

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
д.т.н., проф. С. І. Шматков
«__» _____ 2023 р

Пояснювальна записка

до кваліфікаційної роботи
бакалавра


на тему: «КОМП'ЮТЕРНА СИСТЕМА АВТОМАТИЗАЦІЇ
БІЗНЕС-ПРОЦЕСІВ ФАРМАЦЕВТИЧНОЇ КОМПАНІЇ»

Захищено на засіданні
Атестаційної комісії № __
протокол № __ від __.06.2023 р.
Оцінка _____ / _____
Голова Атестаційної комісії

(підпис)

СКОБ Ю. О.
(прізвище та ініціали)

Виконав:
студент 4 курсу, групи КІ– 41
Галузь знань: 12 – Інформаційні
технології
Спеціальність: 123 – «Комп'ютерна
інженерія»
ЧЕРЕДНИК Олександр Вікторович


(підпис)

Керівник:
старший викладач кафедри теоретичної
та прикладної системотехніки
МОРОЗ Ольга Юріївна


(підпис)

Рецензент:
Професор кафедри фундаментальної та
мовної підготовки Національного
фармацевтичного університету д. ф-мн.,
професор
ПОГОРЄЛОВ Станіслав Вікторович


(підпис)

Харків – 2023

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і трьох додатків. Загальний обсяг роботи складає 69 сторінок, із яких 49 сторінок основної частини з 42 рисунками, 21 найменуваннями списку використаних джерел та трьома додатками.

Метою кваліфікаційної роботи є розробка перспективної комп'ютерної моделі системи управління та автоматизації бізнес-процесів фармацевтичної компанії для українського ринку.

Об'єкт дослідження – процес моделювання системи автоматизації бізнес-процесів фармацевтичної компанії

Предмет дослідження – комп'ютерні моделі автоматизації бізнес-процесів фармацевтичної компанії і методи їх аналізу.

Задача, яка вирішується в кваліфікаційній роботі полягає в розробці комп'ютерної моделі для автоматизації бізнес-процесів фармацевтичної компанії для українського ринку.

Область застосування – розробка веб-додатку, що буде використовуватися для роздрібною торгівлі фармацевтичними представниками.

Ключові слова: комп'ютерна модель, фармацевтична компанія, бізнес-процеси, веб-додаток, розробка веб-додатку.

ABSTRACT

The explanatory note to the bachelor's thesis consists of an introduction, three sections, conclusions, a list of used sources and three appendices. The total volume of the work is 69 pages, of which 49 pages are the main part with 42 figures, 21 names of the list of used sources and three appendices.

The purpose of the qualification work is to develop a computer model of management and automation of business processes of a pharmaceutical company.

The object of the research is the process of developing a computer model of automation of business processes of a pharmaceutical company.

The subject of the research is the computer model of automation of business processes of a pharmaceutical company.

The task that is solved in the qualification work is the development of a computer model for automating the business processes of a pharmaceutical company for the Ukrainian market.

The field of application is the development of a web application that will be used for retail trade by pharmaceutical representatives.

Keywords: computer model, pharmaceutical company, business processes, web application, web application development.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
РОЗДІЛ 1.....	10
АНАЛІЗ СУЧАСНИХ МЕТОДІВ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ ФАРМАЦЕВТИЧНОЇ КОМПАНІЇ	10
1.1 Аналіз сучасних онлайн-інструментів медичних представників в сфері фармацевтичних послуг.....	10
1.2 Аналіз сучасних методологій розробки комп'ютерних систем.....	14
1.3 Аналіз автоматизування бізнес-процесів фармацевтичної компанії.....	19
1.4 Аналіз сучасних методів розробки веб-додатків.....	23
1.5. Мета, предмет, об'єкт та постановка задачі.....	26
Висновки за розділом 1.....	27
РОЗДІЛ 2.....	28
ПРОЕКТУВАННЯ ТА РОЗРОБКА.....	28
КОМП'ЮТЕРНОЇ МОДЕЛІ ВЕБ-ДОДАТКУ.....	28
2.1 Проектування архітектури веб-додатку.....	28
2.2 Рівень доступу до даних.....	31
2.3 Рівень інфраструктури.....	36
2.4 Рівень представлення.....	37
Висновки за розділом 2.....	42
РОЗДІЛ 3.....	43
РЕКОМЕНДАЦІЇ ПО ВИКОРИСТАННЮ КОМП'ЮТЕРНОЇ МОДЕЛІ.....	43
3.1 Сторінки, що доступні усім користувачам.....	43
3.2 Сторінки, що доступні адміністратору та менеджеру.....	47
3.3 Сторінки, що доступні адміністратору.....	48
Висновки за розділом 3.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТКИ.....	57
Додаток А.....	57

Додаток Б.....	59
Додаток В.....	59

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

- SaaS – Software as a service – це модель надання ліцензії на програмне забезпечення за передплатою.
- СУБД – Система управління базами даних.
- IT – Інформаційні технології.
- HTML – Hyper Text Markup Language (англ.) – мова розмітки, яка визначає структуру та надає вигляд веб-сторінок.
- CSS – Cascading Style Sheets (англ.) – це мова стилізації, використовувана для опису зовнішнього вигляду веб-сторінок, які написані з використанням HTML або інших мов розмітки.
- JS – JavaScript (англ.) – це високорівнева, інтерпретована мова програмування, яка використовується для розробки динамічних інтерактивних веб-сторінок.
- .Net – це платформа розробки програмного забезпечення, розроблена компанією Microsoft. Вона надає середовище для виконання програм на різних платформах.
- ORM – Object-Relational Mapping (англ.) – технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних».
- EF Core – Entity Framework Core (англ.) – об'єктно-орієнтована, легковажна та розширювана технологія від компанії Microsoft для доступу до даних.
- SQL – Structured Query Language (англ.) – це стандартна мова запитів, яка використовується для взаємодії з базами даних. Вона дозволяє зберігати, керувати та отримувати доступ до даних, які зберігаються в структурованому форматі у базі даних.
- DOM – Document Object Model (англ.) – це програмний інтерфейс, який представляє структуру та контент веб-сторінки як дерево об'єктів. DOM дозволяє програмам та скриптам отримувати доступ до елементів сторінки, змінювати їх вміст, стилі, атрибути та взаємодіяти з ними.

ВСТУП

Українські фармкомпанії відновлюють інвестиції у виробництво та власні наукоємні розробки. Мова про капіталовкладення на сотні мільйонів гривень в майбутнє надважливої галузі як воєнних, так і мирних часів. Так проявляється впевненість вітчизняних фармацевтів у неминучій перемозі ЗСУ над хворим російським режимом, який можна вилікувати лише силою зброї. [1]

Чому українські фармкомпанії інвестують у свій розвиток навіть у найважчий для країни час – під час війни? Як зауважує Євгенія Піддубна, директорка служби корпоративних комунікацій «Фармак»: «Сьогодні, коли роль сильної національної фармацевтичної галузі доведена викликами пандемії COVID-19 та повномасштабною війною в Україні, варто сфокусуватися на подальшому розвитку її потенціалу. Це і стимулювання інвестицій у дослідження, і прискорення виходу нових генеричних препаратів в Україні через вдосконалення патентного законодавства, і освоєння нових технологій – власних та отриманих за рахунок міжнародного співробітництва». [1]

Актуальність роботи. Після початку повномасштабного російського вторгнення, що змінило все у роботі бізнесу, підприємці потребують підтримки. Від початку війни, через великі масштаби знищення української медичної інфраструктури та порушення логістичних зв'язків, медичні установи стикнулися з нестачею великої кількості лікарських засобів та обладнання. За даними опитування ВООЗ, кожен п'ятий хворий (22%) не зміг придбати необхідні йому ліки, а на тимчасово окупованих територіях і територіях з активними бойовими – кожен третій. Схожа ситуація і з внутрішньо переміщеними особами. [2]

Несподіваною проблемою для українських фармацевтичних компаній стала масована гуманітарна допомога, яка надходить в Україну з інших країн.

Лікувальні установи часто були перевантажені гуманітарними препаратами, а українські фармпідприємства – не змогли постачати свою продукцію у певних сегментах ринку. Щоб розвиватись, навіть в умовах війни, фармацевтичні компанії роблять ставку на прямий продаж препаратів з найбільшим попитом, розробку нових лікарських засобів і вихід на нові ринки.

У цій сфері застосування інформаційних технологій (ІТ) в роботі медичних представників дозволяє фахівцям фармацевтичних компаній знаходити вихід зі складних ситуацій, зокрема працювати з базами даних фармації, отримувати та оперувати інформацією про фармацевтичні засоби задля забезпечення підприємницьких цілей фармацевтичної індустрії та потреб, інтересів і запитів споживчого ринку фармацевтичних засобів.

Використання роботи медичних представників є найбільш ефективним засобом комунікативної політики виробника, особливо для рецептурних ліків. Як правило, медичні представники працюють з окремими препаратами або їх групами, за виділеною територією та зі споживачами (оптовими фірмами, аптеками, лікувально-профілактичними закладами, категоріями хворих). Контролюється діяльність медичного представника регіональними менеджерами та продукт-менеджерами. Саме вони планують діяльність медичного представника, здійснюють та беруть участь у процесі навчання, в подвійних візитах; аналізують щотижневі звіти. Показниками контролю роботи медичного представника є такі: обсяг та динаміка продажу продукції; контроль пересування по території, планування роботи; контроль розподілу часу між важливими та другорядними клієнтами; кількість візитів протягом місяця до клієнтів відповідної категорії; контроль втрат рекламних матеріалів, використання корпоративних фінансів, техніки, обладнання; термін надання інформації на запит керівника, розвиток навичок медичного представника.

[3]

Метою кваліфікаційної роботи є розробка комп'ютерної моделі управління та автоматизації бізнес-процесів фармацевтичної компанії.

Об'єкт дослідження – процес моделювання системи автоматизації бізнес-процесів фармацевтичної компанії.

Предмет дослідження – комп'ютерні моделі автоматизації бізнес-процесів фармацевтичної компанії і методи їх аналізу.

Задача, яка вирішується в кваліфікаційній роботі полягає в розробці перспективної комп'ютерної системи автоматизації бізнес-процесів фармацевтичної компанії, яка допоможе підняти роботу медпредставників на більш високий якісний рівень.

Завдання дослідження

1. Виконати аналіз предметної області задля вивчення ключових моментів у фармацевтичній компанії.

1. Постановка задачі моделювання комп'ютерної системи автоматизації бізнес-процесів фармацевтичної компанії.

2. Розробити модель комп'ютерної системи автоматизації бізнес-процесів фармацевтичної компанії.

3. Розробити рекомендації по впровадженню технології, що використовуються в роботі.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНИХ МЕТОДІВ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ ФАРМАЦЕВТИЧНОЇ КОМПАНІЇ

1.1 Аналіз сучасних онлайн-інструментів медичних представників в сфері фармацевтичних послуг

Фармкомпанії мають багато бізнес-процесів: продажі та фінансові операції, аналіз даних, менеджмент зустрічей фарм-представників з клієнтами, управління персоналом, створення бази даних клієнтів, така складна структура процесів потребує організації. Варто розуміти, що основна мета роботи медпредставника полягає в тому, щоб переконати лікаря або провізора використовувати препарати компанії в своїй практиці і, як наслідок, збільшити обсяги продажу. Спілкування з лікарем, в свою чергу, має проходити на високому професійному рівні з використанням різних технік продажу. Від медичного представника очікується високий ступінь залученості в процес.

Робота з просування препаратів будується перш за все на фаховій зацікавленості лікаря в терапії своїх пацієнтів, і завдання медичного представника – наочно та переконливо інформувати фахівця, що препарати компанії дійсно успішно справляються з цим завданням. [4]

Медичний (фармацевтичний) представник – посада представника фармацевтичної компанії, пов'язана з просуванням на ринок лікарських препаратів. Сьогодні розрізняють такі посади, як медичний представник з просування рецептурних або безрецептурних лікарських препаратів, а також, медичний представник з функціями торгового представника.

До функціональних обов'язків фармацевтичного представника входять проведення інформаційної роботи серед лікарів, провізорів, та ін.; індивідуальні візити, презентації, забезпечення мерчандайзингу в аптеці, контроль продажу лікарських препаратів в аптеках та призначень лікарів. [4]

Завданнями медичного представника є інформування про нові лікарські препарати компанії, нагадування про вже відомі препарати; обговорення призначення препарату в схемах лікування; позиціонування лікарських препаратів, фіксування у свідомості фахівців переваг лікарських препаратів, стимулювання лікарів виписувати рецепти та провізорів – рекомендувати лікарські препарати покупцям; аналіз діяльності конкурентів, створення та/або підтримка іміджу компанії. Сьогодення вимагає компанії йти в ногу з часом, тому сьогодні медичні представники іноді працюють в абсолютно новому форматі – відео-візитів.

Диджиталізація галузі фармації розширює можливості для клінік, аптек, косметичних і медичних центрів. Легка взаємодія з клієнтами, оптимізація роботи представників, забезпечення контролю та покращення обслуговування – це лише невелика частина того, що дає впровадження сучасних методів розробки комп'ютерних систем автоматизації для фармацевтичних компаній.

ZennX Retailer, продукт USM Group.

ZENNX – платформа SaaS, яка надає інноваційне та автоматизоване рішення для обробки громіздких фармацевтичних замовлень від ініціювання до виконання. [5] Постачальники SaaS розміщують сервіси та програми, які доступні клієнтам на вимогу. [6]

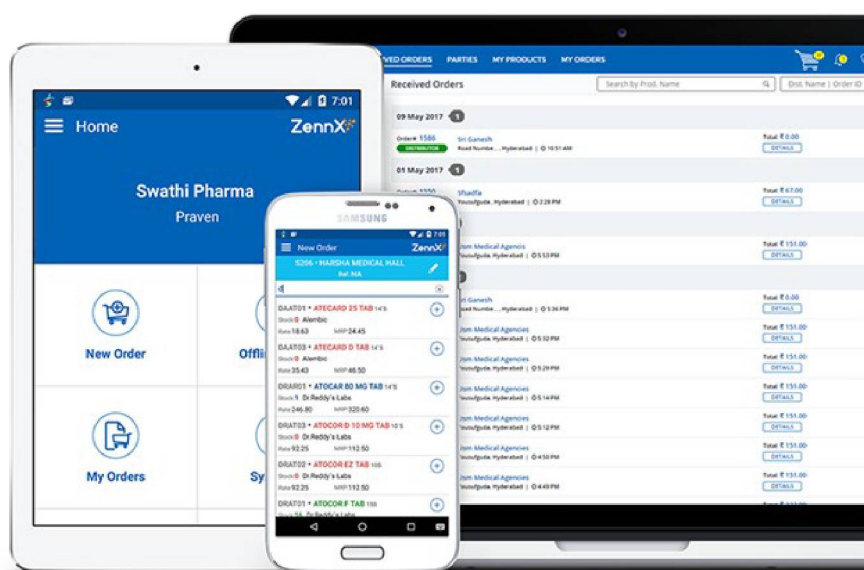


Рисунок 1.1 – Додаток і веб-додаток ZennX-Retailer

– Для роздрібних торговців: для знаходження товару, що немає в наявності, ви можете просто знайти їх у ZENNX і розмістити замовлення потрібному дистриб'ютору. ZENNX покаже актуальну ринкову позицію від ваших дистриб'юторів.

