

РЕФЕРАТ

Пояснювальна записка: ___ с., ___ рис., ___ табл., ___ дод., ___ джерел.

Об'єкт розробки: інформаційна підсистема юридичного відділу.

Мета кваліфікаційної роботи: розробка інформаційної підсистеми електронного архіву документів юридичного відділу банку для оптимізації методів зберігання та обробки електронних документів.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проведено аналіз предметної галузі, визначено актуальність завдання та призначення розробки, розроблена постановка завдання, задані вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі виконано аналіз існуючих рішень, обрано вибір платформи для розробки, виконано проектування і розробка програми, наведено опис алгоритму і структури функціонування програми, визначені вхідні і вихідні дані, наведені характеристики складу параметрів технічних засобів, описаний виклик та завантаження програми, описана робота програми.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню програми та розраховано час на його створення.

Практичне значення роботи полягає в можливості завдяки розробленому додатку автоматизувати реєстрацію документів, виконати розподіл електронних документів між співробітниками, контролювати проходження документів та виконання доручень і документів, а також створювати звіти і журнали.

Актуальність теми кваліфікаційної роботи визначається тим, що в юридичній діяльності, як і в багатьох інших видах діяльності, величезну роль грає вдосконалення методів передачі, отримання, зберігання і переробки інформації. Важливим елементом в юридичній діяльності є збір і обробка статистичної інформації, створення і ведення баз даних, а також використання консультаційних систем.

Список ключових слів: ЮРИДИЧНИЙ ВІДДІЛ, ДОКУМЕНТООБІГ, КОРИСТУВАЧІ, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, ДОДАТОК, ІНФОРМАЦІЙНА СИСТЕМА.

ABSTRACT

Explanatory note: ___ pp., ___ fig., ___ table, ___ appendix, ___ sources.

Object of development: information subsystem of the legal department.

The purpose of the qualification work: development of the information subsystem of the electronic archive of documents of the legal department of the bank to optimize the methods of storage and processing of electronic documents.

The introduction considers the analysis and the current state of the problem, specifies the purpose of the qualification work and the field of its application, provides a justification for the relevance of the topic and clarifies the problem.

In the first section the analysis of the subject area is carried out, the urgency of the task and purpose of development is defined, the statement of the task is developed, requirements to software realization, technologies and software are set.

The second section analyzes existing solutions, selects a platform for development, performs design and development of the program, describes the algorithm and structure of the program, determines the input and output data, provides characteristics of the parameters of hardware, describes the call and download of the program, describes the program .

The economic section determines the complexity of the developed information system, calculates the cost of work to create a program and calculates the time for its creation.

The practical significance of the work lies in the ability to automate the registration of documents, to distribute electronic documents among employees, to control the passage of documents and the execution of orders and documents, as well as to create reports and journals.

The relevance of the topic of qualification work is determined by the fact that in legal activities, as in many other activities, a huge role is played by the improvement of methods of transmission, reception, storage and processing of information. An important element in legal activity is the collection and processing of statistical information, the creation and maintenance of databases, as well as the use of consulting systems.

List of key words: LEGAL DEPARTMENT, DOCUMENT FLOW, USERS, CLIENT-SERVER ARCHITECTURE, APPENDIX, INFORMATION SYSTEM.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних;

ІС – інформаційна система;

ПК – персональний комп'ютер;

ОС – операційна система;

ООП – об'єктно-орієнтоване програмування;

СКБД – система керування базою даних.

MS – Microsoft.

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	10
1.1. Загальні відомості з предметної галузі	10
1.1.1. Опис діяльності юридичного відділу.....	10
1.1.2. Характеристики діяльності юридичного відділу банку.....	11
1.1.3. Особливості електронного обігу документів в бізнесі.....	12
1.2. Призначення розробки та галузь застосування.....	13
1.3. Підстава для розробки.....	14
1.4. Постановка завдання.....	15
1.5. Вимоги до програми або програмного виробу.....	16
1.5.1. Вимоги до функціональних характеристик.....	16
1.5.2. Вимоги до інформаційної безпеки.....	17
1.5.3. Вимоги до складу та параметрів технічних засобів.....	18
1.5.4. Вимоги до інформаційної та програмної сумісності	19
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	20
2.1. Функціональне призначення системи	20
2.2. Опис застосованих математичних методів.....	21
2.3. Опис використаних технологій та мов програмування.....	21
2.4. Опис структури програми та алгоритмів її функціонування ...	25
2.4.1. Опис складових частин системи.....	25
2.4.2. Опис функціонування системи	28
2.4.3. Розробка БД системи.....	30

2.4.5. Алгоритм роботи програми.....	32
2.4.6. Опис взаємодії компонентів програми.....	35
2.5. Обґрунтування та організація вхідних та вихідних даних програми.....	38
2.6. Опис розробленої системи	38
2.6.1. Використані технічні засоби.....	38
2.6.2. Використані програмні засоби.....	39
2.6.3. Виклик та завантаження програми.....	39
2.6.4. Опис інтерфейсу користувача.....	39
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....	47
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту.....	47
3.2. Розрахунок витрат на створення програми.....	50
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
Додаток А. Код програми.....	56
Додаток Б. Відгук керівника економічного розділу.....	80
Додаток В. Перелік файлів на диску.....	81

ВСТУП

З ростом розміру компанії перед керівництвом встає питання про систематизацію і створення механізму поширення та архівації електронних документів. З огляду на сучасний технічний розвиток, найбільш оптимальним, є розробка бази даних типу «Електронний архів документів».

Актуальність теми обумовлена тим, що в юридичній діяльності, як і в багатьох інших видах діяльності, величезну роль грає вдосконалення методів передачі, отримання, зберігання і переробки інформації. Важливим елементом в юридичній діяльності є збір і обробка статистичної інформації, створення і ведення баз даних, а також використання консультаційних систем.

Для цього використовуються сучасні комп'ютерні технології, які допомагають автоматизувати звернення кінцевих користувачів до єдиної бази документів і підвищують рівень управління як окремими управліннями та відділами, так і організацією в цілому.

У даній кваліфікаційній роботі представлена розробка інформаційної підсистеми електронного архіву документів юридичного відділу банку.

Об'єктом розробки в роботі є інформаційна підсистема юридичного відділу.

Метою кваліфікаційної роботи є розробка інформаційної підсистеми електронного архіву документів юридичного відділу банку для оптимізації методів зберігання та обробки електронних документів. Дана система повинна спростити процес зберігання і архівацію електронних документів і прибрати недоліки, властиві використанню документів на паперових носіях.

Головні завдання, що ставляться для виконання роботи:

1. Аналіз основних методів побудови електронних архівів документів.
2. Аналіз компонентів електронних архівів документів та їх технічних методів реалізації.
3. Вибір мови програмування для розробки бази даних електронного архіву документів.

4. Проектування бази даних електронного архіву документів для юридичного відділу банку.

5. Розробка додатку для зручного доступу до бази даних системи.

Головним призначенням даного додатку для юридичного відділу є:

- розробка бази даних електронного архіву документів;
- автоматизація реєстрації документів;
- розподіл електронних документів між співробітниками;
- контроль проходження документів;
- контроль виконання доручень і документів;
- створення звітів і журналів.

Дана програма може бути використана в будь-якому юридичному відділі або управлінні зі схожими задачами функціонування.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

1.1.1. Опис діяльності юридичного відділу

Юридичний відділ включає в себе юридичну службу та відділу юридичної експертизи.

Основними напрямками діяльності відділу юридичної служби є:

1. Пошук, збір, придбання нормативно-правових документів, необхідних для здійснення діяльності підприємством.

2. Організація систематизованого обліку і зберігання нормативних правових актів.

3. Придбання та експлуатація електронних баз даних правової інформації.

4. Перевірка відповідності закону проектів наказів, інструкцій, положень та інших документів правового характеру, що подаються на підпис керівнику підприємства у за наступними напрямками:

- визначення правомочності керівника підприємства видавати наказ (інший акт правового характеру) з даного питання;

- визначення фахівця підприємства або керівника підрозділу, який компетентний підписувати правовий акт по конкретному правовому питанню;

- визначення ступеня необхідності узгодження з підрозділами підприємства;

- визначення правильності посилань на закони та інші нормативні правові акти;

- перевірка етапів погоджень із зацікавленими підрозділами підприємства.

5. Оформлення операцій з нерухомістю.

6. Функції в сфері трудового права.
 7. Ведення справ в судах загальної юрисдикції, арбітражних і господарських судах.
 8. Представництво в органах кримінальної юрисдикції;
 9. Інкасові розпорядження (арешт і зняття арешту з рахунків).
- Основними напрямками діяльності відділу юридичної експертизи є:
1. Ліцензування.
 2. Юридичне оформлення корпоративних відносин і оформлення типових договорів для фізичних осіб.
 3. Розробка та юридична експертиза договорів, проектів і позик в тому числі міжнародних.
 4. Інші функції в сфері цивільного права.
 5. Фінансування та забезпечення заставних позик (включаючи проблемні позики), складання додаткових угод до них.
 6. Здійснення правового аналізу проектів, що надійшли нормативних правових актів з підготовкою висновку відділу.
 7. Здійснює перевірку на предмет відповідності законодавству проектів і договорів, що укладаються банком.
 8. Організовує поповнення інформаційного фонду через підписку на періодичні видання, придбання наукової, довідкової та іншої літератури.
 9. Здійснює підготовку аналітичних матеріалів за дорученнями начальника управління.
 10. Інші напрямки юридичної діяльності в областях юридичної експертизи.

1.1.2. Характеристики діяльності юридичного відділу банку

Створення структури управління банку - це складний довготривалий процес, в якому можуть бути задіяні як окремі фахівці, так і цілі консалтингові компанії. Правильно обрана структура банку, своєчасна її трансформація

відповідно до тих змін внутрішніми і зовнішніми чинниками - важливі умови реалізації банківської стратегії, ефективної діяльності банку.

Банк – це досить велика організація, що включає в себе різні структурні підрозділи, в якому обов'язково має бути присутнім юридичне управління, яке координуватиме роботу банку з усіх аспектів юридичної підтримки.

Основні завдання юридичного управління:

- забезпечення юридичної захищеності компанії і її клієнтів;
- рішення виникаючих проблемних питань;
- експертиза документів для ліцензування;
- стягнення заборгованості з недобросовісних клієнтів;
- розробка юридичних документів всіх форм;
- юридична підтримка при взаємодії з організаціями, державними і муніципальними органами;
- оформлення операцій з нерухомістю;
- аналіз нововведень в нормативно-правових актах та їх впровадження.

1.1.3. Особливості електронного обігу документів в бізнесі

Під інформатизацією розуміється систематичне використання інформації та інформаційних технологій в бізнесі. При цьому до інформаційних технологій можна відносити всі засоби обробки, передачі і використання інформації, включаючи, в тому числі, і комп'ютерні технології.

Для ефективного менеджменту в діяльності будь-якого підприємства інформаційні потоки повинні представляти собою чітко відстежуваний і керований процес. Типовим інформаційним об'єктом, що фіксує і регламентує діяльність на підприємстві, є документ.

Успішність інформатизації організації багато в чому буде залежати від правильного вибору, професійного впровадження і забезпечення експлуатації інформаційної системи.

При цьому вибір програмного продукту для створення архіву документів

стає нагальним завданням для багатьох організацій незалежно від їх організаційно-правових форм.

Потреби в архівації електронних документів і досягнення в області нових інформаційних технологій призводять до постійного зростання пропозицій на ринку ПЗ.

В даний час різними фірмами розробляються, впроваджуються і проходять дослідну експлуатацію велику кількість автоматизованих систем управління, що характеризуються специфічними підходами та комп'ютерно-комунікаційними засобами реалізації. Споживачами є компанії всіх організаційно-правових форм, які можуть бути класифіковані на основі різних принципів: сферам діяльності, формі власності, правовим положенням та ін. Проте, незважаючи на зовнішні відмінності в діяльності організацій проглядаються спільні риси, подібні форми, методи і технології роботи.

Підготовка та прийняття управлінського рішення з питання створення електронного архіву документів є досить складним процесом, що вимагає від керівництва організації обліку багатьох технічних, економічних, організаційних і соціально-психологічних особливостей її розвитку.

Разом з цим, інформатизація вимагає певних фінансових вкладень, через що керівництву організації необхідно визначитися - наскільки цей захід фінансово обґрунтовано і в якому обсязі необхідний.

Тому для юридичного управління і його відділів був зроблений вибір на сторону розробки власного програмного забезпечення.

1.2. Призначення розробки та галузь застосування

При розробці даної ІС для юридичного відділу банку, врахований той факт, що деякі з комерційних систем вже впроваджені і успішно працюють в інших структурних підрозділах банку.

Таким чином, при визначенні необхідного рівня інформатизації було вирішено, що буде досить невеликої програми рівня одного управління, а саме

електронний архів документів.

Система призначена для автоматизації комплексу робіт з документами та об'єднує в собі всі складові цієї роботи:

- забезпечення реєстрації інформації, що надходить (заповнення необхідних атрибутів документу, рух, пошук, оперативне зберігання і використання документів);
- зберігання документів (зберігання інформації/документів, при необхідності їх пошук для використання в роботі);
- автоматичні операції з документами (копіювання, відновлення, знищення);
- організація індексування документів для подальшого їх швидкого пошуку (пошук по типу документу, датою створення, виконавцю);
- аудит і забезпечення безпеки документів (контроль доступу до документу і протоколювання всіх подій з документом, поділ прав користувачів);
- інтеграція з додатками обробки документа (можливість редагування документів в інших додатках);
- організація колективної роботи з документами.

