Figure 4.6, but with validation messages. Red text 'Обязательно к заполнению' (Mandatory for completion) is displayed below the following fields:

- Фамилия: (Family Name)
- Имя: (First Name)
- Телефон: (Phone)
- Пол: (Gender)
- Номер карты: (Card Number)
- Статус: (Status)

The 'День рождения' (Date of Birth) field is pre-filled with '24 ноября 2021 г.' and has a calendar icon. The 'Не отправлять уведомления' (Do not send notifications) checkbox is also visible.

Рисунок 4.7 – Повідомлення про помилку

При відкритті вкладки співробітники користувач бачить перед собою всіх співробітників та кількість наявних у них грошей (Рисунок 4.8)

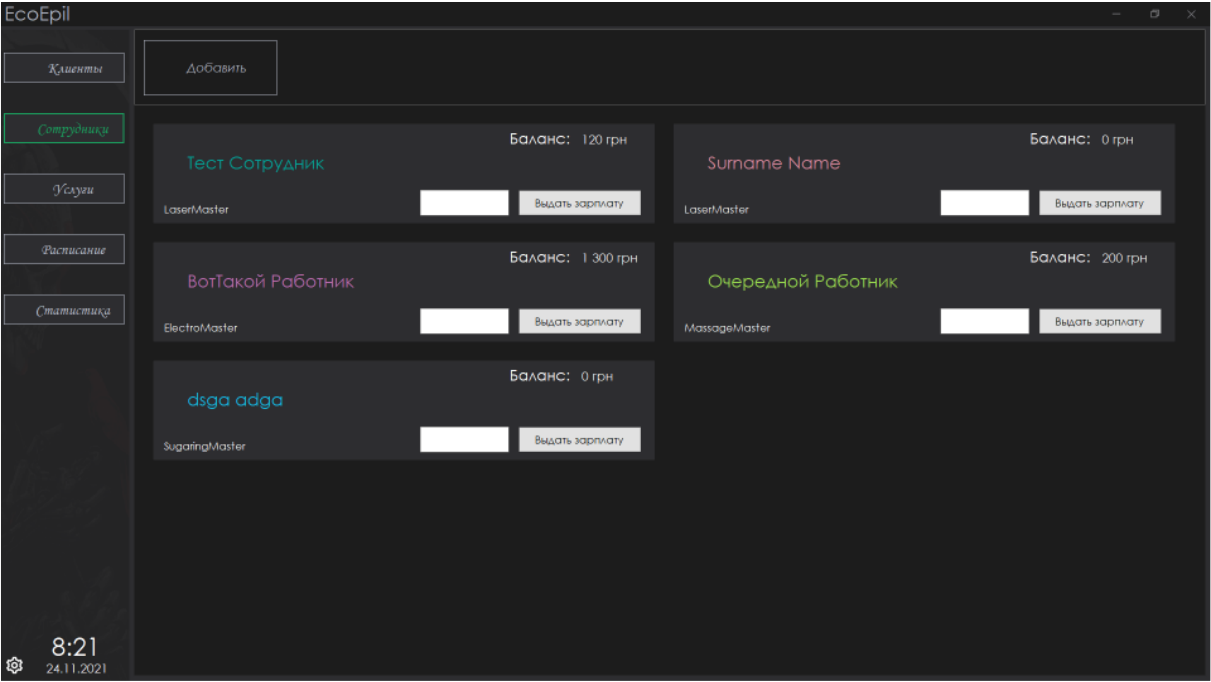


Рисунок 4.8 – Вкладка співробітники

При відкритті вкладки послуги (Рисунок 4.9)

