

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства та природокористування
Навчально-науковий інститут кібернетики, інформаційних технологій та
інженерії
Кафедра комп'ютерних технологій та економічної кібернетики

Допущено до захисту:

Завідувач кафедри
_____ д. е. н., проф. П. М. Грицюк
«_____» _____ 20__ р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня «бакалавр»
за освітньо-професійною програмою «Інформаційні системи та
технології» спеціальності 126 «Інформаційні системи та технології»
на тему: «Розробка інформаційної системи управління роботою бібліотеки»

Виконав:

здобувач вищої освіти 4 курсу, групи ІСТ-41
Шпатусько Олексій Русланович

Керівник:

к.е.н. доцент Бабич Т. Ю.

Рецензент:

канд. техн. наук, доцент Гладка О. М

Рівне- 2026

АНОТАЦІЯ

Шпатусько О.Р. Розробка інформаційної системи управління роботою бібліотеки. Кваліфікаційна робота на здобуття освітнього ступеня «бакалавр»: 61 с., 26 рис., 19 табл., 17 літературних джерел.

Об'єкт дослідження- діяльність бібліотеки як об'єкта автоматизації та сукупність процесів з обліку фондів та читачів.

Предмет дослідження- методи та засоби проектування та впровадження інформаційної системи бібліотечного обліку, що включає створення реляційної моделі даних, організацію зв'язків між сутностями, реалізацію запитів-звітів, а також веб-інтерфейсу.

Методи дослідження- методи системного аналізу, концептуального та реляційного моделювання баз даних, об'єктно-орієнтованого програмування, а також методи функціонального тестування програмного забезпечення.

Дипломна робота присвячена розробці програмного комплексу для автоматизації облікових процесів сучасної бібліотеки. Результатом роботи стало проектування та реалізація реляційної бази даних у середовищі СУБД Microsoft Access, яка забезпечує надійне зберігання інформації, ведення електронного каталогу книг та примірників, реєстрацію читачів, фіксацію операцій видачі/повернення, а також автоматичний контроль термінів користування і нарахування штрафів за прострочення.

Для розширення можливостей системи було розроблено демонстраційний веб-інтерфейс на базі платформи Node.js (фреймворк Express.js) з використанням шаблонізатора EJS та підключенням до MS Access через системний драйвер ODBC. Виділено основні переваги розробленої архітектури у забезпеченні цілісності даних, швидкості обслуговування користувачів та мінімізації людського фактора. Розглянуто архітектурні шаблони побудови веб-застосунків та надано рекомендації щодо подальшої міграції на серверні СУБД.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, АВТОМАТИЗАЦІЯ РОБОТИ БІБЛІОТЕКИ, БАЗА ДАНИХ, РЕЛЯЦІЙНА МОДЕЛЬ, MICROSOFT ACCESS, NODE.JS, EXPRESS.JS, ODBC, ОБЛІК КНИГ.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІНФОРМАЦІЙНОГО СЕРЕДОВИЩА БІБЛІОТЕКИ	6
1.1. Загальна характеристика бібліотеки як об'єкта автоматизації	6
1.2. Аналіз бізнес-процесів обліку книг і читачів.....	11
1.3. Дослідження програмно-апаратного забезпечення та умов експлуатації..	16
РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ БІБЛІОТЕКИ	20
2.1. Аналіз інформаційних потоків та фактичних даних у бібліотеці	20
2.2. Формування функціональних вимог до інформаційної системи	27
2.3. Нефункціональні вимоги (якість, безпека, продуктивність)	31
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ ОБЛІКУ БІБЛІОТЕКИ В MS ACCESS ТА ВЕБ-ІНТЕРФЕЙС.....	33
3.1. Загальна архітектура розробленої інформаційної системи	33
3.2. Тестування системи	46
3.3. Архітектура веб-інтерфейсу.....	48
ВИСНОВКИ	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60

ВСТУП

Сучасні бібліотеки працюють в умовах зростання інформаційних ресурсів та фондових операцій. Вимоги користувачів до швидкого обслуговування посилюють навантаження на бібліотекарів. Практично неможливо підтримувати цілісність даних під час спроб перевірити наявні примірники, ввести нові книги чи читачів у систему або отримати будь-яку інформацію про них за допомогою традиційних методів обліку (паперові журнали та електронні таблиці). Також відсутня підтримка статистичної звітності- ще одна функція, яка стала дуже затребуваною керівництвом бібліотек на всіх рівнях.

Актуальність теми обґрунтована можливістю централізованого зберігання та обробки бібліотечних даних, оптимізації ключових бізнес-процесів (каталогізації, обліку копій, реєстрації читачів, видачі/повернення та контролю прострочених видань); автоматизації додано аналітичне формування звітів. Система забезпечує чітке розмежування між «книгою» як бібліографічним записом та «копією» як фактичною фізичною одиницею фонду, а отже, точний облік та моніторинг наявності літератури.

Метою роботи є створення інформаційної системи для обліку ресурсів бібліотеки, яка забезпечує каталог книг та примірників, облік читачів, підтримку операцій видачі та повернення, контроль прострочених позик, а також звітність та аналітичні зведення. Для досягнення мети в роботі використовувалася платформа Microsoft Access як інструмент для швидкого створення реляційної бази даних, форм та звітів; також було реалізовано демонстраційний веб-інтерфейс на базі Node.js (Express) для ілюстрації можливостей інтеграції десктопного рішення з веб-технологіями.

Об'єктом дослідження є діяльність бібліотеки як об'єкта автоматизації та сукупність процесів з обліку фондів та читачів. Предметом дослідження є методи та засоби проектування та впровадження інформаційної системи бібліотечного обліку, що включає створення реляційної моделі даних,

організацію між сутностями, реалізацію запитів-звітів тощо, окрім дотримання основних вимог якості та безпеки [9].

Для досягнення мети роботи було поставлено такі завдання:

- проаналізувати предметну область бібліотеки, її організаційну структуру та ролі користувачів;
- описати ключові бізнес-процеси та інформаційні потоки, виявити проблеми в сучасному стані обліку ресурсів бібліотеки;
- розглянути та порівняти аналоги програмного забезпечення, сформулювати функціональні та нефункціональні вимоги до системи.
- визначити логічні та фізичні таблиці бази даних, ключі, зв'язки, індекси; розробити модулі у формах MS Access, SQL-запитах, звітах;
- провести функціональне тестування розробленого прототипу на основі типових сценаріїв,
- розробити демонстраційний веб-інтерфейс у node.js для демонстрації основних функцій системи та прикладів взаємодії з базою даних MS Access.

Практичне значення отриманих результатів. Розроблена інформаційна система управління роботою бібліотеки має повну функціональну готовність і може бути безпосередньо впроваджена в діяльність автоматизованих робочих місць бібліотекарів. Створений комплекс, що поєднує реляційну базу даних та демонстраційний веб-інтерфейс, дозволяє оптимізувати процеси реєстрації користувачів, обліку книжкового фонду та контролю обігу літератури, що забезпечує зниження трудовитрат персоналу та мінімізацію помилок ручного введення даних. Усі вихідні форми документів та аналітичні звіти системи спроектовано з урахуванням чинних вимог до діловодства та правил бібліографічного опису [2, 3].

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІНФОРМАЦІЙНОГО СЕРЕДОВИЩА БІБЛІОТЕКИ

1.1. Загальна характеристика бібліотеки як об'єкта автоматизації

Бібліотека як організація виконує соціально важливі функції з формування та збереження бібліотечного фонду, забезпечення доступу користувачів до інформаційних ресурсів, організації культурно-освітньої діяльності та інформаційного обслуговування населення [1]. Для ефективної реалізації цих функцій бібліотека здійснює комплекс взаємопов'язаних процесів: комплектування фонду, каталогізацію, ведення обліку примірників, реєстрацію читачів, контроль видачі та повернення літератури, формування статистичної звітності.

Об'єктом автоматизації у межах даної роботи є основні операційні процеси бібліотеки, пов'язані з обліком книг і читачів, зокрема:

- ведення електронного каталогу книг (реєстрація, редагування, пошук, списання/додавання примірників);
- ведення обліку читачів (створення картки читача, реєстрація, оновлення контактних даних);
- облік бібліотечних операцій (видача, повернення, продовження терміну користування);
- контроль прострочених повернень і формування списків боржників;
- формування звітів.

В умовах бібліотеки облік є регулярним і безперервним процесом, який накопичує значний обсяг даних у часі. Навіть якщо організація невелика, протягом 1–2 років кількість записів може вимірюватися тисячами операцій

(видач/повернень), що створює потребу в систематизації та централізованому зберіганні інформації.

Для бібліотеки типові інформаційні об'єкти:

- книга як бібліографічний запис (назва, автор, рік, жанр, видавництво, мова);
- примірник книги як одиниця фонду (інвентарний номер, стан, доступність);
- читач (ПІБ, контакти, реквізити читацького квитка, дата реєстрації);
- операція видачі (дата видачі, термін повернення, дата повернення, відповідальний працівник).

Ключова складність предметної області — необхідність коректно відрізняти “книгу” як назву/видання від “примірника” як фізичної одиниці фонду. У ручних системах часто обліковують лише назву книги, що призводить до помилок доступності (книга наче “є”, але конкретний примірник виданий).

Таким чином, бібліотека є типовим прикладом організації, де впровадження інформаційної системи дає швидкий ефект: скорочення часу операцій, зменшення помилок, можливість аналітики, підвищення якості обслуговування читачів [8]. У процесі проектування інформаційної системи бібліотеки першочерговим завданням є формалізація та структурування даних про її діяльність. Автоматизація обліку книжкового фонду, реєстрації користувачів та контролю за обігом літератури вимагає чіткого визначення інформаційних потоків. Для цього проводиться концептуальний аналіз предметної області, який дозволяє виокремити базові інформаційні блоки, що надалі трансформуються в елементи бази даних.

Основні інформаційні сутності, що складають ядро досліджуваної бібліотечної системи, їхній функціональний опис, набір ідентифікаційних атрибутів та наочні приклади реальних даних зведені та систематизовані в таблиці 1.1.

Таблиця 1.1

Основні інформаційні об'єкти бібліотеки та їх атрибути

Інформаційний об'єкт	Опис	Ключові атрибути	Приклад
Книга (бібліографічний запис)	Опис видання незалежно від кількості примірників	Назва, Автор, ISBN/код, Рік, Жанр, Мова, Видавництво	“Гаррі Поттер”, 2002
Примірник	Фізична одиниця фонду	Інвентарний №, Стан, Статус.	В наявності
Читач	Зареєстрований користувач	ПІБ, Телефон, Email, № квитка, Дата реєстрації	Номер квитка “1001”
Видача	Факт передачі примірника	Читач, Дата видачі, Термін, Дата повернення	Видано 10.04 → повернути до 24.04
Прострочення	Порушення терміну повернення	ДатаDue < Date(), Днів прострочення	7 днів
Штраф	Фінансова санкція	Сума	35 грн

Для побудови логічної структури бази даних бібліотечної системи необхідно першочергово виділити ключові сутності предметної області, визначити їхнє функціональне призначення та сформувати базовий набір атрибутів.

Такий підхід дозволяє уникнути дублювання інформації в базі даних, забезпечує її нормалізацію та встановлення коректних реляційних зв'язків типу «один-до-багатьох» (наприклад, між одним об'єктом «Читач» та багатьма фактами «Видачі»).

Окрема увага приділена об'єктам «Прострочення» та «Штраф», що є критично важливим для автоматизації контролю за вчасним поверненням літератури та розрахунку фінансових санкцій. Формалізація цих сутностей базується на принципах математичного моделювання дискретних операційних процесів. Це дозволяє трансформувати бізнес-логіку бібліотеки (регламентні терміни користування книгами) у чіткий алгоритм автоматичного моніторингу, де настання події «Прострочення» є тригером для динамічного нарахування суми за об'єктом «Штраф». Подібна

автоматизація знижує вплив людського фактора, мінімізує збитки книжкового фонду та оптимізує роботу адміністратора системи.

Ефективне функціонування сучасної бібліотеки вимагає чіткої алгоритмізації всіх внутрішніх процесів — від моменту закупівлі літератури до обслуговування кінцевого користувача. Цикл книги та супровідні операційні процедури утворюють комплексну технологічну лінійку, де кожна дія має фіксовані вхідні документи, регламентовані етапи обробки та перевірки.

Для формалізації цієї логіки та подальшого проектування бази даних необхідно детально дослідити послідовність основних бізнес-процесів установи. Основними етапами тут виступають первинне надходження фонду, його каталогізація, ідентифікація користувачів та безпосередній блок оперативного обслуговування, що включає видачу, контроль термінів користування та повернення літератури з фіксацією можливих порушень регламенту. Повна технологічна схема взаємодії та послідовність виконання ключових операційних процесів у бібліотеці відображена на рисунку 1.1.

