

РЕФЕРАТ

Кваліфікаційна робота на здобуття освітнього ступеню «бакалавр» за спеціальністю 121 – Інженерія програмного забезпечення. Тернопільський національний технічний університет ім. Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СПс-43, 2023 рік. Пояснювальна записка до кваліфікаційної роботи на здобуття освітнього ступеню «бакалавр» містить: 56 с., 21 рис., 8 табл., 2 додатка.

Ключові слова: база даних, сутність, атрибут, зв'язок, TRANSACT SQL, BIBLIOTEKA, сортування, пошук.

Ця кваліфікаційна робота присвячена розробці інформаційної системи для інвентаризації бібліотечного фонду. Ця система буде здатна автоматизувати всі бібліотечні процеси, включаючи процес інвентаризації фонду. Робота стає актуальною через недосконалість наявних систем безпеки та тестування у використовуваних ліфтових системах.

У процесі проведення дослідження був здійснений аналіз наявних систем інвентаризації, з'ясовані їх позитивні аспекти та недоліки. Був також проведений огляд компонентів, що входять до розроблюваної системи, і обґрунтовано обраний підхід. Забезпечено взаємодію програмного та апаратного забезпечення обраних елементів. Крім того, була розроблена структура та функціональна частина системи.

У результаті була створена інформаційна система інвентаризації бібліотечних фондів, спрямована на спрощення процесу ведення обліку видань у бібліотеці.

ANNOTATION

Qualification work for obtaining a bachelor's degree in specialty 121 - Software engineering. Ternopil National Technical University named after Ivan Pulyuya, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, CPs-43 group, 2023. The explanatory note to the qualification work for obtaining the bachelor's degree contains: 56 pages, 21 figures, 8 tables, 2 appendices.

Keywords: database, entity, attribute, relationship, TRANSACT SQL, LIBRARY, sorting, search.

This qualification work is devoted to the development of an information system for inventorying the library fund. This system will be able to automate all library processes, including the stock inventory process. The work becomes relevant due to the imperfection of existing safety and testing systems in the used elevator systems

In the process of conducting the research, an analysis of the existing inventory systems was carried out, their positive aspects and shortcomings were clarified. An overview of the components included in the developed system was also conducted, and the approach was reasonably chosen. The interaction of the software and hardware of the selected elements is ensured. In addition, the structure and functional part of the system was developed.

As a result, an information system for the inventory of library funds was created, aimed at simplifying the process of keeping records of publications in the library.

ЗМІСТ

РЕФЕРАТ	4
ANNOTATION	5
ВСТУП	7
1 АНАЛІТИЧНА ЧАСТИНА	8
1.1 Опис об'єкта автоматизації.....	8
1.2 Огляд подібних проектних рішень.....	10
2 ПРОЕКТНА ЧАСТИНА	13
2.1 Постановка задачі	13
2.2 Проектування бази даних.....	14
2.3 Проектування інформаційної системи.....	21
3 ПРАКТИЧНА ЧАСТИНА	25
3.1 Проектування інтерфейсу користувача	25
3.2 Опис програмних модулів.....	32
3.3 Опис результатів тестування	34
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	41
4.1 Вимоги з безпеки праці для працівників бібліотеки	41
4.2 Загальні вимоги безпеки в бібліотеці.....	42
ВИСНОВКИ.....	45
ПЕРЕЛІК ПОСИЛАНЬ	46
ДОДАТКИ.....	48
Додаток А ER-діаграма бази даних «Biblioteka»	49
Додаток Б Текст програми «Biblioteka».....	50

ВСТУП

В сучасний період процес інвентаризації бібліотечних фондів відіграє важливу роль у бібліотечній діяльності. Основним завданням інвентаризації є точний облік друкованих та письмових матеріалів у бібліотеці.

Застосування новітніх інформаційних технологій шляхом використання бібліотечних інформаційних систем дозволяє здійснити автоматизацію всіх бібліотечних процесів, включаючи інвентаризацію фонду

Основною ціллю автоматизації будь-якого процесу є звільнення людини від необхідності брати участь у процесах отримання, перетворення, передачі та використання енергії, матеріалів або інформації, а також значне зменшення трудомісткості виконуваних операцій. Комплексна автоматизація бібліотеки надає працівникам можливість відійти від рутинних процесів та підвищити ефективність своєї роботи.

У процесі автоматизації бібліотечних фондів основними технічними засобами є:

комп'ютерне обладнання;

локальна комп'ютерна мережа;

програмне забезпечення для функціонування автоматизованої бібліотечної інформаційної системи.

Автоматизація бібліотеки є складним і постійним процесом, який залежить від багатьох факторів. Швидке та успішне впровадження залежить від наявних технічних можливостей бібліотеки, а також наявності кваліфікованих спеціалістів з інформаційними технологіями у штаті бібліотеки.

Отже, основною метою дипломного проекту є розробка та реалізація структури бази даних для ефективного зберігання різноманітної інформації. Шляхом розробки програмного забезпечення буде досягнуто оптимізації процесу інвентаризації бібліотечного фонду.

1 АНАЛІТИЧНА ЧАСТИНА

1.1 Опис об'єкта автоматизації

Бібліотечний фонд (БФ) є невід'ємною основою кожної бібліотеки. Він є необхідним та невід'ємним елементом, без якого неможливе існування та функціонування цього культурного закладу. На сучасному етапі, перед бібліотекою стоїть ряд суперечливих завдань - активне використання бібліотечного фонду та його максимальне забезпечення тривалим збереженням. Вирішення цих завдань вимагає регулярної інвентаризації фонду - перевірки фактичної наявності матеріалів у підрозділі бібліотеки, яка документується у бібліотечних облікових документах [1].

Документи, що знаходяться в БФ, поділяють на три групи:

1. Видання, до яких відносять: книги, брошури, аркушеві видання, буклети, плакати, журнали, газети.

2. Аудіовізуальні документи, до яких відносяться: фонодокументи, відеодокументи, кінодокументи, фотодокументи, предмети у вигляді окремих фотозображень.

3. Електронні видання, до яких відносять електронні документи (групу електронних документів), які пройшли редакційно-видавничу обробку, призначені для розповсюдження в незмінному вигляді та мають вихідні дані.

Назва та примірник є основними компонентами для обліку всіх типів документів, що входять до бібліотечного фонду, за винятком газет. Кожен окремий екземпляр документа, що включений до бібліотечного фонду або з нього вибуває, вважається примірником. Річний комплект газети та її назва (незалежно від зміни назви видання) є основними елементами для обліку газет в бібліотеці. Річний комплект представляє собою сукупність номерів (випусків) газети, які були накопичені у бібліотечному фонді протягом року.

Інвентаризація фонду здійснюється відповідно до п'ятирічного перспективного плану, який складається та затверджується директором бібліотеки або бібліотечної системи.

Для проведення інвентаризації фонду бібліотеки необхідний наказ директора, який визначає причини і терміни інвентаризації, методи її проведення та склад комісії. Головний бухгалтер обов'язково бере участь у цій комісії. Лідером комісії, який є керівником бібліотеки або його заступником, очолює інвентаризаційний процес.

У невеликих бібліотеках для проведення інвентаризації фонду використовуються індивідуальні облікові форми, зокрема, інвентарна книга. Перевірку фонду здійснюють два працівники. Перший працівник бере документ з полиці, оголошує його інвентарний номер та прізвище автора або перше слово назви. Другий працівник знаходить відповідний запис у індивідуальному обліку. Якщо інформація про документ, взята з полиці, відповідає запису в інвентарній книзі, то в обох документах ставлять позначки про наявність книги у фонді під час інвентаризації, розташовуючи їх у верхньому кутку титульної сторінки книги та в інвентарній книзі.

Після проведення інвентаризації, результати фіксуються у вигляді акта, що має встановлену форму. У цьому акті інвентаризаційна комісія робить висновки щодо правильності організації зберігання фонду у бібліотеці, стану ведення його обліку та використання, виконання інструкцій і розпоряджень, що стосуються цього питання. Крім того, в акті зазначається наявність нестачі примірників та їх загальна вартість. Остаточне затвердження акта інвентаризації здійснює директор бібліотеки.

та загальну її вартість. Директор бібліотеки затверджує акт інвентаризації [2].

1.2 Огляд подібних проектних рішень

Застосування сучасних інформаційних технологій за допомогою бібліотечних інформаційних систем дозволяє автоматизувати всі етапи бібліотечних процесів, включаючи інвентаризацію фонду. Такі технічні можливості для бібліотеки має програмний продукт «УФД/Бібліотека», завдяки якому можна оптимізувати процес інвентаризації бібліотечного фонду.

УФД/Бібліотека – автоматизована бібліотечна інформаційна система (АБІС), що розроблена і підтримується ТОВ «Український фондовий дім» (Київ) з 1998 року. Система призначена для комплексної автоматизації бібліотеки та враховує всі основні виробничі цикли [3].

Система «УФД/Бібліотека» охоплює всі ділянки роботи бібліотеки: комплектування, обробку і каталогізацію, обслуговування читачів і ведення обліку роботи, моніторинг статистичних показників роботи бібліотеки, взаємодію з читачами з питань комплектування, роботу з електронними документами, експорт та імпорт.

Щоб запровадити технологічні розробки з використанням можливостей «УФД/Бібліотека», необхідно закупити сканери з внутрішньою пам'яттю для перевірки в автоматизованому режимі. Відомо, що існують новітні моделі таких сканерів із накопичувачем, що зберігає у своїй пам'яті зчитані штрих-коди. При підключенні сканера до комп'ютера є можливість зберігати ці штрих-коди на його жорсткому диску [4].

Також система має такі ключові можливості як:

- апарат профілів з налаштуванням кожного робочого місця індивідуально;
- апарат шаблонів для перегляду і друку інформації;
- апарат зовнішніх вихідних форм для підготовки нестандартних звітів, визначення прав доступу до інформації для кожної групи користувачів;

– база даних під управлінням SQL сервера.

Визначальним фактором є те, що дана методика дозволить провести інвентаризацію, не зупиняючи процесів обслуговування користувачів бібліотеки.

Система «Evergreen» – ця інтегрована бібліотечна система була розроблена консорціумом PINES з метою використання в обширній публічній бібліотеці та мережі бібліотек цілого штату Джорджія, США. Ця система була призначена для обробки мільйонів записів, що належать сотням бібліотек.

Evergreen надає підтримку основних бібліотечних операцій, включаючи контроль фондів, реєстрацію користувачів та доступ до онлайн-каталогу. В системі Evergreen є функціонал для обчислення і контролю термінів видачі та повернення матеріалів. Ця система дозволяє відстежувати місцезнаходження будь-якої книги, диска або іншого документа у бібліотеці. Крім того, Evergreen має зручний веб-інтерфейс каталогу, що дозволяє користувачам легко знаходити потрібну інформацію, незалежно від місцезнаходження матеріалу

Система «Librarisica» – це безкоштовна інтегрована систем CASSIE, яка надає стандартні функції обслуговування користувачів у читальних залах бібліотеки. Серед цих функцій входять контроль доступу та обмеження часу використання комп'ютерів, керування друком, персональна аутентифікація користувачів та генерація статистичних звітів.

CASSIE також пропонує унікальні візуальні функціональності для керування, що дозволяють бібліотекарям та користувачам взаємодіяти з системою за допомогою зручного графічного інтерфейсу. Більше того, різні співробітники можуть одночасно виконувати диспетчерські функції завдяки цим візуальним інструментам управління.

