

БАКАЛАВРСЬКА РОБОТА

БР. ІІ - 55.00.00.000 ІІЗ

Група ІІ-21-3

Храбатин Юрій

2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Храбатин Юрій Романович

(прізвище, ім'я, по батькові)

УДК 004
(індекс)

БАКАЛАВРСЬКА РОБОТА

Розробка Web-базованої системи для житлово-експлуатаційних

компаній

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач освітнього рівня Храбатин Ю.Р.
(підпис, ініціали та прізвище здобувача)

Науковий керівник Корнута Володимир Андрійович, к.т.н., доцент
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

доц. Бандура В.В.
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2025

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 28 квітня 2025 р.

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області розробки системи для житлово-експлуатаційних компаній	04.05.2025	виконано
2	Моделювання архітектури системи для житлово-експлуатаційних компаній	15.05.2025	виконано
3	Моделювання динамічних та статичних аспектів системи за допомогою UML діаграм	21.05.2025	виконано
4	Моделювання структури бази даних	28.05.2025	виконано
5	Програмна імплементація web-базованої системи для житлово-експлуатаційних компаній	03.06.2025	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	10.06.2025	виконано

Студент – дипломник _____

(підпис)

Керівник роботи _____

(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 77 сторінок, 59 рисунків, список використаних джерел із 30 найменуваннями, 1 додаток.

Мета роботи - реалізувати web-базовану інформаційну систему для житлово-експлуатаційних компаній, яка забезпечує ефективне управління будинками, персоналом, подіями, платежами, повідомленнями та іншими об'єктами.

Об'єкт дослідження - інформаційні процеси управління житлово-експлуатаційною компанією.

Предмет дослідження - методи, моделі та технології розробки web-базованих інформаційних систем управління житлово-комунальними об'єктами.

В першому розділі проведено всебічний аналіз предметної області, сучасних програмних рішень та функціональних вимог до систем управління житлово-експлуатаційними компаніями.

В другому розділі запропоновано архітектуру, здійснено вибір технологічного стеку та побудовано UML-моделі системи, що відображають її логіку та структуру.

В третьому розділі реалізовано веб-систему з повним циклом розробки – від проектування до тестування – з інтеграцією модулів управління користувачами, персоналом, подіями, повідомленнями та базою даних.

Висновок: запропоновано архітектурну модель web-базованої системи управління житлово-експлуатаційною компанією, яка враховує специфіку внутрішніх бізнес-процесів та інтегрує модулі управління персоналом, платежами, повідомленнями та подіями в єдине середовище

КЛЮЧОВІ СЛОВА: WEB-СИСТЕМА, ЖИТЛОВО-ЕКСПЛУАТАЦІЙНА КОМПАНІЯ, ANGULAR, NODE.JS, БАЗА ДАНИХ, UML, АВТОМАТИЗАЦІЯ, СИСТЕМА КЕРУВАННЯ, МОДУЛЬНІСТЬ

ANNOTATION

The bachelor's thesis contains 77 pages, 59 figures, a list of used sources with 30 names, 1 appendix.

The purpose of the work is to implement a web-based information system for housing and utility companies, which provides effective management of buildings, personnel, events, payments, messages and other objects.

The object of research is information processes of housing and utility company management.

The subject of research is methods, models and technologies for developing web-based information systems for managing housing and utility facilities.

The first section provides a comprehensive analysis of the subject area, modern software solutions and functional requirements for the housing and utility company management system.

The second section proposes to create an architecture, select a technological stack and create UML models of the system that reflect its logic and structure.

In the third section, a web system with a full development cycle is implemented - from design to testing - with the integration of user, personnel, event, message and database management modules.

Conclusion: we offer an architectural model of a web-based housing management system that takes into account the specifics of internal business processes and integrates personnel, payment, message and event management modules into a single environment.

KEYWORDS: WEB SYSTEM, HOUSING MANAGEMENT COMPANY, ANGULAR, NODE.JS, DATABASE, UML, AUTOMATION, CONTROL SYSTEM, MODULARITY

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	10
ВСТУП.....	11
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ ЖИТЛОВО-ЕКСПЛУАТАЦІЙНИХ КОМПАНІЙ	14
1.1. Огляд функціональних модулів web-базованої системи для житлово-експлуатаційних компаній	14
1.1.1. Архітектура та технологічна платформа	14
1.1.2. Функціональні модулі системи	15
1.2. Обґрунтування та архітектура розробки системи для житлово-експлуатаційних компаній	16
1.2.1. Значимість проекту.....	17
1.2.2. Мета проекту.....	18
1.2.3. Мотивація проекту.....	18
1.3. Аналіз сучасних систем управління житлово-експлуатаційними компаніями і товариствами	19
1.3.1. Системи на базі веб-технологій	20
1.3.2. Мобільні додатки для управління житловою компанією / товариством	25
1.3.3. Новітні тенденції та додаткові функціональності	27
РОЗДІЛ 2. МОДЕЛЮВАННЯ АРХІТЕКТУРИ СИСТЕМИ ДЛЯ ЖИТЛОВО-ЕКСПЛУАТАЦІЙНИХ КОМПАНІЙ	29
2.1. Критичний аналіз існуючих методологій управління житловими компаніями та системні вимоги для розробки	29

					БР.ІІ – 55.00.00.000 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Розробка Web-базованої системи для житлово-експлуатаційних компаній Пояснювальна записка	<i>Літ.</i>	<i>Арк.</i>	<i>Аркуші</i>
Розроб.		Храбатин Ю.Р					6	
Перевір.		Корнута В.А.						
Реценз.								
Н. Контр.		Піх М.М.						
Затверд.		Бандура В.В.					ІФНТУНГ Ш-21-3	

2.1.1. Недоліки існуючих підходів до комунікації	29
2.1.2. Системні вимоги	30
2.2. Технологічний стек веб-базованої системи управління житлово- експлуатаційною компанією	31
2.2.1. Angular JS (Фронтенд)	31
2.2.2. Bootstrap (Фронтенд-фреймворк для дизайну).....	33
2.2.3. Node.js (Бекенд)	33
2.2.4. MySQL.....	34
2.2.5. Інструмент phpMyAdmin	34
2.2.6. Середовище XAMPP.....	35
2.3. Візуалізація системного дизайну за допомогою Unified Modeling Language	36
2.3.1. Архітектурна діаграма.....	36
2.3.2. Представлення Use Case.....	37
2.4. Моделювання динамічних та статичних аспектів системи за допомогою UML діаграм	38
2.4.1. Представлення діаграми послідовності (Sequence Diagram).....	38
2.4.2. Представлення діаграми класів (Class Diagram)	39
2.4.3. Розробка діаграми активності (Activity Diagram)	40
2.5. Моделювання структури бази даних за допомогою діаграми сутність- зв'язок (ER-модель).....	41

РОЗДІЛ 3. ПРОГРАМНА ІМПЛЕМЕНТАЦІЯ WEB-БАЗОВАНОЇ СИСТЕМИ ДЛЯ ЖИТЛОВО-ЕКСПЛУАТАЦІЙНИХ КОМПАНІЙ

3.1. Системний аналіз запропонованої веб-базованої системи	44
3.1.1. Запропонована архітектура системи.....	44
Модуль безпеки	45
3.1.2. Деталізація функціональності веб-додатка	46
3.2. Початкова імплементація системи управління	47

3.2.1. Налаштування середовища розробки	47
3.2.2. Створення проекту Angular	47
3.2.3. Розробка фронтенду	48
3.2.4. Розробка бекенду (RESTful API).....	48
3.2.5. Інтеграція фронтенду та бекенду	48
3.2.6. Налаштування бази даних	49
3.2.7. Тестування проекту	49
3.3. Представлення структури бази даних.....	49
3.3.1. Створення бази даних та таблиць	50
3.3.3. Приклади структури таблиць та даних	51
3.4. Методологія тестування розроблюваної веб-базованої системи	51
3.5. Тестування додатка: модуль адміністратора	52
3.5.1. Доступ та вхід до системи	52
3.5.2. Управління користувачами	54
3.5.3. Управління будинками	55
3.5.4. Управління персоналом безпеки.....	56
3.5.5. Управління персоналом з прибирання	57
3.5.6. Управління повідомленнями.....	58
3.5.7. Управління подіями	58
3.5.8. Управління платежами	59
3.6. Функціональний аналіз та тестування модуля системи управління	61
3.6.1. Доступ та автентифікація користувача.....	61
3.6.2. Функціональні можливості користувача	62
3.7. Аналіз функціональності модуля безпеки в системі.....	65
3.7.1. Автентифікація та доступ до модуля безпеки	66
3.7.2. Управління відвідувачами.....	66
3.7.3. Моніторинг персоналу з прибирання	68
3.7.4. Доступ до повідомлень та подій	68

3.8. Оцінка та перспективи розвитку веб-системи управління житлово-експлуатаційної компанії	69
3.8.1. Ефективність та переваги впровадження системи	69
3.8.2. Можливі майбутні вдосконалення	70
 ВИСНОВКИ	 72
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	74
ДОДАТКИ	
БІБЛОГРАФІЧНА ДОВІДКА	

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CLI – Command Line Interface – Інтерфейс командного рядка

API – Application Programming Interface – Інтерфейс прикладного програмування

RESTful – Representational State Transfer (architecture style) – Передача репрезентативного стану (архітектурний стиль)

CRUD – Create, Read, Update, Delete – Створення, Читання, Оновлення, Видалення (операції з даними)

XAMPP – Cross-platform (X), Apache (A), MySQL (M), PHP (P), Perl (P) – Крос-платформний (X), Apache (A), MySQL (M), PHP (P), Perl (P) (комплект програмного забезпечення)

UI – User Interface – Користувацький інтерфейс

UX – User Experience – Користувацький досвід

2FA – Two-Factor Authentication – Двофакторна автентифікація

CSS – Cascading Style Sheets – Каскадні таблиці стилів

HTML – HyperText Markup Language – Мова розмітки гіпертексту

RDBMS – Relational Database Management System – Реляційна система управління базами даних

OMG – Object Management Group – Група управління об'єктами

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

ВСТУП

Сучасний розвиток інформаційних технологій істотно трансформуює сфери управління, комунікації та надання послуг. Однією з галузей, що відчуває потребу в цифровій трансформації, є діяльність житлово-експлуатаційних компаній. Управління житловим фондом, облік платежів, взаємодія з мешканцями, контроль за технічним станом об'єктів та оперативне реагування на інциденти – це лише частина завдань, які покладаються на такі компанії. У традиційних умовах вирішення цих задач часто супроводжується значними часовими, ресурсними та комунікаційними витратами.

Web-базовані інформаційні системи мають значний потенціал для автоматизації управлінських процесів у житлово-комунальному господарстві, забезпечення прозорості операцій, зменшення адміністративного навантаження та підвищення якості обслуговування мешканців. У роботі виконано аналіз предметної області, розроблено архітектуру та реалізовано прототип системи управління для житлово-експлуатаційних компаній з використанням сучасних веб-технологій (Angular, Node.js, MySQL). Окрема увага приділена питанням безпеки, зручності користувацького інтерфейсу, модульності та масштабованості системи.

Актуальність роботи

Актуальність дослідження зумовлена нагальною потребою житлово-експлуатаційних компаній у впровадженні цифрових інструментів для підвищення ефективності управління житловим фондом, прозорості фінансових операцій та поліпшення комунікацій з мешканцями. В Україні та багатьох країнах Східної Європи значна частина житлового господарства функціонує на базі застарілих або ручних методів обліку й управління. Це призводить до затримок у вирішенні проблем, відсутності чіткої звітності, недовіри з боку мешканців. Розробка інтегрованої веб-системи дозволяє

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

централізувати основні функції управління, забезпечити доступність інформації в режимі реального часу, підтримати електронний документообіг та інтеграцію з іншими сервісами.

Крім того, поширення смартфонів і доступу до Інтернету створює сприятливі умови для впровадження веб-орієнтованих рішень як з боку адміністрації, так і користувачів. Отже, створення такої системи є логічним кроком на шляху до цифрової трансформації ЖКГ-сектора.

Мета роботи - реалізувати web-базовану інформаційну систему для житлово-експлуатаційних компаній, яка забезпечує ефективне управління будинками, персоналом, подіями, платежами, повідомленнями та іншими об'єктами.

Завдання дослідження

1. Провести аналіз предметної області діяльності житлово-експлуатаційних компаній.
2. Дослідити існуючі програмні рішення та визначити їх переваги та недоліки.
3. Обґрунтувати архітектуру та технології для реалізації системи.
4. Побудувати UML-моделі для представлення структури та логіки роботи системи.
5. Розробити базу даних та реалізувати її структуру.
6. Провести тестування функціональних модулів системи.

Об'єкт дослідження - інформаційні процеси управління житлово-експлуатаційною компанією.

Предмет дослідження - методи, моделі та технології розробки web-базованих інформаційних систем управління житлово-комунальними об'єктами.

Методи дослідження:

- Аналіз – для вивчення предметної області та функціональних потреб користувачів.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

- Моделювання – для побудови UML-діаграм, бази даних і визначення взаємодій.

- Метод прототипування – для розробки MVP-версії веб-системи.

- Методи тестування ПЗ – для перевірки функціональності та виявлення помилок.

- Формальний підхід до архітектурного проєктування – для структурування та модульного поділу системи.

Наукова новизна

У роботі запропоновано архітектурну модель web-базованої системи управління житлово-експлуатаційною компанією, яка враховує специфіку внутрішніх бізнес-процесів та інтегрує модулі управління персоналом, платежами, повідомленнями та подіями в єдине середовище з урахуванням вимог до масштабованості та безпеки.

Практичне значення

Результати дослідження можуть бути використані для впровадження у діяльність житлово-експлуатаційних компаній, ОСББ, керуючих компаній та муніципальних структур, що займаються управлінням житловим фондом. Запропонована система може стати основою для цифрової трансформації сервісів ЖКГ.

Бакалаврська робота містить 77 сторінок, 59 рисунків, 3 розділи список використаних джерел із 30 найменуваннями, 1 додаток.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ ЖИТЛОВО-ЕКСПЛУАТАЦІЙНИХ КОМПАНІЙ

1.1. Огляд функціональних модулів web-базованої системи для житлово- експлуатаційних компаній

У контексті забезпечення ефективного функціонування житлово-експлуатаційних компаній, розробка спеціалізованих веб-базованих систем управління набуває критичного значення. Представлена система є глобальним веб-додатком, спроектованим для оптимізації управління житловими товариствами.

1.1.1. Архітектура та технологічна платформа

Розробка представленої веб-базованої системи здійснювалася на основі сучасних, перевірених часом та галуззю веб-технологій, що забезпечує її високу продуктивність, масштабованість та надійність. Вибір технологічного стеку був зумовлений необхідністю створити ефективний інструмент для управління житловим фондом, здатний задовольняти як поточні, так і майбутні потреби користувачів.

У якості фреймворку для фронтенду було обрано AngularJS. Цей вибір обґрунтовується його здатністю забезпечувати розробку динамічного та високоінтерактивного користувацького інтерфейсу (UI), що є критично важливим для забезпечення інтуїтивно зрозумілої та зручної взаємодії користувачів із системою. AngularJS дозволяє ефективно реалізовувати односторінкові додатки (SPA), що мінімізує перезавантаження сторінок та покращує загальний користувацький досвід.

Для реалізації функціоналу бекенду застосовано Node.js. Ця платформа дозволяє створювати масштабовані та високопродуктивні серверні додатки,

					БР.ІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

які здатні обробляти велику кількість одночасних запитів, що є важливою характеристикою для багатокористувацьких систем. Завдяки асинхронній, подійно-орієнтованій архітектурі Node.js, система демонструє високу швидкість обробки даних та ефективне використання ресурсів.

Система оперує з даними, що зберігаються у реляційній базі даних MySQL. MySQL є одним з найпопулярніших рішень для зберігання структурованих даних, відомим своєю надійністю, швидкістю та широкою підтримкою транзакцій, що гарантує цілісність та консистентність даних.

Серверне середовище розгорнуто на платформі XAMPP. XAMPP є комплексним дистрибутивом Apache, що інтегрує в собі Apache HTTP Server (як веб-сервер), MySQL (як систему управління базами даних), а також інтерпретатори PHP та Perl. Це інтегроване середовище значно спрощує процес розгортання та адміністрування системи, надаючи повний набір необхідних компонентів для її функціонування.

Загалом, така архітектура забезпечує надійну, ефективну та масштабовану платформу для комплексного управління житловим фондом, дозволяючи обробляти значні обсяги інформації та підтримувати високий рівень взаємодії між усіма учасниками житлового товариства.

1.1.2. Функціональні модулі системи

Розроблена система управління складається з трьох ключових функціональних модулів, кожен з яких реалізує специфічні можливості, необхідні для комплексного управління житловим товариством:

1. Модуль адміністратора.

Цей модуль надає розширений функціонал для повного контролю над системою. До його можливостей входить управління користувачами, облік житлових одиниць (будинків/квартир), налаштування параметрів безпеки, керування графіками та виконанням робіт персоналом з прибирання, обробка

									Арк.
									15
Змн.	Арк.	№ докум.	Підпис	Дата	БР.ІІІ – 55.00.00.000 ПЗ				

внутрішніх повідомлень, планування та координація подій, ведення обліку платежів та адміністрування інформації щодо голосувань. Модуль дозволяє додавати та переглядати всі необхідні дані відповідно до визначених функцій.

2. Модуль Користувача.

Призначений для мешканців та надає доступ до релевантної інформації та можливостей взаємодії з системою. Користувачі можуть переглядати інформацію про заплановані події, отримувати внутрішні повідомлення, контролювати статус своїх платежів та брати участь у голосуваннях. Функціонал модуля дозволяє переглядати дані, надані адміністратором, та виконувати дії відповідно до них.

3. Модуль Безпеки.

Цей модуль зосереджений на управлінні аспектами безпеки житлового товариства. Він включає функції обліку відвідувачів, контролю за персоналом з прибирання, моніторингу повідомлень, що стосуються безпеки, та інформації про події, які можуть впливати на безпеку. Модуль надає релевантну інформацію відповідно до конкретної функції.

Кожен з цих модулів розроблений з метою забезпечення зручного та ефективного інтерфейсу для управління всіма деталями щоденних операцій, включаючи комунікацію, організацію зустрічей, планування подій, обробку платежів та координацію робіт.

1.2. Обґрунтування та архітектура розробки системи для житлово-експлуатаційних компаній

В умовах зростаючої складності управління житловим фондом та необхідності оптимізації комунікаційних процесів, розробка автоматизованих систем набуває особливої актуальності. Ця наукова стаття представляє опис розробленої веб-базованої системи, що має на меті

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

централізацію та спрощення управління інформацією в житлових товариствах.

Ручне ведення та зберігання даних у житлово-експлуатаційних компаніях характеризується значними труднощами, що включають високу ймовірність помилок, складність доступу до актуальної інформації та низьку ефективність комунікаційних процесів. З огляду на повсюдне використання інформаційних технологій та зростаючу залежність суспільства від онлайн-сервісів, розробка веб-базованого рішення є логічним кроком для вирішення зазначених проблем. Представлений додаток призначений для інтеграції всієї релевантної інформації щодо житлової системи в єдиний централізований репозиторій, забезпечуючи легкий та безпечний доступ для всіх членів товариства. Застосування сучасних веб-технологій дозволяє не лише спростити повсякденні операції, а й підвищити рівень безпеки за рахунок ефективного обліку відвідувачів, персоналу та інших суб'єктів. Централізоване зберігання даних у базі даних (MySQL) сприяє мінімізації ризиків шахрайства та підвищенню прозорості операцій.

