

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІМ. Ю.М. ПОТЕБНІ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кваліфікаційна робота

другий (магістерський)

(рівень вищої освіти)

на тему **Єдина інформаційна система керування підприємством**
України

Виконав: студент 2 курсу, групи 8.1212-іпз-2
спеціальності 121 Інженерія програмного
забезпечення

(код і назва спеціальності)

освітньої програми Інженерія програмного
забезпечення

(код і назва освітньої програми)

С.В. Меліхов

(ініціали та прізвище)

Керівник доцент, к.т.н., О.М.Міхайлуца
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Рецензент директор ТОВ «Дісітел»

П.О. Лютий

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

Запоріжжя
2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІМ. Ю.М. ПОТЕБНІ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

Кафедра Електроніки, інформаційних систем та програмного забезпечення

Рівень вищої освіти другий (магістерський)

Спеціальність 121 Інженерія програмного забезпечення
(код та назва)

Освітня програма Інженерія програмного забезпечення
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри Т.В. Критська
“ 01 ” вересня 2023 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Меліхову Євгенію Валентиновичу

(прізвище, ім'я, по батькові)

1. Тема роботи Єдина інформаційна система керування підприємством України

керівник роботи Міхайлуца Олена Миколаївна, доцент, к.т.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від 09.10.2023 р. №1577-с

2. Строк подання студентом кваліфікаційної роботи 1 грудня 2023 р.

3. Вихідні дані магістерської роботи

- комплект нормативних документів ;
- технічне завдання до роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- огляд та збір літератури стосовно теми кваліфікаційної роботи;
- огляд та аналіз існуючих рішень та аналогів;
- дослідження технологій, методів та засоби проектування веб-сайтів інтернет магазинів;
- створення програмного продукту та його опис;
- перелік вимог для роботи програми;
- дослідження поставленої проблеми та розробка висновків та пропозицій.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
15 слайдів презентації

6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		Завдання прийняв

7. Дата видачі завдання 01.09.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1	Аналіз предметної області шляхом комунікація із підприємствами для виявлення недоліків існуючих систем	02.09-12.09.23	виконано
2	Формулювання та узгодження із керівником тематики роботи	13.09-14.09.23	виконано
3	Вивчення аналогів, існуючих на ринку програмних продуктів	15.09-21.09.23	виконано
4	Визначення вимог до майбутньої інформаційної системи	22.09-25.09.23	виконано
5	Проектування бази даних та заповнення її інформацією	26.09-29.09.23	виконано
6	Проектування інтерфейсу користувача	30.09-05.09.23	виконано
7	Створення початкового застосунку та його зв'язування із БД	06.10-10.10.23	виконано
8	Реалізація спроектованого інтерфейсу	11.10-21.10.23	виконано
9	Проектування основних алгоритмів роботи системи	22.10-27.10.23	виконано
10	Розробка спроектованих алгоритмів	28.10-10.11.23	виконано
11	Зв'язування розроблених інтерфейсів із алгоритмами	11.11-18.11.23	виконано
12	Тестування інформаційної системи та виправлення недоліків	19.11-21.11.23	виконано
13	Оформлення звіту	22.11-27.11.23	виконано

Студент _____ Є.В. Меліхов
(підпис) (ініціали та прізвище)

Керівник роботи _____ О.М. Міхайлуца
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ І.А. Скрипник
(підпис) (ініціали та прізвище)

АНОТАЦІЯ

Сторінок: 98

Рисунків: 30

Таблиць: 2

Джерел: 27

Меліхов Є. В. Єдина інформаційна система керування підприємством України : кваліфікаційна робота магістра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник О. М. Михайлуца. Запоріжжя : ЗНУ, 2023. 98 с.

Мета роботи полягає у вивченні використовуваних підприємствами програмних застосунків, їхніх альтернатив та реалізації інформаційної системи для керування підприємством, яка містить в собі весь необхідний функціонал.

Предметом дослідження є необхідність створення власного застосунку для задоволення потреб малих та середніх підприємств та зменшення їх фінансового навантаження шляхом підвищення ефективності ведення документообігу.

У процесі дослідження була розглянута проблема розробки одного застосунку замість використання великої кількості платних додатків, що здатна суттєво підвищити ефективність роботи підприємства та зменшити фінансове навантаження на нього.

Ключові слова: інформаційна система для керування підприємством, програма, програмний застосунок, C#, Windows Forms, підприємство, бухгалтерія, інтерфейс.

SUMMARY

Pages: 98

Figures: 30

Tables: 2

Source: 27

Melikhov E. V. Unified information system of management of the enterprise of Ukraine: master's thesis of specialty 121 "Software engineering" / Science head O. M. Mikhailutsa. Zaporizhzhia: ZNU, 2023. 98 p.

The purpose of the work is to study the software applications used by enterprises, their alternatives, and the implementation of an information system for managing the enterprise, which contains all the necessary functionality.

The subject of the study is the need to create an own application to meet the needs of small and medium-sized enterprises and reduce their financial burden by improving the efficiency of document management.

In the research process, the problem of developing one application instead of using a large number of paid applications was considered, which can significantly increase the efficiency of the enterprise and reduce its financial burden.

Keywords: information system for enterprise management, program, software application, C#, Windows Forms, enterprise, accounting, interface.

ЗМІСТ

ВСТУП	8
1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ВИКОРИСТАННЯ ДЕКІЛЬКОХ ЗАСТОСУНКІВ ДЛЯ ВЕДЕННЯ БУХГАЛТЕРСЬКОГО ОБЛІКУ НА ПІДПРИЄМСТВІ.....	13
1.1 Огляд проведених досліджень та наукових джерел.....	13
1.2 Огляд закордонних аналогів західного виробництва.....	16
1.3 Інформаційна система бухгалтерського обліку, яка використовується на території України	20
1.4 Інформаційна система подачі звітності до державних та контролюючих органів, яка використовується на території України ...	24
1.5 Недоліки, які криються у використанні BAS та M.E.Doc.....	25
2 ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	27
2.1 Найдоцільніший формат інформаційної системи	27
2.1.1 Переваги та недоліки розробки системи у вигляді веб- застосунку	27
2.1.2 Переваги та недоліки розробки системи у вигляді десктопного застосунку	30
2.2 Технології розробки настільних застосунків	31
2.3 Вибір архітектури програмного забезпечення.....	33
2.4 Вибір платформи користувачького інтерфейсу.....	34
2.5 Вибір бази даних для розташування серверної частини.....	35
2.6 З'єднання програмного застосунку з базою даних.....	37

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ КЕРУВАННЯ ПІДПРИЄМСТВОМ	38
3.1 Бізнес обґрунтування та вимоги інформаційної системи.....	38
3.2 Вимоги до програмного продукту.....	38
3.2.1 Функціональні вимоги до програмного продукту.....	38
3.2.2 Нефункціональні вимоги до програмного продукту.....	39
3.3 Розробка бази даних інформаційної системи	40
3.4 Розробка інформаційної системи за допомогою Visual Studio 2022 .	42
3.4.1 Проектування інформаційної системи.....	42
3.4.2 Реалізація програмної частини	45
3.4.3 Розробка інтерфейсу застосунку	70
4 ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ РОЗРОБКИ ЄДИНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ КЕРУВАННЯ ПІДПРИЄМСТВОМ	86
4.1 Підвищення надійності вносимої інформації	86
4.2 Використання правил людинно-машинної взаємодії для визначення ефективності єдиної інформаційної системи	88
4.3 Використання продуктових метрик для визначення ефективності єдиної інформаційної системи.....	89
4.4 Вартість розробки єдиної інформаційної системи	92
ВИСНОВКИ.....	95
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	96

ВСТУП

Актуальність теми

За оцінкою багатьох відомих вчених, інформаційні технології та комп'ютеризація стали найбільш значущими інноваціями ХХ століття, котрі зумовили перехід до нового технологічного укладу [20]. Вже наприкінці 80-х років минулого сторіччя у світі почався період, який велика кількість дослідників називають «цифрова революція». Він характеризується швидким розвитком виробництва персональної обчислювальної техніки та її широким поширенням у суспільстві. Раніше комп'ютери, зазвичай, стояли на виробництвах та у дослідних центрах і мали доволі вузьке коло задач. Тепер вони почали опинятися у різних підприємств ще й для ведення діловодства, зберігання архівів у цифровому форматі, а згодом — і у приватних користувачів для побутових задач. Такі розширені вимоги викликали необхідність збільшити кількість програмних продуктів на ринку, що викликало зростання нещодавно сформованої ІТ-сфери [19].

У цей час і сформувалося розуміння того, що підприємствам потрібно діджиталізувати один із базових відділів, який є майже на кожному підприємстві — бухгалтерського обліку. Важко переоцінити важливість бухгалтерії, адже навіть першим наказом, який має видати підприємство після відкриття, повинен бути наказ «Про облікову політику» [24]. Окрім цього, на фінансовій та податковій звітності, яку підприємство подає періодично (щомісячно, щоквартально, щорічно), повинні стояти підписи не лише директора підприємства, а і головного бухгалтера [17].

Цілком очікувано, що в епоху цифрової революції, загального впровадження цифровізованих застосунків та діджиталізованих рішень у всіх сферах життя людини, бухгалтери вже не можуть обходитись складанням паперових первинних документів та звітності. Саме через це, на допомогу фахівцям з бухгалтерського обліку, і була розпочата розробка спеціалізованого програмного забезпечення для ведення обліку та складання звітності підприємства [17].

Окрім цього, певні фактори впливають на те, що використання інформаційної системи не тільки пришвидшує та спрощує оцінку стану підприємства, а і підвищує ступінь захисту інформації [17].

По-перше, ведення електронного документообігу майже повністю унеможливує створення та використання «чорної» бухгалтерії або підробки звітів для приховування розкрадань на підприємстві, адже завжди можна перевірити, які дії зі звітністю були проведені та на що вони були направлені [17].

По-друге, завдяки подібним інформаційним системам, відсутня необхідність власноруч звіряти безліч проведених операцій для отримання інформації про наявність запасів. Достатньо викликати відповідне вікно програмного застосунку, яке відразу надасть бажану інформацію за певний період часу, що значно підвищує швидкість інвентаризацій на підприємствах [17].

Саме завдяки переліченим вище факторам, майже із самого початку цифрової революції, і було вирішено почати повсюдну діджиталізацію бухгалтерського обліку підприємств. Успішна діяльність тієї чи іншої організації багато в чому залежить від того, чи є унікальною система, що використовується, і за допомогою якої здійснюється діяльність бізнесу та контроль над ним. Використання стандартних програм зводить нанівець унікальність будь-якої з технологій, а індивідуальний підхід дозволяє охопити всі сфери бізнесу та сформулювати комплекс робіт по реалізації поставлених завдань. Тобто створення програмного забезпечення, яке повністю розробляється під вимоги підприємств та організацій, є досить перспективним напрямком, оскільки забезпечує найбільш оптимальний підхід для їх функціонування в умовах сучасної економіки. І, як казав Бенджамін Франклін: «Час — це гроші!» [20].

Таким чином, актуальним є створення інформаційної системи для керування діяльністю підприємства замість використання на фірмі декількох застосунків, які вирішують суто обмежене коло завдань.

Мета роботи

Мета роботи полягає у вивченні використовуваних підприємствами програмних застосунків, їхніх альтернатив та реалізації інформаційної системи для керування підприємством, яка містить в собі весь необхідний функціонал.

Об'єкт дослідження

Об'єктом дослідження є процес ведення електронного діловодства на підприємстві.

Предмет дослідження

Предметом дослідження є необхідність створення власного застосунку для задоволення потреб малих та середніх підприємств та зменшення їх фінансового навантаження шляхом підвищення ефективності ведення документообігу.

Методи дослідження

Для вирішення поставленої задачі використовуються наступні методи дослідження:

1. Аналіз особливостей та існуючих рішень (аналогів) для ведення бухгалтерії та електронного документообігу на підприємствах.
2. Аналіз технологій для реалізації інформаційної системи для керування підприємством.

Наукова новизна одержаних результатів

Наукова новизна одержаних результатів дослідження полягає у тому, що для вирішення задачі доцільності створення єдиної інформаційної системи керування підприємством замість використання декількох платних програм використовуються формули підрахунку швидкості програмного застосунку та ефективності заміни декількох існуючих на один загальний.

Практичне значення одержаних результатів

Практичне значення одержаних результатів дослідження полягає у тому, що розробка одного застосунку замість використання великої кількості платних додатків підвищує ефективність роботи підприємства та зменшує фінансове навантаження на нього.

Апробація результатів

Результати роботи, викладені у кваліфікаційній роботі магістра, були опубліковані в матеріалах Міжнародної науково-практичної конференції «Перспективи сталого розвитку в умовах глобалізації в економічному, управлінському та інженерному аспектах» .

Глосарій

C# — це об'єктно-орієнтована мова програмування, яка використовується для розробки програмних застосунків на платформі Microsoft .NET.

DataSet — це клас, який належить простору імен System.Data в ADO.NET і призначений для представлення та маніпулювання набором даних в пам'яті. Він діє як контейнер, який містить одну або кілька таблиць з відношеннями між ними. Цей клас надає зручні можливості для зберігання, модифікації та обробки даних. Він має вбудовану структуру, яка здатна зберігати схему та реальні дані таблиць, що дозволяє виконувати різноманітні операції з ними. *DataSet* дозволяє фільтрувати, сортувати, злити, розділити та змінювати структуру даних, що знаходяться в ньому.

Windows Forms — це набір інструментів для розробки графічних інтерфейсів користувача для програм, які працюють в операційній системі Windows.

Бухгалтерія — це підрозділ підприємства, який відповідає за фінансовий облік та звітність.

Інтерфейс — це зовнішній вигляд програми або системи, що надає користувачеві можливість взаємодіяти з ними. Інтерфейс може містити елементи керування, які дозволяють користувачу взаємодіяти з програмою, такі як кнопки, поля введення, меню тощо.

Інформаційна система для керування підприємством — це комп'ютерна система, призначена для автоматизації різних процесів у підприємстві, таких як фінансове планування, управління персоналом, контроль запасів тощо.

Підприємство — це юридична особа, яка займається виробництвом або наданням послуг з метою отримання прибутку.

Програмний застосунок — це програма, яка створена для вирішення конкретної задачі або набору задач на комп'ютері.

1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ВИКОРИСТАННЯ ДЕКІЛЬКОХ ЗАСТОСУНКІВ ДЛЯ ВЕДЕННЯ БУХГАЛТЕРСЬКОГО ОБЛІКУ НА ПІДПРИЄМСТВІ

1.1 Огляд проведених досліджень та наукових джерел

Взагалі, ідея об'єднання кількох програмних продуктів, які використовуються на підприємстві, задля підвищення ефективності і створення єдиного інформаційного простору, існує доволі довгий час. Всі бачили від цього лише користь:

- уникнення можливих помилок при перенесенні даних з однієї програми до іншої;
- зменшення фінансового навантаження на підприємство, адже за кожен ліцензійний застосунок необхідно платити гроші;
- можливість легкого впровадження додаткового функціоналу, який пов'язаний з діяльністю фірми;
- тощо.

Але для того, щоб можна було взагалі працювати із документами у програмі, їх необхідно цифровізувати. Згодом цей процес отримав назву діджиталізація.

Автор [27], досліджуючи методологічні підходи до трактування цифрової економіки у сучасному розумінні, дав цьому терміну наступну характеристику: «Діджиталізацію можна визначити як сучасний інноваційний етап економічного розвитку, в основі якого знаходиться інтеграція фізичних та цифрових ресурсів у сфері виробництва та споживання, в економіці та суспільстві.». Як зазначають науковці у статті [13], за результатами дослідження Siemens Business Services було з'ясовано, що в окремих компаніях 30% часу робочих груп витрачається на пошуки та погодження документів; 6% документів безповоротно губляться; кожен внутрішній документ копіюється до 20 разів; на

20-25% зростає продуктивність праці персоналу за використання електронного документообігу; вартість архівного збереження електронних документів на 80% нижча порівняно з їхніми паперовими копіями.

Тобто, дивлячись на статистичні дані, можна зазначити, що діджиталізація діловодства дозволила б значно підвищити ефективність підприємств. Але, на жаль, сучасну інтегрованість цифрових рішень у галузях народного господарства України не можна назвати надто високою. Згідно проведеного дослідження [22], станом на 2021 рік:

- більше 60% підприємств використовує діджитал-технології, але лише у разі вимог з боку клієнтів;
- менше 20% знаходяться на початковому етапі впровадження цифрових рішень;
- 15% компаній використовують переваги діджиталізації у значних обсягах;
- лише 3-4% впровадили діджитал-технології у всіх сферах своєї діяльності.

Але, як зазначають науковці у статті [11], необхідною умовою для діджиталізації народного господарства є створення попиту на цифрові технології на всіх рівнях: від громадян та бізнеса до цілих галузей економіки та державних секторів. Саме через це показники цифрової трансформації компаній залишаються на доволі низькому рівні. Проте вже зараз держава розпочала дії щодо популяризації сучасних ІТ-рішень. Як зазначають у статті [9], керівництво країни розпочало цілеспрямовану політику, спрямовану на інноваційний розвиток бізнесу, використання новітніх інформаційних технологій у господарській діяльності, прискорення та підвищення якості взаємодії у ринковому середовищі. Отже, підприємствам саме час розпочинати діджиталізацію своїх відділів.

Велику роль на будь-якому підприємстві займає саме бухгалтерія. Отже, це один з перших підрозділів, який повинна буде торкнутися діджиталізація.

До того ж, велика кількість економістів погоджуються з тим, що майбутнє бухгалтерії саме за цифровими носіями. Наприклад, автори статті [16] зазначають, що впровадження інформаційної системи для ведення обліку, дозволяє впорядкувати його, зменшити кількість помилок, підвищити ефективність відділу і його економічність та збільшити обсяг інформації, яку можна отримати у короткий термін». Окрім цього, науковці у статті [15] зазначають, що побудову інформаційної системи управління підприємством можна розпочинати лише при наявності певних умов та компонентів, а саме: програмного забезпечення з необхідним ступенем функціональності, гнучких схем бізнес-логіки і кваліфікації консультантів, достатньої для побудови системи. Але процес діджиталізації є безперервним і, як зазначають автори у статті [26], повинен включати наступні етапи:

- пошук нових ідей, технологій, рішень;
- апробація технологій в діяльності підприємства на невеликій ділянці;
- застосування технології в масштабах всього підприємства;
- оцінка окупності від застосування інноваційної технології.

Але якщо впроваджувати сучасні цифрові рішення у кожній галузі підприємства окремо, то це призведе до наявності значної кількості застосунків, недоліки яких зазначені вище. Саме тому існує необхідність створення саме єдиної інформаційної системи для керування підприємством. Як зазначають автори [25], подібні застосунки повинні:

- здійснювати збір, систематизацію та обробку інформації;
- проводити пошук, аналіз і оцінку джерел інформації для проведення економічних розрахунків;
- використовувати сучасні прийоми і способи для вирішення економічних завдань;
- використовувати сучасні наукові методологічні та методичні розробки економістів при проведенні аналітичних заходів;
- розробляти напрямки з мобілізації невикористаних резервів, впровадження інноваційних технологій.

Окрім цього, згідно дослідження [10], всі інформаційні системи цього напрямку повинні розроблятися за допомогою концепції CALS (Continuous acquisition and life cycle support). Вона ставить певні вимоги до програмних застосунків, а саме: дотримання принципів ООП, повна інтеграція підсистем між собою, можливість існування системи у автономному режимі (буде доцільним при наявності на підприємстві секретної інформації або наявності комерційної таємниці), єдиний інтерфейс доступу тощо.

1.2 Огляд закордонних аналогів західного виробництва

Якщо подивитись на західні аналоги, то можна виокремити декілька застосунків, які підійдуть для ведення бухгалтерського обліку:

1. Wave — інформаційна система, яка підійде для ведення бухгалтерського обліку відносно невеликих та середніх підприємств [3].

Серед функціональних можливостей програми можна зазначити:

- керування бухгалтерськими проведеннями;
- керування грошовими потоками підприємства;
- нарахування заробітної плати;
- здійснення закупівель [1].

Інтерфейс програмного застосунку наведено на відповідному рисунку (див. Рис. 1.1).

Він повністю безкоштовний, проте має дуже обмежений функціонал (наприклад, відсутня можливість керування запасами та отримання інформації про їхній облік). Отже, цей застосунок призначений для внутрішнього керування підприємством, але не має для цього достатнього функціоналу, що створює необхідність використовувати додаткові інформаційні системи.

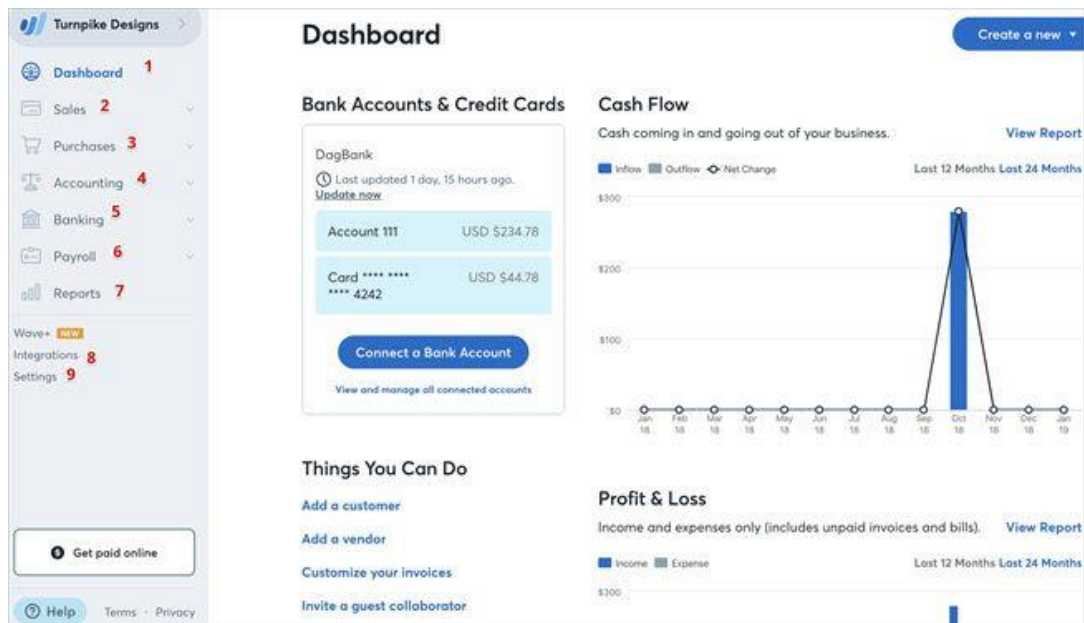


Рисунок 1.1 — Інтерфейс програмного застосунку «Wave» [1]

2. FreshBooks — інформаційна система, яка спеціально розроблена для незалежних підрядників та підприємств середнього розміру [3]. Інтерфейс програми наведено на відповідному рисунку (див. Рис. 1.2).



Рисунок 1.2 — Інтерфейс інформаційної системи «FreshBooks» [1]

Серед функціональних можливостей програми можна зазначити:

- керування платіжними дорученнями та бухгалтерськими рахунками;
- виконання періодичних транзакцій;

- планувальник зборів по закінченню звітного періоду;
- легке під'єднання до деяких онлайн банкінгів [1].

Цей застосунок відноситься до ліцензійного програмного забезпечення (середня підписка коштує 10 \$/місяць), що означає необхідність оформлювати платну підписку, хоча вона і не є надто високою. Водночас, він теж має деякі обмеження у функціоналі (наприклад, також відсутня можливість керування запасами та здійснювати розрахунок заробітної плати). Отже, це програмне забезпечення більш пристосоване для податкового обліку, хоч і має функції ведення бухгалтерії. Однак, цього теж недостатньо для повноцінної роботи бухгалтерського відділу.

3. Херо — один з найкращих програмних продуктів для бухгалтерського обліку на підприємствах невеликого розміру (малі та середні) [3]. Інтерфейс застосунку наведено на відповідному рисунку (див. Рис. 1.3).

