

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
Центр післядипломної освіти

Кафедра

Програмної інженерії

(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти

перший (бакалаврський)

Програмна система обліку доступу та робочого часу робітників підприємства

(тема)

Виконав:

студент 4 курсу, групи ПЗПП-22-2

Бабій В.Г.

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного
забезпечення

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Програмна інженерія

(повна назва освітньої програми)

Керівник доц. кафедри ІІІ Русакова Н.Є.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф.

(підпис)

З.В. Дудар

(посада, прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Центр післядипломної освіти

Кафедра	<u>Програмної інженерії</u> <small>(повна назва)</small>
Рівень вищої освіти	<u>перший (бакалаврський)</u>
Спеціальність	<u>121 – Інженерія програмного забезпечення</u> <small>(код і повна назва спеціальності)</small>
Тип програми	<u>освітньо-професійна</u> <small>(освітньо-професійна або освітньо-наукова)</small>
Освітня програма	<u>Програмна інженерія</u> <small>(повна назва)</small>

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«___» _____ 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Бабію Володимиру Григоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмна система обліку доступу та робочого часу робітників підприємства

затверджена наказом університету від "17" червня 2024 р. № 588 _____

2. Термін подання студентом роботи до екзаменаційної комісії " 22 " 07 2024 р.

3. Вихідні данні до роботи вимоги до розроблюваної програми, вимоги до архітектури системи, електронні ресурси за обраною темою, мови програмування C#, технології .NET Core 3.1, СКБД MSSQL, середовища розробки Visual Studio.

4. Перелік питань, що потрібно опрацювати в роботі вступ, аналіз предметної області, формування вимог до програмного забезпечення, архітектура та проектування розробленого програмного забезпечення, опис прийнятих програмних рішень, тестування програмного забезпечення, висновки, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Позначка про виконання
1	Аналіз проблемної області	6.05.2024 – 10.05.2024	Виконано
2	Розробка постановки задачі	10.05.2024 – 13.05.2024	Виконано
3	Аналіз та моделювання предметної області	13.05.2024 – 15.05.2024	Виконано
4	Проектування БД	15.05.2024 – 18.05.2024	Виконано
5	Розробка алгоритмів	18.05.2024 – 23.05.2024	Виконано
6	Проектування архітектури програмної системи	23.05.2024 – 25.05.2024	Виконано
7	Програмна реалізація системи	25.05.2024 – 12.06.2024	Виконано
8	Підготовка пояснювальної записки	12.06.2024 – 30.06.2024	Виконано
9	Підготовка презентації та доповіді	1.07.2024 – 07.07.2024	Виконано
10	Нормоконтроль	8.07.2024	Виконано
11	Рецензування	10.07.2024	Виконано
12	Занесення диплома в електронний архів	13.07.2024	Виконано
13	Попередній захист	15.07.2024	Виконано
14	Допуск до захисту у зав. кафедри	19.07.2024	Виконано

Дата видачі завдання 6 травня 2024р.

Студент _____ Бабій В.Г.
(підпис) (прізвище, ініціали)

Керівник роботи _____ доц.кафедри ІІІ Русакова Н.Є.
(підпис) (прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи бакалавра містить: 74 с., 22 рисунки, 3 додатки, 11 джерел.

МОВА ПРОГРАМУВАННЯ C#, ОБЛІК ДОСТУПУ, ОБЛІК РОБОЧОГО ЧАСУ, РЕЛЯЦІЙНА БАЗА ДАНИХ, СИСТЕМА КЕРУВАННЯ БАЗОЮ ДАНИХ, ADO.NET, MICROSOFT SQL SERVER, WINDOWS FORMS.

Об'єкт розробки – програмна система Windows Forms, за допомогою якої здійснюється облік доступу працівників до об'єктів підприємства та облік робочого часу працівників на підприємстві.

Мета розробки – забезпечити зберігання інформації та отримання даних для подальшого аналізу ефективності працівників, виявлення систематичних порушників робочого часу на підприємстві, а також контролю доступу на об'єкти підприємства.

Методи рішення завдання – аналіз та моделювання предметної області, концептуальне моделювання, UML-моделювання, об'єктно-орієнтоване програмування, використання СКБД MSSQL.

Результатом роботи є програмний застосунок для автоматизації процесу обліку доступу та робочого часу працівників підприємства, виявлення спроб несанкціонованого доступу на об'єкти підприємства. Програмний застосунок дозволяє зберігати та відображати дані про працівників підприємства, їхні підрозділи, об'єкти підприємства, час реєстрації працівника на об'єкті підприємства під час входу чи виходу з нього. Також програма дозволяє проводити пошук та фільтрацію даних, отримувати статистику по відпрацьованому часу працівників, підтримує задачу автоматизації з імпорту даних про працівників.

C# PROGRAMMING LANGUAGE, ACCESS ACCOUNTING, TIME ACCOUNTING, RELATIONAL DATABASE, DATABASE MANAGEMENT SYSTEM, ADO.NET, MICROSOFT SQL SERVER, WINDOWS FORMS.

The object of development is a Windows Forms software system, which is used to record employee access to company facilities and record the working hours of employees at the company.

The purpose of the development is to ensure the storage of information and the acquisition of data for the further analysis of the efficiency of employees, the identification of systematic violators of working hours at the enterprise, as well as the control of access to the enterprise's facilities.

Task solution methods – analysis and modeling of the subject area, conceptual modeling, UML modeling, object-oriented programming, use of the MSSQL DBMS.

The result of the work is a software application for automating the process of accounting for access and working time of the company's employees, detecting attempts of unauthorized access to the company's facilities. The software application allows you to store and display data about the company's employees, their departments, the company's facilities, the time of the employee's registration at the company's facility when entering or leaving it. The program also allows you to search and filter data, receive statistics on the time worked by employees, and supports the task of automating the import of employee data.

Я, Бабій Володимир Григорович , студент гр. ПЗПП-22-2, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Програмна система обліку доступу та робочого часу робітників підприємства», що буде представлена в екзаменаційну комісію для

публічного захисту, виконана самостійно, в ній не містяться елементи плагіату. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ.....	8
1 Аналіз проблемної галузі.....	10
1.1 Аналіз проблемної галузі.....	10
1.2 Аналіз існуючих аналогів.....	14
1.3 Постановка задачі.....	16
2 Формування вимог до програмної системи.....	18
3 Архітектура та проектування програмного забезпечення.....	21
3.1 Аналіз та моделювання предметної області.....	21
3.2 Розробка алгоритму процедури надання та затвердження доступу.....	28
3.3 Проектування бази даних.....	31
3.4 Розробка архітектури системи.....	35
3.5 Створення UI / UX дизайну системи.....	37
4 Опис програмної реалізації.....	38
4.1 Вибір засобів програмної реалізації.....	38
4.2 Призначення і логічна структура.....	40
4.3 Описання фізичної моделі бази даних.....	42
4.4 Опис програмної реалізації алгоритму процедури затвердження доступу.....	43
4.5 Опис інтерфейсу користувача.....	46
5 Тестування розробленого програмного забезпечення.....	53
5.1 Обґрунтування вибору виду тестування.....	53
5.2 Опис тестування.....	53
Висновки.....	58
Перелік джерел посилання.....	59
Додаток А Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ.....	60
Додаток Б Слайди презентації.....	61
Додаток В Лістинг програмного коду.....	69

ВСТУП

Проблема обліку робочого часу працівників на підприємстві, а також обліку відвідування об'єктів підприємства є актуальною для менеджерів та служби безпеки, коли на підприємстві працює велика кількість людей. Головна проблема, котру повинна вирішувати програмна система для менеджерів та служби безпеки – можливість швидко отримати необхідну інформацію по кожному працівнику: скільки часу працівник провів на робочому місці, скільки запізнь є у працівника, які об'єкти підприємства працівник відвідував, тощо.

Актуальність роботи полягає у створенні такої платформи, яка допоможе менеджерам спростити та автоматизувати облік робочого часу і підвищити загальну ефективність праці, а також допоможе службі безпеки підприємства виявляти спроби несанкціонованого доступу працівників на об'єкти підприємства для подальшого розслідування.

Таким чином, метою розробки є створення системи, яка дозволить вести облік робочого часу працівників підприємства та виявляти спроби несанкціонованого доступу на об'єкти. Це, в свою чергу, дозволить підвищити ефективність праці та її оплати, а також проводити розслідування щодо спроб несанкціонованого доступу.

У ході кваліфікаційної роботи було проведено аналіз предметної галузі та конкурентів, розроблено архітектуру програмної системи та компонентів відповідно до вимог до програмної системи. Було спроектовано та реалізовано програмну систему. Програмна система є легкою для сприйняття, розуміння та швидкого оволодіння функціоналом, отже людина з мінімальним досвідом зможе її використовувати після вивчення відповідної документації.

В ході створення програмної системи були використані середовище розробки Microsoft Visual Studio 2022, платформа ASP.Net, мова програмування C#, інтерфейс програмування додатків Windows Forms та система управління базами даних Microsoft SQL Server .

Робота перевірена на унікальність тексту в мережі інтернет та базі ХНУРЕ (див. додаток А). За результатами кваліфікаційної роботи було розроблено презентацію (див. додаток Б). Лістинг програмного коду наведено в додатку В.

1 АНАЛІЗ ПРОБЛЕМНОЇ ГАЛУЗІ

1.1 Аналіз проблемної галузі

Підприємство є великим і складним організаційним утворенням, що складається з численних підрозділів, будівель або окремих приміщень, розташованих, зазвичай, у різних частинах міста. Усі будівлі та окремі приміщення підприємства мають власну функціональну призначеність і, відповідно, рівень доступу для працівників. Це підприємство займається широким спектром діяльностей, що потребує координації великої кількості працівників і ресурсів, що в свою чергу, потребує впровадження ефективної системи контролю доступу та моніторингу пересування працівників. Кожен працівник підприємства виконує певні обов'язки, відповідно до своєї посади та підрозділу, до якого він закріплений.

Підприємство характеризується наступними особливостями:

- розгалужена структура підрозділів: кожен підрозділ має свою назву і може підпорядковуватись іншому підрозділу. Кількість працівників у підрозділі не обмежена і може дорівнювати нулю (новий або розформований підрозділ);
- різні будівлі з різним рівнем доступу: підприємство розташоване у різних частинах міста і складається з багатьох будівель, кожна з яких має свою назву та адресу;
- інформація про працівників: кожен працівник має унікальний ідентифікатор, ім'я, прізвище, по-батькові, посаду, дату прийому на роботу та інші характеристики. Кожен працівник закріплений за певним підрозділом, але може працювати у різних будівлях.

Менеджмент підприємства, відповідає за загальне управління підприємством, а отже і за його ефективністю. Для оцінки ефективності та оплати праці працівників використовує показники загального обсягу робочого часу, який робітник провів на робочому місці.

Служба безпеки підприємства забезпечує безпеку об'єктів підприємства, видає доступи для кожного працівника, контролює та розслідує спроби несанкціонованого доступу.

Відділ кадрів веде облік працівників та підрозділів, забезпечує кадровий менеджмент, здійснює реєстрацію нових працівників та управління їхніми даними.

Процедура контролю доступу на об'єкти підприємства та підрахунок реального робочого часу, проведеного працівником на робочому місці є досить комплексною і складною без використання засобів автоматизації. І саме для цих частин необхідно вирішити проблеми за допомогою програмної системи.

Отже, програмна система повинна по-перше правильно ідентифікувати працівника на будь-якому об'єкті підприємства без людської допомоги. Це можливо реалізувати за допомогою особистих безконтактних карток. Кожен працівник повинен мати картку з номером, яка використовується для ідентифікації при вході та виході з будівель чи приміщень. Система зчитує номер з картки та ідентифікує закріпленого за цим номером працівника.

По-друге, система повинна фіксувати час реєстрації працівника на об'єкті під час входу та виходу. Коли працівник підносить безконтактну картку до зчитуючого пристрою, система зчитує дані з картки та фіксує реальний поточний час приходу або виходу працівника з будівлі.

По-третє, система повинна виконувати контроль доступу - на основі рівня доступу, закріпленого за кожним працівником, система вирішує, чи має працівник право увійти до конкретної будівлі. В разі, якщо доступ не було дозволено, система повинна записати відповідну спробу, як несанкціонований доступ.

Для того, щоб програмна система дійсно була ефективним інструментом для менеджменту підприємства, служби безпеки підприємства та відділу кадрів та допомагала у вищезгаданих питаннях, розробнику необхідно вирішити наступні проблеми:

- обрати необхідну та правильну архітектуру для системи, яка б мала високу працездатність та дозволяла ефективно взаємодіяти користувачеві з нею;
- обрати відповідну базу даних для зберігання усієї необхідної інформації;
- розробити зручний та інтерактивний інтерфейс для користувачів;
- забезпечити захист даних та безпеку;
- забезпечити формування звітів та статистик;
- створити та розробити алгоритм для взаємодії з існуючими системами на підприємстві.

Для створення такої програмної системи необхідно ретельно підібрати архітектуру, яка буде використовуватися.

Власне архітектура ПЗ – це набір рішень, яким чином повинна бути структура програми або обчислювальної системи, яка включає в себе програмні компоненти, властивості та відносини між ними. Основною ідеєю програмної архітектури є ідея зниження складності системи шляхом абстракції та розмежування повноважень. При виборі певного архітектурного стилю, слід зауважити, що система повинна бути масштабованою та гнучкою, тобто зміни до системи повинні вноситися без зайвих проблем та витрат часу. Таким чином, найкращим варіантом для реалізації є архітектура «клієнт-сервер». У такій архітектурі програма виступає як клієнт, який звертається до сервера (де розміщена база даних) для отримання чи збереження даних. Це один з найпоширеніших видів архітектури для побудови програмних систем.

При виборі способу збереження даних необхідно враховувати всі потреби та обсяг інформації, яку буде зберігати система. Враховуючи переваги та недоліки різних типів баз даних, найбільш вдалим вибором є реляційна база даних. Такий вид бази даних побудований на реляційній моделі, що дозволяє зберігати дані у структурованому вигляді за допомогою таблиць.

Проектування даної бази даних поділяється на три етапи, а саме концептуальний, логічний та фізичний [1].

Концептуальне проектування – це перший крок у побудові бази даних. Власне концептуальна модель – це така модель даних на самому високому рівні абстракції.

Логічне проектування – це процес проектування інформаційної моделі на основі вже існуючих моделей. Фаза логічного проектування бази даних полягає в перетворенні концептуальної моделі даних у логічну модель. Прикладом такої моделі є реляційна модель бази даних. Обов'язковою умовою є те, що реляційна модель повинна бути нормалізована.

Нормалізація – це процес, який полягає в усуненні надмірності даних та виявленні функціональних залежностей. Кожен з кроків цього процесу приводить схему відношень в нормальні форми. Нормальна форма – це сукупність вимог, яким має задовольняти відношення. Зазвичай, процес нормалізації досягає до 3 нормальної форми, а отже кожна з цих форм має свої правила та обмеження.

Відношення знаходиться в першій нормальній формі (1NF) в тому та тільки в тому випадку, коли на перетині будь-якого стовпця та будь-якого запису знаходиться атомарне значення.

Відношення знаходиться в другій нормальній формі (2NF) в тому та тільки в тому випадку, коли воно знаходиться у 1NF, та кожен неключовий атрибут повністю функціонально залежить від первинного ключа відношення.

Відношення знаходиться в третій нормальній формі (3NF) в тому та тільки в тому випадку, якщо воно знаходиться у 2NF та між неключовими атрибутами не має транзитивних залежностей.

Фізичне проектування – це процес створення схеми бази даних для точної СКБД, яка може включати в себе різні обмеження, такі як іменування об'єктів бази даних, або підтримуванні типи даних та інші [2].

Інтерфейс користувача потрібно створювати, використовуючи певні рекомендації щодо зручності та полегшення користування програмою., дотримуватись евристик розробки інтерфейсів, UI/UX дизайну. Оскільки архітектура системи є «клієнт-сервер», інтерфейс буде забезпечений через частину «клієнт».

1.2 Аналіз існуючих аналогів

На сьогоднішній день існує декілька готових рішень для обліку робочого часу та контролю доступу представлених на ринку рішень.

Одним з аналогів системи контролю доступу можна розглядати застосунок ZKTeco ZKBioSecurity від компанії ZKTeco Co., Ltd (див.рис.1).

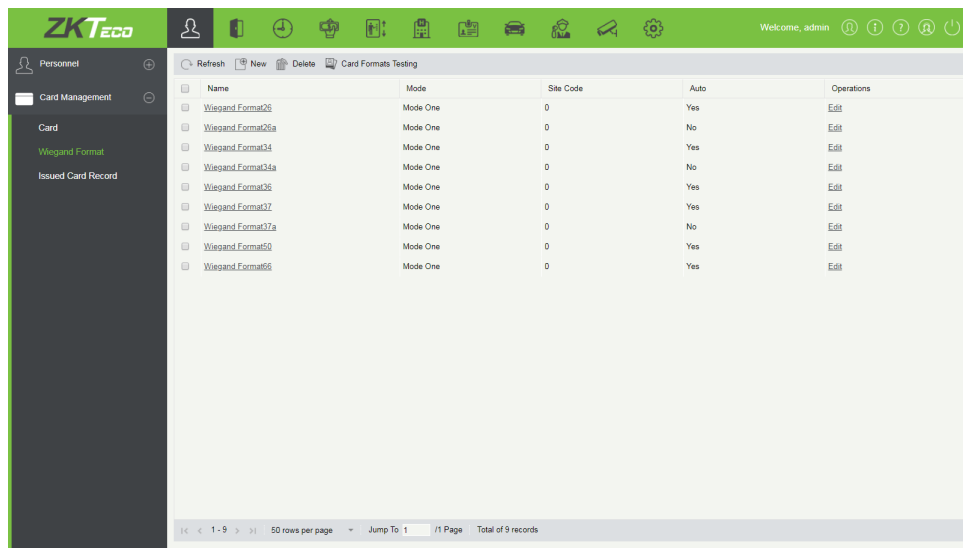


Рисунок 1 – Застосунок ZKTeco ZKBioSecurity (за даними [3])

Це система виробництва компанії «ZKTeco Co., Ltd» яка дає можливість обліку робочого часу та контролю доступу на об’єкти та має власні біометричні термінали. Перевагами цієї системи є біометрична ідентифікація високого рівня та інтегрованість різних модулів. Недоліками цього застосунку можна вважати високу ціну, зокрема й за самі біометричні термінали та обмеження кількості робочих місць, яке дорівнює п’яти.

