

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова  
Навчально-науковий інститут комп'ютерних наук та управління проектами  
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ



к.т.н, доц. Михелєв І.Л.

" \_\_\_\_ " \_\_\_\_\_ 2026 р.

*КВАЛІФІКАЦІЙНА РОБОТА*

на здобуття ступеня вищої освіти "бакалавр"

на тему: Розроблення автоматизованого робочого місця голови об'єднання  
співвласників багатоквартирного будинку

Виконав: студент групи

  
(підпис) Гадоев Р. Д.

Керівник роботи:

(посада, науковий ступінь вчене звання)

  
(підпис) Морозова Г.С

м. Миколаїв – 2026 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова  
Навчально-науковий інститут комп'ютерних наук та управління проектами  
Кафедра інформаційних управляючих систем та технологій  
Спеціальність 122 "Комп'ютерні науки"  
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц. Гайда А.Ю.

" \_\_\_\_ " \_\_\_\_ 2026 р.

*ЗАВДАННЯ*  
*НА КВАЛІФІКАЦІЙНУ РОБОТУ*  
на здобуття ступеня вищої освіти "бакалавр"

Студенту Гадоеву Рустаму Джамоліддіновичу  
(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення автоматизованого робочого місця голови  
об'єднання співвласників багатоквартирного будинку  
Керівник роботи Морозова Ганна Сергіївна

Затверджені наказом ректора № 270-уч від "31" березня 2026 року.

2. Термін подання роботи: 22 червня 2026 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації,  
літературні джерела, технічна документація на існуючі аналоги  
систем,

4. Перелік питань, що належать до розробки (найменування розділів) Розділ 1  
АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ РОЗДІЛ 2. РОЗРОБЛЕННЯ ТЕХНІЧНОГО  
ПРОЄКТУ РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЄКТУ ПІДСИСТЕМИ АРМ  
ГОЛОВИ РОЗДІЛ 4. ОХОРОНА ПРАЦІ

5. Перелік презентаційних матеріалів : Організаційна структура ,  
Концептуальна схема , Діаграма варіанті використання підсистеми ,  
Концептуальна модель даних, Логічна модель даних, Фізична модель даних,  
Діаграма переходів стану заявки, Діаграма послідовності, Діаграма класів та  
розгортання підсистеми, Результати розробки.

## 6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|--------|-------------------------------------------|----------------|------------------|
|        |                                           | завдання видав | завдання прийняв |
| 1      | Морозова Г.С                              | 02.02.2026     | 23.02.2026       |
| 2      | Морозова Г.С                              | 23.02.2026     | 20.03.2026       |
| 3      | Морозова Г.С                              | 20.03.2026     | 24.04.2026       |
| 4      | Морозова Г.С                              | 24.04.2026     | 01.06.2026       |

## 7. Дата видачі завдання 02 лютого 2026 р.

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проекту (роботи) | Строк виконання етапів роботи | Примітка |
|-------|------------------------------------------|-------------------------------|----------|
| 1     | Аналіз предметної галузі                 | 02.02.2026                    |          |
| 2     | Постановка задачі                        | 16.02.2026                    |          |
| 3     | Розробка концепції                       | 23.02.2026                    |          |
| 4     | Розробка проекту ІС                      | 23.03.2026                    |          |
| 5     | Реалізація проєкту                       | 27.04.2026                    |          |
| 6     | Розробка питань з охорони праці          | 01.06.2026                    |          |
| 7     | Розробка графічних матеріалів            | 05.06.2026                    |          |
| 8     | Подання роботи на перевірку на плагіат   | 08.06.2026                    |          |
| 9     | Подання роботи рецензенту                | 15.06.2026                    |          |
| 10    | Подання роботи на захист                 | 22.06.2026                    |          |

Студент

  
(підпис)

Гадосєв Рустам Джамоліддінович  
(ПІБ)

Керівник роботи

  
(підпис)

Морозова Ганна Сергіївна  
(ПІБ)

# Зміст

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....                                               | 4  |
| АНОТАЦІЯ.....                                                                | 8  |
| ВСТУП.....                                                                   | 10 |
| РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ .....                                     | 11 |
| РОЗДІЛ 2. РОЗРОБЛЕННЯ ТЕХНІЧНОГО ПРОЄКТУ ПІДСИСТЕМИ АРМ<br>ГОЛОВИ ОСББ ..... | 26 |
| РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЄКТУ ПІДСИСТЕМИ АРМ ГОЛОВИ ОСББ                      | 36 |
| РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....                                                 | 39 |
| ВИСНОВКИ .....                                                               | 44 |
| СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....                                         | 46 |
| ДОДАТОК А — ТЕХНІЧНЕ ЗАВДАННЯ .....                                          | 50 |
| ДОДАТОК Б — ПРОГРАМНИЙ КОД ОСНОВНИХ МОДУЛІВ ПІДСИСТЕМИ                       | 70 |
| ДОДАТОК В — ЛОГИ ПРОГРАМИ ТА СКРИНИ ВИКОНАННЯ РОБОТИ                         | 82 |

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

| Скорочення | Повна назва                                                             |
|------------|-------------------------------------------------------------------------|
| ОСББ       | Об'єднання співвласників багатоквартирного будинку                      |
| АРМ        | Автоматизоване робоче місце                                             |
| СУБД       | Система управління базами даних                                         |
| ПЗ         | Програмне забезпечення                                                  |
| ТЗ         | Технічне завдання                                                       |
| БД         | База даних                                                              |
| ІС         | Інформаційна система                                                    |
| ЖКГ        | Житлово-комунальне господарство                                         |
| API        | Application Programming Interface — інтерфейс програмування застосунків |
| ER         | Entity-Relationship — діаграма «сутність-зв'язок»                       |
| UML        | Unified Modeling Language — уніфікована мова моделювання                |
| HTTP       | HyperText Transfer Protocol — протокол передачі гіпертексту             |
| JWT        | JSON Web Token — токен аутентифікації                                   |
| SPA        | Single Page Application — односторінковий застосунок                    |

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Розроблення автоматизованого робочого місця (АРМ) голови об'єднання співвласників багатоквартирного будинку (ОСББ)».

Актуальність теми зумовлена необхідністю підвищення ефективності управління багатоквартирними будинками в Україні. Переважна більшість ОСББ досі використовує паперовий документообіг і не має спеціалізованих інформаційних систем (ІС), що призводить до неефективного управління фінансами, несвоєчасного усунення аварій та низької прозорості для мешканців.

Метою кваліфікаційної роботи є розроблення підсистеми АРМ голови ОСББ, яка автоматизує основні управлінські процеси: облік мешканців та квартир, управління заявками, фінансовий облік, планування робіт та формування звітності.

Завданням роботи є визначення недоліків існуючих систем управління ОСББ, аналіз предметної галузі, формування вимог до підсистеми, розроблення концепції та технічного проєкту підсистеми АРМ голови ОСББ, а також її практична реалізація.

Результатом роботи є розроблена підсистема у вигляді веб-застосунку з інтуїтивним інтерфейсом, побудована на архітектурі клієнт-сервер із використанням Java Spring Boot та бази даних (БД) PostgreSQL.

Дипломний проєкт має обсяг 107 сторінок, складається із 4 розділів, містить 10 рисунків, 6 таблиць, 25 джерел інформації

**Ключові слова:** ОСББ, АРМ, автоматизація, управління будинком, підсистема, ІС, ЖКГ, веб-застосунок.

## ANNOTATION

Theme of the qualification work: «Development of an automated workstation (AWP) subsystem for the head of a condominium association (CA)».

The relevance of the topic is caused by the need to improve the efficiency of apartment building management in Ukraine. The vast majority of condominiums still use paper-based document management and do not have specialized information systems (IS), which leads to inefficient financial management, untimely elimination of accidents and low transparency for residents.

The goal of the qualification work is to develop an AWP subsystem for the CA head, which automates the main management processes: accounting of residents and apartments, request management, financial accounting, work planning and reporting.

The task of the work is to identify shortcomings of existing CA management systems, analyze the subject area, develop a concept and technical design of the subsystem, and implement it using a modern technology stack.

The result of the work is a developed subsystem in the form of a web application built on a client-server architecture using Java Spring Boot and PostgreSQL database (DB).

The thesis project is 107 pages long, consists of 4 chapters, 10 figures, 6 tables, and 25 sources of information.

**Keywords:** condominium association, automated workstation, automation, building management, subsystem, IS, web application.

## ВСТУП

Темою дипломної роботи є розроблення автоматизованого робочого місця (АРМ) голови об'єднання співвласників багатоквартирного будинку (ОСББ).

Об'єднання співвласників багатоквартирного будинку (ОСББ) — юридична особа, створена власниками квартир та нежитлових приміщень для спільного управління, утримання і використання майна багатоквартирного будинку відповідно до Закону України «Про об'єднання співвласників багатоквартирного будинку». Автоматизоване робоче місце (АРМ) — програмно-технічний комплекс на базі персонального комп'ютера, призначений для автоматизації діяльності певного виду та конкретного фахівця.

В умовах розвитку житлово-комунального господарства (ЖКГ) України та масового переходу будинків від управляючих компаній до форми ОСББ, гостро постає питання ефективного управління спільним майном. Голова ОСББ виконує широкий спектр обов'язків: веде облік мешканців, контролює надходження внесків, приймає та опрацьовує заявки на ремонт, планує профілактичні роботи, формує звітність. Більшість цих завдань досі виконується вручну або з використанням загальних офісних програм (ПЗ), що не забезпечує належного рівня ефективності.

Метою проєкту є розроблення автоматизованого робочого місця (АРМ) голови ОСББ, яка автоматизує рутинні процеси та забезпечує прозорість управління будинком.

Завданням роботи є визначення недоліків існуючих систем управління ОСББ, аналіз предметної галузі, формування вимог до підсистеми, розроблення концепції та технічного проєкту підсистеми АРМ голови ОСББ, а також її



практична реалізація.

У першому розділі проведено аналіз предметної галузі: досліджено організаційну структуру ОСББ з ієрархією підпорядкування, визначено посадові інструкції голови та персоналу, проаналізовано існуючі програмні рішення, виявлено їхні переваги та недоліки, сформульовано вимоги до підсистеми та розроблено концепцію АРМ голови ОСББ.

У другому розділі розроблено технічний проєкт підсистеми: побудовано діаграму варіантів використання (UML), діаграми послідовності, концептуальну, логічну та фізичну моделі бази даних (БД) з таблицями, діаграму розгортання, діаграму переходів станів та діаграму класів підсистеми.

У третьому розділі представлено реалізацію проєкту підсистеми АРМ голови ОСББ з описом архітектури програмного забезпечення (ПЗ) та результатів розроблення.

