

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: *Комп'ютеризована система розумної книгообмінної будки*

Виконав: студент 4 курсу, групи СІ-42
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

	<u>Щирба Т.А.</u> (підпис)	(прізвище та ініціали)
Керівник	<u>Шингера Н.Я.</u> (підпис)	(прізвище та ініціали)
Нормоконтроль	<u>Тиш Є.В.</u> (підпис)	(прізвище та ініціали)
Завідувач кафедри	<u>Осухівська Г.М.</u> (підпис)	(прізвище та ініціали)
Рецензент	<u>Михалик Д.М.</u> (підпис)	(прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Осухівська Г.М.
(підпис) (прізвище та ініціали)
«27» січня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студента Щирба Тетяна Андріївна
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система розумної книгообмінної будки

Керівник роботи Шингера Наталя Ярославівна, к.т.н., доц. каф. КС
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 27 » січня 2025 року № 4/7-53

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз технічного завдання

2. Проєктна частина

3. Практична частина

4. Безпека життєдіяльності, основи охорони праці

Висновки

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

1. Структурна схема. 2. Блок-схема алгоритму роботи апарату. 3. Схема електрично-принципова. 4. Концептуальна схема бази даних

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>д.т.н., професор Пилипець М. І.</i>		

7. Дата видачі завдання 27.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Розробка технічного завдання.</i>	<i>27.01. – 02.02.</i>	<i>Викон.</i>
2.	<i>Робота над основними розділами – Аналіз технічного завдання. Аналіз предметної області та актуальності</i>	<i>03.02.– 09.03.</i>	<i>Викон.</i>
3.	<i>Робота над основними розділами. Вибір апаратного забезпечення та проектування електричної схеми.</i>	<i>10.03.– 16.04.</i>	<i>Викон.</i>
4.	<i>Робота над основними розділами. Реалізація та розробка вебплатформи: база даних, інтерфейс, логіка угод</i>	<i>17.04. - 15.06.</i>	<i>Викон.</i>
5.	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>26.05. – 02.06.</i>	<i>Викон.</i>
6.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>03.06. – 10.06.</i>	<i>Викон.</i>
7.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>09.06. -15.06.</i>	<i>Викон.</i>
8.	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>16.06.2025р. – 22.06.2025р.</i>	<i>Викон.</i>
9.	<i>Захист кваліфікаційної роботи бакалавра</i>	<i>27.06.</i>	<i>Викон.</i>

Студент

(підпис)

Щирба Т.А.

(прізвище та ініціали)

Керівник роботи

(підпис)

Шингера Н.Я

(прізвище та ініціали)

АНОТАЦІЯ

Щирба Т.А. Комп'ютеризована система розумної книгообмінної будки: робота на здобуття кваліфікаційного ступеня бакалавра: спец. 123 — комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2025.

Ключові слова: PHP, JavaScript, MySQL, ESP32, QR, web, book exchange, automated system.

Кваліфікаційна робота присвячена розробці автоматизованої системи обміну книгами з використанням веб-платформи та апаратного модуля на базі мікроконтролера.

Метою цієї роботи є розробка та впровадження системи, яка забезпечує безпечний та безконтактний обмін книгами шляхом підтвердження транзакцій через веб-сайт та надання фізичного доступу через сканування QR-коду.

У першому розділі кваліфікаційної роботи було проаналізовано актуальність теми та сучасний стан технологій обміну книгами. Було визначено функціональні та технічні вимоги, а також розглянуто різні можливі варіанти реалізації.

У другому розділі було розроблено структуру комп'ютеризованої системи, обґрунтовано вибір апаратних компонентів та створено електричну схему за допомогою KiCad. Описано кожен елемент апаратної системи, включаючи мікроконтролер ESP32-CAM, датчики, реле та модулі живлення.

У третьому розділі було представлено практичну реалізацію системи. Пояснено середовище розробки та логіку прошивки, показано інтеграцію електронних компонентів, а також описано веб-платформу для взаємодії з користувачем та підтвердження транзакцій. Також було окреслено інтерфейс адміністратора та механізми безпеки системи.

ANNOTATION

Shchyrba T.A. Computerized system of a smart book exchange booth: Bachelor's Graduation Thesis: speciality 123 — computer engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2025.

Keywords: PHP, JavaScript, MySQL, ESP32, QR, web, book exchange, automated system.

The qualification work is devoted to the development of an automated book exchange system using a web platform and a microcontroller-based hardware module.

The purpose of this work is to design and implement a system that enables secure and contactless book exchange by confirming transactions through a website and granting physical access via QR code scanning.

In the first section of the qualification work, the relevance of the topic and the current state of book exchange technologies were analyzed. Functional and technical requirements were defined, and various possible implementation options were considered.

In the second section, the structure of the computerized system was developed, the selection of hardware components was justified, and the electrical schematic was created using KiCad. Each element of the hardware system, including the ESP32-CAM microcontroller, sensors, relays, and power supply modules, was described.

In the third section, the practical implementation of the system was presented. The development environment and firmware logic were explained, the integration of electronic components was shown, and the web platform for user interaction and transaction confirmation was described. The administrator interface and system security mechanisms were also outlined.

ЗМІСТ

ВСТУП 9

РОЗДІЛ 1	АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ	10
1.1	Актуальність та аналіз предметної області.....	10
1.2	Функціональні та технічні вимоги до системи	12
1.3	Аналіз можливих рішень поставленого завдання	14
РОЗДІЛ 2	ПРОЄКТНА ЧАСТИНА.....	17
2.1	Розробка загальної структури комп'ютеризованої системи.....	17
2.2	Обґрунтування вибору апаратного забезпечення проектного комп'ютерного засобу.....	18
2.2.1	Мікроконтролер.....	19
2.2.2	Дисплеї.....	21
2.2.3	Сенсори.....	22
2.2.4	Виконавчі елементи	23
2.2.5	Живлення та енергозабезпечення.....	25
2.3	Проектування електрично принципової схеми	27
РОЗДІЛ 3	ПРАКТИЧНА ЧАСТИНА.....	31
3.1	Реалізація проектних рішень	31
3.2	Налаштування середовища для розробки мікроконтролера	33
3.3	Опис прошивки ESP32 та логіки керування	36
3.4	Розробка сайту для обліку книг і підтвердження угод	41
3.4.1	Мова програмування та технології	41

					КС КРБ 123.247.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Щирба Т.А.			Комп'ютеризована система книгообмінної будки			Літ.	Арк.	Аркуші	
Перевір.		Шингера Н.Я.								6	
Реценз.		Михалик Д.М.									
Н. Контр.		Тиш Є.В.						ТНТУ, каф. КС, гр. СІ-42			
Затверд.		Осухівська Г.М.									

3.4.2	Реєстрація та авторизація користувачів	44
3.4.3	Додавання книг із вибором статусів «обмін» або «продаж»	45
3.4.4	Профілі користувачів	47
3.4.5	Система чатів між користувачами	48
3.4.6	Система створення та підтвердження угоди.....	49
3.5	Генерація QR-кодів та перевірка угоди через сайт	50
3.6	База даних	51
3.7	Керування веб-сайтом на рівні адміністратора	54
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ ..		56
4.1	Навчання працюючих та інструктажі з охорони праці	56
4.2	Заходи пожежної безпеки для книгообмінної будки	58
ВИСНОВКИ		62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		63
Додаток А Технічне завдання		
Додаток Б Перелік елементів		
Додаток В Лістинги функцій комп'ютеризованої системи		
Додаток Г Лістинги файлів веб-додатку		
Додаток Е Лістинг функції підключення до бази даних		

СПИСОК СКОРОЧЕНЬ

API – Application Programming Interface

BJT – Bipolar Junction Transistor

CSS – Cascading Style Sheets

DC – Direct Current

DBMS – Database Management System

DIP – Dual In-line Package

DOM – Document Object Model

ESP – Espressif Systems Platform

ESP32 – Espressif Systems 32-bit Microcontroller

GPIO – General Purpose Input/Output

GUI – Graphical User Interface

IDE – Integrated Development Environment

I/O – Input/Output

IP – Internet Protocol

JSON – JavaScript Object Notation

LED – Light Emitting Diode

MCU – Microcontroller Unit

MQTT – Message Queuing Telemetry Transport

NAS – Network Attached Storage

PCB – Printed Circuit Board

QR – Quick Response

SQL – Structured Query Language

					<i>КС КРБ 123.247.00.00 ПЗ</i>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У інформаційному суспільстві доступ до знань, зокрема у формі друкованої книги, залишається актуальним попри цифровізацію більшості сфер життя. Книга — не лише джерело інформації, але й засіб культурного обміну, що сприяє самоосвіті, розвитку критичного мислення та збереженню мовної і літературної спадщини. Водночас багато книг після прочитання залишаються без використання, тоді як інші люди могли б зацікавитися ними.

На цьому тлі виникає потреба в альтернативних формах поширення літератури, зокрема через обмін. Ідеї вільного книгообміну давно застосовуються у вигляді полицок у громадських місцях, буккросингу чи невеликих бібліотек самообслуговування. Проте таким ініціативам притаманні обмеження: відсутність контролю, фізичні втрати книг, неможливість відстеження обігу, а також залежність від людського фактору.

Інтеграція технічних засобів у процес обміну дозволяє створити інфраструктуру, що поєднує безконтактність, зручність та облік. Комп'ютеризована система книгообміну, що використовує вебінтерфейс для керування угодами й апаратну частину для автоматичного доступу до книг, дає змогу вивести процес на якісно новий рівень. Це відкриває перспективи не лише для особистого або локального використання, а й для впровадження у бібліотеках, навчальних закладах, коворкінгах чи житлових комплексах.

У межах проєкту розглядається одна з можливих реалізацій такого підходу - на основі цифрової ідентифікації, обліку, контролю доступу та користувацької взаємодії через інтерфейс.

					КС КРБ 123.247.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Актуальність та аналіз предметної області

У сучасному інформаційному суспільстві книга залишається не лише джерелом знань, а й важливим культурним та соціальним інструментом. Попри розвиток цифрових технологій, попит на друковану літературу зберігається, зокрема серед молоді, студентства та людей, які прагнуть усвідомленого споживання контенту. Проте наявність друкованих видань у вільному обігу не завжди відповідає потребам охочих до читання. У багатьох випадках нові книги є недоступними через їхню вартість, а бібліотечні фонди — обмежені, застарілі або незручно локалізовані. У зв'язку з цим усе більшої актуальності набувають ініціативи, що передбачають обмін книгами між користувачами на безоплатній або умовно-комерційній основі.

У практиці останніх років поширення набули різні форми відкритого доступу до книг: буккросинг, полички в кав'ярнях, обмінні стелажі у коворкінгах, а також ініціативи на кшталт «вільної бібліотеки» в під'їздах чи закладах освіти. Ці підходи мають значний соціальний потенціал, адже стимулюють культурну взаємодію, підвищують доступність літератури та формують навички відповідального обміну.

Разом із тим, більшість таких рішень є неформальними, неструктурованими і часто незахищеними. Відсутність фіксації угод, контролю за станом книг чи за порядком обміну призводить до хаотичності: книги втрачаються, псуються або використовуються не за призначенням. Полички в публічних місцях не мають системи верифікації або взаємної відповідальності, що знижує рівень довіри до таких ініціатив.

					КС КРБ 123.247.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Щирба Т.А.			Аналіз технічного завдання		
Перевір.		Шингера Н.Я.					
Реценз.		Михалик Д.М.					
Н. Контр.		Тиш Є.В.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркуші
						10	
					ТНТУ, каф. КС, гр. СІ-42		

Також варто зазначити, що фізичні форми книгообміну часто залежать від наявності персоналу, погодних умов, доступу до приміщення чи територіальної зручності, що обмежує їхню ефективність у реальному використанні.

На цьому тлі запропонований проєкт вирізняється технологічним підходом до організації книгообміну. Система поєднує вебінтерфейс для пошуку книг, організації угод і спілкування користувачів із фізичною книгообмінною будкою, доступ до якої забезпечується за допомогою QR-коду, що генерується сайтом після підтвердження обома сторонами умов угоди.

Окрім соціальної цінності, сучасні форми обміну літературою все частіше спираються на технологічні рішення, здатні значно підвищити ефективність і зручність таких процесів. Поєднання фізичних об'єктів зокрема, автоматизованих будок для зберігання книг - із вебінтерфейсом створює новий формат взаємодії, де традиційна функція книгообміну отримує цифрове розширення. Інтеграція мережевих технологій дозволяє реалізовувати механізми реєстрації користувачів, створення угод, підтвердження обміну та моніторингу дій, що раніше були неможливими або потребували людського втручання.

У цьому контексті важливим є принцип синхронізації: фізична будка не виконує роль автономної одиниці, а функціонує як частина загальної цифрової системи, де прийняття рішень щодо доступу базується на об'єктивних даних - статусі угоди, зафіксованому у базі даних. Така модель взаємодії усуває випадковість, запобігає зловживанням та підвищує рівень довіри до системи. Користувач більше не взаємодіє з книгообміном хаотично - він діє в межах чітко регламентованого процесу, де кожен крок, від ініціації обміну до фізичного отримання книги, супроводжується системною логікою.

Використання QR-кодів, згенерованих після підтвердження угоди обома сторонами, дозволяє забезпечити гнучкий і захищений доступ до будки. Такий механізм дозволяє уникнути потреби у постійній присутності

					КС КРБ 123.247.00.00 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

персоналу, знижує ймовірність несанкціонованого відкриття комірок, а також гарантує, що обмін відбувається виключно між учасниками, які погодили умови. Рішення, засноване на поєднанні вебфункціоналу з фізичною інфраструктурою, формує новий підхід до організації спільного доступу до ресурсів - більш надійний, прогнозований і прозорий. Саме таке середовище відповідає сучасним вимогам до сервісів з елементами довіри між користувачами [11].

1.2 Функціональні та технічні вимоги до системи

Розробка системи обміну книгами із залученням апаратної частини вимагає чіткого визначення функціональних та технічних вимог, що стосуються як вебінтерфейсу, так і взаємодії з фізичним пристроєм. Система передбачає два взаємопов'язані компоненти: сайт, що має виконувати роль логіки обробки даних та комунікації, і апаратну частину, яка має забезпечувати фізичну реалізацію доступу до книг через спеціальну будку.

З боку сайту функціональність має охоплювати базові можливості вебсервісу з авторизацією користувачів, а також специфічні механізми, пов'язані з логікою обміну. Зокрема, користувач має мати можливість створити особистий профіль, пройти процедуру реєстрації та автентифікації, додати книгу до системи, переглядати книги інших користувачів, а також ініціювати спілкування через інтегрований чат. Важливою особливістю є система створення угод, у межах якої передбачене погодження умов обміну або продажу, з фіксацією стану угоди на кожному з етапів. Ключовим є обов'язкове підтвердження угоди обома сторонами, що уможливорює подальший доступ до книги через фізичну будку [12].

Апаратна частина має виконувати функції авторизації на місці обміну. Після завершення угоди на сайті система генеруватиме унікальний QR-код, який користувач сканує за допомогою мобільного пристрою. QR-код містить

					<i>КС КРБ 123.247.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

посилання на відповідний серверний запит, що верифікує статус угоди. У разі позитивного результату, мікроконтролер активує механізм відкриття замка на визначений проміжок часу. Таким чином, фізичний доступ до книги забезпечується лише після проходження повної процедури підтвердження.

У структурі системи мають оброблятися три основні типи даних: інформація про користувачів (імена, пошта, ID), бібліографічні відомості про книги (назва, автор, жанр, статус доступності), а також деталі угод (учасники, обрана книга, статус, час створення й підтвердження). Вся ця інформація зберігається у базі даних та взаємодіє з вебінтерфейсом і апаратною частиною.

З боку технічної реалізації в апаратному модулі необхідно використати мікроконтролер ESP32 з підтримкою Wi-Fi та камерою, яка забезпечує функцію сканування QR-кодів без зовнішніх пристроїв, як на рисунку 1.1.

