

Міністерство освіти і науки України  
Криворізький національний університет  
Кафедра моделювання та програмного забезпечення

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття ступеня вищої освіти бакалавра**  
зі спеціальності 121 – Інженерія програмного забезпечення

На тему: Проект системи комп'ютеризації контролю кадрів при  
нестабільності структури трудових ресурсів

Засвідчую, що в цій  
кваліфікаційній роботі немає  
запозичень із праць інших  
авторів без відповідних  
посилань.

Студент гр. ІПЗ–22-1

\_\_\_\_\_ / С. В. Родіонов/

Керівник кваліфікаційної  
роботи

\_\_\_\_\_ / І. А. Котов /

Завідувач кафедри

\_\_\_\_\_ / А. М. Стрюк /

Кривий Ріг

2026

Криворізький національний університет

Факультет: Інформаційних технологій  
Кафедра: Моделювання та програмного забезпечення  
Ступінь вищої освіти: бакалавр  
Спеціальність: 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри

\_\_\_\_\_ А. М. Стрюк

«\_\_\_\_» \_\_\_\_\_ 2026 р.

## **ЗАВДАННЯ**

### **на кваліфікаційну роботу**

студенту групи ПЗ–22-1 Родіонову Сергію Володимировичу

1. Тема: Проект системи комп'ютеризації контролю кадрів при нестабільності структури трудових ресурсів

затверджена наказом по КНУ № 284с від «28» травня 2026 р.

2. Термін подання студентом закінченої роботи: «10» червня 2026 р.

3. Вихідні дані по роботі: програмний комплекс для автоматизації контролю кадрів при нестабільності структури трудових ресурсів.

4. Зміст пояснювальної записки (перелік питань, що їх треба розробити):  
проаналізувати підходи до автоматизації контролю кадрів при нестабільності структури трудових ресурсів, проаналізувати існуючі програмні системи автоматизації контролю кадрів, спроектувати застосунок, розробити програмне забезпечення системи автоматизації контролю кадрів при нестабільності структури трудових ресурсів, здійснити тестування розробленого додатку.

5. Перелік ілюстративного матеріалу: функціональна схема, блок–схема алгоритму, зображення екранних форм додатку.

Календарний план:

| №  | Найменування етапів кваліфікаційної роботи                                            | Термін виконання етапів роботи |
|----|---------------------------------------------------------------------------------------|--------------------------------|
| 1  | Пошук науково-практичних джерел та інтернет-ресурсів з тематики роботи                | 02.02.26 – 09.02.26            |
| 2  | Аналіз існуючих підходів і методів вирішення проблеми                                 | 10.02.26 – 23.02.26            |
| 3  | Формулювання актуальності і постановка завдань роботи                                 | 24.02.26 – 02.03.26            |
| 4  | Оформлення матеріалів першого розділу роботи                                          | 03.03.26 – 09.03.26            |
| 5  | Розробка структурної і функціональної схем та основних алгоритмів програмного проєкту | 10.03.26 – 23.03.26            |
| 6  | Оформлення матеріалів другого розділу роботи                                          | 24.03.26 – 06.04.26            |
| 7  | Розробка візуального інтерфейсу, баз даних і програмних модулів проєкту               | 07.04.26 – 20.04.26            |
| 8  | Оформлення матеріалів третього розділу роботи                                         | 21.04.26 – 11.05.26            |
| 9  | Оформлення додатків                                                                   | 12.05.26 – 18.05.26            |
| 10 | Тестування розробленої програмного комплексу                                          | 19.05.26 – 01.06.26            |
| 11 | Оформлення пояснювальної записки та супровідних документів                            | 02.06.26 – 08.06.26            |

Дата видачі завдання: «02» лютого 2026 р.

Студент: \_\_\_\_\_ / С. В. Родіонов /

Керівник роботи: \_\_\_\_\_ / І. А. Котов /

## РЕФЕРАТ

### АВТОМАТИЗАЦІЯ, БАЗА ДАНИХ, БЛОК-СХЕМА, ІНТЕРФЕЙС, МОДЕЛЬ, ОБЛІК, ПРОЄКТ, РЕСУРС, УПРАВЛІННЯ

Пояснювальна записка: 83 с., 41 рис., 14 табл., 1 дод., 23 джерела.

Кваліфікаційна робота: «Проект системи комп'ютеризації контролю кадрів при нестабільності структури трудових ресурсів».

Мета випускової кваліфікаційної роботи: проектування і розробка програмного застосунку для автоматизації процесів планування, управління та обліку кадрів при нестабільності структури трудових ресурсів.

Об'єктом дослідження і проектування є: процеси планування, управління та обліку кадрів при нестабільності структури трудових ресурсів, а також програмні комплекси автоматизації обліку кадрів.

Теоретична частина роботи присвячена аналізу проблеми організації планування, управління та обліку кадрів при нестабільності структури трудових ресурсів, а також існуючих програмних систем обліку кадрів. Сформульовано актуальність розробки системи автоматизації обліку кадрових ресурсів при нестабільності кадрового забезпечення.

Практична частина кваліфікаційної роботи присвячена розробці структурних і функціональних моделей, структур даних, алгоритмів, бази даних, моделей користувацького інтерфейсу, базових програмних модулів, програмних модулів управління і обліку кадрів, програмних модулів узагальнення і звітів за результатами роботи.

Аналіз розробленого програмного комплексу планування, управління та обліку кадрів при нестабільності структури трудових ресурсів показує, що отримані програмні результати можуть бути використані для цілей обліку кадрів в промисловості і інших установах, а також у вищих навчальних закладах. Використання розробленого застосунку підвищить ефективність обліку трудових ресурсів в умовах нестабільності економічних і технічних обставин.

## **ABSTRACT**

AUTOMATION, DATABASE, BLOCK DIAGRAM, INTERFACE, MODEL,  
ACCOUNTING, PROJECT, RESOURCE, MANAGEMENT

Explanatory note: 83 p., 41 fig., 14 tab., 1 supplement, 23 sources.

Qualification work: « Project for a computerized personnel management system in the context of an unstable labor force structure».

The purpose of the final qualifying work: design and development of a software application to automate the processes of planning, management and accounting of personnel in the event of instability in the structure of labor resources.

The object of research and design are: processes of planning, management and accounting of personnel in the event of instability of the structure of labor resources, as well as software complexes for automating personnel accounting.

The theoretical part of the work is devoted to the analysis of the problem of organizing planning, management and accounting of personnel in the event of instability of the structure of labor resources, as well as existing software systems for accounting of personnel. The relevance of developing a system for automating accounting of human resources in the event of instability of staffing is formulated.

The practical part of the qualification work is devoted to the development of structural and functional models, data structures, algorithms, databases, user interface models, basic software modules, software modules for management and accounting of personnel, software modules for generalization and reports on the results of work.

Analysis of the developed software complex for planning, management and accounting of personnel in the event of instability of the structure of labor resources shows that the obtained program results can be used for the purposes of accounting of personnel in industry and other institutions, as well as in higher educational institutions. The use of the developed application will increase the efficiency of accounting of human resources in conditions of instability of economic and technical circumstances.

## ЗМІСТ

|                                                                                                                                  |    |
|----------------------------------------------------------------------------------------------------------------------------------|----|
| ВСТУП.....                                                                                                                       | 8  |
| 1 ПРОБЛЕМА АВТОМАТИЗАЦІЇ КОНТРОЛЮ КАДРІВ ПРИ<br>НЕСТАБІЛЬНОСТІ СТРУКТУРИ ТРУДОВИХ РЕСУРСІВ                                       | 10 |
| 1.1. Аналіз проблеми автоматизації управлінського обліку<br>кадрових ресурсів в організації.....                                 | 10 |
| 1.2. Аналіз загальних принципів управлінського обліку і<br>контролю кадрів.....                                                  | 14 |
| 1.3. Аналіз моделей управлінського обліку при нестабільності<br>структури управління кадровими ресурсами.....                    | 17 |
| 1.4. Аналіз існуючих систем автоматизації управлінського<br>обліку кадрових ресурсів.....                                        | 22 |
| 1.5. Завдання розробки автоматизованої системи<br>управлінського обліку при нестабільності структури кадрів.....                 | 26 |
| 2 РОЗРОБКА МОДЕЛЕЙ І АЛГОРИТМІВ ПРОГРАМНОГО<br>ЗАСТОСУНКУ КОНТРОЛЮ КАДРІВ ПРИ НЕСТАБІЛЬНОСТІ<br>СТРУКТУРИ ТРУДОВИХ РЕСУРСІВ..... | 29 |
| 2.1. Розробка математичних моделей автоматизованої системи<br>управлінського обліку і контролю кадрів.....                       | 29 |
| 2.2. Розробка структури автоматизованої системи<br>управлінського обліку.....                                                    | 34 |
| 2.3. Розробка функціональної моделі програмного<br>застосунок.....                                                               | 39 |
| 2.4. Розробка схеми бази даних автоматизованої системи<br>управлінського обліку і контролю кадрів.....                           | 42 |
| 2.5. Розробка алгоритмів програмних методів системи<br>автоматизації управлінського обліку.....                                  | 46 |
| 2.6. Проектування візуального інтерфейсу програмного<br>комплексу системи автоматизації управлінського обліку.....               | 50 |

|                                                                                                                        |    |
|------------------------------------------------------------------------------------------------------------------------|----|
| 3. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗАЦІЇ<br>УПРАВЛІНСЬКОГО ОБЛІКУ.....                                            | 54 |
| 3.1. Розробка програмних модулів автоматизованої системи<br>управлінського обліку і контролю кадрів.....               | 54 |
| 3.2. Розробка бази даних автоматизованої системи<br>управлінського обліку і контролю кадрів.....                       | 59 |
| 3.3. Розробка програмних модулів управління даними системи<br>управлінського обліку і контролю кадрів.....             | 63 |
| 3.4. Розробка візуального інтерфейсу користувача системи<br>автоматизації управлінського обліку і контролю кадрів..... | 66 |
| 3.5. Розробка модулів документування результатів роботи<br>програмного застосунку.....                                 | 70 |
| ВИСНОВКИ.....                                                                                                          | 74 |
| ПЕРЕЛІК ПОСИЛАНЬ.....                                                                                                  | 76 |
| ДОДАТОК А.....                                                                                                         | 79 |

## ВСТУП

Роль відділу кадрів у розробці стратегії. Розробка стратегічного плану передбачає виявлення, аналіз та збалансування зовнішніх можливостей і загроз для організації з її внутрішніми сильними та слабкими сторонами. Створені на основі цього стратегічні плани повинні використовувати сильні сторони та можливості підприємства і водночас зменшувати або усунути загрози та слабкі сторони. Команда відділу кадрів може відігравати в цьому випадку кілька ролей. Наприклад, вона має чудові можливості для надання конкурентної розвідки, такої як деталі щодо програм мотивації конкурентів, та інформації про майбутні законодавчі зміни, наприклад, у сфері трудового права. Крім того, він здатний надавати консультації щодо внутрішніх людських сильних і слабких сторін організації [1].

Два рівні тестування оцінюють знання фахівців у всіх аспектах управління персоналом, включаючи адміністративні практики, підбір персоналу, розвиток, оплату праці, трудові відносини, а також аспекти охорони праці та безпеки. Ті, хто успішно виконує всі вимоги, отримують сертифікат висококваліфікованого фахівця з управління персоналом. Нещодавно Інститут сертифікації з управління персоналом почав проводити державні сертифікації, запропонувавши іспити для фахівців з управління персоналом [1, 2].

Існує багато методів аналізу посад. Наприклад, комісія з питань державної служби має стандартизовану процедуру порівняння та класифікації посад. Інформація збирається у реєстраційному бланку для аналізу посад. Спочатку записують ідентифікаційні дані та короткий опис посади. Потім перелічують конкретні завдання посади в порядку їхньої важливості. Далі для кожного виду діяльності аналітик вказує такі аспекти, як необхідні знання (наприклад, факти чи принципи, з якими працівник повинен бути ознайомлений для виконання своєї роботи), необхідні навички (наприклад, навички, необхідні для керування машинами чи транспортними засобами) та



необхідні здібності (наприклад, математичні, логічні, навички вирішення проблем та міжособистісні навички).

На основі проведеного ретельного аналізу сформульований перелік завдань кваліфікаційної роботи: розробити моделі і алгоритми програмного застосунку контролю кадрів при нестабільності структури трудових ресурсів; розробити математичні моделі автоматизованої системи управлінського обліку і контролю кадрів [2]; розробити функціональну модель програмного застосунку; розробити схему бази даних автоматизованої системи управлінського обліку і контролю кадрів; розробити алгоритми програмних методів системи автоматизації управлінського обліку; спроектувати візуальний інтерфейс програмного комплексу системи автоматизації управлінського обліку; програмна реалізація системи автоматизації управлінського обліку; розробити програмні модулі автоматизованої системи управлінського обліку і контролю кадрів [2]; розробити базу даних автоматизованої системи управлінського обліку і контролю кадрів; розробити програмні модулі управління даними системи управлінського обліку і контролю кадрів; розробити візуальний інтерфейс користувача системи автоматизації управлінського обліку і контролю кадрів; розробити модулі документування результатів роботи програмного застосунку.

Розроблена програмна система для автоматизації процесів планування, управління та обліку кадрів при нестабільності структури трудових ресурсів дає змогу підвищити ефективність обліку та управління кадровими ресурсами в умовах нестабільності. Система забезпечує можливості управління персоналом, включаючи адміністративні практики, підбір персоналу, розвиток, оплату праці, трудові відносини [2].

На основі отриманих результатів можна зробити висновок, що виконана робота характеризує актуальність теми, демонструє ефективність розроблених програмних модулів.

Результати роботи можуть бути використані для підприємств, організації, проведення, обліку кадрів та оцінювання їх роботи.

# 1 ПРОБЛЕМА АВТОМАТИЗАЦІЇ КОНТРОЛЮ КАДРІВ ПРИ НЕСТАБІЛЬНОСТІ СТРУКТУРИ ТРУДОВИХ РЕСУРСІВ

## 1.1. Аналіз проблеми автоматизації управлінського обліку кадрових ресурсів в організації

Роль відділу кадрів у розробці стратегії. Розробка стратегічного плану передбачає виявлення, аналіз та збалансування зовнішніх можливостей і загроз для організації з її внутрішніми сильними та слабкими сторонами. Створені на основі цього стратегічні плани повинні використовувати сильні сторони та можливості підприємства і водночас зменшувати або усунути загрози та слабкі сторони. Команда відділу кадрів може відігравати в цьому випадку кілька ролей [1, 2]. Наприклад, вона має чудові можливості для надання конкурентної розвідки, такої як деталі щодо програм мотивації конкурентів, та інформації про майбутні законодавчі зміни, наприклад, у сфері трудового права. Крім того, він здатний надавати консультації щодо внутрішніх людських сильних і слабких сторін організації. Наприклад, під час процесу автоматизації виробник сільськогосподарського обладнання сформував надзвичайно талановиту робочу силу в галузі автоматизації своїх заводів, що дозволило йому створити підрозділ нових технологій для надання послуг з автоматизації іншим компаніям. Управління персоналом та технології. Оскільки компанії Dell потрібно було найняти приблизно 15 000 нових співробітників у 51 країні, група з пошуку талантів розуміла, що для залучення та найму кадрів необхідний процес, заснований на вимірюванні показників [2]. На той час у Dell не було централізованої процедури для контролю за глобальним рекрутингом. Менеджери, розташовані за межами США, збирали свої щомісячні статистичні дані щодо найму в електронних таблицях і надсилали їх до головного офісу в Остіні, штат Техас [2].

Отже, можемо говорити про узгодженість корпоративних стратегій та стратегій управління персоналом, яка показана на рис. 1.1.

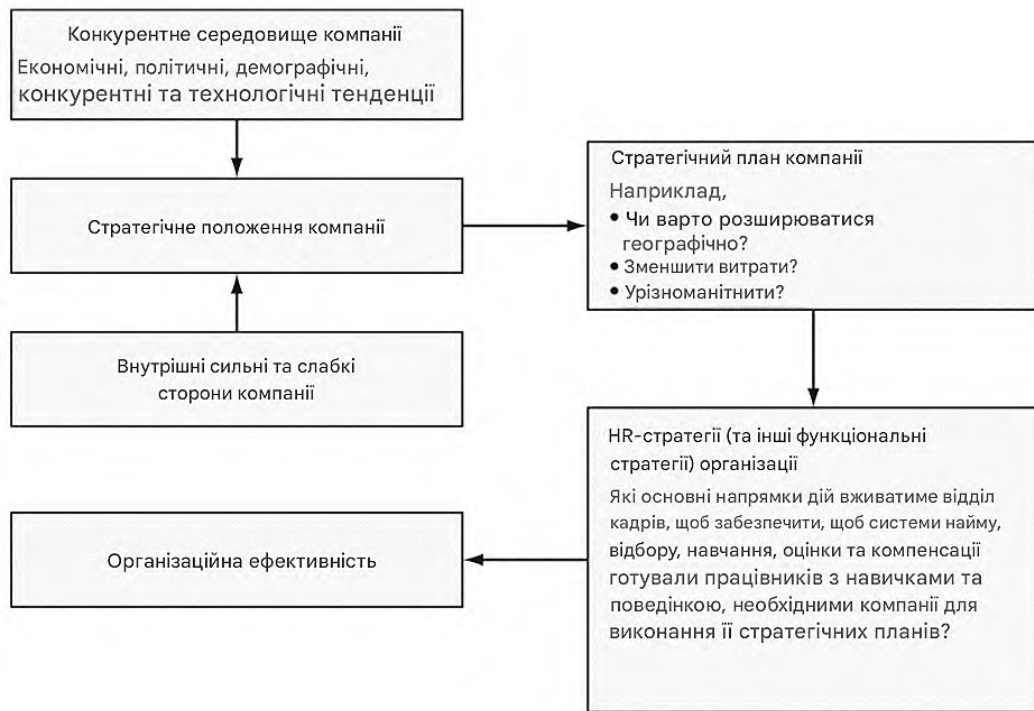


Рисунок 1.1 – Узгодження корпоративних стратегій та стратегій управління персоналом

Dell найняла компанію для встановлення веб-системи відстеження кандидатів, яка автоматизувала процес найму співробітників, обробку їхніх заяв про працевлаштування та відстеження прогресу протягом усього процесу найму [3]. Потім ця компанія співпрацювала з Dell над створенням індивідуальних показників найму, які включали, наприклад, оцінки кандидатів, що їх надавали менеджери з найму. Тож тепер Dell може встановлювати взаємозв'язки між результатами роботи кандидатів та джерелами, звідки їх залучають, і, відповідно, зосереджувати свій найм на найпродуктивніших джерелах. Підвищення продуктивності за допомогою систем управління кадрами (SIRH) [3]. Через внутрішню мережу компанії HR Portals надає співробітникам єдиний пункт доступу, або «вхід», до інформації, пов'язаної з управлінням персоналом. Це дозволяє працівникам, менеджерам та керівникам інтерактивно переглядати та редагувати частину своїх даних [2, 3]. Таким чином, це спрощує процеси управління персоналом і дає можливість менеджерам з управління персоналом більше зосередитися на стратегічних питаннях, які приведені в табл. 1.1.

Таблиця 1.1 – Технологічні рішення, які використовують менеджери з управління персоналом

| Технології                                                | Як HR це використовує                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Постачальники послуг для заявників                        | Послуги соціального страхування надають та керують послугами для роботодавця(наприклад, обробкою заявок на роботу) з власних комп'ютерів віддалено.                                                                                                                    |
| Інтернет-портали                                          | Роботодавці використовують їх, наприклад, щоб дозволити працівникам керувати власними пільгами та оновлювати свою особисту інформацію [3].                                                                                                                             |
| Одночасний відеопотік                                     | Його використовують, наприклад, для полегшення дистанційного навчання.                                                                                                                                                                                                 |
| Мобільний інтернет та доступ до бездротових мереж         | Це служить для полегшення онлайн-доступу співробітників до кадрової діяльності компанії.                                                                                                                                                                               |
| Цифрові персональні програми                              | Наприклад, деякі компанії пропонують новим менеджерам цифрові особисті плани, які містять попередньо завантажену інформацію, щоб допомогти їм краще адаптуватися до нової посади, таку як ключова контактна інформація та цифрові зображення нових співробітників [4]. |
| Програмно забезпечення для моніторингу мереж та Інтернету | Він використовується для моніторингу електронної пошти та інтернет-активності співробітників або для контролю їхньої роботи.                                                                                                                                           |
| Інтегровані інформаційні системи управління персоналом    | Вони використовуються для інтеграції окремих систем управління персоналом роботодавців, наприклад, шляхом автоматичного оновлення списку кваліфікацій працівника після завершення його програми навчання [3, 4].                                                       |
| Електронні підписи                                        | Юридично дійсні електронні підписи, які роботодавець може використовувати для швидшого отримання підписів від кандидатів та працівників.                                                                                                                               |
| Інтернет                                                  | Менеджери широко використовують Інтернет, наприклад, для проведення опитувань щодо заробітної плати.                                                                                                                                                                   |

NCR встановила портал з управління персоналом під назвою HR eXpress, який поділений на три інформаційні розділи: пільги та винагорода, навчання та кар'єрний розвиток, а також цінності та політика NCR у сфері управління персоналом. Компанія також додала «Центр форм» до панелі заголовків сторінки. HR eXpress пропонує співробітникам NCR «швидкий доступ» до всієї інформації, необхідної для виконання завдань з управління персоналом, таких як ті, що пов'язані з корпоративними пільгами та оновленням їхніх персональних даних [4]. Центр форм забезпечує швидкий доступ до будь-якої форми з управління персоналом, яка їм потрібна. Дослідження зазвичай показують, що такі інтернет-додатки («електронні HR») зменшують штат відділу кадрів, прискорюють процеси та знижують витрати на функції управління персоналом [3].

Сертифікація в галузі управління персоналом. У міру того, як завдання менеджера з управління персоналом стають дедалі складнішими, зростає рівень професіоналізації в цій сфері. У Сполучених Штатах понад 60 000 фахівців з управління персоналом вже склали один або обидва іспити на отримання професійної сертифікації в галузі управління персоналом, що проводяться Товариством з управління персоналом (SARM). Інститут сертифікації в галузі управління персоналом (ICRH) при SARM проводить такі іспити [4, 5].

Два рівні тестування оцінюють знання фахівців у всіх аспектах управління персоналом, включаючи адміністративні практики, підбір персоналу, розвиток, оплату праці, трудові відносини, а також аспекти охорони праці та безпеки. Ті, хто успішно виконує всі вимоги, отримують сертифікат APRH (висококваліфікований фахівець з управління персоналом) або PRH (фахівець з управління персоналом) [5]. Нещодавно Інститут сертифікації з управління персоналом почав проводити державні сертифікації, запропонувавши іспити для фахівців з управління персоналом [5].