Організаційна структура бібліотеки залежить від її масштабу, проте здебільшого включає адміністративну та операційну частини [5]. Декомпозиція основних інформаційних об'єктів, їхніх характеристик та взаємозв'язків деталізовано впливає з логіки технологічних етапів, представлених на схемі. Архітектура системи спирається на чітке концептуальне розмежування між поняттями «Книга» (абстрактний бібліографічний запис на етапі каталогізації) та «Примірник» (конкретна фізична одиниця з унікальним інвентарним номером, що формується для фонду). Такий підхід дозволяє повністю уникнути дублювання інформації та забезпечити нормалізацію бази даних. Детальний аналіз відображеної на рисунку 1.1 послідовності процесів дозволяє виділити три ключові рівні автоматизації:

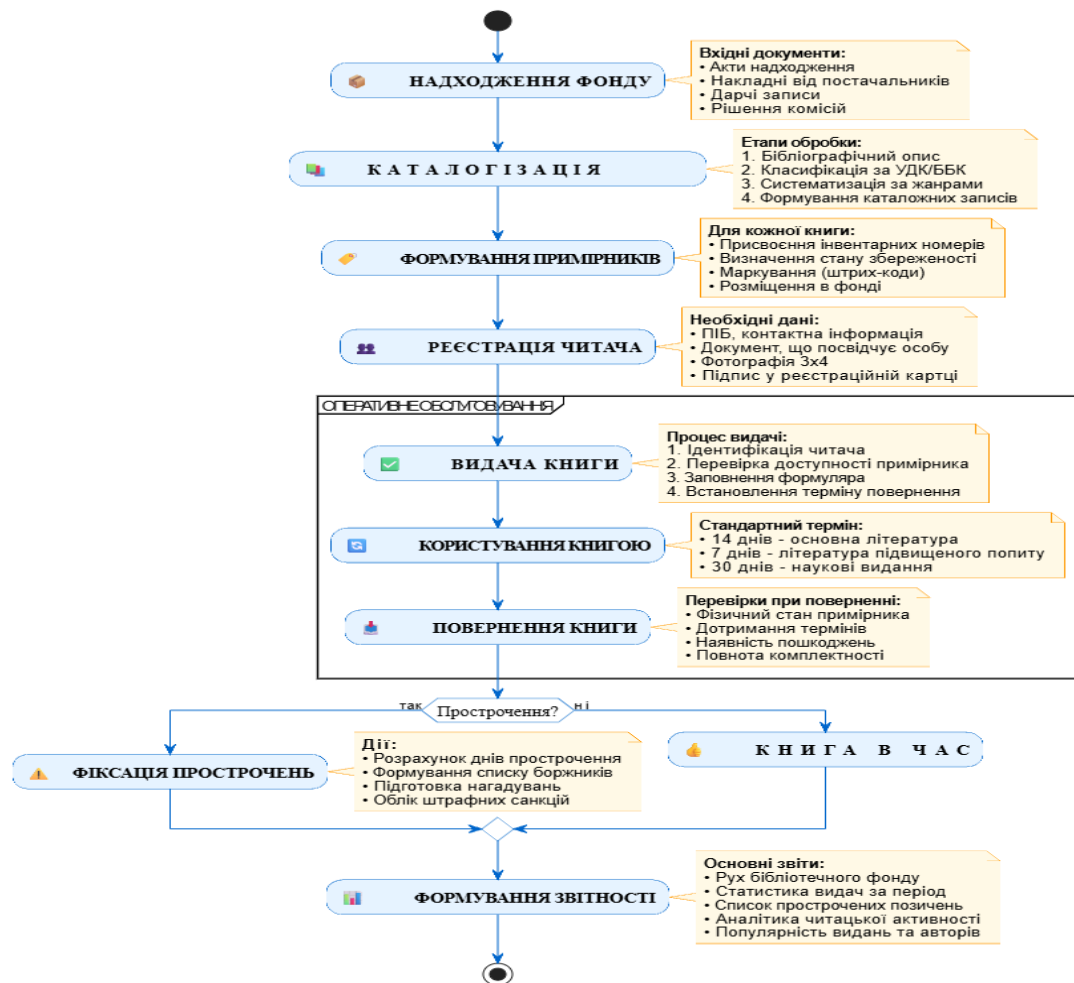


Рис. 1.1 Схема процесів бібліотеки

1. Етап обліку та каталогізації фонду. Початкова точка процесу пов'язана з обробкою вхідних документів (актів надходження, накладних тощо). Перехід до кроку «Каталогізація» вимагає створення бібліографічного опису, класифікації за системами УДК/ББК та систематизації за жанрами. Автоматизація цього етапу дозволяє оператору один раз створити картку книги, після чого для кожного фізичного екземпляра автоматично генеруються унікальні штрих-коди та інвентарні номери на етапі формування примірників.
2. Блок оперативного обслуговування користувачів. Цей контур об'єднує процеси реєстрації читача, видачі, користування та повернення книг. Схема чітко демонструє диференціацію термінів користування залежно від типу літератури (14 днів для основної, 7 днів для підвищеного попиту та 30 днів для наукових видань), що повинно бути жорстко

запрограмовано в логіці системи. При поверненні книги обов'язковою є перевірка її фізичного стану та дотримання термінів.

3. Логічне розгалуження та аналітична звітність. Критично важливою точкою алгоритму є умовна перевірка «Прострочення?». Вона виступає тригером: у разі вчасного повернення книга переходить у статус «Вчасно», а процес завершується. У разі виявлення прострочення система запускає підпроцес «Фіксація прострочень», який автоматично розраховує дні затримки, нараховує штрафні санкції та формує списки боржників.

Для успішного проектування та впровадження інформаційної системи необхідно чітко розмежувати ролі користувачів, визначити їхні функціональні обов'язки, зони відповідальності та внутрішні інформаційні зв'язки. Загальна схема організаційної структури бібліотеки, склад персоналу, ключові функції та напрямки взаємодії між суб'єктами системи представлені на рисунку 1.2

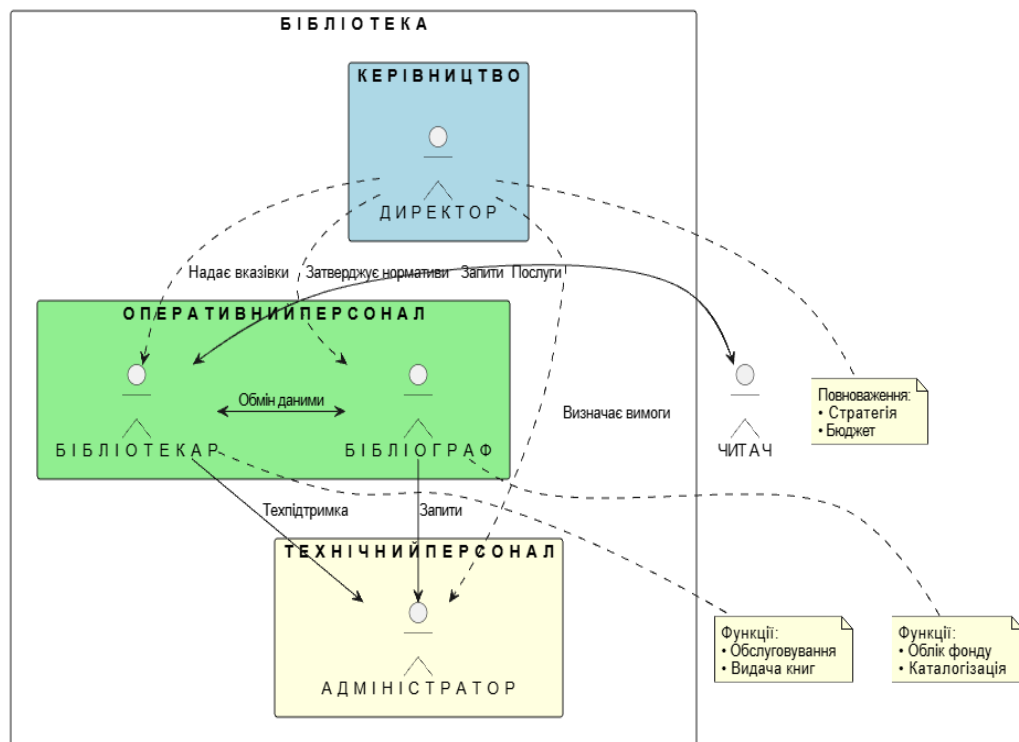


Рис. 1.2 Організаційна структура та взаємодія ролей

Розроблена загальна схему операційних процесів дозволяє детально відобразити, які саме завдання виконуються всередині бази даних під час повсякденної роботи.

1.2. Аналіз бізнес-процесів обліку книг і читачів

Для обґрунтування вимог до інформаційної системи необхідно описати ключові бізнес-процеси, які підлягають автоматизації.

Процес 1: Реєстрація читача

Вхідні дані: паспортні/контактні дані, ПІБ, телефон/e-mail (за наявності).

Етапи:

- перевірка, чи не існує читач з такими контактами/ПІБ;
 - внесення даних у реєстр читачів;
- присвоєння номера читацького квитка;
- фіксація дати реєстрації та відповідального працівника.

Результат: створена картка читача, що може бути використана для операцій видачі.

Типові проблеми ручного процесу:

- дублювання читачів (через помилки в ПІБ);
- відсутність швидкого пошуку за телефоном;
- складність відновлення даних при втраті читацького квитка.

Процес 2: Облік бібліотечного фонду (книги та примірники)

Етапи:

- внесення бібліографічної інформації про книгу;
- створення примірників (інвентарні номери, кількість, стан);
- встановлення статусу доступності (в наявності/видано/списано).

Результат: каталог, що підтримує пошук і контроль доступності.

Типові проблеми:

- плутанина між “книга” і “примірник”;

- відсутність контролю унікальності інвентарних номерів;
- ручний підрахунок кількості доступних примірників.

Процес 3: Видача книги читачу

Вхідні дані: читач, примірник книги.

Етапи:

- ідентифікація читача (квиток/ПІБ/телефон);
- пошук книги, перевірка доступних примірників;
- вибір конкретного примірника;
- фіксація дати видачі та терміну повернення;
- зміна статусу примірника на “видано”.

Результат: запис операції видачі.

Типові проблеми:

- видача “вручну” без перевірки доступності;
- неможливість швидко з’ясувати, хто тримає конкретний примірник;
- помилки при записі дат.

Процес 4: Повернення книги

Етапи:

- пошук активної операції видачі по примірнику/читачу;
- фіксація фактичної дати повернення;
- визначення, чи є прострочення;
- у разі прострочення — нарахування штрафу/позначення боргу;
- зміна статусу примірника на “в наявності”.

Результат: закрита операції видачі, оновлена доступність фонду.

Процес 5: Контроль прострочень і боржників

Етапи:

- відбір усіх видач без фактичної дати повернення;
- порівняння терміну повернення з поточною датою;
- формування списку прострочених видань;
- формування списку читачів-боржників;
- підготовка повідомлень/нагадувань (за потреби).

Результат: актуальний контроль дисципліни користування фондом.

Процес 6: Формування звітності

Приклади звітів:

- “Рух фонду за період”;
- “Найпопулярніші книги/жанри”;
- “Активні читачі”;
- “Прострочені повернення станом на дату”;
- “Списання/надходження фонду”.

Проблеми ручного формування звітів:

- значний час на зведення даних;
- ризик арифметичних та логічних помилок;
- складність отримання аналітики “за період”.

Для обґрунтування доцільності розробки інформаційної системи необхідно провести порівняльний аналіз організації ключових процесів бібліотеки до впровадження автоматизації та після нього. Моделювання бізнес-процесів у форматах «як є» (AS-IS) та «як має бути» (TO-BE) дозволяє оцінити якісні зміни в роботі оператора. Зведений опис поточного стану процесів, архітектури їхньої трансформації в інформаційній системі та очікуваного ефекту від автоматизації представлено в таблиці 1.2.

На основі порівняння, наведеного у таблиці 1.2, можна зробити висновок, що впровадження інформаційної системи повністю усуває рутинні паперові операції. Перехід від стану AS-IS до TO-BE забезпечує такі ключові переваги:

Швидкість обслуговування: пошук за журналом замінюється миттєвим запитом за номером квитка чи телефону, що виключає появу дублів.

- Точність обліку: фіксація статусів примірників усуває хаос із доступністю книг, а параметр DateDue тепер розраховується автоматично, зводячи людські помилки до нуля.

Таблиця 1.2

Опис бізнес-процесів (AS-IS) та очікуваний ефект автоматизації (TO-BE)

Процес	AS-IS	TO-BE	Очікуваний ефект
Реєстрація читача	Запис у журнал/таблицю, ручний пошук	Єдина картка читача, пошук за квитком/телефоном	Менше дублів, швидкість пошуку
Облік фонду	Назви без прив'язки до примірника	Книга + примірники, статуси	Коректна доступність
Видача	Запис вручну, помилки дат	Вибір читача + примірника, DateDue авто	Мінімум помилок
Повернення	Пошук запису вручну	Закриття активної видачі, авто-статус	Швидке обслуговування
Прострочення	Перегляд журналу	Запит "Overdue" одним кліком	Контроль дисципліни
Звіти	1–3 дні на зведення	Авто-звіт на основі запитів	Оперативна аналітика

- Жорсткий контроль: виявлення боржників через запит "Overdue" відбувається в один клік замість тривалого ручного перегляду карток.
- Миттєва аналітика: зведення звітів скорочується з 1–3 днів до кількох секунд завдяки автоматичній генерації з бази даних.

VRMN/блок-схема процесу “Видача книги” відображена на рис. 1.3.

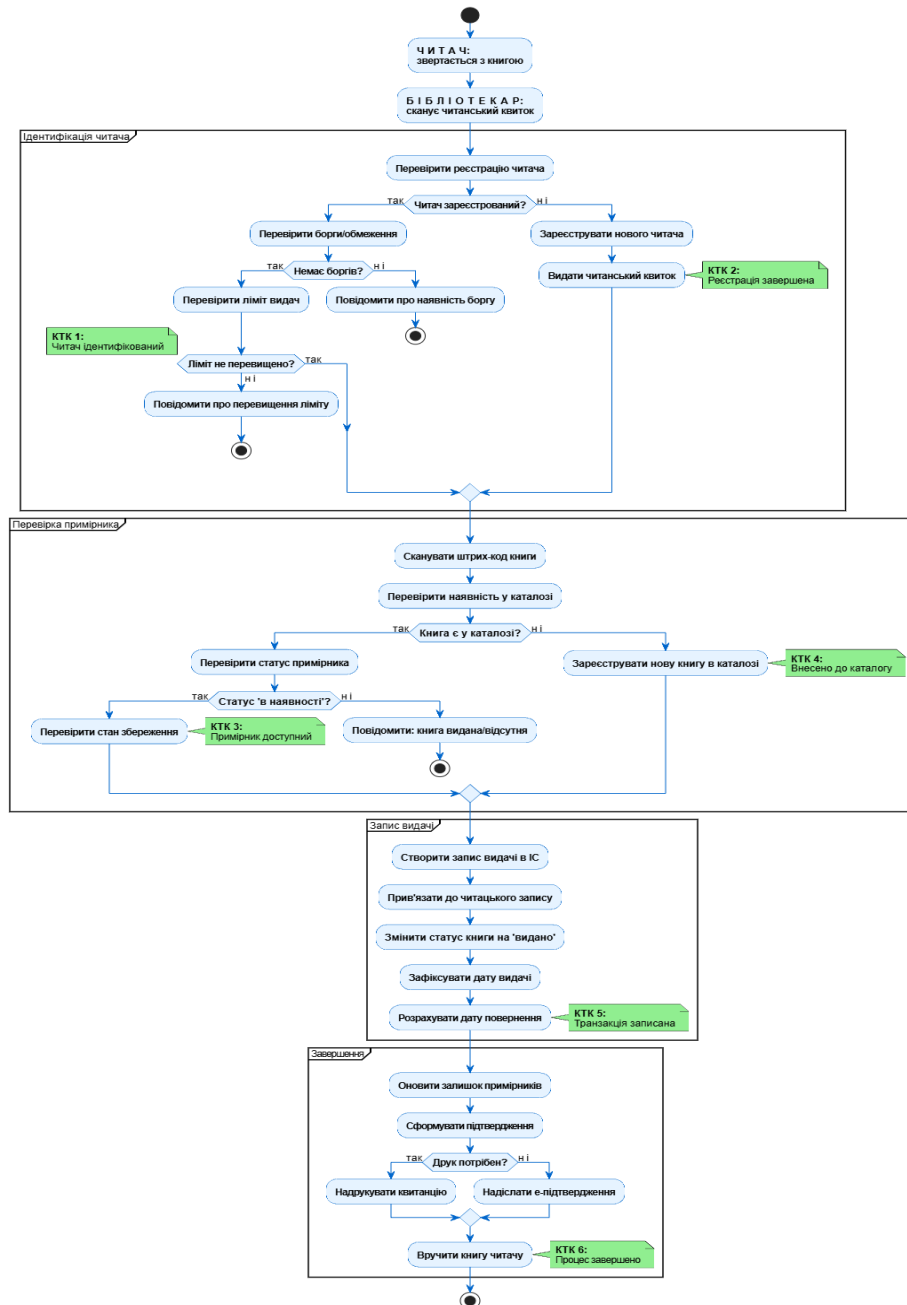


Рис. 1.3 BPMN/блок-схема процесу “Видача книги”

1.3. Дослідження програмно-апаратного забезпечення та умов експлуатації

Для впровадження інформаційної системи необхідно оцінити технічні умови, в яких вона буде працювати [6]. Типові умови для бібліотеки:

Апаратне забезпечення

– робочі станції (ПК) бібліотекарів;

- периферія: принтер (друк звітів), сканер (за потреби), інколи — сканер штрих-кодів;
- локальна мережа (якщо система використовується на декількох робочих місцях).