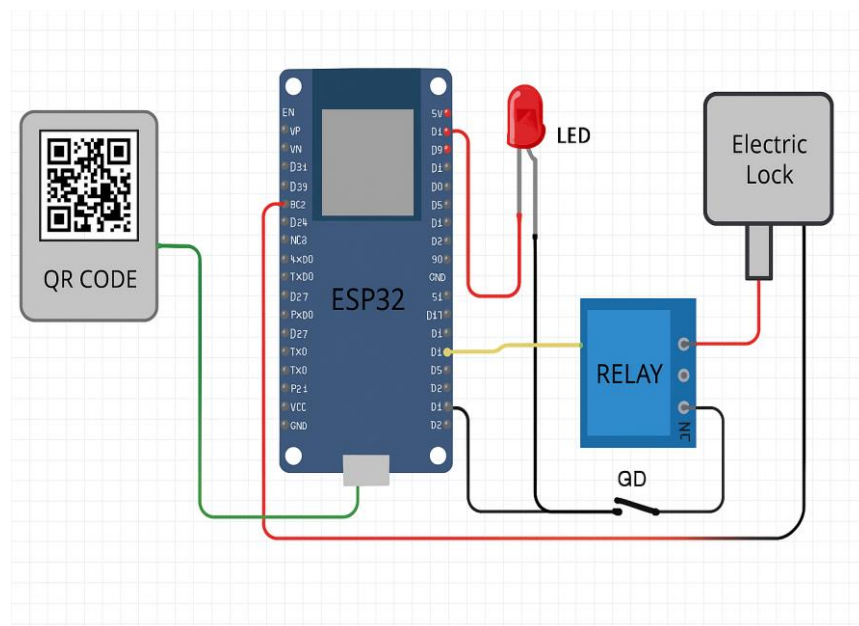


Рисунок 1.1 – Схема електричних з'єднань

Керування фізичним замком повинно реалізуватися через реле, що подає живлення на соленоїдний механізм відкриття. Стан зачинення контролюється за допомогою датчика Холу, що спрацьовує у присутності магнітного поля.

Уся система живиться від акумулятора через модуль зарядки та стабілізації напруги. Завдяки такому технічному поєднанню повинна забезпечитись автономна, надійна та енергоефективна робота пристрою.

1.3 Аналіз можливих рішень поставленого завдання

Під час проектування системи автоматизованого обміну книгами з доступом через QR-код передусім постало питання вибору архітектури та технологій для реалізації вебчастини - адже саме через неї здійснюється основна логіка взаємодії між користувачами та ініціація угод. Найочевиднішим варіантом на ранньому етапі було використання готових платформ або CMS-рішень, зокрема WordPress, які дозволяють швидко побудувати сайт із базовими функціями авторизації та формами. Проте такі інструменти обмежені в налаштуванні логіки угод і не дозволяють гнучко реалізувати процеси двостороннього підтвердження, створення унікальних чатів між користувачами чи формування прив'язаних до угоди QR-кодів. Додатково, розширення функціональності в рамках CMS потребує численних плагінів, що не завжди забезпечує стабільність і контроль над внутрішніми процесами.

Іншим можливим підходом була побудова вебсистеми з використанням фреймворків повного циклу, як-от Laravel для PHP або Django для Python. Такі рішення надають більшу гнучкість і сучасні інструменти, але їх використання передбачає складну конфігурацію середовища, більше навантаження на хостинг і вищі вимоги до ресурсів розробника. У контексті проєкту, що має зосереджуватись на простоті й ефективності реалізації, такий підхід видався надмірно складним.

У підсумку було обрано підхід на основі чистого PHP у поєднанні з JavaScript для клієнтської частини та MySQL як системи управління базами даних. Це дозволило досягти високого рівня керованості всіх процесів,

					<i>КС КРБ 123.247.00.00 ПЗ</i>	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

реалізувати власну логіку угод з їхнім покроковим підтвердженням, впровадити функціонал чату, формування угод із прив'язкою до конкретної книги та QR-коду, а також забезпечити прямий зв'язок між вебчастиною і апаратною частиною. Крім того, цей варіант зручний для хостингу, не вимагає складних залежностей і забезпечує достатній рівень продуктивності для проєкту такого масштабу.

Паралельно з розглядом цифрової логіки проєкту можна проаналізувати альтернативні варіанти фізичної реалізації обміну книгами, які не передбачають електроніки або мережевого з'єднання. Але, незважаючи на свою популярність, такі рішення мають суттєві недоліки. Насамперед, відсутній будь-який контроль за тим, хто і коли бере книги, що призводить до втрат, псування літератури або її повного зникнення. Немає фіксації факту обміну, відсутній зворотний зв'язок між учасниками, а також неможливо забезпечити доступ до книг лише тим, хто дійсно взяв участь в обміні. Відсутність захисту від зовнішнього втручання або погодних умов ще більше ускладнює підтримку таких пунктів у придатному для користування стані.

У процесі розробки апаратної частини системи було розглянуто кілька підходів до організації доступу користувача до фізичної книги. Основна задача полягала в тому, щоб забезпечити контрольований, безпечний та максимально простий механізм розблокування шафки виключно після підтвердження обома сторонами відповідної угоди на сайті. Водночас система мала бути доступною з фінансової точки зору, енергоефективною та придатною для автономної роботи.

На початковому етапі розглядалися такі альтернативи, як RFID-мітки, NFC-карти, клавіатурні панелі або Bluetooth-зв'язок зі смартфоном. Кожен з цих підходів мав свої переваги, але також суттєві обмеження. Наприклад, використання RFID або NFC потребує спеціальних фізичних міток, які необхідно видавати кожному користувачеві. Це ускладнює масштабування системи та створює додаткове навантаження на організацію обміну.

					<i>КС КРБ 123.247.00.00 ПЗ</i>	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

Зчитування таких міток також залежить від якості приймача і може бути нестабільним, особливо при використанні дешевих модулів. Клавіатурні рішення є застарілими й не забезпечують належного рівня захисту, оскільки коди легко передаються стороннім особам, а також потребують пам'яті з боку користувача. Bluetooth-зв'язок, своєю чергою, вимагає парування пристроїв, що ускладнює взаємодію і знижує зручність використання.

У результаті аналізу було прийнято рішення використовувати QR-код як основний механізм авторизації доступу. Це рішення має одразу кілька переваг:

- QR-коди легко генеруються на боці сервера для кожної конкретної угоди, без додаткових витрат на обладнання чи носії.
- Їх можна зчитати безконтактно за допомогою будь-якого смартфона, що повністю відповідає сучасній моделі поведінки користувача.
- QR-код дозволяє передати унікальний ідентифікатор угоди у вигляді URL-адреси, яку обробляє вебсервер, повертаючи відповідь про можливість відкриття кімнатки.

Ключовим апаратним компонентом для реалізації цієї логіки став модуль ESP32-CAM, який поєднує у собі вбудований Wi-Fi-інтерфейс, достатню обчислювальну потужність для обробки запитів, підтримку sleep-режимів для економії енергії, а головне — камеру, здатну зчитувати QR-коди без необхідності підключення зовнішніх сканерів.

					КС КРБ 123.247.00.00 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

2.1 Розробка загальної структури комп'ютеризованої системи

Запропонована комп'ютеризована система обміну книгами базується на поєднанні автономного апаратного модуля з централізованим інформаційним середовищем. У межах цього розділу розглядається структура виключно апаратної частини, яка виконує функції доступу до фізичної комірки у книгообмінній будці. Завданням апаратного модуля є прийняття рішення про відкриття або блокування доступу до книги на основі отриманої авторизаційної інформації [12].

Узагальнена структура пристрою включає п'ять основних функціональних блоків: блок живлення, керуючий контролер, виконавчі пристрої (реле та замок), система індикації та сенсорний модуль для зчитування положення дверцят (див. рис. 2.1).

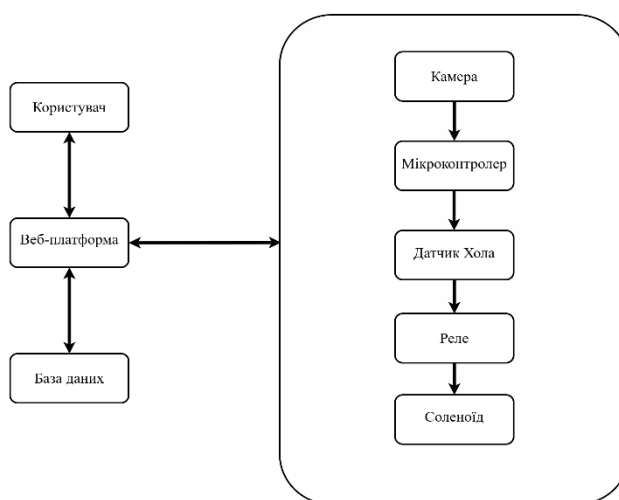


Рисунок 2.1 – Узагальнена структурна схема пристрою для сканування QR-коду

					КС КРБ 123.247.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Щирба Т.А.			Проектна частина	Літ.	Арк.
Перевір.		Шингера Н.Я.					17
Реценз.		Михалик Д.М.				ТНТУ, каф. КС, гр. СІ-42	
Н. Контр.		Тиш Є.В.					
Затверд.		Осухівська Г.М.					

Усі ці компоненти фізично поєднані в єдину схему, яка забезпечує автономну роботу пристрою без необхідності постійного втручання користувача або зовнішніх керуючих сигналів.

2.2 Обґрунтування вибору апаратного забезпечення проектного комп'ютерного засобу

У процесі проектування апаратної частини системи було визначено низку базових вимог, серед яких автономність роботи пристрою, компактність елементної бази, енергоефективність, здатність виконувати базові обчислення, обробляти сигнали з сенсорів і керувати виконавчими механізмами. Система мала забезпечувати надійне функціонування без постійного підключення до електромережі, а також працювати в режимі енергозбереження з можливістю активації у визначений момент взаємодії з користувачем.

Усі компоненти апаратної частини були підібрані з урахуванням необхідності забезпечення повної автономності системи. Як джерело живлення використовується акумуляторний елемент типу 18650 у поєднанні з модулем керування зарядом TP4056, що дає змогу не лише заряджати пристрій через USB-роз'єм, а й захищає елемент живлення від перезаряду або глибокого розряду. Такий підхід дозволяє створити надійну систему резервного живлення, яка не потребує постійного обслуговування або підключення до зовнішніх джерел електроенергії.

Для забезпечення відповідних рівнів напруги, необхідних для роботи різних модулів, до складу системи включено два підвищуючих перетворювачі на базі модуля MT3608. Один із них стабілізує напругу на рівні 5 В для живлення мікроконтролера ESP32, інший забезпечує напругу 12 В для виконавчого елемента — соленоїдного замка. Такий розподіл дозволяє забезпечити ефективну роботу як логічної, так і силової частини системи без перевантаження живлення.

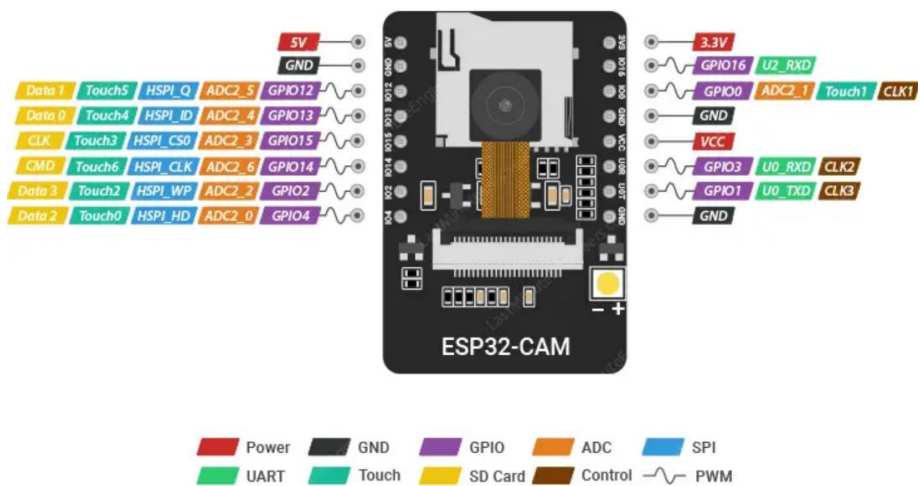
Особливу увагу було приділено питанню визначення поточного стану

					КС КРБ 123.247.00.00 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

комірки. Для цього застосовано цифровий датчик Холу, який реагує на наявність магнітного поля. Магніт фіксується на дверцятах будки, а сенсор — у внутрішній частині корпусу. У разі, якщо дверцята відчинені або нещільно закриті, сигнал від датчика не надходить, що дозволяє контролеру ідентифікувати стан і заблокувати повторне відчинення до моменту правильного закривання. Такий механізм гарантує коректність кожного циклу відкриття-закриття та виключає можливість повторного доступу до книги без оновленої угоди.

2.2.1 Мікроконтролер

У структурі проєктованого пристрою роль керуючого елементу виконує мікроконтролер ESP32-CAM, який було обрано з огляду на його технічні характеристики, функціональну гнучкість та відповідність вимогам до автономного, енергоефективного й комунікаційно здатного пристрою. Дана плата поєднує в собі процесорну частину з ядрами Tensilica Xtensa LX6, вбудовану підтримку Wi-Fi та Bluetooth, набір цифрових портів введення/виведення, а також інтерфейс для підключення камери, зображення плати на рисунку 2.2 [13].



ESP32-CAM Pinout

Рисунок 2.2 – Мікроконтролер ESP32-CAM

Особливістю ESP32-CAM є вбудована камера OV2640 з роздільною здатністю до 2 мегапікселів. Це дозволяє локальне зчитування QR-кодів - ключового елемента авторизації у межах даної системи. Завдяки цьому відпадає потреба у зовнішньому сканері, що зменшує кількість компонентів, спрощує конструкцію та знижує загальну вартість пристрою.

У контексті роботи проєкту ESP32-CAM виконує одразу декілька функцій. Насамперед він забезпечує зчитування QR-коду, після чого формує HTTP-запит до вебсерверу, у відповідь на який отримує інформацію про статус угоди. У разі позитивного результату, контролер генерує сигнал на один із цифрових виходів для активації реле, що замикає коло живлення соленоїдного замка. Додатково контролер керує двокольоровим світлодіодом, а також обробляє сигнал із датчика Холу, що дозволяє контролювати стан дверцят [13].

Характеристики ESP32-CAM представлені в таблиці 2.1.

Таблиця 2.1 – Характеристики ESP32-CAM

Процесор	32-бітний, частота 160-240 МГц
Об'єм RAM	520 КБ
Об'єм Flash-пам'яті	448 КБ
Напруга живлення	2.2 – 3.6В
Зовнішня Flash-пам'ять	4 МБ
Виводів GPIO	8
Інтерфейси	UART, SPI, I2C, PWM
Споживана потужність: при вимкненому спалаху	180 мА
при включеному спалаху	-
Під час передачі WiFi	-
Температурний діапазон: робочий	-20 – 85 °C
Розміри	40,5 x 27 x 4,5 мм
Різне	Кнопка скидання налаштувань

Модуль може перебувати у п'яти режимах. Окрім увімкненого та вимкненого спалаху, є ще глибокий сон, модем сон та легкий сон.

2.2.2 Дисплеї

У проєктованій системі дисплеї як елементи виведення інформації свідомо не застосовуються. Це обумовлено специфікою архітектури рішення, яка передбачає відсутність прямої взаємодії користувача з апаратним модулем. Усі ключові дії — перегляд книг, створення та підтвердження угод, доступ до QR-коду — відбуваються на стороні вебплатформи, доступної з будь-якого мобільного або стаціонарного пристрою з браузером.

Візуальний інтерфейс користувача повністю винесено на рівень програмного забезпечення сайту, що дозволяє спростити апаратну частину, зменшити її енергоспоживання, вартість та габарити. Наявність дисплея на пристрої у цьому випадку не має практичного сенсу, оскільки інформація, яка могла б виводитися, вже доступна користувачу у вебінтерфейсі до моменту фізичної взаємодії з будкою.

Однак важливою частиною функціональності пристрою є можливість зчитування QR-коду, який генерується на сайті після підтвердження угоди між користувачами. Цей QR-код містить унікальне посилання, що дозволяє перевірити статус відповідної угоди. У системі використовується вбудована камера модуля ESP32-CAM, яка забезпечує зчитування цього коду безпосередньо з екрана мобільного пристрою користувача.

Єдиним засобом зворотного візуального зв'язку в апаратному модулі залишається двокольоровий світлодіод, який подає індикативні сигнали про результат перевірки доступу — зелений колір означає дозвіл на відкриття, червоний — відмову (див. рис. 2.3).

					КС КРБ 123.247.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21



Рисунок 2.3 – Світлодіод двокольоровий

Характеристики двоколькового світлодіоду наведені в таблиці 2.2

Таблиця 2.2 – Характеристики ESP32-CAM

Тип включення	Із загальним анодом
Максимальний струм	20 мА
Номінальна напруга (чр/зел)	2.0-2.2 В / 2.8-3.0 В
Колір світіння	Червоний + зелений
Довжина хвилі	620 / 520 нм
Сила світла (чр/зел)	500/500 мкд
Кут огляду	30°

Даний підхід є оптимальним для задачі, яка вирішується, та забезпечує необхідну мінімальну інформативність при максимальній простоті.

2.2.3 Сенсори

У системі контролю доступу до комірки для визначення стану дверцят використовується цифровий датчик Холу SS41A. Його завдання полягає у виявленні наявності або відсутності магнітного поля, що дозволяє непрямим способом контролювати факт закриття дверцят (див. рис. 2.4).



Рисунок 2.4 – Датчик Холу

Постійний магніт закріплюється на рухомій частині — дверцятах будки, тоді як датчик фіксується нерухомо на корпусі. У момент, коли дверцята повністю зачиняються, магніт наближається до сенсора, і той спрацьовує, фіксуючи факт замкнутого положення.