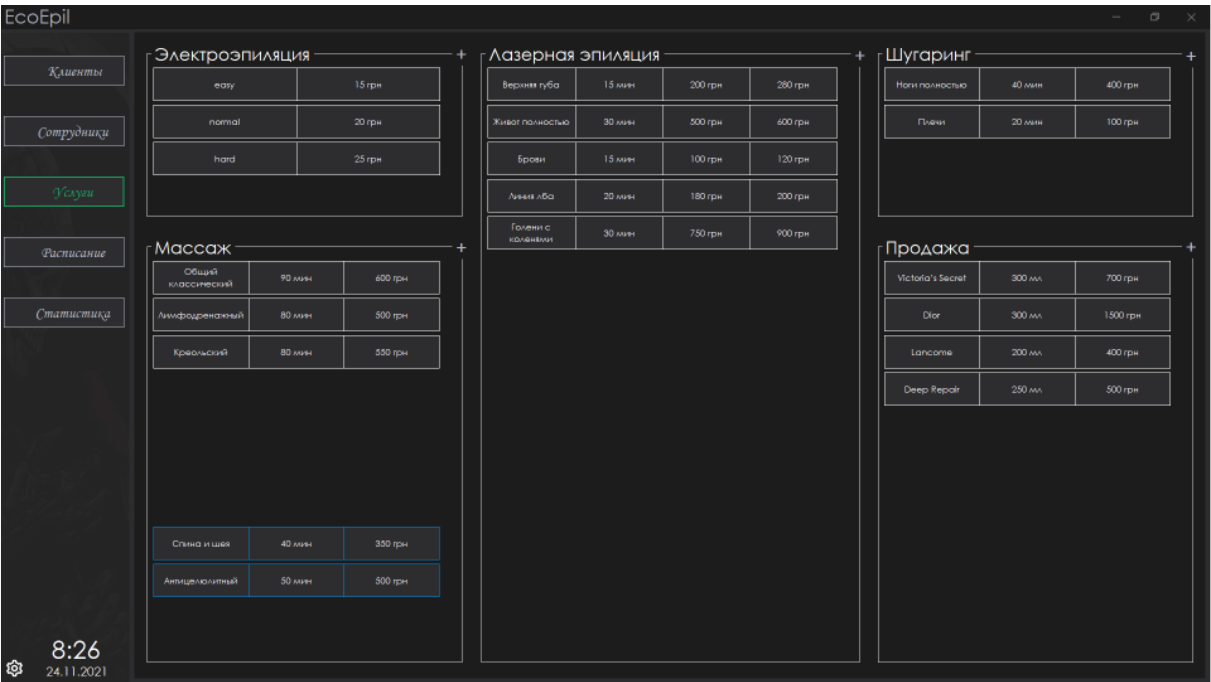


Рисунок 4.9 – Вкладка послуги

Через те, що у послуг різна структура, вікно додавання послуг у кожній категорії своє (Рисунок 4.10 – 4.14):

Рисунок 4.10 – Додавання послуги електроепіляції

Рисунок 4.11 – Додавання послуги масажу

Рисунок 4.12 – Додавання послуги лазерної епіляції

Рисунок 4.13 – Додавання послуги шугарінгу

Рисунок 4.14 – Додавання послуги продажу

При відкритті вкладки розклад, користувач бачить запис клієнтів у вибрані дні. (Рисунок 4.15)

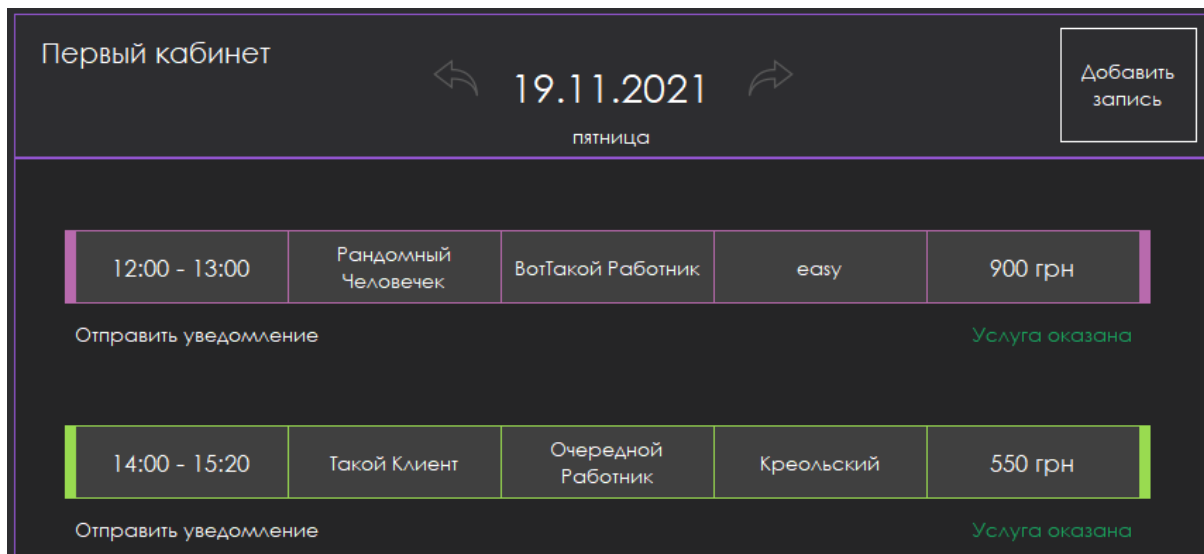


Рисунок 4.15 – Вкладка розклад

- 1) Клієнту можна надіслати повідомлення.
- 2) Запис можна видалити, якщо не було здійснено оплату.
- 3) Можна сплатити зазначений запис.