## 1.2. Аналіз загальних принципів управлінського обліку і контролю кадрів

Управління людськими ресурсами займається практиками та політиками, необхідними для вирішення тих аспектів, які — серед усіх завдань керівництва — стосуються кадрових питань. Зокрема, сюди входять найм, навчання, оцінка, оплата праці та забезпечення безпечного, етичного та справедливого середовища для працівників підприємства [5]. Ці практики та політики, серед іншого, включають:

- проведення аналізу роботи (визначення характеру роботи кожного працівника);
- планування потреб у робочій силі та залучення потенційних працівників [4,5];
- відбір потенційних працівників;
- надання вказівок та навчання працівників;
- оцінка ефективності роботи;
- управління зарплатами та денними ставками (як винагороджувати працівників);
- надання стимулів та пропозицій;
- комунікація (співбесіди, консультування, забезпечення дисципліни).

Також включають те, що повинен знати менеджер щодо: рівних можливостей, етики та позитивних дій; здоров'я, безпеки та етичного ставлення до працівника; розгляду скарг та трудових відносин [6].

Отже, можемо говорити про взаємозв'язок управлінського і фінансового обліку, як показано на рис. 1.2.

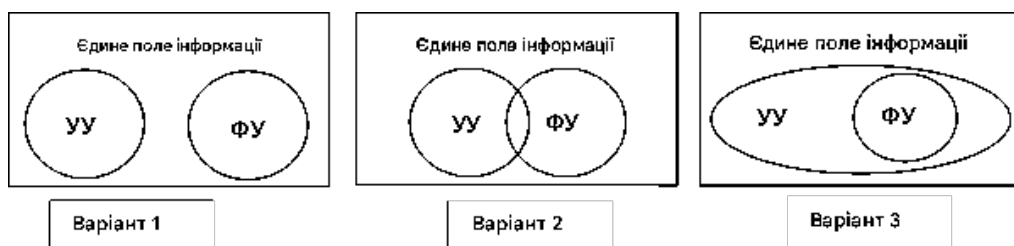


Рисунок 1.2 – Взаємозв'язок управлінського та фінансового обліку

Розглянемо обов'язки менеджерів першої лінії в управлінні персоналом. У будь-якому разі, всі керівники витрачають значну частину свого часу на завдання, пов'язані з управлінням персоналом [6]. Дійсно, робота з людьми завжди була частиною обов'язків менеджера першої лінії [5]. Одна компанія коротко описує, спираючись на наступні загальні заголовки, повноваження менеджерів першої лінії з метою ефективного управління людськими ресурсами:

- призначення відповідної людини на відповідну посаду;
- адаптація нових співробітників у компанії (ознайомлення);
- навчання співробітників новим для них завданням;
- підвищення продуктивності праці кожного співробітника;
- налагодження співпраці та розвиток гармонійних робочих відносин;
- роз'яснення політик та процедур компанії;
- контроль витрат на оплату праці;
- розвиток здібностей кожного працівника;
- створення та підтримання командного духу у відділі;
- захист здоров'я та умов праці працівників.

Графічна ілюстрація взаємодії виробничого та фінансового обліку показана на рис.1.3 [6].

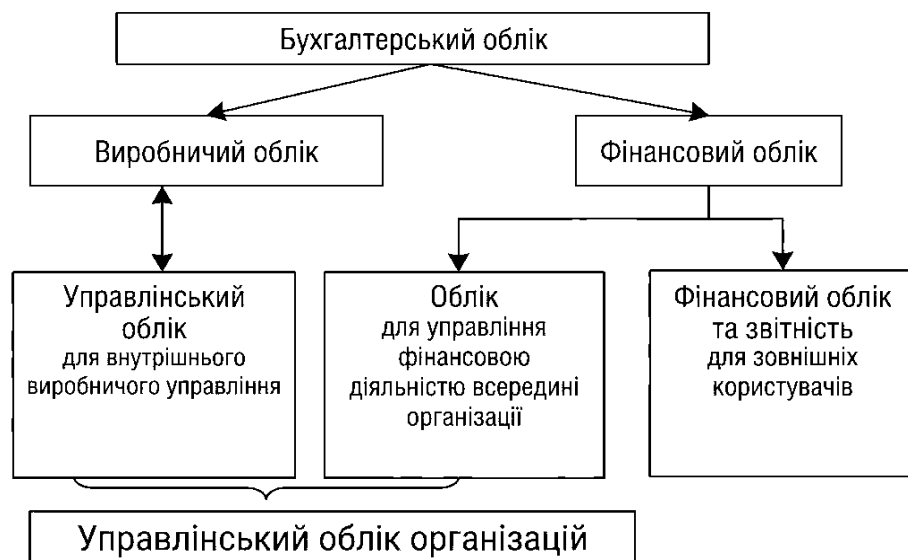


Рисунок 1.3 – Взаємодія фінансового та виробничого обліку

Тепер можна провести порівняльний аналіз фінансового та управлінського обліку, який наведено в табл. 1.2

Таблиця 1.2 – Порівняльна характеристика фінансового та управлінського обліків

|   | Критерій                      | Фінансовий облік                                                                                                                                                                                                                       | Управлінський облік                                                                                                                              |
|---|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Обов'язковість ведення обліку | Ведення є обов'язковим. Обов'язок вести фінансовий облік з використанням аналітичних рахунків визначено законом України «Про бухгалтерський облік», дія якого поширюється на всі організації, що знаходяться на території України [7]. | Ведення залежить від волі керівництва. Збирається та обробляється інформація, цінність якої для управління не нижче витрат на її отримання.      |
| 2 | Мета ведення обліку           | Підготовка інформації для зовнішніх користувачів про активи та зобов'язання компанії, фінансові результати її діяльності, тобто. зовнішньої звітності.                                                                                 | Забезпечення менеджменту компанії достовірною та оперативною інформацією, необхідною для планування, аналізу, контролю, прийняття рішень [5, 7]. |
| 3 | Користувачі інформації        | Зовнішні (акціонери, інвестори, податкові органи, покупці тощо) та внутрішні користувачі                                                                                                                                               | Тільки внутрішні користувачі відповідно до своєї компетенції.                                                                                    |

Набір більш різноманітного персоналу. Як вже було зазначене раніше, склад робочої сили кардинально змінюється. Кількість жінок та представників меншин серед працівників зростатиме, що вимагатиме вжиття спеціальних



заходів для набору людей старшого віку, представників меншин та жінок. Бачимо, що багато факторів сприяють успішному набору працівників з цих категорій. Наприклад, гнучкий графік роботи приваблює та утримує працівників, які є самотніми батьками. Загалом, важливо вживати заходів, які «вказують» на те, що це гарне місце роботи для різноманітних працівників [7]. Передача відповідного повідомлення передбачатиме використання засобів масової інформації, орієнтованих на меншини, різноманітних оголошень, акцентування уваги на політиці інклюзії, а також залучення рекрутерів-жінок та представників меншин.

Існує багато інших методів аналізу посад. Наприклад, комісія з питань державної служби має стандартизовану процедуру порівняння та класифікації посад. Інформація збирається у реєстраційному бланку для аналізу посад. Спочатку записують ідентифікаційні дані та короткий опис посади [6]. Потім перелічують конкретні завдання посади в порядку їхньої важливості. Далі для кожного виду діяльності аналітик вказує такі аспекти, як необхідні знання (наприклад, факти чи принципи, з якими працівник повинен бути ознайомлений для виконання своєї роботи), необхідні навички (наприклад, навички, необхідні для керування машинами чи транспортними засобами) та необхідні здібності (наприклад, математичні, логічні, навички вирішення проблем та міжособистісні навички) [7].

### 1.3. Аналіз моделей управлінського обліку при нестабільності структури управління кадровими ресурсами

Дамо короткий огляд наступних матеріалів, які не слід розглядати як окремі та взаємозалежні, оскільки кожен із них взаємодіє з іншими та впливає на них: усі вони повинні відповідати стратегічному плану роботодавця [6]. Підсумуємо цю ідею на рис. 1.4. Наприклад, те, як ви проводите тестування та співбесіди з кандидатами на посаду та навчаєте й оцінюєте діючих працівників, залежить від конкретних обов'язків та

відповідальності, пов'язаних із посадою. Те, як ви відбираєте та навчаєте працівників, впливатиме на впевненість, з якою вони виконують свої завдання. Результативність працівника і, отже, його оцінка залежать не лише від мотивації самого працівника, а й від того, як були визначені обов'язки посади, а також від того, як відбувався відбір та навчання працівника [7]. Крім того, як ми бачили раніше, кожна з кадрових стратегій роботодавця в кожній сфері, наприклад, як він наймає, відбирає, навчає, оцінює та винагороджує працівників, повинна відповідати стратегічному плану підприємства. Загальна стратегія та основні процеси управління персоналом приведені на рис. 1.4.

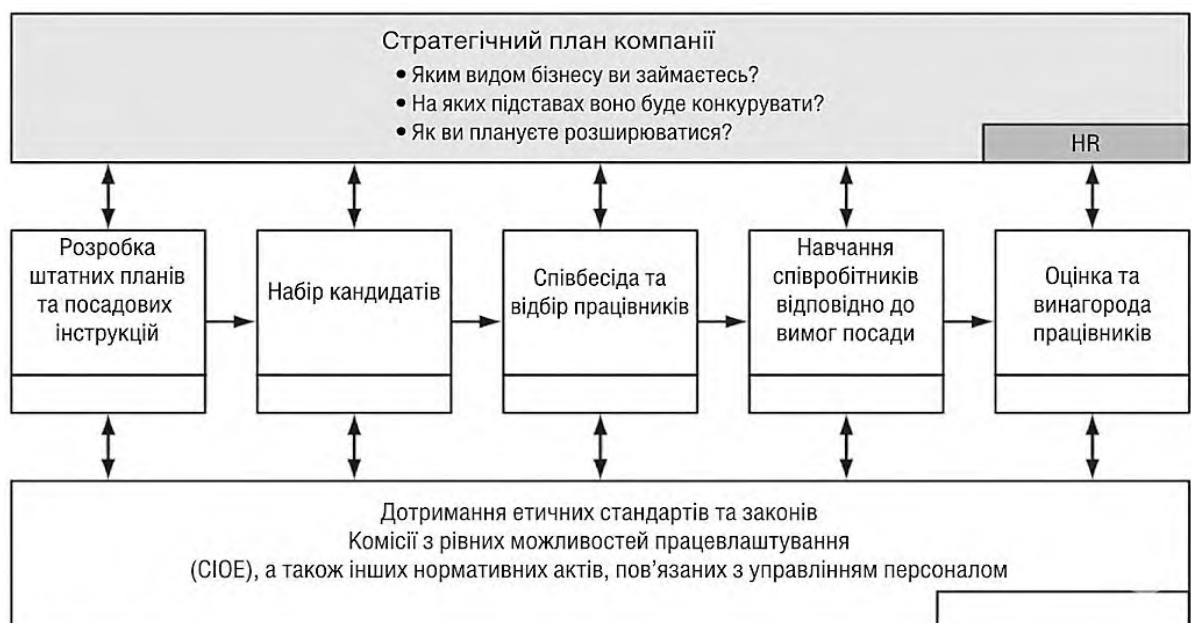


Рисунок 1.4 – Стратегія та основні процеси управління персоналом

Процес, за допомогою якого аналізуються потреби в кадрових ресурсах у міру зміни внутрішнього та зовнішнього середовища підприємства. Він дозволяє забезпечити необхідну кількість підготовлених і кваліфікованих працівників на відповідних посадах у потрібний час. Визначення потреб у кадрових ресурсах та способів їх задоволення з метою реалізації стратегічних планів організації [4, 7].

Стратегічне планування людських ресурсів необхідне для того, щоб підприємства могли передбачати та запобігати відтоку та притоку персоналу.

Мета стратегічного планування людських ресурсів полягає в тому, щоб використовувати ресурси якомога ефективніше, там і тоді, де та коли вони потрібні, з метою досягнення цілей та завдань підприємства відповідно до його стратегії [7].

Поняття «компетентності» стосується внутрішніх рис особистості, пов'язаних із належним виконанням обов'язків на робочому місці, які можуть ґрунтуватися на особистій задоволеності, самооцінці, цінностях, характері, поведінці або знаннях [6]. Тому ефективна структура управління підприємством можна відобразити схемою на рис. 1.5.

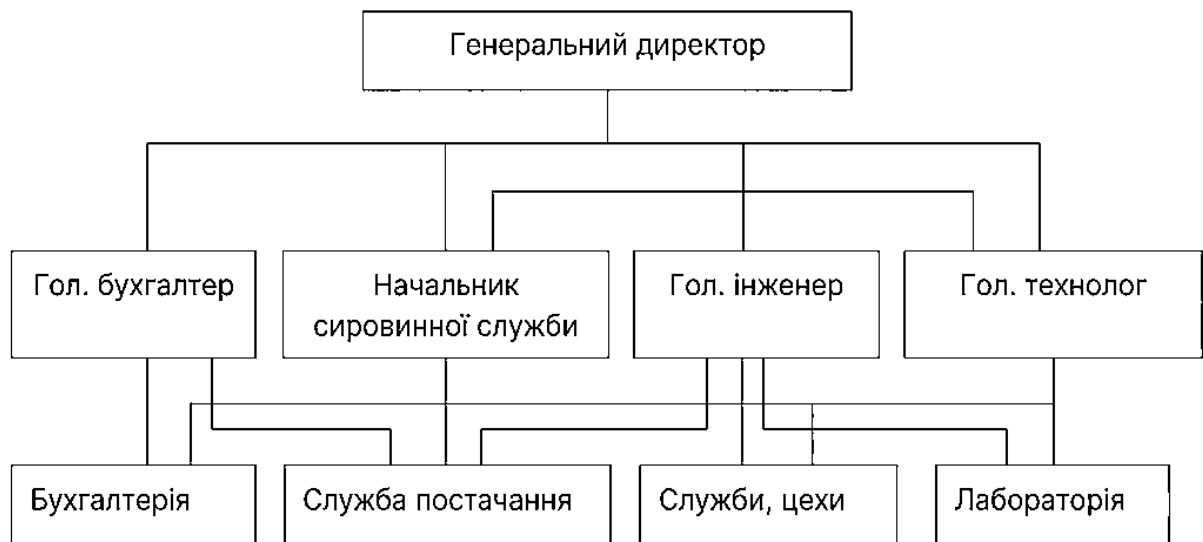


Рисунок 1.5 – Ефективна структура підрозділів промислового підприємства

Ця концепція знаходиться в межах конкретної теми аналізу та опису посад, яка надає пріоритет визначенню конкретних завдань, що призводить до створення досить простих професійних профілів, які майже повністю ґрунтуються на підтвердженні технічних знань, що вважаються необхідними. Оцінка компетентностей вивчає людей, які демонструють успішну роботу, і визначає посаду виходячи з характеристик та поведінки цих людей. Таким чином, посаду можна визначити з точки зору компетентностей, необхідних для її належного виконання. Розрізняють два типи компетенцій [7, 8]:

– основні: це ті, що необхідні для досягнення належного рівня виконання. Зазвичай пов'язані з технічними знаннями або механічними навичками;

– вирішальні: це ті, що необхідні для досягнення успішного рівня виконання, а отже, вищого за належний. Більше пов'язані з поведінкою, цінностями, мотивацією тощо [6].

З урахуванням компетенцій функціональні рекрутингові дії при прийомі на роботу зображені на рис. 1.6.

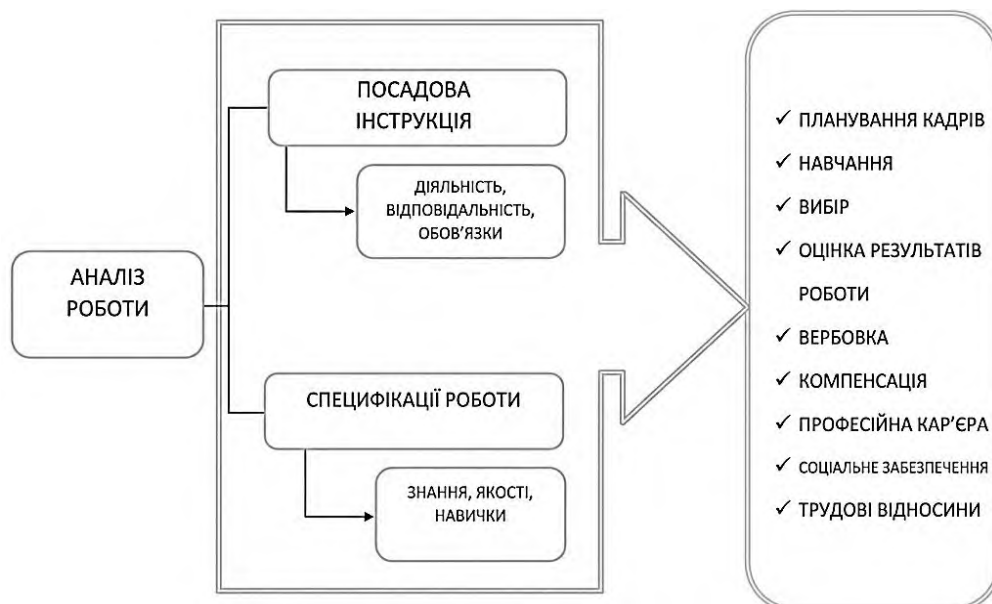


Рисунок 1.6 – Функціональні рекрутингові дії при найманні на роботу

Метою, яка спонукає підприємство розпочати процес аналізу та опису робочих посад, є отримання необхідної інформації, яка дозволить підприємству вчасно досягти місії та цілей, пов'язаних зі стратегічним плануванням, одночасно задовольняючи потреби персоналу, що його складає, сприяючи створенню сприятливого робочого клімату та забезпеченню справедливої оплати праці, тим самим запобігаючи виникненню порівняльних невдоволень серед співробітників [5, 8].

Базуючись на цих принципах, інформаційну модель ефективного менеджменту персоналу можна зобразити на рис. 1.7 [8].



#### 1.4. Аналіз існуючих систем автоматизації управлінського обліку кадрових ресурсів

Управління людськими ресурсами – це та частина управління підприємством, яка стосується політик та процедур, що застосовуються в організації для належного управління людськими ресурсами з метою реалізації їхнього повного потенціалу [7]. У системі управління слід планувати всі кадрові заходи, як економічного, так і адміністративного характеру.

Існує багато систем автоматизації управлінського обліку кадрових ресурсів. Розглянемо і проаналізуємо деякі з них.

Система Rippling показана на рис. 1.8 [8].

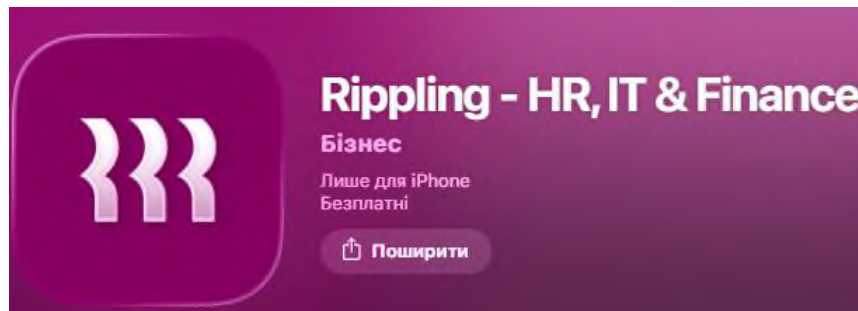


Рисунок 1.8 – Система автоматизації управлінського обліку кадрових ресурсів – Rippling [8]

Rippling - це всеосяжна платформа, призначена для оптимізації різних аспектів керування робочою силою, HR, IT та інших. Він пропонує низку інструментів, щоб допомогти організаціям ефективно керувати своєю діяльністю [8]. Ключові функції включають цифровий годинник для відстеження годинника працівників, планувальник перетягування для створення графіків співробітників і автоматичні робочі процеси, які можна налаштувати з використанням різних джерел даних. Платформа також підтримує навчання співробітників, інтегруючи навчальні курси до графіків співробітників [8, 9].

Однією з визначних особливостей Rippling є його здатність автоматизувати складні робочі процеси у різних відділах [9]. Це включає

створення тригерів і дій на основі різних точок даних, таких як володіння співробітників або інформація про планування. Крім того, Rippling надає надійні можливості звітності, дозволяючи користувачам генерувати власні звіти за допомогою інтерфейсу без кодів [8, 9]. Ця гнучкість дозволяє організаціям аналізувати свої дані значними способами, як-от оцінка тенденцій компенсації чи відстеження часу [9].

Система Innaware, яка показана на рис 1.9 [9, 10].



IW HR&Payroll: Кадровий облік і розрахунок зарплат для України

Рисунок 1.9 – Система Innaware

Спеціалізований модуль IW HR&Payroll, розроблений компанією Innaware, призначений для автоматизації кадрового адміністрування та розрахунку заробітної плати відповідно до чинного законодавства України. Функціонал додатка забезпечує комплексне управління персональними даними співробітників, реєстрацію кадрових наказів і періодів відсутності, а також ведення обліку робочого часу [9]. Завдяки впровадженню IW HR&Payroll досягається централізація кадрового обліку, оптимізується рекрутинг (моніторинг вакансій та аналіз профілів кандидатів), автоматизується контроль за лікарняними й відпустками, а також забезпечується автоматичне нарахування виплат із подальшим формуванням регламентованої звітності [10]. Платформа Microsoft Dynamics 365 дозволяє легко інтегрувати розроблене рішення з будь-якими іншими інформаційними системами, що вже використовуються на підприємстві.

Наступна система BAS – зарплата та управління персоналом показана на рис. 1.10.



Рисунок 1.10 – Система BAS – зарплата та управління персоналом

Ця система стверджує наступне. Назавжди відмовтеся від розрізнених систем і підготуйтеся до використання єдиного джерела достовірної інформації про всіх ваших співробітників. Забезпечте впорядкованість, безпеку та доступність даних, а головне — їхню відповідність вимогам законодавства про захист персональних даних [10].

Наступна система – Vimp ERP – управлінський облік показана на рис. 1.11.



### **Vimp ERP - Управлінський облік та склад для Виробництва і E-commerce**

Рисунок 1.11 – Система Vimp ERP – управлінський облік

Всі системи відрізняються розвиненим графічним інтерфейсом користувача, що допомагає керівнику здійснювати повний контроль над процесами прийому персоналу на роботу, а також відстежувати його кар'єрні та професійні успіхи [10, 11].

Нижче показано систему SMARTFIN.UA, рис. 1.12.