Датчик SS41A реагує на магнітну індукцію в межах приблизно ± 40 Гаусс, що є оптимальним для надійної роботи з невеликим неодимовим магнітом. У звичайному стані вихід сенсора перебуває у високому логічному рівні, при виявленні магнітного поля він змінює свій стан, що може бути зчитано мікроконтролером. У проєкті сигнал з SS41A надходить на вхід GPIO14 модуля ESP32-CAM [14].

2.2.4 Виконавчі елементи

Виконавчі елементи, які реалізують фізичну дію в процесі управління доступом до кімрки, це елементи які безпосередньо реагують на керуючі сигнали з мікроконтролера та забезпечують механічне розблокування або візуальне сповіщення користувача.

Основним виконавчим механізмом є соленоїдний замок, розрахований на напругу 12 В (див. рис. 2.5).

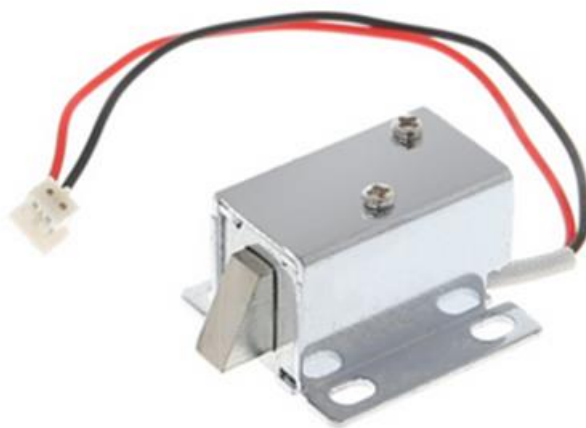


Рисунок 2.5 – Соленоїдний замок

У нормальному стані пристрій перебуває у замкненому положенні. Замок відкривається лише на короткий проміжок часу при подачі живлення, після чого автоматично повертається у закритий стан. Замок має споживання близько 0.4 А, що враховано при проектуванні системи живлення.

Для комутації живлення на замок застосовується релейний модуль на 12 В, здатний комутувати навантаження до 10 А (див. рис. 2.6).

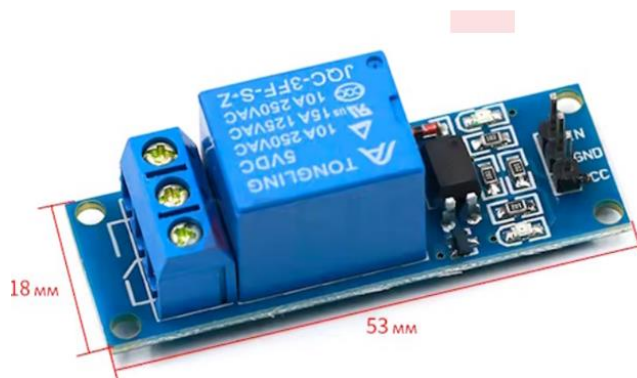


Рисунок 2.6 – Модуль 1-канального реле

Реле керується від мікроконтролера через оптопару, яка забезпечує електричну розв'язку між логічною схемою ESP32 і силовим колом. Такий підхід гарантує безпеку контролера та захист у разі коротких замикань або перевантаження (див. рис. 2.7).

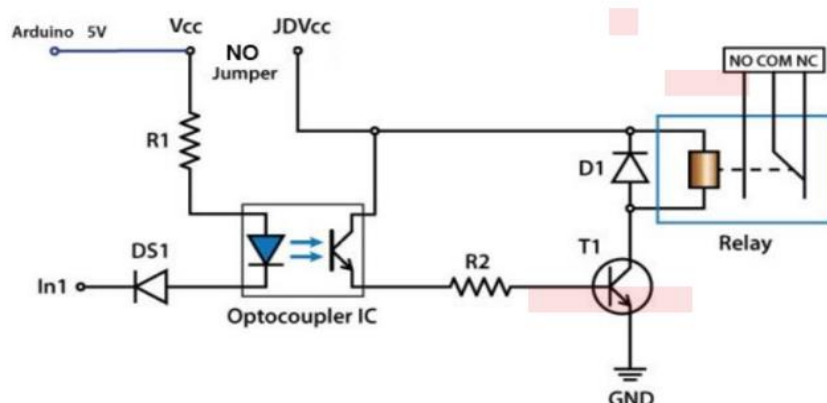


Рисунок 2.7 – Принципова схема для прикладу одного каналу реле

Реле оснащено оптичним розв'язком (опторозв'язкою), що забезпечує електричну ізоляцію між логічною та силовою частинами схеми. При знятті перемички JD-VCC та подачі живлення 5 В окремо на оптопару, модуль здатен коректно функціонувати навіть з мікроконтролерами, які працюють на логічному рівні 3.3 В. Принцип дії модуля базується на інверсній логіці: активація реле відбувається при подачі логічного нуля, деактивація — при логічній одиниці.

2.2.5 Живлення та енергозабезпечення

Живлення апаратної частини проєктованої системи реалізовано на основі літій-іонного акумулятора формату 18650 ємністю 2200 мА·год. Такий тип елемента обрано з огляду на його високу щільність енергії, доступність і можливість багаторазового перезарядження. Акумулятор працює у поєднанні з платою керування зарядом TP4056, яка обладнана роз'ємом Type-C для зручного підключення до джерела струму. Модуль TP4056 також виконує функцію захисту від перезаряду, надмірного розряду, короткого замикання та перегріву, що дозволяє експлуатувати систему без додаткових зовнішніх обмежень або контролю (див. рис. 2.8).

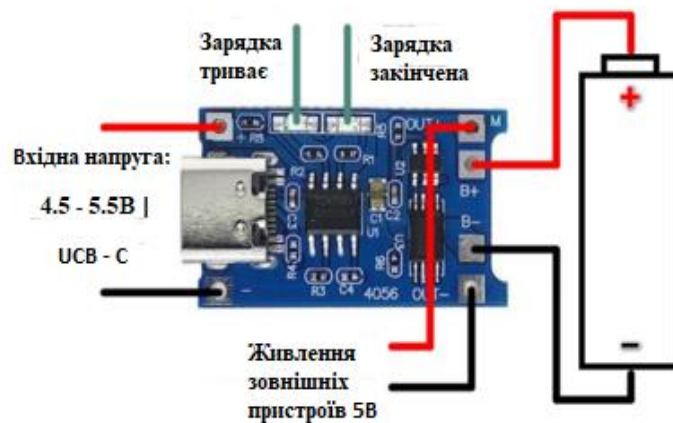


Рисунок 2.8 – Модуль TP4056

Оскільки різні модулі пристрою потребують різних рівнів напруги, у схемі використовуються два підвищуючі перетворювачі на базі мікросхеми MT3608. Один із них забезпечує стабілізовану напругу 5 В для живлення мікроконтролера ESP32-CAM, другий — формує 12 В, необхідні для роботи силових компонентів: реле та соленоїдного замка (див. рис. 2.9).



Рисунок 2.9 – Модуль підвищувача напруги MT3608

Такий розподіл дозволяє ефективно відокремити логічну та виконавчу частини пристрою й забезпечити стабільність їхньої роботи при коливаннях напруги на акумуляторі.

Завдяки застосованим енергоощадним рішенням, система здатна функціонувати в автономному режимі без підзарядки протягом тривалого часу. У режимі сну, коли пристрій очікує взаємодії з користувачем, загальне

					КС КРБ 123.247.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

споживання настільки низьке, що забезпечується до 11 годин безперервної роботи.

У разі активної взаємодії, коли здійснюється зчитування QR-коду, перевірка угоди та відкриття замка, енергоспоживання зростає, однак акумулятора вистачає орієнтовно на 17 повноцінних циклів відкриття комірки.

2.3 Проєктування схеми електрично-принципової

2.3.1 Середовище проєктування електричних схем

Проєктування схеми електрично-принципової апаратної частини системи здійснювалося в середовищі KiCad — відкритій програмній платформі для розробки електронних схем та друкованих плат.

Цей інструмент було обрано завдяки його функціональності, доступності, підтримці великої кількості бібліотек електронних компонентів, а також активній спільноті користувачів.

Середовище KiCad забезпечує зручні засоби для побудови повноцінної електричної принципової схеми, що є основою для подальшої розробки, виготовлення або відлагодження пристрою. У межах роботи над даним проєктом за допомогою цього ПЗ було створено логічну структуру з усіма зв'язками між елементами, а також здійснено перевірку електричних з'єднань, що дозволяє виявити помилки, наприклад, незамкнуті ланцюги або конфлікти сигналів.

Окрім розробки самої схеми, KiCad надає можливість автоматичного формування специфікації елементів (Bill of Materials, BoM), яка використовується як на етапі підбору компонентів, так і під час складання пристрою. При необхідності середовище також дозволяє виконати розводку друкованої плати, підготувати файли для виробництва (Gerber), провести візуалізацію кінцевого продукту та здійснити попереднє 3D-моделювання.

					КС КРБ 123.247.00.00 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

2.3.2 Опис схеми електрично-принципової

Електрично-принципова схема пристрою описує логіку взаємодії між живленням, контролером, сенсорами та виконавчими елементами. Живлення надходить до модуля TP4056 з роз'ємом Type-C, через який здійснюється заряджання літій-іонного акумулятора типу 18650. Вихідна енергія з акумулятора розподіляється на два підвищуючі перетворювачі MT3608: один забезпечує стабілізовану напругу 5 В для живлення мікроконтролера ESP32-CAM, другий — формує 12 В для керування силовими елементами, що наведено в таблиці 2.3.

Таблиця 2.3 – Компоненти всього пристрою

BT1	18650 2200mAh	Акумулятор
D1	LED_Dual_KAK	Двокольоровий світлодіод
J1	Jack-DC	Роз'єм живлення або зарядки
L1	Solenoid_12V_0.5A	Електромагнітний замок
M1	Neodim Magnet	Постійний магніт для спрацьовування датчика Холу
R1	300	Обмежувальний резистор для світлодіода
R2	4k7	Підтягувальний резистор
U1	ESP32_Cam_Module	Центральний мікроконтролер із камерою
U2	TP4056_TypeC	Зарядний модуль
U3	Relay_Module_10A_Opto	Реле
U4, U5	DCDC_BOOST_MT3608	Два підвищуючі перетворювачі
U6	SS41A	Датчик Холу

Мікроконтролер ESP32, отримуючи живлення, працює в режимі очікування сигналу. Через вбудовану камеру виконується зчитування QR-коду, після чого контролер надсилає HTTP-запит на сервер і перевіряє статус

угоди. У разі підтвердження угоди ESP32 зчитує сигнал із цифрового датчика Холу, який визначає, чи дверцята будки перебувають у закритому стані. Якщо всі умови виконані, на вихід GPIO12 подається логічний нуль, що активує реле і дозволяє короткочасно подати 12 В на соленоїдний замок, відкриваючи комірку.

Мікроконтролер ESP32-CAM, має обмежену кількість доступних портів GPIO через вбудовану камеру, що займає частину пінів. З урахуванням цього, розподілення цифрових входів/виходів здійснено максимально оптимально, з урахуванням електричної сумісності та пріоритетності сигналів. Порти які використано наведено в таблиці 2.4

Таблиця 2.4 – Специфікація керуючих контактів модуля ESP32-CAM

Сигнал	Призначення	Пояснення
GPIO12	Open	Лог."0" відчиняє комірку на час 5 секунд
GPIO13	Led_Red	Лог."0" вмикає червоний світлодіод
GPIO14	Door_Closed	Лог."0" сигналізує, що комірка зачинена
GPIO15	Led_Green	Лог."0" вмикає зелений світлодіод

Додатково GPIO13 і GPIO15 керують кольорами двокольорового світлодіода для візуальної індикації результату, а GPIO14 використовується для зчитування сигналу з датчика Холу. Уся схема побудована з урахуванням мінімізації енергоспоживання та забезпечує надійне функціонування пристрою в автономному режимі.

Інші порти ESP32 (зокрема GPIO1, GPIO3, GPIO0 тощо) використовуються або для завантаження прошивки, або для підключення камери, тому вони не задіяні в основній логіці керування пристроєм. Обране розподілення забезпечує надійне функціонування системи та мінімізує конфлікти між периферійними пристроями та функціоналом контролера.

Електрична схема системи розроблена на базі структурної схеми системи на рисунку 2.1. Центральним компонентом схеми є модуль ESP32-

CAM, який зображено на рисунку 2.2. Електрична принципова схема сканера QR-коду для відмикання комірки зображено на рисунку 2.10 [15].

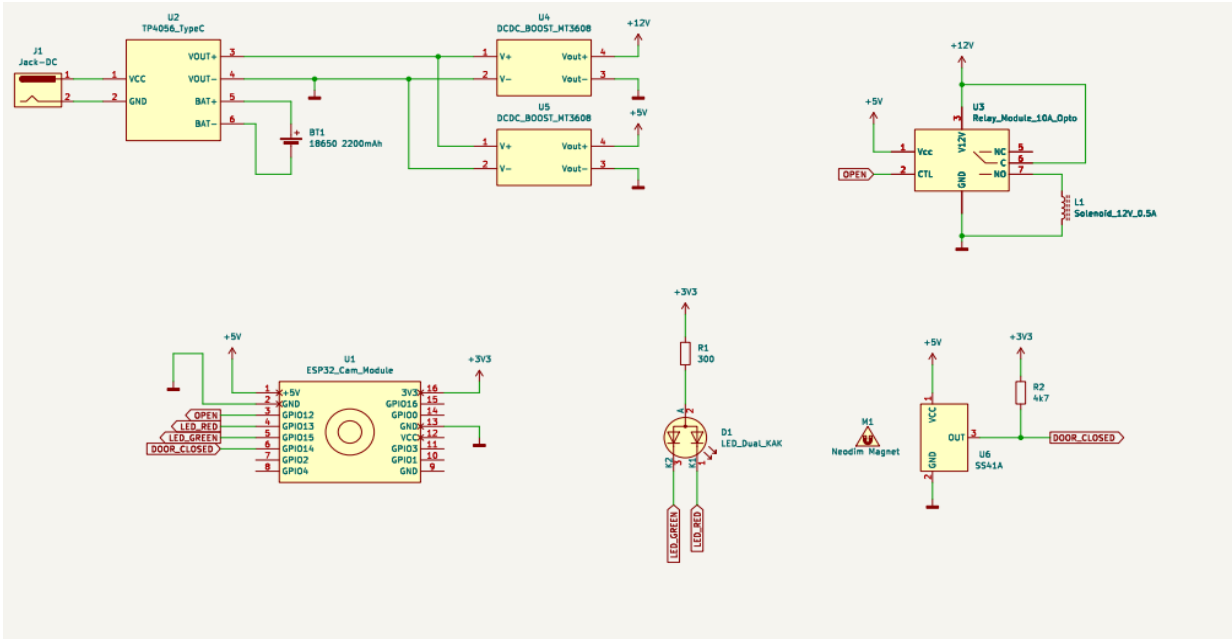


Рисунок 2.10 – Схема електрично-принципова сканера QR-коду для відмикання комірки

Кожен компонент схеми виконує визначену функцію та пов’язаний із мікроконтролером через відповідний порт GPIO.

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1 Реалізація проєктних рішень

На основі розробленої у попередніх розділах архітектури було здійснено поетапну реалізацію апаратної та програмної частини системи. Кожен компонент, передбачений у принциповій електричній схемі, було обґрунтовано з точки зору функціонального призначення, сумісності та енергоефективності. Реалізація охоплює всі рівні функціонування пристрою — від низькорівневих апаратних з'єднань до логіки обміну інформацією з віддаленим сервером через інтернет.

Функціональна логіка роботи апаратної частини представлена у вигляді блок-схеми, яка демонструє послідовність основних дій: очікування у режимі енергозбереження, сканування QR-коду, формування запиту на сервер, перевірка дозволу на відкриття, активація замка та зчитування сигналу про закриття дверцят.

Усі теоретичні рішення, представлені у першому та другому розділах, будуть повністю втілені у практичну реалізацію системи. Структурна блок-схема зображена на рис. 3.1, що описуватиме загальну архітектуру пристрою, стала основою для побудови принципової електричної схеми, у якій кожен обраний компонент виконує чітко визначену функцію. Програмна частина також реалізована відповідно до закладених вимог: мікроконтролер виконує логіку перевірки угоди, взаємодіє із сервером, керує замком, світлодіодом та обробляє сигнал із датчика Холу. Комплексне поєднання апаратної та програмної складових забезпечує повну відповідність функціоналу поставленим задачам і дозволяє системі працювати автономно, надійно та відповідно до сценаріїв, визначених у процесі моделювання.