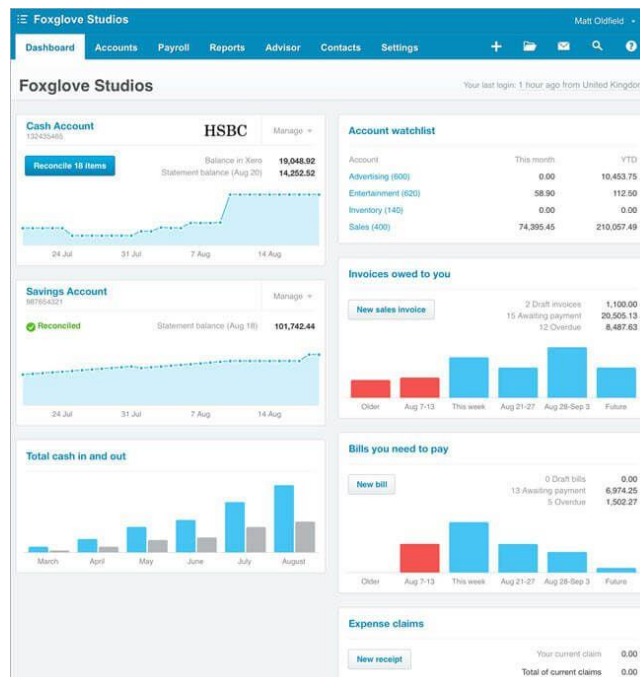


Рисунок 1.3 — Інтерфейс програмного застосунку «Херо» [1]

Серед функціональних можливостей програми можна зазначити:

- керування витратами підприємства;
- генерування рахунку-фактури проданої продукції;

- сплата рахунків за допомогою підключення модулю онлайн-платежів;
- підтримка кількох валют;
- можливість отримання інформації про запаси [1].

Це програмне забезпечення надається за умови надання платної підписки (середня вартість — 30 \$/місяць). Водночас, кожен модуль надається за окрему плату. Хоч це дає можливість обрати функції, які необхідні підприємству, проте робить застосунок відчутним у фінансових видатках суб'єкта господарювання. Отже, це програмне забезпечення найбільш за інших пристосоване для податкового та бухгалтерського обліку, хоч і потребує за це відчутних грошей.

Однак, використання західних аналогів пов'язано з деякими труднощами. Якщо виконання фінансових операцій з підключенням до онлайн-банкінгу та облік запасів не викликають складнощів, то розрахунки заробітної плати і виконання бухгалтерських проведення пов'язані з деякими труднощами. У той час, як західні програмні застосунки використовують у своїй роботі міжнародні стандарти обліку, більшість українських підприємств застосовують національні. Таким чином, для пристосування іноземних інформаційних систем до вітчизняних нормативів необхідне певне доопрацювання, що призведе до додаткових фінансових та часових витрат. Виділяти на це кошти заради того, щоб декілька місцевих фірм придбала ліцензію на використання — нецільно з економічної точки зору. Проте, у тому разі, якщо б український ринок бухгалтерських застосунків був повністю порожній — з високою ймовірністю деякі компанії вирішили б спробувати пропонувати свої продукти. Попит, безумовно, був би, адже альтернативою їхнім застосункам було б паперове діловодство. Проте інформаційна система, яка пристосована для української бухгалтерії — існує.

1.3 Інформаційна система бухгалтерського обліку, яка використовується на території України

Через те, що до набуття Україною незалежності, на її теренах діяли радянські стандарти бухгалтерського обліку, як і по всій території колишнього СРСР, країна не могла відразу почати використовувати міжнародні стандарти. Саме тому, коли з'явилась інформаційна система «1С: Підприємство» разом із модулем «Бухгалтерський облік для України», усі підприємства у державі, які переходили від паперового діловодства на цифрове, почали використовувати саме її. Все через те, що цей застосунок надавав досить розвинений функціонал за задовільні гроші та, що найголовніше, повністю підходив під використовувані стандарти. Після початку розробки національні норми бухгалтерського обліку, 1С: Підприємство почало модернізацію модулю «Бухгалтерський облік для України» під нові критерії. Тому, коли у 1999 році були прийняті нові національні стандарти бухгалтерського обліку, програмний продукт вже мав їхню підтримку. Після потрапляння під санкції та заборони на території України «1С: Підприємство» було перереєстроване під новою назвою — BAS (Business Automation Software).

BAS Бухгалтерія — програмний продукт, призначений для автоматизації бухгалтерського і податкового обліку, зокрема й підготовки обов'язкової (регламентованої) звітності, в організаціях, що здійснюють будь-які види комерційної діяльності: оптову і роздрібну торгівлю, комісійну торгівлю (включно з субкомісією), надання послуг, виробництво тощо. Бухгалтерський та податковий облік ведеться відповідно до чинного законодавства України [2]. Інтерфейс інформаційної системи «BAS Бухгалтерія» наведено на відповідному рисунку (див. Рис. 1.4).

N	Номенклатура	Характеристика	Партія	Виділення	Кількість	Кількість (об'єм)	Од.	Ціна
1	Молоко		<Не використовується>	<Не використовується>	9,000	9,000	шт	700.00
2	З'єднання прима стільниці AL ...	<Не використовується>	<Не використовується>	<Не використовується>	49,000	49,000	шт	0.50
3	З'єднання кутове стільниці AL ...	<Не використовується>	<Не використовується>	<Не використовується>	29,000	29,000	шт	10.00
4	Ганні M4x22	<Не використовується>	<Не використовується>	<Не використовується>	180,000	180,000	шт	14.07
5	Сарозуруп з овальною голівкою	<Не використовується>	<Не використовується>	<Не використовується>	400,000	400,000	шт	0.50
6	Малабок 118-300мм білий SA...	<Не використовується>	комісійний товар	<Не використовується>	69,000	69,000	шт	
7	Гачок RW 0703	<Не використовується>	<Не використовується>	<Не використовується>	96,000	96,000	шт	15.21
8	Демпфер вразній "SMOVE" 010...	<Не використовується>	комісійний товар	<Не використовується>	50,000	50,000	шт	
9	Шафа-купе	<Не використовується>	<Не використовується>	<Не використовується>	2,000	2,000	шт	278.05
10	Профіль ходовий ніжний L<3>м...	<Не використовується>	<Не використовується>	<Не використовується>	50,000	50,000	шт	20.00

Рисунок 1.4 — Інтерфейс програмного застосунку «BAS» [1]

Поточна версія BAS містить наступні функціональні розділи:

- продажі — дозволяє зберігати найменування, коди та ціни товарів (в тому числі — відображає їх зміни у часі разом з користувачем, який це виконував); зберігає надані пільги та знижки, введені націнки; тощо;
- продажі у роздріб — містить у собі інформацію про всі торговельні майданчики роздрібною продажу; має можливість генерації товарного звіту, створення етикеток та друк цінників; тощо;
- закупівлі — можливість внесення інформації про постачальників; контакти контрагентів; видача даних про графік поставок; облік запасів; тощо;
- роботи — надає документи про виконані роботи, надані послуги, тощо;
- виробництво — створення завдань на роботу, відрядних нарядів; загальне керування процесом виробництва; розрахунок собівартості; тощо;
- гроші — створення документів для обліку у банках та касі; розрахунки з підзвітними особами; генерування платіжного календаря; здійснення онлайн-платежів; тощо;
- зарплата — розрахунок заробітної плати; облік робочого часу; формування відповідних відомостей; тощо [12].

Звичайно, що наповнення інформаційної системи буде залежати від типу придбаної ліцензії. За початковий функціонал з найнеобхіднішими можливостями потрібно заплатити цілком підйомну суму, адже, наприклад, BAS Бухгалтерія. ПРОФ, зазначена на сайті, коштує 8 400 грн. на одного користувача, а повноцінна корпоративна (BAS Бухгалтерія КОРП) — 24 000 грн. на рік. [23].

Проте, не дивлячись на таку кількість функцій, у застосунку є і певні значущі недоліки.

1. Деяких можливостей все одно не вистачає. Наприклад, відсутній функціонал заповнення деяких банків даних (як то перелік та вартість товарів у певного постачальника). Таким чином, у разі глобальної зміни каталогу товарів фірми-контрагента доведеться власноруч вносити нову інформацію у застосунок, що вкрай не зручно та створює можливість некоректно внесеної інформації.

2. Відсутні можливості перевірки підприємства. Наприклад, неодноразово з'являлась необхідність провести внутрішній аудит наявності та використання основних засобів підприємства. В такому випадку необхідно мати інформацію про терміни використання та дату введення в експлуатацію матеріального активу. Це все зберігається у базі даних, що наведено на відповідному рисунку (див. Рис. 1.5).

Єврокуб 1000л (Основное средство)

Головне | Налоговые назначения ОС | События ОС организаций (регл. учет)

Записати та закрити | Записати | Створити на підставі | Інвентарна картка ОС (О3-6) | Щє | ?

Найменування: Єврокуб 1000л | Код: 00-000237

Організація: КОНТАКТСІЧ-АКВА НВП ТОВ | Інвентарний номер: 00-000237

Дата свідень: 02.12.2022 23:59:59

Основні свідення | Бухгалтерський учет | Налоговий учет

Первоначальная стоимость: 2 200,00 грн

Балансовая стоимость: 2 200,00 грн

Налоговая группа ОС: Группа 11: Малоценные необоротные материальные активы

Налоговое назначение: Обл. НДС

Начисление амортизации

Текущая амортизация: 2 200,00 грн

Срок полезного использования: 36 (3 года)

Состояния основного средства

Введено в эксплуатацию: 07.02.2019 | Ввод в эксплуатацию ОС АНО-000003 від 07.02.2019 17:00:06

Снято с учета: | Вести документ списания

Рисунок 1.5 — Інтерфейс про основний засіб (Єврокуб)

Однак, під час генерування переліку всіх основних засобів (що може використовуватись, в тому числі, і для аудиту, адже дані виводяться не лише на екран програмного застосунку, а і у Excel або XML), інформація надається в дуже стислому вигляді (див. Рис. 1.6).

Таким чином, для отримання інформації про терміни використання основних засобів необхідно власноруч відкривати кожен з них та звіряти інформацію з поданою звітністю.

Найменування	Код	Полное наименование
Орендовані ОС	00000027	
Придбання (виготовлення) інших необоротних матерія	00000025	
Придбання (виготовлення) основних засобів	00000022	
Придбання (створення) нематеріальних активів	00000026	
Акумуляторна батарея Varta	00-000165	Акумуляторна батарея Varta
Верстат заточувальний BC20-150	00-000387	Верстат заточувальний BC20-150
Гайковерт ударний електричний IW-1000	00-000234	Гайковерт ударний електричний IW-1000
Гайковерт ударний електричний IW-1000	00-000235	Гайковерт ударний електричний IW-1000
Дриль ударний Dnipro-M HD-120P	00-000406	Дриль ударний Dnipro-M HD-120P
Єрокуб 1000л	00-000166	Єрокуб 1000л
Єрокуб 1000л	00-000236	Єрокуб 1000л
Єрокуб 1000л	00-000237	Єрокуб 1000л
Єрокуб 1000л	00-000238	Єрокуб 1000л
Єрокуб 1000л	00-000239	Єрокуб 1000л
Єрокуб 1000л	00-000240	Єрокуб 1000л
Єрокуб 1000л	00-000241	Єрокуб 1000л
Єрокуб 1000л	00-000242	Єрокуб 1000л
Єрокуб 1000л	00-000243	Єрокуб 1000л
Єрокуб 1000л	00-000244	Єрокуб 1000л
Зварювальний апарат MWH-280	00-000245	Зварювальний апарат MWH-280
Зварювальний апарат MWH-280	00-000246	Зварювальний апарат MWH-280
Клапан запобіжний ЗП-ЗК-231	00-000385	Клапан запобіжний ЗП-ЗК-231
Клапан запобіжний ЗП-ЗК-231	00-000386	Клапан запобіжний ЗП-ЗК-231
Кувалда 2000г	00-000388	Кувалда 2000г
Ланцюг Husqvarna H35 14"5018470-52	00-000248	Ланцюг Husqvarna H35 14"5018470-52
Лом 1200	00-000389	Лом 1200

Рисунок 1.6 — Інтерфейс виведення інформації про основні засоби

3. Відсутня можливість подати звітність до державних органів. В застосунку передбачена можливість під'єднувати лише онлайн-банкінг. Функціонал зв'язування з податковою інспекцією та іншими державними та контролюючими органами — відсутня.

Якщо перші 2 проблеми так і не було вирішено, для подолання третьої українські підприємства використовують іншу інформаційну систему, яка має можливість зв'язування з ВАС — М.Е.Дос.

1.4 Інформаційна система подачі звітності до державних та контролюючих органів, яка використовується на території України

М.Е.Дос — інформаційна програмна система, яка призначена для подачі звітності до державних та контролюючих органів у цифровому форматі. Найдешевшою версією застосунку є М.Е.Дос. Бізнес. Цей варіант інформаційної системи стане підприємству у 1620 грн./рік. Основними можливостями застосунку, якій він надає за ці гроші, є:

- створення власних форматів для звітності;
- автоматична обробка типових відповідей від перевіряючих органів;
- надсилання до державних органів відповідних звітів, які підписані за допомогою цифрового ключу;
- інформування при отриманні документів;
- створення та підтримка резервного архіву для цифрових документів;
- можливість приєднання до електронних кабінетів (наприклад, електронний кабінет податкової);
- функціонал для отримання інформації від ВАС [6].

Інтерфейс програмного продукту наведено на відповідному рисунку (див. Рис. 1.7).

Основним недоліком цієї інформаційної системи є то, що вона призначена виключно для здійснення електронного документообігу та подання звітності до контролюючих органів.

Код	Дата модифікації	Стан	Тип	Номер додатку	Найменування
30500106	09.04.2021 21:42:...	Новий	Звітний		Об'єднана звітність ПДФО та ЄСВ
-J0510106			Звітний		(1) Податковий розрахунок сум доходу, нарахованого (сплаченого) на користь платників податків - фізичних осіб, і сум утриманих податків
-J0510106			Звітний		Додаток 1 (Д1) Відомості про нараховування заробітної плати (доходу, грошового забезпечення) застрахованим особам
-J0510106			Звітний		Додаток 1 (Д1) Відомості про нараховування заробітної плати (доходу, грошового забезпечення) застрахованим особам
-J0510106			Звітний		Додаток 1 (Д1) Відомості про нараховування заробітної плати (доходу, грошового забезпечення) застрахованим особам
-J0510206			Звітний		Додаток 2 (Д2) Відомості про осіб, які доглядають за дитиною до досягнення нею трирічного віку та відповідно до закону отримують податкову пільгу
-J0510206			Звітний		Додаток 2 (Д2) Відомості про осіб, які доглядають за дитиною до досягнення нею трирічного віку та відповідно до закону отримують податкову пільгу
-J0510206			Звітний		Додаток 2 (Д2) Відомості про осіб, які доглядають за дитиною до досягнення нею трирічного віку та відповідно до закону отримують податкову пільгу
-J0510306			Звітний		Додаток 3 (Д3) Відомості про осіб, які проходять строкову військову службу
-J0510306			Звітний		Додаток 3 (Д3) Відомості про осіб, які проходять строкову військову службу
-J0510306			Звітний		Додаток 3 (Д3) Відомості про осіб, які проходять строкову військову службу
-J0510406			Звітний		Додаток 4 (Д4) Відомості про суми нарахованого доходу, утриманого та сплаченого податку на доходи фізичних осіб та військових пенсій
-J0510406			Звітний		Додаток 4 (Д4) Відомості про суми нарахованого доходу, утриманого та сплаченого податку на доходи фізичних осіб та військових пенсій
-J0510406			Звітний		Додаток 4 (Д4) Відомості про суми нарахованого доходу, утриманого та сплаченого податку на доходи фізичних осіб та військових пенсій
-J0510506			Звітний		Додаток 5 (Д5) Відомості про трудові відносини осіб та період проходження військової служби
-J0510606			Звітний		Додаток 6 (Д6) Відомості про наявність підстав для обліку стажу окремим категоріям осіб відповідно до законодавства
-J0510606			Звітний		Додаток 6 (Д6) Відомості про наявність підстав для обліку стажу окремим категоріям осіб відповідно до законодавства
-J0510606			Звітний		Додаток 6 (Д6) Відомості про наявність підстав для обліку стажу окремим категоріям осіб відповідно до законодавства

Рисунок 1.7 — Інтерфейс інформаційної системи М.Е.Дос

1.5 Недоліки, які криються у використанні BAS та М.Е.Дос

Саме поєднання двох вищезазначених програмних продуктів (BAS та М.Е.Дос) — це те, без чого не обходиться жодне українське підприємство, яке бажає хоч трохи послабити гніт паперової бюрократії та мінімізувати можливість механічних помилок або «чорної» бухгалтерії. Проте, у використанні цих інформаційних систем криється багато прихованих недоліків, які всі помічають, але знайти щось краще на ринку програмних застосунків у поточний момент просто не мають змоги.

1. Частина документів підприємства ведеться не у спеціальних програмних застосунках, а у звичайних Word та Excel. І не дивлячись на те, що вони суворо типізовані і відрізняються один від одного лише обсягом та даними, залишаючи розмітку документу — незмінною, у BAS та М.Е.Дос відсутні модулі, за допомогою яких користувач може налаштувати шаблон отримання інформації з цих документів. А в кожному з них може бути дуже велика кількість інформації, інколи навіть більше 1000 позицій. Через це, якщо дані необхідно відобразити у BAS або М.Е.Дос, необхідно власноруч переносити їх до цих застосунків, що неодмінно призводить до помилок (адже безперервне вико-

нання шаблонних дій викликає втому та монотонію, що значно знижує уважність фахівців), через що необхідно декілька разів перевіряти введену інформацію окремими особами.

2. Деякі підприємства використовують так звані «галузеві програми». Це інформаційні системи, які полегшують працю штатних співробітників певного напрямлення, проте створюють додаткове фінансове навантаження на фірму, адже ліцензія коштує грошей. Наприклад, будівельні організації використовують програмний комплекс «АВК-5», який генерує документ із кошторисом. Замість цього було б зручніше використовувати певний режим у тому ж застосунку, яким здійснюється загальне керування підприємством. При застосуванні цієї модуль можна налаштувати на друк або формування переліку у бажаному стилі. Це економить час, адже можна створити шаблон розташування інформації у документі на власний розсуд ще на етапі його генерації, та гроші, адже підприємству не потрібно платити певні суми за ще одну ліцензію.

3. Окрім зазначених у попередньому пункті «галузевих програм» (яких може бути декілька), підприємству необхідно сплачувати за ліцензію самих BAS та M.E.Doc. Сукупність усіх цих програмних застосунків створює і справді відчутне фінансове навантаження на фірму.

Отже, рішенням усіх вищезазначених проблем буде розробка та впровадження власного програмного застосунку призначеного для керування підприємством, у якому буде відсутній зайвий функціонал, проте будуть наявними ті можливості, які будуть дійсно потрібними фірмам. Все це дозволить значно підвищити ефективність підприємства шляхом зменшення бюрократичних документів, відсутності механічних помилок при перенесенні інформації з одного застосунку до іншого та зменшить фінансове навантаження на підприємство.

2 ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Найдоцільніший формат інформаційної системи

Для ведення бухгалтерського обліку на підприємствах можна викремити дві версії інформаційної системи: WEB-сайт або desktop-застосунок. Кожна з них має свої переваги та недоліки.

2.1.1 Переваги та недоліки розробки системи у вигляді веб-застосунку

Перший варіант розробки інформаційної системи — технологія її реалізації у WEB-форматі (веб-застосунок). У якості його беззаперечної переваги — майже повна відсутність вимог до апаратної конфігурації комп'ютеру. Завдяки тому, що розгортання відбувається на сервері — веб-застосунок майже не використовує ресурси користувачького пристрою, через що сайт може працювати на будь-якій системі. Через те, що він повинен підтримувати спілкування з базою даних, передачу поточної інформації з пристроїв, команд на них та ще велику кількість функціоналу, створювати його без суворої типізації та об'єктно-орієнтованого стилю буде майже неможливим. У зв'язку з цим, розробники не можуть використовувати при створенні сайту лише найбільш популярну у світі комбінацію для веб-програмування (HTML та JS), зв'язуючи їх з базою завдяки програмній платформі Node.js. Саме через це, у світовій практиці, все частіше використовують технологію ASP .NET для реалізації таких застосунків. Завдяки цьому фреймворку частину функціоналу можна писати за допомогою жорстко типізованої мови програмування C# [21].

ASP.NET — це частина технології .NET, використовувана для написання потужних клієнт-серверних інтернет додатків. Вона дозволяє створю-

вати динамічні сторінки HTML. ASP.NET виникла в результаті об'єднання технології ASP (активні серверні сторінки) і .NET Framework. Вона містить безліч готових елементів управління, використовуючи які можна швидко створювати інтерактивні web-сайти. Також є можливість використовувати сервіси, що надаються іншими сайтами, прозоро для користувачів розроблюваного сайту. Загалом, можливості ASP.NET майже необмежені, порівняно з іншими технологіями [21].

Microsoft .NET Framework — це потужна платформа для розробки програмного забезпечення. Вона надає високопродуктивне, засноване на стандартах, багатомовне середовище, яке дозволяє розширювати існуючі застосунки за рахунок додатків і сервісів останнього покоління, а також вирішувати завдання розгортання і використання інтернет-додатків. .NET Framework складається з трьох основних частин — загальномовного середовища виконання (CLR, common language runtime), ієрархічної множини уніфікованих бібліотек класів і компонентної версії ASP, звану ASP.NET [21].

ASP.NET має велику кількість переваг, завдяки яким все частіше розробники вибирають саме його для створення своїх застосунків, а саме:

- Цей застосунок має «єдиний стек». Це означає, що за допомогою ASP.NET можна створювати не лише ту частину сайту, яку бачить користувач, але і писати на ній сервер, що значно спрощує розробку (full-stack розробка). Це дозволяє уникнути так званого «дублювання коду». Подібне відбувається у тих випадках, коли різні модулі сайту розроблюються на окремих технологіях, що викликає необхідність використовувати різні рішення для виконання схожих завдань [21].

- Технологія Blazor. Вона була додана компанією Microsoft у ASP.NET Core починаючи з версії 3.0. Завдяки цій розробці логіку клієнтської частини застосунку можна писати ще й на C# (або взагалі тільки на цій мові програмування, яка має велику кількість переваг), а не тільки за допомогою JavaScript [21].

- Єдиний пакет SDK і наскрізна підтримка залежностей. Під час розробки сайтів за допомогою більшості сучасних технологій необхідно пильно стежити за версією використовуваних модулів, адже може трапитись ситуація, коли вони можуть оновитися без підтримки старих версій, що призведе до некоректної роботи застосунку. У ASP.NET, під час додавання до пакету SDK (англ. Software Development Kit — засоби для розробки ПО) певного розширення, створюється наскрізне посилання саме на цю версію. Таким чином, навіть після оновлення модулів, застосунок буде звертатися до тієї версії, з використанням якої він був написаний [21].

- Компіляція. Основними конкурентами ASP.NET на ринку є PHP, Node.js або Ruby on Rails. Але між ними є велика різниця. Всі ці платформи є інтерпретованими, а в основі Microsoft .NET Framework полягає мова програмування C#, яка є компільованою мовою. Це надає програмним застосункам, які розроблені за допомогою ASP.NET, значну перевагу у швидкодії через те, що їм не потрібно інтерпретуватися (що економить значну кількість ресурсів системи) [21].

Однак, не дивлячись на переваги веб-застосунку, його використання для створення системи керування підприємством не буде задовільним через наступні причини:

- Розоблювана система не буде відіймати у комп'ютері багато ресурсів, адже не використовує якісь високозатратні розрахунки. Саме тому, для роботи із застосунком керування підприємством, буде цілком достатньо звичайних пристроїв, які зараз є на підприємствах.

- Доступ до застосунку може бути перервано у тому випадку, якщо на підприємстві виникнуть труднощі з під'єднанням до інтернету, чого ніколи не виникне при локальному використанні.

- Розгортання системи буде проходити на сторонніх обчислювальних потужностях, що створює небезпеку для захищеності даних.

Розглянувши всі вищенаведені переваги та недоліки стає зрозумілим той факт, що у поточній концепції використання інформаційної системи

керування підприємством розробка програми у форматі веб-застосунку є недоцільною.