– Для дистриб'юторів: як тільки ви отримаєте замовлення, ZENNX надішле його до вашого платіжного програмного забезпечення для створення рахунків-фактур без ручного втручання.

– Для торгових представників/керівників: контроль над замовленнями та заборгованістю вашого продавця. Актуальна інформація у мобільному додатку.

Переваги SaaS

Оскільки створенням SaaS-сервісу і його підтримкою займається виключно провайдер, клієнт позбавляється необхідності його обслуговувати.

Переваг у SaaS моделі багато, проте є і ряд недоліків. Для деяких компаній вони критичні, тому доводиться відмовлятися від можливостей технологій SaaS.

Передача даних стороннім особам. По суті, сторонній провайдер отримуватиме конфіденційні дані вашої компанії. У більшості випадків це безпечно, але все ж таки ризик є.

Швидкодія системи залежить від швидкості інтернету. При нестабільній його роботі або повній відсутності робота сервісів припиняється.

Альтернативи SaaS

Хмарні платформи. Щоб не надавати конфіденційні дані компанії стороннім провайдерам, можна орендувати потужність комп'ютерів, так звані хмарні сховища. І там зберігати програмне забезпечення та всі дані.

Хостинг програм. Для кожного клієнта виконується встановлення окремої копії програми. Але обслуговування обходиться дорожче, а процес адміністрування складніший.

Diagnosia. Додаток Diagnosia є супутником як лікаря, фармацевта, медичного і фармацевтичного представника, який постійно розширюється. [7] Є можливість знаходити інформацію про спеціалістів, шукайте взаємодію ліків та консультуватися з медичними експертами через експертний чат. Але поки що він у тестовій версії.

MedHelp_UA_bot - зручний чат-бот лікарів та фармацевтів у Viber і Telegram: надання медичної допомоги, потреба в ліках та пошук транспорту за станом здоров'я. [8]

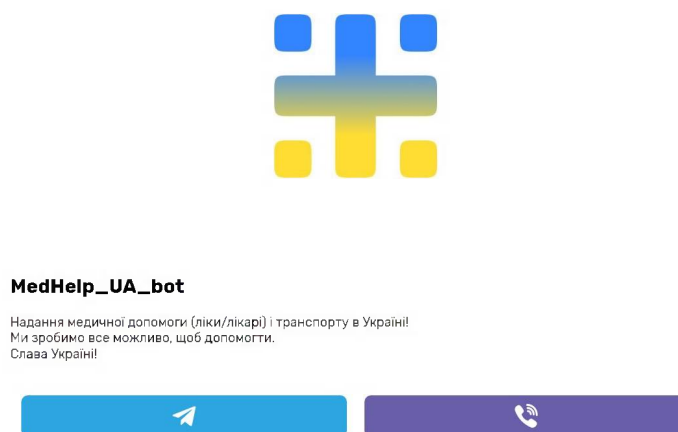


Рисунок 1.2 – MedHelp_UA_bot.

Спираючись на аналіз найбільш популярних додатків можна виділити основні переваги автоматизації процесу зберігання, оновлювання та обміну інформації для фармпредставників:

- 1) Централізований доступ.
- 2) Покращена координація фармацевтичного представника.
- 3) Збільшена безпека та захист даних.
- 4) Зручність та мобільність.
- 5) Покращення ефективності та економії часу.

Ці переваги демонструють значний потенціал застосування веб-додатків та мобільних застосунків у фіксації та обміну медичними даними, що

покращує якість фармацевтичної допомоги та взаємодію між медпредставником та клієнтом.

Задача кваліфікаційної роботи створити сучасну комп'ютерну систему автоматизації бізнес-процесів фармацевтичної компанії, яка допоможе підняти роботу медпредставників на більш високий якісний рівень. Рішення складається з двох частин: мобільний додаток, який встановлюється на смартфони або планшети медичних представників та основна база, з якої в режимі реального часу можна обмінюватися інформацією з офісною системою.

1.2 Аналіз сучасних методологій розробки комп'ютерних систем.

Методологія розробки комп'ютерних систем (програмного забезпечення) – це сукупність методів, що використовують на певних етапах життєвого циклу системи або програмного продукту. Методи в своїй основі мають спільний підхід, концепцію.

Сьогодні існує безліч методологій розробки комп'ютерних систем (програмного продукту), але можна виділити усього декілька, так як вони мають найбільший попит при розробці і відточені на практиці, вони суттєво допомагають організувати процес розробки командою, аналізувати та оцінювати складність задач, їх пріоритезації, сприяють більш тісній комунікації між членами команди та зацікавленими сторонами.

Waterfall model (водопад або каскадна модель) – Має п'ять стадій: затвердження вимог, проектування, кодування, тестування та підтримка.



Рисунок 1.3 – Водоспадна модель розробки.

Під час розробки потрібно чітко слідувати кожній стадії і не повертатися до попередньої поки не буде пройдений увесь цикл, при такому підході можна досягти високих темпів реалізації проекту, але й підходить така модель далеко не для всіх випадків. Як правило, використовують для малих, рідше, для середніх проектів. Також усі вимоги повинні бути чітко поставлені, а бізнес логіка досить простою та очевидною. [9]

V – model (validation and verification) – ця модель має багато спільного з «водопадом», так як також повинна дотримуватись послідовності етапів, але за одним виключенням – тестування. Тут воно найважливіше та проводиться водночас з розробкою, тому і несуттєві коректування дизайну системи також можливі за підсумком тестування функціоналу. Для використання моделі також мають бути чітко поставлені вимоги, як і в попередньої. [9]



Рисунок 1.4 – Модель валідації та перевірки.

Incremental model (інкрементна модель) також відома як «мульти-водоспад». При використанні цієї моделі увесь процес розробки ділиться на декілька циклів замість одного. Головна мета це розробити перший працюючий базовий модуль і поверх нього вже нарощувати функціонал з кожним наступним циклом поки не буде закінчений останній модуль. Ця методологія є логічним розвитком моделі «водопаду» та використовується якщо планується ранній вихід продукту на ринок. Також ця модель має різновид **RAD – model (швидка розробка додатку)**, де компоненти – інкременти розробляються декількома командами паралельно, а не послідовно. [9]



Рисунок 1.5 – Інкрементна модель.

Iterative model (ітеративна модель) – не потребує чітких вимог до логіки та способів реалізації. Спочатку потрібно створити базову робочу версію проекту, але з покриттям майже усього функціонала, який не повинен бути чітко спроектований та реалізований. Кожна версія повинна бути робоча, а при кожній так званій ітерації можна допрацьовувати дизайн та поглиблювати функціонал. [9] [10]



Рисунок 1.6 – Ітеративна модель.

Agile model (гнучка модель) – Модель з нечіткими вимогами до кінцевого продукту. Процес розробки відбувається за допомогою нетривалих циклів, як правило, від двох до чотирьох тижнів, які називаються «Sprint». Також учасники команди кожен день зустрічаються та обговорюють виконану роботу за минулий день «Scrum», проблеми та свої оцінки щодо часу, що залишився на реалізацію задачі. На початку кожного циклу команда оцінює перелік задач на цикл, планує та розподіляє їх. У кінці циклу, плануються нові задачі, або переносяться невиконані на наступний, також демонструють готовий функціонал клієнту, звіряють уявлення заказчика про кінцевий продукт. [11] Ця модель надає найбільшій гнучкості серед усіх, можливі правки вимог, дизайну в любий час. Але й прогнозування та оцінка завершення роботи над розробкою продукту дуже ускладнюється тому підходить для довготривалих проєктів націлених на постійну адаптацію до тенденцій ринку. [9]



Рисунок 1.7 – Модель гнучкої розробки.

Spiral model (спіральна модель) – модель дуже схожа на інкрементну, але підходить лише для великих та дорогих проєктів, в рамках розробки можуть вдатися до наукових досліджень. Методологія передбачає великий

аналіз та оцінку, наслідки невдачі важать більше ніж сам кінцевий продукт. кожен цикл моделі проходить через чотири етапи, а саме: планування, аналіз ризиків, розробка та оцінка результатів. [9] [12]

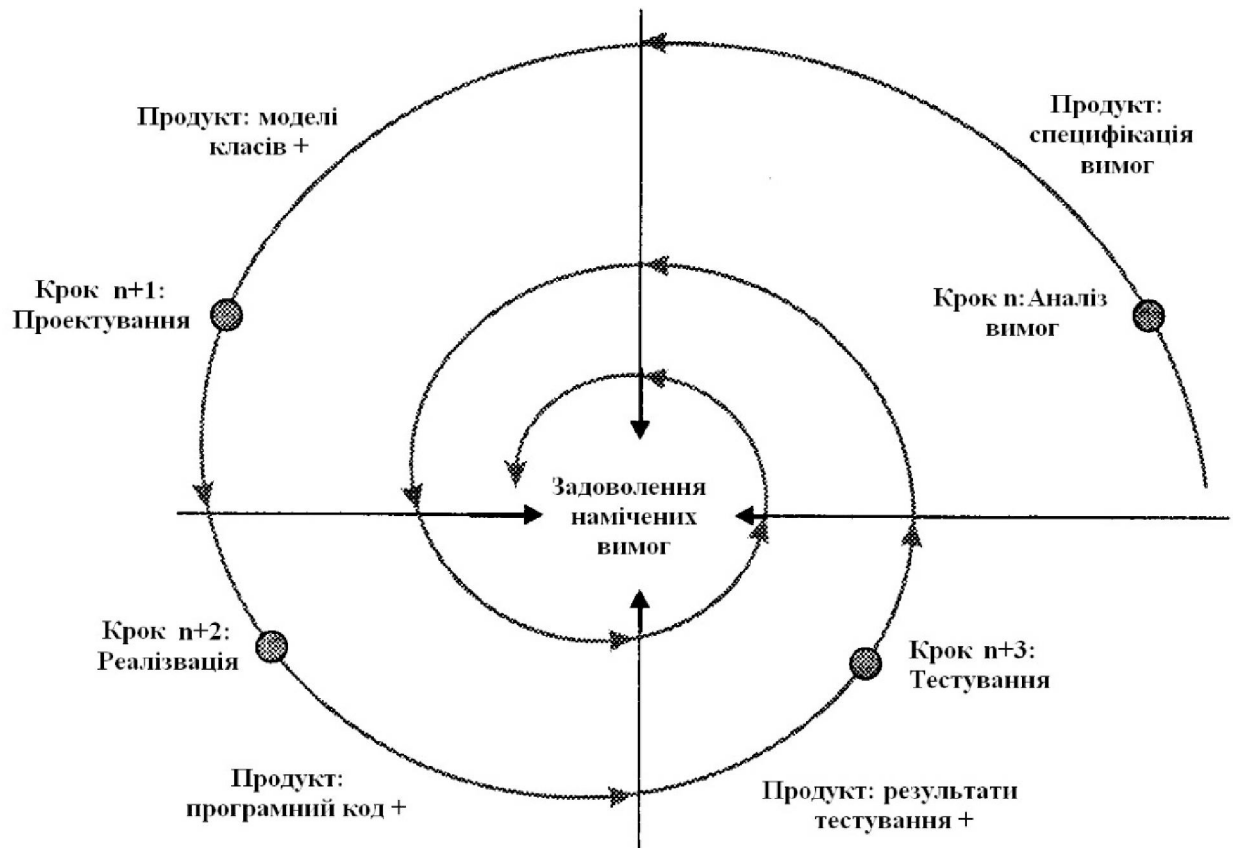


Рисунок 1.8 – Спіральна модель розробки.

На практиці методології розробки систем можуть перетинатись в деяких підходах та методах, деякі витікають із інших, є логічним розвитком попиту. Чітко визначити моделі для проєктів досить складно так як потребують уваги усіх факторів, починаючи від розміру самого проєкту та його вимог щодо функціоналу до бюджету та часу заказчика.

1.3 Аналіз автоматизування бізнес-процесів фармацевтичної компанії.

Сьогодні від медичних компаній вимагається збирати та використовувати більшу кількість даних, ніж будь-коли раніше. Крім того, ці

дані охоплюють різні аспекти сфери бізнесу, зокрема фінанси, аналіз, продажі, бази даних і клінічні дослідження. Отже, виникає необхідність інтеграції комп'ютерної моделі як інструмент автоматизації бізнес-процесів.

Крім того, у сучасному середовищі, орієнтованому на ефективність, неефективні робочі процеси можуть суттєво вплинути на продуктивність, мотивацію та загальну якість роботи працівників. Навіть, здавалося б, рутинні завдання часто стикаються з перешкодами для ефективності, як-от необхідність запитувати інформацію, отримати доступ до кількох програм або очікувати відповіді від інших відділів. Впровадження глобальної комп'ютерної моделі в компанії спрямоване на вирішення всіх цих проблем.

Автоматизований процес – це процес, який більше не вимагає втручання людини і може виконувати завдання, які раніше виконувалися вручну.

Компанії, які успішно впровадили автоматизацію у свою діяльність, спостерігають підвищення ефективності роботи та зростання доходів. Це в першу чергу пояснюється прискореним виконанням завдань і організаційних процесів, що дає змогу виконувати більше роботи за той самий термін.

Одним з показових прикладів рутинного бізнес-процесу, який піддається автоматизації, є генерація повторюваних звітів. Звіти відіграють вирішальну роль, оскільки вони надають керівникам відділів найновішу інформацію, але керівники часто стикаються з проблемами в організації великої кількості отриманих даних і представленні підсумкового аналізу керівництву. Замість того, щоб співробітники вручну вводили дані, створювали графіки в зовнішніх програмах і надсилали їх електронною поштою, усі ці дії можна автоматизувати через централізовану базу даних. Таке звільнення від завдань дозволяє співробітникам зосередитися на інших більш важливих обов'язках, які вимагають їхнього унікального досвіду і не можуть бути виконані лише програмним забезпеченням.