Електронний архів документів повинен створити в юридичному управлінні єдиний документований інформаційний простір, що дає співробітникам засоби для підвищення ефективності управління та спільної роботи над документами на будь-якому робочому місці (в межах їх компетенції), в режимі реального часу.

1.3. Підстава для розробки

Підставою для розробки кваліфікаційної роботи бакалавра на тему «Розробка інформаційної підсистеми електронного документообігу юридичного відділу банку засобами мови С#» є наказ по Національному

1.4. Постановка завдання

Завданням кваліфікаційної роботи бакалавра є розробка електронного обігу документів для юридичного відділу банку

Незважаючи на широке розмаїття готових продуктів даної області на ринку, необхідно зробити універсальну, недорогу систему для зберігання і швидкого пошуку типових договорів і форм, характерних юридичному відділу банку. Система повинна бути призначена для швидкого пошуку проходження документів в організації та надавати такі можливості:

- реєстрацію документів;
- розподіл електронних документів між співробітниками;
- контроль проходження документів;
- контроль виконання доручень і документів;
- створення звітів і журналів.

Розроблена система повинна бути багато користувальницькою та забезпечувати координовану спільну роботу колективу співробітників. Цим має бути обумовлено як структура системи, так і вимоги до її налаштування: ідентифікація користувачів системи, управління правами доступу до оброблюваної інформації, забезпечення цілісності і збереження інформації.

Користувачам системи потрібно привласнити такі права:

- введення в систему нових документів. Користувач, наділений такими правами, може виконувати функції реєстратора документів, а також переглядати (в режимі звіту) список всіх файлів зареєстрованих на сервері;
- оновлення довідників. Користувач, наділений такими правами, може вносити в довідкові таблиці програми нові дані, а також коригувати старі;
- адміністрування системи. Користувач, наділений такими правами, може виконувати функції адміністратора системи, зокрема, створювати облікові записи користувачів, призначати користувачам права, змінювати налаштування

системи.

Облік документів і відстеження їх проходження в організації повинні базуватися на таких основних поняттях:

– реєстрація документа - процедура створення нового документа в системі, яка виконується користувачем, що має право на зміну документів - реєстратором;

– реєстраційна форма документа - набір ознак, що визначають ключові відомості про документ, що дозволяють знайти його в базі даних і відстежувати його проходження;

– контроль виконання документа - відстеження реального проходження документа і термінів його виконання в організації, яке виконує спеціально призначеним користувачем - контролером.

Електронний архів документів повинен створити в юридичному управлінні єдине документоване інформаційний простір, що дає співробітникам засоби для підвищення ефективності управління та спільної роботи над документами на будь-якому робочому місці (в межах їх компетенції), в режимі реального часу.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Система призначена для автоматизації комплексу робіт з документами та об'єднує в собі всі складові цієї роботи:

– забезпечення реєстрації інформації, що надходить (заповнення необхідних атрибутів документа, рух, пошук, оперативне зберігання і використання документів);

– зберігання документів (зберігання інформації/документів, при необхідності її пошук для використання в роботі);

– автоматичні операції з документами (копіювання, відновлення, знищення);

- організація індексування документів для подальшого їх швидкого пошуку (пошук по типу документа, датою створення, виконавцю);
- аудит і забезпечення безпеки документів (контроль доступу до документа і протоколювання всіх подій з документом, поділ прав користувачів);
- інтеграція з додатками обробки документа (можливість редагування документів в інших додатках);
- організація колективної роботи з документами

Проаналізувавши наявні в даний момент на ринку програмні продукти, що відповідають вимогам поставленим проектованій системі, можна поставити більш конкретні завдання, а саме:

- система повинна мати можливість розмежування прав доступу користувачів;
- програма повинна містити в собі модуль складання звітів;
- для зручності сприйняття документи повинні бути заповнені по одній формі, що забезпечується внесенням даних в наявні шаблони;
- ІС повинна містити в собі базу даних співробітників і клієнтів;
- система повинна відповідати фінансовій політиці організації;
- повинна володіти функцією швидкого пошуку необхідних документів;
- за допомогою програми директор компанії повинен мати можливість переглядати результативність роботи кожного співробітника і на цій підставі робити висновки;
- повинна бути реалізована можливість перегляду всієї історії співпраці компанії з конкретними контрагентами або особистостями.

1.5.2. Вимоги до інформаційної безпеки

Система повинна мати клієнт-серверну архітектуру і повинна експлуатуватися в локальній мережі організації. Така структура системи дозволить забезпечити наступні ключові характеристики:

- організацію довільної, легко змінюваної кількості робочих місць, що визначається обсягом робіт, характером вирішуваних в організації завдань і числом необхідних для цього працівників;
- ефективний захист цілісності інформації в режимі багатокористувацького доступу;
- забезпечення схоронності інформації за рахунок її централізованого архівування та відновлення;
- забезпечення конфіденційності інформації на основі управління правами доступу до неї.

1.5.3. Вимоги до складу та параметрів технічних засобів

Для серверних технічних засобів рекомендована конфігурація, що забезпечує цілодобову роботу програми з резервуванням даних:

- процесори - Intel ® Core 2 Duo ™ (E6600) 2.4GHz;
- шина даних - 1066MHz;
- пам'ять 2 x DIMM DDR2-800 1 Гб (Samsung Original);
- Жорсткі диски 3x 250 Гб SATA 2 Seagate ST3250620AS 16 Мб буфер, 7200 RPM;
- інтегрована графіка: NVIDIA ASUS 7300 256 Мб DDR3;

Для клієнтського апаратного забезпечення висуваються такі вимоги:

- процесор класу Intel ® Core 2 Duo ™ (E6600) 2.4GHz;
- SVGA монітор;
- клавіатура;
- маніпулятор «миша»;
- не менше 1 Гб ОЗУ;
- HDD 20,0 Гб;
- доступ до мережі Internet.

Наведені вище технічні характеристики є рекомендованими, тобто при наявності технічних засобів не нижче зазначених розроблене ПЗ буде

функціонувати відповідно до вимог, висунутими замовником щодо надійності, швидкості обробки даних і безпеки.

1.5.4. Вимоги до інформаційної та програмної сумісності

Додаток повинен бути створений за допомогою методів об'єктно-орієнтованого програмування на мові C# і скомпільований з розширенням exe. Він має працювати у середовищах операційної системи Windows.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

Система призначена для швидкого пошуку проходження документів в організації і надає наступні можливості:

- реєстрацію документів;
- розподіл електронних документів між співробітниками;
- контроль проходження документів;
- контроль виконання доручень і документів;
- створення звітів і журналів;

Розроблена система є багато користувальницькою та забезпечує координовану спільну роботу колективу співробітників. Цим обумовлена як структура системи, так і вимоги до її налаштування: ідентифікація користувачів системи, управління правами доступу до оброблюваної інформації, забезпечення цілісності і збереження інформації.

Співробітники організації, зареєстровані в системі і використовують її для обробки документів, яка визначається їх службовими обов'язками, називаються користувачами системи. Залежно від роду діяльності користувачам системи присвоюються права на виконання певних дій і доступ до певних документів.

Користувачам системи можна привласнити такі права:

- введення в систему нових документів. Користувач, наділений такими правами, може виконувати функції реєстратора документів, а також переглядати (в режимі звіту) список всіх файлів зареєстрованих на сервері;
- оновлення довідників. Користувач, наділений такими правами, може вносити в довідкові таблиці програми нові дані, а також коригувати старі;
- адміністрування системи. Користувач, наділений такими правами, може виконувати функції адміністратора системи, зокрема, створювати

облікові записи користувачів, призначати користувачам права, змінювати налаштування системи.

2.2. Опис застосованих математичних методів

Під час проектування та розробки даної ІС жодних математичних методів не використовувалося.

2.3. Опис використаних технологій та мов програмування

Для розробки інформаційної системи була обрана мова програмування C#. За допомогою цієї мови програмування можна якісно створити програмний додаток з максимальними можливостями за даною темою кваліфікаційної роботи. Базу даних створено за допомогою системи керування базами даних MySQL та використано платформу Windows Forms, яка націлена на створення графічного інтерфейсу додатку. Розробка програмного продукту здійснена у середовищі розробки Microsoft Visual Studio 2019, куди вбудований MySQL.

Мова C# була випущена в середині 2000 року групою програмістів компанії Microsoft. Головна особливість нової мови - це велика кількість засобів для розробника додатків - простий доступ до компонентів і низько рівневим бібліотекам платформи Win32, можливість використання системи обміну повідомленнями з об'єктами і компонентами на базі XML і протоколу SOAP.

Microsoft говорить, що визначення мови C# було отримано з C і C++, це відображає багато елементів мови. C# ширше, ніж Java, так як її проектувальники використовували спадкоємство від C++. Крім того в C# додані нові особливості (типу початкового тексту versioning). Щоб точніше розібратися у всьому цьому можна розібрати особливості C#, які чітко збігаються з Java та походять від стандартних C і C++. Особливості, які C# запозичила у цих мов допоможуть розібратися в її структурі. Деякі деталі були

запозичені з мови Java, як то простору імен etc. Ще одне запозичене з Java - це система автоматичного управління пам'яттю. На систему покладено функції виділення пам'яті для різних класів, і її звільнення після виходу з обігу класів. Реалізація інтерфейсів в C# ідентична реалізації в мові Java. Так як мова C# сама не містить бібліотек класів, то в ній з'явилася можливість використання бібліотек інших систем програмування, розроблених компанією Microsoft [1].

Головними особливостями мови C# є:

- інтерфейси: як і в Java, інтерфейс - абстрактне визначення колекції методів. Коли клас або структура виконує інтерфейс, він повинен виконати всі методи, визначені в цьому інтерфейсі. Одиночний клас може виконувати ряд інтерфейсів;

- управління пам'яттю: існує автоматична збірка "сміття", яка забезпечується .NET;

- структури: подібні структурам в C++ і повинні містити визначення даних і методи. Однак, на відміну від C++, структури в C# відмінні від класів і не підтримують успадкування. Однак, подібно Java, структури можуть виконувати інтерфейси;

- перевантаження операторів: деякі оператори можуть бути перевантажені, а деякі ні. Зокрема, жоден з операторів призначення не може бути перевантажений;

- визначення в namespace. При створенні програми, створюється один або більше класів в namespace. У ньому ж (поза класом) можливе оголошення інтерфейсів, enums і structs. Використовуючи ключові слова можна адресувати вміст іншого namespace;

- фундаментальні типи даних. У C# існує більш широке розмаїття типів даних ніж в C, C++ або Java. Типи - bool, byte, ubyte, short, ushort, int, uint, long, ulong, float, double, and decimal. Подібно Java, всі типи мають встановлений розмір. Подібно C і C++ всі типи можуть бути знаковими і без знаковими. Подібно Java, char містить 16-ти бітний unicode символ. У C# новим типом даних є тип decimal, який може містити до 28 десяткових цифр;

– два фундаментальні класи. Клас `object` - базовий клас всіх класів. Клас `string` - також базовий клас. Будучи частиною мови він використовується компілятором, коли користувач створює рядок у програмі, укладаючи її в лапки;

– асемблювання - колекція компільованих класів і здатність до виконання інших елементів мови, які об'єднані в окремому файлі. Якщо це програма, файл має розширення `EXE` [2].

Розробка програмного додатку буде виконується в середовищі розробки `Microsoft Visual Studio`. На сьогоднішній день є одним з кращих засобів розробки додатків. З кожною новою версією це середовище набуває все більше корисних функцій.

Функціональність охоплює всі етапи розробки програмного забезпечення, надаючи сучасні інструменти для написання коду, проектування графічних інтерфейсів, збірки, налагодження і тестування програм. Можливості `Microsoft Visual Studio` можуть бути доповнені шляхом підключення необхідних розширень.

Редактор коду підтримує підсвічування синтаксису, вставку фрагментів коду, відображення структури і пов'язаних функцій. Інтегроване середовище розробки `Microsoft Visual Studio` - це оригінальне середовище яке дозволяє редагувати, налагоджувати і створювати код, а потім публікувати додатки. `Microsoft Visual Studio` включає в себе редактор вихідного коду з підтримкою технології `IntelliSense` і можливістю найпростішого рефакторинга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і відладчик машинного рівня.

Решта вбудованих інструментів включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних.

`Microsoft Visual Studio` надає комплекс інструментів для автоматизації тестування додатків в частині перевірки роботи інтерфейсів, модульного і навантажувального тестування.

Windows Forms Designer. Windows.Forms використовується в Microsoft .NET для створення додатків, забезпечених графічним інтерфейсом. Ґрунтується він на .NET Framework class library і має набагато досконалішу і зручнішу в роботі модель програмування, чим, наприклад, програмні інтерфейси Win32 API або MFC.

Windows Forms - інтерфейс програмування додатків (API), що відповідає за графічний інтерфейс користувача і є частиною Microsoft .NET Framework. Цей інтерфейс спрощує доступ до елементів інтерфейсу Microsoft Windows за рахунок створення обгортки для існуючого Win32 API в керованому коді. Причому керований код - класи, реалізуючі API для Windows Forms, не залежать від мови розробки. Тобто програміст однаково може використати Windows Forms як при написанні програмного забезпечення на C#, C++ та ін. Також Windows.Forms - це набір різних керованих бібліотек, за допомогою яких можна виконати усі необхідні для віконного застосування дії, починаючи від обміну сполученнями з операційною системою для відстеження будь-яких подій клієнтського вікна, закінчуючи діалоговими системами, зв'язком з іншими комп'ютерами по мережі і багатьма іншими можливостями.