Ще одним аналогом є Yaware.TimeTracker від компанії “Yaware” (див.рис. 2).

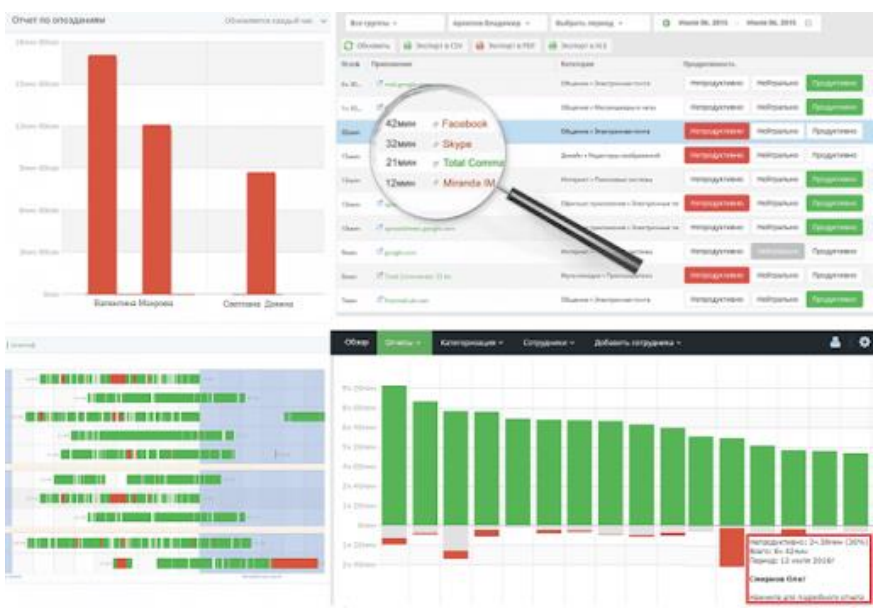


Рисунок 2 – Застосунок Yaware.TimeTracker (за даними [4])

Yaware.TimeTracker працює у фоновому режимі, збирає статистику співробітника, дає повну характеристику роботи. Табель обліку робочого часу впливає на ефективність як офісних, так і віддалених працівників, дає керівнику повний звіт щодо запізнь, реальної завантаженості персоналу, показує співвідношення відпрацьованого часу та активності поза комп'ютером. Кожні 15 хвилин програма формує 18 звітів. Перевагами цієї системи є досить привабливий дизайн та велика кількість звітів. Недоліками цієї системи є досить великі вимоги до апаратного забезпечення, високу ціну, та необхідність додаткового навчання персоналу протягом 30 днів.

Також до аналогів можна віднести програмний застосунок «СКУД» від компанії Smartel, який представляє широкий спектр можливостей обліку доступу та робочого часу (див.рис.3).

Перевагами цього рішення є наявність великої кількості апаратних модулів, які можуть об'єднуватись в єдину систему. Недоліками цього застосунку можна вважати високу ціну, необхідність додаткового апаратного забезпечення, та отримання сертифіката на використання системи.



Рисунок 3 – Застосунок «СКУД» від компанії Smartel (за даними [5])

Незважаючи на дійсно присутніх конкурентів, кожна з систем має свої недоліки, відсутність необхідного функціоналу для майбутнього користувача.

1.3 Постановка задачі

Метою цієї роботи є розробка програмної системи яка допоможе автоматизувати процес обліку доступу та робочого часу працівників підприємства.

Необхідно спроектувати базу даних для збереження інформації стосовно обліку доступу та робочого часу працівників підприємства, та створити програмну систему, що буде забезпечувати наступні функції:

- система повинна відображати, додавати, редагувати та видаляти данні про працівників, підрозділи, об'єкти підприємства;
- система повинна відображати дату та час реєстрації працівника на об'єкті підприємства під час входу чи виходу;
- система повинна підтримувати сортування, пошук та фільтрацію даних;

- система повинна підтримувати обробку даних та формування необхідних звітів та статистик;
- система повинна мати можливість імпорту даних про співробітників з документу Excel, який вивантажується з програмного застосунку по управлінню кадрами підприємства.

На основі цієї інформації, постає такий список завдань, які слід виконати:

- провести аналіз та моделювання предметної області програмної системи;
- спроектувати базу даних для збереження інформації з предметної області;
- спроектувати архітектуру програмної системи;
- розробити алгоритм процедури надання та затвердження доступу працівникам на об'єкти підприємства;
- виконати програмну реалізацію системи та провести тестування створеного програмного продукту.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Розроблювана програмна система має бути масштабованою, адаптивною, стійкою до змін і зручною для користувача. Внесення змін до системи повинно здійснюватися легко і швидко, без зайвих проблем.

Програмна система повинна мати клієнт-серверну архітектуру. Серверна частина повинна відповідати за збереження даних та обробку запитів, а також бути максимально захищеною від несанкціонованого доступу.

Для цієї мети найбільш вдалим вибором буде реляційна база даних MS SQL. Оскільки такий вид бази даних побудований на основі реляційної моделі, тобто бази даних, що зберігає дані у структурованому вигляді за допомогою таблиць. Управління базою даних повинно бути здійснене за допомогою програмного забезпечення MS SQL Server 2019 [6]. Це система управління реляційними базами даних, розроблена Microsoft ще в 1988 році.

Надалі слід забезпечити клієнтську частину програмної системи, яка буде створена в якості окремого незалежного застосунку. Ця частина буде відповідати за відображення даних за допомогою графічного інтерфейсу, відправку запитів до бази даних та обробку отриманої інформації для послідуочого відображення користувачу.

Клієнтська частина повинна підтримувати авторизацію користувача та відображення необхідних даних відповідно до ролі користувача в системі.

Розробка клієнтської частини повинна бути реалізована за допомогою мови програмування C# та технології .NET Core версії 3.1, яка є найбільш оптимальною для даної задачі.

Клієнтська частина повинна буде підключатись та взаємодіяти з базою даних за допомогою технології ADO.NET [7]. Ця технологія розроблена компанією Microsoft, яка надає засоби для доступу до даних та їхнього керування в рамках платформи .NET. Вона є ключовою частиною .NET Framework для роботи з базами

даних і забезпечує інтерфейс для взаємодії з різними джерелами даних, такими як реляційні бази даних, XML-файли, а також деякими іншими структурованими даними. ADO.NET також забезпечує потужний і гнучкий спосіб для роботи з даними в додатках .NET, пропонуючи розробникам великий набір інструментів для ефективної взаємодії з різними джерелами даних.

Під час роботи серверної частини з клієнтською повинна забезпечуватись надійна безпека передачі даних.

Для реалізації механізму аутентифікації та авторизації слід використати аутентифікацію на основі логіну та паролю, пароль в системі необхідно зберігати у зашифрованому вигляді. Авторизація повинна забезпечувати надання доступ до ресурсів системи відповідно до ролі користувача.

Для того, щоб задовольнити вимоги користувача, система повинна мати наступний функціонал:

- можливість авторизуватися у ролі менеджера, співробітника служби безпеки, співробітника відділу кадрів;
- можливість авторизуватися у ролі адміністратора системи;
- управляти користувачами системи у ролі адміністратора;
- створювати бекап даних системи, як адміністратор системи;
- можливість переглянути увесь список працівників, підрозділів та об'єктів підприємства;
- можливість додавати, редагувати та видаляти працівників та підрозділи підприємства у ролі співробітника відділу кадрів;
- можливість додавати, редагувати та видаляти об'єкти підприємства у ролі співробітника служби безпеки;
- можливість затверджувати або відхиляти дозволи, подані на затвердження;
- можливість додавати, редагувати та видаляти дозволи на відвідування об'єктів підприємства працівниками у ролі співробітника служби безпеки;

- можливість переглядати дату та час відвідування об'єктів підприємства працівником у ролі менеджера;
- можливість формувати необхідні звіти по працівникам та об'єктам підприємства;
- можливість формувати статистики по працівникам та об'єктам підприємства;
- можливість імпортувати з файлу Excel дані про співробітників підприємства.

3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз та моделювання предметної області

На підприємстві з розгалуженою структурою підрозділів працює велика кількість людей. Територіально підприємство розташоване у різних частинах міста та складається з багатьох будівель з різним рівнем доступу. Кожен працівник закріплений лише за одним підрозділом, але може працювати у різних будівлях в різні проміжки часу. Щоб потрапити до будівлі у працівника повинен бути відповідний рівень доступу та особиста безконтактна картка. Якщо у працівника немає відповідного рівня доступу, він не зможе зайти в будівлю, або вийти з неї. Проходження до будівлі контролюється системою безпеки, яка зчитує номер особистої картки працівника та записує інформацію про час приходу та виходу кожного працівника.

Кожен підрозділ має свою назву та може підпорядковуватись іншому підрозділу. Кількість працівників не обмежена у одному підрозділі та може дорівнювати нулю, якщо це новий підрозділ, або розформований підрозділ.

Кожен працівник має свій ідентифікатор, ім'я, прізвище, по-батькові, посаду, дату прийому на роботу та інші характеристики. Крім того, кожен працівник пов'язаний з певним підрозділом, де він працює.

Кожний об'єкт підприємства (окрема будівля чи окреме приміщення) має свою назву та адресу. Проходження до будівлі контролюється системою безпеки, яка записує інформацію про час приходу та виходу кожного працівника.

Кожен працівник може працювати в будівлях (приміщеннях), до яких має доступ. Для цього він повинен зареєструватися в системі безпеки при вході та виході, використовуючи особисту безконтактну картку. Якщо працівник повинен перейти в іншу будівлю, він повинен вийти з поточної будівлі, а потім зареєструватися в системі безпеки при вході в нову будівлю.

Для керування підрозділами, працівниками та об'єктами (будівлями та приміщеннями) можна використовувати спеціальну систему управління, яка дозволяє відслідковувати роботу працівників, змінювати рівень доступу до будівель та інше.

Менеджмент підприємства здійснює контроль робочого часу працівників, їхні переміщення протягом робочого дня між об'єктами підприємства для покращення ефективності роботи підприємства.

Служба безпеки підприємства контролює об'єкти підприємства, видає доступи працівникам до об'єктів підприємства на основі виробничих потреб та внутрішніх розпоряджень, контролює та розслідує несанкціонований доступ (або спроби несанкціонованого доступу).

Відділ кадрів підприємства веде облік працівників та структурних підрозділів підприємства.

Інформаційні потреби служби безпеки:

- необхідна інформація про нових співробітників, що приймаються на роботу, для яких необхідно видати доступ до об'єктів підприємства;
- необхідна інформація про звільнених співробітників, для яких необхідно скасувати доступ;
- необхідна інформація про спроби несанкціонованого доступу до об'єктів підприємства.

Інформаційні потреби менеджера підприємства:

- необхідна інформація про запізнення працівників на робочі місця;
- необхідна інформація про загальний час перебування співробітників на робочих місцях;
- необхідна інформація про працівників, які протягом робочого дня знаходились поза приміщенням більше дозволеного часу;
- необхідна інформація по працівникам з кількістю відпрацьованих годин кожного дня впродовж місяця;

– необхідна інформація про список об'єктів, які відвідує найменша кількість працівників протягом місяця.

Загальна контекстна діаграма інформаційної системи наведена на рисунку 4.

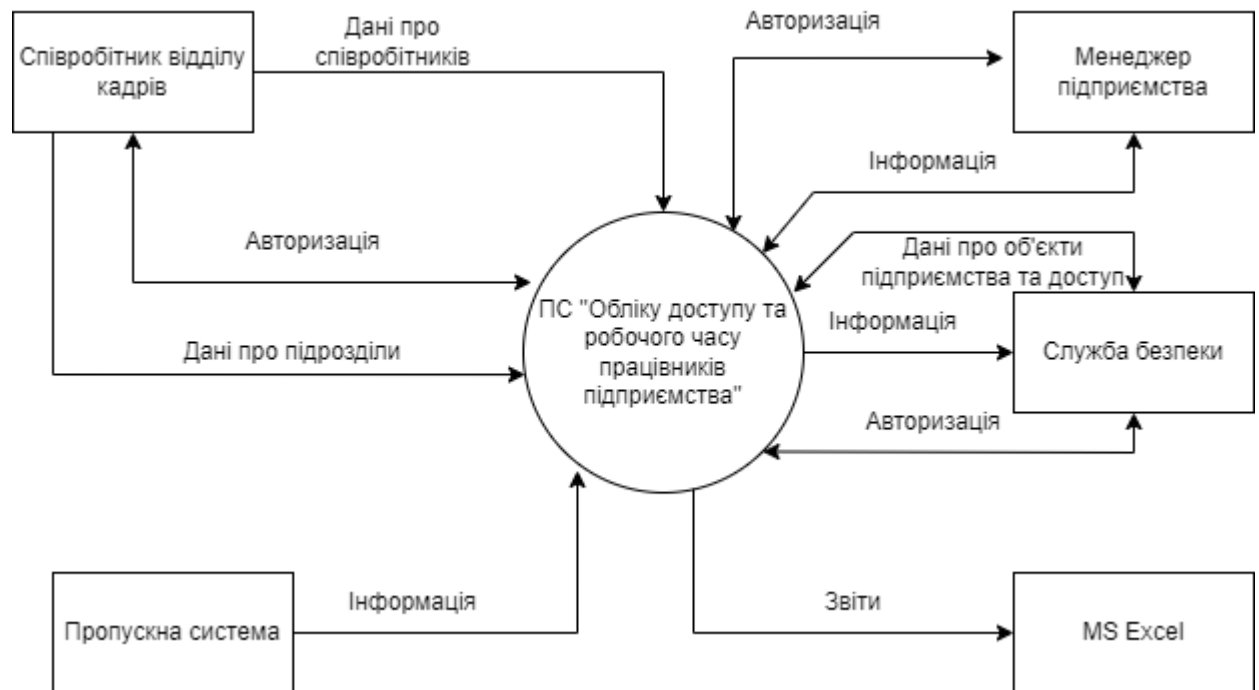


Рисунок 4 – Загальна контекстна діаграма ПС «Облік доступу та робочого часу працівників підприємства» (рисунок виконаний самостійно)

Опис функціонального призначення інформаційної системи, що створюється у вигляді UML-діаграми, наведено на рисунку 5.

Опис концептів програмної області та зв'язків між ними зобразимо у вигляді загальної діаграми класів представлено на рисунку 6.

Зазначимо перелік характеристик для кожного з наведених понять. Поняття підрозділ має наступні властивості: назву підрозділу та телефон. Поняття працівник включає в себе ПІБ, дату народження, посаду, номер особистої картки доступу, дату прийому на роботу і дату звільнення.

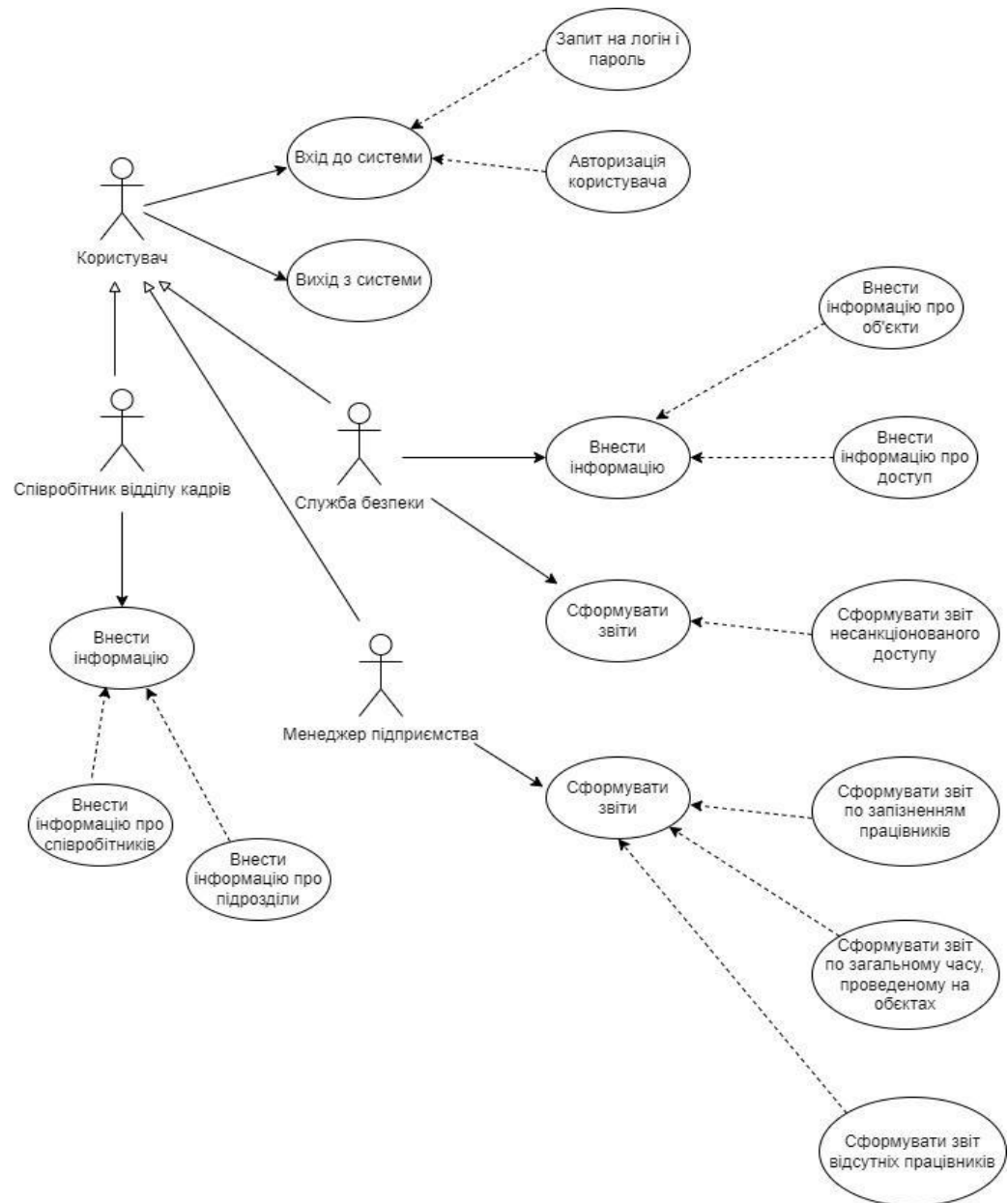


Рисунок 5 – Загальна USE-CASE діаграма (рисунок виконаний самостійно)

Об'єкти підприємства мають зазначені атрибути: назву об'єкта та адресу місцезнаходження.