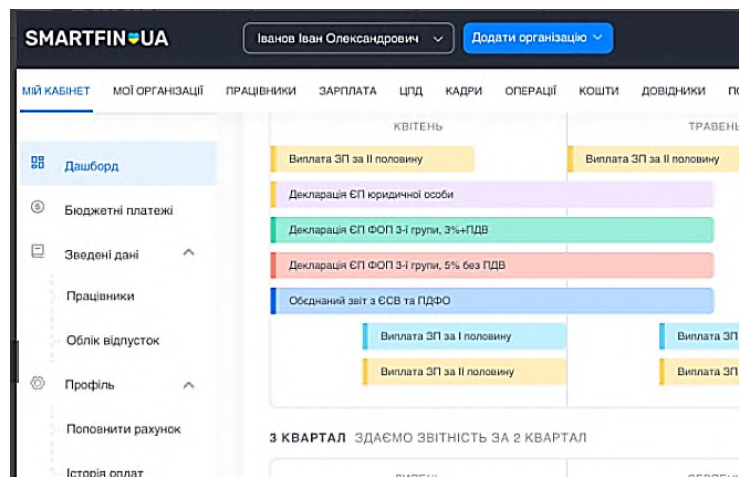


Рисунок 1.12 – Система обліку персоналу SMARTFIN.UA [11]

Наступна проаналізована система – Finmap, показана на рис. 1.13.

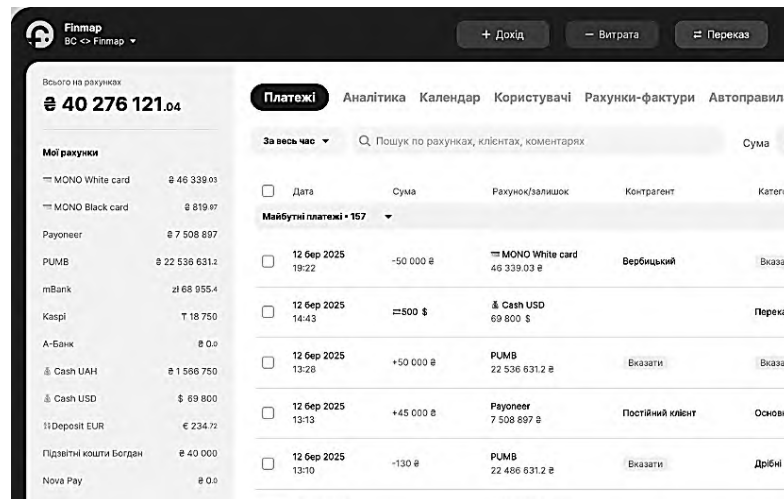
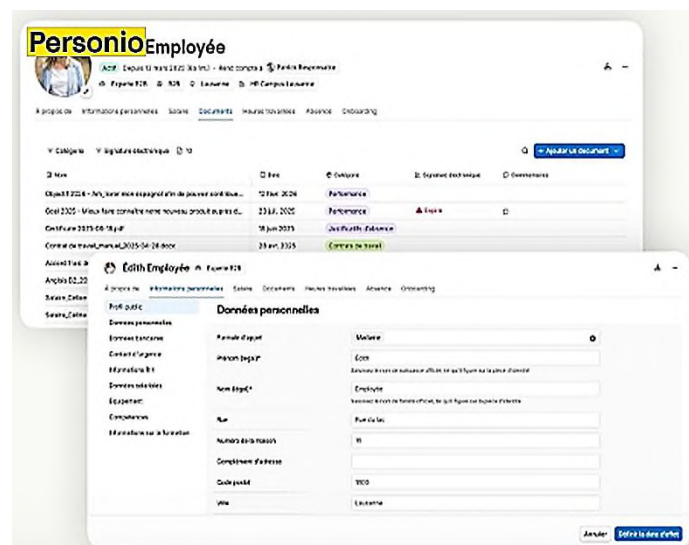


Рисунок 1.13 – Система Finmap [11]

Наступна проаналізована система – Personio, показана на рис. 1.14.



Система 1.14 – Система Personio [12]

### 1.5. Завдання розробки автоматизованої системи управлінського обліку при нестабільності структури кадрів

Аналіз підвищення продуктивності за допомогою систем управління кадрами (SIRH). Через внутрішню мережу компанія надає співробітникам єдиний пункт доступу, або «вхід», до інформації, пов'язаної з управлінням персоналом. Це дозволяє працівникам, менеджерам та керівникам інтерактивно переглядати та редагувати частину своїх даних. Таким чином, це спрощує процеси управління персоналом і дає можливість менеджерам з управління персоналом більше зосередитися на стратегічних питаннях [11]. Компанія встановила портал з управління персоналом під назвою HR eXpress, який поділений на три інформаційні розділи: пільги та винагорода, навчання та кар'єрний розвиток, а також цінності та політика у сфері управління персоналом. Компанія також додала «Центр форм» до панелі заголовків сторінки. HR eXpress пропонує співробітникам «швидкий доступ» до всієї інформації, необхідної для виконання завдань з управління персоналом, таких як ті, що пов'язані з корпоративними пільгами та оновленням їхніх персональних даних [10]. Центр форм забезпечує швидкий доступ до будь-якої форми з управління персоналом, яка їм потрібна. Дослідження зазвичай показують, що такі інтернет-додатки («електронні HR») зменшують штат відділу кадрів, прискорюють процеси та знижують витрати на функції управління персоналом [12].

Управління людськими ресурсами займається практиками, необхідними для вирішення тих аспектів, які — серед усіх завдань керівництва — стосуються кадрових питань. Зокрема, сюди входять найм, навчання, оцінка, оплата праці та забезпечення безпечного, етичного та справедливого середовища для працівників підприємства [12]. Ці практики та політики, серед іншого, включають: проведення аналізу роботи (визначення характеру роботи кожного працівника); планування потреб у робочій силі та залучення потенційних працівників; відбір потенційних працівників; надання вказівок та

навчання працівників; оцінка ефективності роботи; управління зарплатами та денними ставками (як винагороджувати працівників); надання стимулів та пропозицій; комунікація (співбесіди, консультування, забезпечення дисципліни). Також включають те, що повинен знати менеджер щодо: рівних можливостей, етики та позитивних дій, здоров'я, безпеки та етичного ставлення до працівника, розгляду скарг та трудових відносин [12].

Процес, за допомогою якого аналізуються потреби в кадрових ресурсах у міру зміни внутрішнього та зовнішнього середовища підприємства. Він дозволяє забезпечити необхідну кількість підготовлених і кваліфікованих працівників на відповідних посадах у потрібний час. Визначення потреб у кадрових ресурсах та способів їх задоволення з метою реалізації стратегічних планів організації [11, 12].

Стратегічне планування людських ресурсів необхідне для того, щоб підприємства могли передбачати та запобігати відтоку та притоку персоналу [11, 12]. Мета стратегічного планування людських ресурсів полягає в тому, щоб використовувати ресурси якомога ефективніше, там і тоді, де та коли вони потрібні, з метою досягнення цілей та завдань підприємства відповідно до його стратегії.

Метою, яка спонукає підприємство розпочати процес аналізу та опису робочих посад, є отримання необхідної інформації, яка дозволить підприємству вчасно досягти місії та цілей, пов'язаних зі стратегічним плануванням, одночасно задовольняючи потреби персоналу, що його складає, сприяючи створенню сприятливого робочого клімату та забезпеченню справедливої оплати праці, тим самим запобігаючи виникненню порівняльних невдовольень серед співробітників [10, 11].

На основі проведеного аналізу в роботі сформульовані наступні задачі:

- розробити моделі і алгоритми програмного застосування контролю кадрів при нестабільності структури трудових ресурсів;
- розробити математичні моделі автоматизованої системи управлінського обліку і контролю кадрів;

- розробити структуру автоматизованої системи управлінського обліку;
- розробити функціональну модель програмного застосунку;
- розробити схему бази даних автоматизованої системи управлінського обліку і контролю кадрів;
- розробити алгоритми програмних методів системи автоматизації управлінського обліку;
- спроектувати візуальний інтерфейс програмного комплексу системи автоматизації управлінського обліку;
- програмна реалізація системи автоматизації управлінського обліку;
- розробити програмні модулі автоматизованої системи управлінського обліку і контролю кадрів;
- розробити базу даних автоматизованої системи управлінського обліку і контролю кадрів;
- розробити програмні модулі управління даними системи управлінського обліку і контролю кадрів;
- розробити візуальний інтерфейс користувача системи автоматизації управлінського обліку і контролю кадрів;
- розробити модулі документування результатів роботи програмного застосунку.

## 2 РОЗРОБКА МОДЕЛЕЙ І АЛГОРИТМІВ ПРОГРАМНОГО ЗАСТОСУНКУ КОНТРОЛЮ КАДРІВ ПРИ НЕСТАБІЛЬНОСТІ СТРУКТУРИ ТРУДОВИХ РЕСУРСІВ

### 2.1. Розробка математичних моделей автоматизованої системи управлінського обліку і контролю кадрів

Операційні дослідження — це галузь математики, що полягає у використанні математичних моделей, статистики та алгоритмів з метою прийняття рішень [12]; це сучасна наукова дисципліна, яка характеризується застосуванням спеціальних теорій, методів і технік для пошуку рішень проблем управління, організації та контролю, що виникають у різних системах, які існують у природі та створені людиною, таких як різноманітні організації, які вона ідентифікує як організовані системи, фізичні, економічні, екологічні, освітні, соціальні системи тощо [13]. Операційні дослідження використовують методологію для вирішення наукових проблем, беручи до уваги етапи проекту з операційних досліджень.

Збір даних та визначення математичної задачі. На цьому етапі процесу необхідно: описати цілі системи, тобто те, що потрібно оптимізувати; визначити відповідні змінні, незалежно від того, чи є вони керованими; встановити обмеження системи. Також слід врахувати можливі варіанти рішень, часові обмеження для прийняття рішення та обмеження щодо отримання належного рішення [13, 14]. Цей процес визначення проблеми є дуже важливим, оскільки він суттєво вплине на висновки дослідження; неможливо отримати правильну відповідь на неправильно сформульовану проблему. Дуже важливою частиною формулювання математичної моделі є збір даних, оскільки вони дозволять реалізувати модель і забезпечити точність результатів, адже потрібно багато даних, щоб досягти точного розуміння проблеми та надати відповідні вхідні дані для математичної моделі, яка буде сформульована на наступному етапі дослідження [14].

Формулювання математичної моделі. Після того, як переконано, що визначення математичної проблеми сформульовано конкретно та правильно, переходять до формулювання математичної моделі. Ця модель має бути сформульована таким чином, щоб виражати найважливіше в зазначеній проблемі [14]. Математична модель базується на рівняннях та нерівностях, сформульованих у термінах змінних, які виражають суть проблеми, що вирішується, і визначаються залежно від основних характеристик проблеми. Після визначення змінних залежно від цієї проблеми приступають до математичного визначення двох частин, що складають модель. Міра ефективності, яка дозволяє визначити рівень досягнення цілей; зазвичай це функція, яка називається цільовою функцією. Обмеження задачі, які називаються обмеженнями, — це сукупність рівностей або нерівностей, що становлять бар'єри та перешкоди для досягнення цілі [14].

Отримання розв'язку на основі математичних моделей, перевірка моделі та її розв'язку Отримавши модель, приступають до отримання математичного розв'язку, використовуючи різноманітні математичні техніки та методи для розв'язання задач і рівнянь. Розв'язання моделі полягає у знаходженні значень залежних змінних, пов'язаних з керованими компонентами системи, з метою оптимізації, якщо це можливо, або принаймні підвищення ефективності чи результативності системи в рамках, встановлених цілями та обмеженнями задачі [12]. Слід враховувати, що рішення, отримані на цьому етапі процесу, є математичними і повинні інтерпретуватися в реальному світі. Вибір методу вирішення залежить від характеристик моделі.

Перед впровадженням модель має підтвердити свою обґрунтованість шляхом перевірки того, чи дозволяють її результати з певною точністю або приблизністю передбачити наслідки різних варіантів рішень [13]. Для перевірки рішення моделі необхідно зібрати інформацію з метою проведення необхідних випробувань та перевірки відповідно до наступних кроків: науково визначити (включаючи показник ефективності), провести вибірку (включаючи планування експериментів), зменшити кількість даних, використати дані для

перевірки гіпотез, оцінити результати. Якщо ці кроки виконуються повторно щоразу, коли отримуються результати моделі, починається виконання систематичної процедури контролю, яка очищає та пристосовує її до реальності [14].

Встановлення контролю над розв'язком математичної моделі. Розв'язок, визнаний дійсним для задачі, залишається таким, доки умови задачі, неконтрольовані змінні, параметри, взаємозв'язки тощо не зазнають істотних змін. Ця ситуація стає більш ймовірною, коли деякі параметри були оцінені приблизно. З огляду на це, необхідно отримати додаткову інформацію про поведінку рішення у зв'язку зі змінами параметрів моделі; зазвичай це називають аналізом чутливості [14]. Цей етап полягає у визначенні діапазонів варіації параметрів, в межах яких рішення задачі не змінюється.

Впровадження рішення математичної моделі. Коли система для застосування моделі вже розроблена, останній етап полягає у впровадженні перевірених результатів моделі [15]. Це, по суті, означає перетворення цих результатів на детальні операційні інструкції, викладені у формі, зрозумілій особам, які будуть керувати та експлуатувати систему. По завершенні дослідження доцільно, щоб команда з операційних досліджень задокументувала свою методологію достатньо чітко, щоб робота була відтворюваною [14]. Після представлення методології операційних досліджень, яка є дуже корисною дослідницькою логікою, що може бути використана для вдосконалення процесу розподілу завдань між людськими ресурсами. Тут доцільно ввести деякі більш конкретні базові елементи, пов'язані з плануванням проектів за допомогою методів PERT та CPM, які закладуть основу для посилення процесу [15].

Виходячи з принципів побудови математичних моделей, які були викладені вище, насамперед необхідно розглянути можливість побудови математичних моделей формальної оцінки власних якостей та результатів роботи персоналу [15]. І тому розглянемо узагальнену схему оцінки надійності роботи персоналу, показану на рис. 2.1.

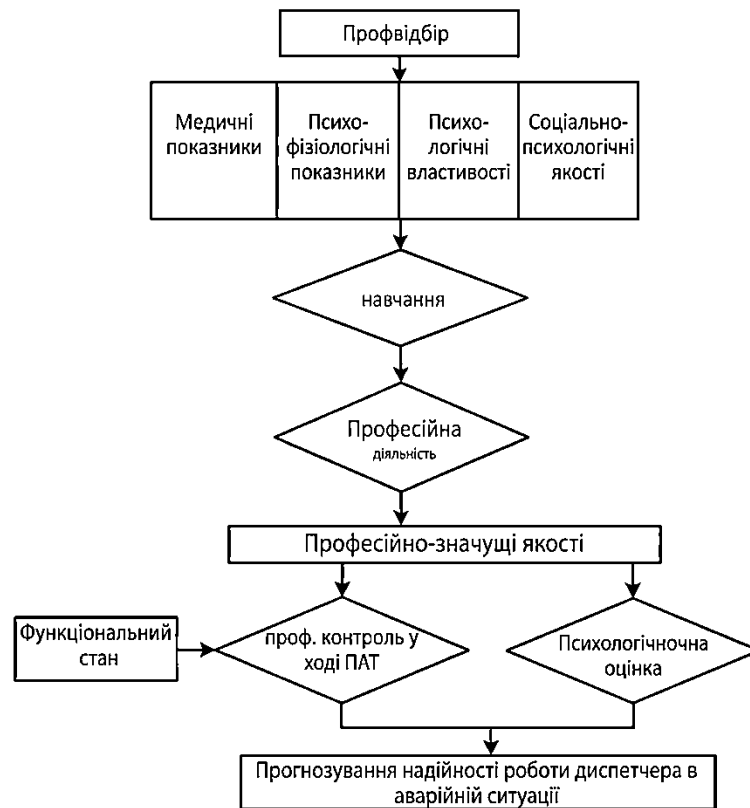


Рисунок 2.1 – Схема оцінки надійності персоналу

Тепер запровадимо оцінки основних показників роботи персоналу [13]:

- Правильно, своєчасно виконані дії –  $q_1$ ;
- Невиконані дії –  $q_2$ ;
- Неправильні дії –  $q_3$ ;
- Дії, виконані із запізненням –  $q_4$ ;
- Дії, виконані раніше необхідного –  $q_5$ ;
- Зайві дії –  $q_6$ ;
- Неоптимальні дії –  $q_7$ .

Отже, проводиться оцінка кожної дії (зокрема і невиконаного необхідного). Запропонована класифікація дій дозволяє отримати наступний набір професійних та психологічних характеристик персоналу [12, 15]. Наведемо основні формули.

$$\alpha = \frac{q_1}{\sum_i q_i}, \quad (2.1)$$

де  $\alpha$  – професіоналізм, необхідна умова висококласної роботи,  $\alpha \rightarrow 1$ .



$$\rho = 1 - \frac{q_2 + q_3 + q_4 + q_5 + q_6}{\sum_i q_i}, \quad (2.2)$$

де  $\rho$  – надійність, необхідна умова надійної роботи,  $\rho \rightarrow 1$  [15].

$$\beta = \frac{q_3 + q_4}{\sum_i q_i}, \quad (2.3)$$

де  $\beta$  – рівень підготовки, необхідна умова високої підготовленості,  $\beta \rightarrow 1$ .

$$\gamma = 1 - \frac{q_5 + q_6}{\sum_i q_i}, \quad (2.4)$$

де  $\gamma$  – стійкість, необхідна умова стійкої роботи, що виключає метушливість  $\gamma \rightarrow 1$  [15].

$$\delta = 1 - \frac{q_4 + q_7}{\sum_i q_i}, \quad (2.5)$$

де  $\delta$  – оперативність, необхідна умова чіткої роботи, що виключає загальмованість,  $\delta \rightarrow 1$  [14].

На основі наведених формул можна провести оцінку ділових якостей персоналу, яка показана у табл. 2.1.

Таблиця 2.1 – Матриця оцінки ділових якостей персоналу

|                   | Посадові обов'язки |     | Ділові якості |     | Особисті якості |     | Бали |            | Оцінка |
|-------------------|--------------------|-----|---------------|-----|-----------------|-----|------|------------|--------|
| Швидкість реакції | 40                 | 0,4 | 30            | 0,4 | 20              | 0,2 | 80   | 16+12+4=32 | 3      |
| Наполегливість    | 25                 | 0,4 | 40            | 0,4 | 25              | 0,2 | 90   | 10+16+5=31 | 3      |

Базова чисельність персоналу для планових сервісних робіт рахується за наступної формулою [14].

$$N_{BP} = C_N = A * \frac{T_{PS}}{60}, \quad (2.6)$$

де  $N_{BP}$  – годинне навантаження;  $A$  – орієнтовна кількість клієнтів, які прибуватимуть щогодини;  $T_{PS}$  – середній час обслуговування, який кожен оператор витрачає на одного клієнта. Формула ділиться на 60, якщо  $T_{PS}$  вимірюється в хвилинах, але потрібно ділити на 3600, якщо  $T_{PS}$  вимірюється в секундах.

Опишемо загальні математичні моделі для оптимального розподілу людських ресурсів у проектах [16].

Визначення змінних:  $X_{ijkd}$  – кількість людських ресурсів кваліфікації  $i$ , які будуть призначені на посаду  $j$  елемента  $k$  роботи в день  $d$ . Цільова функція

$$\min(Z(t)) = \sum_{j=1}^J \sum_{k=1}^K \sum_{d=t}^{t+1} x_{ijkd}. \quad (2.7)$$

Цільова функція мінімізує сумарну суму розподілу людських ресурсів. Щоб сформулювати наведені нижче обмеження, слід зауважити, що вони встановлюються для фіксованої кваліфікації, де передбачається, що людські ресурси, які можна розподілити між різними посадами та об'єктами будівництва, є обмеженими. Перші  $d$  обмежень є такими [16]

$$\sum_{k=1}^K \sum_{j=1}^J a_{ijkd} x_{ijkd} \leq drh_i, d = t, \dots, T + 1. \quad (2.8)$$

Ці обмеження означають, що при визначенні кваліфікації та дати, під час підсумовування загальної кількості людських ресурсів, розподілених між різними посадами та об'єктами, не слід перевищувати наявну кількість людських ресурсів для даної кваліфікації [17].

## 2.2. Розробка структури автоматизованої системи управлінського обліку

У суспільному житті сам прогрес людства призвів до розквіту виробництва, технологій та науки, які стрімко розвиваються в інтересах розвитку та добробуту людства. Сьогодні людина змушена застосовувати дуже складні промислові процеси з необхідною якістю та надійністю, де в багатьох випадках обсяг оброблюваної інформації може становити тисячі аналогових та цифрових змінних [16]. На сьогоднішній день неможливо уявити жоден сучасний виробничий процес без автоматизації. У найпростіших випадках вона може охоплювати датчики та локальні контури управління, а в більш складних застосуваннях — комп'ютерні системи збору даних; системи

SCADA, розподілені системи управління тощо [17]. В автоматизації технологічних процесів, промисловості, лабораторіях та сферах послуг системи управління відіграють фундаментальну роль, оскільки завдяки їм можна гарантувати правильне функціонування, експлуатацію, нагляд та керування такими системами. Нижче показана конфігурація комп'ютерної інформаційної системи управлінського обліку, рис. 2.2 [17].

