

**Міністерство освіти і науки України**  
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет комп'ютерно-інформаційних систем та програмної інженерії  
(повна назва факультету)

Кафедра програмної інженерії  
(повна назва кафедри)

## КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

На тему: Система управління проєктами комунального підприємства "Благоустрій" з використанням технології .NET

Виконав: студент 4 курсу, групи СПс-41  
спеціальності 121 «Інженерія програмного забезпечення»

(шифр і назва напрямку підготовки, спеціальності)

|                   |       |                                                |
|-------------------|-------|------------------------------------------------|
|                   | <hr/> | <u>Пласконь Н.П.</u><br>(прізвище та ініціали) |
| Керівник          | <hr/> | <u>Михалик Д.М.</u><br>(прізвище та ініціали)  |
| Нормоконтроль     | <hr/> | <hr/>                                          |
| Завідувач кафедри | <hr/> | <hr/>                                          |
| Рецензент         | <hr/> | <hr/>                                          |

Тернопіль, 2024

## АНОТАЦІЯ

Кваліфікаційна робота на тему «Система управління проєктами комунального підприємства «Благоустрій» з використанням технології .NET» виконана Пласконем Назаром Петровичем, студентом групи СПс-41, кафедри програмної інженерії, факультету комп'ютерно-інформаційних систем і програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя.

Відомості про обсяг: сторінок – 62, рисунків – 23, таблиць – 7, додатків – 5, посилань – 18.

Об'єкт дослідження – інформативна база комунального підприємства.

Мета роботи – розробка бази даних та реалізація програмного забезпечення для ефективного ведення обліку проєктів, що здійснюються на комунальному підприємстві «Благоустрій».

Метод дослідження – опис об'єкта дослідження через аналіз і обробку відповідних даних, враховуючи задану мету; розробка структури бази даних з використанням методів нормалізації для оптимізації її функціональності; написання програмного забезпечення та відлагодження його в середовищі Visual Studio.

Отримані результати – створено програмний засіб для ведення обліку проєктів на комунальному підприємстві «Благоустрій», який інтегрується з базою даних.

Ключові слова: C#, База даних, Атрибут, Сутність, Зв'язок, SQL Server, Підприємство, Облік, ER-діаграма, Проєкти, Фінанси, Ресурси, Персонал, Замовники.

## ABSTRACT

Qualification work on the topic "Project Management System of the Municipal Enterprise 'Blahoustrii' using .NET Technology" completed by Nazar Plaskon, a student of group SPs-41, Department of Software Engineering, Faculty of Computer Information Systems and Software Engineering, Ternopil Ivan Pului National Technical University.

Information about the volume: pages – 62, figures – 23, tables – 7, appendices – 5, references – 18.

Object of research – the information base of the municipal enterprise.

The aim of the work – development of a database and implementation of software for effective project accounting conducted by the municipal enterprise "Beautification."

Research Method – description of the research object through analysis and processing of relevant data, considering the set goal; development of the database structure using normalization methods to optimize its functionality; writing software and debugging it in the Visual Studio environment.

Results Obtained – a software tool for project accounting in the municipal enterprise "Beautification" has been created, which integrates with the database.

Keywords: C#, Database, Attribute, Entity, Relationship, SQL Server, Enterprise, Accounting, ER Diagram, Projects, Finance, Resources, Personnel, Clients.

## ЗМІСТ

|                                                            |    |
|------------------------------------------------------------|----|
| ВСТУП.....                                                 | 7  |
| 1 АНАЛІТИЧНА ЧАСТИНА .....                                 | 9  |
| 1.1 Опис об'єкту автоматизації .....                       | 9  |
| 1.2 Аналіз схожих проєктних рішень .....                   | 10 |
| 2 ПРОЄКТНА ЧАСТИНА .....                                   | 15 |
| 2.1 Формулювання задачі.....                               | 15 |
| 2.2 Проєктування бази даних.....                           | 16 |
| 2.3 Проєктування програмного продукту .....                | 21 |
| 3 ПРАКТИЧНА ЧАСТИНА .....                                  | 24 |
| 3.1 Проєктування користувацького інтерфейсу .....          | 24 |
| 3.2 Опис модулів програмного забезпечення .....            | 30 |
| 3.3 Результати тестування програми .....                   | 32 |
| 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .....      | 39 |
| 4.1 Актуальність безпеки життєдіяльності людини .....      | 39 |
| 4.2 Проведення інструктажів з охорони праці.....           | 40 |
| ВИСНОВКИ.....                                              | 43 |
| ПЕРЕЛІК ПОСИЛАНЬ .....                                     | 44 |
| ДОДАТКИ.....                                               | 46 |
| ДОДАТОК А Діаграма «сутність-зв'язок».....                 | 47 |
| ДОДАТОК Б Діаграма зв'язків між таблицями бази даних ..... | 48 |
| ДОДАТОК В Код форми головного вікна програми .....         | 49 |
| ДОДАТОК Д Код вкладки «Проекти».....                       | 55 |
| ДОДАТОК Е Диск із кваліфікаційною роботою бакалавра .....  | 62 |

## ВСТУП

Сучасні комунальні підприємства, як і будь-які інші організації, розуміють необхідність постійного розвитку та удосконалення своїх діяльностей для забезпечення комфорту і задоволення потреб місцевого населення. Один із ключових аспектів цього процесу - це автоматизація та впровадження інформаційних систем. Такий підхід дозволяє підприємствам не тільки підвищувати ефективність своєї роботи, але й забезпечувати більш високий рівень обслуговування для своїх клієнтів, що в кінцевому підсумку сприяє підвищенню якості життя мешканців громади.

Сьогодні автоматизація є не лише важливою, але й перспективною стратегією для підприємств сфери благоустрою. Вона дає можливість звільнити співробітників від рутинних операцій, таких як обробка документів, планування робіт або моніторинг стану обладнання. Це, в свою чергу, дозволяє працівникам зосередитися на більш важливих і творчих завданнях, які вимагають їхнього професійного досвіду та знань. Відмінною рисою цього підходу є можливість швидко та ефективно керувати процесами на підприємстві "Благоустрій". Завдяки автоматизації всі процеси стають прозорішими та контрольованішими, що сприяє покращенню якості надання послуг, оптимізації витрат, і, в кінцевому підсумку, задоволенню потреб місцевого населення. Наприклад, автоматизовані системи можуть значно скоротити час на обробку заявок від громадян, швидко реагувати на проблеми та забезпечувати своєчасне виконання робіт з благоустрою.

Однак, щоб досягти максимальних результатів, необхідно розробити та впровадити інформаційну систему, яка дозволить спрощувати та систематизувати всі аспекти роботи комунального підприємства "Благоустрій". Ця система має об'єднувати усі сфери діяльності підприємства, включаючи управління ресурсами, фінансами, а також планування та моніторинг робіт. Використання такої програми забезпечить високу ефективність та сприятиме покращенню життя місцевого населення. Наприклад, система може включати модулі для планування робіт,

управління транспортом, контроль якості виконаних робіт та зворотний зв'язок з мешканцями. Завдяки цьому всі процеси стануть більш організованими, а підприємство зможе швидко адаптуватися до змін та потреб місцевої громади.

Для досягнення максимальної результативності, важливо, щоб усі співробітники комунального підприємства "Благоустрій" взаємодіяли з цією інформаційною системою. Менеджери, бухгалтери, адміністративні працівники різних рангів повинні бути активними користувачами цієї програми. Це включає навчання працівників для роботи з новими інструментами, а також регулярне оновлення системи для забезпечення її актуальності та ефективності. Для забезпечення безпеки та обмеження доступу до конфіденційних даних в системі можливо використовувати відповідну правову політику облікових записів. Важливо запровадити різні рівні доступу для користувачів, щоб забезпечити конфіденційність та захист даних. Це включає використання паролів, двофакторної автентифікації та інших методів захисту інформації.

Отже, впровадження інформаційної системи стає ключовим етапом у розвитку комунального підприємства "Благоустрій" і сприятиме покращенню якості надання комунальних послуг та задоволенню потреб громади. Цей процес потребує комплексного підходу, включаючи підготовку персоналу, технічне забезпечення та постійне вдосконалення системи. В результаті, автоматизація і впровадження інформаційних систем допоможуть підприємству стати більш ефективним, прозорим і орієнтованим на потреби мешканців, що в кінцевому підсумку сприятиме сталому розвитку громади та підвищенню рівня життя її мешканців.

## 1 АНАЛІТИЧНА ЧАСТИНА

### 1.1 Опис об'єкту автоматизації

Комунальне підприємство «Благоустрій» спеціалізується на різноманітних послугах, пов'язаних із благоустроєм міських та прилеглих територій, а також наданні комунальних послуг. Це підприємство відіграє ключову роль у сфері покращення міського середовища та надання комунальних послуг. Воно забезпечує важливі функції у великому місті, сприяючи підвищенню якості життя жителів та охороні природного середовища. Для підвищення ефективності роботи та оптимізації процесів, комунальне підприємство "Благоустрій" планує впровадити автоматизовану систему.

Підприємство надає широкий спектр послуг, орієнтованих на створення комфортного та екологічно чистого середовища для громади та підтримання міської інфраструктури. Діяльність підприємства охоплює [2]:

1. Архітектурне проєктування: Розробка планів для зон відпочинку, парків, скверів, а також вуличних зелених насаджень, що відповідають високим стандартам естетики та функціональності.

2. Комплексне обслуговування об'єктів: Забезпечення належного стану та догляду за парками, скверами, зонами відпочинку, зеленими насадженнями, а також загально доступними приміщеннями.

3. Відтворення рослинності: Посадка нових рослин, догляд за ними та вирощування дерев і чагарників для подальшого використання.

4. Збирання та утилізація відходів: Організація санітарного прибирання вулиць міста, збір та утилізація рослинних відходів та обрізок.

5. Озеленення та впорядкування територій: Роботи з озеленення та поліпшення зовнішнього вигляду міських та прибудинкових територій.

6. Надання комунальних та ремонтно-побутових послуг: Забезпечення комунальних послуг, включаючи ремонт та обслуговування зон відпочинку.

7. Ремонт та реконструкція зелених зон: Виконання капітального та поточного ремонту зелених зон міста.

8. Проектування місць відпочинку та озеленення території: Розробка проєктів та кошторисної документації для покращення міської інфраструктури.

У сучасних умовах кожне підприємство прагне автоматизувати свої виробничі процеси, що дозволяє знизити витрати та полегшити роботу працівників. Автоматизація є важливим та перспективним напрямком у розвитку будь-якої організації. Зменшення участі людини у виробничих процесах та концентрація на основних операціях сприяє покращенню умов праці та економічних показників.

Наразі підприємство «Благоустрій» не має автоматизованої системи обліку. Тому основною метою є розробка програми для обліку проєктів, що дозволить спростити та систематизувати робочі процеси, зробивши їх більш прозорими та ефективними, що, у свою чергу, сприятиме загальному покращенню роботи підприємства.

## 1.2 Аналіз схожих проєктних рішень

На ринку існує чимало програм для обліку, проте кожна з них має свої особливості і не завжди підходить для конкретного підприємства. Розглянемо декілька програм, які могли б відповідати потребам комунального підприємства «Благоустрій».

Однією з найбільш розповсюджених облікових програм є «1С Підприємство». Цей програмний інструмент призначений для автоматизації облікових процесів у різноманітних підприємствах з економічним спрямуванням (див. Рис. 1.1). Завдяки своїм численним перевагам і широкій універсальності, програма швидко здобула популярність серед користувачів. Вона забезпечує ефективний облік різних аспектів діяльності підприємства, пропонуючи високу точність і зручність у використанні.



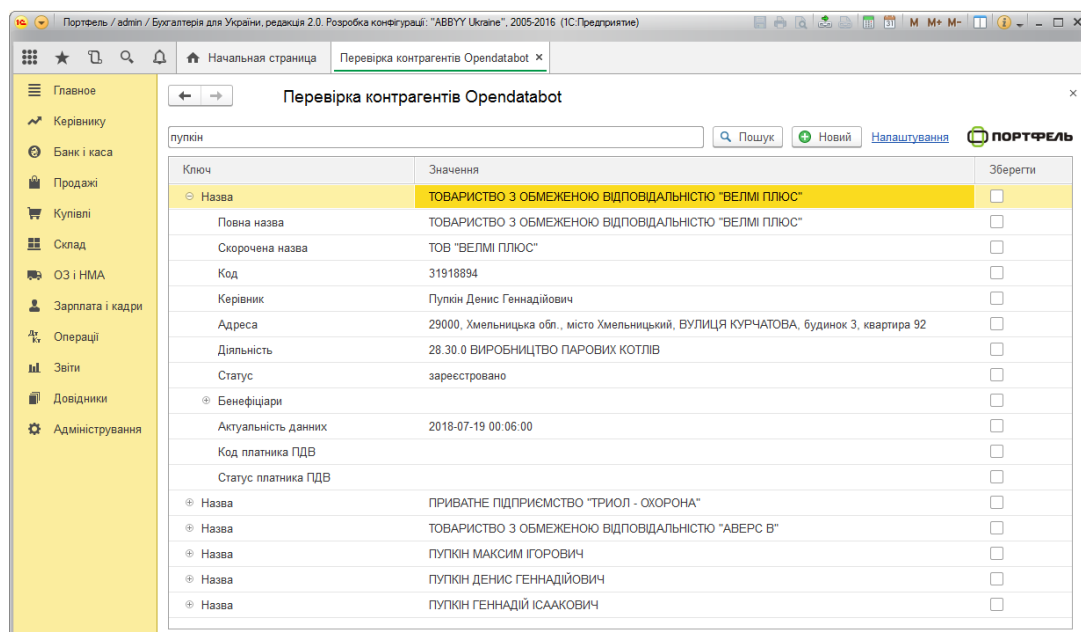


Рисунок 1.1 – Вигляд програми «1С Підприємство»

Розглянемо детальніше переваги використання програми «1С Підприємство» [4]:

- Велика кількість користувачів. Система дозволяє працювати багатьом співробітникам одночасно, що робить її масштабованою і придатною для різних розмірів підприємств.

- Оперативна і всебічна підтримка. Після придбання програми користувачі отримують документи з інформацією про технічну підтримку, доступ до консультацій фахівців та методичних посібників. Це допомагає швидко вирішувати будь-які питання, що виникають під час експлуатації.

- Індивідуальне налаштування. Програмне забезпечення можна адаптувати під специфічні потреби кожного підприємства, розробляючи та налаштовуючи індивідуальні проекти.

- Об'єктно-орієнтована мова програмування. Інструменти, розроблені фахівцями «1С Підприємство», полегшують роботу з конфігураціями та сприяють більш ефективному використанню програми.

- Гнучкість у модифікаціях. Завдяки відкритому коду, програмісти можуть вносити зміни та розширювати функціонал продукту відповідно до конкретних вимог підприємства.

- Уніфікований технологічний інструмент. Програма забезпечує високий рівень стандартизації у розробці, дозволяє масштабувати проєкти та швидко впроваджувати нові технології.

Проте існують і недоліки у використанні програми «1С Підприємство» [4]:

- Обмежений функціонал. Іноді система не може задовольнити всі вимоги підприємства, тому доводиться використовувати додаткові рішення для повного покриття потреб.

- Відсутність спеціалізованих галузевих рішень. Хоча програма є універсальною, для деяких компаній доводиться витрачати значні ресурси на розробку спеціалізованих документів, довідників та звітів.

- Велика кількість постачальників послуг. Це може ускладнити вибір надійного партнера для забезпечення якісної та оперативної підтримки.

- Висока вартість впровадження. Послуги програмістів можуть бути дорогими, але індивідуальний підхід та досягнуті результати часто виправдовують ці витрати.

- Помилки в оновленнях. Оновлення програми іноді можуть містити помилки, що негативно впливає на ефективність її роботи.

Аналогічною програмою є «Microsoft Dynamics NAV», яка є системою управління ресурсами підприємства (ERP). Ця система була спочатку розроблена датською компанією Navision і зараз представлена як один із продуктів у лінійці Microsoft Dynamics (див. Рис. 1.2) [3].