2.1.2 Переваги та недоліки розробки системи у вигляді десктопного застосунку

Другий варіант розробки інформаційної системи — технологія її реалізації у настільному форматі (desktop-застосунок). У цьому випадку програмний застосунок постачається ліцензійним пакетом на підприємство та розгортається всередині нього за рахунок власних потужностей. З його переваг можна виділити наступні фактори:

1. Першим та, мабуть, найважливішим питанням стає забезпечення того, щоб дані не потрапили у треті руки (конкурентів або зловмисників). Інформаційною безпекою на великих підприємствах займаються лише висококваліфіковані, досвідчені професіонали, які добре знаються на своїй справі та розуміють, як запобігти злочинцям. Вони використовують різноманітні засоби захисту. Наприклад:

- захищають комп'ютери програмно. Вони блокують порти USB, щоб неможливо було скопіювати інформацію на флешку;
- встановлюють серйозні антивірусні та анти-шпигунські програми, щоб унеможливити викрадання дані через Інтернет;
- у випадку зберігання всередині внутрішній мережі підприємства інформації, яка є надважливою або вкрай секретною (така, яка підпадає під заборону розголошення комерційної таємниці згідно ст. 232 ККУ [14]) — взагалі забороняють підключати комп'ютер до «світової мережі» [18].

Саме через вищезазначені фактори, співробітники інформаційної безпеки, частіше за все, наполягають на розташуванні серверів у внутрішній мережі компанії. Безперечно, у тому ж місці можна розгорнути і веб-застосунок, проте можуть виникати труднощі з його оновленням (наприклад, для підтримки у новій версії браузера), адже доступ до інформаційної системи поза межами фірми буде відсутній.

2. Desktopна версія застосунку має більш високу автономність. Безумовно, вона залежна від наявності на підприємстві електроенергії, проте це можна вирішити забезпечивши сервер та комп'ютери системою безперебійного живлення. А у тому випадку, якщо на підприємстві зникне інтернет — вирішення питання з доступом до самого сайту та інформації, збереженої на віддалених серверах, буде складно вирішити. Адже багато фірм мають обмеження на підключення до всесвітньої мережі через обставини, які зазначені вище.

3. Швидкість роботи десктопного застосунку буде значно швидше, ніж у його веб-версії. Так, безперечно на дуже слабких та старих персональних комп'ютерах можуть виникати певні затримки у роботі. Проте основні прості системи, скоріш за все, будуть пов'язані із швидкістю отримання інформації з серверу, ніж із завантаженням сторінки або модулю, що ставить веб-версії подекуди навіть у більш скрутне становище.

У якості ж недоліку можна зазначити той факт, що наявність на підприємстві власного серверу створює потребу у людині, яка буде його обслуговувати та підтримувати. Проте, якщо на підприємстві є інтернет та комп'ютери, має бути і посада системного адміністратора. Це людина, яка цілком впорається з вирішенням поточних питань щодо коректної роботи та стосовно цілісності розміщеної у базі даних інформації.

Отже, беручи до уваги усе вищезазначене, можна дійти висновку, що використання технології створення настільного застосунку є найбільш оптимальною та привабливою як у плані зручності та надійності, так і у фінансовому аспекті.

2.2 Технології розробки настільних застосунків

При виборі мови програмування для розробки десктоп-застосунків, важливо розглянути кілька ключових мов та їхні особливості.

Java відома своєю кросплатформовістю, дозволяючи програмам працювати на різних операційних системах без модифікацій. Також, вона має широку спільноту розробників та розгалужену екосистему, що полегшує процес розробки. Однак, деякі розробники вказують на велику витрату ресурсів, особливо пам'яті, що може бути недоліком для деяких десктоп-застосунків.

Python, з іншого боку, виділяється своєю простотою та читабельністю коду. Маючи у власному розпорядженні велику кількість бібліотек для різних завдань безлічі галузей, Python стає привабливим вибором для розробників, які цінують продуктивність та ефективність. Однак, його повільніше виконання, порівняно з компільованими мовами програмування, може бути важливим фактором при виборі для великих та ресурсоємних десктоп-застосунків.

При розгляді мов програмування для розробки десктоп-застосунків важливо враховувати C++, що відзначається високою продуктивністю та близькістю до апаратного забезпечення. Використовуючи C++, розробники можуть створювати десктоп-застосунки з високою швидкістю виконання, що робить цю мову популярною для графічних додатків та ігор.

Однак, на шляху до вибору C++ стоять певні виклики. Складність цієї мови може виявитися перешкодою для початківців, які тільки починають свій шлях у програмуванні. Вивчення концепцій, таких як вказівники та ручне керування пам'яттю, може займати більше часу та зусиль порівняно з іншими мовами. Але є мова програмування, яка хоч і значно повільніше від C++, проте значно легша у вивченні і використанні, та навіть відноситься з нею до одного сімейства.

У останні роки однією з найкращих технологій для створення десктопних застосунків є мова програмування C#. Безумовно, суперечки навколо «першості» чогось майже завжди присутні у будь-якій галузі, що не оминає і сфери мов програмування. Проте велика кількість все ж таки відзначають саме C#. Наприклад, видання Decipherzone навіть поставило його на перше місце минулорічного рейтингу та назвало — найкращім вибором для створення комп'ютерних настільних застосунків [7].

C# — мова програмування, яка була розроблена корпорацією Microsoft, зазнавши впливу своїх попередників — C та C++. Вона має наявну типізацію, є об'єктно-орієнтованою та сучасною. До того ж, має схожий синтаксис з іншими мовами програмування (наприклад, Java), що дозволяє швидко перевчитись на неї програмістам, які працюють з іншими технологіями. До того ж, він легко взаємодіє з власними технологіями простої та швидкої розробки настільних застосунків від корпорації Microsoft: WPF, Mono та Windows Forms. Останній є найпопулярнішим завдяки тому, що його оволодіння займає значно менше часу, а отже і розробка застосунку за його допомогою буде значно дешевше [5].

Windows Forms — платформа користувацького інтерфейсу, яка була розроблена корпорацією Microsoft для створення програмних застосунків на платформі ОС Windows. Дозволяє виконувати розробку інтерфейсу за допомогою інтерактивної панелі (конструктора) у якій розробник збирає сторінку по окремому елементу, шляхом перетягування. Має можливість проектування та створення застосунків різного ступеня складності. Здатен підтримувати доступ до пам'яті комп'ютеру та встановленому на нього обладнанню. Розробка здійснюється за допомогою середовища розробки Visual Studio [4].

Visual Studio — інтегроване середовище розробки (ІСР), має функції для написання, відлагодження та збирання програмного застосунку. Відмінністю від інших засобів створення програм є наявність не тільки стандартних редакторів та відлагоджувальників, а і власні засоби автозавершення команд, платформ створення графічного інтерфейсу (в тому числі — Windows Forms) та компіляції. Завдяки вищезазначеним перевагам є однією з найвикористовуваніших інтегрованих середовищ розробки [8].

2.3 Вибір архітектури програмного забезпечення

Після прийняття рішення стосовно мови програмування інформаційної системи, настає час для вибору архітектури програмного застосунку. Вибір

мови в парі з архітектурою грає вирішальну роль у визначенні ефективності та розвитку десктоп-застосунків. Необхідно глибше вникнути в різні архітектурні концепції, розглядаючи їхні переваги та недоліки, щоб знайти оптимальне рішення для конкретного завдання.

Монолітна архітектура — це простий, але ефективний підхід, де весь код застосунку розташований в єдиному блоку. Це спрощує розробку та тестування, але може ускладнити масштабування та оновлення великих проєктів. Віддавати свій вибір на користь монолітної архітектури варто у тому випадку, коли програмний застосунок досить невеликий та необхідна простота розробки.

Мікросервісна архітектура використовує набір невеликих, незалежних мікросервісів для реалізації функціоналу. Це забезпечує гнучкість та високу доступність, але може ускладнити управління та конфігурацію. Обирати мікросервіси необхідно у тому випадку, коли потрібна велика гнучкість та планується значний розвиток проєкту.

Клієнт-серверна архітектура встановлює чітке розділення між функціоналом на клієнтській та серверній сторонах. Це забезпечує легше масштабування, але може призвести до збільшеної складності в розробці. Необхідно обирати цей підхід, якщо потрібно розподілити обов'язки між клієнтом та сервером, а у майбутньому планується значне розширення інформаційної системи.

2.4 Вибір платформи користувацького інтерфейсу

У віддзеркаленні наукової експозиції, необхідно розглянути декілька платформ користувацького інтерфейсу (UI) для розробки десктоп-застосунків з використанням мови програмування C#. Серед найбільш оптимальних та популярних інструментів виокремлюють такі інноваційні технології як: Windows Presentation Foundation (WPF), Windows Forms, Universal Windows Platform (UWP), та Avalonia.

Windows Presentation Foundation (WPF) — це фреймворк, який відрізняється виразністю та гнучкістю дизайну, дозволяючи розробникам створювати складні та динамічні інтерфейси. Застосовуючи концепцію шаблонізації, WPF пропонує засіб для елегантного та повторного використання коду. На жаль, його обмежена кросплатформовість робить його менш привабливим для проєктів, які розглядають розгортання на різних операційних системах.

Windows Forms — інтуїтивний інструмент для швидкої розробки десктоп-застосунків на основі .NET. Його схильність до простоти розробки та легкість інтеграції з .NET стали його сильними сторонами. Проте, його дизайн може виглядати застарілим порівняно з сучаснішими конкурентами, що не є суттєвим недоліком для розробки офісних застосунків, але дозволяє значно заощаджувати час та гроші.

Universal Windows Platform (UWP) — платформа, що надає гнучкість адаптації до різних пристроїв та високу інтеграцію з Windows 10. Однак, як і у випадку з WPF, обмежена кросплатформовість може ускладнити вибір для проєктів, які націлені на широку аудиторію.

Звертаючи увагу на сучасні тенденції та вимагаючи кросплатформовості, варто враховувати Avalonia. Цей інструмент прагне вирішити обмеження та недоліки попередніх фреймворків, надаючи кросплатформовість та сучасний дизайн. Однак його менша спільнота може викликати питання щодо підтримки та ресурсів.

Отже, у разі розробки офісної інформаційної системи, основною вимогою до якої є надійність та швидкодія, а не кросплатформовість та яскравий дизайн, перевагу варто віддати Windows Forms.

2.5 Вибір бази даних для розташування серверної частини

За останні дослідження мов програмування та архітектурних підходів для десктоп-застосунків, обраний шлях визначення оптимального рішення полягає у виборі системи керування базами даних (СКБД).

Саме тому прийнято рішення розглянути чотири ключові СКБД, які найчастіше використовують на ринку програмних продуктів: MySQL, PostgreSQL, Microsoft SQL Server та MongoDB. Для цього необхідно здійснити аналіз їхніх переваг та недоліків.

1. Відкритість та безкоштовність **MySQL** надають йому привабливості в ряді проектів. Задіяння великої спільноти розробників відзначається високою активністю. Проте, слід зазначити, що порівняно з конкурентами, MySQL може відставати за функціональністю, а це важливий момент при розробці інформаційної системи.

2. **MongoDB** залучає увагу своєю гнучкістю у роботі з неструктурованими даними та швидкою розробкою. Тим не менше, відсутність транзакцій у деяких сценаріях може викликати питання щодо забезпечення консистентності.

3. **PostgreSQL** відзначається розширеною підтримкою типів даних та операцій, що полегшує роботу з різноманітними даними. Однак варто врахувати, що ця система керування базою даних може вимагати значних ресурсів пам'яті.

4. На завершення, **Microsoft SQL Server**, як інтегрований у систему Microsoft, користується досить відчутним попитом завдяки високій продуктивності та інтеграції з власною екосистемою інших програмних рішень. Однак його вартість ліцензій може бути принциповим фактором, особливо для бюджетно орієнтованих проектів.

Вибір СКБД є стратегічним етапом, що вимагає уважного аналізу. Враховуючи вибір у якості мови програмування — C#, а у якості платформи — десктоп-застосунок із застосуванням Windows Forms (що є членами екосистеми Microsoft), ефективніше всього буде використовувати саме Microsoft SQL Server.

2.6 З'єднання програмного застосунку з базою даних

Інформаційна система буда з'єднуватися з базою даних за допомогою `System.Data.DataSet`. Цей клас можна розглядати як контейнер, який містить одну або кілька таблиць, зв'язаних за допомогою відношень. Кожна таблиця в `DataSet` має свій власний об'єкт `DataTable`, який зберігає схему та дані таблиці.

`DataSet` надає розширені можливості та є найефективнішим засобом для маніпулювання даними, включаючи фільтрацію, сортування, злиття, розділення та зміну структури даних.

Він також підтримує можливість виконання операцій з даними в повністю автономному режимі, що означає, що дані можуть бути завантажені з бази даних і збережені в `DataSet`, а подальша робота може відбуватися без постійного з'єднання з серверною частиною інформаційної системи, що може значно розвантажити сам сервер.

Використання саме цього контейнеру є дуже зручним у потчному форматі, адже `DataSet` доступен у будь-якому місці програмного застосунку. У разі зміни структури таблиці відсутня необхідність змінювати якісь моделі у проекті. Достатно лише зайти у конструктор `DataSet` та оновити його структуру (у разі зміни існуючої таблиці) або додати нову (у випадку створення нової таблиці).

Отже, найефективнішим рішенням буде реалізація десктоп-застосунку на базі `Windows Forms` (із використанням мови програмування `C#`), застосовуючи СКБД `Microsoft SQL Server` та зв'язуючи її із інформаційною системою за допомогою `DataSet`.

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ КЕРУВАННЯ ПІДПРИЄМСТВОМ

3.1 Бізнес обґрунтування та вимоги інформаційної системи

У рамках реалізації інформаційної системи для керування підприємством необхідно об'єднати у одному застосунку найбільш бажаний для підприємства функціонал. Це призведе до більш ефективного використання ресурсів фірми та економії її фінансових активів у довгостроковій перспективі, що дозволить більш стрімко розвиватись та підтримувати рентабельність.

В той же час, відсутня необхідність для впровадження можливостей, які або дуже складні для реалізації, або їх функціонал постачається безкоштовно (буде відсутня вигода від розробки). Також, застосунок не буде впроваджуватись у ті відділи, які використовують у своїй роботі лише одне програмне забезпечення, яке не є занадто дорогим, адже перехід на нову програму може завдати незручності та перевантажити інформаційну систему разом з її серверами.

3.2 Вимоги до програмного продукту

Перед початком проектування та розробки інформаційної системи було визначено функціональні та нефункціональні вимоги, яким вона повинна задовольняти.

3.2.1 Функціональні вимоги до програмного продукту

Для початку розробки та повноцінного впровадження інформаційної системи було необхідно створити можливість:

- ведення обліку співробітників;
- аналізу термінів використання основних засобів ;

- роботи із сторонніми документами у програмного застосунку (на прикладі можливості відкриття податкових накладних, які зберігаються на сервері);

- нарахування заробітної плати співробітникам відповідно до відпрацьованих ними годин;

- ведення обліку відпусток та лікарняних;

- визначення розміру податків та зберігання про це інформації;

- надання даних відносно ставок заробітної плати та термінів їхнього застосування.

3.2.2 Нефункціональні вимоги до програмного продукту

Ефективність:

- Програмний продукт повинен забезпечувати швидкий доступ до даних.

- Час відкриття та обробки сторонніх документів повинен бути прийнятним для користувача.

- Операції з базою даних повинні бути оптимізовані для ефективного використання ресурсів сервера MSSQL.

- Програмний продукт повинен бути готовий до масштабування, здатний ефективно працювати при збільшенні обсягу даних та істотному прирості навантаження.

Надійність:

- Програмний продукт повинен забезпечувати стабільну роботу без збоїв та критичних помилок.

- В разі виникнення помилок або відмов системи, програмний продукт повинен забезпечувати коректне відновлення роботи та збереження даних, які в ньому зберігаються.

Легкість використання:

- Інтерфейс користувача повинен бути простим та інтуїтивно зрозумілим для максимального комфорту абсолютної більшості користувачів, які будуть використовувати інформаційну систему.

- Програмний продукт має надавати зручні інструменти для взаємодії з системою та проведення необхідних операцій, які визначені функціональними та нефункціональними вимогами до нього.

- Забезпечення можливості персоналізації інтерфейсу для врахування індивідуальних потреб користувачів.

Безпека:

- Доступ до інформації повинен бути обмеженим та контрольованим.
- Система повинна забезпечувати автентифікацію користувачів.

3.3 Розробка бази даних інформаційної системи

Для зберігання інформації буде використовуватися Microsoft SQL Server Management Studio 18. На поточному етапі проектування база даних інформаційної системи для керування підприємством складається з одинадцяти таблиць. Вся інформація про зв'язаність таблиць та типи даних, які в них містяться, наведена на рисунку (див. Рис. 3.1).

1. *OblicoviZapisy* — таблиця, у якій зберігаються облікова інформація до доступу в систему. Містить логін та пароль для автентифікації в системі та роль користувача, який використав ці дані для входу до програмного застосування. Після того, як база даних підтвердила достовірність інформації, які ввели до інформаційної системи, програмний застосунок отримує роль, яку буде перевіряти у необхідні моменти.

2. *OsnovnyZasoby* — в ній зберігається інформація про наявні на підприємстві основні засоби, терміни їхньої експлуатації, дати прийняття на облік, унікальні коди в системі, повна та скорочені назви.

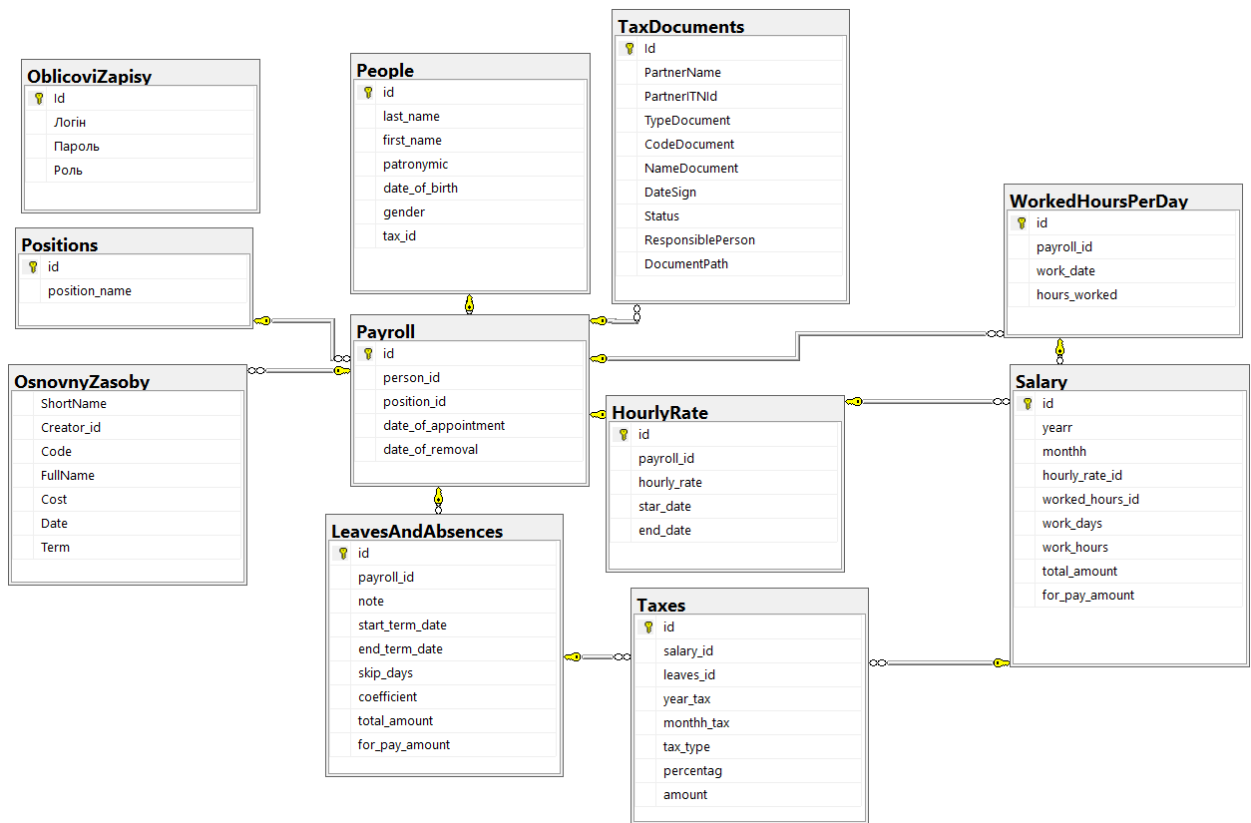


Рисунок 3.1 — Схема бази даних інформаційної системи

3. **People** — таблиця, у якій містяться дані про людей, працюючих на підприємстві. Вона зберігає прізвище, ім'я, по батькові, дата народження, стать та ПІН.

4. **Positions** — в ній зберігається інформація про існуючі на підприємстві посади, на які можуть призначатися співробітники.

5. **Payroll** — таблиця, яка представляє собою робочий табель, у якому поєднується інформація про співробітників, які посади вони займають, дата призначення та зняття з певної посади.

6. **TaxDocuments** — в ній зберігається інформація про податкові накладні (з яким підприємством підписано, що за тип документу, який його код, коли підписаний, хто відповідальна особа від підприємства, шляхи до цих документів тощо).

7. **HourlyRate** — таблиця для зберігання ставок погодинної оплати праці співробітників підприємства, яка містить саму ставку, дати початку та закінчення її дії.

8. `WorkedHoursPerDay` — відображає інформацію про кількість годин, які людина відпрацювала у конкретний день.

9. `Salary` — містить дані про кількість днів та годин, які співробітник підприємства відпрацював у певному році та місяці, яка зарплата йому була нарахована та що виплачено після обрахування податків.

10. `LeavesAndAbsences` — таблиця з інформацією про хвороби та відпустки працівника, у якій зазначено початок та закінчення цього періоду, скільки днів він був відсутнім, за яким коефіцієнтом обчислюється заробітна плата (наприклад, співробітникам із страховим стажем менше 3 років, окрім деяких категорій, лікарняні оплачуються з коефіцієнтом 0.5), які суми нараховані та сплачені після утримання податків.

11. `Taxes` — містить в собі дані про податки, які були нараховані на заробітну плату або відпустку, лікарняний тощо. Зберігаються рік та місяць, за який визначається податок, за якого співробітника сплачується, який його тип, відсоток та сума податку.

3.4 Розробка інформаційної системи за допомогою Visual Studio 2022

Розробка програмного продукту була розподілена на 3 частини:

- проектування інформаційної системи;
- реалізація програмної частини;
- розробка інтерфейсу застосунку.

3.4.1 Проектування інформаційної системи

Інформаційна система побудована з дотриманням стандартів якісного програмування та задовольняючи всім вимогам до програмного продукту, які були визначені у пункту 3.2. Функціонал застосунку, який доступний користувачу, наведено на Use-case діаграмі (див. Рис. 3.2).

Use Case Diagram — це діаграма, яка використовується для моделювання та візуалізації можливих варіантів використання системи з точки зору користувача. Це один з ключових інструментів в аналізі вимог, який допомагає розібратися в тому, як система взаємодіє з її акторами (користувачами, зовнішніми системами або іншими компонентами) та які функціональні можливості вона надає.