					КС КРБ 123.247.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Щирба Т.А.			Практична частина		
Перевір.		Шингера Н.Я.					
Реценз.		Михалик Д.М.					
Н. Контр.		Тиш Є.В.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркушів
						31	
					ТНТУ, каф. КС, гр. СІ-42		

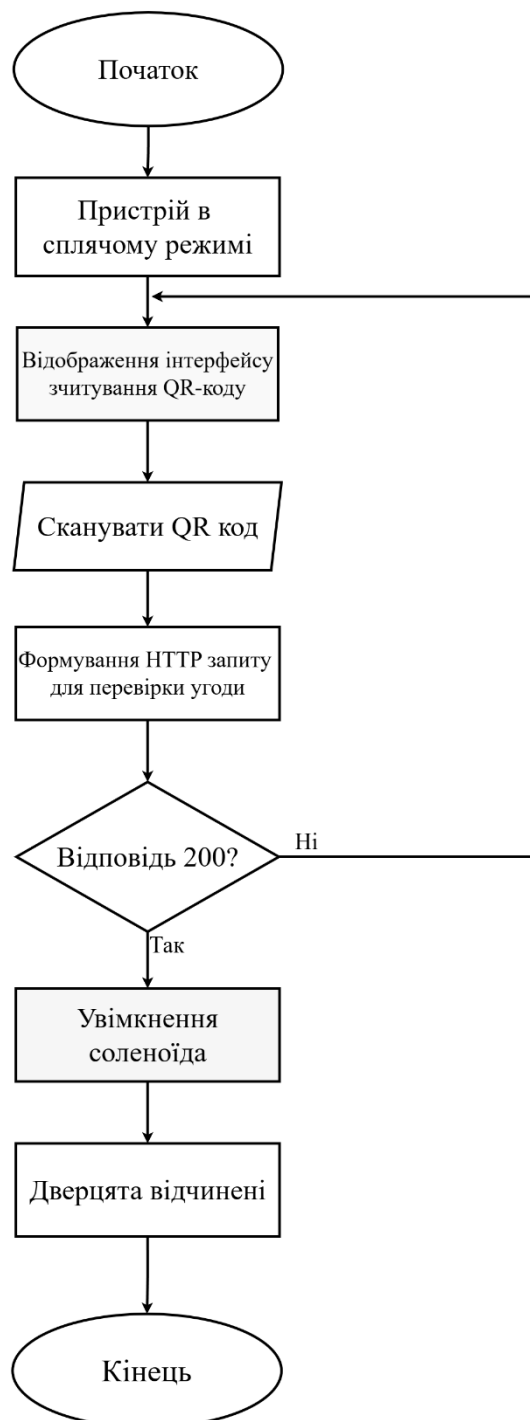


Рисунок 3.1 – Алгоритм роботи книгообмінної будки

Реалізоване рішення передбачає повну інтеграцію апаратної частини з вебінтерфейсом, через який здійснюється облік книг, підтвердження угод та генерація QR-кодів.

3.2 Налаштування середовища для розробки мікроконтролера

Для розробки програмного забезпечення мікроконтролера ESP32-CAM було використано середовище Arduino IDE, яке забезпечує зручний інтерфейс для написання, компіляції та завантаження прошивки на плату. Обрана платформа підтримує широкий спектр мікроконтролерів і має активну спільноту, що спрощує інтеграцію сторонніх бібліотек та модулів [16].

ESP32 не входить до стандартного набору плат Arduino IDE, тому необхідно вручну підключити репозиторій розробника:

1. У меню Файл, далі налаштування (або File, далі Preferences) знайти поле: «Додаткові URL-адреси для менеджера плат».
2. Додати підтримку плати ESP32-CAM до Arduino IDE.
3. Натиснути ОК (див. рис. 3.2).

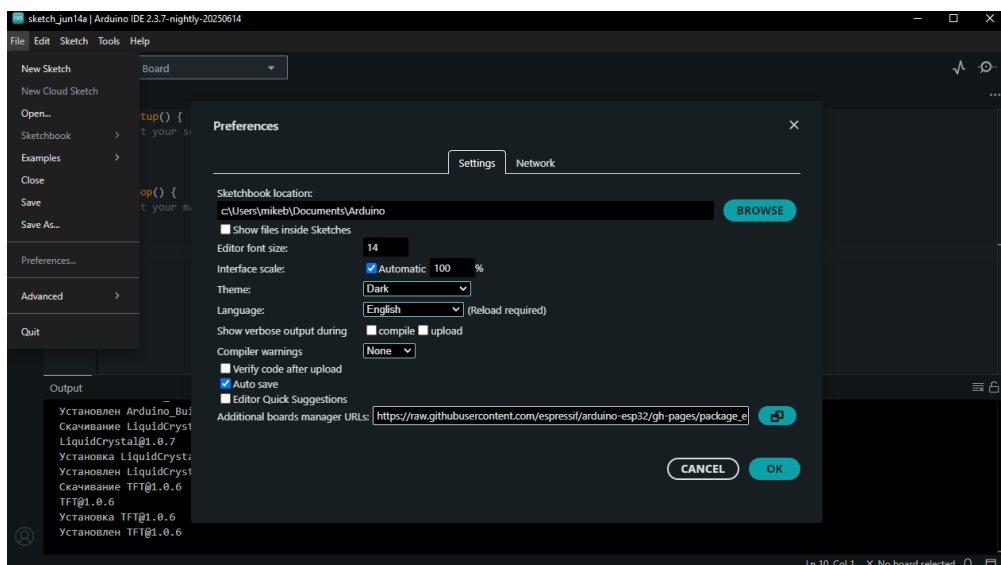


Рисунок 3.2 – Налаштування підтримки ESP32-CAM в Arduino IDE

Після цього в Інструменти, далі Менеджер плат необхідно знайти й встановити "esp32 by Espressif Systems" (див. рис. 3.3).

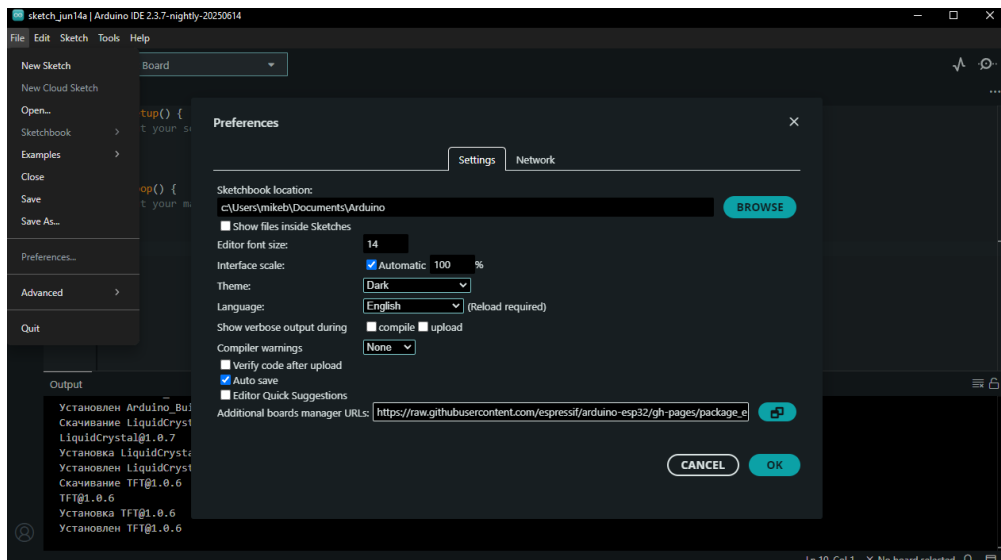


Рисунок 3.3 – Інсталяція пакетів esp32 by Espressif Systems

Після встановлення підтримки платформ ESP32 в Arduino IDE необхідно правильно обрати модель плати, яка відповідає використовуваному апаратному модулю.

У проєкті застосовується мікроконтролер ESP32-CAM, що базується на архітектурі AI Thinker. Тому у меню середовища розробки слід перейти до пункту «Інструменти, далі Плата», де в розділі ESP32 Arduino обрати варіант «AI Thinker ESP32-CAM». Саме ця модель повністю відповідає конфігурації популярних модулів із вбудованою камерою OV2640, слотом для карти пам'яті microSD та стандартним розміщенням GPIO-виводів [16].

Після вибору плати необхідно переконатися, що інші параметри компіляції встановлені коректно.

Параметри компіляції - це налаштування, які визначають, як саме буде зібрано програму перед її завантаженням у мікроконтролер, в книгообмінній будці де використовується ESP32-CAM, після цих налаштувань має:

- працювати камера (OV2640);
- ESP32 має доступ до Wi-Fi;
- керування піном для реле й світлодіодів коректне;
- не виникають помилки при завантаженні прошивки.

У меню «Інструменти» задаються наступні значення, які описано в таблиці 3.1

					КС КРБ 123.247.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Таблиця 3.1 – Параметри компіляції

Швидкість завантаження	115200 бод
Схема розділення пам'яті	Default (1MB SPIFFS)
Частота Flash-пам'яті	40 MHz
Режим Flash-пам'яті	QIO
Рівень налагодження	немає
Порт	автоматично визначається після підключення USB-UART адаптера до комп'ютера

На відміну від більшості плат Arduino, модуль ESP32-CAM не має вбудованого USB-інтерфейсу для програмування. Тому для завантаження прошивки до пристрою необхідно використовувати зовнішній USB-UART адаптер, підключення виконується за схемою описаній в таблиці 3.2

Таблиця 3.2 – Таблиця відповідності контактів при підключенні ESP32-CAM до USB-UART адаптера

ESP32-CAM	USB-UART адаптер
U0R	TX (передає)
U0T	RX (приймає)
GND	GND
5V	VCC (5 V)
IO0	GND (тільки під час прошивки)

Для стабільної роботи з Wi-Fi, HTTP-запитами, цифровими сенсорами та керуванням виходами в прошивці було використано бібліотеки які описано в таблиці 3.3

Таблиця 3.3 – Бібліотеки для взаємодії з компонентами

WiFi.h	підключення до мережі
HTTPClient.h	для формування та відправки HTTP-запитів на сервер
ESP32Servo.h	у випадку використання сервоприводів
Arduino.h	базова бібліотека для роботи з мікроконтролером
FS.h/SPIFFS.h	у разі збереження налаштувань або зображень локально

Якщо в прошивці передбачено ініціалізацію Wi-Fi, формування HTTP-запитів, керування GPIO-виводами, а також обробку даних із сенсорів або камери — ці компоненти активуються згідно з закладеною логікою.

3.3 Опис прошивки ESP32 та логіки керування

У структурі програми виділено декілька окремих функціональних блоків, кожен з яких виконує окрему частину логіки пристрою, значення функцій описано в таблиці 3.4

Таблиця 3.4 – Функціональні компоненти прошивки

scanQRCode()	зчитування URL з QR-коду
checkAccess(String url)	надсилає HTTP-запит на сервер для перевірки статусу угоди
unlock()	активує реле та відкриває замок на заданий час
readHallSensor()	зчитує стан дверцят через цифровий датчик Холу
indicateAccessGranted(), indicateAccessDenied()	візуально повідомляють користувача про результат через світлодіодну індикацію

Програмне забезпечення для мікроконтролера ESP32-CAM реалізоване у вигляді скетчу на мові C++ із використанням бібліотек платформи Arduino. Загальна структура прошивки відповідає стандартній архітектурі, що включає

дві основні функції — `setup()` і `loop()`, а також набір допоміжних функцій для реалізації логіки роботи пристрою.

Функція `setup()` виконується один раз при запуску або перезапуску пристрою. У ній відбувається ініціалізація пінів вводу/виводу (GPIO), підключення до Wi-Fi-мережі, а також початкова індикація стану системи, функція описана в лістингу 3.1.

Лістинг 3.1 – Лістинг функції `setup()`

```
// --- Налаштування портів GPIO ---

#define RELAY_PIN          12  // Пін для керування реле.
// Логічний 0 активує реле (вмикає замок)
#define HALL_SENSOR_PIN    14  // Пін, на який підключено
// цифровий датчик Холу для перевірки закриття дверцят
#define RED_LED_PIN        13  // Пін для червоного
// світлодіодного каналу (сигналізує про відмову)
#define GREEN_LED_PIN       15  // Пін для зеленого
// світлодіодного каналу (сигналізує про дозвіл доступу)

// --- Змінні для зберігання параметрів Wi-Fi ---
const char* ssid           = "Назва_Вашої_Мережі";           // Ім'я
// Wi-Fi мережі (SSID)
const char* password       = "Пароль_Від_WiFi";             // Пароль
// до Wi-Fi

void setup() {
    // Ініціалізація портів
    pinMode(RELAY_PIN, OUTPUT);           // Встановлюємо пін
// реле як вихід
    pinMode(RED_LED_PIN, OUTPUT);         // Встановлюємо пін
// червоного світлодіода як вихід
    pinMode(GREEN_LED_PIN, OUTPUT);       // Встановлюємо пін
// зеленого світлодіода як вихід
    pinMode(HALL_SENSOR_PIN, INPUT);      // Пін датчика Холу
// працює як вхід

    // Початкові значення
    digitalWrite(RELAY_PIN, HIGH);        // Вимикаємо реле
// (лог. 1 = замок вимкнено)
    digitalWrite(RED_LED_PIN, LOW);       // Світлодіоди
// вимкнені
    digitalWrite(GREEN_LED_PIN, LOW);

    // --- Ініціалізація серійного порту ---
    Serial.begin(115200);                 // Вмикаємо серійний
// монітор для діагностики
    delay(1000);                          // Невелика затримка
// перед підключенням Wi-Fi
}
```

					КС КРБ 123.247.00.00 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

// --- Ініціалізація Wi-Fi ---
WiFi.begin(ssid, password);           // Підключаємося до
Wi-Fi
Serial.print("Підключення до Wi-Fi");

while (WiFi.status() != WL_CONNECTED) {
    delay(500);                       // Чекаємо з'єднання
    Serial.print(".");
}

// Виводимо локальну IP-адресу після підключення
Serial.println("");
Serial.println("Wi-Fi підключено");
Serial.print("IP-адреса пристрою: ");
Serial.println(WiFi.localIP());
}

```

Лістинг основної логіки для loop() на ESP32-CAM, яка зчитує QR-код, перевіряє угоду через HTTP-запит до сервера, та керує реле/замком і світлодіодом та описано в лістингу 3.2.

Лістинг 3.2 – Лістинг функції loop()

```

#include <WiFi.h>
#include <HTTPClient.h>

// Функція перевірки доступу до комірки
bool checkDealAccess(String dealUrl) {
    HTTPClient http;
    http.begin(dealUrl); // URL, закодований у QR-коді
    int httpCode = http.GET();

    if (httpCode == 200) {
        String payload = http.getString();
        http.end();
        if (payload == "access_granted") {
            return true;
        }
    }

    http.end();
    return false;
}

// Функція відкриття замка
void unlockBox() {
    digitalWrite(RELAY_PIN, LOW);           // Активуємо реле
(логічний нуль)
    digitalWrite(GREEN_LED_PIN, HIGH);     // Вмикаємо зелений

```

					КС КРБ 123.247.00.00 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

індикатор
    delay(3000); // Тримаємо замок
відкритим 3 секунди
    digitalWrite(RELAY_PIN, HIGH); // Вимикаємо реле
(замок зачиняється)
    digitalWrite(GREEN_LED_PIN, LOW); // Вимикаємо
індикатор
}

// Функція відмови в доступі
void denyAccess() {
    digitalWrite(RED_LED_PIN, HIGH);
    delay(2000);
    digitalWrite(RED_LED_PIN, LOW);
}

String getScannedQR() {
    return "http://yourserver.com/unlock.php?deal_id=123"; //
приклад
}

void loop() {
    String scannedUrl = getScannedQR(); // ←

    // Чекаємо поки дверцята закриті
    if (digitalRead(HALL_SENSOR_PIN) == LOW) {
        if (checkDealAccess(scannedUrl)) {
            unlockBox();
        } else {
            denyAccess();
        }
    } else {
        Serial.println("Дверцята відчинені. Очікування
закриття...");
        delay(2000); // Очікуємо перед повторною перевіркою
    }

    delay(5000); // Чекаємо між ітераціями, щоб уникнути
надмірного опитування
}

```

Функція перевірки угоди через HTTP-запит описано в лістингу 3.3

					КС КРБ 123.247.00.00 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг 3.3 – Лістинг checkAccess(String url)

```
#include <HTTPClient.h>

// Перевірка статусу угоди по URL з QR-коду
bool checkDealAccess(String dealUrl) {
    HTTPClient http;
    http.begin(dealUrl);
    int httpCode = http.GET();

    if (httpCode == 200) {
        String response = http.getString();
        http.end();
        return (response == "access_granted");
    } else {
        http.end();
        return false;
    }
}
```

Після підтвердження доступу система активує виконавчий механізм. Реле комутується логічним нулем, подаючи живлення на соленоїдний замок. Після затримки замок знову блокується, лістинг активації реле та розблокування комірки, що описано в лістингу 3.4.