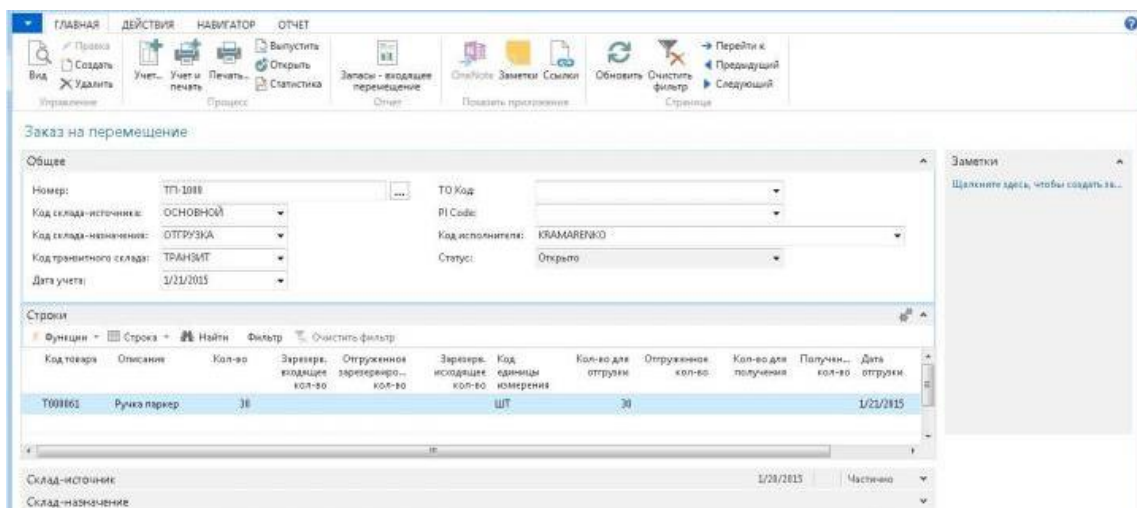


Рисунок 1.2 – Вид программы «Microsoft Dynamics NAV»

Цей продукт володіє багатьма перевагами, серед яких можна виділити такі [5]:

- Широкий спектр функціональності. Програма підходить як для великих, так і для середніх підприємств, завдяки своїй здатності задовольняти різноманітні потреби користувачів.
- Гнучкість налаштувань. Система легко адаптується під специфічні вимоги вашої компанії, дозволяючи швидко та ефективно налаштовувати її відповідно до потреб бізнесу.
- Високий рівень захисту. Всі інформаційні дані та проведені операції забезпечуються надійним захистом, що мінімізує ризики втрати або несанкціонованого доступу.
- Зручне відстеження змін. Програма дозволяє легко моніторити всі зміни у документах, що забезпечує прозорість та контроль над бізнес-процесами.
- Мультизадачність. Система підтримує виконання багатьох завдань одночасно, що підвищує ефективність роботи і економить час.

Однак, програма має і певні недоліки [5]:

- Відставання в деяких функціях. В порівнянні з іншими програмами, цей продукт все ще має деякі обмеження у функціоналі, що може впливати на його конкурентоспроможність.

- Складність у використанні. Для деяких користувачів програма може здатися більш складною у використанні, що потребує додаткового навчання та адаптації.

- Залежність від інших продуктів Microsoft. Програма розрахована на інтеграцію з іншими продуктами Microsoft, що може обмежувати її самостійну функціональність і вимагати додаткових витрат на придбання сумісних продуктів.

- Висока вартість. Програмне забезпечення має значну ціну, що може бути недоступним для деяких підприємств, особливо для малих бізнесів.

Головним недоліком таких програм є те, що вони не завжди орієнтовані на одну конкретну галузь. Це може призводити до недостатньої функціональності для специфічних потреб окремих підприємств.

Для комунального підприємства «Благоустрій» планується розробити програмне забезпечення, яке буде спеціально адаптоване під його потреби. Це дозволить включити лише необхідний функціонал, уникнувши зайвих елементів, які не будуть використовуватися. Таке рішення також сприятиме спрощенню користувацького інтерфейсу та зменшенню розміру програми, що підвищить її зручність та ефективність у використанні.

## 2 ПРОЄКТНА ЧАСТИНА

### 2.1 Формулювання задачі

Метою даної роботи є створення автоматизованої системи обліку проєктів для комунального підприємства «Благоустрій». Ця система повинна забезпечувати облік і розрахунок основних подій підприємства, таких як наявність необхідних ресурсів, чистий дохід та окупність проєктів, а також надавати можливість експортування необхідної інформації для подальшого використання.

Ефективне застосування сучасних комп'ютерних технологій дозволить значно підвищити продуктивність роботи підприємства, що в результаті призведе до зростання його прибутків. На відміну від інших програмних забезпечень, дана система буде здатна обчислювати окупність проєктів, аналізувати витрати та своєчасно сповіщати про майбутні події, що є надзвичайно важливим для оперативного управління підприємством.

Заплановано створити інформаційну систему засобами Visual C#, яка забезпечуватиме облік проєктів на підприємстві «Благоустрій». Для цього необхідно реалізувати взаємодію з базою даних та модифікацію записів за допомогою SQL-запитів [6].

Для розробки даного програмного проєкту потрібно виконати наступні завдання:

1. Передбачити вікно авторизації для захисту інформації від несанкціонованого доступу, що забезпечить безпеку даних.
2. Додати можливість перегляду, додавання, видалення та редагування інформації у таблицях, що дозволить користувачам зручно управляти даними.
3. Реалізувати обчислення необхідної кількості ресурсів для виконання проєктів, що допоможе ефективно планувати використання ресурсів.
4. Забезпечити обчислення окупності проєктів та чистого доходу, що дозволить оперативно оцінювати фінансовий стан підприємства.

5. Передбачити можливість пошуку в таблицях, що дозволить швидко знаходити необхідну інформацію.

6. Реалізувати експорт таблиць у формати Excel та Word, що спростить підготовку звітів та інших документів.

7. Спроекувати вікно з довідкою про програму, щоб користувачі могли легко знайти необхідну інформацію про використання системи.

8. Розробити простий та зрозумілий інтерфейс, що забезпечить зручність використання програми.

Ці завдання спрямовані на створення ефективної і зручної у використанні системи, яка дозволить підприємству «Благоустрій» оптимізувати свої робочі процеси та досягти нових висот у своїй діяльності.

## 2.2 Проєктування бази даних

Розглянемо основні операції, які виконує база даних:

- Введення інформації про проєкти, ресурси, фінанси, персонал та замовників.
- Пошук доступних ресурсів для виконання проєктів.
- Розрахунок чистого доходу з фінансових операцій та окупності проєктів.
- Перевірка наявності необхідних ресурсів для реалізації проєктів.
- Виведення списку замовників за встановленим критерієм пошуку.
- Надання інформації про потрібні ресурси.

Система розрахована на різних типів користувачів, включаючи адміністратора, менеджера та бухгалтера. Адміністратор має повний доступ до всієї інформації, можливість додавання, редагування та видалення записів.

Менеджер отримує доступ до даних про проекти, ресурси, фінанси та замовників, а бухгалтер - до фінансової інформації та персоналу.

Тепер розглянемо основні сутності бази даних, їх назви та особливості використання.

Таблиця 2.1 – Сутності у базі даних та їх опис

| Назва сутності | Опис сутності                                         | Псевдоніми | Особливості застосування                               |
|----------------|-------------------------------------------------------|------------|--------------------------------------------------------|
| 1              | 2                                                     | 3          | 4                                                      |
| Користувачі    | Перелік користувачів, які мають доступ до програми    | uzer       | Обов'язково вказується логін та пароль                 |
| Проекти        | Дані про проекти підприємства                         | projects   | Кожен проект містить крайній термін виконання завдання |
| Фінанси        | Зберігаються дані про фінансові операції підприємства | finance    | Кожна операція поділяється за типом: витрата або дохід |
| Персонал       | Дані про працівників підприємства                     | staff      | Вказується відпустка, коли працівник буде відсутній    |
| Замовники      | Дані про замовника проекту                            | clients    | Вказується вартість виконання проекту замовником       |
| Ресурси        | Перелік ресурсів для виконання проектів               | resource   | Вказується проект у якому використовуються ресурси     |

Розглянемо детальніше сутності, атрибути, їхні домени та властивості бази даних «Blagoustroi»

Таблиця 2.2 – Характеристики атрибутів, сутностей, їхні домени та властивості бази даних «Blagoustroi»

| Вид сутності | Атрибут | Опис сутності               | Тип даних | Обмеження      | Попереднє значення | Псевдонім | Порожнє значення |
|--------------|---------|-----------------------------|-----------|----------------|--------------------|-----------|------------------|
| 1            | 2       | 3                           | 4         | 5              | 6                  | 7         | 8                |
| Користувачі  | id      | Код користувача             | Числовий  | —              | —                  | —         | Ні               |
|              | login   | Логін користувача для входу | Текстовий | Первинний ключ | —                  | —         | Ні               |

Продовження таблиці 2.2

| 1        | 2                  | 3                                      | 4         | 5              | 6 | 7 | 8   |
|----------|--------------------|----------------------------------------|-----------|----------------|---|---|-----|
|          | password           | Пароль користувача                     | Текстовий | —              | — | — | Ні  |
|          | username           | Ім'я користувача                       | Текстовий | —              | — | — | Так |
|          | tables_access      | Доступ до таблиць                      | Текстовий | —              | — | — | Так |
|          | image_url          | Зображення користувача                 | Текстовий | —              | — | — | Так |
| Персонал | id                 | Унікальний код працівника              | Числовий  | Первинний ключ | — | — | Ні  |
|          | ПІБ                | Прізвище, ім'я, по батькові працівника | Текстовий | —              | — | — | Так |
|          | Посада             | Посада працівника                      | Текстовий | —              | — | — | Так |
|          | Графік             | Графік роботи працівника               | Текстовий | —              | — | — | Так |
|          | Відпустка          | Дата відпустки працівника              | Дата      | —              | — | — | Так |
|          | Контакт            | Контакт для зв'язку з працівником      | Текстовий | —              | — | — | Так |
| Ресурси  | id                 | Унікальний код ресурсу                 | Числовий  | Первинний ключ | — | — | Ні  |
|          | Назва              | Назва ресурсу                          | Текстовий | —              | — | — | Так |
|          | Кількість          | Необхідна к-сть ресурсу                | Числовий  | —              | — | — | Так |
|          | Провайдер          | Провайдер ресурсу                      | Текстовий | —              | — | — | Так |
|          | Ціна               | Ціна за одиницю ресурсу                | Числовий  | —              | — | — | Так |
|          | Код проєкту        | Проект у якому використовується ресурс | Числовий  | Зовнішній ключ | — | — | Ні  |
|          | Доступна кількість | Кількість наявного ресурсу             | Числовий  | —              | — | — | Так |



Продовження таблиці 2.2

| 1       | 2              | 3                                   | 4         | 5              | 6 | 7 | 8   |
|---------|----------------|-------------------------------------|-----------|----------------|---|---|-----|
| Проекти | Id             | Унікальний код проекту              | Числовий  | Первинний ключ | — | — | Ні  |
|         | Назва          | Назва проекту                       | Текстовий | —              | — | — | Так |
|         | Опис           | Опис проекту                        | Текстовий | —              | — | — | Так |
|         | Дедлайн        | Крайній термін здачі проекту        | Дата      | —              | — | — | Так |
|         | Відповідальний | Відповідальний за проєкт            | Текстовий | —              | — | — | Так |
|         | Статус         | Статус проекту                      | Текстовий | —              | — | — | Так |
| Фінанси | id             | Унікальний код операції             | Числовий  | Первинний ключ | — | — | Ні  |
|         | Тип            | Вид фінансової операції             | Текстовий | —              | — | — | Так |
|         | Опис           | Опис фінансової операції            | Текстовий | —              | — | — | Так |
|         | Сума           | Сума операції                       | Числовий  | —              | — | — | Так |
|         | Дата           | Дата проведення операції            | Дата      | —              | — | — | Так |
|         | Категорія      | Категорія операції                  | Текстовий | —              | — | — | Так |
| Клієнти | id             | Унікальний код клієнта              | Числовий  | Первинний ключ | — | — | Ні  |
|         | ПІБ            | Прізвище, ім'я, по батькові клієнта | Текстовий | —              | — | — | Так |
|         | Контакт        | Контакт для зв'язку з клієнтом      | Текстовий | —              | — | — | Так |
|         | Код проекту    | Проект клієнта                      | Числовий  | Зовнішній ключ | — | — | Ні  |
|         | Вартість       | Вартість проекту                    | Числовий  | —              | — | — | Так |

При створенні сутностей важливо визначити унікальні атрибути, які будуть використовуватися для їх ідентифікації. Ці атрибути є первинними ключами.

Первинні ключі обрані наступним чином:

- Користувачі → Логін;
- Клієнти → Код;
- Фінанси → Код;
- Проєкти → Код;
- Працівники → Код;
- Ресурси → Код.

Взаємозв'язки між цими сутностями представлені у таблиці 2.3. А у додатку А зображена ER-діаграма.

Таблиця 2.3 – Опис взаємозв'язків між сутностями у базі даних «Blagoustroi»

| Назва сутності 1 | Назва сутності 2 | Назва зв'язку                                                                  | Вид зв'язку |
|------------------|------------------|--------------------------------------------------------------------------------|-------------|
| Проєкт           | Клієнти          | Містить інформацію про проєкт, замовлений клієнтом                             | 1:N         |
| Проєкт           | Фінанси          | Містить інформацію про проєкт, завдяки якому була проведена фінансова операція | 1:N         |
| Проєкт           | Ресурси          | Містить інформацію про проєкт, в якому використовуються ресурси                | 1:N         |
| Проєкт           | Працівники       | Містить інформацію про проєкт, в якому задіяні працівники                      | 1:N         |

Для створення логічної моделі даних, потрібно пройти кілька етапів. На першому етапі необхідно провести спрощення концептуальної моделі шляхом виконання попередніх перетворень, щоб усунути зв'язки, які несумісні з реляційною моделлю даних. Це означає виконання ряду операцій, таких як видалення двосторонніх зв'язків, складних зв'язків, атрибутів з багатьма значеннями, рекурсивних зв'язків та зв'язків з атрибутами. З урахуванням того, що у проєктованій базі даних відсутні складні зв'язки, можна припустити, що концептуальна модель даних переходить у логічну модель без зайвих ускладнень та зберігає свою структуру.

## 2.3 Проєктування програмного продукту

Процес планування проєкту є важливим етапом у створенні програмного забезпечення, оскільки визначаються цілі, завдання, часові рамки, ресурси та ключові показники ефективності. Для цього часто використовується стандартна мова моделювання Unified Modeling Language (UML), яка є універсальним інструментом для розробників.

UML забезпечує набір графічних позначень і методів для розуміння, проєктування та документування програмних систем. Вона включає діаграми класів, об'єктів, компонентів, діяльності, станів та інших структур, що дозволяють моделювати як статичні, так і динамічні аспекти системи. Використання UML дозволяє ефективно комунікувати і координувати зусилля учасників проєкту на всіх стадіях розробки.

Завдяки UML, всі учасники проєкту мають єдине уявлення про структуру та функціонування системи, що знижує ризики непорозумінь та помилок. Графічні моделі допомагають обговорювати ідеї, аналізувати рішення та приймати обґрунтовані рішення, що важливо для успіху командних проєктів.

Таким чином, планування проєкту за допомогою UML сприяє організованому підходу до розробки програмного забезпечення, забезпечуючи високу якість кінцевого продукту і задоволення вимог користувачів.

Для проєктування програмного забезпечення необхідно створити наступні діаграми:

- Діаграма використання: Ця діаграма представляє собою графічну нотацію, яка відображає конкретні варіанти використання системи, акторів, можливі інтерфейси та взаємозв'язки між ними. Вона визначає послідовність дій для взаємодії з актором у кожному конкретному випадку використання [7]. Приклад діаграми використання для програми обліку проєктів можна побачити на рисунку 2.1.

