

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ЦИФРОВОЇ ТРАНСФОРМАЦІЇ БІЗНЕС-ПРОЦЕСІВ КОМПАНІЇ
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS
FOR DIGITAL TRANSFORMATION OF COMPANY BUSINESS PROCESSES

Виконав: студент 2 курсу, групи КНТ-112м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

УДОВИЧЕНКО А. А.

(ПРІЗВИЩЕ та ініціали)

Керівник ПАРХОМЕНКО А. В.

(ПРІЗВИЩЕ та ініціали)

Рецензент ЗЕЛЕНЬОВА І. Я.

(ПРІЗВИЩЕ та ініціали)

ЗАТВЕРДЖУЮ

ЗАВДАННЯ

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

УДОВИЧЕНКА Артема Артуровича

1. Тема проекту (роботи): Дослідження та програмна реалізація методів цифрової трансформації бізнес-процесів компанії. Research and Software Implementation of Methods for Digital Transformation of Company Business Processes

керівник проекту (роботи): к.т.н., доц. ПАРХОМЕНКО Анжеліка Володимирівна

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

2. Строк подання студентом проєкту (роботи): 1 грудня 2023 року

3. Вихідні дані до проєкту (роботи): завдання, література

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): 1. Аналіз методів та засобів цифрової трансформації бізнес-процесів. 2. Дослідження моделей зрілості цифрової трансформації бізнес-процесів. 3. Проєктування програмного забезпечення. 4. Конструювання програмного забезпечення. 5. Тестування та дослідна експлуатація системи. 5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):

Слайди презентації

6. Консультанти розділів проєкту (роботи):

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	ПАРХОМЕНКО А.В., доцент		
Нормоконтроль	БЄЛОВА А.В., асистент		

7. Дата видачі завдання «08» вересня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання
2	Аналіз предметної області	2-3 тиждень	Розділ 1
3	Дослідження методів вирішення проблеми	4-5 тиждень	Розділ 2
4	Проектування програмного забезпечення	6 тиждень	Розділ 3
5	Конструювання програмного забезпечення	7-9 тижні	Розділ 4
6	Тестування та експериментальне дослідження програмного забезпечення.	10 тиждень	Розділ 5
7	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	11 тиждень	ПЗ, Додатки
8	Захист роботи	12 тиждень	

Студент(ка)

_____ Артем УДОВИЧЕНКО
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Анжеліка ПАРХОМЕНКО
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
106 с., 13 табл., 41 рис., 2 додатки, 45 джерел.

БІЗНЕС-ПРОЦЕС, ЦИФРОВА ТРАНСФОРМАЦІЯ, МОДЕЛЬ
ЗРІЛОСТІ, ВІДДАЛЕНИЙ МОНІТОРИНГ, АУТЕНТИФІКАЦІЯ, БАЗА
ДАНИХ.

Об'єкт дослідження – цифрова трансформація бізнес-процесів компанії.

Предмет дослідження – методи та засоби цифрової трансформації бізнес-процесів компанії.

Мета роботи – дослідження та програмна реалізація методів цифрової трансформації бізнес-процесу моніторингу діяльності фахівців компанії для підвищення оперативності прийняття управлінських рішень в режимі онлайн.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, середовище програмування MS Visual Studio Code, мова програмування Golang, платформа Docker, персональний IBM-сумісний комп'ютер з оперативною пам'яттю 16 Гб, монітором, клавіатурою та маніпулятором типу «миша», що функціонує під управлінням операційної системи Ubuntu.

Результати. Створено програмний застосунок, який надає керівнику компанії онлайн платформу для віддаленого моніторингу та контролю виконання бізнес-задач фахівцями компанії.

Висновки. Впровадження розробленої програми дозволить спростити та пришвидшити процес прийняття управлінських рішень керівниками компанії та оперативно реагувати на результати бізнес-діяльності фахівців через браузер на смартфоні або комп'ютері.

Галузь використання – бізнес-діяльність в різних сферах.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 106 pages, 13 tables, 41 figures, 2 appendixes, 45 sources.

BUSINESS PROCESS, DIGITAL TRANSFORMATION, MATURITY MODEL, REMOTE MONITORING, AUTHENTICATION, DATA BASE.

The object of research is digital transformation of the company's business processes.

The subject of research is methods and tools of digital transformation of the company's business processes.

The purpose of the work is research and software implementation of methods of digital transformation of the business process of monitoring the activities of the company's specialists to increase the efficiency of making management decisions online.

Materials, methods and technical tools: object-oriented programming, integrated development environment MS Visual Studio Code, Golang programming language, Docker platform, personal IBM-compatible computer with 16 GB of RAM, monitor, keyboard and "mouse" type manipulator, operating under Ubuntu operating system.

Results. A software application has been created that provides the company manager with an online platform for remote monitoring and control of the execution of business tasks by the company's specialists.

Conclusions. Implementation of the developed software will simplify and speed up the process of management decision-making by the company's managers and will allow prompt response to the results of the business activities of specialists through a browser on a smartphone or computer.

The area of usage – business activities in various fields.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ.....	9
1 Аналіз методів та засобів цифрової трансформації бізнес-процесів	10
1.1 Огляд особливостей цифрової трансформації бізнес-процесів.....	10
1.2 Аналіз об'єктів та процесів цифрової трансформації	14
1.3 Вибір програмної платформи для ЦТ БП.....	16
1.4 Постановка завдань дипломної роботи.....	23
2 Дослідження моделей зрілості цифрової трансформації бізнес-процесів ...	25
2.1 Аналіз особливостей застосування моделей зрілості ЦТ	25
2.2 Дослідження моделей зрілості ЦТ	26
2.3 Методика застосування МЗЦТ для оцінки рівня ЦТ	33
2.4 Висновки до розділу 2	37
3 Проєктування програмного забезпечення	38
3.1 Призначення та вимоги до ПЗ	38
3.2 Розробка архітектури системи	39
3.3 Вибір інструментарію розробки	40
3.4 Висновки до розділу 3	48
4 Конструювання ПЗ.....	49
4.1 Розробка алгоритму функціонування програми	49
4.2 Структура основної програми	56
4.3 Опис особливостей програмної реалізації.....	57
4.4 Висновки до розділу 4	61
5 Тестування та дослідна експлуатація системи.....	62
5.1 Опис застосування	62
5.2 Інструкція з експлуатації програми	62
5.3 Висновки до розділу 5	73

Висновки	74
Перелік джерел посилань	75
Додаток А_Текст програми.....	80
Додаток Б_Слайди презентації	100

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

BANI	– Brittle, Anxious, Nonlinear, Incomprehensible;
BIM	– Building Information Modelin;
BPM	– Business Process Management;
CMMI	– Capability Maturity Model Integration;
CRM	– Customer Relationship Management;
DTMM	– Digital Transformation Maturity Models;
ERP	– Enterprise Resource Planning;
HCM	– Human Capital Management;
HRIS	– Human Resources Information System;
HRMS	– Human Resources Management System;
KPI	– Key Performance Indicators;
TOTP	– Time-based One-Time Password;
VDC	– Virtual Design and Construction;
VUCA	– Volatility, Uncertainty, Complexity, Ambiguity;
2FA	– Two-Factor Authentication;
БД	– база даних;
БП	– бізнес-процес;
ІКТ	– інформаційно-комунікаційні технології;
ІТ	– інформаційні технології;
МЗ	– модель зрілості;
ПЗ	– програмне забезпечення;
СКБД	– система керування базами даних;
ЦТ	– цифрова трансформація.

ВСТУП

Стрімкий розвиток інформаційних технологій (ІТ), а також нові тенденції на світовому ринку призвели до появи нового тренду – цифрової трансформації (ЦТ). Успішний процес ЦТ призводить до зміни існуючих бізнес-моделей та бізнес-процесів (БП), а також створення абсолютно нових шляхом впровадження та використання цифрових технологій [1].

Проекти ЦТ охоплюють дуже широкий спектр задач залежно від сфери їх реалізації. Прикладами для архітектури та будівництва є проекти BIM (Building Information Modelin) або VDC (Virtual Design and Construction). В сфері ІТ це може бути впровадження нової системи ERP (Enterprise Resource Planning) або розробка нового програмного забезпечення (ПЗ), в електронній комерції – запуск нової платформи мерчандайзингу.

Оцінка поточного рівня ЦТ організації, продукту, послуги чи бізнесу є надзвичайно важливою для опису та планування цільового рівня, а також для забезпечення якості на шляху вдосконалення до вищих рівнів автоматизації БП або розробці нової цифрової бізнес-моделі.

В умовах, коли бізнес-діяльність стає все складнішою та розподіленою по різних локаціях, важливо мати ефективну систему віддаленого моніторингу та контролю, щоб забезпечити оптимальну продуктивність та максимальну ефективність управління. Системи віддаленого моніторингу та контролю можуть стати вирішальним фактором для багатьох компаній у підтримці високого рівня виконання БП та бізнес-задач.

Існуючі програмні платформи автоматизації БП зазвичай не надають повного набору інструментів ЦТ та потребують модифікації та розширення функціоналу. Тому, тема роботи є актуальною.

Мета роботи – дослідження та програмна реалізація методів цифрової трансформації бізнес-процесу моніторингу діяльності фахівців компанії для підвищення оперативності прийняття управлінських рішень в режимі онлайн.

1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ БІЗНЕС-ПРОЦЕСІВ

1.1 Огляд особливостей цифрової трансформації бізнес-процесів

ЦТ – це використання сучасних інформаційних технологій для створення нових або модифікації існуючих БП, культури та досвіду клієнтів. Переосмислення бізнесу в епоху цифрових технологій надає організації необхідні інструменти, щоб не відставати від конкурентів і відповідати мінливим вимогам ринку [2].

ЦТ БП – це стратегічна ініціатива, яка передбачає ретельний аналіз і перегляд існуючих БП. Основною метою є підвищення операційної стійкості, продуктивності та загальної ефективності бізнес діяльності [3].

Існує чотири основні напрямки ЦТ бізнес-діяльності (рис.1.1) [2].

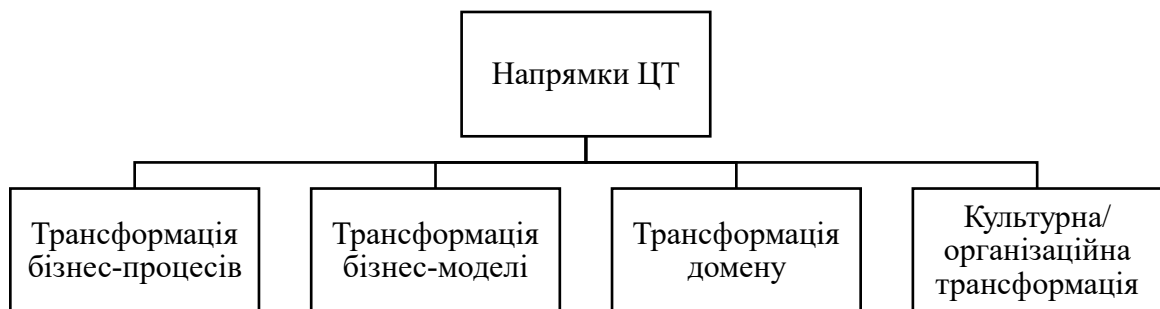


Рисунок 1.1 – Напрямки ЦТ [2]

Трансформація бізнес-процесів – це стратегічна ініціатива, спрямована на підвищення ефективності та результативності діяльності компанії, що передбачає ретельний аналіз існуючих процесів і систем з метою визначення областей для вдосконалення та внесення змін для кращого досягнення цілей компанії.

Трансформація бізнес-процесу передбачає наступні кроки:

- визначення цілей ЦТ;
- встановлення базових показників для оцінки успішності ЦТ;

- залучення всіх зацікавлених сторін;
- складання найкращого сценарію;
- тестування;
- запуск і моніторинг.

Трансформація бізнес-моделі передбачає переосмислення традиційних бізнес-моделей та внесення значних змін в бізнес-практики.

Трансформація домену дозволяє переосмислювати продукти та послуги, розмивати межі галузі та створювати можливості для нетрадиційних конкурентів та нові цінності.

Культурна/організаційна ЦТ дозволяє узгодити свої зусилля з ЦТ зі внутрішніми цінностями та поведінкою компанії, що може вплинути на культуру організації з точки зору розуміння, сприйняття та просування ЦТ [2].

Програми ЦТ БП охоплюють комплексний цикл Трансформація-Оптимізація-Контроль (рис.1.2) [3]-[4].

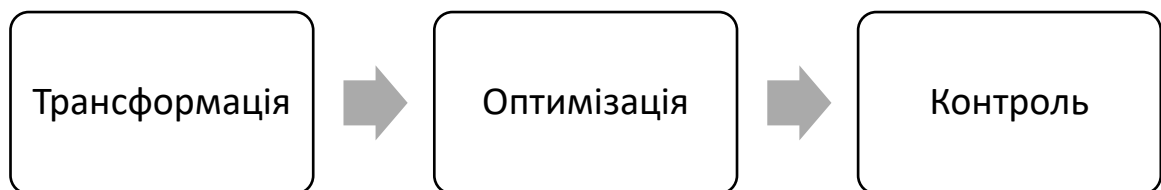


Рисунок 1.2 – Цикл ЦТ БП [3]-[4]

Трансформація стосується повного перегляду та редизайну БП для значного покращення критичних показників ефективності, таких як вартість, обслуговування, якість і швидкість. Цей процес часто передбачає переосмислення фундаментальних способів ведення бізнесу організацією та може призвести до значних змін у бізнес-стратегії.

Оптимізація – це процес найкращого або найефективнішого використання ситуації або ресурсу в БП. Цей крок передбачає доопрацювання

та вдосконалення процесів для підвищення ефективності, зменшення відходів і підвищення прибутковості.

Контроль є завершальним етапом трансформації БП, що передбачає моніторинг і коригування нових процесів, щоб гарантувати, що вони забезпечують бажані результати. Механізми контролю включають показники ефективності, аудити та системи зворотного зв'язку. Ефективний контроль допомагає підтримувати переваги, досягнуті завдяки трансформації та оптимізації, і забезпечує продовження адаптації процесів до мінливих бізнес-потреб і викликів [3].

Основні фази ЦТ БП представлено на рис.1.3 [3].

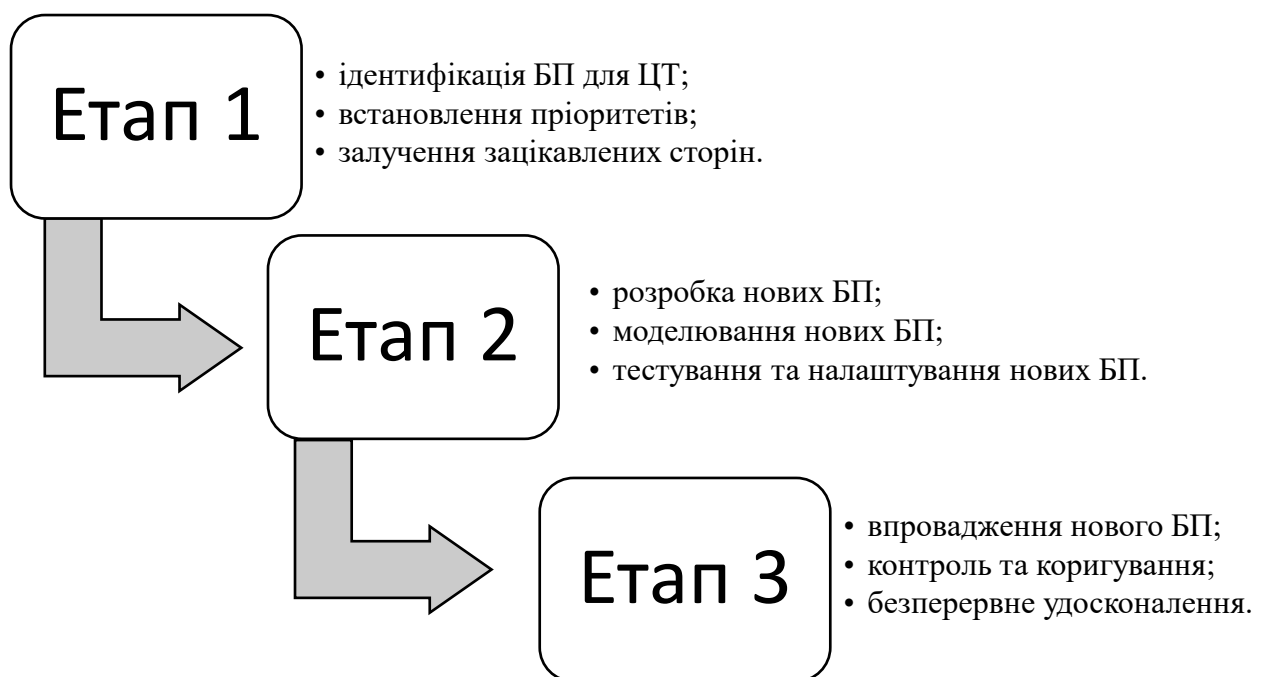


Рисунок 1.3 – Етапи ЦТ БП [3]

Етап 1. Передбачає детальний аналіз поточних бізнес-операцій з метою визначення БП для ЦТ, встановлення пріоритетів на основі потенційного впливу БП на бізнес з урахуванням таких факторів, як економія коштів, підвищення задоволеності клієнтів і підвищення ефективності роботи. Цей етап закладає основу для успішної трансформації. Визначивши правильні

процеси та залучивши зацікавлених сторін на ранній стадії, компанії можуть гарантувати, що їхні зусилля з трансформації матимуть найбільший ефект [3].

Етап 2. Спрямований на створення ефективних процесів, які відповідають потребам бізнесу та його клієнтів. Компанії можуть забезпечити успішність своїх перетворень шляхом проєктування, моделювання та тестування нових процесів. Розробка нових БП передбачає використання інформації, зібраної на етапі ідентифікації та створення нових процесів, що будуть ефективними та здійсненими. Моделювання нових БП дозволяє візуально представити нові процеси, щоб зрозуміти, як саме вони працюватимуть та чим відрізнятимуться від старих. Тестування та налаштування дає змогу перевірити нові процеси в ході пілотних програм, виявити проблеми чи неефективність до повного впровадження нових БП [3].

Етап 3. Передбачає впровадження та моніторинг нових процесів, щоб переконатися, що вони дають заплановані результати. Це передбачає навчання персоналу, оновлення систем і, можливо, реорганізацію відділів або ролей.

Контроль відбувається на основі показників ефективності нових БП з подальшим внесенням коректив у разі необхідності. Постійне вдосконалення передбачає регулярну перевірку та оновлення БП, щоб вони відповідали потребам бізнесу та його клієнтів [3].

Деякі з базових показників цінності ЦТ БП включають:

- вдосконалення досвіду та задоволення клієнтів;
- збільшення фінансових показників;
- підвищення загальної операційної ефективності організації;
- підвищення продуктивності та залучення працівників;
- нові потенційні клієнти [5].

Таким чином, ЦТ – це інвестиція, яка дозволяє заощадити гроші в довгостроковій перспективі та забезпечує кращі можливості для прийняття прибуткових бізнес-рішень. ЦТ надає підприємствам та компаніям багато переваг, зокрема:

- підвищена ефективність завдяки покращеним зв'язкам і оптимізації робочих процесів;
- зниження операційних витрат завдяки цифровому моніторингу та аналітиці, що дозволяє виявляти проблеми, такі як вузькі місця та непотрібні затримки;
- додана цінність для клієнтів за рахунок більш ефективних та індивідуальних послуг;
- краще використання часу персоналу, оскільки автоматизація завдань дозволяє співробітникам витрачати час на більш значущу та складну роботу ніж на рутинні прості операції;
- більше корисних даних завдяки використанню нових технологій, що дає змогу ефективно приймати бізнес-рішення [6].

Однією з важливих сфер ЦТ є перетворення БП шляхом впровадження цифрових операцій, що надають можливість переглядати, керувати та відстежувати всю роботу, використовуючи одну систему. Цей підхід надає наступні переваги:

- видимість – наявність повної картини допомагає виявити слабкі місця та впливати на впливові зміни;
- одна проста платформа, що дозволяє керувати та організувати всю роботу в одній інтуїтивно зрозумілій системі, яка дає змогу впорядкувати процеси та вдосконалювати комунікацію;
- автоматизація повторюваних завдань;
- підвищення ефективності за рахунок чітких часових рамок, правильного розподілу обов'язків та організації робочих процесів [6].

1.2 Аналіз об'єктів та процесів цифрової трансформації

Успіх ЦТ залежить від типу БП, до якого застосовуються зміни (рис. 1.4) [7].

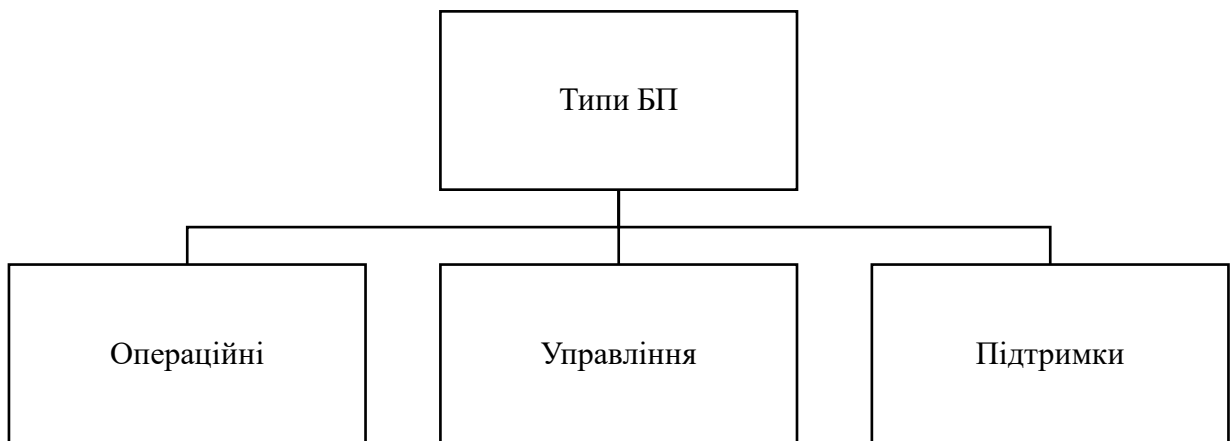


Рисунок 1.4 – Типи бізнес-процесів [7]

Операційні – це процеси, в яких низка послідовних та одночасних заходів створює нову послугу, продукт, політику чи бізнес-методологію.

В процесах управління керівництво ставить організаційні цілі, визначає та окреслює масштабні ініціативи та надає рекомендації та інструкції щодо ділової діяльності.

В процесах підтримки зазначені завдання забезпечують підтримку щоденної експлуатації організації [7].

ЦТ БП – це більше, ніж просто переміщення бізнесу в Інтернет або перетворення аналогових даних у цифровий формат. ЦТ перетинається з іншими поняттями, такими як оцифрування та цифровізація – це терміни, які часто використовуються взаємозамінно, але вони не є синонімом, і розуміння різниці є ключовим для розробки та впровадження ефективної та стійкої стратегії ЦТ. Рисунок 1.5 ілюструє процеси оцифрування, цифровізації та ЦТ та їх співвідношення [8].

Оцифрування – це процес перетворення аналогової інформації в цифрові дані, наприклад, перетворення голосу в аудіо дані або сканування паперового документа для отримання його цифрової копії. Оцифрування полегшує обмін даними та розпізнавання закономірностей у великій кількості інформації.

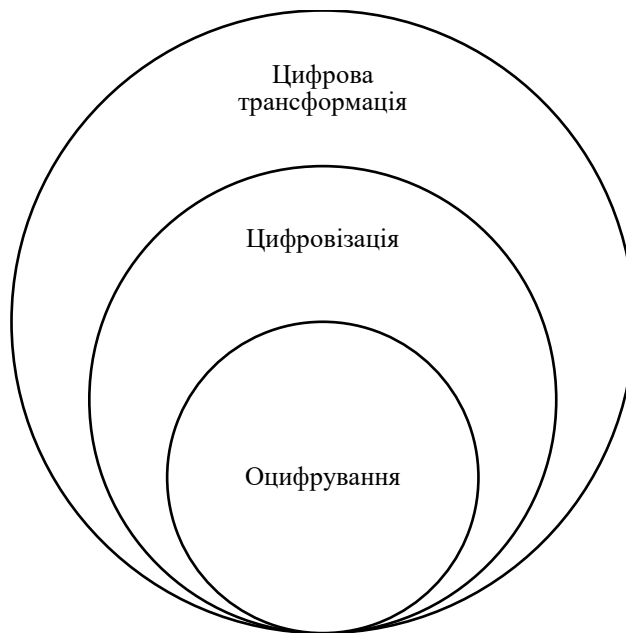


Рисунок 1.5 – Діаграма процесів ЦТ [8]

Цифровізація – автоматизація та вдосконалення (існуючого) процесу за допомогою цифрових інструментів і даних, наприклад, перехід від паперового процесу замовлення та продажу до Інтернет-магазину.

ЦТ – розробка (повністю) нових процесів, послуг, продуктів і бізнес-моделей, які були б неможливі без цифрових технологій і які замінюють існуючі. Це виходить за рамки вдосконалення існуючих процесів і вимагає набагато більш детальної концепції змін і врахування впливу. Охоплює як оцифрування так і цифровізацію [8]-[9].

1.3 Вибір програмної платформи для ЦТ БП

Вибір правильних інструментів та технологічних платформ вимагає ретельного розгляду потреб та цілей компанії, а також узгодження з існуючим технологічним ландшафтом, щоб забезпечити безшовну ЦТ. Наприклад, якщо мета – цифровізація та оптимізація БП, то варіантом вибору є система управління БП (Business Process Management, BPM). Для покращення відносин з клієнтами та продуктивності більш доцільно обрати систему управління

відносинами (Customer Relationship Management, CRM), а для кадрових операцій – систему управління людськими ресурсами (Human Resources Management System, HRMS; Human Resources Information System, HRIS; Human Capital Management, HCM) [10].