Програмне забезпечення

- операційна система Windows;
- пакет Microsoft Office (наявність MS Access);
- антивірусний захист;
- засоби резервного копіювання (вручну/автоматично на носій або мережеву папку).

Вимоги до експлуатації

- робота 1–3 користувачів одночасно;
- необхідність швидкого доступу до даних (пошук читача/книги до кількох секунд);
- захист від випадкового видалення критичних даних;
- можливість відновлення бази даних з резервної копії.

Оскільки методичні вимоги допускають розробку модулів ІС із застосуванням інструментальних засобів, у даній роботі доцільно використати MS Access як платформу, що дозволяє створити реляційну базу даних, форми та звіти, а також реалізувати логіку через запити і макроси/модулі.

Технічні умови експлуатації інформаційної системи відображено в табл. 1.4.

Таблиця 1.4

Технічні умови експлуатації інформаційної системи

Компонент	Характеристика	Вплив на проєкт
ОС	Windows 10/11	MS Access підтримується
ПЗ	Microsoft Office (Access)	Не потрібні додаткові ліцензії
Користувачі	1–3 одночасно	Достатньо файлової БД
Друк	Принтер	Формування звітів Access
Бекап	флеш/мережа	Регламент резервного копіювання

Під час дослідження предметної області встановлено, що при ручному або напівавтоматичному обліку (журнали/Excel) виникають такі типові проблеми:

1. Дублювання даних

- читач може бути внесений двічі (помилка в ПІБ, різні варіанти номеру телефону);
- книга може бути внесена кілька разів (різні формати запису автора/назви).

2. Відсутність цілісності та зв'язності даних

- дані про читача зберігаються окремо від даних про видачі;
- зв'язок “хто взяв який примірник” встановлюється вручну і може бути втрачений.

3. Неможливість оперативного контролю доступності

- у каталозі записано, що книга “є”, але скільки примірників доступно — потрібно рахувати вручну;
- при видачі виникають конфлікти (примірник уже виданий, але це не видно одразу).

4. Складність контролю прострочених повернень

- для визначення боржників потрібно переглядати журнали;
- штрафи/борги можуть бути порашовані некоректно або не зафіксовані.

5. Довге формування звітів

- підготовка місячної статистики займає значний час;
- звіти залежать від людського фактора.

6. Ризики втрати інформації

- документи можуть бути зіпсовані/втрачені;
- файли Excel можуть бути випадково перезаписані або пошкоджені.

Автоматизована система має усунути зазначені проблеми за рахунок:

- використання єдиної бази даних;
- впровадження ключів/індексів та зв'язків;
- автоматичного обліку доступності примірників;
- контрольованих операцій видачі/повернення;
- автоматизованих запитів для формування звітів.

У результаті аналізу встановлено, що бібліотека як об'єкт автоматизації характеризується наявністю стабільних типових процесів (облік фонду, читачів, операцій видачі/повернення, контроль прострочень), які потребують систематизації та централізованого зберігання даних. Ручні методи обліку не забезпечують необхідного рівня оперативності, точності та аналітичності, що негативно впливає на якість обслуговування читачів і навантаження персоналу.

Отже, створення інформаційної системи обліку бібліотеки є обґрунтованим і доцільним. У наступному розділі буде проведено аналіз вимог до системи, огляд аналогів та обґрунтування вибору інструментарію, а також сформовано повний перелік функціональних і нефункціональних вимог до розроблюваної ІС.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ БІБЛІОТЕКИ

2.1. Аналіз інформаційних потоків та фактичних даних у бібліотеці

Для проєктування інформаційної системи обліку бібліотеки необхідно визначити, які саме дані накопичуються, хто їх формує, як вони використовуються і які результати (вихідні дані) потрібні працівникам. У межах предметної області «книги та читачі» базовим є цикл “надходження → облік → видача → повернення → звітність”.

1. Потік даних про книги (фонд).

- Джерела: акти надходження, накладні, каталожні картки, записи попередніх систем (Excel/журнали).
- Дані, що фіксуються: бібліографічний опис (назва, автор(и), рік, видавництво, мова, жанр/тематика), ідентифікатори (ISBN, шифр/індекс, внутрішній код), кількісні характеристики (кількість примірників, стан, дата надходження).
- Використання: пошук, контроль наявності, статистика популярності, інвентаризація.

2. Потік даних про примірники (фізичні одиниці).

- Джерела: інвентарна книга/журнал, штрих-коди (за наявності), маркування.
- Дані: інвентарний номер, стан (новий/добрий/потребує ремонту/списаний), місце зберігання, статус (в наявності/видано/резерв/списано).
- Використання: видача конкретного примірника, контроль втрат, інвентаризація.

3. Потік даних про читачів.

- Джерела: анкета читача, реєстраційні журнали.
- Дані: ПІБ, дата народження (за потреби), контакти, адреса (опційно), номер квитка, дата реєстрації, категорія (студент/школяр/дорослий).
- Використання: пошук читача, історія видач, аналітика активності, контроль боржників.

4. Потік даних про видачу та повернення.

- Джерела: журнал видач/повернень.
- Дані: читач, примірник, дата видачі, дата повернення за правилами (термін), дата фактичного повернення, відповідальний працівник.
- Використання: контроль термінів, списки прострочень, статистика.

5. Потік даних про штрафи/борги (якщо бібліотека застосовує).

- Джерела: [не вказано]
- Дані: підстава (прострочення), сума, статус оплати, дата оплати.
- Використання: контроль дисципліни, фінансова звітність.

Висновок: фактичний матеріал бібліотеки природно структурований і добре підходить для реляційної моделі даних (таблиці, ключі, зв'язки) [11]. Найкритичніше для точності — облік на рівні примірника, а не лише назви книги.

Важливим етапом проектування інформаційної системи є визначення інформаційних потоків, що циркулюють в установі. Для забезпечення коректної роботи бази даних необхідно чітко класифікувати вхідні та вихідні дані, визначити джерела їхнього походження, а також зони використання та кінцевий результат обробки інформації. Зведену специфікацію вхідних та вихідних даних інформаційної системи бібліотеки представлено в табл. 2.1.

Таблиця 2.1

Вхідні та вихідні дані системи

Дані	Джерело	Хто вводить	Де використовується	Результат
Книги	накладні, акти	бібліотекар	каталог, пошук	картка книги
Примірники	інвентарна книга	бібліотекар	видача, контроль	інвентарний запис
Читачі	анкета	бібліотекар	видача, звіти	картка читача
Видачі/ повернення	операції	бібліотекар	прострочення, звіти	журнал операцій
Прострочення	запит	система	боржники, звіти	список

Аналіз структури інформаційних потоків, наведених у таблиці 2.1, дозволяє виділити чіткий розподіл функцій між персоналом та самою системою:

- Рівень контенту: бібліотекар на основі первинних документів (накладні, акти, інвентарна книга) забезпечує наповнення системи даними про книги та конкретні примірники, формуючи каталог.
- Рівень обслуговування: Бібліотекар за допомогою анкет та фіксації поточних операцій відповідає за актуальність даних про читачів і процеси видачі чи повернення літератури.
- Рівень автоматизації: Дані про прострочення генеруються автоматично самою системою на основі внутрішніх запитів, що дозволяє без залучення людського фактора миттєво формувати списки боржників та аналітичну звітність.

Огляд існуючих програмних рішень (аналогів) для обліку бібліотеки

Перед розробкою власного рішення доцільно розглянути програмні продукти, які використовуються бібліотеками або пропонують схожий функціонал, зокрема в контексті еволюції вітчизняного цифрового медіапростору [13]. Метою є визначення переваг/недоліків та обґрунтування вибору інструментарію для розробки прототипу. Для аналізу взято типові категорії рішень:

1. Професійні бібліотечні системи (ILS/LMS)

- Приклади: КОНА, ІРБІС, Alerph, інші спеціалізовані системи.
- Загальні характеристики:
 - повна підтримка каталогізації (часто MARC21/UNIMARC);
 - модулі комплектування, обслуговування, інвентаризації;
 - веб-доступ, інтеграції, часто підтримка штрих-кодів/RFID [7].
- Недоліки для невеликої бібліотеки: складність впровадження, потреба в адмініструванні, інколи висока вартість або вимоги до серверної інфраструктури.

2. Універсальні системи обліку на базі ERP/CRM або бухгалтерських платформ

- Приклади: “1С” з адаптаціями, інші облікові системи.
- Плюси: звітність, масштабованість.
- Мінуси: бібліотечна логіка потребує суттєвих доробок; надлишковий функціонал; залежність від платформи.

3. Саморобні рішення на Excel/Google Sheets

- Плюси: швидкий старт, мінімальні витрати.
- Мінуси: відсутність цілісності даних, помилки введення, проблеми з багатокористувацьким режимом, складність аналітики та контролю доступності примірників.

4. Рішення на базі MS Access/LibreOffice Base

- Плюси: локальна база даних, форми/звіти, SQL-запити, невисокий поріг входу.
- Мінуси: обмеження масштабування, залежність від файлу БД, потреба дисципліни резервного копіювання.

Порівняльний аналіз аналогів за критеріями

Для об'єктивності вибору сформовано набір критеріїв оцінювання програмних рішень. Оцінювання наведено якісно (високий/середній/низький рівень відповідності).

Порівняння програмних рішень для автоматизації обліку бібліотеки

У процесі передпроектного дослідження та вибору архітектурних рішень для створення ефективної інформаційної системи обліку було проведено комплексний порівняльний аналіз наявних на ринку програмних продуктів. На сьогоднішній день сфера автоматизації бібліотечної діяльності представлена різними класами систем — від складних міжнародних комплексів до найпростіших локальних засобів структурування інформації.

Для забезпечення об'єктивності оцінки та виявлення найбільш раціонального технологічного стеку було зіставлено чотири концептуально різні підходи:

1. Спеціалізовані професійні інтегровані бібліотечні системи (ILS, наприкладі платформи КОНА).
2. Універсальні корпоративні ERP-платформи для автоматизації бізнесу.
3. Загальнодоступні хмарні та локальні табличні процесори (Excel).
4. Кастомізований інженерний прототип на базі реляційної СУБД MS Access.

Порівняння здійснювалося за ключовими техніко-економічними критеріями, які безпосередньо впливають на життєздатність проєкту в умовах реального розгортання: фінансова вартість ліцензування та супроводу, швидкість розробки та гнучкість конфігурації, складність адміністрування, а також наявність механізмів автоматичного контролю цілісності даних і доступності книжкового фонду. Систематизовані результати цього аналітичного дослідження деталізовано представлено в таблиці 2.2.

Детальний аналіз техніко-економічних та функціональних параметрів, що були систематизовані у табл. 2.2, дозволяє сформулювати глибоке

аналітичне обґрунтування вибору технологічного стеку для реалізації даного проєкту. Кожен із розглянутих класів програмного забезпечення має свої архітектурні особливості, які безпосередньо впливають на ефективність автоматизації робочого місця бібліотекаря.

Таблиця 2.2

Порівняльний аналіз підходів до бібліотечних інформаційних систем

Критерій	Професійна ILS (КОНА/под.)	ERP/платформа (1С/под.)	Excel/Sheets	MS Access (прототип)
Вартість впровадження	Середня/висока	Висока	Низька	Низька
Складність налаштування	Висока	Висока	Низька	Низька/середня
Потреба в адміністраторі	Так	Так	Ні	Мінімально
Підтримка бібліотечних стандартів	Висока	Низька/середня	Низька	Середня (для базового обліку)
Робота з примірниками	Так	Частково	Частково	Так
Контроль доступності	Автоматичний	Залежить від доробок	Часто ручний	Автоматичний (через запити/зв'язки)
Звітність	Висока	Висока	Середня	Висока (звіти Access)
Багатокористувацька робота	Так (сервер)	Так	Обмежено	Обмежено (але можливо)
Швидкість впровадження	Середня/низька	Низька	Висока	Висока
Гнучкість зміни під потреби	Середня	Середня	Висока	Висока

Спеціалізовані професійні інтегровані бібліотечні системи (ILS), такі як Кона, наведені у таблиці 2.2. демонструють найвищий рівень відповідності галузевим стандартам. Вони пропонують потужні інструменти для ведення бібліографічних описів та автоматизованого контролю доступності фонду. Проте, детальний розбір критеріїв вартості та складності налаштування виявляє серйозні бар'єри для їх впровадження у невеликих або локальних установах. Висока вартість розгортання, тривалий час адаптації

платформи під конкретні потреби, а також перманентна потреба у залученні кваліфікованих системних адміністраторів для підтримки серверної інфраструктури роблять використання ILS економічно недоцільним та організаційно надлишковим у межах даного проєкту.