1.2.1. Значимість проекту

Розроблена система управління житловим товариством має значну значимість для оптимізації його функціонування. Вона забезпечує підвищену ефективність та простоту використання, що є критично важливим для комплексного управління житловим фондом. Додаток надає низку сервісів, таких як система запитів, онлайн-моніторинг та інше, що сприяє оперативному вирішенню питань та мінімізації конфліктних ситуацій у житловому товаристві. Ця система є особливо корисною для секторів зі складною житловою інфраструктурою та у сфері нерухомості, оскільки вона надає сучасні послуги та аналітичні дані. Розробка враховувала потреби як бізнес-користувачів, так і рядових мешканців, забезпечуючи їм унікальний досвід проживання та покращений доступ до операцій, що виконуються

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

адміністрацією. Дизайн додатка включає такі функціональності, як чат-бот, дошка оголошень, система оплати в розстрочку, що підвищує зручність та функціональність системи.

1.2.2. Мета проекту

Основною метою цього проекту є створення ефективної та інтуїтивно зрозумілої системи управління житловим товариством, яка може бути використана як для особистих потреб мешканців, так і для комерційних цілей. Проект спрямований на надання підтримки різним категоріям користувачів у рамках житлового товариства: мешканцям, службі безпеки та адміністрації. Додаток реалізований як система введення/виведення даних, що забезпечує гнучку взаємодію з інформацією. Доступ до системи здійснюється через авторизацію за допомогою зареєстрованих облікових даних (ім'я користувача та пароль), що забезпечує конфіденційність та безпеку даних для користувачів, служби безпеки та адміністратора. Технічна реалізація системи базується на використанні AngularJS для клієнтської частини та Node.js для серверної частини, що гарантує високу продуктивність та масштабованість.

1.2.3. Мотивація проекту

Мотивацією для розробки даної системи слугує наявна неефективність традиційних методів управління житловим товариством, які широко застосовуються в міських регіонах. Ручні процеси комунікації, обліку та координації характеризуються значними обмеженнями та недоліками. Щоденні оголошення, регулярні та позапланові збори, контакти для вирішення побутових питань, попередження безпеки та інші важливі повідомлення часто не доставляються вчасно або не доходять до всіх адресатів через людський фактор та трудомісткість ручної обробки. Це

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

призводить до зниження рівня інформованості мешканців, виникнення конфліктів та загальної неефективності управління.

Для вирішення цих проблем виникла потреба у розробці автоматизованого рішення, що дозволяє мінімізувати ручні зусилля та підвищити ефективність комунікації. Основна мета системи управління товариством полягає в інтеграції різноманітних функціональних можливостей в єдиний додаток, доступний через смартфон або веб-інтерфейс. Це включає управління повідомленнями, оновленнями, пов'язаними з добробутом, та соціальними подіями. Система також сприяє ефективному управлінню подіями, надаючи вичерпну інформацію про них через інтуїтивно зрозумілий інтерфейс. Крім того, додаток призначений для оповіщення користувачів про критичні питання та скарги, що стосуються поточних проблем, що значно покращує оперативність реагування на виклики.

1.3. Аналіз сучасних систем управління житлово-експлуатаційними компаніями і товариствами

Управління житловими компаніями і товариствами в сучасних міських умовах є складним та багатоаспектним завданням, що охоплює широкий спектр повсякденних потреб мешканців, включаючи комунальні послуги, безпеку, комунікацію та організацію соціальної взаємодії. Традиційні методи управління, що ґрунтуються на ручних процесах та паперовому документообігу, демонструють значні недоліки, такі як низька ефективність, відсутність прозорості, висока ймовірність помилок та труднощі в оперативній комунікації. У відповідь на ці виклики, наукове співтовариство та розробники активно працюють над створенням автоматизованих систем, які використовують сучасні інформаційні технології. Нижче представлено огляд ключових досліджень та розробок у цій галузі.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

1.3.1. Системи на базі веб-технологій

В роботі [1] розроблено систему управління житловим товариством за допомогою РНР яка призначена для оптимізації повсякденної діяльності кооперативних товариств. Основним акцентом цього дослідження було створення онлайн-інструментів для ефективного вирішення скарг у охоронюваних та закритих товариствах. Автори зазначають, що існуючі ручні методи подання скарг виявилися трудомісткими та неефективними. Запропонована веб-базована система, реалізована з використанням РНР, забезпечує більш ефективний та зручний для користувача підхід до обробки скарг, дозволяючи відстежувати їхній прогрес та покращуючи комунікацію.

Метою системи є отримання точної інформації про транзакції членів житлової компанії, скорочення робочого часу кожної особи та уникнення надмірності даних. За допомогою цієї системи члени можуть легко відстежувати свої платежі за кожною здійсненою транзакцією. До настання терміну вони також можуть контролювати залишки, яких вони повинні дотримуватися (рис. 1.1).

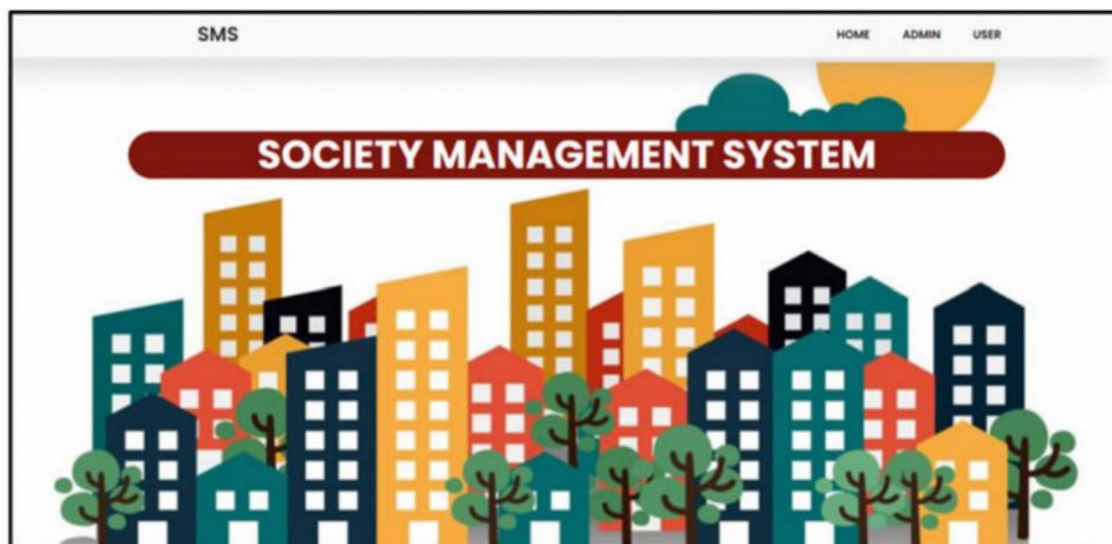


Рисунок 1.1 – Головна сторінка системи

Цільова сторінка веб-додатку – це домашня сторінка, яка складається з двох модулів: входу для адміністратора та входу для користувача (рис. 1.2).

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

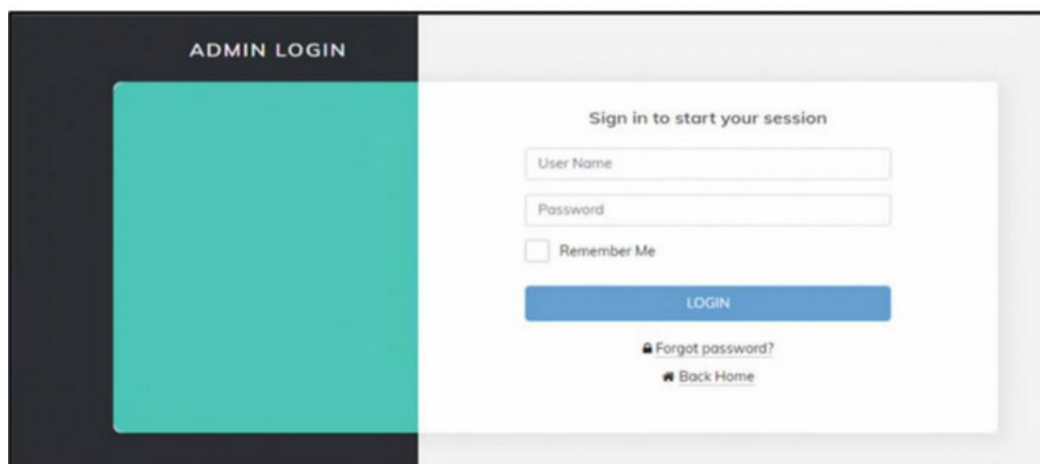


Рисунок 1.2 – Сторінка входу для адміністратора

Цільова сторінка адміністратора, з якої можуть увійти зареєстровані адміністратори системи, показана на рисунку 1.2. Ця Система управління житловим товариством має зручний інтерфейс та функції.

Члени можуть увійти, використовуючи свій номер телефону, додаючи номер корпусу та квартири, в яких вони проживають. Ці члени створюються адміністратором (рис. 1.3).

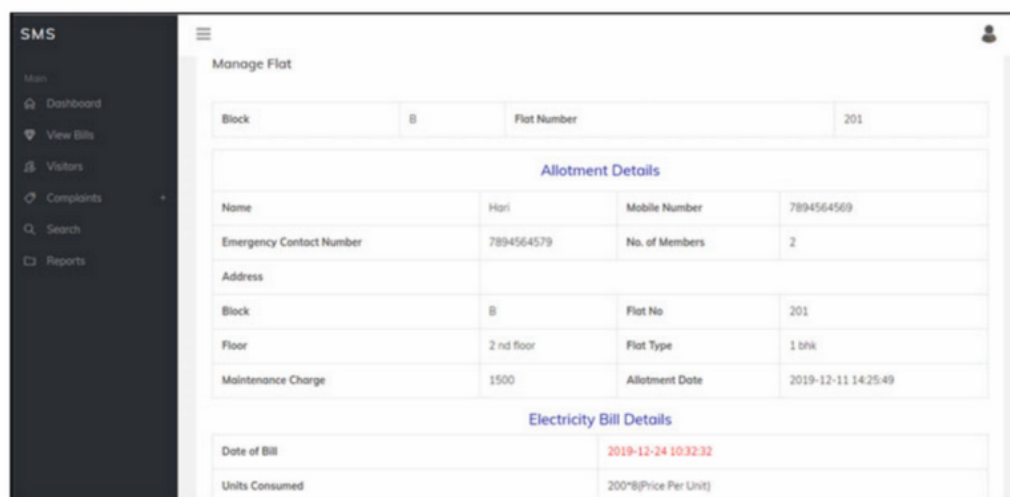


Рисунок 1.3 – Дашборд системи

Панель керування члена надає численні функції, які дозволяють членам перевіряти статус своїх рахунків, відвідувачів та зареєстрованих ними скарг. Адміністратор відповідає за адміністрування та підтримку даних системи.

В цій системі пропонується ефективне та просте рішення для вирішення найважливіших питань, зменшуючи зусилля та зосереджуючись на надійній комунікації для подолання недоліків.

В дослідженні [2] представлено веб-додаток системи управління житловим товариством з рекомендаційною системою, який полегшує комунікацію між мешканцями та покращує соціальну взаємодію через інтеграцію з організаціями.

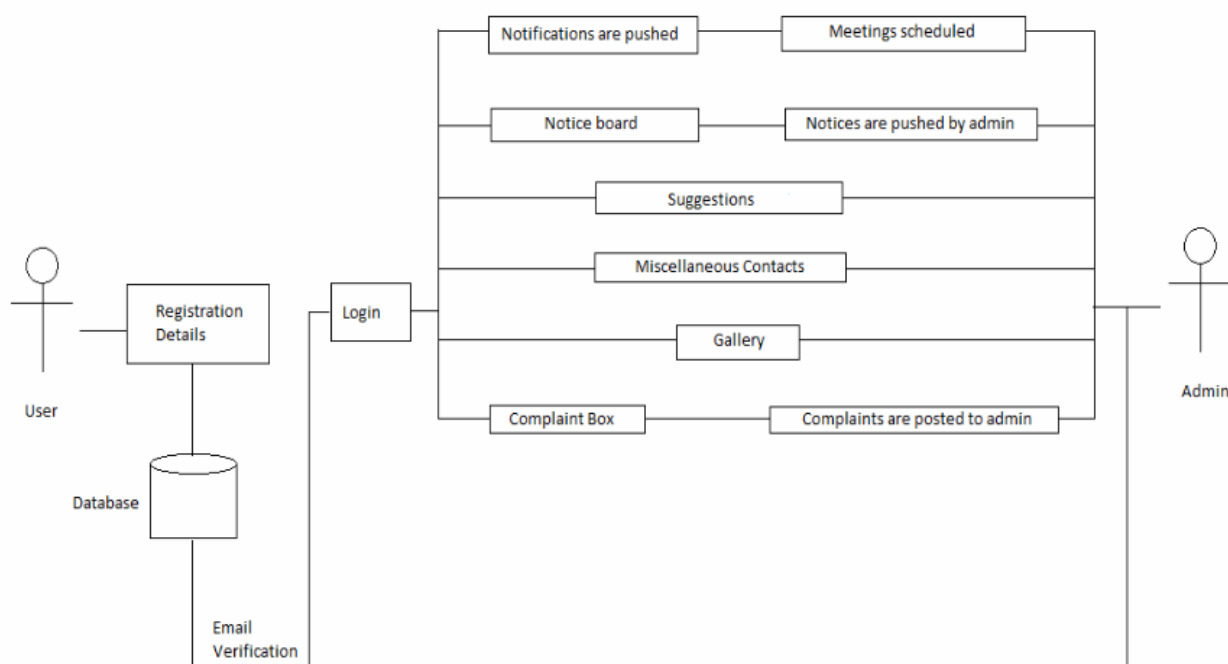


Рисунок 1.4 – Архітектура веб-додатку

Система дозволяє подавати скарги та пропозиції в електронному вигляді, а також інформує мешканців про події та прогрес у вирішенні питань.

В роботі [3] запропонована хмарна система управління житловим товариством, що наголошує на перевагах хмарних рішень для адміністрування житлових товариств. Ця система забезпечує централізоване відстеження нерухомості, онлайн-платежі, системи повідомлень, облік витрат та формування звітів. Важливою особливістю є її користувацька

зручність, автоматизація та можливість взаємодії між членами, а також функції онлайн-голосування та швидкого доступу до контактної інформації.

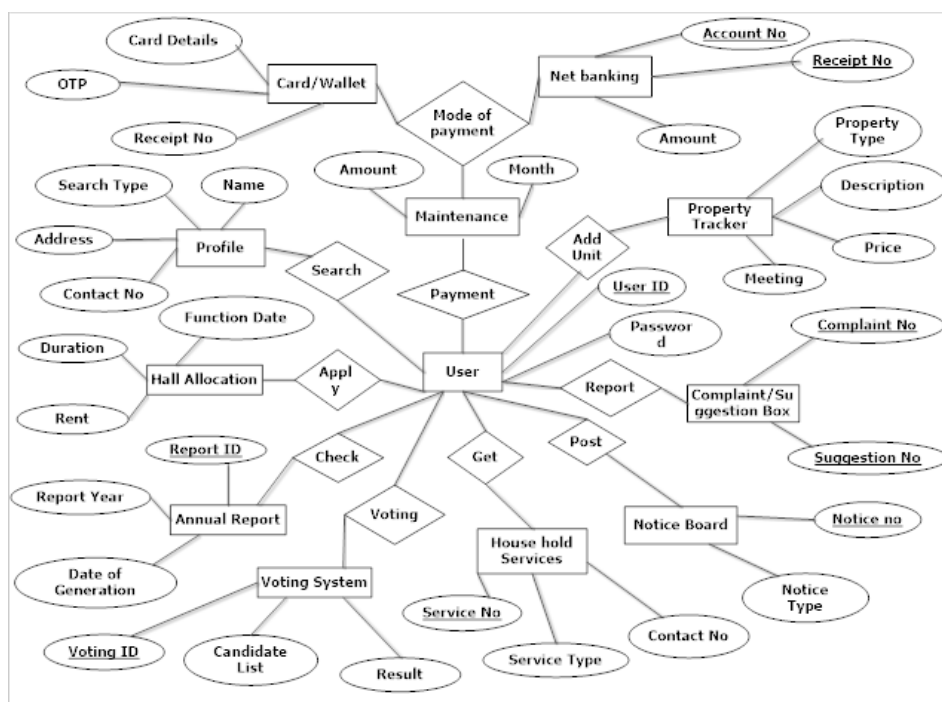


Рисунок 1.5 – Діаграма функціонування хмарної системи управління ЖИТЛОВИМ ТОВАРИСТВОМ

Модуль управління витратами забезпечує комплексний підхід до обліку та аналізу фінансових надходжень та видатків товариства. Його функціонал включає:

- Реєстрація та відстеження витрат. Систематична фіксація всіх фінансових операцій, що здійснюються товариством. Кожен запис містить детальний опис витрати, дату її здійснення та загальну суму. Це забезпечує прозорість та можливість аудиту всіх фінансових потоків.

- Імпорт даних. Можливість завантаження списків витрат у стандартизованому форматі електронних таблиць (наприклад, Microsoft Excel). Ця функція значно спрощує процес введення великих обсягів даних та інтеграцію з існуючими бухгалтерськими системами.

- Генерація звітів. Автоматизоване формування річних звітів про витрати. Ці звіти надають консолідовану інформацію про всі фінансові операції за звітний період, що є критично важливим для аналізу ефективності використання коштів.

ANNUAL REPORT		
Year(2017)		
No.	PARTICULARS	AMOUNT
1.	10th Year Celebration	20,280
2.	Watch-man Sallary	36,000
3.	Electricity charges	98,856
4.	Payments to Workes	50235
5.	Pooja Expenses	80,376
6.	Diwali Celebration	20,560
	TOTAL:	3,06,307

Print

Рисунок 1.6 – Приклад формування річного звіту

- Формування звітів про фонд товариства. На основі даних звітів про витрати система генерує звіт про стан фонду товариства. Цей звіт дозволяє керівництву та членам товариства оперативно оцінювати поточний фінансовий стан та приймати обґрунтовані рішення щодо управління ресурсами.

В роботі [4] також представили веб-базований додаток для цифровізації управління громадою (товариством), що відстежує щоденні сповіщення, зустрічі та культурні заходи, зменшуючи паперовий документообіг.

Система управління товариством – це вебсайт, розроблений для зменшення конфліктів між членами громади. Система має автоматичну функцію розрахунку щомісячних платежів за обслуговування, і член може переглядати статус свого рахунку у своєму обліковому записі.

Основною функцією цього проекту є наявність системи голосування за різні громадські посади, такі як Секретар, Голова, Скарбник тощо. Член може голосувати за кандидатів, які представляють різні ролі в громаді.

Система управління товариством дозволяє членам входити у свій обліковий запис та отримувати оновлення про соціальні події.