Для вирішення завдань ЦТ БП бажано обрати платформу, яка:

- централізує управління даними та забезпечує зрозумілі, повні та стандартизовані дані, отримані з різних джерел;
- забезпечує вирівняний, відповідний ролям доступ до даних у режимі реального часу як з настільних ПК, так і з мобільних пристроїв на будь-якій платформі;
- спрощує автоматизацію процесів, усуваючи як людські помилки, так і необхідність участі людини в рутинних, трудомістких та великих завданнях;
- надає доступ до потужних інструментів аналізу даних, які покращують прийняття рішень, стратегічне планування та бізнесу;
- дозволяє використовувати модульний підхід, масштабування оцифрування процесу, щоб відповідати організаційним ресурсам та загальними цілями для ЦТ;
- забезпечує ітеративне постійне вдосконалення для подальшого підвищення ефективності, швидкості та точності виконання БП [11].

Вибір ПЗ для ЦТ БП є складним завданням, що потребує детального аналізу та порівняння різних платформ з урахуванням низки критеріїв:

- відповідність задачам ЦТ – необхідно визначити, які саме БП будуть трансформовані та які функції повинно мати ПЗ для досягнення поставленої мети;
- вартість ПЗ – ціна ліцензії, підтримки та навчання користувачів для ефективного використання програмного продукту на підприємстві;

- зручність використання – ПЗ повинно бути зручним та легким у використанні, а також мати зрозумілий інтерфейс користувача та докладну документацію;
- підтримка та розвиток ПЗ, оскільки технології постійно змінюються, а потреби підприємства можуть зростати;
- сумісність з іншими програмними засобами, що використовуються на підприємстві та здатність інтегруватися з вже існуючими системами для забезпечення ефективної взаємодії між ними.

Результати дослідження популярних на українському ринку платформ автоматизації БП наведено в табл. 1.1 [12]-[17].

Таблиця 1.1 – Результати дослідження систем автоматизації БП

Назва	Призначення	Переваги	Недоліки
1	2	3	4
Vimpr [12]	Моніторинг, контроль, оптимізація та управління фінансами для малого і середнього бізнесу	Простота та зручність використання, широкі можливості інтеграції з іншими програмними продуктами	Процес налаштування та інтеграції може потребувати багато часу та зусиль недосвідчених користувачів; сервіс не є безкоштовним; не підходить для всіх типів бізнесу
Quincefin [13]	Автоматизація складського обліку, процеси контролю за переміщенням товарів, сировини або продукції.	Доступ до даних з будь-якого пристрою, підключеного до Інтернету; можливості інтеграції з іншими програмними продуктами; гнучкий налаштовуваний інтерфейс	Не є універсальним; є платним; доступ до фінансової інформації третім особам, що може бути ризикованим з точки зору конфіденційності даних.

Продовження таблиці 1.1

1	2	3	4
Finmap [14]	Автоматизація БП, що дозволяє контролювати фінансовий потік підприємства та управляти ним	Безкоштовний пробний період; зручний та інтуїтивно зрозумілий інтерфейс; можливість підключати банківські рахунки та інші джерела фінансової інформації.	Деяко обмежений функціонал у порівнянні з конкурентними продуктами; інтерфейс може бути недостатньо зручним та неінтуїтивним у деяких моментах.
Pipeliner CRM [15]	Управління продажами, як для приватного, так і для колективного використання в компаніях, де важливу роль відіграють продажі.	Безкоштовний доступ до системи з обмеженням місця зберігання даних; можливості інтеграції з іншими продуктами; доступ з мобільного телефону та ПК, збирання даних про клієнтів через сайт та Google-форми.	Висока вартість програми, складність налаштувань та використання без попередньої підготовки та навчання.
Bookkeeper [16]	Облік діяльності, складання та подання звітності підприємств і ФОП	Онлайн доступ; автоматичне імпортування операцій з банківських рахунків; можливість ведення обліку декількох юросіб; простий та зрозумілий інтерфейс.	Не є безкоштовним; має застарілий та не дуже приємний інтерфейс.
BAS [17]	Автоматизація та оптимізація БП в компаніях різних розмірів та в різних галузях.	Налаштування під конкретні потреби клієнта; високий рівень захисту даних; детальний аналіз БП для вдосконалення стратегії компанії.	Висока вартість; складність використання, що вимагає кваліфікованих фахівців; ризики безпеки; залежність від інтернет-з'єднання

Приклади інтерфейсів систем BIMP та Quincefin представлено на рис. 1.6-1.7 [12]-[13].

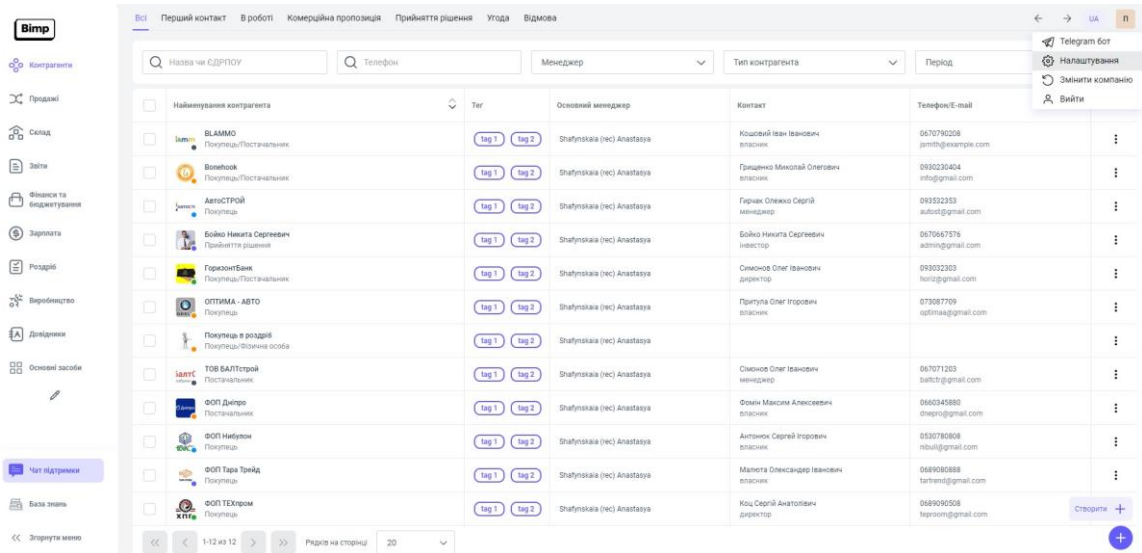


Рисунок 1.6 – Приклад інтерфейсу системи BIMP [12]

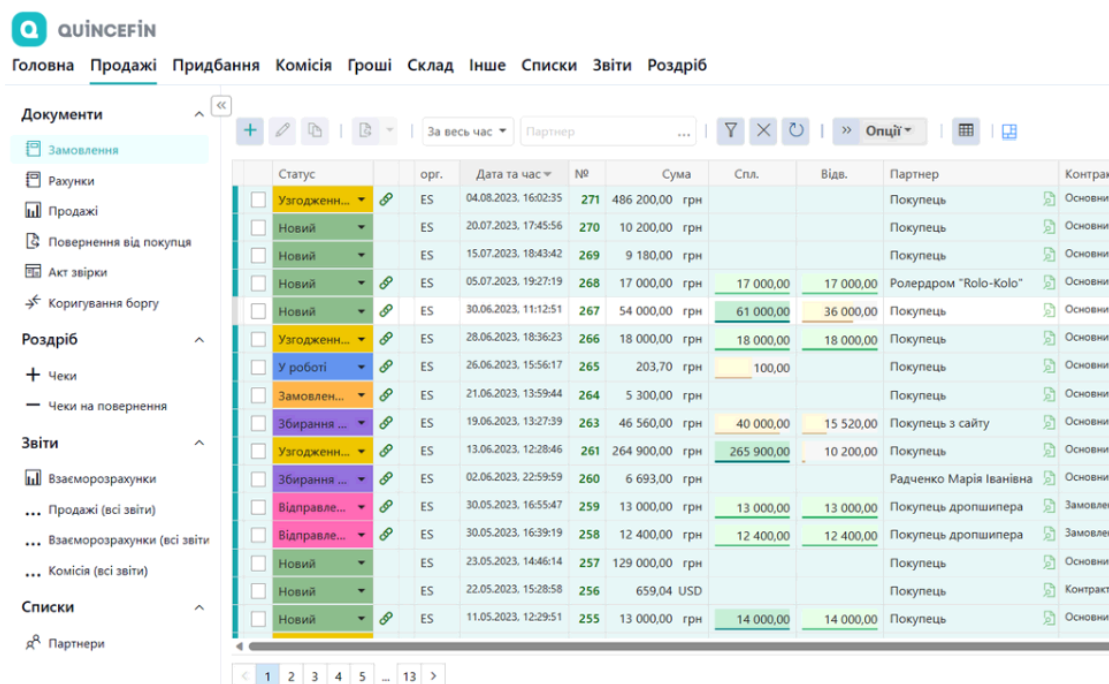


Рисунок 1.7 – Приклад інтерфейсу системи Quincefin [13]

Приклади інтерфейсів систем Finmar та Pipeliner CRM представлено на рис. 1.8-1.9 [14]-[15].

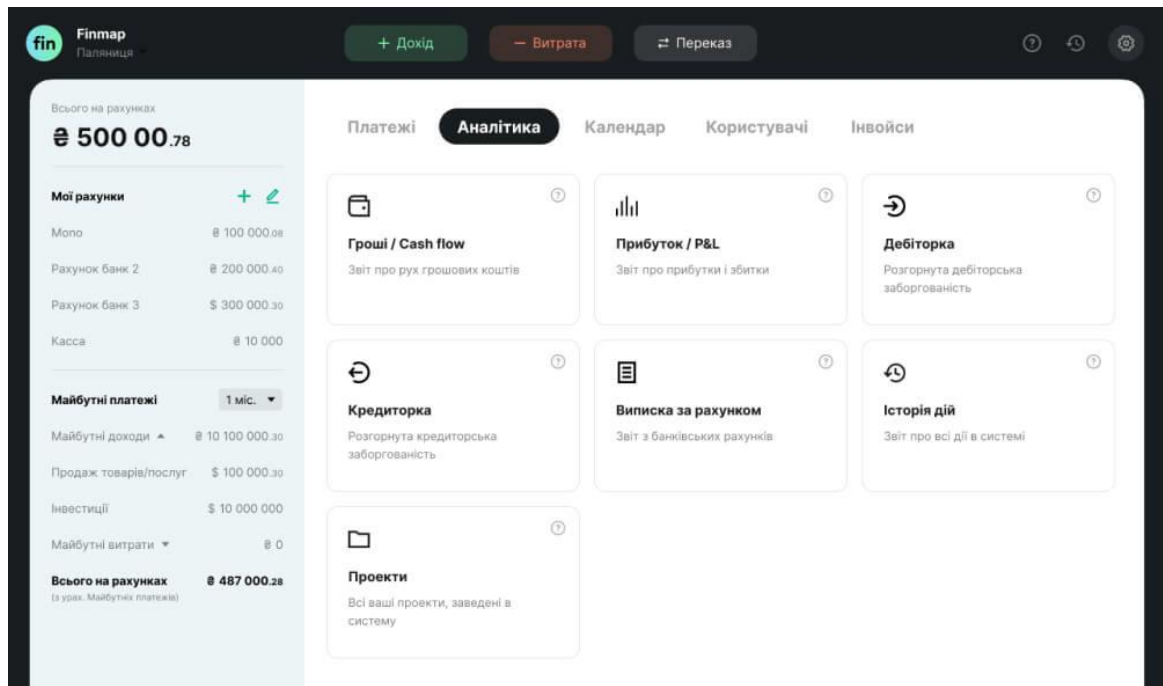


Рисунок 1.8 – Приклад інтерфейсу системи Finmap [14]

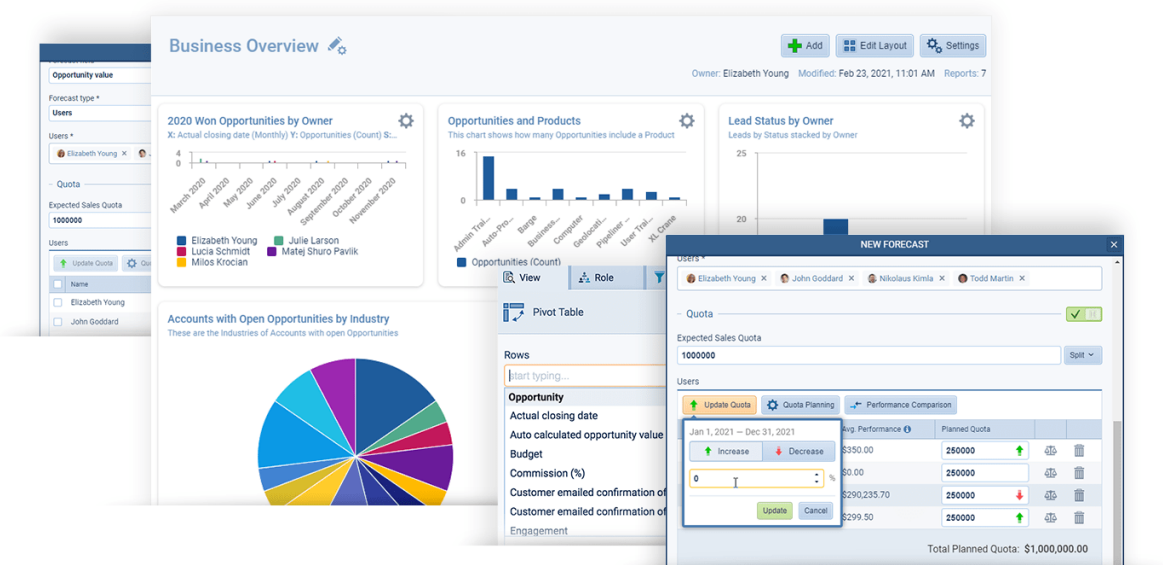


Рисунок 1.9 – Приклад інтерфейсу системи Pipeliner CRM [15]

Приклади інтерфейсів систем Bookkeeper та BAS представлено на рис. 1.10-1.11 [16]-[17].

Отже, кожна система має свій функціонал, переваги та недоліки, а також особливості організації інтерфейсу. Тому, вибір засобів ЦТ є складною задачею, що потребує урахування цілої низки критеріїв.

БукКіпер сервіс онлайн-бухгалтерії

зберегти зберегти та закрити оновити видалити провести друк/експорт

Акт введення в експлуатацію 1 від 28.11.2018

Оклад: Основний склад

Матеріально-відповідальна особа: Петренко П. Б.

+ додати рядок

№ з.п.	Назва	Кількість	Первісна вартість	Група ОЗ, НМА, НА	Паспорт	Рік виготовлення	Інвентарний номер	Термін корисного використання (місяців)	Ліквідаційна вартість (грн.)	
1	Стіл	4	500,00	Инстру...	333	2017	777	120	10	×
2	Кре	2	1 200,00	Инстру...	444	2016	666	60	20	×
3	Стул	2	600,00	Инстру...	555	2016	888	72	20	×
Разом			2 300,00							

+ додати рядок

Примітка:

про систему контакти умови використання

Рисунок 1.10 – Приклад інтерфейсу системи Bookkeeper [16]

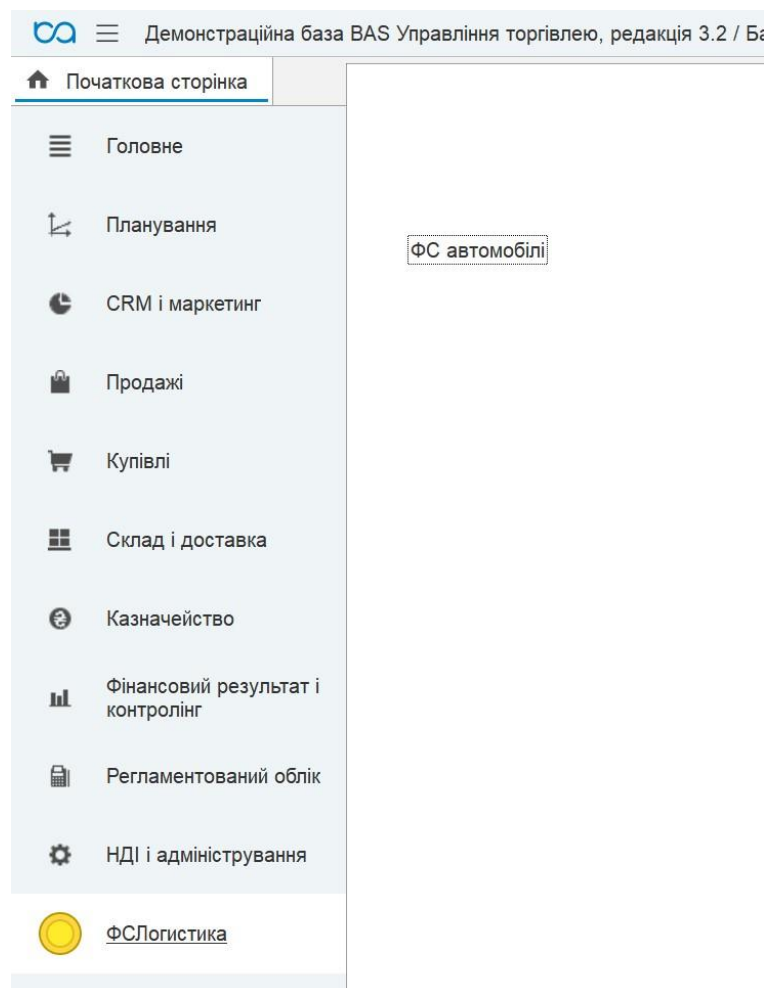


Рисунок 1.11 – Приклад інтерфейсу системи BAS ERP [17]

Окрім вказаних в таблиці 1.1 особливостей, система BAS має ще декілька важливих переваг, що зумовлюють її популярність серед українських компаній:

- гнучкість та налаштування: система може бути налаштована під конкретні потреби компанії; за допомогою різноманітних модулів та інструментів, BAS дозволяє створювати унікальні рішення для кожного клієнта;
- автоматизація бізнес-процесів: BAS дозволяє значно зменшити час та зусилля, необхідні для виконання бізнес-процесів; автоматизовані процеси BAS забезпечують швидкий та точний облік фінансів, управління складом, контроль за взаємовідносинами з клієнтами та багато іншого;
- інтеграція з іншими системами: BAS може бути легко інтегрована з іншими системами, що дозволяє клієнтам використовувати різноманітні інструменти та сервіси, не обмежуючи себе тільки цією системою;
- покращений аналіз даних: BAS дозволяє проводити детальний аналіз БП та виявляти проблемні місця у роботі компанії; за допомогою аналітичних інструментів BAS можна виявити можливості для оптимізації БП;
- безпека даних: BAS забезпечує високий рівень захисту даних клієнтів; всі дані, що зберігаються в системі, є конфіденційними та захищеними від несанкціонованого доступу;
- підтримка клієнтів: BAS має професійну службу підтримки, яка завжди готова надати клієнтам підтримку щодо використання системи [17].

1.4 Постановка завдань дипломної роботи

Як показали проведені дослідження, ЦТ БП є складною задачею, що потребує комплексного підходу з точки зору організації циклу Трансформація-Оптимізація-Контроль, врахування багатьох факторів та особливостей бізнес-діяльності підприємств та компаній.

Цифрову трансформацію БП можна здійснити за допомогою проєктів , які реалізують кроки, необхідні для переходу від існуючої «As-Is» ситуації до майбутньої «To-Be» ситуації. Основні фактори для проєктів ЦТ включають:

- мінливе, невизначене, складне, неоднозначне (VUCA, volatility, uncertainty, complexity, ambiguity), а також крихке, тривожне, нелінійне, незрозуміле (BANI, brittle, anxious, nonlinear, incomprehensible) середовище проєкту;
- поступовий та ітераційний підхід до досягнення стабільності та впливу ЦТ на бізнес-діяльність компанії.

Серйозною проблемою є відсутність універсальних підходів, моделей та програмних платформ, які відповідають всім задачам ЦТ БП. У зв'язку з цим, виникає потреба в розширенні функціональних можливостей існуючих на підприємствах систем автоматизації БП, розробці додаткових сервісів, які допоможуть здійснити ЦТ БП, оптимізувати роботу компанії та збільшити прибуток.

Огляд ринку показав, що система BAS є популярною на українському ринку, особливо для малого та середнього бізнесу. Вона забезпечує широкий набір можливостей, але не надає функцій віддаленого моніторингу виконання бізнес-задач. Тому, тема роботи є актуальною.

Мета роботи – дослідження та програмна реалізація методів ЦТ БП моніторингу діяльності фахівців компанії для підвищення оперативності прийняття управлінських рішень в режимі онлайн.

Завдання роботи:

- дослідити існуючі моделі зрілості ЦТ БП;
- виконати розробку методики оцінки рівня ЦТ БП;
- виконати проєктування та конструювання ПЗ для віддаленого моніторингу бізнес-діяльності фахівців, що використовують платформу BAS ERP.

2 ДОСЛІДЖЕННЯ МОДЕЛЕЙ ЗРІЛОСТІ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ БІЗНЕС-ПРОЦЕСІВ

2.1 Аналіз особливостей застосування моделей зрілості ЦТ

Моделі зрілості цифрової трансформації (МЗЦТ) (DTMM, Digital Transformation Maturity Models) структурують ЦТ на окремі етапи та припускають, що більш просунутий етап у процесі трансформації відповідає вищому рівню зрілості щодо ЦТ. Цей підхід може застосовуватися не для всіх сценаріїв ЦТ, оскільки він передбачає, що трансформація має чітку початкову точку низької зрілості та визначену кінцеву точку високої зрілості. Крім того, передбачається, що процес можна структурувати на етапи, які ведуть до визначеного рівня зрілості. Моделі зрілості з визначеними рівнями зрілості широко використовуються для структурування та управління трансформацією організацій, наприклад, модель (методологія) удосконалення процесів та покращення показників діяльності (CMMI, Capability Maturity Model Integration), яка визначає зрілість управління процесами [18].

Загалом, моделі зрілості можна застосовувати до багатьох процесів трансформації, де існуюча ситуація «As-Is» перетворюється на бажану майбутню ситуацію «To-Be». Модель зрілості може допомогти оцінити ситуацію «As-Is» (поточний рівень зрілості) і визначити мету трансформації (майбутній рівень зрілості). Крім того, вона надає інформацію про рівні зрілості між поточним і майбутнім рівнем зрілості. Це дає вказівки та попередньо структурований шлях трансформації для досягнення майбутнього рівня зрілості крок за кроком. Крім того, розроблені моделі зрілості забезпечують інструменти та вказівки для оцінки поточної ситуації та переходу від одного рівня зрілості до наступного [18].

Процеси ЦТ, які можуть виграти від використання моделей зрілості, включають:

- ЦТ організацій, наприклад, компаній, відділів;

- ЦТ бізнес-процесів;
- ЦТ бізнес-моделей, ринків;
- ЦТ продуктів або послуг;
- ЦТ компетенцій людей [18].

Для розуміння того, як використовувати модель зрілості для оцінки та забезпечення якості проєкту важливо наступне:

- мотивація: що таке ЦТ і чому вона важлива, які чинники впливають на ЦТ, що таке проєкт ЦТ;
- вступ до моделей зрілості ЦТ і опис простої моделі з чотирма рівнями зрілості;
- використання МЗЦТ з характеристиками кожного рівня зрілості та відповідними питаннями для оцінки рівня зрілості:
 - як оцінити існуючий стан «As-Is»;
 - як спланувати майбутній стан «To-Be»;
 - вимірювання прогресу за допомогою ключових показників ефективності (KPI, Key Performance Indicators) [18]-[19].

2.2 Дослідження моделей зрілості ЦТ

2.2.1 Класифікація моделей зрілості ЦТ

На сьогоднішній день, немає чіткої таксономії чи класифікації МЗЦТ. Багато з них розроблено для особливих випадків використання, наприклад, Industrie 4.0 Maturity Index [20]. Таким чином, конкретний процес ЦТ вимагатиме вибору та адаптації моделі зрілості, яка добре відповідає проблемі.

Приклади типових випадків використання МЗЦТ:

- опис статусу організації щодо ЦТ, визначення цільового рівня зрілості, до якого організація хоче розвиватися;
- оцінка розриву між поточним рівнем зрілості та цільовим рівнем, а також планування підвищення зрілості та створення проєктів ЦТ.

МЗЦТ були запропоновані у великій кількості для різних областей застосування та для конкретних типів трансформації [21]-[23].

Аналіз показав, що існує кілька типів моделей зрілості (МЗ) (рис. 2.1) [24]-[25].

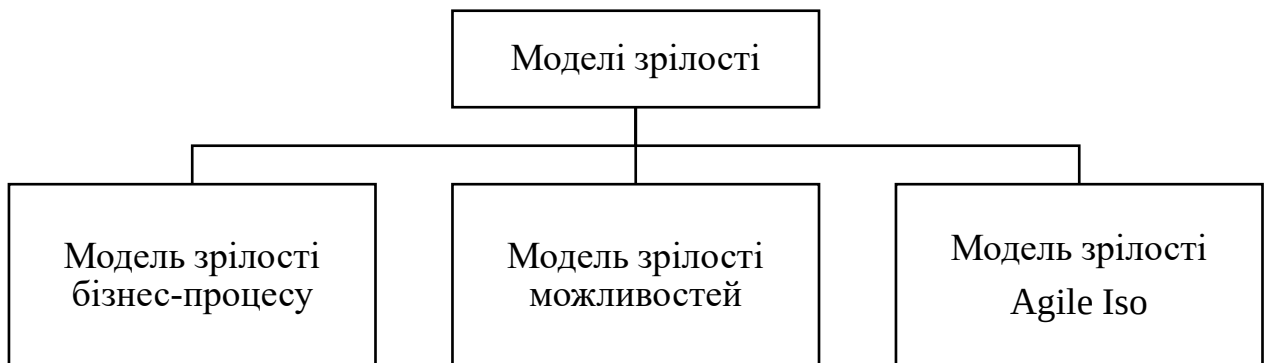


Рисунок 2.1 – Типи моделей зрілості [24]-[25]

Модель зрілості БП зазвичай використовує п'ять рівнів для оцінки зрілості організації:

- початковий: характеризує непослідовні практики управління або команди, які реагують на кризи, але не прогнозують їх;
- керований: визначає команди та підприємства, які мають основу управління, але окремі команди всередині бізнесу все ще працюють ізольовано з мінімальною співпрацею або доказами включення стратегій покращення;
- стандартизований: рівень управління процесами, означає, що бізнес знає про свої процеси та працює над узгодженістю та рівномірним наданням;
- передбачуваність: на цьому рівні організації використовують свою інфраструктуру процесів і можливості активів для досягнення надійних результатів шляхом контролю варіацій своїх результатів;
- оптимізація: на цьому рівні компанії постійно вдосконалюються та зосереджуються на інноваціях [24]-[25].