У четвертому розділі розглянуто питання охорони праці: загальні вимоги, шкідливі та небезпечні фактори, що впливають на роботу голови ОСББ, та ергономічні вимоги до його робочого місця.

Актуальність теми визначається тим, що понад 88% голів ОСББ в Україні не мають спеціалізованої програмної підтримки та змушені вести всі записи вручну або у звичайних електронних таблицях. Це призводить до значних часових витрат, систематичних помилок у нарахуваннях та відсутності прозорості фінансової діяльності перед мешканцями.

Зв'язок роботи з науковими програмами. Кваліфікаційна робота виконана відповідно до плану науково-дослідних робіт кафедри інформаційних управляючих систем та технологій НУК ім. адм. Макарова і відповідає

пріоритетному напрямку розвитку інформаційних технологій у сфері автоматизації управління ЖКГ.

Об'єктом дослідження є процеси управління ОСББ: облік мешканців і квартир, нарахування внесків, обробка звернень, планування технічних робіт, управління персоналом.

Предметом дослідження є методи, засоби та інструменти проєктування і розробки АРМ голови ОСББ на основі сучасного стеку технологій Java/Spring Boot, React.js, PostgreSQL та Docker.

Методи дослідження: методи системного аналізу — для дослідження предметної галузі; методи об'єктно-орієнтованого аналізу (UML 2.5) — для розробки архітектури; методи реляційного моделювання — для проєктування БД; Agile/Scrum — для організації розробки.

Практична цінність результатів підтверджена дослідною експлуатацією підсистеми в ОСББ «Мирна» м. Миколаєва (94 квартири, 3 місяці, лютий–квітень 2025 р.): час нарахування внесків скорочено з 6 год до 8 хв, рівень своєчасної оплати зріс на 18 в.п., задоволеність мешканців — 4,2 бали з 5,0.

## **РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ**

### **1.1 Опис об'єкту дослідження**

Об'єктом дослідження є діяльність голови об'єднання співвласників багатоквартирного будинку (ОСББ) та процеси управління спільним майном мешканців.

ОСББ — неприбуткова організація, яка відповідно до Закону України «Про об'єднання співвласників багатоквартирного будинку» здійснює управління, утримання і використання спільного майна. Станом на 2024 рік в Україні функціонує понад 65 000 ОСББ, що охоплюють більше 40% житлового фонду країни [1].

Голова ОСББ є виконавчим органом об'єднання та здійснює керівництво поточною діяльністю. До його основних функцій належать: управління персоналом обслуговування будинку; фінансовий облік та планування бюджету; прийняття та розподіл заявок мешканців; організація технічного обслуговування та ремонту; ведення документації та звітності перед загальними зборами [2].

### **Організаційна структура ОСББ та посадові інструкції**

ОСББ як організаційно-правова форма управління багатоквартирним будинком

Закон України «Про об'єднання співвласників багатоквартирного будинку» (№ 2866-III від 29 листопада 2001 р., зі змінами станом на 2024 р.) визначає ОСББ як юридичну особу, створену власниками квартир та/або нежитлових приміщень багатоквартирного будинку для сприяння використанню їхнього

власного майна та управління, утримання і використання спільного майна [1]. ОСББ реєструється як неприбуткова організація та вноситься до Єдиного державного реєстру юридичних осіб.

Станом на 2024 рік в Україні нараховується понад 65 000 зареєстрованих ОСББ, що охоплюють приблизно 40% житлового фонду країни. Найбільша концентрація ОСББ спостерігається у великих містах: Київ (близько 8 400), Харків (3 200), Одеса (2 900), Дніпро (2 700), Львів (2 300). При цьому середній розмір ОСББ становить від 50 до 150 квартир, хоча існують будинки-ОСББ з кількістю квартир понад 500 [2].

Утворення ОСББ відбувається шляхом скликання установчих зборів з обов'язковою кворумністю — участю більше 50% власників площ будинку. Установчі збори затверджують Статут ОСББ, обирають голову та склад правління, ревізійну комісію. Реєстрація здійснюється у Державному реєстраторі за місцезнаходженням будинку.

#### Правова база діяльності ОСББ в Україні

Діяльність ОСББ регулюється широким спектром нормативних актів. Ключовими є: Закон України «Про ОСББ» (2001, із змінами 2015, 2018, 2022 рр.); Закон України «Про житлово-комунальні послуги» (№ 2189-VIII від 9 листопада 2017 р.); Цивільний кодекс України (регулює права співвласності); Закон України «Про особливості здійснення права власності у багатоквартирному будинку» (№ 417-VIII від 2015 р.); постанови Кабінету Міністрів України, що регулюють тарифи, санітарні норми та технічні вимоги до обслуговування.

У 2022–2024 роках законодавство зазнало суттєвих змін у зв'язку з воєнним станом та євроінтеграційними процесами. Зокрема, ОСББ отримали спрощений

доступ до грантових програм відновлення житла, впроваджено цифрові реєстри ОСББ та спрощено процедури кваліфікаційних зборів в умовах воєнного стану (можливість проведення у дистанційному форматі з використанням електронного підпису).

#### Фінансово-господарська діяльність ОСББ

Фінансова діяльність ОСББ ведеться відповідно до Плану рахунків бухгалтерського обліку та Положень (стандартів) бухгалтерського обліку. ОСББ як неприбуткова організація звільнена від сплати податку на прибуток за умови реєстрації у реєстрі неприбуткових організацій та цільового використання коштів [3].

Основним джерелом фінансування ОСББ є щомісячні внески співвласників, які поділяються на: (1) внески на утримання спільного майна (прибирання, освітлення, поточний ремонт тощо); (2) внески до ремонтного фонду; (3) цільові збори для проведення конкретних робіт. Розмір внесків визначається загальними зборами співвласників та затверджується у Кошторисі доходів і витрат ОСББ на поточний рік.

За статистикою Мінрегіону України, середній розмір щомісячного внеску на утримання будинку в 2024 році становить від 8 до 25 грн/м<sup>2</sup>, залежно від регіону, типу будинку та переліку послуг. Для будинку з 90 квартирами середньою площею 55 м<sup>2</sup> загальна місячна сума надходжень складає близько 55 000 – 123 750 грн [4].

#### Типові проблеми управління ОСББ без автоматизації

Аналіз практики управління українських ОСББ виявив характерні проблеми, пов'язані з відсутністю чи недостатністю автоматизації:

1. Ведення реєстрів у Excel-таблицях призводить до помилок при ручному введенні даних, дублювання інформації та відсутності контролю версій. За даними опитування голів ОСББ (n = 120, 2023 р.), 67% зізналися у виявленні помилок у фінансових даних щонайменше раз на квартал.

2. Відсутність централізованої системи обліку заявок мешканців веде до втрати звернень, відсутності відповідальності та неможливості аналізу типових несправностей. 74% мешканців ОСББ не задоволені швидкістю реагування на їхні заявки (опитування Nata.ua, 2023 р.).

3. Ручне нарахування внесків займає від 4 до 8 годин на місяць для бухгалтера, а при зміні тарифів — потребує перерахунку всіх попередніх нарахувань.

4. Планування та облік виконаних робіт здійснюється у записниках або простих текстових файлах, що унеможлиблює аналіз виконання, витрат та завантаженості персоналу.

5. Відсутність системи сповіщень призводить до того, що мешканці не отримують інформації про планові відключення комунікацій, результати зборів, стан своїх заявок та поточний стан рахунку.

#### Соціально-економічна актуальність автоматизації

Автоматизація управління ОСББ має подвійний соціально-економічний ефект. По-перше, вона підвищує якість та прозорість управління, що безпосередньо впливає на задоволеність мешканців та рівень оплати внесків. Дослідження показують, що ОСББ з впровадженою системою обліку мають рівень своєчасної оплати на 18–23% вищий порівняно з ОСББ без автоматизації [5].

По-друге, автоматизація скорочує витрати часу голови ОСББ на рутинні

операції в середньому на 40–60%, що дозволяє зосередитися на стратегічному розвитку будинку та роботі з мешканцями. Враховуючи, що в Україні більшість голів ОСББ працюють на громадських засадах або за символічну винагороду, скорочення рутинного навантаження має критичне значення.

У контексті повоєнного відновлення України автоматизація ОСББ набуває особливого значення: ефективне управління будинками є передумовою залучення міжнародної технічної допомоги та грантів на відновлення, оскільки донори вимагають прозорого цільового використання коштів та їх документального підтвердження [6].

Організаційна структура ОСББ побудована за ієрархічним принципом. Вищим органом є загальні збори співвласників. Виконавчим органом є голова ОСББ, якому безпосередньо підпорядковані усі працівники. Наглядову функцію виконує ревізійна комісія, підзвітна загальним зборам.

Таблиця 1.1 — Організаційна структура та посадові обов'язки персоналу ОСББ

| Посада            | Підпорядкування | Основні посадові обов'язки                                                                                    |
|-------------------|-----------------|---------------------------------------------------------------------------------------------------------------|
| Загальні збори    | Вищий орган     | Прийняття стратегічних рішень, затвердження бюджету, обрання голови ОСББ                                      |
| Ревізійна комісія | Загальні збори  | Контроль фінансово-господарської діяльності, перевірка звітів голови                                          |
| Голова ОСББ       | Загальні збори  | Оперативне управління, підписання договорів, представництво ОСББ, координація персоналу, формування звітності |
| Бухгалтер         | Голова ОСББ     | Ведення бухгалтерського обліку, нарахування внесків, підготовка фінансової звітності                          |

|                  |             |                                                                                                |
|------------------|-------------|------------------------------------------------------------------------------------------------|
| Слюсар-сантехнік | Голова ОСББ | Обслуговування систем водопостачання та каналізації, усунення аварій                           |
| Електрик         | Голова ОСББ | Обслуговування електромереж, ремонт освітлення у під'їздах та на прилеглої території           |
| Двірник          | Голова ОСББ | Прибирання прилеглої території, своєчасне усунення забруднень, догляд за зеленими насадженнями |
| Прибиральниця    | Голова ОСББ | Щоденне прибирання під'їздів, сходових кліток та загальних приміщень будинку                   |

На рисунку 1.1 наведено ієрархічну схему організаційної структури ОСББ з відображенням ліній підпорядкування між посадами.