Усередині .NET Framework, Windows Forms реалізується у рамках простору імен System.Windows.Forms. Додаток Windows Forms є об'єктно-орієнтованим застосуванням, підтримуваний Microsoft .NET Framework.

На відміну від пакетних програм, велика частина часу витрачається на очікування від користувача яких-небудь дій, як, наприклад, введення тексту в текстове поле або кліку мишкою по кнопці. Оскільки Windows.Forms, по суті, повинна включати сотні організованих класів, щоб забезпечувати усі необхідні можливості розробникові .NET Framework розбита на ряд ієрархічних розділів, що мають свої імена.

Вся інформація додатку буде зберігатись в мережевій базі даних, яка знаходиться на безкоштовному сервері MySQL.

MySQL є реляційною базою даних, яка широко використовується. Дані та зв'язки між даними організовані за допомогою таблиць. Кожен стовпець у

таблиці має ім'я і тип. Кожен рядок представляє окремий запис або елемент даних в таблиці, який містить значення для кожного з стовпців.

MySQL заснований на моделі клієнт-сервер. Ядром MySQL є сервер MySQL, який обробляє всі інструкції бази даних (або команди). Сервер MySQL доступний у вигляді окремої програми для використання в мережевому середовищі клієнт-сервер і у вигляді бібліотеки, яка може бути вбудована (або пов'язана) в окремі додатки. MySQL працює разом з декількома службовими програмами, які підтримують адміністрування баз даних MySQL. Команди відправляються MySQL через клієнт MySQL, який встановлений на комп'ютері. Він також дозволяє зберігати дані і отримувати до них доступ через кілька механізмів зберігання, включаючи InnoDB, CSV і NDB.

В додатку обрано механізм збереження InnoDB. Це одна з обраних підсистем низького рівня в СУБД MySQL, входить в усі стандартні збірки для різних операційних систем. Основною відмінністю InnoDB від інших підсистем низького рівня MySQL є наявність механізму транзакцій і зовнішніх ключів.

MySQL також здатна реплікувати дані і таблиці секціонування для підвищення продуктивності і довговічності. З метою безпеки MySQL використовує систему прав доступу і зашифрованих паролів, яка забезпечує перевірку на основі хоста. Клієнти MySQL можуть підключатися до MySQL, використовуючи кілька протоколів, включаючи сокети TCP/IP на будь-якій платформі.

2.4. Опис структури системи та алгоритмів її функціонування

2.4.1. Опис складових частин системи

Створення структури управління банку - це складний довготривалий процес, в якому можуть бути задіяні як окремі фахівці, так і цілі консалтингові компанії. Правильно обрана структура банку, своєчасна її трансформація відповідно до тих змін внутрішніми і зовнішніми чинниками - важливі умови реалізації банківської стратегії, ефективної діяльності банку.

Залежно від кількості і обсягу виконуваних робіт, також може змінюватися і загальне число співробітників.

На рис. 2.1 представлена організаційна структура юридичного відділу банку.

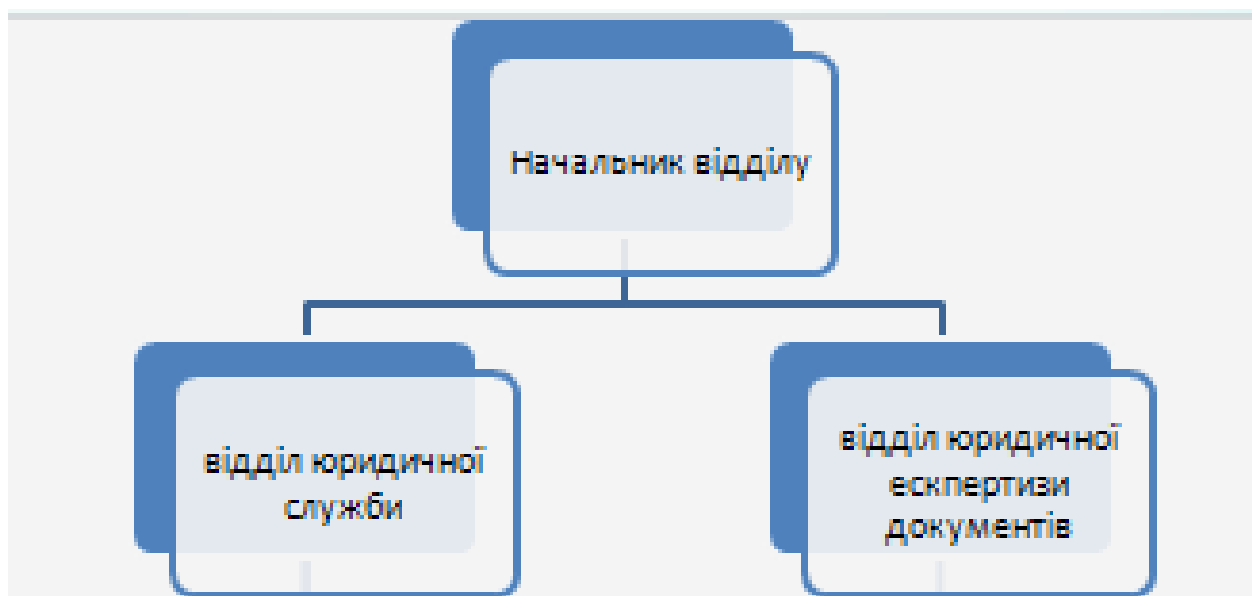


Рис. 2.1. Організаційна схема юридичного відділу банку

На чолі юридичного управління знаходиться начальник управління.

Безпосередньо начальнику юридичного управління підпорядковуються два відділи:

1. Відділ юридичної служби.
2. Відділ юридичної експертизи.

Функції начальника управління:

- самостійно і оперативно приймає і реалізовує управлінські рішення;
- контролювати, аналізувати і прогнозувати наслідки прийнятих рішень;
- планування роботи відділів;
- навчання працівників відділів;
- ставити перед підлеглими завдання;

– забезпечувати взаємодію відділів з іншими структурними підрозділами;

– проводити наради, семінари з правових питань;

– інші функції, виконуються начальником.

Юридичний відділ включає в себе юридичну службу та відділу юридичної експертизи.

Основними напрямками діяльності відділу юридичної служби є:

1. Пошук, збір, придбання нормативно-правових документів, необхідних для здійснення діяльності підприємством.

2. Організація систематизованого обліку і зберігання нормативних правових актів.

3. Придбання та експлуатація електронних баз даних правової інформації.

4. Перевірка відповідності закону проектів наказів, інструкцій, положень та інших документів правового характеру, що подаються на підпис керівнику підприємства.

5. Оформлення операцій з нерухомістю.

6. Функції в сфері трудового права.

7. Ведення справ в судах загальної юрисдикції, арбітражних і господарських судах.

8. Представництво в органах кримінальної юрисдикції;

9. Інкасові розпорядження (арешт і зняття арешту з рахунків).

Основними напрямками діяльності відділу юридичної експертизи є:

1. Ліцензування.

2. Функції в сфері цивільного права.

3. Напрямки юридичної діяльності в областях юридичної експертизи.

2.4.2. Опис функціонування системи

Система призначена для швидкого пошуку проходження документів в організації і надає наступні можливості:

- реєстрацію документів;
- розподіл електронних документів між співробітниками;
- контроль проходження документів;
- контроль виконання доручень і документів;
- створення звітів і журналів.

Розроблена система є багато користувальницькою та забезпечує координовану спільну роботу колективу співробітників. Цим обумовлена як структура системи, так і вимоги до її налаштування: ідентифікація користувачів системи, управління правами доступу до оброблюваної інформації, забезпечення цілісності і збереження інформації.

Співробітники організації, зареєстровані в системі і використовують її для обробки документів, яка визначається їх службовими обов'язками, називаються користувачами системи. Залежно від роду діяльності користувачеві системи адміністратором, присвоюються права на виконання певних дій і доступ до певних документів.

Користувачам системи можна привласнити такі права:

- введення в систему нових документів. Користувач, наділений такими правами, може виконувати функції реєстратора документів, а також переглядати (в режимі звіту) список всіх файлів зареєстрованих на сервері;
- оновлення довідників. Користувач, наділений такими правами, може вносити в довідкові таблиці програми нові дані, а також коригувати старі.
- адміністрування системи. Користувач, наділений такими правами, може виконувати функції адміністратора системи, зокрема, створювати облікові записи користувачів, призначати користувачам права, змінювати налаштування системи.

До документів при реєстрації додається кілька смислових частин, які в

свою чергу, займають відповідні місця в різних таблицях бази даних. Концептуальна модель побудови електронного архіву документів відображена на рис. 2.2

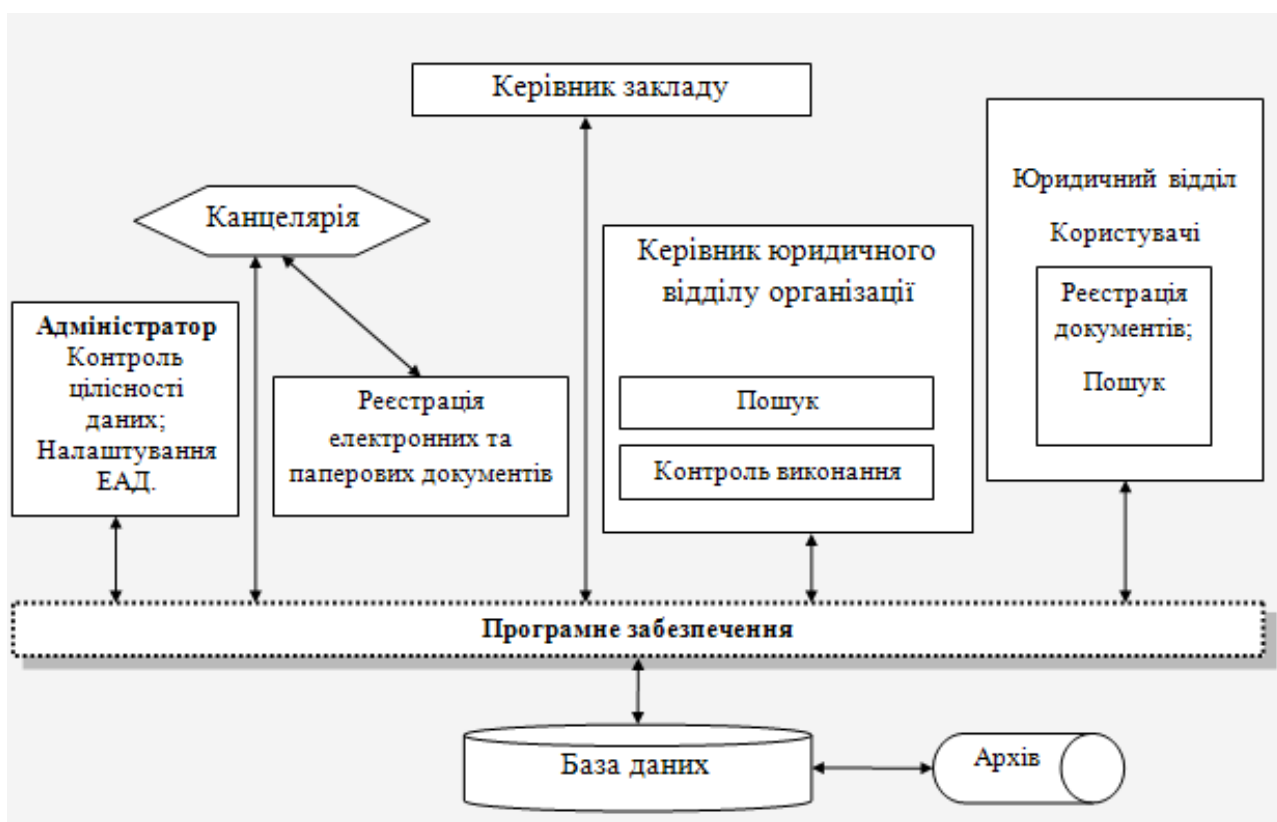


Рис. 2.2. Концептуальна модель побудови електронного архіву документів

При створенні нових документів користувачі, включені в один зі списків, за замовчуванням отримують відповідні права - перегляду або редагування документа. Користувачам, не включених до списків, право доступу може бути призначено лише адміністратором системи, який має доступ до налаштувань програми. Всі інші користувачі не мають доступу до документів.

Так як система являє собою клієнт-серверний варіант, то проектування програми велося паралельно в двох формах: клієнтської і адміністративної (серверної).

Система, з якою працює багато користувачів, повинна мати централізоване управління. Цю функцію виконує спеціально виділений співробітник або співробітники, які здійснюють єдину узгоджену політику настройки та управління системою, яких називають адміністраторами системи.

В обов'язки адміністратора системи входить, зокрема, наступне:

- відображення в системі поточної структури організації;
- реєстрація користувачів;
- перегляд системних повідомлень і повідомлень про помилки;
- актуалізація таблиць бази даних.

Крім зазначених функцій, адміністратор системи також забезпечує запуск і зупинку сервера електронного архіву документів, налаштування та обслуговування системи, створення резервних копій баз даних системи і відновлення працездатності системи у разі збоїв.

Програма-сервер електронного архіву документів буде виконувати кілька функцій:

- обробку запитів клієнтів;
- висновок службової інформації (логів);
- служити посередником між клієнтом і базою даних;
- здійснювати процедуру реєстрації користувачів.