У моделі рівні ранжуються від найвищого до найнижчого, причому перший рівень потребує найбільшого вдосконалення, а п'ятий рівень є повністю реалізованою та ефективною операцією. Моделі зрілості бізнес-процесів також наголошують на доменних областях процесів, які є стратегіями, які компанії та команди можуть використовувати для досягнення наступного рівня моделі. Процеси домену можуть стосуватися управління вимогами, моніторингу та контролю робочих одиниць, стратегій управління змінами та гарантії продукту.

Модель зрілості можливостей. Спочатку розроблена для розробки ПЗ, модель зрілості можливостей оцінює зрілість організації або систем розробки ПЗ шляхом порівняння з найкращими галузевими практиками. Вимірюючи результати та призначаючи рівні зрілості, компанії та групи розробників можуть використовувати свої моделі для оцінки своєї обізнаності щодо БП, ефективних методів управління та областей для вдосконалення. Усі моделі зрілості використовують перераховані вище рівні з деякими варіаціями для опису процесів. З моделями можливостей рівні більш чітко пов'язані з процесами розробки [24]-[25].

Гнучка модель зрілості ISO. На відміну від інших моделей зрілості, модель Міжнародної організації стандартизації (ISO) працює, щоб усунути деяку критику, яку люди мають щодо моделей зрілості. Завдяки стандартизації рівнів гнучкі моделі встановлюють чіткіше визначені очікування, визначені міжнародним органом. Впровадження гнучкої методології також може принести переваги підприємствам [24]-[25].

Деякі моделі зрілості недостатньо деталізовані, щоб надати корисні оцінки. Стандартизовані гнучкі моделі працюють для боротьби з цим, зосереджуючись менше на послідовній розробці, а більше на окремих фазах життєвого циклу розробки. Багато гнучких методологій використовують спринти, які зосереджують команди розробників і розділяють проекти на більш доступні фази. Застосування ітераційного підходу до розробки дозволяє

командам більш ефективно оцінювати нюанси та визначати сфери вдосконалення [24]-[25].

Огляд існуючих моделей зрілості [22] виявив, що 9 моделей зрілості з 22 досліджених використовують 4 дискретні рівня для опису зрілості.

Тому 4-ступенева модель може бути рекомендована як спрощений загальний підхід, в якому описано 4 загальні етапи зрілості ЦТ. Автори [22] наголошують на важливості залучення ключових зацікавлених сторін, таких як керівники, менеджери проєктів та ІТ-лідери до процесу відбору для забезпечення узгодженості з цілями організації.

Структурні рівні МЗЦТ представлена в табл. 2.1 [22].

Таблиця 2.1 – Структурні рівні МЗЦТ [22]

Рівень зрілості	Назва	Опис
1	Базова цифровізація	Використання базових технологій, таких як підключення до Інтернету, наявність вебсайту та використання електронної пошти для покращення
2	Розширена цифровізація	Використання передових технологій для покращення роботи на додаток до Базової цифровізації.
3	Розумна цифровізація	Інтегроване використання цифрових технологій, таких як онлайн платформи та автоматизоване керування ланцюжком поставок на додаток до Розширеної цифровізації
4	Трансформативна цифровізація	Цифрові технології та можливості управління, що дозволяють трансформувати бізнес, наприклад цифрові активи, що створюють нові бізнес-моделі на додаток до Розумної цифровізації

2.2.2 Вимоги та критерії вибору МЗЦТ

Загалом, МЗЦТ для планування та управління проектом ЦТ мають відповідати наступним вимогам:

- описувати окремі рівні зрілості, які легко зрозуміти;
- пропонувати критерії, характеристики або контрольні точки, для оцінки існуючої ситуації перед запланованим кроком ЦТ;
- пропонувати опис передбачуваної майбутньої ситуації як мети ЦТ;
- описувати послідовні рівні зрілості, які дозволяють розробити шлях підвищення зрілості з визначеними етапами між наявною та майбутньою ситуацією;
- надавати практичні приклади організацій у подібних галузях, які успішно використовували певні МЗЦТ [21]-[22].

Для використання МЗЦТ у плануванні та управлінні проектом ЦТ необхідно вибрати відповідну МЗЦТ і, якщо необхідно, виконати її адаптацію до випадку використання. Критерії для вибору МЗЦТ можуть включати:

- відповідність конкретному об'єкту, суб'єкту чи випадку ЦТ, наприклад, чи призначено це для трансформації організації, бізнес-моделі, розробки нового продукту чи пропозиції послуг, розвитку компетенції особи чи команди? Обраний рівень зрілості визначає рівень зрілості об'єкта, суб'єкта чи організації, тому модель має відповідати цьому;
- відповідність певній області застосування, наприклад, професійній сфері (наприклад, управління, освіта) і відповідній галузі (наприклад, ІТ, автомобільна промисловість, енергетика);
- відповідність діапазону або розміру трансформації, наприклад, компанії, міста, системи освіти;
- відповідність необхідній деталізації планування;
- відповідність знанням користувача, зручність використання [21]-[22].

2.2.3 Оцінка ситуацій «As-Is» та «To-Be»

Перед застосуванням МЗЦТ необхідно мати можливість оцінити поточний стан або рівень зрілості ЦТ, щоб зрозуміти, з чого проєкт ЦТ повинен початися і на якій основі запускається етап ЦТ.

ЦТ розглядається як безперервний процес, у якому кожен етап – це лише крок або відстань. Таким чином, ситуація «As-Is» може не бути нульовим станом, і необхідно оцінити, на якому етапі зараз знаходиться об'єкт, суб'єкт або випадок ЦТ. Ситуація «To-Be» повинна описувати чіткі цілі та, можливо, ключові показники ефективності (KPI), яких потрібно досягти в певний момент часу.

Однак важко визначити точні цифри, яких необхідно досягти, і точні терміни, оскільки проєкти ЦТ не плануються традиційним способом через притаманну невизначеність. Натомість пропонується поступовий або дослідницький підхід, коли цілі та шлях трансформації часто коригуються, а проєкт виконується ітераціями. Більш актуальною є оцінка внеску проєкту в запланований вплив, який необхідний для стійкої ЦТ [26].

2.2.4 Проведення комплексного аналізу

Структурований комплексний аналіз поточного стану або рівня зрілості ЦТ «As-Is» можна провести за допомогою контрольних списків або анкет. У таблиці 2.2 наведено набір категорій [22], які є прикладом для типових контрольних точок в оцінці зрілості ЦТ. Ці категорії та запитання можна адаптувати для кожного рівня зрілості, представленого в таблиці 2.1. Це дозволить сформувати контрольний список, де відповіді на кожне запитання можна буде пов'язати з певним рівнем зрілості.

Таблиця 2.2 – Контрольний список [22]

Категорія	Запитання
Цифровізація сервісів	Формати запитів, виданих компанією та переданих постачальникам, підрядникам тощо.
Використання ПЗ та можливостей апаратного забезпечення	Використання ПЗ в повсякденних завданнях. Використання ПЗ для управління документами та завданнями. Використання сховища цифрових даних (хмарного або внутрішнього).
Витрати на ІКТ	Середньорічні витрати на ІКТ.
Навчання цифровим можливостям	Навчання персоналу цифровим можливостям організації.
Інвестиції в дослідження	Середній річний бюджет на R&D.

2.2.5 Використання ІТ-інструментів для оцінки рівня ЦТ

На додаток до контрольних списків і анкет, інші практичні ІТ-інструменти для оцінки процесів або зрілості, такі як SWOT-аналіз або відображення потоку створення цінності, можуть допомогти організаціям визначити додаткові характеристики, наприклад, сильні та слабкі сторони, можливості та загрози.

Крім того, можна використовувати модель готовності IMPULS [27], яка є онлайн-анкетною, яка дозволяє організаціям виконувати самооцінку та порівнювати з іншими компаніями. Вона включає шість вимірів із шістьма рівнями зрілості, рівні зрілості від рівнів 0 до 5 і згруповані в три мета-етапи, відомі як Новачки (рівні 0 і 1), Початківці (рівні 2) і Лідери (рівні 3, 4 і 5) та класифікуються як Аутсайдер, Початківець, Середній, Досвідчений, Експерт і Кращий виконавець (рис. 2.2) [27].

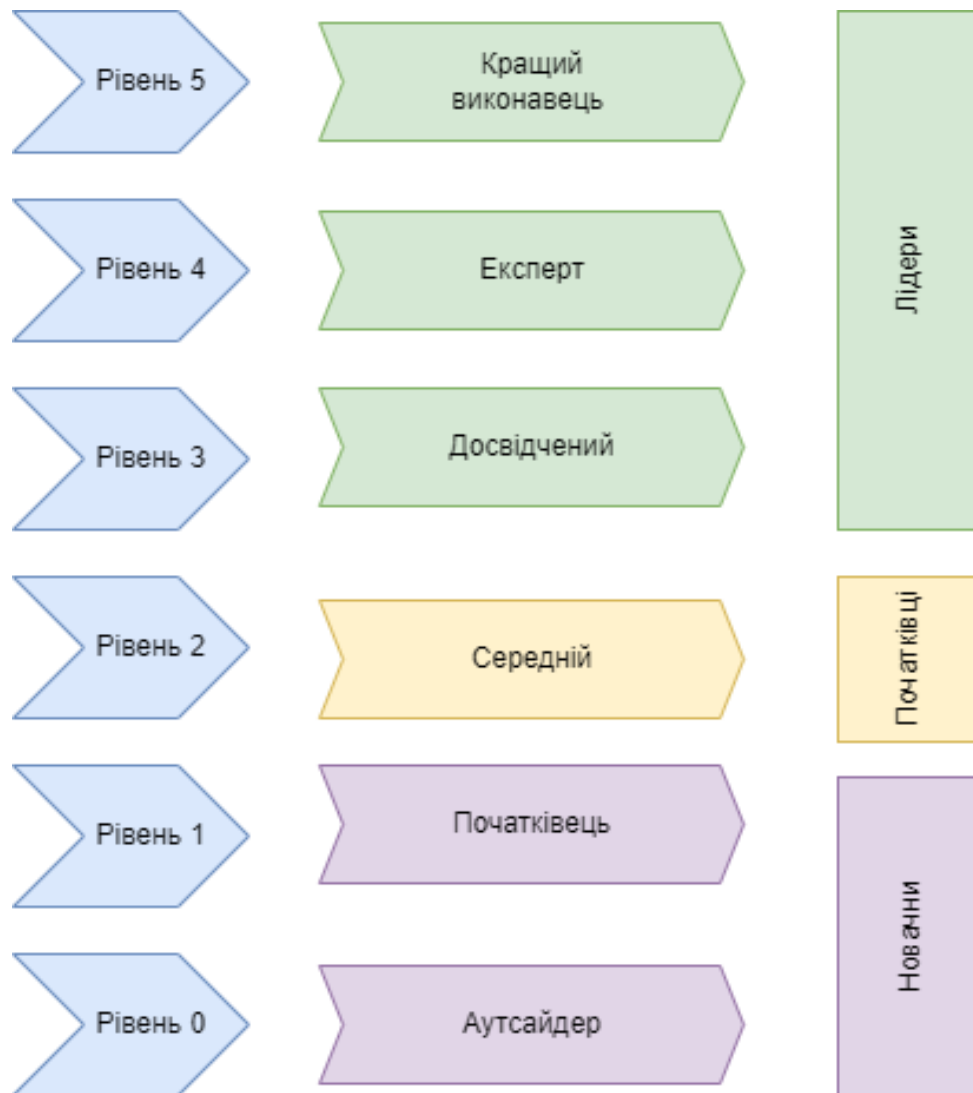


Рисунок 2.2 – Модель IMPULS [27]

2.3 Методика застосування МЗЦТ для оцінки рівня ЦТ

Запропонована методика застосування МЗЦТ наведена у вигляді схеми алгоритму на рис. 2.3.

Блок 1. Початок.

Блок 2. Визначення етапів ЦТ. Запитання, наведені в таблиці 2.2, можуть допомогти у визначенні поточного рівня МЗЦТ з таблиці 2.1. Це вимагає адаптації запитань із відповідями, які відображають певний рівень зрілості. Однак без експертної оцінки не можна вибрати дискретний рівень МЗЦТ і не можна визначити чіткий шлях для її застосування.

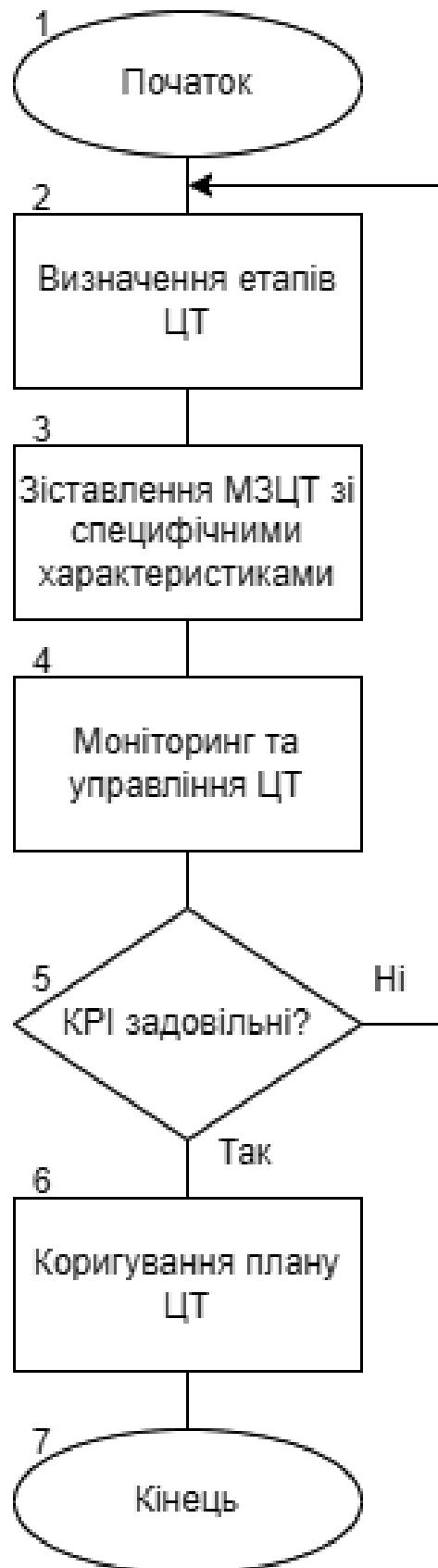


Рисунок 2.3 – Схема алгоритму методики застосування МЗЦТ

Блок 3. Зіставлення МЗЦТ із специфічними характеристиками. Крім таблиці 2.2, наведені в табл. 2.3 специфічні характеристики можуть допомогти безпосередньо оцінити стан «As-Is» на основі загальної МЗЦТ з таблиці 2.1. Ці характеристики можна використовувати як питання або точки оцінки для відображення поточного стану «As-Is» і допомогти зробити рішення щодо того, як перейти у стан «To-Be».

Таблиця 2.3 – Приклад відображення критеріїв на рівні МЗЦТ [22]

Рівень зрілості	Назва	Специфічні характеристики
1	2	3
1	Базова цифровізація	Наявність виділеного широкосмугового доступу для роботи. Наявність вебсайту, що представляє діяльність організації? Використання електронної пошти для спілкування з зацікавленими сторонами щодо питань, пов'язаних з роботою? Маєте підписку на платформи соціальних мереж як суб'єкт господарювання? Організаційні дані зберігаються локально. Відсутність формального навчання щодо цифрових можливостей. Жодних інвестицій у дослідження та розробку (R&D) Інвестиції в ІКТ становлять менше 1% обороту організації.
2	Розширена цифровізація	Використання аналітики даних соціальних мереж? Використання платформ SaaS або спеціальних платформ електронної комерції для закупівлі товарів і послуг? Використання багатосторонніх інструментів для співпраці та спілкування (наприклад, MS Teams, Zoom, Slack, Trello) Організаційні дані зберігаються в хмарі. Спеціальне навчання щодо нових цифрових можливостей. Інвестиції в дослідження та розробки (R&D) складають менше 1% обороту організації. Інвестиції в ІКТ становлять від 1% до 3% обороту організації.

Продовження таблиці 2.3

1	2	3
3	Розумна цифровізація	Використання порівняльного аналізу/динамічного зворотного зв'язку щодо показників ефективності. Внутрішня або зовнішня керована мережа, що забезпечує обмін даними та обмін даними. Використання спеціального програмного забезпечення для управління ланцюгами поставок і логістики. Використання програмного забезпечення для моделювання бізнес-процесів. Розширене та інтегроване використання IoT в організаційному просторі. Структуровані зовнішні навчальні програми щодо нових цифрових можливостей Інвестиції в дослідження та розробки (НДДКР) становлять від 1% до 3% від обороту організації. Інвестиції в ІКТ становлять від 1% до 3% обороту організації.
4	Трансформативна цифровізація	Використання повністю інтегрованої спеціально розробленої системи ERP для всієї організації з високим рівнем хмарної інтеграції. Використання передових кіберфізичних систем для експлуатації та управління будівлею. Використання розширених цифрових ресурсів, які динамічно представляють фізичне середовище (наприклад, Digital Twin) Інвестиції в дослідження та розробки, спрямовані на трансформацію бізнесу. Цифрові інвестиції добре управляються завдяки сильному об'єднанчому баченню, управлінню та корпоративній цифровій культурі, які можуть передбачити подальші зміни та втілити їх мудро. Структуровані внутрішні навчальні програми щодо нових цифрових можливостей. Інвестиції в дослідження та розробки (R&D) становлять більше 3% від обороту організації. Інвестиції в ІКТ становлять більше 5% обороту організації.

Блок 4. Моніторинг та управління ЦТ. КРІ традиційно використовуються як механізм звітності та обліку, метод ведення рахунку. Виконавці працювали над досягненням КРІ, і в багатьох випадках вони безпосередньо використовувалися як критерії успіху або принаймні як показники успіху для проєктів. Однак для того, щоб ефективно використовувати КРІ, слід зосередитись на їх використанні як на методі управління ЦТ [26]. Це вимагає визначення КРІ, які пов'язані з кінцевим результатом і впливом проєкту, слугуючи орієнтиром для того, щоб рухатися правильним шляхом. Технікою для цього може служити орієнтований на результат моніторинг і управління.

Існують численні ІТ-інструменти для відстеження КРІ та керування ними як частиною або набором завдань, зокрема Jira, MeisterTask, Confluence, Trello тощо. Вони допомагають розподіляти навантаження, призначати дати та цілі, відстежувати залежності та зв'язки тощо.

Блок 5. Результати задовільні? Якщо так, то перехід на Блок 7, якщо Ні, повернення на Блок 2.

Блок 6. Коригування плану ЦТ. Організація має регулярно переглядати та коригувати свої плани ЦТ на основі відгуків, отриманих результатів і мінливої ринкової динаміки, забезпечуючи узгодженість зі стратегічними цілями та максимізуючи шанси на успіх. При цьому КРІ мають розглядатися як засіб досягнення вимірності ЦТ, а не як її мета [26].

Блок 7. Кінець.

2.4 Висновки до розділу 2

В даному розділі відповідно до мети та задач дипломної роботи було проведено дослідження особливостей та типів МЗЦТ та запропоновано методику застосування МЗЦТ БП.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Призначення та вимоги до ПЗ

Проектоване ПЗ призначене для ЦТ БП віддаленого моніторингу діяльності фахівців, що в своїй діяльності використовують платформу BAS ERP.

Функціональні вимоги:

- реєстрація нових користувачів;
- аутентифікація та авторизація користувачів;
- перевірка дозволів користувача за токеном;
- додавання нових номерів телефону для користувача;
- виведення списку менеджерів системи та їх номерів телефону;
- виведення списку задач на підтвердження;
- показ детальної інформації про бізнес-задачу;
- підтвердження бізнес-задачі;
- спілкування з зовнішнім джерелом даних.

Окрім цього аналіз показав, що програма повинна забезпечувати наступні нефункціональні вимоги:

- зручність реєстрації нового користувача;
- зручність входу в наявний акаунт;
- інтуїтивно-зрозумілий адаптивний інтерфейс;
- можливість модифікації програми під нові потреби бізнесу.

На основі аналізу функціональних та нефункціональних вимог було розроблено діаграму прецедентів, що демонструє взаємодію користувачів та зовнішніх сервісів з програмою (рис. 3.1).

Для реалізації поставлених задач програма буде отримувати інформацію з системи BAS, обробляти дані та зберігати у локальній базі даних (БД). Користувач зможе взаємодіяти з цими даними, а результати взаємодії будуть, за потреби, передані до BAS ERP.

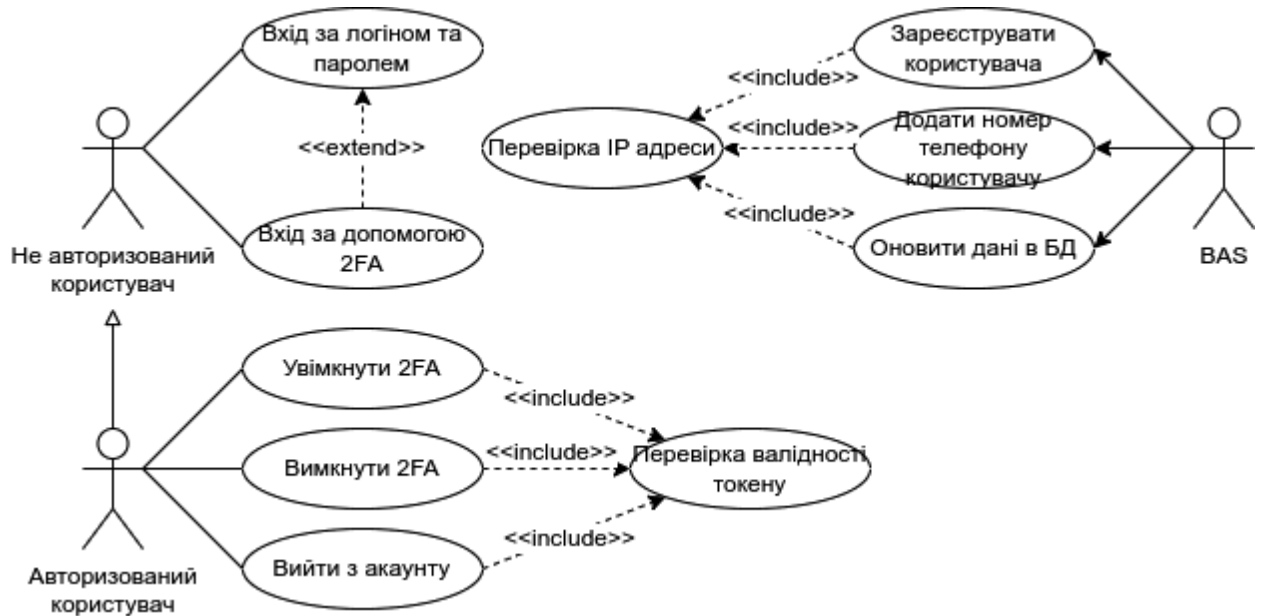


Рисунок 3.1 – Діаграма прецедентів

3.2 Розробка архітектури системи

Архітектура системи (рис. 3.2) об’єднує три блоки, відображає зв’язки між ними та типи даних, що циркулюють в системі.

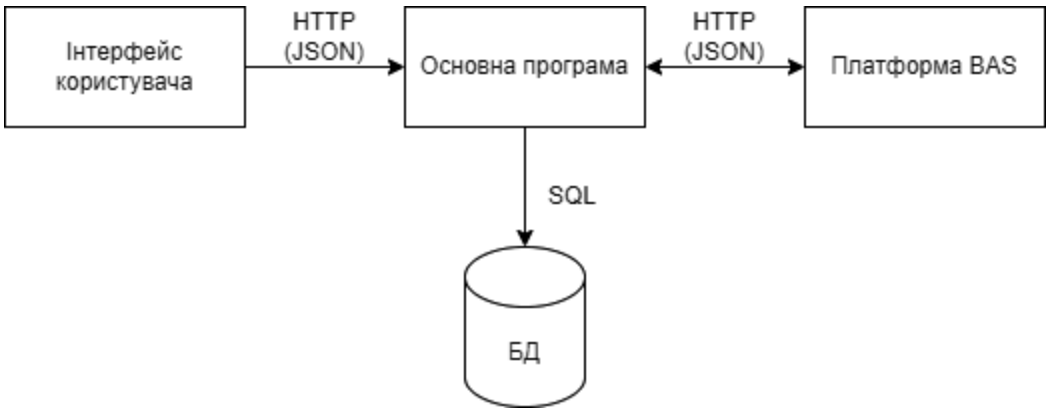


Рисунок 3.2 – Узагальнена архітектура системи

Блок інтерфейсу представляє собою вебсторінку, що дозволить робити HTTP запити до основної програми та отримувати й виводити відповіді на ці запити.

Основна програма відповідає за авторизацію та аутентифікацію користувачів, отримання та аналіз даних з платформи BAS, зберігання та створення інформації про користувачів, а також за формування відповідей на запити користувачів.

Локальна БД призначена для зберігання оперативної інформації, що дозволить зменшити навантаження на головну БД платформи BAS та оперативніше реагувати на запити, що надходять в систему.