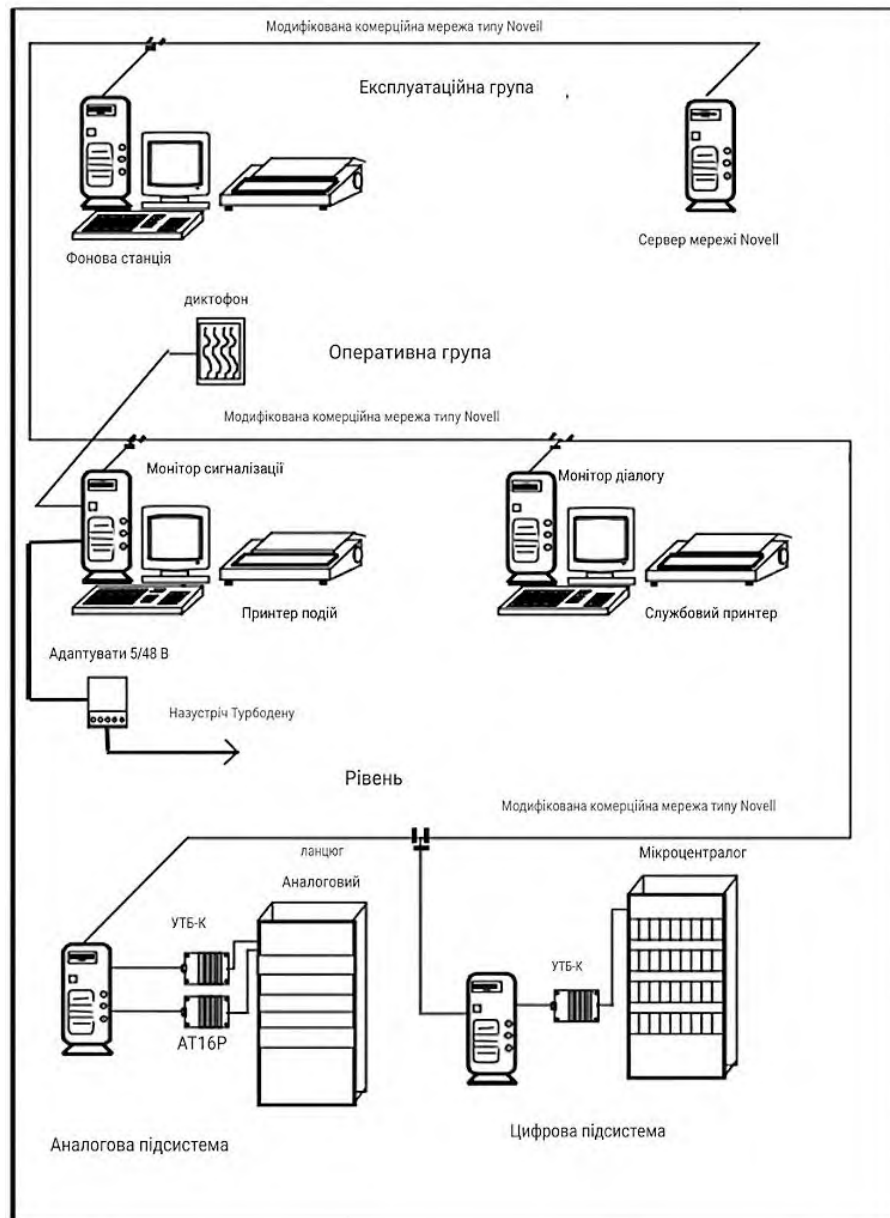


Рисунок 2.2 – Конфігурація комп'ютерної інформаційної системи управлінського обліку

Функції автоматизованої системи. В основному функції, що виконуються в автоматизованій системі, можна класифікувати за чотирма

типами: 1. Функції управління 2. Функції обробки даних та контролю за роботою установки. 3. Функції зв'язку. 4. Інформаційно-обчислювальні функції [15, 17].

Функції управління. Функції управління є необхідними та мають вирішальне значення для забезпечення належного функціонування системи. Серед них можна виділити: встановлення режимів роботи процесу; формування та передача сигналів до пристроїв управління; створення послідовностей команд для запуску, зупинки, реагування на ненормальні умови тощо; прийняття рішень; організація передачі інформації; нагляд за загальним функціонуванням системи та координація факторів; встановлення функціональних пріоритетів між елементами системи та з іншими системами на різних рівнях; управління та експлуатація банків даних. Слід зазначити, що ці функції є типовими для ієрархічної системи автоматизації [18].

Функції обробки та управління процесом. Включають дії, які виконує автоматична система з точки зору власне управління, серед них: збір та перевірка даних; лінеаризація та калібрування вимірювань; арифметична та логічна обробка даних; аналіз критичних значень змінних (наприклад: межі, тенденції тощо); перетворення шкал та в інженерні одиниці виміру; нормалізація даних; корекція результатів; автоматичне регулювання (дрейф) тощо [18].

Комунікаційні функції. До цієї групи функцій належать такі: комунікація між елементами установи; комунікація між процесом і людиною; комунікація між різними підсистемами у можливих режимах роботи установи.

Інформаційно-обчислювальні функції [17]. Це функції, спрямовані на подання інформації та звітів оператору процесу, серед яких: індикація змінних та параметрів процесу; історичний запис інформації; виявлення відхилень змінних від встановлених меж (верхня межа, нижня межа, максимальний коефіцієнт зміни тощо); виявлення станів та блокування процесу; розпізнавання та обробка умов; непряме вимірювання параметрів, розрахованих на основі інших вимірювань; розрахунок та аналіз загальних

показників функціонування процесу; розрахунок техніко-економічних показників експлуатації процесу (наприклад: грами палива на кожну кВт/год, вироблену на теплоелектростанції); діагностика та прогнозування процесу, що є дуже корисним для своєчасного запобігання та проведення технічного обслуговування [15]; звіти про інциденти та тривожні сигнали процесу; підготовка інформації та обмін нею з іншими ієрархічними рівнями обробки.

Архітектурна основа системи була побудована з використанням патерна MVC з метою забезпечити поділ обов'язків між усіма її складовими шарами і тим самим спростити процес розробки [16], як це показано на рис. 2.3.

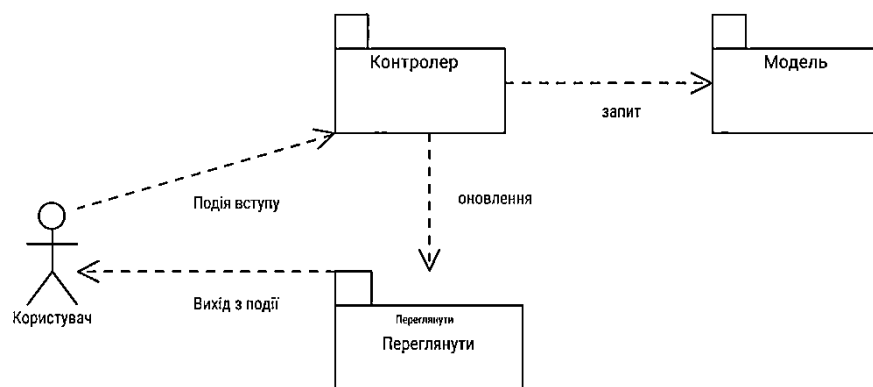


Рисунок 2.3 – Потік повідомлень в архітектурі MVC

В якості шаблону, вибраного для забезпечення незалежності подання від моделі, була використана архітектура MVC, яка виключає прив'язку моделі до подання, хоча уявлення, як і раніше, має доступ до форм, що знаходяться в моделі [17]. У системі взаємодія уявлень із моделлю здійснюється виключно через контролер. Архітектура MVC фактично є модифікованою реалізацією MVC0; основна зміна полягає в тому, що модель більше не запускає події у своїх уявленнях.

Система, що розробляється, має бути децентралізованою. Децентралізовані системи — це такі, в яких завдання розподіляються між елементами, що складають або входять до їхньої структури, що є проявом децентралізації «інтелекту» і, звісно, передбачає наявність більш ніж одного обчислювального модуля [18]. Ці системи можуть бути концентрованими або

розподіленими. Система є концентрованою, коли обчислювальні елементи обов'язково повинні бути розташовані в одному місці, тоді як у розподіленій системі обчислювальні елементи розташовані в віддалених місцях [18]. Логічну взаємодію модулів у децентралізованій системі зображено на рис.2.4.

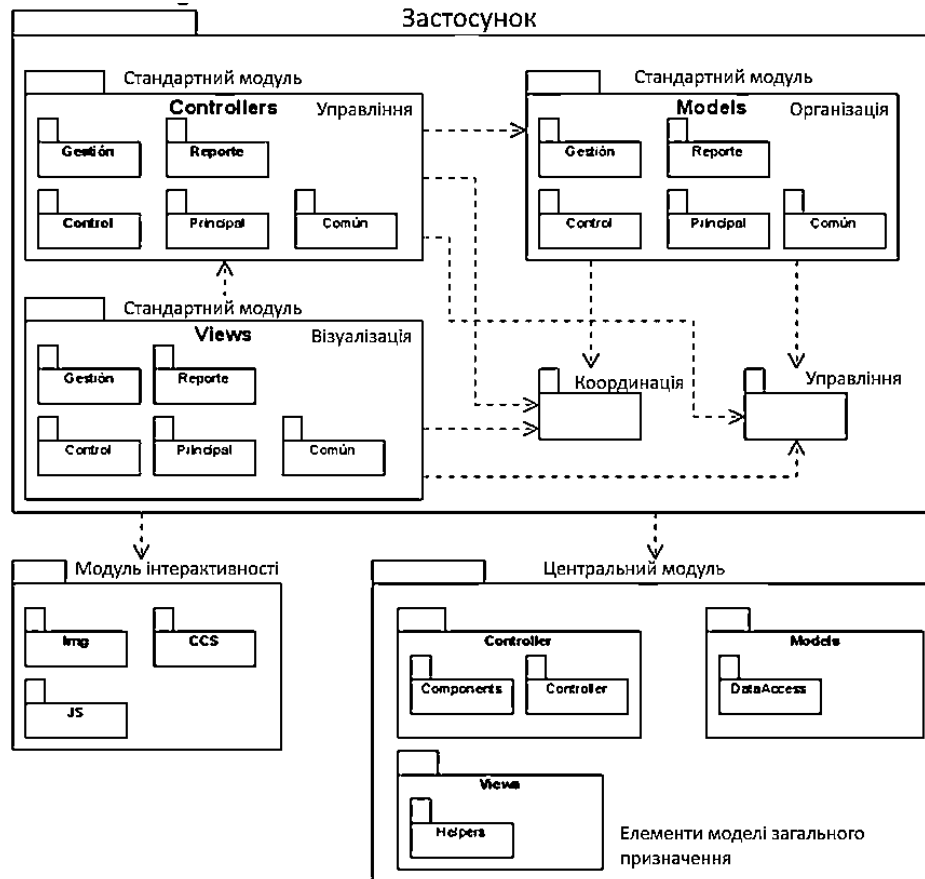


Рисунок 2.4 – Логічна структура взаємодії модулів системи

Системи розподіленого управління (СРУ) з'явилися як результат розвитку мікропроцесорів у середині 70-х років, коли з'явилися перші мікропроцесорні СРУ, призначені для заміни традиційних електронних приладових панелей, що склалися з цифрових систем, які забезпечували кращу компактність та візуалізацію інформації, використання відеотерміналів, розподілених функцій контролерів та елементів, що дозволяли легше виконувати функції регулювання, стратегії управління, послідовності управління, синхронізації та підрахунку, щоб назвати лише деякі з них. Таким чином, можна спроектувати структуру застосунку автоматизованої системи управлінського обліку [18].

### 2.3. Розробка функціональної моделі програмного застосунку

Функціональна модель показує послідовність подій, що виникають під час виконання функцій блоками проектованої системи. Рекомендується використовувати розроблений шаблон, оскільки він змушує всі запити, що надходять до додатка, проходити через контролер. Контролер забезпечує єдину точку входу, яка контролює та управляє запитам, що надходять від клієнтів системи. Завдяки цій єдиній точці входу не доводиться повторювати одну й ту саму логіку управління [18, 19]. Зазвичай він використовується разом із диспетчером, який відповідає за перенаправлення потоку виконання до відповідного модулю. Цей диспетчер розподілу дій може бути реалізований самим контролером або знаходитися в окремому класі. Робота функціональної моделі приведена на рис. 2.5.

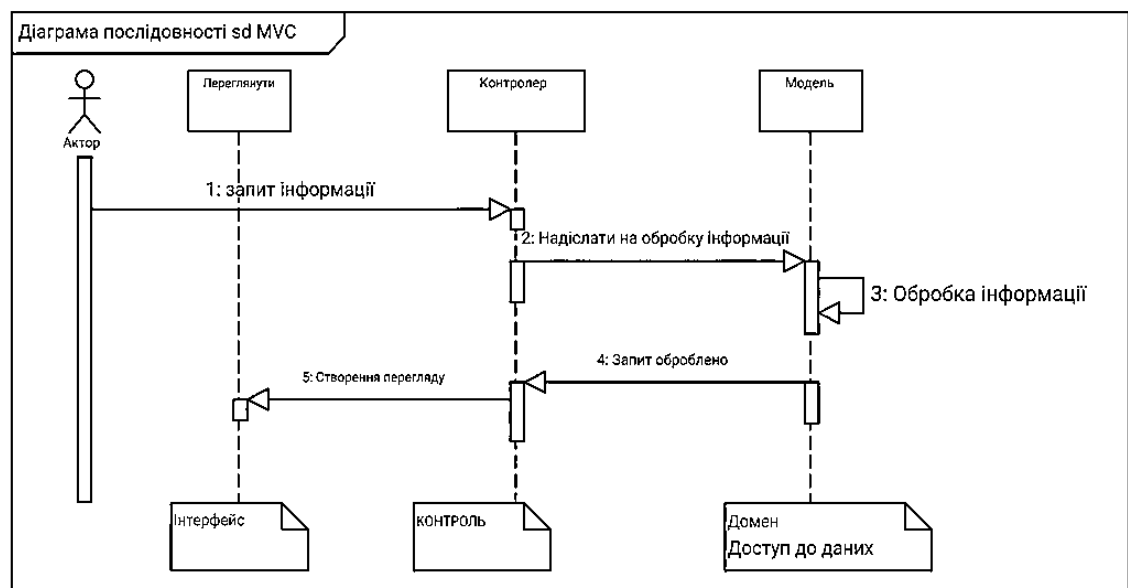


Рисунок 2.5 – Функціональна діаграма послідовності моделі MVC

Шаблон «Controller» виконує роль єдиного файлу входу в додаток. Відповідальність за виконання завдання покладається на клас, який має або може мати відповідні дані (атрибути). Клас містить усю необхідну інформацію для виконання дорученого йому завдання [19]. Слід враховувати, що це застосовується, поки розглядаються ті самі аспекти системи: бізнес-логіка.

Необхідно покласти відповідальність за контроль потоку подій системи на конкретні класи. Це полегшує централізацію дій (перевірки, безпека тощо). Контролер не виконує ці дії, а делегує їх іншим класам, з якими підтримує модель високої згуртованості [19]. Дуже поширеною помилкою є покладення на нього надмірної відповідальності та високого рівня зв'язку з рештою компонентів системи. З урахуванням останньої обставини функціональну схему можна відобразити наступним чином, рис. 2.6.

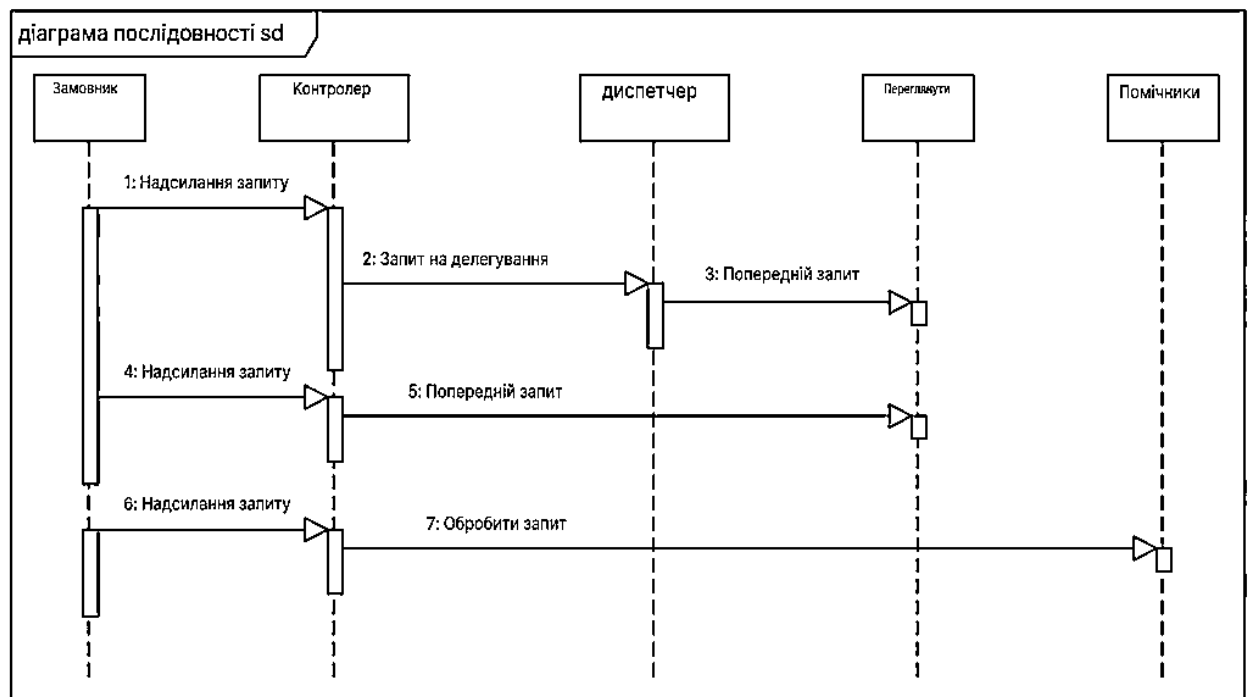


Рисунок 2.6 – Схема послдовальности работы шаблона Controller

Мовні шаблони використовуються в процесах впровадження, обслуговування та розгортання; їх зазвичай визнають стандартами кодування та проектування [19]. Вони описують, як кодувати та представляти загальновідомі типові операції в новому середовищі або в межах групи, забезпечуючи читабельність та передбачуваність. Розглянемо функціональну схему впровадження блоків застосунку, якій проектується [19]. На схемі в першу чергу показано розміщення основних компонентів, що входять до складу системи, які показані на рис. 2.7.



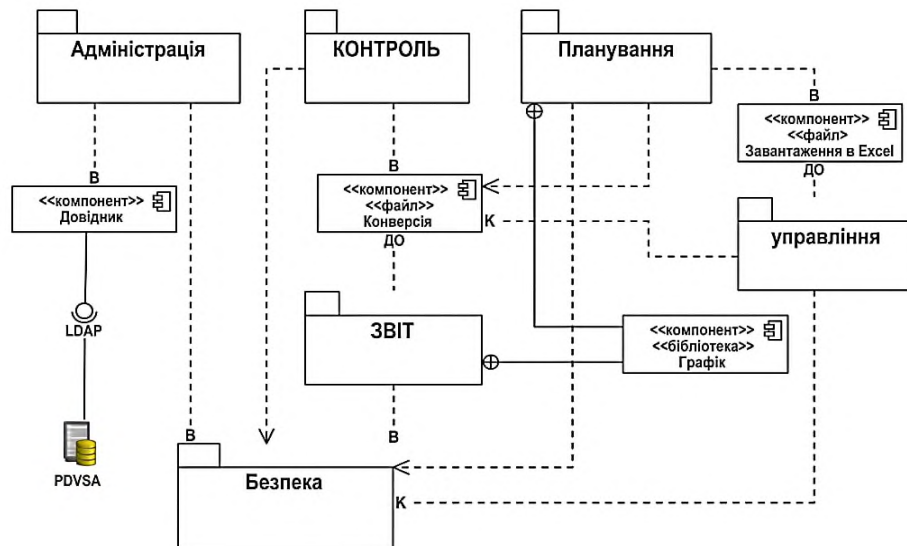


Рисунок 2.7 – Функціональна схема впровадження блоків застосунку

У загальному сенсі та з точки зору управління існує два типові способи організації автоматизованих систем: централізована та децентралізована структура. Система вважається централізованою, коли вона має єдиний орган управління, який відповідає за організацію роботи решти системи. У таких системах усі сигнали від датчиків надходять до комп'ютера, звідки сигнали управління повертаються до відповідних виконавчих механізмів [19].

Загальна схема структури комп'ютерної інформаційної системи, встановленої на підприємстві, відповідає розробленої моделі. На нижньому рівні використовуються зазвичай два комп'ютера, один для зчитування аналогових сигналів, а інший — для дискретних сигналів. На другому рівні знаходяться два ПК: один для підсистеми сигналізації, а інший — для зображень, діалогів, звітів тощо [17]. На верхньому рівні розташований ПК для експлуатації системи, тобто там, де уповноважений персонал здійснює зміни, заповнює базу даних тощо. Крім того, є сервер для всієї системи. Використовується локальна мережа з модифікаціями для роботи в режимі реального часу [19].

До основних функцій системи належать такі: безперервне та надійне зчитування, зберігання та відображення інформації про стан компонентів

системи підприємства, інших вимірювань та сигналів. Виконання команд управління, ініційованих оператором. Повідомлення оператора про виявлені зміни на об'єкті, як ті, що не вважаються нормальними, так і зміни, що відбуваються в ході повсякденної експлуатації об'єкта. Зміни зберігаються в системі для подальшого аналізу [19]. Загальні додатки, засновані на інформації, отриманій системою, такі як: звіти, графіки тенденцій, історія змінних, розрахунки, прогнози, виявлення аномалій тощо.

#### 2.4. Розробка схеми бази даних автоматизованої системи управлінського обліку і контролю кадрів

Розробка підходу для створення бази даних системи управлінського обліку і контролю кадрів передбачає досягнення таких цілей: визначення понять баз даних та алгоритмів, розробка алгоритму, що полегшує застосування методології, яка базується на етапах проектування баз даних. Поняття методології — це сукупність послідовних методів, пов'язаних між собою спільними принципами. Етап методології полягатиме у вивченні моделей та етапів проектування бази даних, які будуть використовуватися під час проведення дослідження, розробці алгоритму та аналізі його рівня складності [18]. Цікавість у пропозиції алгоритму, що полегшує проектування бази даних, полягає в тому, що це логічний та простий у виконанні процес.

Методологія — це сукупність методів, які допомагають передбачити успіх системи, тоді як модель — це послідовність кроків, яких слід дотримуватися для планомірного досягнення мети, тобто кроки, правила та процедури для вирішення проблеми; з іншого боку, методологія здійснює аналіз сильних і слабких сторін на основі моделей, пам'ятаючи, що не існує методу, кращого за інший, все залежатиме від системи бази даних, яку потрібно впровадити [18]. Була розроблена модель з метою спрощення проектування бази даних за допомогою діаграм «сутності-зв'язок». Ставиться завдання представити логічну структуру бази даних у вигляді діаграми.

Визначаються основні елементи проектування бази даних, що стосуються сутностей, атрибутів або полів та відносин між сутностями. На сьогоднішній день існує кілька моделей баз даних, серед яких можна згадати: ієрархічну модель бази даних, реляційну модель, мережеву модель, об'єктно-орієнтовану модель бази даних, модель «сутності-відношення», модель документів, модель «сутності-атрибут-значення», зіркоподібну схему, об'єктно-реляційну модель.

Загальна схема бази даних обліку персоналу представлена на рис. 2.8.