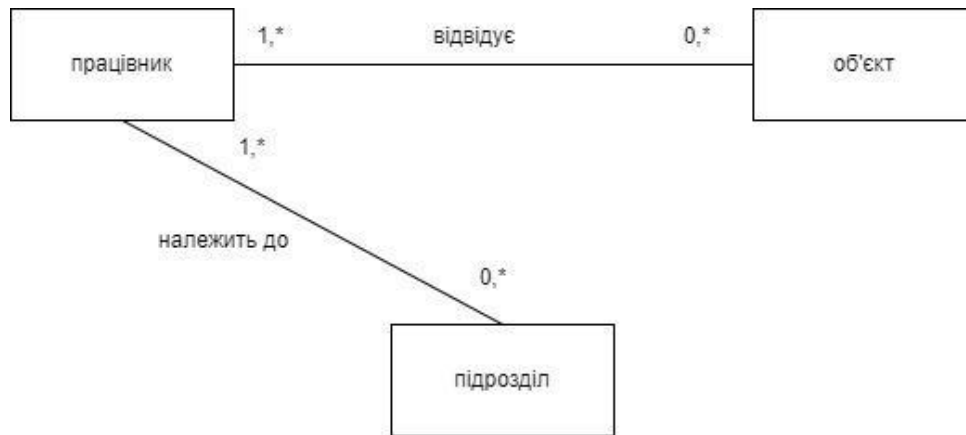


Рисунок 6 – Загальна діаграма класів (рисунок виконаний самостійно)

Для спрощення роботи відділу кадрів, служби безпеки та менеджерів під час роботи з інформацією вони повинні мати можливість:

- сортувати працівників за ПІБ, посадою, датою народження;
- здійснювати пошук інформації про працівників за його повним або частковим ПІБ, про підрозділ за його повною або частковою назвою, про об'єкт підприємства за його назвою та адресою;
- здійснювати фільтрацію інформації по працівникам по датам народження, датам прийому на роботу та датам звільнення, а також по підрозділу, до якого вони належать.

Співробітники відділу кадрів, служби безпеки та менеджери підприємства для підвищення безпеки та ефективності праці на підприємств повинні мати можливість отримання наступної статистики:

- про працівників, які найчастіше запізнювались на робоче місце (тобто час першої реєстрації на об'єкті був пізніше початку робочого дня: підраховується кількість таких запізнь по кожному працівнику за місяць);
- про список працівників, які протягом робочого дня знаходились поза приміщенням більше ніж дозволений час (протягом робочого дня працівники покидають приміщення на перекур, чи на обідню перерву, тощо; підраховується кількість часу, яку працівник провів поза межами

приміщення між першою реєстрацією (початок робочого дня) та останньою реєстрацією (закінчення робочого дня);

- про список об’єктів, які відвідує найменша кількість працівників протягом місяця;
- про список працівників, які мають прогули (тобто не зареєструвались на жодному з об’єктів протягом робочого дня).

Оскільки інформація про нових працівників надходять з зовнішньої програми в електронному вигляді в форматі .xlsx, то необхідна автоматизація з імпорту цієї інформації в БД. Запропоновану автоматизацію зображено за допомогою діаграми станів (див. рис. 7).

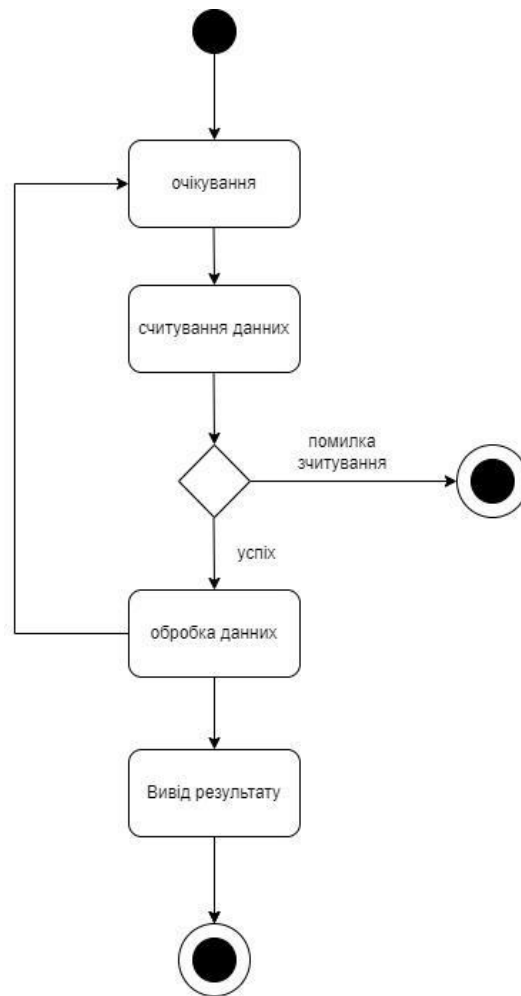


Рисунок 7 – Діаграма станів імпорту інформації з зовнішньої системи (рисунок виконано самостійно)

Опис існуючого документообігу складається з наступних документів:

- звіт по працівникам «Щоденний звіт» з кількістю відпрацьованих годин кожного дня впродовж місяця: ПІБ працівника, відділ, дата, кількість відпрацьованих годин;
- звіт «Несанкціонований доступ» по спробам несанкціонованого доступу на об'єкти підприємства: об'єкт підприємства, ПІБ працівника, дата.

В проблемній області існує ряд обмежень, які можна віднести до обмежень цілісності стосовно ідентифікації:

- кожен працівник буде ідентифікуватися сурогатним ідентифікатором, оскільки ПІБ працівників можуть співпадати;
- кожний підрозділ на підприємстві однозначно ідентифікується сурогатним номером, оскільки назва підрозділу може змінюватись, і можуть з'являтися нові підрозділи з назвою, як у попередніх;
- кожний об'єкт на підприємстві однозначно ідентифікується сурогатним ідентифікатором, оскільки декілька об'єктів можуть знаходитись за однією адресою та мати однакову назву.

В проблемній області існує ряд обмежень, які можна віднести до обмежень цілісності стосовно зв'язків:

- кожний працівник може належати тільки до одного підрозділу, але може змінювати підрозділ протягом роботи;
- кожний підрозділ може містити багато працівників, але може і не містити жодного, наприклад, коли тільки створюється новий підрозділ, або розформовується існуючий;
- на кожний об'єкт підприємства може заходити багато працівників, а може і ніхто не заходити.

З урахуванням ліцензійного ПЗ, що є на підприємстві необхідно дотримуватися наступних обмежень:

- всі документи формувати для редакторів MS Word або MS Excel;

– використовувати СКБД MSSQL.

Лінгвістичні відносини, що існують в ПО роботи підприємства, повинні нормалізуватися документацією з організації робочого процесу та термінологією, що прийнята в галузі роботи підприємства.

3.2 Розробка алгоритму процедури надання та затвердження доступу

Для того, щоб працівник зміг відвідати об'єкт підприємства, йому необхідно надати відповідний дозвіл на доступ до об'єкту. За вимогами правил безпеки на підприємстві надання дозволу не може здійснюватися одноосібно. Після створення заявки на дозвіл особою 1, її має бути перевірено та підтверджено (або відхилено) особою 2. Це гарантує, що доступ надається лише після ретельної перевірки та згоди відповідальних осіб. Така процедура забезпечує підвищений рівень безпеки і запобігає несанкціонованому доступу до об'єктів підприємства. Крім того, всі дії з обробки заявок фіксуються у відповідній системі для подальшого моніторингу та аудиту.

Зобразимо життєвий цикл сутності «Дозвіл» у вигляді діаграми. Існує безліч способів подання діаграм такого типу. На рисунку 8 наведена запозичена у Джеймса Марта діаграма-"огорожа", яка відображає життєвий цикл сутності "Дозвіл" у спрощеній системі.

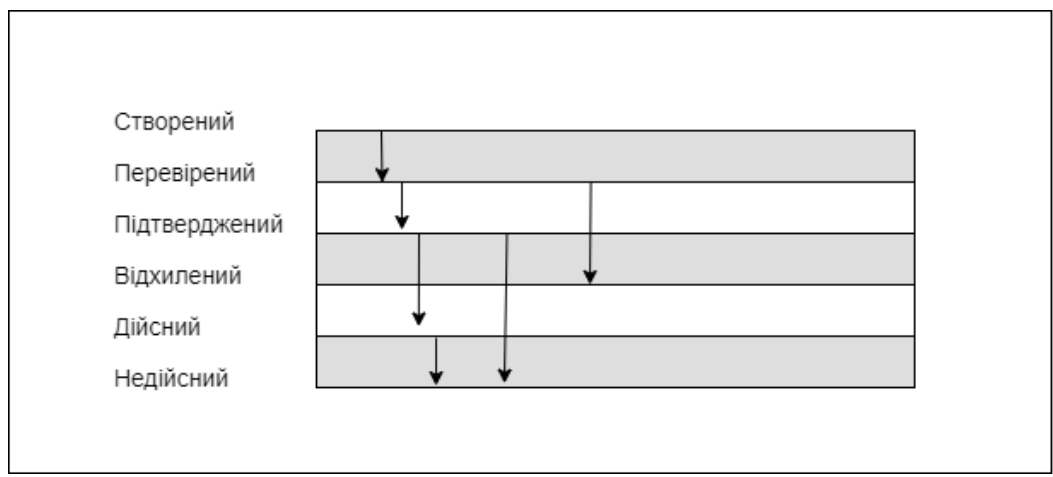


Рисунок 8 – Життєвий цикл «Дозволу» (рисунок виконано самостійно)

На першому етапі дозвіл створюється особою 1, де вказуються наступні дані:

- назва дозволу;
- працівник (його ПІБ), для якого створюється дозвіл на доступ;
- об'єкт підприємства, на відвідування якого видається дозвіл;
- дата початку дії дозволу;
- дата закінчення дії дозволу.

На цьому етапі особа 1 може зберегти дозвіл для подальшого редагування (в цьому разі дозвіл зберігається в системі з типом «Створений») або відправити на підтвердження (в такому разі дозволу виставляється статус «Перевірений»). Після відправки на підтвердження особа 1 не зможе редагувати або видалити дозвіл.

На наступному етапі особа 2 переглядає дозвіл та переводить його в статус «Підтверджений» або «Відхилений».

Якщо дозвіл було підтверджено, він починає діяти починаючи з дати, вказаної у полі «Дата початку дії» і переходить в статус «Дійсний». По настанню дати «Дата закінчення дії дозволу», дозвіл переходить в статус «Недійсний».

Відхилені дозволи більше не використовуються системою, а дані про них можуть бути використані для аудиту безпеки.

Запропоновану діаграму зміни станів по затвердженню доступу зображено на рисунку 9.

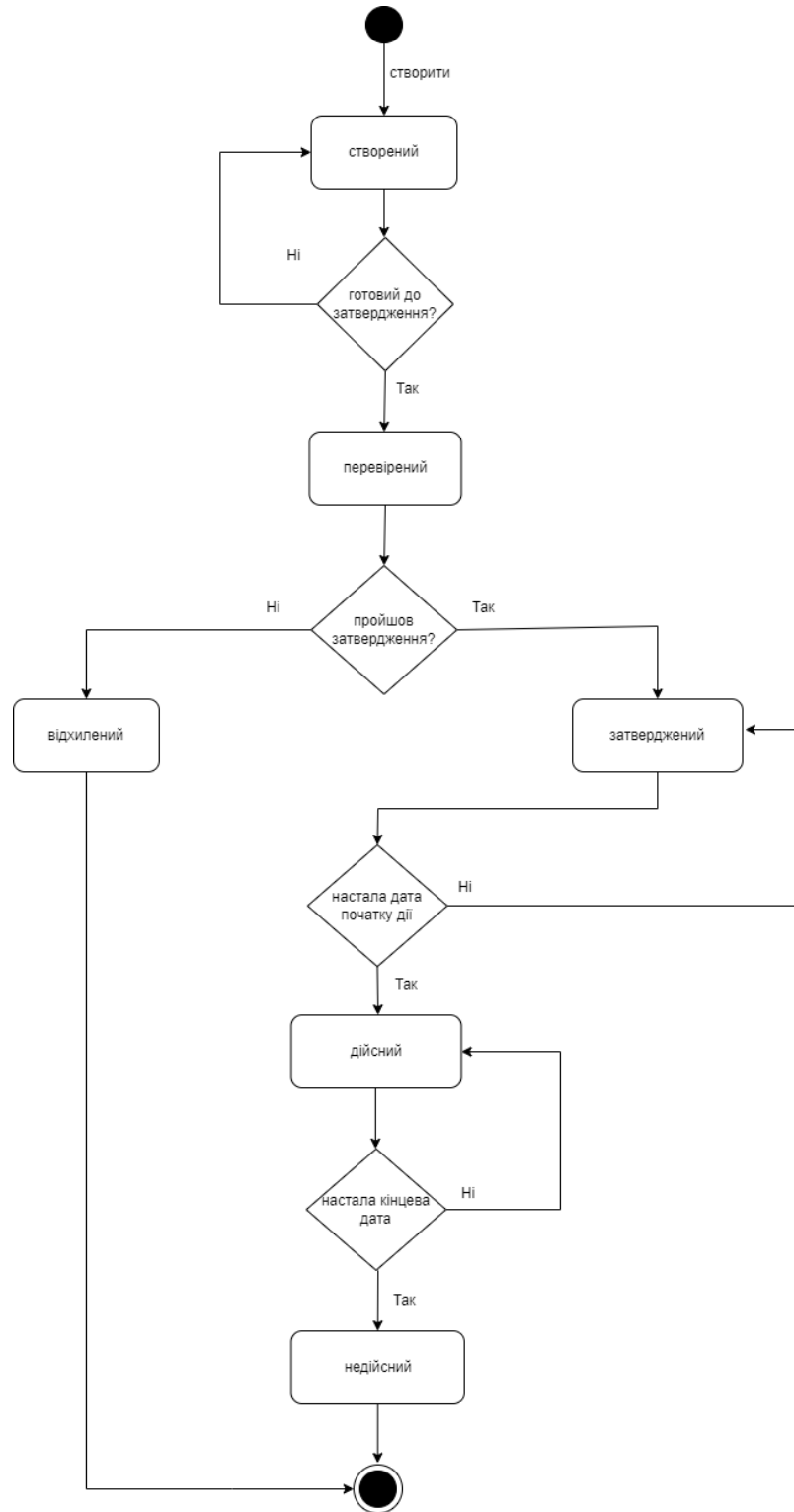


Рисунок 9 – Діаграма зміни станів доступу (рисунок виконано самостійно)

3.3 Проектування бази даних

Після аналізу предметної області, виконаного в п.3.1, необхідно визначити тип бази даних для використання. Найбільш ефективним варіантом є реляційна база даних. За допомогою реляційної моделі даних слід створити сутності у вигляді таблиць, визначити атрибути сутностей (стовпці таблиць) та зв'язки між ними. Кінцевим результатом має бути база даних у третій нормальній формі.

Реляційні бази даних ґрунтуються на реляційній моделі – інтуїтивно зрозумілому, наочному поданні інформації у вигляді таблиць. Кожен рядок у таблиці такої бази даних являє собою запис унікальним ідентифікатором – ключем. Стовпці таблиць мають атрибути даних, що дає змогу встановлювати зв'язки між елементами даних. Саме цьому, реляційна модель бази даних є найбільш ефективною для вирішення задачі даного проекту (див. рис. 10).

Нормальна форма – це вимога, що висувається до структури таблиць в теорії реляційних баз даних для усунення з бази зайвих функціональних залежностей між атрибутами. В нашому проекті при нормалізації виконується приведення таблиць до третьої нормальної форми. Відношення знаходиться у третій нормальній формі, коли воно знаходиться у першій та другій нормальній формі, тобто всі атрибути атомарні, кожний не ключовий атрибут повністю функціонально залежить від первинного ключа відношення та між неключовими атрибутами немає транзитивних залежностей.

В усіх відношеннях всі атрибути є атомарними, тому вони знаходяться в 1 НФ.

При зведенні до другої нормальної форми було доведено, що всі відношення не містять неповних функціональних залежностей, тобто в складі потенційного ключа відсутня менша підмножина атрибутів від якої можна також вивести дану функціональну залежність.