При відкритті вкладки статистика користувач може оцінити роботу салону за позиціями, що його цікавлять. (Рисунок 4.16)

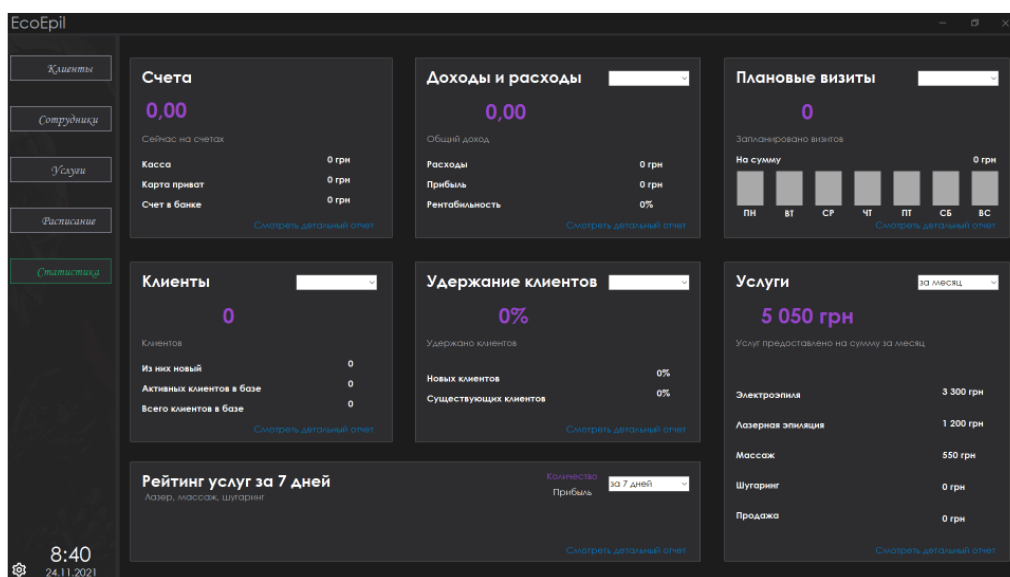


Рисунок 4.16 – Вкладка статистика

ВИСНОВКИ

В процесі виконання проекту проведено аналіз предметної області, що дозволило сформулювати потреби користувачів, задачі, які потрібно вирішити системою, список користувачів (адміністратор, співробітник, інвестор). Сформовані потреби даних та створена база даних на 14 таблиць для зберігання та маніпулювання інформацією.

Для розробки систему обрано дворівнева архітектура клієнт-сервера, шаблон проектування MVP, СУБД PostgreSQL, мову програмування C# та систему управління пакетами Npgsql.

Інтерфейс побудовано за потребами користувачів та забезпечує зручне вирішення задач Про.

 Пемухов Д.С.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Двухуровневые архитектура «клиент-сервер» [Электронный ресурс] - <https://studfiles.net/preview/5265892/page:3/>. – 05.06.2019
2. Малахов Є.В., Розновець О.І. Організація баз даних: методичні вказівки до курсового проектування з навчального курсу // Є.В. Малахов, О.І. Розновець / Одеса: Одес. нац.ун-т ім. І. І. Мечникова, 2021. – 64 с.
3. PostgreSQL — объектно-реляционная система управления базами данных [Электронный ресурс] – Режим доступа: <https://web-creator.ru/articles/postgresql>.
4. Малахов Є.В., Проектування баз даних та їх реалізація засобами стандартного SQL та PostgreSQL: Навч. посіб. для студ. вищих навч. закладів / Є.В. Малахов, О.А. Блажко, М.Г. Глава // Одеса: ВМВ, 2012. – 248 с.