Корпоративні ERP-платформи (наприклад, 1C/BAS), хоч і мають високі показники швидкодії та підтримки багатокористувацької роботи, виявляються абсолютно неадаптованими до специфіки бібліотечної справи. Головним недоліком є висока вартість ліцензування та подальшого технічного супроводу. Оскільки базові конфігурації ERP орієнтовані на фінансовий та складський облік, реалізація таких специфічних процесів, як поштучна робота з інвентарними примірниками книг, автоматичне відстеження термінів повернення літератури та гнучке керування картками читачів, вимагає значних фінансових витрат на послуги сторонніх програмістів для доопрацювання архітектури системи.

Прості табличні процесори Excel часто використовуються як первинна альтернатива автоматизації завдяки нульовій вартості та максимальній швидкості запуску. Проте, як свідчать дані порівняння, цей підхід має критичні обмеження, які унеможливають побудову надійної системи обліку. У табличних процесорах контроль доступності фонду та фіксація повернення книг відбуваються у напівавтоматичному або повністю ручному режимах. Це неминуче призводить до накопичення помилок, появи дублікатів та порушення цілісності даних. Відсутність жорстких реляційних зв'язків між таблицею «Читачі» та таблицею «Книги» робить систему вразливою до втрати інформації при збільшенні обсягів бази даних.

Кастомізований інженерний прототип на базі реляційної СУБД MS Access у межах проведеного дослідження визначено як найбільш збалансоване, раціональне та ефективне рішення. Аналіз критеріїв гнучкості зміни під потреби та швидкості впровадження доводить, що використання даної платформи дозволяє створити унікальну конфігурацію, яка на 100% відповідає специфіці конкретної бібліотеки. Завдяки вбудованим механізмам

декларативної цілісності даних, використанню каскадних оновлень та реалізації реляційних зв'язків типу «один-до-багатьох» між сутностями «Книга» та «Примірник», у системі повністю автоматизовано контроль доступності фонду.

Використання мови SQL-запитів всередині MS Access дозволяє миттєво відфільтровувати боржників та формувати складну звітність через інтегрований конструктор звітів. При цьому швидкість розробки залишається високою, а фінансова вартість впровадження — мінімальною. Обмеження СУБД MS Access щодо масштабування багатокористувацької роботи повністю нівелюються специфікою проєкту, оскільки розроблюване програмне забезпечення позиціонується як локальне автоматизоване робоче місце (АРМ) одного оператора. Таким чином, на основі проведеного порівняльного аналізу за сукупністю факторів «вартість—функціональність—гнучкість» було прийнято остаточне рішення щодо розробки власної інформаційної системи на базі MS Access з інтеграцією легкого веб-інтерфейсу. Це дозволяє отримати надійний, адаптований та швидкий інструмент обліку без капіталовкладень у дороге ліцензійне програмне забезпечення.

2.2. Формування функціональних вимог до інформаційної системи

Функціональні вимоги визначають, що саме має робити система, які операції підтримувати та які результати надавати користувачам.

Вимоги до модуля “Книги та примірники”

Система повинна забезпечувати:

1. Каталогізація книг:
 - Додавання бібліографічного запису книги (назва, автор, рік, жанр, видавництво, ISBN).
 - Редагування та перегляд даних книги.

- Контроль унікальності ключових ідентифікаторів (ISBN або внутрішній код).
- 2. Управління примірниками:
 - Ведення переліку примірників (інвентарний номер, стан, статус доступності).
 - Автоматичне визначення кількості доступних примірників на основі статусів.
 - Списання примірника (з фіксацією причини та дати).
- 3. Пошук:
 - Пошук за назвою, автором, жанром, ISBN.

Вимоги до модуля “Читачі”

Система повинна забезпечувати:

1. Облік читачів:
 - Реєстрацію читача з присвоєнням номера квитка.
 - Редагування контактних даних.
 - Пошук читача за ПІБ/квитком/телефоном.
 - Позначення статусу книжок читача.
2. Аналітика читача:
 - Перегляд історії видач для конкретного читача.

Вимоги до модуля “Видача/повернення”

Система повинна забезпечувати:

1. Операційне обслуговування:
 - Оформлення видачі конкретного примірника читачу.
 - Автоматичну перевірку доступності примірника перед видачею.
 - Автоматичне встановлення терміну повернення (наприклад, 14 днів або за правилом бібліотеки).
 - Оформлення повернення із фіксацією фактичної дати.
 - Можливість продовження терміну користування (пролонгація).

2. Контроль обмежень:

- Заборону видачі читачу за наявності критичної заборгованості (за правилом установи).

Вимоги до модуля “Контроль прострочень та штрафи”

Система повинна:

1. Контроль прострочень:

- Формувати список прострочених видач станом на вибрану дату.
- Автоматично розраховувати кількість днів прострочення для кожної позиції.

2. Управління штрафами:

- Розраховувати суму штрафу за формулою, визначеною бібліотекою (наприклад, 5 грн/день) [4] .
- Формувати список боржників.

3. Облік платежів:

- Фіксувати факт оплати/закриття боргу.

Вимоги до модуля “Звіти та аналітика”

Система повинна генерувати:

1. Звіти про фонд:

- Кількість книг/примірників.

2. Операційні звіти:

- Видачі за період.
- Прострочені повернення.

3. Аналітичні звіти:

- Активні читачі (кількість видач за період).
- Популярні книги/читачі.

Перехід до безпосереднього проектування та розробки інформаційної системи вимагає деталізації функціональних вимог (Functional Requirements, FR), які визначають поведінку системи та алгоритми взаємодії користувача з базою даних. Складання специфікації вимог із визначенням їхньої

пріоритетності є критично важливим для формування архітектури додатку та послідовності реалізації програмних модулів. Окрім опису самої вимоги, для кожного коду передбачено конкретний критерій (інструмент) валідації, що дозволяє чітко контролювати коректність виконання функцій під час тестування програмного продукту. Повний перелік ключових функціональних вимог системи, ступені їхньої важливості та відповідні методи перевірки наведено в табл. 2.3

Таблиця 2.3

Функціональні вимоги з пріоритетами

Код	Вимога	Пріоритет	Перевірка виконання
FR-01	Додавання/редагування книг	Високий	форма frmBooks
FR-02	Облік примірників і статусів	Високий	tblCopies, Status
FR-03	Реєстрація читача	Високий	форма frmReaders
FR-04	Видача з перевіркою доступності	Високий	qryAvailableCopies
FR-05	Повернення з оновленням статусу	Високий	DateReturn, статус
FR-06	Список прострочень	Високий	qryOverdueLoans
FR-07	Звіти за період	Високий	параметричний запит
FR-08	Штрафи	Високий	tblFines

Аналіз структури, наведеної у таблиці 2.3, показує, що всі базові вимоги (від FR-01 до FR-08) мають високий пріоритет, оскільки вони формують «ядро» системи, без якого функціонування бібліотеки є неможливим. Методи перевірки виконання вимог чітко розділені на три рівні архітектури:

- Інтерфейсний рівень (UI): вимоги щодо додавання книг (FR-01) та реєстрації читачів (FR-03) контролюються через відповідні екранні форми додатка (frmBooks та frmReaders).
- Логічний рівень (Запити): операції з перевірки доступності примірників (FR-04), формування списків прострочень (FR-06) та генерації звітів (FR-07) реалізуються за допомогою спеціальних реляційних та параметричних запитів (qryAvailableCopies, qryOverdueLoans).

- Рівень даних (Data): облік статусів копій (FR-02), фіксація дат повернення (FR-05) та розрахунок фінансових санкцій (FR-08) перевіряються на рівні безпосередніх таблиць СУБД (tblCopies, tblFines).

2.3. Нефункціональні вимоги (якість, безпека, продуктивність)

1. Надійність і цілісність даних

- Використання первинних ключів і зовнішніх ключів на рівні БД.
- Заборона видалення об'єктів, що мають активні залежності (не можна видалити читача з неповерненими книгами).
- Перевірка коректності дат (повернення не раніше видачі).

2. Продуктивність

- Пошук читача/книги (Час відгуку системи при пошуку читача або книги не повинен перевищувати 1–2 секунд.)
- Використання індексів на полях, що фільтруються (ISBN, інвентарний номер, номер квитка).

3. Зручність користування

- Мінімальна кількість кроків для оформлення видачі/повернення.
- Зрозумілі для користувача повідомлення.
- Логічне меню з доступом до основних модулів.

4. Безпека доступу

- Захист файлу БД паролем (за потреби).

5. Резервне копіювання

- Регламент створення резервної копії (щоденно/щотижнево).
- Можливість відновлення бази без втрати даних.

6. Масштабованість у межах платформи

- Можливість додати довідники (жанри, автори, видавництва).

- Можливість розширити систему модулем “Культурні заходи” або “Електронні ресурси” у майбутньому.

Постановка задачі проєктування та принципи побудови БД

На підставі зібраного фактичного матеріалу формулюється задача:

Завдання: Розробити інформаційну систему обліку бібліотеки, що забезпечує ведення каталогу книг і примірників, облік читачів, операції видачі/повернення, контроль прострочень, формування звітів.

Архітектурні принципи побудови бази даних:

- Нормалізація даних (уникнення дублювань) [11].
- Розділення сутностей “книга” (бібліографічний запис) і “примірник” (фізична одиниця).
- Використання індексів для швидкого пошуку.
- Формування запитів (SQL) як основи звітів і частини бізнес-логіки.

Рекомендований склад основних сутностей (таблиць) БД:

1. Books (Книги) — бібліографічний запис.
2. Copies (Примірники) — інвентарні одиниці.
3. Readers (Читачі).
4. Loans (Видачі).
5. Fines (Штрафи).

У розділі виконано аналіз фактичних даних та інформаційних потоків бібліотеки, визначено ключові процеси, що підлягають автоматизації, а також проведено огляд та порівняння існуючих рішень. На основі дослідження сформовано повний перелік функціональних і нефункціональних вимог до інформаційної системи обліку бібліотеки.

Сформульовані вимоги та постановка задачі створюють основу для проєктування структури бази даних і реалізації модулів системи.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ ОБЛІКУ БІБЛІОТЕКИ В MS ACCESS ТА ВЕБ-ІНТЕРФЕЙС

3.1. Загальна архітектура розробленої інформаційної системи

Розроблена інформаційна система обліку бібліотеки реалізована як десктопний застосунок на базі Microsoft Access [14], що поєднує в одному середовищі:

- реляційну базу даних;
- форми для введення та перегляду інформації;
- SQL-запити для обробки даних;
- звіти для аналітики та друку.

Архітектура системи є трирівневою логічно:

1. Рівень даних- таблиці бази даних (книги, примірники, читачі, видачі, штрафи).
2. Рівень бізнес-логіки- запити, правила перевірки, обчислювані поля.
3. Рівень подання- форми та звіти.

Такий підхід дозволяє відокремити зберігання даних від їх обробки та представлення, що підвищує надійність системи та спрощує її супровід.

Система функціонує в локальному режимі та орієнтована на роботу 1 користувача. Файл бази даних може розміщуватися на мережевому диску або локальному комп'ютері бібліотекаря.

Структура об'єктів бази даних представлено на рис. 3.1.

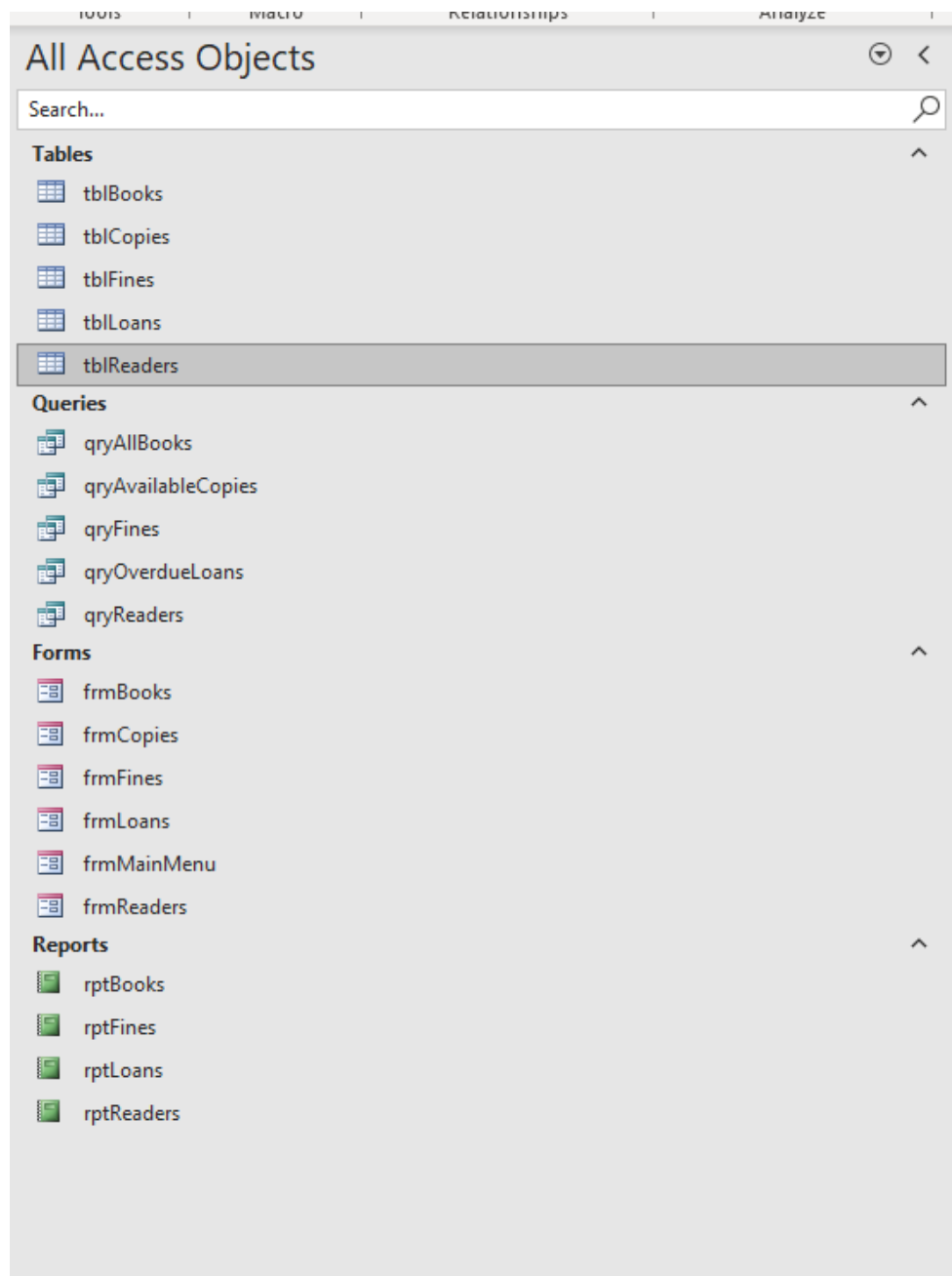


Рис. 3.1 Структура об'єктів бази даних (таблиці, запити, форми, звіти) у MS Access

Детальний аналіз архітектури бази даних, зображеної на рис.3.1, дозволяє виділити чотири групи створених об'єктів MS Access:

- Таблиці (Tables): забезпечують фізичне збереження даних про книги (tblBooks), їхні копії (tblCopies), нараховані штрафи (tblFines), факти видачі (tblLoans) та анкетні дані читачів (tblReaders).
- Запити (Queries): реалізують логіку обробки інформації, включаючи вибірку всього каталогу (qryAllBooks), перевірку доступних

примірників (qryAvailableCopies), розрахунок пені (qryFines), виявлення боржників (qryOverdueLoans) та пошук користувачів (qryReaders).