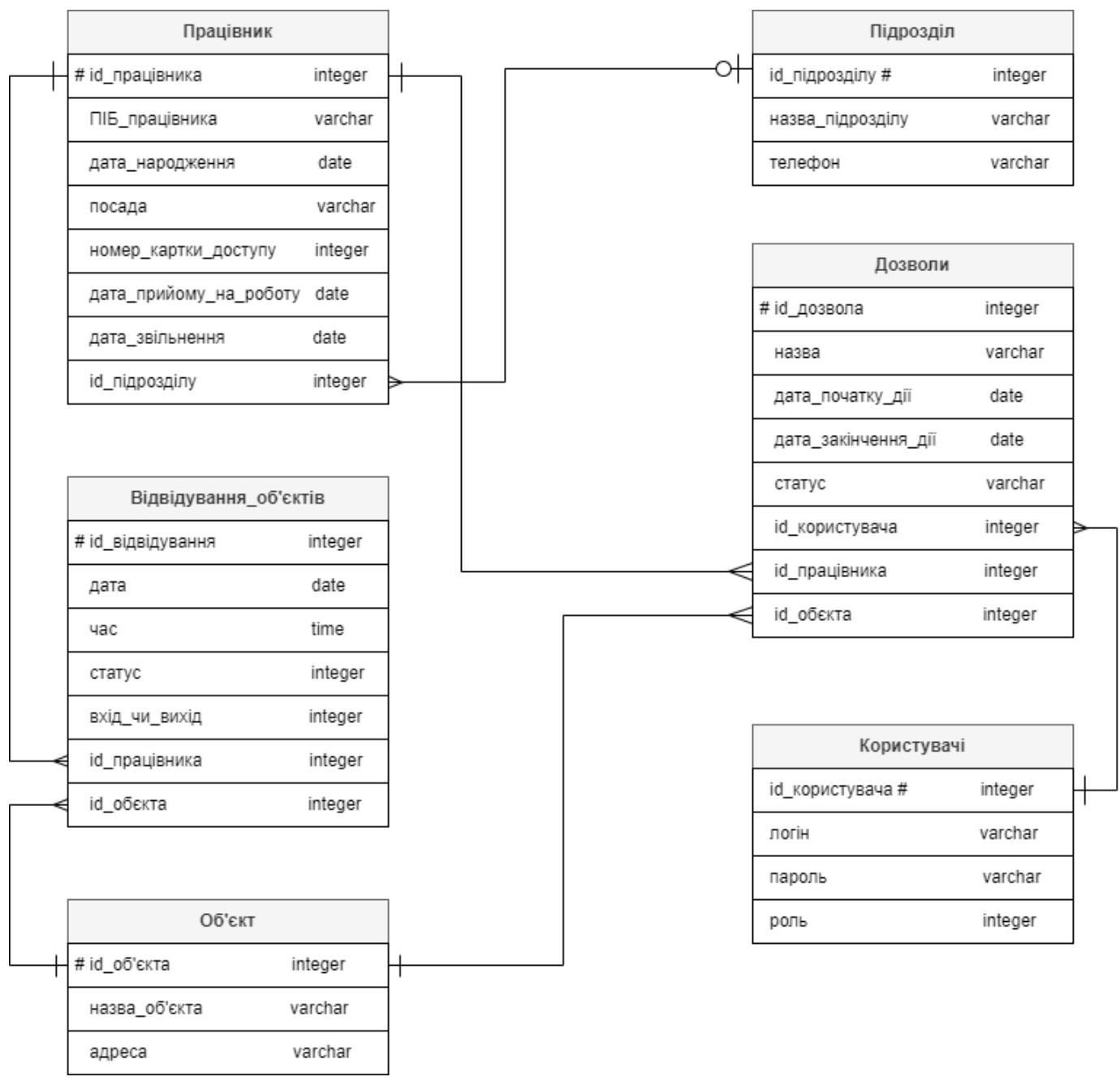


Рисунок 10 – Схема реляційної бази даних (рисунок виконано самостійно)

При зведенні до третьої нормальної форми була проведена перевірка на відсутність транзитивних функціональних залежностей.

Отже, отримана схема БД знаходиться в 3НФ.

3.4 Розробка архітектури системи

Архітектура програмного забезпечення – це спосіб структурування системи та її елементів на певному етапі її роботи. Розробка архітектури системи програмного додатку, є критично важливим етапом, що визначає загальну структуру, ефективність, масштабованість і безпеку системи. Основою для розробки архітектури є вибір архітектурного стилю.

У даному випадку найбільш підходящим варіантом є клієнт-серверна архітектура. Цей стиль забезпечує розділення завдань між сервером, який відповідає за збереження та обробку даних, і клієнтом, який надає інтерфейс користувача для взаємодії з системою. У ролі сервера виступає реляційна база даних, а в ролі клієнта програмний застосунок.

Архітектура програмної системи зображена на рисунку 11.

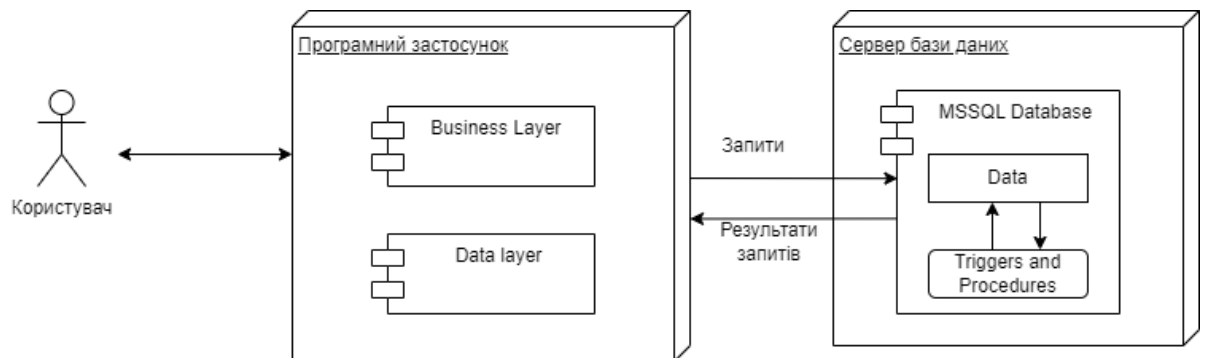


Рисунок 11 – Загальна схема архітектури програмної системи (рисунок виконано самостійно)

Клієнтська частина відповідальна за взаємодію з користувачем. Вона може бути реалізована як веб-додаток, мобільний додаток або настільний додаток. Основні завдання клієнтської частини:

- введення та відображення даних користувачем;
- надсилання запитів до сервера;
- обробка відповідей від сервера.

Настільні додатки (desktop applications) мають ряд переваг у порівнянні з веб- і мобільними додатками, які можуть зробити їх кращим вибором для певних завдань і середовищ.

Для реалізації нашої програмної системи було обрано настільний додаток, оскільки він має більш високу продуктивність за рахунок прямого доступу до ресурсів комп'ютера, таких як процесор, оперативна пам'ять, графічна карта, що забезпечує більш швидку обробку даних. Також настільний додаток використовує вбудовані механізми операційної системи для контролю доступу до даних і ресурсів, що підвищує загальний рівень безпеки, що в нашому випадку є доволі критичним. Оскільки для програмного застосунку не потрібно використовувати браузер, то будуть відсутні ризики, пов'язані з веб-уразливостями, такими як XSS або CSRF.

Серверна частина - база даних зберігає всі необхідні дані для функціонування системи. Вибір типу бази даних (реляційна або нереляційна) залежить від специфічних вимог додатку. Основні завдання бази даних:

- збереження структурованих даних;
- забезпечення швидкого доступу до даних;
- підтримка цілісності та консистентності даних.

Для реалізації нашої програмної системи було обрано реляційну базу даних Microsoft SQL Server (MSSQL) оскільки вона є потужною реляційною системою управління базами даних (RDBMS), яка надає широкий спектр можливостей для обробки даних, аналітики та забезпечення безпеки [9]. Основними переваги використання MSSQL є:

- надійність та висока продуктивність;
- масштабованість;
- безпека;
- інтеграція з іншими продуктами Microsoft;
- зручність адміністрування;
- підтримка високої доступності.

На серверній частині буде реалізована логіка зміни статусу доступу за допомогою тригерів та процедур, які виконуються за розкладом.

3.5 Створення UI / UX дизайну системи

Створення UI/UX дизайну для настільного застосунку є комплексним процесом, що включає вивчення потреб користувачів, проектування інтерфейсу, розробку візуального дизайну та юзабіліті-тестування. Використання системного підходу до розробки дизайну забезпечує створення інтуїтивно зрозумілого, зручного та ефективного користувацького інтерфейсу, що сприяє підвищенню задоволеності користувачів та успіху продукту на ринку.

На першому етапі розробимо схему навігації, що відображає структуру застосунку та взаємозв'язки між різними розділами і функціями. Схема навігації є ключовим елементом дизайну користувацького інтерфейсу (UI), який визначає структуру та взаємозв'язки між різними екранами та розділами застосунку. Ефективна навігація дозволяє користувачам легко знаходити необхідну інформацію та виконувати свої завдання.

Аналізуючи користувачів та сценарії, які користувачі повинні виконувати у застосунку, визначимо наступні основні розділи та функції:

- працівники підприємства: користувачі, відповідно до їхньої ролі будуть переглядати, додавати, редагувати та видаляти дані;
- підрозділи підприємства: користувачі, відповідно до їхньої ролі будуть переглядати, додавати, редагувати та видаляти дані про підрозділи підприємства, а також виставляти працівникам підрозділ, в якому вони працюють;
- об'єкти підприємства: користувачі, відповідно до їхньої ролі будуть переглядати, додавати, редагувати та видаляти дані про об'єкти підприємства;

- дозволи: користувачі, відповідно до їхньої ролі будуть переглядати, додавати, редагувати та видаляти дані про дозволи;
- відвідування об'єктів підприємства: користувачі, відповідно до їхньої ролі будуть переглядати дані про відвідування працівниками об'єктів підприємства;
- звіти: користувачі, відповідно до їхньої ролі будуть формувати та переглядати доступні звіти;
- статистики: користувачі, відповідно до їхньої ролі будуть формувати та переглядати доступні статистики;
- імпорт даних з зовнішньої системи: користувачі, відповідно до їхньої ролі будуть імпортувати дані з файлу Excel;
- аутентифікація та авторизація: для роботи в системі користувачам потрібно пройти аутентифікацію і авторизацію.

Схема навігації представлена на рисунку 12.

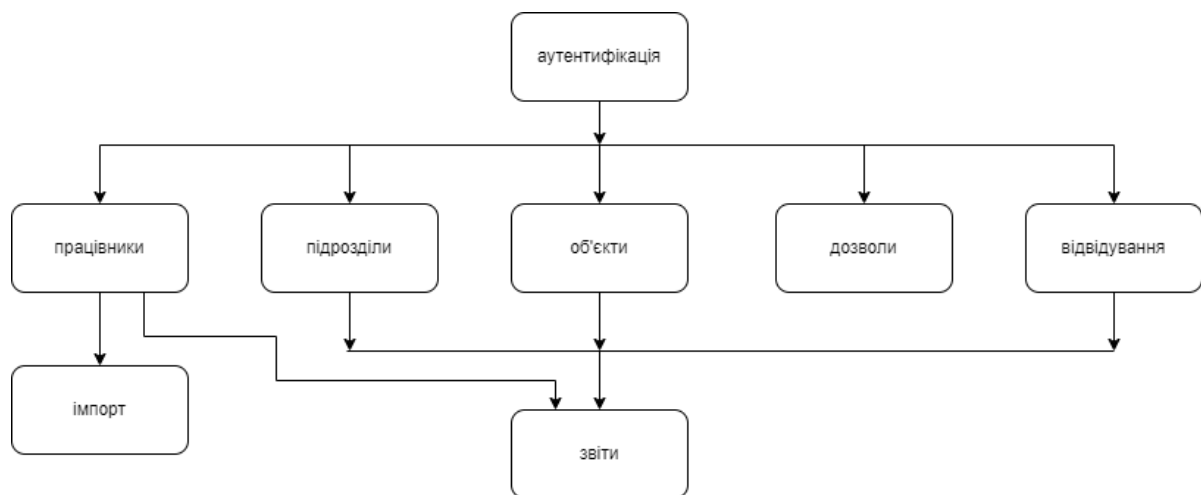


Рисунок 12 – Схема навігації програмної системи (рисунок виконано самостійно)

На рисунку 13 наведено макет головного екрану.

Access and Working Time Accounting - Manager access

Файл Перегляд Редагування Відвідування Доступ Звіти Статистики

Працівники

ПІБ Підрозділ Усі

Посада Дата прийняття на роботу з 3 липня 2024 р. по 3 липня 2024 р.

ID	Назва підрозділу	ПІБ	Дата народження	Посада	Номер картки	Дата прийняття	Дата звільнення
1	Виробничий відділ	Портнов Віктор Іванович	20.01.1978	Робітник	12301	01.05.2005	01.01.2100
2	Бухгалтерія	Сидоровська Олена Ва...	02.04.1970	Бухгалтер	12308	20.07.2003	01.01.2100
3	Дирекція	Тихонов Петро Васильо...	17.12.1999	Менеджер	12307	10.04.2019	01.01.2100
4	Технічний відділ	Петриковський Іван Ро...	20.05.1998	Робітник	12305	15.03.2018	01.01.2100
6	ІТ відділ	Бурлака Юрій Радимович	01.10.1999	ІТ спеціаліст	12306	21.11.2019	01.01.2100
8	Технічний відділ	Міховський Йосеф Ма...	03.06.1994	Робітник	12308	12.03.2023	01.01.2100
9	Контроль якості	Демчина Максим Зоря...	26.01.1989	Робітник	12309	14.11.2009	01.01.2100
10	Відділ фінансового забезпечення	Маланок Федір Вікторо...	14.11.1992	Фінансист першої кате...	12310	21.11.2009	01.01.2100
11	Бухгалтерія	Гасюк Едуард Азарович	19.09.2001	Бухгалтер першої катег...	12311	14.11.2022	01.01.2100
12	Бухгалтерія	Голубинський Кузьма П...	13.10.1992	Бухгалтер другої катего...	12312	06.02.2020	01.01.2100
13	Відділ фінансового забезпечення	Міцкевич Лідія Яромирі...	17.07.1985	Провідний фінансист	12313	16.06.2004	01.01.2100
14	Відділ фінансового забезпечення	Пшенична Анжеліка Тар...	20.01.1989	Фінансист	12314	14.11.2023	01.01.2100
15	Виробничий відділ	Павлів Валерій Давидо...	13.06.1984	Токар	12315	23.01.2015	01.01.2100
16	Виробничий відділ	Захаренко Єгор Милосл...	30.06.1984	Токар	12316	14.11.2023	01.01.2100
17	Виробничий відділ	Копитко Едуард Янович	14.11.2023	Електромонтер	12317	14.06.2017	01.01.2100
18	Охорона	Токарчук Євген Добро...	27.11.1959	Охоронець	12318	18.11.2010	01.01.2100

Рисунок 13 – Макет головного екрану (рисунок виконано самостійно)

Прийняті проектні рішення будуть покладені в основу подальшої програмної реалізації системи.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

4.1 Вибір засобів програмної реалізації

Для розробки програмної системи необхідно ретельно обрати відповідні технології, щоб забезпечити досягнення поставлених цілей під час проектування. Деякі з них вже були згадані на етапах проектування та аналізу системи.

В ролі серверної частини використовується СКБД MSSQL. Ця СКБД була обрана як оптимальна в рамках виконання даної задачі, так як вона відповідає усім нашим вимогам, а саме:

- реляційна структура - MSSQL зберігає дані у вигляді таблиць зі стовпцями і рядками. Вона підтримує відносини між таблицями, що дозволяє ефективно управляти даними і забезпечує цілісність даних;
- мова запитів - MSSQL використовує мову запитів SQL (Structured Query Language) для взаємодії з базою даних. SQL надає потужні можливості для створення, читання, оновлення та видалення даних;
- підтримка індексів: MSSQL підтримує індекси, які допомагають прискорити пошук та фільтрацію даних. Індекси дозволяють ефективно виконувати запити і забезпечують швидкий доступ до даних;
- транзакції: MSSQL підтримує транзакції, що забезпечують атомарність, консистентність, ізолюваність та стійкість (ACID properties) під час операцій з базою даних. Це важливо для забезпечення цілісності даних та уникнення проблем конкуренції при одночасному доступі до даних;
- масштабованість: MSSQL дозволяє масштабувати базу даних, починаючи з невеликих проектів і розширюючись до великих розподілених систем. Вона підтримує горизонтальне масштабування (шарування) і вертикальне масштабування (покращення обладнання).
- безпека: MSSQL надає можливості для захисту даних, включаючи аутентифікацію користувачів, контроль доступу та шифрування даних. Вона

дозволяє налаштувати рівні доступу для різних користувачів і забезпечує захист від несанкціонованого доступу до даних.