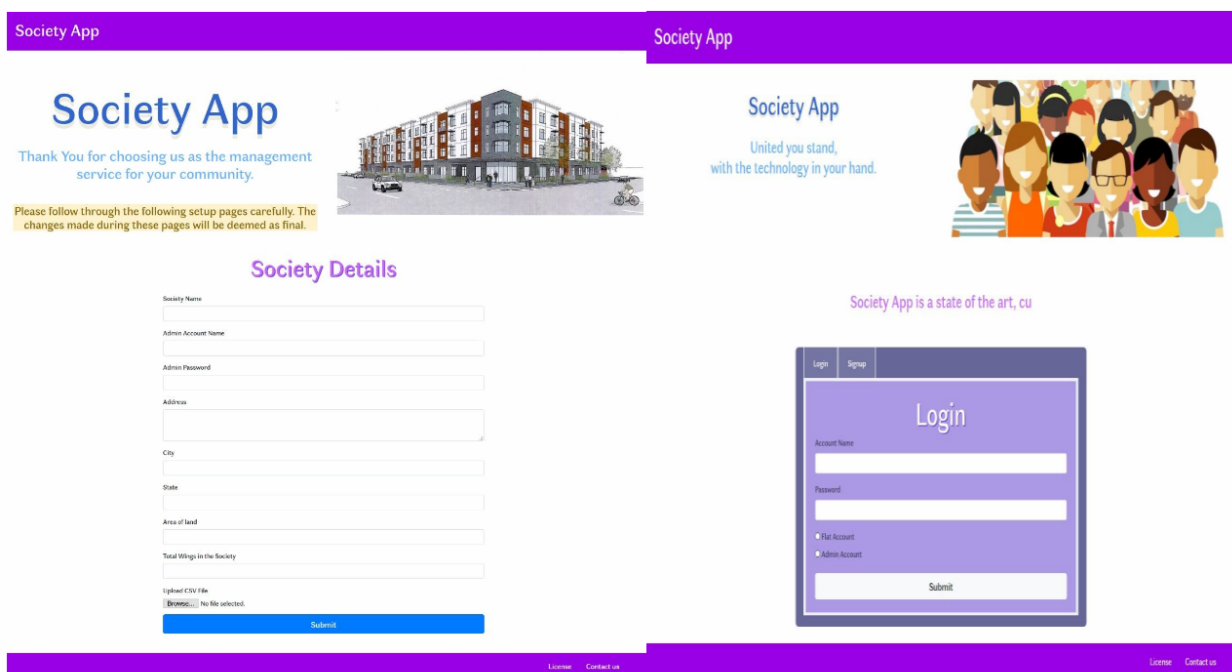


Рисунок 1.7 – Веб-ресурс “Система управління товариством”

Система управління товариством втілює фактичну роботу товариства, дозволяючи постачальникам послуг з управління житловими товариствами швидко впроваджувати та експлуатувати безпечні та недорогі додатки, при цьому отримуючи вигоду від нижчих витрат на обслуговування та оновлення протягом усього терміну співпраці.

1.3.2. Мобільні додатки для управління житловою компанією / товариством

Дослідження [5] "Додаток для управління товариством на Android" підкреслює важливість автоматизації управління житловим товариством, оскільки традиційні методи комунікації часто страждають від недоліків, таких як відсутність прозорості та неефективна доставка важливих повідомлень. Автори пропонують мобільний додаток для Android, який

					БР.ІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

централізує такі послуги, як розгляд скарг, організація зустрічей, обмін повідомленнями та доступ до правил.

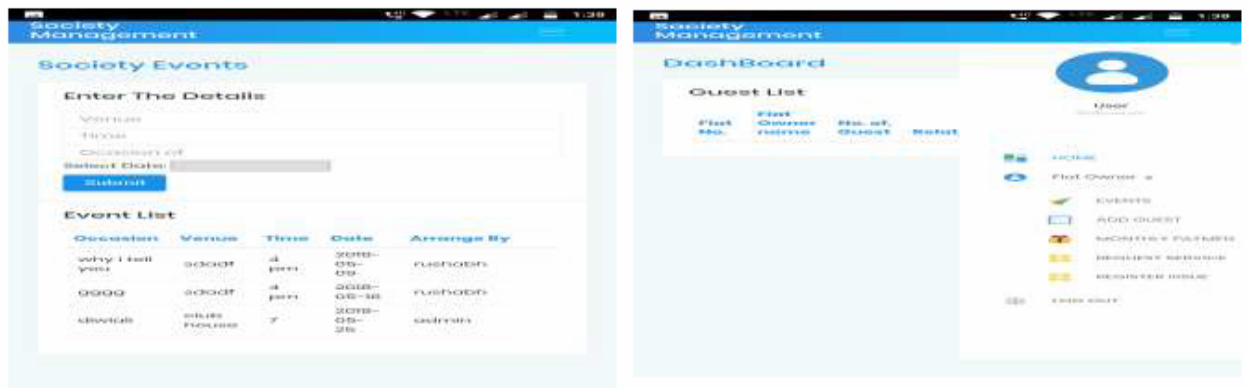


Рисунок 1.8 - Додаток для управління товариством на Android

Щоб подолати недоліки існуючої системи, пропонується розумніший та ефективніший спосіб вирішення критичних питань, зменшуючи зусилля та вдосконалюючи надійну комунікацію. Різні функції в товаристві, такі як скарги, зустрічі, оголошення, пропозиції, правила, різноманітні контакти, будуть доступні в одному місці, щоб користувач міг їх переглядати та використовувати за необхідності.

Цей додаток, який буде корисним для всіх членів товариства, щоб постійно отримувати актуальну інформацію, пов'язану з товариством. Ця система дозволяє входити в систему для персоналу/охорони/власників за допомогою унікального QR-коду, що містить їхнє ім'я, пароль, адресу, номер телефону.

В роботі [6] "Управління житловим товариством" також зосереджуються на автоматизації повсякденних дій товариства за допомогою додатка для Android. Їхня система включає функції календаря, дотримання правил, домашньої допомоги, а також зберігання інформації про оголошення та заходи. Тут провели дослідження реалізації подібного додатка для Android, підтверджуючи актуальність мобільних рішень для повсякденних потреб управління житловим фондом.

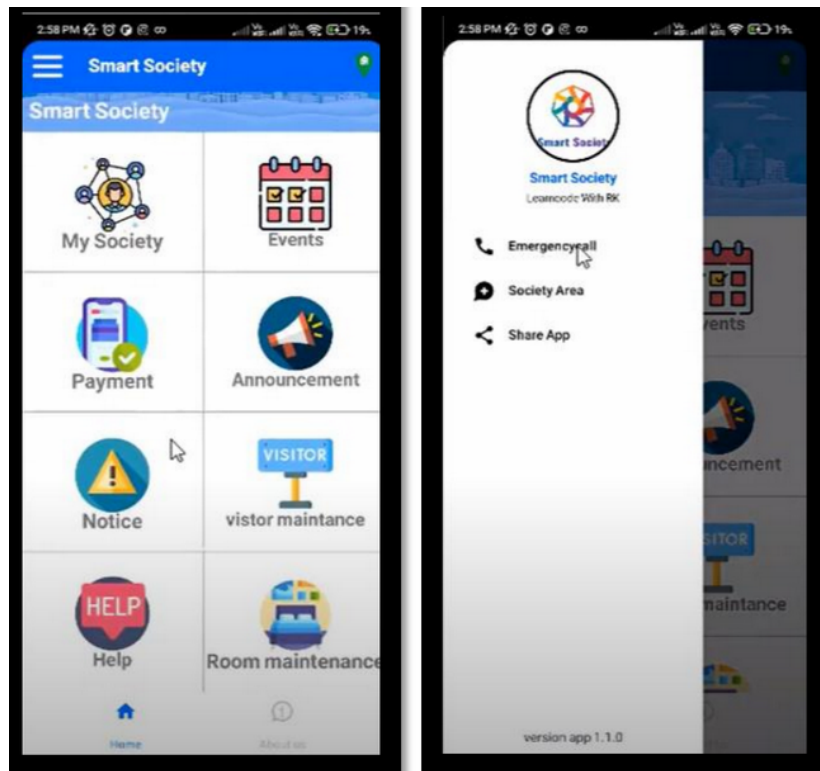


Рисунок 1.9 – Додаток управління житловим товариством

Останні розробки, такі як "Додаток Society Connect: Е-менеджер для житлового товариства" демонструють подальший прогрес у цій галузі. Цей додаток для Android спрощує збір платежів за обслуговування, бронювання об'єктів, обмін повідомленнями, а також управління відвідувачами та персоналом. Він оптимізує процеси, зменшує паперову роботу та підвищує прозорість, дозволяючи користувачам оплачувати рахунки онлайн, переглядати історію платежів та отримувати сповіщення про події.

1.3.3. Новітні тенденції та додаткові функціональності

У контексті пандемії COVID-19, в роботі [7] автор підкреслив важливість систем, які мінімізують необхідність особистих контактів. Його система передбачає функції щоденних повідомлень, графіків зустрічей, оновлень про культурні заходи, розподілу коштів та спеціалізованих послуг охорони здоров'я, включаючи функцію SOS. Також інтегрований чат-бот для полегшення навігації користувачів.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

Перспективним напрямком є інтеграція Інтернету речей (IoT) у системи управління. В [9] описують систему управління житловим товариством з використанням IoT, досліджують можливості підключення фізичних об'єктів до Інтернету для автоматизації повсякденних дій. Це дозволяє спростити та покращити функціонування товариства, автоматизуючи рутинні процеси.

Огляд літератури свідчить про активний розвиток автоматизованих систем для управління житловими товариствами. Перехід від традиційних ручних методів до веб- та мобільних додатків, а також інтеграція новітніх технологій, таких як IoT та хмарні обчислення, є ключовими тенденціями. Ці системи спрямовані на підвищення ефективності комунікації, забезпечення прозорості, спрощення адміністративних процесів та підвищення рівня життя мешканців. Подальші дослідження будуть зосереджені на вдосконаленні функціоналу, інтеграції нових технологій та забезпеченні високого рівня безпеки та зручності користувача.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

РОЗДІЛ 2. МОДЕЛЮВАННЯ АРХІТЕКТУРИ СИСТЕМИ ДЛЯ ЖИТЛОВО-ЕКСПЛУАТАЦІЙНИХ КОМПАНІЙ

2.1. Критичний аналіз існуючих методологій управління житловими компаніями та системні вимоги для розробки

Управління житловими компаніями в сучасних урбаністичних умовах стикається з низкою значних викликів, зумовлених обмеженнями традиційних комунікаційних та інформаційних систем. Незважаючи на поширеність загальних дошок оголошень та спроби інтеграції автоматизованих чат-інфраструктур, ці підходи демонструють суттєві недоліки, які перешкоджають ефективному та прозорому функціонуванню товариств.

2.1.1. Недоліки існуючих підходів до комунікації

Використання традиційних методів управління інформацією в житлових товариствах, таких як фізичні дошки оголошень, характеризується низкою системних недоліків:

- Відсутність надійності та автентичності інформації: Інформація, розміщена на загальних дошках, часто піддається маніпуляціям, включаючи несанкціоновані зміни, некоректне розміщення або навіть фальсифікацію даних. Це створює передумови для непорозумінь та підриває довіру до офіційних повідомлень. Ручний спосіб оновлення інформації також підвищує ризик людської помилки.

- Низька доступність інформації: Залежність від фізичної дошки оголошень призводить до обмеженої доступності інформації. Члени товариства можуть ігнорувати або забувати перевіряти оновлення, а адміністратори можуть несвоєчасно оновлювати дані. Це порушує

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

оперативну взаємодію та призводить до інформаційного вакууму для значної частини мешканців.

- Відсутність механізмів підтвердження: Ручний процес комунікації не дозволяє достовірно відстежувати доставку повідомлень до всіх зацікавлених сторін. Відсутність зворотного зв'язку або підтвердження отримання інформації унеможливорює оцінку ефективності комунікації та може спричиняти затримки у прийнятті рішень або усуненні проблем.

- Трудомісткість процесу: Підтримка фізичної дошки оголошень вимагає постійної відповідальності та значних часових витрат з боку призначеної особи. Це створює додаткове навантаження на персонал та може конфліктувати з їхніми особистими графіками, що негативно позначається на регулярності та якості оновлення інформації.

Крім того, хоча автоматизовані чат-рішення можуть бути корисними для загальної комунікації, вони, як правило, не надають спеціалізованого середовища для цільових оголошень або обговорень конкретних питань управління житлом. Це призводить до перевантаження інформаційного простору нерелевантними даними, що ускладнює пошук необхідної інформації та відволікає від основної теми.

2.1.2. Системні вимоги

Для розробки ефективної та функціональної веб-базованої системи управління житловим товариством були визначені наступні апаратні та програмні вимоги.

Компонент	Специфікація
Додаток	Веб-додаток
Пам'ять (ОЗУ)	4 ГБ
Сховище	64 ГБ
Жорсткий диск	160 ГБ
Клавіатура	Стандартна клавіатура Windows
Процесор	Процесор Intel i3 або еквівалент

Рисунок 2.1 – Апаратні вимоги

Компонент	Специфікація
Операційна система	Windows (будь-яка сучасна версія)
IDE	Visual Studio Code
Мови програмування	Angular JS, NodeJS, TypeScript
База даних	MySQL
Інструменти	VS Code, XAMPP, phpMyAdmin, MySQL Workbench, Stripe
Сервер	Apache Tomcat

Рисунок 2.2 – Системні вимоги

2.2. Технологічний стек веб-базованої системи управління житлово-експлуатаційною компанією

Розроблена веб-базована система управління житловою компанією базується на сучасних та ефективних технологіях, що забезпечують її функціональність, масштабованість та зручність використання. Системний ландшафт розгорнуто в локальному середовищі з використанням інтегрованого пакету XAMPP, який включає сервер Apache Tomcat для веб-сервера та MySQL для управління базами даних. Для розробки інтерфейсної частини (фронтенду) використано фреймворк Angular JS, тоді як для реалізації серверної логіки (бекенду) застосовано платформу Node JS. Візуалізація даних та адміністрування бази даних здійснюються за допомогою phpMyAdmin. Інтеграція платіжного шлюзу Stripe забезпечує можливість здійснення онлайн-платежів. Середовищем розробки виступає Visual Studio Code.

2.2.1. Angular JS (Фронтенд)

AngularJS є потужною JavaScript-бібліотекою з відкритим вихідним кодом, що використовується для розробки інтерфейсів користувача. У поточному проекті застосовано одну з останніх версій Angular, яка є одним з

провідних фреймворків для створення односторінкових додатків (Single Page Applications, SPA). Ключові особливості Angular включають:

- Типобезпечність: Код розробляється з використанням TypeScript, що забезпечує статичну типізацію та підвищує надійність та зручність підтримки коду.

- Компонентна архітектура: Додатки будуються на основі компонентів, що сприяє модульній структурі, перевикористанню коду та полегшує керування складними інтерфейсами.

- Шаблонна структура файлів: Проект Angular характеризується наявністю шаблонних файлів, що спрощує початкове налаштування. Конфігураційна інформація проекту, включаючи точки входу, зберігається у файлі angular.json.

- Життєвий цикл додатка: Виконання додатка починається з файлу main.ts, який спрямовує потік до кореневого модуля app.module.ts. Цей модуль, позначений декоратором @NgModule, оголошує всі компоненти, що використовуються в додатку, роблячи їх доступними для Angular. Підключення точки входу до кореневого компонента (наприклад, app.component.ts) здійснюється через app.module.ts.

Приклади структур HTML-файлів до та після компіляції додатка наведено нижче, демонструючи інтеграцію згенерованих JavaScript-файлів.

Лістинг 2.1. Структура HTML-файлів до компіляції додатка

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>My Hello World App!</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