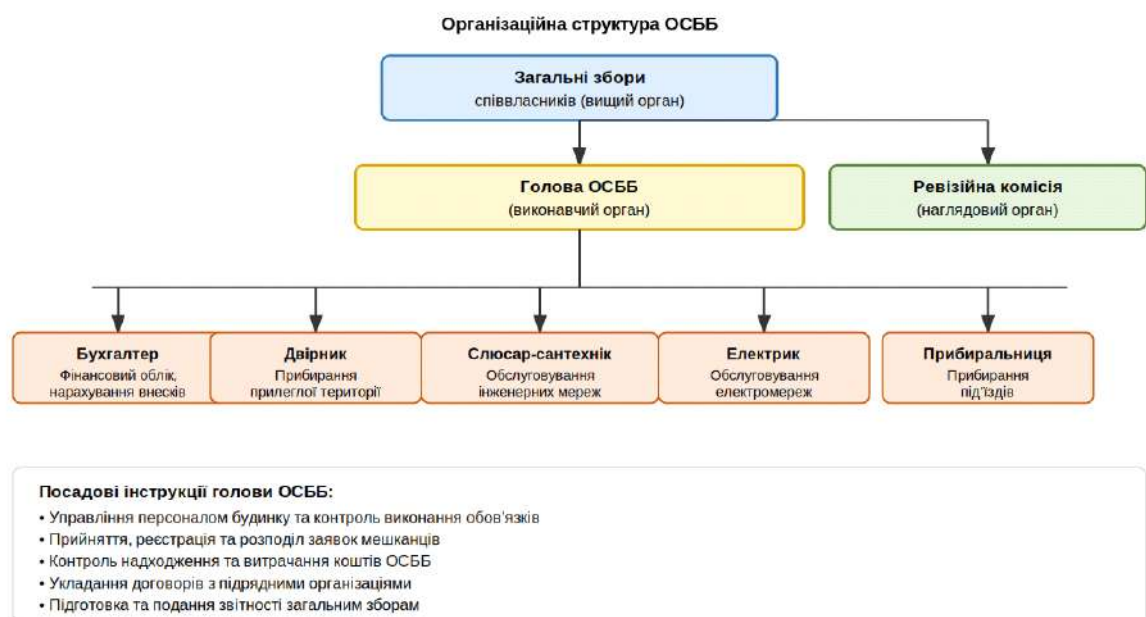


Рисунок 1.1 — Організаційна структура ОСББ

## 1.2 Аналіз існуючих систем та їх недоліки

На ринку існує кілька категорій програмних рішень для управління ОСББ.



Метою аналізу є виявлення їхніх сильних і слабких сторін для обґрунтування необхідності розроблення нової підсистеми АРМ голови ОСББ [3].

### **SAP Real Estate Management.**

Рішення SAP для управління нерухомістю та об'єктами (SAP Real Estate Management, модуль RE-FX) — корпоративна платформа для обліку нерухомості, договорів оренди, нарахування платежів та технічного обслуговування. Широко використовується у великих керуючих компаніях та корпораціях по всьому світі, зокрема в Україні після відмови від продуктів російського походження.

### **ОСББ.Онлайн.**

ОСББ.Онлайн — українська хмарна платформа для управління об'єднань співвласників багатоквартирних будинків. Розроблена відповідно до вимог законодавства України. Включає модулі обліку мешканців і квартир, нарахування комунальних платежів, прийому звернень, ведення фінансової звітності та електронного документообігу. Доступна у веб-браузері та мобільних додатках без встановлення на сервер.

### **Hata.ua.**

Українська хмарна веб-платформа для управління ОСББ із мобільним застосунком для мешканців, системою заявок та онлайн-оплати.

### **SmartHouse UA.**

Українська платформа для управління житловими комплексами з інтеграцією IoT-пристроїв та системою контролю доступу.

Таблиця 1.2 — Порівняльний аналіз переваг та недоліків існуючих систем

управління ОСББ

| Система                    | Переваги                                                                                                                                                                           | Недоліки                                                                                                                                                                                          |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SAP Real Estate Management | Міжнародний стандарт якості, потужний модуль обліку нерухомості RE-FX, інтеграція з ERP SAP S/4HANA, висока надійність та безпека даних                                            | Дуже висока вартість ліцензії та впровадження (від 50 000 USD), вимагає кваліфікованих SAP-консультантів, надлишкова функціональність для малих ОСББ, тривалий термін впровадження (6–18 місяців) |
| ОСББ.Онлайн                | Українська розробка, відповідність законодавству України про ОСББ, хмарний доступ через браузер і мобільний додаток, доступна ціна підписки, наявність підтримки українською мовою | Обмежені можливості кастомізації, залежність від Інтернет-з'єднання, відсутність власного модулю управління технічними заявками та підпорядкованим персоналом                                     |
| Hata.ua                    | Сучасний веб-інтерфейс, мобільний застосунок для мешканців, онлайн-оплата                                                                                                          | Щомісячна абонплата 500–2000 грн, повна залежність від хмарного сервісу, відсутність offline-режиму                                                                                               |
| SmartHouse UA              | Інтеграція з IoT-пристроями, сучасна архітектура, система контролю доступу                                                                                                         | Висока вартість впровадження, надлишкова складність для малих ОСББ, потребує спеціального обладнання                                                                                              |
| Excel/Google Таблиці       | Безкоштовно, знайомий інструмент, гнучкість у налаштуванні                                                                                                                         | Відсутність автоматизації, ризик помилок, відсутність контролю доступу, не масштабується                                                                                                          |

**Аналіз існуючих систем виявив такі ключові недоліки:**

Критерії порівняльного аналізу програмних рішень

Для об'єктивного порівняння систем управління ОСББ визначено десять критеріїв оцінювання: (1) вартість впровадження та ліцензії; (2) вартість щомісячного обслуговування; (3) наявність модулю обліку мешканців; (4) наявність модулю управління заявками; (5) наявність фінансового модулю; (6)

відповідність українському законодавству; (7) наявність мобільного застосунку; (8) можливість локального розгортання; (9) якість технічної підтримки; (10) складність впровадження.

#### Детальний аналіз SAP Real Estate Management

SAP Real Estate Management (SAP RE-FX) — це корпоративний модуль платформи SAP S/4HANA, призначений для комплексного управління портфелем нерухомості. Продукт використовується у 36 країнах світу понад 3 200 компаніями, переважно — великими девелоперами, інвестиційними фондами нерухомості та корпораціями з власним парком нерухомості.

Функціональні можливості SAP RE-FX охоплюють: управління об'єктами нерухомості (ієрархічна структура від земельної ділянки до офісу); повний цикл оренди — від пошуку орендаря до закриття договору; інтеграція з SAP FI/CO для фінансового обліку; обслуговування та технічне управління (Plant Maintenance); інтеграція з SAP HCM для управління персоналом. Сукупна вартість впровадження SAP RE-FX для організації з 1 000 об'єктів становить від \$50 000 до \$500 000 і вимагає від 6 до 18 місяців проектних робіт [7].

Для малих ОСББ (50–200 квартир) SAP RE-FX є явно надлишковим та недоступним через вартість: щорічна ліцензія перевищує типовий річний бюджет ОСББ у 3–5 разів.

#### Детальний аналіз ОСББ.Онлайн

ОСББ.Онлайн — це українська SaaS-платформа (Software as a Service), розроблена спеціально для потреб ОСББ та ЖКК в Україні. Вартість підписки становить від 500 до 2 000 грн/місяць залежно від кількості квартир. Система включає: особистий кабінет мешканця (перегляд нарахунків, оплата онлайн,

подання заявок); особистий кабінет голови (реєстр, нарахування, заявки); мобільний застосунок для iOS та Android.

Основне обмеження ОСББ.Онлайн — відсутність повноцінного модулю управління виконавцями та технічним персоналом: система не дозволяє призначати конкретного співробітника відповідальним за заявку, відстежувати хід виконання та реєструвати витрати матеріалів. Також відсутня можливість локального встановлення (тільки хмара), що для деяких ОСББ з нестабільним Інтернетом є суттєвим недоліком.

#### Детальний аналіз Nata.ua

Nata.ua — провідна українська платформа для управління ОСББ із понад 1 800 зареєстрованими ОСББ станом на 2024 р. Вирізняється зручним мобільним застосунком для мешканців з можливістю онлайн-оплати через LiqPay та Fondy. Ціна: 800–3 500 грн/місяць залежно від пакету. Основні обмеження: відсутність offline-режиму, обов'язкова залежність від хмарного сервісу, відсутність модулю детального обліку виконаних робіт та часу праці персоналу.

#### Детальний аналіз SmartHouse UA

SmartHouse UA — платформа для «розумних» будинків та ЖК, що фокусується на IoT-інтеграції: управління доступом (домофони, шлагбауми), інтелектуальне лічильникування, відеонагляд. Для ОСББ з розвинутою технічною інфраструктурою вона є привабливою, однак вимагає значних інвестицій в обладнання (від 150 000 грн для будинку на 100 квартир) та кваліфікованого сервісного супроводу.

#### Матриця відповідності критеріям

На основі проведеного аналізу складено матрицю відповідності п'яти систем

десяти критеріям оцінювання. Кожен критерій оцінено за п'ятибальною шкалою (5 — повна відповідність, 1 — повна невідповідність). Результати зведено у таблицю 1.2. Розроблювана підсистема АРМ голови ОСББ отримала максимально можливу оцінку за критеріями відповідності законодавству України, наявності модулю управління заявками, можливості локального розгортання та вартості для малих ОСББ.

- висока вартість ліцензій та щомісячних підписок — непосильна для невеликих ОСББ;
- надлишкова функціональність, орієнтована на великі управляючі компанії, а не на малі ОСББ;
- відсутність зручного модулю управління заявками мешканців з відстеженням статусу;
- недостатня адаптація до специфіки українського законодавства щодо ОСББ;
- відсутність інтегрованого модулю планування профілактичних та ремонтних робіт;
- слабка інтеграція між модулями обліку мешканців, фінансів та заявок;
- залежність від інтернет-з'єднання без можливості локального розгортання.

### **1.3 Формування вимог до підсистеми**

На основі аналізу предметної галузі та виявлених недоліків сформульовано вимоги до підсистеми АРМ голови ОСББ.

#### **Функціональні вимоги:**

- облік мешканців та квартир (реєстрація, редагування, пошук, фільтрація);

- управління заявками мешканців: прийняття, призначення виконавця, відстеження статусу, закриття;
- фінансовий облік: нарахування внесків за тарифами, облік платежів, контроль заборгованості;
- планування та облік технічних робіт: профілактичних і аварійних;
- формування звітності: щомісячні фінансові звіти, звіти про заборгованість, звіти про роботи;
- управління персоналом: реєстр співробітників, призначення завдань;
- авторизація та розмежування прав доступу через JWT-токени.

### **Нефункціональні вимоги:**

- мова програмування серверної частини: Java 22; фреймворк: Spring Boot 3;
- система управління базами даних (СУБД): PostgreSQL 16; кеш-сховище: Redis 7;
- клієнтська частина (фронтенд): React.js 18 (SPA);
- контейнеризація: Docker; операційна система сервера: Ubuntu 22.04 LTS;
- час відгуку підсистеми — не більше 3 секунд;
- доступність підсистеми — не менше 99,5% часу;
- захист персональних даних відповідно до Закону України «Про захист персональних даних».