Серед недоліків використання та обслуговування сучасних інформаційних систем інвентаризації БФ можна відмітити необхідність особливої професійної підготовки персоналу бібліотеки, а також високу цінову політику розробників програмного забезпечення. Для звичайних районних бібліотек такі витрати на

придбання, встановлення та обслуговування програмних продуктів для автоматизації інвентаризації можуть бути неприйнятно великими.

Отже, з'явилася потреба у розробці безкоштовної програми, яка забезпечувала б комплексну автоматизацію процесу інвентаризації БФ. Для цієї системи важливими характеристиками були: наявність шаблонів для друку звітів, можливість встановлення прав доступу до інформації для різних груп користувачів, а також зручний і зрозумілий інтерфейс [5].

2 ПРОЕКТНА ЧАСТИНА

2.1 Постановка задачі

На сучасному етапі становлення інформаційного суспільства однією з основних тенденцій розвитку електронних інформаційних ресурсів та бібліотечно-інформаційних технологій є розробка бази даних бібліотеки і створення електронних бібліотек, що дозволяє полегшити роботу працівників.

Основним завданням дипломного проекту є розробка та створення бази даних для бібліотечного фонду, яка має на меті забезпечити ефективну інвентаризацію на підприємстві. Ця база даних також спрямована на виявлення можливих помилок, які можуть виникнути під час шифрування документів, оформлення видачі книг користувачам, а також на виявлення заборгованості користувачів перед бібліотекою.

В результаті використання ручної роботи, працівники бібліотеки затрачують значну кількість нераціонального часу і часто допускають помилки. Однак, створення інформаційної системи дозволить автоматизувати процес інвентаризації бібліотечного фонду, що призведе до швидкого та ефективного оброблення даних. Це в суттєвій мірі зменшить час, що витрачається працівниками бібліотеки [6].

Вхідними даними буде служити інформації про бібліотечний фонд, працівників бібліотеки та користувачів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати та узагальнити вітчизняний і зарубіжний досвід із електронного бібліотечно-інформаційного забезпечення;
- вивчити етапи інвентаризації бібліотечних фондів;
- спроектувати базу даних, враховуючи методи нормалізації;
- створити інформаційну систему для автоматизації процесу інвентаризації.

2.2 Проектування бази даних

База даних є необхідною компонентою у сфері створення системи автоматизації інвентаризації бібліотечного фонду. Вона надасть можливість детально вести облік всього наявного фонду бібліотеки.

Внаслідок реалізації дипломного проекту буде створена інформаційна база даних, яка буде включати всю необхідну інформацію про бібліотечний фонд.

Процес реєстрації нового користувача виконується шляхом використання бази даних, тоді як перевірка існуючих користувачів відбувається під час процедури авторизації. Крім того, зберігання та виведення повної інформації про бібліотечний фонд також залежить від бази даних [7].

Після проведення аналізу предметної області, будуть ідентифіковані ключові інформаційні елементи (сутності), що необхідні для розробки бази даних. Для виявлення сутностей можна використовувати різні методи, одним із яких є вивчення аналогічних програм і проведення аналізу предметної області. Кожна ідентифікована сутність отримує зрозуміле для користувача ім'я, яке відповідає її значенню та контексту. Інформацію про типи сутностей бази даних, їх опис, псевдоніми та особливості використання представимо у вигляді таблиці 2.1.

Таблиця 2.1 – Опис сутностей бази даних «Biblioteka»

Ім'я сутності	Опис	Псевдоніми	Особливості виконання
Газети	Описується інформація про газети, які знаходяться в бібліотечному фонді	Газети	Кожна газета має свій унікальний номер, який закріплений за працівником
Книги	Описується інформація про книги, які знаходяться в бібліотечному фонді	Книги	Кожна книга має свій унікальний номер, який закріплений за працівником
Журнали	Описується інформація про журнали, які знаходяться в бібліотечному фонді.	Журнали	Кожний журнал має свій унікальний номер, який закріплений за працівником
Інвентарна карта	Описує весь бібліотечний фонд	Інвентарна_карта	Введеться облік усього бібліотечного фонду
Працівники	Описує інформацію про працівників бібліотеки	Працівники	Кожен працівник має свій особистий запис
Користувачі	Зберігаються дані про користувачів	users	Кожен користувач має свій персональний кабінет і свій тариф

Опис атрибутів сутностей бази даних «Biblioteka» знаходиться в таблиці 2.2.

Таблиця 2.2 – Опис атрибутів сутностей бази даних «Biblioteka»

Назва сутності	Атрибут	Опис	Домен	Обмеження	Значення за замовчуванням	Псевдонім	Значення Null
1	2	3	4	5	6	7	8
Працівники	Код працівника	Код працівника	Ціле число	Первинний ключ	-	id	Ні
	ПІБ	ПІБ працівника	Стрічка	255	-	ПІБ	Так

	Посада	Посада працівника	Стрічка	255	-	Посада	Так
--	--------	-------------------	---------	-----	---	--------	-----

Продовження таблиці 2.2

1	2	3	4	5	6	7	8
	Адреса	Адреса проживання працівника	Стрічка	255	-	Адреса	Так
	Телефон	Мобільний номер працівника	Стрічка	255	-	Телефон	Так
Інвентарна_карта	Код інвентарної карти	Код інвентарної карти	Ціле число	Первинний ключ	-	id	Ні
	Інвентарний номер	Інвентарний номер	Стрічка	255	-	Інвентарний номер	Так
	Назва книги	Назва книги	Стрічка	255	-	Назва	Так
	Ціна відшкодування	Ціна відшкодування	Стрічка	255	-	Ціна_відшкодування	Так
	Відповідальний	Відповідальний	Стрічка	255	-	Відповідальний	Так
Книги	Код книги	Унікальний ідентифікатор	Ціле число	Первинний ключ	-	id	Ні
	Назва книги	Назва книги	Стрічка	255	-	Назва	Так
	Автор книги	Автор книги	Стрічка	255	-	Автор	Так
	Видавництво	Видавництво газети	Стрічка	255	-	Видавництво	Так
	Рік видання	Рік видання	Стрічка	255	-	Рік_видання	Так
	Стан	Стан книги	Стрічка	255	-	Стан	Так
	Інвентарний номер	Унікальний номер книги	Стрічка	255	-	Інвентарний_номер	Так
Книги	Ціна відшкодування	Ціна відшкодування втраченої чи пошкоджен	Стрічка	255	-	Ціна відшкодування	Так

		ої книги					
Журнали	Код журналу	Унікальний ідентифікатор	Ціле число	Первинний ключ	-	id	Ні
	Назва журналу	Назва журналу	Стрічка	255	-	Назва	Так

Продовження таблиці 2.2

1	2	3	4	5	6	7	8
	Автор журналу	Автор журналу	Стрічка		-	Автор	Так
	Видавництво	Видавництво газети	Стрічка	255	-	Видавництво	Так
	Рік видання	Рік видання	Стрічка	255	-	Рік_видання	Так
Журнали	Стан	Стан журналу	Стрічка	255	-	Стан	Так
	Інвентарний номер	Унікальний номер журналу	Стрічка	255	-	Інвентарний_номер	Так
	Ціна відшкодування	Ціна відшкодування втраченої чи пошкодженого журналу	Стрічка	255	-	Ціна відшкодування	Так
Користувачі	Код користувача	Унікальний ідентифікатор	Ціле число	Первинний ключ	—	id	Ні
	Ім'я	Ім'я користувача	Стрічка	50		Login	Так
	Пароль	Пароль користувача	Стрічка	50		Password	Так
	Категорія користувача	Статус користувача admin або user	Стрічка	25		Kategor	Так
Газети	Код газети	Унікальний ідентифікатор	Ціле число	Первинний ключ	—	id	Ні
	Назва газети	Назва газети	Стрічка	255		Назва	Так
	Автор газети	Автор газети	Стрічка	255		Автор	Так

	Видавництво	Видавництво газети	Стрічка	255		Видавництво	Так
	Рік видання	Рік видання	Стрічка	255		Рік_видання	Так

Продовження таблиці 2.2

1	2	3	4	5	6	7	8
	Стан	Стан газети	Стрічка	255	-	Стан	Так
	Інвентарний номер	Унікальний номер газети	Стрічка	255	-	Інвентарний_номер	Так
	Ціна відшкодування	Ціна відшкодування втраченої чи газети	Стрічка	255	-	Ціна_відшкодування	Так

Під час розробки бази даних необхідно побудувати ER-діаграму, для чого спочатку потрібно визначити первинні ключі сутностей та типи зв'язків, а також їх кардинальність. Кожній сутності призначається первинний ключ. Проаналізувавши предметну область встановлено зв'язки між сутностями (див. табл. 1.3).

Таблиця 2.3 – Опис зв'язків між сутностями бази даних «Biblioteka»

Імя сутності	Назва зв'язку	Імя сутності	Кардинальність
Інвентарна_карта	Має	Книги	1:N
Інвентарна_карта	Має	Журнали	1:N
Інвентарна_карта	Має	Газети	1:N
Інвентарна_карта	Має	Працівники	1:N

Далі ми повинні вибрати модель для проектування бази даних, і ми використаємо реляційну модель даних для створення логічної схеми даних.

Спочатку розглянемо атрибути сутностей бази даних «Bibloteka». У концептуальній моделі даних багатозначні атрибути відсутні.

Проведемо аналіз зв'язків між сутностями бази даних "Biblioteka" з метою виявлення зв'язків "багато-до-багатьох", складних та рекурсивних зв'язків. Оскільки на попередньому етапі ми визначили зв'язки типу "один-до-багатьох" в базі даних "Biblioteka", тепер концептуальна модель перетворюється на логічну схему, де кожна сутність стає відношенням. Зв'язки між відношеннями моделюються за допомогою первинних та зовнішніх ключів.

Нормалізація - це процес, який спрямований на усунення дублювання даних та протиріч у логічній моделі даних. Основна ціль нормалізації полягає у створенні ефективної бази даних, де зайві або непотрібні дані виключаються.

Першим кроком буде аналіз відношень бази даних "Biblioteka" та їх нормалізація до третьої нормальної форми. Відношення вважається нормалізованим до першої нормальної форми, якщо кожен атрибут відношення є атомарним та не містить повторюваних груп даних.

Розглянемо створені відношення: «Книги», «Журнали», «Газети», «Працівники» та «Інвентарна_карта». Серед них лише у відношенні «Працівники» є багатозначний атрибут «ПІБ», який необхідно розділити на три окремих атрибути «Прізвище», «Ім'я», «По_батькові». Усі інші відношення не містять багатозначних атрибутів, тому вони будуть нормалізованими до першої нормальної форми.

Відношення знаходиться у другій нормальній формі, якщо воно знаходиться у першій нормальній формі і містить функціональні залежності між описовими атрибутами та ключем. Проаналізуємо кожне з відношень бази даних «Biblioteka». Відношення містять простий первинний ключ і їх описові атрибути функціонально залежать від ключа. Тому можна вважати, що кожне відношення бази даних «Biblioteka» знаходиться в другій нормальній формі.

Кожне відношення бази даних буде вважатися нормалізованим до третьої нормальної форми, оскільки воно вже відповідає вимогам першої та другої нормальних форм, і між атрибутами не існує транзитивних функціональних залежностей. Після аналізу всіх відношень в базі даних "Biblioteka", було виявлено відсутність таких транзитивних залежностей. Тому можна стверджувати, що всі відношення бази даних "Biblioteka" відповідають третій нормальній формі.

Враховуючи аналіз предметної області, створимо структуру таблиць за допомогою SQL-запитів наведених нижче .