Платформа BAS – це головний сервіс, з якого будуть отримуватися дані та до якого буде відсилатися інформація про статуси бізнес-задач та бізнес-процесів, які виконують користувачі.

3.3 Вибір інструментарію розробки

3.3.1 Аналіз та вибір методу аутентифікації користувачів

Розробка ефективних методів аутентифікації є дуже важливим завданням для будь-якої системи чи послуги, особливо в мережі Інтернет [28].

Діапазон методів аутентифікації дуже великий:

- ідентифікатор /пароль: щоб відкрити сеанс на комп'ютері або аутентифікуватися в Інтернеті;
- PIN-код (персональний ідентифікаційний номер): для розблокування смарт-картки;
- RFID-картка або відбиток пальця для доступу до будівлі;
- система розпізнавання обличчя з вебкамерою: для відкриття сесії в Інтернеті;
- USB-жетон;
- маркер одноразового пароля та ін. [29].

Кожен із методів аутентифікації має певну галузь використання та властиві переваги та недоліки. Метою аутентифікації є перевірка ідентичності особи з заданим рівнем довіри. Додаткові методи аутентифікації, такі як

двофакторна аутентифікація або аутентифікація за допомогою токенів, можуть забезпечити вищий рівень безпеки для користувачів.

Одним з найчастіше використовуваних методів додаткової аутентифікації є двофакторна аутентифікація (Two-Factor Authentication, 2FA), яка полягає в тому, щоб користувач підтвердив свою ідентичність за допомогою двох різних методів. Основні види 2FA включають використання різних даних та засобів [30]:

- пароль/код доступу, який надсилається на мобільний телефон;
- пароль/скан відбитка пальця або розпізнавання голосу;
- смарткартка або USB-ключ/пароль або PIN-код;
- ключ картки доступу/скан відбитка пальця.

Порівняння різних методів 2FA наведено в таблиці 3.1.

Таблиця 3.1 – Порівняння методів аутентифікації [29]-[30]

Категорія	Пароль	SMS	USB-ключ	TOTP	Біометрія
Безпека	Слабка	Слабка	Сильна	Сильна	Сильна
Використання	Зручний	Зручний	Складніше	Зручний	Зручний
Комфорт	Зручний	Зручний	Потребує флешки	Зручний	Потребує додатковий пристрій
Додаткові витрати	Немає	На кожне SMS	Покупка флешки	Немає	Покупка пристрою біометричної ідентифікації

Протоколи аутентифікації є важливою складовою безпеки інформаційних систем і мереж. Їх розробляють і підтримують провідні організації для забезпечення безпечної та простої аутентифікації на різних

системах, сервісах та вебсайтах. Ці протоколи дозволяють підтверджувати ідентичність користувачів, що забезпечує доступ до систем і мереж тільки авторизованим особам, що в свою чергу дозволяє зменшити ризики порушення безпеки та зловживання даними.

Результати порівняння протоколів аутентифікації наведено в табл. 3.2.

Таблиця 3.2 – Порівняння протоколів аутентифікації [31]-[34]

Категорія	OAuth	SAML	Kerberos	LDAP
1	2	3	4	5
Призначення	Протокол авторизації для вебзастосунків	Протокол авторизації для різних застосунків	Протокол аутентифікації для локальних мереж	Протокол каталогу для зберігання даних користувачів
Використання	Для авторизації користувачів на різних сайтах	Для авторизації користувачів на різних сайтах	Для авторизації та аутентифікації користувачів	Для зберігання та доступу до інформації про користувачів
Підтримка	Вебсайти та мобільні застосунки	Вебсайти та мобільні застосунки	Локальні мережі	Локальні мережі та сервери застосунків
Формат	Токени	XML	Квитки	Дерева

Продовження таблиці 3.2

1	2	3	4	5
Механізм	Авторизація за допомогою стороннього сервісу	Авторизація з використанням цифрових підписів	Авторизація з використанням зашифрованих квитків	Аутентифікація за допомогою логіну та паролю
Безпека	TLS, SSL	XML-підписи, SSO	Шифрування	TLS, SSL, шифрування паролю
Розширюваність	Масштабується до багатьох сервісів	Підтримує багато механізмів авторизації	Обмежена	Масштабується до багатьох користувачів

Аналіз показав, що згідно зі стандартом інтернету RFC 6238, алгоритм TOTP є розширенням алгоритму одноразового пароля (OTP), а саме алгоритму на основі HMAC-одноразового пароля (HOTP), який визначено у RFC 4226, для підтримки фактора рухомого часу. Алгоритм HOTP визначає алгоритм OTP на основі подій, де рухомих фактором є лічильник подій. Варіант OTP на основі часу надає короткочасні значення OTP, що бажано для підвищення безпеки [35].

Цей алгоритм може використовуватися в широкому спектрі мережевих додатків, від віддаленого доступу до віртуальної приватної мережі (VPN) та входу до Wi-Fi мережі до транзакційно-орієнтованих вебзастосунків. Автори вважають, що спільний алгоритм сприятиме впровадженню 2FA в Інтернеті, дозволяючи забезпечити взаємодію між комерційними та відкритими реалізаціями [36].

Таким чином, на основі проведених досліджень було вирішено реалізувати аутентифікацію на основі логіну та паролю. Окрім цього програма надасть можливість активації 2FA на основі TOTP коду.

3.3.2 Вибір мови та середовища програмування

Результати порівняльного аналізу мов програмування наведено в таблиці 3.3.

Таблиця 3.3 – Порівняння мов програмування [37]

Характеристика	Golang	C++	Python	Java
Синтаксис	Легкий	Важкий	Легкий	Середній
Продуктивність	Висока	Висока	Низька	Середня
Навчання	Легкий	Важкий	Легкий	Середній
Конкур. придатність	Висока	Середня	Середня	Низька
Розширюваність	Висока	Висока	Висока	Висока
Підтримка ООП	+	+	+	+
Підтримка ФП	+-	+	+	+-

Однією з важливих переваг мови Golang є те, що це мова програмування, що компілюється. Через це програмам, що реалізовані цією мовою притаманна швидкодія (інколи навіть вища, ніж у програм на C++). Також, не дивлячись на те, що Golang – це мова зі статичною типізацією, в ній є також динамічний варіант типізації. Завдяки простоті цієї мови, програми, написані на ній, можуть бути легко перероблені професіональними програмістами під потреби свого проєкту [37].

В табл. 3.4 наведено результати порівняння середовищ програмування. Аналіз показав, що VSCode є гарним вибором для розробки програм мовою програмування Go.

Таблиця 3.4 – Порівняння середовищ розробки [38]-[40]

Характеристика	VSCode	Sublime Text	IntelliJ IDEA	Eclipse
Відкритість	+	-	-	+
Швидкодія	++	+++	+	+
Підтримка мов програмування	+	+	+	-
Відлагодження	+	-	+	+
Git	+	+	+	+
Docker	+	-	+	+
IntelliSense	+	-	+	+
Вбудований термінал	+	-	+	+
Наявність безкоштовної версії	+	-	-	+

Використовуючи розширення Go для Visual Studio Code, можливо отримати такі функції, як IntelliSense, навігація по коду, пошук символів, тестування, налагодження та багато інших, які допоможуть у розробці [39].

3.3.3 Вибір СКБД

В таблиці 3.5 наведено результати порівняльного аналізу популярних систем керування базами даних (СКБД), які використовуються для роботи з даними в різних типах вебзастосунків та вебсайтів [41]-[43].

Аналіз показав, що PostgreSQL — це передова система реляційних баз даних корпоративного класу з відкритим кодом.

Таблиця 3.5 – Порівняння СКБД [41]-[43]

Характеристика	PostgreSQL	Oracle	SQL Server	MySQL
Відкритість	+	-	-	+
Ліцензія	PostgreSQL License	Proprietary	Proprietary	GPL
Розширюваність	++	++	+	++
SQL	+	+	+	+
JSON	+	+	+	+
XML	+	+	+	+
NoSQL	+	-	-	+-
Повнотекстовий пошук	+	+	+	+
Реплікація	+	+	+	+
Індексіція	++	++	++	++
BLOB	+	+	+	+
Швидкодія	++	++	++	+

PostgreSQL підтримує як SQL (реляційні), так і JSON (нереляційні) запити. PostgreSQL використовується як основне рішення для багатьох веб, а також мобільних і аналітичних застосунків [41]-[43].

Зокрема, при співставленні PostgreSQL і SQL Server було відзначено наступні відмінності:

- право власності та ліцензування: PostgreSQL — це система з відкритим вихідним кодом, керована спільнотою, а SQL Server належить корпорації Microsoft;

- операційні системи: PostgreSQL призначена для роботи в декількох операційних системах, включаючи Linux, macOS і Windows, тоді як SQL Server в першу чергу призначена для роботи в Windows, але також може працювати і в Linux;

- вартість: PostgreSQL можна використовувати безкоштовно, тоді як SQL Server вимагає ліцензії і може виявитися дорогим, особливо для великих організацій;
- масштабованість: SQL Server має готові пакети, які посилюють масштабування, в той час як PostgreSQL вимагає більше ручних зусиль для масштабування БД;
- особливості: PostgreSQL пропонує більш широкий набір функцій, ніж SQL Server, включаючи підтримку власного типу даних JSON, наслідування таблиць та багато інших розширених функцій. SQL Server, з іншого боку, пропонує функції, специфічні для продуктів Microsoft, такі як інтеграція з Microsoft Office;
- продуктивність: SQL Server забезпечує більш високу продуктивність для певних програм завдяки своїй традиційній структурі, в той час як PostgreSQL відома своєю продуктивністю та може обробляти більші обсяги даних;
- типи даних користувача: PostgreSQL підтримує тип даних користувача PostgreSQL, що робить її трохи більш гнучкою, ніж SQL Server;
- вбудовані функції: PostgreSQL має більший набір вбудованих функцій порівняно з SQL Server;
- хранимі процедури: PostgreSQL підтримує сховища процедур, написаних на кількох мовах, тоді як SQL Server підтримує сховища процедур, написаних тільки на T-SQL [44].

Тому, як СКБД було вирішено обрати об'єктно-реляційну систему керування базами даних PostgreSQL, яка забезпечує швидкодію, різноманітний функціонал та є доволі популярною, а отже є спільнота розробників, що мають досвід роботи з нею.

3.4 Висновки до розділу 3

В цьому розділі, відповідно до мети та завдань дипломної роботи виконано аналіз вимог до проєктованого ПЗ, розроблено архітектуру програмної системи, проведено огляд та обґрунтовано вибір інструментарію розробки: мова програмування Golang, середовище розробки Visual Studio Code, СКБД PostgreSQL. Прийнято рішення щодо організації аутентифікації користувачів системи на основі логіну та паролю. Окрім цього програма буде мати можливість активації 2FA на основі TOTP коду.

4 КОНСТРУЮВАННЯ ПЗ

4.1 Розробка алгоритму функціонування програми

Узагальнений алгоритм функціонування основної програми наведено на рисунку 4.1.

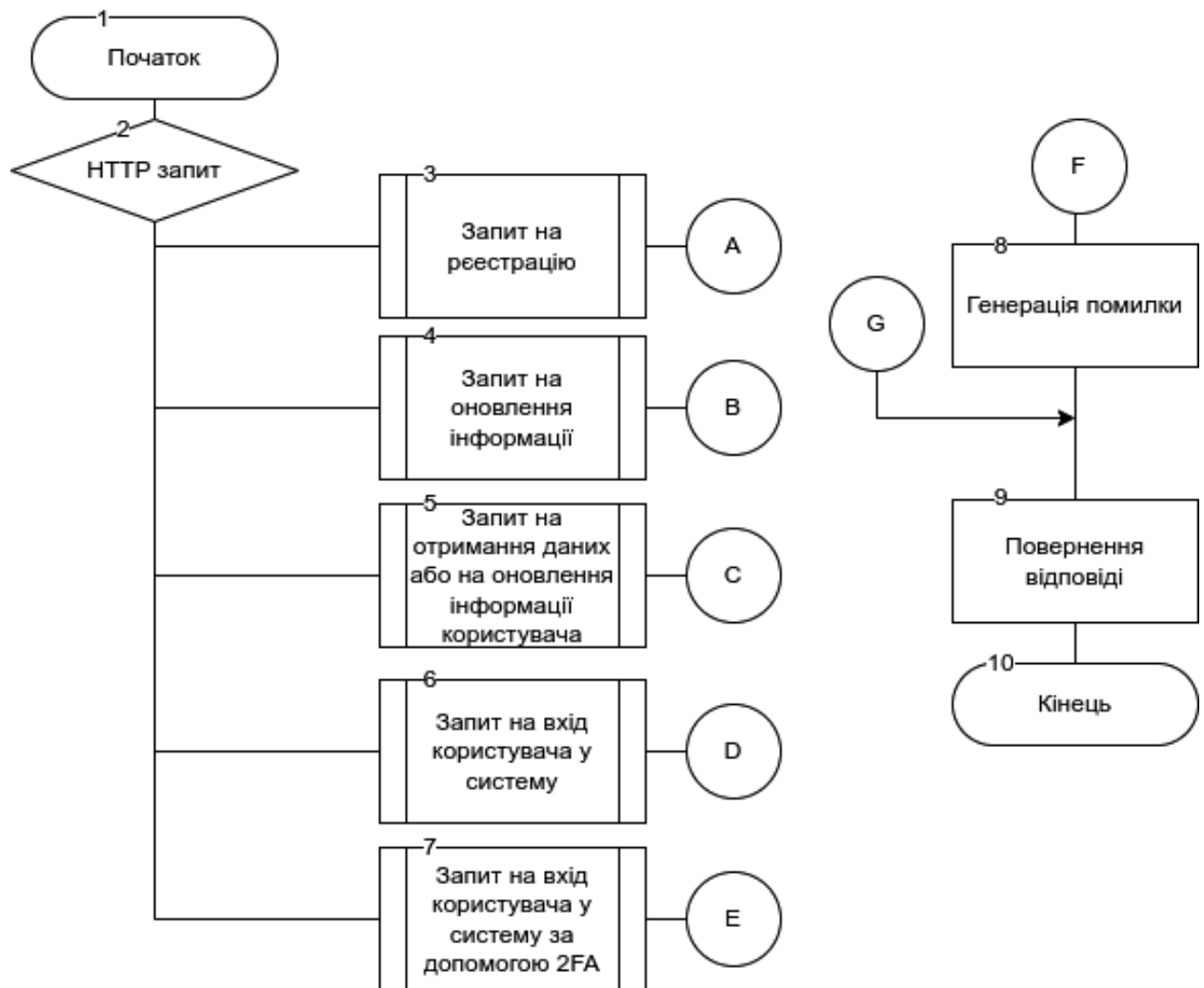


Рисунок 4.1 – Узагальнений алгоритм функціонування основної програми

Блок 1. Початок.

Блок 2. HTTP-запит. Передбачає, що користувач може здійснювати різні типи запитів на:

- реєстрацію;

- оновлення інформації;
- отримання даних або на оновлення інформації користувача;
- вхід в систему;
- вхід в систему за допомогою 2FA.

Блоки 3-7 це функції, що мають виконуватися при отриманні певного типу запиту. Детальні схеми алгоритмів цих функцій наведено далі.

Блок 8. Генерація помилки. Передбачає створення та журналювання помилки, що виникла при перевірці даних введених користувачем та подальше повернення помилки користувачеві у Блоці 9.

Блок 9. Повернення відповіді. Передбачає, що користувач буде отримувати підготовлену відповідь у вигляді рядка або JSON файлу на кожен свій запит до програми.

Блок 10. Кінець.

На рис. 4.2 представлено схему алгоритму процедури Запит на реєстрацію.

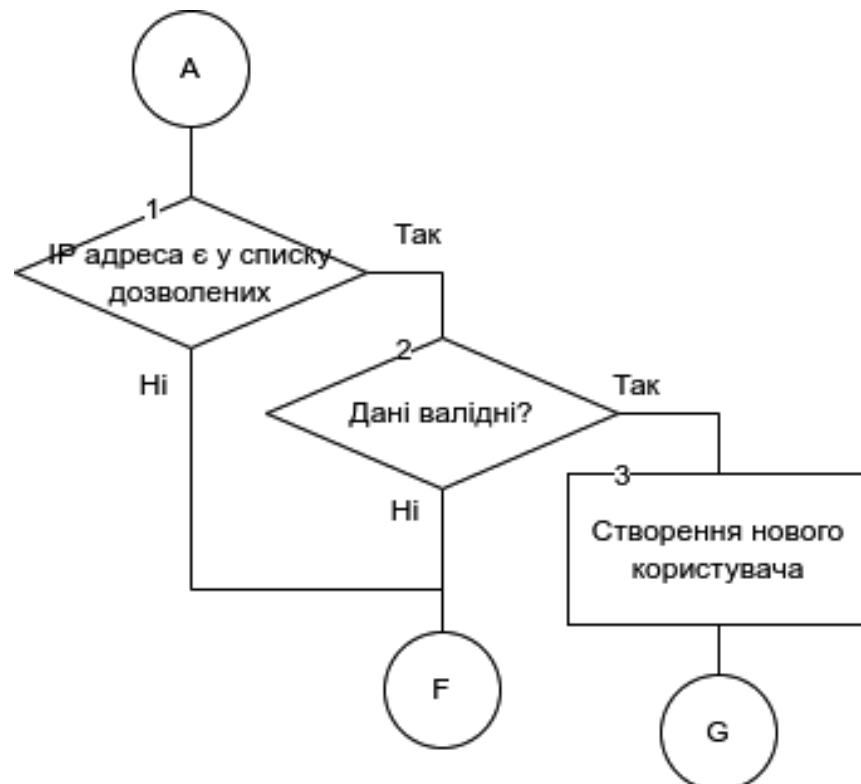


Рисунок 4.2 – Запит на реєстрацію

Блок 1. IP адреса є у списку дозволених? Передбачає перевірку IP-адреси, з якої надійшов запит на реєстрацію. Якщо цієї адреси немає у списку дозволених, то програма переходить до виконання Блоку 8 алгоритму функціонування основної програми. Інакше виконується Блок 2 поточного алгоритму.

Блок 2. Дані валідні? Передбачає перевірку на наявність та валідність необхідних для подальшої роботи даних. Якщо дані не валідні, то програма переходить до виконання Блоку 8 алгоритму функціонування основної програми. Інакше виконується Блок 3 поточного алгоритму.

Блок 3. Створення нового користувача. Передбачає створення нового користувача, додавання його номерів телефону у БД та подальше повернення повідомлення про успіх користувачеві у Блок 9 алгоритму функціонування основної програми.

На рисунку 4.3 наведено схему алгоритму функції Запит на оновлення інформації.

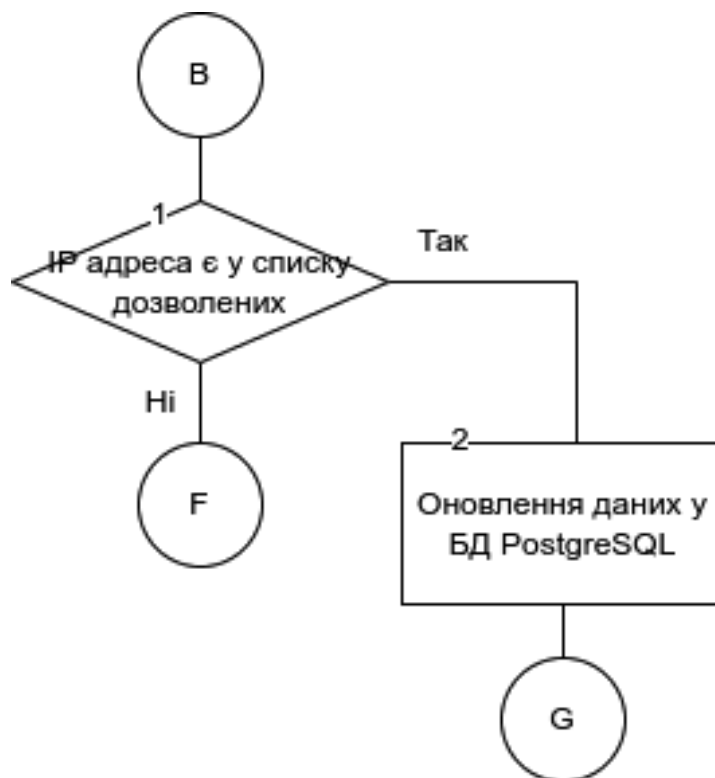


Рисунок 4.3 – Запит на оновлення інформації

Блок 1. IP адреса є у списку дозволених? Передбачає перевірку IP-адреси, з якої надійшов запит на оновлення інформації. Якщо цієї адреси немає у списку дозволених, то програма переходить до виконання Блоку 8 алгоритму функціонування основної програми. Інакше виконується Блок 2 поточного алгоритму.

Блок 2. Оновлення даних у базі даних PostgreSQL. Передбачає оновлення даних у БД та подальше повернення повідомлення про успіх користувачеві у Блок 9 алгоритму функціонування основної програми.

На рис. 4.4 наведено схему алгоритму функції Запит на отримання даних або на оновлення інформації користувача.

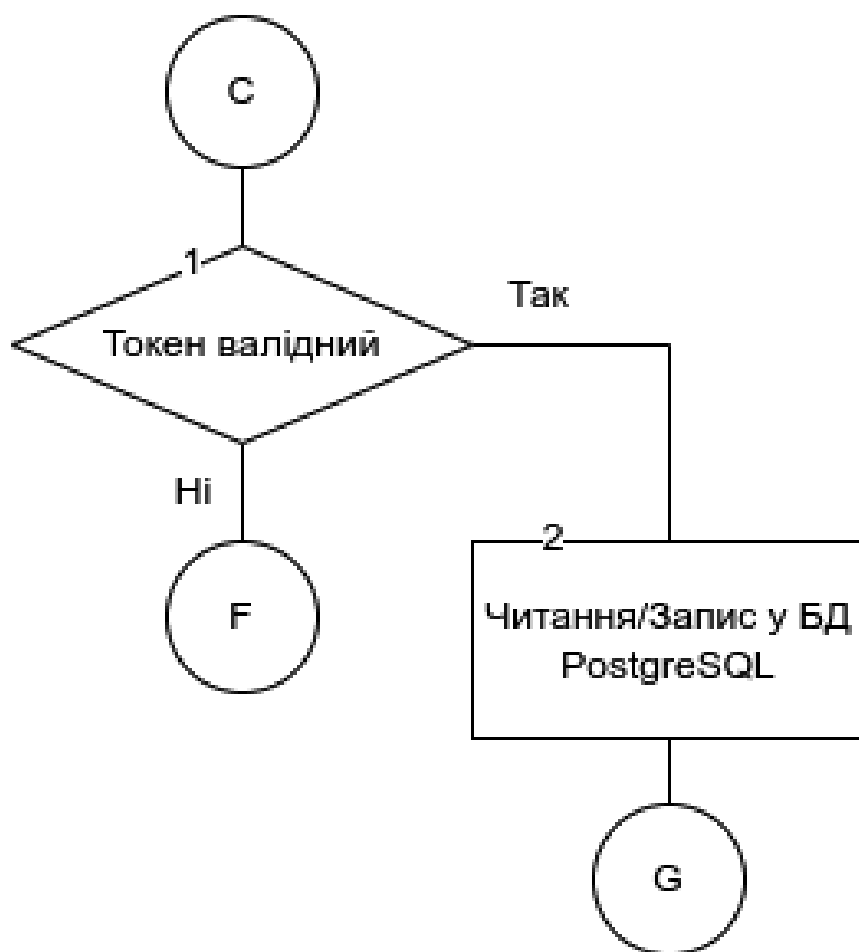


Рисунок 4.4 – Запит на отримання даних або на оновлення інформації користувача

Блок 1. Токен валідний? Передбачає перевірку токена користувача, з якого надійшов запит на оновлення даних користувача або отримання даних з БД. Якщо токен не валідний або термін його дії завершений, то програма переходить до виконання Блоку 8 алгоритму функціонування основної програми. Інакше виконується Блок 2.

Блок 2. Читання/запис у БД PostgreSQL. Передбачає читання даних з БД або оновлення даних користувача у БД, підготовку даних для відповіді та подальше повернення цих даних користувачеві у Блоці 9 алгоритму функціонування основної програми.

На рис. 4.5 наведено схему алгоритму функції Запит на вхід користувача у систему.

Блок 1. Дані валідні? Передбачає перевірку на наявність та валідність необхідних для подальшої роботи даних. Якщо дані не валідні, то перехід до виконання Блоку 8 алгоритму функціонування основної програми. Інакше виконується Блок 2.

Блок 2. Отримання користувача з БД PostgreSQL. Передбачає отримання даних користувача за його номером телефону з БД.

Блок 3. Паролі співпадають? Передбачає перевірку на однаковість пароля у БД та того паролю, що ввів користувач. Якщо паролі не збігаються, то програма переходить до виконання Блоку 8 алгоритму функціонування основної програми. Інакше виконується Блок 4 поточного алгоритму.

Блок 4. ОТР увімкнуто? Передбачає перевірку на те, чи увімкнув користувач 2FA. Якщо 2FA не була увімкнута, то програма переходить до виконання Блоку 5, інакше виконується Блок 6.

Блок 5. Генерація токена. Передбачає генерацію токена для подальшої роботи з сервісом та повернення токена користувачеві у Блоці 9 алгоритму функціонування основної програми.

Блок 6. Генерація тимчасового ідентифікатору. Передбачає генерацію тимчасового ідентифікатору та збереження даних користувача для подальшої

2FA та цього ідентифікатору користувачеві у Блоці 9 алгоритму функціонування основної програми.