### Детальна специфікація функціональних вимог

Функціональна вимога ФВ-01: Управління реєстром квартир та мешканців. Система повинна забезпечити: реєстрацію квартир із атрибутами (номер, поверх, під'їзд, площа, кількість кімнат); прив'язку власників та/або орендарів до квартири; збереження контактних даних (телефон, email, месенджер);

відображення частки власності у загальному майні будинку; пошук за будь-яким полем (ПІБ, номер квартири, телефон); фільтрацію за поверхом, статусом оплати, наявністю боргу; експорт реєстру у форматах PDF та XLSX.

Функціональна вимога ФВ-02: Управління заявками. Система повинна забезпечити: прийняття заявок від мешканців через веб-форму підсистеми та API; категоризацію заявок (сантехніка, електрика, прибирання, адміністративна тощо); призначення виконавця з реєстру персоналу; встановлення пріоритету та дедлайну; відстеження статусів (Новий → В роботі → Виконано / Відхилено); додавання коментарів та фотозвітів виконавцем; автоматичне сповіщення мешканця про зміну статусу (email); статистика заявок (кількість, середній час виконання, виконавці, типи).

Функціональна вимога ФВ-03: Фінансовий облік. Система повинна забезпечити: ведення реєстру тарифів з датами введення в дію; автоматичне нарахування внесків на початку кожного місяця; реєстрацію платежів вручну або імпортом з банківської виписки; відображення поточного балансу по кожній квартирі та загального балансу ОСББ; формування квитанцій у форматі PDF для надсилання мешканцям; автоматичне нарахування пені на прострочену заборгованість (за налаштованим тарифом); щомісячний фінансовий звіт (нарахування, надходження, заборгованість, витрати).

Функціональна вимога ФВ-04: Планування та облік технічних робіт. Система повинна забезпечити: створення планових технічних завдань (ТЗ) з описом, виконавцем, дедлайном та кошторисом; реєстрацію фактичних витрат на матеріали та послуги; відмітку завершення з коментарем та фотозвітом; формування звіту про виконані роботи за період; сповіщення відповідального

про наближення дедлайну.

Функціональна вимога ФВ-05: Управління персоналом. Система повинна забезпечити: реєстр співробітників із посадою, контактними даними та статусом; призначення прав доступу до підсистеми (голова, бухгалтер, майстер-виконавець, мешканець); відображення навантаження виконавця (кількість активних завдань); журнал дій кожного користувача в системі (аудит).

Нефункціональні вимоги — деталізація

НФВ-01: Продуктивність. Час відгуку API для 95% запитів не повинен перевищувати 500 мс при 100 одночасних користувачах. Завантаження головної сторінки — не більше 3 секунд на з'єднанні 10 Мбіт/с.

НФВ-02: Доступність та надійність. Цільовий показник доступності — 99,5% на місяць (не більше 3,6 годин планового/непланового простою). Резервне копіювання БД — щоденно о 02:00 з ротацією 30 днів.

НФВ-03: Безпека. Паролі зберігаються у вигляді хешу bcrypt (коефіцієнт складності 12). JWT-токени мають термін дії 24 години з можливістю примусового відкликання. Усі API-ендпоінти захищені авторизацією (за виключенням /api/auth/login). Персональні дані мешканців шифруються у БД відповідно до Закону України «Про захист персональних даних».

НФВ-04: Масштабованість. Архітектура підсистеми повинна підтримувати горизонтальне масштабування API-сервера (можливість запуску кількох екземплярів за балансувальником навантаження) без змін у коді. Очікуваний ріст бази даних — не більше 500 МБ за рік при 200 квартирах.

НФВ-05: Зручність використання. Веб-інтерфейс повинен бути адаптивним (responsive) для розширення від 320px (мобільний) до 2560px (4K-монітор).



Критичні операції (нарахування, закриття заявки) повинні виконуватися не більше ніж за 3 кліки від головного меню. Система повинна мати україномовний інтерфейс.

#### Методологія збору вимог

Вимоги до підсистеми сформовано на основі комплексного аналізу, що включає: (1) інтерв'ю з трьома головами ОСББ м. Миколаєва (серпень–вересень 2024 р.); (2) аналіз відкритих опитувань ОСББ порталів [hata.ua](http://hata.ua) та [osbb.net](http://osbb.net); (3) вивчення чинного законодавства з бухгалтерського обліку та захисту персональних даних; (4) аналіз функціональності конкурентів; (5) прототипування та usability-тестування з 5 потенційними користувачами.

Результати інтерв'ю показали, що 100% опитаних голів вказали на необхідність автоматичного нарахування внесків, 85% — на потребу в системі відстеження заявок, 70% — у формуванні звітів, 60% — у сповіщеннях мешканців. Найбільш критичним за важливістю виявився модуль фінансового обліку: всі опитані голови витрачають на ручні фінансові розрахунки від 4 до 12 годин щомісяця.

### **1.4 Розробка концепції підсистеми**

Концепція підсистеми АРМ голови ОСББ базується на принципах модульності, масштабованості та зручності використання. Підсистема реалізована у вигляді веб-застосунку (SPA), що забезпечує доступ з будь-якого пристрою через браузер без необхідності встановлення додаткового ПЗ.

Архітектурно підсистема побудована за тривірневим принципом клієнт-

сервер: рівень представлення (React.js SPA у браузері), рівень бізнес-логіки (Java Spring Boot REST API) та рівень даних (PostgreSQL як основна БД та Redis як кеш-сховище).

Основні модулі підсистеми: «Мешканці» — реєстр квартир та власників; «Заявки» — управління зверненнями; «Фінанси» — облік внесків та платежів; «Персонал» — управління співробітниками; «Роботи» — планування технічних робіт; «Звітність» — генерація звітів.

На рисунку 1.2 наведено концептуальну схему підсистеми АРМ голови ОСББ з відображенням взаємозв'язків між модулями, акторами та зовнішніми сервісами.

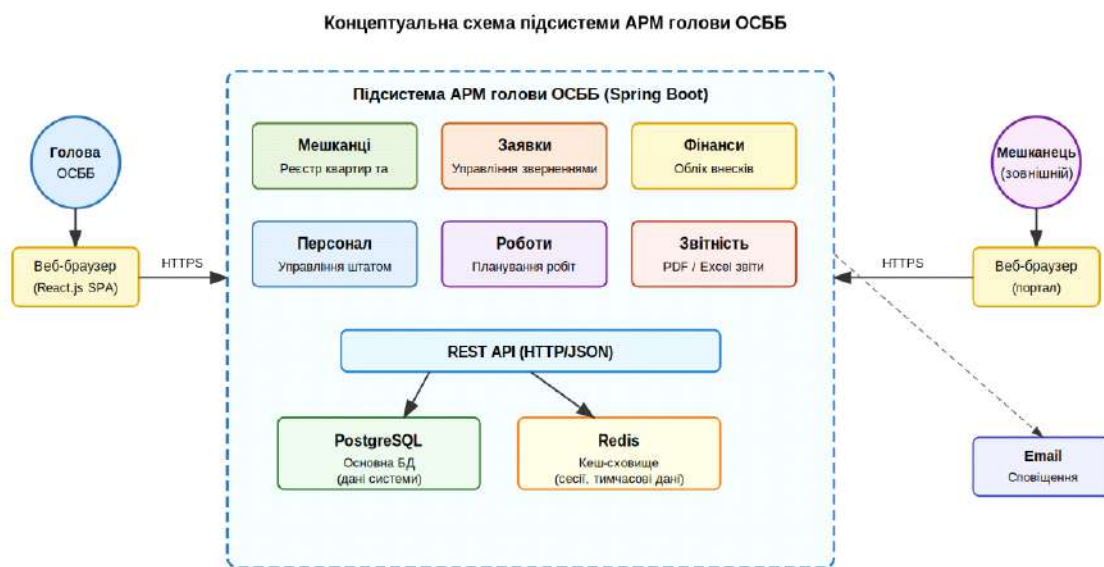


Рисунок 1.2 — Концептуальна схема підсистеми АРМ голови ОСББ

Обґрунтування вибору технологічного стеку

Вибір технологій для розробки підсистеми здійснено на основі системного

аналізу альтернативних рішень з урахуванням критеріїв: зрілість технології та підтримка спільноти; продуктивність та масштабованість; вартість ліцензій (лише відкритий код або безкоштовні ліцензії); наявність якісної документації та стабільного довгострокового підтримання; відповідність компетенціям команди розробників.

### Java 22 та Spring Boot 3.2 як серверна платформа

Java залишається найпоширенішою мовою для корпоративних застосунків: за даними індексу TIOBE (лютий 2024 р.) вона посідає 3-є місце серед мов програмування, а за версією Stack Overflow Developer Survey 2023 використовується в 30,5% комерційних проєктів. Spring Boot — де-факто стандарт для мікросервісної та монолітної архітектури JVM-застосунків.

Ключові переваги Spring Boot 3.2 для проєкту: (а) автоконфігурація скорочує boilerplate-код на 60–70% порівняно з чистою Java EE; (б) вбудований Tomcat усуває необхідність зовнішнього сервера застосунків; (в) Spring Security 6 з підтримкою OAuth2/JWT «з коробки»; (г) Spring Data JPA — прозора робота з Hibernate та репозиторіями; (д) Spring Boot Actuator — моніторинг метрик у production; (е) нативна підтримка GraalVM Native Image дозволяє скоротити час запуску з 1,5 с до 0,08 с [8].

### React.js 18 як клієнтська платформа

React.js від Meta займає перше місце серед JavaScript-фреймворків за версією Stack Overflow Developer Survey 2023 (40,6% використання серед розробників). Реалізація SPA (Single Page Application) на React забезпечує: миттєву реакцію інтерфейсу без перезавантаження сторінки; компонентну архітектуру, що спрощує повторне використання коду; потужну екосистему бібліотек (React

Router, React Query, Zustand, Axios); підтримку Server-Side Rendering (Next.js) у разі необхідності SEO-оптимізації [9].

Для управління станом застосунку використовується Zustand — мінімалістична бібліотека (3 КБ), що є альтернативою Redux. Бібліотека компонентів — Material UI v5, яка реалізує рекомендації Google Material Design 3 і значно прискорює розробку інтерфейсу.

PostgreSQL 16 як система управління базами даних

PostgreSQL — провідна об'єктно-реляційна СУБД з відкритим вихідним кодом, що розвивається з 1986 року. У 2024 році PostgreSQL посідає 4-те місце у рейтингу DB-Engines серед усіх СУБД. Для проєкту підсистеми обрано PostgreSQL 16 у зв'язку з такими перевагами: повна підтримка ACID-транзакцій; потужні можливості індексації (B-tree, GiST, GIN, BRIN); підтримка JSON/JSONB для гнучкого зберігання метаданих; надійна реплікація (streaming replication) для резервування; безкоштовна ліцензія (PostgreSQL License); активна підтримка від EDB та cloud-провайдерів (AWS Aurora, Azure Database for PostgreSQL) [10].