Лістинг 2.1 Текст SQL-запиту для створення таблиці «users»

```
CREATE TABLE [dbo].[users](
[id] int NOT NULL PRIMARY KEY,
[Login] varchar (50) NULL,
[Password] varchar (50) NULL,
[Kategor] varchar (50) NULL)
```

Лістинг 2.2 Текст SQL-запиту для створення таблиці «Інвентарна_картка»

```
CREATE TABLE [dbo].[Інвентарна_картка](
[id] int NOT NULL PRIMARY KEY,
[Інвентарна_картка] varchar (50) NULL,
[Назва] varchar (50) NULL,
[Ціна_відшкодування] varchar (50) NULL,
[Відповідальний] varchar (50) NULL)
```

Лістинг 2.3 Текст SQL-запиту для створення таблиці «Працівники»

```
CREATE TABLE [dbo].[Працівники](
[id] int NOT NULL PRIMARY KEY,
[ПІБ] varchar (50) NULL,
[Посада] varchar (50) NULL,
[Адреса] varchar (50) NULL,
[Телефон] varchar (50) NULL)
```

Лістинг 2.4 Текст SQL-запиту для створення таблиці «Газети»

```
CREATE TABLE [dbo].[Газети](
[id] int NOT NULL PRIMARY KEY,
[Назва] varchar (50) NULL,
[Автор] varchar (50) NULL,
[Видавництво] varchar (50) NULL,
[Рік_видання] varchar (50) NULL,
[Стан] varchar (50) NULL,
[Інвентарний_номер] varchar (50) NULL,
[Ціна_відшкодування] varchar (50) NULL)
```

Лістинг 2.5 Текст SQL-запиту для створення таблиці «Журнали»

```
CREATE TABLE [dbo].[Журнали](
[id] int NOT NULL PRIMARY KEY,
[Назва] varchar (50) NULL,
[Автор] varchar (50) NULL,
[Видавництво] varchar (50) NULL,
[Рік_видання] varchar (50) NULL,
```

Продовження лістингу 2.5

```
[Стан] varchar (50) NULL,  
[Інвентарний_номер] varchar (50) NULL,  
[Ціна_відшкодування] varchar (50) NULL)
```

Лістинг 2.6 Текст SQL-запиту для створення таблиці «Книги»

```
CREATE TABLE [dbo].[Книги](  
[id] int NOT NULL PRIMARY KEY,  
[Назва] varchar (50) NULL,  
[Автор] varchar (50) NULL,  
[Видавництво] varchar (50) NULL,  
[Рік_видання] varchar (50) NULL,  
[Стан] varchar (50) NULL,  
[Інвентарний_номер] varchar (50) NULL,  
[Ціна_відшкодування] varchar (50) NULL)
```

2.3 Проектування інформаційної системи

Інформаційно-логічна модель відображає структуру даних предметної області у формі колекції інформаційних об'єктів та зв'язків між ними.

Створення UML-моделі є важливим етапом підготовки до розробки бази даних та програмного забезпечення. Ця модель дозволяє представити систему у вигляді графічних діаграм із стандартизованими символами та нотацією UML. Вона надає можливість легко перевести модель в програмний код, що спрощує процес розробки та забезпечує зрозумілість системи для розробників інших компонентів [10].

Для того, щоб спроектувати інформаційну систему, потрібно побудувати наступні діаграми:

- діаграма використання (Use case);
- діаграма класів (Class diagram);
- діаграма послідовності (Sequence diagram).

Діаграма використання є графічним засобом для представлення конкретних сценаріїв використання системи. Вона включає акторів, які взаємодіють з системою, а також елементи, що відображають можливі дії та взаємодії між акторами та системою. Ця діаграма надає зрозуміле візуальне відображення варіантів використання та відношень між елементами системи, спрощуючи аналіз та комунікацію між учасниками розробки. Компоненти діаграми варіантів використання можуть бути вміщені в прямокутник, що представляє систему в цілому. Додатково до графічних елементів, діаграма може містити текстові пояснення, що розкривають сенс або семантику окремих компонентів. Це допомагає уточнити і пояснити деталі та взаємозв'язки, що представлені на діаграмі варіантів використання.

Відношення між прецедентами «Перегляд інформації» і «Формування звітів» – буде відношення включення, оскільки коли актор «Бібліотекар» переглядає інформацію, то не завжди при цьому він робить вибірку інформації.

Відношення між прецедентами «Додавання даних», «Редагування даних» та «Видалення даних» – буде відношення розширення, оскільки, коли актор «Адміністратор» виконує додавання інформації, тоді він не може одночасно редагувати її чи видалити. Діаграма використання для АІС БІБЛІОТЕКА наведена на рисунку 2.1



Рисунок 2.1 – Діаграма використання для АІС БІБЛІОТЕКА

Діаграма класів – це статичне представлення структури моделі. Відображає статичні елементи, такі як: класи, типи даних, їх зміст та відношення.

Діаграма класів містить багато елементів, які взаємодіють між собою, щоб відображати декларативні знання про предметну область. Ці знання виражаються за допомогою базових понять мови UML, таких як класи, інтерфейси та їхні відношення та компоненти

Діаграма класів для АІС БІБЛІОТЕКА зображує функцію авторизації, при успішній авторизації користувач переходить на головну сторінку. Головна сторінка забезпечує можливість отримати доступ до бази даних та взаємодіяти з нею. База даних дозволяє виконувати операції додавання, редагування та видалення інформації.

Діаграма класів наведена на рисунку 2.2 (див. рис. 2.2).

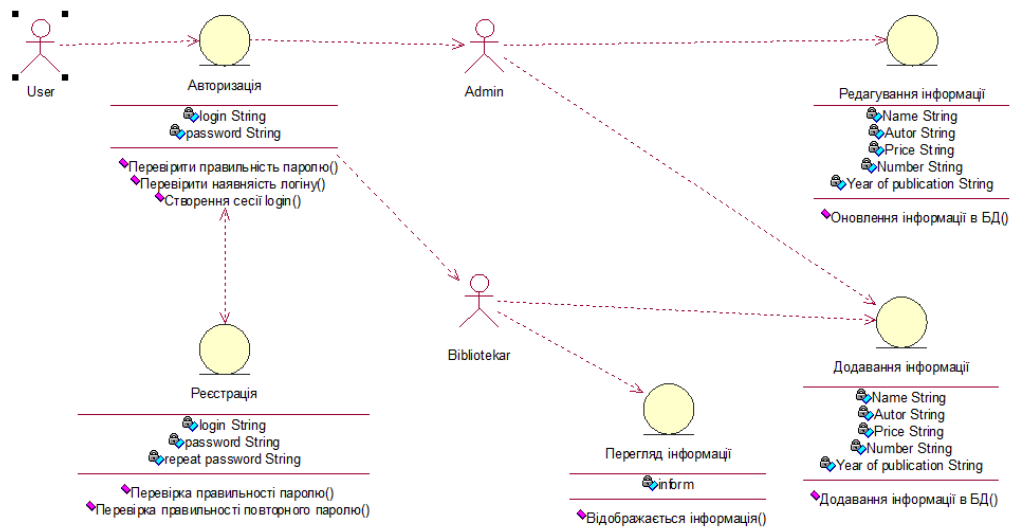


Рисунок 2.2 – Діаграма класів для АІС БІБЛІОТЕКА

Діаграма послідовності (sequence diagram) є основним способом відображення взаємодії об'єктів в часі.

Діаграма послідовностей демонструє паралельну роботу різних процесів або об'єктів у вигляді вертикальних ліній, які зображаються одночасно.

Повідомлення, які були надіслані, відображаються на діаграмі послідовностей у формі горизонтальних ліній, розташованих в порядку їх відправлення.

Діаграма послідовності показує механізм авторизації та роботу процесу для різних типів користувачів. Для успішного виконання функції, необхідно ввести необхідну інформацію, звернутись до бази даних та отримати результат, який потрібно порівняти з введеними даними. Після цього відбудеться перехід на головну форму і доступ буде наданий в залежності від типу користувача.

Діаграма послідовності наведена на рисунку 2.3 (див. рис. 2.3).

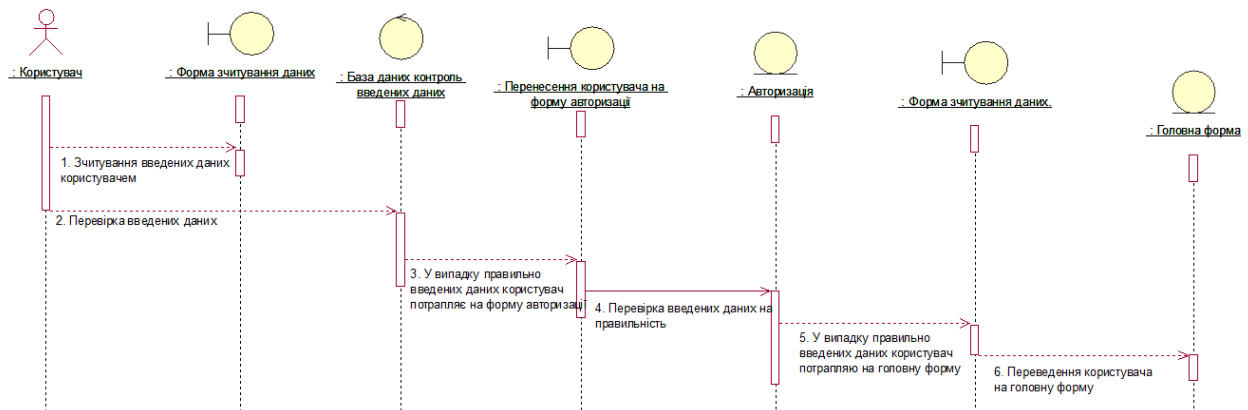


Рисунок 2.3 – Діаграма послідовності для АІС БІБЛІОТЕКА

На даному етапі було розроблено інформаційну систему для АІС "БІБЛІОТЕКА" шляхом створення UML-діаграм, зокрема, діаграм прецедентів, класів та послідовностей [11].

3 ПРАКТИЧНА ЧАСТИНА

3.1 Проектування інтерфейсу користувача

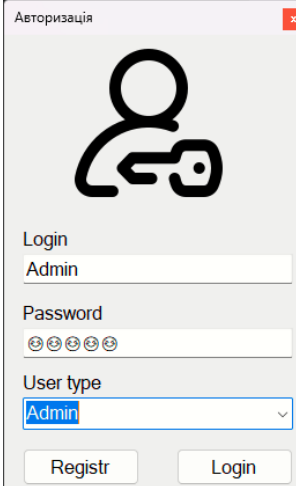
Основним завданням дипломного проекту є розробка інформаційної системи інвентаризації бібліотечних фондів. Для досягнення цієї мети, мною буде проведений детальний аналіз предметної області, вивчена робота підприємства та розроблена відповідна база даних.

Для ефективного проектування бази даних для інвентаризації бібліотечних фондів необхідно дотримуватися основних правил та вимог, що стосуються цього процесу. Це включає ретельне вивчення правил і процедур, пов'язаних з інвентаризацією, ознайомлення з відповідною документацією підприємства та врахування методів нормалізації та обмежень цілісності даних.


При розробці інтерфейсу необхідно пам'ятати, що:

- інтерфейс – найважливіша частина проекту яка впливає на характер рішень, які приймає користувач.
- інтерфейс має бути гнучким, та інтуїтивно зрозумілим для користувача.

Перед початком роботи з програмою користувача вітає форма авторизації (див. Рис 3.1).