					БР.ІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

Лістинг 2.2. Структура HTML-файлів після компіляції додатка

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>My Hello World App!</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
  <script type="text/javascript" src="runtime.js"></script>
  <script type="text/javascript" src="es2015-polyfills.js" nomodule></script>
  <script type="text/javascript" src="polyfills.js"></script>
  <script type="text/javascript" src="styles.js"></script>
  <script type="text/javascript" src="vendor.js"></script>
  <script type="text/javascript" src="main.js"></script>
</body>
</html>
```

2.2.2. *Bootstrap* (Фронтенд-фреймворк для дизайну)

Для забезпечення адаптивного та естетично привабливого макета інтерфейсу користувача використано фреймворк Bootstrap. Це дозволяє створювати елементи UI, які коректно відображаються на різних пристроях та розмірах екранів, забезпечуючи зручність користувача.

У даному проєкті Bootstrap використовується для макета фронтенду (layout). Це означає, що він допомагає структурувати вміст сторінки, розташовувати елементи (наприклад, меню, форми, таблиці) у правильному порядку, забезпечувати їхню візуальну привабливість та адаптивність до різних розмірів екранів. Завдяки Bootstrap, інтерфейс системи буде виглядати професійно та буде зручним для користувачів, незалежно від пристрою, з якого вони отримують доступ до системи.

2.2.3. *Node.js* (Бекенд)

Node.js є крос-платформовим середовищем виконання JavaScript, що дозволяє розробникам створювати серверні додатки з використанням JavaScript. На відміну від традиційного використання JavaScript у веб-браузерах для взаємодії з користувачем, Node.js розширює його можливості

					БР.ІІ – 55.00.00.000 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

на серверний рівень. Це дозволяє реалізовувати логіку обробки запитів, виконання SQL-запитів до бази даних для отримання інформації та формування відповідей для клієнтської частини. Використання Node.js забезпечує високу продуктивність та масштабованість бекенду.

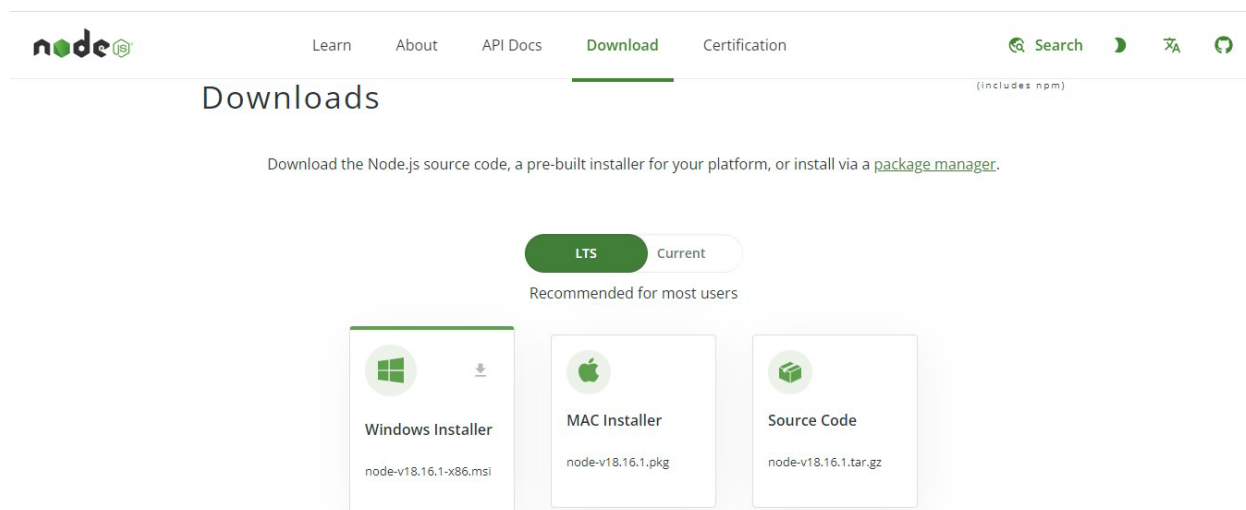


Рисунок 2.3 - Node.js

2.2.4. MySQL

MySQL є реляційною системою управління базами даних (РСУБД), яка використовується для зберігання, маніпулювання та видалення структурованих даних. Дані організовані у вигляді таблиць, що складаються з рядків та стовпців, забезпечуючи ефективне зберігання та доступ до інформації про житлове товариство.

2.2.5. Інструмент phpMyAdmin

phpMyAdmin є безкоштовним веб-додатком з відкритим вихідним кодом, що надає графічний інтерфейс користувача (GUI) для управління базою даних MySQL. Цей інструмент значно спрощує процеси візуалізації даних, виконання запитів, адміністрування таблиць та користувачів бази даних.

					БР.ІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

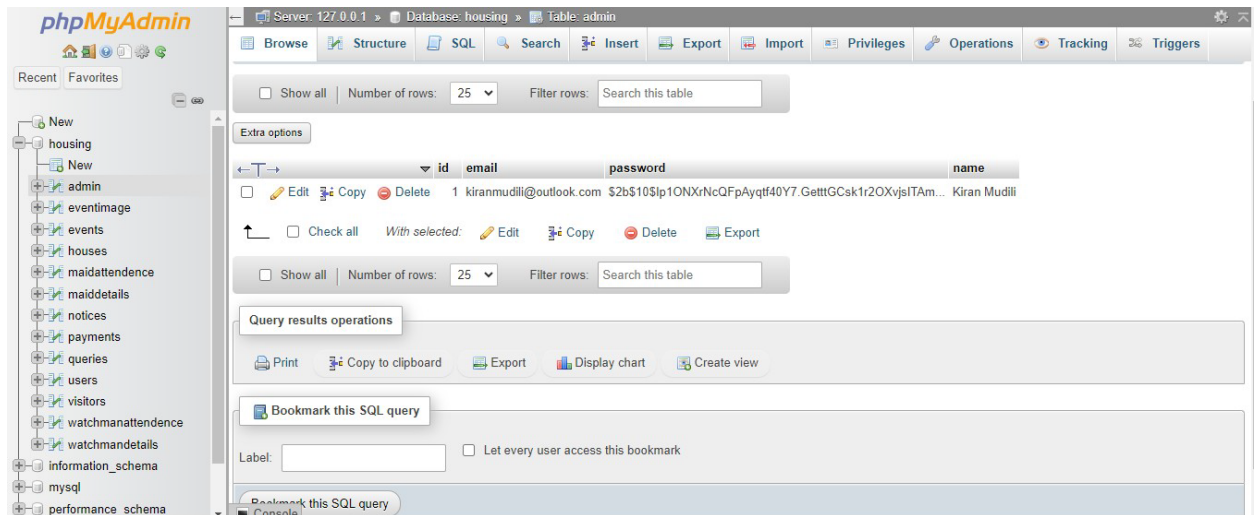


Рисунок 2.4 – phpMyAdmin

2.2.6. Середовище ХАМРР

ХАМРР – це популярний програмний пакет з відкритим вихідним кодом, який надає повне локальне веб-серверне середовище для розробки та тестування веб-додатків.

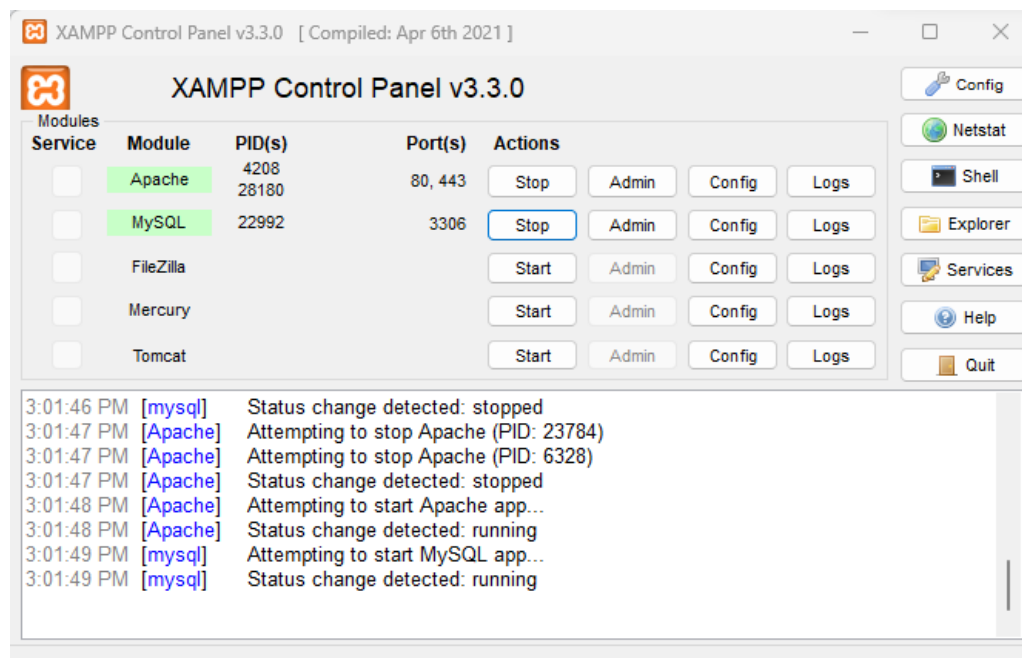


Рисунок 2.5 – ХАМРР

ХАМРР інтегрує Apache HTTP Server та MySQL, а також включає phpMyAdmin для зручного управління базою даних. Це середовище ідеально

						Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

підходить для розробки та тестування веб-додатків на локальній машині, проте не рекомендується для використання у виробничих середовищах.

2.3. Візуалізація системного дизайну за допомогою Unified Modeling Language

Ефективна комунікація архітектурних та функціональних аспектів розробленої системи може бути досягнута за допомогою стандартизованих діаграм, зокрема з використанням Unified Modeling Language (UML). UML є уніфікованою мовою моделювання, яка широко застосовується в галузі об'єктно-орієнтованого програмного інжинірингу. Розроблена та підтримувана Object Management Group (OMG), UML слугує загальним інструментом для візуалізації, специфікації, конструювання та документування артефактів програмних систем. Основними елементами UML є метамодель, яка визначає синтаксис та семантику мови, та нотація, що надає графічні символи для побудови моделей.

2.3.1. Архітектурна діаграма

Для візуалізації загальної структури та взаємодії ключових компонентів системи використовується архітектурна діаграма. Ця діаграма надає високорівневе уявлення про організацію системи, її основні шари, модулі та потоки даних між ними.

Ця діаграма ілюструє взаємодію користувачів, адміністраторів та служби безпеки з системою управління житловою компанією. Вона показує послідовність дій та доступні функції для кожної ролі.

Згідно діаграми користувач має наступні можливості:

Login - Користувач входить до системи.

View event - Переглядає інформацію про події.

View notice - Переглядає оголошення та повідомлення.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

Payment - Здійснює платежі.

Logout - Виходить із системи.

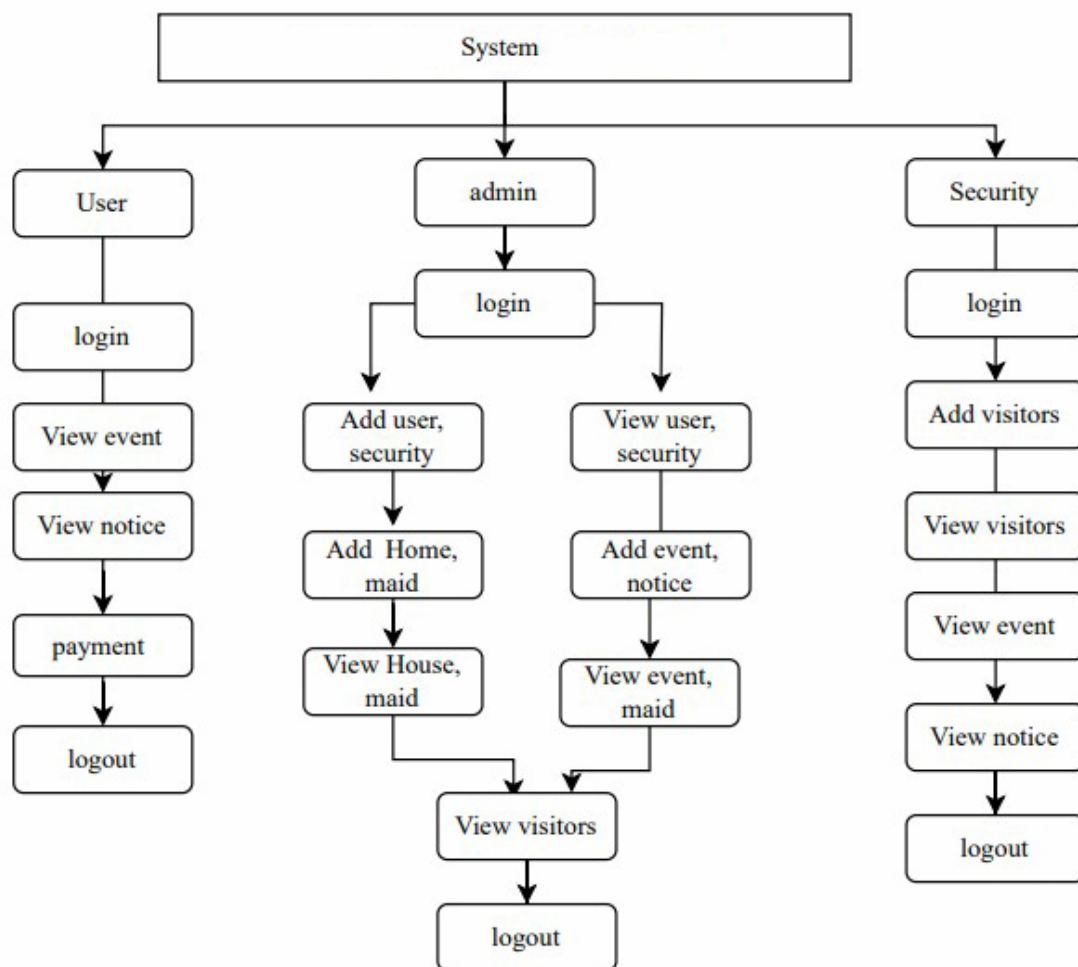


Рисунок 2.6 – Архітектурна діаграма

2.3.2. Представлення Use Case

Представлення use case є поведінковим аспектом моделювання в UML, що відображає функціональність системи з точки зору зовнішніх акторів. Воно створюється на етапі аналізу вимог та надає схематичне зображення усіх функціональних можливостей, які система повинна надавати. Кожен use case описує послідовність дій, які система виконує у відповідь на дії актора, що взаємодіє з системою для досягнення певної мети.

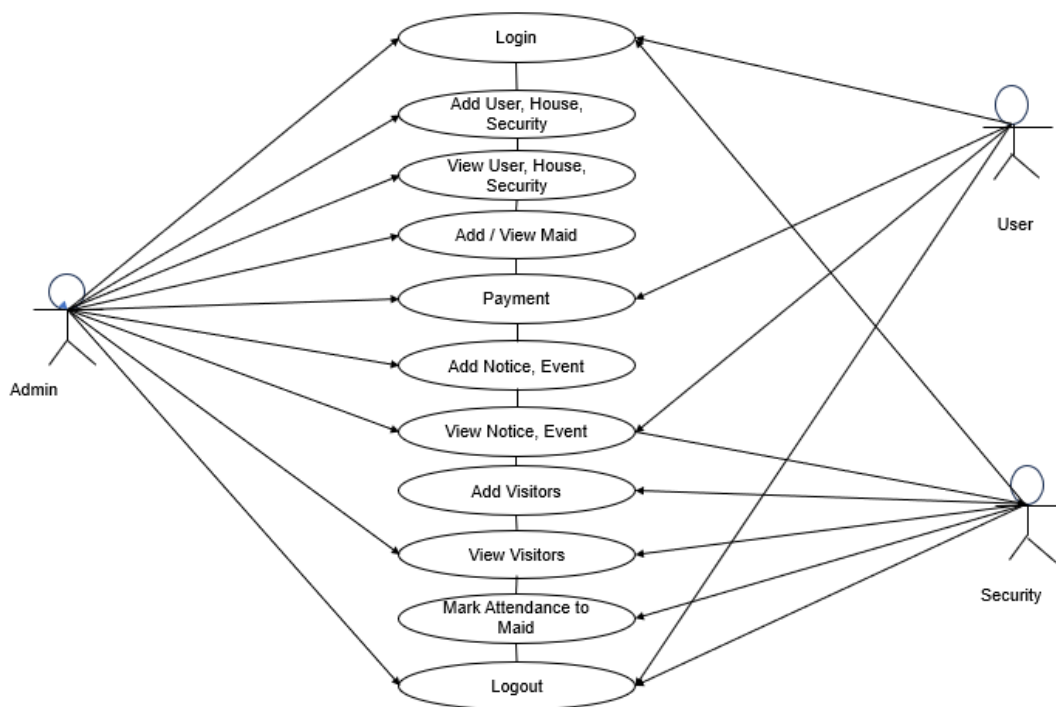


Рисунок 2.7 – Use Case діаграма

2.4. Моделювання динамічних та статичних аспектів системи за допомогою UML діаграм

Для всебічного розуміння архітектури та поведінки розробленої системи управління житловою компанією, окрім загального представлення Use Case, застосовуються детальніші діаграми Unified Modeling Language (UML). Ці діаграми дозволяють візуалізувати як послідовність взаємодій між об'єктами, так і статичну структуру системи, а також потоки її активностей.

2.4.1. Представлення діаграми послідовності (Sequence Diagram)

Представлення послідовності, також відоме як діаграма послідовності, є одним з типів комунікаційних діаграм UML. Воно деталізує, яким чином операції та об'єкти системи взаємодіють між собою у часовій послідовності, показуючи порядок виклику методів та обмін повідомленнями. Ця діаграма є особливо корисною для моделювання сценаріїв використання (Use Cases) та

демонструє потік управління та обробки даних між різними компонентами системи.

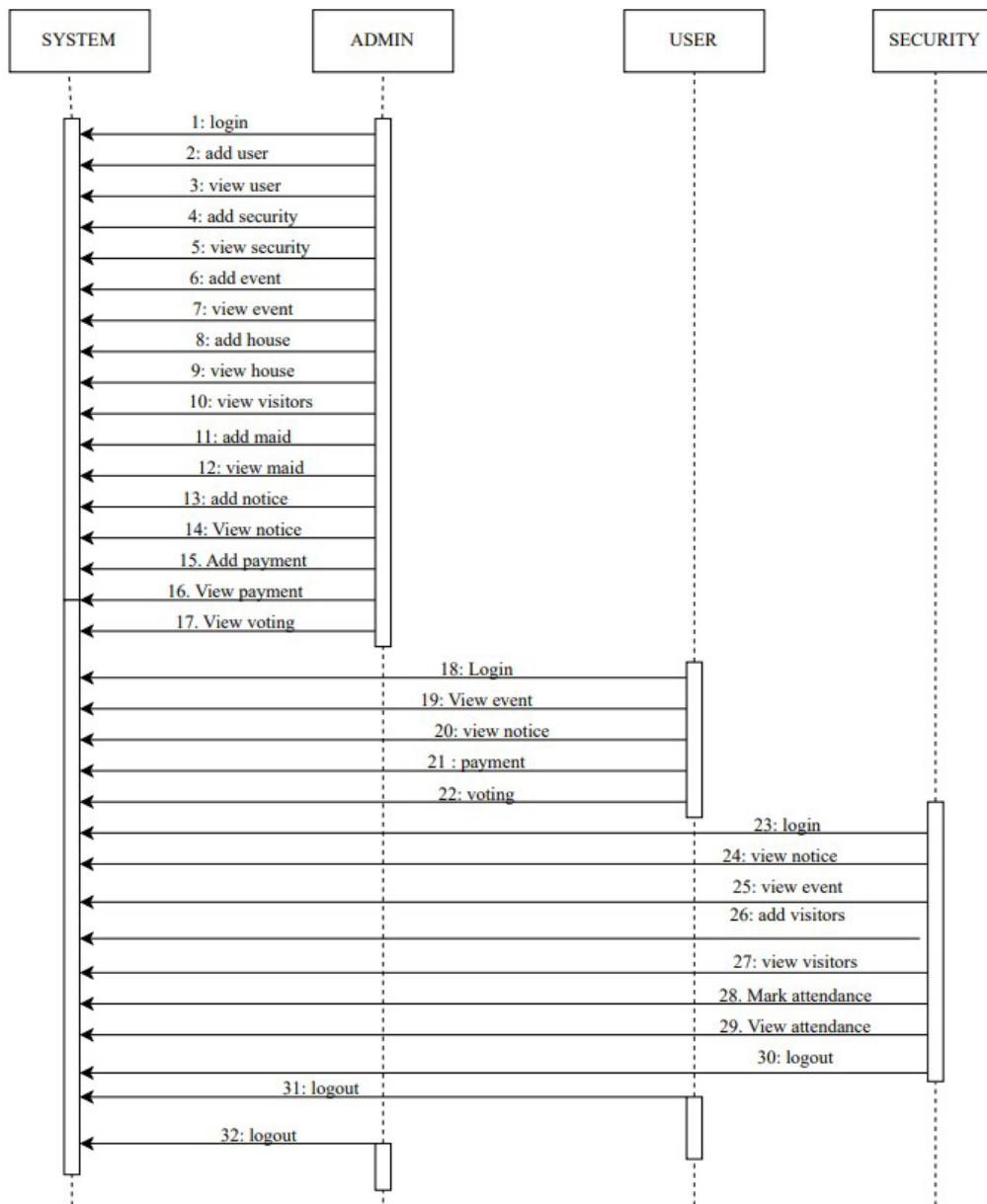


Рисунок 2.8 – Діаграма послідовності

2.4.2. Представлення діаграми класів (Class Diagram)

Представлення класу, або діаграма класів, є ключовим статичним структурним представленням в UML. Вона описує логічну структуру системи, відображаючи класи, їхні атрибути (властивості), операції (методи) та різні типи зв'язків між класами (наприклад, асоціації, агрегації, композиції,

узагальнення). Ця діаграма є фундаментальною для об'єктно-орієнтованого дизайну, оскільки вона надає вичерпну інформацію про дані, що зберігаються в системі, та поведінку, пов'язану з цими даними.

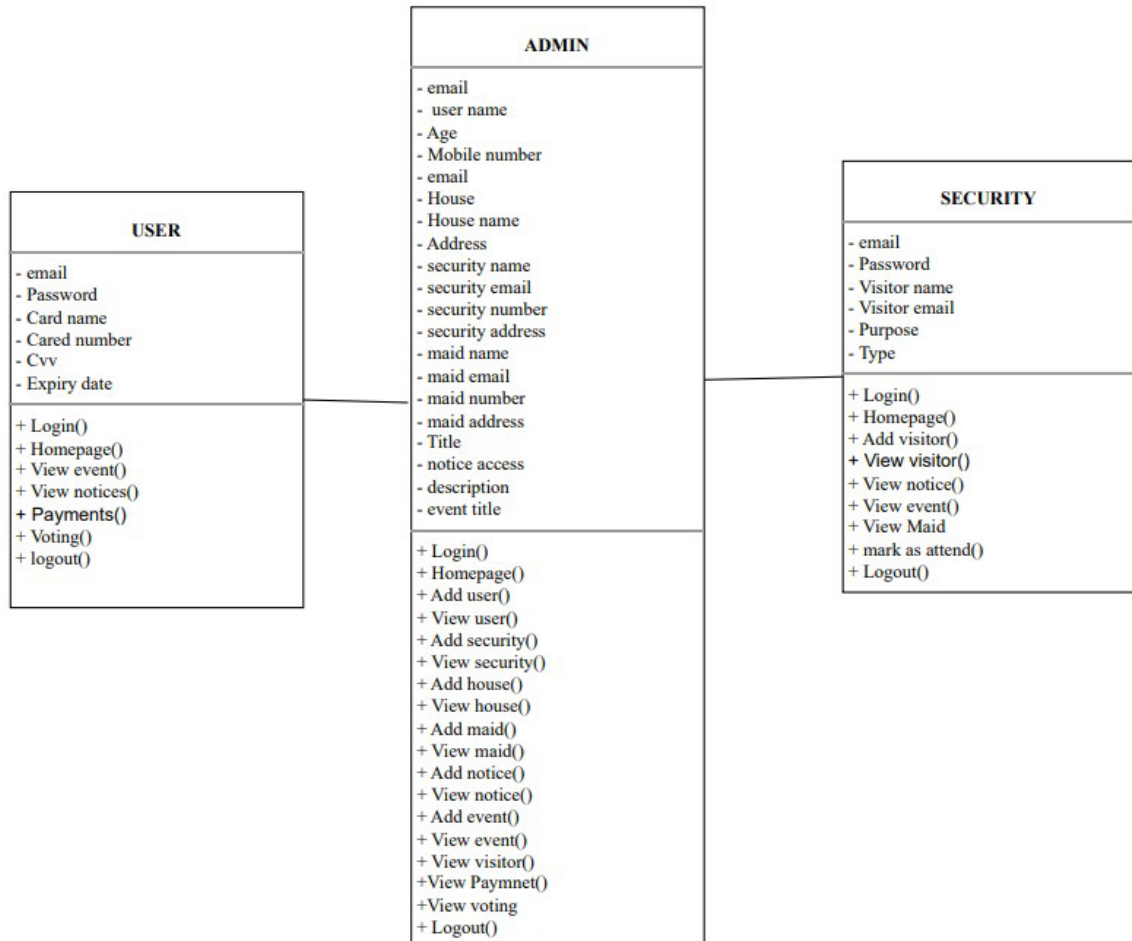


Рисунок 2.9 – Діаграма класів

2.4.3. Розробка діаграми активності (Activity Diagram)

Представлення активності, або діаграма активності, є візуальним зображенням покрокових функцій та дій, що виконуються в системі. Вона здатна моделювати складні потоки управління, підтримуючи паралелізм (одночасне виконання дій), ітерації (повторення дій) та вибір (умовні переходи). У контексті UML, діаграми активності часто використовуються для моделювання бізнес-процесів, робочих потоків або послідовності кроків

обробки елементів у системі. Вони надають чітке уявлення про весь потік управління, від початку до завершення певної діяльності.

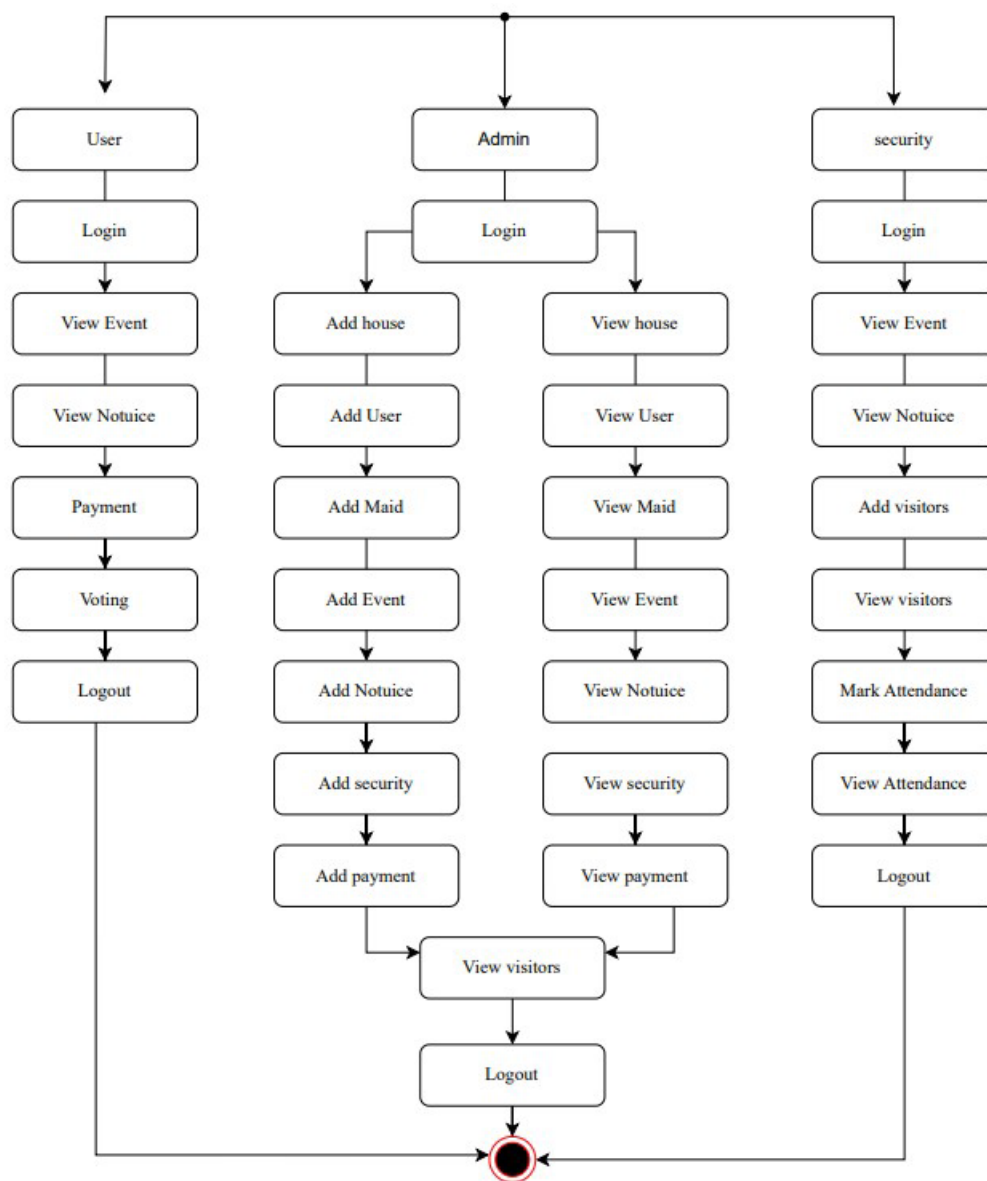


Рисунок 2.10 – Діаграма активності

2.5. Моделювання структури бази даних за допомогою діаграми сутність-зв'язок (ER-модель)

Для ефективного проектування та документування реляційних баз даних широко використовується модель Сутність-Зв'язок (Entity-Relationship,

ER-модель). ER-модель є високорівневим концептуальним представленням даних, що слугує "планом" або "дизайном" бази даних, який згодом може бути реалізований у конкретній системі управління базами даних.

Основними елементами ER-моделі є:

- **Набір Сутностей (Entity Set):** Сутність є об'єктом або концептом реального світу, який має самостійне значення та може бути однозначно ідентифікований. У контексті системи управління інформацією, сутність може відповідати таблиці в реляційній базі даних. Набір сутностей об'єднує сутності одного типу.

- **Атрибути (Attributes):** Кожна сутність характеризується набором атрибутів, які описують її властивості. Атрибути відповідають стовпцям у таблицях бази даних. Кожна сутність у наборі сутностей має однаковий набір атрибутів.