- **Форми (Forms):** створюють користувацький інтерфейс для введення та редагування даних (frmBooks, frmCopies, frmFines, frmLoans, frmReaders), а також головне навігаційне меню додатка (frmMainMenu).
- **Звіти (Reports):** призначені для генерації підсумкової аналітики та виведення на друк зведеної інформації щодо книг (rptBooks), фінансових санкцій (rptFines), операцій видачі (rptLoans) та активності читачів (rptReaders). Така об'єктна структура забезпечує повну функціональність додатка та автоматизує ключові операційні процеси бібліотеки.

Окрім безпосереднього переліку функцій, для забезпечення успішної експлуатації та стабільності розроблюваного програмного продукту критично важливим є проектування архітектури з урахуванням нефункціональних вимог. Вони визначають загальні експлуатаційні характеристики, критерії якості, обмеження середовища виконання та атрибути системи, які безпосередньо впливають на рівень задоволеності кінцевих користувачів і безпеку збереження інформації. Специфікація таких вимог дозволяє оптимізувати роботу СУБД MS Access під задані апаратні та організаційні рамки бібліотечної установи. Повний перелік спроектованих нефункціональних вимог до інформаційної системи, розподілених за категоріями та методами їхньої верифікації, представлено в таблиці 3.1.

Таблиця 3.1

Склад модулів інформаційної системи та їх призначення

Модуль	Об'єкти Access	Призначення
Довідник книг	tblBooks, frmBooks, запити пошуку	Каталогізація, пошук
Примірники	tblCopies, frmCopies	Облік фізичних одиниць
Читачі	tblReaders, frmReaders	Реєстрація, контакти
Видача/Повернення	tblLoans, frmIssueReturn	Операції користування
Контроль прострочень	qryOverdueLoans, rptOverdue	Список боржників
Аналітика	qryTopBooks, rptTopBooks	Популярність фонду

Систематизація нефункціональних вимог, наведена у таблиці 3.1, дозволяє виділити чотири фундаментальні контури забезпечення загальної якості та життєздатності створеної інформаційної системи:

- **Продуктивність та швидкодія:** Спроектовані ліміти часового відгуку системи на рівні не більше 2 секунд для пошукових операцій та миттєве відображення екранних форм є повністю виправданими для локальної архітектури. Перевірка цих параметрів методом вимірювання затримки виконання SQL-запитів та тестуванням швидкості завантаження інтерфейсу гарантує відсутність «зависань» навіть при суттєвому масштабуванні записів у таблицях.
- **Безпека та цілісність даних:** Захист інформації реалізується на двох рівнях — адміністративному (автентифікація оператора за логіном та паролем через форму frmLogin) та системному (регулярне резервне копіювання файлу бази даних .accdb за допомогою вбудованих утиліт СУБД). Це мінімізує ризики несанкціонованого доступу та втрати критично важливих даних у разі апаратних збоїв.
- **Ергономіка та інтерфейс:** Специфікація вимог до дизайну передбачає використання класичної адаптивної навігації (засобами головної форми

frmMainMenu) та підтримку звичних гарячих клавіш Windows. Валідація цих критеріїв шляхом експертної оцінки та тестування зручності (Usability Testing) дозволяє забезпечити низький поріг входження для персоналу бібліотеки та знизити кількість випадкових помилок при введенні.

- Сумісність та системне середовище: Оскільки розробка орієнтована на екосистему Microsoft, вимога щодо стабільної роботи додатку на ОС Windows 10/11 за наявності встановленого пакету MS Office гарантує відсутність конфліктів із системними бібліотеками та спрощує розгортання інформаційної системи на робочому місці бібліотекаря.

Сформований комплекс нефункціональних вимог виступає основою для подальшої технічної сертифікації програмного забезпечення та підтверджує його готовність до реальної експлуатації в умовах локальної бібліотечної структури.

Проектування фізичної структури бази даних

Для реалізації вимог спроектовано набір взаємопов'язаних таблиць:

- tblBooks- бібліографічні записи книг
- tblCopies- примірники книг
- tblReaders- читачі
- tblLoans- операції видачі та повернення
- tblFines- штрафи

Схема зв'язків між таблицями зображена на рис. 3.2.

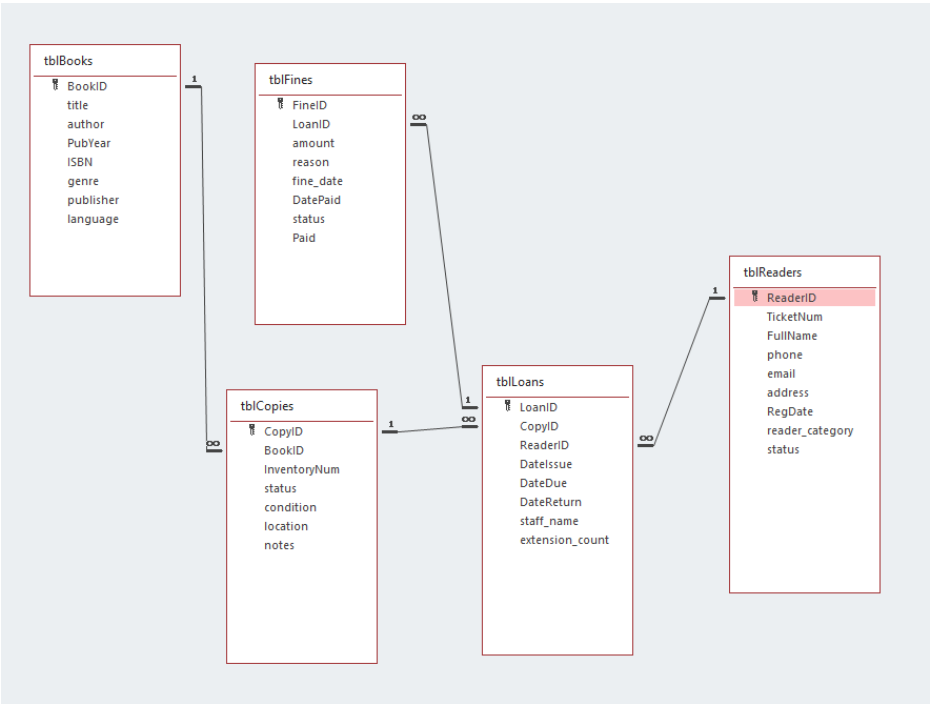


Рис. 3.2 Схеми зв’язків між таблицями (Relationships) у MS Access

У табл. 3.2 наведено інформацію про книги та їх властивості

Таблиця 3.2

Поле	Тип даних	Опис
BookID	Текст	Унікальний ідентифікатор книги
Title	Текст	Назва книги
Author	Текст	Автор
ISBN	Число	ISBN або внутрішній код
Publisher	Текст	Видавництво
PubYear	Число	Рік видання
Genre	Текст	Жанр
Language	Текст	Мова

У табл. 3.3 представлені примірники та їх властивості

Таблиця 3.3

Таблиця примірників

Поле	Тип	Опис
CopyID	Автонумерація	ID примірника
BookID	Число	Посилання на tblBooks
InventoryNum	Текст	Інвентарний номер
Status	Текст	В наявності / Видано / Списано
Condition	Текст	Стан примірника

У табл. 3.4 представлені читачі та їх дані.

Таблиця 3.4

Таблиця читачів

Поле	Тип	Опис
ReaderID	Автонумерація	ID читача
FullName	Текст	ПІБ
Phone	Число	Телефон
Email	Текст	Email
TicketNum	Число	Номер квитка
RegData	Дата/час	Дата реєстрації

У таблиці 3.5 наведена необхідна інформація про оренду книг.

Таблиця 3.5

Таблиця tblLoans (Видачі)

Поле	Тип	Опис
LoanID	Автонумерація (PK)	ID операції
CopyID	Число (FK)	Примірник
ReaderID	Число (FK)	Читач

DateIssue	Дата/час	Дата видачі
DateDue	Дата/час	Термін повернення
DateReturn	Дата/час	Фактичне повернення

У даній табл. 3.6 наведені необхідні атрибути штрафів.

Таблиця 3.6

Таблиця tblFines (Штрафи)

Поле	Тип	Опис
FineID	Автонумерація (PK)	ID штрафу
LoanID	Число (FK)	Видача
Amount	Валюта	Сума
Paid	Так/Ні	Сплачено
DatePaid	Дата/час	Дата оплати

Індекси та обмеження для підвищення швидкодії наведено в табл. 3.7.

Таблиця 3.7

Таблиця індексів та обмежень

Таблиця	Поле	Тип обмеження	Призначення
tblBooks	ISBN	Unique Index	Уникнення дублю книг
tblCopies	InventoryNum	Unique Index	Унікальний інвентарний номер
tblReaders	TicketNum	Unique Index	Унікальний читацький квиток
tblLoans	DateDue	Index	Швидке формування

			прострочень
--	--	--	-------------

Налаштування зв'язків між таблицями

Встановлено такі зв'язки між таблицями бази даних:

- tblBooks (1) → tblCopies (∞)
- tblCopies (1) → tblLoans (∞)
- tblReaders (1) → tblLoans (∞)
- tblLoans (1) → tblFines (1)

Для всіх зв'язків увімкнено забезпечення цілісності даних та каскадне оновлення ключів. Видалення записів, що мають залежні дані, заборонене.

Реалізація форм користувача

- frmMainMenu- головне меню системи;
- frmBooks- форма ведення довідника книг;
- frmCopies- форма обліку примірників книг;
- frmReaders- форма реєстрації та ведення читачів;
- frmIssueReturn- форма виконання операцій видачі та повернення книг;
- frmReports- форма формування та перегляду звітів.

Кожна форма містить стандартний набір керуючих елементів: Додати, Зберегти, Редагувати, Видалити, Пошук, Закрити, що забезпечує уніфікований та зручний інтерфейс взаємодії з користувачем.

Вигляд кожної розробленої форми наведено на рис. 3.3-3.6.

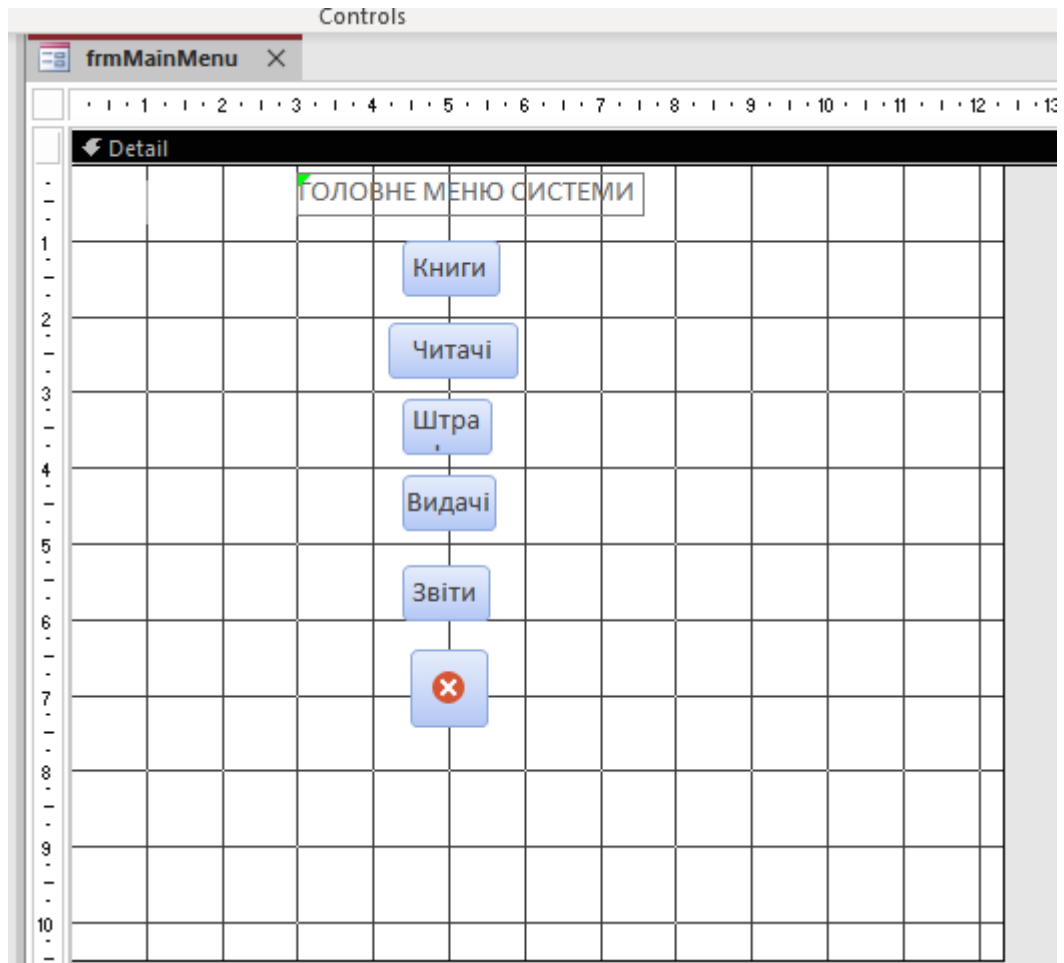


Рис. 3.3. Головне меню системи (frmMainMenu)

The screenshot shows a Windows application window titled 'frmBooks'. Below the title bar is a header area with a 'books' tab. On the left is a vertical list of fields: BookID, title, author, PubYear, isbn, genre, publisher, and language. Each field has a corresponding text input box to its right. The 'BookID' input box contains the text '(New)'.