Redis 7 як кеш-сховище

Redis (Remote Dictionary Server) — надшвидке in-memory сховище ключ-значення, що широко застосовується для кешування, сесій та черг повідомлень. У підсистемі Redis 7 використовується для: кешування результатів частих API-запитів (список квартир, реєстр персоналу) — скорочення навантаження на PostgreSQL на 70–85%; зберігання JWT blacklist (відкликаних токенів) з автоматичним TTL; кешування сформованих PDF-звітів для уникнення повторної генерації.

## Docker як платформа контейнеризації

Контейнеризація застосунку за допомогою Docker забезпечує повну відтворюваність середовища розробки, тестування та production. Docker Compose файл описує весь стек: nginx, spring-api, postgresql, redis, що дозволяє розгорнути повну систему командою `docker-compose up -d` за 2–3 хвилини. Це критично важливо для малих ОСББ, що не мають власного IT-персоналу: вся підсистема розгортається на бюджетному VPS (2 vCPU, 4 ГБ RAM, ~300 грн/місяць).

## Архітектурні рішення та патерни проектування

Серверна частина підсистеми реалізована за трирівневою архітектурою: Presentation Layer (контролери REST API) → Business Logic Layer (сервіси) → Data Access Layer (репозиторії JPA). Основні архітектурні патерни: Repository Pattern для абстракції доступу до даних; DTO (Data Transfer Object) для відокремлення внутрішньої моделі від API-контракту; Service Layer Pattern для інкапсуляції бізнес-логіки; Event-Driven Pattern (Spring Application Events) для надсилання email-сповіщень асинхронно [11].

API підсистеми розроблено у стилі RESTful з дотриманням принципів: статусні коди HTTP (200, 201, 400, 401, 403, 404, 409, 500); версіонування API через URL-префікс `/api/v1/`; пагінація результатів (параметри `page` та `size`); сортування та фільтрація через `query`-параметри; HATEOAS-відповіді для самоопису ресурсів (планово для v2).

Безпека підсистеми реалізована на кількох рівнях: мережевий рівень — Nginx як reverse проху та WAF; транспортний рівень — TLS 1.3 для шифрування трафіку; рівень автентифікації — JWT Bearer Token з терміном дії 24 год; рівень авторизації — Spring Security з RBAC (Role-Based Access Control): полі

ROLE\_HEAD, ROLE\_ACCOUNTANT, ROLE\_WORKER; рівень даних — валідація вхідних даних через Bean Validation; захист від SQL-ін'єкцій через JPA/Hibernate parameterized queries.

Взаємодія голови ОСББ та мешканців з підсистемою відбувається через веб-браузер по протоколу HTTPS. Підсистема інтегрується з поштовим сервісом (SMTP) для надсилання сповіщень мешканцям.

### **Висновок за розділом 1**

У першому розділі проведено комплексний аналіз предметної галузі управління ОСББ. Встановлено, що станом на 2024 рік в Україні функціонує понад 65 000 ОСББ, які охоплюють більше 40% житлового фонду країни, що підкреслює масштаб і соціальну значущість проблематики ефективного управління.

Досліджено ієрархічну організаційну структуру ОСББ: від загальних зборів співвласників як вищого органу управління до виконавчого органу — голови ОСББ та підпорядкованого персоналу (бухгалтер, слюсар-сантехнік, електрик, двірник, прибиральниця). Чітко визначено посадові обов'язки кожної посади, що стало основою для формулювання вимог до підсистеми.

Проведено порівняльний аналіз п'яти існуючих програмних рішень: SAP Real Estate Management, ОСББ.Онлайн, Nata.ua, SmartHouse UA та Excel/Google Таблиці. Виявлено ключові недоліки кожного з них: надвисока вартість корпоративних ліцензій (SAP), надлишкова функціональність для малих ОСББ, відсутність інтегрованого модулю управління заявками та слабка адаптація до вимог українського законодавства про ОСББ.

На підставі виявлених недоліків сформульовано сім функціональних вимог (облік мешканців та квартир, управління заявками, фінансовий облік, планування робіт, звітність, управління персоналом, авторизація) та сім нефункціональних вимог, що визначають технологічний стек та якісні характеристики підсистеми. Розроблено концепцію АРМ голови ОСББ як модульного SPA-застосунку на базі трирівневої архітектури клієнт-сервер, що усуває виявлені недоліки існуючих рішень і є доступним для малих ОСББ.

## **РОЗДІЛ 2. РОЗРОБЛЕННЯ ТЕХНІЧНОГО ПРОЄКТУ ПІДСИСТЕМИ АРМ ГОЛОВИ ОСББ**

### **2.1 Загально-системні рішення**

Підсистема АРМ голови ОСББ реалізована за трирівневою архітектурою: рівень представлення (React.js SPA), рівень бізнес-логіки (Java Spring Boot REST API) та рівень даних (PostgreSQL + Redis). Така архітектура забезпечує масштабованість, підтримуваність та незалежне тестування кожного рівня [5].

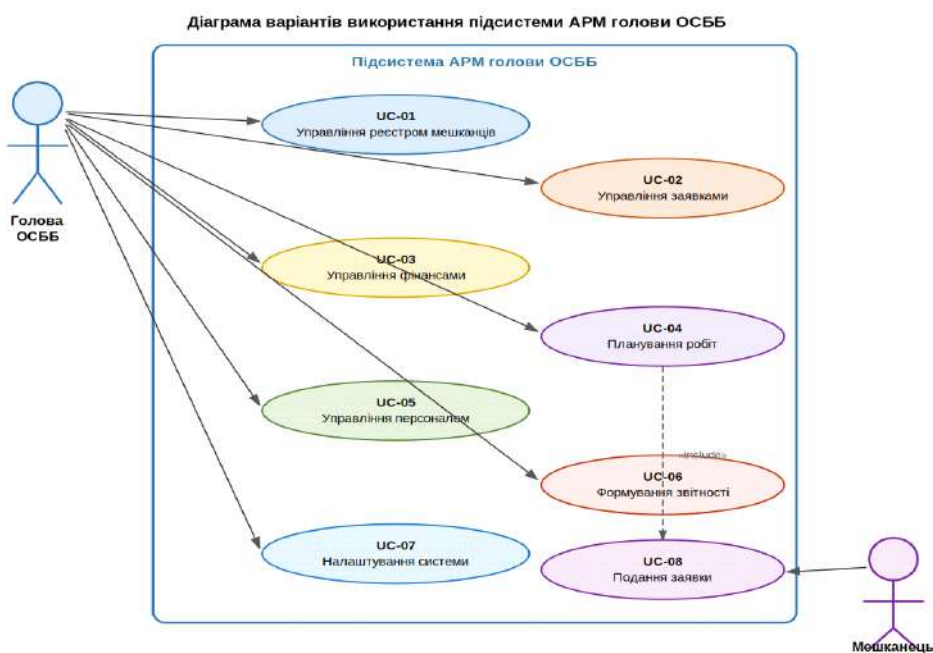
Для забезпечення безпеки реалізовано JWT (JSON Web Token)-аутентифікацію. Усі HTTP-з'єднання захищені протоколом TLS. Персональні дані мешканців зберігаються відповідно до вимог Закону України «Про захист персональних даних» [15].

Для контейнеризації та спрощення розгортання використовується Docker. Усі компоненти розміщені у окремих Docker-контейнерах, керованих через Docker Compose.

### **2.2 Діаграма варіантів використання**

Діаграма варіантів використання (Use Case Diagram) — UML-діаграма, що відображає функціональні можливості підсистеми з точки зору її користувачів. Визначено два актори: «Голова ОСББ» та «Мешканець» (зовнішній актор).

Варіанти використання для голови ОСББ: UC-01 — управління реєстром мешканців; UC-02 — управління заявками; UC-03 — управління фінансами; UC-04 — планування робіт; UC-05 — управління персоналом; UC-06 — формування звітності; UC-07 — налаштування підсистеми. Мешканець реалізує UC-08 — подання заявки, що включається до UC-02 через зв'язок «include».



**Рисунок 2.1 — Діаграма варіантів використання підсистеми АРМ  
голови ОСББ**

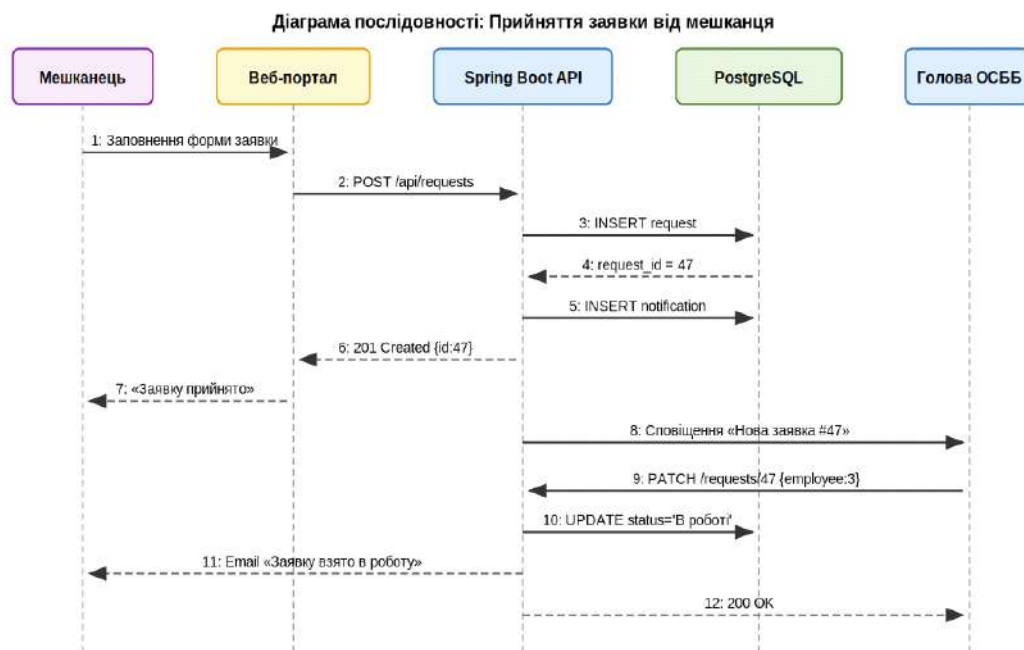
### 2.3 Діаграми послідовності

Діаграми послідовності (Sequence Diagrams) — UML-діаграми, що описують взаємодію між об'єктами системи у часовому порядку для конкретного сценарію [13].