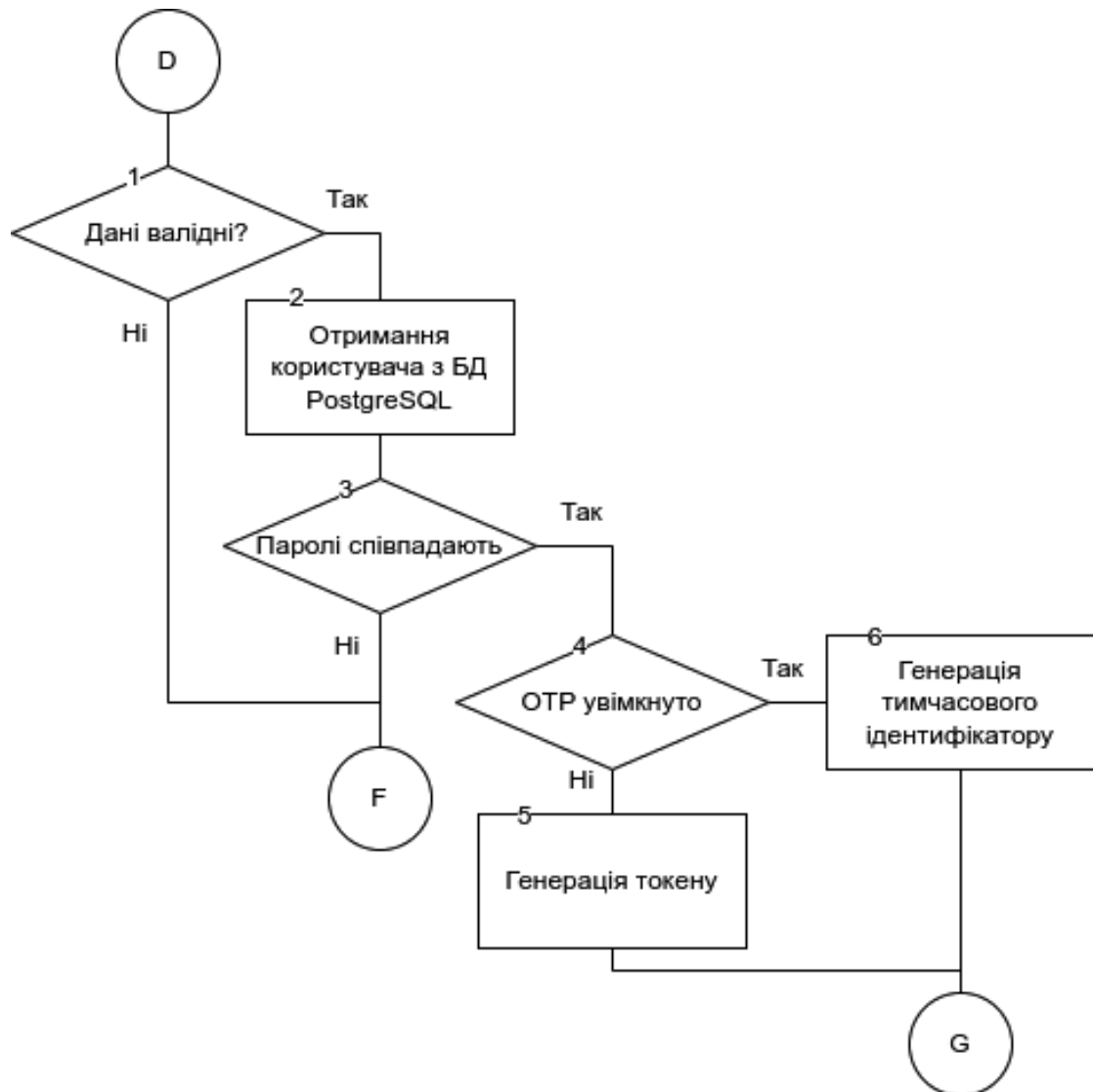


Рисунок 4.5 – Запит на вхід користувача у систему

На рис. 4.6 наведено схему алгоритму функції Запит на вхід користувача у систему за допомогою 2FA.

Блок 1. Дані валідні? Передбачає перевірку на наявність та валідність необхідних для подальшої роботи даних. Якщо дані не валідні, то програма переходить до виконання Блоку 8 алгоритму функціонування основної програми. Інакше виконується Блок 2.

Блок 2. Отримання користувача з кешу за тимчасовим ідентифікатором. Передбачає отримання користувача за представленим тимчасовим ідентифікатором з кешу.

Блок 3. Пароль OTP валідний? Передбачає перевірку на валідність пароля TOTP, що ввів користувач. Якщо пароль не валідний, то програма переходить до виконання Блоку 8 алгоритму функціонування основної програми. Інакше виконується Блок 4 поточного алгоритму.

Блок 4. Генерація токена. Передбачає генерацію токена для подальшої роботи з сервісом та повернення токена користувачеві у Блоці 9 алгоритму функціонування основної програми.

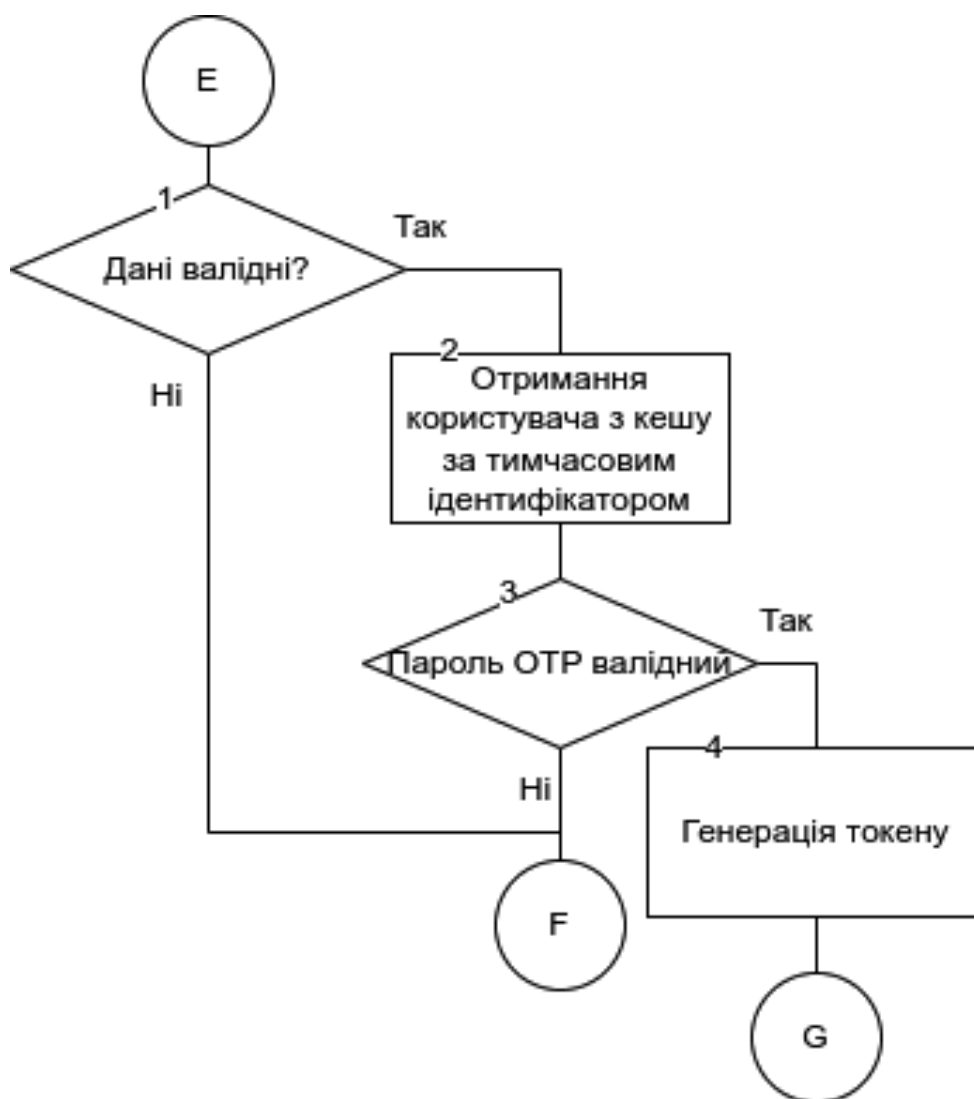


Рисунок 4.6 – Запит на вхід користувача у систему за допомогою 2FA

4.2 Структура основної програми

Основна програма містить два блоки: User та Tasks. Блок User містить модулі: Реєстрація, Аутентифікація, Перевірка IP адреси. Блок Tasks містить і модулі: Отримання списку бізнес-задач, Оновлення статусу бізнес-задачі, Створення бізнес-задачі. Окрім основних блоків та модулів на структурній схемі (рис. 4.7) показано вебсервер, який приймає http запити та БД PostgreSQL, в якій зберігаються дані.

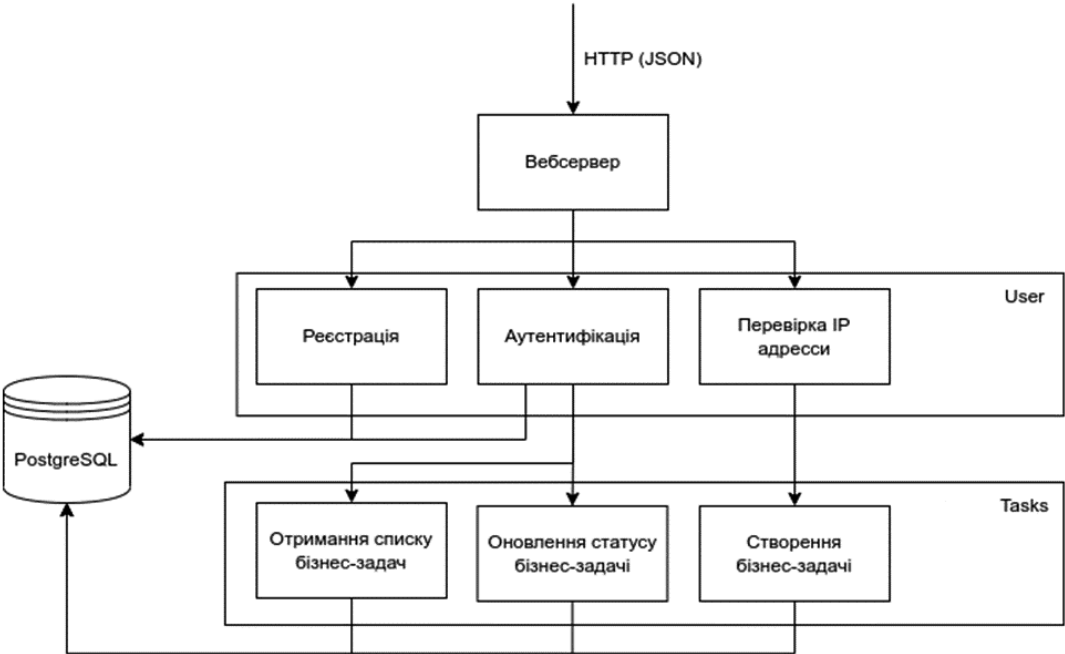


Рисунок 4.7 – Структура основної програми

Опис складових програми наведено в таблиці 4.1.

Таблиця 4.1 – Складові програми

Блок	Опис
User	Відповідає за реєстрацію користувачів та перевірку валідності токену користувача.
Tasks	Відповідає за роботу з бізнес-задачами.

4.3 Опис особливостей програмної реалізації

4.3.1 Блок User

Блок User надає рішення, що забезпечують базовий функціонал для аутентифікації користувачів. Назви функцій та їх опис наведено в таблиці 4.2.

Таблиця 4.2 – Функції блоку User

Назва функції	Опис
1	2
Register	Відповідає за створення нового користувача. Пароль нового користувача хешується за допомогою бібліотеки bcrypt. Приймає на вхід інформацію о користувачі. На виході або віддає картинку для 2FA (якщо була увімкнута при реєстрації) або не віддає нічого.
Login	Відповідає за створення токена для вже наявного користувача, якщо він не увімкнув 2FA. Окрім цього перевіряє коректність пароля та логіну. На вході приймає логін і пароль. На виході або віддає токен або віддає згенерований випадковий ідентифікатор та просить ввести код totp.
TwoFA	Відповідає за створення токена для вже наявного користувача, якщо він увімкнув 2FA. На вході приймає ідентифікатор з функції Login і код totp. На виході віддає токен.
AddAlternativeNumber	Відповідає за додання альтернативного номеру для входу в акаунт користувача.

Продовження таблиці 4.2

1	2
EnableTwoFA	Відповідає за увімкнення 2FA для вже наявного користувача. Приймає на вхід ідентифікатор користувача та його токен. На виході віддає картинку для 2FA.
DisableTwoFA	Відповідає за вимкнення 2FA для вже наявного користувача. Приймає на вхід ідентифікатор користувача та його токен. На виході не віддає нічого.
Logout	Відповідає за вихід користувача з системи та видалення його токenu. Приймає на вхід токен користувача. На виході не віддає нічого.

4.3.2 Блок Tasks

Блок Tasks надає рішення, що забезпечують базовий функціонал для створення та оновлення статусу бізнес-задач. Назви функцій та їх опис наведено в таблиці 4.3.

Таблиця 4.3 – Складові блоку Tasks

Назва функції	Опис
1	2
GetTasks	Відповідає за отримання списку бізнес-задач користувача. Приймає на вхід ідентифікатор користувача. На виході віддає всі бізнес-задачі, що призначені цьому користувачеві.

Продовження таблиці 4.3

1	2
GetVacations	Відповідає за отримання списку запитів на відпустку користувача. Приймає на вхід ідентифікатор користувача. На виході віддає всі запити на відпустку, що призначені цьому користувачеві.
GetPayments	Відповідає за отримання списку запитів на оплату послуги користувача. Приймає на вхід ідентифікатор користувача. На виході віддає всі запити на оплату послуги, що призначені цьому користувачеві.
SaveTasks	Відповідає за створення нових бізнес-задач. Приймає на вхід інформацію о бізнес-задачах. На виході віддає статус “Успішно”.
SaveVacations	Відповідає за створення нових запитів на відпустку. Приймає на вхід інформацію о запитах на відпустку. На виході віддає статус “Успішно”.
SavePayments	Відповідає за створення нових запитів на оплату послуги. Приймає на вхід інформацію о запитах на оплату послуги. На виході віддає статус “Успішно”.
UpdateTaskStatus	Оновлює статус бізнес-задачі або підтверджуючи або відхиляючи її. Приймає на вхід ідентифікатор задачі та її новий статус. На виході віддає статус “Успішно”.

4.3.3 Структура БД

Дані у БД зберігаються в 3 різних таблицях: "users", "contact_info", "tasks". Структура та зв'язки таблиць БД наведені на рисунку 4.8.

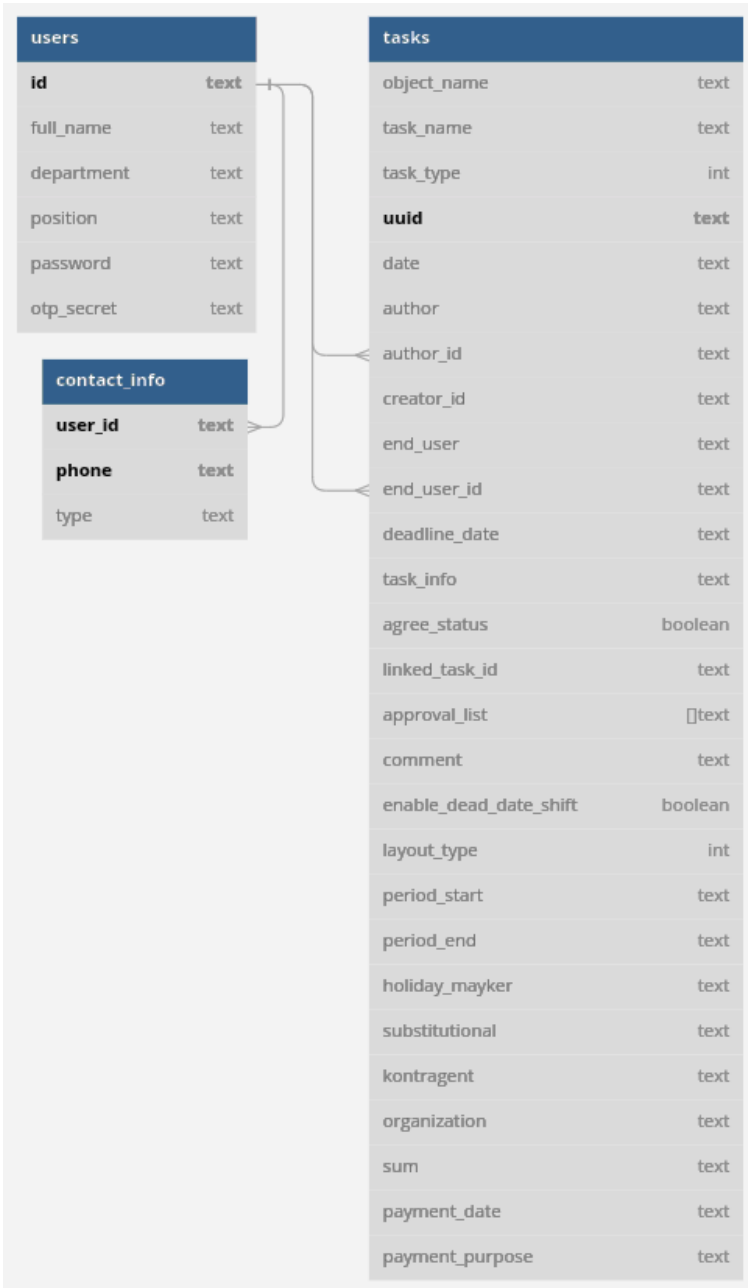


Рисунок 4.8 – Узагальнена структура БД

Таблиця "users" містить основну інформацію про користувачів, таку як повне ім'я, відділ, посада, пароль та інші важливі дані.

Таблиця "contact_info" містить контактну інформацію користувачів, таку як номер телефону та його тип.

Таблиця "tasks" містить повну інформацію про бізнес-задачу, її тип, статус, а також посилання на користувача, що створив цю бізнес-задачу та посилання на користувача для якого ця бізнес-задача призначена.

4.3.3 Інтерфейс програми

Інтерфейс програми – це будь-яка html сторінка, що надає необхідний для роботи програми функціонал. В даній роботі в якості вебсторінки використано програму Postman, зовнішній вигляд інтерфейсу якої представлено на рисунку 4.9. Postman використовується для тестування та автоматизації API, дозволяючи розробникам створювати, організовувати та виконувати запити API, відстежувати відповіді та співпрацювати з членами команди [45].

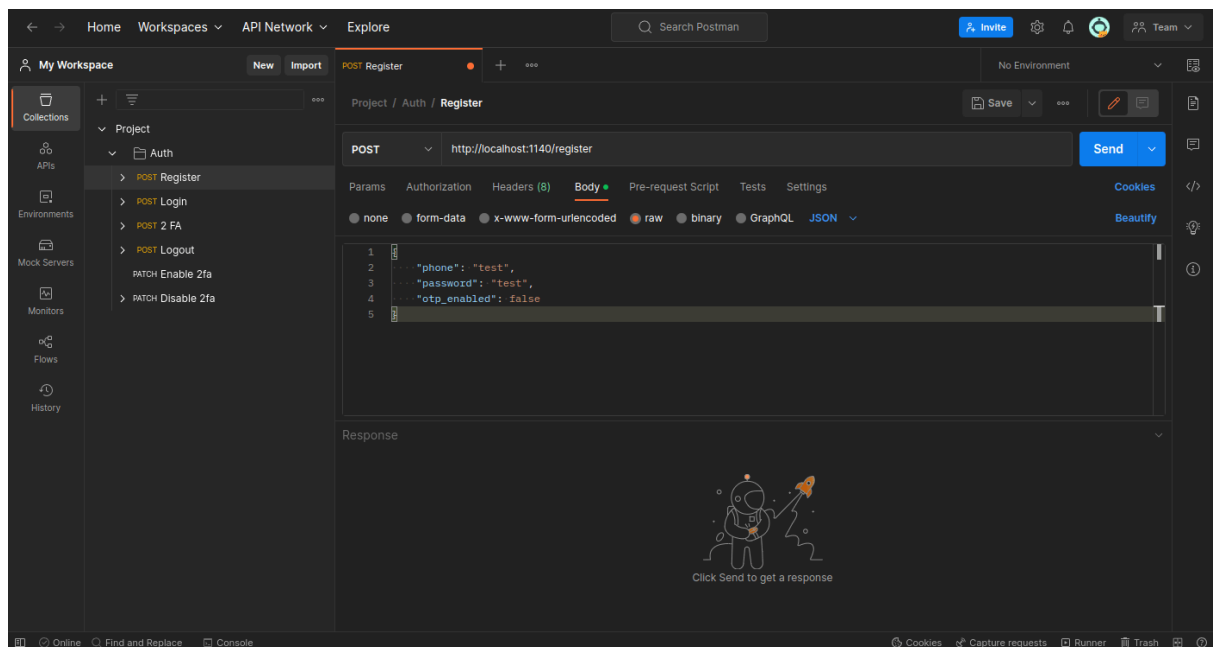


Рисунок 4.9 – Зовнішній вигляд інтерфейсу програми Postman

4.4 Висновки до розділу 4

В цьому розділі відповідно до мети та завдань роботи виконано конструювання ПЗ, розроблено базові алгоритми та структуру основної програми, а в якості інтерфейсу було запропоновано використати програму Postman. Наведено детальний список розроблених блоків програми та їх функціональних модулів.

5 ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ СИСТЕМИ

5.1 Опис застосування

ПЗ призначене для ЦТ БП віддаленого моніторингу та контролю бізнес-задач користувачів системи BAS. Воно може використовуватися для затвердження результатів виконання бізнес-задач та створення нових бізнес-задач у віддаленому режимі, через браузер на будь-якому пристрої (смартфон, планшет).

Для запуску проєкту потрібен комп'ютер з встановленою мовою програмування Golang версії не нижче за 1.19, що здатен підтримувати контейнери Docker. Наприклад, персональний IBM-сумісний комп'ютер з оперативною пам'яттю 16 Гб, монітором, клавіатурою та маніпулятором типу «миша», що функціонує під управлінням операційної системи Ubuntu.

5.2 Інструкція з експлуатації програми

Спочатку треба встановити відповідну версію мови програмування Golang та Docker. Також буде корисним перевірити налаштування, що знаходяться у папці settings й змінити їх за потреби. Створити .env файл можливо на основі файлу .env.example шляхом перейменування та зміни налаштувань у самому файлі з використанням текстового редактора.

Щоб запустити програму потрібно скомпілювати її за допомогою команди “go build .” та запустити скомпільований файл через термінал. Це також треба мати на увазі при спробі запустити проєкт через Docker. Приклад docker-compose файлу є у папці deploy/local.

Тестування програми можливе з використанням вбудованих засобів тестування Golang через команду “go test” або засобами програми Postman або її аналогів.

Для коректної роботи програми окрім запуску самої програми потрібно ще запустити Docker-контейнер з PostgreSQL, приклад якого є у файлі `docker-compose` у папці `deploy/local`.

Структура проєкту (рис. 5.1) містить 9 папок: “`cmd`”, “`config`”, “`deploy`”, “`internal`”, “`migrations`”, “`pkg`”, “`postman`”, “`server`”, “`settings`”. Окрім цього в корені є 6 файлів: “`main.go`”, “`Dockerfile`”, “`.env.example`”, “`.gitignore`”, “`go.mod`”, “`go.sum`”.

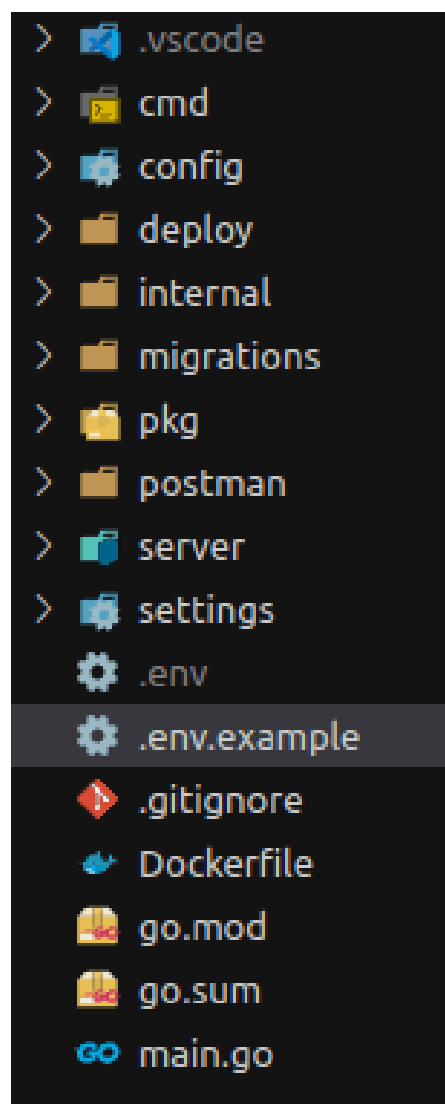


Рисунок 5.1 – Структура кореневої папки програми

В таблиці 5.1 наведено детальний опис елементів кореневої папки проєкту.

Таблиця 5.1 – Опис складових кореневої папки проєкту

Назва елементу	Опис
1	2
cmd	В цій папці знаходиться ДІ та функції, що потрібні для запуску проєкту
config	В цій папці знаходяться структури з налаштуваннями проєкту
deploy	В цій папці знаходяться файли, які потрібні для запуску необхідних для роботи програми в цілому, контейнерів
internal	В цій папці знаходиться внутрішній функціонал проєкту (його сервіси)
migrations	В цій папці знаходяться міграції для СКБД PostgreSQL
pkg	В цій папці знаходяться зовнішні залежності проєкту. Як правило це бібліотеки, що потрібні для роботи проєкту
postman	В цій папці знаходяться файли для тестування проєкту через програму Postman
server	В цій папці знаходиться реалізація серверної частини проєкту
settings	В цій папці знаходяться файли налаштувань які можна, але не рекомендовано змінювати
.gitignore	Цей файл зберігає в собі назви та шаблони файлів, що не повинні потрапити у GitHub
Dockerfile	В цьому файлі містяться налаштування для запуску проєкта у Docker-контейнері
main.go	В цьому файлі зберігається код запуску проєкту
.env.example	В цьому файлі містяться змінні налаштування проєкту
go.mod	В цьому файлі знаходяться залежності проєкту від інших бібліотек. Окрім цього там зберігається системна інформація щодо назви проєкту та версії Golang
go.sum	В цьому файлі знаходяться хеші залежностей проєкту від інших бібліотек

В папці `internal`. містяться наступні папки та елементи:

- папка `user` – містить у собі реалізацію сервісу `User` й функції для аутентифікації та роботи з профілем користувача;
- папка `tasks` – містить у собі реалізацію сервісу `Tasks` й функції для створення та оновлення бізнес-задач.