ДОДАТОК А

Таблиця А.1 - Перелік задач користувачів інформаційної системи

Номер	Задача	Вхідні дані	Вихідні дані
Адміністратор			
A1	Створити клієнта	Прізвище, Ім'я, По батькові, Стать, Телефон, Повідомлення, День народження, Статус, Примітка, Фото	Новий клієнт
A2	Редагувати клієнта	Клієнт, що змінюється, Нові дані клієнта	Оновлена інформація про клієнта
A3	Додати співробітника	Прізвище, Ім'я, По-батькові, Телефон, Посада, День народження, Персональний колір	Новий співробітник
A4	Реєстрація посад співробітників	Назва, Ставка, Відсоток від послуг, Відсоток від продажу	Нова посада
A5	Видача зарплати співробітнику	Ідентифікаційний номер співробітника, Кількість грошей	Оновлена інформація про співробітника
A6	Додавання послуги Електроепіляції	Категорія, ціна за хвилину	Нова послуга електроепіляції
A7	Додавання послуги масажу	Назва, Час, Ціна, Бонус	Нова послуга масажу
A8	Додавання послуги Лазерної епіляції	Назва, Час, Ціна для жінок, Ціна для чоловіків, Бонус	Нова послуга лазерної епіляції
A9	Додавання послуги Шугарінгу	Назва, Час, Ціна	Нова послуга шугарінгу
A10	Додавання послуги Продажу	Назва, Об'єм, Ціна	Нова послуга продажу

Продовження таблиці А.1

Співробітник			
C1	Додати запис на електроепіляцію	Дата, Клієнт, Майстер, Час початку, Тривалість	Ідентифікатор запису
C2	Додати запис на (Лазерну епіляцію, Масаж, Шугарінг)	Дата, Клієнт, Майстер, Час початку, Процедура	Ідентифікатор запису
C3	Видалити запис	Ідентифікатор запису	Відсутні
Інвестор			
I1	Перегляд статистики	Відсутні	Статистика
I2	Перегляд розкладу	Відсутні	Розклад

ДОДАТОК Б

Запити на створення таблиць бази даних

```

CREATE TABLE client
(
    client_id integer NOT NULL,
    client_name varying(50) NOT NULL,
    client_surname varying(50) NOT NULL,
    client_patronymic varying(50),
    gender varying(50) NOT NULL,
    phone varying(50) NOT NULL,
    notice_bool boolean,
    birthday date,
    num integer NOT NULL,
    status varying(50) NOT NULL,
    note text,
    link_photo varying(100),
    CONSTRAINT client_pkey PRIMARY KEY (client_id)
);

CREATE TABLE worker
(
    worker_id int PRIMARY KEY,
    worker_name varchar(50) NOT NULL,
    worker_surname varchar(50) NOT NULL,
    worker_patronymic varchar(50),
    phone varchar(50) NOT NULL,
    birthday date,
    color int NOT NULL,
    worker_money numeric CHECK (worker_money >= 0),
    worker_positions int NOT NULL,
    FOREIGN KEY (worker_positions) REFERENCES
positions(positions_id)
);

CREATE TABLE positions
(
    positions_id integer NOT NULL,
    positions_name varying(50) NOT NULL,
    rate smallint NOT NULL,
    percentage_of_services smallint NOT NULL,
    percentage_of_sales smallint NOT NULL,
    CONSTRAINT positions_pkey PRIMARY KEY (positions_id)
);

```

```

CREATE TABLE visit
(
    visit_id integer NOT NULL,
    cabinet varying(50) NOT NULL,
    visit_date date NOT NULL,
    worker_id integer NOT NULL,
    client_id integer NOT NULL,
    final_amount numeric NOT NULL,
    time_start time without time zone NOT NULL,
    time_end time without time zone NOT NULL,
    visit_status varying(50) NOT NULL,
    notification boolean NOT NULL,
    PRIMARY KEY (visit_id),
    FOREIGN KEY (client_id) REFERENCES client (client_id),
    FOREIGN KEY (worker_id) REFERENCES worker (worker_id)
);

```

```

CREATE TABLE electro
(
    electro_id integer NOT,
    electro_name varying(50) NOT NULL,
    price_per_minute integer NOT NULL,
    CONSTRAINT electro_pkey PRIMARY KEY (electro_id)
);