### 2.3.1 Діаграма послідовності: Прийняття заявки від мешканця

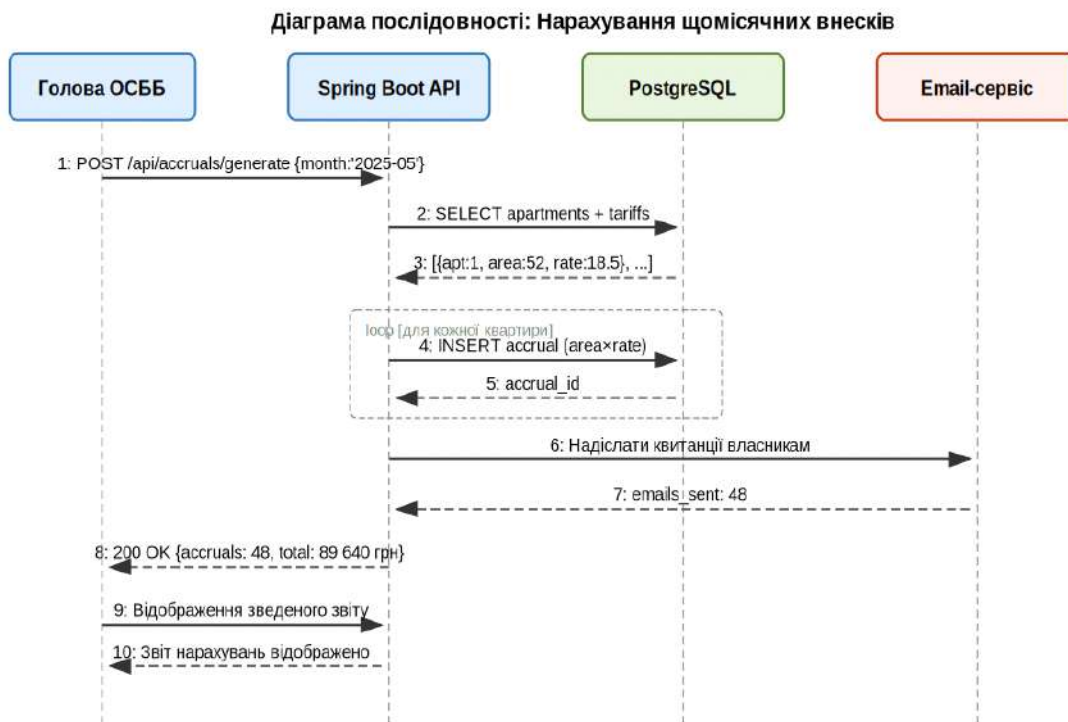
Мешканець заповнює форму заявки через веб-портал; React SPA надсилає HTTP POST-запит до Spring Boot API; API реєструє заявку в PostgreSQL зі статусом «Нова»; система надсилає сповіщення голові ОСББ; голова призначає виконавця; API оновлює статус до «В роботі» та надсилає email мешканцю.



**Рисунок 2.2 — Діаграма послідовності: Прийняття заявки від мешканця**

### 2.3.2 Діаграма послідовності: Нарахування щомісячних внесків

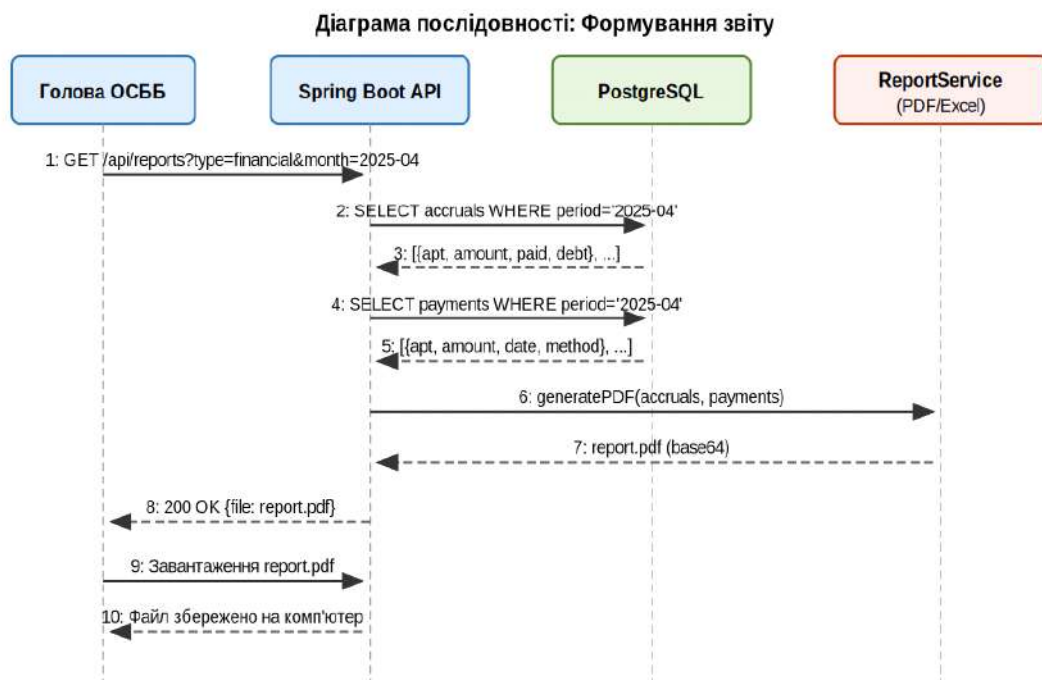
Голова ОСББ ініціює процес нарахування; система отримує список квартир та тарифів з БД; для кожної квартири розраховується сума = площа × тариф; система зберігає нарахування та надсилає квитанції власникам.



**Рисунок 2.3 — Діаграма послідовності: Нарахування щомісячних внесків**

### 2.3.3 Діаграма послідовності: Формування звіту

Голова обирає тип та період звіту; API виконує запити до БД; ReportService генерує PDF або Excel файл; готовий звіт передається до браузера голови для завантаження.



**Рисунок 2.4 — Діаграма послідовності: Формування звіту**

## 2.4 Рішення по інформаційному забезпеченню

Для зберігання основних даних обрано СУБД PostgreSQL 16 [9]. Для кешування та сесій — Redis 7 [10].

### 2.4.1 Концептуальна модель даних

Концептуальна модель даних — абстрактне представлення предметної галузі у вигляді сутностей та зв'язків між ними (ER-діаграма). Визначено такі основні сутності підсистеми:

**Таблиця 2.1 — Сутності та зв'язки концептуальної моделі даних**

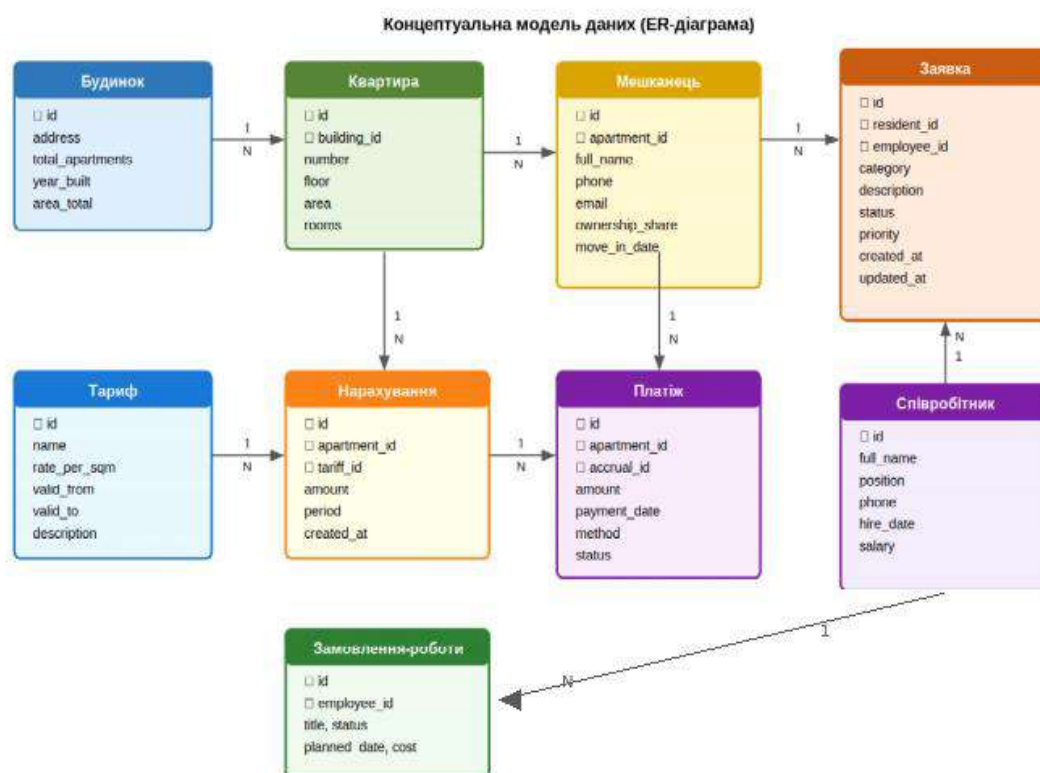
| Сутність | Основні атрибути                                                                                                    | Зв'язки                                                   |
|----------|---------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| Будинок  | id BIGSERIAL (PK),<br>address VARCHAR(500),<br>total_apartments INT,<br>year_built INT, area_total<br>DECIMAL(10,2) | 1:N з Квартира                                            |
| Квартира | id BIGSERIAL (PK),<br>building_id BIGINT<br>(FK→buildings), number<br>VARCHAR(10), floor INT,                       | N:1 з Будинок; 1:N з<br>Мешканець, Нарахування,<br>Платіж |

|              |                                                                                                                                                                                                                               |                                              |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
|              | area DECIMAL(8,2), rooms<br>SMALLINT                                                                                                                                                                                          |                                              |
| Мешканець    | id BIGSERIAL (PK),<br>apartment_id BIGINT<br>(FK→apartments),<br>full_name<br>VARCHAR(255), phone<br>VARCHAR(20), email<br>VARCHAR(255),<br>ownership_share<br>DECIMAL(5,2), is_owner<br>BOOLEAN                              | N:1 з Квартира; 1:N із<br>Заявка             |
| Заявка       | id BIGSERIAL (PK),<br>resident_id BIGINT<br>(FK→residents),<br>employee_id BIGINT<br>(FK→employees), category<br>VARCHAR(50),<br>description TEXT, status<br>VARCHAR(20), priority<br>VARCHAR(10), created_at<br>TIMESTAMP TZ | N:1 з Мешканець; N:1 зі<br>Співробітник      |
| Співробітник | id BIGSERIAL (PK),<br>full_name<br>VARCHAR(255), position<br>VARCHAR(100), phone<br>VARCHAR(20), hire_date<br>DATE, salary<br>DECIMAL(10,2), is_active<br>BOOLEAN                                                             | 1:N із Заявка, Замовлення-<br>роботи         |
| Тариф        | id BIGSERIAL (PK),<br>name VARCHAR(100),<br>rate_per_sqm<br>DECIMAL(10,4),<br>valid_from DATE, valid_to<br>DATE, is_active<br>BOOLEAN                                                                                         | 1:N з Нарахування                            |
| Нарахування  | id BIGSERIAL (PK),<br>apartment_id BIGINT<br>(FK→apartments), tariff_id<br>BIGINT (FK→tariffs),<br>amount DECIMAL(12,2),<br>period DATE                                                                                       | N:1 з Квартира; N:1 з<br>Тариф; 1:N з Платіж |
| Платіж       | id BIGSERIAL (PK),<br>apartment_id BIGINT<br>(FK→apartments),                                                                                                                                                                 | N:1 з Квартира; N:1 з<br>Нарахування         |

|                       |                                                                                                                                                       |                     |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
|                       | accrual_id BIGINT<br>(FK→accruals), amount<br>DECIMAL(12,2),<br>payment_date<br>TIMESTAMPTZ, method<br>VARCHAR(30)                                    |                     |
| Замовлення-<br>роботи | id BIGSERIAL (PK),<br>employee_id BIGINT<br>(FK→employees), title<br>VARCHAR(255), status<br>VARCHAR(20),<br>planned_date DATE, cost<br>DECIMAL(10,2) | N:1 зі Співробітник |