Авторизація



Login
Admin

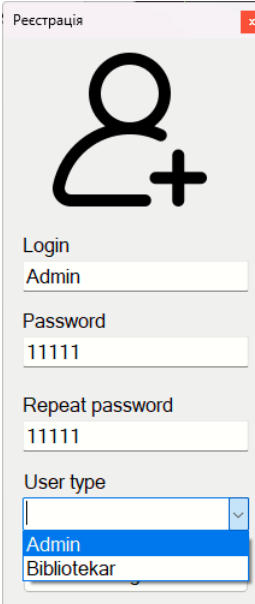
Password
●●●●●

User type
Admin

Registr Login

Рисунок 3.1 – Вигляд вікна авторизації

Якщо користувач використовує програму вперше, йому потрібно натиснути кнопку "Registr", після чого відкриється вікно, де користувачу будуть запропоновані поля для введення необхідних реєстраційних даних (див. рис 3.2).



The image shows a registration window titled "Регістрація" (Registration). At the top, there is a large icon of a person with a plus sign. Below the icon, there are four input fields: "Login" with the text "Admin", "Password" with "11111", and "Repeat password" with "11111". At the bottom, there is a "User type" dropdown menu with a blue highlight on the "Admin" option and "Bibliotekar" listed below it.

Рисунок 3.2 – Вигляд вікна реєстрації

Після успішної авторизації користувачу відкривається головна форма програми, яка надає доступ до основного функціоналу. (див. рис 3.3).

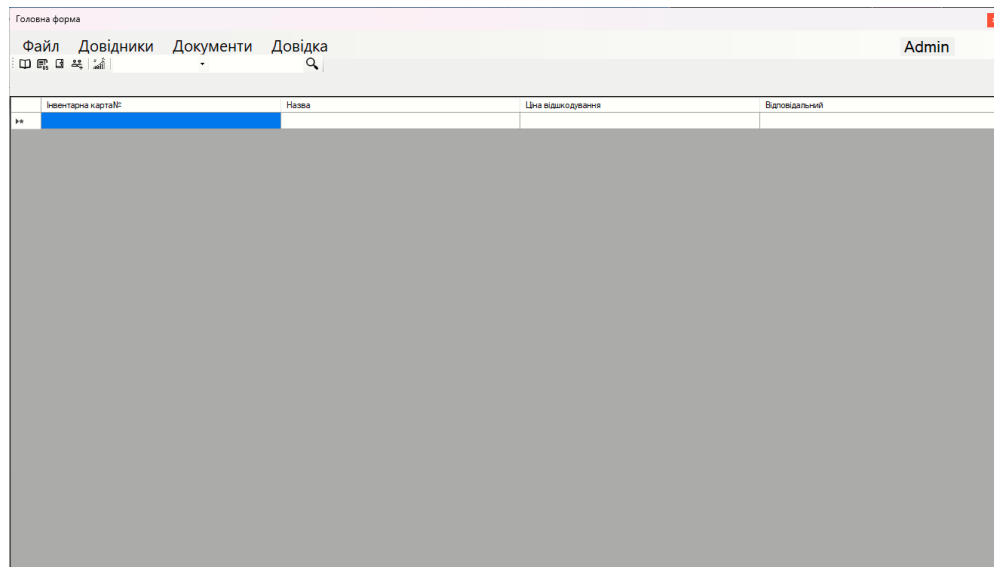


Рисунок 3.3 – Головна форма програми

Головна форма надає користувачу доступ до різноманітних функціональних можливостей, пропонуючи широкий спектр різноманітних опцій та інструментів:

- додавання нових даних про бібліотечний фонд;
- редагування даних про бібліотечний фонд;
- видалення даних про бібліотечний фонд;
- пошук необхідних даних за різними критеріями;
- експорт даних в Microsoft Word, Excel.

Щоб внести нові дані про бібліотечний фонд, потрібно перейти на відповідну вкладку у головному меню (див. рис 3.4), після цього відкривається вікно, де можна переглянути інформацію про наявні в бібліотеці книги (див. рис 3.5).

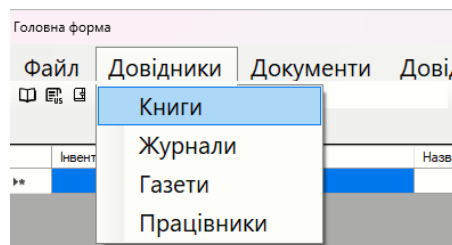
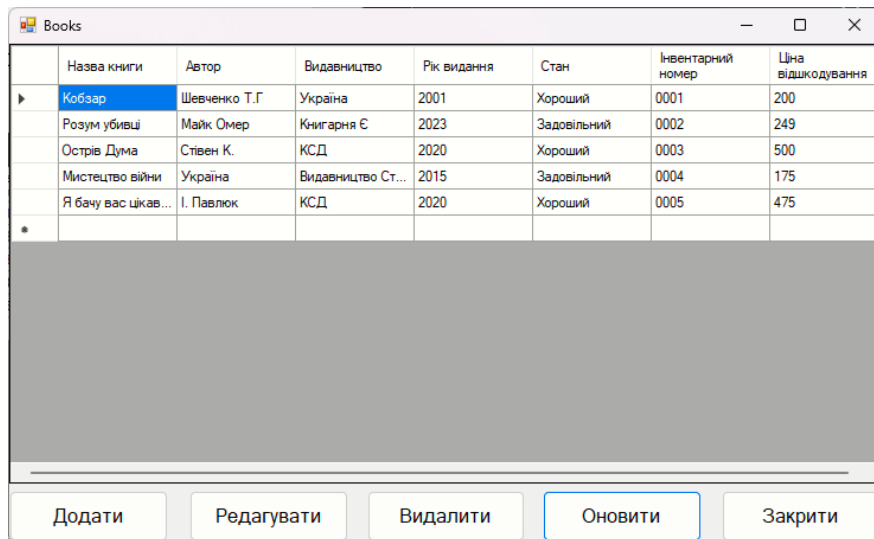


Рисунок 3.4 – Вкладки головного меню для додавання інформації]

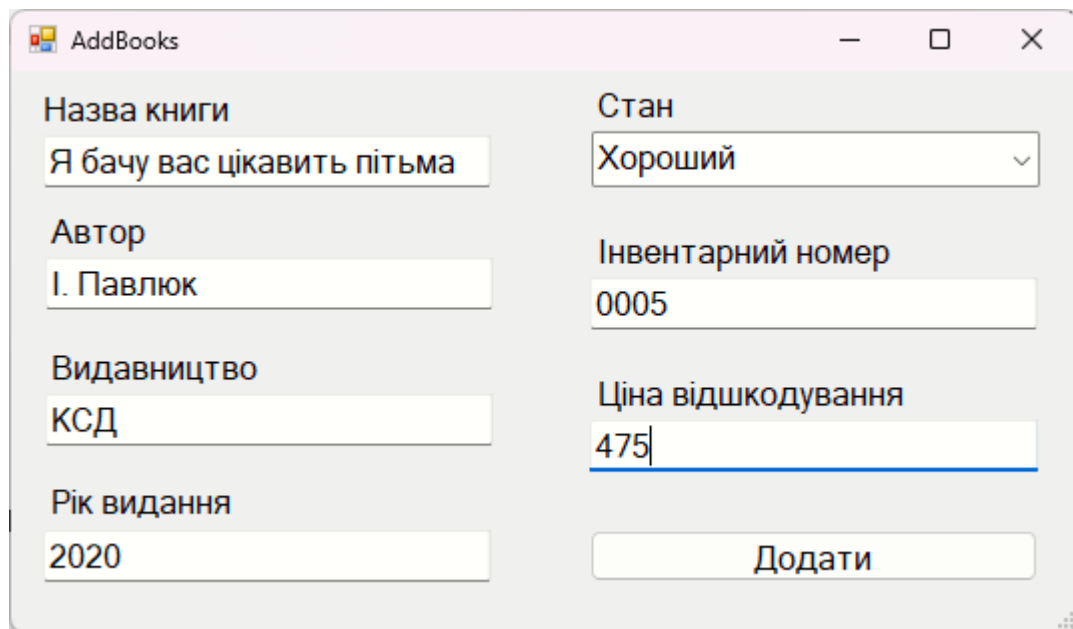


	Назва книги	Автор	Видавництво	Рік видання	Стан	Інвентарний номер	Ціна відшкодування
▶	Кобзар	Шевченко Т.Г	Україна	2001	Хороший	0001	200
	Розум убивці	Майк Омер	Книгарня Є	2023	Задовільний	0002	249
	Острів Дума	Стівен К.	КСД	2020	Хороший	0003	500
	Мистецтво війни	Україна	Видавництво Ст...	2015	Задовільний	0004	175
	Я бачу вас цікав...	І. Павлюк	КСД	2020	Хороший	0005	475
*							

Buttons: Додати, Редагувати, Видалити, Оновити, Закрити

Рисунок 3.5 – Форма для перегляду інформації бібліотечного фонду

Для того, щоб користувач додав нову інформацію про книгу, необхідно виконати наступні кроки: спочатку натиснути кнопку "Додати", що призведе до відкриття нової форми. На цій формі користувач повинен ввести всю необхідну інформацію про книгу, після чого слід натиснути кнопку "Додати" (див. рис 3.6).



AddBooks

Назва книги: Я бачу вас цікавить пітьма

Автор: І. Павлюк

Видавництво: КСД

Рік видання: 2020

Стан: Хороший

Інвентарний номер: 0005

Ціна відшкодування: 475

Додати

Рисунок 3.6 – Форма додавання нової книги

Крім того, користувач має можливість створити інвентарну картку. Для цього необхідно перейти на вкладку "Документи" у головному меню, обрати опцію "Інвентарна картка" і натиснути кнопку "Створити" (див. Рис 3.6). Після цього з'явиться форма, де користувач може переглянути список існуючих інвентарних карток (див. рис 3.7).

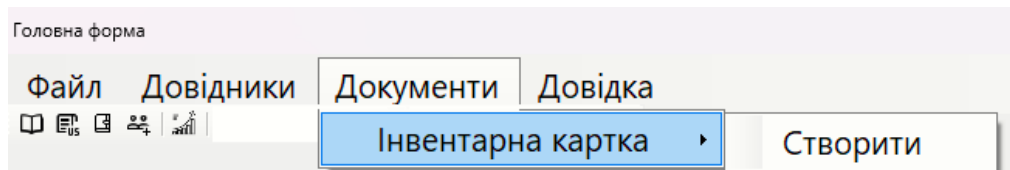


Рисунок 3.6 – Вкладка головного меню для створення інвентарної картки

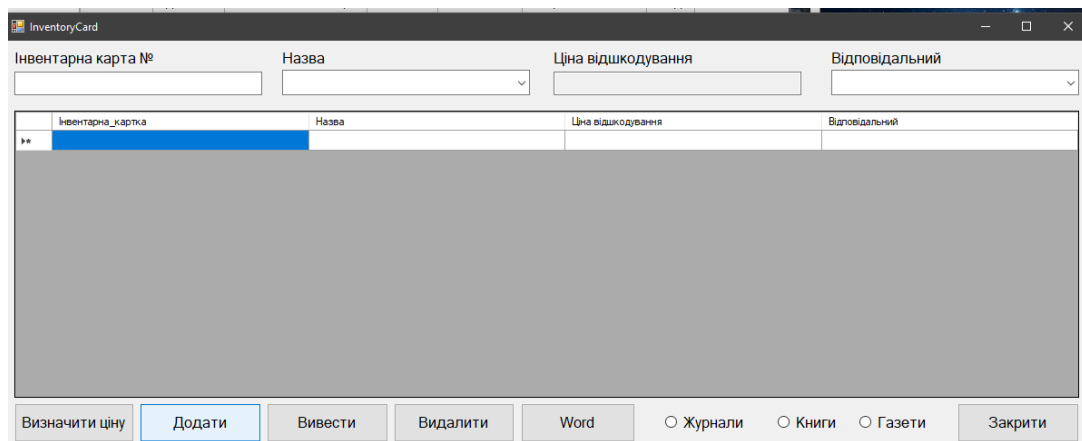


Рисунок 3.7 – Форма для перегляду інформації про інвентарні картки

Для створення нової інвентарної картки, користувач повинен натиснути кнопку "Додати". Це відкриє нову форму, де буде потрібно ввести всю необхідну інформацію (див. рис 3.8).