Для створення клієнтського додатку було обрано технологію Windows Forms на мові програмування C#. Це рішення прийнято на основі кількох важливих факторів, що забезпечують ефективну реалізацію вимог проекту:

а) простота та зручність розробки:

- 1) швидкий старт: Windows Forms надає зручне та інтуїтивно зрозуміле середовище для створення графічних інтерфейсів. Інструменти, такі як Visual Studio, дозволяють швидко почати розробку, використовуючи візуальний редактор форм;
- 2) багата бібліотека компонентів: Windows Forms включає велику кількість стандартних компонентів (кнопок, текстових полів, меню та ін.), які можна легко використовувати та налаштувати для створення зручного користувацького інтерфейсу;

б) інтеграція з MSSQL:

- 1) ADO.NET: Windows Forms чудово інтегрується з ADO.NET, що дозволяє ефективно працювати з базою даних MSSQL. Це забезпечує надійну та швидку взаємодію з серверною частиною, включаючи виконання запитів, збереження та отримання даних;
- 2) Entity Framework: Використання Entity Framework спрощує роботу з даними за допомогою ORM (об'єктно-реляційного відображення), що зменшує кількість коду та підвищує продуктивність розробників [10];

в) висока продуктивність:

- 1) легка вага: Windows Forms є легкою технологією з невеликими системними вимогами, що дозволяє розробляти швидкі та ефективні додатки;
- 2) швидка обробка: використання C# забезпечує високу продуктивність виконання коду, що є важливим для інтерактивних настільних додатків;

г) зручність у підтримці та оновленні:

- 1) зрозумілий код: Розробка на C# з використанням Windows Forms дозволяє створювати зрозумілий та структурований код, що полегшує підтримку та оновлення додатку;
- 2) модульність: Windows Forms підтримує модульний підхід до розробки, що спрощує впровадження нових функцій та змін.

4.2 Призначення і логічна структура

З точки зору функціоналу, в проекті реалізовано управління трьома базовими сутностями – працівниками, підрозділами та об'єктами підприємства.

Структура розділена наступним чином. Головна форма (Main Form) - це головне вікно програми, яке містить основні елементи керування та відображає головний інтерфейс користувача. На головній формі розташоване меню та відображено основні сутності: дані по працівникам, підрозділам та об'єктам.

Форми для введення даних (Data Input Forms) - ці форми використовуються для введення даних користувачем. Наприклад, форма для створення нового запису, редагування існуючого запису або введення параметрів для пошуку. Форми для введення даних містить текстові поля, випадаючі списки, елементи роботи з датами, кнопки підтвердження тощо.

Форми для відображення даних (Data Display Forms) - ці форми використовуються для відображення даних з бази даних через дата-адаптери та дата-сети.

Форма авторизації - форма для введення логіну та паролю користувача.

Форми для статистик - окремі форми для відображення необхідних статистик.

Форми для звітів - окремі форми, які дозволяють формувати необхідні звіти та експортувати отримані дані до MS Excel.

Структура коду проекту зображена на рисунку 14. Різний функціонал зберігається в окремих файлах, котрі на етапі компіляції та зборки проекту об'єднуються в єдиний файл.

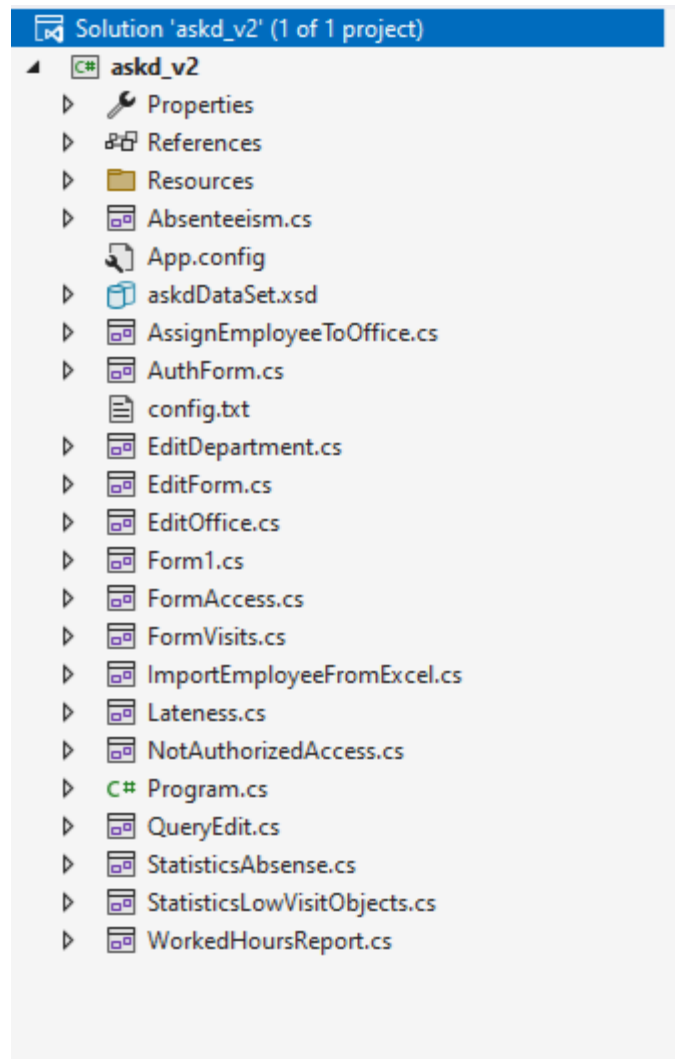


Рисунок 14 - Структура коду проекту (рисунок виконано самостійно)

Основна точка входу в програму знаходиться в файлі Program.cs, де міститься клас Program та метод Main(). В цьому місці визначається глобальна змінна globalrole, яка відповідає за роль користувача. В програмі передбачено 3 ролі: менеджер, працівник відділу кадрів та служба безпеки. Після запуску програми відображається основна форма MainForm. На початку завантаження основної форми

MainForm встановлюється підключення до бази даних, підвантажуються необхідні дані і для користувача відображається форма авторизації (AuthenticationForm).

Якщо користувач вірно введе логін та пароль, то форма авторизації зникне і користувачу буде присвоєно одну з ролей: менеджер, співробітник відділу кадрів чи служба безпеки. В іншому випадку для користувача буде виведено сповіщення про невірні логін чи пароль.

Підключення до бази даних MS SQL Server відбувається за технологією ADO.Net. Для цього в додатку використовуються наступні елементи: DataSet та BindingSource. Відображення даних з таблиць БД відбувається за допомогою компонентів DataGridView та TableAdapter. За допомогою об'єкта Connection відбувається встановлення підключення до джерела даних. Об'єкт Command дозволяє виконувати операції з даними БД. Об'єкт DataReader зчитує отримані в результаті запиту дані. Об'єкт DataSet призначений для зберігання даних із БД та дозволяє працювати з ними незалежно від БД. І об'єкт DataAdapter є посередником між DataSet та джерелом даних.

4.3 Описання фізичної моделі бази даних

При реалізації фізичної моделі бази даних було створено шість таблиць. Нижче наведено команди на їх створення.

```
CREATE TABLE department (
  department_id INT IDENTITY,
  dep_name VARCHAR(100) NOT NULL,
  dep_phone VARCHAR (15),
  PRIMARY KEY (department_id));
```

```
CREATE TABLE employee (
  employee_id INT IDENTITY,
  PIB VARCHAR (100),
  date_of_birth DATE,
  job_title VARCHAR (100),
  card_number INT,
  hire_date DATE,
  dismissal_date DATE,
```

```

department_id INT REFERENCES department(department_id) ON DELETE SET NULL ON
UPDATE CASCADE,
PRIMARY KEY (employee_id));

```

```

CREATE TABLE office (
office_id INT IDENTITY,
office_name VARCHAR (100),
office_address VARCHAR (100),
PRIMARY KEY (office_id));

```

```

CREATE TABLE access (
access_id INT IDENTITY,
access_name VARCHAR (100) NOT NULL,
access_start DATE,
access_end DATE,
access_status VARCHAR (20) NOT NULL,
access_users INT REFERENCES users(users_id) ON DELETE CASCADE,
employee_id INT REFERENCES employee(employee_id) ON DELETE SET NULL ON UPDATE
CASCADE,
office_id INT REFERENCES office(office_id) ON DELETE SET NULL ON UPDATE
CASCADE,
PRIMARY KEY (access_id));

```

```

CREATE TABLE visit (
visit_id INT IDENTITY,
visit_date DATE,
visit_time TIME,
status INT,
enter_exit INT,
employee_id INT REFERENCES employee(employee_id) ON DELETE CASCADE,
office_id INT REFERENCES office(office_id) ON DELETE CASCADE,
PRIMARY KEY (visit_id));

```

```

CREATE TABLE users (
users_id INT IDENTITY,
users_login VARCHAR (100) NOT NULL,
users_pass VARCHAR (100) NOT NULL,
users_role INT,
PRIMARY KEY (users_id));

```

4.4 Опис програмної реалізації серверної частини системи

На першому етапі, коли доступ створюється співробітником служби безпеки, він може мати статус лише ‘створений’ або ‘перевірений’.

На серверній частині створимо тригер, який унеможливить додавання доступу з іншими статусами.

```

CREATE OR ALTER TRIGGER trg_check_insert_access_status
ON access
AFTER INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE access_status NOT IN ('створений', 'перевірений')
    )
    BEGIN
        UPDATE access
        SET access_status = 'створений'
        FROM access a
        JOIN inserted i ON a.access_id = i.access_id
        WHERE i.access_status NOT IN ('створений', 'перевірений');
    END
END;

```

Цей тригер призначений для контролю додавання нових доступів і дозволяє додати доступ тільки зі статусами 'створений' або 'перевірений'.

На наступному етапі, після того як доступ був відправлений на затвердження, він може бути або підтверджений або відхилений. Для цього створимо тригер, який контролює переведення доступу зі статусу 'перевірений' в статус 'підтверджений' або 'відхилений'.

```

CREATE TRIGGER trg_update_approve_access_status
ON access
AFTER UPDATE
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN deleted d ON i.access_id = d.access_id
        WHERE d.access_status = 'перевірений'
        AND i.access_status NOT IN ('підтверджений', 'відхилений')
    )
    BEGIN
        UPDATE access
        SET access_status = 'перевірений'
        FROM access a
        JOIN inserted i ON a.access_id = i.access_id
        JOIN deleted d ON a.access_id = d.access_id
        WHERE d.access_status = 'перевірений'
    END
END;

```

```

        AND i.access_status NOT IN ('підтверджений', 'відхилений');
    END
END;

```

На наступному етапі, якщо дозвіл був затверджений, та настала дата початку дії дозволу, то дозвіл переводиться у статус ‘дійсний’. Для цього створимо процедуру UpdateAccessStatusToActive, яка буде запускатись кожного дня за розкладом:

```

CREATE PROCEDURE UpdateAccessStatusToActive
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE access
    SET access_status = 'дійсний'
    WHERE access_status = 'підтверджений'
    AND access_start = CAST(GETDATE() AS DATE);
END;

```

Створимо завдання в SQL Server Agent з назвою «Execute UpdateAccessStatusToActive», яке буде виконуватись раз на день о 3й годині. Таким чином, ми автоматизуємо процес зміни access_status з 'підтверджений' на 'дійсний' щодня, якщо access_start відповідає сьогоднішній даті.

На наступному етапі, якщо у дійсного доступу закінчився термін дії, його необхідно перевести в статус ‘недійсний’. Для цього створимо процедуру “UpdateAccessStatusToInactive”:

```

CREATE PROCEDURE UpdateAccessStatusToInactive
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE access
    SET access_status = 'недійсний'
    WHERE access_status = 'дійсний'
    AND access_end < CAST(GETDATE() AS DATE);
END;

```

Створимо нове завдання в SQL Server Agent з назвою «Execute UpdateAccessStatusToInactive», яке буде виконуватись раз на день о 4й годині. . Таким чином, ми автоматизуємо процес зміни access_status з 'дійсний' на 'недійсний' щодня, якщо access_end менший за сьогоднішню дату.

4.5 Опис інтерфейсу користувача

Взаємодія користувача з програмним застосунком відбувається за допомогою графічного інтерфейсу. Після запуску програмного застосунку користувачеві буде відображено форму аутентифікації (див. рис. 15).

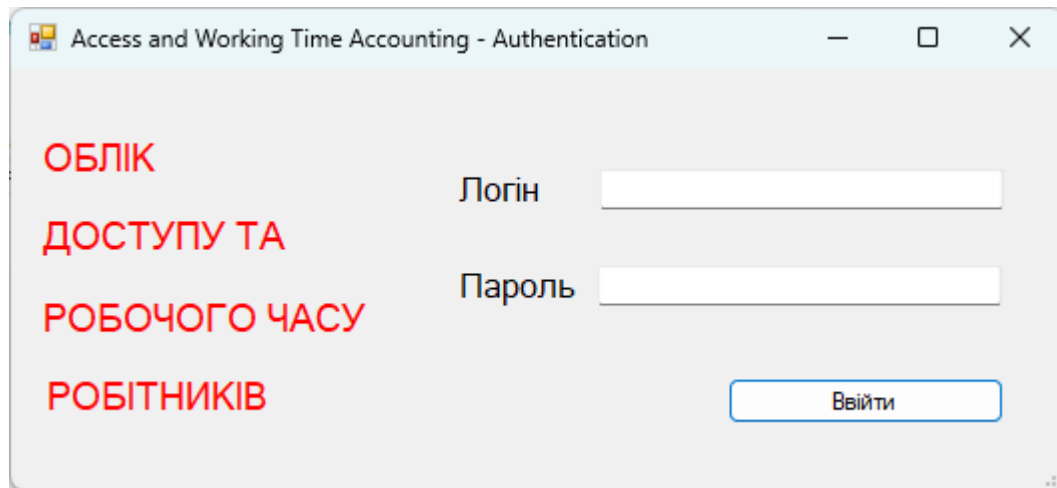


Рисунок 15 – Форма аутентифікації (рисунок виконано самостійно)

Після вводу коректного логіну та паролю користувачеві буде відображено головний екран, на якому відображається список працівників підприємства. У верхній частині розташоване меню для переходу в інші розділи системи. Далі розташовані поля пошуку та фільтрації, а нижче таблиця зі списком працівників та пов'язаною з ними інформацією (див.рис.16).

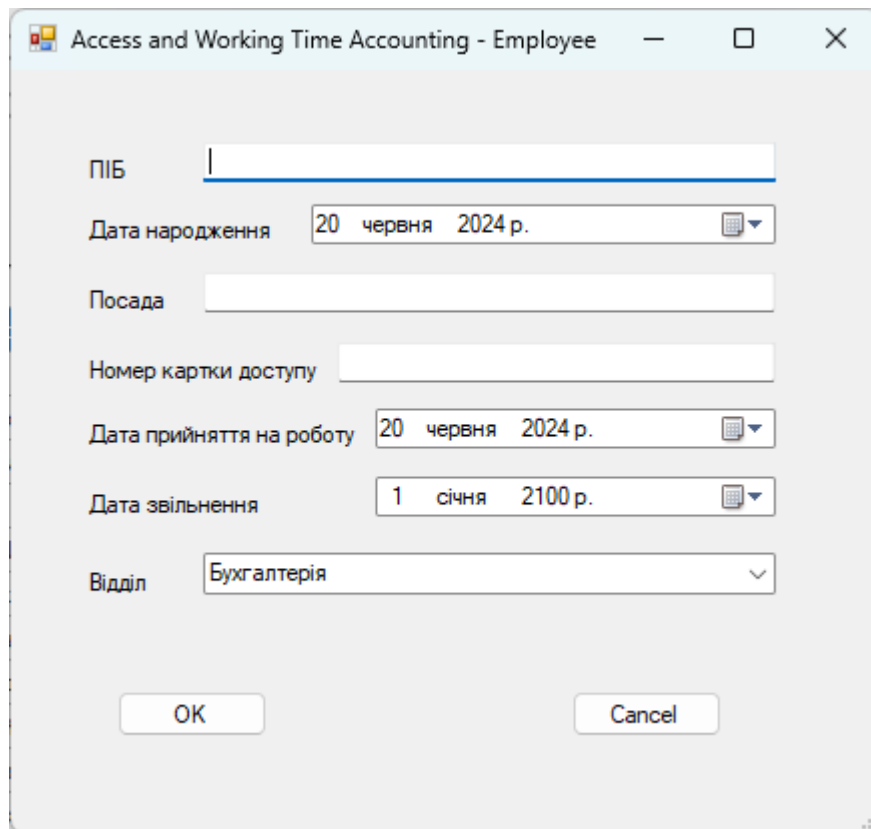
ID	Назва підрозділу	ПІБ	Дата народження	Посада	Номер картки	Дата прийняття	Дата звільнення
1	Виробничий відділ	Портнов Віктор Іванович	20.01.1978	Робітник	12301	01.05.2005	01.01.2100
2	Бухгалтерія	Сидоровська Олена Ва...	02.04.1970	Бухгалтер	12308	20.07.2003	01.01.2100
3	Дирекція	Тихонов Петро Васильо...	17.12.1999	Менеджер	12307	10.04.2019	01.01.2100
4	Технічний відділ	Петриковський Іван Ро...	20.05.1998	Робітник	12305	15.03.2018	01.01.2100
6	ІТ відділ	Бурлака Юрій Радимович	01.10.1999	ІТ спеціаліст	12306	21.11.2019	01.01.2100
8	Технічний відділ	Міхновський Йосеф Ма...	03.06.1994	Робітник	12308	12.03.2023	01.01.2100
9	Контроль якості	Демчина Максим Зоря...	26.01.1989	Робітник	12309	14.11.2009	01.01.2100
10	Відділ фінансового забезпечення	Маланок Федір Вікторо...	14.11.1992	Фінансист першої кате...	12310	21.11.2009	01.01.2100
11	Бухгалтерія	Гасюк Едуард Азарович	19.09.2001	Бухгалтер першої катег...	12311	14.11.2022	01.01.2100
12	Бухгалтерія	Голубинський Кузьма П...	13.10.1992	Бухгалтер другої катего...	12312	06.02.2020	01.01.2100
13	Відділ фінансового забезпечення	Міцкевич Лідія Яромирі...	17.07.1985	Провідний фінансист	12313	16.06.2004	01.01.2100
14	Відділ фінансового забезпечення	Пшенична Анжеліка Тар...	20.01.1989	Фінансист	12314	14.11.2023	01.01.2100
15	Виробничий відділ	Павлів Валерій Давидо...	13.06.1984	Токар	12315	23.01.2015	01.01.2100
16	Виробничий відділ	Захаренко Єгор Милосл...	30.06.1984	Токар	12316	14.11.2023	01.01.2100
17	Виробничий відділ	Копитко Едуард Янович	14.11.2023	Електромонтер	12317	14.06.2017	01.01.2100
18	Охорона	Токарчук Євген Добро...	27.11.1959	Охоронець	12318	18.11.2010	01.01.2100

Рисунок 16 – Форма роботи зі списком працівників (рисунок виконано самостійно)

Для відображення списку підрозділів підприємства необхідно обрати меню «Перегляд» та перейти в пункт «Підрозділи». Після цього буде відображено форму з підрозділами підприємства. Для відображення інформації по об'єктах підприємства необхідно обрати пункт меню «Перегляд» та перейти в підменю «Об'єкти».