Користувачі в клієнтській частині системи мають такі права:

- виконувати функції реєстратора документів, а також перегляду (в режимі звіту) списку всіх файлів, зареєстрованих на сервері;
- оновлювати довідники, вносити в довідкові таблиці програми нові дані, а також коригувати старі.

На рис. 2.4 представлена структура основних модулів системи і зв'язку між ними.

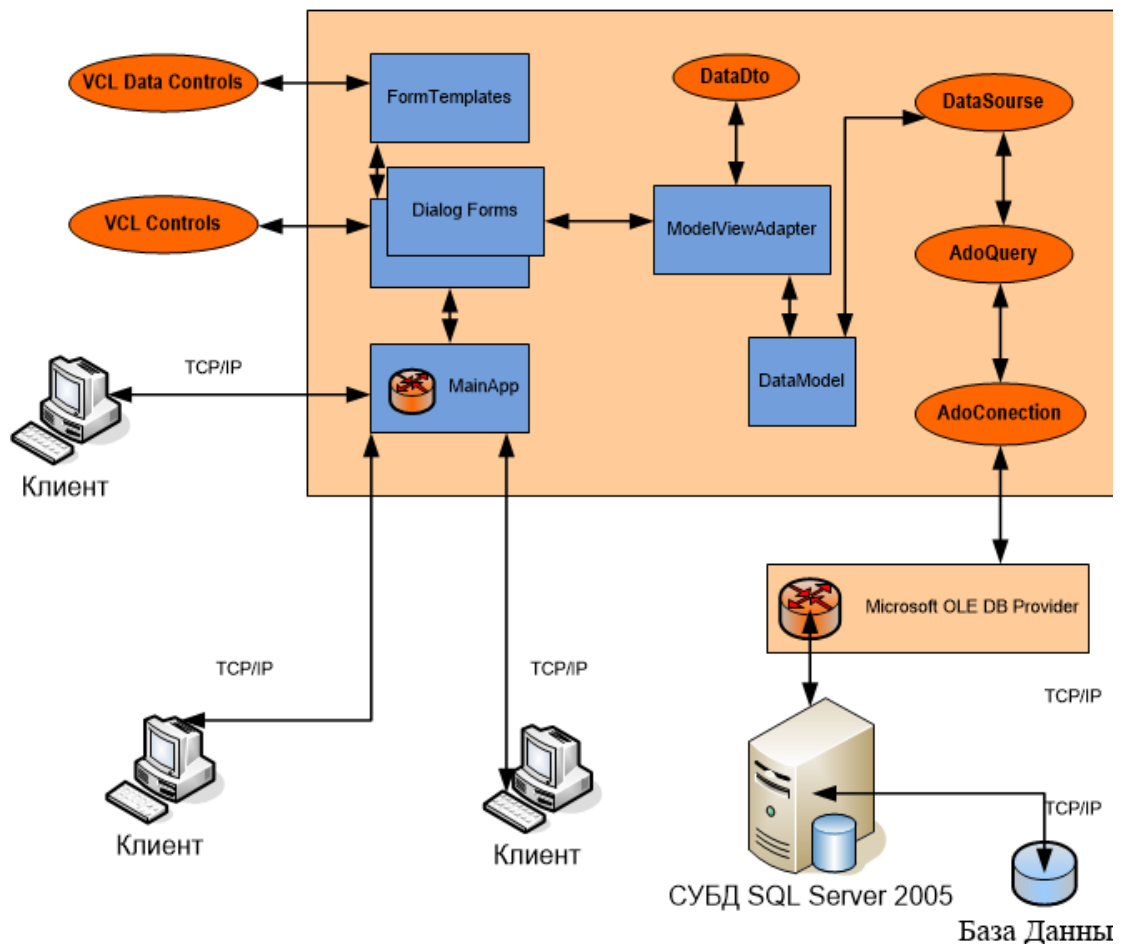


Рис.2.3. Основні модулі системи

2.4.3. Розробка БД системи

Програма буде складатися з двох основних частин: клієнта і сервера. Сервер з'єднується з базою даних, а клієнти, в свою чергу, підключаються до даного сервера для обміну інформацією, виконанням централізованих розрахунків, отриманням і записом даних. За систему керування базою даних обрана система MySQL.

Самим документам при реєстрації присвоюється кілька реквізитів, що дозволяють швидко знайти їх за допомогою пошуку в базі даних. Також при реєстрації відділяється сам текст від документа, і це дає можливість виробляти пошук по вмісту документа.

База даних системи повинна складатися з ряду необхідних таблиць, таких як таблиця користувачів системи (A_Users), таблиць текстових повідомлень

(A_VhodDocs, A_IshodDocs та іншими), таблиці структури документа (ZH_DOCS_TYPE) та інших таблиць, не кажучи вже про більш детальному поділі зазначеної структури. Структура таблиць бази даних представлена на рис. 2.5.

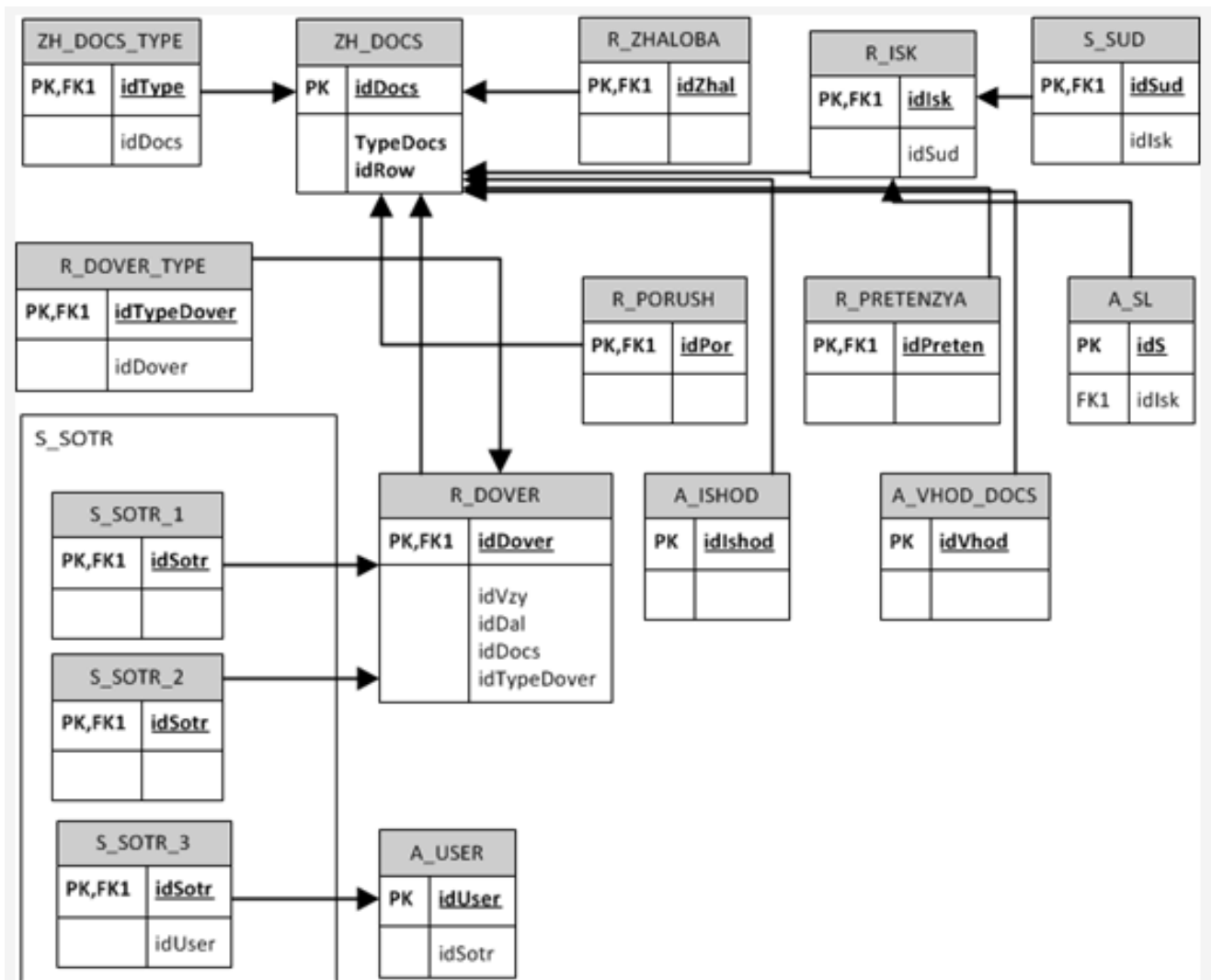


Рис. 2.5. Структура бази даних

З'єднання клієнтів з сервером буде здійснюватися по протоколу TCP/IP, щоб уникнути втрат важливих даних. До того ж нинішні мережеві технології дозволяють використовувати високошвидкісні канали зв'язку за відносно невеликі фінансові вкладення.

При такій схемі база даних і серверна частина програми електронного архіву документів можуть перебувати фізично на різних серверах. Це дозволить

знизити вимоги до обладнання і знизити навантаження на окремі вузли мережевої інфраструктури.

2.4.4. Алгоритм роботи програми

Функціонування системи документообігу юридичного відділу банку повинно бути засноване на використанні спеціальної програми - сервера електронного архіву документів. Сервер електронного архіву документів буде здійснювати основні функції, що забезпечують роботу користувачів: отримання запитів, звернення до бази даних, обробку інформації, видачу результатів.

Так як на сервері відбувається реєстрація користувачів, то клієнт повинен, в першу чергу, пройти процедуру авторизації. Після відправки введених даних клієнт отримує або позитивну відповідь, в разі правильного введення імені користувача і пароля, або повідомлення про відмову доступу. З програмою зможуть працювати тільки користувачі, які успішно пройшли процедуру авторизації.

До тих пір, поки основна програма не запущена, взаємодія користувачів з системою електронного архіву документів і робота з документами неможлива.

На рис. 2.6 і 2.7 показані спроектовані вікна серверної частини програми «Юридичне управління» при її першому запуску, а зокрема майбутнє тестування з'єднання з базою даних, яка розташована на тому ж сервері при різних режимах аутентифікації.

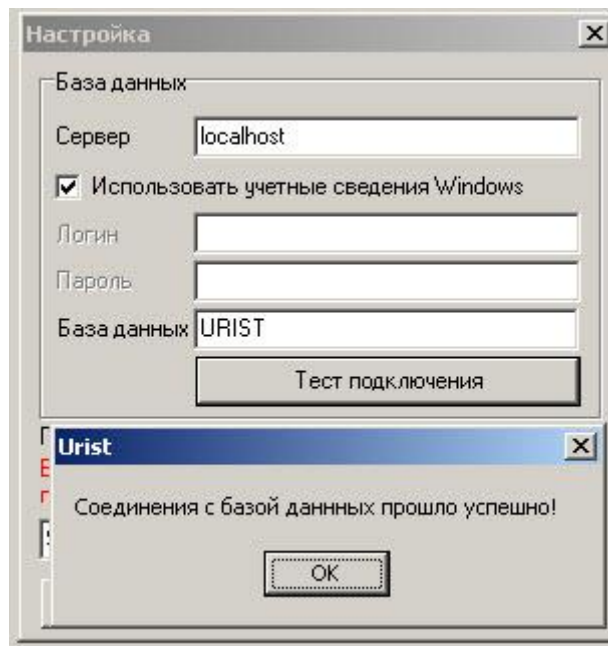


Рис. 2.6. Тестування з'єднання програми з базою даних без введення пароля адміністратора

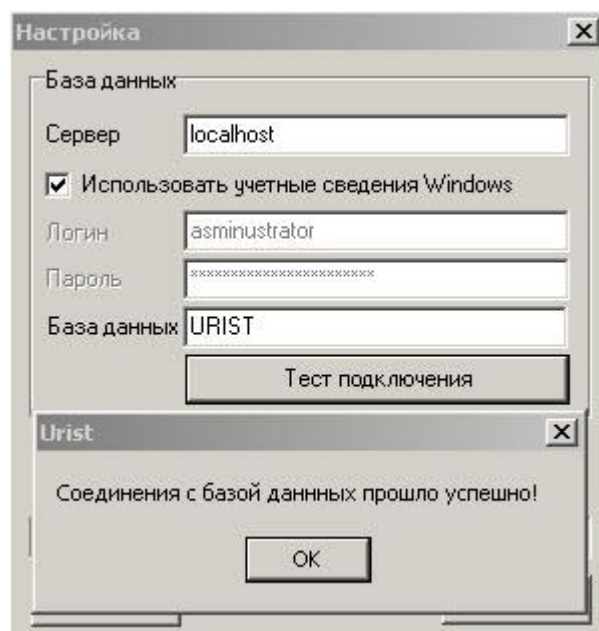


Рис. 2.7. Тестування з'єднання програми з базою даних в режимі введення пароля адміністратора.

З'єднання серверної частини програми з базою даних в різних режимах аутентифікації пояснюється тим, що при вже наявному доступі на головний

сервер, немає необхідності встановлювати додаткові паролі для входу в базу даних. У всіх організаціях (фактично), за це відповідає група компетентних людей, і крім них доступ до сервера не має ніхто інший. У тих випадках, коли в ролі сервера буде використовуватися звичайна робоча станція, на якій пересічний користувач буде виконувати свою повсякденну роботу, для розмежування прав доступу необхідно встановлювати пароль для зв'язку з базою даних.

У разі недоступності сервера або некоректно введення логіна і\або пароля, система повинна видавати повідомлення про помилку і продовження роботи з програмної надалі буде неможливо. Продовжити роботу можна буде тільки після коректної авторизації, або відновлення роботи основного сервера електронного архіву документів.