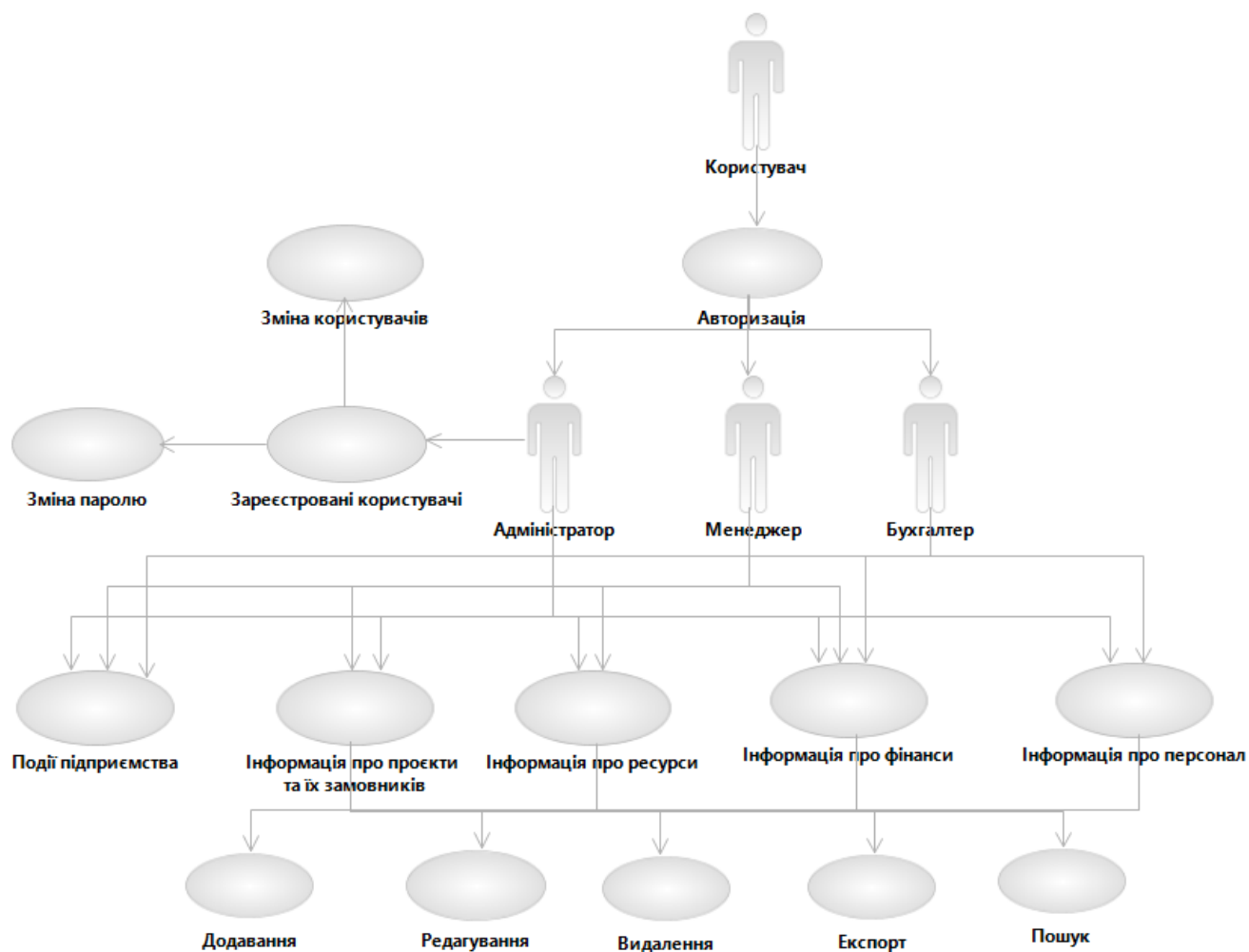


Рисунок 2.1 – Діаграма використання для обліку проєктів

– Діаграма класів: Ця діаграма є статичним представленням структури моделі, яке включає символи класів та їх зв'язки. Іконки класів зазвичай мають форму прямокутників, які можуть містити ім'я класу, його атрибути та операції. Така діаграма допомагає візуалізувати структуру програми та взаємозв'язки між класами [8]. Діаграма класів для програми обліку проєктів представлена на рисунку 2.2.

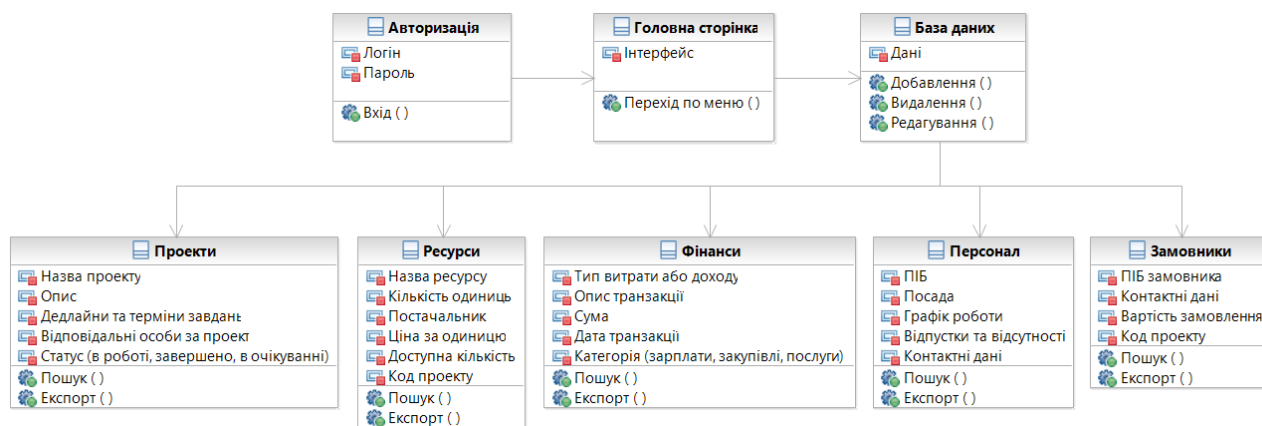


Рисунок 2.2 – Діаграма класів для програми обліку проєктів

– Діаграма послідовності: Ця діаграма надає візуальне представлення взаємодії об'єктів у конкретному сценарії. Кожному об'єкту сценарію відповідає лінія життя, яка відображає послідовність подій. Візуалізація взаємодії об'єктів у сценарії включає вертикальні пунктирні лінії, що позначають життєвий цикл об'єктів, та крайні праві лінії, які відображають зовнішнє середовище, з яким взаємодіє сценарій [9]. Діаграма послідовностей для програми обліку проєктів наведена на рисунку 2.3.

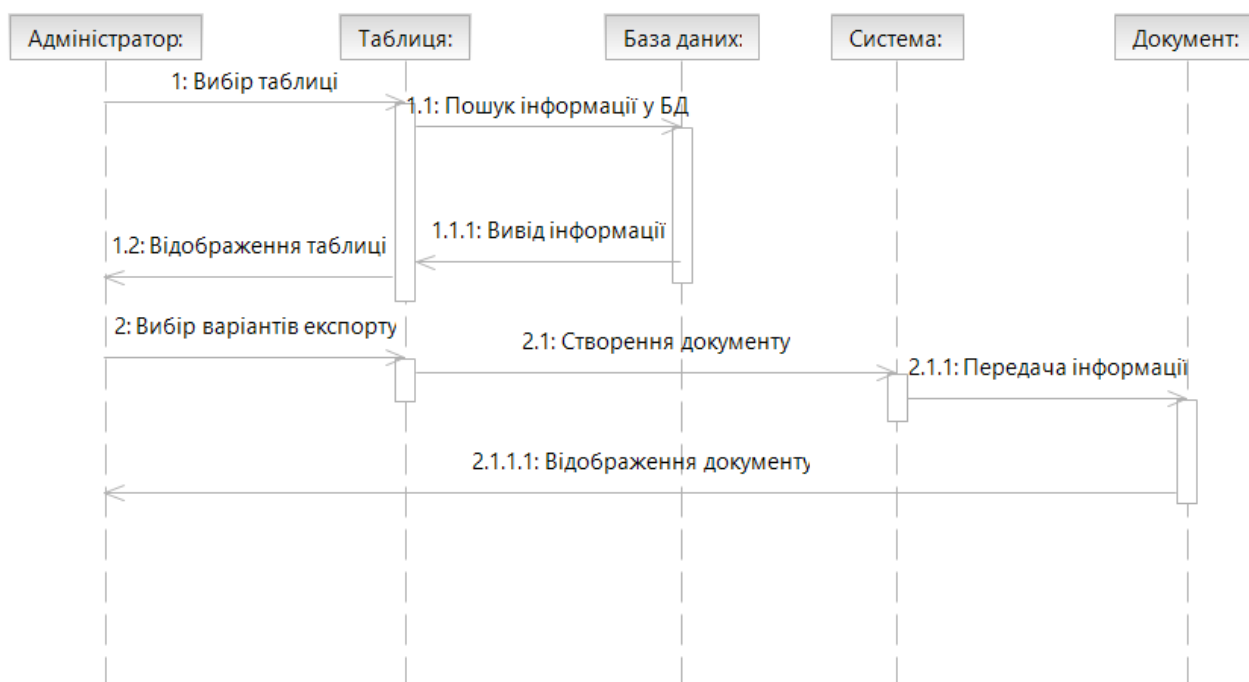


Рисунок 2.3 – Діаграма послідовностей для програми обліку проєктів

## 3 ПРАКТИЧНА ЧАСТИНА

### 3.1 Проектування користувацького інтерфейсу

Розроблена програма включає в себе вікно для авторизації користувачів, яке з'являється при запуску. Візуальне зображення цього вікна представлено на рисунку 3.1.

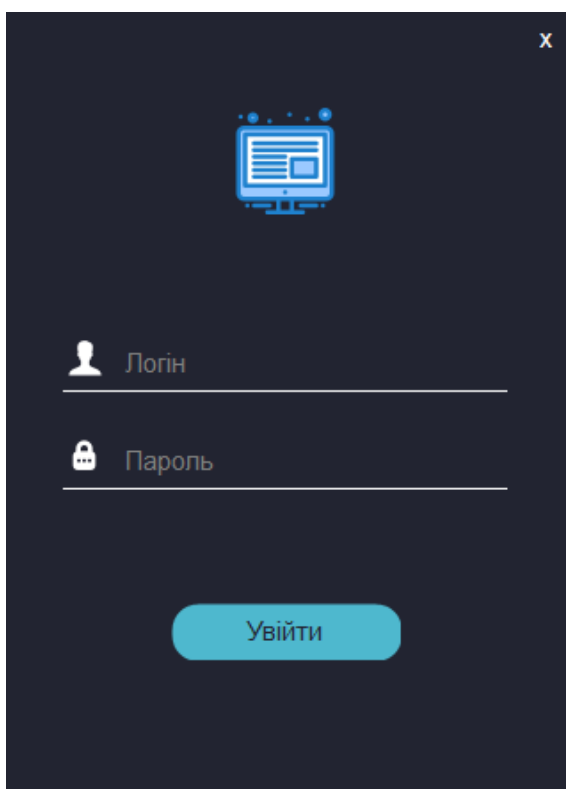


Рисунок 3. 1 – Вікно авторизації

Після того як користувач успішно авторизується в системі, відкривається головне вікно програми (див. Рис. 3.2). У цьому основному вікні розташовані різноманітні кнопки, які надають доступ до всіх функцій проєкту та забезпечують зручну навігацію. Окрім цього, є робоча зона, де відображається вся необхідна інформація. У цій зоні також можна побачити таблицю, що містить дані про різні проєкти комунального підприємства.

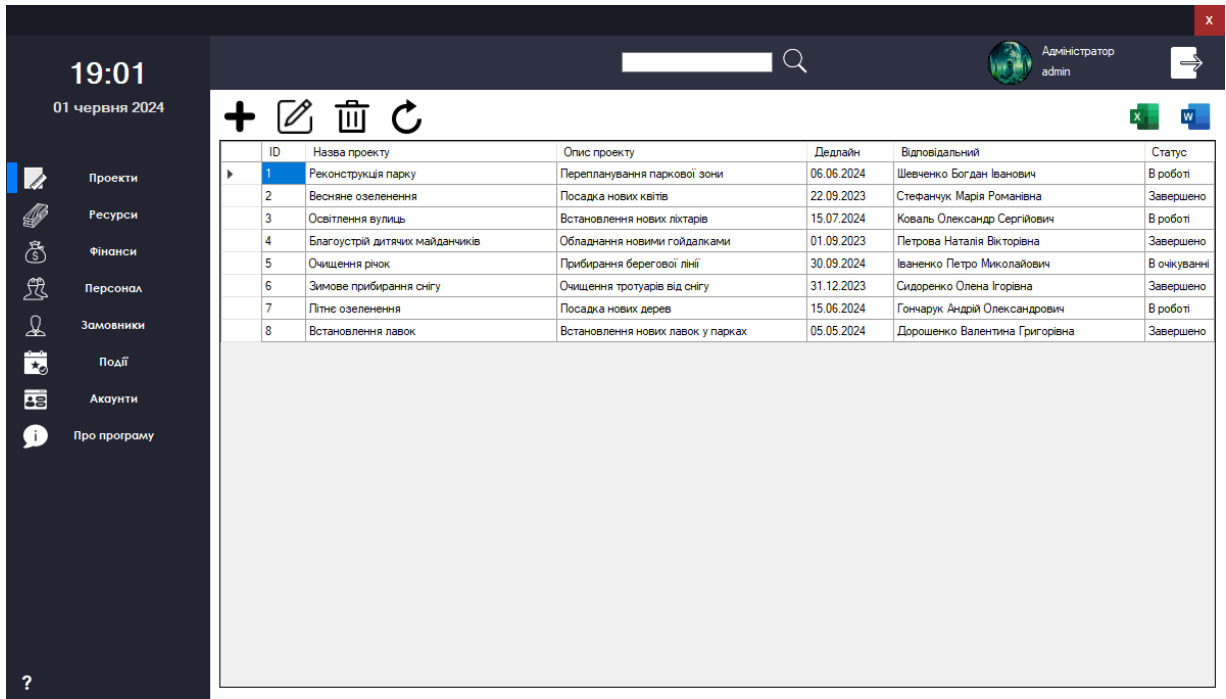


Рисунок 3. 2 – Головне вікно програми з інформацією про проекти

Актуалізація даних про ресурси, включаючи їх додавання, видалення та редагування, здійснюється у розділі «Ресурси», що представлений на рисунку 3.3. Інтерфейс вкладки для внесення нової інформації про ресурс можна побачити на рисунку 3.4.