Організації, які використовують комплексну автоматизацію процесів, розширюють можливості своїх співробітників, звільняючи їх від монотонних

адміністративних обов'язків у повсякденній рутині. Це звільнення дозволяє працівникам перенаправляти свої навички та таланти на залучення проектів, які відповідають їхнім інтересам і є стратегічно вигідними для компанії.

Програмне забезпечення для автоматизації робочих процесів може застосовуватися до широкого кола бізнес-процесів:

- Ефективне управління часом: програмне забезпечення для автоматизації робочого процесу сприяє ефективному управлінню часом і спрощує процеси планування зустрічей.
- Продажі: програмне забезпечення для управління взаємовідносинами з клієнтами містить функції керування робочим процесом, автоматизацію заповнення форм і співпрацю між відділами.
- Аналіз: автоматизація робочого процесу може бути використана для аналітичних цілей, допомагаючи в ефективному виконанні завдань аналізу даних.
- Фінансові операції: автоматизація покращує узгодженість платежів, спрощує вимоги відповідності та дозволяє точніше прогнозувати доходи та процеси збору.
- Інформаційна безпека: програмне забезпечення для автоматизації зменшує ризики безпеки.

Переваги інтеграції комп'ютерної моделі як інструмент автоматизування бізнес-процесів фармацевтичної компанії:

- Ефективність використання часу: час є цінним ресурсом, і автоматизація відіграє вирішальну роль у збереженні дорогоцінних хвилин. Автоматизуючи завдання, компанії можуть виділяти більше часу та ресурсів на іншу продуктивну діяльність, підвищуючи загальну ефективність.
- Зменшення людських помилок: чим більше людей залучено до технічних процесів, тим вища ймовірність виникнення помилок. Автоматизація мінімізує людські помилки, усуваючи повторювані та

трудомісткі завдання, такі як введення даних вручну, забезпечуючи більшу точність і надійність.

- Підвищення безпеки: автоматизація забезпечує виявлення в режимі реального часу та реагування на проблеми безпеки, зміцнюючи загальні заходи безпеки організації. Завдяки автоматизації звітів про інциденти та інтеграції з існуючими інструментами безпеки компанії можуть ефективно пом'якшувати потенційні загрози.

- Оптимізація командних ресурсів: автоматизація дозволяє підприємствам оптимізувати організаційну структуру своїх співробітників. Замість того, щоб розширювати робочу силу для виконання додаткових завдань, автоматизація оптимізує процеси та зміцнює різні відділи. Ця оптимізація покращує швидкість, ефективність і результативність у всій організації.

- Забезпечення масштабованості: зростання є загальною метою для бізнесу, і комп'ютерна модель відіграє вирішальну роль у досягненні масштабованості. Компанії можуть легко обслуговувати більшу клієнтську базу, незалежно від їхнього географічного розташування. Миттєва масштабованість забезпечує конкурентну перевагу, дозволяючи компаніям розширювати свої пропозиції, продукти, комунікаційні платформи та охоплення ринку.

Сьогодні компанії все частіше використовують цифрові рішення для управління різними аспектами своєї організації. Інтеграція комп'ютерної моделі відіграє важливу роль у підвищенні ефективності роботи з кожною використовуваною програмою. Це допомагає покращити бізнес-процеси та робочі процеси, що призводить до зменшення бюджетних витрат та підвищення прибутковості бізнесу.

1.4 Аналіз сучасних методів розробки веб-додатків

Веб-додаток – це програмне забезпечення або програма, яку можна відкрити за допомогою будь-якого браузера.

Розробка веб-додатків включає в себе інтеграцію зовнішніх (Front-end) і внутрішніх (Back-end) компонентів, що дозволяє створювати графічний інтерфейс користувача (GUI), який полегшує взаємодію користувача з програмою. Цей процес ґрунтується на таких основних веб-технологіях і інструментах на стороні клієнта (FE), як HTML, CSS і JavaScript

HTML служить мовою розмітки, яка визначає структуру веб-сторінок, надаючи їм візуальний вигляд. [13] Він є основою для будь-якої веб-програми. І навпаки, CSS діє як мова таблиць стилів, покращуючи візуальне представлення та естетичну привабливість документів HTML. [14]

JavaScript – мультипарадигмова мова програмування відіграє важливу роль у маніпулюванні об'єктною моделлю документа HTML (DOM) для надання інтерактивного інтерфейсу користувачам. Це дає змогу реалізувати динамічні функції, такі як спливаючі повідомлення, перевірка введених форм і адаптивне коригування макета на основі дій користувача або клацань миші. Ці технології виконуються веб-браузерами, щоб забезпечити зручний веб-інтерфейс. [15]

На стороні сервера веб-розробка передбачає створення комп'ютерних програм і баз даних, які обслуговують функціональні можливості та технічне обслуговування програми. Серверні технології зазвичай охоплюють такі мови програмування, як .Net (C#), Java, PHP, Ruby, Python, Node.js, а також різноманітні фреймворки.

Зв'язок між клієнтом і сервером забезпечується за допомогою протоколів зв'язку. Протоколи - набір правил для обміну повідомленнями в мережі зв'язку. [16] Існують наступні протоколи:

- HTTP - спочатку розроблений для передачі гіпертекстових документів у форматі HTML, перетворився на протокол прикладного рівня, який полегшує передачу різних типів даних.
- TCP/IP - це набір мережевих протоколів, які дозволяють комп'ютерам і пристроям спілкуватися один з одним через Інтернет. Він забезпечує надійний і стандартизований метод передачі даних, гарантуючи належну доставку та прийом пакетів даних між пристроями.
- FTP - це стандартний мережевий протокол, який використовується для передачі файлів між клієнтом і сервером у комп'ютерній мережі. Він забезпечує простий і ефективний спосіб віддаленого завантаження, завантаження та керування файлами. FTP працює за архітектурою клієнт-сервер, де клієнт ініціює з'єднання та надсилає команди на сервер для виконання операцій передачі файлів. Він зазвичай використовується для обслуговування веб-сайтів, оновлення програмного забезпечення та обміну файлами.
- SMTP - це стандартний протокол зв'язку, який використовується для надсилання та доставки повідомлень електронної пошти між серверами електронної пошти.
- SOAP - це протокол обміну повідомленнями, який використовується для обміну структурованою інформацією на основі XML між веб-службами в мережі.
- REST - це архітектурний стиль і протокол, які використовуються для розробки мережевих програм. Це легкий і масштабований підхід, який підкреслює простоту, гнучкість і використання стандартних методів HTTP для зв'язку. RESTful API (інтерфейси прикладного програмування) дотримуються принципів REST і дозволяють клієнтам взаємодіяти з ресурсами сервера через Інтернет.

Для збереження та легкого доступу до даних використовують системи управління базами даних. [17] Найпопулярніші серед них:

- Реляційна система керування базами даних (RDBMS): RDBMS є найбільш широко використовуваним типом СУБД, заснованим на реляційній моделі. Він організовує дані в таблиці з рядками та стовпцями та забезпечує зв'язок між таблицями за допомогою ключів.
- Об'єктно-орієнтована система керування базами даних (OODBMS): OODBMS розроблена для обробки складних і взаємопов'язаних структур даних, де об'єкти з атрибутами та методами зберігаються та керуються безпосередньо.
- Ієрархічна система керування базами даних: ієрархічна СУБД організовує дані в деревоподібній структурі з батьківсько-дочірніми зв'язками між записами. Цей тип СУБД в основному використовується в застарілих системах і спеціалізованих програмах.
- Система керування мережевою базою даних: мережева СУБД організовує дані за допомогою мережевої моделі, де записи з'єднані у графоподібну структуру.
- Система керування базами даних NoSQL: СУБД NoSQL – це широка категорія, яка включає нереляційні бази даних, призначені для обробки великомасштабних розподілених даних і забезпечення високої продуктивності та масштабованості.
- Система керування базами даних у пам'яті (IMDBMS): IMDBMS зберігає дані переважно в пам'яті, а не на диску, що забезпечує швидший доступ до даних і обробку. Він зазвичай використовується в програмах, які потребують аналізу даних у реальному часі та високошвидкісних транзакцій.

Зрештою, зараз повсякчасно використовують фреймворки для розробки веб-додатків.

Фреймворк – це деякий набір інструментів, бібліотек та шаблонів для створення програмного забезпечення в тій сфері, під яку був створений, рідше бувають загальнопризначеними. Фреймворки допомагають дуже сильно прискорити процес розробки, бо вже покривають потреби та задачі, що

потрібно вирішувати з абсолютно кожним проектом. В ІТ-суспільстві постійно говорять, що не потрібно винаходити та розробляти інструменти, які вже існують та вирішують поставлені задачі, якщо це робиться не для навчання. І це правда, бо фреймворк зберігають час та, як правило, мають більш ефективні рішення, бо розробка фреймворку або бібліотеки – це окремий проект де вся увага приділяється тільки розробці ефективного рішення. Якби сказав, що можна провести аналогію з кухарем відомого ресторану, він може сам виростити та підготувати усі продукти для страви, але скільки клієнти чекали своєю їжу? Та й ніхто не дає гарантії того, що він професіональний фермер. Тому розробники намагаються якомога рідше вдаватися до розробки своїх інструментів під час роботи над проектом.

Сьогодні найпопулярнішими фреймворками для розробки графічних інтерфейсів веб-проектів на стороні клієнта (Front-end):

- Для мови JavaScript – AngularJS, ReactJS, VueJS. Це фреймворки для розробки графічних інтерфейсів на стороні клієнта (Front-end).

Для Back-end розробки:

- PHP – Laravel, Symfony, CodeIgniter.
- Java – Spring, JavaServer Faces (JSF), Play Framework.
- .Net (C#) – ASP.NET, .NET Core.
- Go - Gin, Revel, Echo.
- Python – Django, Flask, Pyramid.
- Ruby - Ruby on Rails, Sinatra, Hanami.

1.5. Мета, предмет, об'єкт та постановка задачі.

Метою кваліфікаційної роботи є розробка перспективної комп'ютерної моделі системи управління та автоматизації бізнес-процесів фармацевтичної компанії для українського ринку.

Об'єкт дослідження – процес моделювання системи автоматизації бізнес-процесів фармацевтичної компанії.

Предмет дослідження – комп’ютерні моделі автоматизації бізнес-процесів фармацевтичної компанії і методи їх аналізу.

Постановка задачі: розробити комп’ютерну систему автоматизації бізнес-процесів фармацевтичної компанії використовуючи обрану мною мову програмування C# та фреймворк ASP.NET MVC, СУБД – Microsoft sql server.

У веб-додатку повинний бути розроблений наступний функціонал: авторизація користувачем. Розподілення доступу на ролі. Панель керування користувачами, оновлення їх інформації та паролю. Панель керування продуктами, створення звітів продаж. Перегляд звітів продаж та закупівель. Панелі клієнтури. Панель-календар зустрічей – можливість створити зустріч та стежити за її станом. Панелі ефективності персоналу.

Висновки за розділом 1.

Було розглянуто основні інструменти, методи та поняття які дозволили отримати загальне уявлення про основні складові частини процесу розробки комп’ютерної системи та веб-додатку, автоматизації бізнес-процесів фармацевтичної компанії. Також були оглянуті технології створення веб-додатків, що допомогло обрати інструменти розробки та виділити чіткі кроки реалізації проекту. Також поставлені задачі, які необхідно реалізувати.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА

КОМП'ЮТЕРНОЇ МОДЕЛІ ВЕБ-ДОДАТКУ

2.1 Проектування архітектури веб-додатку.

Для розуміння вимог користувачів та визначення функціональності веб-додатку використовується діаграма use case (діаграма використання). Ця діаграма дозволяє ідентифікувати основні актори, які взаємодіють з додатком, та описує їх взаємодію з системою через різні сценарії використання.

На діаграмі use case можуть бути відображені різні варіанти використання додатку, такі як реєстрація користувачів, авторизація, створення, редагування та видалення об'єктів, а також інші дії, які виконуються користувачами або іншими акторами.

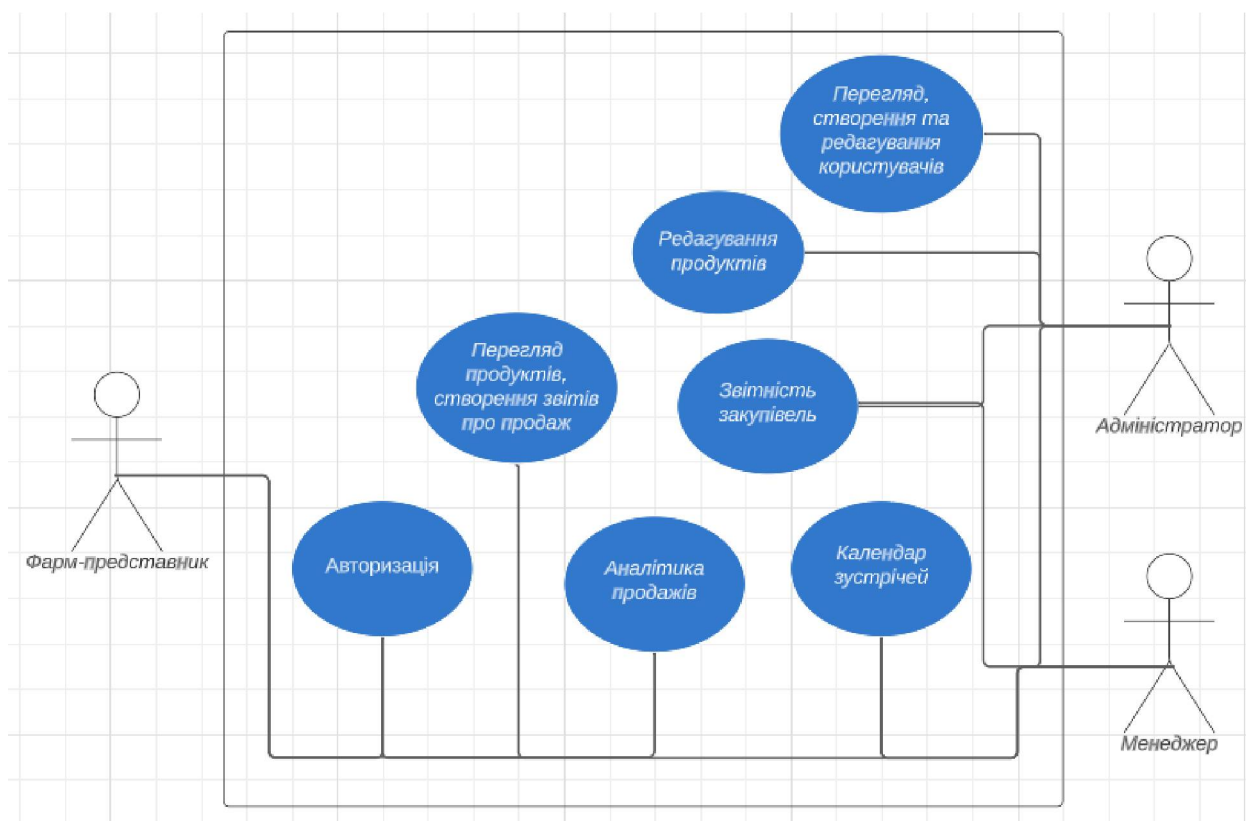


Рисунок 2.1 – Діаграма використання веб-додатку.