```

```

CREATE TABLE laser
(
    laser_id integer NOT,
    laser_name varying(50) NOT NULL,
    laser_time integer NOT NULL,
    price_woman integer NOT NULL,
    price_man integer NOT NULL,
    bonus boolean NOT NULL,
    CONSTRAINT laser_pkey PRIMARY KEY (laser_id)
);

```

```

CREATE TABLE message
(
    message_id integer NOT,
    message_name varying(50) NOT NULL,
    message_time integer NOT NULL,
    price integer NOT NULL,
    bonus boolean NOT NULL,
    CONSTRAINT message_pkey PRIMARY KEY (message_id)
);

```

```

CREATE TABLE shugaring
(
    shugaring_id integer NOT,

```

```

        shugaring_name varying(50) NOT NULL,
        shugaring_time integer NOT NULL,
        price integer NOT NULL,
        CONSTRAINT shugaring_pkey PRIMARY KEY (shugaring_id)
    );

CREATE TABLE sell
(
    sell_id integer NOT,
    sell_name varying(50) NOT NULL,
    volume integer,
    price integer NOT NULL,
    CONSTRAINT sell_pkey PRIMARY KEY (sell_id)
);

CREATE TABLE visit_electro
(
    visit_id integer NOT NULL,
    electro_id integer NOT NULL,
    FOREIGN KEY (electro_id) REFERENCES electro
(electro_id),
    FOREIGN KEY (visit_id) REFERENCES visit (visit_id)
);

CREATE TABLE visit_laser
(
    visit_id integer NOT NULL,
    laser_id integer NOT NULL,
    FOREIGN KEY (laser_id) REFERENCES laser (laser_id),
    FOREIGN KEY (visit_id) REFERENCES visit (visit_id)
);

CREATE TABLE visit_message
(
    visit_id integer NOT NULL,
    message_id integer NOT NULL,
    FOREIGN KEY (message_id) REFERENCES message
(message_id),
    FOREIGN KEY (visit_id) REFERENCES visit (visit_id)
);

CREATE TABLE shugaring
(
    shugaring_id integer NOT,
    shugaring_name varying(50) NOT NULL,
    shugaring_time integer NOT NULL,
    price integer NOT NULL,
    CONSTRAINT shugaring_pkey PRIMARY KEY (shugaring_id)
);

```

ДОДАТОК В

Запити на створення тригерів та функцій

```

CREATE OR REPLACE FUNCTION collision(
visitdate date, cab character varying,
tstart time without time zone,
tend time without time zone, workerid integer)
    RETURNS boolean
    LANGUAGE 'sql'

COST 100
VOLATILE PARALLEL UNSAFE

AS $BODY$

SELECT EXISTS(select *
               from visit
               where (visit_date = visitDate and cabinet = cab
and tEnd > time_start and tEnd <= time_end)
                  or (visit_date = visitDate and cabinet = cab
and tStart >= time_start and tStart < time_end)
                  or (visit_date = visitDate and cabinet = cab
and tStart >= time_start and tEnd <= time_end)
                  or (visit_date = visitDate and cabinet = cab
and tStart <= time_start and tEnd >= time_end)

                  or (visit_date = visitDate and worker_id =
workerId and cabinet != cab and tEnd > time_start and tEnd <=
time_end)

                  or (visit_date = visitDate and worker_id =
workerId and cabinet != cab and tStart >= time_start and tStart
< time_end)

                  or (visit_date = visitDate and worker_id =
workerId and cabinet != cab and tStart >= time_start and tEnd <=
time_end)

                  or (visit_date = visitDate and worker_id =
workerId and cabinet != cab and tStart <= time_start and tEnd >=
time_end))

$BODY$;

CREATE OR REPLACE VIEW public.client_view
AS
SELECT client.client_id,
       client.client_surname,
       client.client_name,
       client.phone,
       client.num,
       client.status
FROM client;
CREATE OR REPLACE FUNCTION public.pay_out()
    RETURNS trigger

```

```
        LANGUAGE 'plpgsql'
        COST 100
        VOLATILE NOT LEAKPROOF
    AS $BODY$
    BEGIN
        INSERT INTO pay_history(worker_id, worker_money, datetime)
VALUES
        (NEW.worker_id, NEW.worker_money, current_timestamp);

    RETURN NEW;
    END;
$BODY$;

CREATE TRIGGER pay_out_trigger
AFTER UPDATE
ON worker
FOR EACH ROW
EXECUTE PROCEDURE pay_out();
```