В папці `pkg` містяться наступні папки та елементи:

- папка `bcrypt` – містить у собі функції для хешування паролів, а також для їх подальшої перевірки;
- папка `cache` – містить у собі функції для збереження та отримання даних з кешу;
- папка `jwt` – містить у собі функції для генерації та валідації `jwt` токенів;
- папка `otp` – містить у собі функції для генерації та валідації `otp` кодів;
- папка `postgres` – містить у собі функції для роботи з СКБД `PostgreSQL`.

Після запуску програми необхідно перейти в програму `Postman`, завантажити туди колекцію з папки `postman` та перевірити, що адреса, яка вказана у файлах конфігурації програми (за замовчуванням це `0.0.0.0:1140`) збігатися з адресою в запитах `Postman`. Після цього користувачу буде доступний список запитів, наведений на рисунку 5.2.

Далі користувач повинен зареєструватись у системі використовуючи запит `Register`. Запит на реєстрацію нового користувача представлений на рисунку 5.3.

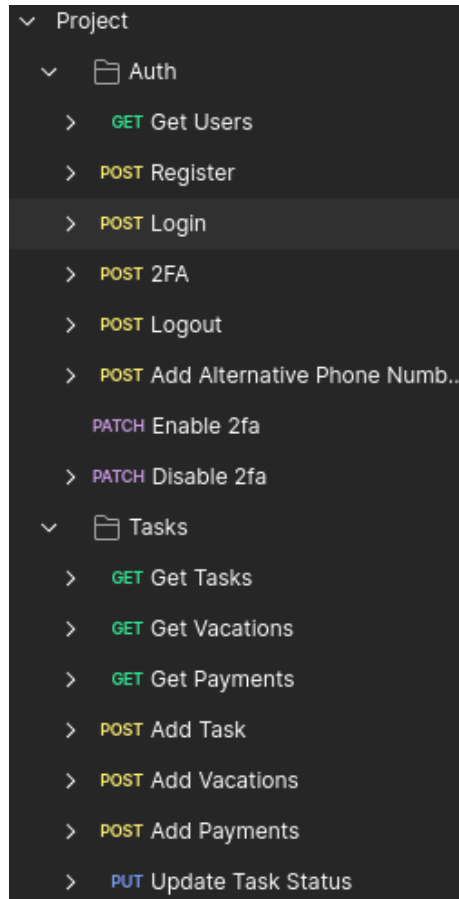


Рисунок 5.2 – Список доступних запитів

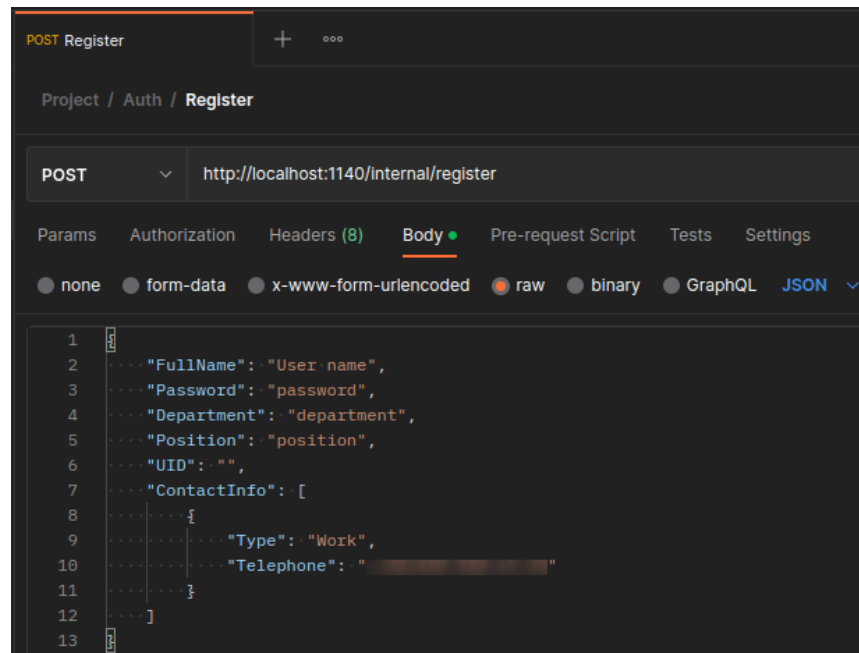


Рисунок 5.3 – Запит на реєстрацію нового користувача

Після реєстрації або якщо користувач вже зареєстрований у системі йому потрібно увійти у свій акаунт. Для цього потрібно використати запит Login, який наведений на рисунку 5.4.

Якщо користувач увімкнув 2FA, то йому знадобиться використати також запит 2FA, що наведений на рисунку 5.5.

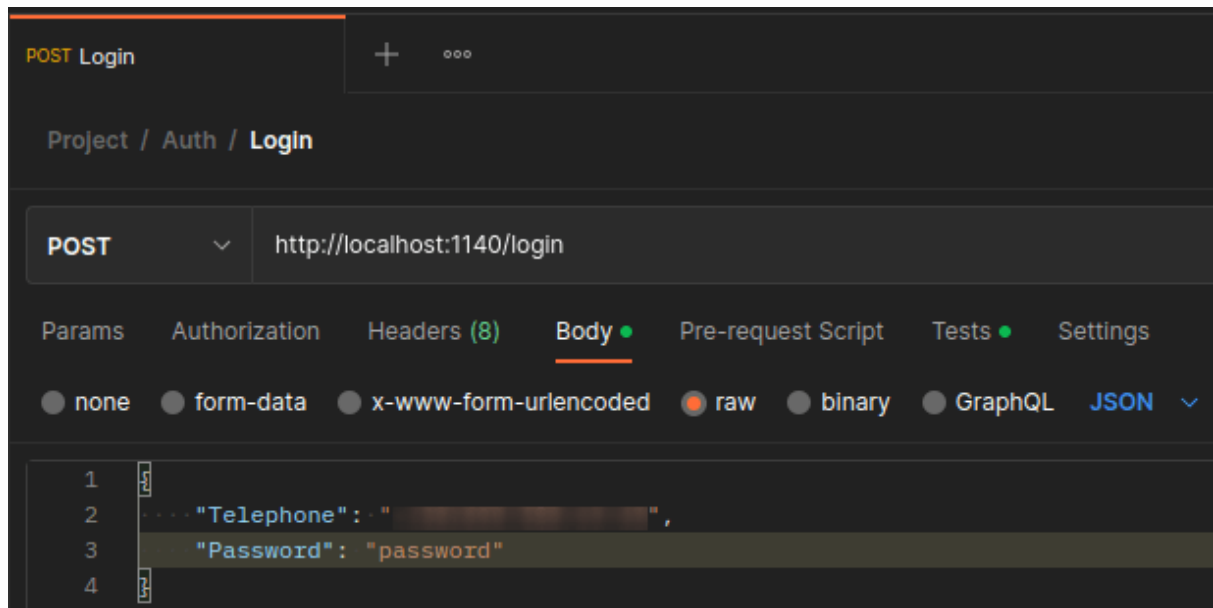


Рисунок 5.4 – Запит Login

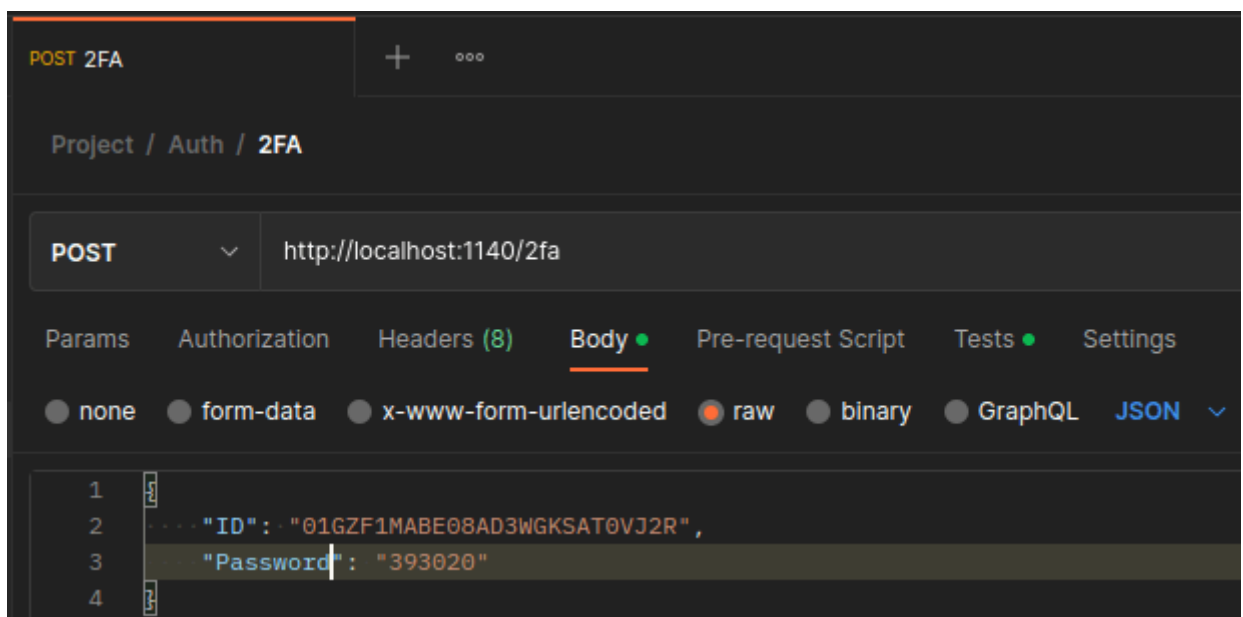


Рисунок 5.5 – Запит 2FA

Якщо користувач не увімкнув 2FA при реєстрації, але йому потрібно увімкнути її, то потрібно використати запит Enable 2fa, що доступний лише після входу до акаунту. Цей запит наведений на рисунку 5.6.

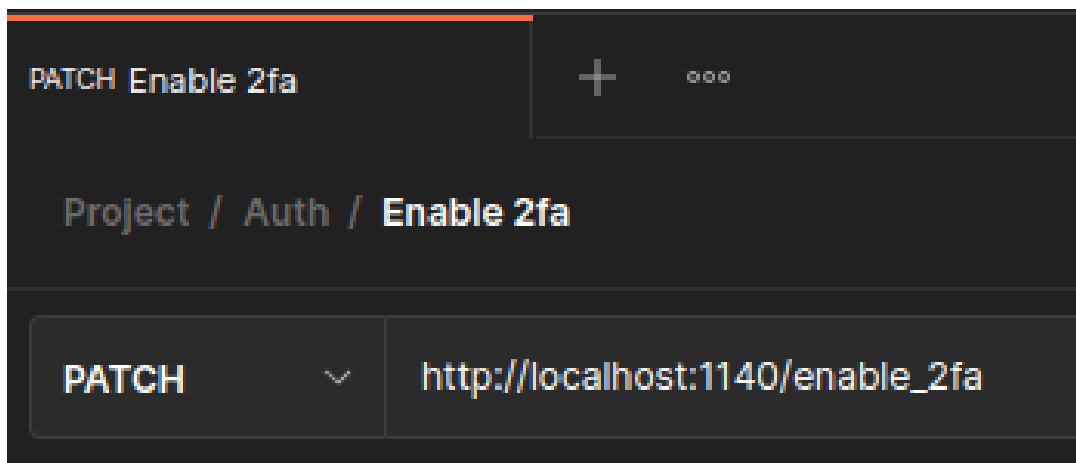


Рисунок 5.6 – Запит Enable 2fa

Якщо користувачу не потрібна 2FA, то її можна виключити у запиті Disable 2fa, що доступний лише після входу до акаунту. Цей запит наведений на рисунку 5.7.

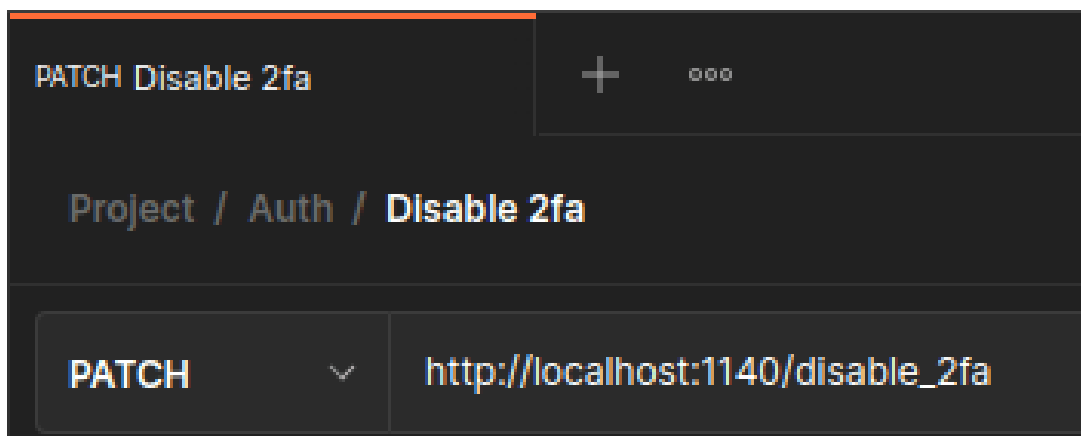


Рисунок 5.7 – Запит Disable 2fa

Якщо користувач хоче додати альтернативний номер телефону для входу в акаунт, то він може зробити це у запиті Add Alternative Phone Number,

що доступний лише після перевірки на існування IP-адреси у списку дозволених. Цей запит наведений на рисунку 5.8.

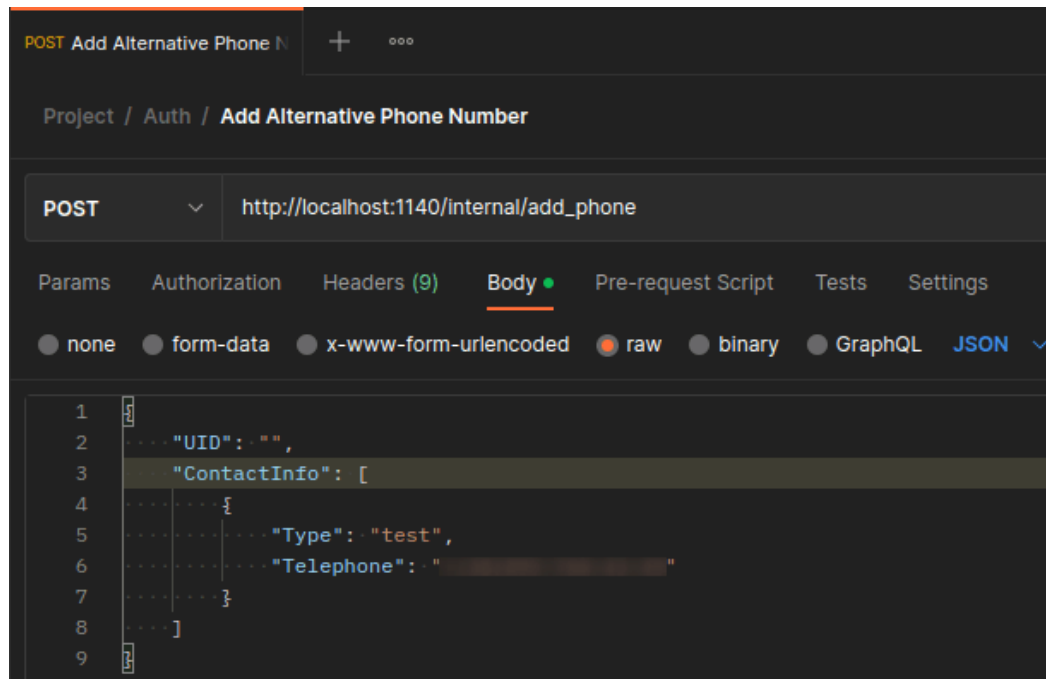


Рисунок 5.8 – Запит Add Alternative Phone Number

Якщо користувач хоче вийти зі свого акаунту, то він може зробити це у запиті Logout, що доступний лише після входу до акаунту. Цей запит наведений на рисунку 5.9.

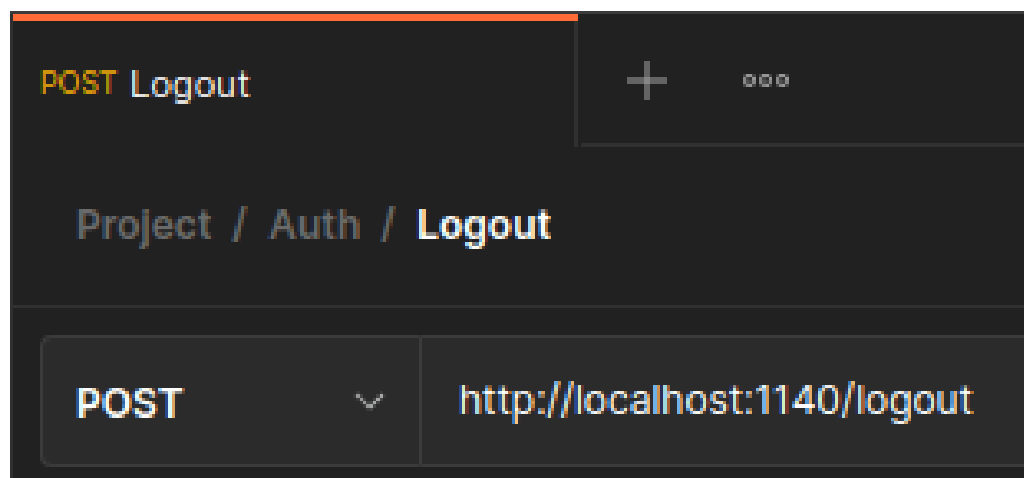


Рисунок 5.9 – Запит Logout

Для отримання списку бізнес-задач, що призначені користувачеві, необхідно використати запит Get Tasks, що доступний лише після входу до акаунту. Цей запит наведений на рисунку 5.10.

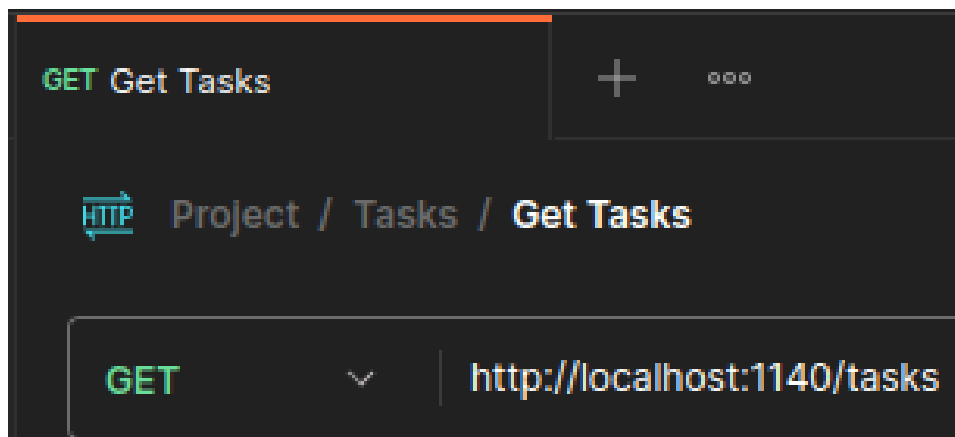


Рисунок 5.10 – Запит Get Tasks

Для отримання списку запитів на відпустку, що призначені користувачеві, необхідно використати запит Get Vacations, що доступний лише після входу до акаунту. Цей запит наведений на рисунку 5.11.

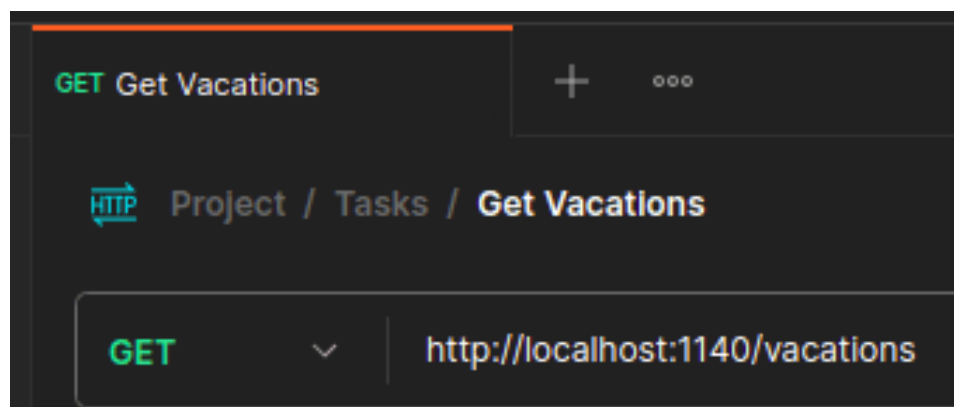


Рисунок 5.11 – Запит Get Vacations

Для отримання списку запитів на оплату послуги, що призначені користувачеві, необхідно використати запит Get Payments, що доступний лише після входу до акаунту. Цей запит наведений на рисунку 5.12.

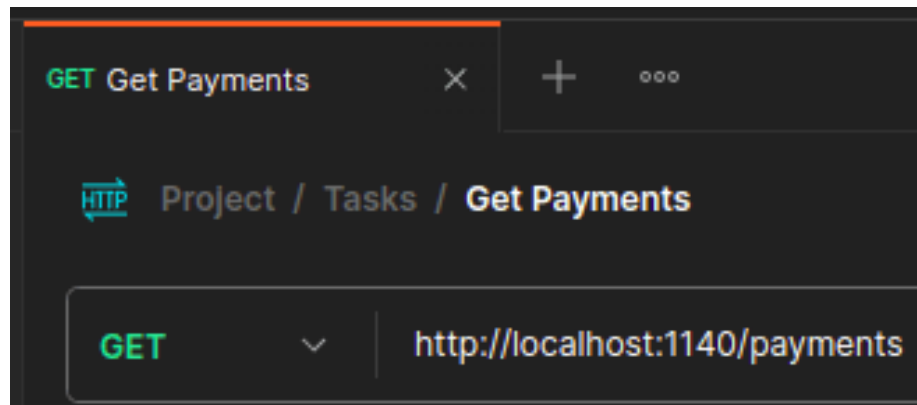


Рисунок 5.12 – Get Payments

Для створення нових бізнес-задач необхідно використати запит Add Task, що доступний лише після перевірки на існування IP-адреси у списку дозволених. Цей запит наведений на рисунку 5.13.

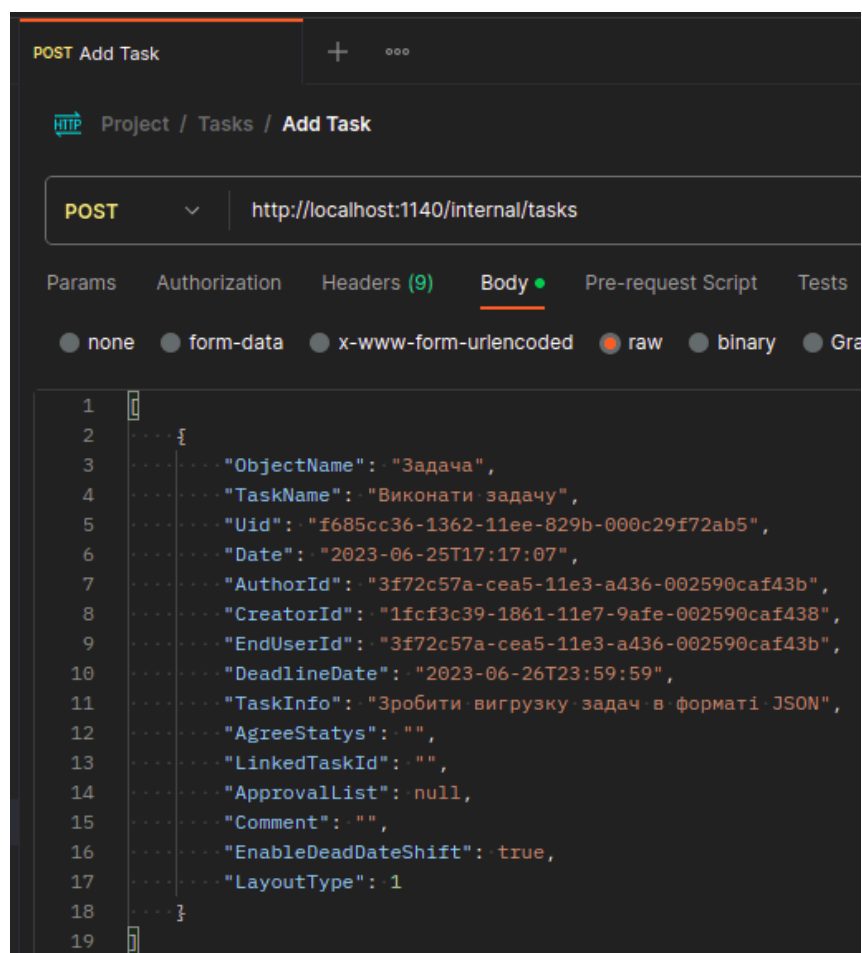


Рисунок 5.13 – Запит Add Task

Для створення нових запитів на відпустку необхідно використати запит Add Vacations, що доступний лише після перевірки на існування ІР-адреси у списку дозволених. Цей запит наведений на рисунку 5.14.

```

1  {
2  }
3
4  "ObjectName": "Согласование отпуска",
5  "TaskName": "Утвердить директором департамента",
6  "UserId": "4d20f018-daa5-11ed-851d-002590caf439",
7  "Date": "2023-02-20T13:04:00",
8  "Author": " ",
9  "AuthorId": "99ea4e5c-665a-11e0-b2eb-005056ae1b1c",
10 "EndUserId": null,
11 "EndUserId": "3f72c57a-cea5-11e3-a436-002590caf43b",
12 "DeadlineDate": "2023-02-20T16:04:00",
13 "TaskInfo": "Прому предоставить дополнительный отпуск в связи с днем рождения с 22.02.2023 по 23.02.2023 и Основной ежегодный отпуск на один день 24.02.2023г.",
14 "AgreeStatus": "",
15 "LinkedTaskId": "",
16 "Approvallist": null,
17 "Comment": "",
18 "EnableDeadDateShift": true,
19 "LayoutType": 1,
20 "PeriodStart": "2023-07-01T00:00:00",
21 "PeriodEnd": "2023-07-10T00:00:00",
22 "HolidayMaker": " ",
23 "Substitutional": " "
24 }

```

Рисунок 5.14 – Запит Add Vacations

Для створення нових запитів на оплату послуги необхідно використати запит Add Payments, що доступний лише після перевірки на існування ІР-адреси у списку дозволених. Цей запит наведений на рисунку 5.15.

```

1  {
2  }
3
4  "ObjectName": "Согласование счета на оплату поставщика",
5  "TaskName": "Проверить исполнителем по договору",
6  "UserId": "4d20f018-daa5-11ed-851d-002590caf439",
7  "Date": "2023-04-14T12:18:27",
8  "Author": " ",
9  "AuthorId": "3f72c57a-cea5-11e3-a436-002590caf43b",
10 "EndUserId": "3f72c57a-cea5-11e3-a436-002590caf43b",
11 "DeadlineDate": "2023-04-18T12:18:27",
12 "TaskInfo": "Оплата  зр.  від 14 квітня 2023 р.",
13 "AgreeStatus": "",
14 "LinkedTaskId": "",
15 "Approvallist": null,
16 "Comment": "В связи с внеплановым переводом  с  на  / Состояние: Согласован дата: 18.04.2023 14:58:26 /",
17 "EnableDeadDateShift": true,
18 "LayoutType": 1,
19 "Kontragent": " ",
20 "Organization": " ",
21 "Sum": "1200",
22 "PaymentDate": "14.04.2023 0:00:00",
23 "PaymentPurpose": "Оплата  від 14 квітня 2023 р."
24 }
25

```

Рисунок 5.15 – Запит Add Payments

Для оновлення статусу наявних бізнес-задач необхідно використати запит Update Task Status, що доступний лише після входу до акаунту. Цей запит наведений на рисунку 5.16.