На представленій вище діаграмі використання можна побачити деякі заплановані сценарії для реалізації у веб-додатку, а також ролі для користувачів.

Для розробки веб-додатку була обрана трирівнева архітектура – це шаблон проектування, який розділяє програму на окремі рівні, кожен зі своїми обов'язками та функціями. [18] Її класичний варіант:

- Рівень представлення: є рівнем взаємодії програми з користувачем за допомогою інтерфейсу користувача (UI).
- Рівень бізнес-логіки: є рівнем збереження моделей, компонентів, обробки бізнес-логіки. Запрошує дані у рівні даних, обробляє та передає до рівня представлення.
- Рівень доступу до даних: є рівнем збереження даних та моделей сутностей. Він взаємодіє з базою даних або зовнішніми джерелами даних для виконання таких операцій, як читання, запис і оновлення даних. Цей рівень інкапсулює деталі технологій доступу до даних, таких як SQL-запити або структури об'єктно-реляційного відображення (ORM).

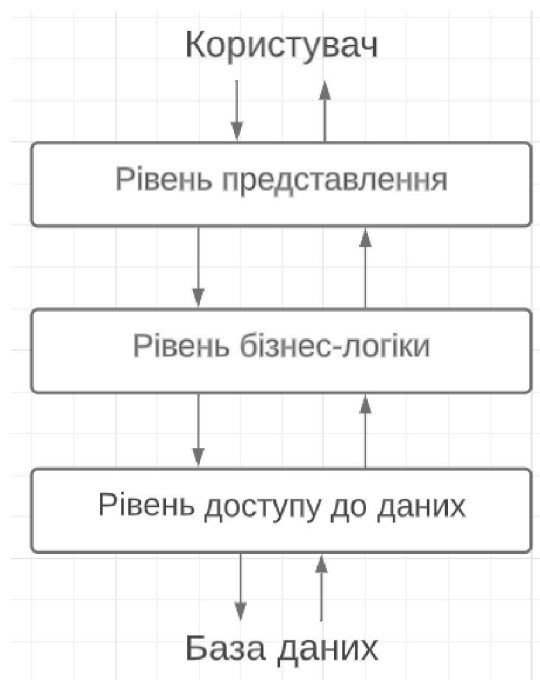


Рисунок 2.2 – Трирівнева архітектура.

У багаторівневій архітектурі важливо підкреслити, що різні рівні мають певні обмеження щодо їх взаємодії. Наприклад, рівень представлення не повинен отримувати прямий доступ до бази даних або рівня доступу до даних. Замість цього він повинен покладатися на рівень бізнес-логіки.

Рівень доступу до даних працює незалежно від інших рівнів, зосереджуючись виключно на отриманні та зберіганні даних. Рівень бізнес-логіки залежить від рівня доступу до даних для отримання та обробки даних. Нарешті, рівень презентації залежить від рівня бізнес-логіки для обробки бізнес-операцій і надання необхідних даних для представлення користувачам.

Таким чином досягається слабкий зв'язок та підтримка модульності програми. Це полегшує обслуговування та надає можливість замінювати реалізації компонентів, не впливаючи на інші частини програми.

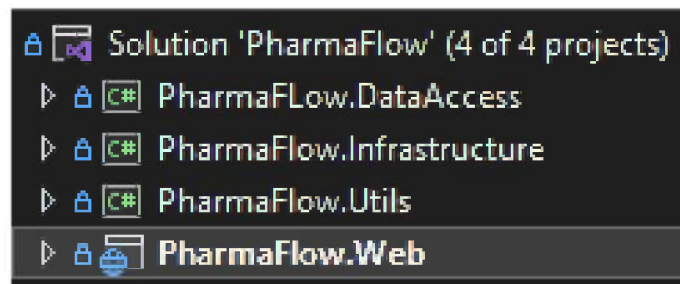


Рисунок 2.3 – Проекти веб-додатку.

На скріншоті провідника рішень ми можемо побачити чотири рівня, що розділені шляхом створення різних проектів. PharmaFlow.Web – рівень представлення, PharmaFlow.DataAccess – рівень доступу до даних. В ході роботи було прийняте рішення, що бізнес логіка веб-додатку не є занадто складною для виділення її до окремого рівня, натомість було запропоновано розподілити цю логіку між контролерами рівня представлення та репозиторіями рівня доступу до даних. Тож PharmaFlow.Infrastructure – рівень що зберігає моделі DTO – шаблон проектування, що використовується для передачі даних між рівнями програми. Рівень інфраструктури потрібен нам,

щоб позбутися залежності рівнів представлення та доступу до даних через відсутність проміжного рівня обробки даних – рівня бізнес логіки. PharmaFlow.Utils не є рівнем даних, це бібліотека, що була створена задля реалізації та використання різних необхідних програмних рішень у всьому проекті.

2.2 Рівень доступу до даних.

В якості СУБД я використав Microsoft SQL Server (MS SQL Server) з реляційною базою даних. Також я використовую технологію Entity Framework Core (EF Core) – фреймворк об'єктно реляційного перетворення даних.

EF Core дозволяє розробникам працювати з базами даних, використовуючи об'єктно-орієнтовані принципи. Він перетворює таблиці бази даних на класи, а стовпці бази даних – на властивості класу. [19]

```
// Database settings.  
services.AddDbContext<PharmaFlowDbContext>();
```

Рисунок 2.5 – Реєстрація класу, що наслідує DbContext.

Клас DbContext у Entity Framework Core (EF Core) є основною точкою входу для взаємодії з базою даних. Він надає API високого рівня для операцій CRUD, запитів, відстеження змін і керування транзакціями. За допомогою властивостей «DbSet» він представляє набори сутностей, що відповідають таблицям бази даних. «DbContext» спрощує взаємодію з базою даних, абстрагуючись від низькорівневих деталей, дозволяючи розробникам зосередитися на бізнес-логіці програми та маніпулюванні даними.

Також в запропонованій моделі використано підхід «First code»: де спочатку створюються моделі – сутності та конфігурації. Таким чином усувається потреба в скриптах SQL взагалі, для чого EF Core і був створений.

```

public record UserPersistence
{
    15 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    public Guid ID { get; set; }

    8 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    public required string FirstName { get; set; }

    8 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    public required string LastName { get; set; }

    10 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    public required UserRolePersistence Role { get; set; } = UserRolePersistence.Pharmacist;

    20 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    public required string Email { get; set; }

    8 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    public required string PasswordHash { get; set; }

    17 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    public UserStatePersistence State { get; set; } = UserStatePersistence.Active;

    // Sales reports.
    1 reference | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    public List<InputReportPersistence> InputReports { get; set; } = new();
}

```

Рисунок 2.6 – Модель сутності користувача.

На рис 2.6 ми можемо побачити сутність користувача дещо незвичайного типу «record» - це насправді модифікатор для визначення посилального типу «record class», який з'явився в С# 9. Цей тип надає вбудовані функції для інкапсуляції даних й насправді є так званим «синтаксичним цукром» і є надбудовою над звичайним класом. Має ідентифікатор «ID» та властивості з корисною інформацією.

Далі у «DbContext» додав «DbSet» з аргументом типу «UserPersistence» задля представлення набору сутності, що відповідає таблиці бази даних.

```

17 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
public DbSet<UserPersistence> Users { get; set; }

```

Рисунок 2.7 – Властивість «DbSet» з аргументом «UserPersistence»

Також потрібна конфігурація типу сутності задля налаштування відображення в таблиці й визначення додаткової поведінки або обмежень. Передбачає використання «Fluent API» або анотацій даних для налаштування різних

аспектів сутності, таких як ім'я таблиці, імена стовпців, зв'язки та типи даних.

```

internal class UserPersistenceMap : IEntityTypeConfiguration<UserPersistence>
{
    0 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    public void Configure(EntityTypeBuilder<UserPersistence> builder)
    {
        builder.ToTable("user");

        builder.HasKey(u => u.ID);

        builder.Property(u => u.ID)
            .ValueGeneratedOnAdd()
            .HasColumnName("user_id");

        builder.Property(u => u.FirstName)
            .HasMaxLength(255)
            .IsRequired();

        builder.Property(u => u.LastName)
            .HasMaxLength(255)
            .IsRequired();

        builder.Property(u => u.Email)
            .HasMaxLength(255)
            .IsRequired();

        builder.Property(u => u.PasswordHash)
            .HasMaxLength(1024)
            .IsRequired();

        builder.Property(u => u.Role)
            .HasConversion<int>();

        builder.Property(u => u.State)
            .HasConversion<int>();
    }
}

```

Рисунок 2.8 – Клас конфігурації сутності користувача.

На рис 2.8 бачимо, що крім налаштувань властивостей імені, імейлу, ролі та стану, даємо назву таблиці, що буде створена на основі набору та визначення первинного ключа.

На рис 2.9 бачимо діаграму бази даних, що була створена на основі моделей – сутностей за допомогою підходу «First Code» EF Core. Також EF Core пропонує загальний API, який абстрагує базову систему бази даних. Це означає, що при переході на іншу систему керування базами даних (СУБД),

основні зміни у проекті стосуватимуться налаштування та підключення до нового постачальника. Код, відповідальний за роботу з даними, залишиться в основному незмінним.



Рисунок 2.9 – Діаграма бази даних веб-додатку

В додаткового рівня абстракції для рівня доступу до даних я використав шаблон «Repository». Цей шаблон суттєво допомагає розробці

шляхом повторного використання коду, структурує доступ до даних та забезпечує його інтерфейс.

```

6 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
public interface IUserRepository
{
    2 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    Task<List<UserDto>> GetUserListAsync();

    3 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    Task<UserDto> GetUserByEmailAsync(string email);

    2 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    Task AddUserAsync(CreateUserRequest request);

    2 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    Task UpdateUserAsync(UpdateUserRequest request);

    2 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    Task UpdateUserEmailAsync(Guid userID, string email);

    2 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    Task UpdateUserPasswordAsync(Guid userID, string passwordHash);

    2 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    Task RemoveUserByIDAsync(Guid userID);
}

```

Рисунок 2.10 – Інтерфейс репозиторію користувача з оголошенням його методів.

На рис 2.10 бачимо, що методи репозиторію використовують DTO моделі щоб позбутися залежності від будь-яких моделей інших рівнів.

На рис 2.11 приклад реалізації одного з методів репозиторію, а саме «GetUserListAsync». Звертаючись до «DbSet» користувачів фільтруємо бажаний список по стану. EF Core трансліює C# код sql-запиту та навпаки з реляційного до об'єктно орієнтованого представлення даних при отриманні. Повертаємо лист користувачів, попередньо перетворивши його на лист моделей DTO.

```

public class UserRepository : IUserRepository
{
    private readonly PharmaFlowDbContext _db;

    0 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    public UserRepository(PharmaFlowDbContext db)
    {
        _db = db;
    }

    2 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    public async Task<List<UserDto>> GetUserListAsync()
    {
        try
        {
            List<UserPersistence> users = await _db.Users
                .Where(u => u.State == UserStatePersistence.Active)
                .ToListAsync();

            return users.ToUserDtoList();
        }
        catch (Exception)
        {
            throw;
        }
    }
}

```

Рисунок 2.11 – Реалізація методу репозиторія для отримання списку користувачів

2.3 Рівень інфраструктури.

Рівень реалізує шаблон DTO тому містить лише моделі, при тому крім цих моделей, які в основному агрегують в собі об'єктні залежності сутностей у вигляді звичайних властивостей примітивних типів, є також деякі «Requests» моделі. Вони містять мінімальну кількість параметрів потрібну тільки конкретному методу репозиторію рівню даних, тим самим позбавляють необхідності передавання звичайних DTO моделей з деякою кількістю неініціалізованих властивостей. [20]

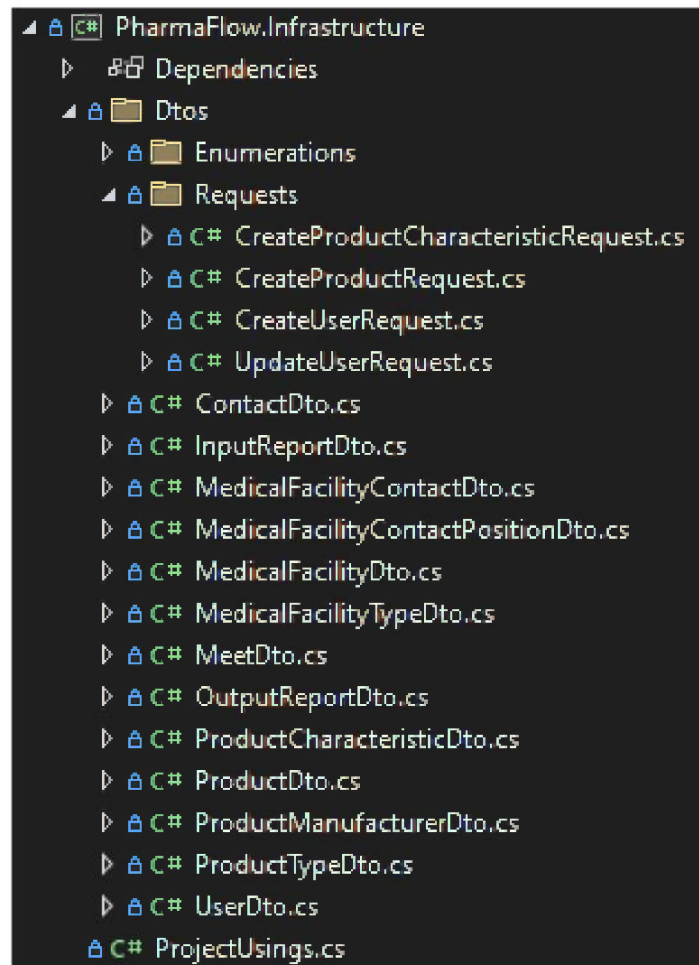


Рисунок 2.12 – Моделі рівню інфраструктури.