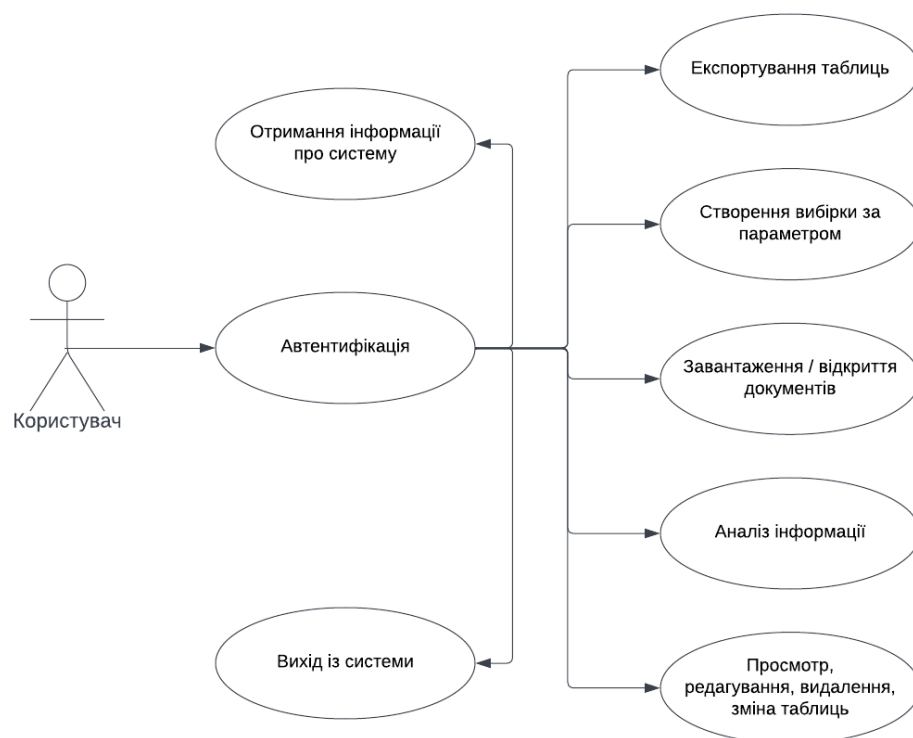


Рисунок 3.2 — Use-case діаграма

Для того, щоб було зрозуміло, як саме ці можливості користувача були втілені у інформаційну систему, наведена діаграма класів (див. Рис. 3.3).

Class Diagram — це інструмент визначення та візуалізації структури моделі системи. Вона надає статичне представлення класів, типів даних та їх взаємозв'язків. Діаграма класів використовується для визначення структури системи, опису класів і їх атрибутів, методів, спадкування, асоціацій, агрегацій та інших відношень між класами. Вона допомагає розуміти структуру системи,

визначати взаємозв'язки між класами, їх характеристики та взаємодію у межах системи.

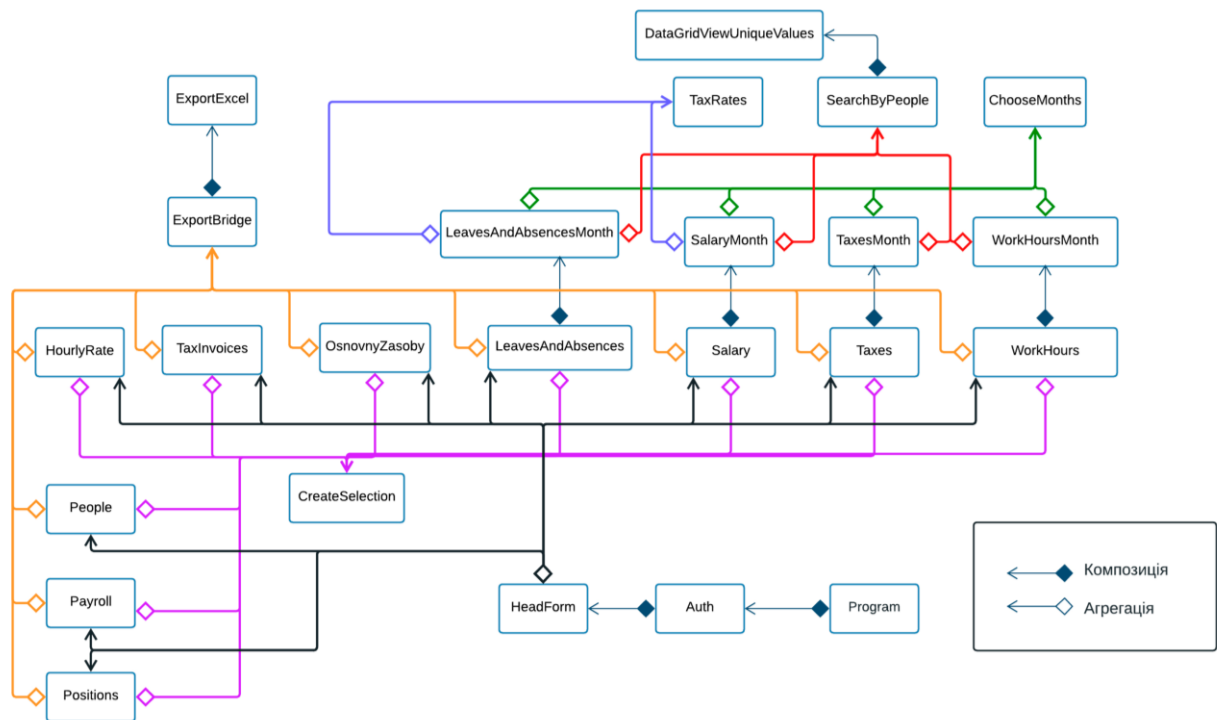


Рисунок 3.3 — Діаграма класів

Програмна частина інформаційної системи для керування підприємством складається з 24 класів, які можна умовно розділити на наступні групи:

- **Класи загальної взаємодії (1-3).** Вони призначені для підготовки програмного застосунку перед роботою з користувачем (відбувається зв'язок з базою даних, проводиться автентифікація, після чого надається виборі для подальшого відкриття вікон).
- **Класи відображення інформації (4-13).** Їхня задача полягає у групуванні та наданні користувачу тих даних, які зберігаються у таблицях на сервері.
- **Класи конкретизації періоду відображення інформації (14-17).** Вони є допоміжними для класів відображення інформації. В залежності від вибору користувача можуть надавати дані за весь період роботи інформаційної системи, за певний місяць, перераховувати значення у разі зміни.

- **Класи взаємодії (18-22).** Вони створені для виконання певних дій над інформацією, яку надає інформаційна система (конкретні дії залежать від їхнього призначення), та можуть бути розширені та деталізовані за допомогою модульних класів.

- **Модульні класи (23-24).** Якщо клас взаємодії має виконувати кілька схожих процедур із одним і тим самим набором даних (наприклад, експортування у різних форматах), доцільніше буде використовувати його у форматі шаблону проектування «Міст» (рівень абстракції), а рівень реалізації виносити у спеціально розроблені для цього модульні класи.

Більш детальний розгляд цих класів та опис деяких методів розглянуто далі (див. п. 3.5.2).

3.4.2 Реалізація програмної частини

1. Клас Program був створений автоматично та необхідний для ініціалізації програмного застосунку та передачу керування розробленим користувачам класам.

2. Клас Auth створено для проходження автентифікації користувачем інформаційної системи. Він зчитує логін та пароль з вікна програми та порівнює з тим, який знаходиться у базі даних (див. Лістинг 1).

Лістинг 1 — Функціонал кнопки «Логін»

```
private void buttonLogin_Click(object sender,
EventArgs e)
{
    try
    {
        string login = textBoxLogin.Text;
        string password = textBoxPass.Text;
        bool matchFound = false;
        foreach (DataGridViewRow row in
oblicoviZapisyDataGridView.Rows)
        {
            string rowLogin =
row.Cells[1].Value.ToString();
            string rowPassword =
row.Cells[2].Value.ToString();
```

```

        roleDB =
row.Cells[3].Value.ToString();
        if (login == rowLogin && password
== rowPassword)
        {
            matchFound = true;
            break;
        }
    }
    if (matchFound)
    {
        HeadForm headForm = new
HeadForm(this);

        this.Visible = false;
        headForm.Visible = true;
    }
    else
    {
        MessageBox.Show("Помилкові логін
або пароль. Спробуйте знову.");
    }
}
catch {
    MessageBox.Show("Помилкові логін або пароль.
Спробуйте знову.");
}
}

```

3. Клас `HeadForm` відкривається після проходження користувачем автентифікації. Служить для доступу до інших функціональних вікон за допомогою кнопок на формі.

4. Клас `OsnovnyZasoby` служить для виведення на екран інформації про основні засоби. Містить в собі функціонал проведення аналізу на протерміновані або придатні до експлуатації запаси, здійснення вибірки даних а також для їхнього експортування у excel (див. Лістинг 2).

Лістинг 2 — Функціонал аналізу термінів експлуатації ОЗ

```

private void аналізТермінівЕксплуата-
ціїToolStripMenuItem_Click(object sender, EventArgs e)
{
    DateTime date;
    DataTable bufferTable = new DataTable();
    int sizeBufferTable = 0;

```

```

        foreach (DataGridViewColumn column in
osnovnyZasobyDataGridView.Columns)
        {

bufferTable.Columns.Add(column.HeaderText,
column.ValueType);
        }
        foreach (DataGridViewRow row in
osnovnyZasobyDataGridView.Rows)
        {
            date =
Convert.ToDateTime(row.Cells[4].Value);
            date =
date.AddMonths(Convert.ToInt32(row.Cells[5].Value));
            if (DateTime.Now > date)
            {
                DataRow bufferRow =
bufferTable.NewRow();
                for (int i = 0; i <
osnovnyZasobyDataGridView.Columns.Count; i++)
                {
                    if (row.Cells[i].Value != null)
                    {
                        bufferRow[i] =
row.Cells[i].Value;
                    }
                    else
                    {
                        bufferRow[i] =
DBNull.Value;
                    }
                }
                bufferTable.Rows.Add(bufferRow);
                sizeBufferTable++;
            }
        }
        if (sizeBufferTable == 0)
        {
            MessageBox.Show("Відсутні протерміно-
вані основні засоби");
        }
        else
        {

bufferTable.Rows.RemoveAt(bufferTable.Rows.Count - 1);
        osnovnyZasobyDataGridView.DataSource =
bufferTable;
            Text = "Система керування підприємством
(Протерміновані основні засоби)";

```

```

        analizOpen = true;
    }
}

```

5. Клас `TaxInvoices` призначений для взаємодії з таблицею податкових накладних. Він містить в собі необхідний функціонал для завантаження або відкриття різних документів, здійснення вибірки, а також експорту у excel (див. Лістинг 3).

Лістинг 3 — Функціонал завантаження документів

```

private void завантажитиToolStripMenuItem_Click (object
sender, EventArgs e) {
    using (var folderDialog = new
FolderBrowserDialog())
    {
        DialogResult result =
folderDialog.ShowDialog();
        if (result == DialogResult.OK &&
!string.IsNullOrEmpty(folderDialog.SelectedPath))
        {
            string targetFolder =
folderDialog.SelectedPath;
            foreach (DataGridViewRow row in
taxDocumentsDataGridView.Rows)
            {
                string documentPath =
row.Cells[columnPath].Value?.ToString() ?? string.Empty;
                string fileName =
Path.GetFileName(documentPath);
                string destinationPath =
Path.Combine(targetFolder, fileName);
                if
(!string.IsNullOrEmpty(fileName)) // Проверка на пустую
строку
                {
                    try
                    {
                        File.Copy(documentPath,
destinationPath);
                    }
                    catch (IOException ex)
                    {
                        MessageBox.Show($"Не
вдалося скопіювати файл {fileName}: {ex.Message}");
                    }
                }
            }
        }
    }
}

```



```

        }
        catch
        {
            MessageBox.Show($"Виникла помилка.");
        }
    }
    Process.Start(targetFolder);
}
}
}

```

6. Клас People призначений для взаємодії з особистими даними користувачів. Містить функціонал для здійснення вибірки даних, а також для експортування у excel.

7. Клас Positions необхідний для виводу інформації про посади, які наявні у інформаційній системі. Є можливість здійснення фільтрації інформації та експортування даних.

8. Клас Payroll служить для виведення на екран інформації про таблиць співробітників підприємства. Містить в собі функціонал проведення аналізу на звільнених співробітників, здійснення вибірки даних і їхнього експортування у excel (див. Лістинг 4).

Лістинг 4 — Функціонал аналізу на наявність звільнених співробітників

```

private void звільненіСпівробітникиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DataTable bufferTable = new DataTable();
    int sizeBufferTable = 0;

    foreach (DataGridViewColumn column in payrollDataGridView.Columns)
    {
        bufferTable.Columns.Add(column.Name, column.ValueType);
    }

    foreach (DataGridViewRow row in payrollDataGridView.Rows)
    {

```

```

        if (row.Cells[6].Value != null &&
row.Cells[6].Value.ToString() != "")
        {
            DataRow bufferRow =
bufferTable.NewRow();
            for (int i = 0; i <
payrollDataGridView.Columns.Count; i++)
            {
                if (row.Cells[i].Value != null)
                {
                    bufferRow[i] =
row.Cells[i].Value;
                }
                else
                {
                    bufferRow[i] =
DBNull.Value;
                }
                bufferTable.Rows.Add(bufferRow);
                sizeBufferTable++;
            }
        }
        if (sizeBufferTable == 0)
        {
            MessageBox.Show("Відсутні звільнені
співробітники");
        }
        else
        {
            payrollDataGridView.DataSource = new
BindingSource(bufferTable, null);
            Text = "Система керування підприємством
(Співробітники.Звільнено)";
            analizOpen = true;
        }
    }
}

```

9. Клас `HourlyRate` відповідає за вивід на екран інформації про ставки заробітної плати, впроваджені на підприємстві, до якої посади вони застосовуються та період їхньої дії. Він має можливості здійснення вибірки та експортування даних у бажаний формат.

10. Клас `LeavesAndAbsences` надає користувачу інформацію про лікарняні та відпустки, які брали співробітники підприємства за певний період або за весь час. Також присутні можливості виконання фільтрації даних, їхнього експорту та пошуку за ПІБ та посадою співробітника.

Виведення інформації відбувається за допомогою SQL-запиту до БД (див. Лістинг 5).

Лістинг 5 — SQL-запит для виведення інформації про лікарняні/відпустки

```

SELECT      LeavesAndAbsences.id,
LeavesAndAbsences.payroll_id, LeavesAndAbsences.note,
LeavesAndAbsences.start_term_date,
LeavesAndAbsences.end_term_date,
LeavesAndAbsences.coefficient, Positions.position_name,
                People.tax_id,
People.last_name, People.first_name, People.patronymic,
LeavesAndAbsences.skip_days, HourlyRate.hourly_rate,
LeavesAndAbsences.total_amount,
LeavesAndAbsences.for_pay_amount
FROM        LeavesAndAbsences INNER JOIN
                Payroll ON
LeavesAndAbsences.payroll_id = Payroll.id INNER JOIN
                People ON Payroll.person_id =
People.id INNER JOIN
                Positions ON
Payroll.position_id = Positions.id INNER JOIN
                HourlyRate ON Payroll.id =
HourlyRate.payroll_id

```

Для перерахунку відпускних/лікарняних для всіх співробітників, із зміною інформації про сплачені податки, використовується збережена процедура (див. Лістинг 6).

Лістинг 6 — Збережена процедура, яка перераховує розмір відпусток/лікарняних за весь час (враховуючи податки)

```
USE EnterpriseManagementDatabase;
```

```

-- Оновлення таблиці LeavesAndAbsences з розрахунком
total_amount та skip_days
UPDATE LeavesAndAbsences
SET
    total_amount = (DATEDIFF(day, start_term_date,
end_term_date) + 1)* coefficient * HourlyRate.hourly_rate,
    skip_days = DATEDIFF(day, start_term_date,
end_term_date) + 1
FROM
    LeavesAndAbsences
JOIN Payroll ON LeavesAndAbsences.payroll_id =
Payroll.id
JOIN HourlyRate ON Payroll.id = HourlyRate.payroll_id;

-- Видалення записів про податки, які вже були за цей
період
DELETE FROM Taxes
WHERE EXISTS (
    SELECT 1
    FROM LeavesAndAbsences la
    WHERE Taxes.leaves_id = la.id
);

-- Додавання нових записів про податки до таблиці Taxes
INSERT INTO Taxes (leaves_id, tax_type, percentag,
amount, year_tax, monthh_tax)
SELECT
    la.id AS leaves_id,
    'ECB' AS tax_type,
    @ESVRate AS percentag,
    (la.total_amount * @ESVRate) AS amount,
    YEAR(la.start_term_date) AS year_tax,
    MONTH(la.start_term_date) AS monthh_tax
FROM
    LeavesAndAbsences la
WHERE
    YEAR(la.end_term_date) = @SelectedYear
    AND MONTH(la.end_term_date) = @SelectedMonth
    AND la.total_amount IS NOT NULL;

INSERT INTO Taxes (leaves_id, tax_type, percentag,
amount, year_tax, monthh_tax)
SELECT
    la.id AS leaves_id,
    'ПДФО' AS tax_type,
    @PDFORate AS percentag,
    (la.total_amount * @PDFORate) AS amount,
    YEAR(la.start_term_date) AS year_tax,
    MONTH(la.start_term_date) AS monthh_tax

```

```

FROM
    LeavesAndAbsences la
WHERE
    YEAR(la.end_term_date) = @SelectedYear
    AND MONTH(la.end_term_date) = @SelectedMonth
    AND la.total_amount IS NOT NULL;

INSERT INTO Taxes (leaves_id, tax_type, percentag,
amount, year_tax, monthh_tax)
SELECT
    la.id AS leaves_id,
    'B3' AS tax_type,
    @VZRate AS percentag,
    (la.total_amount * @VZRate) AS amount,
    YEAR(la.start_term_date) AS year_tax,
    MONTH(la.start_term_date) AS monthh_tax
FROM
    LeavesAndAbsences la
WHERE
    YEAR(la.end_term_date) = @SelectedYear
    AND MONTH(la.end_term_date) = @SelectedMonth
    AND la.total_amount IS NOT NULL;

-- Оновлення поля for_pay_amount в таблиці
LeavesAndAbsences
UPDATE LeavesAndAbsences
SET for_pay_amount = total_amount - (
    SELECT COALESCE(SUM(amount), 0)
    FROM Taxes
    WHERE Taxes.leaves_id = LeavesAndAbsences.id AND
Taxes.tax_type != 'ECB'
);

```

Пошук за ПІБ та/або посадою може виконуватись лише після завантаження таблиці, адже система пропонує лише ті варіанти заповнення, стосовно яких є інформація у наведеному вікні. Для цього користувач повинен натиснути відповідну кнопку на панелі швидкого доступу (див. Лістинг 7).

Лістинг 7 — Виконання дій у разі натискання кнопки пошуку

```

private void по-
шукToolStripMenuItem_Click(object sender, EventArgs e)
{
    List<string> uniquePositions =
GetUniqueColumnValues("position_name");

```

```

        List<string> uniqueLastNames =
GetUniqueColumnValues("last_name");
        List<string> uniqueFirstNames =
GetUniqueColumnValues("first_name");
        List<string> uniquePatronymics =
GetUniqueColumnValues("patronymic");

        DataGridViewUniqueValues data = new
DataGridViewUniqueValues
        {
            UniquePositions = uniquePositions,
            UniqueLastNames = uniqueLastNames,
            UniqueFirstNames = uniqueFirstNames,
            UniquePatronymics = uniquePatronymics
        };
        searchByPeople = new
SearchByPeople(headForm, data);

        searchByPeople.SearchButtonClicked +=
SearchPeople_SearchButtonClicked;
        searchByPeople.SearchFormClosing +=
SearchPeople_FormClosing;

        this.Enabled = false;
        searchByPeople.Visible = true;
        searchByPeople.Location = new
Point(this.Location.X, this.Location.Y);
    }

```

На початку виконання попереднього методу були оброблено стовпці, стосовно яких дозволено здійснювати пошук (див. Лістинг 8).

Лістинг 8 — Отримання унікальних значень з таблиці

```

private List<string>
GetUniqueColumnValues(string columnName)
{
    return leavesAndAbsencesDataGridView.Rows
        .Cast<DataGridViewRow>()
        .Select(row =>
row.Cells[columnName].Value?.ToString())
        .Where(value =>
!string.IsNullOrEmpty(value))
        .Distinct()
        .OrderBy(value => value)
        .ToList();
}

```

Під час виконання цього методу викликається клас-форма SearchByPeople, яка надає користувачу вибір з 4 параметрів, за якими здійснюється пошук (вона буде детальніше описана у пункті 19 цього підрозділу). Після її виконання спеціальний фільтр перевіряє, які саме поля були обрані для пошуку та чи обрані взагалі (див. Лістинг 9).

Лістинг 9 — Фільтр для отримання обраних полів

```

        public void ApplyFilter(Dictionary<string,
string> selectedValues)
        {
            List<string> filterConditions = new
List<string>();
            if (selectedValues["Position"] != "-Нічого не
обрано-")
            {
                filterConditions.Add($"position_name =
'{selectedValues["Position"]}'");
            }
            if (selectedValues["Surname"] != "-Нічого не об-
рано-")
            {
                filterConditions.Add($"last_name =
'{selectedValues["Surname"]}'");
            }
            if (selectedValues["Name"] != "-Нічого не об-
рано-")
            {
                filterConditions.Add($"first_name =
'{selectedValues["Name"]}'");
            }
            if (selectedValues["Patronymic"] != "-Нічого не
обрано-")
            {
                filterConditions.Add($"patronymic =
'{selectedValues["Patronymic"]}'");
            }
            if (filterConditions.Count != 0)
            {
                string filter = string.Join(" AND ",
filterConditions);
                leavesAndAbsencesBindingSource.Filter =
filter;

                selectionStart = true;
            }
        }
    
```

```

        Text = "Система керування підприємством (Заробітна плата.Вибірка)";
    }
    searchByPeople.Close();
}

```

11. Клас Salary відповідає за вивід на екран інформації про заробітну плату співробітників або за конкретний місяць, який цікавить користувача, або за весь період роботи підприємства. Також у класі реалізована можливість розрахунку заробітної плати із відрахуванням податків

Він має можливості здійснення вибірки, експортування даних у бажаний формат та пошуку за ПІБ та/або посадою. (див. Лістинг 10). Ставки податків є змінними та зазначаються у окремому класі (зазначено у п. 20 цього розділу).

Лістинг 10 — Розрахунок заробітної плати за весь період (із зміною нарахованих податків)

```

        USE EnterpriseManagementDatabase;
-- Оновлення таблиці Salary з розрахунком total_amount
MERGE INTO Salary AS target
USING (
    SELECT
        hr.id AS hourly_rate_id,
        YEAR(wd.work_date) AS yearr,
        MONTH(wd.work_date) AS monthh,
        COUNT(DISTINCT wd.work_date) AS work_days,
        SUM(wd.hours_worked) AS work_hours,
        SUM(hr.hourly_rate * wd.hours_worked) AS
total_amount
    FROM
        HourlyRate hr
    JOIN Payroll p ON hr.payroll_id = p.id
    JOIN People pe ON p.person_id = pe.id
    JOIN WorkedHoursPerDay wd ON p.id = wd.payroll_id
    GROUP BY
        hr.id, YEAR(wd.work_date), MONTH(wd.work_date)
) AS source
ON target.hourly_rate_id = source.hourly_rate_id
AND target.yearr = source.yearr
AND target.monthh = source.monthh
WHEN MATCHED THEN
    UPDATE SET
        target.work_days = source.work_days,

```



```

        target.work_hours = source.work_hours,
        target.total_amount = source.total_amount
    WHEN NOT MATCHED THEN
        INSERT (hourly_rate_id, yearr, monthh, work_days,
work_hours, total_amount, for_pay_amount)
        VALUES (source.hourly_rate_id, source.yearr,
source.monthh, source.work_days, source.work_hours,
source.total_amount, 0.0);
    -- Оновлення полів year_tax та monthh_tax в таблиці
Taxes
    UPDATE Taxes
    SET
        year_tax = YEAR(wd.work_date),
        monthh_tax = MONTH(wd.work_date)
    FROM
        Taxes t
    JOIN LeavesAndAbsences la ON t.leaves_id = la.id
    JOIN Payroll p ON la.payroll_id = p.id
    JOIN WorkedHoursPerDay wd ON p.id = wd.payroll_id;

    -- Видалення всіх існуючих записів про податки
    DELETE FROM Taxes
    WHERE EXISTS (
        SELECT 1
        FROM Salary sa
        WHERE Taxes.salary_id = sa.id
    );

    -- Додавання нових записів про податки до таблиці Taxes
    INSERT INTO Taxes (salary_id, tax_type, percentag,
amount, year_tax, monthh_tax)
    SELECT
        s.id AS salary_id,
        'ECB' AS tax_type,
        @ESVRate AS percentag,
        (s.total_amount * @ESVRate) AS amount,
        s.yearr AS year_tax,
        s.monthh AS monthh_tax
    FROM
        Salary s;
    INSERT INTO Taxes (salary_id, tax_type, percentag,
amount, year_tax, monthh_tax)
    SELECT
        s.id AS salary_id,
        'ПДФО' AS tax_type,
        @PDFORate AS percentag,
        (s.total_amount * @PDFORate) AS amount,
        s.yearr AS year_tax,
        s.monthh AS monthh_tax

```

```

FROM
    Salary s;

INSERT INTO Taxes (salary_id, tax_type, percentag,
amount, year_tax, monthh_tax)
SELECT
    s.id AS salary_id,
    'B3' AS tax_type,
    @VZRate AS percentag,
    (s.total_amount * @VZRate) AS amount,
    s.yearr AS year_tax,
    s.monthh AS monthh_tax
FROM
    Salary s;
-- Оновлення поля for_pay_amount в таблиці Salary
UPDATE Salary
SET for_pay_amount = total_amount - (
    SELECT SUM(amount)
    FROM Taxes
    WHERE Taxes.salary_id = Salary.id AND
Taxes.tax_type != 'ECB'
);

```

12. Клас Taxes надає користувачу інформацію про податки, які підприємство сплатило за нього впродовж певного період або за весь час (в тому числі ті, які не утримуються напряму із заробітної плати). Також присутні можливості виконання фільтрації даних, їхнього експорту та пошуку за ПІБ та посадою співробітника (див. Лістинг 11).