Лістинг 3.4 – Лістинг функцій unlock() та readHallSensor()

```
void unlock() {
    digitalWrite(RELAY_PIN, LOW);           // Активуємо реле
    digitalWrite(GREEN_LED_PIN, HIGH);     // Візуальна
індикація
    delay(3000);                           // Затримка
відкриття
    digitalWrite(RELAY_PIN, HIGH);          // Вимикаємо реле
    digitalWrite(GREEN_LED_PIN, LOW);       // Вимикаємо
індикатор
}
bool readHallSensor() {
    return (digitalRead(HALL_SENSOR_PIN) == LOW); // LOW –
дверцята зачинені
}
```

					КС КРБ 123.247.00.00 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Структурованість коду, модульний підхід до реалізації функцій та використання стандартних бібліотек платформи Arduino дозволяють легко адаптувати програму до змін у конфігурації або розширення функціоналу. Завдяки взаємодії з сервером через HTTP-запити пристрій здатен приймати рішення про надання доступу в режимі реального часу, що забезпечує синхронність між апаратною частиною та вебінтерфейсом.

3.4 Розробка сайту для обліку книг і підтвердження угод

3.4.1 Мова програмування та технології

Супровідною частиною функціонування апаратної системи виступає вебінтерфейс, який реалізує адміністрування користувачів, облік книг, ініціацію угод та підтвердження доступу. Для розробки вебсайту було використано зв'язку технологій: PHP (на стороні сервера), MySQL (для зберігання даних), Bootstrap 5 (для інтерфейсної верстки) та JavaScript (для динамічних клієнтських елементів).

Архітектура вебресурсу побудована за класичною моделлю клієнт–сервер, де користувачі взаємодіють із інтерфейсом через браузер, а серверна частина виконує обробку запитів, формування HTML-відповідей, доступ до бази даних, перевірку прав доступу та генерацію динамічного вмісту (див. рис. 3.8) [17].

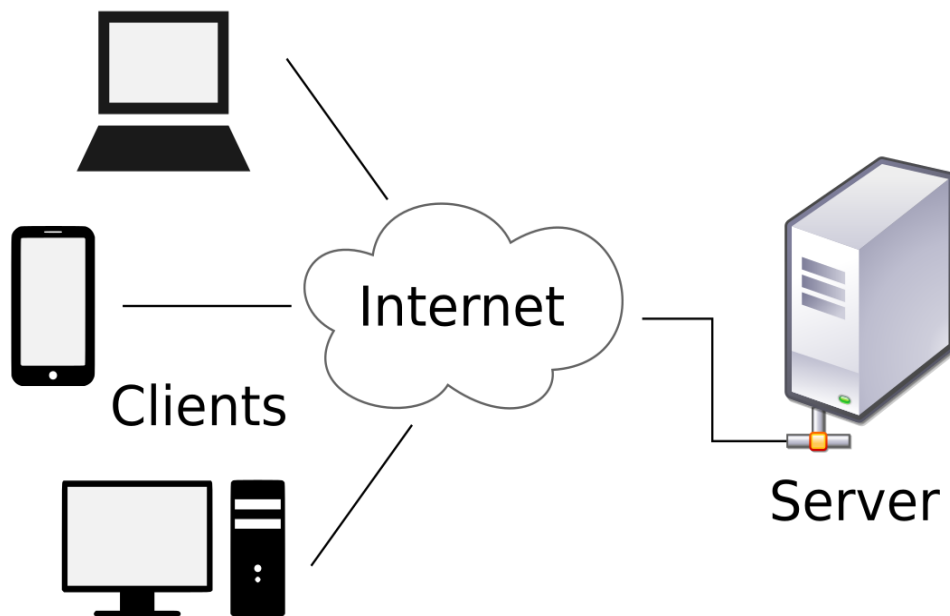


Рисунок 3.8 – Клієнт-серверна архітектура

Обрана зв’язка технологій забезпечила ефективну взаємодію клієнтської та серверної частини, дозволяючи реалізувати всі необхідні функції для обліку книг та підтвердження угод у межах єдиної інтегрованої платформи.

Користувач працює через браузер, де формуються запити HTTP-методами (GET або POST), що передаються до серверних обробників. Сервер, у свою чергу, виконує необхідні дії — звернення до бази даних, перевірку прав доступу, зміну статусів або формування HTML/JSON-відповіді — й надсилає результат клієнтові.

Клієнтська частина відповідає за візуалізацію сторінки, обробку форм, відображення книг, повідомлень і результатів взаємодії. Серверна частина обробляє всю логіку: реєстрацію, авторизацію, створення чатів, додавання книг, створення та підтвердження угод, керування адміністративною панеллю, а також формування QR-кодів для доступу до будки.

Усі компоненти, реалізовані на PHP, поділяються за функціональним призначенням:

- `accept_deal.php` та `confirm_deal.php` — відповідають за зміну статусу угоди на «підтверджено» після того, як обидві сторони погодились на обмін. Ці модулі оновлюють поля `from_confirmed`, `to_confirmed` і `status` у

					<i>КС КРБ 123.247.00.00 ПЗ</i>	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

таблиці deals;

- add_book.php — обробляє додавання книги до системи. Логіка включає перевірку статусу (обмін чи продаж), автоматичне встановлення ціни (0 грн для обміну) та запис даних у таблицю books;

- admin_dashboard.php — формує адміністративну панель, де відображаються списки користувачів, книг і звернень до підтримки. Також реалізовано механізм відповіді на звернення з оновленням відповідного запису в БД;

- book.php — завантажує дані про книгу за її ID, а також виконує фільтрацію за жанрами, використовуючи параметри з URL. Це дозволяє формувати динамічний каталог із можливістю відбору;

- chat.php — реалізує чат між двома користувачами. Крім відображення повідомлень, файл виконує перевірку доступу до чату, обробляє статуси угод, перевіряє підтвердження сторін і, за потреби, змінює статус угоди на «виконано» (completed);

- create_deal.php — створює нову угоду між користувачами після ініціації в чаті. Перевіряє, чи існує книга, чи чат належить користувачам, і чи немає вже активної угоди, після чого записує дані в таблицю deals;

- contact.php — реалізує форму підтримки. Користувач може надіслати звернення, яке зберігається в support_messages, а історія звернень виводиться на тій самій сторінці;

- start_chat.php — створює новий чат, якщо ще не існує. Перевіряє відповідність користувачів і книги, уникаючи повторного створення для тієї самої пари;

- scanner.html — сторінка для зчитування QR-кодів із мобільного пристрою. Після зчитування браузер автоматично перенаправляє користувача на URL, який міститься в коді, що дозволяє ініціювати HTTP-запит на відкриття будки.

Компоненти взаємодіють через єдину базу даних та використовують систему сесій для контролю доступу.

					КС КРБ 123.247.00.00 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

3.4.2 Реєстрація та авторизація користувачів

Однією з базових функцій вебплатформи є механізм реєстрації та авторизації користувачів, що забезпечує персоналізований доступ до функціоналу системи та дозволяє зберігати історію взаємодії кожного користувача з сервісом. Цей механізм реалізовано за допомогою серверного програмування на мові PHP з використанням бази даних MySQL для зберігання облікових записів.

Реєстрація передбачає створення нового облікового запису шляхом заповнення користувачем відповідної форми, у якій вказуються ім'я, прізвище, електронна адреса, пароль та інші допоміжні атрибути. Паролі зберігаються в базі у зашифрованому вигляді за допомогою функції хешування, що виключає прямий доступ до чутливої інформації навіть у разі витоку даних (див. рис. 3.9).

Реєстрація

Ім'я

Прізвище

Електронна пошта

Пароль

Підтвердження паролю

Зареєструватися

[Вже маєте акаунт? Увійдіть](#)

Рисунок 3.9 – Реєстрація нового користувача на веб сайті

Після реєстрації користувач має змогу пройти авторизацію, вводячи логін і пароль у відповідну форму. Дані перевіряються на сервері, і у разі збігу збережених значень з введеними — створюється сесія, що фіксує стан користувача як автентифікованого. Сесійний механізм забезпечує

					КС КРБ 123.247.00.00 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

безперервний доступ до персонального кабінету, дозволяє відслідковувати поточну активність користувача та формує основу для обмеження доступу до певних сторінок ресурсу (див. рис. 3.10).

Рисунок 3.10 – Авторизація на сайті

Механізми авторизації включають перевірку прав доступу, захист від несанкціонованого доступу та обмеження прямого переходу до внутрішніх частин сайту без входу до системи. Усі запити, що вимагають ідентифікації користувача (наприклад, перегляд профілю, ініціація угоди, доступ до чату), супроводжуються перевіркою активної сесії. У разі її відсутності користувача автоматично перенаправляють на сторінку входу.

3.4.3 Додавання книг із вибором статусів «обмін» або «продаж»

Можливість користувача додавати власні книги до системи для подальшого обміну або продажу реалізована через інтерфейс особистого кабінету, де зареєстрований користувач може заповнити форму додавання книги із зазначенням базових атрибутів.

Форма передбачає введення таких параметрів, як назва книги, автор, жанр, опис, зображення обкладинки, а також вибір одного зі статусів: «обмін»

					КС КРБ 123.247.00.00 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

або «продаж». Обраний статус безпосередньо впливає на подальший функціонал взаємодії з цією книгою. У випадку, якщо обрано статус «обмін», система автоматично встановлює вартість книги як нульову, що відображає її недоступність для купівлі. Якщо ж обрано статус «продаж», користувач додатково має змогу вказати бажану ціну, яка зберігається у відповідному полі бази даних.

Уся інформація про книгу після додавання зберігається у таблиці books у базі MySQL та прив'язується до ідентифікатора користувача, що дозволяє реалізувати відображення лише тих книг, якими володіє відповідний акаунт. Водночас усі додані книги доступні у загальному каталозі, де їх можуть переглядати інші користувачі, фільтрувати за жанром, статусом або іншими параметрами (див. рис. 3.11).

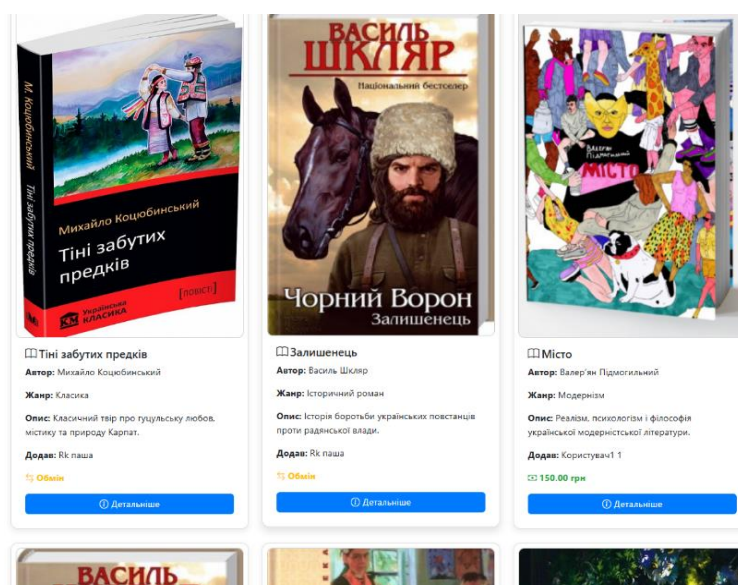


Рисунок 3.11 – Загальний каталог з книгами

Механізм додавання книги також виконує перевірку введених даних на стороні сервера. Якщо деякі поля не заповнено або містять недопустимі значення (наприклад, відсутність ціни у режимі продажу або нечислове значення вартості), система повертає повідомлення про помилку та пропонує користувачеві внести коригування. Після успішного додавання книга миттєво стає доступною для перегляду іншими користувачами, що дозволяє розпочати комунікацію в межах платформи.

					КС КРБ 123.247.00.00 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

3.4.4 Профілі користувачів

Профіль користувача є центральним елементом персоналізованої взаємодії з вебплатформою. Він містить ключову інформацію про зареєстровану особу, а також об'єднує в єдиному інтерфейсі всі дії, здійснені цим користувачем у межах системи. Основною метою реалізації профілю є забезпечення доступу до персональних даних, історії взаємодії, списку доданих книг, активних та завершених угод.

Інтерфейс профілю включає відображення базових персональних відомостей: ім'я, прізвище, електронна адреса, а також аватар (при бажанні користувача). У розділі профілю виводиться перелік усіх книг, які було додано цим користувачем із зазначенням їхніх атрибутів — назви, автора, жанру, статусу («обмін» або «продаж») та поточного стану (вільна, у процесі угоди, завершена) (див. рис. 3.12).

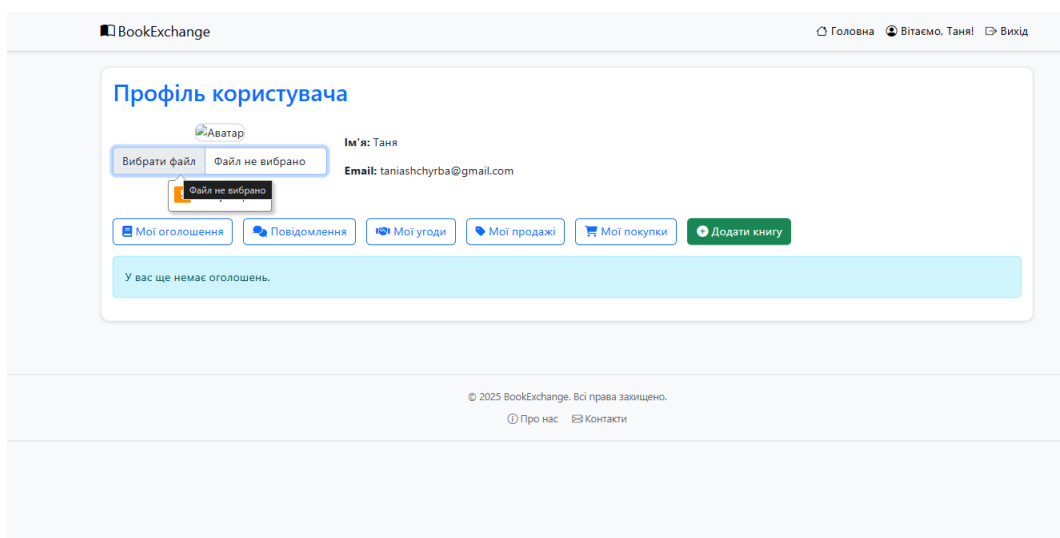


Рисунок 3.12 – Профіль користувача на веб-сайті

Окремий важливий підрозділ профілю присвячено угодам. Користувач має змогу переглядати статуси всіх угод, у яких він брав участь — як ініціатор, так і отримувач. Кожна угода містить дані про відповідну книгу, іншу сторону, час створення, підтвердження та результат. Якщо угода перебуває у стані очікування підтвердження або завершення, користувач може перейти до чату для подальшої взаємодії.

					КС КРБ 123.247.00.00 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

3.4.5 Система чатів між користувачами

Для забезпечення прямої комунікації між учасниками системи у вебплатформі реалізовано внутрішню систему чатів. Основною функцією цього модуля є надання можливості обговорення умов обміну або продажу книги до моменту створення угоди. Завдяки чатам користувачі можуть уточнювати деталі, ставити запитання, домовлятись про зручний час чи спосіб передачі книги.

Ініціація чату відбувається автоматично при першому зверненні одного користувача до іншого щодо конкретної книги. Кожен чат є двостороннім, закріпленим за унікальною парою користувачів та конкретною книгою. Повторне звернення до тієї ж книги не створює новий чат, а відкриває вже наявний, що дозволяє уникнути дублювання діалогів.

Для захисту інформації та підтримання цілісності даних кожен чат доступний лише його учасникам. Перед завантаженням вмісту виконується перевірка, чи поточний автентифікований користувач є стороною відповідного чату. У разі невідповідності система блокує доступ і повертає повідомлення про помилку (див. 3.13).

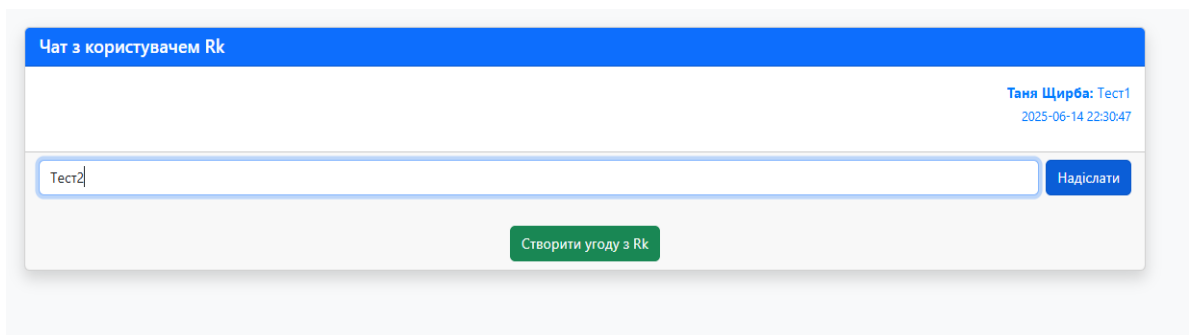


Рисунок 3.13 – Чатування на веб-сайті

Система повідомлень реалізована на основі асинхронної передачі даних, де кожне нове повідомлення зберігається у базі даних із прив'язкою до ID чату, автора, часу відправлення та вмісту. При завантаженні сторінки чату виконується вибірка всіх повідомлень, що дозволяє користувачам переглядати повну історію листування. Вивід повідомлень оформлено у форматі

					КС КРБ 123.247.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

«відправник – текст – час», із візуальним поділом за автором.