2.4 Рівень представлення.

Для розробки рівня представлення була обрана платформа ASP.NET MVC – це фреймворк для розробки веб-додатків, що реалізує архітектурний шаблон модель – представлення – контролер (MVC). [21]

- Модель: інкапсулює дані та бізнес-логіку програми.
- Представлення: інтерфейс користувача, що використовує моделі задля відображення або збору та збереження даних. Використовують синтаксис HTML, CSS та Razor – це мова розмітки, яка використовується в ASP.NET MVC, поєднує мову розмітки HTML та код на сервері (C#) задля створення кінцевого результату, що надсилається до браузера клієнта.

- Контролер: посередник між моделями для представленнями. Обробляє запити, зберігає або отримує запрошені дані з бази даних та вибирає відповідне представлення для створення відповіді у вигляді веб-сторінки.

Саме в цьому проекті (PharmaFlow.Web) знаходиться точка входу до програми.

```
internal class Program
{
    0 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    private static void Main(string[] args)
    {
        Host.CreateDefaultBuilder(args)
            .ConfigureWebHostDefaults(webBuilder => webBuilder.UseStartup<Startup>()).Build().Run();
    }
}
```

2.13 Точка входу до програми.

Метод «Host.CreateDefaultBuilder», який можна побачити на рис 2.13 створює хост-конструктор за замовчуванням, також можна вказати аргумент для методу «UseStartup» для налаштувань. Зазвичай клас аргументу має відповідну до метода назву. Хост-конструктор автоматично налаштовує конфігурацію програми використовуючи джерела конфігурації, таких як файл «appsettings.json».

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "SqlConnection": "Server=pharmaflow.database.windows.net;Database=PharmaFlow;User Id=alexandr.cherednyk;Password=pa33word!"
  }
}
```

Рисунок 2.14 – Файл-джерело конфігурації.

На рис 2.14 можна побачити налаштування постачальника журналів за замовчуванням та строку підключення бази даних веб-додатку.

Також хост-конструктор встановлює контейнер впровадження залежностей (DI) за замовчуванням, що дозволяє реєструвати та застосовувати служби в усій програмі. Він забезпечує зручний спосіб конфігурації та керування залежностями за допомогою принципів інверсії контролю.

Впровадження залежностей (DI) – механізм ASP.NET, який дозволяє зв'язати об'єкти системи через абстракції (інтерфейси), таким чином залежність об'єктів стає нижча, а сама система більш гнучкою. На рис 2.15 можна побачити приклад інверсії контролю, де зіставляється інтерфейс зі своєю реалізацією. Після налаштування DI для деякого сервісу, можна скористатись впровадженням залежностей, як приклад: впровадження репозиторія в контролері користувача рис 2.16.

```
public void ConfigureServices(IServiceCollection services)
{
    // Add services to the container.
    services.AddRazorPages();

    services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)
        .AddCookie(options =>
        {
            options.LoginPath = new PathString("/Identity/Account/Login");
            options.AccessDeniedPath = new PathString("/Identity/Account/Login");
        });

    // Database settings.
    services.AddDbContext<PharmaFlowDbContext>();

    //IoC.
    services.AddScoped<IUserRepository, UserRepository>();
    services.AddScoped<IProductRepository, ProductRepository>();
    services.AddScoped<IReportRepository, ReportRepository>();
    services.AddScoped<IMedicalFacilityRepository, MedicalFacilityRepository>();
    services.AddScoped<IContactRepository, ContactRepository>();
    services.AddScoped<IReportRepository, ReportRepository>();
    services.AddScoped<IMeetRepository, MeetRepository>();
}
```

2.15 Інверсія контролю репозиторіїв.

```

public class UserController : Controller
{
    private readonly IUserRepository _userRepository;

    0 references | Alexandr Cherednyk, 9 days ago | 1 author, 1 change
    public UserController(IUserRepository userRepository)
    {
        _userRepository = userRepository;
    }
}

```

2.16 Впровадження залежності для репозиторія користувача.

Для захисту програми від несанкціонованого доступу були налаштовані автентифікація та авторизація. На рис 2.15 метод «AddAuthentication» налаштований на основі файлів «cookie». При успішній авторизації користувач буде автентифікований та отримує файли «cookie» рис 2.17.

```

private async Task Authenticate(UserViewModel user)
{
    var claims = new List<Claim>
    {
        new Claim(ClaimsIdentity.DefaultNameClaimType, user.Email),
        new Claim(ClaimsIdentity.DefaultRoleClaimType, Enum.GetName(typeof(UserRoleViewModel), user.Role!)),
    };

    ClaimsIdentity id = new(
        claims,
        CookieAuthenticationDefaults.AuthenticationScheme,
        ClaimsIdentity.DefaultNameClaimType,
        ClaimsIdentity.DefaultRoleClaimType);

    await HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, new ClaimsPrincipal(id));
}

```

2.17 Автентифікація на основі cookie-файлів.

В свою чергу контролери використовують фільтри автентифікація рис 2.18, та якщо користувач не увійшов до системи, або не має достатньо прав, йому буде відказано в доступі.

```

[Authorize(Roles = "Admin")]
1 reference | Alexandr Cherednyk, 8 days ago | 1 author, 2 changes
public class UserController : Controller
{

```

2.18 фільтр автентифікації для контролера користувача.

Також в роботі використовується обласна маршрутизація рис 2.19, її застосовують для організації та розділення різних функціональних розділів або модулів у програмі. Області забезпечують спосіб інкапсуляції пов'язаних контролерів, представлень та інших ресурсів у окремі папки, що забезпечує кращу організацію та зручність обслуговування. Для рішення задачі кваліфікаційної роботи створювалися області відповідно до ролей користувачів.

```
app.UseEndpoints(endpoints =>
{
    endpoints.MapControllerRoute(
        name: "areas",
        pattern: "{area=Pharmacist}/{controller=Calendar}/{action=GetCalendar}"
    );
});
```

Рисунок 2.19 – Налаштування обласної маршрутизації.

На рис 2.20 приклад роботи кінцевої точки контролера для отримання представлення панелі користувачів. За допомогою раніше впровадженої залежності репозиторія робимо запит даних до рівня даних. Отримавши список моделей DTO користувачів, перетворюємо його на лист моделей представлення та відправляємо на запит представлення із відповідними даними моделі.