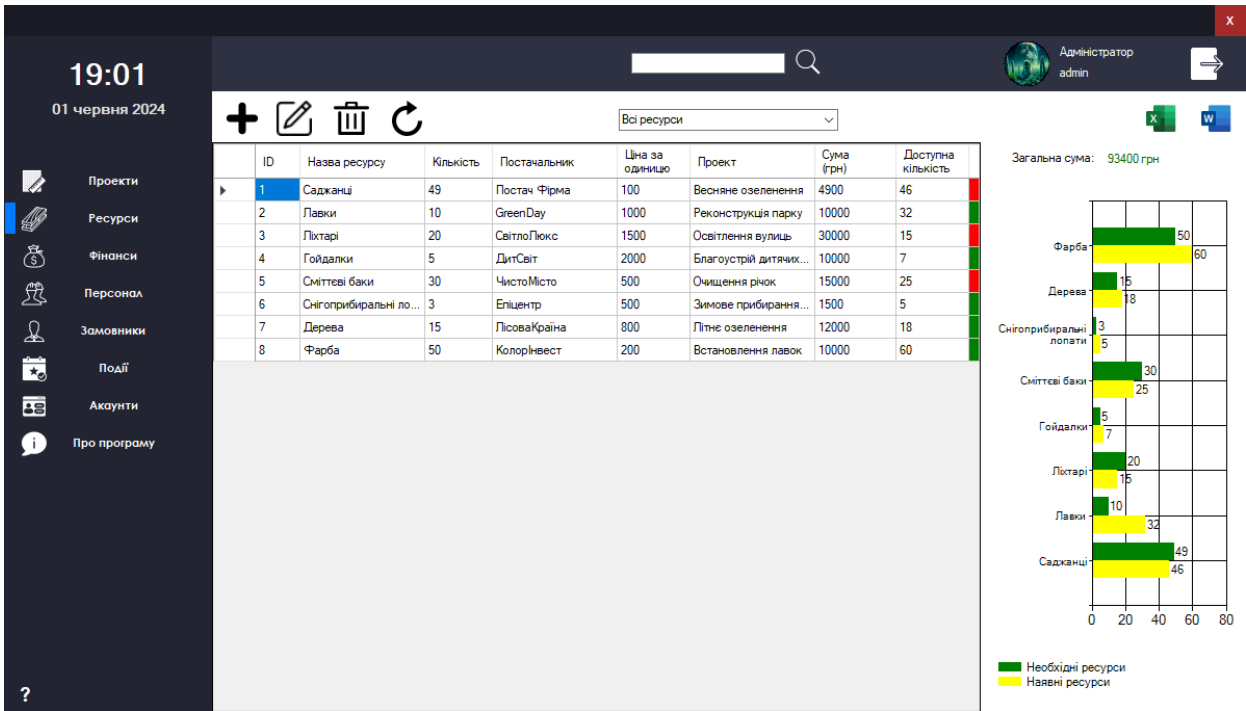


Рисунок 3. 3 – Вигляд вкладки «Ресурси»

Рисунок 3. 4 – Інтерфейс вікна додавання нового ресурсу

Інформацію про фінансові операції можна переглядати та редагувати у розділі «Фінанси» (див. Рис. 3.5).

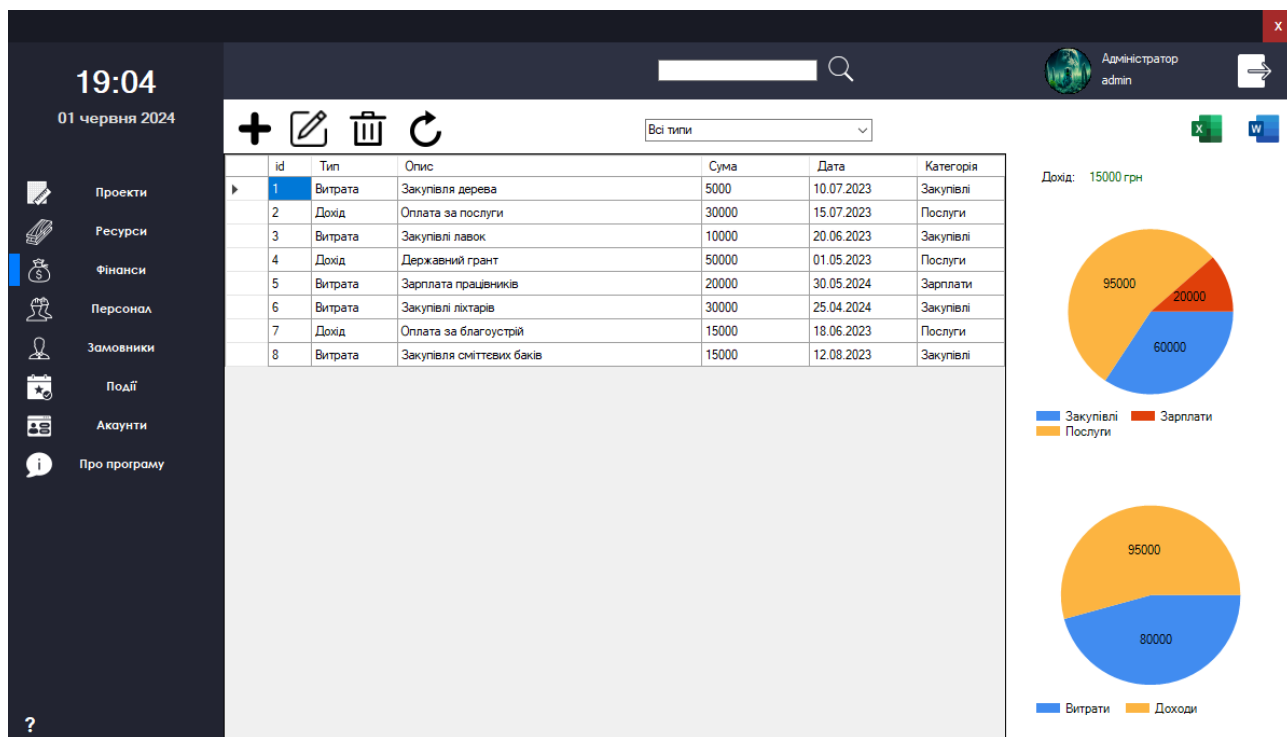


Рисунок 3. 5 – Вигляд вкладки «Фінанси»

У розділі "Персонал" ви зможете отримати повну інформацію про персонал та їх графік роботи. Саме цю вкладку можна побачити на рисунку 3.6.



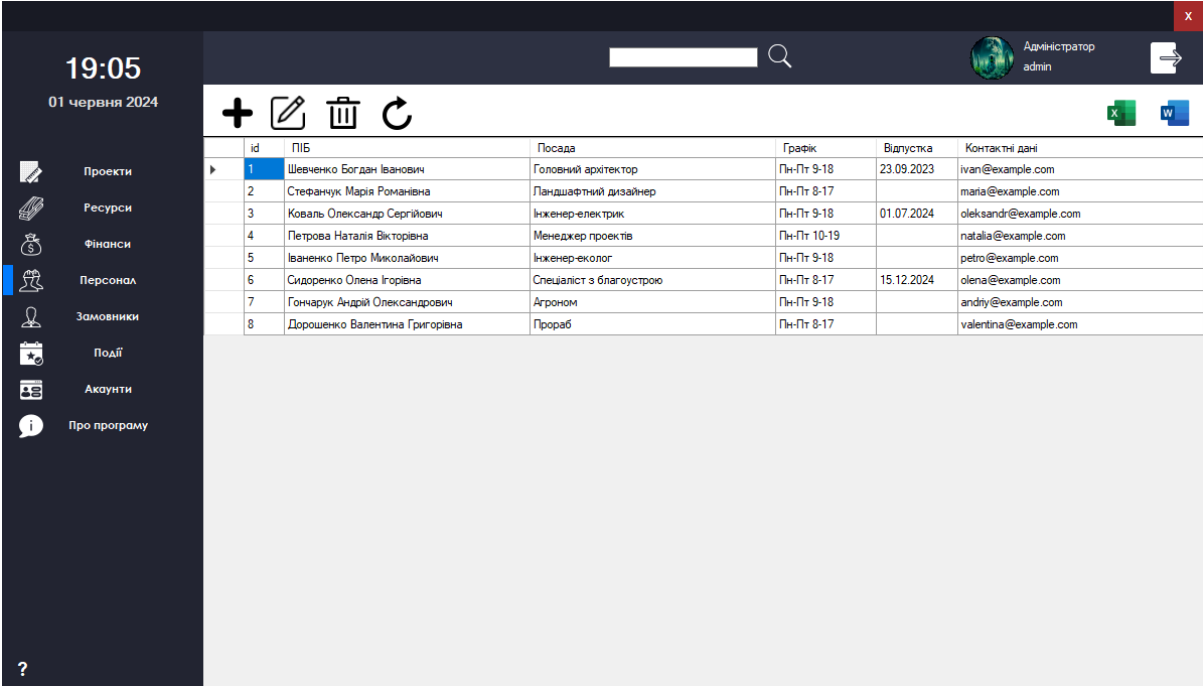


Рисунок 3.6 – Вигляд вкладки «Персонал»

Для внесення змін щодо замовників, таких як додавання нового клієнта, видалення чи редагування інформації щодо вже наявних, слід перейти до вкладки «Замовники» (див. Рис. 3.7).

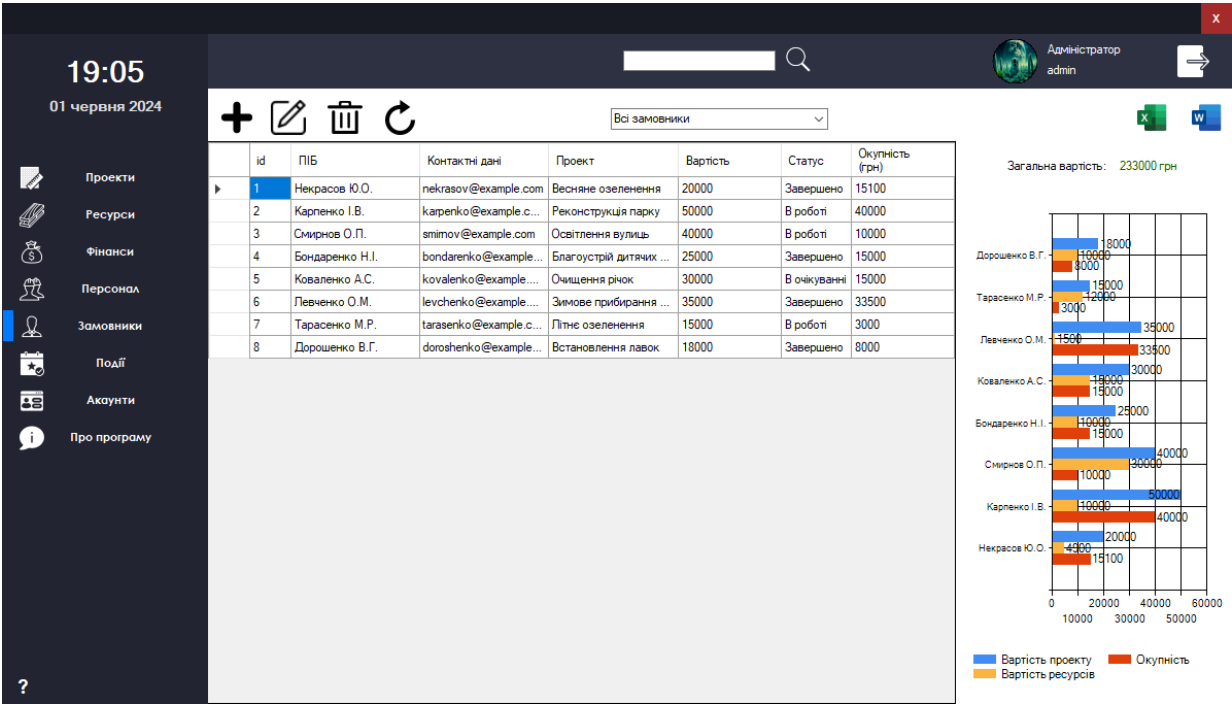


Рисунок 3. 7 – Вигляд вкладки «Замовники»

В рамках даного проекту передбачено включення розділу, який відображатиме актуальні події, такі як відпустки працівників, завершення проєктів і недостатність ресурсів. Цей розділ має назву "Події". Зразок вигляду розділу для відображення інформації про майбутні події наведено на рисунку 3.8.

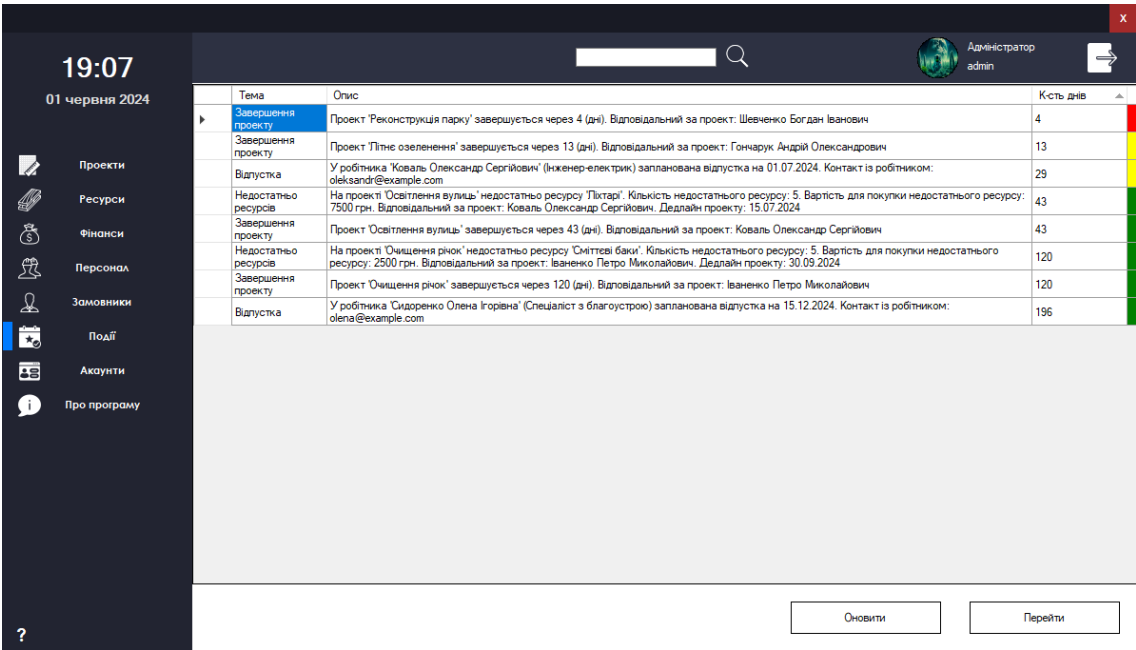


Рисунок 3. 8 – Вигляд вкладки «Події»

Адміністратор програми має право на зміну паролів зареєстрованих користувачів, що доступно у розділі «Акаунти», який зображено на рисунку 3.9.

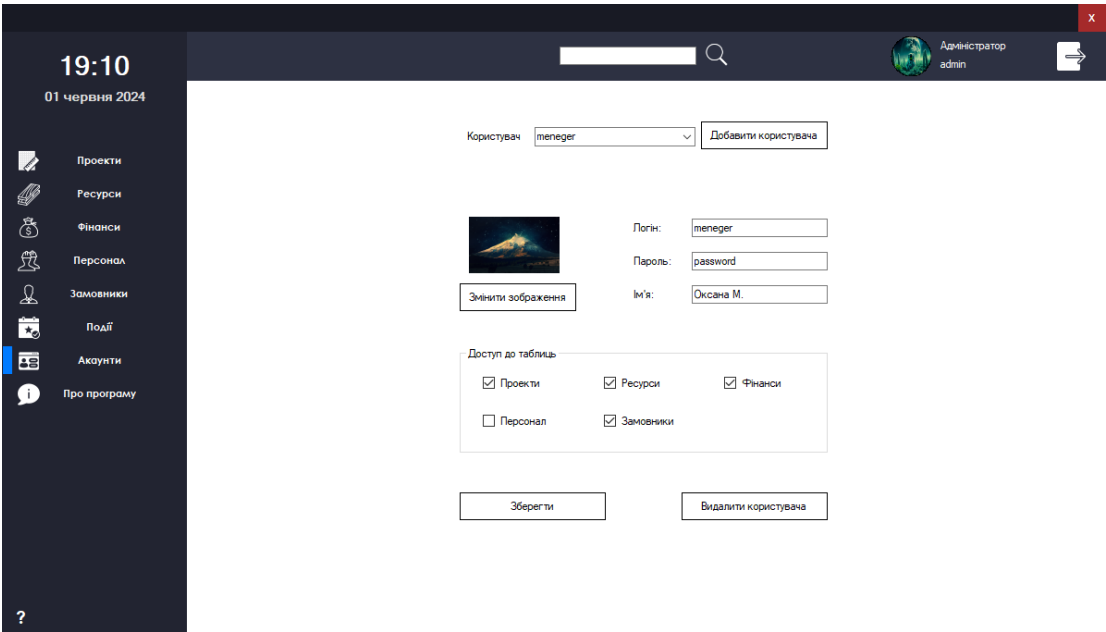


Рисунок 3. 9– Вигляд вкладки «Акаунти»

Інформація, що описує програму, може бути знайдена у вкладці з назвою «Про програму», яку можна знайти нижче на рисунку 3.10.