Лістинг 11 — SQL-запит для виведення інформації на екран

```

SELECT
    Taxes.id, Positions.position_name,
    People.tax_id, People.last_name, People.first_name,
    People.patronymic, Salary.yearr, Salary.monthh,
    Taxes.tax_type, Taxes.percentag, Taxes.amount,
    Taxes.salary_id, Taxes.leaves_id,
    Taxes.year_tax,
    Taxes.monthh_tax
FROM
    Taxes LEFT OUTER JOIN
        (SELECT
            Salary_1.id
        AS salary_id, Payroll.id AS payroll_id, Salary_1.yearr,
        Salary_1.monthh
        FROM
            Salary
        AS Salary_1 LEFT OUTER JOIN

```

```

HourlyRate ON Salary_1.hourly_rate_id = HourlyRate.id LEFT
OUTER JOIN

WorkedHoursPerDay ON Salary_1.worked_hours_id =
WorkedHoursPerDay.id LEFT OUTER JOIN

Payroll ON COALESCE (HourlyRate.payroll_id,
WorkedHoursPerDay.payroll_id) = Payroll.id) AS Salary ON
Taxes.salary_id = Salary.salary_id LEFT OUTER JOIN
        LeavesAndAbsences ON
Taxes.leaves_id = LeavesAndAbsences.id LEFT OUTER JOIN
        Payroll AS PayrollLeaves ON
LeavesAndAbsences.payroll_id = PayrollLeaves.id LEFT OUTER
JOIN
        Payroll AS PayrollSalary ON
Salary.payroll_id = PayrollSalary.id INNER JOIN
        People ON COALESCE
(PayrollLeaves.person_id, PayrollSalary.person_id) =
People.id INNER JOIN
        Positions ON COALESCE
(PayrollLeaves.position_id, PayrollSalary.position_id) =
Positions.id;

```

13. Клас `WorkHours` надає інформацію про дати роботи співробітників та кількість годин, які вони тоді відпрацювали. Він має можливості здійснення вибірки, експортування даних у бажаний формат та пошук за ПІБ та/або посадою.

14. Клас `LeavesAndAbsencesMonth` надає користувачу можливість обрати місяць та рік для відображення інформації про лікарняні та відпустки, а також зробити їхні розрахунки за цей період, із врахуванням податків (див. Лістинг 12).

Лістинг 12 — SQL-запит для розрахунку сум та податків для лікарняних/відпусток за певний місяць

```

USE EnterpriseManagementDatabase;

-- Оновлення таблиці LeavesAndAbsences з розрахунком
total_amount та skip_days
UPDATE LeavesAndAbsences
SET

```

```

        total_amount = (DATEDIFF(day, start_term_date,
end_term_date) + 1) * coefficient * HourlyRate.hourly_rate,
        skip_days = DATEDIFF(day, start_term_date,
end_term_date) + 1
    FROM
        LeavesAndAbsences
    JOIN Payroll ON LeavesAndAbsences.payroll_id =
Payroll.id
    JOIN HourlyRate ON Payroll.id = HourlyRate.payroll_id;
-- Видалення записів про податки, які вже були за цей
період
DELETE FROM Taxes
WHERE EXISTS (
    SELECT 1
    FROM LeavesAndAbsences la
    WHERE Taxes.leaves_id = la.id
);
-- Додавання нових записів про податки до таблиці Taxes
INSERT INTO Taxes (leaves_id, tax_type, percentag,
amount, year_tax, monthh_tax)
SELECT
    la.id AS leaves_id,
    'ECB' AS tax_type,
    @ESVRate AS percentag,
    (la.total_amount * @ESVRate) AS amount,
    YEAR(la.start_term_date) AS year_tax,
    MONTH(la.start_term_date) AS monthh_tax
FROM
    LeavesAndAbsences la
WHERE
    YEAR(la.end_term_date) = @SelectedYear
    AND MONTH(la.end_term_date) = @SelectedMonth
    AND la.total_amount IS NOT NULL;
INSERT INTO Taxes (leaves_id, tax_type, percentag,
amount, year_tax, monthh_tax)
SELECT
    la.id AS leaves_id,
    'ПДФО' AS tax_type,
    @PDFORate AS percentag,
    (la.total_amount * @PDFORate) AS amount,
    YEAR(la.start_term_date) AS year_tax,
    MONTH(la.start_term_date) AS monthh_tax
FROM
    LeavesAndAbsences la
WHERE
    YEAR(la.end_term_date) = @SelectedYear
    AND MONTH(la.end_term_date) = @SelectedMonth
    AND la.total_amount IS NOT NULL;

```

```

INSERT INTO Taxes (leaves_id, tax_type, percentag,
amount, year_tax, monthh_tax)
SELECT
    la.id AS leaves_id,
    'B3' AS tax_type,
    @VZRate AS percentag,
    (la.total_amount * @VZRate) AS amount,
    YEAR(la.start_term_date) AS year_tax,
    MONTH(la.start_term_date) AS monthh_tax
FROM
    LeavesAndAbsences la
WHERE
    YEAR(la.end_term_date) = @SelectedYear
    AND MONTH(la.end_term_date) = @SelectedMonth
    AND la.total_amount IS NOT NULL;
-- Оновлення поля for_pay_amount в таблиці
LeavesAndAbsences
UPDATE LeavesAndAbsences
SET for_pay_amount = total_amount - (
    SELECT COALESCE(SUM(amount), 0)
    FROM Taxes
    WHERE Taxes.leaves_id = LeavesAndAbsences.id AND
Taxes.tax_type != 'ECB');

```

15. Клас SalaryMonth забезпечує функціонал не лише для обрання місяця та року з метою подальшого відображення інформації про заробітну плату у певний період, а і дозволяє проводити її розрахунки за цей період із врахуванням податків (див. Лістинг 13).

Лістинг 13 — SQL-запит для розрахунку сум та податків для заробітної плати за певний місяць

```

MERGE INTO Salary AS target
USING (
    SELECT
        hr.id AS hourly_rate_id,
        YEAR(wd.work_date) AS yearr,
        MONTH(wd.work_date) AS monthh,
        COUNT(DISTINCT wd.work_date) AS work_days,
        SUM(wd.hours_worked) AS work_hours,
        SUM(hr.hourly_rate * wd.hours_worked) AS
total_amount
    FROM
        HourlyRate hr
    JOIN Payroll p ON hr.payroll_id = p.id

```

```

        JOIN People pe ON p.person_id = pe.id
        LEFT JOIN WorkedHoursPerDay wd ON p.id =
wd.payroll_id
        WHERE
            YEAR(wd.work_date) = @SelectedYear
            AND MONTH(wd.work_date) = @SelectedMonth
        GROUP BY
            hr.id, YEAR(wd.work_date),
MONTH(wd.work_date)
    ) AS source
    ON target.hourly_rate_id = source.hourly_rate_id
    AND target.yearr = source.yearr
    AND target.monthh = source.monthh
    WHEN MATCHED THEN
        UPDATE SET
            target.work_days = source.work_days,
            target.work_hours = source.work_hours,
            target.total_amount = source.total_amount
    WHEN NOT MATCHED THEN
        INSERT (hourly_rate_id, yearr, monthh,
work_days, work_hours, total_amount, for_pay_amount)
        VALUES (source.hourly_rate_id, source.yearr,
source.monthh, source.work_days, source.work_hours,
source.total_amount, 0.0);
    -- Оновлення полів year_tax та monthh_tax в таблиці
Taxes
    UPDATE Taxes
    SET
        year_tax = YEAR(wd.work_date),
        monthh_tax = MONTH(wd.work_date)
    FROM
        Taxes t
    JOIN LeavesAndAbsences la ON t.leaves_id = la.id
    JOIN Payroll p ON la.payroll_id = p.id
    JOIN WorkedHoursPerDay wd ON p.id = wd.payroll_id;
    -- Видалення існуючих записів про податки для пев-
ного року та місяця
    DELETE FROM Taxes
    WHERE
        salary_id IN (
            SELECT id
            FROM Salary sa
            WHERE yearr = @SelectedYear
            AND monthh = @SelectedMonth
            AND Taxes.salary_id = sa.id
        );
    -- Додавання нових записів про податки до таблиці
Taxes

```

```

INSERT INTO Taxes (salary_id, tax_type, percentag,
amount, year_tax, monthh_tax)
SELECT
    s.id AS salary_id,
    'ECB' AS tax_type,
    @ESVRate AS percentag,
    (s.total_amount * @ESVRate) AS amount,
    s.yearr AS year_tax,
    s.monthh AS monthh_tax
FROM
    Salary s
WHERE
    s.yearr = @SelectedYear
    AND s.monthh = @SelectedMonth;
INSERT INTO Taxes (salary_id, tax_type, percentag,
amount, year_tax, monthh_tax)
SELECT
    s.id AS salary_id,
    'ПДФО' AS tax_type,
    @PDFORate AS percentag,
    (s.total_amount * @PDFORate) AS amount,
    s.yearr AS year_tax,
    s.monthh AS monthh_tax
FROM
    Salary s
WHERE
    s.yearr = @SelectedYear
    AND s.monthh = @SelectedMonth;
INSERT INTO Taxes (salary_id, tax_type, percentag,
amount, year_tax, monthh_tax)
SELECT
    s.id AS salary_id,
    'B3' AS tax_type,
    @VZRate AS percentag,
    (s.total_amount * @VZRate) AS amount,
    s.yearr AS year_tax,
    s.monthh AS monthh_tax
FROM
    Salary s
WHERE
    s.yearr = @SelectedYear
    AND s.monthh = @SelectedMonth;
-- Оновлення поля for_pay_amount в таблиці Salary
UPDATE Salary
SET for_pay_amount = total_amount - (
    SELECT SUM(amount)
    FROM Taxes
    WHERE Taxes.salary_id = Salary.id AND
Taxes.tax_type != 'ECB'

```

```

)
WHERE
    yearr = @SelectedYear
    AND monthh = @SelectedMonth;
";

```

16. Клас `TaxesMonth` надає користувачу можливість обрати місяць та рік для відображення інформації сплачені підприємством податки за співробітників.

17. Клас `WorkHoursMonth` забезпечує функціонал лише для обрання місяцю та року з метою подальшого відображення інформації стосовно відпрацьованих днів та годин у певний період.

18. Клас `ChooseMonths` дозволяє співставляти введені користувачем місяці у вікнах конкретизації періоду відображення інформації із їхніми числовими представленнями (від 1 до 12 для кожного з місяців відповідно календаря, 0 у випадку помилкового введення).

19. Клас `SearchByPeople` дозволяє здійснювати пошук співробітників за ПІБ/посадами шляхом порівняння обраних користувачем значень, які надані у вигляді випадаючого списку, із таблицею, отриманою з БД (див. Лістинг 14).

Лістинг 14 — Методи класу, які відповідають за пошук співробітників

```

public SearchByPeople(HeadForm headForm,
DataGridViewUniqueValues data)
{
    InitializeComponent();
    this.headForm = headForm;
    this.data = data;
    comboBoxPositions.Items.Add("-Нічого не об-
рано-");
    comboBoxPositions.Items.AddRange(data.UniquePositions.ToArray());
    comboBoxPositions.SelectedIndex = 0;
    comboBoxSurname.Items.Add("-Нічого не об-
рано-");
    comboBoxSurname.Items.AddRange(data.UniqueLastNames.ToArray());
    comboBoxSurname.SelectedIndex = 0;
    comboBoxName.Items.Add("-Нічого не обрано-");
};

```



```

comboBoxName.Items.AddRange(data.UniqueFirstNames.ToArray()
);
        comboBoxName.SelectedIndex = 0;
        comboBoxPatronymic.Items.Add("-Нічого не
обрано-");
comboBoxPatronymic.Items.AddRange(data.UniquePatronymics.To
Array());
        comboBoxPatronymic.SelectedIndex = 0;
    }
    public Dictionary<string, string>
GetSelectedValues()
    {
        Dictionary<string, string> selectedValues =
new Dictionary<string, string>();
        selectedValues["Position"] =
comboBoxPositions.SelectedItem.ToString();
        selectedValues["Surname"] =
comboBoxSurname.SelectedItem.ToString();
        selectedValues["Name"] =
comboBoxName.SelectedItem.ToString();
        selectedValues["Patronymic"] =
comboBoxPatronymic.SelectedItem.ToString();

        return selectedValues;
    }

```

20. Клас взаємодії `TaxRates` зберігає в собі інформацію про податкові ставки, за якими здійснюється нарахування (див. Лістинг 15). У разі збільшення інформаційної системи та подальшого її впровадження буде доцільним створення окремої таблиці у БД із зазначенням податкової ставки та термінів її дії (за аналогією з класом `HourlyRate` з п.9 цього підрозділу).

Лістинг 15 — Клас із визначеними податковими ставками

```

public static class TaxRates
{
    public const double ESVRate = 0.22;
    public const double PDFORate = 0.18;
    public const double VZRate = 0.015;
}

```

21. Клас `ExportBridge` служить для підготовки до експорту інформації. Основний його функціонал полягає в отриманні папки призначення (див. Лістинг 16).

Лістинг 16 — Функціонал, необхідний для отримання папки призначення

```

public void ExelExport(DataGridView dG)
    {
        try
        {
            SaveFileDialog openFileDialog = new
SaveFileDialog
            {
                InitialDirectory = "c:\\",
                Filter = "Excel files
(*.xlsx)|*.xlsx|Excel 2007 (*.xls)|*.xls",
                FilterIndex = 1
            };

            if (openFileDialog.ShowDialog() ==
DialogResult.OK)
            {
                DataTable dt =
ExportExcel.DataGridView_To_Datatable(dG);
                dt.exportToExcel(openFileDialog.FileName);
                ExelExport(openFileDialog.FileName);
            }
            catch (Exception ex) {
                MessageBox.Show(ex.Message);
            }
        }

        private void ExelExport(string path)
        {
            DialogResult result = MessageBox.Show("Ба-
жаєте відкрити збережений файл?", "Підтвердження",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);

            if (result == DialogResult.Yes)
            {
                try
                {

```



```

        }
        else
        {
            bufferRow[i] =
DBNull.Value;
        }
    }

bufferTable.Rows.Add(bufferRow);
}
}
dataGridView.DataSource = bufferTable;
}
}
}

```

23. Клас `ExportExcel` призначений для безпосереднього експорту даних `excel`-форматі (див. Лістинг 18).

Лістинг 18 — Функціонал, призначений для експорту у `excel`

```

public static void exportToExcel(this
System.Data.DataTable DataTable, string ExcelFilePath =
null)
{
    try
    {
        int ColumnsCount;
        if (DataTable == null || (ColumnsCount
= DataTable.Columns.Count) == 0)
            throw new Exception("ExportToExcel:
Null or empty input table!\n");

        Microsoft.Office.Interop.Excel.Application Excel = new
Microsoft.Office.Interop.Excel.Application();

        Excel.Workbooks.Add();

        Microsoft.Office.Interop.Excel._Worksheet Worksheet =
Excel.ActiveSheet;

        object[] Header = new
object[ColumnsCount];
        for (int i = 0; i < ColumnsCount; i++)
            Header[i] =
DataTable.Columns[i].ColumnName;
        Microsoft.Office.Interop.Excel.Range
HeaderRange =

```

```

Worksheet.get_Range((Microsoft.Office.Interop.Excel.Range) (
Worksheet.Cells[1,
    1]),
(Microsoft.Office.Interop.Excel.Range) (Worksheet.Cells[1,
ColumnsCount]));
    HeaderRange.Value = Header;
    HeaderRange.Interior.Color =
System.Drawing.ColorTranslator.ToOle(System.Drawing.Color.L
ightGray);
    HeaderRange.Font.Bold = true;
    int RowsCount = DataTable.Rows.Count;
    object[,] Cells = new object[RowsCount,
ColumnsCount];
    for (int j = 0; j < RowsCount; j++)
        for (int i = 0; i < ColumnsCount;
i++)
            Cells[j, i] =
DataTable.Rows[j][i];

Worksheet.get_Range((Microsoft.Office.Interop.Excel.Range) (
Worksheet.Cells[2, 1]),
    (Microsoft.Office.Interop.Excel.Range) (Worksheet.Cells[
RowsCount + 1, ColumnsCount])).Value = Cells;
    if (ExcelFilePath != null &&
ExcelFilePath != "")
    {
        try
        {
Worksheet.SaveAs(ExcelFilePath);

            Excel.Quit();
        }
        catch (Exception ex)
        {
            throw new
Exception("ExportToExcel: Excel file could not be saved!
Check filepath.\n"
            + ex.Message);
        }
    }
    else // no filepath is given
    {
        Excel.Visible = true;
    }
}
catch (Exception ex)
{

```

```

        throw new Exception("ExportToExcel: \n"
+ ex.Message);
    }
}

```

24. Клас `DataGridViewUniqueValues` зберігає списку унікальних значень для здійснення пошуку (див. Лістинг 19)

Лістинг 19 — Списки унікальних значень, за якими буде здійснюватися пошук

```

public class DataGridViewUniqueValues
{
    public List<string> UniquePositions { get; set;
}
    public List<string> UniqueLastNames { get; set;
}
    public List<string> UniqueFirstNames { get;
set; }
    public List<string> UniquePatronymics { get;
set; }
}

```

3.4.3 Розробка інтерфейсу застосунку

Через те, що для розробки інформаційної системи керування підприємством були обрані Windows Forms, C# а база даних розгорнута на Microsoft SQL Server — інтерфейс програмного застосунку буде виконано у класичному бізнес-стилі програм від корпорації Microsoft.

Першим функціональним вікном, з яким буде взаємодіяти користувач, призначене для його автентифікації у інформаційній системі для керування підприємством (див. Рис. 3.4).

Вона складається з двох елементів `TextBox`. Перше призначене для логіну, друге — для паролю (введена інформація приховується символом ‘*’).

Окрім цього, є кнопки для входу до системи та закриття програмного застосунку.

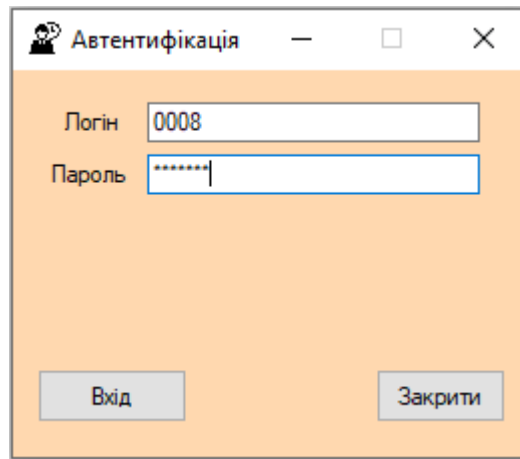


Рисунок 3.4 — Вікно для автентифікації у інформаційній системі

У разі коректного введення інформації відкривається головне вікно (див. Рис. 3.5). Воно складається з компонента MenuStrip (на якому розташовані кнопки «Інформація про застосунок» та «Вийти з профілю») та окремих 10 кнопок, які відкривають вікна роботи з таблицями.

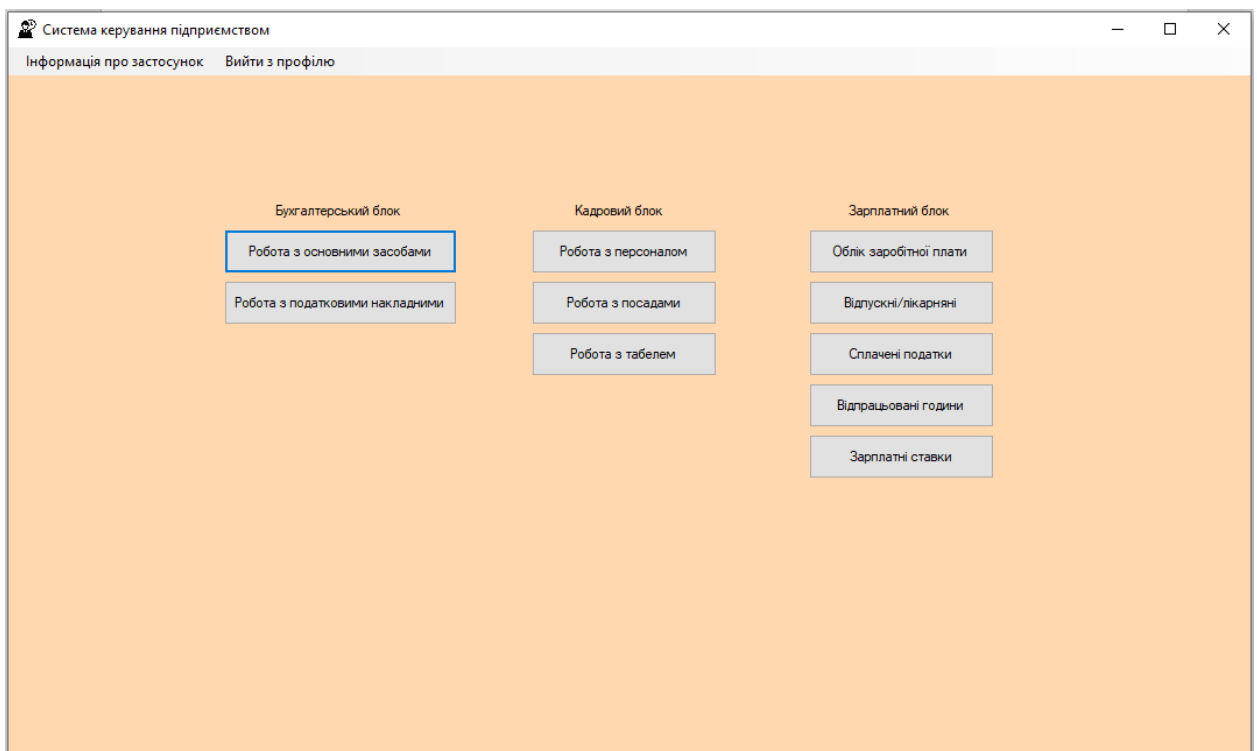


Рисунок 3.5 — Головне вікно системи керування підприємством

1. «Інформація про застосунок». При натисканні спливає вікно, у якому міститься інформація про розроблювальний програмний продукт (див. Рис. 3.6).

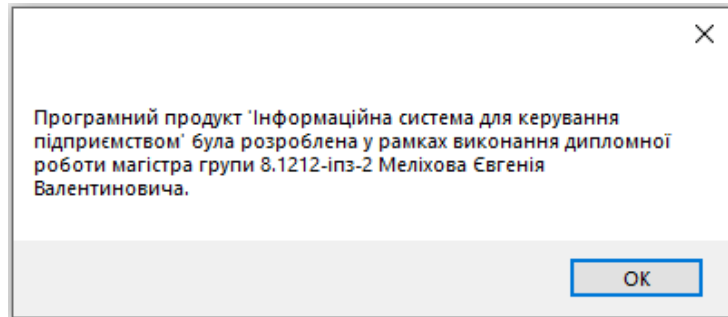


Рисунок 3.6 — Вікно з інформацією про застосунок

2. «Вийти з профілю». Ця кнопка присутня на кожному вікні, окрім автентифікації. Взаємодія з нею призведе до закриття всіх вікон та повернення до першого вікна.