```
[HttpGet]
0 references | Alexandr Cherednyk, 8 days ago | 1 author, 2 changes
public async Task<IActionResult> GetUserList()
{
    try
    {
        List<UserDto> users = await _userRepository.GetUserListAsync();
        List<UserViewModel> response = users.ToUserViewModelList();
        return View("UserPanel", response);
    }
    catch (Exception)
    {
        return View("~/Views/Error.cshtml");
    }
}
```

Рисунок 2.20 – Реалізація кінцевої точки контролера для отримання представлення панелі користувачів.

Висновки за розділом 2.

В даному розділі був розроблений веб-додаток для автоматизації бізнес процесів фармацевтичної компанії використовуючи мову програмування C# та фреймворк ASP.NET MVC, з використанням дещо адаптованого архітектурного шаблону – трирівневої архітектури та СУБД – Microsoft sql server. Також були використанні різноманітні архітектурні шаблони та підходи для вирішення глобальних та локальних задач, що виникли під час розробки веб-додатку. Був розроблений наступний функціонал: авторизація користувачем, розподілення доступу на ролі, панель керування користувачами, оновлення їх інформації та паролю, панель керування продуктами, створення звітів продаж, перегляд звітів продаж та закупівель, панелі клієнтури, панель-календар зустрічей – можливість створити зустріч та стежити за її станом, панелі ефективності персоналу.

РОЗДІЛ 3.

РЕКОМЕНДАЦІЇ ПО ВИКОРИСТАННЮ КОМП'ЮТЕРНОЇ МОДЕЛІ

3.1 Сторінки, що доступні усім користувачам.

Коли користувач намагається перейти до будь-якої сторінки, йому потрібно пройти авторизацію. Самому зареєструватись неможливо, доступ та аккаунт видається тільки адміністратором задля безпеки даних системи.

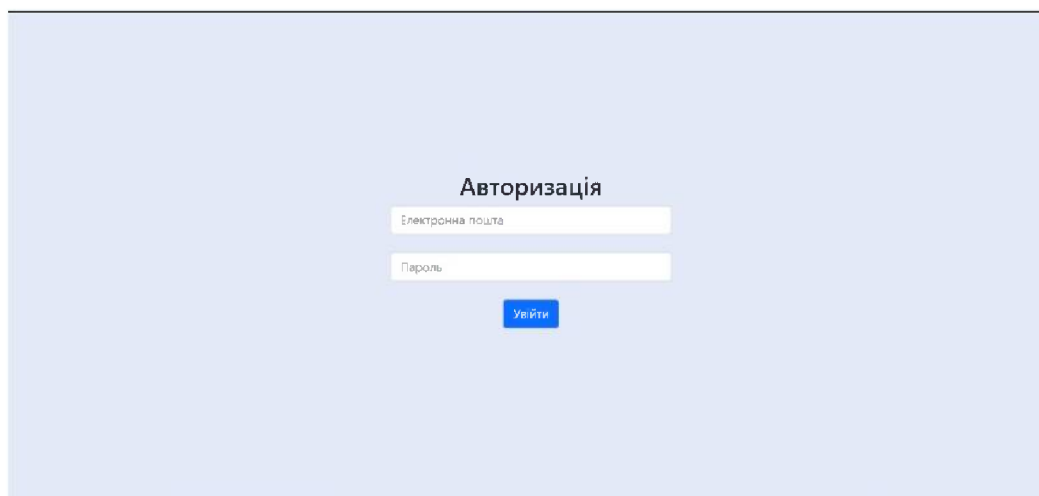


Рисунок 3.1 – Сторінка авторизації.

Домашня сторінка – календар зустрічей з відображеною картою для орієнтування між адресами та списком з загальною інформацією зустрічей.

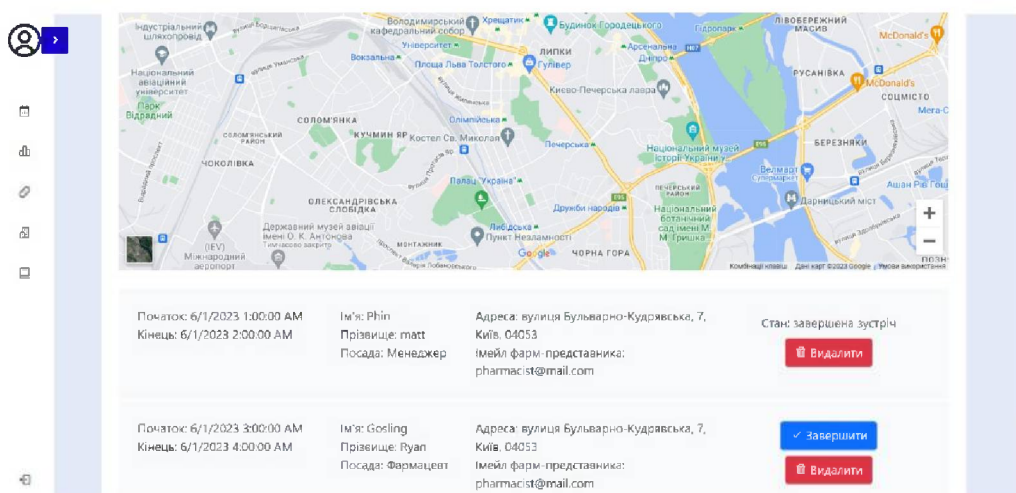


Рисунок 3.2 – Сторінка календаря зустрічей.

Кожен користувач може створювати зустріч з для себе з деяким працівником на відповідній сторінці, обравши час початку та кінця зустрічі й працівника для зустрічі.

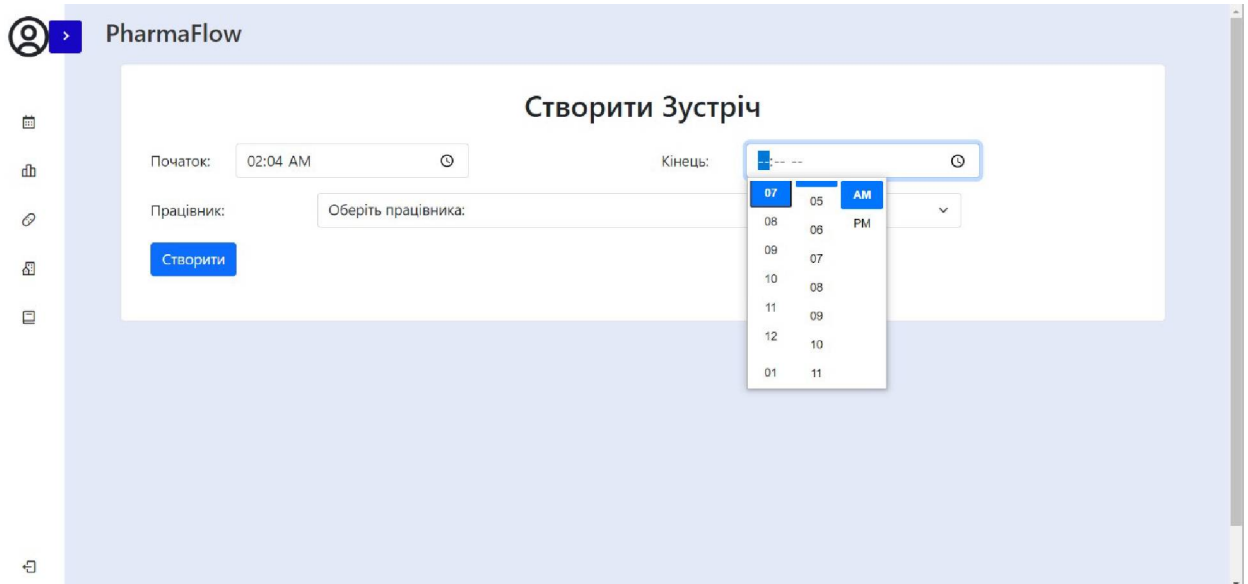


Рисунок 3.3 – Сторінка створення зустрічі.

Сторінка контактів клієнтів, для її демонстрації я авторизувався як адміністратор так як для фарм-представника не відображаються кнопки менеджменту запису.



Рисунок 3.4 – Сторінка контактів клієнтів.

Сторінка клієнтів з місцями їх роботи. Ролі адміністратора та менеджера також бачуть та можуть додати контакти працівників до місця роботи, або видалити записи.

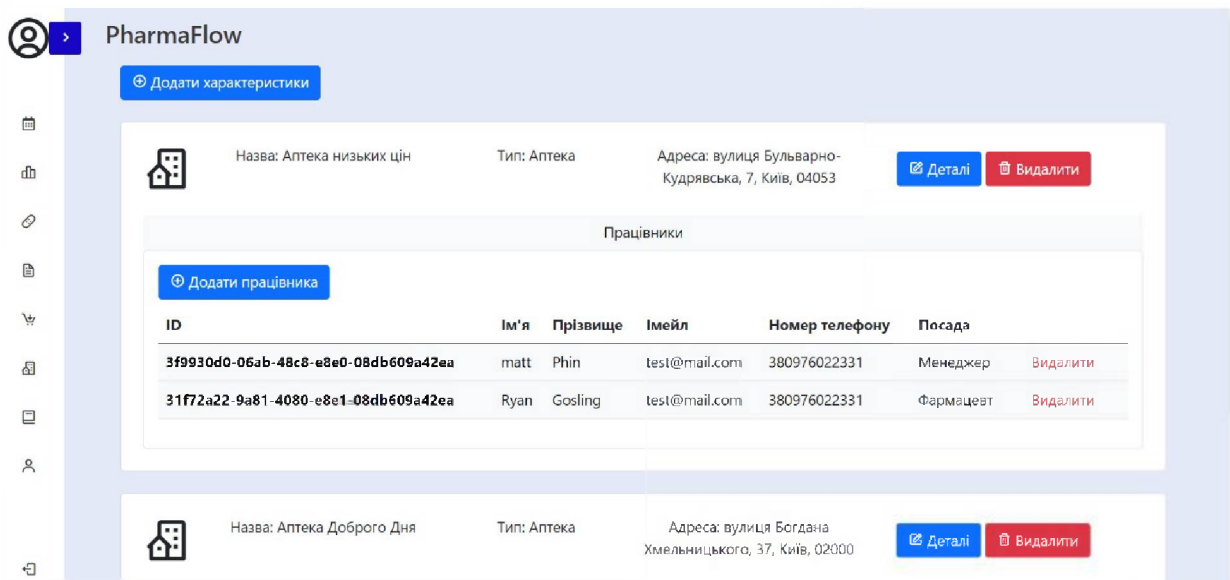


Рисунок 3.5 – Сторінка клієнтів з місцями їх роботи.

На сторінці продуктів крім відображення списку продукції ролі адміністратора та менеджера також можуть бачити кнопки для додавання характеристик таких як типи продуктів, та кнопку додавання продукту, видалення з бази даних.

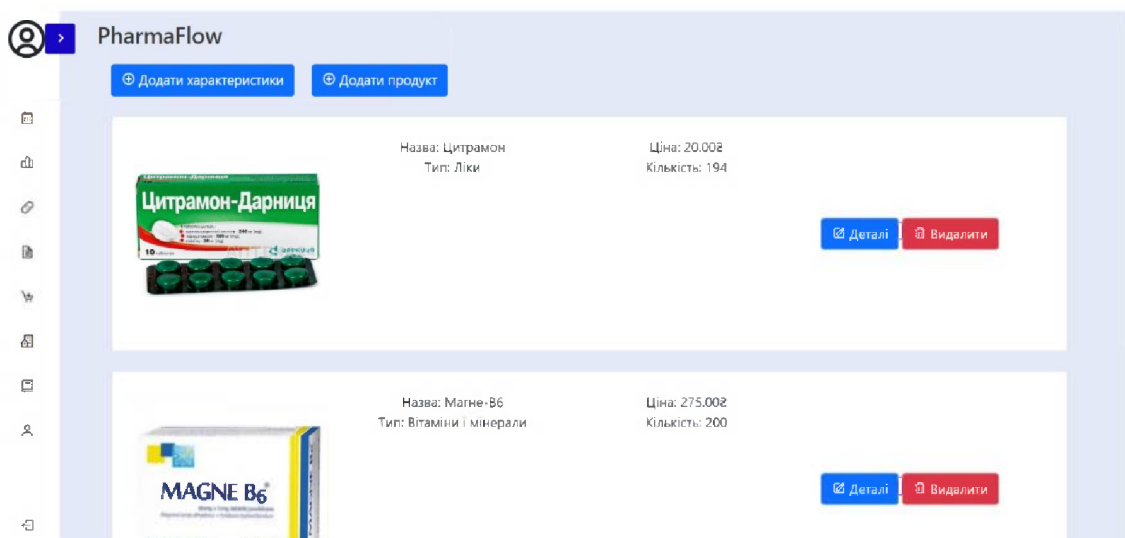
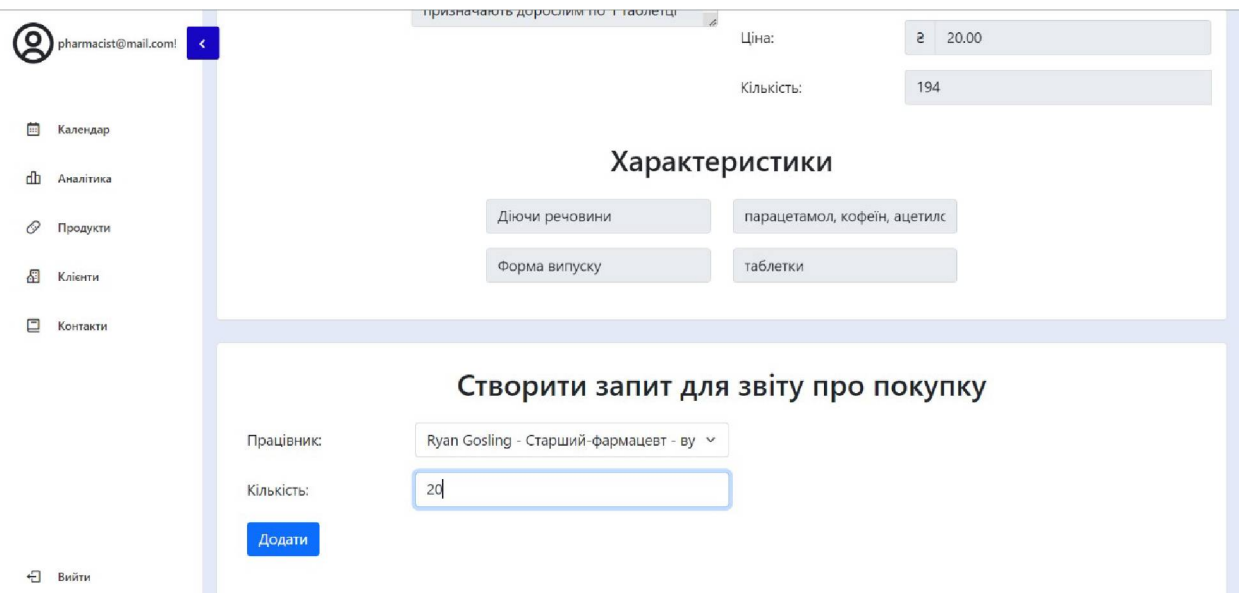


Рисунок 3.6 – Сторінка продуктів.

Сторінка деталей продукту. Також на цій сторінці можна створити запит продажу продукту. Але якщо користувач має ролі адміністратора або менеджера, то його звіт одразу стає підтвердженим.



The screenshot displays a web interface for a pharmacist. At the top left, there is a user profile for 'pharmacist@mail.com!' with a navigation menu containing 'Календар', 'Аналітика', 'Продукти', 'Клієнти', 'Контакти', and 'Вийти'. The main content area shows product details for 'призначають дорослим по 1 таблетці'. The price is set to 20.00 and the quantity to 194. Below this, the 'Характеристики' (Characteristics) section lists 'Діючі речовини' (Active ingredients) as 'парацетамол, кофеїн, ацетилс' and 'Форма випуску' (Release form) as 'таблетки'. At the bottom, a section titled 'Створити запит для звіту про покупку' (Create request for purchase report) includes a dropdown menu for 'Працівник' (Employee) set to 'Ryan Gosling - Старший-фармацевт - ву', a text input for 'Кількість' (Quantity) with '20' entered, and a blue 'Додати' (Add) button.

Рисунок 3.7 – Сторінка створення запиту/звіту про продаж

Сторінка звітності продажу за місяць, крім таблиці звітів продаж, можна побачити графік продаж за місяць. Якщо навести курсор миші на точку, що відповідає звіту з продажу, можна побачити дату та час підтвердження звіту.



Рисунок 3.8 – Сторінка звітності з продажу.

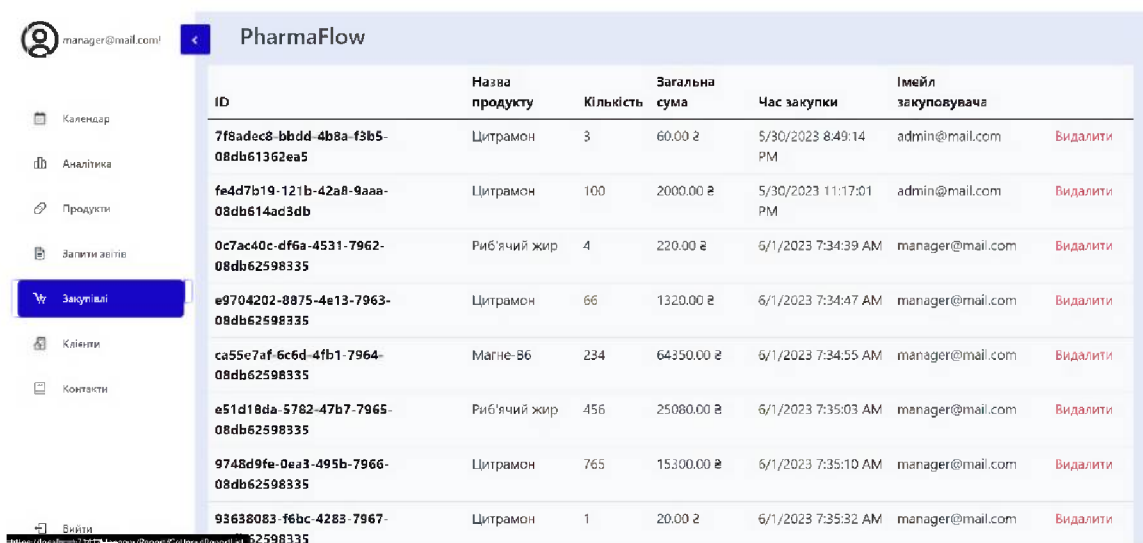
3.2 Сторінки, що доступні адміністратору та менеджеру.

Сторінка запитів звітів продажів, та з подальшою можливістю підтвердження або видалення запиту.

ID	Назва продукту	Кількість	Загальна сума	Час створення	Створив	ID працівника	Підтвердити	Видалити
4485146f-68f5-4276-3e76-08db61f42422	Цитрамон	5	100.00 ₴	5/31/2023 7:29:00 PM	pharmacist@mail.com	a0219414-c12b-44e7-e8e2-08db609a42ea	Підтвердити	Видалити
e6455484-a19d-4509-3e77-08db61f42422	Магне В6	10	2750.00 ₴	5/31/2023 7:29:15 PM	pharmacist@mail.com	539e6d80-387b-4837-e8e3-08db609a42ea	Підтвердити	Видалити
16b54986-c285-4dee-3e78-08db61f42422	Риб'ячий жир	25	1375.00 ₴	5/31/2023 7:29:31 PM	pharmacist@mail.com	3f9930d0-06ab-48c8-e8e0-08db609a42ea	Підтвердити	Видалити
5db512b5-8545-4920-a09d-08db6257a75b	Цитрамон	30	600.00 ₴	6/1/2023 7:21:21 AM	pharmacist@mail.com	3f9930d0-06ab-48c8-e8e0-08db609a42ea	Підтвердити	Видалити
3d46747c-ffa5-427f-a09e-08db6257a75b	Риб'ячий жир	5	275.00 ₴	6/1/2023 7:29:23 AM	pharmacist@mail.com	3f9930d0-06ab-48c8-e8e0-08db609a42ea	Підтвердити	Видалити
fcaa262e-7c1e-40ba-a09f-08db6257a75b	Магне В6	12	3300.00 ₴	6/1/2023 7:29:32 AM	pharmacist@mail.com	31f72a22-9a81-4080-e8e1-08db609a42ea	Підтвердити	Видалити
7f84445d-4fe8-4ca9-a0a0-08db6257a75b	Риб'ячий жир	60	3300.00 ₴	6/1/2023 7:29:42 AM	pharmacist@mail.com	31f72a22-9a81-4080-e8e1-08db609a42ea	Підтвердити	Видалити
94de1c90-4d89-460e-8e8e-08db6257a75b	Риб'ячий жир	17	935.00 ₴	6/1/2023 7:29:54 AM	pharmacist@mail.com	31f72a22-9a81-4080-e8e1-08db609a42ea	Підтвердити	Видалити

Рисунок 3.9 – Сторінка запитів звітів з продаж

Сторінка звітів закупівель продуктів з загальною інформацією про користувача, що створив звіт закупівлі, назву, час закупки, кількість та

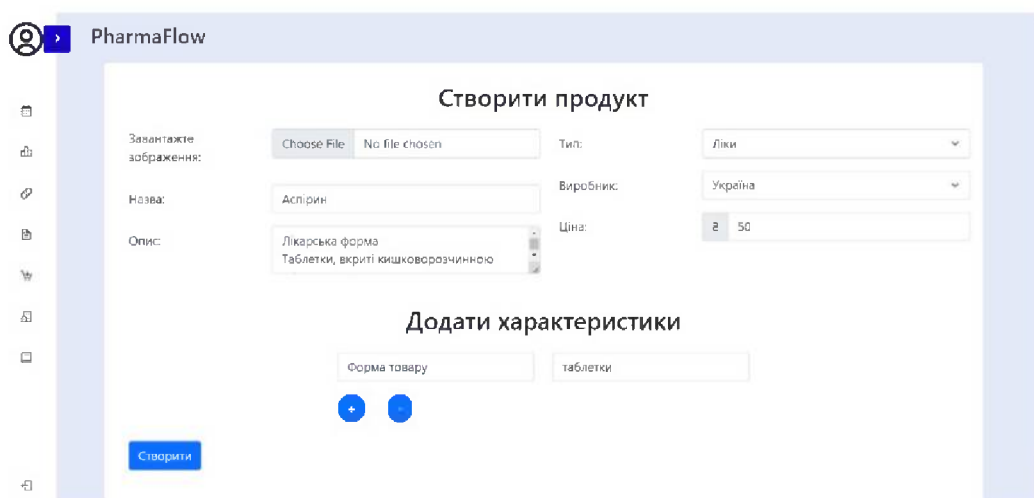


ID	Назва продукту	Кількість	Загальна сума	Час закупки	Ім'ял закуповувача	
7f8adec8-bbdd-4b8a-f3b5-08db61362ea5	Цитрамон	3	60.00 ₪	5/30/2023 8:49:14 PM	admin@mail.com	Видалити
fe4d7b19-121b-42a8-9aaa-08db614ac3db	Цитрамон	100	2000.00 ₪	5/30/2023 11:17:01 PM	admin@mail.com	Видалити
0c7ac40c-df6a-4531-7962-08db62598335	Риб'ячий жир	4	220.00 ₪	6/1/2023 7:34:39 AM	manager@mail.com	Видалити
e9704202-8875-4e13-7963-08db62598335	Цитрамон	66	1320.00 ₪	6/1/2023 7:34:47 AM	manager@mail.com	Видалити
ca55c7af-6c6d-4fb1-7964-08db62598335	Магне-Б6	234	64350.00 ₪	6/1/2023 7:34:55 AM	manager@mail.com	Видалити
e51d18da-5762-47b7-7965-08db62598335	Риб'ячий жир	456	25080.00 ₪	6/1/2023 7:35:03 AM	manager@mail.com	Видалити
9748d9fe-0ea3-495b-7966-08db62598335	Цитрамон	765	15300.00 ₪	6/1/2023 7:35:10 AM	manager@mail.com	Видалити
93638083-f6bc-4283-7967-08db62598335	Цитрамон	1	20.00 ₪	6/1/2023 7:35:32 AM	manager@mail.com	Видалити

загальну суму.

Рисунок 3.10 – Сторінка звітів закупівель.

На сторінці створення продукту можна завантажити зображення продукту, заповнити назву, опис, тип, який вже доданий до бази даних а якщо ні то можна це зробити шляхом переходу до сторінки продуктів Рис. 3.6 та перейти до додавання характеристик.



Створити продукт

Завантажте зображення: No file chosen

Тип:

Назва:

Виробник:

Опис:

Ціна:

Додати характеристики

Рисунок 3.11 – Сторінка створення продукту.

3.3 Сторінки, що доступні адміністратору.

Сторінка панелі користувачів, на якій можна побачити загальну інформацію про користувачів та можливості додавання, редагування та видалення записів.



Рисунок 3.12 – Сторінка користувачів.

Сторінка оновлення даних користувачів. Також у відповідних блоках можна змінювати електронну пошту та пароль користувача.

PharmaFlow

Оновити дані користувача

ID:

Ім'я:

Прізвище:

Змінити електронну пошту користувача

Електронна пошта:

Змінити пароль користувача

Пароль:

Підтвердьте пароль:

Рисунок 3.13 – Сторінка оновлення даних користувачів.

Сторінка створення акаунту користувача з полями особистої інформації, електронної пошти та паролю.

admin@mail.com PharmaFlow

Створити користувача

Ім'я:

Прізвище:

Роль:

Електронна пошта:

Пароль:

Підтвердьте пароль:

Вийти
<https://localhost:57141/Admin/User/GetUserList>

Рисунок 3.14 – Сторінка створення акаунту користувача.

Автоматизація постійно повторювальних операцій, а також їх структурування є дуже зручним для фарм-представника. Увесь цей функціонал а також єдиний ресурс даних, як до інформації клієнта так і до загальної інформації про продукт та його кількість, дозволяє робітнику стати більш ефективним та автономним, бо більше не треба бігати від одного відділу до іншого в пошуку та структурування необхідної інформації для вирішення повсякденних задач. Також фарм-представник має змогу відстежувати та створювати зустрічі з клієнтами, а також відстежувати свою ефективність в плані продаж, це надає змогу аналізувати та планувати подальші дії до можливих покращень в своїй роботі. Також до кожної зустрічі також зберігається адреса та відповідна помітка на мапі, для зручності й орієнтування фарм-представника, якому, в свою чергу, буде набагато легше планувати свій маршрут.

Адміністратор має змогу видачі доступу до ресурсу та контролювати акаунти користувачів, їх інформацію. Менеджер, в свою чергу, має можливість менеджменту продукції та звітів про продаж фарм-представників з ідентифікаторами та адресами клієнтів.

Висновки за розділом 3.

Виконано аналіз реалізованого веб-додатку для вирішення задачі автоматизації бізнес-процесів фармацевтичної компанії.

Веб-додаток надає зручний інструмент, що має наступні переваги:

- Структурування багатьох бізнес процесів в одній комп'ютерній моделі.
- Єдина база даних дозволяє швидко отримати потрібну інформацію.
- Підвищення ефективності та автономності працівника як робочої одиниці.

- Безпека та розподіл доступу до ресурсу. Також користувач не зможе увійти у систему самотужки, доки для нього не виділять відповідний акант адміністратори.
- Аналітичний інструмент у вигляді графіку планів продаж.
- Мінімалістичний та інтуїтивно зрозумілий інтерфейс користувача, що дозволяє досить швидко зорієнтуватися у функціях веб-додатку.

ВИСНОВКИ

Сьогодні новітні технології справляють глибокий вплив на сферу фармації. Завдяки автоматизованому рішенню обробки громіздких фармацевтичних замовлень медпредставник може не просто здійснювати пошук необхідних медпрепаратів, але й дізнаватись їх вартість, місце продажу, обирати найвигіднішу для нього цінову позицію, визначати препарати-аналоги, а також навіть сумісність декількох препаратів між собою або їх взаємодію з іншими лікарськими засобами.

В даній роботі виконано:

- Аналіз сучасних онлайн-інструментів медичних представників у сфері фармацевтичних послуг, сучасних методологій розробки комп'ютерних систем, автоматизування бізнес-процесів фармацевтичної компанії та методів розробки веб-додатків.
- Спроектований та розроблений веб-додаток, спроектована його загальна архітектура. Також при розробці були використані різноманітні шаблони та підходи реалізації функціоналу.
- Розглянуті та протестовані основні сторінки веб-додатку. Описані можливості для користувача з різними правами доступу.

Також до основних переваг розробленого веб-додатку можна віднести:

- **Простота використання.** Веб-інтерфейс мінімалістичний та інтуїтивно зрозумілий користувачу.
- **Економія часу.** Робітники можуть швидко освоїти програму для початку роботи.
- **Невисока вартість.** Такий тип додатку має відносно невелику вартість розробки та значно нижчу вартість використання ніж великі CRM.

Перечислені переваги досить затребувані на ринку серед малого та середнього бізнесу.

Впровадження ІТ в фармацію є необхідним кроком у сучасному світі, де технології впливають на всі сфери нашого життя. ІТ-рішення в медицині пропонують низку переваг, які полегшують доступ до медичних послуг, покращують діагностику, сприяють ефективному лікуванню та підвищують якість надання медичної допомоги.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Як українські фармкомпанії розвиваються під час війни. [Електронний ресурс]. –режим доступу: URL: <https://biz.nv.ua/ukr/markets/farmaceutika-yak-ukrajinski-kompaniji-rozvivayut-virobnictvo-i-tehnologiji-pid-chas-viyni-50284810.html> (дата звернення – 30 лист. 2022р).
2. Медицина під час війни: з якими викликами стикнулася медична система. Негативні наслідки та позитивні зміни. [Електронний ресурс]. – режим доступу: URL <https://mind.ua/openmind/20255116-medicina-pid-chas-vijni-z-yakimi-viklikami-stiknulasya-medichna-sistema> (дата звернення – 31 березня 2023).
3. Фармацевтична енциклопедія JavaScript [Електронний ресурс]. – режим доступу: URL: <https://www.pharmencyclopedia.com.ua/article/1366/medichniy> (дата звернення - 16.02.2023).
4. 10 міфів про роботу медичного представника. [Електронний ресурс]. – режим доступу: URL: <https://www.apteka.ua/article/512089> (дата звернення – 11.12.2022).
5. Чому ZENNX? [Електронний ресурс]. – режим доступу: URL: <http://zennx.in/> (дата звернення – 11 квіт. 2023 р.).
6. Що таке SaaS сервіс простими словами. [Електронний ресурс]. – режим доступу: URL: <https://wezom.com.ua/ua/blog/saas-prilozheniya> (дата звернення – 02 трав. 2023 р.).
7. Додаток Diagnosia [Електронний ресурс]. – режим доступу: URL: <https://play.google.com/store/apps/details?id=com.diagnosia.fachinformation&hl=uk&gl=US> (дата звернення – 11 квіт. 2023 р.).

8. у Взаємодія. MedHelp_UA_bot. Надання медичної допомоги (ліки/лікари) і транспорту в Україні. [Електронний ресурс]. – режим доступу: URL: <https://viyna.net/medichna-dopomoga>. (дата звернення – 11 лютого 2023 р.).