Рисунок 3.8 – Форма для додавання інформації про інвентарні картки

Для зручного та швидкого доступу до різних операцій у програмі розроблено спеціальне меню швидкого доступу (див. рис 3.9).

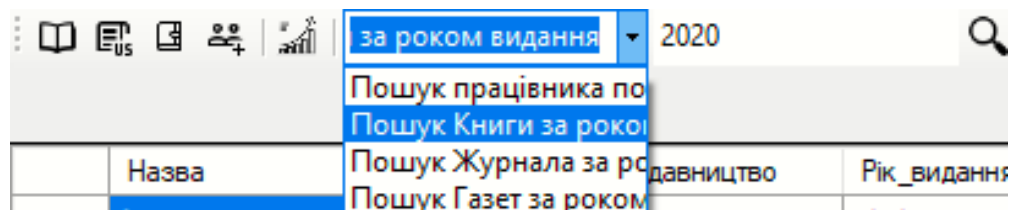


Рисунок 3.9 – Меню швидкого доступу

Користувач має можливість здійснювати пошук необхідної інформації за заданими критеріями відповідно до своїх потреб. Для цього в меню швидкого доступу користувач може вибрати відповідні пошукові критерії, ввести умову та натиснути кнопку "Пошук" (див. рис 3.10).

Головна форма

Файл Довідники **Документи** Довідка

Пошук Книги за роком 2020

	Назва	Автор	Видавництво	Рік_видання	Стан	Інвентарний_номє	Ціна_відшкодуван
▶	Острів Дума	Стівен К.	КСД	2020	Хороший	0003	500
	Я бачу вас цікав...	І. Павлюк	КСД	2020	Хороший	0005	475
*							

Рисунок 3.10 – Результат пошуку

Користувач має можливість експортувати результати пошуку в форматі Excel або Word, якщо це відповідає його потребам. Для здійснення експорту необхідно клацнути правою кнопкою миші на таблиці з результатами, а потім з'явиться контекстне меню, з якого користувач може обрати тип експорту, який йому потрібен (див. рис 3.11).

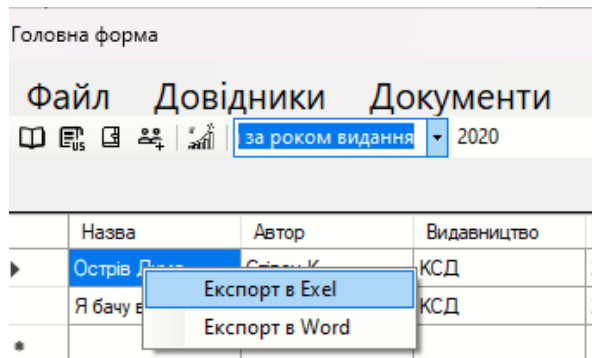


Рисунок 3.11 – Експорт даних

Після вибору вкладки "Довідка" у головному меню (див. рис. 3.12) та натискання пункту "Довідка", користувач отримає доступ до довідника, де він зможе знайти необхідну інформацію про використання програми. Цей довідник надасть користувачеві інструкції та пояснення щодо різних функцій та можливостей програми (див. рис 3.13). При виборі пункту "Про програму" у меню "Довідка", користувачу буде надана інформація про програму. Ця інформація може містити загальний опис програми, версію, автора або команду розробників, а також інші важливі деталі, які стосуються програми (див. рис 3.13).

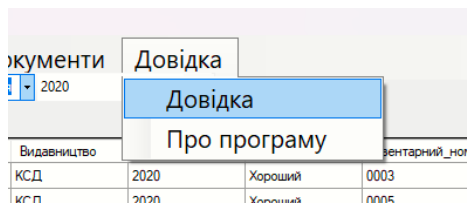


Рисунок 3.11 – Вкладки головного меню для відкриття довідки

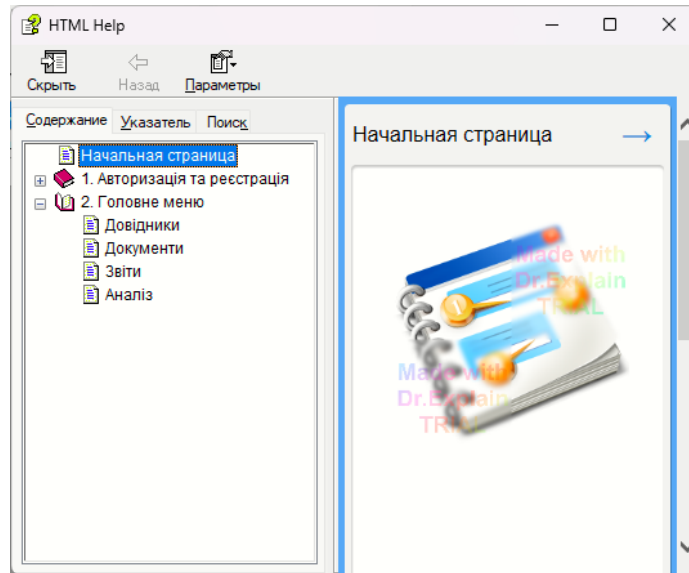


Рисунок 3.12 – Довідник

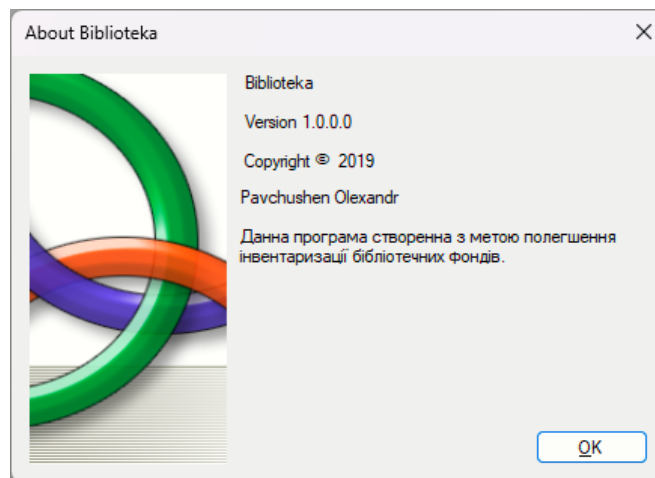


Рисунок 3.13 – Форма інформації про програму

3.2 Опис програмних модулів

На основі складеного технічного завдання та вимог користувачів ми визначимо та надамо опис основних програмних файлів.

Програма розроблена за допомогою: візуального середовища Visual Studio 2017 мови програмування C# та Transact SQL у системі керування базами даних SQL Server [13]. Для розробки проекту було обрано фреймворк .NET Framework як надбудову для Microsoft Visual Studio 2017. Microsoft .NET – програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і веб-застосунків. Ця платформа пропонує кілька готових API-інтерфейсів для розробки веб додатків (ASP.NET), десктопних додатків (WPF), веб аплетів (Silverlight), роботи з базами даних (ADO.NET) тощо. Також .NET Framework і дозволяє звертатися до різноманітних джерел даних, до яких необхідно спочатку підключитися [12].

Для отримання даних з бази використовується об'єкт SqlDataReader, за допомогою якого можна зчитувати записи бази даних. Крім цього в проекті використовуються класи SqlDataAdapter і DataSet. DataSet – це сховище даних, з якими можна працювати незалежно від наявності підключення, а SqlDataAdapter заповнює DataSet даними з БД. Для отримання даних через об'єкт SqlDataAdapter необхідно організувати підключення до БД і виконати команду SELECT. Отримавши дані в DataSet, можна виконувати з ними різні операції: видаляти, змінювати, додавати нові записи. Компонента DataSet дозволяє працювати з базою за допомогою стандартних класів бібліотеки, інструментами класів C# та мови SQL [14].

Опис основних програмних модулів наведено у таблиці 3.1.

Таблиця 3.1 - Опис програмних модулів

№	Назва програмного модуля	Короткий опис програмного модуля
1	Login.cs	Форма для авторизації користувача в програмі.
2	Registr.cs	Форма для реєстрації користувача в програмі.
3	About.cs	Форма з усією інформацією про проект.
4	MainForms.cs	Головна форма програми де користувач може вибрати наступну лію яку він буде виконувати.

5	InventoryCard.cs	Форма яка відображає інвентарну картку.
6	Logs.cs	Форма яка відображає журнали які є в наявності бібліотеки.
7	Books.cs	Форма яка відображає книги які є в наявності бібліотеки.
8	Newspapers.cs	Форма яка відображає газети які є в наявності бібліотеки.
9	Worker.cs	Форма яка відображає список усіх працівників бібліотеки.
10	AddBooks.cs	Форма для додавання нової книги в бібліотечний фонд.
11	AddInventoryCard.cs	Форма для створення інвентарної картки.
12	AddLogs.cs	Форма для додавання нового журналу в бібліотечний фонд.
13	AddNewspapers.cs	Форма для додавання нової газети в бібліотечний фонд.
14	AddWorker.cs	Форма для додавання нового працівника в список працівників бібліотеки.

Описані, в таблиці 3.1 Програмні модулі відображають структуру розробленого програмного засобу та надають інформацію про їхнє призначення, способи взаємодії та можливості використання у рамках проектної структури [15].

3.3 Опис результатів тестування

Тестування програмного забезпечення (ПЗ) є важливою складовою процесу його розробки і включає в себе ряд заходів, спрямованих на досягнення високої якості програмного продукту.

Тестування включає в себе кероване експериментування з програмним продуктом з метою виявлення помилок, тобто неточностей, що можуть бути допущені розробниками ПЗ, а також пошук і усунення дефектів у програмному забезпеченні.

Оскільки число можливих тестів навіть для невеликих програмних компонентів є практично нескінченним, стратегія тестування полягає в умілому плануванні та виконанні тестів з урахуванням обмежень у часі та ресурсах. Таким чином, програмне забезпечення піддається тестуванню за допомогою стандартних виконань програми з метою виявлення багів (помилки) та інших дефектів.

Тестування проникає через увесь життєвий цикл програмного забезпечення, починаючи з фази проектування і триваючи протягом тривалого періоду експлуатації. Воно нерозривно пов'язане з управлінням вимогами та змінами, оскільки його ціль полягає в переконанні, що програма відповідає встановленим вимогам [16].

Розглянемо такі види тестування як функціональне і ручне, результати роботи яких наведено у таблиці 3.2 та 3.5.

Функціональне тестування полягає у створенні тест-кейсів для перевірки усіх функцій ПЗ. Набір тест-кейсів для АІС «Biblioteka» наведено у таблиці 3.2.