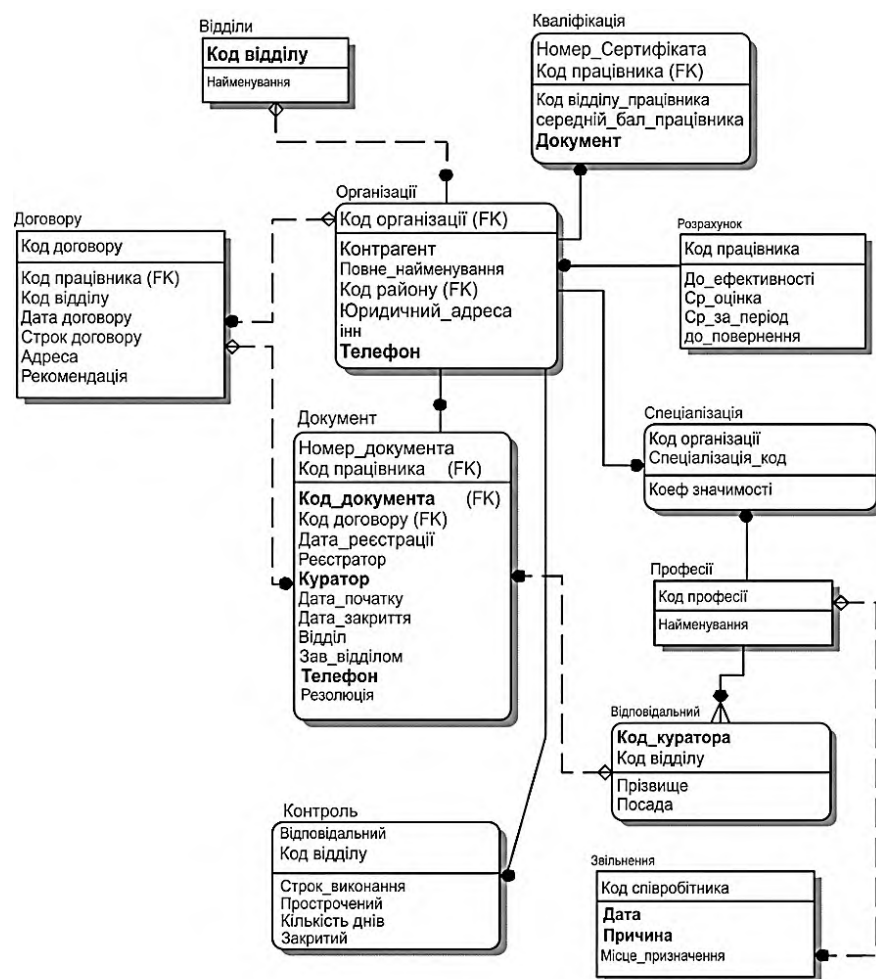


Рисунок 2.8 – Інформаційна модель бази даних управлінського обліку персоналу промислового підприємства

Для реалізації методології побудови бази даних пропонується використовувати загальний алгоритм [19]. Алгоритм — це послідовність кроків, які слід виконувати в певному порядку та за обмежений проміжок часу; кожен наступний крок вказує, що саме потрібно зробити. Алгоритм завжди

має початок і кінець, тому слід бути на сто відсотків упевненим, що за його допомогою вдасться досягти мети, намагаючись при цьому допустити мінімальну кількість помилок. Абстрактним способом формулювання стратегії є «алгоритм», тобто послідовність інструкцій, кожна з яких представляє чітко визначене завдання і може бути виконана за скінченний проміжок часу та з використанням скінченної кількості обчислювальних ресурсів [20]. Основною вимогою є те, що алгоритм повинен завершуватися за скінченну кількість кроків, завдяки чому його можна використовувати як інструкцію в іншому, більш складному алгоритмі. За допомогою методології алгоритм оптимізує час створення бази даних [19, 20].

На цьому етапі слід використовувати реляційну модель. Як уже зазначалося, ця модель базується на таблицях, які мають бути пов'язані між собою. Використання цієї моделі допомагає краще зрозуміти, як пов'язані дані, та перевірити, чи дотримуються бізнес-правила, надані клієнтом. Сутності перетворюються на таблиці, а відносини типу «багато до багатьох» стають таблицями [20]. При роботі з пов'язаними таблицями можна додавати інформацію, відому як записи або кортежі, і перевіряти відсутність дублювання інформації. У разі необхідності атрибуту можна перенести до первинного ключа, щоб усунути дублювання інформації. Крім того, модель застосовує нормалізацію [19]. Нормалізація — це процес спрощення взаємозв'язків між полями запису. За допомогою нормалізації набір даних у записі замінюється кількома записами, які є простішими та передбачуваними, а отже, і зручнішими в роботі. При проектуванні бази даних рекомендується дотримуватися принаймні третьої нормальної форми; це не гарантує повного контролю над надмірністю даних, але принаймні дозволяє обробляти інформацію в контрольований спосіб. Іншим дуже важливим питанням є безпека доступу до бази даних [20]. Цей процес повністю належить до компетенції розробника, але вважається, що логічна фаза є найбільш підходящою на етапі проектування для очищення схеми.

В основі бази даних лежать класифікатори, що показані в табл. 2.2.

Таблиця 2.2 – Класифікатори, що використовуються у системі управління персоналом

| Найменування кодової множини | Довжина коду | Система кодування | Вид класифікатора  |
|------------------------------|--------------|-------------------|--------------------|
| Код_працівника               | 4 байта      | Порядковий        | Загальносистемний  |
| Код_відділу                  | 1 байта      | Порядковий        | Загальносистемний  |
| Код_підрозділу               | 1 байта      | Порядковий        | Внутрішньофірмовий |
| Код_виду_роботи              | 4 байта      | Порядковий        | Загальносистемний  |
| Номер_договору               | 4 байта      | Порядковий        | Внутрішньофірмовий |
| Код_посади                   | 4 байта      | Порядковий        | Загальносистемний  |
| Код_спеціальності            | 4 байта      | Порядковий        | Загальносистемний  |
| Ід_код                       | 1 байт       | Порядковий        | Внутрішньофірмовий |
| Підпорядкованість_вгору      | 1 байт       | Порядковий        | Загальносистемний  |
| Підпорядкованість_вниз       | 1 байт       | Порядковий        | Загальносистемний  |

У цьому питанні існують розбіжності. Є пропозиції, які передбачають включення в схеми «сутності-зв'язку» багатозначного атрибута як окремої сутності, а інші пропонують зображати його як атрибут. У нашому випадку, незалежно від того, чи є атрибут простим чи складеним, якщо відомо, що він матиме обмежену і не надто велику кількість входжень, то він стане частиною сутності, яку описує (за умови, що поняття, яке він представляє, не пов'язане з іншими сутностями схеми E/R) [21]. Прикладом може бути: «Щодо співробітника цікаво зберігати його ІПН, ім'я, адресу та номери телефонів»; у цьому випадку атрибут «Телефон» є багатозначним атрибутом сутності «Співробітник». Наприклад, Викладач характеризується своїм ім'ям, ІПН, адресою та кампусами, на яких він викладає. У цьому випадку атрибут «Кампус» також міг би бути багатозначним атрибутом сутності Професор, але він пов'язаний з іншими сутностями і повинен відображатися як сутність.

## 2.5. Розробка алгоритмів програмних методів системи автоматизації управлінського обліку

Розробимо алгоритми функціонування модулів системи автоматизації управлінського обліку. Для цього узагальнимо взаємозв'язок між аналізом проблеми та отриманим рішенням (елементами) системи [21]. Можна помітити, що єдиного рішення проблеми ніколи не існує; завдання проектувальника полягає в тому, щоб знайти найкраще з можливих рішень, застосовуючи критерій техніко-економічної оцінки, який не завжди є абсолютним [21]. Від аналізу статичних та динамічних характеристик сигналів, що вимірюються, залежить проектування системи збору даних, для якої також існують різні альтернативні рішення.

Слід враховувати кількість транзакцій і запитів, які необхідно виконувати в базі даних, а також обсяг дискового простору, з огляду на вертикальне розширення системи [20]. Вертикальна масштабованість є найпростішою, оскільки передбачає розширення апаратного забезпечення одного з вузлів, тобто заміну апаратних компонентів на більш потужні, таких як жорсткий диск, пам'ять, процесор тощо; однак це також може бути повна заміна апаратного забезпечення на більш потужне. Зусилля, необхідні для такого зростання, мінімальні, оскільки це не має наслідків для програмного забезпечення, адже потрібно лише створити резервну копію та перенести системи на нове обладнання [21]. На завершення етапу не можна не згадати про документацію: завжди важливо документувати та підписувати угоди. Клієнт завжди має бути залучений до кожного етапу проектування. Слід пам'ятати, що на цьому етапі підтверджуються вимоги щодо апаратного забезпечення та СУБД, тому проектувальник повинен знати перелік можливих витрат та зручність придбання, а також рівні безпеки, де клієнт вирішуватиме, що він може покрити, а що ні.

Блок-схема алгоритму обліку документів про персонал показана на рис. 2.9.

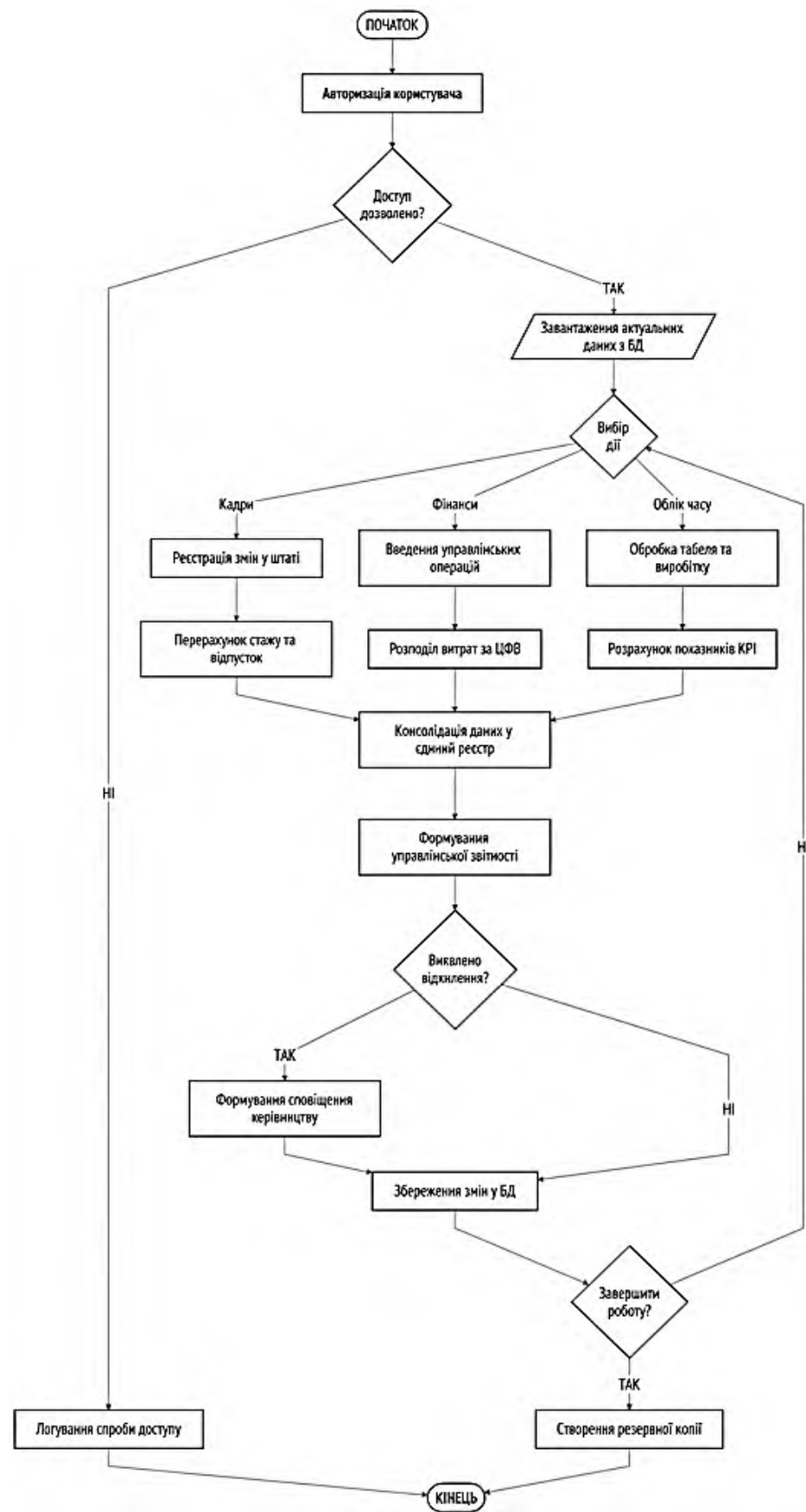


Рисунок 2.9 – Загальна блок-схема алгоритму обліку документів про персонал

Блок-схема — це графічне зображення алгоритму, яке наочно демонструє кроки або процеси, що необхідно виконати для вирішення задачі. Правильне її складання є надзвичайно важливим, оскільки саме на її основі пишеться програма на певній мові програмування [22]. Якщо блок-схема є повною та правильною, перенесення її на мову програмування є відносно простим і зрозумілим. Важливо підкреслити, що блок-схема показує місця походження та призначення даних, перетворення, яким піддаються дані, місця, де дані зберігаються в системі, та канали, якими циркулюють дані [21].

Визначення: узагальнення — це розумове об'єднання об'єктів у алгоритмічну концепцію; це надзвичайно складний процес, під час якого розумова діяльність спрямована на виявлення суттєвих і загальних рис математичного та/або обчислювального об'єкта, а також на встановлення взаємозв'язків між ними, утворюючи нові властивості, зв'язки та правила, які конкретизуються за допомогою прикладів [21]. Це передбачає поглиблення та збагачення знань особи для встановлення певних спільних і водночас суттєвих ознак між різними об'єктами або явищами, що аналізуються. Важливість її формування: це базова навичка для індукції, тобто для отримання припущень щодо сукупності об'єктів, явищ або відносин на основі аналізу окремих або особливих випадків [18]. Вона є дуже важливою для алгоритмізації, оскільки йдеться не про пошук алгоритму, який дасть рішення для кожної з поставлених конкретних проблемних ситуацій, а про пошук загальних моделей алгоритмів, що вирішують класи задач. Узагальнення може бути не тільки у формі висновку або визначення понять, а й у формі пояснень або доказів.

Для розробки алгоритмів необхідне швидко та усвідомлено засвоювати математичний та обчислювальний зміст процесу. Зв'язок з іншими блоками системи відбувається за допомогою порівняння, яке підкреслює загальність низки об'єктів або явищ, та абстрагування від загального, щоб дійти до глибокого розуміння реальності через поняття, судження та висновки [20]. У ній присутні такі фундаментальні навички, як аналіз, синтез, абстрагування, ідентифікація. Вона конкретизується за допомогою алгоритму, псевдокоду або



блок-схем як форми висновку, що перетворює її на узагальнені знання. За допомогою цих принципів побудуємо блок-схему алгоритму взаємодії застосунку з базою даних, рис. 2.10.

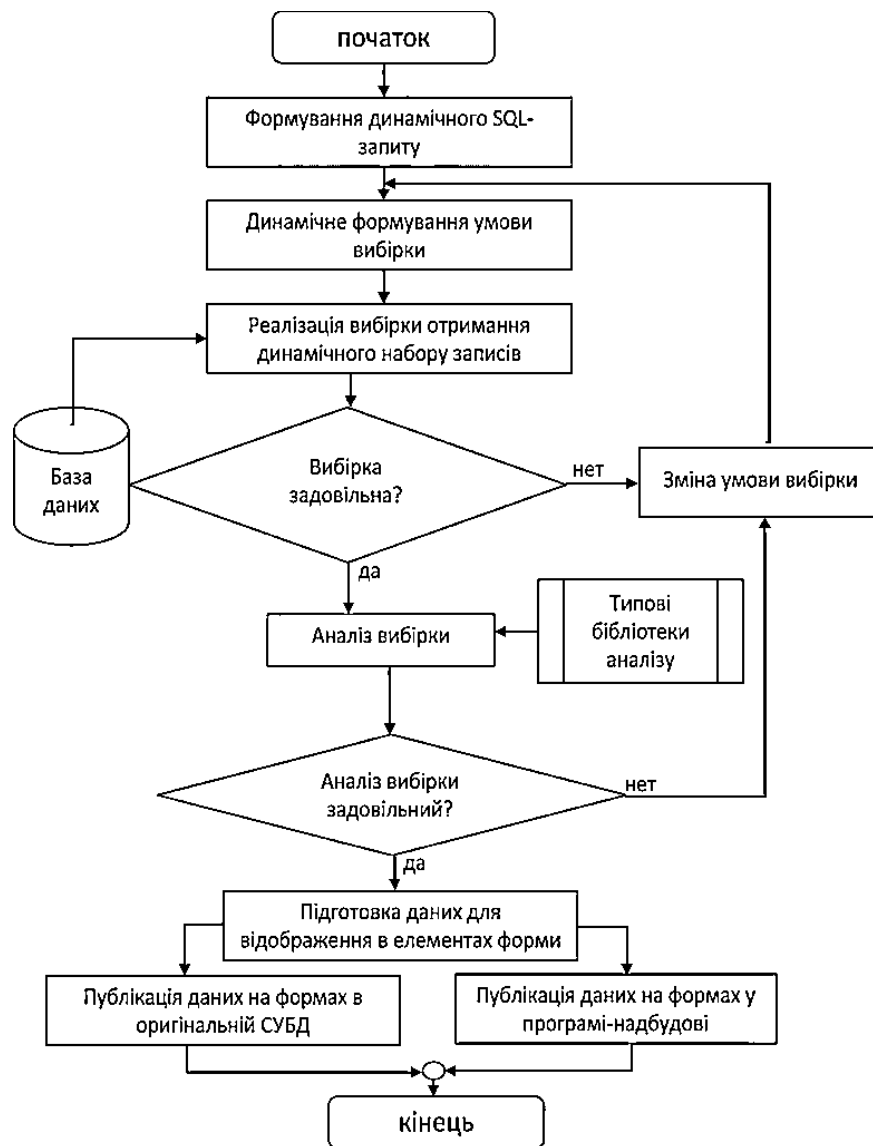


Рисунок 2.10 – Блок-схема алгоритму взаємодії застосунку з базою даних

Прикладні системи призначені для задоволення потреб підприємства чи організації. Усі підприємства мають багато спільного: їм потрібно наймати персонал, виставляти рахунки, вести бухгалтерський облік тощо [21]. Компанії, що працюють в одній галузі, використовують загальні галузеві програми. Так, окрім загальних бізнес-функцій, усім телекомунікаційним компаніям потрібні системи для з'єднання дзвінків, управління мережами та виставлення рахунків клієнтам тощо [22].

## 2.6. Проєктування візуального інтерфейсу програмного комплексу системи автоматизації управлінського обліку

У сфері взаємодії людини з машиною до сьогоднішнього дня використовуються два основні підходи, які ґрунтуються на двох різних аспектах відносин людини з машиною. Один із них є діалогічним і з'явився раніше. Вона обирає звичайну мову для спілкування з комп'ютером. Прикладом цього є операційні системи DOS та Unix, де всі команди вводяться з командного рядка (command-line) за допомогою клавіатури [22].

Другий підхід є більш сучасним. Вона характеризується як ергономічна і спрямована на максимально ефективне використання різних периферійних пристроїв, використовуючи опції екранів, зображень, графіки, команд, з метою підвищення функціональності, зручності використання та швидкості взаємодії з комп'ютером [22]. У цьому контексті взаємодія з користувачем здійснюється через графічний інтерфейс користувача (GUI).

Предметом інтерфейсу є комунікація, тобто передача повідомлень, а також процеси перетворення, необхідні для успішної реалізації цієї передачі. Комунікація користувача з додатком реалізується через спосіб його представлення і є посередником між користувачем та системою, що визначається як інтерфейс (interface) [21].

Інтерфейс користувача (user interface) — це те, що бачить користувач під час навігації по програмному забезпеченню; тобто це технологічний засіб взаємодії двох окремих систем — людини та комп'ютера. З цією метою він виконує всі необхідні операції кодування, декодування та управління потоком інформації [23].

Інтерфейс користувача відіграє дуже важливу роль, особливо в системах із високим рівнем складності та автоматизованих функцій. Це пов'язано з тим, що відомі випадки, коли такі системи створюють додаткові труднощі, які в кінцевому підсумку доводиться вирішувати користувачеві. Така ситуація стає більш зрозумілою, якщо врахувати, що користувач звик до автоматичних

функцій і процесів, потрапляючи таким чином у стан бездіяльності [22]. Таким чином, користувач має зіткнутися з ситуацією, яка вимагає швидких реакцій та рішучості, тоді як система, замість того щоб полегшити йому завдання, перевантажує його непотрібною інформацією, що ускладнює його роботу [18]. Саме такі ситуації роблять важливими ролі користувацького інтерфейсу, а також проектування взаємодії людини з машиною. Основні аспекти діалогу з користувачем наведені в табл. 2.3.

Таблиця 2.3 – Повідомлення системи у процесі діалогу з користувачем

| Повідомлення                               | Пояснення                                                                                   | Дії у відповідь                                                 |
|--------------------------------------------|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| Неправильний пароль                        | Повідомлення з'являється у разі неправильного введення пароля під час входу до системи [17] | Ввести вірний пароль                                            |
| Доступ заблокований                        | Користувач ввів неправильний пароль більше 3 разів                                          | Зв'язатися з адміністратором системи                            |
| Цей обліковий запис відсутній              | Користувач ввів неправильне ім'я користувача, або обліковий запис відсутній [18]            | Ввести правильне ім'я користувача або зареєструватися в системі |
| На даний звітний період відомості відсутні | На даний звітний період у базі даних відсутні відомості                                     | Вибрати інший звітний період                                    |
| Звіт зараз недоступний.                    | Дані для звіту формуються адміністратором                                                   | Повторити спробу отримання звіту пізніше                        |
| Одне з полів форми не заповнене            | Відсутні дані в одному з полів форми                                                        | Заповнити порожнє поле введення                                 |

Важливу роль в оцінці системи відіграє поняття зручності використання (usability). Англійський термін usable означає «придатний для використання».

Термін «зручність роботи» був затверджений на противагу популярному ще кілька років тому терміну «зручність використання». Міжнародний стандарт ISO 9241 визначає зручність використання системи як її здатність функціонувати ефективно та результативно, забезпечуючи при цьому суб'єктивне задоволення її користувачів [15]. Ці процеси показані на рис. 2.11.