9. Методологія розробки програмного забезпечення: дивимось на роботу команди зсередини. [Електронний ресурс]. – режим доступу: URL: <https://wezom.com.ua/ua/blog/metodologija-razrabotki-programmnogo-obespechenija> (дата звернення – 15 квітня. 2023).
10. Ітеративна модель розробки. 5 уроків, які ми засвоїли. [Електронний ресурс]. – режим доступу: URL: <https://dou.ua/lenta/articles/iterative-development-model/> (дата звернення – 15 квітня. 2023).
11. Agile – гнучка методологія розробки програмного забезпечення. [Електронний ресурс]. – режим доступу: URL: <https://wezom.com.ua/ua/blog/agile-gibkaya-metodologiya-razrabotki> (дата звернення – 16 квітня. 2023).
12. Спіральна модель Боема (Boehm's Spiral Model). [Електронний ресурс]. – режим доступу: URL: <https://www.maxzosim.com/boehms-spiral-model/> (дата звернення – 16 квітня. 2023).
13. HTML документація. [Електронний ресурс]. – режим доступу: URL: <https://developer.mozilla.org/ru/docs/Web/HTML> (дата звернення – 17 квітня. 2023).
14. CSS документація. [Електронний ресурс]. – режим доступу: URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення – 17 квітня. 2023).
15. JS документація. [Електронний ресурс]. – режим доступу: URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення – 17 квітня. 2023).

16. Протоколи передавання даних: їх типи та особливості. [Електронний ресурс]. – режим доступу: URL: <https://highload.today/uk/protokoli-peredavannya-danih-yih-tipi-ta-osoblivosti/> (дата звернення – 17 квітня. 2023).
17. СУБД: які бувають, як вибрати. [Електронний ресурс]. – режим доступу: URL: <https://highload.today/uk/subd-yaki-buvayut-yak-vibrati/> (дата звернення – 18 квітня. 2023).
18. Using a Three-Tier Architecture Model. [Електронний ресурс]. – режим доступу: URL: <https://learn.microsoft.com/en-us/windows/win32/cos-sdk/using-a-three-tier-architecture-model> (дата звернення – 20 квітня. 2023).
19. Офіційна документація розробників Entity Framework Core. [Електронний ресурс]. – режим доступу: URL: <https://learn.microsoft.com/en-us/ef/core/> (дата звернення – 20 квітня. 2023).
20. Навіщо використовують DTO. Приклади в Java-застосунках. [Електронний ресурс]. – режим доступу: URL: <https://dou.ua/forums/topic/34411/> (дата звернення – 25 квітня. 2023).
21. Офіційна документація ASP.NET MVC 5. [Електронний ресурс]. – режим доступу: URL: <https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started> (дата звернення – 27 квітня. 2023).

ДОДАТКИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) Бакалавр
Галузь знань: 12 – Інформаційні технології
Спеціальність: 123 – Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри теоретичної
та прикладної системотехніки
д.т.н., проф. Шматков С. І.



«17» листопада 2022 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Чередника Олександра Вікторовича

1. Тема роботи «Комп'ютерна система автоматизації бізнес-процесів фармацевтичної компанії»
керівник роботи Мороз Ольга Юріївна, старший викладач
затверджені наказом по університету від «23» травня 2023 року № 4101-5/895
2. Строк подання студентом роботи 26 травня 2023
3. Перелік питань, які потрібно розробити
 1. Аналіз предметної області задля вивчення ключових моментів у фармацевтичній компанії.
 2. Постановка задачі моделювання комп'ютерної системи автоматизації бізнес-процесів фармацевтичної компанії.
 3. Розробка моделі комп'ютерної системи автоматизації бізнес-процесів фармацевтичної компанії.
 4. Управління базою даних клієнтів(створення єдиної бази лікарів, лікувально-профілактичних закладів та аптек).
 5. Керування продажами(ведення, розподіл та аналіз планів продажу, індивідуальних планів, показників продажів).
 6. Управління робочим місцем медичного представника, регіонального менеджера та менеджера з продуктів.
 7. Формулювання та організація бази знань щодо єдиного інформаційного ресурсу.
 8. Розробка рекомендації по впровадженню технології, що використовуються в роботі (Microsoft Dynamics CRM, Microsoft Windows Server, Microsoft SQL Server, Microsoft Office).
4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1.	Аналіз і підбір літератури та створення літературної бази для розробки моделі	19.10.2022–16.01.2023
2.	Аналіз сучасних прикладів комп'ютерних систем автоматизації бізнес-процесів фармацевтичної компанії	19.10.2022–16.01.2023
3.	Аналіз предметної області задля вивчення ключових моментів роботи менеджерів у фармацевтичній компанії	19.10.2022–16.01.2023
4.	Побудова моделі сервісу та порівняння аналогів існуючих систем автоматизації бізнес-процесів фармацевтичної компанії	16.01.2023 – 20.02.2023
5.	Розробка та тестування автоматизованої онлайн-системи ключових бізнес-процесів фармацевтичної компанії	20.02.2023–24.04.2023
6.	Розробка технічного завдання на розроблену систему	20.02.2023–24.04.2023
7.	Розробка програми і методики тестування розробленої	20.02.2023–24.04.2023
8.	Розробка інструкцій для користувача	20.02.2023–24.04.2023
9.	Оформлення пояснювальної записки	24.04.2023 – 25.05.2023
10.	Підготовка доповіді на тему кваліфікаційної роботи на науково-технічну конференцію	24.04.2023 – 25.05.2023
11.	Оформлення звіту за результатами переддипломної практики	24.04.2023 – 25.05.2023
12.	Передзахист кваліфікаційної роботи	Травень 2023
13.	Представлення кваліфікаційної роботи керівнику та рецензенту	Травень 2023

5. Дата видачі завдання 19 жовтня 2022

Студент

Чередник О. В.

ініціали, прізвище



підпис

Керівник роботи

Мороз О.Ю.

ініціали, прізвище



підпис

Додаток Б

Затверджую

« » 2023 р.

**Технічне завдання
на розробку програмного виробу «Комп'ютерна система
автоматизації бізнес-процесів фармацевтичної компанії»**

1.	Введення	1.1. Назва: Комп'ютерна система автоматизації бізнес-процесів фармацевтичної компанії 1.2. Галузь застосування: Інформаційні технології
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп'ютерна інженерія 2.2. Завдання на кваліфікаційну роботу бакалавра № <u> </u> 4101-5/895 <u> </u> від «23» <u> </u> 05 <u> </u> 2023 (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3.	Призначення розробки	3.1. Мета розробки: спрощення та оптимізація процесів роботи з клієнтами, включаючи процеси замовлення, доставки та обробки замовлень, взаємодії зі споживачами та партнерами. 3.2. Призначення розробки надає можливість полегшити та прискорити роботу фармацевтичної компанії, що в свою чергу допоможе збільшити ефективність її діяльності. 3.3. Вихідні дані розробки: CRM-система для оптимізації бізнес-процесів фармацевтичної компанії, а саме адміністрування, тайм-менеджмент, звітність продажів; вхідні дані: інформація про користувачів, клієнтів та їх компанії.
4.	Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: можливість адміністрування, ведення облікових записів користувачів, планування зустрічей представників фармацевтичних компаній та слідкування за звітностями. 4.2. Вимоги до надійності: забезпечення безперебійної роботи програмного виробу при будь-яких вимогах користувача в рамках призначення виробу . 4.3.Вимоги до умов експлуатації: немає 4.4. Вимоги до складу і параметрів технічних засобів: для виконання програми повинен підходити ПК із

		<p>будь-якою операційною системою сімейства Windows, Linux/Unix, Mac OS X, OS/2, Amiga. Крім того, для роботи потрібний інтерпретатор мови програмування.</p> <p>4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows 10, підтримка мови програмування, підтримка різних платформ.</p> <p>4.6. Вимоги до маркування та упаковки: вимоги до маркування та упакування не представляються.</p> <p>4.7. Вимоги до транспортування і зберігання: вимоги до транспортування та зберігання не представляються.</p> <p>4.8. Спеціальні вимоги: спеціальні вимоги до програмного виробу не пред'являються.</p>	
5.	Вимоги до програмної документації	<p>Програмою документацією до виробу «Метод аналізу інформативності змінних стану при діагностиці систем з використанням інформаційних критеріїв» вважати:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Методику розрахунку інформативності змінних стану (у вигляді глав 3.2 та 3.3 пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Опис виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи)</p>	
6.	Вимоги до техніко-економічних показників	<p>Програмою документацією до виробу «Комп'ютерна система автоматизації бізнес-процесів фармацевтичної компанії» вважати:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Опис програмного виробу (представити в Розділі 3 пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Джерела базової інформації.</p>	
7.	Стадії і етапи розробки	Дата	Назва етапу
		19.10.2022– 16.01.2023	Аналіз і підбір літератури та створення літературної бази для розробки моделі
		19.10.2022– 16.01.2023	Аналіз сучасних прикладів комп'ютерних систем автоматизації бізнес-процесів фармацевтичної компанії

	19.10.2022– 16.01.2023	Аналіз предметної області зادля вивчення ключових моментів роботи менеджерів у фармацевтичній компанії
	16.01.2023 –20.02.2023	Побудова моделі сервісу та порівняння аналогів існуючих систем автоматизації бізнес-процесів фармацевтичної компанії
	20.02.2023– 24.04.2023	Розробка та тестування автоматизованої онлайн- системи ключових бізнес-процесів фармацевтичної компанії
	20.02.2023– 24.04.2023	Розробка технічного завдання на розроблену систему
	20.02.2023– 24.04.2023	Розробка програми і методики тестування розробленої
	20.02.2023– 24.04.2023	Розробка інструкцій для користувача
	24.04.2023 –25.05.2023	Оформлення пояснювальної записки
	24.04.2023 –25.05.2023	Підготовка доповіді на тему кваліфікаційної роботи на науково-технічну конференцію
	24.04.2023 –25.05.2023	Оформлення звіту за результатами переддипломної практики
	Травень 2023	Передзахист кваліфікаційної роботи
	Травень 2023	Представлення

		кваліфікаційної роботи керівнику та рецензенту
8.	Порядок контролю і приймання програмного продукту (моделі)	<ol style="list-style-type: none"> 1. Перевірку ходу розробки програми виконувати раз в 3 тижні. 2. Захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати на паперових носіях в 1 примірнику і в електронному вигляді в 1 примірнику на CD-R компакт-диску.

Виконавець
студент групи КІ- 41
Чередник О. В.



Замовник
старший викладач
Мороз О. Ю.



Додаток В

Програма і методика випробувань програмного виробу

«Комп'ютерна система автоматизації бізнес-процесів фармацевтичної компанії»

1. Об'єкт випробувань

- 2 Назва програмного виробу : «Комп'ютерна система автоматизації бізнес-процесів фармацевтичної компанії»
- 3 Галузь застосування : Інформаційні технології
- 4 Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

2. Мета випробувань

Перевірка відповідності функціональності програмної реалізації системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

3. Загальні положення

1. Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

2. Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

4. Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступним вимогам:

- 4.1. Вимоги до функціональних характеристик: можливість адміністрування, ведення облікових записів користувачів, планування зустрічей представників

фармацевтичних компаній та слідкування за звітностями.
 4.2. Вимоги до надійності: забезпечення безперебійної роботи програмного виробу при будь-яких вимогах користувача в рамках призначення виробу .

4.3. Вимоги до умов експлуатації: немає

4.4. Вимоги до складу і параметрів технічних засобів: для виконання програми повинен підходити ПК із будь-якою операційною системою сімейства Windows, Спеціальні вимоги (не пред'являються). Linux/Unix, Mac OS X, OS/2, Amiga. Крім того, для роботи потрібний інтерпретатор мови програмування.

4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows 10, підтримка мови програмування, підтримка різних платформ.

4.6. Вимоги до маркування та упаковки: вимоги до маркування та упаковки не представляються.

4.7. Вимоги до транспортування і зберігання: вимоги до транспортування та зберігання не представляються.

4.8. Спеціальні вимоги: спеціальні вимоги до програмного виробу не пред'являються.

5. Вимоги до програмної документації

Документацією до виробу «Комп'ютерна система планування ІТ-проектів» вважати:

1) Документація по мові програмування та додаткові мануали.

2) Програму і методичку випробувань розробленої програми (представити як Додаток В до пояснювальної записки до кваліфікаційної роботи).

3) Опис програмного виробу (представити в Розділі 3 пояснювальної записки до кваліфікаційної роботи).

4) Джерела базової інформації.

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Засоби випробувань представлено на ПК на яких встановлено наступні програмні засоби: інтерпретатор мови програмування.

6.2 Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

- ознайомчий (1-й етап);
- власне випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

- 1) Перевірку комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.
- 2) Перевірку якості програмної документації. Перевірку здійснювати за критерієм відповідності вимогам ГОСТ 19.301-79 ЕСПД. «Програма і методика випробувань».

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

- 1) Перевірку відповідності технічних характеристик програми вимогам технічного завдання.
- 2) Перевірку ступеня виконання функціональних вимог до програми.
- 3) Методику проведення перевірок:
 - a) Запустити програмне забезпечення.
 - b) Порядок проведення випробувань:
 - Зробити налаштування.
 - Перевірити чи працює програма.
 - Перевірити чи формується звіт.
- 4) Якщо перевірки на першому та другому етапах виконано успішно, то виріб вважається таким, що пройшов випробування.

Для проведення випробувань пропонується тест 1, тест 2 та тест 3.

Тест 1

1. Перевірка виконання програми;
2. Конфігурація сервісу адміністрування та налаштування програмного інтерфейсу;
3. Отримання відповіді серверу про успішне створення об'єкту.