Програма-сервер «Юридичне управління» буде виконуватися на комп'ютері, підключеному до мережі організації, який також називають сервером. Для сервера електронного архіву документів, можливо, виділити окремий комп'ютер або використовувати одну з робочих станцій мережі, вибір залежить від обсягу робіт з документами. Для приблизної оцінки можна сказати, що при малих (до сотні в день) потоках документів цілком допустимо запускати сервер електронного архіву документів на робочій станції середнього класу, при цьому вона одночасно може використовуватися для звичайної офісної роботи без серйозного зниження швидкості. Необхідно відзначити, що реєстрація документів і створення пов'язаних з ними завдань і повідомлень призводять до зростання розміру бази даних і збільшення займаного нею дискового простору, тому слід забезпечити достатню ємність жорстких дисків з урахуванням перспектив зростання потоку документів.

Після успішного підключення до сервера користувачеві надається головне вікно для роботи з програмою. За своєю суттю це повинно бути SDI (однодокументне) вікно з головним меню і панеллю інструментів.

2.4.5. Опис взаємодії компонентів програми

Для дотримання принципів об'єктно-орієнтованого програмування потрібно розділити код за смисловим ознаками в окремі класи. Виходячи з функцій, які виконуються програмою-сервером, можна виділити три основні класи:

- клас інтерфейсу для взаємодії з користувачем UMenu;
- клас мережевої взаємодії з клієнтами - UUchet;
- клас зв'язку з базою даних - UConfig.

Всі ці класи взаємопов'язані і служать для обробки команд клієнтів. Спочатку ініціалізується сама програма, потім включаються зазначені основні класи. Мережевий клас отримує команди від клієнта, для їх виконання використовує допоміжний клас взаємодії з базою даних, при необхідності відправляє відповідь клієнту і відображає результати своєї роботи на призначений для користувача інтерфейс. Також можуть використовуватися додаткові структурні ланки. За допомогою такого поділу на класи реалізується інкапсуляція - це механізм програмування, який об'єднує дані і код в одному блоці, що оберігає їх від втручання ззовні і неправильного використання.

«Інкапсуляція дозволяє об'єднати дані і код в об'єкт і приховати реалізацію об'єкта від користувача. При цьому користувачеві надається тільки специфікація (інтерфейс) об'єкта. Користувач може взаємодіяти з об'єктом тільки через цей інтерфейс ».

Щоб визначити стан сервера і правильність виконання запитів клієнтів, необхідно використовувати компонент для запису службової інформації в реальному часі. Цей поповнюваний список, так званий журнал подій або лог, повинен розташовуватися в основному вікні програми-сервера, щоб найбільш зручним чином відобразити інформацію для адміністратора системи. Сюди повинні потрапляти, головним чином, результати обробки запитів до бази даних, так як це є дуже вразливим місцем в системі, особливо якщо програма-сервер і сховище даних знаходяться фізично на різних комп'ютерах.

Для скорочення числа додаткових програм в інтерфейс сервера необхідно вбудувати засіб реєстрації користувачів системи. В якості основних ідентифікаційних даних має бути внесено ім'я користувача (логін або псевдонім), справжнє ім'я, прізвище та по батькові, а також пароль. Також необхідно врахувати і додаткові ідентифікатори користувача в організації: номер телефону, адреса знаходження і e-mail. Щоб підвищити рівень безпеки системи, пароль потрібно зберігати в базі даних в зашифрованому вигляді.

Останній, але важливий елемент в сервері - це параметри настройки з'єднання з базою даних і її стартове розгортання. Щоб сервер міг використовувати як джерело даних довільний доступний вузол мережі, необхідно дати можливість адміністратору задавати параметри підключення до сервера бази даних. У параметри підключення входить IP-адреса комп'ютера, що містить базу даних, ім'я самої бази, логін і пароль користувача, що має доступ до вчинення процедур читання, записи і поновлення таблиць бази даних. Також до сервера повинна бути приєднана утиліта для початкового створення таблиць в базі і їх заповнення необхідними значеннями. Це дозволить без допомоги сторонніх програм (можливо навіть платних) і знання мови SQL підготувати дану систему електронного архіву документів до роботи. Параметри підключення також відображені в файлі конфігурації програми. Далі представлена ще одна частина конфігураційного файлу, що відповідає за з'єднання з базою даних.

Запити до серверної програми повинні надходити по протоколу TCP / IP, так як дані не повинні губитися по шляху проходження. Програмним засобом передачі даних було вирішено використовувати набір низькорівневих класів Net Sockets, що дозволяють працювати з керованими сполуками.

Опис частини, конфігураційного файлу програми, що відповідає за обробку класів Net Sockets описано нижче:

```
Socket s = new Socket (AddressFamily.InterNetwork,  
SocketType.Stream, ProtocolType.Tcp);  
<Configuration>
```

```

<System.diagnostics>
  <Sources>
    <Source name = "System.Net.Sockets">
      <Listeners>
        <Add name = "Sockets" />
      </Listeners>
    </Source>
  </Sources>
  <Switches>
    <Add name = "System.Net.Sockets" value = "31" />
  </Switches>
  <SharedListeners>
    <Add name = "Sockets" type =
      "System.Diagnostics.TextWriterTraceListener"
      initializeData = "Sockets.log" />
  </SharedListeners>
  <Trace autoflush = "true" />
</System.diagnostics>
</Configuration>

```

Так як клієнтів в даній системі може бути кілька (програмно не повинно бути обмежень в кількості клієнтів; їх число обмежується пропускною спроможністю мережі і продуктивністю обладнання), то програма-сервер повинна працювати з ними окремо. Таким чином, планується виділяти клієнтів в самостійні потоки, які будуть народжуватися при отриманні сигналу про новому підключенні і закриватися при від'єднанні користувача.

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Дана інформаційна система забезпечує роботу з наступними документами:

- вхідні документи - будь-які типи юридичних документів, які потрапляють в канцелярію і реєструються в журналі обліку документів;
- вихідні документи - будь-які типи юридичних документів, які виходять з канцелярії і реєструються в журналі обліку документів;
- внутрішні документи - юридичні документи, які звертаються усередині юридичного управління і не реєструються в журналі обліку документів.

Вхідними даними є та інформація, що вноситься користувачем в дані документи, а також дані про користувачів, співробітників тощо, а вихідною є дані, що формуються в результаті обробки вхідної інформації та формуються за відповідними запитами та зберігаються у відповідних файлах.

2.6. Опис роботи розробленої системи

2.6.1. Використані технічні засоби

Для розробки та тестування даної системи використовувався ПК з наступними характеристиками:

- процесор із тактовою частотою 1 ГГц 32-розрядний (x86);
- оперативна пам'ять 1 гігабайт (ГБ) (для 32-розрядної версії);
- графічний пристрій із підтримкою DirectX 9 і драйвером WDDM 1.0.

2.6.2. Використані програмні засоби

Для розробки інформаційної системи була обрана мова програмування C#. За допомогою цієї мови програмування можна якісно створити програмний додаток з максимальними можливостями за даною темою кваліфікаційної роботи. Базу даних створено за допомогою системи керування базами даних MySQL та використано платформу Windows Forms, яка націлена на створення графічного інтерфейсу додатку. Розробка програмного продукту здійснена у середовищі розробки Microsoft Visual Studio 2019.

2.6.3. Виклик та завантаження програми

Для установки ІС потрібно папку Justice скопіювати на потрібний комп'ютер, відкрити файл Justice.exe і виконати всі кроки Майстра установки.

2.6.4. Опис інтерфейсу користувача

Після запуску програми на екрані з'являється вікно авторизації. Після успішної авторизації користувачу доступне головне вікно програми, яке представлено на рис. 2.8.



Рис. 2.8. Головне вікно програми.

Головне вікно програми складається з наступних пунктів:

- реєстрація;
- довідники;

- сервіс;
- вихід з програми.

У пункті меню «Реєстрація» (рис. 2.9) для користувача у відповідних закладках надається весь основний функціонал програми-клієнта, а саме:

- «Канцелярія» - призначена для введення і реєстрації вхідних поштових та інших повідомлень;
- «Доручення» - призначена для введення і реєстрації нотаріальних та інших типів довіреностей, які видаються або отримані юридичним управлінням та банком в цілому;
- «Позовні заяви» - призначена для введення і реєстрації позовних заяв, що пред'являються або отримані юридичним управлінням та банком в цілому;
- «Доручення» - призначена для введення і реєстрації всіх видів доручень, що видаються або отримані юридичним управлінням та банком в цілому;
- «Претензії» - призначена для введення і реєстрації претензій, що пред'являються або отримані юридичним управлінням та банком в цілому;
- «Скарги» - призначена для введення і реєстрації скарг, що пред'являються або отримані юридичним управлінням та банком в цілому;
- «Слухання» - призначена для введення інформацій про поточні, колишні та майбутні слухання в суді.

Всю інформацію в даних пунктах користувачам клієнтської програми можна переглядати та редагувати.

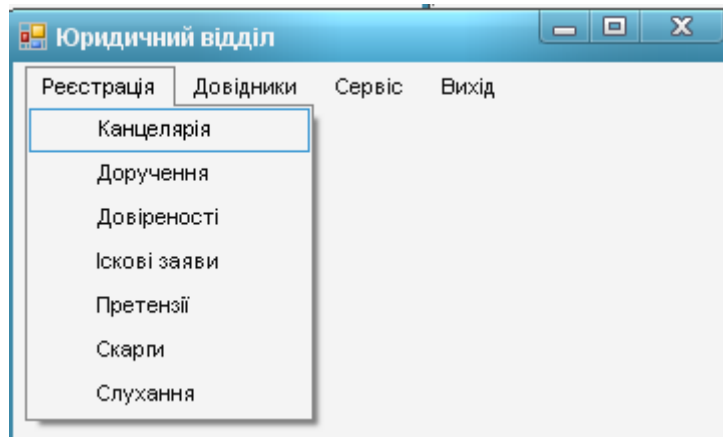


Рис. 2.9. Пункт меню «Реєстрація»

Підпункт «Канцелярія» включає в себе перелік вхідних, вихідних і внутрішніх документів.

Вхідні документи - будь-які типи юридичних документів, які потрапляють в канцелярію і реєструються в журналі обліку документів. Перелік вхідних документів представлений на рис. 2.11.

	Номер вхідного	Дата вхідного	Номер вихідного	Дата вихідного	Зміст	Відправник	Отримувач
▶ ВХІДНІ	1	23.10.2020	1	23.11.2020	Відповідь на запит	Іванов І.І.	фірма1
ВИХІДНІ	6	12.05.2021	11	13.05.2021	Відповідь на скаргу	Петров П.П.	фірма2
ВНУТРІШНІ	12	23.05.2021					

Рис. 2.11. Перелік вхідних документів.

Вихідні документи - будь-які типи юридичних документів, які виходять з канцелярії і реєструються в журналі обліку документів (рис. 2.12).

	Номер вихідного	Дата вихідного	Отримувач	Відправник	Виконавець	Зміст	Місцезнаходженн	Дата контролю	Дата виконан
▶ ВХІДНІ	1	23.11.2020	фірма1	Іванов І.І.	Хамсін А.Л.	Відповідь на запит	пр. Яворницько...	23.11.2020	23.11.2020
ВИХІДНІ	11	13.05.2021	фірма2	Петров П.П.	Моршинін А.Р.	Вимога щодо надання	вул. Гончара, 17	15.05.2021	13.05.2021
ВНУТРІШНІ	23	15.05.2021	фірма3	Сидоров С.С.	Порогіков М.Д.	Заявка на стягнення	пр. Поля, 113	15.05.2021	14.05.2021
	24	16.05.2021	фірма3	Сидоров С.С.	Вольський П.Л.	Заявка на привід	пл. Соборна, 34	16.05.2021	16.05.2021

Рис. 2.12. Перелік вихідних документів

Внутрішні документи - юридичні документи, які обертаються усередині юридичного управління і не реєструються в журналі обліку документів (рис. 2.13).

	Номер виділого	Дата виділого	Отримувач	Номер виділого	Дата виділого	Відправник	Виконавець	Дата контролю	Дата виконан
ВХІДНІ	1	23.11.2020	Хамсін А.Л.	1	23.11.2020	Іванов І.І.	Хамсін А.Л.	23.11.2020	23.11.2020
ВИХІДНІ	11	13.05.2021	Моршинін А.Р.	11	13.05.2021	Петров П.П.	Моршинін А.Р.	15.05.2021	13.05.2021
ВНУТРІШНІ	3	15.05.2021	Поротіков М.Д.	23	15.05.2021	Сидоров С.С.	Поротіков М.Д.	15.05.2021	14.05.2021
	4	16.05.2021	Вольський П.Л.	24	16.05.2021	Сидоров С.С.	Вольський П.Л.	16.05.2021	16.05.2021

Рис. 2.13. Перелік внутрішніх документів.

Підпункт «Доручення» - включає в себе доручення клієнтів і доручення які, складають співробітники юридичного управління (рис. 2.14):

Номер	Дата отримання	Зміст	Джерело надходження	Виконавець	Дата виконання	стан	контроль
1	12.04.2021	отримання матеріальн...	начальник відділу	Хоменко А.Р.	15.04.2021	виконано	так
2	11.05.2021	відвідання семінару	відділ кадрів	Холод А.О.	20.05.2021	не виконано	так

Рис. 2.14. Підпункт «Доручення»

Підпункт «Довіреності» - включає в себе всі виписані довіреності, складені юридичним управлінням та іншими управліннями банку (рис. 2.15).