Наступні функції можна розділити на 3 логічні блоки: бухгалтерський облік, кадровий блок та зарплатний блок.

Бухгалтерський блок складається з двох кнопок:

1. «Робота з основними засобами». При натисканні відкривається вікно роботи з основними засобами.

2. «Робота з податковими накладними». Взаємодія з цією кнопкою викликає форму для податкових накладних.

Кадровий блок складається з трьох кнопок:

1. «Робота з персоналом». При натисканні відкривається вікно роботи з персоналом.

2. «Робота з посадами». Взаємодія з цією кнопкою викликає форму для роботи з посадами.

3. «Робота з табелем». При натисканні відкривається вікно роботи з табелем персоналу.

Зарплатний блок вже містить у собі п'ять кнопок:

1. «Облік заробітної плати». При натисканні відкривається вікно роботи із заробітною платою.
2. «Сплачені податки». Призначені для висвітлювання даних стосовно нарахованих податків на заробітні плати співробітників та їхні відпустки та лікарняні.
3. «Відпрацьовані години». При натисканні відкривається вікно з інформацією про відпрацьовані персоналом години на підприємстві.
4. «Зарплатні ставки». Надає інформацію про ставки погодинної оплати праці та дати їх застосування до відповідних нарахунків.
5. «Відпускні/лікарняні». При натисканні відкривається вікно з інформацією про оплату відпусток та лікарняних.

Наступне вікно відповідає за роботу з основними засобами (див. Рис. 3.7).

Тип	код	назва	Вартість	Дата введення	Термін
Акумулятор	00-000165	Акумуляторна батарея Varta	1495,00	01.11.2013	24
Акумулятор	00-000166	Акумуляторна батарея Varta	1495,00	01.11.2022	24
Акумулятор	00-000167	Акумуляторна батарея Varta	1495,00	01.11.2022	24
Єврокуб	00-000236	Єврокуб 1000л	2200,00	22.11.2022	36
Єврокуб	00-000237	Єврокуб 1000л	2200,00	22.11.2022	36
Єврокуб	00-000238	Єврокуб 1000л	2200,00	22.11.2022	36
Єврокуб	00-000239	Єврокуб 1000л	2200,00	22.11.2022	36
Єврокуб	00-000240	Єврокуб 1000л	2200,00	22.11.2022	36
Єврокуб	00-000241	Єврокуб 1000л	2200,00	22.11.2022	36
Дриль	00-000406	Дриль ударна Ddnipro-m PB-120P	3789,00	01.12.2022	48
Дриль	00-000407	Дриль ударна Ddnipro-m PB-120P	3789,00	01.12.2022	48
Дриль	00-000408	Дриль ударна Ddnipro-m PB-120P	3789,00	01.12.2022	48
Зварювальний апарат IGBT	00-000489	Зварювальний апарат IGBT Dnipro-M SAB-14D MINI	5675,00	03.12.2022	48
Клапан запобіжний	00-000494	Клапан запобіжнийЗА-ЗНС-231	586,00	04.12.2022	36
Клапан запобіжний	00-000495	Клапан запобіжнийЗА-ЗНС-232	586,00	04.12.2022	36

Рисунок 3.7 — Вікно для роботи з основними засобами

Вікно має 6 функціональних кнопок, розташованих на MenuStrip (деякі з них об'єднані у групи).

1. «Головна». Повертає користувача на головне вікно.
2. «Повернутися». В залежності від поточного стану програми або повертає користувача на головне вікно, або скидає всі зміни, здійснені на цій формі.
3. «Аналітика → Закінчився термін експлуатації». Робить вибірку, виводячи на екран записи, де термін експлуатації перевищує дату введення до експлуатації.
4. «Аналітика → Придатні до використання». Робить вибірку, виводячи на екран записи, де термін експлуатації ще є меншим за дату введення.
5. «Експорт → Excel». Експортує таблицю у форматі Excel до місця, яке вказує користувач.
6. «Вийти з профілю». Описано вище.

Окрім цього, записи можна змінювати, додавати та видаляти натисканням на них. А у разі, коли необхідно вивести на екран декілька записів із схожими параметрами — існує можливість проведення фільтрації. Для цього необхідно двічі натиснути на параметр, за яким необхідно зробити вибірку.

Наступне вікно відповідає за роботу з податковими документами (див. Рис.3.8). Форма має той самий функціонал, як і попередня, за виключенням: відсутня можливість аналізу але наявна можливість відкривати та завантажувати документи.

Шляхи до них зберігаються у стовпчику із назвою «Шлях до документу». Якщо користувачу необхідно відкрити певний документ, він може двічі натиснути по стовпчику Id.

А при натисканні на кнопку «Завантажити» у MenuStrip викликається вікно, у якому необхідно вказати шлях до папки, у яку вивантажаться обрани документи.

Система керування підприємством (Податкові документи)

Головна Повернутися Завантажити Експорт Вийти з профілю

1 для 16

	Id	Ім'я контрагента	ІПН контрагента	Тип документу	Код документу	Назва документу
▶	1	ТОВ "ВелБуд"	2643803415	Накладна	VNPALW2C	Видаткова накладна ТОВ "ВелБуд" на сир
	2	ТОВ "Будідея"	3845709126	Рахунок	RKH34567	Рахунок для ТОВ "Будідея"
	3	ТОВ "Будівельник-Профі"	5901832467	Накладна	INV78901	Видаткова накладна для ТОВ "Будівельни
	4	ТОВ "МайстерГрад"	7650981234	Договір	CNTR23456	Договір з ТОВ "МайстерГрад" на будівни
	5	ТОВ "ЕнергоБуд"	9021345678	Специфікація	SPC78901	Специфікація для ТОВ "ЕнергоБуд"
	6	ТОВ "АрхіТек"	6512034879	ВнутрішняНакладна	INT12345	Внутрішня накладна для ТОВ "АрхіТек"
	7	ТОВ "БетонБуд"	4789562103	ІншаНакладна	OTH67890	Інша накладна для ТОВ "БетонБуд"
	8	ТОВ "СтальБуд"	1238904567	Договір	CNTR34567	Договір з ТОВ "СтальБуд" на постачання
	9	ТОВ "ЕкоБуд"	8765432190	Рахунок	RKH78901	Рахунок для ТОВ "ЕкоБуд"
	10	ТОВ "ДекорБуд"	2341098576	Накладна	INV23456	Видаткова накладна для ТОВ "ДекорБуд"
	11	ТОВ "ІнжБуд"	8907654321	Специфікація	SPC56789	Специфікація для ТОВ "ІнжБуд"
	12	ТОВ "ТеплоБуд"	3456712098	ІншаНакладна	OTH12345	Інша накладна для ТОВ "ТеплоБуд"
	13	ТОВ "Будівельний Альянс"	6789453210	Рахунок	RKH67890	Рахунок для ТОВ "Будівельний Альянс"
	14	ТОВ "ЕлектроБуд"	4567890123	Договір	CNTR90123	Договір з ТОВ "ЕлектроБуд" на електром
	15	ТОВ "АрхіСтрой"	1098765432	Накладна	INV56789	Видаткова накладна для ТОВ "АрхіСтрой"

Рисунок 3.8 — Вікно для роботи з податковими документами

Наступні три вікна призначені для роботи із кадровим блоком. Перше з них необхідно для взаємодії з інформацією про співробітників (див. Рис. 3.9).

Система керування підприємством (Співробітники)

Головна Повернутися Експорт Вийти з профілю

1 для 17

	Id	Прізвище	Ім'я	По батькові	Дата народження	Стать	ІПН
▶	1	Петренко	Іван	Миколайович	15.05.1990	Чоловіча	3645678901
	2	Коваленко	Олена	Василівна	22.09.1985	Жіноча	2890123456
	3	Сидоренко	Микола	Петрович	10.03.1992	Чоловіча	3623456789
	4	Богданова	Оксана	Володимирівна	08.07.1994	Жіноча	2834567890
	5	Мельник	Іван	Тарасович	25.11.1988	Чоловіча	3690123458
	6	Андрієнко	Софія	Олександрівна	18.02.1991	Жіноча	2856789012
	7	Шевченко	Олександр	Іванович	03.06.1993	Чоловіча	3678901234
	8	Ковальчук	Анна	Сергіївна	12.12.1987	Жіноча	2845678901
	9	Григоренко	Павло	Олегович	29.09.1996	Чоловіча	3612098745
	10	Денисенко	Юлія	Вікторівна	05.08.1993	Жіноча	2834567890
	11	Литвиненко	Артем	Сергійович	18.12.1990	Чоловіча	3658901234
	12	Павленко	Марія	Андріївна	02.05.1986	Жіноча	2812345670
	13	Гордієнко	Олег	Ігорович	14.02.1995	Чоловіча	3690123458
	14	Сергієнко	Тетяна	Володимирівна	23.07.1997	Жіноча	3658712098
	15	Карпенко	Денис	Петрович	30.11.1989	Чоловіча	2845678901
	16	Смирнов	Володимир	Петрович	31.10.1993	Чоловіча	2864758901

Рисунок 3.9 — Вікно для роботи із співробітниками

Друге вікно блоку для роботи з кадрами фірми зберігає перелік посад, які є на підприємстві та можуть бути призначені працюючим на ньому людям (див. Рис. 3.10).

Третє вікно необхідно для роботи із табелем співробітників. Він поєднує у собі персональну інформацію про співробітників та посади, які вони займають на підприємстві, включаючи дату призначення та звільнення (див. Рис. 3.11).

До того ж, присутня функція аналізу з метою виявлення у таблиці тих співробітників, які були звільнені (тих з них, що в поточний момент часу працюють, можна вивести, якщо зробити вибірку по строкам, у яких дата зняття з посади — пуста).

Наступним логічним блоком є робота із заробітними платами співробітників. Перша кнопка відповідає за роботу із заробітною платою. Після її натискання користувач бачить вікно для вибору місяця та року відображення даних (див. Рис. 3.12).

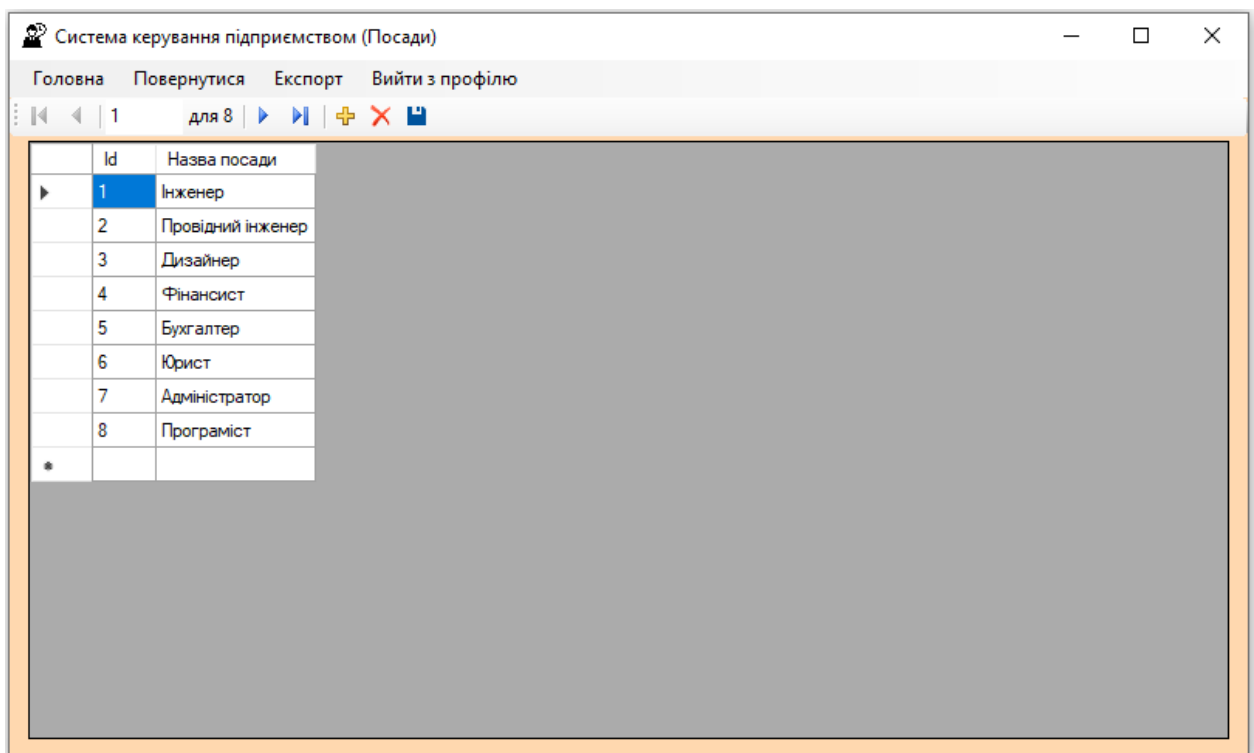


Рисунок 3.10 — Вікно для роботи із посадами

Система керування підприємством (Табель співробітників)

Головна Повернутися Аналіз Експорт Вийти з профілю

1 для 16

	Id	Посада	Прізвище	Ім'я	По батькові	Дата призначення	Дата зняття
▶	1	Інженер	Петренко	Іван	Миколайович	01.01.2021	
	2	Провідний інженер	Коваленко	Олена	Василівна	01.02.2021	
	3	Дизайнер	Сидоренко	Микола	Петрович	01.03.2021	
	4	Фінансист	Богданова	Оксана	Володимирівна	01.04.2021	
	5	Бухгалтер	Мельник	Іван	Тарасович	01.05.2021	
	6	Юрист	Андрієнко	Софія	Олександрівна	01.06.2021	
	7	Адміністратор	Шевченко	Олександр	Іванович	01.07.2021	01.06.2023
	8	Програміст	Ковальчук	Анна	Сергіївна	01.08.2021	
	9	Інженер	Григоренко	Павло	Олегович	01.09.2021	01.06.2023
	10	Провідний інженер	Денисенко	Юлія	Вікторівна	01.10.2021	
	11	Інженер	Литвиненко	Артем	Сергійович	01.01.2021	
	12	Провідний інженер	Павленко	Марія	Андріївна	01.02.2021	
	13	Дизайнер	Гордієнко	Олег	Ігорович	01.03.2021	
	14	Фінансист	Сергієнко	Тетяна	Володимирівна	01.04.2021	
	15	Бухгалтер	Карпенко	Денис	Петрович	01.05.2021	
	16	Юрист	Смирнов	Володимир	Петрович	01.06.2021	

Рисунок 3.11 — Вікно для роботи із таблицею співробітників

Система керування підприємством (Заробітна плата)

Оберіть місяць та рік відображення заробітної плати

Місяць Рік

Листопад 2023

Відкрити за весь час Розрахувати за весь час

Відкрити Розрахувати

Повернутися

Рисунок 3.12 — Вікно для вибору місяця та року заробітної плати

Вікно складається із 5 кнопок:

- **Відкрити за весь час.** У разі її натискання відкривається вікно із таблицею нарахованої заробітної плати за весь час роботи інформаційної системи.

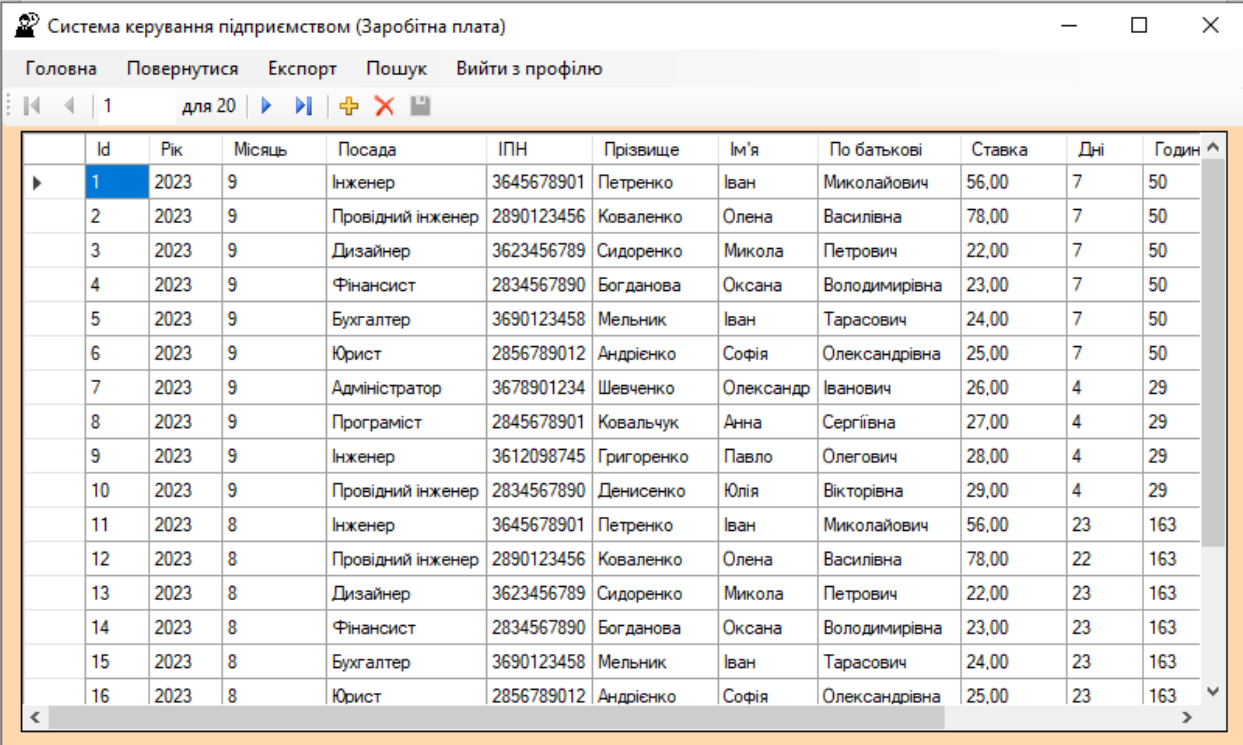
- **Відкрити.** В цьому випадку вікно заробітної плати відкривається лише за той період, який користувач обрав за допомогою двох випадючих списків.

- **Розрахувати за весь час.** У разі її натискання, перед завантаженням вікна заробітної плати, буде проведено розрахунок всіх заробітних плат за час роботи інформаційної системи (включаючи розрахунки податків, які зберігаються у окремій таблиці).

- **Розрахувати.** В цьому випадку інформаційна система зробить розрахунок лише за визначений користувачем період часу.

- **Повернутись.** Ця функція повертає користувача до Головного вікна.

У випадку натискання будь-якої кнопки на формі, окрім останньої, відкривається вікно з інформацією про заробітні плати співробітників підприємства (див. Рис. 3.13).



Id	Рік	Місяць	Посада	ІПН	Прізвище	Ім'я	По батькові	Ставка	Дні	Годин
1	2023	9	Інженер	3645678901	Петренко	Іван	Миколайович	56,00	7	50
2	2023	9	Провідний інженер	2890123456	Коваленко	Олена	Василівна	78,00	7	50
3	2023	9	Дизайнер	3623456789	Сидоренко	Микола	Петрович	22,00	7	50
4	2023	9	Фінансист	2834567890	Богданова	Оксана	Володимирівна	23,00	7	50
5	2023	9	Бухгалтер	3690123458	Мельник	Іван	Тарасович	24,00	7	50
6	2023	9	Юрист	2856789012	Андрієнко	Софія	Олександрівна	25,00	7	50
7	2023	9	Адміністратор	3678901234	Шевченко	Олександр	Іванович	26,00	4	29
8	2023	9	Програміст	2845678901	Ковальчук	Анна	Сергіївна	27,00	4	29
9	2023	9	Інженер	3612098745	Григоренко	Павло	Олегович	28,00	4	29
10	2023	9	Провідний інженер	2834567890	Денисенко	Юлія	Вікторівна	29,00	4	29
11	2023	8	Інженер	3645678901	Петренко	Іван	Миколайович	56,00	23	163
12	2023	8	Провідний інженер	2890123456	Коваленко	Олена	Василівна	78,00	22	163
13	2023	8	Дизайнер	3623456789	Сидоренко	Микола	Петрович	22,00	23	163
14	2023	8	Фінансист	2834567890	Богданова	Оксана	Володимирівна	23,00	23	163
15	2023	8	Бухгалтер	3690123458	Мельник	Іван	Тарасович	24,00	23	163
16	2023	8	Юрист	2856789012	Андрієнко	Софія	Олександрівна	25,00	23	163

Рисунок 3.13 — Вікно з інформацією стосовна заробітної плати

Воно надає користувачу інформацію про рік та місяць, за які нарахована заробітна плата, посаду та ПІБ співробітника, який посадовий оклад йому

призначено, скільки днів та годин йому зараховано та суми до утримання податків і фактично виплачену. Є можливість здійснювати вибірку, експортувати дані у певний формат та проводити пошук працівників за параметрами (див. Рис. 3.14).

The image shows a software window titled "Система керування підприємством (Пошук)". Inside the window, there is a heading "Оберіть бажані параметри". Below this heading, there are four dropdown menus arranged in a 2x2 grid. The top-left dropdown is labeled "Посада" and the top-right is labeled "Прізвище". The bottom-left dropdown is labeled "Ім'я" and the bottom-right is labeled "По батькові". All four dropdown menus currently display the text "-Нічого не обрано-". At the bottom of the window, there are two buttons: "Пошук" on the left and "Повернутися" on the right.

Рисунок 3.14 — Вікно пошуку співробітників

Користувач має можливість шукати по 4 параметрах: посада, прізвище, ім'я, по батькові. Обирати необхідно певне значення із випадючих списків. Вони формуються на підставі даних, які наведені у попередній формі за період, який обрав користувач. Причому, можна здійснювати пошук у будь-яких варіаціях (наприклад, шукати лише посаду та прізвище, або лише ім'я тощо). Після обирання пошукової комбінації та натискання кнопки «Пошук» відкривається вікно із результатами.

Наступною можливістю інформаційної системи є перегляд інформації про лікарняні та відпустки. Для цього обирається бажана кнопка на Головній формі, після чого відкривається вікно для вибору періоду відображення даних (див. Рис. 3.15). Користувач, як і із вікном заробітної плати, може здійснити розрахування лікарняних/відпусток або відкрити вже наявні. До того ж, для обох з цих можливостей доступні два варіанти: або відкриття за весь час роботи інформаційної системи, або лише за конкретний місяць.

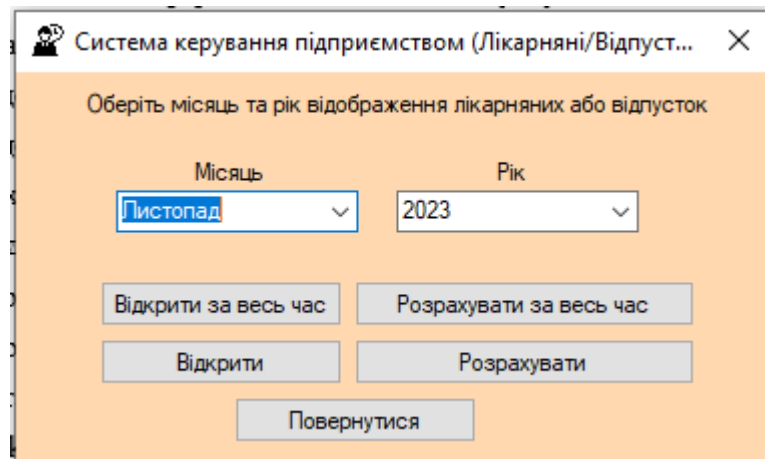


Рисунок 3.15 — Вікно вибору періоду відображення відпусток/лікарняних

Будь яка дія, окрім натискання кнопки «Повернутись», відкриває вікно із відображенням лікарняних/відпусток (див. Рис. 3.16).