Для редагування інформації необхідно перейти в меню «Редагування» та обрати один з пунктів підменю «Додати», «Видалити» або «Змінити» в залежності від необхідної дії. Також редагування доступне за допомогою контекстного меню при натисканні правої кнопки миші.

Для додавання нового працівника необхідно на формі перегляду працівників перейти в пункт меню «Редагування» та обрати пункт підменю «Додати». Буде відображено форму з необхідними для заповнення полями (див.рис.17).



Access and Working Time Accounting - Employee

ПІБ

Дата народження 20 червня 2024 р.

Посада

Номер картки доступу

Дата прийняття на роботу 20 червня 2024 р.

Дата звільнення 1 січня 2100 р.

Відділ Бухгалтерія

OK Cancel

Рисунок 17 – Додавання нового працівника (рисунок виконано самостійно)

Після заповнення усіх необхідних полів і натисканні кнопки «ОК», інформація про нового працівника буде збережена в базі даних і з'явиться на формі перегляду працівників.

Аналогічним чином відбуваються дії з сутностями «Підрозділи» та «Об'єкти». Для перегляду відвідування об'єктів працівниками користувачу необхідно перейти в пункт меню «Відвідування» і обрати підменю «Відвідування об'єктів». На екрані з'явиться форма зі списком відвідування об'єктів працівниками за обрану дату (див.рис.18).

The screenshot shows a software window titled "Access and Working Time Accounting - Visits". At the top, there is a date selector set to "20 червня 2024 р." and a search field for "ПІБ" (Last Name, First Name, Middle Name) with "Очистити" and "Шукати" buttons. Below is a table with columns: Дата, Час, Статус, Вхід/Вихід, ПІБ, and Об'єкт. The table contains 20 rows of visit records for the date 20.06.2024, all with a status of "успіх" (success) and a type of "вхід" (arrival). The objects visited include "Головний офіс" and "Виробниче приміщення №1".

Дата	Час	Статус	Вхід/Вихід	ПІБ	Об'єкт
20.06.2024	07:32:45	успіх	вхід	Портнов Віктор Іванович	Головний офіс
20.06.2024	07:35:08	успіх	вхід	Демчина Максим Зорянович	Головний офіс
20.06.2024	07:35:26	успіх	вхід	Майданюк Злата Валентинівна	Головний офіс
20.06.2024	07:38:07	успіх	вхід	Костенко Едуард Йосипович	Головний офіс
20.06.2024	07:39:59	успіх	вхід	Токарчук Євген Добромірович	Головний офіс
20.06.2024	07:41:06	успіх	вхід	Лушик Олег Юліанович	Виробниче приміщення №1
20.06.2024	07:46:46	успіх	вхід	Голубинський Кузьма Пилипович	Головний офіс
20.06.2024	07:47:41	успіх	вхід	Міцкевич Лідія Яромірівна	Головний офіс
20.06.2024	07:49:47	успіх	вхід	Павлів Валерій Давидович	Головний офіс
20.06.2024	07:49:51	успіх	вхід	Міхновський Йосиф Максимович	Головний офіс
20.06.2024	07:51:40	успіх	вхід	Рубани Артур Фролович	Головний офіс
20.06.2024	08:01:00	успіх	вхід	Тихонов Петро Васильович	Головний офіс
20.06.2024	08:03:07	успіх	вхід	Петриковський Іван Романович	Головний офіс
20.06.2024	08:04:44	успіх	вхід	Копитко Едуард Янович	Головний офіс
20.06.2024	08:07:22	успіх	вхід	Сидоровська Олена Василівна	Виробниче приміщення №1
20.06.2024	08:10:08	успіх	вхід	Захаренко Єгор Милославович	Головний офіс
20.06.2024	08:22:17	успіх	вхід	Пшенична Анжеліка Тарасівна	Головний офіс
20.06.2024	08:24:38	успіх	вхід	Бурлака Юрій Радимович	Головний офіс
20.06.2024	08:24:46	успіх	вхід	Парфенюк Євген Валентинович	Головний офіс
20.06.2024	08:24:47	успіх	вхід	Гасюк Едуард Азарович	Виробниче приміщення №1

Рисунок 18 – Відвідування об’єктів працівниками (рисунок виконано самостійно)

Користувач має можливість переглянути відвідування за будь-яку дату, вказавши необхідну дату у відповідному полі, та має можливість знайти інформацію про конкретного працівника використовуючи пошук по прізвищу.

Для того, щоб потрапити на конкретний об’єкт підприємства, працівнику необхідно видати доступ до цього об’єкта. У разі, якщо працівник має дійсний доступ до об’єкту, то на формі відвідувань візит на об’єкт буде позначений як успішний. В протилежному випадку візит на об’єкт буде позначено як «відмова». Для перегляду наявних дозволів необхідно перейти в пункт меню «Доступ» і далі обрати підменю «Доступ на об’єкти», після чого буде відображено форму зі списком наявних дозволів (див.рис.19).

ID	Назва	Дата початку	Дата закінчення	ПІБ	Об'єкт	Статус
2	Доступ	15.06.2023	15.06.2024	Сидоровська Олена Василівна	Головний офіс	дійсний
3	Доступ 1	15.06.2023	15.06.2024	Портнов Віктор Іванович	Виробниче приміщення №1	дійсний
4	Доступ на склад	12.06.2023	01.11.2023	Портнов Віктор Іванович	Склад №1	дійсний
5	Доступ на склад 2	14.06.2023	16.06.2024	Бурлака Юрій Радимович	Склад №2	дійсний
6	Доступ на виробниче	15.06.2023	15.06.2024	Тихонов Петро Васильович	Виробниче приміщення №1	дійсний
7	новий доступ	03.12.2023	03.12.2024	Демчина Максим Зорянович	Головний офіс	дійсний
8	access	10.12.2023	12.12.2023	Портнов Віктор Іванович	Головний офіс	дійсний
9	access2	10.12.2023	12.12.2023	Портнов Віктор Іванович	Головний офіс	дійсний
11	Доступ 11	05.06.2024	05.06.2025	Невров Олексій Петрович	Склад №2	перевірений
1010	Доступ	01.06.2024	31.12.2026	Тихонов Петро Васильович	Головний офіс	створений
1011	Доступ	01.06.2024	31.12.2026	Петраковський Іван Романович	Головний офіс	створений
1012	Доступ	01.01.2024	31.12.2026	Портнов Віктор Іванович	Головний офіс	дійсний
1013	Доступ	01.01.2024	31.12.2026	Сидоровська Олена Василівна	Головний офіс	дійсний
1014	Доступ	01.01.2024	31.12.2026	Тихонов Петро Васильович	Головний офіс	дійсний
1015	Доступ	01.01.2024	31.12.2026	Петраковський Іван Романович	Головний офіс	дійсний
1017	Доступ	01.01.2024	31.12.2026	Бурлака Юрій Радимович	Головний офіс	дійсний
1019	Доступ	01.01.2024	31.12.2026	Мішовський Йосиф Максимович	Головний офіс	дійсний
1020	Доступ	01.01.2024	31.12.2026	Демчина Максим Зорянович	Головний офіс	дійсний
1021	Доступ	01.01.2024	31.12.2026	Маланок Федір Вікторович	Головний офіс	дійсний

Рисунок 19 – Доступи на об'єкти (рисунок виконано самостійно)

Для створення нового доступу на об'єкти необхідно натиснути кнопку «Додати доступ», і як результат, буде відображена форма додавання нового доступу (див.рис.20).

Access and Working Time Accounting - Assign Employee to Office

Працівник: Бурлака Юрій Радимович

Об'єкт: Автопарк

Дата початку: 20 червня 2024 р.

Дата закінчення: 20 червня 2025 р.

Назва: Доступ

Зберегти

Відправити на затвердження

Рисунок 20 – Створення нового доступу (рисунок виконано самостійно)

Працівник служби безпеки, в особі 1 має змогу створити новий доступ та зберегти його для подальшого редагування або відправити на затвердження. Після відправки на затвердження доступ з'явиться у загальному списку доступів з статусом «перевірений». Право затвердити доступ має будь-який інший працівник

служби безпеки. При спробі затвердити доступ створений власноруч, користувач отримає повідомлення про неможливість даної операції.

Дійсні доступи не підлягають редагуванню, але можуть бути скасовані. Для скасування будь-якого дійсного доступу необхідно обрати відповідний доступ, та використовуючи контекстне меню обрати пункт «скасувати доступ». В цьому разі доступ перейде в статус «недійсний», а кінцева дата дії дозволу оновиться на поточну. Недійсні дозволи, кінцева дата яких перевищує один місяць будуть видалені з системи автоматично.

У меню «Звіти» доступні два типи звітів: звіт з несанкціонованого доступу та звіт з відпрацьованих годин.

У звіті з несанкціонованого доступу відображається інформація по працівниках які намагались потрапити на об'єкт підприємства без наявного дозволу на доступ (див.рис.21). Після перегляду інформації за обраний місяць та рік – є можливість експорту даних в Microsoft Excel.

Дата	Час	Статус	Вхід/Вихід	ПІБ	Об'єкт
03.06.2024	14:14:20	відмова	вихід	Костенко Едуард Йосипович	Склад №1

Рисунок 21 – Несанкціонований доступ на об'єкти (рисунок виконано самостійно)

У звіті з відпрацьованих годин відображається інформація по кожному працівнику з кількістю відпрацьованих годин за обраний місяць (див.рис.22). За необхідності інформацію можна експортувати до Microsoft Excel.

ПІБ	Підрозділ	Місяць, Рік	Відпрацьовані Години
Портнов Віктор Іванович	Виробничий відділ	06.2024	181
Сидоровська Олена Василівна	Бухгалтерія	06.2024	187
Тихонов Петро Васильович	Дирекція	06.2024	190
Петриковський Іван Романович	Технічний відділ	06.2024	190
Бурлака Юрій Радимович	ІТ відділ	06.2024	181
Міхновський Йосеф Максимович	Технічний відділ	06.2024	189
Демчина Максим Зорянович	Контроль якості	06.2024	191
Маланюк Федір Вікторович	Відділ фінансового забезпечення	06.2024	190
Гасюк Едуард Азарович	Бухгалтерія	06.2024	191
Голубинський Кузьма Пилипович	Бухгалтерія	06.2024	194
Міцкевич Лідія Яромірівна	Відділ фінансового забезпечення	06.2024	188
Пшенична Анжеліка Тарасівна	Відділ фінансового забезпечення	06.2024	190
Павлів Валерій Давидович	Виробничий відділ	06.2024	191
Захаренко Єгор Милославович	Виробничий відділ	06.2024	190
Копитко Едуард Янович	Виробничий відділ	06.2024	189

Рисунок 22 – Відпрацьовані години протягом місяця (рисунок виконано самостійно)

У розділі статистик відображається інформація про запізнення на робоче місце, відсутності робітника протягом робочого дня, найменш відвідуваним об’єктам та прогулам.

5 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Обґрунтування вибору виду тестування

Після завершення розробки програмної системи необхідно приділити увагу важливому етапу – тестуванню. Існує безліч різних методів тестування, кожен з яких належить до певної категорії. Наприклад, за рівнем тестування виділяють такі підходи, як модульне тестування, інтеграційне та системне.

Зважаючи на архітектуру та специфіку розроблювального програмного засобу, тестування слід розділити на дві основні частини: тестування серверної частини та тестування клієнтської частини системи. Обидві частини проходять модульне тестування.

Модульне тестування – це метод, який передбачає окреме тестування кожного модуля коду програми. Модуль – це найменша частина програми, яку можна протестувати. Кожен метод або функцію програми слід розглядати як модуль, який підлягає тестуванню [11].

Визначивши частини системи для тестування та відповідний підхід, необхідно обрати ступінь тестування. Існують два основні типи тестування: ручне та автоматизоване. Ручне тестування передбачає перевірку тест-кейсів вручну без використання інструментів автоматизації. Автоматизоване тестування, навпаки, використовує набір інструментів для автоматизації процесу перевірки тест-кейсів.

На даному етапі, коли основною метою є виявлення помилок та неточностей у системі, автоматизоване тестування є оптимальним варіантом.

5.2 Опис тестування

Для проведення автоматизованого тестування будемо використовувати MSTest для тестування клієнтської частини та tSQLt для тестування серверної частини. MSTest – це фреймворк для тестування, розроблений компанією Microsoft, який добре інтегрується з екосистемою .NET і Visual Studio. Він є ідеальним вибором для

тестування коду на C#, оскільки надає можливість легко створювати, запускати та аналізувати тести безпосередньо в середовищі розробки. MSTest підтримує різні типи тестування, включаючи модульне та інтеграційне, і дозволяє використовувати анотації для визначення тестових методів, що робить процес тестування більш зрозумілим і організованим.

tSQLt – це фреймворк для модульного тестування бази даних, призначений для роботи з SQL Server. Він дозволяє розробникам писати тести для своїх T-SQL процедур, функцій, тригерів та інших об'єктів бази даних безпосередньо в середовищі SQL Server. Основна мета tSQLt – забезпечити ефективний та автоматизований процес тестування, що сприяє підвищенню якості коду та зменшенню ризику помилок.

Приклад програмної реалізації одного з тест-кейсів для тестування процедури редагування робітників:

```
[TestClass]
public class EditFormTests
{
    private Mock<SqlConnection> mockSqlConnection;
    private Mock<SqlCommand> mockSqlCommand;
    private Mock<SqlDataReader> mockSqlDataReader;

    [TestInitialize]
    public void Setup()
    {
        mockSqlConnection = new
Mock<SqlConnection>(Program.connection_string);
        mockSqlCommand = new Mock<SqlCommand>();
        mockSqlDataReader = new Mock<SqlDataReader>();

        mockSqlCommand.Setup(cmd =>
cmd.ExecuteReader()).Returns(mockSqlDataReader.Object);
        mockSqlConnection.Setup(conn => conn.Open());
        mockSqlConnection.Setup(conn =>
conn.CreateCommand()).Returns(mockSqlCommand.Object);
    }

    [TestMethod]
    public void EditForm_Constructor_ShouldPopulateComboBox()
    {
        // Arrange
```

```

int id = 1;
string pib = "Test PIB";
DateTime birthDay = new DateTime(1990, 1, 1);
string job_title = "Test Job Title";
int card_number = 123456;
DateTime hire_date = new DateTime(2020, 1, 1);
DateTime dismissal_date = new DateTime(2021, 1, 1);
int id_department = 1;

// Setup the mock data reader to return test data
mockSqlDataReader.SetupSequence(reader => reader.Read())
    .Returns(true)
    .Returns(false);
mockSqlDataReader.Setup(reader =>
reader["dep_name"]).Returns("Test Department");
mockSqlDataReader.Setup(reader =>
reader["department_id"]).Returns(id_department);

// Act
var form = new EditForm(id, pib, birthDay, job_title,
card_number, hire_date, dismissal_date, id_department)
{
    SqlConnection = mockSqlConnection.Object
};
// Assert
Assert.AreEqual(1, form.comboBox_Department.Items.Count);
Assert.AreEqual("Test Department",
((ComboBoxItem)form.comboBox_Department.Items[0]).Text);
Assert.AreEqual(id_department.ToString(),
((ComboBoxItem)form.comboBox_Department.Items[0]).Value);
Assert.AreEqual(0, form.comboBox_Department.SelectedIndex);

Assert.AreEqual(pib, form.textBox_PIB.Text);
Assert.AreEqual(birthDay,
form.dateTimePicker_BirthDay.Value);
Assert.AreEqual(job_title, form.textBox_JobTitle.Text);
Assert.AreEqual(card_number.ToString(),
form.textBox_CardNumber.Text);
Assert.AreEqual(hire_date,
form.dateTimePicker_HireDate.Value);
Assert.AreEqual(dismissal_date,
form.dateTimePicker_DismissalDate.Value);
}
}

```

Приклад програмної реалізації одного з тест-кейсів, який призначений для тестування тригера на серверній частині:

```

CREATE PROCEDURE
AccessTests.[test_trg_check_insert_access_status_ShouldSetCreatedStatus]
AS

```

```

BEGIN
    -- Створення тестової таблиці "access"
    EXEC tSQLt.FakeTable @TableName = 'dbo.access';
    -- Вставка тестових даних
    INSERT INTO access (access_name, access_start, access_end,
access_status, access_users, employee_id, office_id)
    VALUES
        ('Test Access 1', '2024-01-01', '2024-12-31', 'новий статус', 1,
1, 1), -- Повинен бути змінений на 'створений'
        ('Test Access 2', '2024-01-01', '2024-12-31', 'створений', 1, 1,
1), -- Повинен залишитися без змін
        ('Test Access 3', '2024-01-01', '2024-12-31', 'перевірений', 1,
1, 1); -- Повинен залишитися без змін

    -- Виконання тригера вручну
    EXEC trg_check_insert_access_status;

    -- Перевірка результатів
    DECLARE @Expected TABLE (
        access_id INT,
        access_name VARCHAR(100),
        access_start DATE,
        access_end DATE,
        access_status VARCHAR(20),
        access_users INT,
        employee_id INT,
        office_id INT
    );

    INSERT INTO @Expected (access_id, access_name, access_start,
access_end, access_status, access_users, employee_id, office_id)
    VALUES
        (1, 'Test Access 1', '2024-01-01', '2024-12-31', 'створений', 1,
1, 1),
        (2, 'Test Access 2', '2024-01-01', '2024-12-31', 'створений', 1,
1, 1),
        (3, 'Test Access 3', '2024-01-01', '2024-12-31', 'перевірений',
1, 1, 1);

    EXEC tSQLt.AssertEqualsTable @Expected, 'access';
END;