- **Набір Зв'язків (Relationship Set):** Зв'язок представляє асоціацію або взаємодію між двома або більше сутностями. Набір зв'язків об'єднує зв'язки одного типу. Зв'язки можуть мати власні атрибути та характеризуватися кратністю (наприклад, один-до-одного, один-до-багатьох, багато-до-багатьох).

ER-представлення забезпечує візуалізацію логічної структури бази даних шляхом ілюстрації взаємозв'язків між наборами сутностей та їхніми атрибутами. Це дозволяє розробникам та аналітикам:

- **Концептуалізувати дані:** Створити чітке та зрозуміле уявлення про дані, що зберігатимуться в системі, незалежно від конкретної технології реалізації.

- **Виявити взаємозв'язки:** Ідентифікувати та визначити типи взаємодії між різними сутностями системи.

- **Зменшити надмірність даних:** Оптимізувати структуру бази даних, мінімізуючи дублювання інформації.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

- Покращити комунікацію: Надати стандартизований та зрозумілий спосіб документування структури бази даних для всіх зацікавлених сторін.

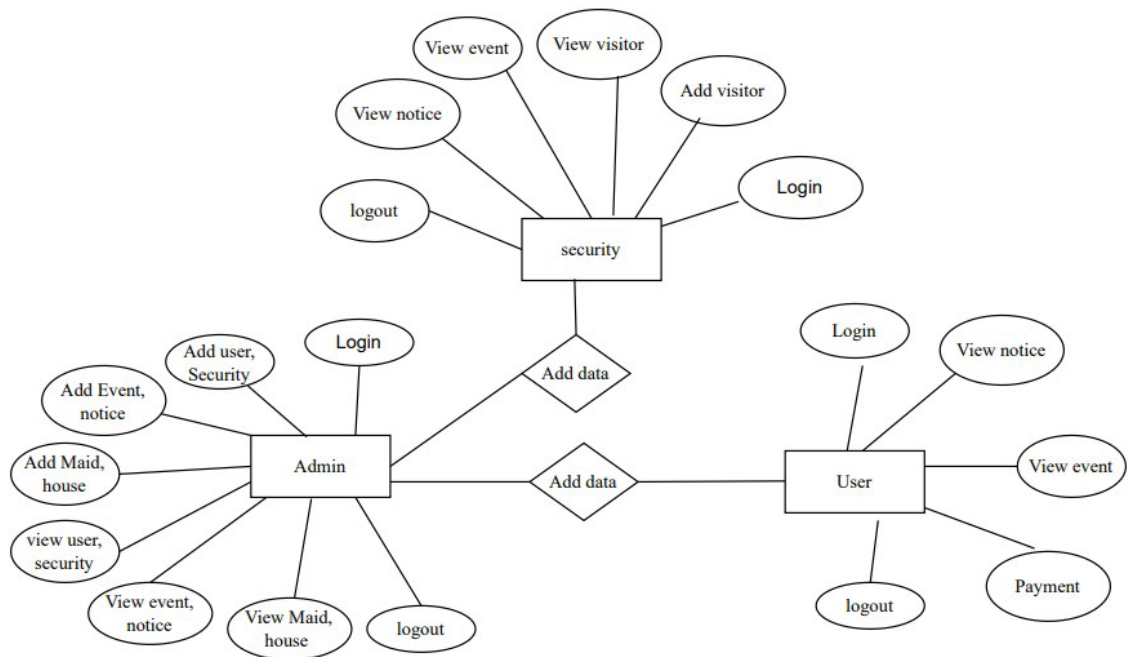


Рисунок 2.11 – ER діаграма системи

Таким чином, ER-модель є фундаментальним інструментом у процесі проектування баз даних, що забезпечує систематизований підхід до організації та управління інформацією в складних програмних системах.

РОЗДІЛ 3. ПРОГРАМНА ІМПЛЕМЕНТАЦІЯ WEB-БАЗОВАНОЇ СИСТЕМИ ДЛЯ ЖИТЛОВО-ЕКСПЛУАТАЦІЙНИХ КОМПАНІЙ

3.1. Системний аналіз пропонованої веб-базованої системи

Представлений документ присвячений системному аналізу розробленого веб-додатку, призначеного для оптимізації управління житловими товариствами та мінімізації існуючих конфліктів. Система забезпечує централізований доступ до інформації про події та соціальну активність у межах товариства для всіх його членів.

3.1.1. Запропонована архітектура системи

Розроблений додаток структурований за трьома основними функціональними модулями: адміністратор, користувач та безпека. Інформація про профілі користувачів, події, дані відвідуваності та бронювання квартир зберігається у централізованій базі даних.

Модуль адміністратора

Модуль Адміністратора надає повний спектр можливостей для контролю та управління системою. Його компоненти включають:

- Управління користувачами: Функції додавання та перегляду облікових записів користувачів.
- Управління будинками: Можливості додавання та перегляду інформації про житлові будинки.
- Управління безпекою: Функції додавання та перегляду даних щодо персоналу служби безпеки.
- Управління персоналом з прибирання: Додавання та перегляд інформації про прибиральниць.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

- Управління повідомленнями: Функціонал для створення та перегляду внутрішніх повідомлень.

- Управління подіями: Додавання та перегляд інформації про заплановані події.

- Перегляд відвідувачів: Доступ до реєстру відвідувачів.

- Управління платежами: Додавання та перегляд даних про фінансові платежі.

- Голосування: Адміністрування системи голосування.

Усі критичні повідомлення щодо функціонування товариства автоматично розсилаються кожному члену.

Модуль користувача

Модуль користувача надає доступ до основних функцій системи для зареєстрованих членів товариства.

Він включає наступні компоненти:

- Події: Перегляд інформації про майбутні та минулі події.

- Повідомлення: Отримання та перегляд системних повідомлень.

- Платежі: Перегляд статусу та історії платежів.

- Голосування: Участь у голосуваннях.

Модуль безпеки

Модуль безпеки розроблений для управління функціями, пов'язаними з контролем доступу та безпекою. Його компоненти включають:

- Відвідувачі: Функція додавання нових відвідувачів.

- Прибиральниці: Перегляд інформації про прибиральниць та відмітка їхньої присутності.

- Повідомлення: Доступ до відповідних повідомлень.

- Події: Перегляд інформації про події.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

3.1.2. Деталізація функціональності веб-додатка

Система реалізована як веб-додаток з окремими точками входу для кожної ролі.

Вхід адміністратора

Модуль входу адміністратора забезпечує доступ до розширеного функціоналу системи для зареєстрованих адміністраторів, інформація про яких зберігається в базі даних. Після успішної автентифікації адміністратор отримує можливість управляти всіма аспектами системи, включаючи управління користувачами (додавання/перегляд), будинками (додавання/перегляд), безпекою (додавання/перегляд), персоналом з прибирання (додавання/перегляд), повідомленнями (додавання/перегляд), подіями (додавання/перегляд), перегляд відвідувачів, управління платежами (додавання/перегляд) та адміністрування системи голосування. Система забезпечує автоматичне надсилання необхідних повідомлень усім членам товариства.

Вхід користувача

Модуль входу користувача призначений для автентифікації членів товариства. Доступ до системи надається після введення коректного імені користувача та пароля. Для нових користувачів реєстрація здійснюється адміністратором. Процес реєстрації включає введення основних даних, таких як ім'я, електронна пошта, номер мобільного телефону, вік, номер будинку/квартири тощо. Після успішного входу користувачі отримують доступ до функціоналу, що включає перегляд подій, повідомлень, управління платежами та участь у голосуваннях.

Вхід безпеки

Модуль входу безпеки реалізовано аналогічно модулю входу користувача. Персонал служби безпеки повинен пройти процес

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

автентифікації за допомогою зареєстрованих облікових даних (ім'я користувача та пароль). Реєстрація нових співробітників служби безпеки здійснюється адміністратором системи.

3.2. Початкова імплементація системи управління

Цей розділ описує методологію розробки веб-базованої системи управління житловим товариством, починаючи з налаштування середовища розробки. Розглянемо основні процес розробки системи.

3.2.1. Налаштування середовища розробки

Початковим етапом було налаштування необхідного програмного середовища. Це включало:

- Встановлення Node.js: Інсталяція Node.js, що є фундаментальною платформою для бекенду.
- Глобальне встановлення Angular CLI: Виконання команди `npm install -g @angular/cli` для встановлення інтерфейсу командного рядка Angular, який є ключовим інструментом для управління проектами Angular.
- Встановлення XAMPP: Інсталяція XAMPP, інтегрованого пакету, що надає компоненти Apache (веб-сервер), MySQL (система управління базами даних) та PHP. Після встановлення було виконано конфігурацію програмного забезпечення відповідно до специфічних вимог проекту.

3.2.2. Створення проекту Angular

Після налаштування середовища було ініційовано новий проект Angular. Цей процес включав:

- Відкриття терміналу або командного рядка та вибір директорії для проекту.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

- Виконання команди `ng new housing-society-management` для генерації базової структури проекту Angular.

- Запуск сервера розробки за допомогою команди `ng serve` для локального тестування фронтенд-частини.

3.2.3. Розробка фронтенду

Розробка інтерфейсу користувача здійснювалася за допомогою фреймворку Angular. Це включало:

- Створення модулів та компонентів за допомогою Angular CLI для структурування архітектури фронтенду.

- Проектування та реалізація візуальних елементів, таких як шаблони (HTML), таблиці стилів (CSS) та логіка компонентів (TypeScript/Angular), для забезпечення функціонального та естетичного інтерфейсу.

3.2.4. Розробка бекенду (RESTful API)

Для реалізації серверної логіки було розроблено RESTful API з використанням Node.js та Express.js. Ключові аспекти включали:

- Визначення маршрутів та реалізація операцій для аутентифікації користувачів, управління послугами, управління даними мешканців тощо.

- Підключення бекенду до бази даних MySQL за допомогою відповідних бібліотек.

- Реалізація моделі бази даних та операцій CRUD (Create, Read, Update, Delete) для взаємодії з даними.

- Забезпечення належної обробки аутентифікації та авторизації в API для гарантування безпеки даних.

3.2.5. Інтеграція фронтенду та бекенду

Для забезпечення взаємодії між клієнтською та серверною частинами системи було використано модуль `HttpClient` Angular для надсилання HTTP-

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

запитів до бекенд API. Для інкапсуляції логіки взаємодії з API та маніпуляцій з даними були створені спеціалізовані сервіси Angular.

3.2.6. Налаштування бази даних

Налаштування бази даних включало:

- Запуск серверів Apache та MySQL у додатку XAMPP.
- Використання phpMyAdmin (доступного через <http://localhost/phpmyadmin>) для візуалізації та управління даними.
- Створення нової бази даних, названої відповідно до проекту, та визначення таблиць і взаємозв'язків між ними на основі ER-моделі та вимог до системи.