Таблиця 3.2 – Функціональне тестування Test-case для АІС «Biblioteka»

ID	Тип	Опис	Очікувано	Реально	Pass/ Fail
1	2	3	4	5	6
01_All	positive	Перевірка відображення на різних розширеннях монітора	Усі компоненти знаходяться на своїх місцях, немає ніяких спотворень інтерфейсу	Усі компоненти знаходяться на своїх місцях, немає ніяких спотворень інтерфейсу	PASS
02_All	positive	Функція відображення паролю char *	При введенні паролю усі символи відображаються у вигляді *	При введенні паролю усі символи відображаються у вигляді *	PASS
03_All	positive	Функція переходу по компонентах клавішею Tab (function button Tab)	Повинен здійснюватися перехід курсора між полями для вводу та кнопками	При переході функція клавіші Tab працює у всіх випадках	PASS
04_All	negative	У поле логіну	Поле з помилкою	Поле з помилкою	PASS

		користувача введено невірний логін			
05_All	negative	У поле паролю користувача введено невірний пароль	Поле з помилкою	Поле з помилкою	PASS
06_All	Positive	У поле логіну і паролю введено вірні данні	Вхід в систему відповідно облікового запису	Вхід в систему відповідно до облікового запису	PASS
07_All	positive	Вхід під обліковим записом користувача	Доступні лише ті функції програми, які визначенні правами доступу	Доступні лише ті функції програми, які визначенні правами доступу	PASS

Продовження таблиці 3.2

1	2	3	4	5	6
08_All	positive	У поле «Пошук працівника по ПІБ» під час пошуку введенні некоректні дані	Поле з помилкою	Поле з помилкою	PASS
09_All	positive	У поле «Пошук Книги за роком видання» під час пошуку введенні некоректні дані	Поле з помилкою	Поле з помилкою	PASS
10_All	positive	У поле «Пошук Журнала за роком видання» під час пошуку введенні некоректні дані	Поле з помилкою	Поле з помилкою	PASS

Для перевірки коректного відображення сторінок написано тести, результати яких наведено у таблиці 3.3 та 3.4.

Таблиця 3.3 – Опис тест кейсу перевірка відображення сторінки

Дія	Очікуваний результат	Результат тесту
Відкрити сторінку «Авторизація»	<ul style="list-style-type: none"> – форма «Авторизація»; – назва форми – Авторизація; – на формі три поля для введення даних логін, пароль та тип користувача; – кнопка Вхід доступна; – кнопка Реєстрація доступна; 	PASS

Назва: Перевірка відображення сторінки.

Дія: Відкрити сторінку «Авторизація».

Перевірка: Перевірити, щоб сторінка, яка відображається, співпадала з сторінкою на рисунку 3.14 (див. рис. 3.14).

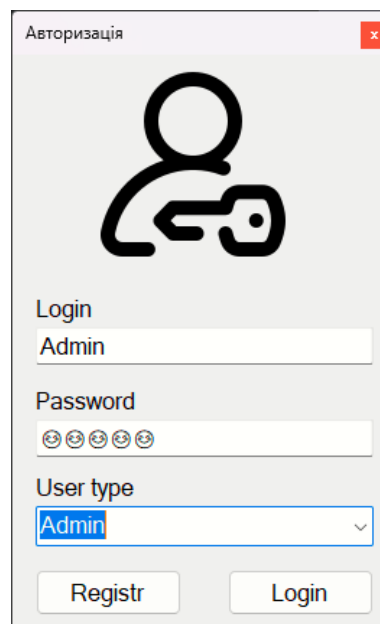


Рисунок 3.14 – Вікно «Авторизація»

Таблиця 3.4 – Опис тест кейсу перевірка відображення сторінки

Дія	Очікуваний результат	Результат тесту
Відкрити сторінку «Перегляд в режимі адміністратора»	<ul style="list-style-type: none"> – Форма «Перегляд в режимі адміністратора»; – Відображення типу користувача; – На формі є область 	PASS

	<p>для пошуку потрібної інформації</p> <p>У верхній частині форми є меню для перегляду або редагування допоміжних таблиць БД</p>	
--	--	--

Назва: Перевірка відображення сторінки.

Дія: Відкрити сторінку «Перегляд в режимі адміністратора».

Перевірка: Перевірити, щоб сторінка, яка відображається, співпадала з сторінкою, що на рисунку 3.15 (див. рис. 3.15).

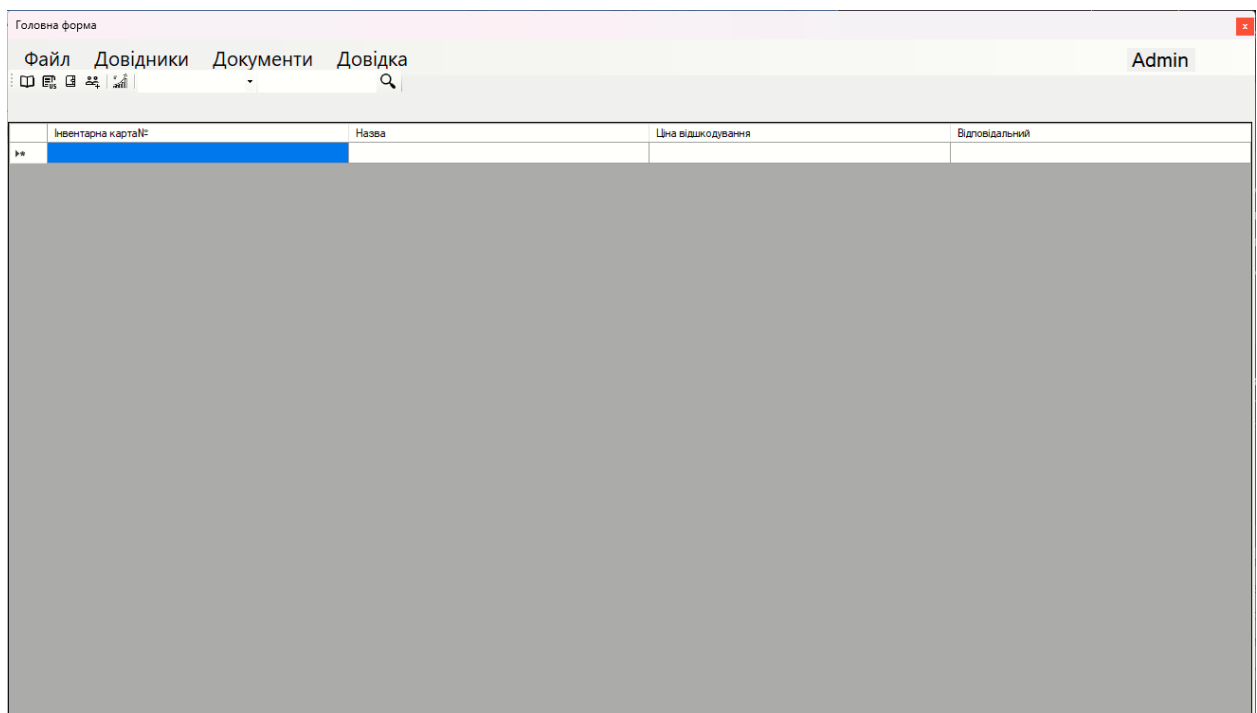


Рисунок 3.15 – Головне вікно програми

Під час тестування програми було виявлено недоліки, опис яких наведено у таблиці 3.5

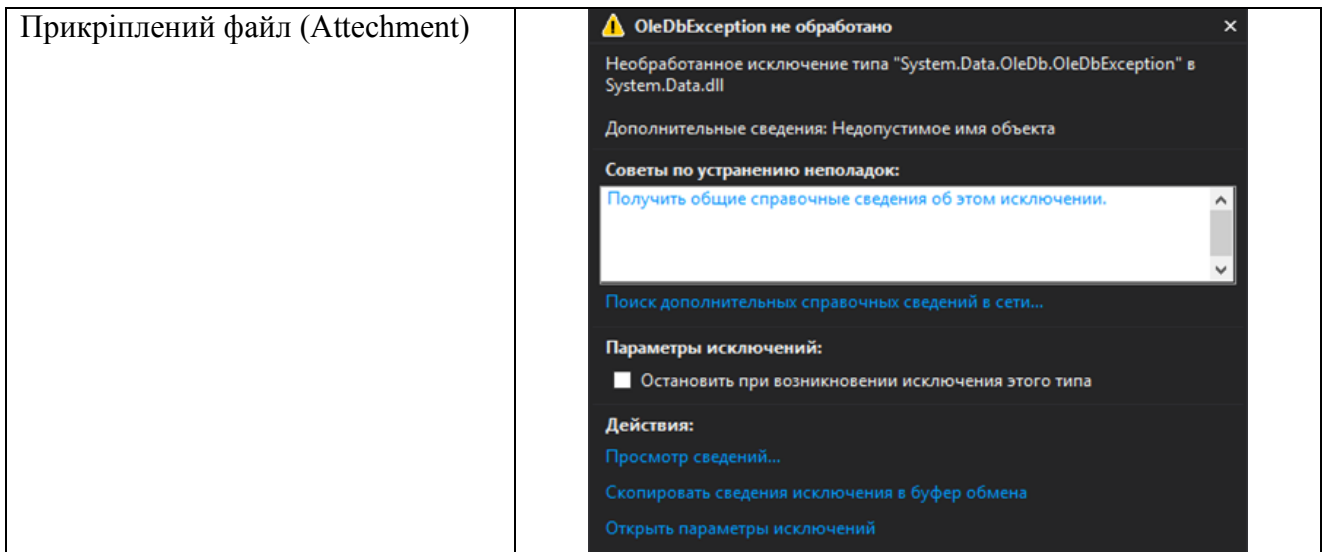
Таблиця 3.5 – Опис Bug-report для АІС «Biblioteka»

Автоматизована інформаційна система «Biblioteka»	
ID номер	Тест 2
Короткий опис (Summary)	При додаванні запису з пустими полями, відбудеться зависання програми

Проект (Project)	«Biblioteka»
Компонент додатку (Component)	Функціонал додавання запису
Номер версії (Version)	Версія при якій було виявлено баг 1.1
Серйозність (Severity)	S1 Блокуючий (Blocker)
Пріоритет (Priority)	P1 Високий (High)
Статус (Status)	Критичний
Автор (Author)	Павчишен Олександр

Продовження таблиці 3.5

1	2
Призначення для виправлення (Assigned To)	Павчишен Олександр
ОС	Операційна система Windows 10 build 1803, .NetFramework 4.5 Процесор: Intel Core i3-8130U 2.2GHz ОЗУ: 48 Гб Тип системи: 64-розрядна операційна система, процесор x64
Опис	
Кроки відтворення (Steps to Reproduce)	Запуск ПЗ Авторизація під користувачем «admin» В меню навігації вибрати «Робота з додаванням даних в БД» Заповнити не усі необхідні поля
Фактичний результат (Result)	Зависання програми
Очікуваний результат (Expected Result)	Виведення помилки і прохання заповнити усі необхідні поля
Додаток	



Для вирішення проблеми було використано, систему опрацювання помилок:

Лістинг 3.1 – Опрацювання помилки

```
if ((textBox1.Text.Equals("")) || (textBox2.Text.Equals("")) ||
(textBox3.Text.Equals("")) || (textBox4.Text.Equals("")))

```

Продовження лістингу 3.1

```
    MessageBox.Show("Ви не заповнили усі поля введення!", "Увага!",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation,
MessageBoxDefaultButton.Button1);
else form1.close();

```

Вона перевіряє, чи усі поля форми заповнені.

Отже, після проведеного набору тестів, можна сказати, що програм без помилок не буває, а проведення тестів на всіх етапах життєвого циклу ПЗ забезпечує якість додатку та покращує його роботу з кінцевим користувачем [17].