```

Приклад програмної реалізації одного з тест кейсів, який призначений для тестування процедури на серверній частині:

```

CREATE PROCEDURE
AccessTests.[test_UpdateAccessStatusToActive_ShouldSetStatusToActive]
AS
BEGIN
    EXEC tSQLt.FakeTable @TableName = 'dbo.access';

```

```

INSERT INTO access (access_id, access_name, access_start,
access_end, access_status, access_users, employee_id, office_id)
VALUES
    (1, 'Test Access 1', CAST(GETDATE() AS DATE), '2024-12-31',
'підтверджений', 1, 1, 1), -- Повинен бути змінений на 'дійсний'
    (2, 'Test Access 2', CAST(GETDATE() AS DATE), '2024-12-31',
'непідтверджений', 1, 1, 1), -- Не повинен бути змінений
    (3, 'Test Access 3', DATEADD(day, -1, CAST(GETDATE() AS DATE)),
'2024-12-31', 'підтверджений', 1, 1, 1); -- Не повинен бути змінений

-- Виконання процедури
EXEC UpdateAccessStatusToActive;

-- Перевірка результатів
DECLARE @Expected TABLE (
    access_id INT,
    access_name VARCHAR(100),
    access_start DATE,
    access_end DATE,
    access_status VARCHAR(20),
    access_users INT,
    employee_id INT,
    office_id INT
);

INSERT INTO @Expected (access_id, access_name, access_start,
access_end, access_status, access_users, employee_id, office_id)
VALUES
    (1, 'Test Access 1', CAST(GETDATE() AS DATE), '2024-12-31',
'дійсний', 1, 1, 1), -- Статус змінено на 'дійсний'
    (2, 'Test Access 2', CAST(GETDATE() AS DATE), '2024-12-31',
'непідтверджений', 1, 1, 1), -- Статус не змінено
    (3, 'Test Access 3', DATEADD(day, -1, CAST(GETDATE() AS DATE)),
'2024-12-31', 'підтверджений', 1, 1, 1); -- Статус не змінено

EXEC tSQLt.AssertEqualsTable @Expected, 'access';
END;

```

Під час тестування було створено 35 різних тест-кейсів, які слід було покрити модульним тестуванням. Усі знайдені помилки та дефекти були виправлені, що призвело до покращення програмної системи.

ВИСНОВКИ

При виконанні кваліфікаційної роботи було спроектовано програмну систему для автоматизації процесу обліку доступу та робочого часу робітників підприємства. Був проведений аналіз проблемної області та конкуруючих систем. Під час аналізу проблемної області були відокремлені проблеми, які слід вирішити, створюючи дану програмну систему. Аналіз конкуруючих систем дав змогу зрозуміти об'єм роботи для створення системи, яка буде забезпечувати усі необхідні потреби користувача. Таким чином, була поставлена задача та сформовані вимоги до програмного продукту.

На наступному етапі було спроектовано архітектуру програмної системи, розроблено схему бази даних та алгоритму процедури надання та затвердження доступу. Обрано засоби програмної реалізації, підготовлено макет дизайну головного екрану програмної системи та описано фізичну модель бази даних.

Програмна система була створена і протестована з використанням технології .NET Core і мови програмування C#. Для зберігання даних була обрана СКБД MSSQL.

Результатом роботи стало успішне виконання поставленого завдання: розроблено програмну систему, яка є масштабованою, гнучкою, стійкою до будь-яких змін і зручною для користувача, та яка відповідає поставленим вимогам.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Date C.J. Database Design and Relational Theory: Normal Forms and All That Jazz. Apress, 2019. 470 P. ISBN 978-148-425-539-1. DOI: <https://doi.org/10.1007/978-1-4842-5540-7>
2. John Sharp Microsoft Visual C# Step by Step 10th Edition / John Sharp – Microsoft Press, 2022. - 832 с.
3. ZKTeco [Електронний ресурс] - Режим доступу до ресурсу: <https://zkteco.pro/> (дата звернення 12.06.2024)
4. Yaware [Електронний ресурс] - Режим доступу до ресурсу: <https://yaware.com/> (дата звернення 12.06.2024)
5. SmartelUA [Електронний ресурс] - Режим доступу до ресурсу: <https://smartel.ua/> (дата звернення 12.06.2024)
6. Wiley Online Library [Електронний ресурс] – Режим доступу до ресурсу: <https://onlinelibrary.wiley.com/> (дата звернення 10.06.2024)
7. Little Free Library [Електронний ресурс] – Режим доступу до ресурсу: <https://littlefreelibrary.org/> (дата звернення 11.06.2024)
8. Open Library [Електронний ресурс] – Режим доступу до ресурсу: <https://openlibrary.org/> (дата звернення 11.06.2024)
9. Murach J. Murach's SQL Server 2019 for Developers / J. Murach, B. Syverson. – Fresno, CA, United States: Mike Murach and Associates, Inc., 2020. – 674 с.
10. Lerman J. Programming Entity Framework: Code First / J. Lerman, R. Miller. – New York: O'Reilly Media, Inc., 2011. – 194 с.
11. Gayathri Mohan: Full Stack Testing. A Practical Guide for Delivering High Quality Software / G. Mohan - O'Reilly Media, 2022. – 406 с.

ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Дата звіту 7/12/2024
Дата редагування ---



Звіт не був оцінений.

метадані

Заголовок

2024_Б_ПІ_ПЗПІ-22-2_Бабій_В_Г_скорочений

Автор

Бабій Володимир Григорович

Науковий керівник / Експерт

Вадим Юрійович Нечволод

підрозділ

Харківський національний університет радіоелектроніки

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		1
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		48

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2



61946

Кількість символів

7885

Кількість слів

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	2022_Б_ПІ_ПЗПІ_18_4_Семко_Д 5/30/2024 Kharkiv National University of Radio Electronics (Kharkiv National University of Radio Electronics)	214	2.71 %
2	2024_Б_ПІ_ПЗПІ_22_2_Душко_А_В_скорочений 7/11/2024 Kharkiv National University of Radio Electronics (Харківський національний університет радіоелектроніки)	147	1.86 %

ДОДАТОК Б

Слайди презентації

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Програмна система обліку доступу та робочого часу
робітників підприємства

Виконав: ст.гр. ПЗПп-22-2 Бабій В.Г.
Керівник: доц. кафедри ПІ Русакова Н.Є.

Рисунок Б.1 – Титульний слайд

АКТУАЛЬНІСТЬ СИСТЕМИ ТА АНАЛОГИ

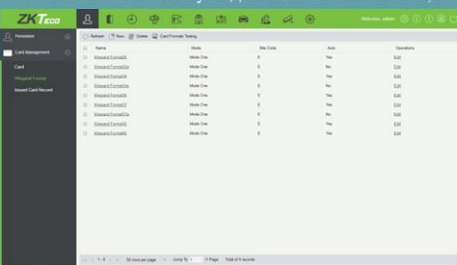
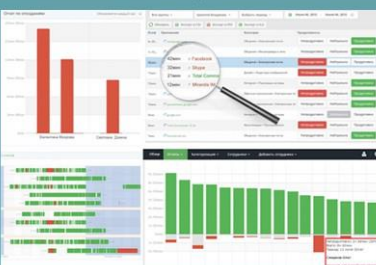

Ефективна система обліку робочого часу спрощує підрахунок оплати праці та підвищує координацію бюджетного процесу

Контроль доступу на об'єкти підприємства забезпечує підвищення рівня безпеки та зниження ризику небажаних інцидентів

ZKTeco ZKBioSecurity від компанії ZKTeco Co., Ltd

Yaware.TimeTracker від компанії Yaware

СКУД від компанії Smartel

2

Рисунок Б.2 – Актуальність системи та аналоги

ПОСТАНОВКА ЗАДАЧІ

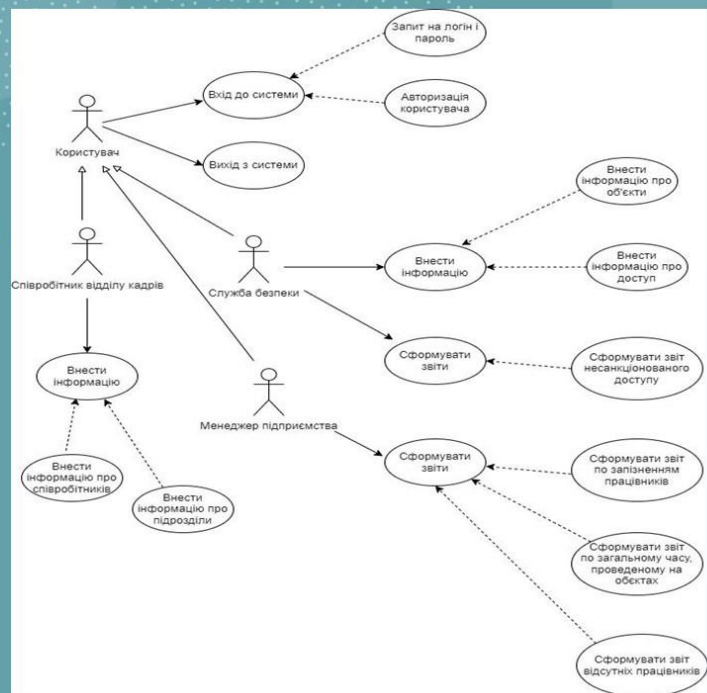
- провести аналіз та моделювання предметної області програмної системи;
- спроектувати базу даних для збереження інформації з предметної області;
- розробити алгоритм процедури надання та затвердження доступу працівникам на об'єкти підприємства;
- спроектувати архітектуру програмної системи;
- виконати програмну реалізацію системи та провести тестування створеного програмного продукту.

3

Рисунок Б.3 – Постановка задачі

АНАЛІЗ ТА МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Загальна USE CASE
діаграма



4

Рисунок Б.4 – Аналіз та моделювання предметної області



Рисунок Б.5 – Проєктування бази даних



Рисунок Б.6 – Алгоритм процедури надання та затвердження доступу



Рисунок Б.7 – Проєктування архітектури

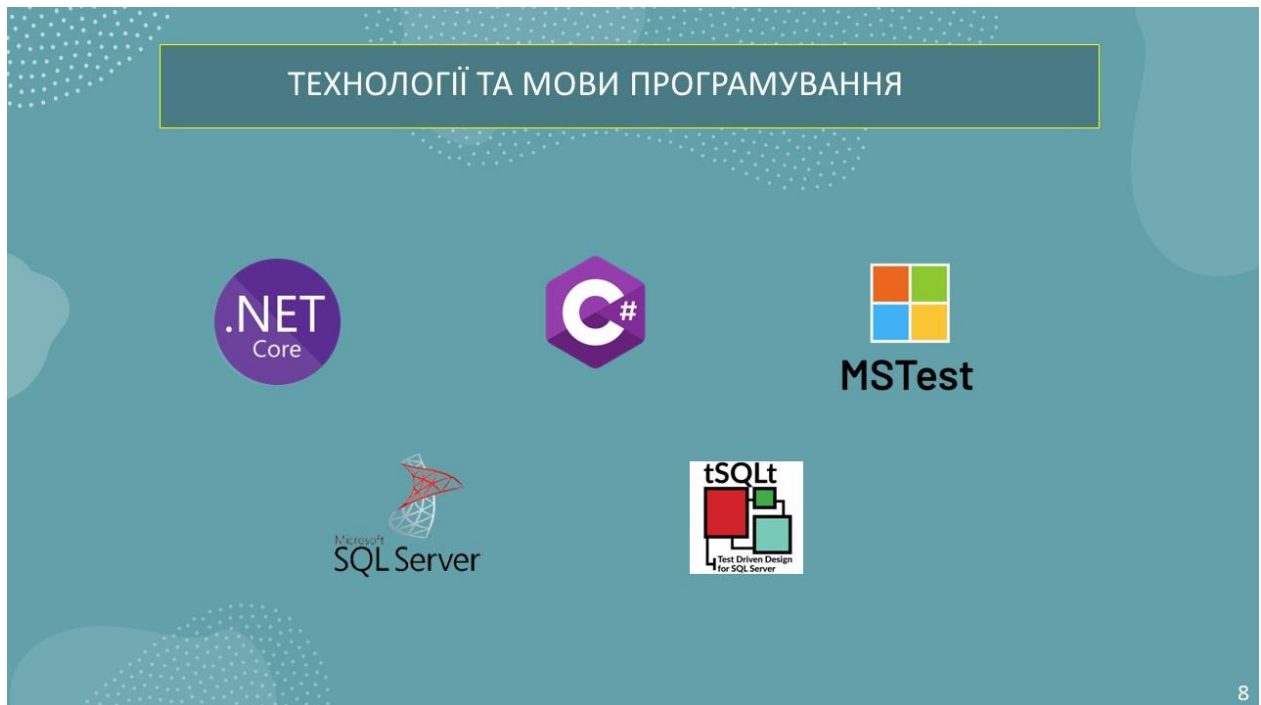


Рисунок Б.8 – Технології та мови програмування

Серверна частина

```

CREATE OR ALTER TRIGGER trg_check_insert_access_status
ON access
AFTER INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE access_status NOT IN ('створений',
        'перевірений')
    )
    BEGIN
        UPDATE access
        SET access_status = 'створений'
        FROM access a
        JOIN inserted i ON a.access_id = i.access_id
        WHERE i.access_status NOT IN ('створений',
        'перевірений');
    END
END;

```

```

CREATE PROCEDURE UpdateAccessStatusToActive
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE access
    SET access_status = 'дійсний'
    WHERE access_status = 'підтверджений'
    AND access_start = CAST(GETDATE() AS DATE);
END;

```

```

CREATE PROCEDURE UpdateAccessStatusToInactive
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE access
    SET access_status = 'недійсний'
    WHERE access_status = 'дійсний'
    AND access_end < CAST(GETDATE() AS DATE);
END;

```

9

Рисунок Б.9 – Серверна частина

ІНТЕРФЕЙС

Форма авторизації

ID	Назва підрозділу	ПІБ	Дата народження	Посада	Номер картки	Дата прийняття	Дата звільнення
1	Виробничий відділ	Портнов Віктор Іванович	20.01.1978	Робітник	12301	01.05.2005	01.01.2100
2	Бухгалтерія	Сидоровська Олена Ва...	02.04.1970	Бухгалтер	12308	20.07.2003	01.01.2100
3	Дирекція	Тилонов Петро Васильович	17.12.1959	Менеджер	12307	10.04.2019	01.01.2100
4	Технічний відділ	Петровський Іван Ро...	20.05.1998	Робітник	12305	15.03.2018	01.01.2100
6	ІТ відділ	Бирлава Юрій Радикович	01.10.1999	ІТ спеціаліст	12306	21.11.2019	01.01.2100
8	Технічний відділ	Митовський Іосиф Ма...	03.06.1994	Робітник	12308	12.03.2023	01.01.2100
9	Контроль якості	Димченко Максим Зоря...	26.01.1989	Робітник	12309	14.11.2009	01.01.2100
10	Відділ фінансового забезпечення	Малавкін Федор Віктор...	14.11.1992	Фінансист першої катег...	12310	21.11.2009	01.01.2100
11	Бухгалтерія	Гасюк Євгенія Азарівна	19.09.2001	Бухгалтер першої катег...	12311	14.11.2022	01.01.2100
12	Бухгалтерія	Голубенський Кузьма П...	13.10.1992	Бухгалтер другої катег...	12312	06.02.2020	01.01.2100
13	Відділ фінансового забезпечення	Мухомин Павло Франко...	17.07.1985	Промисловий фінансист	12313	16.06.2004	01.01.2100
14	Відділ фінансового забезпечення	Пилипенко Анастасія Тар...	20.01.1989	Фінансист	12314	14.11.2023	01.01.2100
15	Виробничий відділ	Павленко Валерій Дмитро...	13.06.1984	Токар	12315	23.01.2015	01.01.2100
16	Виробничий відділ	Захарченко Євген Микола...	30.06.1984	Токар	12316	14.11.2023	01.01.2100
17	Виробничий відділ	Колотко Євгенія Янівна	14.11.2023	Електромонтер	12317	14.06.2017	01.01.2100
18	Охорона	Токарчук Євген Дмитро...	27.11.1959	Охоронець	12318	18.11.2010	01.01.2100