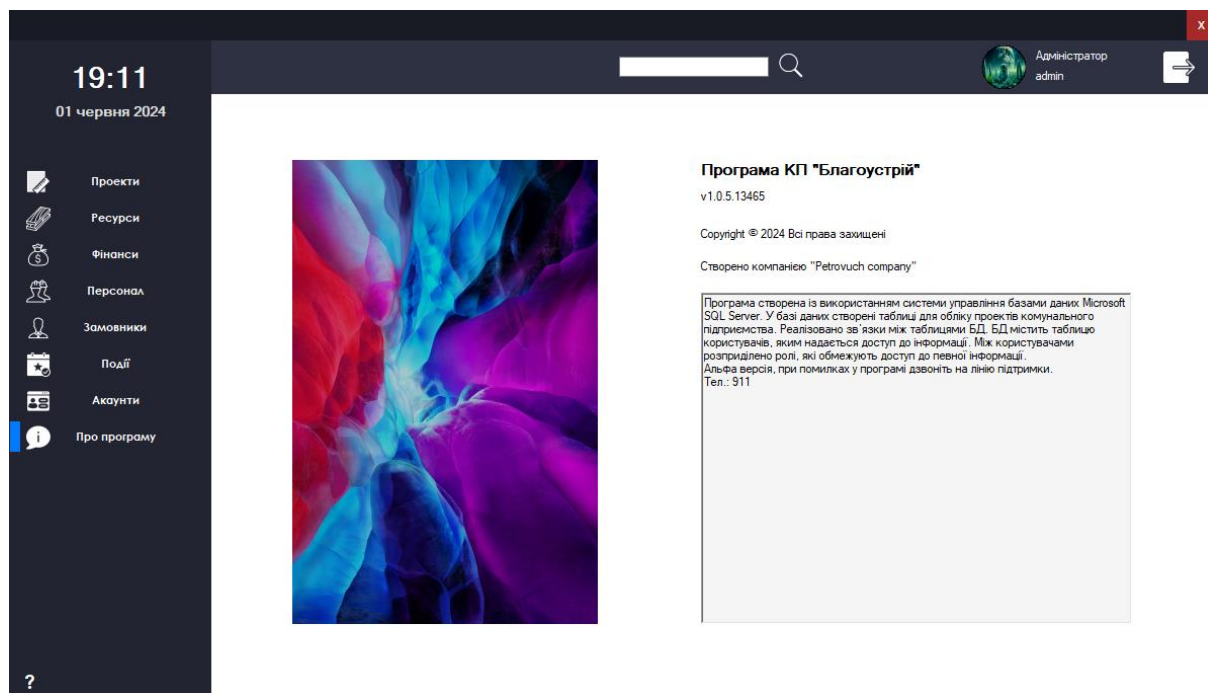


Рисунок 3. 10 – Вигляд вкладки «Про програму»

На рисунку 3.11 показано довідку до програми, що відкривається при натисненні кнопки із знаком питання.

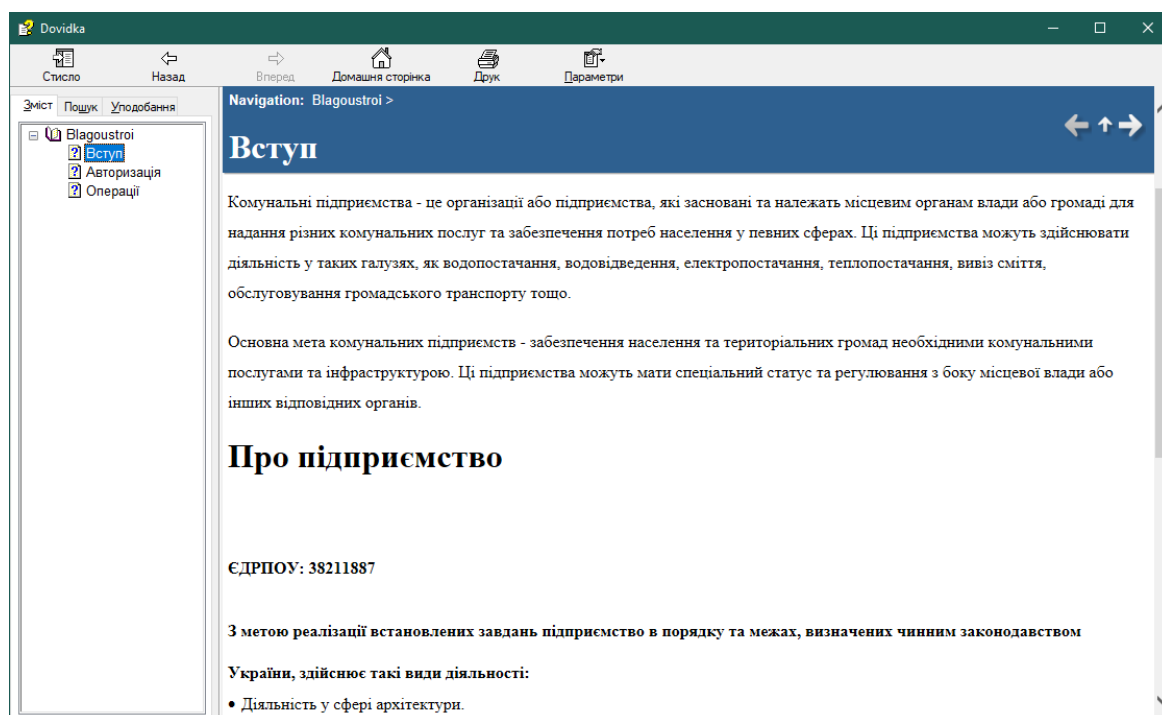


Рисунок 3. 11 – Вигляд довідки програми

### 3.2 Опис модулів програмного забезпечення

Програма була розроблена у середовищі Visual Studio, із використанням мови програмування C#. Для збереження даних використовується база даних, створена на платформі управління базами даних Microsoft SQL Server. Взаємодія з базою даних відбувається шляхом виконання SQL-запитів [10].

Для відображення даних про ресурси з бази даних у таблицю використовується SQL-запит, зображений на рисунку 3.12.

```
Select r.id_resource, r.name_resource,
r.count, r.provider, r.price, p.name
AS projects, r.available_amount
From resource r LEFT JOIN projects p
ON r.kod_project = p.id
```

Рисунок 3.12 – SQL-запит для виведення інформації про ресурси з бази даних

Процес занесення інформації про ресурси до бази даних здійснюється шляхом виконання запиту мовою SQL, який можна знайти на рисунку 3.13.

```
INSERT INTO resource VALUES (" + nomer + ", '" +
form_addresource.textBox1.Text + "', " +
form_addresource.numericUpDown1.Value + ", '" +
form_addresource.textBox2.Text + "', " +
form_addresource.textBox3.Text + ", NULL, " +
form_addresource.numericUpDown2.Value + ")
```

Рисунок 3.13 – SQL-запит для введення інформації про ресурси в базу даних

Для виконання процесу вилучення інформації про ресурси використовується SQL-запит, що зображений на рисунку 3.14.

```
DELETE FROM resource Where id_resource=" +  
dataGridView1.CurrentRow.Cells[0].Value
```

Рисунок 3.14 – SQL-запит для видалення інформації про ресурси з бази даних

Для збереження відредагованих даних про ресурси було використано SQL-запит, який можна побачити на рисунку 3.15.

```
UPDATE resource SET name_resource='" +  
form_addresource.textBox1.Text + "', count='" +  
form_addresource.numericUpDown1.Value + "', provider='" +  
form_addresource.textBox2.Text + "', price='" +  
form_addresource.textBox3.Text + "', kod_project='" + id_project + "',  
available_amount='" + form_addresource.numericUpDown2.Value + "  
Where id_resource=" + dataGridView1.CurrentRow.Cells[0].Value
```

Рисунок 3.15 – SQL-запит для редагування інформації про ресурси в базі даних

У таблиці 3.1 надано докладний огляд основних програмних модулів.

Таблиця 3. 1 – Опис програмних модулів програми підприємства

| №  | Ім'я програмного модуля | Огляд програмного модуля                                                      |
|----|-------------------------|-------------------------------------------------------------------------------|
| 1  | 2                       | 3                                                                             |
| 1  | AboutForm               | Описується короткий опис програми, її версія та назва                         |
| 2  | AccountsForm            | Форма для зміни даних акаунтів адміністратором, а також їх доступу до таблиць |
| 3  | AddAccountForm          | Форма для додавання нового користувача програми                               |
| 4  | AddClientForm           | Форма для заповнення даних про нового замовника підприємства                  |
| 5  | AddFinanceForm          | Форма для додавання нової фінансової операції                                 |
| 6  | AddProjectForm          | Форма для додавання нового проєкту                                            |
| 7  | AddResourceForm         | Форма для додавання нового ресурсу                                            |
| 8  | AddStaffForm            | Форма для додавання нового працівника                                         |
| 9  | ChangePassForm          | Форма для зміни паролю та інших даних акаунту                                 |
| 10 | ClientsForm             | Виведення інформації про замовників та керування нею                          |
| 11 | LoginForm               | Авторизація користувачів у програму                                           |
| 12 | MainForm                | Головне вікно програми                                                        |

Продовження таблиці 3.1

| 1  | 2                | 3                                                            |
|----|------------------|--------------------------------------------------------------|
| 13 | NoPermissionForm | Виведення повідомлення про відсутність доступу до інформації |
| 14 | NotificationForm | Виведення сповіщень про події комунального підприємства      |
| 15 | FinancesForm     | Виведення та керування інформацією про фінансові операції    |
| 16 | ProjectsForm     | Виведення та керування інформацією про проекти               |
| 17 | ResourceForm     | Виведення та керування інформацією про ресурси               |
| 18 | StaffForm        | Виведення та керування інформацією про персонал              |

Код програми наведено у додатках В та Д.

### 3.3 Результати тестування програми

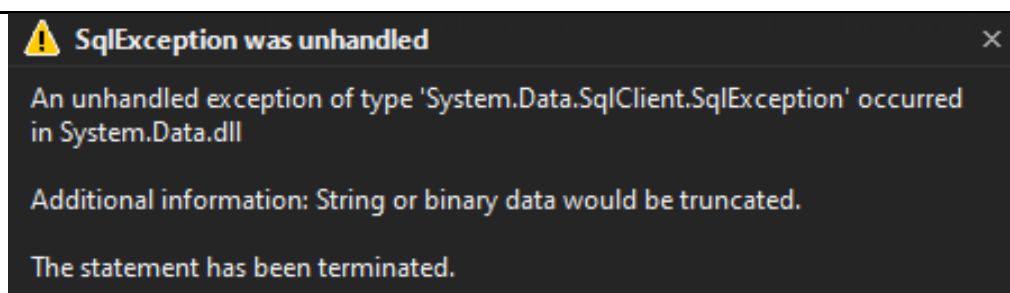
Перед початковим використанням програми рекомендується провести тестування для мінімізації виникнення помилок у подальшому. Тестування включатиме в себе наступні етапи:

- тестування вручну;
- тестування функціональності;
- тестування модулів.

Ручне тестування — це процес, при якому тестувальник вручну виконує тестові сценарії без використання автоматизації. У цьому випадку тестувальник виступає в ролі кінцевого користувача програми, випробовуючи її функціональні можливості з метою виявлення помилок та недоліків. Зазвичай для ефективного проведення ручного тестування тестувальник використовує попередньо складений план, який містить детальні вказівки щодо випадків тестування [11].

Внаслідок виконання ручного тестування здійснюється складання структури Bug-report. У таблиці 3.2 наведено описання структури Bug-report.

Таблиця 3. 2 - Структура баг-репорту

| Програма обліку проєктів комунального підприємства            |                                                                                                                                                                                                                                                              |
|---------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1                                                             | 2                                                                                                                                                                                                                                                            |
| Короткий опис                                                 | При редагуванні імені користувача на таке що перевищує ліміт символів вказаний у БД, виникає помилка збереження                                                                                                                                              |
| Проект                                                        | BlagoustriyManagement                                                                                                                                                                                                                                        |
| Модуль додатку                                                | AccountsForm                                                                                                                                                                                                                                                 |
| Номер версії                                                  | Версія v 1.0.5                                                                                                                                                                                                                                               |
| Критичність проблеми                                          | S2 Критичний (Critical)                                                                                                                                                                                                                                      |
| Важливість дефекту                                            | P1 Високий (High)                                                                                                                                                                                                                                            |
| Автор                                                         | Пласконь Назар                                                                                                                                                                                                                                               |
| Призначення                                                   | Пласконь Назар                                                                                                                                                                                                                                               |
| Середовище                                                    |                                                                                                                                                                                                                                                              |
| ОС / Пакет оновлень / Пакет версій / Характеристики апаратури | Windows 10 Professional / Intel Core i3-7020U / RAM 8 Гб / NVIDIA GeForce MX130                                                                                                                                                                              |
| Опис                                                          |                                                                                                                                                                                                                                                              |
| Спосіб відтворення                                            | <ol style="list-style-type: none"> <li>1. Відкрити вкладку акаунти.</li> <li>2. Вибрати одного користувача зі списку.</li> <li>3. У полі з іменем користувача, ввести значення що перевищує 20 символів.</li> <li>4. Натиснути кнопку «Зберегти».</li> </ol> |
| Отриманий результат                                           | Програма вилітає через помилку збереження                                                                                                                                                                                                                    |
| Необхідний результат                                          | Виведення вікна сповіщення, яке вказує про неможливість збереження вказаної інформації                                                                                                                                                                       |
| Додатково                                                     |                                                                                                                                                                                                                                                              |
| Додаткова інформація                                          |                                                                                                                                                                          |

Помилка була усунута шляхом додавання виключення до коду, який виконує збереження інформації у базу даних. Деталі цього виключення можна знайти у лістингу 3.1.

## Лістинг 3.1 – Код виключення для виправлення помилки збереження

```

try {
    string tableAccess = "";
    if (checkBoxProjects.Checked) tableAccess += "projects ";
    if (checkBoxResource.Checked) tableAccess += "resources ";
    if (checkBoxFinances.Checked) tableAccess += "finances ";
    if (checkBoxStaff.Checked) tableAccess += "staff ";
    if (checkBoxClients.Checked) tableAccess += "clients ";
    LoginForm fl = new LoginForm(); fl.connection.Open();
    SqlCommand command = fl.connection.CreateCommand();
    var x = new LoginForm.XORCipher();
    var encryptedMessageByPass1 = x.Encrypt(textBox2.Text);
    command.CommandText = "UPDATE user SET login='" + textBox1.Text +
    "', password='" + encryptedMessageByPass1 + "', username='" +
    textBox3.Text + "', tables_access='" + tableAccess + "', image_url='"
    + pictureBox1.ImageLocation + "' Where login='" + comboBox1.Text + "'";
    command.ExecuteNonQuery();
    fl.connection.Close();
    if (textBox1.Text == "admin") {
        mainFormInstance.pictureBox1.ImageLocation =
        pictureBox1.ImageLocation;
        mainFormInstance.label1.Text = textBox3.Text; }
    MessageBox.Show("Дані успішно змінено!", "",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
    comboBox1.Text = textBox1.Text; }
catch {
    MessageBox.Show("Не вдалося змінити дані!", "Помилка",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    load_User_Info(comboBox1.Text); }

```

Тепер необхідно виконати процедуру функціонального тестування.

Функціональне тестування включає у себе перевірку функціональних можливостей програмного продукту, а також виявлення відмінностей між фактичним функціоналом та очікуваним, згідно з вимогами та специфікацією. Цей процес спрямований на імітацію реального використання системи з метою виявлення можливих проблем [12].

У результаті функціонального тестування формується набір тест-кейсів, які допомагають у перевірці відповідності програмного забезпечення вимогам та специфікації.

Інформацію про результати тестування програми, проведеного функціональним методом, можна знайти у таблицях 3.3 та 3.4.