3.4.6 Система створення та підтвердження угоди

Створення угоди можливе лише в межах активного чату, що забезпечує попередню комунікацію між користувачами. Один із учасників може ініціювати угоду, обравши конкретну книгу іншого користувача. Після цього в базі даних створюється запис угоди зі статусом «очікує підтвердження» (pending), у якому зберігаються ідентифікатори обох сторін, ID книги, дата створення, тип операції (обмін або продаж), а також ознака підтвердження з кожного боку.

Сторони угоди мають змогу незалежно підтвердити або відхилити пропозицію. Для підтвердження користувач переходить до відповідного чату, де доступна кнопка прийняття умов. У разі якщо обидва учасники підтвердили свою згоду, статус угоди змінюється на «підтверджено» (accepted), що дозволяє перейти до наступного етапу — генерації QR-коду для фізичного відкриття будки. Якщо хоча б одна зі сторін не погодиться, угоду буде анульовано або залишено у статусі очікування без наслідків.

Кожен крок у процесі підтвердження супроводжується записом часової мітки, що дозволяє зафіксувати порядок дій і уникнути спірних ситуацій. Також реалізовано перевірку на наявність уже активної угоди в межах поточного чату, щоб уникнути дублювання або конфлікту транзакцій. Це гарантує, що на момент відкриття комірки буде дійсна лише одна угода, що підтверджена обома сторонами.

Механізм реалізовано у вигляді серверного скрипта, що змінює статус угоди в базі даних після взаємної згоди. При цьому до кожної угоди формується унікальний ідентифікатор, який в подальшому використовується для створення QR-коду та перевірки доступу з боку апаратної частини системи.

					КС КРБ 123.247.00.00 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

3.5 Генерація QR-кодів та перевірка угоди через сайт

Після підтвердження угоди обома сторонами вебплатформа автоматично генерує унікальний QR-код, який містить посилання на серверний обробник, до якого додається параметр з ідентифікатором угоди. Такий QR-код слугує електронним ключем доступу до будки і є обов'язковим елементом для подальшого фізичного відкриття комірки через апаратну частину.

У розділі «Мої угоди» QR-код супроводжується інформацією про книгу, учасників та дату підтвердження, що дозволяє швидко отримати доступ до необхідних даних під час використання пристрою.

Сканування коду здійснюється з боку користувача — наприклад, за допомогою смартфона або безпосередньо через камеру ESP32-CAM. У разі коли QR-код потрапляє до мікроконтролера, він ініціює HTTP-запит до сервера за зазначеним URL. Серверний скрипт `unlock.php` приймає параметр `deal_id`, виконує запит до бази даних і перевіряє наступні дані:

- чи існує така угода;
- чи має вона статус «підтверджено»;
- чи обидва користувачі підтвердили участь;
- чи угода ще не була використана.

У разі відповідності всім умовам сервер формує JSON-відповідь зі статусом `access_granted`, що сигналізує ESP32 про дозвіл відкриття замка. У разі помилки або недійсного запиту повертається відповідь `access_denied`, після чого пристрій блокує виконання дії.

Для запобігання повторному або несанкціонованому використанню QR-коду після першого успішного звернення до `unlock.php` угода автоматично змінює статус на «виконано» (`completed`).

					КС КРБ 123.247.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

3.6 База даних

Під час проєктування бази даних було дотримано основних принципів нормалізації, зокрема уникнення надлишкового дублювання інформації та забезпечення логічного поділу даних на окремі сутності. Кожна таблиця відповідає одній логічній одиниці, а зв'язки між ними реалізовані через зовнішні ключі, що забезпечує цілісність та узгодженість інформації на всіх етапах взаємодії користувачів із системою.

Для забезпечення збереження, обробки та структурованого доступу до даних у межах вебплатформи було розроблено реляційну базу даних, реалізовану на основі системи управління базами даних MySQL.

У межах реалізації проєкту система бази даних виконує ключову роль у забезпеченні цілісності, структурованості та швидкого доступу до даних. Основними об'єктами зберігання є користувачі, книги, угоди, повідомлення чату, замовлення та звернення до служби підтримки. Організація даних реалізована на основі реляційної моделі з використанням СУБД MySQL, що забезпечує ефективну роботу з великою кількістю зв'язаних сутностей та підтримує механізми транзакційності, цілісності й оптимізації запитів.

Застосування індексів на полях пошуку (ID користувача, статус угоди тощо) дозволяє досягти високої швидкодії в процесі запитів. Усі дані, що обробляються системою, мають часові мітки, які фіксують дії користувача й дозволяють реалізувати хронологічний аналіз або аудит.

Модель даних охоплює всі ключові сутності, пов'язані з функціонуванням системи: користувачів, книги, чати, угоди, замовлення, повідомлення підтримки, а також адміністративні облікові записи [18].

Структура бази побудована за принципами нормалізації та логічного зв'язування таблиць через зовнішні ключі, що дозволяє забезпечити цілісність даних та уникати дублювання інформації. В основі лежить таблиця users, яка виступає центральною сутністю для більшості взаємозв'язків. Решта таблиць

					<i>КС КРБ 123.247.00.00 ПЗ</i>	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

пов'язані з нею безпосередньо або опосередковано через інші сутності (наприклад, deals, orders, messages) [18].

У моделі реалізовано підтримку двосторонніх чатів з обмеженням доступу лише для відповідних користувачів, а також механізм зберігання повідомлень із часовими мітками. Важливим компонентом є таблиця угод (deals), у якій відображається процес погодження між сторонами та поточний стан транзакції. Це забезпечує можливість автоматичного підтвердження доступу до фізичної будки лише у разі успішного погодження обома учасниками [18].

Оскільки взаємодія з базою здійснюється виключно через серверні скрипти, без прямого доступу з клієнтської сторони, це значно знижує ризики несанкціонованого втручання та підвищує загальну стабільність системи.

Облікові дані користувачів, зокрема паролі, зберігаються у вигляді хешів із використанням криптографічно стійких алгоритмів (наприклад, bcrypt через функцію password_hash() у PHP).

Звернення до служби підтримки та їх обробка адміністратором фіксуються у відповідних таблицях, що дозволяє зберігати історію комунікацій та відповіді. Адміністративна таблиця (admin) має окрему систему авторизації та не перетинається з базовою системою користувачів, що підвищує безпеку розмежування прав доступу.

Загальна структура бази даних представлена у таблиці 3.5.

					КС КРБ 123.247.00.00 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.5 – Структура бази даних вебплатформи

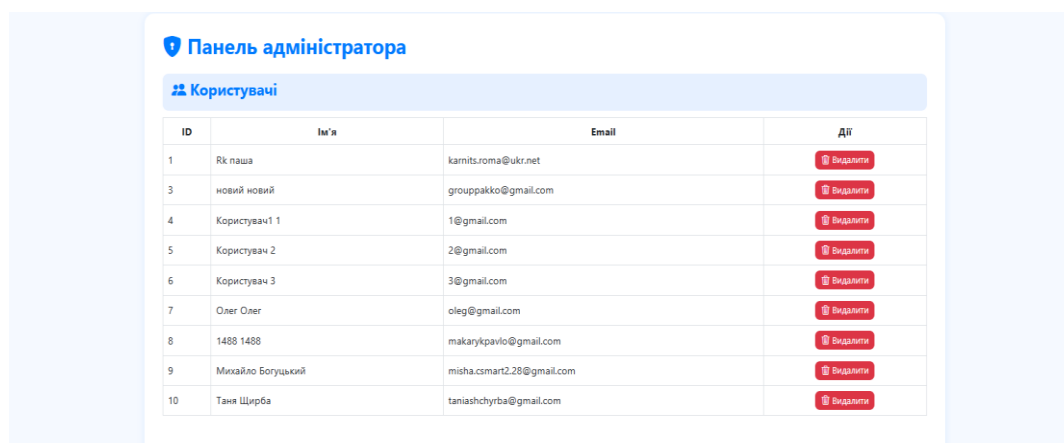
№	Назва таблиці	Призначення таблиці	Основні поля
1	users	Зберігає облікові записи користувачів	idusers, first_name, last_name, email, password, avatar, created_at
2	books	Містить дані про книги, додані користувачами	idbooks, title, author, genre, description, price, status, images, created_at
3	chats	Фіксує чати між двома користувачами щодо певної книги	idchats, user1_id, user2_id, book_id, created_at
4	messages	Зберігає окремі повідомлення між користувачами	idmessages, chat_id, sender_id, message, created_at
5	deals	Містить інформацію про створені угоди	iddeals, chat_id, book_id, from_user_id, to_user_id, status, from_confirmed, to_confirmed, created_at
6	orders	Відображає прямі замовлення книг у випадку продажу	idorders, user_id, book_id, status, created_at
7	support_messages	Містить звернення користувачів до адміністрації	idsupport_messages, user_id, messages, created_at
8	admin	Зберігає облікові дані адміністратора	idadmin, login, password

Розроблена структура бази даних охоплює всі ключові аспекти логіки функціонування системи та дозволяє ефективно реалізувати як типові, так і взаємозалежні операції між користувачами. Завдяки чіткій організації таблиць і визначеним зв'язкам між ними забезпечується узгодженість інформації, а також можливість швидкого виконання запитів на читання, оновлення та перевірку умов доступу.

Наявність окремих таблиць для угод, повідомлень, замовлень і адміністративних звернень дозволяє масштабувати систему без втрати цілісності або функціональності. Такий підхід формує надійну основу для подальшого розвитку платформи, включно з аналітикою, статистикою або інтеграцією з апаратними модулями.

3.7 Керування веб-сайтом на рівні адміністратора

Однією з основних функцій адміністративного розділу є перегляд усіх зареєстрованих користувачів. У таблиці відображається базова інформація про кожен обліковий запис — ім'я, прізвище, електронна адреса, дата реєстрації, а також активність у системі (див. рис. 3.14).



Панель адміністратора

Користувачі

ID	Ім'я	Email	Дії
1	Rk паша	kamits.roma@ukr.net	 Видалити
3	новий новий	grouppakko@gmail.com	 Видалити
4	Користувач11	1@gmail.com	 Видалити
5	Користувач 2	2@gmail.com	 Видалити
6	Користувач 3	3@gmail.com	 Видалити
7	Олег Олег	oleg@gmail.com	 Видалити
8	1488 1488	makarykpavlo@gmail.com	 Видалити
9	Михайло Богущкий	misha.csmart2.28@gmail.com	 Видалити
10	Таня Щирба	taniashchyba@gmail.com	 Видалити

Рисунок 3.14 – Таблиця зареєстрованих користувачів

Це дозволяє відстежувати загальний стан платформи та, за потреби, здійснювати блокування або деактивацію облікових записів.

Окрім цього, адміністратор має доступ до повного списку книг, які було додано користувачами. Для кожної книги доступний її опис, статус, власник та час додавання (див. рис. 3.15).

Книги

Назва	Автор	Користувач	Статус	Дії
Тіні забутих предків	Михайло Коцюбинський	Rk паша	exchange	РедагуватиВидалити
Залишенець	Василь Шкляр	Rk паша	exchange	РедагуватиВидалити
Місто	Валер'ян Підмогильний	Користувач 1	available	РедагуватиВидалити
Чорний ворон	Василь Шкляр	Користувач 1	exchange	РедагуватиВидалити
Кайдашева сім'я	Іван Нечуй-Левицький	Користувач 2	exchange	РедагуватиВидалити
Лісова пісня	Леся Українка	Користувач 2	exchange	РедагуватиВидалити

Рисунок 3.15 – Повний список книг на веб сайті

У разі виявлення невідповідного вмісту або порушення правил користування системою, адміністратор може заблокувати окрему книгу, зробивши її невидимою для інших, не видаляючи при цьому сам запис з бази.

Адміністративна панель також містить розділ для керування чатами між користувачами (див. рис. 3.16).

Повідомлення (чат)

Олег Олег:

Привіт

2025-05-01 10:47:41

Відповідь адміністратора:

Привіт

Відповідь надіслана: 2025-05-01 10:48:07

Rk паша:

Потрібна допомога \

2025-05-01 10:23:58

Відповідь адміністратора:

з чим?

Відповідь надіслана: 2025-05-01 10:27:02

Вийти

Рисунок 3.16 – Керування чатом зі сторони адміністратора

Адміністратор має змогу переглядати зміст діалогів, що може бути корисним у випадках виникнення скарг або порушень. Крім того, реалізована можливість залишати відповіді на звернення, які користувачі надсилають через вбудовану форму підтримки.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Навчання працюючих та інструктажі з охорони праці

Однією з важливих складових забезпечення безпечних умов праці в ІТ-сфері, зокрема при підготовці фахівців інженерів, є навчання, підвищення кваліфікації та постійне оновлення знань з питань охорони праці. Загальні вимоги навчання визначає Закон України «Про охорону праці» (стаття 18 «Навчання з питань охорони праці»).

Проведення навчання і перевірки знань з охорони праці регламентується Типовим положенням про порядок проведення навчання і перевірки знань з питань охорони праці, затверджене наказом Держкомітету України з нагляду за охороною праці від 26.01.2005 № 15. Цей документ встановлює єдиний порядок організації навчання як для працівників підприємств, так і для майбутніх фахівців у закладах вищої освіти [23].

Основним нормативним документом, що встановлює порядок та види навчання і перевірки знань з охорони праці є НПАОП 0.00-4.12-05. Цей порядок спрямовано на реалізацію в Україні системи безперервного навчання з питань охорони праці.

Вступний інструктаж з питань охорони праці проводять з усіма працівниками за програмою, що розроблена службою охорони праці з урахуванням особливостей виробництва і орієнтовного переліку питань вступного інструктажу. Його мета – ознайомлення з загальними правилами охорони праці, пожежної безпеки, виробничої санітарії, з правилами внутрішнього розпорядку, поведінки на території закладу, з питаннями профілактики виробничого травматизму, організацією роботи з охорони праці. Інструктаж проводить інженер з охорони праці у відповідно обладнаному

					КС КРБ 123.247.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Щирба Т.А.			Безпека життєдіяльності, основи охорони праці		
Перевірів		Шингера Н.Я.					
Консульт.		Гурик О.Я.					
Н. Контр.		Тиш Є.В.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркуші
						56	
					ТНТУ, каф. КС, гр. СІ-42		

кабінеті. Проведення інструктажу реєструється в спеціальному журналі.

Навчання з охорони праці передбачено як обов'язкове для керівників, головних спеціалістів, посадових осіб, відповідальних за безпечне виконання робіт, а також для осіб, залучених до виконання робіт підвищеної небезпеки. До таких можуть належати і фахівці з комп'ютерної інженерії, що працюють з високовольтним обладнанням, системами електроживлення серверних приміщень, проектують або супроводжують автоматизовані системи керування технологічними процесами [24].

Первинний інструктаж проводиться до початку роботи безпосередньо на робочому місці з працівниками:

- новоприйнятим (постійно чи тимчасово) на підприємство або до фізичної особи, яка використовує найману працю;
- які переводяться з одного структурного підрозділу підприємства до іншого; який виконуватиме нову для нього роботу;
- відрядженим працівником іншого підприємства, який бере безпосередню участь у виробничому процесі на підприємстві.
- проводиться з учнями, курсантами, слухачами та студентами навчальних закладів до початку трудового або професійного навчання;
- перед виконанням кожного навчального завдання, пов'язаного з використанням різних механізмів, інструментів, матеріалів.

Після проходження спеціального навчання перевірка знань проводиться комісією підприємства, що створюється наказом керівника. Якщо підприємство не має можливості утворити таку комісію (відповідно до п. 4.4 Типового положення), перевірку знань може здійснювати комісія спорідненого підприємства або територіального органу Держпраці.

Перед перевіркою знань з питань охорони праці на підприємстві для працівників організується навчання: лекції, семінари та консультації.

Повторний інструктаж проводиться в терміни, визначені нормативноправовими актами з охорони праці, які діють у галузі, або роботодавцем (фізичною особою, яка використовує найману працю) з

					КС КРБ 123.247.00.00 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

урахуванням конкретних умов праці, але не рідше:

- на роботах з підвищеною небезпекою - 1 раз на 3 місяці;
- 7 для решти робіт - 1 раз на 6 місяців.

Всі працівники, незалежно від посади, зобов'язані пройти навчання з питань охорони праці, включно з тими, хто відповідає за організацію та безпеку виконання робіт, згідно з типовим положенням про порядок проведення навчання та перевірки знань з питань охорони праці, від 26.01.2005р., №15 (зі змінами від 2017 року №140 – НПАОП 0.00-4.36-05). Законодавство не забороняє і не зобов'язує проходити навчання в акредитованих навчальних закладах, однак така практика широко застосовується в Україні.