Номер	Дата початку	Дата закінчення	Довірена особа	Особа, що надає доручення	Зміст	Дата народження1	Дата народження2
1	15.04.2021	15.05.2021	Усачев О.Л.	Загорний А.Л.	на право отримання коштів	12.03.1983	1.09.1989
2	11.04.2021	11.07.2021	Холод А.П.	Компан К.Г.	на продаж активів	12.04.1974	12.05.1967
3	15.01.2020	17.01.2021	Макеєв П.О.	Худий В.Д.	на представлення в суда	13.02.1988	7.04.1977

Рис. 2.15. Підпункт «Довіреності»

Підпункт «Позовні заяви» включають в себе позовні заяви, які складає сам банк і позовні заяви клієнтів банку, які обробляють співробітники юридичного управління. На рис. 2.16 представлений перелік позовних заяв, з

якими працюють юристи банку.

Номер	Район	Суддя	Істець	Відповідач	Суть	Сума	Адвокат	стан	дата впровадження
1	Красногвардей...	Смольняков В.Д.	Коротков Р.О.	Іванов І.І.	відшкодування збитків	15000000 грн	Лагайко Є.О.	на розгляді	1.04.2020
2	Центральний	Жижка Л.О.	Прогоров Л.Д.	Сидоров С.С.	арешт майна	не оцінено	Вопронов П.О.	завершено	1.01.2021

Рис. 2.16. Перелік позовних заяв клієнтів

Підпункт «Претензії» (рис. 2.17) включає в себе претензійні заяви клієнтів банку і окремих управлінь банку зовнішнім структурам, які обробляють співробітники юридичного управління.

Номер	Дата отримання	Від кого	отримувач	дата відправлення	дата вручення	зміст	стан	результат	сума збитків
1	5.02.2021	Копитов А.Л.	Іванов І.І.	10.12.2020	6.02.20...	відмова в кредиті	на розгляді		
2	2.03.2021	Чжев В.М.	Петров П.П.	1.02.2021	3.03.20...	відмова в транз...	розглянуто	відмовити	0
3	4.04.2021	Сонько А.Д.	Петров П.П.	31.03.2021	4.04.20...	відмова в кредиті	розглянуто	відмовити	0
4	5.04.2021	Комов І.Д.	Іванов І.І.	30.03.2021	6.05.20...	відмова в консу...	на розгляді		

Рис. 2.17. Перелік претензійних заяв

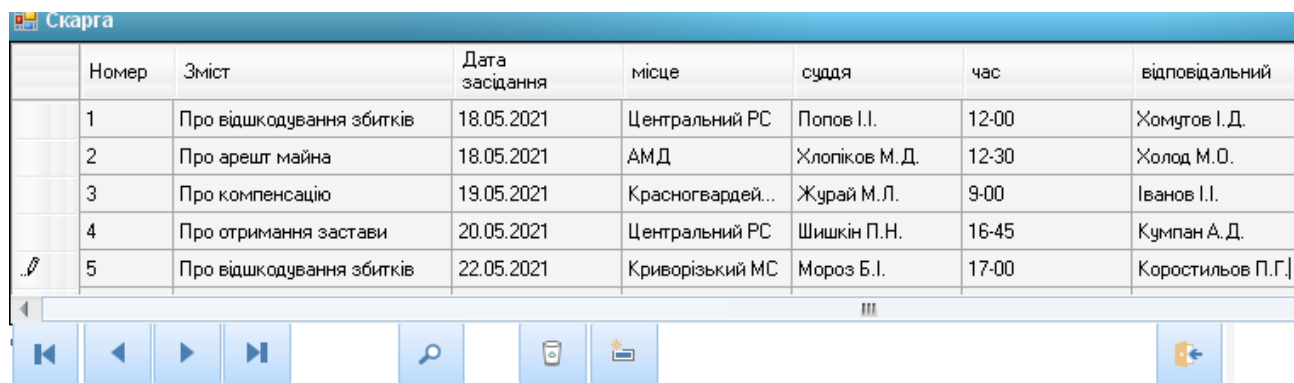
Підпункт «Скарги» - включає в себе заяви зі скаргами клієнтів банку і окремих управлінь банку зовнішнім структурам, які обробляють співробітники юридичного управління (рис. 2.18).

Номер	Дата реєстрації	Зміст	отримувач	дата отримання	виконавець	примітка	стан	результат	дата відповіді
1	14.04.2021	некоректне заповнення д...	Холод А.Р.	15.04.2021	Хорішко А.Д.	проблемний кліє...	розглянуто	догана	20.05.2021
2	15.05.2021	розторгнення договору	Іванов І.І.	16.05.2021	Мудрак П.Ж.	проблемний кліє...	розглянуто	відмова	20.05.2021
3	13.05.2021	не отримання кредиту	Сидоров С.С.	16.05.2021	Поропків І.О.		на розгляді		

Рис. 2.18. Підпункт «Скарги»

Підпункт «Слухання» - в даному меню відображені всі минулі, поточні, а

також заплановані слухання розгляду справ в судах (рис. 2.19).



Номер	Зміст	Дата засідання	місце	суддя	час	відповідальний
1	Про відшкодування збитків	18.05.2021	Центральний РС	Попов І.І.	12-00	Хомутов І.Д.
2	Про арешт майна	18.05.2021	АМД	Хлопиков М.Д.	12-30	Холод М.О.
3	Про компенсацію	19.05.2021	Красногвардей...	Журай М.Л.	9-00	Іванов І.І.
4	Про отримання застави	20.05.2021	Центральний РС	Шишкін П.Н.	16-45	Кумпан А.Д.
5	Про відшкодування збитків	22.05.2021	Криворізький МС	Мороз Б.І.	17-00	Коростильов П.Г.

Рис. 2.19. Підпункт «Слухання»

Наступним елементом меню програми є меню «Довідники» (рис. 2.20), яке також ділиться на кілька підміню:

- «Контрагенти» - в цьому довіднику відображено інформацію про всіх контрагентів в минулому та теперішньому часі. Також в даному меню можна редагувати і заносити нових контрагентів (рис. 2.21);

- «Суди» - список всіх судів міста та області, в яких розглядаються, або можуть розглядатися справи, що стосуються діяльності банку (рис. 2.22);

- «Співробітники» в даному підміню представлений список співробітників, які коли-небудь працювали та працюють в установі. Також доступна коригування вже наявних записів і внесення нових записів (рис. 2.23).

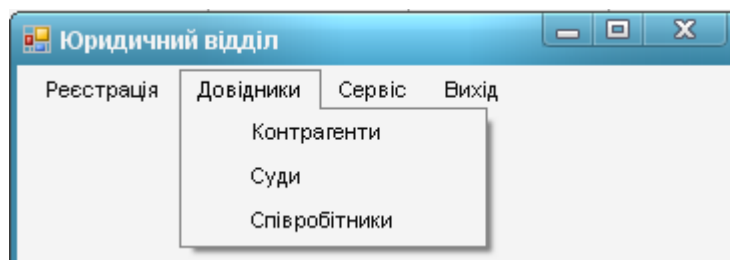


Рис. 2.20. Меню «Довідники»

Контрагенти								
Номер	Код контрагента	Назва	ІНН	юр. адреса	телефон	керівник	бухгалтер	
1	11425	ЗАТ "Дніпробуд"	123456789	Дніпро, Плекан...	0509475214	Комов П.Д.	Наумов І.Л.	
2	76243	ВАТ "СелтікПро"	987456321	Дніпро, Поля 113	0632547412	Іванкін П.Д.	Карсілов П.Т.	
3	48763	ФОП "Козлов"	147852369	Каменське, Світ...	0665478521	Воропаєв П.С.	Комунський А.Ш.	
4	22343	ФОП "ЛОМАКІН"	963258741	Крийий Ріг, Дру...	0971254785	Ломакін А.А.	Хоров Д.Д.	
5	11232	ПП "Коротсиль...	123654789	Дніпро, Слобож...	0441254785	Маркін В.Д.	Щука С.М.	

Рис. 2.21. Підменю «Контрагенти»

Суди				
Номер	Назва	адреса	телефон	
1	Центральний РС	пр. Яворницького, 42	7445241	
2	Красногвардейськ...	пр. Пушкіна 47	3457412	
3	Криворізький МС	вул. Черненко, 25	1254754	
4	АМД РС	пр. Слобожанський, 46	7364512	

Рис. 2.22. Підменю «Суди»

Співробітники										
Табельний номер	Прізвище	Ім'я	По батькові	паспорт	ІНН	адреса проживання	телефон	посада	дата прийняття	
124565	Іванов	Іван	Іванович	АЕ 34526543	123654789	пр. Пушкіна, 25/...	0505412547	юрист	12.03.2008	
458412	Петров	Петро	Петрович	АН 3452673	741258963	вул. Тополина 2...	0666521478	юрист	13.08.2002	
452148	Сидоров	Сидор	Сидорович	КВ 2453652	369852147	вул. Тітова 12/24	0971452541	спеціаліст	3.02.2026	
145236	Холод	Олексій	Іванович	АД 3542434	321456987	Вул. Янгарна, 2	0635214587	кур'єр	8.04.2019	

Рис. 2.23. Підменю «Співробітники»

Наступним, елементом меню програми є меню «Сервіс», яке також ділиться на кілька підменю (рис. 2.24):

- «Перегляд файлів» - перегляд списку файлів, які зберігаються в базі даних архіву;
- «Налаштування» - це меню призначається для управління настройками програми-клієнта і, не доступно пересічному користувачеві.
- Облікові записи - це меню призначається для управління обліковими записами користувачів програми-клієнта і, не доступно пересічному

користувачеві.

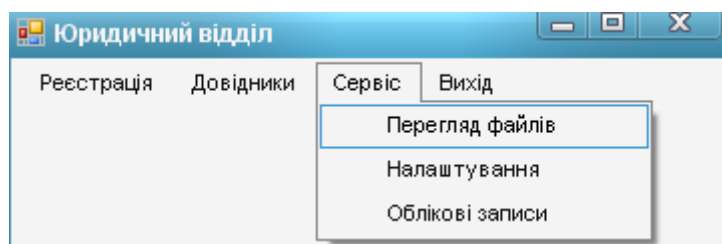


Рис. 2.24. Меню «Сервіс»

Залежно від «глибини» відкривання вкладок, склад функціональних кнопок може змінюватися в більшу або меншу сторону. Не залежно від того, скільки функціональних клавiш мається на відміну від «кореневої» вкладки, в будь-якому меню програми призначення цих клавiш залишається незмінним.

Всі документи, що додані користувачами через програму-клієнта, повинні бути в обов'язковому порядку зареєстровані на сервері електронного архіву документів. Якщо якийсь документ не був зареєстрований, то інформація про нього не буде доступна всім користувачам. При введенні даних про документ у програму-клієнт, необхідно обов'язково заповнити всі поля (щоб надалі полегшити іншим користувачам і собі, пошук за ключовими словами і датами).

Реєстрація файлів на сервері проводиться користувачами вручну.

Після того, як була проведена реєстрація документа на сервері електронного документообігу, цей документ стає доступним для користування всім співробітникам, які працюють з даною програмою.

РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Вхідні дані:

- передбачуване число операторів - 1650;
- коефіцієнт складності програми - 1,5;
- коефіцієнт корекції програми в ході її розробки - 0,06;
- годинна заробітна плата програміста, грн / год - 55.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_u + t_a + t_n + t_{om} + t_d, \text{ людино-годин,} \quad (3.1)$$

де t_u - витрати праці на підготовку й опис поставленої задачі (50),

t_n - витрати праці на дослідження алгоритму рішення задачі,

t_a - витрати праці на розробку блок-схеми алгоритму;

t_{om} - витрати праці на програмування по готовій блок-схемі,

t_{otl} - витрати праці на налагодження програми на ЕОМ,

t_d - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \times C \times (1 + p), \text{ людино-годин,} \quad (3.2)$$

де q - передбачуване число операторів,

C - коефіцієнт складності програми,

p - коефіцієнт кореляції програми в ході її розробки.

$$Q = 1650 \cdot 1,5 \cdot (1 + 0,06) = 2623,5 \text{ людино-годин.}$$

Витрати праці на вивчення опису задачі ти визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{QB}{(75 \dots 85)K}, \text{ людино-годин,} \quad (3.3)$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі; $B=1.2 \dots 1.5$,

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності.