Таблиця 3. 3 – Стислий огляд тест-кейсів

| №  | Тип      | Опис                                                                                                                 | Очікувана реакція                                                    | Фактична реакція                                                | Pass/Fail |
|----|----------|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|-----------------------------------------------------------------|-----------|
| 1  | 2        | 3                                                                                                                    | 4                                                                    | 5                                                               | 6         |
| 01 | positive | Перевірка можливості експорту в документи Word та Excel                                                              | Всі дані можна експортувати у вигляді таблиць в документи Word/Excel | Всі дані експортуються у вигляді таблиць в документи Word/Excel | Pass      |
| 02 | positive | Перевірка правильності обчислення доходу (при сумі загальних витрат 80000 грн і загальній сумі надходжень 95000 грн) | Дохід 15000 грн                                                      | Дохід 15000 грн                                                 | Pass      |
| 03 | positive | Перевірка обчислення окупності проєкту (при вартості ресурсів 10000 грн і вартості проєкту 18000 грн)                | Окупність проєкту 8000 грн                                           | Окупність проєкту 8000 грн                                      | Pass      |
| 04 | positive | Перевірка обчислення кількості днів, до дедлайну проєкту (при дедлайні 30.09.24 і актуальною датою 03.06.24)         | До настання дедлайну 118 днів                                        | До настання дедлайну 118 днів                                   | Pass      |

Таблиця 3. 4 – Детальний огляд тест-кейсів

|                                   |                                                                                                                      |                                                                 |                 |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|-----------------|
| Найменування тесту                | Перевірка можливості експорту в документи Word та Excel                                                              |                                                                 |                 |
| Функція                           | Microsoft.Office.Interop.Excel<br>Microsoft.Office.Interop.Word                                                      |                                                                 |                 |
| Дата тестування                   | 03/06/2024                                                                                                           |                                                                 |                 |
| Дія                               | Очікувана реакція                                                                                                    | Фактична реакція                                                | PASS/FAIL/BLOCK |
| Експорт в документи Word та Excel | Всі дані можна експортувати у вигляді таблиць в документи Word/Excel                                                 | Всі дані експортуються у вигляді таблиць в документи Word/Excel | Pass            |
| Найменування тесту                | Перевірка правильності обчислення доходу (при сумі загальних витрат 80000 грн і загальній сумі надходжень 95000 грн) |                                                                 |                 |
| Функція                           | System.Data.SqlClient                                                                                                |                                                                 |                 |
| Дата тестування                   | 03/06/2024                                                                                                           |                                                                 |                 |
| Дія                               | Очікувана реакція                                                                                                    | Фактична реакція                                                | PASS/FAIL/BLOCK |
| Обчислення доходу підприємства    | Дохід 15000 грн                                                                                                      | Дохід 15000 грн                                                 | Pass            |
| 1                                 | 2                                                                                                                    |                                                                 |                 |

Продовження таблиці 3.4

| 1                                              | 2                                                                                                            |                               |                 |
|------------------------------------------------|--------------------------------------------------------------------------------------------------------------|-------------------------------|-----------------|
| Найменування тесту                             | Перевірка обчислення окупності проєкту (при вартості ресурсів 10000 грн і вартості проєкту 18000 грн)        |                               |                 |
| Функція                                        | System.Data.SqlClient                                                                                        |                               |                 |
| Дата тестування                                | 03/06/2024                                                                                                   |                               |                 |
| Дія                                            | Очікувана реакція                                                                                            | Фактична реакція              | PASS/FAIL/BLOCK |
| Обчислення окупності проєкту                   | Окупність проєкту 8000 грн                                                                                   | Окупність проєкту 8000 грн    | Pass            |
| Найменування тесту                             | Перевірка обчислення кількості днів, до дедлайну проєкту (при дедлайні 30.09.24 і актуальною датою 03.06.24) |                               |                 |
| Функція                                        | System.Data.SqlClient                                                                                        |                               |                 |
| Дата тестування                                | 03/06/2024                                                                                                   |                               |                 |
| Дія                                            | Очікувана реакція                                                                                            | Фактична реакція              | PASS/FAIL/BLOCK |
| Обчислення кількості днів, до дедлайну проєкту | До настання дедлайну 118 днів                                                                                | До настання дедлайну 118 днів | Pass            |

Завершальним кроком у процесі тестування буде виконання процедури модульного тестування.

Модульне тестування, що є необхідним етапом у процесі розробки програмного забезпечення, включає в себе використання різноманітних методів з метою запобігання виявлення дефектів та розробки стратегій для зменшення ризиків, часових витрат і фінансових зусиль, потрібних для розробки програмного продукту. Виконання цього виду тестування може здійснюватися розробником програмного забезпечення або інженером під час створення програмного продукту в межах життєвого циклу розробки. Основна ціль проведення модульного тестування полягає в виявленні та виправленні помилок, які можуть виникнути в процесі проєктування програмного забезпечення. Ця стратегія націлена на забезпечення високої якості програмного продукту згідно з вимогами процесу контролю якості [13].

В файл LoginTests.cs, який входить у склад проєкту WindowsFormsApplication1Tests.Test, буде внесено модульні тести (див. лістинг 3.2).

### Лістинг 3.2 – Тест для перевірки функції шифрування

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
namespace WindowsFormsApplication1Tests
{
    [TestClass]
    public class LoginTests
    {
        [TestMethod]
        public void Encrypt_pass()
        {
            // arrange
            var x = new XORCipher();
            string arg="password";
            string expected = "ufxx|twi";
            // act
            string result = x.Encrypt(arg);
            // assert
            Assert.AreEqual(expected,result);
        }
        [TestMethod]
        public void Decrypt_pass()
        {
            // arrange
            var x = new XORCipher();
            string arg = "firms";
            string expected = "admin";
            // act
            string result = x.Decrypt(arg);
            // assert
            Assert.AreEqual(expected, result);
        }
    }
}
```

У лістингу 3.3 наведено програмний код, який виконує процес шифрування.

### Лістинг 3.3 – Код функції шифрування

```
public class XORCipher
{
    public string Encrypt(string text)
    {
        string res = "";
        for (int i = 0; i < text.Length; i++)
        { res += (char)((text[i] + 5)); }
        return res;
    }
    public string Decrypt(string text)
    {
        string res = "";
```

### Продовження лістингу 3.3