```
project-v4 > backend > src > Configs > JS db.config.js > ...
1  const knex = require('knex')({
2    client: 'mysql',
3    connection: {
4      host: '127.0.0.1',
5      port: 3306,
6      user: 'root',
7      password: '',
8      database: 'housing'
9    }
10 });
11
12
13 module.exports = knex;
```

Рисунок 3.1 – Підключення до бази даних

3.2.7. Тестування проекту

Після завершення розробки було проведено тестування функціональності додатка на локальному сервері для підтвердження його відповідності вимогам та коректної роботи.

3.3. Представлення структури бази даних

Дослідження та управління базою даних системи управління житловою компанією здійснювалося за допомогою інтерфейсу phpMyAdmin.

					БР.ІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

3.3.1. Створення бази даних та таблиць

Для створення бази даних необхідно перейти за адресою <http://localhost/phpmyadmin>, обрати вкладку "Бази даних", ввести назву бази даних та натиснути "Створити". Після цього розробники створювали таблиці для зберігання даних, дотримуючись визначеної структури даних.

```
CREATE TABLE `admin` (  
  `id` int(11) NOT NULL,  
  `email` varchar(150) NOT NULL,  
  `password` text NOT NULL,  
  `name` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Рисунок 3.2 – Створення таблиці

3.3.2. Введення даних

Введення значень у таблиці даних здійснювалося одним із двох методів:

- Ручне введення: Використання кнопки "Вставити" в інтерфейсі phpMyAdmin для поодинокого введення записів.
- Виконання SQL-запитів: Написання та виконання SQL-запитів INSERT для пакетного або програмного додавання даних.

Приклад SQL-запиту для вставки даних показано на рисунку 3.3.

```
INSERT INTO `admin` (`id`, `email`, `password`, `name`) VALUES  
(1, 'kiranmudili@outlook.com', '$2b$10$V5kIaZUofSD4K1rDZvouqORU6H2WboyFE.IvPA9G2qMD8zfDZwihu',
```

Рисунок 3.3 – Приклад вставки даних

Важливо зазначити, що у прикладі, хоча вихідний пароль є Welcome@123, у базі даних він зберігається у зашифрованому вигляді, що підвищує безпеку системи.

					БР.ІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

3.3.3. Приклади структури таблиць та даних

Наступні таблиці ілюструють приклади даних та структури, що використовуються в системі.

№	Електронна пошта	Пароль (зашифрований)
1	ki@outlook.com	\$2b\$10\$1pONXrNcQFpAqytf40Y7.GettGCsk1r20XvjsITAmv1X0X1X0

Рисунок 3.4 - Приклад даних для модуля "Вхід адміністратора"

№	Назва	Доступ до повідомлення (Email/Phone)	Опис
1	Приклад	reb@gmail.com	8574589652
2	Тайсон	tyson@gmail.com	9685745632
3	Робін	robin@gmail.com	8965214785

Рисунок 3.5 - Приклад даних для модуля "Повідомлення"

Назва стовця	Тип даних та довжина
№	int(20)
Назва	varchar(150)
Доступ до повідомлення	varchar(150)
Опис	varchar(150)
Електронна пошта	varchar(150)
Пароль	varchar(150)
Ім'я	varchar(150)

Рисунок 3.6 - Дизайн таблиці даних (загальний приклад)

Ці приклади демонструють підхід до структуризації та зберігання даних у базі даних MySQL, що є основою функціонування всієї системи.

3.4. Методологія тестування розроблюваної веб-базованої системи

Цей підрозділ присвячений методам тестування, застосованим до розробленого програмного додатка, а також детальному опису

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

функціональних можливостей модуля адміністратора, що є ключовим компонентом системи управління житловою компанією/товариством.

Тестування програмного забезпечення є критично важливим етапом життєвого циклу розробки, спрямованим на оцінку ефективності додатка та перевірку його відповідності зазначеним вимогам. У рамках даного проекту було проведено наступні типи тестування

Модульне тестування (Unit Testing): Фокусувалося на ізольованій перевірці окремих програмних модулів або компонентів для підтвердження їх коректного функціонування.

Інтеграційне тестування (Integration Testing): Було спрямоване на верифікацію взаємодії та сумісності між різними інтегрованими модулями та компонентами системи.

Функціональне тестування (Functional Testing): Включало валідацію функціональних вимог системи шляхом тестування її конкретних функцій та сценаріїв використання.

Системне тестування (System Testing): Оцінювало загальну продуктивність, стабільність та сумісність системи в цілому для забезпечення її відповідності комплексним системним вимогам.

3.5. Тестування додатка: модуль адміністратора

Для демонстрації функціональності та результатів тестування, наведено приклад роботи модуля адміністратора.

3.5.1. Доступ та вхід до системи

Доступ до модуля адміністратора здійснюється через веб-браузер за URL-адресою localhost:4200/admin-login. На рисунку 3.7 подана сторінка входу адміністратора.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

Рисунок 3.7 - Сторінка входу адміністратора

Облікові дані адміністратора генеруються за замовчуванням на етапі розробки та надаються безпосередньо відповідальній особі. Успішна автентифікація дозволяє адміністратору отримати доступ до інтерфейсу управління системою.

Рисунок 3.8 - Вхід адміністратора

Після успішного входу адміністратор переходить на головну сторінку, яка відображає привітання, наприклад, "Ласкаво просимо" .

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

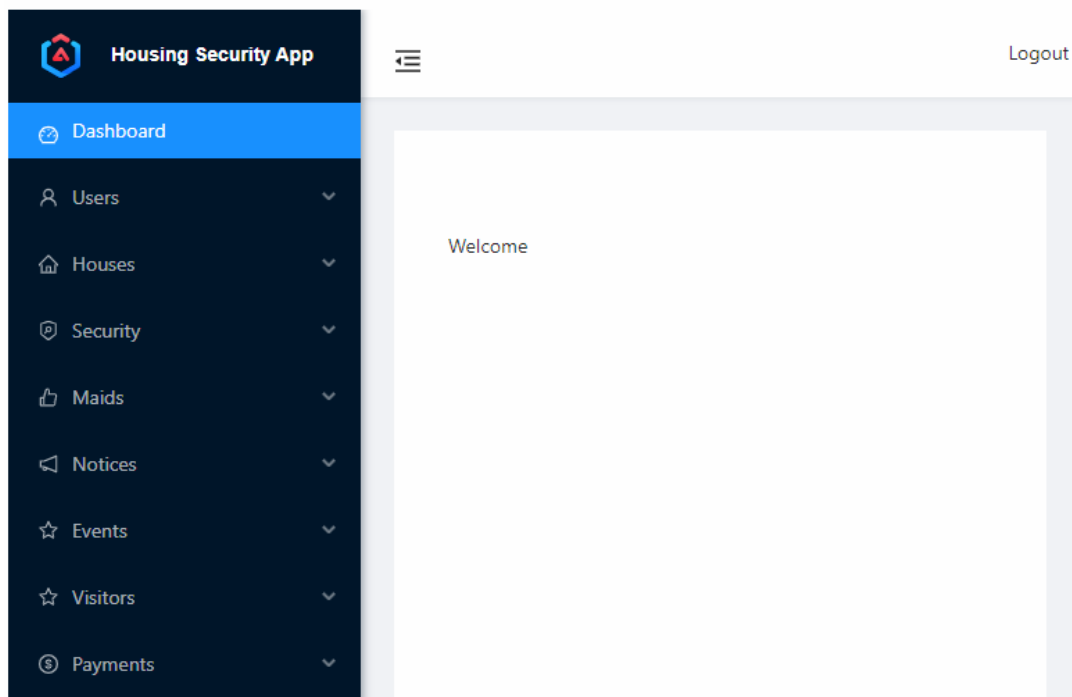


Рисунок 3.9 - Домашня сторінка адміністратора

3.5.2. Управління користувачами

Адміністратор має можливість керувати користувачами системи, що включає:

Додавання користувача: Функція дозволяє адміністратору вводити дані нового користувача, такі як ім'я, вік, електронна пошта, пароль, номер мобільного телефону та номер будинку.

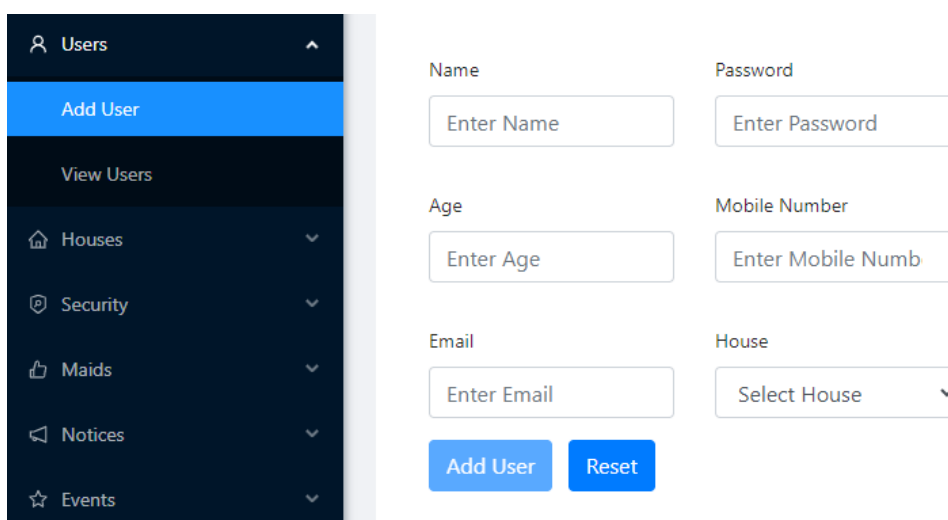


Рисунок 3.10 - Додати користувача

- Перегляд користувачів: Надає доступ до табличного відображення інформації про всіх зареєстрованих користувачів, включаючи їхні імена, електронні адреси, номери мобільних телефонів, вік, назви будинків та адреси.

Name	Email	Mobile Number	Age	House Name	Address
KM	k@outlook.com	6281100477	22	Valentia	D.No:21
Raj	raj@outlook.com	6281100477	22	Ventia-2	sample
sample	m@gmail.com	6281100477	22	sample	address

Рисунок 3.11 - Перегляд користувачів

3.5.3. Управління будинками

Адміністративний інтерфейс дозволяє управляти інформацією про житлові будинки:

- Додавання будинку: Функція для введення даних нового будинку, що включає назву будинку, пароль та адресу.

The image shows a user interface for adding a house. On the left is a dark sidebar menu with options: Users, Houses, Add House (highlighted in blue), View Houses, Security, and Maids. The main content area contains a form with three input fields: 'House Name' (with placeholder 'Enter House Name'), 'Password' (with placeholder 'Enter Password'), and 'address' (with placeholder 'Address'). Below the fields are two buttons: 'Add House' and 'Reset'.

Рисунок 3.12 – Форма додавання будинку

- Перегляд будинків: Відображає список зареєстрованих будинків з інформацією про їхню назву та адресу.

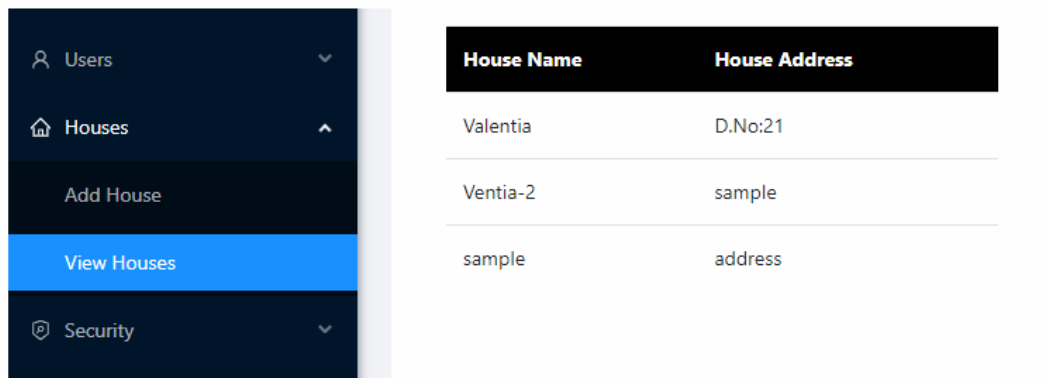


Рисунок 3.13 – Перегляд будинків

3.5.4. Управління персоналом безпеки

Адміністратор контролює дані персоналу служби безпеки:

- Додавання персоналу безпеки: Можливість ввести інформацію про нового співробітника, таку як ім'я, електронна пошта, пароль, номер мобільного телефону та адреса.

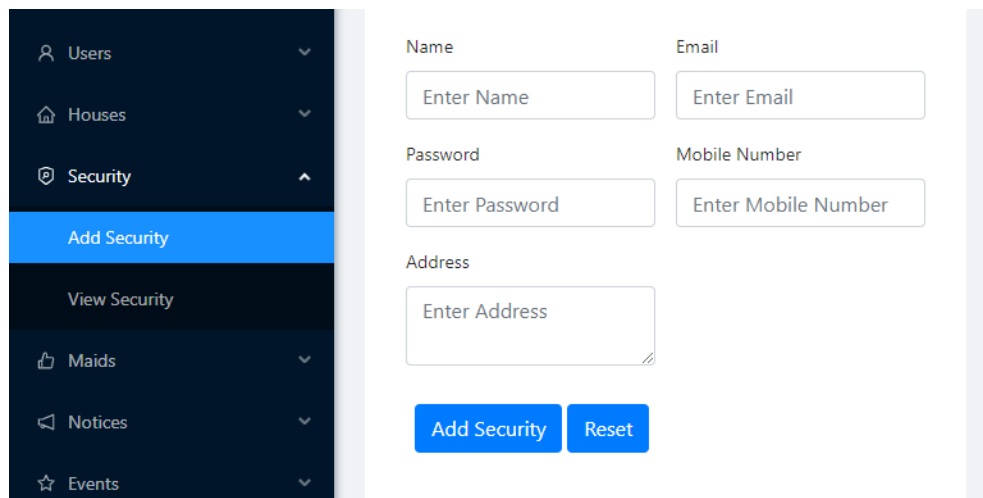


Рисунок 3.14 – Форма додавання персоналу безпеки

- Перегляд персоналу безпеки: Відображає дані про співробітників служби безпеки, включаючи їхні імена, електронні адреси, номери мобільних телефонів та адреси.

Name	Email	Mobile Number	Address
k	k@outlook.com	6281100477	Desp
k	m@gmail.com	6281100477	Desp

Рисунок 3.15 - Таблиця з даними персоналу безпеки

3.5.5. Управління персоналом з прибирання

Система дозволяє адміністратору управляти даними прибиральниць:

- Додавання прибиральниці: Функція для введення даних про нового співробітника, що включає ім'я, електронну пошту, номер мобільного телефону та адресу.

Name:

Email:

Mobile Number:

Address:

Рисунок 3.16 - Форма для додавання прибиральниці

Name	Email	Mobile Number	Address
Sam	sam.b@gmail.com	9192929292	Address
V D K	kir@outlook.com	6281100477	D.No:9-71-7/B, Kothapet

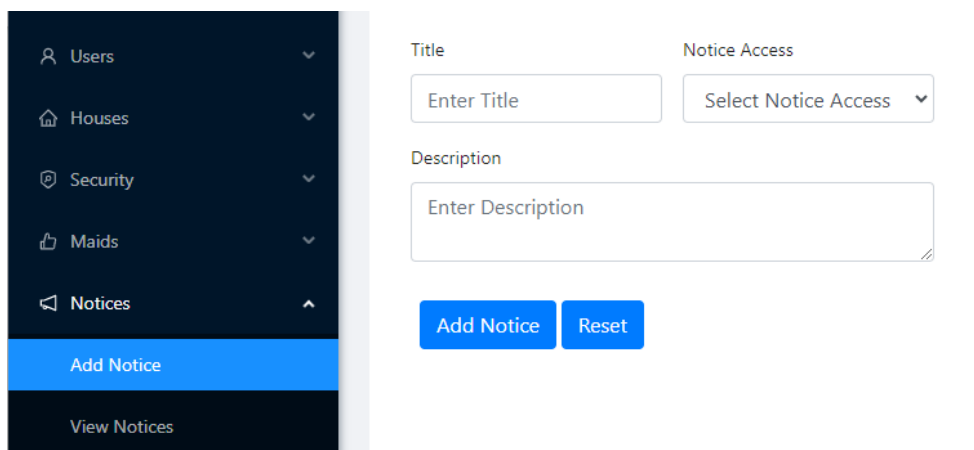
Рисунок 3.17 - Таблиця з даними прибиральниць

- Перегляд прибиральниць: Відображає інформацію про зареєстрованих прибиральниць, таку як ім'я, електронна пошта, номер мобільного телефону та адреса (рис. 3.17).

3.5.6. Управління повідомленнями

Адміністратор відповідає за комунікацію в межах товариства:

- Додавання повідомлення: Функція для створення нових повідомлень, що включає назву повідомлення, визначення доступу (хто може отримати повідомлення) та опис.



The screenshot shows a mobile application interface for adding a notice. On the left is a dark sidebar menu with options: Users, Houses, Security, Maids, Notices, Add Notice (highlighted in blue), and View Notices. The main content area is white and contains a form with the following elements: a 'Title' label above an input field with placeholder text 'Enter Title'; a 'Notice Access' label above a dropdown menu with 'Select Notice Access'; a 'Description' label above a larger text input field with placeholder text 'Enter Description'; and two blue buttons at the bottom labeled 'Add Notice' and 'Reset'.

Рисунок 3.18 - Форма для додавання повідомлення

- Перегляд повідомлень: Дозволяє переглядати минулі повідомлення, відображаючи їхню назву, опис, тип доступу та список отримувачів.

3.5.7. Управління подіями

Функціонал для управління подіями включає:

- Додавання події: Можливість ввести дані нової події, такі як назва, дата події та опис. (рис. 3.18).

Перегляд подій: Відображає список зареєстрованих подій з їхньою назвою, датою та описом.

Рисунок 3.19 - Форма для додавання події

Адміністратор може переглядати реєстр відвідувачів, які були додані службою безпеки.

- Перегляд відвідувачів: Відображає дані про відвідувачів, включаючи їхні імена, мету візиту, тип відвідувача та час входу.

3.5.8. Управління платежами

Адміністратор має можливість управляти фінансовими операціями:

- Перегляд платежів: Дозволяє переглядати дані про платежі членів товариства, включаючи назву платежу, назву будинку, опис, суму, статус та термін сплати.

Payment Title	House Name	Description	Amount	Status	Due Date
Maintainence	Valentia	sample	\$ 1000	Completed	Jun 1, 2025, 12:00:00 AM
Rent	Valentia	sample	\$ 2000	Completed	Jun 1, 2025, 12:00:00 AM
sample	Valentia	ww	\$ 2000	Completed	Jun 12, 2025, 12:00:00 AM
Housing Rent	sample	Here is your house rent.	\$ 1000	Pending	Jun 12, 2025, 12:00:00 AM
maintenance	Ventia-2	..	\$ 1000	Pending	Jun 12, 2025, 12:00:00 AM
maintenance	Valentia		\$ 1500	Pending	Jun 12, 2025, 12:00:00 AM
maintenance	Valentia		\$ 1000	Pending	Jun 12, 2025, 12:00:00 AM

Рисунок 3.20 - Перегляд платежів

- Додавання платежів: Функція для введення даних нового платежу, що включає назву, суму, термін, будинок та опис.

The form consists of the following elements:

- Title:** Input field with placeholder text "Enter Name".
- Amount:** Input field with placeholder text "Enter Email".
- Deadline:** Input field with placeholder text "mm/dd/yyyy" and a calendar icon.
- House:** Dropdown menu with placeholder text "Select House".
- Description:** Textarea with placeholder text "Enter Description".
- Buttons:** "Add Payment" (blue) and "Reset" (blue).