Витрати праці на розробку алгоритму рішення задачі:

$$t_u = \frac{2623,5 \cdot 1,4}{78 \cdot 1,2} = 39,24, \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_\alpha = \frac{Q}{(20 \dots 25)K} \text{ людино-годин.,} \quad (3.4)$$

$$t_\alpha = \frac{2623,5}{21 \cdot 1,2} = 104,1 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20..25)K} \quad \text{людино-годин,} \quad (3.5)$$

$$t_n = \frac{2623.5}{21 \cdot 1.2} = 104.1 \quad \text{людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

$$t_{\text{отл}} = \frac{Q}{(4...5)K} \quad \text{людино-годин,} \quad (3.6)$$

$$t_{\text{отл}} = \frac{2623.5}{4 \cdot 1.28} = 546.56 \quad \text{людино-годин.}$$

- за умови комплексного налагодження завдання:

$$t_{\text{отл}}^{\text{к}} = 1,2 \cdot t_{\text{отл}} ; \quad (3.7)$$

$$t_{\text{отл}}^{\text{к}} = 546.56 \cdot 1,2 = 655,872$$

Витрати праці на підготовку документації:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \quad \text{людино-годин,} \quad (3.8)$$

де $t_{\partial p}$ - трудомісткість підготовки матеріалів і рукопису.

$$t_{\partial p} = \frac{Q}{(15..20)K}, \quad \text{людино-годин.,} \quad (3.9)$$

$$t_{\partial p} = \frac{2623.5}{15 \cdot 1.2} = 145,75 \quad \text{людино-годин.}$$

$t_{\partial o}$ - трудомісткість редагування, печатки й оформлення документації:

$$t_{\partial o} = 0,75 \cdot t_{\partial p}, \quad \text{людино-годин,} \quad (3.10)$$

$$tdo = 0,75 \cdot 145,75 = 109.31$$

$$t_{ii} = 145.78 + 109.31 = 255.06 \text{ людино-годин.}$$

Отримуємо трудомісткість розробки програмного забезпечення:

$$t = 50 + 39.24 + 104.10 + 104.10 + 546.56 + 255.06 = 1049.06 \text{ люд-год.}$$

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ Кпо включають витрати на заробітну плату виконавця програми Зз/п і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{по} = З_{зп} + З_{мв}, \text{ грн,} \quad (3.11)$$

Заробітна плата виконавців визначається за формулою:

$$З_{зп} = t \cdot C_{спр}, \text{ грн,} \quad (3.12)$$

де t - загальна трудомісткість, людино-годин,

$C_{спр}$ - середня годинна заробітна плата програміста, грн/година.

$$C_{сі} = 1049.06 \cdot 55 = 57698.3 \text{ їдї.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{мв} = t_{отл} \times C_{см}, \text{ грн,} \quad (3.12)$$

де $t_{отл}$ - трудомісткість налагодження програми на ЕОМ, год.,

$C_{см}$ - вартість машино-години ЕОМ, 3,5 грн/год.

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

$$Q_A = 546.56 \times 3.5 = 1913 \text{ \textit{ãđĭ}}.$$

$$\hat{E}_{\text{it}} = 57698.3 + 1913 = 59611.3 \text{ \textit{ãđĭ}}.$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p} \text{ міс.} \quad (3.13)$$

де B_k - число виконавців,

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

$$T = \frac{1049}{1 \cdot 176} = 6 \text{ міс.}$$

Висновки: визначено трудомісткість розробленої інформаційної системи (1049 люд-год), проведений підрахунок вартості роботи по створенню програми (59611 грн.) та розраховано час на його створення (6 міс).

ВИСНОВКИ

Метою кваліфікаційної роботи є розробка інформаційної підсистеми електронного архіву документів юридичного відділу банку для оптимізації методів зберігання та обробки електронних документів. Дана система повинна спростити процес зберігання і архівацію електронних документів і прибрати недоліки, властиві використанню документів на паперових носіях.

Система призначена для швидкого пошуку проходження документів в організації і надає наступні можливості:

- реєстрацію документів;
- розподіл електронних документів між співробітниками;
- контроль проходження документів;
- контроль виконання доручень і документів;
- створення звітів і журналів.

Розроблена система є багатокористувальницькою та забезпечує координовану спільну роботу колективу співробітників. Цим обумовлена як структура системи, так і вимоги до її налаштування: ідентифікація користувачів системи, управління правами доступу до оброблюваної інформації, забезпечення цілісності і збереження інформації.

Користувачам системи можна привласнити такі права:

- введення в систему нових документів. Користувач, наділений такими правами, може виконувати функції реєстратора документів, а також переглядати (в режимі звіту) список всіх файлів зареєстрованих на сервері;
- оновлення довідників. Користувач, наділений такими правами, може вносити в довідкові таблиці програми нові дані, а також коригувати старі.
- адміністрування системи. Користувач, наділений такими правами, може виконувати функції адміністратора системи, зокрема, створювати облікові записи користувачів, призначати користувачам права, змінювати налаштування системи.

Для розробки інформаційної системи була обрана мова програмування

C#. За допомогою цієї мови програмування можна якісно створити програмний додаток з максимальними можливостями за даною темою кваліфікаційної роботи. Базу даних створено за допомогою системи керування базами даних MySQL та використано платформу Windows Forms, яка націлена на створення графічного інтерфейсу додатку. Розробка програмного продукту здійснена у середовищі розробки Microsoft Visual Studio 2019.

В економічному розділі визначено трудомісткість розробленої інформаційної системи (1049 люд-год), проведений підрахунок вартості роботи по створенню програми (59611 грн.) та розраховано час на його створення (6 міс).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Безбородько І.Г. Все про бази даних - Львів 2013, 2014 - 276с.
2. Бейлі Л. Вивчаємо SQL. - Львів: Символ-Плюс, 2000. - 120 с.
3. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання : (ГОСТ 7.1-2003, IDT) : ДСТУ ГОСТ 7.1:2006. – Чинний з 2007–07–01. – К. : Держспоживстандарт України, 2007. – 47 с. – (Система стандартів з інформації, бібліотечної та видавничої справи) (Національний стандарт України).
4. Бусигін Б.С., Коротенко Г.М., Коротенко Л.М. Прикладна інформатика. Підручник для студентів комп'ютерних спеціальностей. – Дніпропетровськ: Видавництво НГУ, 2004. – 559 с. URL: <http://www.programmer.dp.ua/book-ua-k01.php>. дата звернення: 15.03.2019.
5. Бусыгин Б.С., Дивизинюк М.М., Коротенко Г.М., Коротенко Л.М. Введение в современную информатику. Учебник. – Севастополь: Издательство СНУЯЭиП, 2005. – 644 с. / URL: <http://www.programmer.dp.ua/book-ru-k02.php>. дата звернення: 15.01.2018.
6. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення : ДСТУ 3008-95. – Чинний від 1996–01–01. – К. : Держстандарт України, 1996. – 39 с.
7. ДСТУ 2394-94 Інформація та документація. Комплектування фонду, бібліографічний опис, аналіз документів. Терміни та визначення. – Чинний від 01.01.1995. - Київ : Держстандарт України, 1994. – 88 с.
8. Е. Троелсен, Ф. Джепикс Мова програмування С# 7 і платформи .NET і .NET Core 8, Москва, Діалектика, 2018.
9. Інагородній В.О. Visual Studio 2017 - Київ 2016, 2018 - 377с.
10. Інтернет-сайт URL: <https://uk.wikipedia.org> Дата звернення: 15.03.2021.
11. Інтернет-сайт URL: <https://www.microsoft.com/> Дата звернення: 15.03.2021.
12. Інтернет-сайт СофтКаталог URL: www.softcatalog.info/ru/obzor. Дата

звернення: 15.03.2021.

13. Інтернет-сайт Етапи тестування програмних додатків URL: www.lemon.com/site. Дата звернення: 15.03.2021.

14. Криворученко К.П. Професійний С# - Харків 2014, 2015 - 100с.

15. Мацяшек Л.А. «Анализ требований и проектирование систем» – М.: *Издательский дом Вильямс*, 2002. - 432 с.

16. Методичні вказівки з виконання економічного розділу в дипломних проектах студентів спеціальності “Комп’ютерні системи ” / Укладачі О.Г. Вагонова, Нікітіна О.Б. Н.Н. Романюк – Дніпропетровськ: Національний гірничий університет. – 2013. – 23с.

17. Методичні рекомендації до виконання кваліфікаційних робіт бакалаврів напряму підготовки 122 «Комп’ютерні науки» галузі знань 12 Інформаційні технології/, Л.М. Коротенко, О.С. Шевцова; Нац. гірн. ун-т. – Д.: ДВНЗ НГУ, 2019. – 65 с.

18. Методичні рекомендації щодо написання, оформлення та представлення учнівських науково-дослідницьких робіт учнів – членів Малої академії наук України / Г.Г. Півняк, Л.М. Коротенко, І.М. Удовик, Є.М. Головня – Д.: ДВНЗ «Національний гірничий університет», 2017. – 24 с.

19. Скорочення слів в українській мові у бібліографічному описі. Загальні правила та вимоги : ДСТУ 3582-97. – Чинний від 1998-07-01. – К. : Держстандарт України, 1998. – 24 с. – (Державний стандарт України).

20. Офіційний сайт середовища розробки Visual Studio Code URL: <https://code.visualstudio.com/docs> Дата звернення: 15.03.2021.

КОД ПРОГРАМИ

```
Main.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using WpfApp1.Sections;

namespace WpfApp1
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            Home_page home_Page = new Home_page();
            Frame_main.Navigate(home_Page);
        }

        private void Exit_Click(object sender, RoutedEventArgs e)
        {
            this.Close();
        }
    }
}
```

```

}

private void WindowMinimize_Click(object sender, RoutedEventArgs e)
{
    this.WindowState = WindowState.Minimized;
}

private void Grid_MouseDown(object sender, MouseButtonEventArgs e)
{
    try
    {
        this.DragMove();
    }
    catch
    {
    }
}

private void bt_section_1_Click(object sender, RoutedEventArgs e)
{
    Section_1 section_1 = new Section_1();
    Frame_main.Navigate(section_1);
}

private void bt_section_2_Click(object sender, RoutedEventArgs e)
{
    Section_2 section_2 = new Section_2();
    Frame_main.Navigate(section_2);
}

private void bt_section_3_Click(object sender, RoutedEventArgs e)
{
    Section_3_theory section_3 = new Section_3_theory();
    Frame_main.Navigate(section_3);
}

```



```

private void Menu_Click(object sender, RoutedEventArgs e)
{
    if (Section_1.mark >= 5)
    {
        bt_section_2.IsEnabled = true;
        if (Section_2.mark >= 5)
        {
            bt_section_3.IsEnabled = true;
        }
        else
        {
            bt_section_3.IsEnabled = false;
        }
    }
    else
    {
        bt_section_2.IsEnabled = false;
    }
}

private void bt_introduction_Click(object sender, RoutedEventArgs e)
{
    Introduction introduction = new Introduction();
    Frame_main.Navigate(introduction);
}

private void bt_cval_Click(object sender, RoutedEventArgs e)
{
    Pracktice pracktice = new Pracktice();
    Frame_main.Navigate(pracktice);
}
}

par.Clear();

```

```

listb_analizet.Items.Clear();
//обработка инфо из файлов

//bonded for atoms
int i_bonded = 0, i_notrealistic = 0, i_angles = 0, i_dihedralas = 0;
for (int i = 0; i < param.Length; i++)
{
    if (param[i].StartsWith("#bonding parameters") == true)
    {
        i_bonded = i;
    }
    else if (param[i].StartsWith("#not realistic bonds") == true)
    {
        i_notrealistic = i;
    }
    else if (param[i].StartsWith("#angles") == true)
    {
        i_angles = i;
    }
    else if (param[i].StartsWith("#dihedrals") == true)
    {
        i_dihedralas = i;
    }
}

for (int i = i_dihedralas + 1; i < param.Length; i++)
{
    for (int j = 0; j < 10; j++)
    {
        dihedrals_param[i - i_dihedralas - 1, j] = param[i].Split(tab)[j];
    }
}

for (int i = i_angles + 1; i < i_dihedralas; i++)

```

```

{
  for (int j = 0; j < 5; j++)
  {
    angles_param[i - i_angles - 1, j] = param[i].Split(tab)[j];
  }
}

for (int i = i_notrealistic + 1; i < i_angles; i++)
{
  for (int j = 0; j < 4; j++)
  {
    not_realistic[i - i_notrealistic - 1, j] = param[i].Split(tab)[j];
  }
}

for (int i = i_bonded + 2; i < i_notrealistic; i++)
{
  for (int j = 0; j < 4; j++)
  {
    bonds_param[i - i_bonded - 2, j] = param[i].Split(tab)[j];
  }
}

for (int i = 1; i < i_bonded; i++)
{
  param_sab = param[i].Split(tab);
  for (int j = 0; j < 4; j++)
  {
    param_sub[j] = param_sab[j + 1];
  }
  string[] clone = new string[param_sub.Length];
  for (int k = 0; k < clone.Length; k++)
  {
    clone[k] = param_sub[k];
  }
}

```

```

    par.Add(param_sab[0], clone);
}

///  

int i_angels = 0;
for (int i = 0; i < connections.Length; i++)
{
    if (connections[i].StartsWith("#angles") == true)
    {
        i_angels = i;
    }
}
i_angels = i_angels + 1;
for (int i = 1; i < i_angels - 1; i++)
{
    bonds_sab = connections[i].Split(tab);
    bonds_sub[i - 1, 0] = bonds_sab[0];
    bonds_sub[i - 1, 1] = bonds_sab[1];
}

int i_dihedrals = 0;
for (int i = 0; i < connections.Length; i++)
{
    if (connections[i].StartsWith("#dihedrals") == true)
    {
        i_dihedrals = i;
    }
}
i_dihedrals = i_dihedrals + 1;
for (int i = i_angels; i < i_dihedrals - 1; i++)
{
    angels_sab = connections[i].Split(tab);
    angels_sub[i - 16, 0] = angels_sab[0];
    angels_sub[i - 16, 1] = angels_sab[1];
}