```

        for (int i = 0; i < text.Length; i++)
        { res += (char)((text[i] - 5)); }
        return res;
    }
}

```

Рисунок 3.16 відображає отримані результати проходження тестів.

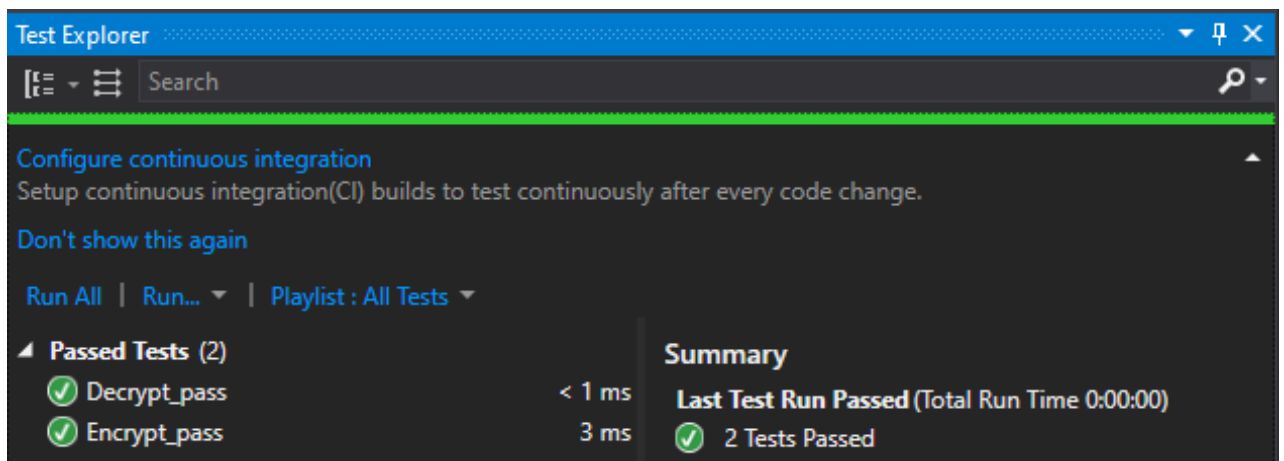


Рисунок 3.16 – Результат проходження тесту

Після проведення трьох різновидів тестів можна зробити висновок, що програма містила невелику кількість помилок, які були виправлені після їх виявлення.

## 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Актуальність безпеки життєдіяльності людини

Безпека життєдіяльності є однією з найважливіших сфер сучасного суспільства, яка забезпечує умови для безпечної роботи та життя людини. Особливо це актуально для підприємств, що займаються благоустроєм міста, таких як комунальні підприємства, де існує широкий спектр діяльності, пов'язаної з різними ризиками для здоров'я та життя працівників.

На підприємствах, які займаються благоустроєм, рівень травматизму може бути значним через специфіку виконуваних робіт. Основні причини травматизму включають фізичні навантаження під час посадки та догляду за рослинами, збирання та утилізації відходів, роботу з механізмами та інструментами, використання газонокосарок, бензопил, садових інструментів тощо.

На підприємстві можна виділити кілька основних факторів, що впливають на безпеку працівників [14]:

- Вражаючі фактори;
- Небезпечні фактори;
- Шкідливі фактори.

Вражаючі фактори включають випадки падіння з висоти під час обрізки дерев або монтажу декоративних елементів на висоті, а також небезпечні ситуації при роботі з важкою технікою, які можуть призвести до загибелі.

Небезпечні фактори виникають при порушенні правил техніки безпеки під час використання електроінструментів, що може викликати травми, опіки чи ураження електричним струмом. Крім того, фізичні навантаження можуть викликати раптове погіршення здоров'я, як-от біль у спині. А постійний шум від робочого обладнання може призвести до зниження слуху.

Шкідливі фактори включають тривалий вплив хімічних речовин, які використовуються для догляду за рослинами, що може викликати хронічні захворювання або алергічні реакції. Пил і забруднення, що виникають під час

збирання та утилізації відходів, можуть спричинити респіраторні захворювання. Додатково, робота в умовах високих або низьких температур, тривалий вплив ультрафіолетового випромінювання при виконанні завдань на відкритому повітрі можуть викликати теплові удари, опіки чи гіпотермію.

Важливо враховувати законодавчі акти України, які регулюють питання безпеки праці та здоров'я працівників. Відповідно до Закону України «Про загальнообов'язкове державне соціальне страхування», всі працівники мають право на соціальне страхування у разі нещасних випадків на виробництві та професійних захворювань. Це забезпечує компенсацію у разі втрати працездатності або тимчасової непрацездатності, а також фінансову допомогу в разі виникнення нещасного випадку [15].

Також Постанова Кабінету Міністрів України «Про затвердження Порядку проведення медичних оглядів працівників певних категорій» визначає обов'язковість проведення періодичних медичних оглядів для працівників, які працюють в умовах підвищеної небезпеки або піддаються впливу шкідливих факторів. Цей порядок забезпечує своєчасне виявлення професійних захворювань і запобігає їхньому розвитку, що є важливою складовою профілактики виробничих травм і захворювань [16].

Таким чином, комплексна оцінка всіх потенційних небезпек і розробка відповідних заходів захисту є критично важливими для забезпечення безпеки працівників на підприємствах. Це включає впровадження належних процедур безпеки, навчання персоналу, використання засобів індивідуального захисту та регулярний моніторинг умов праці, відповідно до чинного законодавства України.

#### 4.2 Проведення інструктажів з охорони праці

Інструктажі з охорони праці є невід'ємною частиною системи забезпечення безпеки на підприємстві. Вони спрямовані на ознайомлення працівників з

правилами та вимогами охорони праці, а також навчання їх правильним діям у небезпечних ситуаціях. Для комунального підприємства, що займається благоустроєм та має значну кількість працівників, які працюють за ПК, проведення інструктажів з охорони праці має особливе значення.

Основні види інструктажів з охорони праці включають [17]:

- Вступний інструктаж: проводиться з усіма новими працівниками незалежно від їх попереднього досвіду та кваліфікації. Метою цього інструктажу є ознайомлення нових співробітників з основними правилами та вимогами охорони праці на підприємстві, а також з організаційною структурою, основними небезпеками і заходами щодо їх запобігання. Цей інструктаж проводиться фахівцем з охорони праці або іншим відповідальним працівником, який має відповідну підготовку.

- Первинний інструктаж на робочому місці: проводиться безпосередньо на робочому місці перед початком виконання обов'язків. Він включає ознайомлення працівника з конкретними умовами праці, можливими небезпеками та заходами їх запобігання. Також працівник отримує інформацію про безпечні методи роботи з обладнанням, інструментами і матеріалами. Цей інструктаж є важливим для нових працівників і тих, хто переводиться на нове робоче місце.

- Повторний інструктаж: проводиться не рідше одного разу на півроку з метою закріплення знань працівників щодо безпечних методів роботи та правил охорони праці. Повторні інструктажі дозволяють підтримувати високу обізнаність працівників про безпеку на робочому місці, а також надають можливість актуалізувати інформацію про нові вимоги та правила.

- Позаплановий інструктаж: проводиться у випадках змін у технологічному процесі, введення в дію нового обладнання, а також у разі виявлення порушень правил охорони праці. Також позаплановий інструктаж проводиться після аварій та нещасних випадків, які сталися на підприємстві, для запобігання їх повторенню.

- Цільовий інструктаж: проводиться при виконанні разових робіт, які не передбачені посадовими обов'язками працівника, або у разі необхідності

проведення робіт підвищеної небезпеки. Цей інструктаж спрямований на ознайомлення працівника з особливостями виконання конкретного завдання і необхідними заходами безпеки.

Згідно із Законом України "Про охорону праці", стаття 18, роботодавець зобов'язаний забезпечити організацію навчання та інструктажів з охорони праці. Це включає не лише вступні та первинні інструктажі, а й повторні, позапланові та цільові інструктажі [18].

Проведення інструктажів з охорони праці повинно бути документально оформленим. Ведення журналу реєстрації інструктажів є обов'язковим, що дозволяє контролювати своєчасність та повноту проведених заходів. У журналі зазначаються дати проведення інструктажів, теми, імена інструкторів та підписані працівниками відмітки про проходження інструктажу.

Ефективність інструктажів значною мірою залежить від кваліфікації та компетенції осіб, що їх проводять. Тому важливо, щоб інструктори мали відповідну підготовку та постійно підвищували свою кваліфікацію. Крім того, необхідно використовувати сучасні методи навчання, включаючи мультимедійні презентації, навчальні відеоролики та інтерактивні заняття, що сприяють кращому засвоєнню матеріалу працівниками. Впровадження сучасних технологій дозволяє зробити процес навчання більш ефективним і цікавим для працівників.

Для працівників, які працюють за ПК, особлива увага приділяється питанням ергономіки робочого місця, профілактиці професійних захворювань, пов'язаних із тривалою роботою за комп'ютером, та правилам регулярних перерв для зниження напруження очей і запобігання проблемам зі здоров'ям. Наприклад, інструктаж може включати рекомендації щодо правильної посадки, розташування монітора, клавіатури і миші, а також вправи для зняття напруги з очей і м'язів.

Таким чином, регулярне проведення інструктажів з охорони праці є ключовим елементом забезпечення безпеки життєдіяльності на комунальному підприємстві. Це дозволяє мінімізувати ризики для здоров'я працівників, підвищити їх обізнаність щодо правил безпеки та створити сприятливі умови для ефективної та безпечної роботи.



## ВИСНОВКИ

Впровадження комп'ютерної системи обліку на комунальному підприємстві "Благоустрій" сприятиме швидкому та ефективному організуванню та обробці облікової інформації, що в свою чергу призведе до значного поліпшення роботи підприємства в цілому. Перехід до автоматизованих облікових процесів призведе до численних корисних наслідків, включаючи зниження кількості помилок у обліку, підвищення швидкості здійснення операцій, збільшення контролю над інформацією та скорочення часу, потрібного для введення даних.

Ефективність стратегій розвитку підприємства в значній мірі залежить від інтеграції інформаційних систем, бо саме рівень їх упровадження підвищує конкурентоспроможність та досягнення мети компанії. Таким чином, застосування технологій управління інформацією в сфері бізнесу сприяє збільшенню фінансової вигідності діяльності.

Під час виконання завдання було розроблено інформаційну систему, яка базується на мові програмування C#. Крім того, була розроблена та впроваджена база даних з метою обробки, зберігання та відображення інформації. Для отримання необхідних даних із цієї бази використовуються SQL-запити.

Система призначена для проведення обліку та аналізу ключових подій, що відбуваються на підприємстві, а також надає можливість експортувати потрібні дані у різноманітні документи.

У програмі здійснено авторизацію користувачів, розрахунок чистого доходу, визначення окупності кожного з проєктів, пошук потрібної інформації у вказаних таблицях, а також впроваджено функцію перевірки необхідних ресурсів для реалізації проєкту. Крім того, реалізовано можливість експорту інформації з таблиць в документи Excel і Word.

Впровадження нової програми на комунальному підприємстві "Благоустрій" сприятиме підвищенню ефективності роботи, що, в свою чергу, призведе до зростання фінансових показників компанії.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Комунальні підприємства, установи та організації | Інститут масової інформації [Електронний ресурс] – Режим доступу до ресурсу: <https://imi.org.ua/advice/komunalni-pidpriemstva-ustanovi-ta-organizatsiji-i2365>.
2. Комунальне підприємство “Благоустрій” [Електронний ресурс] – Режим доступу до ресурсу: [http://blagoustriy-kp.ho.ua/?page\\_id=31](http://blagoustriy-kp.ho.ua/?page_id=31).
3. Благоустрій виконавчого комітету Гусятинської селищної ради [Електронний ресурс] – Режим доступу до ресурсу: [https://youcontrol.com.ua/catalog/company\\_details/36145729/](https://youcontrol.com.ua/catalog/company_details/36145729/).
4. Переваги та недоліки програми 1С:Підприємство | Група компаній А4 [Електронний ресурс] – Режим доступу до ресурсу: <https://a4.com.ua/programa-1spidpriemstvo-perevagi-ta-nedoliki/>.
5. Семенюк Р. А. Microsoft Dynamics NAV [Електронний ресурс] – Режим доступу до ресурсу: <http://eprints.zu.edu.ua/28193/1/35.pdf>.
6. Створення запитів з використанням мови SQL – Інформатика для економістів – Підручники для студентів онлайн [Електронний ресурс] – Режим доступу до ресурсу: [https://stud.com.ua/53372/informatika/stvorenniya\\_zapitiv\\_vikoristannyam\\_movi](https://stud.com.ua/53372/informatika/stvorenniya_zapitiv_vikoristannyam_movi).
7. Що таке Діаграма Use Case і як вона застосовується [Електронний ресурс] – Режим доступу до ресурсу: <https://zdorov.zapisi.cx.ua/ukraincyam/shho-take-diagrama-use-case-i-yak-vona-zastosovuietsya.html>.
8. Діаграми класів – UA5.org [Електронний ресурс] – Режим доступу до ресурсу: <https://ua5.org/oop/392-diagrami-klasiv.html>.
9. Діаграма послідовності (Sequence Diagrams) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.maxzosim.com/sequence-diagrams/>.
10. Основні SQL-команди і запити з прикладами [Електронний ресурс] – Режим доступу до ресурсу: <https://proglib.io/p/osnovnye-sql-komandy-i-zaprosy-s-primerami-kotorye-dolzhen-znat-kazhdy-razrabotchik-2022-04-14>.

11. Ручне тестування – що це таке, типи, процеси, підходи, інструменти та інше! [Електронний ресурс] – Режим доступу до ресурсу: <https://www.zaptest.com/uk/%D1%80%D1%83%D1%87%D0%BD%D0%B5-%D1%82%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D1%89%D0%BE-%D1%86%D0%B5-%D1%82%D0%B0%D0%BA%D0%B5-%D1%82%D0%B8%D0%BF%D0%B8-%D0%BF%D1%80%D0%BE%D1%86>.

12. Що таке функціональне тестування? Типи та приклади [Електронний ресурс] – Режим доступу до ресурсу: <https://www.guru99.com/uk/functional-testing.html>.

13. Що таке модульне тестування? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.guru99.com/uk/unit-testing-guide.html>.

14. Потенційна безпека життєдіяльності людини. Основні поняття [Електронний ресурс] – Режим доступу до ресурсу: <https://dl.tntu.edu.ua/content.php?cid=299141>.

15. Про загальнообов'язкове державне соціальне страхування: Закон України від 23.09.1999, № 1105-XIV [Електронний ресурс] – Режим доступу: <http://zakon.rada.gov.ua/laws/show/1105-14>.

16. Про затвердження Порядку проведення медичних оглядів працівників певних категорій: Наказ Міністерства охорони здоров'я України від 21.05.2007, № 246 [Електронний ресурс] – Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0846-07#Text>.

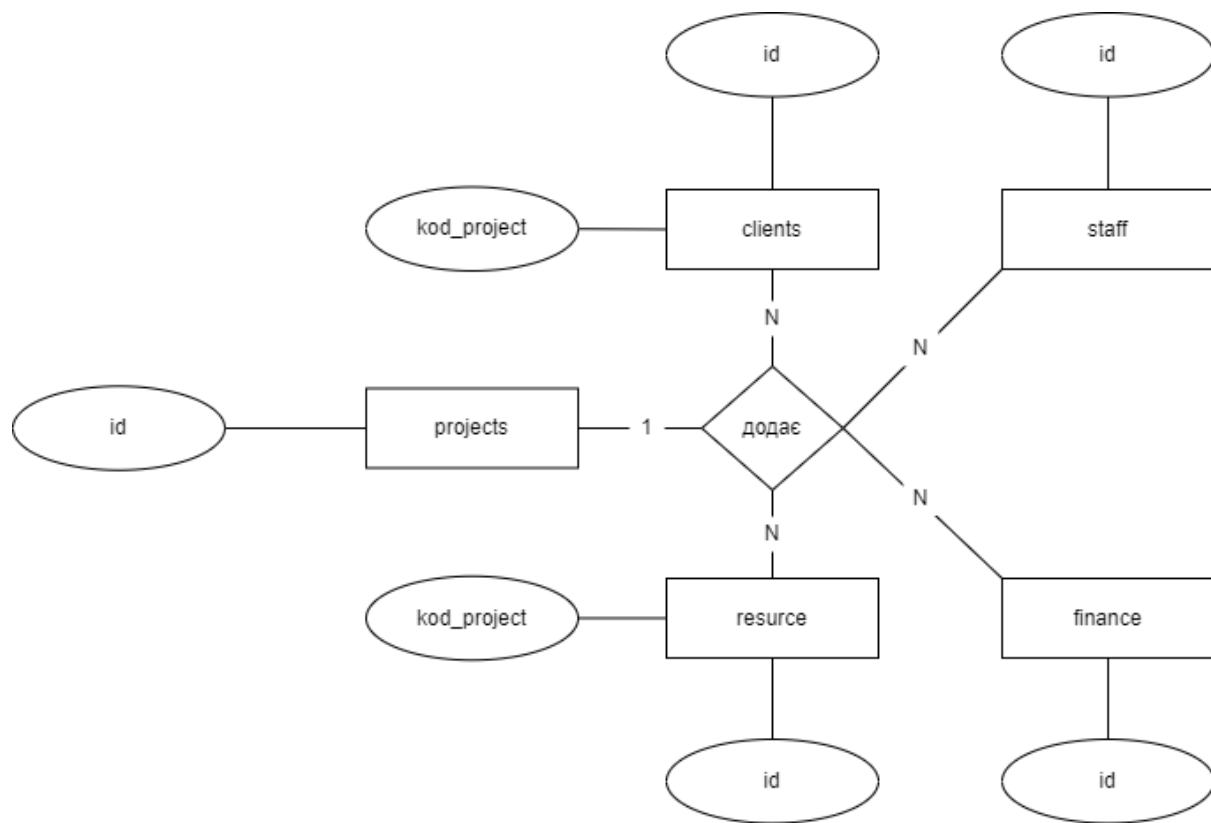
17. «Безпека життєдіяльності»: навч. посібник / Стиценко Т.Є., Пронюк Г.В., Сердюк Н.М., Хондак І.І. – Харків: ХНУРЕ, 2018. – С. 72–73.

18. Про охорону праці: Закон України від 14.10.1992, № 2694-XII [Електронний ресурс] – Режим доступу: <http://zakon.rada.gov.ua/laws/show/2694-12>.

## ДОДАТКИ

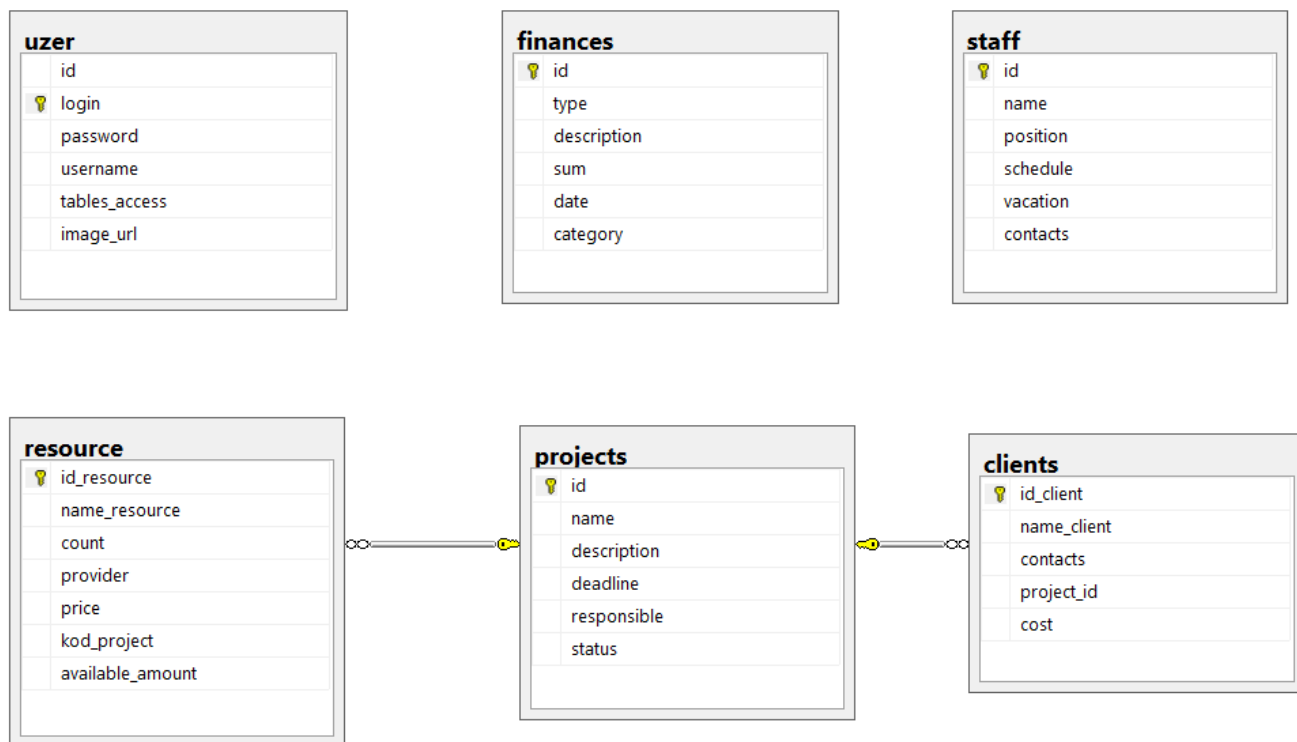
ДОДАТОК А

Діаграма «сутність-зв'язок»



## ДОДАТОК Б

## Діаграма зв'язків між таблицями бази даних



## ДОДАТОК В

## Код форми головного вікна програми

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace BlagoustriyManagement
{
    public partial class MainForm : Form
    {
        public ProjectsForm projectsForm;
        public ResourcesForm resourcesForm;
        public FinancesForm financesForm;
        public StaffForm staffForm;
        public ClientsForm clientsForm;
        public NotificationForm notificationForm;
        public MainForm()
        {
            InitializeComponent();
        }
        public void loadform(object Form)
        {
            if (this.panelMain.Controls.Count > 0)
            {
                this.panelMain.Controls.RemoveAt(0);
            }
            Form f = Form as Form;
            f.TopLevel = false;
            f.Dock = DockStyle.Fill;
            this.panelMain.Controls.Add(f);
            this.panelMain.Tag = f;
            f.Show();
        }
        public bool checkPermission(string role, string table)
        {
            LoginForm fl = new LoginForm();
            fl.connection.Open();
            SqlCommand table_accessSql = new SqlCommand("Select
tables_access From uzer Where login='"+role+"'", fl.connection);
            string table_access =
table_accessSql.ExecuteScalar().ToString();
            fl.connection.Close();
            if(table_access.Contains(table))
                return true;
        }
    }
}

```

```

        else
            return false;
    }
    private void MainForm_Load(object sender, EventArgs e)
    {
        timer1.Tick += new EventHandler(timer1_Tick);
        timer1.Start();
        panelActive.Top = buttonProjects.Top;
        LoginForm fl = new LoginForm();
        fl.connection.Open();
        SqlCommand image_urlSql = new SqlCommand("Select
image_url From user Where login='" + label2.Text + "'",
fl.connection);
        if (image_urlSql.ExecuteScalar().ToString() != "")
            pictureBox1.ImageLocation =
image_urlSql.ExecuteScalar().ToString();
        else pictureBox1.ImageLocation =
@"..\..\Resources\icons8-user-100-white.png";
        fl.connection.Close();
        if (checkPermission(label2.Text, "projects"))
        {
            panelActive.Top = buttonProjects.Top;
            projectsForm = new ProjectsForm();
            loadform(projectsForm);
        }
        else if (checkPermission(label2.Text, "resources"))
        {
            panelActive.Top = buttonResources.Top;
            resourcesForm = new ResourcesForm();
            loadform(resourcesForm);
        }
        else if (checkPermission(label2.Text, "finances"))
        {
            panelActive.Top = buttonFinances.Top;
            financesForm = new FinancesForm();
            loadform(financesForm);
        }
        else if (checkPermission(label2.Text, "staff"))
        {
            panelActive.Top = buttonStaff.Top;
            staffForm = new StaffForm();
            loadform(staffForm);
        }
        else if (checkPermission(label2.Text, "clients"))
        {
            panelActive.Top = buttonClients.Top;
            clientsForm = new ClientsForm();
            loadform(clientsForm);
        }
        else loadform(new NoPermissionForm());
        if (label2.Text == "admin")
            buttonAccounts.Text = "Акаунти";
        else
    }

```



```

        buttonAccounts.Text = "Аккаунт";
    }
    private void buttonProjects_Click(object sender, EventArgs
e)
    {
        panelActive.Top = buttonProjects.Top;
        if (checkPermission(label2.Text, "projects"))
        {
            projectsForm = new ProjectsForm();
            loadform(projectsForm);
        }
        else
            loadform(new NoPermissionForm());
    }
    private void buttonResources_Click(object sender, EventArgs
e)
    {
        panelActive.Top = buttonResources.Top;
        if (checkPermission(label2.Text, "resources"))
        {
            resourcesForm = new ResourcesForm();
            loadform(resourcesForm);
        }
        else
            loadform(new NoPermissionForm());
    }
    private void buttonClose_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
    private void buttonFinances_Click(object sender, EventArgs
e)
    {
        panelActive.Top = buttonFinances.Top;
        if (checkPermission(label2.Text, "finances"))
        {
            financesForm = new FinancesForm();
            loadform(financesForm);
        }
        else loadform(new NoPermissionForm());
    }
    private void buttonStaff_Click(object sender, EventArgs e)
    {
        panelActive.Top = buttonStaff.Top;
        if (checkPermission(label2.Text, "staff"))
        {
            staffForm = new StaffForm();
            loadform(staffForm);
        }
        else loadform(new NoPermissionForm());
    }
    private void buttonClients_Click(object sender, EventArgs e)