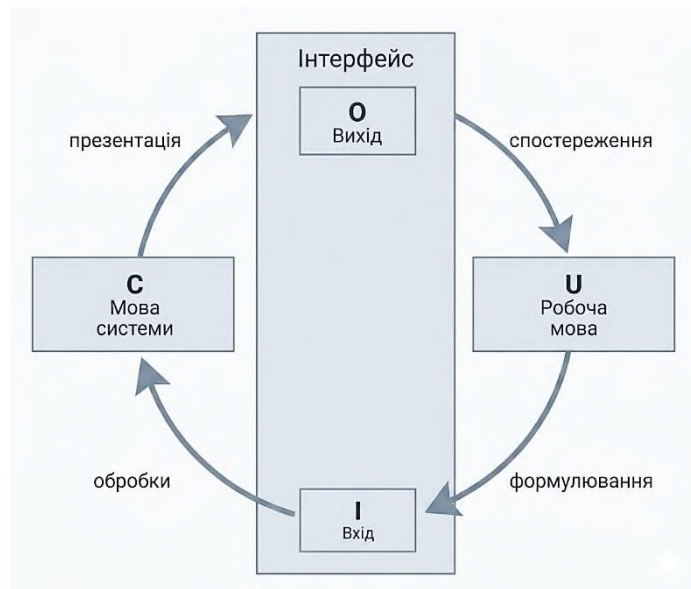


Рисунок 2.11 – Процес взаємодії с інтерфейсом користувача

Кожен із цих компонентів має свою мову. Взаємодія вимагає перекладу між різними мовами. Проблеми виникають, коли переклад неможливо виконати правильно [21]. Як бачимо на рис. 2.11, взаємодія поділяється на чотири складові:

- користувач (U) спілкується з системою через вхід (Input). Цей процес називається артикуляцією (articulation) і описує запит мовою завдань (task language);
- вхід (Input) передає команду через процес виконання (performance), і команда виконується системою (S) мовою системи (core language) [21];
- система видає на вихід (output) результат дії, і контроль повертається до користувача через фазу презентації (presentation);
- користувач відновлює контроль через фазу спостереження (observation) і переформулює нові цілі.

Термін «стиль взаємодії» охоплює всі способи, за допомогою яких користувачі спілкуються або взаємодіють з комп'ютерними системами. Інтерактивні системи та, загалом, додатки широко використовують ці категорії взаємодії [22]. У більшості випадків ці стилі можуть співіснувати, але те, які стилі в кінцевому підсумку будуть включені, залежить від функціональності самої системи та випадків її використання.

Приклади макетів інтерфейсу користувача системи, яка розробляється показані нижче на рис. 2.12 та рис. 2.13.

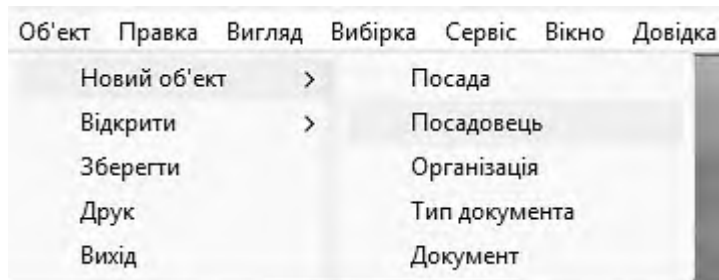


Рисунок 2.12 – Макет головного меню застосунку

Рисунок 2.13 – Макет форми для вводу даних

Взаємодія дає користувачеві можливість кодувати, організовувати, обробляти та/або перетворювати отриману інформацію. По суті, це можливість «відповіді» системи на дію користувача. Ця відповідь надається у вигляді поля альтернативних варіантів, що залежать від вибору користувача.

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗАЦІЇ УПРАВЛІНСЬКОГО ОБЛІКУ

#### 3.1. Розробка програмних модулів автоматизованої системи управління обліку і контролю кадрів

Загальна модель процесу розробки програмного забезпечення включає набір структурних та загальних заходів, дій та робочих завдань. Кожна з моделей процесу може бути описана за допомогою окремого технологічного потоку: опису того, як структурні заходи, дії та завдання організовані послідовно та хронологічно. Шаблони процесу можуть використовуватися для вирішення типових проблем, що виникають у процесі розробки програмного забезпечення [18].

Описові моделі процесу застосовуються вже багато років з метою введення порядку та структури у розробку програмного забезпечення. Кожна з таких моделей пропонує дещо відмінний потік процесу, але всі вони передбачають виконання одного й того ж набору загальних структурних заходів: комунікація, планування, моделювання, побудова та розробка [22].

Послідовні моделі процесів, такі як каскадна та V-модель, є найстарішими парадигмами у сфері програмного забезпечення. Вони передбачають лінійний хід процесу, який часто не відповідає сучасним реаліям (постійні зміни, системи, що розвиваються, стислі терміни тощо) світу програмного забезпечення. Проте вони мають застосування в ситуаціях, коли вимоги чітко визначені та стабільні [19].

Атрибути даних визначають властивості об'єкта даних і мають одну з трьох різних характеристик. Вони використовуються для іменування екземпляра об'єкта даних, опису екземпляра або посилання на інший екземпляр в іншій таблиці [20]. Крім того, один або кілька атрибутів повинні бути визначені як ідентифікатор — тобто атрибут-ідентифікатор стає «ключем», коли потрібно знайти екземпляр об'єкта даних. У деяких випадках

значення для ідентифікатора (ідентифікаторів) є унікальними, хоча це не є обов'язковою вимогою. Щодо об'єкта даних, розумним ідентифікатором може бути серійний номер об'єкту даних [21].

На лістингу нижче, рис. 3.1 наведено фрагмент програмного коду додавання нового об'єкта персоналізованого обліку кадрів.

```
#include "main.h"
#include "org.h"
#include "DataModule.h"
#include "AddOccupation.h"
#include "AddPersona.h"
#include "AddOrganization.h"
#include "AddDocType.h"
#include "AddDocument.h"
#include <fstream.h>
#include "Open.h"
#include "PrintPreview.h"
#include "Search.h"
#include "ElectionCondition.h"
#include "Rpt1.h"
#include "DB.hpp"
#include "about.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
__fastcall TfrmMain::TfrmMain(TComponent* Owner)
    : TForm(Owner)
{
}

void __fastcall TfrmMain::N25Click(TObject *Sender)
{
    frmMain->Close();
}

void __fastcall TfrmMain::N26Click(TObject *Sender)
{
    Application->CreateForm(__classid(TfrmAddOrganization), &frmAddOrganization);
}

void __fastcall TfrmMain::N4Click(TObject *Sender)
{
    if(ActiveMDIChild->Caption == "Додавання нової посади")
    {
        frmAddOccupation->SaveOccupation();
        frmAddOccupation->Close();
        return;
    }

    if(ActiveMDIChild->Caption == "Додавання нового посадовця")
    {
        frmAddPersona->SavePersona();
        frmAddPersona->Close();
        return;
    }
}
```

Рисунок 3.1 – Фрагмент програмного коду додавання нового об'єкта персоналізованого обліку кадрів

Набір атрибутів, що підходить для певного об'єкта даних, визначається з урахуванням контексту задачі. Атрибути для посадовця могли б добре підійти для програми, яку використовує відділ кадрів, але були б марними для автомобільної компанії, якій потрібно створити програмне забезпечення для управління виробництвом [20].

Тому програма має враховувати специфіку саме кадрового обліку. На лістингу рис. 3.2 показано процедури додавання нової організації та нового документа кадрового обліку.

```
if(ActiveMDIChild->Caption == "Додавання нової організації")
{
    frmAddOrganization->SaveOrganization();
    frmAddOrganization->Close();
    return;
}

if(ActiveMDIChild->Caption == "Додавання нового типу документа")
{
    frmAddDocType->SaveDocType();
    frmAddDocType->Close();
    return;
}
```

Рисунок 3.2 – Процедури додавання нової організації та нового документа кадрового обліку

У загальному вигляді проблему компоновання програмного комплексу кадрового обліку як завдання обліку інформаційних об'єктів можна описати в такий спосіб [17]. Опис рішення є частиною простору проблеми. Але це рішення слід шукати після того, як визначено всі сутності. Класи аналізу проявляються в одному з таких видів:

- зовнішні об'єкти (наприклад, інші системи, пристрої та люди), які генерують або споживають інформацію, яку використовуватиме комп'ютерна система;

- речі (звіти, екрани, листи, сигнали тощо), що є частиною інформаційної сфери проблеми [20];

- події (такі як передача власності або виконання серії рухів роботом), що відбуваються в контексті функціонування системи;



- ролі (менеджер, інженер, продавець тощо), які виконують особи, що взаємодіють із системою;
- організаційні одиниці (підрозділ, група, команда тощо), що мають значення для застосування [21];
- місця (виробничий цех або вантажна платформа), що визначають контекст проблеми та загальну функцію системи;
- структури (датчики, чотириколісні транспортні засоби, комп'ютери тощо), що визначають клас об'єктів або пов'язані з ними класи.

У зв'язку з викладеними положеннями, основні етапи ініціалізації програми характеризуються наступним лістингом, рис. 3.3.

```
void __fastcall TfrmMain::N27Click(TObject *Sender)
{
    Application->CreateForm(__classid(TfrmAddPersona), &frmAddPersona);
}

void __fastcall TfrmMain::N31Click(TObject *Sender)
{
    Application->CreateForm(__classid(TfrmAddDocType), &frmAddDocType);
}

void __fastcall TfrmMain::N28Click(TObject *Sender)
{
    Application->CreateForm(__classid(TfrmAddDocument), &frmAddDocument);
}
```

Рисунок 3.3 – Етапи ініціалізації програми

Операції визначають поведінку об'єкта. Хоча існує багато різних типів операцій, зазвичай їх поділяють на чотири основні категорії: операції, що певним чином обробляють дані (наприклад, додають, видаляють, редагують, відбирають тощо), операції, що виконують обчислення, операції, що запитують про стан об'єкта, та операції, що стежать за об'єктом щодо настання контрольної події [18]. Ці функції виконуються за допомогою операцій над атрибутами або їхніми асоціаціями. Отже, операція повинна «знати» про природу атрибутів та асоціацій класу. Щоб визначити, чи правильно розподілена інтелектуальна функціональність системи, слід проаналізувати обов'язки, зазначені в кожній карті CRC, щоб з'ясувати, чи не має якийсь клас занадто довгого переліку обов'язків [22].

Це свідчить про концентрацію інтелектуальної функціональності.<sup>18</sup> Крім того, обов'язки кожного класу повинні мати однаковий рівень абстракції. Наприклад, серед операцій, перелічених для агрегованого класу під назвою «ПеревіритиОсобу», перевіряючий зазначає дві відповідальності: підрахувати кількість осіб та видалити завершені перевірки [20]. Перша операція (відповідальність) передбачає складну математичну та логічну процедуру. На рис. 3.4 показано лістинг методів додавання новій посади та нового посадовця.

```
void __fastcall TfrmMain::N5Click(TObject *Sender)
{
    if(ActiveMDIChild->Caption == "Додавання нової посади")
    {
        Application->CreateForm(__classid(TfrmPrintPreview), &frmPrintPreview);
        frmPrintPreview->Caption = "Друк назви посади";
        frmPrintPreview->pre->Lines->Add(" ");
        frmPrintPreview->pre->Lines->Add("Дата звіта: " + Date().DateString());
        frmPrintPreview->pre->Lines->Add(" ");
        frmPrintPreview->pre->Lines->Add(" ПОСАДА: " + frmAddOccupation->editNewOccupation->Text);
        return;
    }

    if(ActiveMDIChild->Caption == "Додавання нового посадовця")
    {
        Application->CreateForm(__classid(TfrmPrintPreview), &frmPrintPreview);
        frmPrintPreview->Caption = "Друк відомостей о посадовце";
        frmPrintPreview->pre->Lines->Add(" ");
        frmPrintPreview->pre->Lines->Add("Дата звіта: " + Date().DateString());
        frmPrintPreview->pre->Lines->Add(" ");
        frmPrintPreview->pre->Lines->Add(" ПРИЗВИЩЕ: " + frmAddPersona->editFamilia->Text);
        frmPrintPreview->pre->Lines->Add(" ИМ'Я: " + frmAddPersona->editName->Text);
        frmPrintPreview->pre->Lines->Add(" ПО БАТЬКОВІ: " + frmAddPersona->editOtchestvo->Text);
        frmPrintPreview->pre->Lines->Add(" ОРГАНІЗАЦІЯ: " + frmAddPersona->cmbOrganization->Text);
        frmPrintPreview->pre->Lines->Add(" ПОСАДА: " + frmAddPersona->cmbOccupation->Text);
        frmPrintPreview->pre->Lines->Add(" КАБІNET: " + frmAddPersona->editCabinet->Text);
        frmPrintPreview->pre->Lines->Add(" ТЕЛЕФОН: " + frmAddPersona->editTelephone->Text);
        return;
    }
}
```

Рисунок 3.4 – Лістинг методів додавання новій посади та нового посадовця

Більшість функцій системи — це проста офісна діяльність. Оскільки ці операції не мають однакового рівня абстракції, видалення завершених перевірок слід розмістити серед обов'язків класу «ПеревіритиВхідніДані», який входить до агрегованого класу «ПеревіритиОсобу» [21].

На нижчому рівні абстракції процедури модель вимог описує обробку, яка буде здійснюватися за допомогою операцій класу аналізу. Ці операції

оперують атрибутами класу та взаємодіють як класи, що співпрацюють між собою для досягнення бажаної поведінки [20].

Незалежно від рівня абстракції процедури, діаграма активності використовується для представлення її деталей. На рівні аналізу діаграми активності слід використовувати лише там, де функціональність є відносно складною. Значна частина складності багатьох веб-додатків полягає не в функціях, які вони надають, а в характері інформації, до якої здійснюється доступ, та у способах її обробки [18].

На лістингу рис. 3.5 показано процедуру додавання нового документа про працівника.

```
if(ActiveMDIChild->Caption == "Додавання нового документа")
{
    AnsiString s;
    Application->CreateForm(__classid(TfrmPrintPreview), &frmPrintPreview);
    frmPrintPreview->Caption = "Друк відомостей о документе";
    frmPrintPreview->pre->Lines->Add(" ");
    frmPrintPreview->pre->Lines->Add("Дата звіта: " + Date().DateToString());
    frmPrintPreview->pre->Lines->Add(" ");
    frmPrintPreview->pre->Lines->Add(" ВХІДНИЙ НОМЕР: " + frmAddDocument->editNumber->Text);
    frmPrintPreview->pre->Lines->Add(" ВИХІДНИЙ НОМЕР: " + frmAddDocument->editOutgoingNumber->Text);
    frmPrintPreview->pre->Lines->Add(" ОБ'ЕМ (сторінок): " + frmAddDocument->editVolume->Text);
    s = frmAddDocument->chbExecuted->Checked ? "да" : "нет";
    frmPrintPreview->pre->Lines->Add(" ВИКОНАНО: " + s);
    frmPrintPreview->pre->Lines->Add(" НАЗВА: " + frmAddDocument->memoName->Text);
    frmPrintPreview->pre->Lines->Add(" РЕЗОЛЮЦІЯ: " + frmAddDocument->memoResolution->Text);
    frmPrintPreview->pre->Lines->Add(" ТИП: " + frmAddDocument->cmbType->Text);
    frmPrintPreview->pre->Lines->Add(" ОРГАНІЗАЦІЯ-ДЖЕРЕЛО: " + frmAddDocument->cmbSource->Text);
    frmPrintPreview->pre->Lines->Add(" ОРГАНІЗАЦІЯ-ПРИЙМАЧ: " + frmAddDocument->cmbReceive->Text);
    frmPrintPreview->pre->Lines->Add(" АВТОР: " + frmAddDocument->cmbAuthor->Text);
    frmPrintPreview->pre->Lines->Add(" КУРАТОР: " + frmAddDocument->cmbCurator->Text);
    frmPrintPreview->pre->Lines->Add(" ВИКОНАВЕЦЬ: " + frmAddDocument->cmbWorker->Text);
    frmPrintPreview->pre->Lines->Add(" МІСЦЕ РОЗТАШУВАННЯ: " + frmAddDocument->editAllocate->Text);
}
```

Рисунок 3.5 – Процедура додавання нового документа про працівника

### 3.2. Розробка бази даних автоматизованої системи управлінського обліку і контролю кадрів

Для практичної розробки бази даних необхідно визначитися з сутностями предметної області. Ми визначили, що «Співробітник» є узагальненням для «Інженера», «Секретаря» та «Техніка». Отже, перерозподілили атрибути між сутностями [18]. Розміщуємо ідентифікатор №-Співр. та загальні дескриптори Ім'я-Співр., Адреса-Сім'ї, Дата-Народження, Опис-Посади та Заробітна плата в супероб'єкті Співробітник, розміщуємо

специфічний дескриптор Спеціальність в об'єкті Інженер; поміщаємо специфічний дескриптор в об'єкт Секретар, і, нарешті, поміщаємо специфічні дескриптори Досвід та Роки-досвіду в об'єкт Робітник [21]. Ці залежності показані у табл. 3.1.

Таблиця 3.1 – Структура таблиці «Організація»

| Найменування       | Тип       | Опис                                          |
|--------------------|-----------|-----------------------------------------------|
| Контрагент         | числовий  | Первинний ключ                                |
| Повне найменування | строковий | Повна юридична назва Контрагента              |
| Код району         | числовий  | Зовнішній ключ, посилається на таблицю Райони |
| Юридична адреса    | строковий | Юридична адреса контрагента                   |
| ПІН                | строковий | ПІН контрагента                               |
| Телефон            | строковий | Номер контактного телефону контрагента        |

Сутність «Договори» містить інформацію про договори з працівниками підприємства, табл. 3.2 [20].

Таблиця 3.2 – Структура таблиці "Договори"

| Найменування   | Тип        | Опис                                               |
|----------------|------------|----------------------------------------------------|
| Код договору   | числовий   | Первинний ключ                                     |
| Код працівника | числовий   | Зовнішній ключ, посилається на таблицю Контрагенти |
| Код відділу    | числовий   | Код відділу на підприємстві                        |
| Дата договору  | дата / час | Дата укладення договору                            |
| Строк договору | числовий   | Передбачуваний строк продовження договору          |
| Адреса         | строковий  | Адреса працівника для договору                     |
| Рекомендація   | строковий  | Рекомендація працівника                            |

У таблиці «Документ» містяться дані щодо контролю інформації на працівника, табл. 3.3 [19].

Таблиця 3.3 - Структура таблиці "Документ"

| Найменування    | Тип        | Опис                                         |
|-----------------|------------|----------------------------------------------|
| Номер документа | числовий   | Первинний ключ                               |
| Код працівника  | числовий   | Код працівника                               |
| Код документа   | числовий   | Код документа в реєстрі документів           |
| Код договору    | числовий   | Код договору в реєстрі договорів             |
| Дата реєстрації | дата / час | Дата реєстрації документа про працівника     |
| Реєстратор      | строковий  | Відповідальна особа, яка реєструє працівника |
| Куратор         | строковий  | Особа, яка контролює документ про працівника |
| Дата початку    | дата / час | Дата відкриття документа на працівника       |
| Дата закриття   | дата / час | Дата закриття документа на працівника        |
| Відділ          | строковий  | Відділ, куди направляється працівник         |
| Зав відділом    | строковий  | Завідувач відділу працівника                 |
| Телефон         | строковий  | Телефон відділу працівника                   |
| Резолюція       | строковий  | Резолюція начальника відділу на документі    |

У таблиці «Відповідальний» міститься інформація про відповідальну особу, яка контролює працівника, таблиця 3.4 [22].

Таблиця 3.4 - Структура таблиці "Відповідальний"

| Найменування | Тип       | Опис                          |
|--------------|-----------|-------------------------------|
| Код куратора | числовий  | Первинний ключ                |
| Код відділу  | числовий  | Код відділу                   |
| Прізвище     | строковий | Прізвище відповідальної особи |
| Посада       | строковий | Посада відповідальної особи   |

Сутність "Контроль" містить інформацію про контроль документа відповідальною особою, табл. 3.5.

Таблиця 3.5 - Структура таблиці "Контроль"

| Найменування    | Тип        | Опис                                                               |
|-----------------|------------|--------------------------------------------------------------------|
| Відповідальний  | строковий  | Первинний ключ                                                     |
| Код відділу     | числовий   | Код відділу працівника                                             |
| Строк виконання | дата / час | Строк, протягом якого має бути розглянутий документ про працівника |
| Прострочений    | числовий   | Ознака прострочення документа                                      |
| Кіл-ть днів     | числовий   | Кількість днів прострочення розгляду документа                     |
| Закритий        | числовий   | Ознака закриття документа на працівника                            |

Сутність "Кваліфікація" містить інформацію про кваліфікацію працівника, яку він отримав до приходу на підприємство, де ведеться його облік та контроль, табл. 3.6.

Таблиця 3.6 - Структура таблиці "Кваліфікація"

| Найменування            | Тип       | Опис                                  |
|-------------------------|-----------|---------------------------------------|
| Номер Сертифіката       | строковий | Номер сертифіката працівника          |
| Код працівника          | числовий  | Код працівника                        |
| Код відділу працівника  | числовий  | Код відділу, в якому працює працівник |
| Середній бал працівника | числовий  | Середній бал за атестацією працівника |
| Документ                | числовий  | Код документа в таблиці документ      |

Таблиця «Кваліфікація» заповнюється за результатами тестування працівника відповідно до затвердженого плану проведення іспитів та атестації персоналу [16, 21].

Сутність «Звільнення» містить інформацію про причини звільнення працівників, а також про місце призначення працівника, табл. 3.7. При цьому має враховуватись дата звільнення для дотримання вимог законодавства. У таблиці вказується код співробітника, яким здійснюється зв'язок з таблицею «Договори».

Таблиця 3.7 - Структура таблиці "Звільнення"

| Найменування      | Тип        | Опис                              |
|-------------------|------------|-----------------------------------|
| Код співробітника | числовий   | Первинний ключ                    |
| Дата              | дата / час | Дата звільнення                   |
| Причина           | строковий  | Причина звільнення                |
| Місце призначення | строковий  | Нове місце призначення працівника |

У таблиці «Розрахунок» міститься інформація про результати розрахунку коефіцієнтів, що характеризують ефективність праці кожного працівника, табл. 3.8 [22].

Таблиця 3.8 - Структура таблиці "Розрахунок"

| Найменування   | Тип      | Опис                                                                                        |
|----------------|----------|---------------------------------------------------------------------------------------------|
| Код працівника | числовий | Код працівника                                                                              |
| К ефективності | числовий | Коефіцієнт ефективності праці працівника                                                    |
| Ср оцінка      | числовий | Середня загальна оцінка праці працівника                                                    |
| Ср за період   | числовий | Середня загальна оцінка праці працівника за певний контрольний період                       |
| К повернення   | числовий | Коефіцієнт, що характеризує швидкість повернення вкладених у працівника коштів підприємства |

У таблиці «Спеціалізація» міститься інформація про спеціалізацію працівника за фахом на підприємстві, табл. 3.9).

Таблиця 3.9 - Структура таблиці "Спеціалізація"

| Найменування      | Тип      | Опис                                           |
|-------------------|----------|------------------------------------------------|
| Код організації   | числовий | Код організації                                |
| Спеціалізація код | числовий | Код спеціалізації                              |
| Коеф значущості   | числовий | Коефіцієнт значущості спеціалізації працівника |

### 3.3. Розробка програмних модулів управління даними системи управлінського обліку і контролю кадрів

Тут розглядається детальніше поняття: працівники, які працюють на підприємстві. Об'єкт «працівник» ідентифікується за номером та має обов'язкові атрибути з одним значенням: «Ім'я», «Адреса» та «Контактний телефон» [19]. Щоб вказати, до якого відділу він прикріплений, необхідно включити бінарну взаємодію (Працює) з об'єктом відділ. Слід враховувати, що працівник може одночасно працювати лише в одній службі і що у службі має працювати принаймні один працівник (хоча ця інформація і не описана явно). Отже, тип відповідності взаємозв'язку «Працює» буде 1:N [21].