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Вимоги з безпеки праці для працівників бібліотеки

Відповідно до вимог типового положення про службу охорони праці, затвердженого наказом Державного комітету України з нагляду за охороною праці, в бібліотеках призначається відповідальна особа з питань охорони праці або створюється служба охорони праці [18].

У бібліотеці повинна бути розроблена інструкція з охорони праці для бібліотекара, що є нормативним документом та містить обов'язкові для дотримання працівником вимоги з охорони праці при виконання ним робіт, визначених його функціональними обов'язками на робочому місці.

До роботи допускаються бібліотекар:

- після вивчення інструкції та проходження медичного огляду;
- пройдено курс навчання за професією, а при необхідності стажування протягом 2 - 15 змін під керівництвом досвідчених працівників;
- проведено вступний та первинний (на робочому місці) інструктаж з питань охорони праці, пожежної безпеки;
- проведено інструктаж і перевірку знань з електробезпеки і отримано

II кваліфікаційну групу.

У процесі роботи бібліотекар проходить повторний інструктаж на робочому місці 1 раз в 6 місяців.

Робоче місце бібліотекара повинно бути забезпечене необхідним обладнанням (в тому числі зручним кріслом для відпочинку) та утримуватись у чистоті, при необхідності бібліотекару видається безоплатно спецодяг та спецвзуття.

При виявленні будь-яких несправностей обладнання, потрібно повідомити про це керівництво і до усунення неполадок до роботи не приступати.

Бібліотекар повинен знати і виконувати вимоги нормативних актів, інструкцій з охорони праці, правил користування засобами колективного та індивідуального захисту [19].

Забороняється приступати до роботи у стані алкогольного або наркотичного сп'яніння, а також в хворобливому або стомленому стані.

Бібліотекар повинен співпрацювати з керівництвом підприємства у справі створення безпечних і нешкідливих умов праці, повідомляти про порушення вимог охорони праці і пожежного режиму.

Кожен працівник бібліотеки повинен знати місце знаходження медичної аптечки, вміти користуватись медикаментами з аптечки і надавати першу долікарську допомогу. Бібліотекар повинен виконувати вимоги пожежної безпеки, знати розміщення та вміти користуватись первинними засобами пожежогасіння [19].

4.2 Загальні вимоги безпеки в бібліотеці

Приміщення бібліотек укомплектовується справними та перевіреними засобами пожежогасіння відповідно до вимог Типових норм належності вогнегасників, затверджених наказом Міністерства України з питань надзвичайних ситуацій. Засоби пожежогасіння розміщують біля входів, у вестибюлях, у коридорах, проходах та інших доступних місцях. У бібліотеках розробляються схеми евакуації користувачів і працівників, а також план заходів та дій адміністрації та працівників на випадок виникнення пожежі.

Усі світильники загального та місцевого освітлення обладнуються захисною арматурою. Використання світильників загального та місцевого освітлення без захисної арматури заборонено. У приміщеннях сховищ розміщення силових та освітлювальних розподільчих пристроїв не допускається. Освітлення в будівлях і приміщеннях бібліотек має забезпечувати

нормативний рівень освітлення на робочих місцях. У приміщеннях для каталогів, робочих кімнатах і кабінетах застосування лише місцевого освітлення не дозволяється. У приміщеннях книгосховищ проходи облаштовуються постійним штучним освітленням. Вмикання освітлення здійснюється пусковою арматурою, яка розташовується при вході в приміщення [20].

Працівники щодня перед вмиканням електрообладнання повинні візуально перевірити наявність та справність заземлювальних мереж.

Усі приміщення бібліотек повинні бути обладнані системами вентиляції і кондиціонування. Усі повітропроводи, повітряні канали, глушники та вентиляційне обладнання необхідно не рідше одного разу на півріччя очищувати від пилу. Повітрообмін у читальних залах, фондосховищах, реставраційних майстернях та робочих приміщеннях повинен забезпечувати параметри мікроклімату відповідно до Санітарних норм мікроклімату виробничих приміщень, установлених постановою Головного державного санітарного лікаря України. Режим роботи систем вентиляції повинен забезпечувати в залах для користувачів повітряне середовище в межах зазначених параметрів і підтримувати його протягом часу перебування користувачів.

Читальні зали в бібліотеках повинні забезпечувати комфортні і безпечні умови перебування в них і безперешкодну евакуацію у випадку пожежної або іншої небезпеки. Рух по проходах повинен бути безперешкодним. Розташування меблів в читальних залах не повинно перешкоджати руху працівників.

Каталожні шафи повинні розташовуватися вздовж стін приміщення. Висота шаф повинна бути не більше ніж 2 метри, щоб була можливість діставати верхні ящики, стоячи на підлозі. Вага картотечних ящиків не повинна перевищувати 7 кг. Для запобігання випаданню ящиків при їх максимальному витяганні вони обладнуються відповідними обмежувальними пристроями. Ширина проходів між каталожними шафами та встановленими робочими столами повинна бути не менше ніж 1,2 м. Робоче місце чергового

бібліотекара, адміністратора розташовують в такому місці, щоб була можливість спостерігати за усім приміщенням.

Приміщення бібліотек, де експлуатується комп'ютерна техніка, облаштовуються відповідно до вимог Правил охорони праці під час експлуатації електронно-обчислювальних машин, затверджених наказом Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду. Під час роботи з комп'ютерною технікою слід дотримуватися режимів праці і відпочинку відповідно до вимог відповідних державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [21].

ВИСНОВКИ

Під час виконання завдання індивідуального проекту було проведено аналіз предметної області, а також вдосконалено вміння використовувати візуальне середовище Visual Studio 2017 та мову програмування C#.

Було розроблено та реалізовано базу даних під назвою "Biblioteka", яка призначена для зберігання всієї необхідної інформації про бібліотечний фонд. Ця база даних була створена у середовищі Microsoft SQL Server Management Studio 2017, використовуючи SQL-запити.

У результаті було створено інформаційну систему, яка призначена для інвентаризації бібліотечних фондів з метою спрощення процесу обліку видань у бібліотеці. Застосування цієї системи значно підвищить ефективність та прискорить процес інвентаризації.

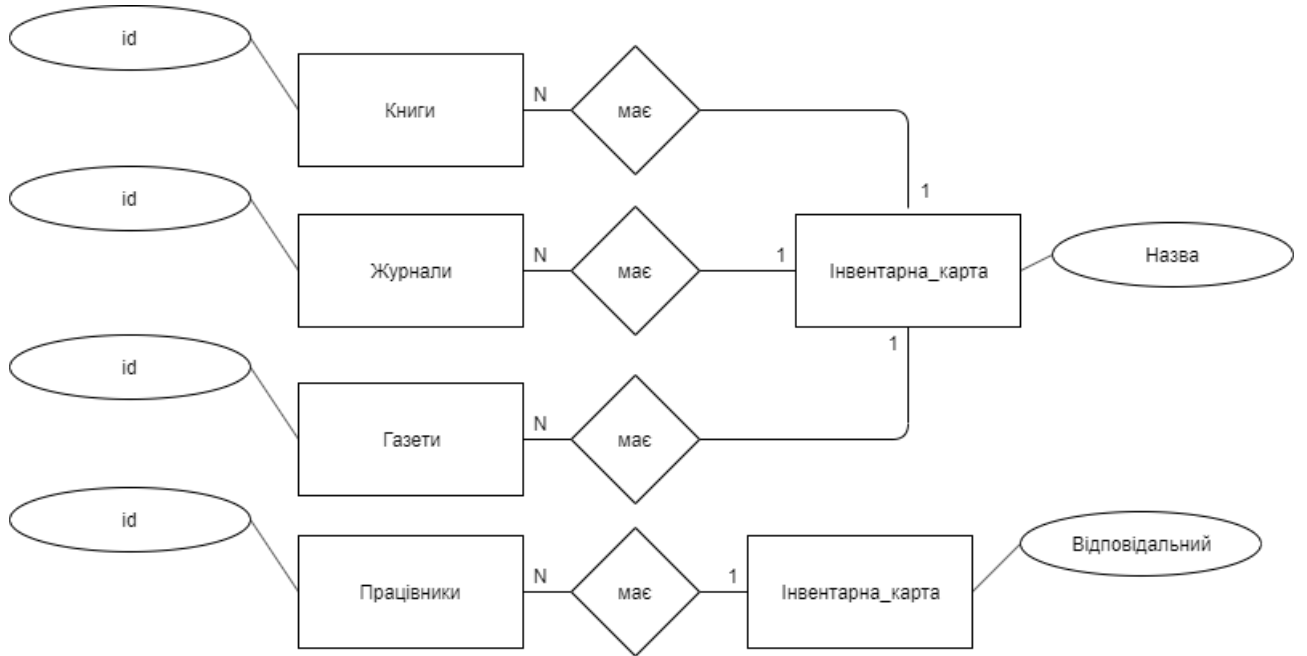
ПЕРЕЛІК ПОСИЛАНЬ

1. Воронько К. Л. Бібліотечні фонди: Навч. для бібл. – М. : Книжкова палата, 1992. – 199 с.
2. Столяров. Ю. Н. Бібліотечний фонд: підручник. – М. : Книжкова палата, 1991. – 271 с.
3. Вікіпедія. УФД. Бібліотека. URL: <https://uk.wikipedia.org/wiki/УФД/Бібліотека> (дата звернення 27.01.2020).
4. УФД. Бібліотека. Версія 2.4. Довідкова інформація. URL: <http://ush.com.ua/info/Products/LibraryHelp/Index.html> (дата звернення 12.02.2020).
5. Вікіпедія. Коша. URL: <https://uk.wikipedia.org/wiki/Коша> (дата звернення 22.02.2020).
6. Шаров С. В., Осадчий В. В. Бази даних та інформаційні системи : навч. посіб. Мелітополь : МДПУ ім. Б. Хмельницького, 2014. 352 с.
7. Берко А. Ю., Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 2. Системи управління базами даних та знань : навч. посіб. Львів : Магнолія-2006, 2012. 584 с.
8. Пасічник В. В., Резніченко В. А. Організація баз даних та знань : підруч. для вузів. К. : Видавнича група ВНУ, 2006. 384 с.
9. Дейт К. Дж. Введення в системи баз даних : пер. з англ. М. : Вільямс, 2005. 1328 с.
10. Уніфікована мова моделювання UML. URL: <http://www.znannya.org/view=uml> (дата звернення 27.04.2020).
11. Методичні вказівки для побудови uml діаграм до дипломного проекту для студентів спеціальності 121 «Інженерія програмного забезпечення» / упоряд. А. Л. Біленький. Гусятин : Вид-во ГК ТНТУ, 2019. 13 с.

12. Платформа .NET та її застосування для ООП URL: <https://www.znannya.org/?view=csharp-dotNET> (дата звернення: 29.04.2020).
13. Мова програмування С# 9 і платформа.NET 5: основні принципи та практики програмування. – Ендрю Троелсен, Філіп Джепікс, 2023. – 632 с.
14. Робота з базами даних в С# та .NET. URL: <https://metanit.com/sharp/ado.php> (дата звернення 02.05.2020).
15. Тестування програмного забезпечення. URL: https://uk.wikipedia.org/wiki/Тестування_програмного_забезпечення (дата звернення 12.05.2020).
16. Охорона праці при роботі з комп'ютерною технікою. URL: <http://oppb.com.ua/content/ohorona-praci-pry-roboti-z-kompyuternoyu-tehnikoyu> (дата звернення 05.05.2020).
17. Організація робочого місця. URL: http://www.proect60.narod/index35_21.html (дата звернення 05.05.2020).
18. Жидецький В. Ц. Охорона праці користувачів комп'ютерів. Львів : Афіша, 2000. 176 с.
19. Гандзюк М. П., Желібо Є. П., Халімовський М. О. Основи охорони праці : підручник / за ред. М. П. Гандзюка. К. : Каравела, 2008 . 384 с.
20. Безпека Життєдіяльності URL: <https://kpmu.km.ua/upload/medialibrary/8ca/BYEITYENA-zykhkhzhekyaoersfkhk-Iatsuszyeshch.pdf> (дата звернення 05.05.2020).
21. Довідник спеціаліста з охорони праці. URL: <https://www.sop.com.ua/> (дата звернення 08.05.2020).