```

```

{
    panelActive.Top = buttonClients.Top;
    if (checkPermission(label2.Text, "clients"))
    {
        clientsForm = new ClientsForm();
        loadform(clientsForm);
    } else loadform(new NoPermissionForm());
}
private void buttonEvents_Click(object sender, EventArgs e)
{
    panelActive.Top = buttonEvents.Top;
    notificationForm = new NotificationForm(this);
    loadform(notificationForm);
}
private void buttonAbout_Click(object sender, EventArgs e)
{
    panelActive.Top = buttonAbout.Top;
    loadform(new AboutForm());
}
private void buttonHelp_Click(object sender, EventArgs e)
{ Help.ShowHelp(this, "Dovidka.chm"); }
private void buttonAccounts_Click(object sender, EventArgs
e)
{
    panelActive.Top = buttonAccounts.Top;
    if (label2.Text == "admin")
    { loadform(new AccountsForm(this)); }
    else { loadform(new ChangePassForm(this)); }
}
private void buttonLogOut_Click(object sender, EventArgs e)
{
    LoginForm lg = new LoginForm();
    this.Close();
    lg.Show();
}
private void timer1_Tick(object sender, EventArgs e)
{
    int h = DateTime.Now.Hour;
    int m = DateTime.Now.Minute;
    string time = "";
    if (h < 10)
    { time += "0" + h; }
    else
    { time += h; }
    time += ":";
    if (m < 10) { time += "0" + m; }
    else { time += m; }
    labelTime.Text = time;
    labelDate.Text = DateTime.Now.ToString("dd MMMM yyyy",
new System.Globalization.CultureInfo("uk-UA"));
}
// Загальний метод для пошуку в DataGridView

```

```

        private void SearchInDataGridView(DataGridView dataGridView,
string searchText)
        {
            if (search == 0)
            {
                search++;
                buttonSearch.BackColor = Color.FromArgb(0, 123,
255);

                for (int i = 0; i < dataGridView.RowCount; i++)
                {
                    bool rowVisible = false;
                    for (int j = 0; j < dataGridView.ColumnCount;
j++)
                    {
                        if (dataGridView.Rows[i].Cells[j].Value !=
null &&
dataGridView.Rows[i].Cells[j].Value.ToString().ToLower().Contains(se
archText.ToLower()))
                        {
                            rowVisible = true;
                            break;
                        }
                    }
                    dataGridView.Rows[i].Visible = rowVisible;
                }
            }
            else
            {
                search--;
                buttonSearch.BackColor = Color.Transparent;
                textBox1.Text = "";
                for (int i = 0; i < dataGridView.RowCount; i++)
                { dataGridView.Rows[i].Visible = true; }
            }
        }
// Отримання активного DataGridView
private DataGridView GetActiveDataGridView()
{
    if (panelActive.Top == buttonProjects.Top)
    { return projectsForm.dataGridView1; }
    else if (panelActive.Top == buttonResources.Top)
    { return resourcesForm.dataGridView1; }
    else if (panelActive.Top == buttonFinances.Top)
    { return financesForm.dataGridView1; }
    else if (panelActive.Top == buttonStaff.Top)
    { return staffForm.dataGridView1; }
    else if (panelActive.Top == buttonClients.Top)
    { return clientsForm.dataGridView1; }
    else if (panelActive.Top == buttonEvents.Top)
    { return notificationForm.dataGridView1; }
    else
    { return null; }
}

```

```

    }
    int search = 0;
    private void buttonSearch_Click(object sender, EventArgs e)
    {
        DataGridView activeDataGridView =
GetActiveDataGridView();
        if (activeDataGridView != null)
        {
            SearchInDataGridView(activeDataGridView,
textBox1.Text);
        }
    }
    int TogMove;
    int MValX;
    int MValY;
    private void panelHeader_MouseDown(object sender,
MouseEventArgs e)
    {
        TogMove = 1;
        MValX = e.X;
        MValY = e.Y;
    }
    private void panelHeader_MouseUp(object sender,
MouseEventArgs e)
    {
        TogMove = 0;
    }
    private void panelHeader_MouseMove(object sender,
MouseEventArgs e)
    {
        if (TogMove == 1)
        {
            this.SetDesktopLocation(MousePosition.X - MValX,
MousePosition.Y - MValY);
        }
    }
}
}}
```

## ДОДАТОК Д

## Код вкладки «Проекти»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Word = Microsoft.Office.Interop.Word;
namespace BlagoustriyManagement
{
    public partial class ProjectsForm : Form
    {
        public ProjectsForm()
        {
            InitializeComponent();
        }
        public void Tabl_Refresh()
        {
            dataGridView1.Rows.Clear();
            LoginForm fl = new LoginForm();
            fl.connection.Open();
            SqlCommand command = new SqlCommand("Select id, name,
description, deadline, responsible, status From projects",
fl.connection);
            SqlDataReader reader = command.ExecuteReader();
            if (reader.HasRows)
            {
                while (reader.Read())
                {
                    dataGridView1.Rows.Add(reader["id"],
reader["name"], reader["description"],
((DateTime)reader["deadline"]).ToShortDateString(),
reader["responsible"], reader["status"]);
                }
            }
            reader.Close();
            fl.connection.Close();
        }
        public void Save_Table()
        {
            try

```

```

        {
            LoginForm fl = new LoginForm();
            fl.connection.Open();
            SqlCommand command = new SqlCommand();
            command.Connection = fl.connection;
            int kol =
dataGridView1.Rows.GetRowCount(DataGridViewElementStates.Visible);
            for (int i = 0; i < kol; i++)
            {
                int id = Convert.ToInt32(dataGridView1[0,
i].Value);
                string name = Convert.ToString(dataGridView1[1,
i].Value);
                string description =
Convert.ToString(dataGridView1[2, i].Value);
                DateTime deadline =
Convert.ToDateTime(dataGridView1[3, i].Value);
                string responsible =
Convert.ToString(dataGridView1[4, i].Value);
                string status =
Convert.ToString(dataGridView1[5, i].Value);
                command.CommandText = "UPDATE projects SET name
= '" + name + "', description = '" + description + "', deadline = '"
+ deadline.ToString("yyyyMMdd") + "', responsible = '" + responsible
+ "', status = '" + status + "' WHERE id=" + id;
                SqlDataReader reader = command.ExecuteReader();
                reader.Close();
            } fl.connection.Close();
        }
        catch (Exception ex) {
            MessageBox.Show("При збереженні відбулася помилка: "
+ ex.Message, "Помилка збереження", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }
    private void button3_Click(object sender, EventArgs e)
    {
        LoginForm fl = new LoginForm();
        fl.connection.Open();
        AddProjectForm form_addproject = new AddProjectForm();
        incorrectValues:
        form_addproject.ShowDialog();
        SqlCommand count = new SqlCommand("Select COUNT(id) From
projects", fl.connection);
        int nomer =
Convert.ToInt32(count.ExecuteScalar().ToString());
        if (form_addproject.DialogResult == DialogResult.OK)
        {
            try
            {
                nomer++;
                SqlCommand command =
fl.connection.CreateCommand();

```

```

        command.CommandText = "INSERT INTO projects
VALUES(" + nomer + ", '" + form_addproject.textBox1.Text + "', '" +
        form_addproject.textBox2.Text + "', '" +
form_addproject.dateTimePicker1.Value.ToString("yyyyMMdd") + "', '"
+ form_addproject.textBox3.Text + "', '" +
form_addproject.comboBox1.Text + "')";
        command.ExecuteNonQuery();
    }
    catch (Exception ex) {
        MessageBox.Show("При добавлені нового запису
відбулася помилка: " + ex.Message, "Помилка добавлення",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        goto incorrectValues;
    }
    form_addproject.Close();
    this.Show();
    fl.connection.Close();
}
else
{
    form_addproject.Close();
    this.Show();
    fl.connection.Close();
}
}
private void button1_Click(object sender, EventArgs e)
{
    LoginForm fl = new LoginForm();
    fl.connection.Open();
    AddProjectForm form_addproject = new AddProjectForm();
    form_addproject.button1.Text = "Редагувати";
    form_addproject.Text = "Редагувати проект";
    form_addproject.textBox1.Text =
dataGridView1.CurrentRow.Cells[1].Value.ToString();
    form_addproject.textBox2.Text =
dataGridView1.CurrentRow.Cells[2].Value.ToString();
    form_addproject.dateTimePicker1.Value =
Convert.ToDateTime(dataGridView1.CurrentRow.Cells[3].Value);
    form_addproject.textBox3.Text =
dataGridView1.CurrentRow.Cells[4].Value.ToString();
    form_addproject.comboBox1.Text =
dataGridView1.CurrentRow.Cells[5].Value.ToString();
    incorrectValueEdit:
    form_addproject.ShowDialog();
    if (form_addproject.DialogResult == DialogResult.OK)
    { try
        {
            SqlCommand command =
fl.connection.CreateCommand();
            command.CommandText = "UPDATE projects SET
name='" + form_addproject.textBox1.Text + "', description='" +

```

```

        form_addproject.textBox2.Text + "'",
deadline='" +
form_addproject.dateTimePicker1.Value.ToString("yyyyMMdd") + "',
responsible='" +
        form_addproject.textBox3.Text + "'",
status='" + form_addproject.comboBox1.Text + "' Where id=" +
dataGridView1.CurrentRow.Cells[0].Value;
        command.ExecuteNonQuery();
    }
    catch (Exception ex) {
        MessageBox.Show("При редагуванні відбулася
помилка: " + ex.Message, "Помилка редагування",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        goto incorrectValueEdit;
    }
    form_addproject.Close();
    this.Show();
    fl.connection.Close();
}
else
{
    form_addproject.Close();
    this.Show();
    fl.connection.Close();
}
}
private void button4_Click(object sender, EventArgs e)
{
    Save_Table();
    int tmp =
Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value);
    LoginForm fl = new LoginForm();
    fl.connection.Open();
    DialogResult result = MessageBox.Show("Видалити вибраний
запис? ВАЖЛИВО: Разом із цим записом будуть видалені всі інші
пов'язані записи!!!", "Попередження", MessageBoxButtons.OKCancel);
    if (result == DialogResult.OK)
    {
        int del_id =
Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value);
        SqlCommand command = fl.connection.CreateCommand();
        command.CommandText = "DELETE FROM projects Where
id=" + dataGridView1.CurrentRow.Cells[0].Value;
        command.ExecuteNonQuery();
        command.CommandText = "SELECT COUNT(*) FROM
projects";
        object counts = command.ExecuteScalar();
        int count = Convert.ToInt32(counts);
        for (; del_id < (count + 1); del_id++)
        {
            command.CommandText = "UPDATE projects SET id="
+ del_id + " Where id=" + (del_id + 1);

```



```

        command.ExecuteNonQuery();
    }
    // Видалення пов'язаних записів з ресурсів (По
    одному щоб змінити id_resource)
    command.CommandText = "SELECT COUNT(*) FROM resource
Where kod_project=" + dataGridView1.CurrentRow.Cells[0].Value;
    int del_count =
    Convert.ToInt32(command.ExecuteScalar());
    for (int i = 1; i < (del_count + 1); i++)
    {
        command.CommandText = "Select TOP 1 id_resource
From resource Where kod_project=" +
dataGridView1.CurrentRow.Cells[0].Value;
        int del_res_id =
        Convert.ToInt32(command.ExecuteScalar());
        command.CommandText = "DELETE FROM resource
Where id_resource=" + del_res_id;
        command.ExecuteNonQuery();
        command.CommandText = "SELECT COUNT(*) FROM
resource";
        int count_res =
        Convert.ToInt32(command.ExecuteScalar());
        for (; del_res_id < (count_res + 1);
del_res_id++)
        {
            command.CommandText = "UPDATE resource SET
id_resource=" + del_res_id + " Where id_resource=" + (del_res_id +
1);
            command.ExecuteNonQuery();
        }
    }
    // Видалення пов'язаних записів з замовників (По
    одному щоб змінити id_resource)
    command.CommandText = "SELECT COUNT(*) FROM clients
Where project_id=" + dataGridView1.CurrentRow.Cells[0].Value;
    del_count=Convert.ToInt32(command.ExecuteScalar());
    for (int i = 1; i < (del_count + 1); i++)
    {
        command.CommandText = "Select TOP 1 id_client
From clients Where project_id=" +
dataGridView1.CurrentRow.Cells[0].Value;
        int del_client_id =
        Convert.ToInt32(command.ExecuteScalar());
        command.CommandText = "DELETE FROM clients Where
id_client=" + del_client_id;
        command.ExecuteNonQuery();
        command.CommandText = "SELECT COUNT(*) FROM
clients";
        int count_clients =
        Convert.ToInt32(command.ExecuteScalar());
        for (; del_client_id < (count_clients + 1);
del_client_id++)

```

```

        {
            command.CommandText = "UPDATE clients SET
id_client=" + del_client_id + " Where id_client=" + (del_client_id +
1);

            command.ExecuteNonQuery();

        }
    }
    MessageBox.Show("Запис видалено", "Повідомлення",
MessageBoxButtons.OK, MessageBoxIcon.Information);
}
fl.connection.Close();
}
private void button6_Click(object sender, EventArgs e)
{ Tabl_Refresh(); }
private void button7_Click(object sender, EventArgs e)
{
    Microsoft.Office.Interop.Excel.Application ExcelApp =
new Microsoft.Office.Interop.Excel.Application();
    ExcelApp.SheetsInNewWorkbook = 1;
    ExcelApp.Application.Workbooks.Add(Type.Missing);
    ExcelApp.Application.Workbooks[1].Worksheets[1].Name =
"Проекти";
    ExcelApp.Columns.ColumnWidth = 20;
    for (int i = 1; i < dataGridView1.ColumnCount + 1; i++)
    {
        ExcelApp.Cells[1, i] = dataGridView1.Columns[i -
1].HeaderText;
    }
    for (int i = 0; i < dataGridView1.ColumnCount; i++)
    {
        for (int j = 0; j < dataGridView1.RowCount; j++)
        {
            ExcelApp.Cells[j + 2, i + 1] = dataGridView1[i,
j].Value;
        }
    }
    ExcelApp.Visible = true;
}
private void button8_Click(object sender, EventArgs e)
{
    Word._Application application = new Word.Application();
    Word._Document document = new Word.Document();
    Object missingObj = System.Reflection.Missing.Value;
    Object trueObj = true;
    Object falseObj = false;
    Object templatePathObj =
System.IO.Path.GetFullPath(@"file1.docx");
    document = application.Documents.Add(ref
templatePathObj, ref missingObj, ref missingObj, ref missingObj);
    Object bookmarkNameObj = "a";
    Word.Range bookmarkRange = null;

```

```

        bookmarkRange = document.Bookmarks.get_Item(ref
bookmarkNameObj).Range;
        Object behavior =
Word.WdDefaultTableBehavior.wdWord9TableBehavior;
        Object autoFitBehavior =
Word.WdAutoFitBehavior.wdAutoFitFixed;
        int rows = dataGridView1.RowCount + 1;
        int columns = dataGridView1.ColumnCount;
        document.Tables.Add(bookmarkRange, rows, columns, ref
missingObj, ref missingObj);
        for (int i = 1; i < dataGridView1.ColumnCount + 1; i++)
        {
            document.Tables[1].Cell(1, i).Range.Text =
dataGridView1.Columns[i - 1].HeaderText;
        }
        for (int i = 0; i < dataGridView1.ColumnCount; i++)
        {
            for (int j = 0; j < dataGridView1.RowCount; j++)
            {
                document.Tables[1].Cell(j + 2, i + 1).Range.Text
= Convert.ToString(dataGridView1[i, j].Value);
            }
        }
        application.Visible = true;
    }
    private void ProjectsForm_Load(object sender, EventArgs e)
    {
        Tabl_Refresh();
    }
}
}

```

## ДОДАТОК Е

Диск із кваліфікаційною роботою бакалавра