Первинний індекс (primary index) створюється на основі первинного ключа, точніше, на основі надмножини первинного ключа. На відміну від



цього, вторинний індекс, так званий «щільний» (dense index) створюється для кожного значення в таблиці. На відміну від цього, розріджений індекс (наприклад, одне значення для кожної сторінки таблиці) створюється як кластеризований (clustered index), для якого дані таблиці зберігаються на жорсткому диску в тому ж порядку, що й базовий індекс [15]. Для кожної таблиці можемо мати саме один такий індекс. З урахуванням цього створений програмний модуль підключення до відповідних таблиць бази даних. Лістинг показаний на рис. 3.6.

```
#include <vcl.h>
#pragma hdrstop
#include "DataModule.h"
#include "Rpt1.h"
#include "ElectionCondition.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TDataMod *DataMod;

__fastcall TDataMod::TDataMod(TComponent* Owner)
: TDataModule(Owner)
{
    dbName = "ctrldoc.mdb";
    AnsiString s;
    s = "Provider=Microsoft.Jet.OLEDB.4.0;";
    s += "Data Source=" + ExtractFilePath(Application->ExeName) + dbName + ".";
    s += "Persist Security Info=False;";

    ADOConn->ConnectionString = s;
    ADOConn->Open();
    cond1 = "";
    cond2 = "";
    condDate = "";
    condTipDocID = "";
    condDocID = "";
    condDatePlan = "";
    condNvxod = "";
    condNvixod = "";
    ReportHeader = "";
}
```

Рисунок 3.6 – Модуль підключення до відповідних таблиць бази даних

Для збереження інформації про дату та кількість призначених працівників необхідно додати багатозначний атрибут, що складається з цих даних [18]. Він буде багатозначним, оскільки один і той самий працівник міг отримати одну і ту саму посаду під час одного перебування працівника кілька разів і в різних посадах, але в різні дати. У тристоронньому взаємозв'язку «Призначення» слід враховувати, що не допускається, щоб два різних працівника отримували одну і ту саму посаду в точно один і той самий день.



Тому було б доцільніше включити інформацію про години та хвилини в дату призначення. Взаємозв'язок «Призначення» не означає, що кожен працівник отримав призначення, хоча мінімальна кардинальність взаємозв'язку дорівнює одиниці. Мінімальні кардинальності взаємозв'язку вказують, що завжди, коли було призначення на певну дату, принаймні одному працівнику це мав зробити принаймні один керівник і було виписано принаймні одне призначення [20]. Одна технологія підключення до джерела даних показано на рис. 3.7.

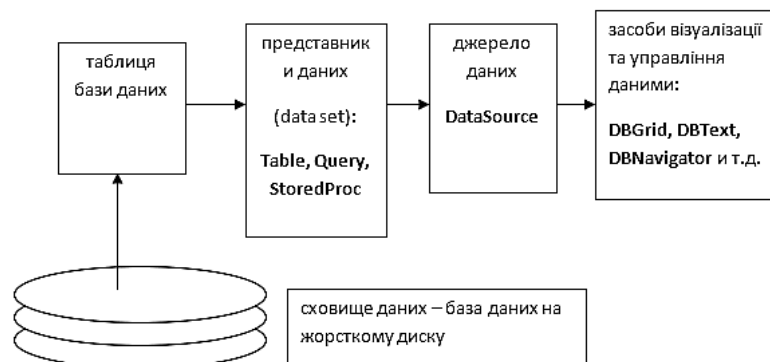


Рисунок 3.7 – Технологія підключення до джерела даних

Мета бази даних полягає в тому, щоб усі заходи з догляду та поліпшення, що здійснюються щодо всіх екземплярів, були зареєстровані. Для цього в БД зберігатиметься перелік заходів, яким може піддаватися будь-який екземпляр даних [22]. З одного боку, зберігаються всі деталі кожного екземпляра, включаючи тип, коментарі, фотографія та інформація про те, чи йдеться про деталі; ця інформація дозволяє порівняти одну структуру з іншою, щоб визначити, чи була робота виконана правильно і чи сприяла вона покращенню ефективності. Також реєструються виміри кожного екземпляра даних, нормалізація, із зазначенням типу нормалізації та опису, і, нарешті, знищення надлишків даних, щодо яких зберігається опис.

Методи, у свою чергу, оголошуються шляхом попереднього визначення рівня доступу. У цьому випадку метод має публічний доступ (для цього ми використовуємо зарезервоване слово `public`), що означає, що будь-який зовнішній об'єкт може викликати операцію, інкапсульовану в методі. Існують

й інші можливості, наприклад `private`, метод можна використовувати лише всередині класу, в якому він оголошений [21]. Після рівня доступу йде тип даних, який повертає метод, або не повертає жодного значення; щоб вказати на це, використовується інше. Лістинг модулю вибірки та видалення даних проілюстрований на рис. 3.8.

```
long TDataMod::LastIDRecord(AnsiString TableName)
{
    long n;
    cmd = "SELECT ";
    cmd += "    MAX([id]) AS [IDmax] ";
    cmd += "FROM ";
    cmd += "    [" + TableName + "] ";
    ADODataset1->Close();
    ADODataset1->CommandText = cmd;
    ADODataset1->Open();
    n = (long)ADODataset1->FieldByName("IDmax")->AsVariant;
    ADODataset1->Close();
    return n;
}

void TDataMod::DeleteRecord(AnsiString TableName, long IDrecord)
{
    cmd = "DELETE ";
    cmd += "FROM ";
    cmd += "    [" + TableName + "] ";
    cmd += "WHERE ";
    cmd += "    [id] = " + (AnsiString)IDrecord;
    ADONcn->BeginTrans();
    ADONcn->Execute(cmd);
    ADONcn->CommitTrans();
}
```

Рисунок 3.8 – Лістинг модулю вибірки та видалення даних

Усі методи повинні мати власний блок коду, в якому, інкапсульовано їх завдання. Розглянутий метод ініціалізує та викликає клас доступу до даних, атрибут якого є об'єктом, що має метод, завданням якого є виведення рядка з таблиці, отриманим як параметр [19].

### 3.4. Розробка візуального інтерфейсу користувача системи автоматизації управлінського обліку і контролю кадрів

Ще однією важливою характеристикою користувача, яку враховують, є його вподобання. Користувач може віддавати перевагу одним вузлам або посиланням перед іншими, а також одним частинам форми перед іншими з різних причин. Такі уподобання можуть бути абсолютними або відносними,

залежно від поточного елемента, мети або вмісту. Вподобання користувача відрізняються від інших частин моделі користувача в декількох аспектах. На відміну від інших частин моделі, вподобання не виводяться системою, а користувач повинен безпосередньо або опосередковано повідомити про них систему [20]. Це більше нагадує статичну адаптивність (adaptability), ніж динамічну (adaptivity). Різниця полягає в тому, що системи адаптивних гіпермедіа можуть узагальнити уподобання користувача та застосувати їх для досягнення адаптивності в нових контекстах [21].

Інтереси користувача не є новою характеристикою для моделювання користувача — точніше кажучи, це друга за давністю характеристика після знань користувача, але вона не використовувалася в ранніх адаптивних гіпермедійних системах [22]. З розвитком гіпермедійних систем пошуку інформації через Інтернет (web Information Retrieval hypermedia systems), які намагаються моделювати довгострокові інтереси користувачів і які використовують, паралельно з короткостроковою метою пошуку користувача з метою поліпшення фільтрації інформації та рекомендацій. Системи пошуку інформації включають системи адаптивних закладок (adaptive bookmarking systems), які в більшості випадків використовуються для реалізації інтересів користувача [14, 22].

Одна з основних форм інтерфейсу користувача зображена на рис. 3.9.

Додавання нового документа

☐ Виконаний

| Вхідний номер        | Вихідний номер       | Вихідний номер РДА   | Об'єм (стор)                   |
|----------------------|----------------------|----------------------|--------------------------------|
| <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text" value="0"/> |

Назва

Резолюція

Тип документа  Дата створення

Організація - джерело  Дата реєстрації

Організація - приймач  Дата резолюції

Автор  Дата видачі виконавцю

Куратор  Дата виконання - план

Виконавець  Дата виконання - Факт

Місце знаходження в даний момент

Рисунок 3.9 – Форма інтерфейсу користувача застосунку

Ця функція також стала популярною в різних онлайн-інформаційних системах. У цих системах інтереси користувача слугують основою для підбору відповідних елементів інтерфейсу [17]. Проте системи, що керуються програмним забезпеченням, можуть контролювати ширший спектр умов, ніж електромеханічні системи. Їх можна відносно легко адаптувати. Вони використовують апаратне забезпечення, яке має дуже високу вбудовану надійність, а також є фізично компактним і легким. Системи, що керуються програмним забезпеченням, забезпечують складну взаємопов'язану безпеку. Вони підтримують стратегії управління, що скорочують час, який люди мають проводити в небезпечних середовищах [21]. Хоча програмне управління може створити більше можливостей для збою системи, воно також забезпечує кращий моніторинг та захист і, як наслідок, сприятиме підвищенню безпеки системи. Неможливо створити систему, яка була б на сто відсотків безпечною, тому суспільство має вирішити, чи варті наслідки випадкової аварії тих переваг, які дає використання передових технологій. Лістинг програмного коду забезпечення роботи інтерфейсу користувача з базою даних приведено на рис. 3.10.

```
void TDataMod::CreateList(TComboBox *Combo,
                        TListBox *IDList,
                        AnsiString TableName,
                        AnsiString FieldName,
                        AnsiString WHEREpart)
{
    ADODataset1->Close();
    ADODataset1->CommandText = "SELECT [id], [" + FieldName + "] " +
                              "FROM [" + TableName + "] " +
                              WHEREpart + " " +
                              "ORDER BY [" + FieldName + "]";
    DataMod->ADODataset1->Open();

    if(DataMod->ADODataset1->IsEmpty())
    {
        DataMod->ADODataset1->Close();
        return;
    }

    DataMod->ADODataset1->First();
    Combo->Clear();
    IDList->Clear();
    while(!DataMod->ADODataset1->Eof)
    {
        Combo->Items->Add(DataMod->ADODataset1->FieldByName(FieldName)->AsString);
        IDList->Items->Add(DataMod->ADODataset1->FieldByName("id")->AsString);
        DataMod->ADODataset1->Next();
    }

    DataMod->ADODataset1->Close();
}
```

Рисунок 3.10 – Лістинг програмного коду забезпечення роботи інтерфейсу користувача з базою даних

Індивідуальні характеристики користувача представлені сукупністю ознак, які визначають користувача як особистість. Прикладом є фактори/коефіцієнти особистості (наприклад, інтроверт/екстраверт), когнітивні фактори [22] та стилі навчання (learning styles). Як і походження користувача, так і індивідуальні характеристики є стабільними рисами користувача, які або не зміняться взагалі, або зміняться лише через тривалий проміжок часу. На відміну від фону користувача, індивідуальні характеристики традиційно визначаються не за допомогою простого інтерв'ю, а за допомогою спеціально розроблених психологічних тестів [19]. Хоча дослідники вивчають використання індивідуальних характеристик для адаптації в багатьох сферах, наразі не можна сказати, що це є успішним процесом. Однак багато хто погоджується з важливістю моделювання та використання індивідуальних характеристик, розбіжність полягає в тому, які саме з цих характеристик слід використовувати і яким чином [15]. Адаптивність середовища користувача — це новий вид адаптації, що виник завдяки системам, заснованим на Інтернеті. Вона стала важливим фактором, оскільки користувачі одного й того ж серверного веб-додатку можуть віртуально брати участь з будь-якого місця та використовувати різне обладнання для адаптації через середовища користувача [20].

Форма застосунку для встановлення критеріїв вибірки зображена на рис. 3.11.

Умови вибірки

☒ Документ станом на дату 04.02.2026

☒ Період виконання документів плановий: с 10.02.2026 по 30.05.2026

Тип документа  
табель

Номер документа вхідний 111/46314

Номер документа вихідний

☒ Дата виконання (за планом) 10.05.2026

OK

Рисунок 3.11 – Форма застосунку для встановлення критеріїв вибірки

Ситуація з автоматизованим моделюванням користувача в адаптивних застосунках видається досить конкретною. На відміну від інших типів систем додатків, спостереження за діями користувача в застосунку надає недостатньо інформації для процесу моделювання користувача. Більшість відомих технік автоматизації моделювання користувача ледь можуть бути застосовані в цій сфері. Єдина інформація, яку система може записати щодо дій користувача, — це шлях, яким віна пройшла у до користувача, та час, який він провів на кожному вузлі, що його відвідав. Шлях і, загалом, моделі навігації є цікавим джерелом інформації, але досить складно оновлювати модель користувача, використовуючи лише ці дані. Деякі системи використовують моделі навігації, щоб визначити рівень досвіду користувача, тоді як інші визначають навчальну мету користувача на основі шляху, яким він пройшов [21].

### 3.5. Розробка модулів документування результатів роботи програмного застосунку

Застосунок має бути розроблений таким чином, щоб передбачати наступний крок користувача. Наприклад, розглянемо застосунок для підтримки клієнтів, розроблений виробником комп'ютерних принтерів. Користувач запитує об'єкт вмісту, що містить інформацію про драйвер принтера для нової операційної системи. Розробник застосунку повинен передбачити, що користувач, можливо, захоче завантажити драйвер, і повинен надати засоби навігації, що дозволяють це зробити, не вимагаючи від користувача пошуку цієї функції [19]. Комунікація. Інтерфейс повинен повідомляти про стан будь-якої дії, ініційованої користувачем. Повідомлення може бути очевидним (наприклад, текстове повідомлення) або неявним (наприклад, зображення аркуша паперу, що рухається через принтер, щоб вказати, що друк триває). Інтерфейс також повинен повідомляти про стан користувача (наприклад, його ідентифікацію) та його місцезнаходження в ієрархії вмісту додатку. Послідовність. Використання елементів навігації,

меню, піктограм та естетики (колір, форма та розміщення) має бути послідовним у застосунку [21]. Наприклад, якщо синій підкреслений текст означає навігаційне посилання, у вмісті ніколи не повинно бути тексту з такими характеристиками, який не є посиланням. Крім того, об'єкт, скажімо, жовтий трикутник, який використовується для позначення попереджувального повідомлення перед тим, як користувач запустить певну функцію або дію, не повинен використовуватися для інших цілей в жодному іншому місці застосунку [22].

Спираючись на викладені принципи, у програмі розроблено засоби модулів документування результатів роботи програмного застосунку. Один з них зображено на рис. 3.12.

| Інформація про виконання документів:      |                         |                                                                                     |                      |                  |                         |
|-------------------------------------------|-------------------------|-------------------------------------------------------------------------------------|----------------------|------------------|-------------------------|
| станом на 05.05.2026                      |                         |                                                                                     |                      |                  |                         |
| Період виконання документів плановий з .. |                         |                                                                                     |                      |                  |                         |
| № П/П                                     | Вхідний номер Дата      | Вихідний номер Дата Тип документа                                                   | Назва документу      | Термін виконання | Виконавець              |
| Куратор:                                  |                         |                                                                                     |                      |                  |                         |
| 1                                         | 873/21<br>17.04.2026    | 4563/21<br>17.04.2026<br>заявление<br>Управление Приднестровской<br>железной дороги | Заява про звільнення | 25.04.2026       | Петров Петр<br>Петрович |
| Куратор:                                  |                         |                                                                                     |                      |                  |                         |
| 2                                         | 1121/128<br>17.04.2026  | 235/77<br>17.04.2026<br>заява<br>Криворізький центр зайнятості                      | Заява про звільнення | 25.04.2026       | Петров Петр<br>Петрович |
| Куратор:                                  |                         |                                                                                     |                      |                  |                         |
| 3                                         | 24752/274<br>17.04.2026 | 9875/88<br>17.04.2026<br>заява<br>Криворізький центр зайнятості                     | Заява про звільнення | 25.04.2026       | Петров Петр<br>Петрович |

Рисунок 3.12 – Модуль документування результатів роботи програмного застосунку

Для оцінки ефективності генерації звітів проведемо порівняльний аналіз. Щоб проілюструвати це, уявіть, що велика компанія веде облік свого обладнання та пов'язану з ним базу даних, яка відстежує технічне обслуговування та ремонт обладнання [20]. Таким чином відстежується, які заявки на технічне обслуговування були подані щодо різних одиниць обладнання, яке планове технічне обслуговування заплановано, коли було виконано технічне обслуговування, скільки часу на це пішло тощо. Ця застаріла система спочатку використовувалася для формування щоденних



робочих списків для персоналу з технічного обслуговування, але з часом до неї додалися нові модулі. Вони надають дані про витрати на технічне обслуговування кожної одиниці обладнання та інформацію, яка допомагає розрахувати вартість робіт з технічного обслуговування, що мають бути виконані зовнішніми підрядниками. Система працює як система «клієнт-сервер» із спеціальним клієнтським програмним забезпеченням. Зараз компанія хоче надати доступ до цієї системи в режимі реального часу з портативних терміналів, якими користується персонал з технічного обслуговування. Вони будуть безпосередньо оновлювати систему даними про час та ресурси, витрачені на технічне обслуговування, а також звертатися до системи, щоб дізнатися про наступне завдання з технічного обслуговування. Крім того, персонал потребує доступу до системи для реєстрації запитів на технічне обслуговування та перевірки їхнього статусу. Нижче приведений лістинг програмного модулю для формування даних про документ с метою створення звіту, рис. 3.13 [20].

```
cmd += " [лицо_1].[ФИО] AS [виконавець], ";
cmd += " [документ].[дата_видачі_виконавцю], ";
cmd += " iif([документ].[стан] = 0, '#' + RusDateToUSADate(Date()).DateString()) + '# - [документ].[дата_ви] ";
cmd += " iif([документ].[стан] = 0, 'не виконаний', 'виконаний') AS [стан] ";
cmd += "FROM ";
cmd += " ((([документ] ";
cmd += "         INNER JOIN [особа] ON [документ].[ID_куратора] = [особа].[id]) ";
cmd += "         INNER JOIN [особа] AS [лицо_1] ON [документ].[ID_виконавця] = [лицо_1].[id]) ";
cmd += "         INNER JOIN [тип_документа] ON [документ].[ID_типу_документа] = [тип_документа].[id]) ";
cmd += "         INNER JOIN [організація] ON [документ].[ID_організація_джерело] = [організація].[id] ";
cmd += "WHERE " + condition + " ";
cmd += "ORDER BY [документ].[ID_куратора]";

frmRpt1->QRGroup1->Expression = "[куратор]"; // групування за цим полем
frmRpt1->DataSet1->Connection = DataMod->ADOCnn;
frmRpt1->DataSet1->Close();
frmRpt1->DataSet1->CommandText = cmd;
frmRpt1->DataSet1->Open();

// призначення полів
frmRpt1->lblHeader->Caption = ReportHeader;
if (frmElectionConditions->chDate->Checked)
    frmRpt1->lblDate->Caption = frmElectionConditions->dtpDate->Date.DateString();
else
    frmRpt1->lblDate->Caption = Date(); // DataMod->USADateToRUSDate(DataMod->condDate);

frmRpt1->txtNumVxod->DataSet = frmRpt1->DataSet1;
frmRpt1->txtNumVxod->DataField = "вихідний_номер";
frmRpt1->txtDateCreate->DataSet = frmRpt1->DataSet1;
frmRpt1->txtDateCreate->DataField = "дата_створення";
frmRpt1->txtNumVixod->DataSet = frmRpt1->DataSet1;
frmRpt1->txtNumVixod->DataField = "вихідний_номер";
frmRpt1->txtDateRegistration->DataSet = frmRpt1->DataSet1;
frmRpt1->txtDateRegistration->DataField = "дата_реєстрації";
frmRpt1->txtTipDoc->DataSet = frmRpt1->DataSet1;
frmRpt1->txtTipDoc->DataField = "тип";
frmRpt1->txtOrgSource->DataSet = frmRpt1->DataSet1;
frmRpt1->txtOrgSource->DataField = "організація_джерело";
frmRpt1->txt2->DataSet = frmRpt1->DataSet1;
frmRpt1->txt2->DataField = "назва";
frmRpt1->txt3->DataSet = frmRpt1->DataSet1;
frmRpt1->txt3->DataField = "дата_виконання_план";
frmRpt1->txt4->DataSet = frmRpt1->DataSet1;
frmRpt1->txt4->DataField = "виконавець";
```

Рисунок 3.13 – Програмний модуль отримання даних для формування звіту  
про документ робітника



Продовження програмного модуля показано на рис. 3.14.

```
frmRpt1->txtResolution->DataSet = frmRpt1->DataSet1;
frmRpt1->txtResolution->DataField = "резольція";
frmRpt1->txtDataResolution->DataSet = frmRpt1->DataSet1;
frmRpt1->txtDataResolution->DataField = "дата_резольції";
frmRpt1->txt5->DataSet = frmRpt1->DataSet1;
frmRpt1->txt5->DataField = "дата_видачі_виконавцю";
frmRpt1->txt6->DataSet = frmRpt1->DataSet1;
frmRpt1->txt6->DataField = "затримка";
frmRpt1->txt7->DataSet = frmRpt1->DataSet1;
frmRpt1->txt7->DataField = "стан";
frmRpt1->txtCurator->DataSet = frmRpt1->DataSet1;
frmRpt1->txtCurator->DataField = "куратор";

frmRpt1->rpt1->DataSet = frmRpt1->DataSet1;
ShowCriteria();
frmRpt1->rpt1->Preview();
}
void TDataMod::Report1()
{
    AnsiString cmd = "";
    AnsiString condition = " 1 = 1 ";

    if (condDate != "")
        condition += " AND [документ].[дата_видачі_виконавцю] <= #" + condDate + "# ";

    if (cond1 != "")
    {
        condition += " AND ([документ].[дата_виконання_план] >= #" + cond1 + "# ";
        condition += " AND [документ].[дата_виконання_план] <= #" + cond2 + "#) ";
    }

    if (condTipDocID != "")
        condition += " AND [документ].[ID_типу_документа] = " + condTipDocID;

    if (condNvxod != "")
        condition += " AND [документ].[вхідний_номер] = '" + condNvxod + "' ";

    if (condNvixod != "")
        condition += " AND [документ].[вихідний_номер] = '" + condNvixod + "' ";

    if (condDatePlan != "")
        condition += " AND [документ].[дата_виконання_план] <= #" + condDatePlan + "# ";

    cmd += "SELECT ";
    cmd += " [документ].[ID_типу_документа], ";
    cmd += " [тип_документа].[тип], ";
    cmd += " [організація].[назва] AS [організація_джерело], ";
    cmd += " [документ].[ID_куратора], ";
    cmd += " [документ].[ID_виконавця], ";
    cmd += " [документ].[вхідний_номер], ";
    cmd += " [документ].[вихідний_номер], ";
    cmd += " [документ].[назва], ";
    cmd += " [документ].[дата_створення], ";
    cmd += " [документ].[дата_реєстрації], ";
    cmd += " [документ].[дата_виконання_план], ";
    cmd += " [документ].[дата_виконання_факт], ";
    cmd += " [документ].[резольція], ";
    cmd += " [документ].[дата_резольції], ";
    cmd += " [особа].[ПІБ] AS [куратор], ";
    cmd += " [особа_1].[ПІБ] AS [виконавець], ";
}
```

Рисунок 3.14 – Продовження програмного модуля з рис. 3.13

Що стосується інтерфейсу, то було проведено повне проектування застосунка. Було зроблено спробу надати йому строгий вигляд. При його розробці було використано шрифт типу Arial, літери з'єднали між собою, прагнучі надати офіційний вигляд [18]. Метою було, щоб ця цілісність інтерфейсу не допускала багатозначного тлумачення при нерозривному зв'язку між системою та користувачем.