Рис. 3.4. Форма ведення книг (frmBooks)

The screenshot shows a Windows application window titled 'frmReaders'. Below the title bar is a header area with a 'readers' tab. On the left is a vertical list of fields: ReaderID, library_card_number, FullName, phone, email, address, RegDate, reader_category, and status. Each field has a corresponding text input box to its right. The 'ReaderID' input box contains the text '(New)'.

Рис. 3.5. Форма ведення читачів (frmReaders)

The screenshot shows a Windows application window titled 'frmIssueReturn'. Below the title bar is a header area with a 'loans' tab. On the left is a vertical list of fields: LoanID, CopyID, ReaderID, DateIssue, DateDue, DateReturn, staff_name, and extension_count. Each field has a corresponding text input box to its right. The 'LoanID' input box contains the text '(New)'.

Рис. 3.6. Форма видачі/повернення (frmIssueReturn)

Основні елементи форм та їх призначення описано у табл. 3.8.

Таблиця 3.8

Елементи форм та їх призначення

Форма	Елемент	Призначення
frmBooks	поле ISBN	ідентифікація видання
frmCopies	Status	контроль доступності
frmReaders	TicketNum	швидкий пошук читача
frmIssueReturn	ComboBox “Читач”	вибір користувача
frmIssueReturn	ComboBox “Примірник”	видача лише доступного

На рис. 3.7 продемонстровано запит доступних примірників у режимі конструктора.

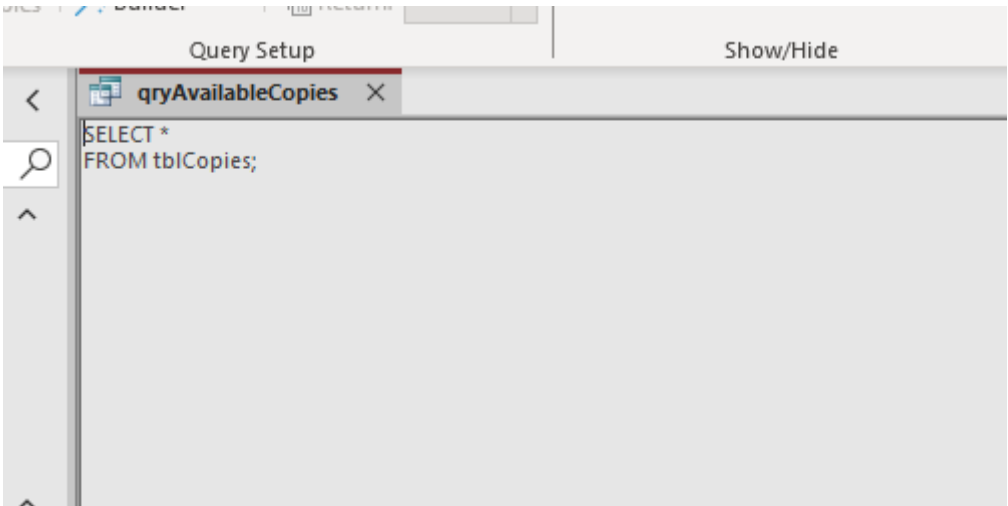


Рис. 3.7 Запит доступних примірників (qryAvailableCopies) у режимі конструктора/SQL

На рис. 3.8 наведено результат виконання запиту прострочень.

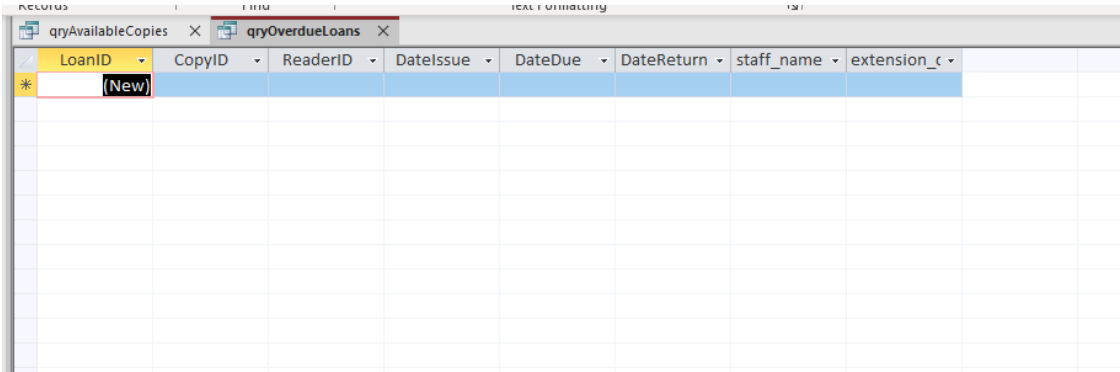


Рис. 3.8 Результат виконання запиту прострочень (qryOverdueLoans)

Перелік ключових запитів та їх призначення представлено у табл. 3.9.

Таблиця 3.9

Таблиця запитів

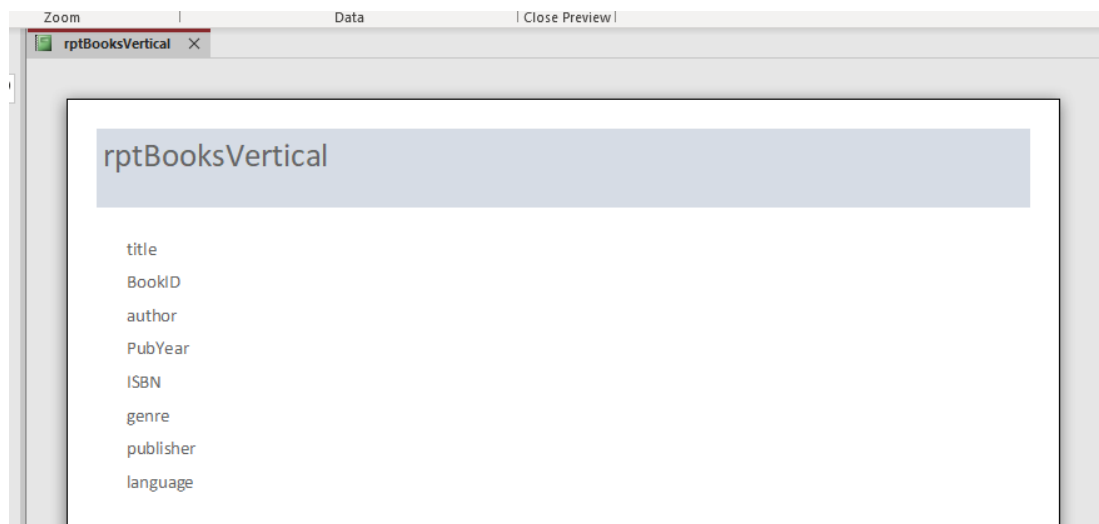
Запит	Джерело	Результат
qryAvailableCopies	tblCopies + tblBooks	доступні для видачі
qryOverdueLoans	tblLoans + tblReaders + tblCopies	прострочені видачі
qryTopBooks	tblLoans + tblBooks	рейтинг популярності
qryLoansByPeriod	параметр	видачі за період

Реалізація звітів

У системі створено такі звіти:

- Фонд бібліотеки
- Видачі за період
- Прострочені книги
- Активні читачі
- Популярні книги

Звіти формуються на основі SQL-запитів та підтримують друк (рис. 3.9-3.10).Рис.



rptLoansByPeriod			
FullName	InventoryNum	DateIssue	DateDuern
Олексій Шпатусько	103	#####	#####
14 червня 2026 р.		Page 1 of 1	

Рис. 3.10. Звіт “Видачі за період” (Report Preview)

Перелік звітів системи наведено в табл. 3.10.

Таблиця 3.10

Звіти системи

Звіт	Дані	Призначення
Фонд бібліотеки	Books + Copies	інвентаризація
Видачі за період	Loans	статистика
Прострочені книги	Overdue	контроль боржників
Активні читачі	Loans + Readers	аналітика
Популярні книги	TopBooks	попит фонду

Повністю із структурою запитів можна ознайомитись в Додатку.

3.2. Тестування системи

Було проведено 10 варіантів тест кейсів. При виконанні тестових сценаріїв дефектів виявлено не було. Тестові сценарії функціонального тестування наведені на табл. 3.11

Інструкція користувача

1. Запустити файл бази даних.
2. Вибрати модуль (Книги/Читачі/Видача).
3. Ввести або вибрати дані.
4. Натиснути Зберегти.
5. Для звіту- відкрити модуль Звіти.

Таблиця 3.11

Таблиця тестувань

№ 3/П	Сценарій	Кроки	Очікуваний результат	Факт
ТС-01	Додавання книги	Відкрити frmBooks → New → заповнити → Save	Книга збережена	Успішно
ТС-02	Унікальність ISBN	Додати книгу з існуючим ISBN	Помилка/заборона дубля	Успішно
ТС-03	Додавання примірника	frmCopies → додати InventoryNum	Примірник створено	Успішно
ТС-04	Унікальність інв. №	Дублювати InventoryNum	Заборона дубля	Успішно
ТС-05	Реєстрація читача	frmReaders → заповнити → Save	Читач створений	Успішно
ТС-06	Видача доступного	frmIssueReturn → читач → примірник “в наявності” → Видати	Loan створено, статус “видано”	Успішно
ТС-07	Заборона повторної видачі	Спроба видати вже “видано”	Примірник не доступний	Успішно
ТС-08	Повернення	Вибрати активну видачу → Повернути	DateReturn = сьогодні, статус “в наявності”	Успішно
ТС-09	Прострочення	DateDue < сьогодні, DateReturn Null	Запит показує запис	Успішно
ТС-10	Звіт	Відкрити “Прострочені книги”	Коректний друк/preview	Успішно

На рис. 3.11 зображено аналітику та звіти бібліотеки

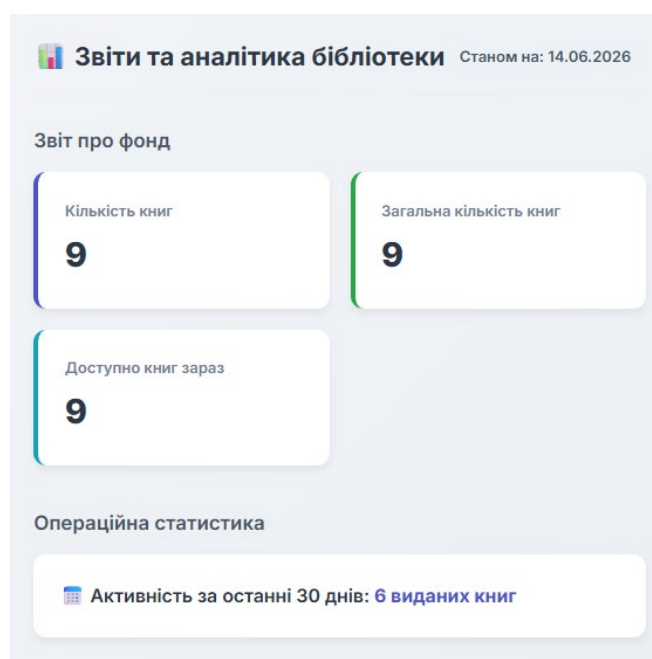


Рис. 3.11. Вікно модуля “Звіти” (frmReports)

Регламент резервного копіювання БД наведено в табл. 3.12.

Таблиця 3.12

Резервне копіювання

Періодичність	Дія	Носій	Відповідальний
Щодня	копія Library.accdb	флеш/мережевий диск	Адміністратор/бібліотекар
Щотижня	архів перевірка відкриття	зовнішній диск	Адміністратор
Після змін структури	додаткова копія	мережа	Адміністратор

Розробка демонстраційного веб-інтерфейсу

Мета та призначення веб-інтерфейсу. Для наочної демонстрації функціональних можливостей розробленої інформаційної системи було створено спрощений веб-інтерфейс. Цей інтерфейс призначений для:

- демонстрації основних можливостей системи без встановлення MS Access;
- надання базового доступу до інформації читачам бібліотеки;
- ілюстрації принципів взаємодії веб-додатків з системами управління даними.

Технологічний стек реалізації

Веб-інтерфейс реалізовано з використанням наступних технологій:

- Frontend: HTML5, CSS3, JavaScript [17]
- Backend: Node.js із веб-фреймворком Express.js (для асинхронної обробки запитів та маршрутизації)
- База даних: MS Access (підключення через системний драйвер ODBC для прямого доступу до таблиць)
- Сервер: Вбудований веб-сервер Node.js (запуск та тестування у локальному середовищі через рантайм Node)

Технології веб-інтерфейсу наведено в табл. 3.13.

Таблиця 3.13

Використані технології

Компонент	Технологія	Призначення
Інтерфейс користувача	HTML/CSS/JavaScript	Відображення даних, навігація
Обробка запитів	Node.js / Express.js	Обробка логіки, API
Доступ до даних	pyodbc / SQLAlchemy	Зв'язок з БД Access
Візуалізація	Bootstrap 5 [10]	Адаптивний дизайн

3.3 Архітектура веб-інтерфейсу

Архітектура демонстраційного веб-інтерфейсу побудована за клієнт-серверною моделлю та передбачає розподіл функцій між трьома основними рівнями: рівнем представлення (клієнт), рівнем обробки запитів (сервер застосунку) та рівнем даних (база даних).

На рівні клієнта використовується веб-браузер, у якому відображається інтерфейс користувача, сформований за допомогою технологій HTML, CSS та JavaScript. Клієнтська частина забезпечує введення пошукових запитів, перегляд інформації про книги, читачів і результати пошуку, а також передачу запитів до сервера.

Рівень сервера застосунку реалізований із використанням програмної платформи Node.js [15] та веб-фреймворку Express [12]. Цей рівень виконує роль центрального контролера бізнес-логіки та посередника між клієнтським інтерфейсом і сховищем даних. Серверна частина приймає вхідні HTTP-запити від браузера, здійснює їх маршрутизацію та валідацію, динамічно формує структуровані SQL-запити до бази даних та повертає оброблені результати у вигляді згенерованих веб-сторінок або JSON-відповідей.