Проходження навчання у спеціалізованих установах дозволяє підприємствам підвищити рівень підготовки персоналу, зменшити кількість нещасних випадків, та створити безпечні умови праці, що у підсумку забезпечує збереження життя та здоров'я працівників і скорочення витрат, пов'язаних із виробничим травматизмом.

4.2 Заходи пожежної безпеки для книгообмінної будки

В умовах воєнного стану, зумовленого збройною агресією проти України, питання пожежної безпеки стає надзвичайно актуальним. Через постійні обстріли цивільної та промислової інфраструктури значно зростає ймовірність виникнення пожеж, що вимагає високого рівня готовності до їх швидкого виявлення, локалізації та ліквідації.

У таких умовах особливо важливим є створення ефективної системи пожежогасіння, а також правильний вибір і застосування відповідних методів і засобів реагування. Пожежна безпека є невід'ємною складовою загальної системи промислової безпеки, тому її забезпечення передбачає комплекс заходів. Зокрема, необхідно оснастити території підприємств, будівлі, споруди, виробничі приміщення та технологічне обладнання первинними

					КС КРБ 123.247.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

засобами пожежогасіння для оперативного реагування на початкові етапи загоряння.

Вимога оснащення засобами пожежогасіння поширюється також на об'єкти, включно з книгообмінними будками, де можуть бути встановлені системи пожежогасіння, пожежної сигналізації або пожежні крани. Засоби пожежогасіння — це речовини або їх комплекси, придатні для використання людиною з метою локалізації та ліквідації пожежі на її початковій стадії (ДСТУ 2272:2006 «Пожежна безпека. Терміни та визначення основних понять»). Вони застосовуються на початку пожежі, щоб запобігти її поширенню та ліквідувати займання. Найефективнішим первинним засобом пожежогасіння для таких невеликих об'єктів, як книгообмінна будка, є вогнегасник.

Порядок розміщення вогнегасників у книгообмінній будці регламентується Правилами експлуатації вогнегасників та типовими нормами їх забезпечення, затвердженими наказом Міністерства внутрішніх справ України від 15 січня 2018 року № 25. Цей нормативний документ поширюється на будівлі і приміщення різного призначення, включно з такими малими об'єктами, як книгообмінні будки, підприємства, установи і організації незалежно від форми власності і виду діяльності, а також на механічні транспортні засоби [19].

Під час вибору засобів пожежогасіння потрібно врахувати фізико-хімічні та пожежонебезпечні властивості горючих речовин і матеріалів, їх взаємодію з вогнегасними речовинами, а також площу виробничих приміщень, відкритих майданчиків та установок. Відповідальними особами за своєчасне й повне оснащення об'єктів засобами пожежогасіння, їх технічне обслуговування, навчання працівників правилам користування вогнегасниками є власники цих об'єктів або орендарі згідно з договором оренди.

До початку експлуатації об'єкти — будинки, споруди, приміщення, технологічні установки — повинні бути забезпечені первинними засобами

					<i>КС КРБ 123.247.00.00 ПЗ</i>	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

пожежогасіння відповідно до наказу МВС України від 15.01.2018 № 25. Необхідну кількість таких засобів для кожного поверху та приміщення визначає відповідальна особа за пожежну безпеку на об'єкті.

Переносні вогнегасники, які застосовуються найчастіше, включають порошкові, вуглекислотні, аерозольні та водні вогнегасники залежно від типу пожежі та специфіки об'єкта. Вогнегасники необхідно:

- навішувати на вертикальні конструкції на висоті не більше ніж 1,5 м від рівня підлоги до нижнього торця вогнегасника та на відстані від дверей, достатній для їх повного відкривання;
- встановлювати у шафи пожежних кран-комплектів, спеціальні тумби, на підставки, що надійно закріплені на підлозі (за умови конструктивної можливості), або у пожежні щити (стенди).

З урахуванням специфіки книгообмінної будки необхідно використовувати вогнегасники, придатні для гасіння пожеж класів А (горючі тверді матеріали, зокрема папір і дерево) та В (горючі рідини). Найоптимальнішими є порошкові або водяні вогнегасники, для книгообмінної будки було обрано порошковий вогнегасник ВП - 6 який ефективно локалізує пожежі на початковій стадії (див. рис. 4.1).



Рисунок 4.1 – Порошковий вогнегасник ВП - 6

					КС КРБ 123.247.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

У середовищі, де експлуатується комп'ютерне обладнання та реалізуються інженерні рішення на базі апаратного та програмного забезпечення, пожежна безпека є критично важливою складовою загальної безпеки. У таких приміщеннях, відповідно до чинних нормативно-правових актів України, зокрема наказу МВС №1417 від 30.12.2014 року, передбачається обов'язкова наявність систем виявлення та оповіщення про пожежу, первинних засобів пожежогасіння.

У контексті комп'ютерної інженерії особливу увагу приділяють вибору засобів пожежогасіння, сумісних з високочутливим електронним обладнанням. Застосування технічних рішень має відповідати національним стандартам щодо пожежної безпеки, що регламентують вимоги до технічних засобів та експлуатаційних умов. Вибір засобів пожежогасіння здійснюється відповідно до вимог ДСТУ 4297:2004 «Пожежна безпека. Техніка пожежогасіння. Вогнегасники. Загальні технічні вимоги».

З огляду на технічну специфіку інженерних обчислювальних систем, важливо також забезпечити відповідність електромереж вимогам електробезпеки, організувати належну вентиляцію для запобігання перегріву компонентів, використовувати матеріали з підвищеною вогнестійкістю та передбачити можливість безперешкодної евакуації в разі надзвичайної ситуації.

Дотримання вимог чинного законодавства щодо вибору, розміщення та експлуатації вогнегасників є обов'язковим елементом загальної системи безпеки. Регулярне технічне обслуговування вогнегасників, проведення інструктажів для відповідальних осіб, а також доступність засобів пожежогасіння сприятимуть швидкому реагуванню у разі виникнення загоряння та мінімізації ризиків для майна і життя користувачів.

					КС КРБ 123.247.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було досягнуто наступне:

1. Проведено аналітичний огляд предметної області, виявлено недоліки існуючих рішень щодо організації доступу до фізичних книг та обґрунтовано актуальність автоматизованого підходу з використанням цифрових технологій.

2. Розроблено узагальнену структуру комп'ютеризованої системи, яка включає апаратний модуль керування доступом та вебплатформу для обліку книг і підтвердження угод.

3. Обґрунтовано вибір апаратних компонентів, зокрема мікроконтролера ESP32-CAM, виконавчих елементів (реле, соленоїд), сенсора Холу, елементів живлення та індикації з урахуванням вимог до автономності та надійності.

4. Реалізовано електричну принципову схему системи за допомогою середовища KiCad, що дозволило формалізувати апаратну частину для подальшого виготовлення та тестування.

5. Створено прошивку для ESP32 з реалізацією логіки сканування QR-коду, перевірки угоди через HTTP-запит до сервера та керування замком залежно від результату перевірки.

6. Розроблено вебсайт з використанням PHP, MySQL, Bootstrap та JavaScript, який забезпечує реєстрацію користувачів, додавання книг, обмін повідомленнями, створення та підтвердження угод, а також генерацію QR-кодів для доступу до будки.

7. Реалізовано механізм, що забезпечує автоматизоване фізичне відкриття комірки на основі підтвердженої угоди.

У межах проєкту було повністю реалізовано замкнутий цикл — від ідеї й архітектури до практичної реалізації функціональної системи для автоматизованого обміну книгами.

					КС КРБ 123.247.00.00 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жаровський Р.О., Луцик Н.С., Осухівська Г.М., Паламар А.М., Тиш Є.В. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2024. 39 с.
2. Буров Є.В., Митник М.М. Комп'ютерні мережі. Підручник. Том другий. Львів: «Магнолія 2006», 2024. 204 с.
3. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.
4. Kharchenko A., Bodnarchuk I., Yatsysyn V. The Method for Comparative Evaluation of Software Architecture with Accounting of Trade-offs. American Journal of Information Systems. 2014. Vol. 2, No. 1. P. 20-25.
5. Palamar A., Palamar M., Osukhivska H. Real-time Health Monitoring Computer System Based on Internet of Medical Things. CEUR Workshop Proceedings, 3rd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2023), Ternopil, Ukraine, Opole, Poland, November 22–24, 2023. Vol. 3628. P. 106-115.
6. Shabliiy N., Lupenko S., Lutsyk N., Yasniy O., Malyshevskaya O. Keystroke dynamics analysis using machine learning methods. Applied Computer Science. 2021. Vol. 17, No. 4. P. 75-83.
7. Voloskyi V., Leshchyshyn Y., Romanyshyn N., Palamar A., Tarasenko L. Method and algorithm for efficient cell balancing in the lithium-ion battery control system. CEUR Workshop Proceedings, The 1st International Workshop on Bioinformatics and Applied Information Technologies (BAIT 2024), Zboriv, Ukraine, October 02-04, 2024. Vol. 3842. P. 258-267.
8. Yasniy O., Lutsyk N., Demchyk V., Osukhivska H., Malyshevskaya O. The prediction of structural properties of Ni-Ti shape memory alloy by the supervised

machine learning methods. ITTAP 2023: 73–78. <https://ceur-ws.org/Vol-3628/short1.pdf>

9. Микитишин А. Г., Митник М. М., Стухляк П. Д. Телекомунікаційні системи та мережі. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017. 384 с.

10. Озарків Т., Жаровський Р. Метод оптимізації EIGRP протоколу для підвищення продуктивності передачі даних в комп'ютерних мережах. Матеріали XI науково-технічна конференція Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2023. С.167

11. Використання згенерованого QR-коду. URL: <https://apix-drive.com/ua/blog/useful/schto-take-qr-kod> (дата звернення 10.03.2025).

12. Валідація комп'ютеризованих систем. URL: <https://val-center.com/ua/service/computerized-system-validation> (дата звернення 10.03.2025).

13. ESP-32 CAM docs. URL: <https://docs.ai-thinker.com/en/esp32-cam> (дата звернення 10.03.2025).

14. Датчик Холу. URL: <https://surl.li/lxamtu> (дата звернення 10.03.2025).

15. Розробка електрично принципової схеми. URL: <https://studfile.net/preview/7291791/page:5/> (дата звернення 01.04.2025).

16. Налаштування Arduino IDE URL: https://geekmatic.in.ua/ua/arduino_ide_with_wifi_esp8266 (дата звернення 01.04.2025).

17. Клієнт-серверна архітектура сайту. <https://oppb.com.ua/news/garantiyi-prav-na-ohoronu-praci> (дата звернення 17.04.2025).

18. Структура бази даних. URL: <https://foxminded.ua/skhemy-bazy-danyh/> (дата звернення 11.05.2025).

19. Види первинних засобів пожежогасіння. URL: <https://surl.li/oeuslt> (дата звернення 26.05.2025).

					КС КРБ 123.247.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

20. Типове положення про порядок проведення навчання та перевірки знань з питань охорони праці, від 26.01.2005 р., №15 (зі змінами від 2017 року № 140 – НПАОП 0.00-4.36-05).

21. ДСТУ 2272:2006. Охорона праці. Терміни та визначення основних понять. — [На заміну ДСТУ 2272-93 ; чинний від 2007-01-01]. — Вид. офіц. — Київ : Держспоживстандарт України, 2006. — 16 с.

22. ДСТУ 4297:2004. Пожежна безпека. Вогнегасники. Загальні технічні вимоги, методи випробувань. — [Чинний від 2005-07-01]. — Вид. офіц. — Київ : Держспоживстандарт України, 2005. — 14 с.

23. Основи охорони праці: Підручник 2-ге видання, доповнене та перероблене. К.Н.Ткачук, М.О. Халімовський, В.В. Зацарний, С.В. Зеркалов Р.В. Сабарно, О.І. Полукаров, В.С. Козяков, Л.О. Мітюк. За ред. К.Н. Ткачука і М.О. Халімовського – К.: Основа, 2006. 448с

24. Гурик О.Я., Король О.І., Сенчишин В.С. Методичні вказівки до лабораторної роботи з дисципліни “Основи охорони праці”. Тернопіль, 2006.

					КС КРБ 123.247.00.00 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС

_____ Осухівська Г.М.

“ ____ ” _____ 2025 р

КОМП'ЮТЕРИЗОВАНА СИСТЕМА РОЗУМНОЇ КНИГООБМІННОЇ БУДКИ

ТЕХНІЧНЕ ЗАВДАННЯ

на 8 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІ-42

_____ к.т.н., доц. Шингера Н.Я.

_____ Щирба Т.А.

« ____ » _____ 2025 р.

« ____ » _____ 2025 р.

Тернопіль 2025

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютеризована система розумної книгообмінної будки».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.247.00.00

1.2 Виконавець

Студент групи СІ-42, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Щирба Т.А.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/7-53 від 27.01.2025 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 27.01.2025 р.

Плановий термін завершення виконання кваліфікаційної роботи – 23.06.2025 р.

1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи – наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

Розроблювана система призначена для:

- організації безконтактного доступу до книг.
- інтеграції фізичного пристрою з механізмом замикання та вебплатформи.
- забезпечення автоматизованого керування доступом до комірки

2.2 Мета створення системи

Метою кваліфікаційної роботи є створення повноцінної автоматизованої системи для безконтактного книгообміну, що поєднує автономний апаратний модуль на базі мікроконтролера ESP32 та вебплатформу для управління обміном книгами.

2.3 Характеристика об'єкту

Об'єктом є електронно-керована будка для обміну книгами, інтегрована з сайтом, що виконує облік користувачів, зберігає історію угод та генерує QR-

коди для доступу до конкретної комірки, відповідно до підтвердженої угоди між учасниками.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

3.1.1 Вимоги до структури та функціонування системи

Система повинна включати:

- автономний апаратний модуль з мікроконтролером, механізмом блокування та сенсорами.
- вебплатформу для користувацької взаємодії, обліку книг та угод.
- серверну частину для обробки запитів, генерації QR-кодів та перевірки дозволів.

3.1.2 Вимоги до способів та засобів зв'язку між компонентами системи

Обмін між апаратною та вебчастиною здійснюється через Wi-Fi-з'єднання у вигляді HTTP-запитів.

3.1.3 Вимоги до режимів функціонування системи

Передбачено:

- нормальний режим функціонування;
- аварійний режим функціонування.

Усі компоненти повинні продовжувати роботу в межах основної логіки або переходити в безпечний стан у разі збою.

3.1.4 Вимоги по діагностуванню системи

Система повинна підтримувати базову діагностику: моніторинг статусу датчика Холу, індикацію помилок світлодіодом, повідомлення на серійну

консоль ESP32. Для тестування вебчастини передбачено режим відображення журналу угод і логів доступу.

3.1.5 Перспективи розвитку, проектування системи

Проект допускає масштабування - додавання нових комірок, адміністрування декількох будок через єдиний вебінтерфейс, впровадження багаторівневого доступу, а також підключення камери для фіксації відкриття.

3.2 Показники призначення

Система повинна бути гнучкою до змін у структурі бази даних, розширення функціональності інтерфейсу, а також адаптивною до змін в апаратному складі.

3.2.1 Вимоги до надійності

Система повинна зберігати працездатність у разі:

- перебоїв живлення (наявність акумуляторного модуля).
- помилок доступу до серверу (збереження безпечного стану).
- локальних апаратних збоїв (індикація через LED, збереження журналу помилок).

3.3 Вимоги до безпеки

Живлення та керування замком повинні бути реалізовані з урахуванням електробезпеки. Комірка має бути замкнена в пасивному стані (тобто за замовчуванням закрита). У разі короткого замикання або втрати контролю – система не повинна допускати небезпечних станів.

3.3.1 Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Мікроклімат в приміщеннях повинен відповідати нормам виробничого мікроклімату по ДСН 3.3.6.042-99:

- температуру повітря в межах від +10°C до +35°C;
- відносну вологість повітря при 25°C в межах від 30% до 80%;
- атмосферний тиск 760 ± 25 мм рт. ст.

Періодичне технічне обслуговування використовуваних технічних засобів має проводитися відповідно до вимог технічної документації, але не рідше ніж один раз на рік.

Періодичне технічне обслуговування і тестування технічних засобів повинні включати обслуговування і тестування всіх використовуваних засобів, датчики, контролери, системи передачі даних, пристрої безперебійного живлення.

На підставі результатів тестування технічних засобів повинні проводитися аналіз причин виникнення виявлених дефектів і прийматися заходи по їх ліквідації.

3.4 Вимоги до захисту інформації від несанкціонованого доступу

Система повинна забезпечувати захист від несанкціонованого доступу на рівні не нижче визначеного вимогами, що висуваються до категорії 1Д відповідно до чинного нормативного документа “Автоматизовані системи. Захист від несанкціонованого доступу до інформації. Класифікація автоматизованих систем”.