## ВИСНОВКИ

Виконана випускна кваліфікаційна робота: "Проект системи комп'ютеризації контролю кадрів при нестабільності структури трудових ресурсів». Робота присвячена актуальній темі - проєктуванню і розробці програмного застосунку для автоматизації процесів планування, управління та обліку кадрів при нестабільності структури трудових ресурсів [15].

Актуальність теми кваліфікаційної роботи полягає у необхідності вирішення проблеми організації планування, управління та обліку кадрів при нестабільності структури трудових ресурсів [18]. Під час виконання кваліфікаційної роботи було спроєктовано та реалізовано програму застосунку для обліку трудових кадрів при нестабільності структури трудових ресурсів.

Під час виконання кваліфікаційної роботи були отримані наступні результати:

- розроблені моделі і алгоритми програмного застосунку контролю кадрів при нестабільності структури трудових ресурсів; розроблені математичні моделі автоматизованої системи управлінського обліку і контролю кадрів;

- розроблена структура автоматизованої системи управлінського обліку; розроблена функціональна модель програмного застосунку;

- розроблена схема бази даних автоматизованої системи управлінського обліку і контролю кадрів;

- розроблені алгоритми програмних методів системи автоматизації управлінського обліку;

- спроєктований візуальний інтерфейс програмного комплексу системи автоматизації управлінського обліку; програмно реалізована система автоматизації управлінського обліку; розроблені програмні модулі автоматизованої системи управлінського обліку і контролю кадрів;

- розроблена база даних автоматизованої системи управлінського обліку і контролю кадрів; розроблені програмні модулі управління даними системи

управлінського обліку і контролю кадрів; розроблений візуальний інтерфейс користувача системи автоматизації управлінського обліку і контролю кадрів;  
– розроблені модулі документування результатів роботи програмного застосунку.

Розроблена програмна система для автоматизації процесів планування, управління та обліку кадрів при нестабільності структури трудових ресурсів дає змогу підвищити ефективність обліку та управління кадровими ресурсами в умовах нестабільності. Система забезпечує можливості управління персоналом, включаючи адміністративні практики, підбір персоналу, розвиток, оплату праці, трудові відносини.

На основі отриманих результатів можна зробити висновок, що виконана робота характеризує актуальність теми, демонструє ефективність розроблених програмних модулів.

Результати роботи можуть бути використані для підприємств, організації, проведення, обліку кадрів та оцінювання їх роботи.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Гавкалова Н. Л., Маркова Н. С. Цифровізація процесів управління персоналом на підприємствах : монографія. Харків : ХНЕУ ім. С. Кузнеця, 2021. 184 с.
2. Гривківська О. В., Кир'язова Т. О. Інформаційні системи і технології в управлінні персоналом : навч. посіб. Київ : Центр учбової літератури, 2022. 210 с.
3. Дороніна М. С., Сазонова Т. О. Автоматизація кадрового діловодства та обліку персоналу в сучасних організаціях : підручник. Полтава : ПУЕТ, 2023. 315 с.
4. Ситник В. Ф., Краснюк С. О. Інформаційні системи та технології менеджменту й обліку на підприємствах : навч.-метод. посібник. Київ : КНЕУ, 2020. 272 с.
5. Канцур І. Г., Кононова О. Є., Хмарська І. А. Управління персоналом в умовах цифрової економіки. *Економіка та держава*. 2022. № 2. С. 103–107.
6. Ткаченко Н. В. Цифрові інструменти моніторингу та обліку ефективності роботи персоналу в дистанційному режимі. *Вчені записки ТНУ імені В. І. Вернадського. Серія: Економіка і управління*. 2023. Т. 34 (73). № 2. С. 143–149.
7. Бєлий Д. С. Автоматизована система обліку та аудиту робочого часу працівників на підприємстві. *Інформаційні технології в освіті, науці та виробництві* : матеріали IV Міжнар. наук.-практ. конф. (Миколаїв, 18–19 травня 2022 р.). Миколаїв : ЧНУ ім. Петра Могили, 2022. С. 54–56.
8. Мельник В. П. Проблеми та перспективи автоматизації кадрового обліку на вітчизняних підприємствах. *Сучасні тенденції розвитку менеджменту та обліку* : матеріали Всеукр. наук. конф. (Львів, 12 листопада 2021 р.). Львів : ЛНУ ім. І. Франка, 2021. С. 201–203.

9. Букало Н. А. Автоматизація обліку та кадрового адміністрування в сучасних умовах трансформації бізнесу. *Економічний часопис Волинського національного університету імені Лесі Українки*. 2021. № 3 (27). С. 45–51.

10. Коломієць О. В., Петренко М. І. Трансформація систем обліку персоналу на підприємствах під впливом ERP-технологій. *Вісник Київського національного університету технологій та дизайну. Серія: Економічні науки*. 2023. № 4 (162). С. 89–96.

11. Шевченко А. М., Бойко О. В. Автоматизований облік робочого часу та розрахунків із персоналом: сучасний стан та тренди інновацій. *Бізнес Інформ*. 2024. № 1. С. 214–221.

12. Олійник Т. В. Сучасне програмне забезпечення для автоматизації кадрового обліку та розрахунку заробітної плати (HR & Payroll). *Економіка, управління та адміністрування*. 2025. № 2 (102). С. 67–73.

13. Лисенко С. О., Кравченко Ю. М. Впровадження хмарних рішень для автоматизації обліку персоналу на малих та середніх підприємствах. *Інфраструктура ринку*. 2022. Вип. 65. С. 112–117.

14. Ковальчук І. В., Сидоренко А. О. Автоматизація процесів оцінювання та обліку компетенцій персоналу організації. *Стратегічні орієнтири розвитку економіки в умовах глобалізації* : зб. тез доповідей Міжнар. наук.-практ. конф. (Одеса, 16–17 січня 2025 р.). Одеса : ОНЕУ, 2025. С. 134–136.

15. Трофименко О. Г., Прокоп Ю. В., Швайко І. Г. Програмування мовою C++ : навч. посіб. Одеса : Фенікс, 2021. 344 с.

16. Josuttis, N. M. C++20 - The Complete Guide. Independence : O'Reilly Media, 2021. 450 p.

17. Meyers, S. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14 (with updates for C++20). New York : O'Reilly Media, 2020. 336 p.

18. Костін Ю. Д., Організація обчислювальних процесів та програмування на C++ : підручник. Харків : ХНУРЕ, 2022. 280 с.

19. Крєневич А. П. С++ Орієнтоване програмування : підручник. Київ : ВПЦ "Київський університет", 2020. 255 с.
20. Васильєв О. М. Програмування на С++ для початківців : навч. посіб.
21. Stroustrup, B. A Tour of C++. 3rd ed. Boston : Addison-Wesley, 2022. 320 p.
22. Lippman, S. B., Lajoie, J., Moo, B. E. C++ Primer. 6th ed. Boston : Addison-Wesley, 2023. 920 p.
23. Gregoire, M. Professional C++. 6th ed. Indianapolis : John Wiley & Sons, 2024. 1184 p.

## ДОДАТОК А

### Лістинги програмних модулів

Фрагмент програмного коду додавання нового об'єкта персоналізованого обліку кадрів

```
#include "main.h"
#include "org.h"
#include "DataModule.h"
#include "AddOccupation.h"
#include "AddPersona.h"
#include "AddOrganization.h"
#include "AddDocType.h"
#include "AddDocument.h"
#include <fstream.h>
#include "Open.h"
#include "PrintPreview.h"
#include "Search.h"
#include "ElectionCondition.h"
#include "Rpt1.h"
#include "DB.hpp"
#include "about.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
__fastcall TfrmMain::TfrmMain(TComponent* Owner)
    : TForm(Owner)
{
}

void __fastcall TfrmMain::N25Click(TObject *Sender)
{
    frmMain->Close();
}

void __fastcall TfrmMain::N26Click(TObject *Sender)
{
    Application->CreateForm(__classid(TfrmAddOrganization), &frmAddOrganization);
}

void __fastcall TfrmMain::N4Click(TObject *Sender)
{
    if(ActiveMDIChild->Caption == "Додавання нової посадки")
    {
        frmAddOccupation->SaveOccupation();
        frmAddOccupation->Close();
        return;
    }

    if(ActiveMDIChild->Caption == "Додавання нового посадовця")
    {
        frmAddPersona->SavePersona();
        frmAddPersona->Close();
        return;
    }
}
```



## Процедури додавання нової організації та нового документа кадрового обліку

```

if(ActiveMDIChild->Caption == "Додавання нової організації")
{
    frmAddOrganization->SaveOrganization();
    frmAddOrganization->Close();
    return;
}

if(ActiveMDIChild->Caption == "Додавання нового типу документа")
{
    frmAddDocType->SaveDocType();
    frmAddDocType->Close();
    return;
}

```

### Етапи ініціалізації програми

```

void __fastcall TfrmMain::N27Click(TObject *Sender)
{
    Application->CreateForm(__classid(TfrmAddPersona), &frmAddPersona);
}

void __fastcall TfrmMain::N31Click(TObject *Sender)
{
    Application->CreateForm(__classid(TfrmAddDocType), &frmAddDocType);
}

void __fastcall TfrmMain::N28Click(TObject *Sender)
{
    Application->CreateForm(__classid(TfrmAddDocument), &frmAddDocument);
}

```

## Процедура додавання нового документа про працівника

```

if(ActiveMDIChild->Caption == "Додавання нового документа")
{
    AnsiString s;
    Application->CreateForm(__classid(TfrmPrintPreview), &frmPrintPreview);
    frmPrintPreview->Caption = "Друк відомостей о документе";
    frmPrintPreview->pre->Lines->Add(" ");
    frmPrintPreview->pre->Lines->Add("Дата звіта: " + Date().DateString());
    frmPrintPreview->pre->Lines->Add(" ");
    frmPrintPreview->pre->Lines->Add(" ВХІДНИЙ НОМЕР: " + frmAddDocument->editNumber->Text);
    frmPrintPreview->pre->Lines->Add(" ВИХІДНИЙ НОМЕР: " + frmAddDocument->editOutgoingNumber->Text);
    frmPrintPreview->pre->Lines->Add(" ОБ'ЕМ (сторінок): " + frmAddDocument->editVolume->Text);
    s = frmAddDocument->chbExecuted->Checked ? "да" : "нет";
    frmPrintPreview->pre->Lines->Add(" ВИКОНАНО: " + s);
    frmPrintPreview->pre->Lines->Add(" НАЗВА: " + frmAddDocument->memoName->Text);
    frmPrintPreview->pre->Lines->Add(" РЕЗОЛЮЦІЯ: " + frmAddDocument->memoResolution->Text);
    frmPrintPreview->pre->Lines->Add(" ТИП: " + frmAddDocument->cmbType->Text);
    frmPrintPreview->pre->Lines->Add(" ОРГАНІЗАЦІЯ-ДЖЕРЕЛО: " + frmAddDocument->cmbSource->Text);
    frmPrintPreview->pre->Lines->Add(" ОРГАНІЗАЦІЯ-ПРИЙМАЧ: " + frmAddDocument->cmbReceive->Text);
    frmPrintPreview->pre->Lines->Add(" АВТОР: " + frmAddDocument->cmbAuthor->Text);
    frmPrintPreview->pre->Lines->Add(" КУРАТОР: " + frmAddDocument->cmbCurator->Text);
    frmPrintPreview->pre->Lines->Add(" ВИКОНАВЕЦЬ: " + frmAddDocument->cmbWorker->Text);
    frmPrintPreview->pre->Lines->Add(" МІСЦЕ РОЗТАШУВАННЯ: " + frmAddDocument->editAllocate->Text);
}

```



## Лістинг методів додавання новій посади та нового посадовця

```

void __fastcall TfrmMain::N5Click(TObject *Sender)
{
    if(ActiveMDIChild->Caption == "Додавання нової посади")
    {
        Application->CreateForm(__classid(TfrmPrintPreview), &frmPrintPreview);
        frmPrintPreview->Caption = "Друк назви посади";
        frmPrintPreview->pre->Lines->Add(" ");
        frmPrintPreview->pre->Lines->Add("Дата звіта: " + Date().DateString());
        frmPrintPreview->pre->Lines->Add(" ");
        frmPrintPreview->pre->Lines->Add(" ПОСАДА: " + frmAddOccupation->editNewOccupation->Text);
        return;
    }

    if(ActiveMDIChild->Caption == "Додавання нового посадовця")
    {
        Application->CreateForm(__classid(TfrmPrintPreview), &frmPrintPreview);
        frmPrintPreview->Caption = "Друк відомостей о посадовце";
        frmPrintPreview->pre->Lines->Add(" ");
        frmPrintPreview->pre->Lines->Add("Дата звіта: " + Date().DateString());
        frmPrintPreview->pre->Lines->Add(" ");
        frmPrintPreview->pre->Lines->Add(" ПІЗВИЩЕ: " + frmAddPersona->editFamilia->Text);
        frmPrintPreview->pre->Lines->Add(" ИМ'Я: " + frmAddPersona->editName->Text);
        frmPrintPreview->pre->Lines->Add(" ПО БАТЬКОВІ: " + frmAddPersona->editOtchestvo->Text);
        frmPrintPreview->pre->Lines->Add(" ОРГАНІЗАЦІЯ: " + frmAddPersona->cmbOrganization->Text);
        frmPrintPreview->pre->Lines->Add(" ПОСАДА: " + frmAddPersona->cmbOccupation->Text);
        frmPrintPreview->pre->Lines->Add(" КАБІНЕТ: " + frmAddPersona->editCabinet->Text);
        frmPrintPreview->pre->Lines->Add(" ТЕЛЕФОН: " + frmAddPersona->editTelephone->Text);
        return;
    }
}

```

## Модуль підключення до відповідних таблиць бази даних

```

#include <vcl.h>
#pragma hdrstop
#include "DataModule.h"
#include "Rpt1.h"
#include "ElectionCondition.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TDataMod *DataMod;

__fastcall TDataMod::TDataMod(TComponent* Owner)
: TDataModule(Owner)
{
    dbName = "ctrldoc.mdb";
    AnsiString s;
    s = "Provider=Microsoft.Jet.OLEDB.4.0;";
    s += "Data Source=" + ExtractFilePath(Application->ExeName) + dbName + ";";
    s += "Persist Security Info=False;";

    ADOConn->ConnectionString = s;
    ADOConn->Open();
    cond1 = "";
    cond2 = "";
    condDate = "";
    condTipDocID = "";
    condDocID = "";
    condDatePlan = "";
    condNvxod = "";
    condNvixod = "";
    ReportHeader = "";
}

```

## Лістинг модулю вибірки та видалення даних

```

long TDataMod::LastIDRecord(AnsiString TableName)
{
    long n;
    cmd = "SELECT ";
    cmd += "    MAX([id]) AS [IDmax] ";
    cmd += "FROM ";
    cmd += "    [" + TableName + "] ";
    ADODataset1->Close();
    ADODataset1->CommandText = cmd;
    ADODataset1->Open();
    n = (long)ADODataset1->FieldByName("IDmax")->AsVariant;
    ADODataset1->Close();
    return n;
}

void TDataMod::DeleteRecord(AnsiString TableName, long IDrecord)
{
    cmd = "DELETE ";
    cmd += "FROM ";
    cmd += "    [" + TableName + "] ";
    cmd += "WHERE ";
    cmd += "    [id] = " + (AnsiString)IDrecord;
    ADONcn->BeginTrans();
    ADONcn->Execute(cmd);
    ADONcn->CommitTrans();
}

```

## Лістинг програмного коду забезпечення роботи інтерфейсу користувача з базою даних

```

void TDataMod::CreateList(TComboBox *Combo,
                        TListBox *IDList,
                        AnsiString TableName,
                        AnsiString FieldName,
                        AnsiString WHEREpart)
{
    ADODataset1->Close();
    ADODataset1->CommandText = "SELECT [id], [" + FieldName + "] " +
                              "FROM [" + TableName + "] " +
                              WHEREpart + " " +
                              "ORDER BY [" + FieldName + "]";
    DataMod->ADODataset1->Open();

    if(DataMod->ADODataset1->IsEmpty())
    {
        DataMod->ADODataset1->Close();
        return;
    }

    DataMod->ADODataset1->First();
    Combo->Clear();
    IDList->Clear();
    while(!DataMod->ADODataset1->Eof)
    {
        Combo->Items->Add(DataMod->ADODataset1->FieldByName(FieldName)->AsString);
        IDList->Items->Add(DataMod->ADODataset1->FieldByName("id")->AsString);
        DataMod->ADODataset1->Next();
    }

    DataMod->ADODataset1->Close();
}

```

## Програмний модуль отримання даних для формування звіту про документ робітника

```

void TDataMod::Report1()
{
    AnsiString cmd = "";
    AnsiString condition = " 1 = 1 ";

    if (condDate != "")
        condition += " AND [документ].[дата_видачі_виконавцю] <= #" + condDate + "# ";

    if (cond1 != "")
    {
        condition += " AND ([документ].[дата_виконання_план] >= #" + cond1 + "# ";
        condition += " AND [документ].[дата_виконання_план] <= #" + cond2 + "#) ";
    }

    if (condTipDocID != "")
        condition += " AND [документ].[ID_типу_документа] = " + condTipDocID;

    if (condNvxod != "")
        condition += " AND [документ].[вхідний_номер] = '" + condNvxod + "' ";

    if (condNvixod != "")
        condition += " AND [документ].[вихідний_номер] = '" + condNvixod + "' ";

    if (condDatePlan != "")
        condition += " AND [документ].[дата_виконання_план] <= #" + condDatePlan + "# ";

    cmd += "SELECT ";
    cmd += " [документ].[ID_типу_документа], ";
    cmd += " [тип_документа].[тип], ";
    cmd += " [організація].[назва] AS [організація_джерело], ";
    cmd += " [документ].[ID_куратора], ";
    cmd += " [документ].[ID_виконавця], ";
    cmd += " [документ].[вхідний_номер], ";
    cmd += " [документ].[вихідний_номер], ";
    cmd += " [документ].[назва], ";
    cmd += " [документ].[дата_створення], ";
    cmd += " [документ].[дата_реєстрації], ";
    cmd += " [документ].[дата_виконання_план], ";
    cmd += " [документ].[дата_виконання_факт], ";
    cmd += " [документ].[резольція], ";
    cmd += " [документ].[дата_резольції], ";
    cmd += " [особа].[ПІБ] AS [куратор], ";
    cmd += " [особа_1].[ПІБ] AS [виконавець], ";
    cmd += " [документ].[дата_видачі_виконавцю], ";
    cmd += " iif([документ].[стан] = 0, '#' + RusDateToUSADate(Date()).DateToString() + '# - [документ].[дата_ви'
    cmd += " iif([документ].[стан] = 0, 'не виконаний', 'виконаний') AS [стан] ";
    cmd += "FROM ";
    cmd += " ((([документ] ";
    cmd += " INNER JOIN [особа] ON [документ].[ID_куратора] = [особа].[id]) ";
    cmd += " INNER JOIN [особа_1] ON [документ].[ID_виконавця] = [особа_1].[id]) ";
    cmd += " INNER JOIN [тип_документа] ON [документ].[ID_типу_документа] = [тип_документа].[id]) ";
    cmd += " INNER JOIN [організація] ON [документ].[ID_організація_джерело] = [організація].[id]
    cmd += "WHERE " + condition + " ";
    cmd += "ORDER BY [документ].[ID_куратора]";

    frmRpt1->QRGroup1->Expression = "[куратор]"; // групування за цим полем
    frmRpt1->DataSet1->Connection = DataMod->ADOCnn;
    frmRpt1->DataSet1->Close();
    frmRpt1->DataSet1->CommandText = cmd;
    frmRpt1->DataSet1->Open();

    // призначення полів
    frmRpt1->lblHeader->Caption = ReportHeader;
    if (frmElectionConditions->chDate->Checked)
        frmRpt1->lblDate->Caption = frmElectionConditions->dtpDate->Date.DateToString();
    else
        frmRpt1->lblDate->Caption = Date(); // DataMod->USADateToRUSDate(DataMod->condDate);

    frmRpt1->txtNumVxod->DataSet = frmRpt1->DataSet1;
    frmRpt1->txtNumVxod->DataField = "вхідний_номер";
    frmRpt1->txtDateCreate->DataSet = frmRpt1->DataSet1;
    frmRpt1->txtDateCreate->DataField = "дата_створення";
    frmRpt1->txtNumVixod->DataSet = frmRpt1->DataSet1;
    frmRpt1->txtNumVixod->DataField = "вихідний_номер";
    frmRpt1->txtDateRegistration->DataSet = frmRpt1->DataSet1;
    frmRpt1->txtDateRegistration->DataField = "дата_реєстрації";
    frmRpt1->txtTipDoc->DataSet = frmRpt1->DataSet1;
    frmRpt1->txtTipDoc->DataField = "тип";
    frmRpt1->txtOrgSource->DataSet = frmRpt1->DataSet1;
    frmRpt1->txtOrgSource->DataField = "організація_джерело";
    frmRpt1->txt2->DataSet = frmRpt1->DataSet1;
    frmRpt1->txt2->DataField = "назва";
    frmRpt1->txt3->DataSet = frmRpt1->DataSet1;
    frmRpt1->txt3->DataField = "дата_виконання_план";
    frmRpt1->txt4->DataSet = frmRpt1->DataSet1;
    frmRpt1->txt4->DataField = "виконавець";
}

```