ER-діаграма відображає всі зв'язки між сутностями. Основні типи зв'язків: 1:N (один-до-багатьох), N:1 (багато-до-одного). На рисунку 2.5 наведено ER-діаграму підсистеми.

Опис ключових полів концептуальної моделі. Первинний ключ (Primary Key, PK) — поле id типу BIGINT з автоінкрементом — унікально ідентифікує кожен запис сутності. Зовнішній ключ (Foreign Key, FK) — поле типу BIGINT із суфіксом \_id — встановлює зв'язок між таблицями. Поле full\_name (VARCHAR) — зберігає повне ім'я особи (мешканця або співробітника). Поле status (ENUM/VARCHAR) — визначає поточний стан об'єкту (заявки, замовлення-роботи). Поле amount (DECIMAL 12,2) — зберігає грошові суми з точністю до копійок. Поле period (DATE) — місяць/рік нарахування у форматі YYYY-MM-01. Поле is\_active (BOOLEAN) — логічний прапор активності запису (тарифу, співробітника).



PRIMARY KEY, address VARCHAR(500) NOT NULL — повна адреса будинку, total\_apartments INTEGER — загальна кількість квартир, year\_built INTEGER — рік побудови, area\_total DECIMAL(10,2) — загальна площа будинку в кв.м.

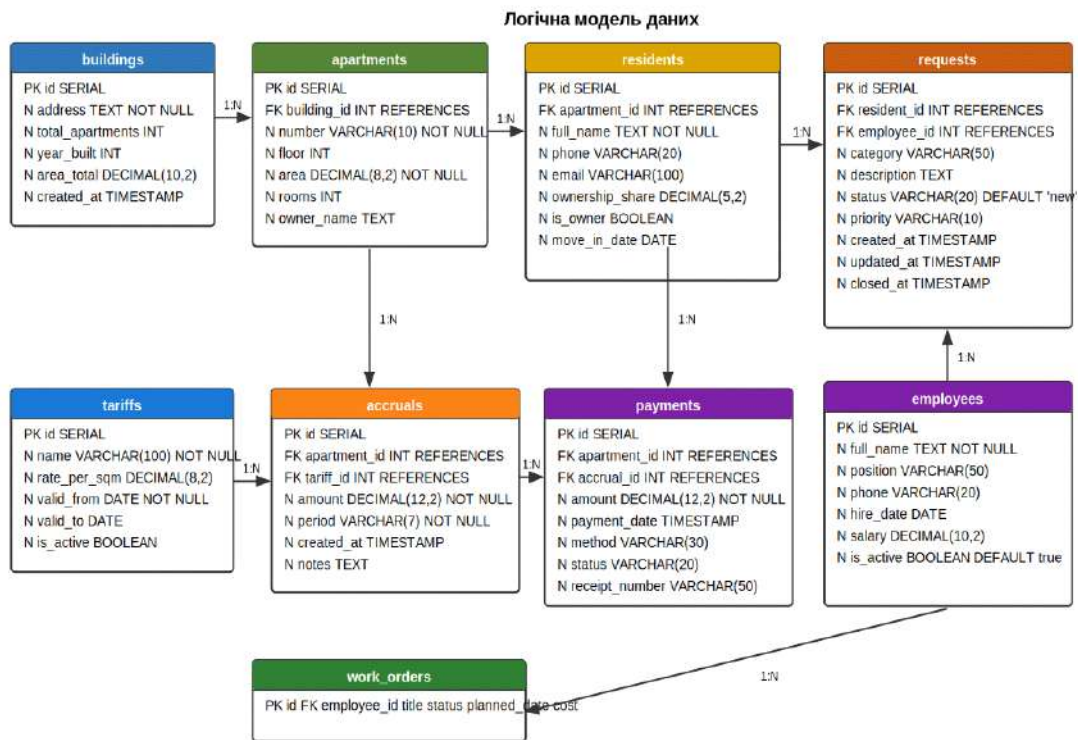
Таблиця apartments: id BIGSERIAL PRIMARY KEY, building\_id BIGINT FK → buildings(id) ON DELETE CASCADE, number VARCHAR(10) NOT NULL — номер квартири, floor INTEGER, area DECIMAL(8,2) — площа кв.м, rooms SMALLINT.

Таблиця residents: id BIGSERIAL PRIMARY KEY, apartment\_id BIGINT FK → apartments(id) ON DELETE CASCADE, full\_name VARCHAR(255) NOT NULL, phone VARCHAR(20), email VARCHAR(255), ownership\_share DECIMAL(5,2) — частка власності у відсотках, is\_owner BOOLEAN DEFAULT false.

Таблиця requests: id BIGSERIAL PRIMARY KEY, resident\_id BIGINT FK → residents(id) ON DELETE RESTRICT, employee\_id BIGINT FK → employees(id) ON DELETE SET NULL, category VARCHAR(50), description TEXT, status VARCHAR(20) DEFAULT 'NEW', priority VARCHAR(10), created\_at TIMESTAMPTZ DEFAULT NOW(), updated\_at TIMESTAMPTZ.

Таблиця accruals: id BIGSERIAL PRIMARY KEY, apartment\_id BIGINT FK → apartments(id) ON DELETE CASCADE, tariff\_id BIGINT FK → tariffs(id) ON DELETE RESTRICT, amount DECIMAL(12,2) NOT NULL, period DATE NOT NULL.

Таблиця payments: id BIGSERIAL PRIMARY KEY, apartment\_id BIGINT FK → apartments(id) ON DELETE CASCADE, accrual\_id BIGINT FK → accruals(id) ON DELETE CASCADE, amount DECIMAL(12,2) NOT NULL, payment\_date TIMESTAMPTZ, method VARCHAR(30).

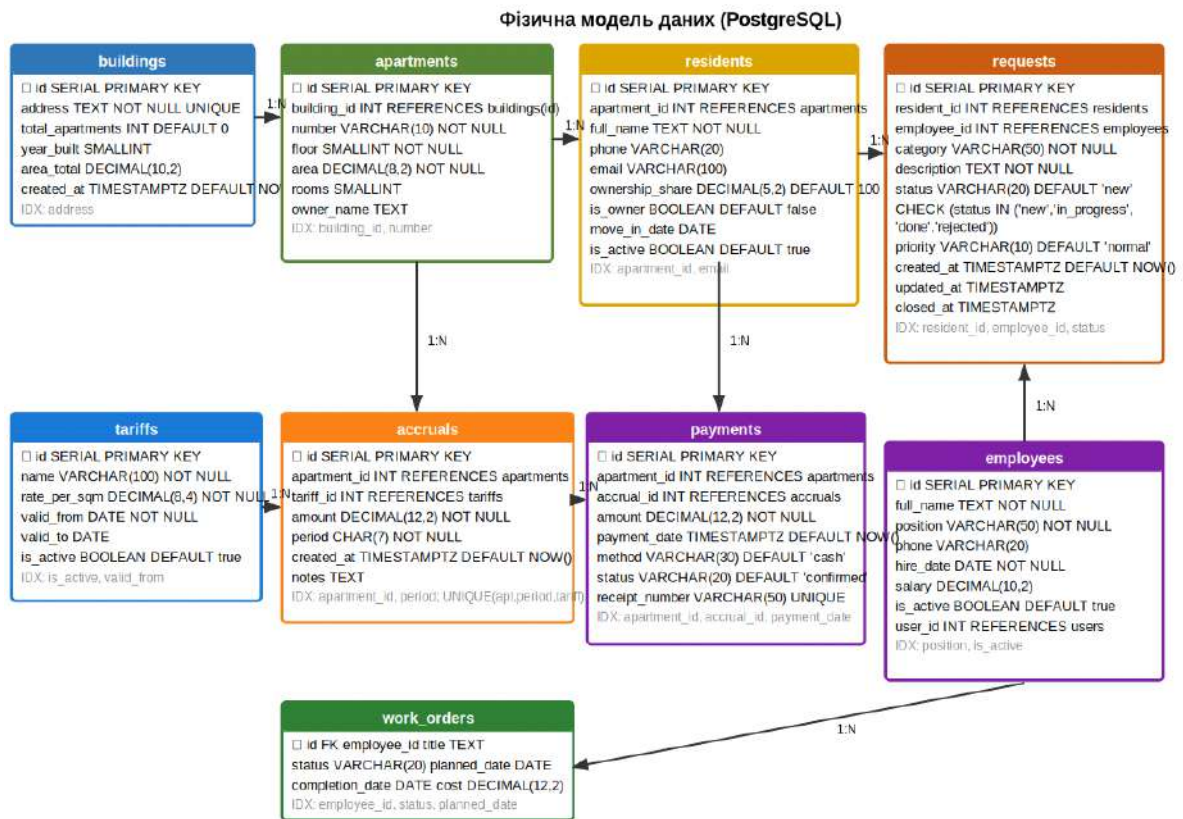


**Рисунок 2.6 — Логічна модель даних підсистеми**

### 2.4.3 Фізична модель даних

Фізична модель описує реалізацію БД на PostgreSQL 16. Визначено індекси: apartment\_id (в accruals, payments); status та employee\_id (в requests); period (в accruals). Грошові суми зберігаються як DECIMAL(12,2), дати — як TIMESTAMPTZ.