Компоненти підсистеми захисту повинні забезпечувати:

- автентифікацію користувача під час входу до системи.
- перевірку прав доступу до ресурсів (угод, профілів, чату, комірок).
- заборону доступу до сторонніх облікових записів і адміністративних функцій.

- виключення повторного використання QR-кодів після підтвердженої дії.

Рівень захищеності вебплатформи, бази даних і фізичної взаємодії через мікроконтролер повинен відповідати вимогам щодо захисту інформації в умовах відкритого середовища доступу та обробки даних через мережу Інтернет.

3.4.1 Вимоги по збереженню інформації при аваріях

Дані угод, книг та користувачів мають зберігатися в постійній базі даних на сервері.

3.4.2 Вимоги по стандартизації і уніфікації

Використовувані компоненти повинні відповідати загальноприйнятим стандартам для безпечної роботи із напругою до 12 В. Програмна частина повинна бути уніфікованою за структурою, що дозволяє іншим розробникам швидко розширити або змінити функціонал.

3.4.3 Вимоги до функцій (завдань), що виконуються системою:

- облік користувачів, книг, угод.
- підтвердження угоди обома сторонами.
- генерація QR-коду для доступу до комірки.
- передача запиту на відкриття будки через інтернет.
- відкриття фізичного замка при підтвердженні.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:

- а) Структурна схема.
- б) Блок схема роботи комп'ютеризованої системи.
- в) Схема електрична принципова.
- г) Концептуальна схема бази даних.

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання.	27.01.2025 р. – 02.02.2025 р.
2	Робота над основними розділами – Аналіз технічного завдання. Аналіз предметної області та актуальності	03.02.2025 р. – 09.03.2025 р.
3	Робота над основними розділами. Вибір апаратного забезпечення та проектування електричної схеми.	10.03.2025 р. – 16.04.2025 р.
4	Робота над основними розділами. Реалізація та розробка вебплатформи: база даних, інтерфейс, логіка угод	17.04.2025 р. - 15.06.2025 р.
5	Безпека життєдіяльності, основи охорони праці	26.05.2025 р. – 02.06.2025 р.
6	Оформлення пояснювальної записки і графічного матеріалу	03.06.2025 р. – 10.06.2025 р.
7	Перевірка на академічний плагіат, перевірка керівником та консультантами	09.06.2025 р. - 15.06.2025 р.
8	Попередній захист кваліфікаційної роботи бакалавра	16.06.2025р. – 22.06.2025р.
9	Захист кваліфікаційної роботи бакалавра	27.06.2025 р.

6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

Додаток Б
Перелік елементів

Додаток В

Лістинги функцій комп'ютеризованої системи

```
// --- Налаштування портів GPIO ---

#define RELAY_PIN          12 // Пін для керування реле.
// Логічний 0 активує реле (вмикає замок)
#define HALL_SENSOR_PIN    14 // Пін, на який підключено
// цифровий датчик Холу для перевірки закриття дверцят
#define RED_LED_PIN        13 // Пін для червоного
// світлодіодного каналу (сигналізує про відмову)
#define GREEN_LED_PIN      15 // Пін для зеленого
// світлодіодного каналу (сигналізує про дозвіл доступу)

// --- Змінні для зберігання параметрів Wi-Fi ---
const char* ssid          = "Назва_Вашої_Мережі";          // Ім'я
// Wi-Fi мережі (SSID)
const char* password      = "Пароль_Від_WiFi";             // Пароль
// до Wi-Fi

void setup() {
    // Ініціалізація портів
    pinMode(RELAY_PIN, OUTPUT);          // Встановлюємо пін
// реле як вихід
    pinMode(RED_LED_PIN, OUTPUT);        // Встановлюємо пін
// червоного світлодіода як вихід
    pinMode(GREEN_LED_PIN, OUTPUT);      // Встановлюємо пін
// зеленого світлодіода як вихід
    pinMode(HALL_SENSOR_PIN, INPUT);     // Пін датчика Холу
// працює як вхід

    // Початкові значення
    digitalWrite(RELAY_PIN, HIGH);       // Вимикаємо реле
// (лог. 1 = замок вимкнено)
    digitalWrite(RED_LED_PIN, LOW);      // Світлодіоди
// вимкнені
    digitalWrite(GREEN_LED_PIN, LOW);

    // --- Ініціалізація серійного порту ---
    Serial.begin(115200);                 // Вмикаємо серійний
// монітор для діагностики
    delay(1000);                          // Невелика затримка
// перед підключенням Wi-Fi

    // --- Ініціалізація Wi-Fi ---
    WiFi.begin(ssid, password);           // Підключаємося до
// Wi-Fi
    Serial.print("Підключення до Wi-Fi");

    while (WiFi.status() != WL_CONNECTED) {
```

```

        delay(500); // Чекаємо з'єднання
        Serial.print(".");
    }

    // Виводимо локальну IP-адресу після підключення
    Serial.println("");
    Serial.println("Wi-Fi підключено");
    Serial.print("IP-адреса пристрою: ");
    Serial.println(WiFi.localIP());
}

#include <WiFi.h>
#include <HTTPClient.h>

// Функція перевірки доступу до комірки
bool checkDealAccess(String dealUrl) {
    HTTPClient http;
    http.begin(dealUrl); // URL, закодований у QR-коді
    int httpCode = http.GET();

    if (httpCode == 200) {
        String payload = http.getString();
        http.end();
        if (payload == "access_granted") {
            return true;
        }
    }

    http.end();
    return false;
}

// Функція відкриття замка
void unlockBox() {
    digitalWrite(RELAY_PIN, LOW); // Активуємо реле
    (логічний нуль)
    digitalWrite(GREEN_LED_PIN, HIGH); // Вмикаємо зелений
індикатор
    delay(3000); // Тримаємо замок
відкритим 3 секунди
    digitalWrite(RELAY_PIN, HIGH); // Вимикаємо реле
(замок зачиняється)
    digitalWrite(GREEN_LED_PIN, LOW); // Вимикаємо індикатор
}

// Функція відмови в доступі
void denyAccess() {
    digitalWrite(RELAY_PIN, HIGH);
    delay(2000);
    digitalWrite(RELAY_PIN, LOW);
}

// Імітація отриманого URL із QR-коду

```

```

String getScannedQR() {
    // У реальній системі – результат зчитування камери або
    стороннього декодувальника
    return "http://yourserver.com/unlock.php?deal_id=123"; //
приклад
}

void loop() {
    String scannedUrl = getScannedQR(); // ← тимчасовий
симулятор

    // Чекаємо поки дверцята закриті
    if (digitalRead(HALL_SENSOR_PIN) == LOW) {
        if (checkDealAccess(scannedUrl)) {
            unlockBox();
        } else {
            denyAccess();
        }
    } else {
        Serial.println("Дверцята відчинені. Очікування
закриття...");
        delay(2000); // Очікуємо перед повторною перевіркою
    }

    delay(5000); // Чекаємо між ітераціями, щоб уникнути
надмірного опитування
}
// Перевірка статусу угоди по URL з QR-коду
bool checkDealAccess(String dealUrl) {
    HTTPClient http;
    http.begin(dealUrl);
    int httpCode = http.GET();

    if (httpCode == 200) {
        String response = http.getString();
        http.end();
        return (response == "access_granted");
    } else {
        http.end();
        return false;
    }
}

void unlockBox() {
    digitalWrite(RELAY_PIN, LOW); // Замок активовано
(логічний 0)
    digitalWrite(GREEN_LED_PIN, HIGH); // Індикація –
зелений
    delay(3000); // Тримати
відчиненим 3 с
    digitalWrite(RELAY_PIN, HIGH); // Замок знову
замкнений
    digitalWrite(GREEN_LED_PIN, LOW); // Індикацію вимкнено
}

```

Додаток Г

Лістинги файлів веб-додатку

Лістинг файлу «about.php»

```
<?php
session_start();
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0"/>
    <title>Про нас - BookExchange</title>
    <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.3/dist/css/boot
strap.min.css" rel="stylesheet">
    <link href="https://cdn.jsdelivrivr.net/npm/bootstrap-
icons@1.10.5/font/bootstrap-icons.css" rel="stylesheet">
    <style>
        body {
            background-color: #fff;
            color: #212529;
            font-family: 'Segoe UI', sans-serif;
        }

        .navbar, footer {
            background-color: #f8f9fa;
            border-bottom: 1px solid #ddd;
        }

        .navbar-brand, .nav-link {
            color: #000 !important;
        }
```



```

        .navbar-nav .nav-link:hover {
            color: #555 !important;
        }

        .content {
            padding: 30px 0;
        }

        footer {
            padding: 20px 0;
            color: #666;
            font-size: 14px;
        }

        .footer-menu a {
            color: #666;
            margin: 0 10px;
            text-decoration: none;
        }

        .footer-menu a:hover {
            color: #000;
        }
    </style>
</head>
<body>

<!-- Навігація -->
<nav class="navbar navbar-expand-lg shadow-sm">
    <div class="container">
        <a class="navbar-brand" href="index.php"><i class="bi
bi-book-half me-1"></i>BookExchange</a>

        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
            <span class="navbar-toggler-icon"></span>
        </button>

```

```

        <div class="collapse navbar-collapse justify-content-
end" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item"><a class="nav-link"
href="index.php"><i class="bi bi-house"></i> Головна</a></li>
                <?php if (isset($_SESSION['first_name'])): ?>
                    <li class="nav-item">
                        <a class="nav-link" href="profile.php">
                            <i class="bi bi-person-circle"></i> Вітаємо, <?=
htmlspecialchars($_SESSION['first_name']) ?>!
                        </a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="logout.php">
                            <i class="bi bi-box-arrow-right"></i> Вихід
                        </a>
                    </li>
                <?php else: ?>
                    <li class="nav-item"><a class="nav-link"
href="login.php"><i class="bi bi-box-arrow-in-right"></i>
Вхід</a></li>
                    <li class="nav-item"><a class="nav-link"
href="register.php"><i class="bi bi-pencil-square"></i>
Реєстрація</a></li>
                <?php endif; ?>
            </ul>
        </div>
    </div>
</nav>

<!-- Про нас -->
<div class="container content">
    <h2 class="text-center mb-4">Про BookExchange</h2>
    <p>BookExchange – це онлайн-платформа, яка дозволяє
користувачам обмінюватися книгами або продавати їх. Ми прагнемо
зробити доступ до книг зручним і доступним для кожного, хто
бажає поділитися своїми улюбленими творами з іншими.</p>

```

<h4>Наша місія</h4>

<p>Ми віримо в силу читання і хочемо допомогти людям знаходити нові книги без зайвих витрат. Завдяки нашій платформі кожен може знайти цікаві книги для себе, обмінятися ними або придбати за вигідною ціною.</p>

<h4>Наші цінності</h4>

Доступність

Екологічність (повторне використання книг)

Спільнота

Освітній розвиток

<h4>Як це працює?</h4>

<p>Користувачі можуть реєструватися на нашій платформі, додавати свої книги та обирати ті, які вони хочуть отримати в обмін. Це простий і безпечний спосіб обміну книжками серед людей з усього світу.</p>

</div>

<!-- Футер -->

<footer class="text-center mt-5 border-top">

<div class="container">

<p>© 2025 BookExchange. Всі права захищено.</p>

<div class="footer-menu">

<i class="bi bi-house-door"></i>
Головна

<i class="bi bi-envelope"></i>
Контакти

</div>

</div>

</footer>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootst
rap.bundle.min.js"></script>

</body>

```
</html>
```

Лістинг файлу «accept_deal.php»

```
<?php
session_start();
require 'db.php';

if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit;
}

$deal_id = $_POST['deal_id'];

$stmt = $pdo->prepare("UPDATE deals SET status = 'accepted'
WHERE id = ?");

$stmt->execute([$deal_id]);

// Можна зробити редирект назад до чату
header("Location: " . $_SERVER['HTTP_REFERER']);
exit;
```

Лістинг файлу «add_book.php»

```
<?php
session_start();
require 'db.php';

if (!isset($_SESSION['user_id'])) {
    header('Location: login.php');
    exit();
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
```

```

$title = $_POST['title'] ?? '';
$author = $_POST['author'] ?? '';
$description = $_POST['description'] ?? '';
$genre = $_POST['genre'] ?? '';
$status = $_POST['status'] ?? 'available';

// Якщо обмін — ціна 0.00, якщо продаж — беремо з форми
або ставимо 0.00, якщо порожнє
if ($status === 'exchange') {
    $price = 0.00;
} else {
    $price = isset($_POST['price']) &&
is_numeric($_POST['price']) ? floatval($_POST['price']) : 0.00;
}

// Фото
$image = null;
if (isset($_FILES['image']) && $_FILES['image']['error']
=== UPLOAD_ERR_OK) {
    $imageName = basename($_FILES['image']['name']);
    $imagePath = 'uploads/' . $imageName;
    if (move_uploaded_file($_FILES['image']['tmp_name'],
$imagePath)) {
        $image = $imagePath;
    }
}

// Запит до бази
$query = "INSERT INTO books (title, author, description,
price, status, user_id, genre, image, created_at)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, NOW())";
$stmt = $pdo->prepare($query);
$stmt->execute([
    $title, $author, $description, $price, $status,
    $_SESSION['user_id'], $genre, $image
]);

```

```

        echo "<p style='color: green;*> Книга успішно
додана!</p>";
    }
    ?>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Додати книгу</title>
    <!-- Підключення Bootstrap для стилів -->
    <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.3/dist/css/boot
strap.min.css" rel="stylesheet">
    <!-- Підключення Font Awesome для іконок -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0-beta3/css/all.min.css" rel="stylesheet">
    <style>
        .form-group {
            margin-bottom: 1.5rem;
        }
        .form-control {
            border-radius: 0.375rem;
            padding: 0.5rem 0.75rem;
        }
        label {
            font-weight: bold;
        }
        .btn-primary {
            background-color: #007bff;
            border: none;
            padding: 0.5rem 1.5rem;
            font-size: 1rem;
        }
        .btn-primary:hover {

```

```

        background-color: #0056b3;
    }
    .container {
        max-width: 600px;
        margin-top: 30px;
    }
</style>
</head>
<body class="bg-light">

<div class="container">
    <div class="card shadow-sm">
        <div class="card-body">
            <h1 class="text-primary mb-4"><i class="fas fa-
book"></i> Додати нову книгу</h1>

            <form action="add_book.php" method="POST"
enctype="multipart/form-data">
                <div class="form-group">
                    <label for="title">Назва книги:</label>
                    <input type="text" id="title" name="title"
class="form-control" required>
                </div>

                <div class="form-group">
                    <label for="author">Автор:</label>
                    <input type="text" id="author"
name="author" class="form-control" required>
                </div>

                <div class="form-group">
                    <label for="status">Тип
пропозиції:</label>
                    <select id="status" name="status"
class="form-control" required onchange="togglePrice()">
                        <option
value="available">Продаж</option>

```

```

                                <option
value="exchange">Обмін</option>
                                </select>
                            </div>

                            <div id="priceField" class="form-group">
                                <label for="price">Ціна (грн):</label>
                                <input type="number" id="price"
name="price" class="form-control">
                            </div>

                            <div class="form-group">
                                <label for="description">Опис:</label>
                                <textarea id="description"
name="description" class="form-control" rows="4"
required></textarea>
                            </div>

                            <div class="form-group">
                                <label for="genre">Жанр:</label>
                                <input type="text" id="genre" name="genre"
class="form-control" required>
                            </div>

                            <div class="form-group">
                                <label for="image">Фото книги:</label>
                                <input type="file" id="image" name="image"
accept="image/*" class="form-control" required>
                            </div>

                                <button type="submit" class="btn btn-primary
mt-3"><i class="fas fa-plus-circle"></i> Додати книгу</button>
                            </form>
                        </div>
                    </div>
                </div>

```



```
<script>
    function togglePrice() {
        const status =
document.getElementById('status').value;
        const priceField =
document.getElementById('priceField');
        const priceInput = document.getElementById('price');

        if (status === 'exchange') {
            priceField.style.display = 'none';
            priceInput.removeAttribute('required');
            priceInput.value = '';
        } else {
            priceField.style.display = 'block';
            priceInput.setAttribute('required', 'required');
        }
    }
    window.onload = togglePrice;
</script>

</body>
</html>
```

Додаток Е

Лістинг функції підключення до бази даних

```
<?php
$host = 'localhost'; // Хост бази даних
$dbname = 'book_exchange'; // Назва бази даних
$username = 'root'; // Ваше ім'я користувача MySQL (наприклад,
root)
$password = 'root'; // Ваш пароль MySQL

try {
    // Підключення до бази даних через PDO
    $pdo = new
PDO("mysql:host=$host;port=3307;dbname=$dbname", $username,
$password);
    // Налаштування на викидання винятків у разі помилки
    $pdo->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    // Якщо виникла помилка під час підключення
    die("Не вдалося підключитися до бази даних: " . $e-
>getMessage());
}
?>
```