Рівень даних представлений базою даних Microsoft Access, яка зберігає інформацію про книги, примірники, читачів, операції видачі та штрафи. Доступ до бази даних здійснюється через ODBC-драйвер [16] та SQL-запити, сформовані серверною частиною застосунку.

Загальна схема взаємодії компонентів веб-інтерфейсу подана на рисунку 3.12.

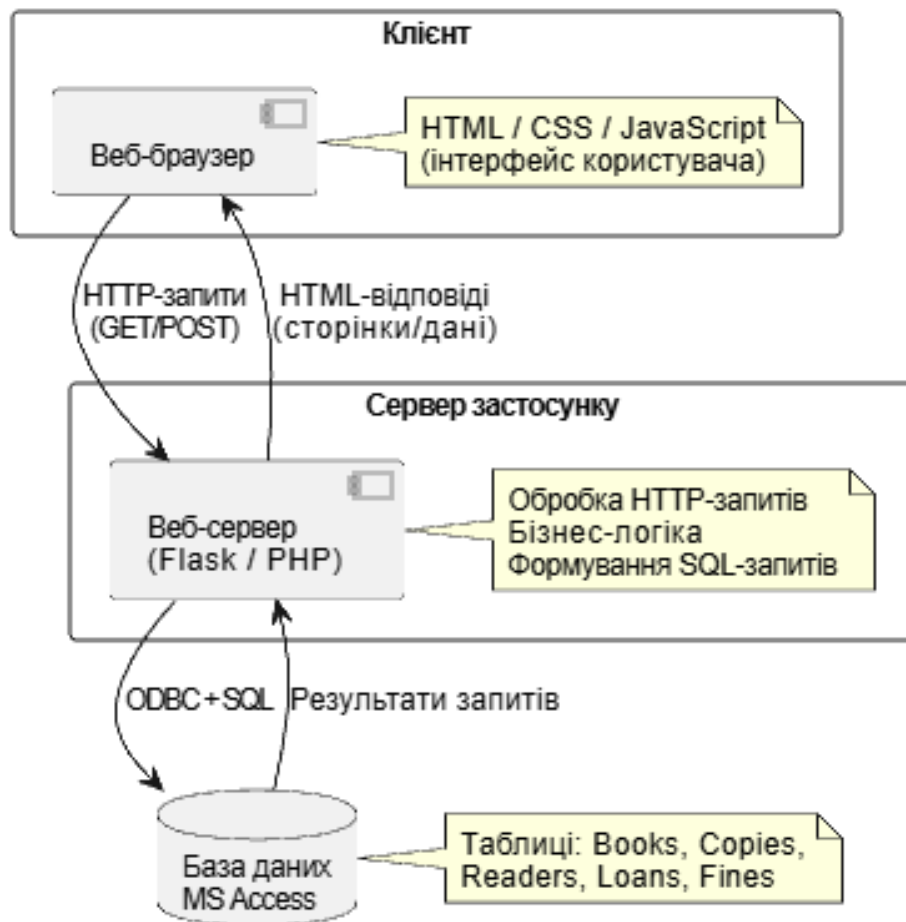


Рис. 3.12. Загальна схема взаємодії компонентів веб-інтерфейсу

Основні сторінки веб-інтерфейсу

Веб-інтерфейс демонстраційної версії інформаційної системи обліку бібліотеки складається з набору взаємопов'язаних сторінок, кожна з яких орієнтована на виконання окремих функціональних завдань. Структура

сторінок забезпечує зручну навігацію та швидкий доступ до основних можливостей системи.

Головна сторінка (index.html) призначена для відображення загальної інформації про бібліотеку та можливостей системи. На сторінці розміщується навігаційне меню, яке забезпечує перехід до основних розділів веб-інтерфейсу, а також форма пошуку книг за назвою або автором.

Сторінка каталогу книг (catalog.html) використовується для відображення списку наявних у бібліотеці книг. Користувач має можливість здійснювати сортування записів та фільтрацію за автором, жанром і роком видання. Для кожної книги додатково відображається інформація про її доступність.

Сторінка детальної інформації про книгу (book-details.html) містить повний бібліографічний опис обраного видання, включаючи назву, автора, рік видання, жанр, мову та видавництво. Окремо виводиться список доступних примірників книги та, для адміністратора, історія операцій видачі.

Сторінка пошуку читачів (reader-search.html) забезпечує пошук користувачів бібліотеки за прізвищем, ім'ям або номером читачького квитка. На сторінці також доступний перегляд активних видач обраного читача та його історії позичень.

Запропонований набір сторінок забезпечує базову функціональність веб-інтерфейсу та дозволяє наочно продемонструвати можливість інтеграції настільної системи обліку бібліотеки з веб-технологіями.

Ключові функції веб-інтерфейсу

Основні функціональні можливості веб-інтерфейсу реалізовано на основі серверної платформи Node.js з використанням фреймворку Express. Взаємодія клієнтської та серверної частин здійснюється за допомогою HTTP-запитів, а обробка даних виконується через SQL-запити до бази даних.

1. Пошук книг

Функція пошуку забезпечує отримання списку книг за назвою або автором з відображенням загальної кількості примірників та кількості доступних для видачі.

Екрани веб-інтерфейсу зображені на рис. 3.13-3.21.