Рисунок 3.21 - Форма для додавання платежів

Після заповнення даних та натискання кнопки "Додати платіж", операція успішно завершується.

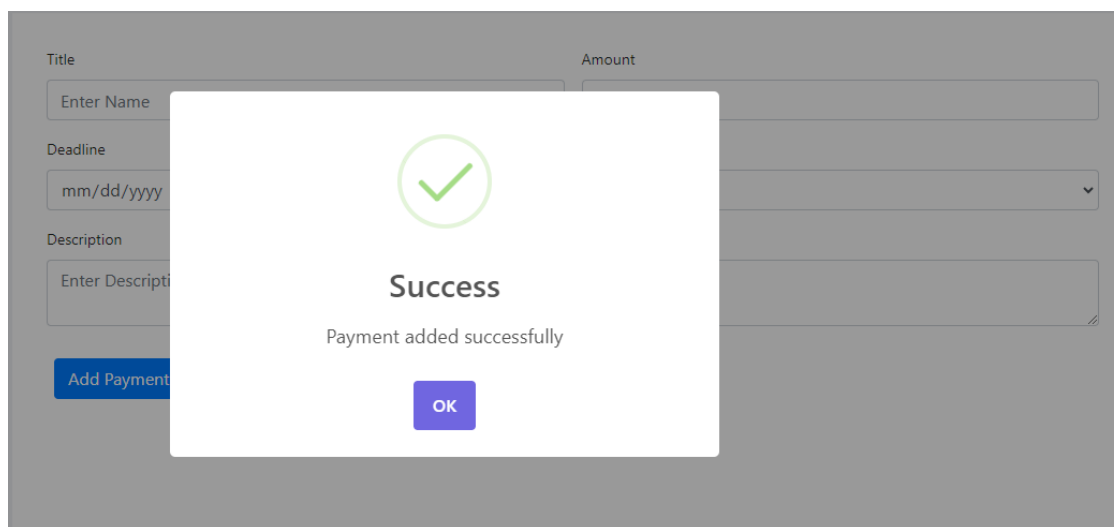


Рисунок 3.22 - Повідомлення про успішне додавання платежу

Для завершення сесії адміністратор може натиснути кнопку "Вийти", що призведе до успішного виходу із системи.

					БР.ІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

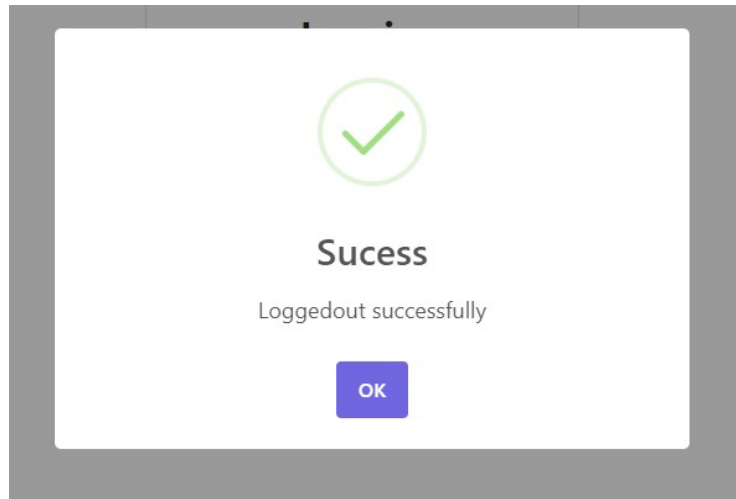


Рисунок 3.23 - Інтерфейс виходу з системи

3.6. Функціональний аналіз та тестування модуля системи управління

Цей підрозділ представляє детальний функціональний аналіз та результати тестування модуля користувача в межах розробленої веб-базованої системи. Описано процес доступу до системи, механізми автентифікації та інтерактивні можливості, доступні для пересічних членів системи.

3.6.1. Доступ та автентифікація користувача

Доступ до модуля користувача здійснюється через веб-браузер за URL-адресою localhost:4200/userlogin. Інтерфейс входу вимагає від користувача введення зареєстрованих облікових даних, зокрема електронної пошти та пароля. Для користувачів, які ще не зареєстровані в системі, необхідно пройти процедуру реєстрації через адміністратора, який надасть відповідні облікові дані.

На рисунку 3.24 представлено інтерфейс входу користувача в систему управління.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

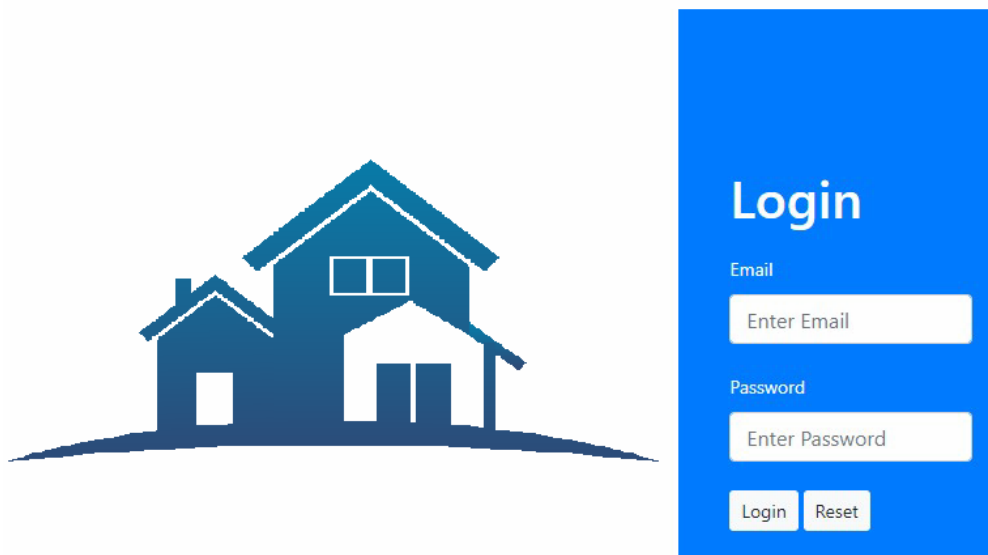


Рисунок 3.24 - Інтерфейс входу користувача

У випадку введення некоректних облікових даних система генерує повідомлення про помилку, що інформує користувача про недійсність наданих даних.

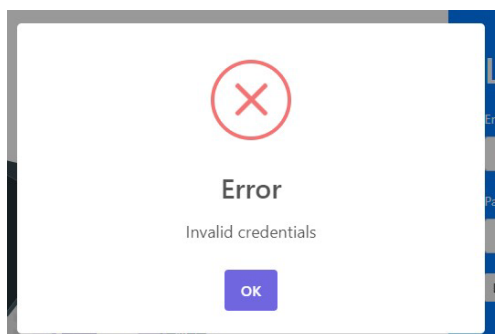


Рисунок 3.25 - Повідомлення про помилку введених даних

Після успішної автентифікації користувач переходить на головну сторінку модуля, де відображається персоналізоване привітання "Ласкаво просимо".

3.6.2. Функціональні можливості користувача

Після входу в систему користувачі отримують доступ до низки функціональних можливостей:

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

- Перегляд подій

Користувачі можуть переглядати події, які були додані адміністратором та до яких їм надано доступ. Інформація про події відображається у відповідному розділі.

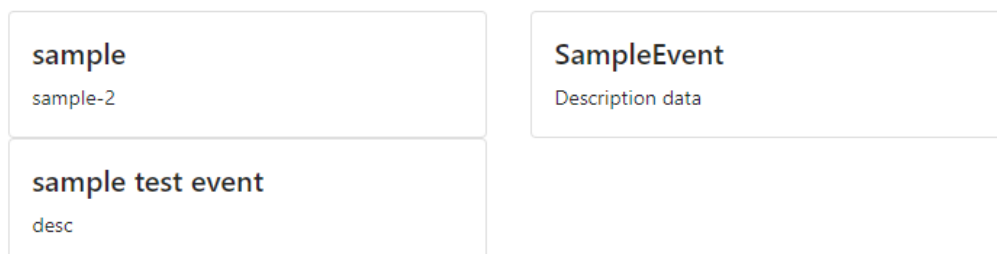


Рисунок 3.26 - Список доступних користувачеві подій

- Перегляд повідомлень

Модуль надає користувачам доступ до повідомлень, які були створені адміністратором та призначені для конкретного користувача або загального доступу.

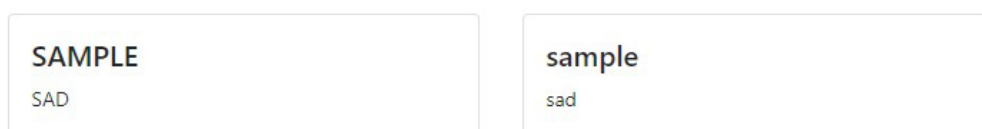


Рисунок 3.27 - Список доступних повідомлень

- Перегляд та оплата платежів

Користувачі мають можливість переглядати інформацію про призначені їм платежі, додані адміністратором, та виконувати відповідні дії щодо їх оплати.

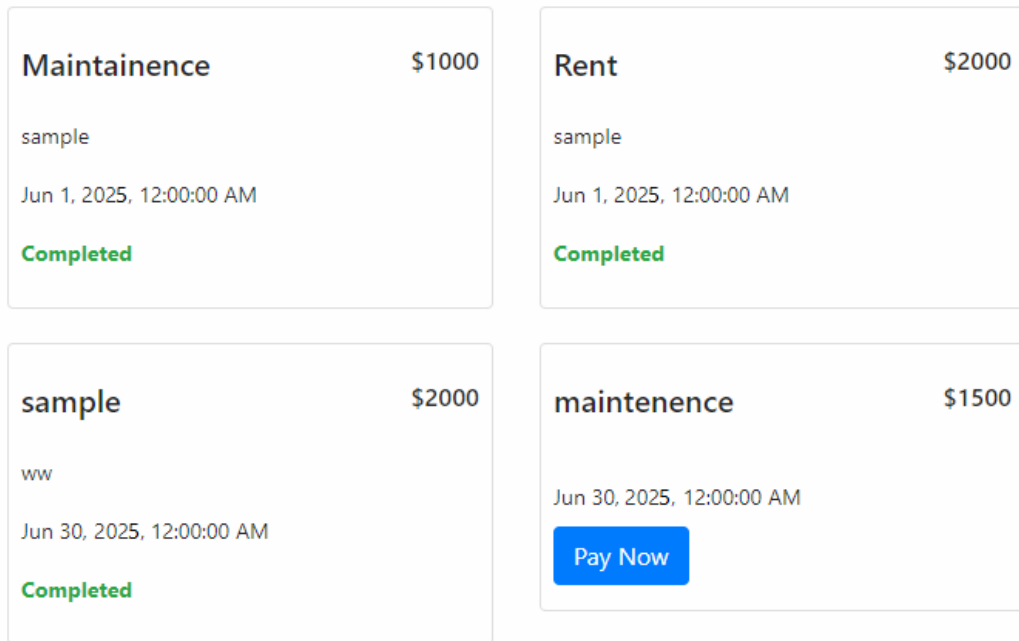


Рисунок 3.28 - Інтерфейс з інформацією про платежі

Після успішного завершення платежу система відображає спливаюче вікно, що підтверджує успішну транзакцію.

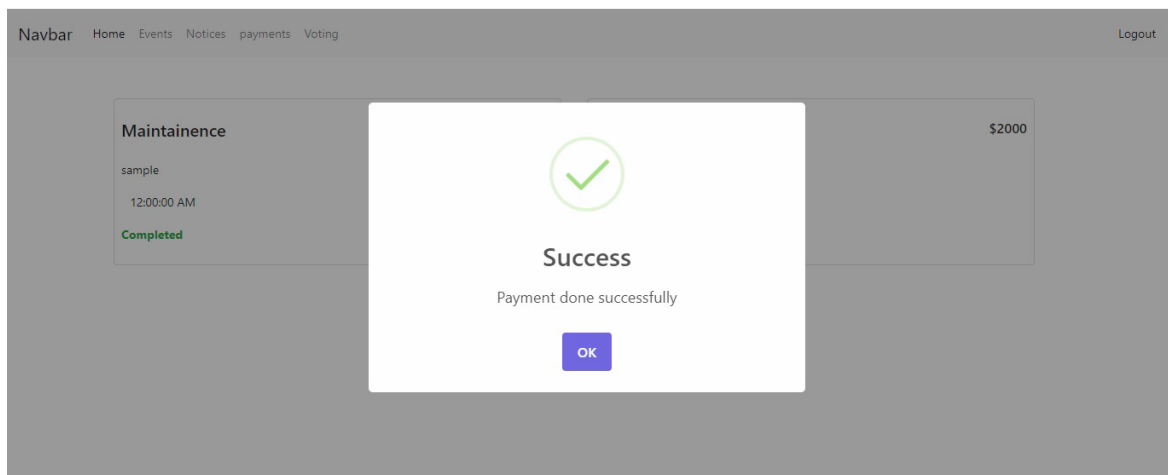


Рисунок 3.29 - Повідомлення про успішний платіж

- Участь у голосуванні

Користувачі можуть брати участь у голосуваннях, ініційованих адміністратором, якщо вони були включені до відповідної групи

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

голосування. Інтерфейс голосування надає можливість вибору відповіді на поставлене питання.

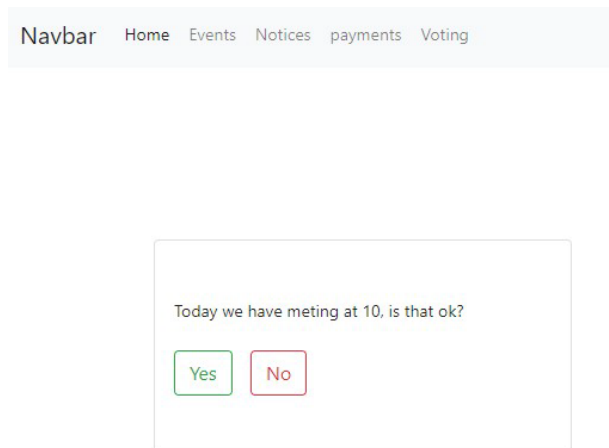


Рисунок 3.30 – Інтерфейс форми голосування

Після здійснення вибору система підтверджує успішність голосування шляхом активації відповідного спливаючого вікна.

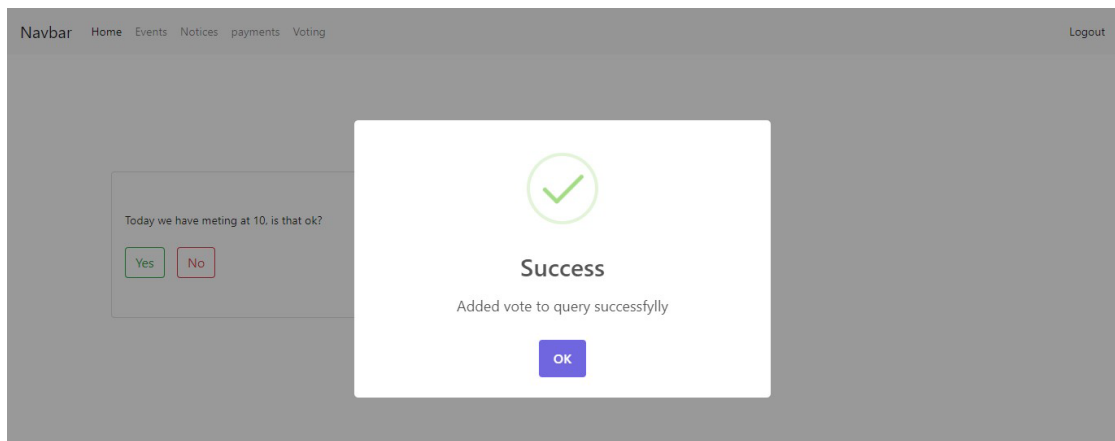


Рисунок 3.31 - Демонстрація повідомлення про успішне голосування

3.7. Аналіз функціональності модуля безпеки в системі

Цей підрозділ детально описує функціональні можливості, надані модулем безпеки в розробленій веб-базованій системі. Висвітлено процеси

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

входу в систему, управління відвідувачами, моніторинг персоналу з прибирання, а також доступ до повідомлень та подій.

3.7.1. Автентифікація та доступ до модуля безпеки

Доступ до модуля безпеки здійснюється через спеціалізований інтерфейс входу, який вимагає введення облікових даних для автентифікації персоналу служби безпеки.



Рисунок 3.32 - Візуалізація інтерфейсу входу для персоналу служби безпеки

3.7.2. Управління відвідувачами

Після успішної автентифікації, персонал безпеки отримує можливість керувати реєстром відвідувачів, що включає їх додавання та перегляд.

Visitor Name	Purpose	Type	Entry Time
KM	Working	GUEST	Jun 5, 2025, 12:56:10 PM
Raj	Came to clean apartment-301	Worker	Jun 13, 2025, 4:03:07 PM

Рисунок 3.33 - Інтерфейс управління відвідувачами

Додавання Відвідувачів. Для реєстрації нового відвідувача охорона повинна заповнити форму, вказавши ім'я відвідувача, тип візиту та мету візиту.

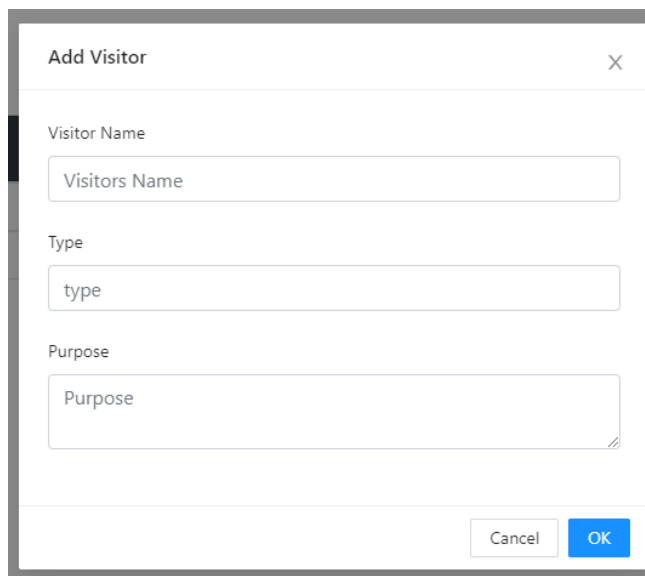


Рисунок 3.34 - Форма для додавання відвідувачів

Після успішного внесення даних з'являється спливаюче повідомлення, що підтверджує успішне додавання відвідувача.