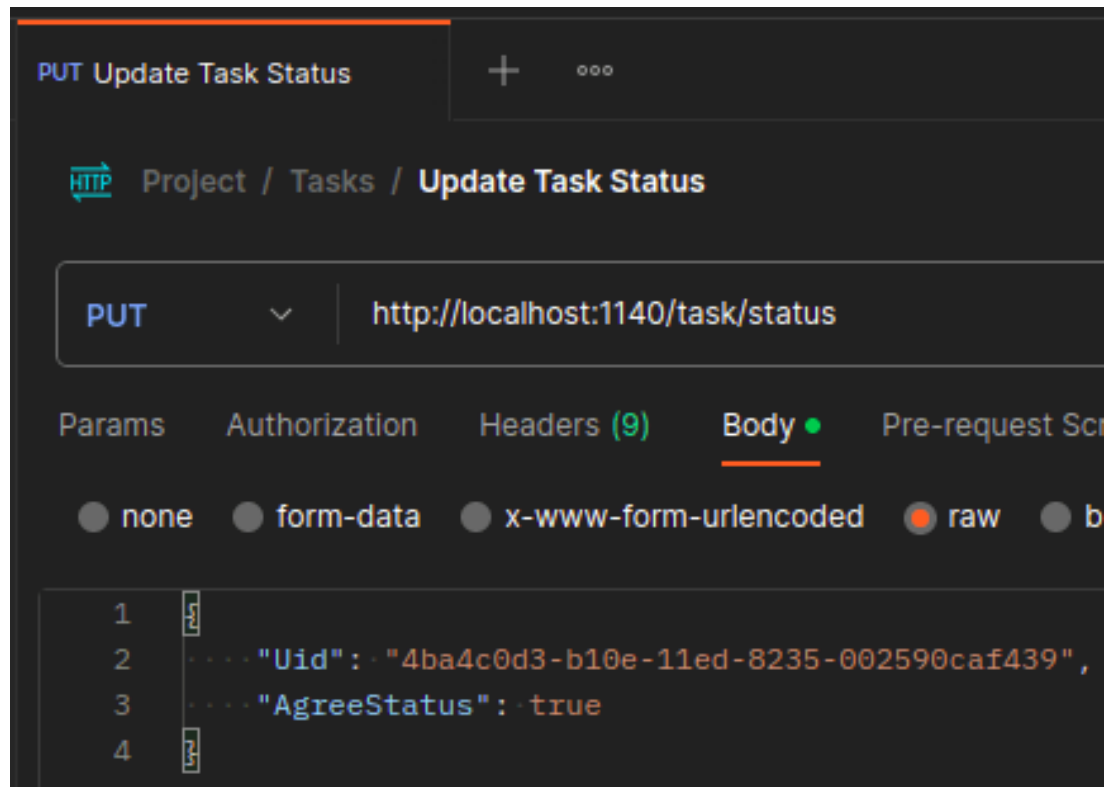


Рисунок 5.16 – Запит Update Task Status

5.3 Висновки до розділу 5

В цьому розділі відповідно до мети та задач дипломної роботи було розглянуто особливості запуску та використання розробленого ПЗ, описані системні вимоги, надані базові інструкції з експлуатації програми.

ВИСНОВКИ

Як показали проведені дослідження, ЦТ фундаментально вплинула на бізнес-діяльність компаній. Тим не менш, ЦТ – це дуже неоднозначна тенденція, що потребує впровадження нових методів, моделей та засобів.

В результаті виконання дипломної роботи було створено програмний застосунок, який надає керівнику компанії онлайн платформу для віддаленого моніторингу та контролю бізнес-діяльності фахівців компанії, які використовують систему BAS ERP.

Наукова новизна роботи полягає в тому, що розроблено методику використання МЗЦТ для оцінки рівня ЦТ компанії, яка дозволяє забезпечити якість процесів ЦТ та намітити подальші шляхи для переходу на вищі рівні ЦТ.

Практична цінність розробки полягає в тому, що впровадження розробленої програми дозволить спростити та пришвидшити процес прийняття управлінських рішень керівниками компанії та дозволить оперативно реагувати на результати бізнес-діяльності фахівців через браузер на смартфоні або комп'ютері.

Завдяки простій архітектурі та програмним рішенням дана програмна реалізація є легко масштабованою та модифікованою. Вона надає надійний механізм двофакторної аутентифікації, щоб не допустити зловмисників до даних акаунту користувачів, що особливо важливо при роботі в мережі Інтернет.

Проведене тестування показало, що створений програмний застосунок дозволяє виконати ЦТ процесу моніторингу бізнес-діяльності фахівців компанії та перейти від рівня Розширеної цифровізації до Розумної цифровізації згідно з обраною 4-рівневою моделлю зрілості процесів ЦТ.

В подальшій роботі функціонал програми можливо розширити шляхом додавання хмарного сховища для зберігання цифрових даних компанії.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Stjepić, A.M. Mastering digital transformation through business process management: Investigating alignments, goals, orchestration, and roles / A.M. Stjepić, L. Ivančić, D.S. Vugec // Journal of Entrepreneurship, Management and Innovation. – 2020. – Vol.16(1). – P. 41-73.
2. The 4 main areas of the digital transformation process [Electronic resource] / T. Vaughan. – Access mode: <https://www.poppulo.com/blog/what-are-the-4-main-areas-of-digital-transformation>.
3. Business process transformation: the ultimate guide [Electronic resource]. – Access mode: <https://www.digital-adoption.com/business-process-transformation/>.
4. What is business process transformation? [Electronic resource]. – Access mode: https://www.softwareag.com/en_corporate/resources/process-management/article/business-process-transformation.html.
5. Business process transformation - an ultimate guide for 2024 [Electronic resource]. – Access mode: <https://kissflow.com/workflow/bpm/business-process-transformation/#:~:text=Business%20Process%20Transformation%20involves%20radically,a%20new%20digital%20transformation%20implementation>.
6. What are the 4 main areas of digital transformation? [Electronic resource]. – Access mode: <https://www.enate.io/blog/4-main-areas-of-digital-transformation>.
7. How business process digitization can transform your company [Electronic resource]. – Access mode: <https://planergy.com/blog/business-process-digitization/>.
8. Digitize business processes to transform your work, team and organization [Electronic resource]. – Access mode: <https://praxie.com/digital-organization-business-process-transformation-through-digitization/>.

9. Process digitalization in digital transformation [Electronic resource] / P. Robledo. – Access mode: <https://albatian.com/en/blog-ingles/process-digitalization-in-digital-transformation/>.
10. Key stages in the successful digitization of business processes [Electronic resource] / C. Sarah. – Access mode: <https://lemonlearning.com/blog/digitization-of-business-processes>.
11. How business process digitization can transform your company [Electronic resource] / L. Del Vecchio. – Access mode: <https://planergy.com/blog/business-process-digitization/>.
12. Огляд Bimp – українського сервісу управління ресурсами та фінансами [Електронний ресурс] – Режим доступу: <https://itc.ua/ua/articles/oglyad-bimp-ukrayinskogo-servisu-upravlinnya-resursamy-ta-finansamy/>.
13. Quincefin [Електронний ресурс] – Режим доступу: <https://quincefin.com/>.
14. Finmap [Електронний ресурс] – Режим доступу: <https://finmap.online/>.
15. Pipeliner CRM [Electronic resource]. – Access mode: <https://www.pipelinersales.com/>.
16. Bookkeeper [Електронний ресурс] – Режим доступу: <https://bookkeeper.kiev.ua/>.
17. BAS [Електронний ресурс] – Режим доступу: <https://www.bas-soft.eu/>.
18. CMMI Institute: capability maturity model integration (CMMI), [Electronic resource]. – Access mode: <https://www.cmmiinstitute.com/>.
19. Digital transformation: a multidisciplinary reflection and research agenda / [P. C. Verhoef, T. Broekhuizen, Y. Bart et al.] // Journal of Business research. – 2021. – Vol. 122. – P. 889-901.

20. Industrie 4.0 maturity index. Managing the digital transformation of companies (acatech STUDY) / [G. Schuh, R. Anderl, R. Dumitrescu et al.] // Computer Science, Business. – 2020. – 64 p.

21. Ngereja, B. Outcome specification: digital transformation maturity model (DTMM) [Electronic resource] / B. Ngereja, B. Hussein, C. Wolff. – Access mode: <https://prodit-alliance.eu/>.

22. A strategic framework for digital maturity of design and construction through a systematic review and application / [S. Perera, X. Jin, P. Das et al.] // J. Ind. Inf. Integr. – 2023. - Vol. 31. – P. 1-17.

23. Mettler, T. Towards a classification of maturity models in information systems / T. Mettler, P. Rohner, R. Winter // Management of the interconnected world. – Physica-Verlag HD. – 2010. – P. 333-340.

24. What is maturity model? Definition, types and use [Electronic resource]. – Access mode: <https://chisellabs.com/glossary/what-is-the-maturity-model-and-how-to-use-it/>.

25. What are maturity models? (With definition, types and benefits) [Electronic resource]. – Access mode: <https://www.indeed.com/career-advice/career-development/what-are-maturity-models>.

26. Schräge, M. How the wrong KPIs doom digital transformation / M. Schräge, V. Muttreja, A. Kwan // MIT Sloan manag. rev. – 2022. – Vol. 63, № 3. – P. 35-40.

27. Industry 4.0 readiness. Online self-check for businesses [Electronic resource]. – Access mode: <https://www.industrie40-readiness.de/?lang=en>.

28. Makhsud, U. Identification and authentication / U. Makhsud // International journal of academic pedagogical research (IJAPR). – 2021. – №5. – C. 39–47.

29. A review on authentication methods / [S. Z. S. Idrus, E. Cherrier, C. Rosenberger, J.-J. Schwartzmann] // Australian journal of basic and applied sciences. – 2013. – P. 1-13.

30. Shepherd, J. The ultimate authentication playbook [Electronic resource] / J. Shepherd. – Access mode: <https://www.okta.com/blog/2019/02/the-ultimate-authentication-playbook/>.
31. OAuth [Electronic resource]. – Access mode: <https://oauth.net/about/introduction/>.
32. SAML [Electronic resource]. – Access mode: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security.
33. Kerberos [Electronic resource]. – Access mode: <https://www.kerberos.org/software/tutorial.html>.
34. Zeilenga, K. Lightweight directory access protocol (LDAP): technical specification road map [Electronic resource] / K. Zeilenga. – Access mode: <https://www.rfc-editor.org/rfc/rfc4510>.
35. M'Raihi, D. TOTP: time-based one-time password algorithm [Electronic resource] / [D. M'Raihi, S. Machani, M. Pei, J. Rydell]. – Access mode: <https://www.rfc-editor.org/rfc/rfc6238>.
36. Plata, I. T. Application of time-based one time password (TOTP) algorithm for human resource e-leave tracking web app / I. T. Plata, J. L. Calpito // International journal of scientific & technology research. – 2020. – P. 1-8.
37. Як і для чого вивчати Golang. Переваги і недоліки мови [Електрон. ресурс]. – Режим доступу: <https://dou.ua/forums/topic/41933/>.
38. Visual studio code & Golang <https://medium.com/@fightermanlviv/visual-studio-code-golang-4b6f89f020dc>.
39. Go in Visual studio code [Electronic resource]. – Access mode: <https://code.visualstudio.com/docs/languages/go>.
40. Kahlert, T. Visual studio code tips & tricks / T. Kahlert, K. Giza. – Deutschland: Microsoft Deutschland GmbH. – 2016. – Vol. 1. – 26 p.
41. PostgreSQL: The world's most advanced open source Relational database [Electronic resource]. – Access mode: <https://www.postgresql.org/>.

42. What is PostgreSQL? [Electronic resource]. – Access mode: <https://www.postgresqltutorial.com/postgresql-getting-started/what-is-postgresql/>.

43. What is PostgreSQL? [Electronic resource]. – Access mode: <https://aws.amazon.com/ru/rds/postgresql/what-is-postgresql/>.

44. PostgreSQL vs. SQL Server – everything you need to know [Electronic resource]. – Access mode: <https://www.astera.com/knowledge-center/postgresql-sql-server/#:~:text=Performance%3A%20SQL%20Server%20offers%20better,more%20flexible%20than%20SQL%20Server.>

45. Swagger vs Postman | Top 10 differences you should know [Electronic resource]. – Access mode: <https://testsigma.com/blog/swagger-vs-postman/>.

ДОДАТОК А
Текст програми

A.1 Текст файла user.go

```

func (u *Service) Register(ctx context.Context, req
*Register) error {
    id := ulid.Make().String()
    contactInfo := make([]postgres.ContactInfo, 0,
len(req.ContactInfo))
    for _, info := range req.ContactInfo {
        phoneNumber := phonenumber.Parse(info.Phone, "")
        if phoneNumber == "" {
            return ErrInvalidPhone
        }

        contactInfo = append(contactInfo,
postgres.ContactInfo{
            UserID: id,
            Type:    info.Type,
            Phone:   "+" + phoneNumber,
        })
    }

    encryptedPassword, err := u.bcrypt.Encode(ctx,
req.Password)
    err = u.db.CreateUser(ctx, &postgres.User{
        ID:            id,
        Department:    req.Department,
        Position:      req.Position,
        FullName:      req.FullName,
        ForeignID:     req.ForeignID,
        Password:      encryptedPassword,
        ContactInfo:   contactInfo,
    })

    for i := range contactInfo {
        err = u.db.AddContactInfo(ctx, &contactInfo[i])
    }

    return nil
}

func (u *Service) Login(ctx context.Context, req *Login)
(string, bool, error) {
    phoneNumber := phonenumber.Parse(req.Phone, "")
    if phoneNumber == "" {

```

```

        return "", false, ErrInvalidPhone
    }

    user, err := u.db.GetUserByPhone(ctx,
"+"+phoneNumber)
    if !u.bcrypt.Validate(ctx, user.Password,
req.Password) {
        return "", false, ErrInvalidPassword
    }

    if user.OTPSecret != "" {
        id := ulid.Make().String()
        u.cache.Set(ctx, id, OTPData{UserID: user.ID,
OTPSecret: user.OTPSecret}, 3*time.Minute)
        return id, true, nil
    }

    token, err := u.jwt.GenerateToken(ctx, user.ID)
    return token, false, err
}

func (u *Service) AddAlternativeNumber(ctx
context.Context, req *AddAlternativeNumber) error {
    id, err := u.db.GetUserIDByForeignID(ctx,
req.ForeignID)

    for _, info := range req.ContactInfo {
        phoneNumber := phonenumbers.Parse(info.Phone, "")
        if phoneNumber == "" {
            return ErrInvalidPhone
        }

        err := u.db.AddContactInfo(ctx,
&postgres.ContactInfo{
            UserID: id,
            Type: info.Type,
            Phone: "+" + phoneNumber,
        })
    }

    return nil
}

```

```

func (u *Service) OTPCheck(ctx context.Context, req
*TwoFA) (string, error) {
    data, found := u.cache.Get(ctx, req.ID)
    if !found {
        return "", ErrInvalid2FAID
    }

    u.cache.Delete(ctx, req.ID)
    ok := u.otp.ValidateKey(ctx, req.OTPPassword,
data.OTPSecret)
    if !ok {
        return "", ErrInvalidOtpCode
    }

    return u.jwt.GenerateToken(ctx, data.UserID)
}

func (u *Service) EnableTwoFA(ctx context.Context,
userID, token string) ([]byte, error) {
    image, secret, err := u.otp.GenerateKey(ctx,
userID)
    err = u.db.EnableOTP(ctx, userID, secret)
    u.Logout(ctx, token)
    return image, nil
}

func (u *Service) DisableTwoFA(ctx context.Context,
userID, token string) error {
    err := u.db.DisableOTP(ctx, userID)
    u.Logout(ctx, token)
    return nil
}

```

A.2 Текст файла auth.go

```

func AuthMiddleware(auth *jwt.JWT) echo.MiddlewareFunc {
    return echojwt.WithConfig(echojwt.Config{
        ErrorHandler: func(ctx echo.Context, err error)
error {
            var errorText string

            if errors.Is(err, jwt.ErrExpired) ||
errors.Is(err, jwt.ErrUserGet) {

```

```

        errorText = err.Error()
    } else {
        errorText = "Failed to validate token"
    }

    return ctx.JSON(http.StatusUnauthorized,
map[string]any{
        "error": errorText,
    })
},

    ParseTokenFunc: func(ctx echo.Context,
accessToken string) (any, error) {
        userID, err :=
auth.ValidateToken(ctx.Request().Context(),
accessToken)

        ctx.Set("token", accessToken)
        ctx.Set("userID", userID)
        return userID, nil
    },
))
}

```

A.3 Текст файла app.go

```

package cmd

import (
    "remote_monitoring_and_controlling/config"
    "remote_monitoring_and_controlling/internal/user"
    "remote_monitoring_and_controlling/pkg/bcrypt"
    "remote_monitoring_and_controlling/pkg/jwt"
    "remote_monitoring_and_controlling/pkg/otp"
    "remote_monitoring_and_controlling/pkg/postgres"
    "remote_monitoring_and_controlling/server"
    "remote_monitoring_and_controlling/server/handlers"

    "go.uber.org/fx"
)

func Exec() fx.Option {
    return fx.Options(

```

```

    fx.Provide(
        config.Get,
        bcrypt.NewBcrypt,
        otp.NewOTP,

        postgres.NewPostgres,
        fx.Annotate(
            annotationDupl[postgres.Postgres],
            fx.As(new(user.DB)),
        ),

        jwt.NewJWT,
        user.NewUserService,
        handlers.NewHandler,
        server.NewHTTPServer,
    ),
    fx.Invoke(
        prepareHooks,
    ),
)
}

func annotationDupl[T any](v *T) *T {
    return v
}

func prepareHooks(server *server.HTTPServer, postgres
*postgres.Postgres, lc fx.Lifecycle) {
    lc.Append(fx.Hook{OnStart: postgres.Start, OnStop:
postgres.Stop})
    lc.Append(fx.Hook{OnStart: server.Start, OnStop:
server.Stop})
}

```

A.4 Текст файла postgres.go

```

package postgres

import (
    "context"
    "errors"
    "fmt"

```

```

    "remote_monitoring_and_controlling/config"

    "github.com/golang-migrate/migrate/v4"
    mpostgres "github.com/golang-
migrate/migrate/v4/database/postgres"
    _ "github.com/golang-
migrate/migrate/v4/source/file" // Needs for correct
migrations
    "gorm.io/driver/postgres"
    "gorm.io/gorm"
    "gorm.io/gorm/clause"
    glog "gorm.io/gorm/logger"
)

type Postgres struct {
    db *gorm.DB
    cfg *config.PostgresConfig
}

func (p *Postgres) GetUsers(ctx context.Context)
([]User, error) {
    var result []User
    return result,
p.db.WithContext(ctx).Model(&User{}).Preload("ContactIn
fo").
        Select([]string{
            "users.id",
            "users.department",
            "users.position",
            "users.full_name",
        }).Find(&result).Error
}

func (p *Postgres) GetUserByPhone(ctx context.Context,
phone string) (User, error) {
    var result User
    return result, p.db.WithContext(ctx).Model(&User{}).
        Select([]string{
            "users.id",
            "users.department",
            "users.position",
            "users.full_name",

```

```

        "users.password",
        "users.otp_secret",
    })).
    Joins("INNER JOIN contact_info ON users.id =
contact_info.user_id").
    Where("contact_info.phone = ?", phone).
    First(&result).Error
}

func (p *Postgres) CreateUser(ctx context.Context, user
*User) error {
    return
p.db.Model(user).WithContext(ctx).Create(user).Error
}

func (p *Postgres) AddContactInfo(ctx context.Context,
contactInfo *ContactInfo) error {
    return
p.db.Model(contactInfo).WithContext(ctx).Clauses(clause
.OnConflict{
        Columns:      []clause.Column{{Name: "user_id"},
{Name: "phone"}},
        DoUpdates:
clause.AssignmentColumns([]string{"type"}),
    }).Create(contactInfo).Error
}

func (p *Postgres) EnableOTP(ctx context.Context,
userID, secret string) error {
    return
p.db.Model(&User{}).WithContext(ctx).Where("id = ?",
userID).Update("otp_secret", secret).Error
}

func (p *Postgres) DisableOTP(ctx context.Context,
userID string) error {
    return
p.db.Model(&User{}).WithContext(ctx).Where("id = ?",
userID).Update("otp_secret", "").Error
}

func (p *Postgres) DeleteUser(ctx context.Context,
userID string) error {

```

```

        return
    p.db.Model(&User{}).WithContext(ctx).Where("id = ?",
    userID).Delete(&User{}).Error
}

func (p *Postgres) Start(ctx context.Context) (err error)
{
    p.db, err = newDB(p.cfg.DSN, p.cfg.MigrationURL)
    return
}

func (p *Postgres) Stop(ctx context.Context) error {
    sql, err := p.db.WithContext(ctx).DB()
    if err != nil {
        return fmt.Errorf("Postgres.Stop get database
error: %w", err)
    }

    err = sql.Close()
    if err != nil {
        return fmt.Errorf("Postgres.Stop connection
close error: %w", err)
    }

    return nil
}

func NewPostgres(cfg *config.Config) *Postgres {
    return &Postgres{
        cfg: &cfg.PostgresConfig,
    }
}

func newDB(dsn, migrationsURL string) (*gorm.DB, error)
{
    client, err := gorm.Open(postgres.Open(dsn),
    &gorm.Config{
        Logger: glog.Default.LogMode(glog.Silent),
    })
    if err != nil {
        return nil, fmt.Errorf("Postgres.newDB gorm
connection open error: %w", err)
    }
}

```

```

        if err := migrations(client, migrationsURL); err !=
nil {
            return nil, fmt.Errorf("Postgres.newDB gorm
migrations error: %w", err)
        }

        return client, nil
    }

func migrations(client *gorm.DB, migrationsURL string)
error {
    db, err := client.DB()
    if err != nil {
        return fmt.Errorf("Postgres.migrations failed to
get sql connection: %w", err)
    }

    driver, err := mpostgres.WithInstance(db,
&mpostgres.Config{})
    if err != nil {
        return fmt.Errorf("Postgres.migrations create
migration instance error: %w", err)
    }

    m, err := migrate.NewWithDatabaseInstance(
        migrationsURL,
        "postgres",
        driver,
    )
    if err != nil {
        return fmt.Errorf("Postgres.migrations migration
init error: %w", err)
    }

    if err := m.Up(); err != nil && !errors.Is(err,
migrate.ErrNoChange) {
        return fmt.Errorf("Postgres.migrations migration
error: %w", err)
    }

    return nil
}

```

A.5 Текст файла jwt.go

```

package jwt

import (
    "context"
    "encoding/base64"
    "errors"
    "fmt"
    "time"

    "remote_monitoring_and_controlling/config"
    "remote_monitoring_and_controlling/settings"

    "remote_monitoring_and_controlling/pkg/bcrypt"
    "remote_monitoring_and_controlling/pkg/cache"

    "github.com/dchest/uniuri"
    "github.com/golang-jwt/jwt/v5"
    "github.com/sethvargo/go-password/password"
)

type JWT struct {
    secret string

    saltCache *cache.Cache[string]
}

func (j *JWT) GenerateToken(ctx context.Context, userID
string) (accessToken string, err error) {
    token := jwt.New(jwt.SigningMethodHS512)

    claims := make(jwt.MapClaims)
    claims["user"] = userID
    claims["iat"] = time.Now().Unix()
    token.Claims = claims

    salt, err := password.Generate(10, 3, 3, false, true)
    if err != nil {
        salt = uniuri.NewLen(10)
    }