```

```

    angels_sub[i - 16, 2] = angels_sab[2];
}

for (int i = i_dihedrals; i < connections.Length; i++)
{
    dihedrals_sab = connections[i].Split(tab);
    dihedrals_sub[i - 33, 3] = dihedrals_sab[3];
}

//create atoms
string decSep = NumberFormatInfo.CurrentInfo.NumberDecimalSeparator;
for (int i = 1; i < coord.Length; i++)
{
    coord_sab = coord[i].Split(tab);
    string name = coord_sab[1];
    ptNames[i - 1] = name;
    for (int j = 0; j < 6; j++)
    {
        if (j == 0)
        {
            var mass = float.Parse(par[name][0].Replace(".", decSep));
            var charge = float.Parse(par[name][1].Replace(".", decSep));
            var sigma = float.Parse(par[name][2].Replace(".", decSep));
            var epsilon = float.Parse(par[name][3].Replace(".", decSep));
            atoms[i - 1] = new Atom(i, name, mass, charge, sigma, epsilon, coord_n_speed);
            listb_analizet.Items.Add(atoms[i - 1].Describe());
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfApp1.Sections
{
    public partial class Section_2_test : Page
    {
        public static int a1_1, a1_2, a2, a3_1, a3_2, a3_3, a4, a5;

        public Section_2_test()
        {
            InitializeComponent();
        }
    }
}

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;

```

```

using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Windows.Xps.Packaging;

namespace WpfApp1.Sections
{
    public partial class Section_2_theory : Page
    {
        public Section()
        {
            InitializeComponent();
            XpsDocument doc = new XpsDocument("../Sections/Section2.xps", FileAccess.Read);
            dc_viewer.Document = doc.GetFixedDocumentSequence();
            dc_viewer.Zoom = 80;
            doc.Close();
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;

```

```

using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Windows.Xps.Packaging;

namespace WpfApp1.Sections
{
    public partial class Section_3_theory : Page
    {
        public ()
        {
            InitializeComponent();
            XpsDocument doc = new XpsDocument("../Sections/Section3.xps", FileAccess.Read);
            dc_viewer.Document = doc.GetFixedDocumentSequence();
            dc_viewer.Zoom = 80;
            doc.Close();
        }
    }
}

```

Program.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace man2
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {

```



```
Application.EnableVisualStyles();
Application.SetCompatibleTextRenderingDefault(false);
Application.Run(new Form1());
}
}
}
```

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace m
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form F5 = new Form5();
            F5.Show();
            this.Hide();
        }

        private void button4_Click(object sender, EventArgs e)
```

```
{
    Form F2 = new Form2();
    F2.Show();
    this.Hide();
}

private void button5_Click(object sender, EventArgs e)
{
    Form F3 = new Form3();
    F3.Show();
    this.Hide();
}

private void button3_Click(object sender, EventArgs e)
{
    Form F4 = new Form4();
    F4.Show();
    this.Hide();
}

private void button2_Click(object sender, EventArgs e)
{
    Form F6 = new Form6();
    F6.Show();
    this.Hide();
}

private void button6_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void button1_Click_1(object sender, EventArgs e)
{
    Form F7 = new Form7();
```

```

        F7.Show();
        this.Hide();
    }

}
}

```

Form10.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

namespace mani2

```

{
    public partial class Form10 : Form
    {
        public Form10()
        {
            InitializeComponent();
        }

        private void Form10_Load(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form F7 = new Form7();

```

```
        F7.Show();
        this.Hide();
    }
}
}
```

Form11.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace man2018
{
    public partial class Form11 : Form
    {
        public Form11()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form F7 = new Form7();
            F7.Show();
            this.Hide();
        }
    }
}
```

```

Form2.cs
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace man2
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void Form2_Load(object sender, EventArgs e)
        {
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Form F7 = new Form7();
            F7.Show();
            this.Hide();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form F3 = new Form3();
            F3.Show();
        }
    }
}

```

```
        this.Hide();
    }
}
}
```

Form3.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace man2
```

```
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form F7 = new Form7();
            F7.Show();
            this.Hide();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Form F2 = new Form2();
```

```

        F2.Show();
        this.Hide();
    }
}
}

```

Data.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Windows;
using System.Windows.Media;
using static Modeling.wElementCreator;
namespace Modeling
{
    static class Data
    {
        public static event Action ElementsChanged;
        //налаштування за замовчанням. Існують тільки тут.
        public static string DataBasePathDefault { get; } = Environment.CurrentDirectory +
        "\\Sources\\Elements.xml";
        //public
        public static readonly double dataScale;
        public static readonly char splitSymbol;
        public static readonly double maxSpeedRatio;
        public static readonly double minSpeedRatio;
        public static readonly Brush groundColor;
        public static readonly Brush grassColor;
        public static readonly Brush skyColor;
        public static readonly Brush normalBorderBrush;
        public static readonly Brush chosentBorderBrush;
        public static readonly Brush guidesForceBrush;
        public static readonly Brush guidesBackBrush;
        public static readonly Brush promptsForeground;
        public static readonly List<Element> Elements;
        //Інкапсулюються властивостями
    }
}

```

```

private static string dataBasePath;
private static double minimalCalculateTime;
private static int maxFramesCount;
private static double scrollStep;
private static double minScale;
private static double translateFields;
private static double averageSpeedRatio;
    //Властивості
public static string DataBasePath
{
    get
    {
        return dataBasePath;
    }
    set
    {
        trySaveData();
        dataBasePath = value;
        Elements.Clear();
        if (ElementsChanged != null)
            ElementsChanged();
        loadData();
    }
}
public static double MinimalCalculateTime {
    get { return minimalCalculateTime; }
    set
    {
        if (value < MinTimeDefault)
            minimalCalculateTime = MinTimeDefault;
        else
            minimalCalculateTime = value;
        isSettingsSave = false;
    }
}
public static int MaxFramesCount {
    get { return maxFramesCount; }
    set
    {
        if (value < MaxFramesCountDefault)
            maxFramesCount = MaxFramesCountDefault;
        else

```



```

        maxFramesCount = value;
        isSettingsSave = false;
    }
}

public static double ScrollStep
{
    get { return scrollStep; }
    set
    {
        if (value < 1)
            scrollStep = 1;
        else
            if (value > 10)
                scrollStep = 10;
            else
                scrollStep = value;
        isSettingsSave = false;
    }
}

public static double MinScale
{
    get { return minScale; }
    set
    {
        if (value <= 0)
            minScale = MinScaleDefault;
        else
            if (value > 1)
                minScale = 1;
            else
                minScale = value;
        isSettingsSave = false;
    }
}

public static double TranslateFields
{
    get { return translateFields; }
    set
    {
        if (value < 0)
            translateFields = 0;
        else
            translateFields = value;
    }
}

```

```

        isSettingsSave = false;
    } }
public static double AverageSpeedRatio
{
    get { return averageSpeedRatio; }
    set
    {
        if (value < minSpeedRatio)
            averageSpeedRatio = minSpeedRatio;
        else
            if (value > maxSpeedRatio)
                averageSpeedRatio = maxSpeedRatio;
            else
                averageSpeedRatio = value;
        isSettingsSave = false;
    } }
public static String TheoryPath { get; }
//private
private static readonly string[] explanationPaths;
private static Stack<Window> windowHistory;
private static string settingsPath;
private static bool isLoading;
private static bool isDataSave { get; set; }
private static bool isSettingsSave { get; set; }
//TODO: абсолютний шлях до теорії.
static Data()
{
    splitSymbol = ' ';
    dataScale = 100;
    maxSpeedRatio = 5;
    minSpeedRatio = 0.2;
    groundColor = new SolidColorBrush( Color.FromRgb(93, 31, 0));
    grassColor = new SolidColorBrush(Colors.Green);
    skyColor = new SolidColorBrush(Color.FromRgb(0, 190, 255));
    normalBorderBrush = new SolidColorBrush(Colors.Black);
    chosentBorderBrush = new SolidColorBrush(Color.FromRgb(0,0,150));
    guidesForceBrush = new SolidColorBrush(Colors.Black);

```

```

guidesBackBrush = new SolidColorBrush(Colors.Blue);
promptsForeground = new SolidColorBrush(Colors.BlanchedAlmond);
Elements = new List<Element>();
windowHistory = new Stack<Window>();
explanationPaths =new string[] { Environment.CurrentDirectory +
"\\Sources\\Explanations\\"+wExplanation.ExplanationType.MenuMain.ToString()+".html",
        Environment.CurrentDirectory +
"\\Sources\\Explanations\\"+wExplanation.ExplanationType.MenuModeling.ToString()+".html",
        Environment.CurrentDirectory +
"\\Sources\\Explanations\\"+wExplanation.ExplanationType.ElementCreator.ToString()+".html",
        Environment.CurrentDirectory +
"\\Sources\\Explanations\\"+wExplanation.ExplanationType.Modeling.ToString()+".html"};
}
//Ініціалізація. Повинна бути виконана при старті програми.
public static void initialize()
{
    try
    {
        /
        using (StreamReader sr = new StreamReader(settingsPath))
        {
            Dictionary<String, String> dat = new Dictionary<String, String>();
            bool isCorrect = true;
            string[] temp = sr.ReadLine().Split(' ','');
            dat.Add(temp[0], temp[temp.Length-1]);
            temp = sr.ReadLine().Split(' ','');
            dat.Add(temp[0], temp[temp.Length - 1]);
            temp = sr.ReadLine().Split(' ','');
            dat.Add(temp[0], temp[temp.Length - 1]);
            temp = sr.ReadLine().Split(' ','');
            dat.Add(temp[0], temp[temp.Length - 1]);
            temp = sr.ReadLine().Split(' ','');
            dat.Add(temp[0], temp[temp.Length - 1]);
            temp = sr.ReadLine().Split(' ','');
            dat.Add(temp[0], temp[temp.Length - 1]);
            temp = sr.ReadLine().Split(' ','');
            dat.Add(temp[0], temp[temp.Length - 1]);
            MinimalCalculateTime = double.Parse(dat["MinimalCalculateTime"]);

```

```

if (double.Parse(dat["MinimalCalculateTime"]) != MinimalCalculateTime)
    isCorrect = false;
MaxFramesCount = int.Parse(dat["MaxFramesCount"]);
if (double.Parse(dat["MaxFramesCount"]) != MaxFramesCount)
    isCorrect = false;
MinScale = double.Parse(dat["MinScale"]);
if (double.Parse(dat["MinScale"]) != MinScale)
    isCorrect = false;
TranslateFields = double.Parse(dat["TranslateFields"]);
if (double.Parse(dat["TranslateFields"]) != TranslateFields)
    isCorrect = false;
ScrollStep = double.Parse(dat["ScrollStep"]);
if (double.Parse(dat["ScrollStep"]) != ScrollStep)
    isCorrect = false;
AverageSpeedRatio = double.Parse(dat["AverageSpeedRatio"]);
if (double.Parse(dat["AverageSpeedRatio"]) != AverageSpeedRatio)
    isCorrect = false;
        if (dat["DataBasePath"].StartsWith("\\"))
            temp[0] = Environment.CurrentDirectory + dat["DataBasePath"];
DataBasePath = temp[0];
if (!isDataSave)
{
    DataBasePath = DataBasePathDefault;
    isCorrect = false;
}
//Якщо досі isCorrect - true, то налаштування у файлі коректні
if (isCorrect)
    isSettingsSave = true;
isDataSave = true;          }          }
catch (Exception e)
{
    if (e is IndexOutOfRangeException || e is FormatException || e is
OverflowException || e is ArgumentNullException)
        MessageBox.Show("Файл: "+settingsPath+ " має не вірний формат.           else
        MessageBox.Show(e.Message, "Помилка", MessageBoxButton.OK,
MessageBoxImage.Error);
//Заносимо дефолтні значення

```

```

MinimalCalculateTime = MinTimeDefault;
MaxFramesCount = MaxFramesCountDefault;
MinScale = MinScaleDefault;
TranslateFields = TranslateFieldsDefault;
ScrollStep = TranslateFieldsDefault;
AverageSpeedRatio = AverageSpeedRatioDefault;
DataBasePath = DataBasePathDefault;
}
finally
{
    //Програма завантажилась
    isLoading = true;
}
}
private static void loadData()
{
    try
    {
        using (FileStream fs = new FileStream(DataBasePath, FileMode.Open))
        {
            DataSet ds = new DataSet();
            ds.ReadXml(fs, XmlReadMode.InferTypedSchema);
            fs.Close();
            ElementCreator.createElements(ds.Tables[0]); //завжди одна таблиця
            isDataSave = true;
        }
    }
    catch
    {
    }
    if (!isDataSave && isLoading)
        if (MessageBox.Show("Данні не збережені. Зберегти?", "Збереження",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            saveData();
}
public static void trySaveSettings() //зберегти налаштування з підтвердженням
{
    if (!isSettingsSave)
        saveSettings();
}
public static void saveChange() //зберегти усе без підтвержень
{
    saveData();
    saveSettings();
}

```

```

}
private static void saveSettings()
{
    using (StreamWriter sw = new StreamWriter(settingsPath))
    {
        sw.WriteLine("MinimalCalculateTime: " + minimalCalculateTime.ToString());
        sw.WriteLine("MaxFramesCount: " + maxFramesCount.ToString());
        sw.WriteLine("MinScale: " + MinScale.ToString());
        sw.WriteLine("TranslateFields: " + TranslateFields.ToString());
        sw.WriteLine("ScrollStep: " + ScrollStep.ToString());
        sw.WriteLine("AverageSpeedRatio: " + AverageSpeedRatio.ToString());
        if(dataBasePath.Contains(Environment.CurrentDirectory))
            sw.WriteLine("DataBasePath: " +
DataBasePath.Replace(Environment.CurrentDirectory, ""));
        else
            sw.WriteLine("DataBasePath: " + DataBasePath);
        isSettingsSave = true;
    }
}

```

ДОДАТОК Б
ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Диплом_ doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Диплом_.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація_.ppt	Презентація кваліфікаційної роботи