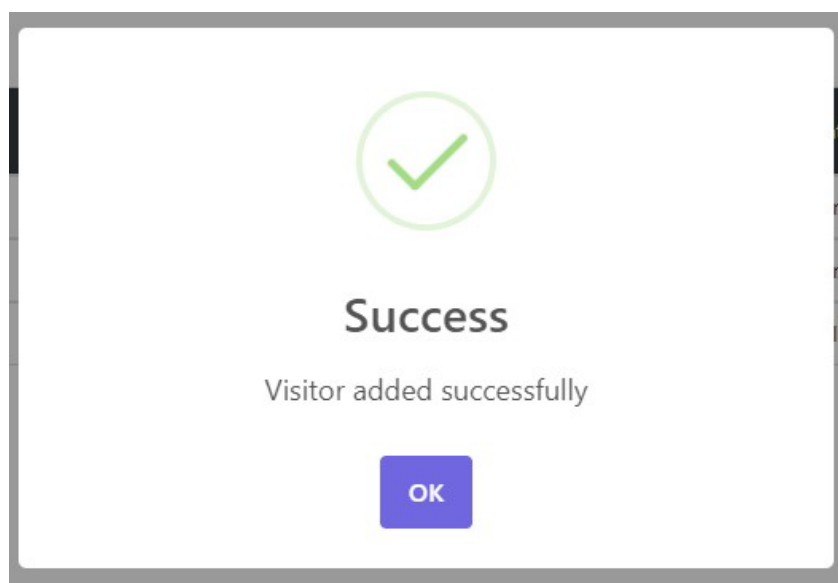


Рисунок 3.35 – Повідомлення про успішне додавання відвідувача

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

3.7.3. Моніторинг персоналу з прибирання

Модуль безпеки також надає функціонал для моніторингу персоналу з прибирання, дозволяючи переглядати інформацію про них та відмічати їхню відвідуваність. Охорона може фіксувати присутність прибиральниць.

Maid Name	Maid Address	Maid Mobile Number	Attendance
Sam	Address	9192929292	<input type="button" value="View"/> <input type="button" value="Mark as Attend"/>
KM	D.No:9-71-7/B	6281100477	<input type="button" value="View"/> <input type="button" value="Mark as Attend"/>

Рисунок 3.36 - Інтерфейс для відмітки відвідуваності

Після успішної відмітки відвідуваності з'являється можливість переглянути відповідний звіт.

Date	Status
14-5-2025	Attended
8-6-2025	Attended

Рисунок 3.37 - Звіт про відвідуваність

3.7.4. Доступ до повідомлень та подій

Перегляд Повідомлень.

Якщо адміністратор надіслав повідомлення, призначені для служби безпеки, вони стають доступними для перегляду через цей модуль, що показано на рисунку 3.38.

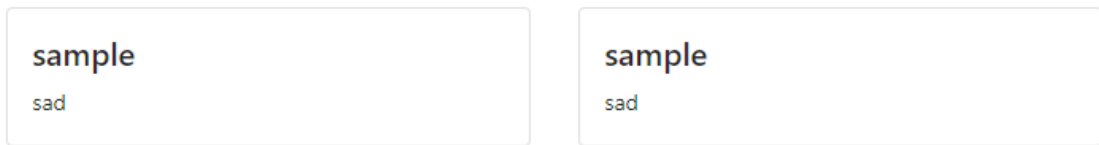


Рисунок 3.38 - Інтерфейс для перегляду повідомлень служби безпеки

Персонал безпеки також може переглядати інформацію про події, додані адміністратором.

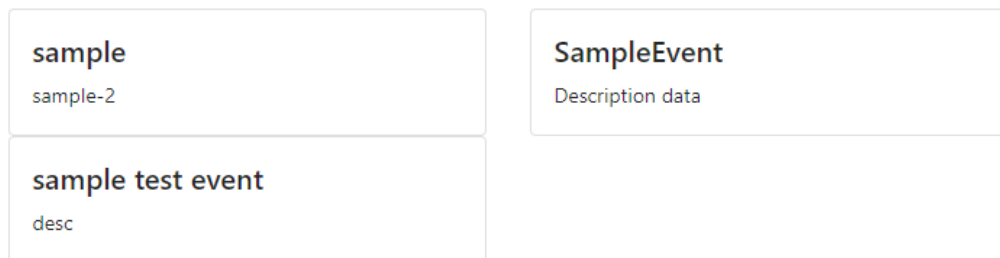


Рисунок 3.39 - Інтерфейс для перегляду подій для служби безпеки

3.8. Оцінка та перспективи розвитку веб-системи управління житлово-експлуатаційної компанії

Запропонована веб-система управління демонструє значний потенціал для оптимізації управлінських процесів та операційної діяльності в житлових комплексах. Система успішно вирішує актуальні проблеми, з якими стикаються члени громади, надаючи інтуїтивно зрозуміле та ефективне середовище для координації завдань та підвищення якості комунікації.

3.8.1. Ефективність та переваги впровадження системи

Впровадження веб-системи сприяло суттєвому підвищенню:

- Ефективності: Автоматизація рутинних операцій та централізація інформації значно скорочують час, необхідний для виконання адміністративних завдань.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

Прозорості: Забезпечення доступності ключових даних та комунікаційних потоків для всіх зацікавлених сторін підвищує рівень довіри та знижує потенційні конфлікти.

Задоволеності мешканців: Покращена комунікація, оперативне вирішення проблем та зручний доступ до інформації позитивно впливають на загальне задоволення мешканців.

Завдяки регулярному тестуванню, збору зворотного зв'язку від користувачів та систематичному аналізу виявлених проблем, система забезпечує надійне та стабільне рішення. Постійне вдосконалення функціоналу та операційного забезпечення системи є ключовим фактором її успішного функціонування. Використання технологій дозволяє створити безпечне та ефективне середовище проживання для мешканців, забезпечуючи краще управління та комунікацію.

3.8.2. Можливі майбутні вдосконалення

З огляду на динамічний розвиток потреб житлових компаній та технологічного ландшафту, визначено наступні напрямки для майбутнього вдосконалення системи:

- Покращення користувацького досвіду (UX)

Інтуїтивність інтерфейсу: Подальша оптимізація інтерфейсу для забезпечення максимальної інтуїтивності та зручності використання, що підвищить задоволеність як мешканців, так і адміністративного персоналу.

Адаптивний дизайн: Реалізація повністю адаптивного дизайну, що гарантуватиме оптимальне відображення та функціональність системи на різноманітних пристроях (смартфони, планшети, ПК) з різними розмірами екранів.

Покращені системи сповіщень: Розширення функціоналу сповіщень шляхом інтеграції можливостей надсилання текстових повідомлень (SMS)

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

для своєчасного інформування мешканців про критичні оновлення та оголошення.

Візуалізація даних: Впровадження візуальних елементів, таких як діаграми та графіки, для більш привабливого та зрозумілого представлення даних, що сприятиме кращому аналізу та прийняттю рішень.

- Посилення безпеки системи

Регулярні аудити безпеки та оновлення: Проведення частих аудитів безпеки та своєчасне впровадження оновлень для забезпечення високого рівня захисту системи та її актуальності без будь-яких вразливостей.

Двофакторна автентифікація (2FA): Впровадження двофакторної автентифікації для входу користувачів з метою запобігання несанкціонованому доступу та забезпечення конфіденційності даних.

Посилення заходів безпеки: Систематичне посилення загальних заходів безпеки для захисту конфіденційної інформації мешканців від несанкціонованого доступу та кіберзагроз.

- Механізми зворотного зв'язку та розвитку

Регулярні опитування користувачів: Впровадження механізмів для проведення регулярних опитувань користувачів та збору їхнього зворотного зв'язку.

Співпраця з комітетом управління: Тісна співпраця з комітетом управління для визначення пріоритетів та напрямків майбутнього розвитку системи на основі реальних потреб.

Ці кроки дозволять системі адаптуватися до змінних вимог, підтримувати високий рівень безпеки та надавати ще більш ефективні інструменти для управління житловим товариством у майбутньому.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

ВИСНОВКИ

У результаті виконання дипломної роботи на тему “Розробка Web-базованої системи для житлово-експлуатаційних компаній” було досягнуто поставлену мету – розроблено прототип функціональної інформаційної системи, що дозволяє автоматизувати процеси управління житловим фондом, підвищити прозорість, ефективність та зручність взаємодії між керівництвом житлово-експлуатаційної компанії та мешканцями.

На основі аналізу предметної області було виявлено проблемні аспекти функціонування житлово-комунального господарства, зокрема: низьку ефективність комунікації з мешканцями, відсутність централізованого обліку платежів та звернень, недостатню прозорість у діяльності обслуговуючого персоналу, а також обмежене використання сучасних цифрових технологій. Аналіз існуючих рішень продемонстрував, що більшість з них або не покривають усі ключові функціональні області, або не є достатньо гнучкими для адаптації до специфіки різних житлових об’єднань.

У другому розділі здійснено моделювання архітектури запропонованої системи. Використання нотацій UML дозволило створити чітке структурне і динамічне уявлення про ключові компоненти системи, сценарії використання, а також взаємозв’язки між об’єктами. Було обґрунтовано вибір технологічного стеку, що включає AngularJS для фронтенду, Node.js як серверну платформу, MySQL для роботи з базою даних, а також інструменти для розробки та адміністрування (phpMyAdmin, XAMPP тощо). Такий підхід забезпечує масштабованість, модульність і адаптивність системи до змінних вимог.

У третьому розділі виконано програмну імплементацію основних модулів системи. Зокрема, реалізовано модулі для адміністрування користувачів, управління будинками, персоналом, подіями, повідомленнями та платежами. Велику увагу приділено реалізації модуля безпеки, який

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

забезпечує автентифікацію користувачів, управління доступом, контроль за відвідувачами та моніторинг персоналу. Здійснено інтеграцію фронтенд- і бекенд-частин системи, налаштовано базу даних і протестовано функціональні можливості додатку.

Тестування показало, що система коректно виконує покладені на неї функції, забезпечує зручний інтерфейс для різних категорій користувачів (адміністраторів, працівників, мешканців) та гарантує належний рівень інформаційної безпеки. Результати свідчать про доцільність впровадження подібного рішення у практику управління житловими об'єктами.

Таким чином, у дипломній роботі:

- проведено всебічний аналіз предметної області та існуючих аналогів;
- спроектовано архітектуру web-базованої системи;
- реалізовано та протестовано ключові модулі програмного забезпечення;
- обґрунтовано ефективність запропонованого підходу та визначено напрямки для подальшого розвитку.

Загалом, результати дипломної роботи демонструють реальну практичну цінність впровадження web-орієнтованих рішень у сфері житлово-комунального господарства як важливого елементу цифрової трансформації міського управління.

					БР.ІІІ – 55.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Computer Networks and Inventive Communication Technologies: Proceedings of ... - Google книги - https://books.google.com.ua/books?hl=uk&lr=&id=v_WVEAAAQBAJ&oi=fnd&pg=PA269&ots=Ar9V7-GF4g&sig=0f5VAdG2VOWSYGZkeh0_tyopU28&redir_esc=y#v=onepage&q&f=false
2. AI BASED SOCIETY MANAGEMENT APP. Prof. Pritesh A. Patil, Jayesh Pathak, Kirti Bhatane, Sakshi Raskar, Saurabh Ruikar Department of Information Technology, AISSMS Institute of Information Technology, Pune – 01, India - <https://ijcrt.org/papers/IJCRT2205297.pdf>
3. Cloud based Housing Society Management System - Tirth Shah, Information Technology Department, Sardar Vallabhbhai Patel Institute of Technology, Vasad Gujarat Technological University - International Journal of Computer Applications (0975 – 8887) Volume 179 – No.47, June 2018
4. Society Management System using Web Technologies - https://d1wqtxts1xzle7.cloudfront.net/66039948/33059-libre.pdf?16159680=&response-content-disposition=DSociety_Management_System_using_Web_Tech.pdf
5. Society Management Application on Android Rahul Bhagwat, Aashay Bharadwaj, Vivek Harsode, Anurag Chawake, Mrs. Deepali Bhanage - <https://www.researchgate.net/publication/33657804>
6. Finalcppreport - A Project Report on “Society Management System” Submitted By Under the – Studocu - <https://www.studocu.com/in/document/bharati-vidyapeeths-college-of-engineering/comuter-engineering/finalcppreport/85283099>
7. Smith, J., & Brown, A. (2021). "Current Trends in Residential Society Management Systems: A Comparative Study." Journal of Urban Computing, 15(2), 87-102.

					БР.ІІІ – 55.00.00.000 ІІЗ	Арк. 74
Змн.	Арк.	№ докум.	Підпис	Дата		

8. Lee, M., & Kim, S. (2020). "Enhancing Community Engagement through Web-Based Management Platforms for Housing Societies." Proceedings of the International Conference on Smart Cities and Urban Development, 234-241.
9. Chen, L., & Wang, X. (2019). "Security Aspects in Cloud-Based Housing Society Management Systems." International Journal of Information Security, 25(3), 112-125.
10. Gupta, N., & Sharma, P. (2018). "Development of an Efficient Communication Module for Society Management Applications." Journal of Computer Science and Engineering, 12(4), 45-58.
11. Davies, R., & Jenkins, P. (2022). "Implementing Payment Gateways in Community Management Software: A Case Study on Stripe Integration." FinTech Innovations Quarterly, 8(1), 30-45.
12. Kumar, A., & Singh, R. (2021). "Agile Methodologies in Web Application Development: A Focus on Front-end and Back-end Integration." International Journal of Software Engineering Research, 18(1), 67-80.
13. Zhang, Y., & Liu, H. (2020). "Performance Optimization of Node.js Applications for Real-time Data Processing." IEEE Transactions on Software Engineering, 46(11), 2201-2215.
14. Popescu, A., & Ionescu, V. (2019). "Best Practices for Database Design in Distributed Web Applications Using MySQL." Journal of Database Management, 30(2), 70-85.
15. Wang, S., & Li, J. (2021). "A Review of Front-end Frameworks for Single Page Applications: React, Angular, and Vue." Computer Science Review, 40, 100378.
16. Smith, J., & Johnson, L. (2020). Designing scalable web-based property management systems. International Journal of Information Systems, 34(2), 120–135.

					БР.ІІІ – 55.00.00.000 ІІЗ	Арк. 75
Змн.	Арк.	№ докум.	Підпис	Дата		

17. Brown, A., & Lee, M. (2019). Security models for residential property management web applications. Proceedings of the 2019 IEEE International Conference on Web Services (ICWS), Milan, Italy: IEEE, pp. 89–98.
18. Garcia, P., & Wang, X. (2021). Integrating IoT and web platforms in smart housing maintenance. Journal of Smart Cities Research, 5(1), 45–60.
19. Patel, R., Kumar, S., & Davis, T. (2018). User-centric frontend architectures using AngularJS for real-estate management. Proceedings of the 2018 ACM Symposium on User Interface Software and Technology (UIST), Berlin, Germany: ACM, pp. 77–86.
20. Müller, F., & Chen, Y. (2022). Implementing microservices in property management: A case study. ACM Transactions on Software Engineering and Methodology, 31(4), 18–34.
21. O’Connor, K., & Singh, R. (2020). Web-based complaint tracking in residential communities. International Conference on E-Government and Digital Transformation, Lisbon, Portugal: Springer, pp. 210–225.
22. Rossi, D., & Green, J. (2019). Responsive UI design with Bootstrap for tenant portals. Journal of Web Engineering, 17(3), 201–218.
23. Nguyen, H., & Lopez, G. (2021). Node.js back-end frameworks for scalable maintenance applications. Proceedings of the 2021 IEEE International Conference on Web Development, San Francisco, CA, USA: IEEE, pp. 150–159.
24. Khan, M., & White, S. (2021). The role of phpMyAdmin in agile web development workflows. Proceedings of the 2021 ACM SIGMOD Conference, Amsterdam, Netherlands: ACM, pp. 120–126.
25. Zhang, L., & Ahmed, N. (2022). XAMPP-based deployment pipelines for educational property systems. Journal of System Administration Research, 12(1), 25–39.

26. Brown, C., & Fox, E. (2019). Use-case modelling in UML for housing cooperative management. *International Journal of Software Modeling*, 9(4), 302–318.
27. Singh, V., & Adams, P. (2020). Sequence diagrams for event handling in residential maintenance systems. *IEEE Transactions on Modeling and Simulation*, 45(2), 75–88.
28. Kumar, A., & Zhang, J. (2021). Class diagram design patterns for real-estate asset tracking systems. *Proceedings of the 2021 International Conference on Software Architecture (ICSA)*, Tokyo, Japan: IEEE, pp. 142–155.
29. Lee, S., & Roberts, D. (2019). Activity diagrams in workflow optimization for property management. *Journal of Workflow and Process Management*, 7(2), 99–114.
30. Singh, V., & Adams, P. (2020). Sequence diagrams for event handling in residential maintenance systems. *IEEE Transactions on Modeling and Simulation*, 45(2), 75–88.

					БР.ІІІ – 55.00.00.000 ІІЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

Додаток А

Програмні коди

TypeScript: main.ts

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { AppModule } from '../app/app.module';
import { environment } from '../environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

CSS: Style.css

```
.table thead {
  background-color: black !important;
  color: white !important;
  font-weight: bold;
}
```

TypeScript: test.ts

```
// Цей файл необхідний для karma.conf.js і рекурсивно завантажує всі файли .spec і framework
import 'zone.js/testing';
import { getTestBed } from '@angular/core/testing';
import {
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting
} from '@angular/platform-browser-dynamic/testing';

declare const require: {
  context(path: string, deep?: boolean, filter?: RegExp): {
    keys(): string[];
    <T>(id: string): T;
  };
};

// Спочатку ініціалізуємо середовище тестування Angular.
getTestBed().initTestEnvironment(
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting(),
  { teardown: { destroyAfterEach: true } },
);

// Потім знаходимо всі тести.
const context = require.context('./', true, /\.spec\.ts$/);
// І завантажуємо модулі.
context.keys().map(context);
```

JavaScript: Admin.controller.js

```
const AdminModel = require("../Models/Admin.model");
const bcryptHelper = require("../Helpers/bcrypt.helper");
const jwtHelper = require("../Helpers/jwt.helper");
const AdminController = {};

AdminController.loginAdmin = async(req, res) => {
  try {
    let email = req.body.email;
    let password = req.body.password;
    AdminModel.getEmailByAdmin(email).then(async(response) => {
      let adminDetails = JSON.parse(JSON.stringify(response[0]));
      let check = await bcryptHelper.comparePassword(password, adminDetails.password);
      console.log(check)
      if (check) {
        delete adminDetails.password
        token = await jwtHelper.generateToken(adminDetails);
        res.send({ token: token, adminDetails: adminDetails });
      } else {
        res.send({ status: "error", message: "Unable to login password" })
      }
    }).catch((e) => {
      console.log(e);
      res.send({ status: "error", message: "Unable to login" })
    })
  } catch (e) {
    console.log(e);
  }
  AdminController.addAdmin = async(req, res) => {
    try {
      let adminDetails = req.body;
      adminDetails["password"] = await bcryptHelper.encryptPassword(adminDetails["password"])
      AdminModel.addAdmin(adminDetails).then((response) => {
        res.send({ status: "success", message: "Admin added successfully" })
      }).catch(e => {
        console.log(e)
        res.send({ status: "error", message: "Unable to add admin" })
      })
    } catch (e) {
      res.send({
        status: "error",
        message: "Unable to insert admin"
      })
    }
  }
}

AdminController.getAdmins = (req, res) => {
  try {
    // Логіка для отримання адміністраторів
  } catch (e) {
    console.log(e);
    res.send({ status: "error", message: "Unable to get admin details" })
  }
}

module.exports = AdminController;
```

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи: “ Розробка Web-базованої системи для житлово-експлуатаційних компаній ”

Обсяг пояснювальної записки: 77 аркушів.

Дата закінчення роботи: 10 червня 2025 р.

Підпис студента _____