10

Рисунок Б.10 – Інтерфейс

ІНТЕРФЕЙС ДЛЯ МЕНЕДЖЕРА

Застосування для дат: 3 місяці 2024 р. Пошук: ПІБ

Дата	Час	Статус	Вид/Вид	ПІБ	Об'єкт
03.07.2024	07:33:01	успіх	вхід	Діланова Максим Зоранович	Виробниче приміщення №1
03.07.2024	07:33:08	успіх	вхід	Тихонов Петро Васильович	Виробниче приміщення №1
03.07.2024	07:36:48	успіх	вхід	Пашин Олег Юрійович	Головний офіс
03.07.2024	07:37:03	успіх	вхід	Малачук Федір Вікторович	Головний офіс
03.07.2024	07:37:34	успіх	вхід	Міновський Йосиф Максимович	Головний офіс
03.07.2024	07:43:55	успіх	вхід	Тонкачкін Євген Діброварович	Головний офіс
03.07.2024	07:45:16	успіх	вхід	Портнов Віктор Іванович	Головний офіс
03.07.2024	07:50:00	успіх	вхід	Гасюк Едуард Азарович	Головний офіс
03.07.2024	07:57:24	успіх	вхід	Кочинко Євгенія Йосифович	Головний офіс
03.07.2024	08:01:56	успіх	вхід	Кочинко Євгенія Йосифович	Виробниче приміщення №1
03.07.2024	08:02:21	успіх	вхід	Рубанов Артур Фролович	Головний офіс
03.07.2024	08:18:43	успіх	вхід	Маданська Злата Валентинівна	Головний офіс
03.07.2024	08:19:23	успіх	вхід	Голубинський Кузьма Пилипович	Головний офіс
03.07.2024	08:19:26	успіх	вхід	Бурлака Юрій Радикович	Головний офіс
03.07.2024	08:20:25	успіх	вхід	Павленко Валерій Давидович	Головний офіс
03.07.2024	08:23:17	успіх	вхід	Пашинка Анастасія Тарасівна	Головний офіс
03.07.2024	08:24:22	успіх	вхід	Петриковський Іван Романович	Головний офіс
03.07.2024	08:26:11	успіх	вхід	Мікуличев Влад Промислов	Головний офіс
03.07.2024	08:29:32	успіх	вхід	Сидоровська Олена Василівна	Головний офіс
03.07.2024	09:30:15	успіх	вхід	Захарченко Євген Миколайович	Головний офіс

Відвідування об'єктів працівниками

Кількість відпрацьованих годин

Місяць: Листопад Рік: 2024

ПІБ	Пароль	Місяць Рік	Відпрацьовані Години
Портнов Віктор Іванович	Виробничий вхід	07.2024	210
Сидоровська Олена Василівна	Бухгалтерія	07.2024	209
Тихонов Петро Васильович	Дирекція	07.2024	205
Петриковський Іван Романович	Технічний вхід	07.2024	207
Бурлака Юрій Радикович	IT вхід	07.2024	202
Міновський Йосиф Максимович	Технічний вхід	07.2024	209
Діланова Максим Зоранович	Контроль якості	07.2024	209
Малачук Федір Вікторович	Відділ фінансового забезпечення	07.2024	207
Гасюк Едуард Азарович	Бухгалтерія	07.2024	206
Голубинський Кузьма Пилипович	Бухгалтерія	07.2024	209
Мікуличев Влад Промислов	Відділ фінансового забезпечення	07.2024	208
Пашинка Анастасія Тарасівна	Відділ фінансового забезпечення	07.2024	207
Павленко Валерій Давидович	Виробничий вхід	07.2024	208
Захарченко Євген Миколайович	Виробничий вхід	07.2024	208
Кочинко Євгенія Йосифович	Виробничий вхід	07.2024	206

Рисунк Б.11 – Інтерфейс

ІНТЕРФЕЙС ДЛЯ СЛУЖБИ БЕЗПЕКИ

Пошук: ПІБ

Вибірвати доступ в статусі: діловий (3) неділовий (1) опису затвердження (1) створений (2)

ID	Назва	Дата початку	Дата закінчення	ПІБ	Об'єкт	Статус
1	Доступ	12.06.2023	12.06.2024	Сидоровська Олена Василівна	Головний офіс	діловий
2	Доступ 1	15.06.2023	15.06.2024	Портнов Віктор Іванович	Виробниче приміщення №1	діловий
4	Доступ на склад	12.06.2023	01.11.2023	Портнов Віктор Іванович	Склад №1	діловий
5	Доступ на склад 2	14.06.2023	16.06.2024	Бурлака Юрій Радикович	Склад №2	діловий
6	Доступ на виробництво	15.06.2023	15.06.2024	Тихонов Петро Васильович	Виробниче приміщення №1	діловий
7	новий доступ	03.12.2023	03.12.2024	Діланова Максим Зоранович	Головний офіс	діловий
8	автосев	10.12.2023	12.12.2023	Портнов Віктор Іванович	Головний офіс	діловий
9	автосев2	10.12.2023	12.12.2023	Портнов Віктор Іванович	Головний офіс	діловий
11	Доступ 11	05.06.2024	05.06.2025	Міров Олег Петрович	Склад №2	неперевірений
1010	Доступ	01.06.2024	31.12.2026	Тихонов Петро Васильович	Головний офіс	створений
1011	Доступ	01.06.2024	31.12.2026	Петриковський Іван Романович	Головний офіс	створений
1012	Доступ	01.01.2024	31.12.2026	Портнов Віктор Іванович	Головний офіс	діловий
1013	Доступ	01.01.2024	31.12.2026	Сидоровська Олена Василівна	Головний офіс	діловий
1014	Доступ	01.01.2024	31.12.2026	Тихонов Петро Васильович	Головний офіс	діловий
1015	Доступ	01.01.2024	31.12.2026	Петриковський Іван Романович	Головний офіс	діловий
1017	Доступ	01.01.2024	31.12.2026	Бурлака Юрій Радикович	Головний офіс	діловий
1019	Доступ	01.01.2024	31.12.2026	Міновський Йосиф Максимович	Головний офіс	діловий
1020	Доступ	01.01.2024	31.12.2026	Діланова Максим Зоранович	Головний офіс	діловий
1021	Доступ	01.01.2024	31.12.2026	Малачук Федір Вікторович	Головний офіс	діловий

Доступ на об'єкти підприємства

Створення нового доступу

Працівник: Бурлака Юрій Радикович

Об'єкт: Автопарк

Дата початку: 3 місяці 2024 р.

Дата закінчення: 3 місяці 2025 р.

Назва: Доступ

Рисунк Б.12 – Інтерфейс

ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Таблиця 1 – Приклади тест-кейсів

№	Опис	Кроки відтворення	Очікуєий результат	Статус
1	Перевірка авторизації користувача	Запустити програму, ввести правильні логін та пароль, натиснути кнопку «ввійти»	Успішна авторизація, відображається форма зі списком співробітників	Успішно
2	Перевірка додавання нового працівника	Перейти в меню «Редагування» - > «Додати». В формі, яка відобразилась заповнити поля «ПІБ», «дата народження», «посада», «номер картки доступу», «дата прийняття на роботу», «відділ» і натиснути кнопку «ОК»	Новий робітник з'явиться у загальному списку робітників. Дані, які вносили на формі додавання робітника відповідають даним в таблиці робітників	Успішно

13

Рисунок Б.13 – Тестування програмного забезпечення

ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

```

CREATE PROCEDURE AccessTests.[test_trg_check_insert_access_status_ShouldSetCreatedStatus]
AS
BEGIN
    -- Створення тестової таблиці "access"
    EXEC tSQLt.FakeTable @TableName = 'dbo.access';
    -- Вставка тестових даних
    INSERT INTO access (access_name, access_start, access_end, access_status, access_users, employee_id, office_id)
    VALUES
        ('Test Access 1', '2024-01-01', '2024-12-31', 'новий статус', 1, 1, 1), -- Повинен бути змінений на 'створений'
        ('Test Access 2', '2024-01-01', '2024-12-31', 'створений', 1, 1, 1), -- Повинен залишитися без змін
        ('Test Access 3', '2024-01-01', '2024-12-31', 'перевірений', 1, 1, 1); -- Повинен залишитися без змін

    -- Виконання тригера вручну
    EXEC trg_check_insert_access_status;

    -- Перевірка результатів
    DECLARE @Expected TABLE (access_id INT, access_name VARCHAR(100), access_start DATE, access_end DATE, access_status VARCHAR(20),
        access_users INT, employee_id INT, office_id INT);

    INSERT INTO @Expected (access_id, access_name, access_start, access_end, access_status, access_users, employee_id, office_id)
    VALUES
        (1, 'Test Access 1', '2024-01-01', '2024-12-31', 'створений', 1, 1, 1),
        (2, 'Test Access 2', '2024-01-01', '2024-12-31', 'створений', 1, 1, 1),
        (3, 'Test Access 3', '2024-01-01', '2024-12-31', 'перевірений', 1, 1, 1);

    EXEC tSQLt.AssertEqualsTable @Expected, 'access';
END;

```

Приклад програмної реалізації одного з тест-кейсів, який призначений для тестування тригера на серверній частині

14

Рисунок Б.14 – Тестування програмного забезпечення

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було:

- проведено аналіз та моделювання предметної області програмної системи;
- розроблено схему базу даних;
- спроектовано архітектуру програмної системи;
- розроблено алгоритм процедури надання та затвердження доступу працівникам на об'єкти підприємства;
- виконано програмну реалізацію системи та проведено тестування створеного програмного продукту.

15

Рисунок Б.15 – Висновки

ДОДАТОК В

Лістинг програмного коду

Приклад коду В.1 – Фрагмент коду, який реалізує формування звіту по запізненнях:

```
{
    Dictionary<string, int> whoislate = new Dictionary<string,
int>();
    DataTable dt = new DataTable();
    dt.Clear();
    dt.Columns.Add("ПІБ", typeof(string));
    dt.Columns.Add("Підрозділ", typeof(string));
    dt.Columns.Add("Місяць.Рік", typeof(string));
    dt.Columns.Add("Кількість запізнень", typeof(string));

    string selected_month =
((ComboBoxItem)cb_month.SelectedItem).HiddenValue;
    string selected_year =
((ComboBoxItem)cb_year.SelectedItem).HiddenValue;
    string selected_hour =
((ComboBoxItem)cb_hours.SelectedItem).HiddenValue;
    string selected_minutes =
((ComboBoxItem)cb_minutes.SelectedItem).HiddenValue;

    using (SqlConnection sqlConnection = new
SqlConnection(@Program.connection_string))
    {
        string sqlQuery = "select min(visit_time) as comintime,
employee_id, DAY(visit_date) as day1 from visit where visit_status='успіх'
and visit_inout='вхід' and MONTH(visit_date)='"+ selected_month + "' and
YEAR(visit_date)='"+ selected_year + "' group by employee_id,
DAY(visit_date)";

        SqlCommand sqlCmd = new SqlCommand(sqlQuery, sqlConnection);
        sqlConnection.Open();
        SqlDataReader sqlReader = sqlCmd.ExecuteReader();

        while (sqlReader.Read())
        {
            var comintime = sqlReader["comintime"].ToString();
            string[] hhmm = comintime.Split(':');
            string hh = hhmm[0];
            string mm = hhmm[1];
            int usercomintime = Convert.ToInt32(hh) * 60 +
Convert.ToInt32(mm);
            int stardaytime = Convert.ToInt32(selected_hour) * 60 +
Convert.ToInt32(selected_minutes);
            if (usercomintime > stardaytime)
            {
```

```

        if
(whoislate.ContainsKey(sqlReader["employee_id"].ToString()))
        {
            whoislate[sqlReader["employee_id"].ToString()]++;
        }
        else
        {

whoislate.Add(sqlReader["employee_id"].ToString(), 1);
        }
    }
}

//відсортований словник запізнь
var sortedDict = whoislate.OrderByDescending(x =>
x.Value).ToDictionary(x => x.Key, x => x.Value);

int cnt = 1;
string sql_in = "";
int get_count;
int selected_top =
Convert.ToInt32(((ComboBoxItem)cb_top.SelectedItem).HiddenValue);

if (sortedDict.Count() > selected_top)
{
    get_count = selected_top;
}
else
{
    get_count = sortedDict.Count();
}
foreach (var emp in sortedDict)
{
    if (cnt < get_count)
    {
        sql_in = sql_in + emp.Key+", ";
        cnt++;
    }
    else if (cnt == get_count)
    {
        sql_in = sql_in + emp.Key;
        cnt++;
    }
    else
    {
        break;
    }
}

if (sql_in.Length>0)
{

```

```

        using (SqlConnection sqlConnection = new
SqlConnection(@Program.connection_string))
        {
            string sqlQuery = "select e.employee_id, e.pib,
d.dep_name from employee e left join department d on
e.department_id=d.department_id where e.employee_id in (" + sql_in + ")";
            SqlCommand sqlCmd = new SqlCommand(sqlQuery,
sqlConnection);

            sqlConnection.Open();
            SqlDataReader sqlReader = sqlCmd.ExecuteReader();
            while (sqlReader.Read())
            {
                dt.Rows.Add(new object[] {
sqlReader["pib"].ToString(), sqlReader["dep_name"].ToString(), selected_month
+ "." + selected_year, sortedDict[sqlReader["employee_id"].ToString()] });
            }
        }

        dataGridView1.DataSource = dt;
        dataGridView1.Columns[0].Width = 250;
        dataGridView1.Columns[1].Width = 250;
    }
}

```

Приклад коду В.2 – Фрагмент коду, який реалізує формування звіту по відпрацьованих годинах:

```

{
    List<List<string>> myList = new List<List<string>>();
    DataTable dt = new DataTable();
    dt.Clear();
    dt.Columns.Add("ПІБ", typeof(string));
    dt.Columns.Add("Підрозділ", typeof(string));
    dt.Columns.Add("Місяць.Рік", typeof(string));
    dt.Columns.Add("Відпрацьовані Години", typeof(string));

    //додаємо працівників
    using (SqlConnection sqlConnection = new
SqlConnection(@Program.connection_string))
    {
        string sqlQuery = "select e.employee_id, e.PIB, d.dep_name
from employee e left join department d on e.department_id=d.department_id;";
        SqlCommand sqlCmd = new SqlCommand(sqlQuery, sqlConnection);
        sqlConnection.Open();
        SqlDataReader sqlReader = sqlCmd.ExecuteReader();

        while (sqlReader.Read())
        {
            var employee_id = sqlReader["employee_id"].ToString();
            var pib = sqlReader["PIB"].ToString();
            var dep_name = sqlReader["dep_name"].ToString();

```

```

        myList.Add(new List<string>() { employee_id, pib,
dep_name, "0", "", "" });
    }
}

//отримуємо місяць і рік, і розраховуємо дані
string selected_month =
((ComboBoxItem)cb_month.SelectedItem).HiddenValue;
string selected_year =
((ComboBoxItem)cb_year.SelectedItem).HiddenValue;
var lastdayofmonth =
DateTime.DaysInMonth(Convert.ToInt32(selected_year),
Convert.ToInt32(selected_month));
for (int i = 1; i <= Convert.ToInt32(lastdayofmonth); i++)
{
    string day;
    if (i < 10)
    {
        day = "0" + i.ToString();
    }
    else
    {
        day = i.ToString();
    }
    string visitdate = selected_year + "-" + selected_month + "-"
+ day;
    using (SqlConnection sqlConnection = new
SqlConnection(@Program.connection_string))
    {
        string sqlQuery = "select * from (select min(visit_time)
as comintime, employee_id from visit where visit_date='" + visitdate + "' and
visit_status='ycnix' and visit_inout='вхід' group by employee_id) a left join
(select max(visit_time) as comouttime, employee_id from visit where
visit_date='" + visitdate + "' and visit_status='ycnix' and
visit_inout='вихід' group by employee_id) b on a.employee_id=b.employee_id;";
        SqlCommand sqlCmd = new SqlCommand(sqlQuery,
sqlConnection);

        sqlConnection.Open();
        SqlDataReader sqlReader = sqlCmd.ExecuteReader();

        while (sqlReader.Read())
        {
            var comintime = sqlReader["comintime"].ToString();
            var employee_id =
sqlReader["employee_id"].ToString();
            var comouttime = sqlReader["comouttime"].ToString();
            foreach (var myitem in myList)
            {
                if (myitem[0] == employee_id)
                {
                    int workinghourdayinsec =
convertTimeToSec(comouttime) - convertTimeToSec(comintime);

```

```

        myitem[3] =
Convert.ToString(Convert.ToInt32(myitem[3]) + workinghourdayinsec);
    }
    }
    sqlReader.Close();
    sqlConnection.Close();
}
}
//end for days

foreach (var emp in myList)
{
    int wh = Convert.ToInt32(emp[3]) / 3600;
    dt.Rows.Add(new object[] { emp[1], emp[2], selected_month +
"." + selected_year, wh.ToString() });
}

dataGridView1.DataSource = dt;
dataGridView1.Columns[0].Width = 260;
dataGridView1.Columns[1].Width = 260;
dataGridView1.Columns[2].Width = 100;
dataGridView1.Columns[3].Width = 110;
dataGridView1.Refresh();
button2.Enabled = true;
}

```

Приклад коду В.3 – Фрагмент коду, який реалізує експорт datagrid в Excel:

```

{
    button2.Enabled = false;
    // creating Excel Application
    Microsoft.Office.Interop.Excel._Application app = new
Microsoft.Office.Interop.Excel.Application();
    // creating new Workbook within Excel application
    Microsoft.Office.Interop.Excel._Workbook workbook =
app.Workbooks.Add(Type.Missing);
    // creating new Excelsheet in workbook
    Microsoft.Office.Interop.Excel._Worksheet worksheet = null;
    // see the excel sheet behind the program
    app.Visible = true;
    // get the reference of first sheet. By default its name is
Sheet1.
    // store its reference to worksheet
    worksheet = workbook.Sheets[1];
    worksheet = workbook.ActiveSheet;
    // changing the name of active sheet
    worksheet.Name = "Exported from gridview";
    // storing header part in Excel
    for (int i = 1; i < dataGridView1.Columns.Count + 1; i++)
    {

```

```
        worksheet.Cells[1, i] = dataGridView1.Columns[i -
1].HeaderText;
    }
    // storing Each row and column value to excel sheet
    for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)
    {
        for (int j = 0; j < dataGridView1.Columns.Count; j++)
        {
            worksheet.Cells[i + 2, j + 1] =
dataGridView1.Rows[i].Cells[j].Value.ToString();
        }
    }
    button2.Enabled = true;
}
```