```

```

        accessToken, err =
token.SignedString([]byte(j.secret + salt))
        if err != nil {
            return "", fmt.Errorf("Error while generating
token string: %w", err)
        }

        j.saltCache.Set(ctx, accessToken, salt,
settings.TokenDefaultExpiration)
        return
    }

func (j *JWT) ValidateToken(ctx context.Context,
tokenString string) (string, error) {
    token, err := jwt.Parse(tokenString, func(token
*jwt.Token) (any, error) {
        if _, ok :=
token.Method.(*jwt.SigningMethodHMAC); !ok {
            return nil, errors.New("Invalid token
signing method")
        }

        salt, found := j.saltCache.Get(ctx, tokenString)
        if !found {
            return nil, errors.New("Salt not found")
        }

        return []byte(j.secret + salt), nil
    })
    if err != nil {
        return "", fmt.Errorf("Failed to parse token:
%w", err)
    }

    claims, ok := token.Claims.(jwt.MapClaims)
    if !ok || !token.Valid {
        return "", errors.New("Failed to get claims")
    }

    userID, ok := claims["user"].(string)
    if !ok || userID == "" {
        return "", ErrUserGet
    }

```

```

        return userID, nil
    }

    func (j *JWT) DeleteSalt(ctx context.Context, token
string) {
        j.saltCache.Delete(ctx, token)
    }

    func NewJWT(cfg *config.Config, bcrypt *bcrypt.Bcrypt)
(*JWT, error) {
        accessHash, err :=
bcrypt.Encode(context.Background(),
base64.StdEncoding.EncodeToString([]byte(cfg.JWTSecret)
))
        if err != nil {
            return nil, fmt.Errorf("Failed to encode jwt
access secret")
        }

        return &JWT{secret: accessHash, saltCache:
cache.NewCache[string]()}, nil
    }

```

A.6 Текст файла otp.go

```

package otp

import (
    "bytes"
    "context"
    "fmt"
    "image/png"

    "remote_monitoring_and_controlling/settings"

    "github.com/pquerna/otp"
    "github.com/pquerna/otp/totp"
)

type OTP struct{}

func (t *OTP) GenerateKey(ctx context.Context, userID
string) ([]byte, string, error) {
    key, err := totp.Generate(totp.GenerateOpts{

```

```

        Issuer:      "A remote system for monitoring and
controlling business processes and tasks",
        AccountName: userID,
        Period:      settings.OTPExpiration,
        SecretSize:  settings.SecretSize,
        Digits:      otp.DigitsSix,
        Algorithm:   otp.AlgorithmSHA512,
    })
    if err != nil {
        return nil, "", fmt.Errorf("Failed to generate
totp key: %w", err)
    }

    var buf bytes.Buffer
    img, err := key.Image(200, 200)
    if err != nil {
        return nil, "", fmt.Errorf("Failed to convert
totp key into image: %w", err)
    }

    err = png.Encode(&buf, img)
    if err != nil {
        return nil, "", fmt.Errorf("Failed to encode totp
image into bytes buffer: %w", err)
    }

    return buf.Bytes(), key.Secret(), nil
}

func (t *OTP) ValidateKey(ctx context.Context, key,
secret string) bool {
    return totp.Validate(key, secret)
}

func NewOTP() *OTP {
    return new(OTP)
}

```

A.7 Текст файла config.go

```

package config

import (

```

```

    "sync"

    "github.com/joho/godotenv"
    "github.com/kelseyhightower/envconfig"
)

type Config struct {
    AppConfig
    PostgresConfig
}

type AppConfig struct {
    AppAddr string
    `envconfig:"APP_SERVER_ADDRESS"`
    JWTSecret string `envconfig:"JWT_SECRET"
    required:"true"`
    PermittedIPs []string `envconfig:"PERMITTED_IPS"`
}

type PostgresConfig struct {
    DSN string `envconfig:"DB_DSN"
    required:"true"`
    MigrationURL string `envconfig:"DB_MIGRATION_URL"
    default:"file://migrations"`
}

var (
    once sync.Once
    config *Config
)

func Get() (*Config, error) {
    var err error
    once.Do(func() {
        var cfg Config
        _ = godotenv.Load(".env")

        if err = envconfig.Process("", &cfg); err != nil
    {
        return
    }

    config = &cfg
    })
}

```

```
    return config, err
}
```

A.8 Текст файла handlers/user.go

```
package handlers

import (
    "net/http"
    "remote_monitoring_and_controlling/internal/user"

    "github.com/labstack/echo/v4"
    "github.com/rs/zerolog/log"
)

func (h *Handler) GetUsers(ctx echo.Context) error {
    users, err :=
h.user.GetUsers(ctx.Request().Context())
    if err != nil {
        log.Error().Str("function",
"GetUsers").Err(err).Msg("Failed to get users")
        return ctx.JSON(http.StatusInternalServerError,
newHTTPError(checkErr(err, ErrRegister)))
    }

    return ctx.JSON(http.StatusOK, users)
}

func (h *Handler) Register(ctx echo.Context) error {
    var req user.Register
    if err := ctx.Bind(&req); err != nil {
        log.Error().Str("function",
"Register").Err(err).Msg("Failed to bind user data")
        return ctx.JSON(http.StatusBadRequest,
newHTTPError(ErrBind))
    }

    if err := h.validator.Struct(&req); err != nil {
        log.Error().Str("function",
"Register").Err(err).Msg("Failed to validate user data")
        return ctx.JSON(http.StatusBadRequest,
newHTTPError(ErrBind))
    }
}
```

```

    }

    err    :=    h.user.Register(ctx.Request().Context(),
&req)
    if err != nil {
        log.Error().Str("function",
"Register").Err(err).Msg("Failed to register user")
        return ctx.JSON(http.StatusInternalServerError,
newHTTPError(checkErr(err, ErrRegister)))
    }

    return ctx.NoContent(http.StatusOK)
}

func (h *Handler) Login(ctx echo.Context) error {
    var req user.Login
    if err := ctx.Bind(&req); err != nil {
        log.Error().Str("function",
"Login").Err(err).Msg("Failed to bind user data")
        return ctx.JSON(http.StatusBadRequest,
newHTTPError(ErrBind))
    }

    if err := h.validator.Struct(&req); err != nil {
        log.Error().Str("function",
"Login").Err(err).Msg("Failed to validate user data")
        return ctx.JSON(http.StatusBadRequest,
newHTTPError(ErrBind))
    }

    token,          twoFAEnabled,          err          :=
h.user.Login(ctx.Request().Context(), &req)
    if err != nil {
        log.Error().Str("function",
"Login").Err(err).Msg("Failed to login user")
        return ctx.JSON(http.StatusInternalServerError,
newHTTPError(checkErr(err, ErrLogin)))
    }

    if twoFAEnabled {
        return ctx.String(http.StatusAccepted, token)
    }

    return ctx.String(http.StatusOK, token)
}

```

```

}

func (h *Handler) AddAlternativeNumber(ctx echo.Context)
error {
    var req user.AddAlternativeNumber
    if err := ctx.Bind(&req); err != nil {
        log.Error().Str("function",
            "AddAlternativeNumber").Err(err).Msg("Failed to bind
            alternative phone number")
        return ctx.JSON(http.StatusBadRequest,
            newHTTPError(ErrBind))
    }

    err :=
h.user.AddAlternativeNumber(ctx.Request().Context(),
    &req)
    if err != nil {
        log.Error().Str("function",
            "AddAlternativeNumber").Err(err).Msg("Failed to add
            alternative phone number")
        return ctx.JSON(http.StatusInternalServerError,
            newHTTPError(checkErr(err, ErrAddPhone)))
    }

    return ctx.NoContent(http.StatusOK)
}

func (h *Handler) TwoFA(ctx echo.Context) error {
    var req user.TwoFA
    if err := ctx.Bind(&req); err != nil {
        log.Error().Str("function",
            "TwoFA").Err(err).Msg("Failed to bind 2fa data")
        return ctx.JSON(http.StatusBadRequest,
            newHTTPError(ErrBind))
    }

    if err := h.validator.Struct(&req); err != nil {
        log.Error().Str("function",
            "TwoFA").Err(err).Msg("Failed to validate 2fa data")
        return ctx.JSON(http.StatusBadRequest,
            newHTTPError(ErrBind))
    }
}

```

```

        token,                                err                                :=
h.user.OTPCheck(ctx.Request().Context(), &req)
    if err != nil {
        log.Error().Str("function",
"TwoFA").Err(err).Msg("Failed to login user")
        return ctx.JSON(http.StatusInternalServerError,
newHTTPError(checkErr(err, ErrLogin)))
    }

    return ctx.String(http.StatusOK, token)
}

func (h *Handler) EnableTwoFA(ctx echo.Context) error {
    userID, ok := ctx.Get("userID").(string)
    if !ok {
        log.Error().Str("function",
"EnableTwoFA").Msg("Failed to get userID")
        return ctx.JSON(http.StatusBadRequest,
newHTTPError(ErrBind))
    }

    token, ok := ctx.Get("token").(string)
    if !ok {
        log.Error().Str("function",
"EnableTwoFA").Msg("Failed to get token")
        return ctx.JSON(http.StatusBadRequest,
newHTTPError(ErrBind))
    }

    image,                                err                                :=
h.user.EnableTwoFA(ctx.Request().Context(), userID,
token)
    if err != nil {
        log.Error().Str("function",
"EnableTwoFA").Err(err).Msg("Failed to login user")
        return ctx.JSON(http.StatusInternalServerError,
newHTTPError(ErrLogin))
    }

    return ctx.Blob(http.StatusOK, "image/png", image)
}

func (h *Handler) DisableTwoFA(ctx echo.Context) error {
    userID, ok := ctx.Get("userID").(string)

```

```

        if !ok {
            log.Error().Str("function",
                "DisableTwoFA").Msg("Failed to get userID")
            return ctx.JSON(http.StatusBadRequest,
                newHTTPError(ErrBind))
        }

        token, ok := ctx.Get("token").(string)
        if !ok {
            log.Error().Str("function",
                "DisableTwoFA").Msg("Failed to get token")
            return ctx.JSON(http.StatusBadRequest,
                newHTTPError(ErrBind))
        }

        err := h.user.DisableTwoFA(ctx.Request().Context(),
            userID, token)
        if err != nil {
            log.Error().Str("function",
                "DisableTwoFA").Err(err).Msg("Failed to login user")
            return ctx.JSON(http.StatusInternalServerError,
                newHTTPError(ErrLogin))
        }

        return ctx.NoContent(http.StatusOK)
    }

func (h *Handler) Logout(ctx echo.Context) error {
    token, ok := ctx.Get("token").(string)
    if !ok {
        log.Error().Str("function",
            "Logout").Msg("Failed to get token")
        return ctx.JSON(http.StatusBadRequest,
            newHTTPError(ErrBind))
    }

    h.user.Logout(ctx.Request().Context(), token)
    return ctx.NoContent(http.StatusOK)
}

```

ДОДАТОК Б
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ БІЗНЕС-ПРОЦЕСІВ КОМПАНІЇ

Виконав
студент групи КНТ-112м

Артем УДОВИЧЕНКО

Керівник
к.т.н. доцент

Анжеліка ПАРХОМЕНКО

Запоріжжя, 2023

Рисунок Б.1 – Слайд 1

Актуальність роботи

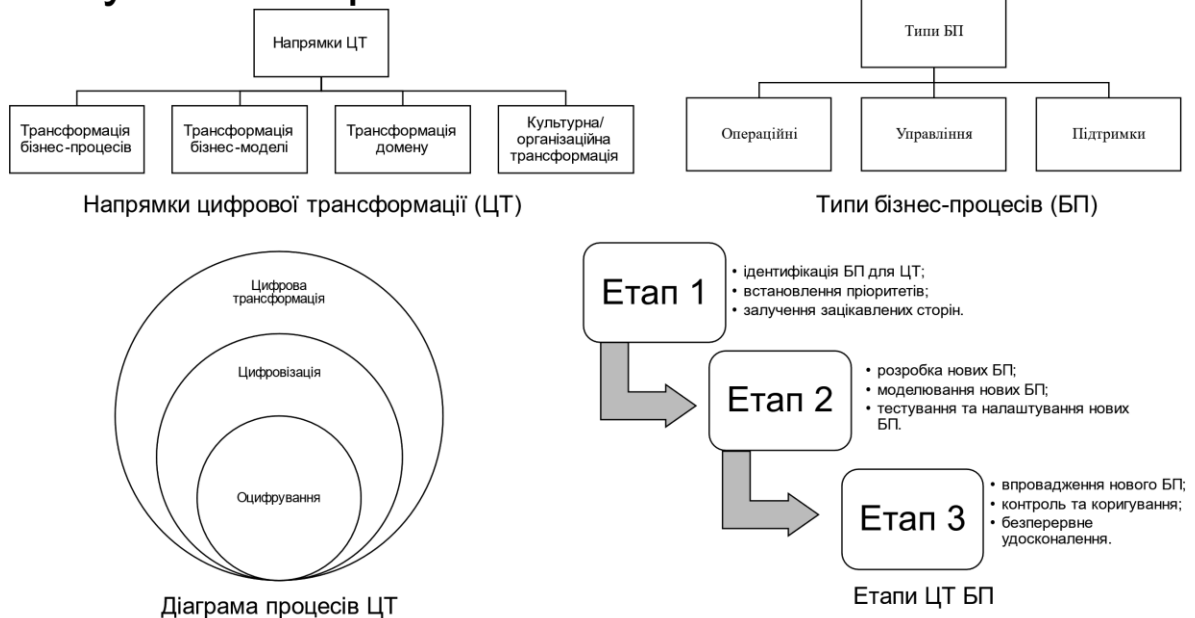


Рисунок Б.2 – Слайд 2

Мета та задачі дипломної роботи

Мета роботи – дослідження та програмна реалізація методів цифрової трансформації (ЦТ) бізнес-процесу (БП) моніторингу діяльності фахівців компанії для підвищення оперативності прийняття управлінських рішень в режимі онлайн.

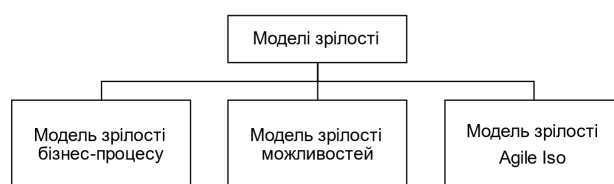
Завдання роботи:

- дослідити існуючі моделі зрілості ЦТ БП;
- виконати розробку методики оцінки рівня ЦТ БП;
- виконати проєктування та конструювання програмного забезпечення для віддаленого моніторингу бізнес-діяльності фахівців, що використовують платформу BAS ERP.

3

Рисунок Б.3 – Слайд 3

Дослідження моделей зрілості цифрової трансформації бізнес-процесів



Типи моделей зрілості

Рівень зрілості	Назва	Опис
1	Базова цифровізація	Використання базових технологій, таких як підключення до Інтернету, наявність вебсайту та використання електронної пошти для покращення ділових операцій
2	Розширена цифровізація	Використання передових технологій для покращення роботи на додаток до Базової цифровізації.
3	Розумна цифровізація	Інтегроване використання цифрових технологій, таких як онлайн платформи та автоматизоване керування ланцюжком поставок на додаток до Розширеної цифровізації
4	Трансформативна цифровізація	Цифрові технології та можливості управління, що дозволяють трансформувати бізнес, наприклад цифрові активи, що створюють нові бізнес-моделі на додаток до Розумної цифровізації

Структурні рівні моделі зрілості ЦТ

4

Рисунок Б.4 – Слайд 4

Розробка методики застосування моделі зрілості цифрової трансформації

Контрольний список для аналізу рівня зрілості ЦТ

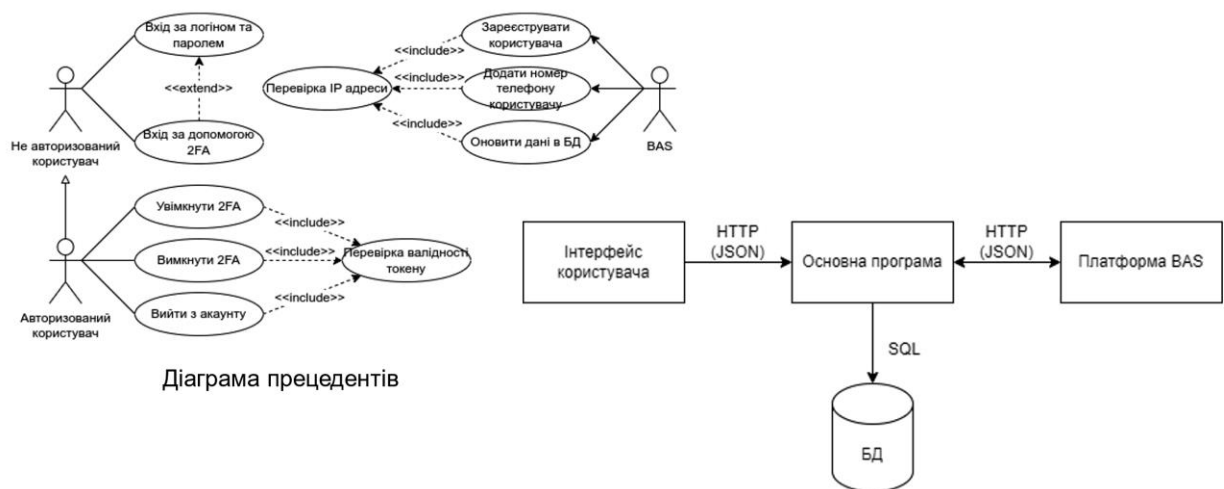
Категорія	Запитання
Цифровізація сервісів	Формати запитів, виданих компанією та переданих постачальникам, підрядникам тощо. Методи спілкування із сторонніми компаніями та організаціями.
Використання ПЗ та можливостей апаратного забезпечення	Використання ПЗ в повсякденних завданнях. Використання ПЗ для управління документами та завданнями. Використання сховища цифрових даних (хмарного або внутрішнього).
Витрати на ІКТ	Середньорічні витрати на ІКТ.
Навчання цифровим можливостям	Навчання персоналу цифровим можливостям організації.
Інвестиції в дослідження та розробки (R&D)	Середній річний бюджет на R&D.

Схема алгоритму методики застосування моделі зрілості ЦТ



Рисунок Б.5 – Слайд 5

Проектування ПЗ



Діаграма прецедентів

Узагальнена архітектура системи

Рисунок Б.6 – Слайд 6

Проектування ПЗ

Категорія	Пароль	SMS	USB-ключ	TOTP	Біометрія
Безпека	Слабка	Слабка	Сильна	Сильна	Сильна
Використання	Зручний	Зручний	Складніший	Зручний	Зручний
Комфорт	Зручний	Зручний	Потребує флешки	Зручний	Потребує додатковий пристрій
Додаткові витрати	Немає	На кожне SMS	Покупка флешки	Немає	Покупка пристрою біометричної ідентифікації

Порівняння методів аутентифікації

Характеристика	VSCode	Sublime Text	IntelliJ IDEA	Eclipse
Відкритість	+	-	-	+
Швидкодія	++	+++	+	+
Підтримка мов програмування	+	+	+	-
Відлагодження	+	-	+	+
Git	+	+	+	+
Docker	+	-	+	+
IntelliSense	+	-	+	+
Вбудований термінал	+	-	+	+
Наявність безкоштовної версії	+	-	-	+

Порівняння середовищ розробки

Характеристика	Golang	C++	Python	Java
Синтаксис	Легкий	Важкий	Легкий	Середній
Продуктивність	Висока	Висока	Низька	Середня
Навчання	Легкий	Важкий	Легкий	Середній
Конкурентоспроможність	Висока	Середня	Середня	Низька
Розширюваність	Висока	Висока	Висока	Висока
Підтримка ООП	+	+	+	+
Підтримка ФП	+~	+	+	+~

Порівняння мов програмування

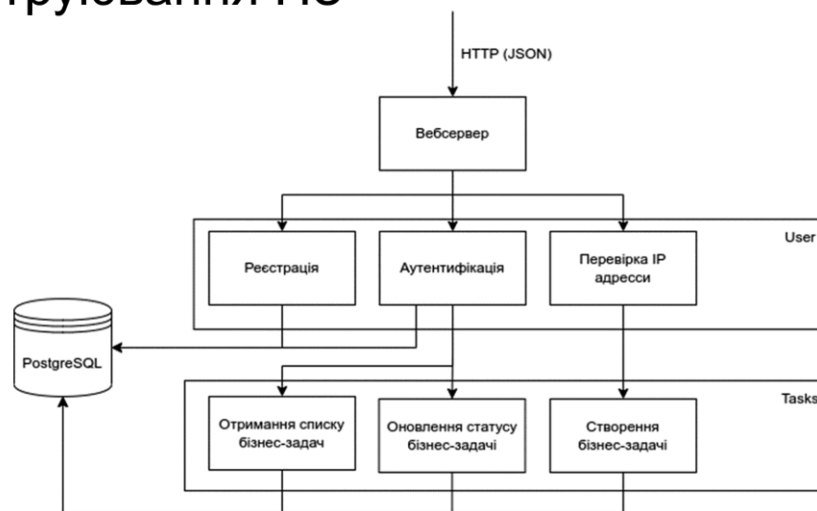
Ліцензія	PostgreSQL License	Proprietary	Proprietary	GPL
Розширюваність	++	++	+	++
SQL	+	+	+	+
JSON	+	+	+	+
XML	+	+	+	+
NoSQL	+	-	-	+~
Повнотекстовий пошук	+	+	+	+
Реплікація	+	+	+	+
Індексація	++	++	++	++
BLOB	+	+	+	+
Швидкодія	++	++	++	+

Порівняння СКБД

7

Рисунок Б.7 – Слайд 7

Конструювання ПЗ



Структура основної програми

8

Рисунок Б.8 – Слайд 8

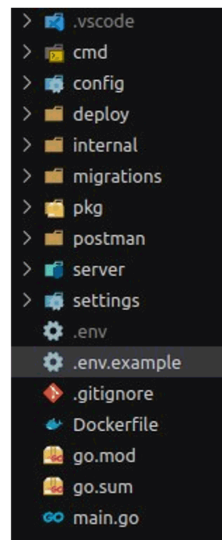
Конструювання ПЗ



9

Рисунок Б.9 – Слайд 9

Конструювання ПЗ



Структура кореневої папки

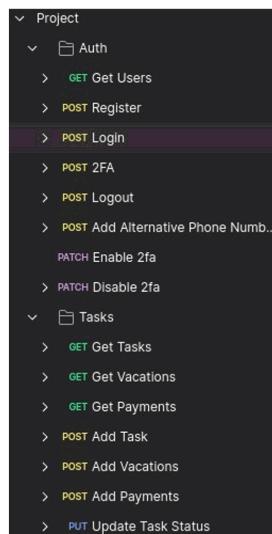
users	tasks
id text	object_name text
full_name text	task_name text
department text	task_type int
position text	uuid text
password text	date text
otp_secret text	author text
	author_id text
	creator_id text
	end_user text
	end_user_id text
	deadline_date text
	task_info text
	agree_status boolean
	linked_task_id text
	approval_list [text]
	comment text
	enable_dead_date_shift boolean
	layout_type int
	period_start text
	period_end text
	holiday_maker text
	substitutional text
	kontragent text
	organization text
	sum text
	payment_date text
	payment_purpose text

Узагальнена структура БД

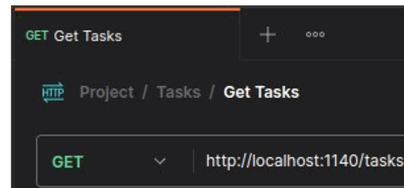
10

Рисунок Б.10 – Слайд 10

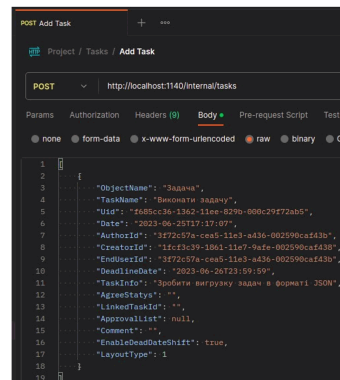
Результати тестування програми



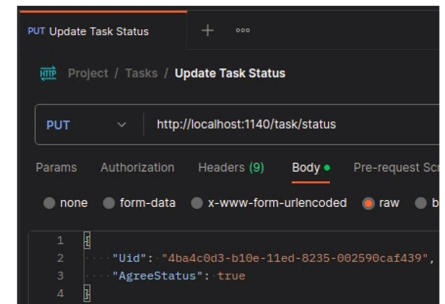
Список доступних запитів



Запит Get Tasks



Запит Add Task



Запит Update Task Status

11

Рисунок Б.11 – Слайд 11

Висновки

- В результаті виконання дипломної роботи було створено програмний застосунок, який надає керівнику компанії онлайн платформу для віддаленого моніторингу та контролю бізнес-діяльності фахівців компанії, які використовують систему BAS ERP.
- Наукова новизна роботи полягає в тому, що розроблено методику використання моделі зрілості цифрової трансформації (ЦТ) для оцінки рівня ЦТ компанії, яка дозволяє забезпечити якість процесів ЦТ та намітити подальші шляхи для переходу на вищі рівні ЦТ.
- Практична цінність розробки полягає в тому, що впровадження розробленої програми дозволить спростити та пришвидшити процес прийняття управлінських рішень керівниками компанії та дозволить оперативно реагувати на результати бізнес-діяльності фахівців через браузер на смартфоні або комп'ютері.
- Проведене тестування показало, що створений програмний застосунок дозволяє виконати ЦТ процесу моніторингу бізнес-діяльності фахівців компанії та перейти від рівня Розширеної цифровізації до Розумної цифровізації згідно з обраною 4-рівневою моделлю зрілості процесів ЦТ.
- В подальшій роботі функціонал програми можливо розширити шляхом додавання хмарного сховища для зберігання цифрових даних компанії.

12

Рисунок Б.12 – Слайд 12