Id	Посада	Прізвище	Ім'я	По батькові	Тип	Ставка	Початок	Кінець	Дні	Кс
1	Фінансист	Богданова	Оксана	Володимирівна	Лікарняна	63,00	24.08.2023	30.08.2023	7	0,6
2	Фінансист	Богданова	Оксана	Володимирівна	Відпустка	63,00	28.10.2023	31.10.2023	4	0,6
3	Інженер	Петренко	Іван	Миколайович	Відпустка	56,00	01.11.2023	30.11.2023	30	1,0
4	Провідний інженер	Коваленко	Олена	Василівна	Відпустка	78,00	01.11.2023	30.11.2023	30	1,0
5	Дизайнер	Сидоренко	Микола	Петрович	Відпустка	62,00	01.11.2023	30.11.2023	30	1,0
6	Фінансист	Богданова	Оксана	Володимирівна	Відпустка	63,00	01.11.2023	30.11.2023	30	0,6
7	Бухгалтер	Мельник	Іван	Тарасович	Відпустка	64,00	01.11.2023	30.11.2023	30	1,0
8	Юрист	Андрієнко	Софія	Олександрівна	Відпустка	65,00	01.11.2023	30.11.2023	30	1,0
9	Адміністратор	Шевченко	Олександр	Іванович	Відпустка	66,00	01.11.2023	30.11.2023	30	0,6
10	Програміст	Ковальчук	Анна	Сергіївна	Відпустка	250,00	01.11.2023	30.11.2023	30	1,0
11	Інженер	Григоренко	Павло	Олегович	Відпустка	65,00	01.11.2023	30.11.2023	30	1,0
12	Провідний інженер	Денисенко	Юлія	Вікторівна	Відпустка	112,00	01.11.2023	30.11.2023	30	1,0
13	Інженер	Литвиненко	Артем	Сергійович	Відпустка	65,00	01.11.2023	30.11.2023	30	1,0
14	Провідний інженер	Павленко	Марія	Андріївна	Відпустка	74,00	01.11.2023	30.11.2023	30	1,0
15	Дизайнер	Гордієнко	Олег	Ігорович	Відпустка	64,00	01.11.2023	30.11.2023	30	1,0
16	Фінансист	Сергієнко	Тетяна	Володимирівна	Відпустка	87,00	01.11.2023	30.11.2023	30	1,0

Рисунок 3.16 — Вікно відображення відпускових та лікарняних

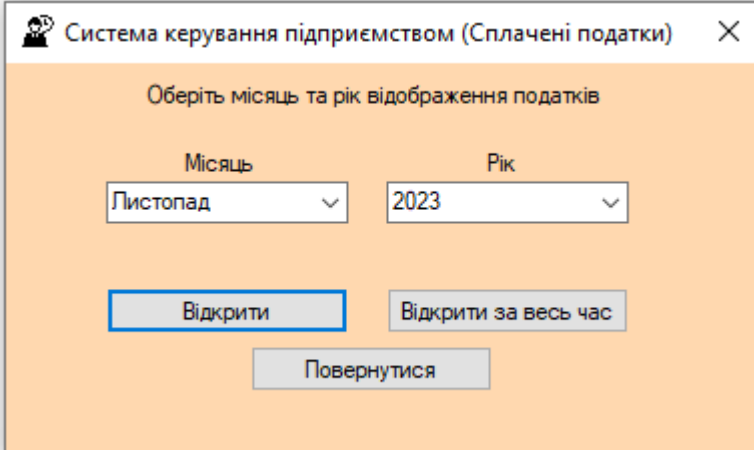
Наповнення дуже схоже на вікно відображення заробітної плати. Відрізняються вони лише механізмом розрахунків, а саме: для обчислення суми береться до уваги кількість днів та зараховується по 8 годин до кожного. Після цього рахуєма сума помножується на коефіцієнт, який встановлено для

співробітника (чим менше страхового стажу — тим менше відсоток виплати). Також є можливості: пошуку, експортування та здійснення вибірки.

І з заробітної плати, і з відпускних/лікарняних, сплачуються податки, які теж можна переглянути за допомогою інформаційної системи (див. Рис. 3.17).

На відміну від двох попередніх вікон вибору періоду, ця не має функції розрахунків, адже всі дані до неї вносяться під час нарахування сум для співробітників. Тому тут можна лише обрати період відображення (за певний місяць або всі дані).

Натискання на кнопки «Відкрити» або «Відкрити за весь час» приводять до виклику вікна роботи із податками (див. Рис. 3.18). Воно надає користувачу інформацію стосовно: періоду сплати податку, посади та ПІБ співробітника, типу податку, його відсотку та безпосередньої суми платежу.



Система керування підприємством (Сплачені податки)

Оберіть місяць та рік відображення податків

Місяць Рік

Листопад 2023

Відкрити Відкрити за весь час

Повернутися

Рисунок 3.17 — Вікно вибору періоду відображення податків

Це вікно, як і попередні, надає функціонал для аналітичної вибірки інформації, експортування у довільний формат та пошуку за ПІБ та/або посадою співробітника.

Система керування підприємством (Сплатнені податки)

Головна Повернутися Експорт Пошук Вийти з профілю

1 для 54

	Рік	Місяць	Посада	ІПН	Прізвище	Ім'я	По батькові	Тип податку	Відсоток	Сум
▶	2023	8	Фінансист	2834567890	Богданова	Оксана	Володимирівна	ЄСВ	0,22	465,
	2023	10	Фінансист	2834567890	Богданова	Оксана	Володимирівна	ЄСВ	0,22	266,
	2023	11	Інженер	3645678901	Петренко	Іван	Миколайович	ЄСВ	0,22	295,
	2023	11	Провідний інженер	2890123456	Коваленко	Олена	Василівна	ЄСВ	0,22	411,
	2023	11	Дизайнер	3623456789	Сидоренко	Микола	Петрович	ЄСВ	0,22	327,
	2023	11	Фінансист	2834567890	Богданова	Оксана	Володимирівна	ЄСВ	0,22	199,
	2023	11	Бухгалтер	3690123458	Мельник	Іван	Тарасович	ЄСВ	0,22	337,
	2023	11	Юрист	2856789012	Андрієнко	Софія	Олександрівна	ЄСВ	0,22	343,
	2023	11	Адміністратор	3678901234	Шевченко	Олександр	Іванович	ЄСВ	0,22	209,
	2023	11	Програміст	2845678901	Ковальчук	Анна	Сергіївна	ЄСВ	0,22	132,
	2023	11	Інженер	3612098745	Григоренко	Павло	Олегович	ЄСВ	0,22	343,
	2023	11	Провідний інженер	2834567890	Денисенко	Юлія	Вікторівна	ЄСВ	0,22	591,
	2023	11	Інженер	3658901234	Литвиненко	Артем	Сергійович	ЄСВ	0,22	343,
	2023	11	Провідний інженер	2812345670	Павленко	Марія	Андріївна	ЄСВ	0,22	390,
	2023	11	Дизайнер	3690123458	Гордієнко	Олег	Ігорович	ЄСВ	0,22	337,
	2023	11	Фінансист	3658712098	Сергієнко	Тетяна	Володимирівна	ЄСВ	0,22	459,

Рисунок 3.18 — Вікно роботи з податками

Передостання кнопка надає функціонал роботи із годинами, які відпрацював співробітник. Спочатку, як і із таблицею податків, відкривається відкно вибору періоду (див. Рис. 3.19).

Система керування підприємством (Відпрацьовані год...)

Оберіть місяць та рік відображення відпрацьованих годин

Місяць: Листопад

Рік: 2023

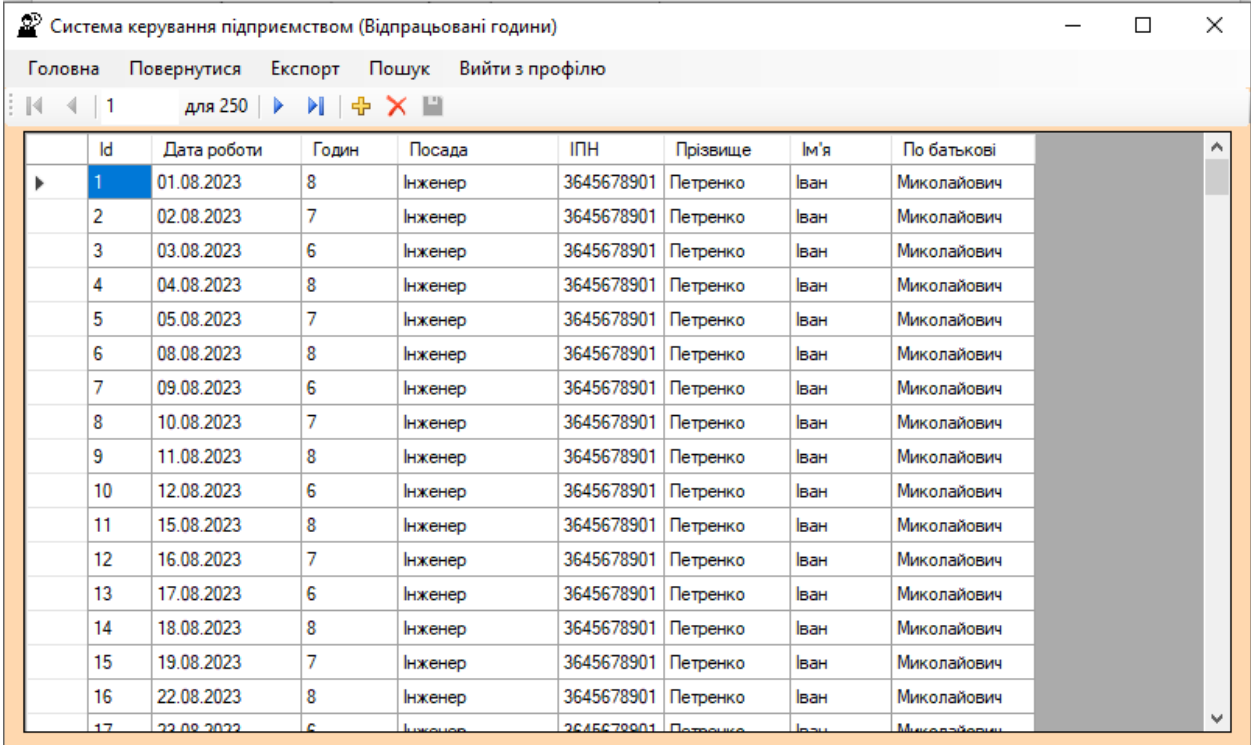
Відкрити

Відкрити за весь час

Повернутися

Рисунок 3.19 — Вікно вибору періоду для відпрацьованих годин

Після обирання періоду (місяць або вся наявна інформація), відкривається вікно для відображення даних про відпрацьовані години (див. Рис. 3.20).

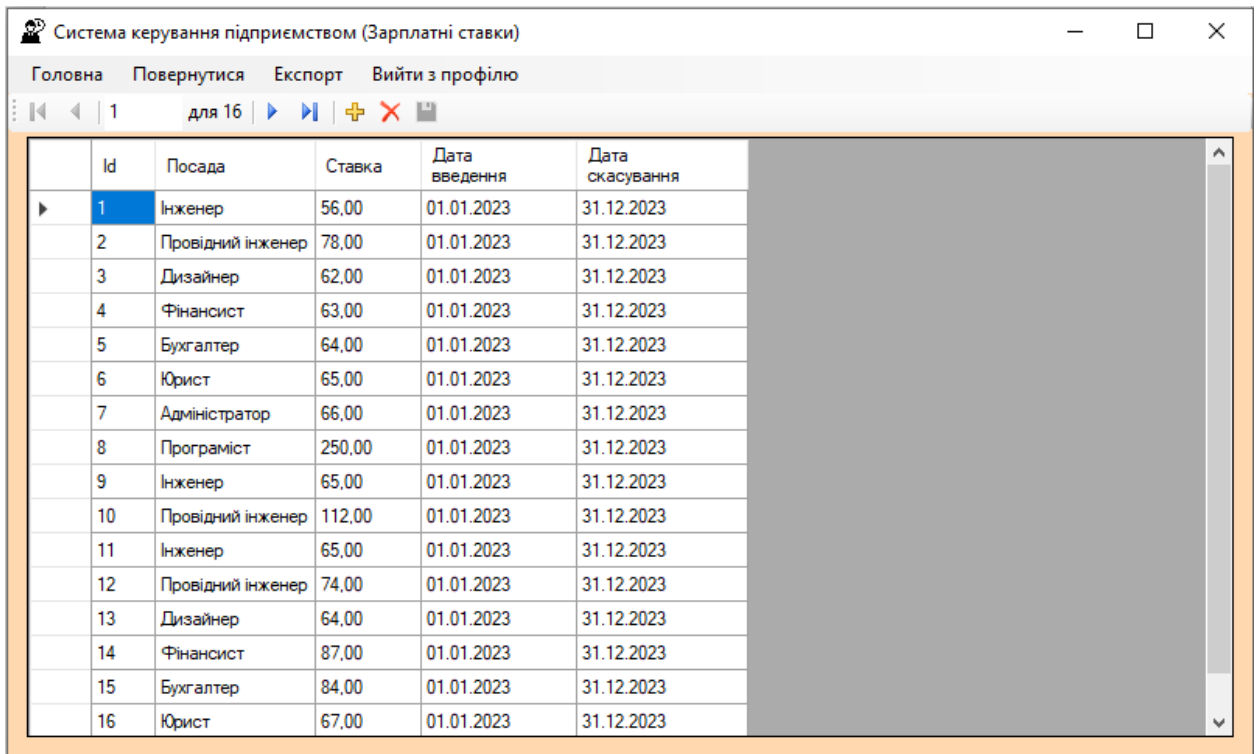


Id	Дата роботи	Годин	Посада	ІПН	Прізвище	Ім'я	По батькові
1	01.08.2023	8	Інженер	3645678901	Петренко	Іван	Миколайович
2	02.08.2023	7	Інженер	3645678901	Петренко	Іван	Миколайович
3	03.08.2023	6	Інженер	3645678901	Петренко	Іван	Миколайович
4	04.08.2023	8	Інженер	3645678901	Петренко	Іван	Миколайович
5	05.08.2023	7	Інженер	3645678901	Петренко	Іван	Миколайович
6	08.08.2023	8	Інженер	3645678901	Петренко	Іван	Миколайович
7	09.08.2023	6	Інженер	3645678901	Петренко	Іван	Миколайович
8	10.08.2023	7	Інженер	3645678901	Петренко	Іван	Миколайович
9	11.08.2023	8	Інженер	3645678901	Петренко	Іван	Миколайович
10	12.08.2023	6	Інженер	3645678901	Петренко	Іван	Миколайович
11	15.08.2023	8	Інженер	3645678901	Петренко	Іван	Миколайович
12	16.08.2023	7	Інженер	3645678901	Петренко	Іван	Миколайович
13	17.08.2023	6	Інженер	3645678901	Петренко	Іван	Миколайович
14	18.08.2023	8	Інженер	3645678901	Петренко	Іван	Миколайович
15	19.08.2023	7	Інженер	3645678901	Петренко	Іван	Миколайович
16	22.08.2023	8	Інженер	3645678901	Петренко	Іван	Миколайович
17	22.08.2023	6	Інженер	3645678901	Петренко	Іван	Миколайович

Рисунок 3.20 — Вікно відображення відпрацьованих годин

На ньому користувач може побачити конкретну дату, кількість годин, коли працював конкретний співробітник (про нього відображено наступну інформацію: посада, індивідуальний податковий номер, Прізвище, Ім'я та по батькові). Як і у всіх вікнах, в яких вказано ПІБ та/або посаду, є можливості здійснення пошуку, вибірки даних або проведення експорту.

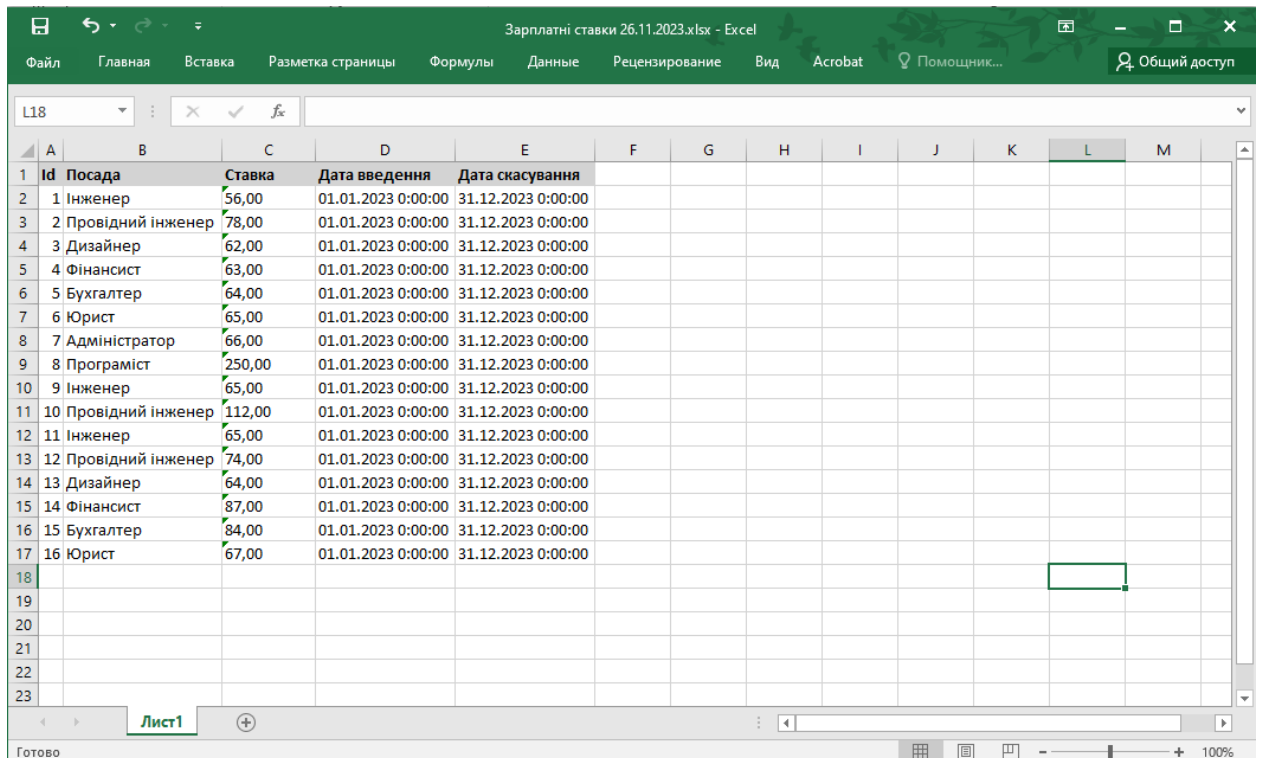
Останнє функціональне вікно інформаційної системи відповідає за відображення зарплатних ставок, які були впроваджені або досі діють на підприємстві (див. Рис. 3.21). Окрім самої суми у гривнях, зазначені асоційована посада та дати, у яких зазначені ставки можуть використовуватись.



	Id	Посада	Ставка	Дата введення	Дата скасування
▶	1	Інженер	56,00	01.01.2023	31.12.2023
	2	Провідний інженер	78,00	01.01.2023	31.12.2023
	3	Дизайнер	62,00	01.01.2023	31.12.2023
	4	Фінансист	63,00	01.01.2023	31.12.2023
	5	Бухгалтер	64,00	01.01.2023	31.12.2023
	6	Юрист	65,00	01.01.2023	31.12.2023
	7	Адміністратор	66,00	01.01.2023	31.12.2023
	8	Програміст	250,00	01.01.2023	31.12.2023
	9	Інженер	65,00	01.01.2023	31.12.2023
	10	Провідний інженер	112,00	01.01.2023	31.12.2023
	11	Інженер	65,00	01.01.2023	31.12.2023
	12	Провідний інженер	74,00	01.01.2023	31.12.2023
	13	Дизайнер	64,00	01.01.2023	31.12.2023
	14	Фінансист	87,00	01.01.2023	31.12.2023
	15	Бухгалтер	84,00	01.01.2023	31.12.2023
	16	Юрист	67,00	01.01.2023	31.12.2023

Рисунок 3.21 — Вікно з інформацією про ставки заробітної плати

Так само, як і більшість попередніх вікон, є можливість проведення фільтрації та експортування до excel (див. Рис. 3.22).



	Id	Посада	Ставка	Дата введення	Дата скасування
1	1	Інженер	56,00	01.01.2023 0:00:00	31.12.2023 0:00:00
2	2	Провідний інженер	78,00	01.01.2023 0:00:00	31.12.2023 0:00:00
3	3	Дизайнер	62,00	01.01.2023 0:00:00	31.12.2023 0:00:00
4	4	Фінансист	63,00	01.01.2023 0:00:00	31.12.2023 0:00:00
5	5	Бухгалтер	64,00	01.01.2023 0:00:00	31.12.2023 0:00:00
6	6	Юрист	65,00	01.01.2023 0:00:00	31.12.2023 0:00:00
7	7	Адміністратор	66,00	01.01.2023 0:00:00	31.12.2023 0:00:00
8	8	Програміст	250,00	01.01.2023 0:00:00	31.12.2023 0:00:00
9	9	Інженер	65,00	01.01.2023 0:00:00	31.12.2023 0:00:00
10	10	Провідний інженер	112,00	01.01.2023 0:00:00	31.12.2023 0:00:00
11	11	Інженер	65,00	01.01.2023 0:00:00	31.12.2023 0:00:00
12	12	Провідний інженер	74,00	01.01.2023 0:00:00	31.12.2023 0:00:00
13	13	Дизайнер	64,00	01.01.2023 0:00:00	31.12.2023 0:00:00
14	14	Фінансист	87,00	01.01.2023 0:00:00	31.12.2023 0:00:00
15	15	Бухгалтер	84,00	01.01.2023 0:00:00	31.12.2023 0:00:00
16	16	Юрист	67,00	01.01.2023 0:00:00	31.12.2023 0:00:00
18					
19					
20					
21					
22					
23					

Рисунок 3.22 — Вікно експорту зарплатних ставок

Отже, перед початком розробки інформаційної системи було отримане переконання у необхідності об'єднання найбільш бажаного для підприємства функціоналу до єдиної інформаційної системи.

Визначені вимоги до застосунку, задоволення яких надасть можливість впровадити його до діючих підприємств та побороти недоліки існуючих рішень.

Виконано розробку бази даних, структура якої дозволяє розширювати інформаційну систему. Було визначено механізм для з'єднання БД із розроблюваним застосунком.

Після цього була розроблена єдина інформаційна система для керування підприємством, яка задовільняє всім вимогам до програмного продукту, що підтверджується UseCase та Class діаграмами. До того ж, приведено лістинги найголовніших методів та елементів, а інтерфейс програмного застосунку описано в повному обсязі.

Забезпечено можливість масштабування функціоналу в майбутньому за допомогою модульної архітектури системи. Це дає змогу легко додавати нові функції та вдосконалювати існуючі, забезпечуючи підприємству гнучкість у використанні інформаційної системи.

Завершивши розробку інформаційної системи, проведено інтенсивне тестування для переконання у стабільності та надійності програмного продукту. Виявлені і виправлені помилки, що дозволило досягти високого рівня якості в роботі системи.

Узагальнюючи, розроблена інформаційна система становить ефективний інструмент для оптимізації управління підприємством, об'єднуючи в собі найбільш важливі функції та забезпечуючи стійкість, надійність і легкість в експлуатації.