ДОДАТКИ

Додаток А
ER-діаграма бази даних «Biblioteka»



Додаток Б

Текст програми «Biblioteka»

```

private void поступленіToolStripMenuItem_Click(object sender,
EventArgs e)
{
    dataGridView1.Rows.Clear();
    Form1 f1 = new Form1();
    f1.connection.Open();
    ZvitPrice zvitPrice = new ZvitPrice();
    zvitPrice.ShowDialog();
    if (zvitPrice.DialogResult == DialogResult.OK)
    {
        SqlCommand command = new SqlCommand("Select
Інвентарна_карта, Назва, Ціна_відшкодування, Відповідальний " +
        "From Інвентарна_карта Where Ціна_відшкодування>" +
zvitPrice.textBox1.Text + "' and Ціна_відшкодування<" +
zvitPrice.textBox2.Text + "'", f1.connection);
        SqlDataReader reader = command.ExecuteReader();
        if (reader.HasRows)
        {
            while (reader.Read())
            {
                dataGridView1.Rows.Add(reader["Інвентарна_карта"], reader["Назва"],
                reader["Ціна_відшкодування"], reader["Відповідальний"]);
            }
        }
    }
}
private void заВідповідальнимToolStripMenuItem_Click(object
sender, EventArgs e)
{
    dataGridView1.Rows.Clear();
    Form1 f1 = new Form1();
    f1.connection.Open();
    ZvitVidpovid zvitVidpovid = new ZvitVidpovid();
    zvitVidpovid.ShowDialog();
    if (zvitVidpovid.DialogResult == DialogResult.OK)
    {
        SqlCommand command = new SqlCommand("Select id From
Працівники Where ПІБ= '" + zvitVidpovid.comboBox1.Text + "'",
f1.connection);
        int vidpovid =
Convert.ToInt32(command.ExecuteScalar().ToString());

```

Продовження Додатку Б


```

        SqlCommand com = new SqlCommand("Select Інвентарна_карта,
Назва, Ціна_відшкодування, Відповідальний From Інвентарна_карта Where
Відповідальний='" + vidpovid + "'", f1.connection);
        SqlDataReader reader = com.ExecuteReader();
        if (reader.HasRows)
        {
            while (reader.Read())
            {
                dataGridView1.Rows.Add(reader["Інвентарна_карта"], reader["Назва"],
reader["Ціна_відшкодування"], reader["Відповідальний"]);
            }
        }
    }
}
private void MainForm_Load(object sender, EventArgs e)
{
    Form1 FM = new Form1();

    if (FM.textBox3.Text == "Bibliotekar")//(FM.DialogResult ==
DialogResult.OK)
    {
        //string s = FM.comboBox1.Text().To;
        //MessageBox.Show(s);
        звітиToolStripMenuItem.Enabled = false;
        аналізToolStripMenuItem.Visible = false;
    }
    if (FM.comboBox1.Text == "Buhalter")
    {
        довідникиToolStripMenuItem.Visible = false;
    }
}
private void експортВExcelToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Microsoft.Office.Interop.Excel.Application ExceApp = new
Microsoft.Office.Interop.Excel.Application();
    ExceApp.SheetsInNewWorkbook = 1;
    ExceApp.Application.Workbooks.Add(Type.Missing);
    ExceApp.Application.Workbooks[1].Worksheets[1].Name = "Звіт";
    ExceApp.Columns.ColumnWidth = 20;
    for (int i=1; i < dataGridView1.ColumnCount; i++)

```

Продовження Додатку Б

```

        {
            ExceApp.Cells[1, i+1] = dataGridView1.Columns[i-
1].HeaderText;
        }
        for (int i=0; i < dataGridView1.ColumnCount; i++)
        {
            for(int j=0; j<dataGridView1.RowCount; j++)
            {
                ExceApp.Cells[j + 2, i + 1] = dataGridView1[i,
j].Value;
            }
        }
        ExceApp.Visible = true;
    }
    private void экспортBWordToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        Word._Application application = new Word.Application();
        Word._Document document = new Word.Document();
        Object missingObj = System.Reflection.Missing.Value;
        Object trueObj = true;
        Object falseObj = false;
        Object templatePathObj = "C:\\Users\\Павчишен
Олександр\\source\\repos\\Biblioteka\\Biblioteka\\file.docx";
        document = application.Documents.Add(ref templatePathObj, ref
missingObj, ref missingObj, ref missingObj);
        Object bookmarkNameObj = "a";
        Word.Range bookmarkRange = null;
        bookmarkRange = document.Bookmarks.get_Item(ref
bookmarkNameObj).Range;
        Object behavior =
Word.WdDefaultTableBehavior.wdWord9TableBehavior;
        Object autoFitBehavior =
Word.WdAutoFitBehavior.wdAutoFitFixed;
        int rows = dataGridView1.RowCount + 1;
        int columns = dataGridView1.ColumnCount;
        document.Tables.Add(bookmarkRange, rows, columns, ref
missingObj, ref missingObj);
        for(int i=1; i<dataGridView1.ColumnCount + 1; i++)
        {
            document.Tables[1].Cell(1, i).Range.Text =
dataGridView1.Columns[i - 1].HeaderText;
        }
        for (int i = 0; i < dataGridView1.ColumnCount; i++)
        {
            Продовження Додатку Б

            for (int j = 0; j < dataGridView1.RowCount; j++)

```

```

        {
            document.Tables[1].Cell(j + 2, i + 1).Range.Text =
Convert.ToString(dataGridView1[i, j].Value);
        }
    }
    application.Visible = true;
}

private void проПрограмуToolStripMenuItem_Click(object sender,
EventArgs e)
{
    AboutBox1 about = new AboutBox1();
    about.ShowDialog();
    this.Show();
}
private void довідкаToolStripMenuItem1_Click(object sender,
EventArgs e)
{
    Help.ShowHelp(this, "help.chm");
}
private void toolStripButton5_Click(object sender, EventArgs e)
{
    Form1 f1 = new Form1();
    f1.connection.Open();
    if (toolStripComboBox1.Text == "Пошук працівника за
прізвищем")
    {
        SqlCommand command = new SqlCommand("Select ПІБ, Посада,
Адреса, Телефон From Працівники " +
        " Where ПІБ='" + toolStripTextBox1.Text + "'",
f1.connection);
        SqlDataReader reader = command.ExecuteReader();
        dataGridView1.Rows.Clear();
        dataGridView1.Columns.Clear();

        if (reader.HasRows)
        {
            while (reader.Read())
            {
                dataGridView1.Columns.Add("Column1", "ПІБ");
                dataGridView1.Columns.Add("Column2", "Посада");

                dataGridView1.Columns.Add("Column3", "Адреса ");
                dataGridView1.Columns.Add("Column4", "Телефон");
            }
        }
    }
}

```

Продовження Додатку Б

```

        dataGridView1.Rows.Add(reader["ПІБ"],
reader["Посада"], reader["Адреса"], reader["Телефон"]);
    }
}
if (toolStripComboBox1.Text == "Пошук Журнала за роком
видання")
{
    dataGridView1.Rows.Clear();
    dataGridView1.Columns.Clear();
    SqlCommand command = new SqlCommand("Select Назва, Автор,
Видавництво, Рік_видання, Стан, Інвентарний_номер, Ціна_відшкодування
From Журнали " +
        " Where Рік_видання='" + toolStripTextBox1.Text +
"", f1.connection);
    SqlDataReader reader = command.ExecuteReader();

    if (reader.HasRows)
    {
        while (reader.Read())
        {
            dataGridView1.Columns.Add("Column1", "Назва");
            dataGridView1.Columns.Add("Column2", "Автор");
            dataGridView1.Columns.Add("Column3", "Видавництво
");
            dataGridView1.Columns.Add("Column4",
"Рік_видання");
            dataGridView1.Columns.Add("Column5", "Стан");
            dataGridView1.Columns.Add("Column6",
"Інвентарний_номер");
            dataGridView1.Columns.Add("Column7",
"Ціна_відшкодування");
            dataGridView1.Rows.Add(reader["Назва"],
reader["Автор"], reader["Видавництво"], reader["Рік_видання"],
reader["Стан"], reader["Інвентарний_номер"],
reader["Ціна_відшкодування"]);
        }
    }
}
if (toolStripComboBox1.Text == "Пошук Газет за роком
видання")
{
    dataGridView1.Rows.Clear();
        Продовження Додатку Б

    dataGridView1.Columns.Clear();

```

```

        SqlCommand command = new SqlCommand("Select Назва, Автор,
Видавництво, Рік_видання, Стан, Інвентарний_номер, Ціна_відшкодування
From Газети " +
            " Where Рік_видання='" + toolStripTextBox1.Text +
            "'", f1.connection);
        SqlDataReader reader = command.ExecuteReader();
        if (reader.HasRows)
        {
            while (reader.Read())
            {
                dataGridView1.Columns.Add("Column1", "Назва");
                dataGridView1.Columns.Add("Column2", "Автор");
                dataGridView1.Columns.Add("Column3", "Видавництво
");
                dataGridView1.Columns.Add("Column4",
"Рік_видання");
                dataGridView1.Columns.Add("Column5", "Стан");
                dataGridView1.Columns.Add("Column6",
"Інвентарний_номер");
                dataGridView1.Columns.Add("Column7",
"Ціна_відшкодування");
                dataGridView1.Rows.Add(reader["Назва"],
reader["Автор"], reader["Видавництво"], reader["Рік_видання"],
reader["Стан"], reader["Інвентарний_номер"],
reader["Ціна_відшкодування"]);
            }
        }
        if (toolStripComboBox1.Text == "Пошук Книги за роком
видання")
        {
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();
            SqlCommand command = new SqlCommand("Select Назва, Автор,
Видавництво, Рік_видання, Стан, Інвентарний_номер, Ціна_відшкодування
From Книги " +
            " Where Рік_видання='" + toolStripTextBox1.Text +
            "'", f1.connection);
            SqlDataReader reader = command.ExecuteReader();
            if (reader.HasRows)
            {
                while (reader.Read())
                {

```

Продовження Додатку Б

```

                dataGridView1.Columns.Add("Column1", "Назва");
                dataGridView1.Columns.Add("Column2", "Автор");

```

```
");
dataGridView1.Columns.Add("Column3", "Видавництво");
"Рік_видання");
dataGridView1.Columns.Add("Column4",
dataGridView1.Columns.Add("Column5", "Стан");
dataGridView1.Columns.Add("Column6",
"Інвентарний_номер");
dataGridView1.Columns.Add("Column7",
"Ціна_відшкодування");
dataGridView1.Rows.Add(reader["Назва"],
reader["Автор"], reader["Видавництво"], reader["Рік_видання"],
reader["Стан"], reader["Інвентарний_номер"],
reader["Ціна_відшкодування"]);
    }
}
}
}
```