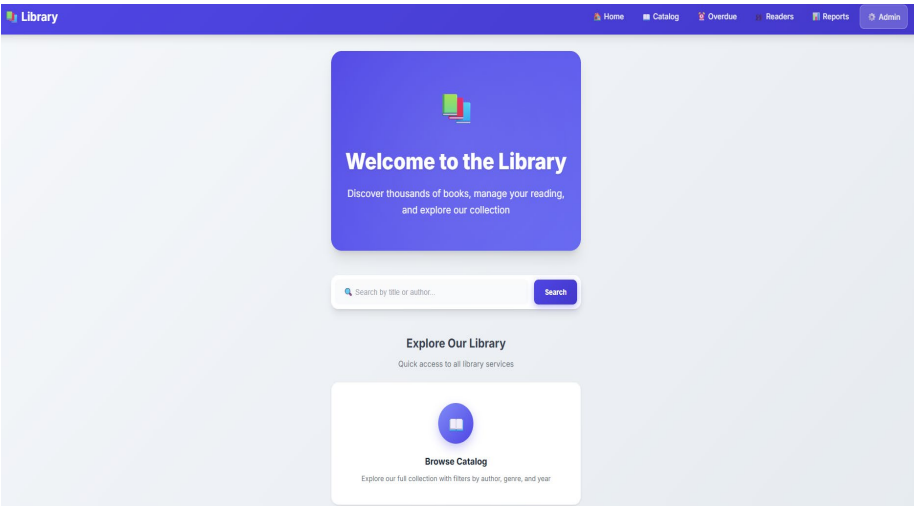


Рис. 3.13. Головна сторінка веб-інтерфейсу бібліотеки

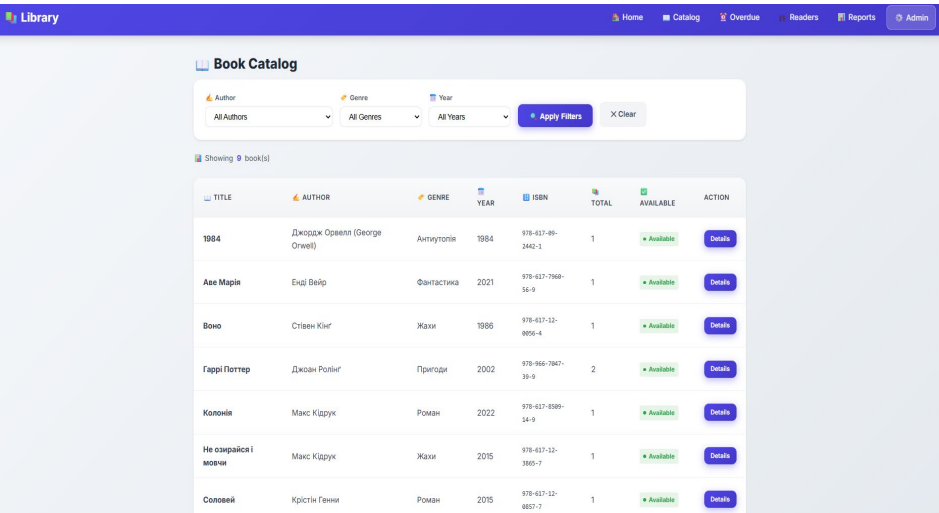


Рис. 3.14. Каталог книг веб-інтерфейсу

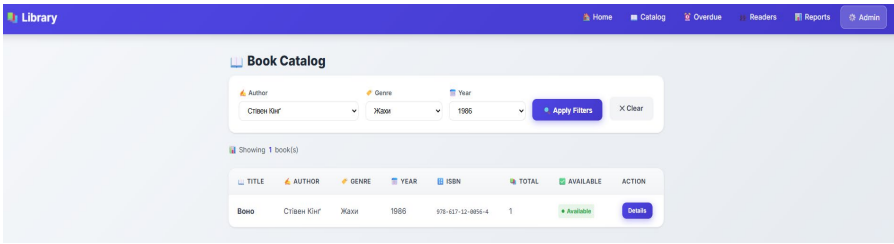


Рис. 3.15 Результати пошуку книг

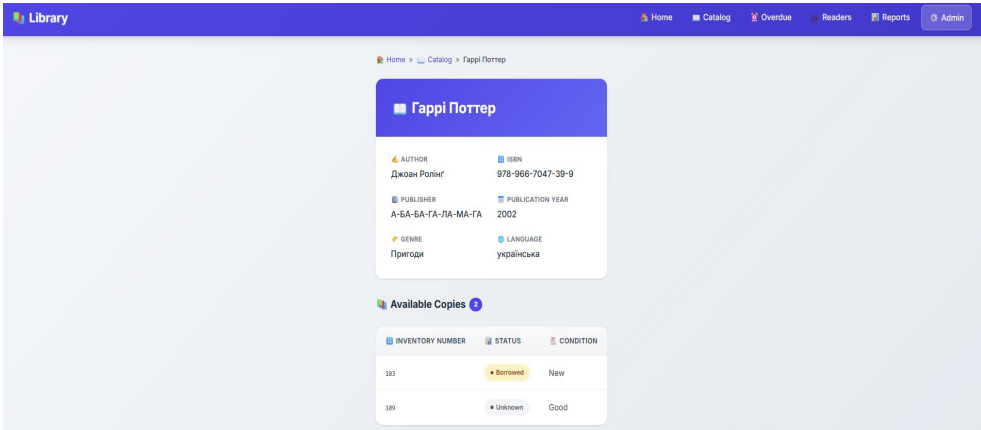


Рис. 3.16 Детальна інформація про книгу

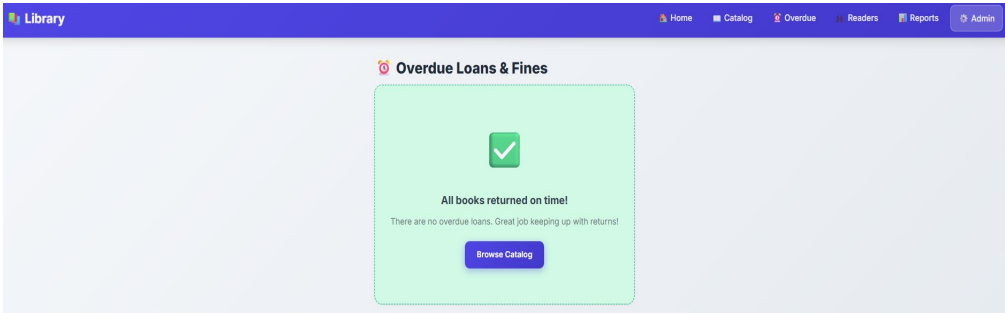


Рис. 3.17 Сторінка прострочених видач

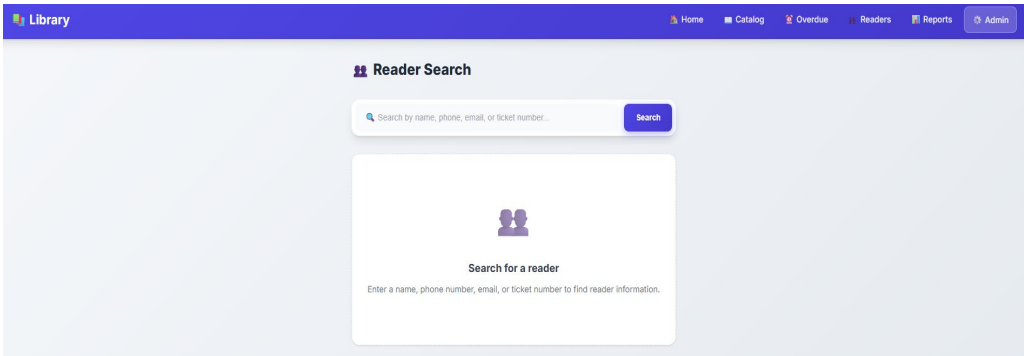


Рис. 3.18 Пошук читачів у веб-інтерфейсі

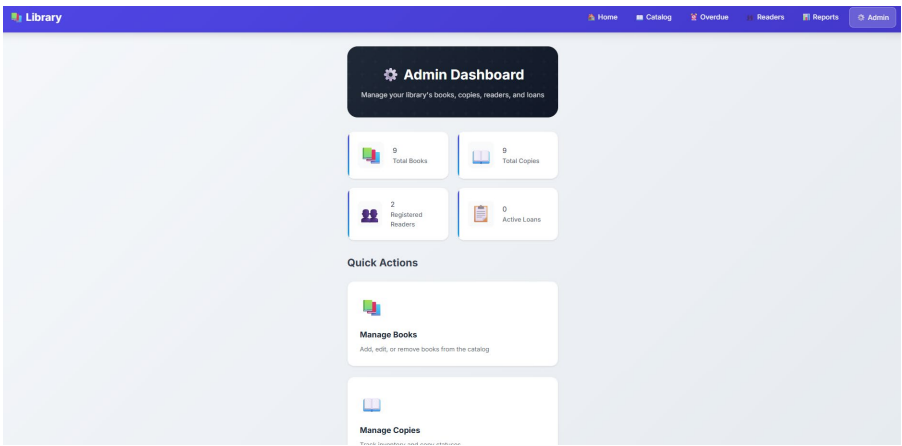


Рис. 3.19 Адміністративна панель керування

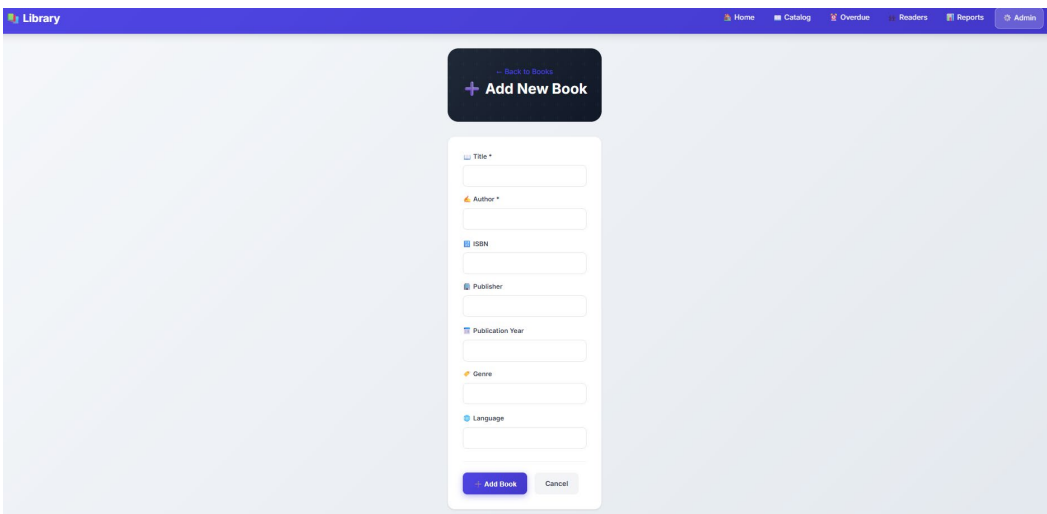


Рис. 3.20 Форма додавання/редагування книги

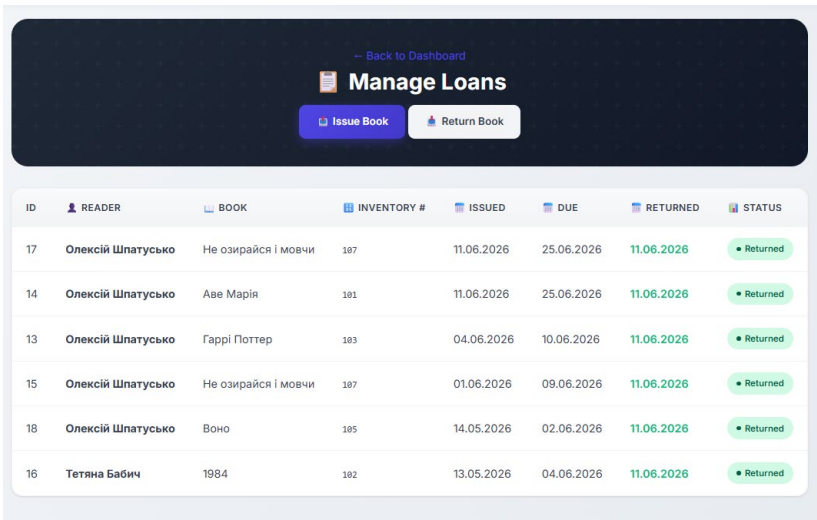


Рис. 3.21 Форма оформлення видачі та повернення книги

На рис. 3.22. зображена повна структура файлів та каталогів.

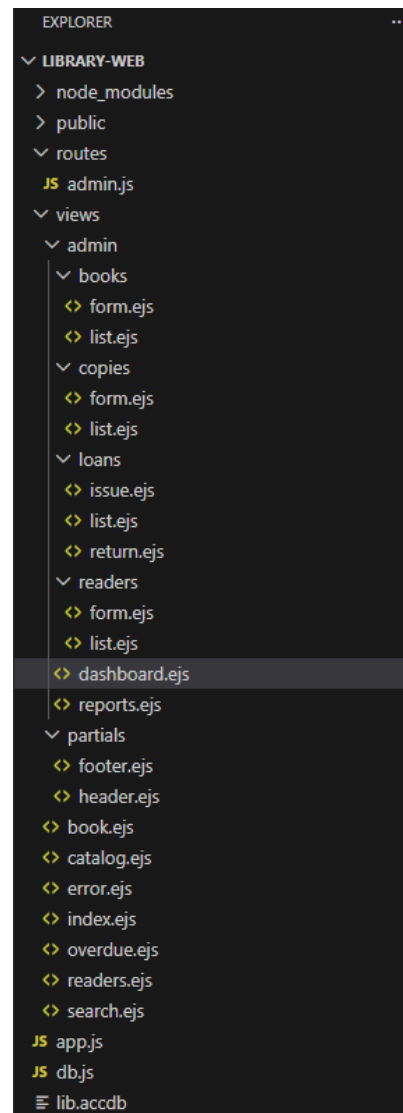


Рис. 3.22 Структура файлів та каталогів

Інструкція з запуску веб-інтерфейсу

Необхідне програмне забезпечення:

- Node.js версії 18 або вище
- Менеджер пакетів npm
- ODBC Driver for Microsoft Access
- Операційна система Windows

Підготовка проєкту

Перейти у каталог проєкту:

- `cd library-web`
- Ініціалізувати Node.js-проєкт (якщо ще не виконано):
- `npm init -y`

Встановити необхідні залежності:

- `npm install express odbc ejs`
- Переконалися, що файл бази даних `lib.accdb` знаходиться в кореневій папці проєкту.
- Запуск веб-сервера
- `node app.js`

Після успішного запуску в консолі з'явиться повідомлення про старт сервера.

Доступ до системи

У веб-браузері перейти за адресою:

- `http://localhost:3000`
- Після цього відкриється головна сторінка веб-інтерфейсу бібліотеки.

Обмеження та можливості розширення

Поточні обмеження:

- адміністративна панель реалізована, однак відсутня система автентифікації користувачів;
- не передбачено розмежування ролей доступу (адміністратор, бібліотекар, читач);
- веб-інтерфейс орієнтований на роботу в локальному середовищі та не розгорнутий у хмарній інфраструктурі;
- обмежена кількість одночасних підключень у зв'язку з використанням файлової бази даних MS Access;
- відсутній механізм автоматичного резервного копіювання бази даних через веб-інтерфейс.

Технічні рекомендації для впровадження:

- впровадження системи авторизації та керування ролями користувачів;
- реалізація онлайн-бронювання та попереднього замовлення книг;

- додавання функції продовження терміну користування через веб-інтерфейс;
- інтеграція системи автоматичних сповіщень про прострочення (email, SMS або месенджери);
- міграція з MS Access на серверну СУБД (MySQL, PostgreSQL) для підвищення продуктивності та масштабованості;
- розгортання веб-застосунку на хмарному сервері.

Висновки по веб-інтерфейсу

Демонстраційний веб-інтерфейс розроблено на платформі Node.js з використанням фреймворку Express. Цей демонстраційний веб-інтерфейс демонструє, наскільки ефективно система бібліотечного обліку, створена в MS Access, може інтегруватися з сучасними веб-технологіями. Веб-додаток надає доступ до основної інформації про книги, примірники, читачів та видані книги, окрім функцій пошуку та фільтрації, перегляду прострочених видань, роботи з адміністративною панеллю, додавання, редагування, видалення записів.

В основному він має зручну навігацію, сучасний візуальний інтерфейс та адаптивний макет, що підходить для різних пристроїв.

Результат можна розглядати як прототип веб-версії системи, на основі якої відбудуватиметься подальший розвиток, зокрема реалізація авторизації користувачів, розподілу ролей, онлайн-бронювання книг та міграція на серверну СУБД.

Розроблений веб-інтерфейс демонструє перспективність обраної архітектури та можливість поетапного переходу від локальної настільної системи до повноцінної веб-бібліотечно-інформаційної системи.

У цьому розділі описано реалізовані модулі бібліотечної інформаційної системи, розробленої на базі Microsoft Access, та демонстраційного веб-інтерфейсу, створеного на Node.js. Також можливе використання архітектури

системи, структури бази даних, побудови таблиць та зв'язків, реалізації форм користувача, SQL-запитів та звітів.

Система забезпечує повноцінний циклічний характер процесів обліку бібліотечного фонду, реєстрації та підтримки читачів, видачі та повернення примірників, контролю прострочених позик та формування звітів. Той факт, що була впроваджена адміністративна панель разом із веб-інтерфейсом, свідчить про можливість подальшого розвитку функціональності цієї системи, а також інтеграції десктопного рішення із сучасними веб-технологіями.

Результат може бути використаний як робочий прототип бібліотечної інформаційної системи, а також як основа для подальшого розвитку, зокрема шляхом впровадження авторизації користувачів, розмежування ролей доступу та переходу на серверну СУБД. Основні теоретичні та практичні результати дослідження були представлені на науковій конференції, де викликали інтерес та обговорення серед учасників, що підтверджує актуальність та практичну цінність виконаної роботи.

ВИСНОВКИ

Розробка інформаційної системи виконана відповідно до завдань кваліфікаційної роботи, дозволила закріпити теоретичні знання та набуті практичних навичок у галузі аналізу предметних областей, проектуванні баз даних.

Робота передбачала аналіз бібліотеки як об'єкта, що підлягає автоматизації: визначення її організаційної структури; дослідження інформаційних потоків та опис ключових бізнес-процесів (реєстрація читачів, облік фонду у процесі видачі та повернення). Виявлено, що використання децентралізованих паперових носіїв та розрізнених електронних таблиць призводить до дублювання інформації, ускладнює пошук і підвищує ризик виникнення помилок, у зв'язку з чим було науково обґрунтовано та доведено доцільність розробки й впровадження спеціалізованого програмного комплексу для автоматизації облікових процесів бібліотеки.

На основі зібраного фактичного матеріалу було визначено функціональні та не функціональні вимоги до інформаційної облікової системи для бібліотек. Основні модулі включають: каталог книг, облік читачів, управління операціями видачі-повернення та контроль простроченої заборгованості й нарахування штрафів, а також звітно-аналітичний модуль.

У першій частині аналізується діяльність бібліотеки як об'єкта автоматизації разом із дослідженням умов її роботи. Друга частина містить аналіз фактичних даних, огляд аналогів, а також формулювання вимог і принципів побудови баз даних. У третьому розділі представлено реалізацію в MS Access, що описує структуру бази даних; запити; звіти; результати тестування, а також розробку демонстраційного веб-інтерфейсу на Node.js.

В рамках роботи було розроблено інформаційну систему в Microsoft Access. Створено реляційну базу даних з первинними та зовнішніми ключами, індексами та забезпечено цілісність посилань. Таблиці та зв'язки

реалізовані разом із користувацькими формами, SQL-запитами та звітами, що підтримують базові операції та аналітику. Було проведено функціональне тестування за певними сценаріями, яке підтвердило коректність роботи ключових модулів.

Результати показують, що завдяки розробленій інформаційній системі виконання типових операцій обслуговування читачів бібліотеки стало більш ефективним, зменшилась кількість помилок введення, підвищилась оперативність контролю за наявністю книжкового фонду та було забезпечено формування надійної статистичної звітності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гуржій А. М. Основи інформаційних технологій [Текст] : підручник / А. М. Гуржій, Н. І. Поворознюк, В. В. Самсонов. Київ : Літера ЛТД, 2023. 336 с.
2. ДСТУ 4163:2020. Державна уніфікована система документації. Вимоги до оформлення документів [Текст] : чинний від 2021-09-01. Київ, 2020. 50 с.
3. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання [Текст] : чинний від 2016-07-01. Київ, 2016. 53 с.
4. Ляшенко І. М. Основи математичного моделювання [Текст] : навч. посібник / І. М. Ляшенко, М. В. Коробова, А. М. Столяр. Тернопіль : Богдан, 2006. 304 с.
5. Павленко Л. А. Corporate Information Systems [Текст] : підручник / Л. А. Павленко. Харків : ІНЖЕК, 2005. 416 с.
6. Павлиш В. А. Основи інформаційних систем і технологій [Текст] : навч. посібник / В. А. Павлиш, Л. К. Гліненко. Львів : Видавництво Львівської політехніки, 2018. 384 с.
7. Положення про електронну бібліотеку [Електронний ресурс] : Чернігівська обласна універсальна наукова бібліотека імені В. Г. Короленка : вебсайт. URL: <http://libkor.com.ua/elektronna-biblioteka-polozhennia> (дата звернення: 06.02.2026).
8. Попик В. Формування національного науково-інформаційного простору: інтеграція й кооперація зусиль бібліотек, наукових установ та інформаційних центрів України [Текст] / В. Попик // Бібліотечний вісник. 2015. № 3. С. 3–11. URL: http://nbuv.gov.ua/UJRN/bv_2015_3_3 (дата звернення: 06.02.2026).
9. Стандарт вищої освіти України. Перший (бакалаврський) рівень вищої освіти, галузь знань «Інформаційні технології», спеціальність

- 126 «Інформаційні системи та технології» [Електронний ресурс]. Київ : МОН України, 2018. URL: <https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/2018/11/14/12-2018-11-14-126.pdf> (дата звернення: 06.02.2026).
10. Bootstrap Documentation [Electronic resource] : official website. URL: <https://getbootstrap.com/> (дата звернення: 06.02.2026).
 11. Date C. J. An Introduction to Database Systems [Text] / C. J. Date. 8th ed. Pearson / Addison-Wesley, 2003. 1024 p.
 12. Express.js Documentation [Electronic resource] : official website. URL: <https://expressjs.com/> (дата звернення: 06.02.2026).
 13. Maryina O. Libraries of Ukraine in the digital media space: theoretical and methodological foundations of development [Text] : diss. ... Dr. Sciences in Social Communications : 27.00.03 / O. Maryina ; Kharkiv State Academy of Culture.- Kharkiv, 2018.
 14. Microsoft Access Documentation [Electronic resource] : official documentation. URL: <https://learn.microsoft.com/en-us/documentation/> (дата звернення: 06.02.2026).
 15. Node.js Documentation [Electronic resource] : official website. URL: <https://nodejs.org/en/docs/> (дата звернення: 06.02.2026).
 16. odbc [Electronic resource] : documentation of the package for Node.js (npm). URL: <https://www.npmjs.com/package/odbc> (дата звернення: 06.02.2026).
 17. W3Schools Online Web Tutorials [Electronic resource] : educational website. URL: <https://www.w3schools.com/> (дата звернення: 06.02.2026).