```

namespace PharmaFlow.AdministrationService;

0 references | 0 changes | 0 authors, 0 changes
internal class Program
{
    0 references | 0 changes | 0 authors, 0 changes
    private static async Task Main(string[] args)
    {
        await Host.CreateDefaultBuilder(args)
            .ConfigureWebHostDefaults(webBuilder => webBuilder.UseStartup<Startup>()).Build().RunAsync();
    }
}

2 references | 0 changes | 0 authors, 0 changes
public class Startup
{
    0 references | 0 changes | 0 authors, 0 changes
    public Startup(IConfiguration configuration)
    {
        Configuration = configuration;
    }

    5 references | 0 changes | 0 authors, 0 changes
    public IConfiguration Configuration { get; }

    0 references | 0 changes | 0 authors, 0 changes
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddControllers();

        services.AddAuthentication(o =>
        {
            o.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;
            o.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;
        })
        .AddJwtBearer(cfg =>
        {
            cfg.RequireHttpsMetadata = false;
            cfg.Authority = Configuration["KeycloakAuthentication:Authority"];
            cfg.Audience = Configuration["KeycloakAuthentication:ClientId"];

            cfg.TokenValidationParameters = new TokenValidationParameters()
            {
                ValidateAudience = true,
                ValidateIssuerSigningKey = true,
                ValidateIssuer = true,
                ValidIssuer = Configuration["KeycloakAuthentication:Issuer"],
                ValidateLifetime = true,
            };
        });
    }
}

```

```

services.AddEndpointsApiExplorer();
services.AddSwaggerGen();

services.AddCodeFirstGrpc(config =>
{
    config.ResponseCompressionLevel = System.IO.Compression.CompressionLevel.Optimal;
});

services.AddDbContext<PharmacyAccessDbContext>(options =>
    options.UseSqlServer(Configuration.GetConnectionString("SqlConnection")));

services.AddScoped<IPharmacyServiceGrpc, PharmacyServiceGrpc>();
}

0 references | 0 changes | 0 authors, 0 changes
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    using (var serviceScope = app.ApplicationServices.GetService<IServiceScopeFactory>().CreateScope())
    {
        var context = serviceScope.ServiceProvider.GetRequiredService<PharmacyAccessDbContext>();
        context.Database.Migrate();
    }

    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseSwagger();
        app.UseSwaggerUI();

        app.UseCors(policy => policy
            .AllowAnyHeader()
            .AllowAnyMethod()
            .AllowAnyOrigin());
    }

    app.UseHttpsRedirection();

    app.UseRouting();

    app.UseAuthentication();
    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllers();
        endpoints.MapGrpcService<IPharmacyServiceGrpc>();
    });
}

```

Рис. В.1 Тест 1

Тест 2

1. Перевірка виконання програми
2. Файл зберігання деяких програмних налаштувань;
3. Отримання результатів.

```

{
  "ConnectionStrings": {
    "SqlConnection": "Server=pharmaflow.database.windows.net;Database=PharmaFlow;User Id=alexandr.cherednyk;Password=pa33word!"
  },
  "KeycloakAuthentication": {
    "ServerAddress": "https://pharmaflowkeycloak.azurewebsites.net",
    "Realm": "pharmaflow",
    "Authority": "https://pharmaflowkeycloak.azurewebsites.net/auth/realms/PharmaFlow",
    "Issuer": "https://pharmaflowkeycloak.azurewebsites.net/auth/realms/PharmaFlow",
    "ClientId": "pharmaflow_app",
    "ClientSecret": "secret-get-it-in-keycloakConsole-client-credentials"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "Kestrel": {
    "Endpoints": {
      "Grpc": {
        "Url": "http://*:8888",
        "Protocols": "Http2"
      },
      "webApi": {
        "Protocols": "Http1",
        "Url": "http://*:443"
      }
    }
  }
}

```

Рис. В.2 Тест 2

Тест 3

1. Перевірка виконання програми
2. Реалізація методу програмного інтерфейсу на основі Grpc;
3. Отримання результату.

```

4 references | 0 changes | 0 authors, 0 changes
internal class PharmacyServiceGrpc : IPharmacyServiceGrpc
{
    private readonly ILogger<PharmacyServiceGrpc> _logger;
    private readonly PharmacyAccessDbContext _db;

    0 references | 0 changes | 0 authors, 0 changes
    public PharmacyServiceGrpc(
        ILogger<PharmacyServiceGrpc> logger,
        PharmacyAccessDbContext db)
    {
        _logger = logger;
        _db = db;
    }

    2 references | 0 changes | 0 authors, 0 changes
    public async Task<PharmacyKeyGrpc> CreatePharmacyAsync(PharmacyPayloadGrpc requestGrpc, CallContext context = default)
    {

```

```

2 references to 'ID changes' | 0 actions | 0 changes
public async Task<PharmacyKeyGrpc> CreatePharmacyAsync(PharmacyPayloadGrpc requestGrpc, CallContext context = default)
{
    if (string.IsNullOrEmpty(requestGrpc.Name))
    {
        throw new RpcException(new Status(StatusCode.InvalidArgument, "Invalid arguments..."), "Invalid arguments...");
    }

    try
    {
        PharmacyPersistence pharmacy = new()
        {
            Name = requestGrpc.Name,
        };

        _db.Pharmacies.Add(pharmacy);
        await _db.SaveChangesAsync();

        return new PharmacyKeyGrpc
        {
            ID = pharmacy.ID,
        };
    }
    catch (OperationCanceledException)
    {
        throw new RpcException(new Status(StatusCode.Cancelled, "Operation was cancelled."), "Operation was cancelled.");
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, "Creating pharmacy error.");
        throw new RpcException(new Status(StatusCode.Unknown, "Unknown error."), "Unknown error.");
    }
}

```

Рис. В.3 Тест 3

Тест вважається пройденим, якщо відбуваються вказані операції і їх відображення у програмному продукті.

Висновки: тест 1 успішно пройшов випробування, тест 2 успішно пройшов випробування і тест 3 успішно пройшов випробування. Випробування пройшло успішно.

Виконавець: студент групи КІ-41, Чередник О. В.