4 ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ РОЗРОБКИ ЄДИНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ КЕРУВАННЯ ПІДПРИЄМСТВОМ

4.1 Підвищення надійності вносимої інформації

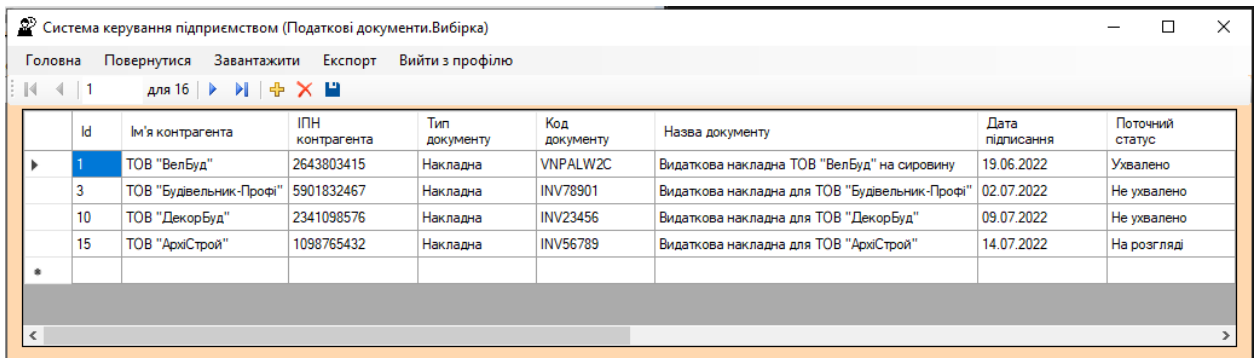
Як вже було зазначено у п.1.5, на підприємствах існують певні документи, інформація для заповнення яких зберігається у різних програмних застосунках. Таким прикладом є податкові накладні між поточною фірмою та її контрагентом.

Майже вся інформація про інше підприємство зберігається у застосунку BAS: Бухгалтерія для України, а саме: ім'я контрагента, його ППН, тип документу, його назва тощо. Але обмін із державними установами виконується через інформаційну платформу обміну інформацією застосунку М.Е.Дос. Таким чином, існуюче підприємство змушене експортувати ці дані шляхом формування певного документу, відвантажувати його до застосунку та власноруч дублювати певні поля, щоб відправити його на обробку до установи. До того ж, відсутні можливості проведення згруповань документів (окрім пошуку за параметрами).

Розроблена єдина інформаційна система керування підприємством зберігає всі зазначені параметри у власній базі даних. Звичайно, що можливість експортування інформації у різні формати існує. Але це знадобиться лише для відправлення до інших підприємств поштою або роздрукування на принтері.

Також дуже корисною є функція групування. Користувач може відкрити вікно керування податковими накладними, двічі натиснути на комірку «накладна», після чого інформаційна система продемонструє вичерпну інформацію стосовно його запиту (див. Рис. 4.1). Після цього, бухгалтер може завантажити архів із цими документами (наприклад, якщо йому потрібно надіслати на перевірку до контролюючого органу всі накладні із певними параметрами). Для цього, після відповідної фільтрації даних (для досягнення бажаного переліку),

він натискає кнопку «Завантажити» у верхньому меню, обирає бажану папку та отримує в неї всі документи, які були зазначені на екрані.



Id	Ім'я контрагента	ІПН контрагента	Тип документа	Код документа	Назва документа	Дата підписання	Поточний статус
1	ТОВ "ВелБуд"	2643803415	Накладна	VNPALW2C	Видаткова накладна ТОВ "ВелБуд" на сировину	19.06.2022	Ухвалено
3	ТОВ "Будівельник-Профі"	5901832467	Накладна	INV78901	Видаткова накладна для ТОВ "Будівельник-Профі"	02.07.2022	Не ухвалено
10	ТОВ "ДекорБуд"	2341098576	Накладна	INV23456	Видаткова накладна для ТОВ "ДекорБуд"	09.07.2022	Не ухвалено
15	ТОВ "АрхіСтрой"	1098765432	Накладна	INV56789	Видаткова накладна для ТОВ "АрхіСтрой"	14.07.2022	На розгляді

Рисунок 4.1 — Інформація про створені накладні

Таким чином, виключаються всі можливі механічні помилки при перенесенні інформації з одного застосунку до іншого, що значно підвищує цілісність даних та підвищує ефективність співробітників, адже більше немає потреби у кропіткому перенесенні інформації.

Подальший розвиток цієї переваги для інформаційної системи полягає у налагодження зв'язку із різними державними органами. В подальшому може бути розроблене API для інтеграції до електронних кабінетів користувачів податкової, пенсійного фонду, сектору обов'язкового державного страхування тощо. У нашій країні вже зараз існують схожі шаблони швидкої інтеграції. Наприклад, id.gov.ua, що представляє собою універсальну платформу для електронної автентифікації користувачів, яку використовують у якості альтернативи входу до системи та власної ідентифікації людини всі державні онлайн-кабінети.

Схоже API, у разі попиту, можна впровадити і до поточної інформаційної системи, що дуже значно підвищить попит на існуючому ринку програмних продуктів для керування підприємством.

4.2 Використання правил людинно-машинної взаємодії для визначення ефективності єдиної інформаційної системи

Подолати проблему механічних помилок при перенесенні інформації із одного застосунку до іншого вдалось подолати шляхом їхнього об'єднання у єдину інформаційну систему. Але залишається питання зручності зміни двох застосунків на один. Вивченням цього питання займається дослідний напрямок під назвою «Людино-машинна взаємодія».

Людино-машинна взаємодія (англ. human-computer interaction, HCI) — це науковий напрямок, який існує і розвивається з метою вдосконалення методів розробки, оцінки та впровадження інтерактивних комп'ютерних систем, призначених для використання людиною, а також для дослідження різних аспектів цього використання. Одним з найвідоміших методів цього дослідження є застосування закону, який вивів німецький дослідник Едуард Вільям Хік.

Закон Хіка – це експериментально підтверджена теорія, яка стверджує, що обсяг інформації, яка надходить до людського мозку, впливає на час, який витрачається на процес прийняття рішення. Цей закон дозволяє аналізувати діяльність оператора: розраховувати час, який йому потрібен для виконання конкретного завдання, визначати швидкість отримання ним інформації з урахуванням його психофізіологічних можливостей.

Закон Хіка виражається формулою 4.1, де: T - середнє значення часу реакції, n - число рівно можливих альтернативних сигналів, a і b - коефіцієнти пропорційності (у поточному прикладі дорівнюють 1).

$$T = a + b \times \log_2(n + 1) \quad (4.1)$$

Варіанти, які доступні користувачу при використанні об'єднаної інформаційної системи (варіант 1) та застосування на підприємствах старого варіанту із двома застосунками (варіант 2), наведені на таблиці 4.1.

Таблиця 4.1 — Два варіанта інформаційних систем

Варіант 1	Варіант 2
Єдина інформаційна система <ul style="list-style-type: none"> • Облік співробітників • Нарахування зарплати • Податкові накладні • Облік запасів 	BAS Бухгалтерія <ul style="list-style-type: none"> • Облік співробітників • Облік запасів • Нарахування зарплати M.E.Doc. Бізнес <ul style="list-style-type: none"> • Податкові накладні
$\text{Log}_2(4+1) = 2,322$	$\text{Log}_2(2+1) = 1,585$
$T = 1 + 1 * 2,322 = 3,322$	$T = 2 * (1 + 1 * 1,585) = 5,17$

Грунтуючись на даних, отриманих з табл. 4.1 (згідно Закону Хіка), можна стверджувати, що надання користувачу всього функціоналу одразу у одній інформаційній системі зазвичай є більш ефективним, ніж використання декількох програмних продуктів.

4.3 Використання продуктових метрик для визначення ефективності єдиної інформаційної системи

Окрім відсутності необхідності у перенесенні даних між різними застосунками, ефективність єдиної інформаційної системи можна довести за допомогою продуктових метрик.

Продуктові метрики - це вимірювання, які використовуються для оцінки ефективності та успішності продукту або сервісу. Вони допомагають компаніям отримувати об'єктивну інформацію про те, як користувачі взаємодіють з продуктом, його функціональністю та рівень якості обслуговування.

Продуктові метрики можуть бути різноманітними і залежать від конкретного продукту або сервісу.

1. Коефіцієнт утримання клієнтів (CRR, англ. Customer Retention Rate) визначає здатність компанії зберігати довгострокові відносини з клієнтами.

Високий рівень CRR свідчить про наявність достатньої кількості лояльних клієнтів, які забезпечують стабільний обсяг продажів для підприємства.

Низький рівень утримання клієнтів вказує на необхідність витрат на взаємодію з користувачами. У випадку неможливості утримати поточних клієнтів, виникає необхідність залучення нових, що завжди є витратнішим.

CRR розраховується шляхом ділення різниці між кількостями клієнтів системи на кінець періоду та новими, що прийшли за цей час, на початкове значення користувачів. Для підтримування оптимальної тенденції розвитку інформаційної системи кількість нових користувачів повинна щонайменше на 40% перевищувати тих клієнтів, які відмовились від користування.

Запропоновано акцентувати увагу на роботі з поточною базою клієнтів. Важливо пам'ятати, що кожен лояльний користувач програмного застосунку може стати безкоштовною рекламою інформаційної системи та сприятиме залученню нових клієнтів.

Дуже схожою на CRR є формула коефіцієнту віттоку клієнтів, проте високе значення коефіцієнту для неї є гіршими ознаками та призводить до негативних наслідків.

2. Коефіцієнт відттоку клієнтів (CR, англ. Churn rate) визначає рівень втрати клієнтів компанією протягом певного періоду. Високий показник Churn Rate свідчить про те, що компанія втрачає значну кількість клієнтів, що може впливати на стабільність її бізнесу.

Низький рівень відтоку свідчить про те, що компанія успішно утримує клієнтів та забезпечує їхню лояльність. Зменшення Churn Rate є важливим для забезпечення сталого обсягу клієнтської бази.

CR розраховується простіше за попередника: кількість клієнтів на початку періоду сумують з новими користувачами, після чого від отриманого значення віднімають поточну кількість активних учасників та помножують на 100%.

Прийнятний рівень показника залежить від розмірів фінального програмного застосунку та кількості часу після його оприлюднення:

- Якщо в поточний момент інформаційна система знаходиться на етапі первинного впровадження, допустимим показником є відток у 10% та більше.
- У випадку формування первинного кола користувачів та знаходження на ринку впродовж певного часу — показник не повинен перевищувати значення 3%.
- Коли мова буде йти вже про дуже великий інформаційний застосунок, серед клієнтів якого будуть великі промислові підприємства, значення CR повинно бути менше 1%.

З погляду стратегії управління, зниження Churn Rate може забезпечити ефективніше використання ресурсів, так як утримання існуючих клієнтів часто виявляється менш витратним, ніж залучення нових. Розуміння та моніторинг цього показника дозволяє компанії вчасно вживати необхідні заходи для утримання клієнтів та покращення їхнього задоволення єдиною інформаційною системою.

Наступним показником, який потребує аналізу, є значення середнього доходу на одного користувача.

3. Середній дохід на користувача (ARPU, англ. Average Revenue Per User) є ключовою метрикою в області бізнесу, що визначає середню суму доходу, яку компанія отримує від кожного активного користувача протягом конкретного періоду.

Високий показник ARPU свідчить про те, що компанія успішно генерує дохід від своєї клієнтської бази, що може бути наслідком ефективних стратегій ціноутворення, удосконалення продукту чи вищого рівня відмінності послуги.

Низький показник ARPU вказує на те, що компанія отримує менше доходу від кожного користувача за певний період. Це може бути результатом низьких цін на продукт чи послугу, неефективних стратегій ціноутворення або

недостатньої споживчої активності користувачів. Він може викликати турботу, оскільки це пряма тенденція до скорочення загального доходу компанії, особливо якщо витрати на утримання користувачів залишаються на занадто високому рівні. Зазвичай компанії стежать за збільшенням ARPU, щоб покращити ефективність своєї діяльності та збільшити прибуток від кожного клієнта.

Не дивлячись на велику важливість показника, його розрахунок є досить легким. Це відношення загального доходу за період (у чисельнику) до кількості користувачів за період (у знаменнику).

Аналіз ARPU є важливим для бізнес-планування та прийняття рішень. Збільшення цієї метрики може вказувати на можливість збільшення доходу на користь компанії, а також на більшу цінність продукту чи послуги для користувачів. Ця метрика також може служити показником ефективності маркетингових та продажових стратегій, а також якісного обслуговування клієнтів, що впливає на їхню готовність співпрацювати з компанією на тривалий термін.

4.4 Вартість розробки єдиної інформаційної системи

Для розрахунку вартості інформаційної системи разом з проектуванням та програмним продуктом необхідно підрахувати час, який було витрачено на цю роботу та його вартість.

Поточна версія Єдиної інформаційної системи для керування підприємством була розроблена приблизно за 25 робочих дні. У разі її виходу на ринок в поточному варіанті буде необхідно розробити супроводжувальну документацію, що займе ще приблизно 7 днів.

Робочі години, які були витрачені на розробку інформаційної системи, розглянуті більш детально у відповідній таблиці (див. Табл. 4.2).

Таблиця 4.2 — Час, витрачений на розробку інформаційної системи

№	Кількість годин	Виконана робота
1	7	Спілкування із бухгалтерами та керівниками підприємств для визначення програм, які використовуються на їхніх підприємствах, та виділення існуючих недоліків
2	18	Вивчення існуючих рішень та застосовуваних технологій у галузі розробки програмного забезпечення
3	22	Визначення вимог та функціоналу майбутнього застосунку на підставі отриманої інформації
4	8	Проектування Баз даних
5	7	Заповнення бази даних тестовою інформацією
6	9	Створення початкового програмного застосунку та його зв'язування з БД
7	20	Проектування алгоритмів роботи застосунку
8	4	Проектування інтерфейсу єдиної інформаційної системи
9	18	Створення візуальних форм у програмному застосунку
10	26	Розробка раніше спроектованих алгоритмів
11	18	Зв'язування візуальних форм та розроблених раніше алгоритмів
12	16	Формулювання та розробка запитів до бази даних
13	30	Тестування інформаційної системи та виправлення недоліків
14	56	Необхідний час для розробки супроводжувальної документації, інструкцій користувача тощо

Таким чином, на розробку єдиної інформаційної системи для керування підприємством було витрачено 259 робочих годин. Середня ставка заробітної

плати розробника на ринку праці становить 156 грн. 25 коп.. Отже, помноживши кількість робочих годин на ставку заробітної плати була отримана вартість програмного застосунку: 40 468 грн. 75 коп.

Як зазначено у п. 1.3-1.4, вартість корпоративної версії (BAS Бухгалтерія КОРП) коштує 24 000 грн./рік, М.Е.Дос. Бізнес. — 1620 грн./рік. Таким чином, комплекс лише з двох цих програмних застосунків коштує 25 620 грн./рік. Ця сума, сплачена лише одним підприємством, менше, ніж за 2 роки, повністю покриває вартість розробки єдиної інформаційної системи для керування підприємством.

Повністю зрозумілим є той факт, що розроблена система, на поточному етапі, не може в повному обсязі конкурувати із існуючими програмними продуктами та потребує певного доопрацювання для повноцінного впровадження на працюючі підприємства. Але, за приблизними оцінками, наявність хоча б 5 клієнтів здатна повністю покрити вартість створення повноцінної інформаційної системи.

Окрім цього, головною метою її розробки є демонстрація самої можливості об'єднання кількох існуючих програм у одну з метою мінімізації механічних помилок. До того ж, в поточному стані інформаційної системи спроектована та впроваджена можливість швидкого розширення застосунку за допомогою створення відповідних модулів, а також наявна база для подальшого розширення та нарощення функціоналу (вже налагоджений зв'язок з базою даних, автентифікація тощо).

Отже, спираючись на розроблену інформаційну систему, можна дійти до висновку, що була доведена можливість створення єдиної інформаційної системи вітчизняного виробництва, яка у значному обсязі підвищує ефективність роботи певних відділів підприємства, підвищує їхню надійність та цілісність інформації, яка зберігається та використовується на ньому.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було поставлено та виконано наступні задачі:

1. Визначена предметна область та проаналізована існуюча проблематика. В ході спілкування із представниками підприємств було з'ясовано, що у разі використання на підприємстві декількох програм значно знижується надійність (адже при ручному перенесенні інформації із одного застосунку в інший виникають помилки) та ефективність (адже необхідно постійно перемикатися між декількома додатками). Також було вивчено західні та вітчизняні аналоги, в ході чого стало зрозуміло, що на ринку відсутні застосунки, які відповідають всім потребам сучасної бухгалтерії та електронного діловодства. Таким чином, питання створення єдиної інформаційної системи для керування підприємством у якості альтернативи використанню декількох застосунків є недослідженим.

2. Розглянуті різні платформи та технології для розробки інформаційно системи. При вивченні переваг та недоліків перевага була віддана desktop-версії. Оптимальними засобами реалізації програмного застосунку були обрані: C#, Windows Forms, Microsoft SQL Server Management Studio 18 та Visual Studio.

3. Розроблено бізнес-обґрунтування та вимоги до програмного продукту, після чого спроектовані та розроблені єдина інформаційна система керування підприємством та серверна база даних до неї. Також описано її інтерфейс та найважливіші модулі.

4. Досліджено результати розробки інформаційної системи. В ході дослідження було отримано переконання у підвищенні ефективності роботи підприємства та зменшенні вартості використання програмних продуктів.

Таким чином, розроблена інформаційна система відповідає всім вимогам, які були висунуті до програмного застосунку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 11 найкращих програм бухгалтерського обліку для малого бізнесу. MyServerName.com : веб-сайт. URL: <https://uk.myservername.com/11-best-accounting-software> (дата звернення: 20.11.2023).
2. BAS Бухгалтерія. Продукти та рішення для масового ринку. BAS Бухгалтерія : веб-сайт. URL: <https://www.bas-soft.eu/soft/bas-mass/bas-accounting> (дата звернення: 20.11.2023).
3. Best Accounting Software for Small Businesses. Investopedia : веб-сайт. URL: <https://www.investopedia.com/best-accounting-software-for-small-business-5069679> (дата звернення: 20.11.2023).
4. Desktop Guide (Windows Forms .NET). Microsoft : веб-сайт. URL: <https://learn.microsoft.com/uk-ua/dotnet/desktop/winforms/overview/?view=netdesktop-6.0> (дата звернення: 20.11.2023).
5. Learn C#. Microsoft : веб-сайт. URL: <https://dotnet.microsoft.com/en-us/learn/csharp> (дата звернення: 20.11.2023).
6. М.Е.Дос.Бізнес. Миттєвий та законний обмін будь-якими електронними документами з вашими контрагентами. М.Е.Дос : веб-сайт. URL: https://medoc.ua/elektronniy_dokumentoobig (дата звернення: 20.11.2023).
7. Top 10 Programming Languages for Desktop Apps In 2021. Decipherzone : веб-сайт. URL: <https://www.decipherzone.com/blog-detail/top-programming-languages-for-desktop-apps-in-2021> (дата звернення: 20.11.2023).
8. Welcome to the Visual Studio IDE. Microsoft : веб-сайт. URL: <https://learn.microsoft.com/uk-ua/visualstudio/get-started/visual-studio-ide?view=vs-2022> (дата звернення: 20.11.2023).
9. Гуренко А.В., Гашутіна О.Е. Напрями розвитку систем управління в умовах діджиталізації бізнесу в Україні. Економіка та суспільство. 2018. №19. С. 739-745.
10. Замрій І.В., Вишнівський В.В. Структура єдиного інформаційного простору підприємства з критичною інфраструктурою. Телекомунікаційні та інформаційні технології. 2021. № 3 (72). С. 12-24.

11. Коломієць Г. М., Глушач Ю.С. Цифрова економіка: контроверсійність змісту і впливу на господарський розвиток. *Бізнес Інформ*. 2017. № 7. С. 137-143.
12. Конфігурація управління невеликою фірмою для України. Інформаційна система ІТС : веб-сайт. URL: <https://its.1c.ua/db/unfuaua#content:210> (дата звернення: 20.11.2023).
13. Королюк Т., Кравчук Н., Карп І. Стан та тенденції розвитку організації документообігу в інформаційній системі управління підприємством. *Галицький економічний вісник*. 2020. № 6 (67). С. 79-89.
14. Кримінальний кодекс України : Кодекс, Закон від 05.04.2001 № 2341-III. URL: <https://zakon.rada.gov.ua/laws/show/2341-14#Text> (дата звернення: 20.11.2023).
15. Кузнецова С.О., Котелевська А.П. Сучасні проблеми і шляхи вдосконалення управлінського обліку витрат підприємства. *Науковий вісник Міжнародного гуманітарного університету*. 2020. № 45. С. 188-192.
16. Ларченко О. В. Ефективність впровадження інформаційних систем в управління підприємством. *Таврійський науковий вісник*. 2020. № 1. С. 278-284.
17. Макаренко А.П., Меліхов Є.В. Документування обліку запасів за допомогою програмного забезпечення. *Перспективи сталого розвитку в умовах глобалізації в економічному, управлінському та інженерному аспектах* : матеріали Міжнарод. наук.-практ. конф., 3-4 квіт. 2022 р. Запоріжжя : ІННІ ім. Ю.М. Потебні ЗНУ, 2022. С. 162-164.
18. Меліхов Є.В., Макаренко А.П. Інформаційна безпека у еру інформаційних технологій. *Наукові та практичні підходи до проведення економічних, товарознавчих, будівельних експертиз* : матеріали круглого столу, 24 квіт. 2019 р. Запоріжжя : ІІ ЗНУ, 2019. С. 124-126.
19. Меліхов Є.В., Міхайлуца О. М. Дослідження напрямку управління проектами як основа сталого розвитку цифрової сфери. *Цифрова трансформація*

мація соціо-економічних, управлінських та освітянських систем сучасного суспільства : матеріали Міжнарод. наук.-практ. конф., 23-24 квіт. 2022 р. Запоріжжя : ІННІ ім. Ю.М. Потебні ЗНУ, 2022. С. 186-190.

20. Меліхов Є.В., Міхайлуца О. М. Необхідність розробки спеціального програмного забезпечення для ефективної роботи підприємства. Економіко-правові дискусії : 2021 рік : матеріали II Міжнарод. наук.-практ. Інтерн. конф., 14 квіт. 2021 р. Кропивницький : ЛА НАУ, 2021. С. 23-26.

21. Меліхов Євгеній, Міхайлуца Олена. Використання full-stack розробки сайту на технології ASP.NET як невід’ємна складова техніко-економічного розвитку промисловості України. *Формування сучасних концепцій менеджменту організацій та адміністрування в умовах цифровізації* : матеріали Міжнарод. наук.-практ. конф., 23-24 вересня 2021 р. Запоріжжя : Видавничий дім «Гельветика», 2021. С. 349-353.

22. Миколюк О. А. Управління підприємством в умовах діджиталізації економіки. Вісник Хмельницького національного університету. Економічні науки. 2021. № 4. С. 142-146.

23. Прайс-лист BAS. Софтком : веб-сайт. URL: <https://www.softcom.ua/ua/bas/price-list/> (дата звернення: 20.11.2023).

24. Про бухгалтерський облік та фінансову звітність в Україні : Закон України від 16.07.1999 № 996-XIV. URL: <https://zakon.rada.gov.ua/laws/show/996-14#Text> (дата звернення: 20.11.2023).

25. Пуцентейло П., Гуменюк О. Інформаційне забезпечення аналітичної діяльності в управлінні підприємством. Міжнародний науковий журнал. 2019. Випуск 1-2. С. 74-82.

26. Юрчук Н. П. Інформаційні системи і технології як інновація у системі управління бізнес-процесами. Ефективна економіка. 2018. № 5. URL: <http://www.economy.nauka.com.ua/?op=1&z=6323> (дата звернення: 20.11.2023).

27. Яценко В. В. Діджиталізація — сучасний фактор розвитку бізнес-процесів. Ефективна економіка. 2022. № 2. URL: <http://www.economy.nauka.com.ua/?op=1&z=10042> (дата звернення: 20.11.2023).

Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ

Я, Меліхов Євгеній Валентинович, студент 2 курсу, денної форми навчання, Інженерного навчально-наукового інституту ім. Ю.М. Потєбні ЗНУ, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти ipz118bd-10@stu.zsea.edu.ua,

- підтверджую, що написана мною кваліфікаційна робота на тему: «**Єдина інформаційна система керування підприємством України**» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений;

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

- згоден на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою Інтернет - системи, в також архівування моєї роботи у базі даних цієї системи.

Дата 30.11.2023 Підпис _____ Меліхов Євгеній Валентинович
(студент)

Дата 30.11.2023 Підпис _____ Міхайлуца Олена Миколаїна
(науковий керівник)