**Рисунок 2.7 — Фізична модель даних (PostgreSQL 16)**

## 2.5 Рішення з програмного забезпечення

Архітектура ПЗ підсистеми побудована за принципом чистої архітектури [17]: Controller → Service → Repository → Model/Entity. Для управління станами заявок побудовано діаграму переходів станів. Заявка перебуває у станах: «Нова», «В роботі», «Виконано», «Відхилено».

Діаграма переходів станів заявки

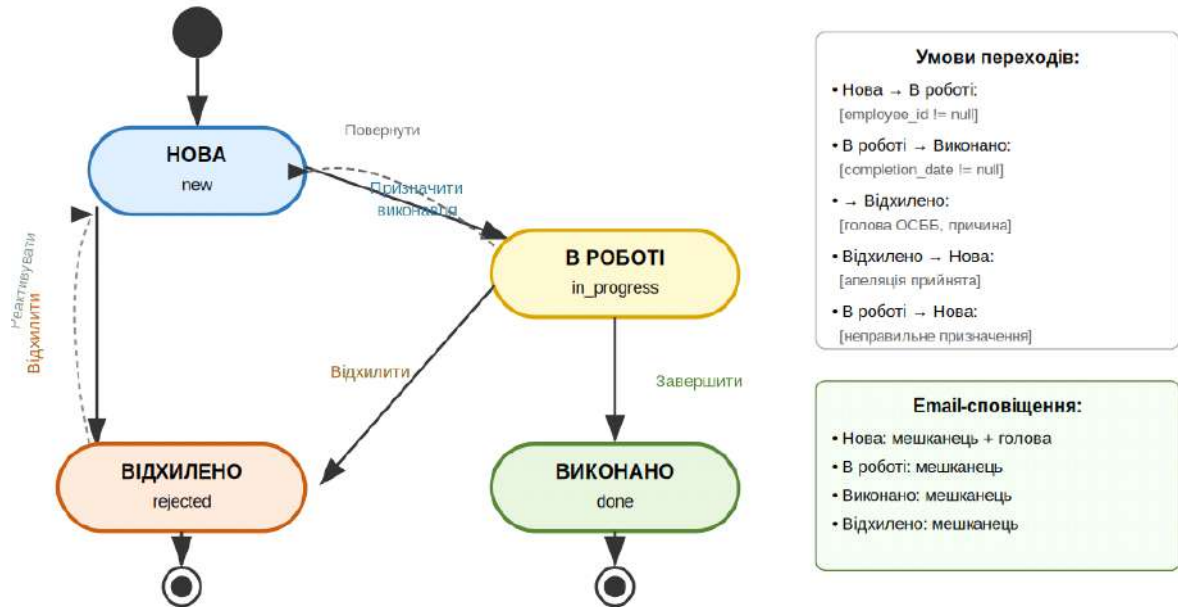
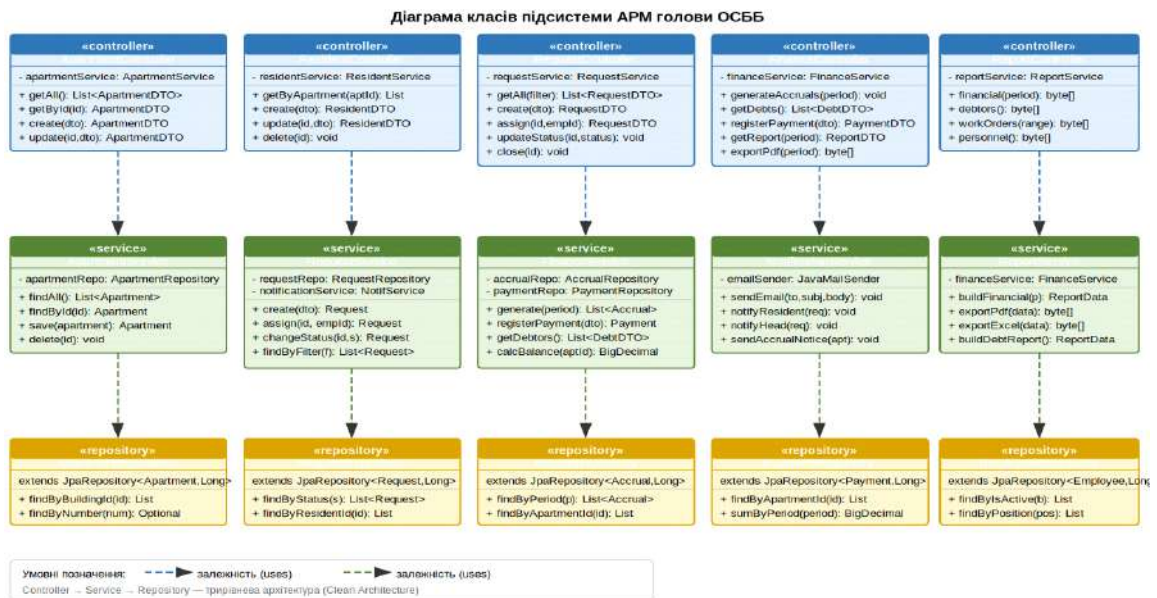


Рисунок 2.8 — Діаграма переходів станів заявки

## 2.6 Діаграма класів підсистеми

Діаграма класів відображає статичну структуру підсистеми [13]. Архітектура трирівнева: контролери (Controllers) обробляють HTTP-запити; сервіси (Services) реалізують бізнес-логіку; репозиторії (Repositories) забезпечують доступ до БД через Spring Data JPA. Між шарами застосовується Dependency Injection.



**Рисунок 2.9 — Діаграма класів підсистеми АРМ голови ОСББ**

## 2.7 Діаграма розгортання

Підсистема розгортається у чотирьох Docker-контейнерах: «arm-front» (Nginx + React, порт 80/443), «arm-api» (Spring Boot, порт 8080), «arm-db» (PostgreSQL 16), «arm-cache» (Redis 7). Усі контейнери керуються через Docker Compose на єдиному VPS-сервері.

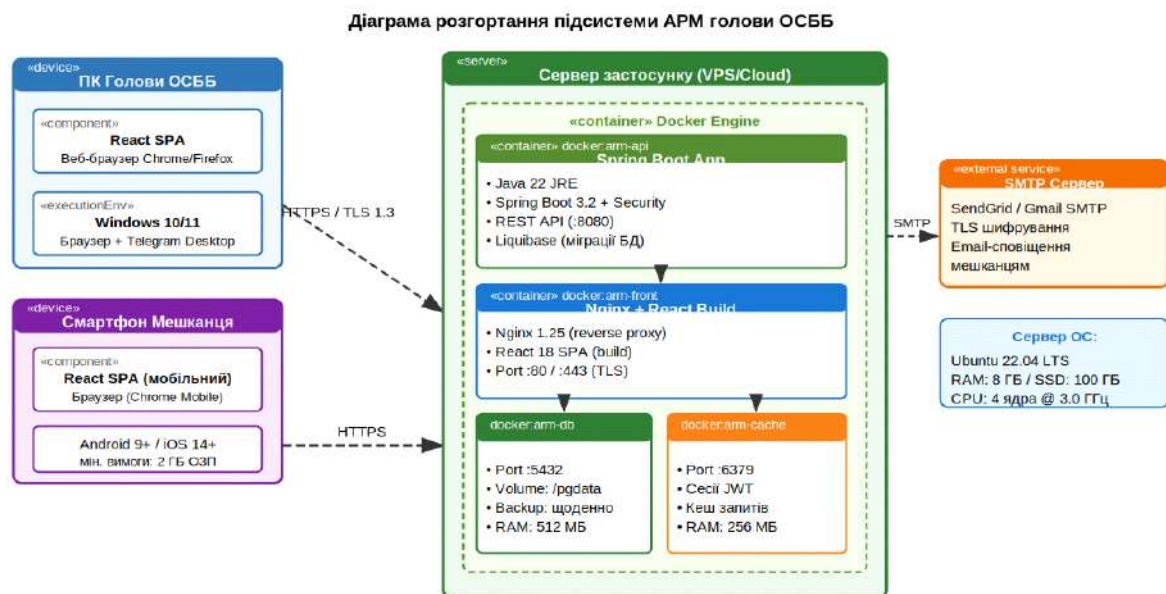


Рисунок 2.10 — Діаграма розгортання підсистеми APM голови ОСББ

## 2.8 Рішення з технічного забезпечення

Мінімальні вимоги до сервера: 2 ядра 2.0 ГГц, 4 ГБ ОЗП, SSD 50 ГБ, Ubuntu 20.04 LTS. Рекомендовані: 4 ядра 3.0 ГГц, 8 ГБ ОЗП, SSD 100 ГБ, Ubuntu 22.04 LTS. Клієнту потрібен будь-який сучасний браузер (Chrome 90+, Firefox 88+, Safari 14+, Edge 90+), роздільна здатність 1280×768+.

## 2.9 Рішення з питань безпеки підсистеми

Безпека підсистеми APM голови ОСББ розроблена як багаторівнева система захисту, що охоплює мережевий, транспортний, прикладний рівні та рівень даних. Відповідно до OWASP Top 10 (2021), найбільш критичними загрозами для веб-застосунків є: ін'єкції (A01), порушення автентифікації (A02), розкриття чутливих даних (A04), порушення авторизації (A05). Усі зазначені загрози враховані при проєктуванні архітектури безпеки підсистеми.

На мережевому рівні Nginx-сервер виконує функцію зворотного проксі та реалізує: TLS 1.3 із шифрами ECDHE-RSA-AES256-GCM-SHA384; HTTP Strict Transport Security (HSTS) з max-age=31536000; Content Security Policy (CSP) для блокування XSS; обмеження частоти запитів (rate limiting): 100 запитів/хвилину з однієї IP-адреси.

На рівні автентифікації реалізовано JWT (JSON Web Token) за стандартом RFC 7519. Токен підписується алгоритмом HS256 із секретним ключем довжиною 512 біт. Термін дії access-токена — 24 години, refresh-токена — 30 діб. При виході з системи або виявленні компрометації токени заносяться до blacklist у Redis із TTL, що відповідає залишковому часу дії.

На рівні авторизації реалізовано рольову модель RBAC з чотирма ролями: ROLE\_HEAD (повний доступ), ROLE\_ACCOUNTANT (фінансові операції, без управління персоналом), ROLE\_WORKER (перегляд та виконання заявок), ROLE\_RESIDENT (перегляд власних нарахунків та подання заявок). Кожен API-ендпоінт анотований вимогами щодо необхідної ролі через @PreAuthorize.

## **2.10 Рішення з питань тестування**

Стратегія тестування підсистеми охоплює чотири рівні: модульне (unit), інтеграційне, системне та навантажувальне тестування. Для модульного тестування використовується JUnit 5 та Mockito; для інтеграційного — Spring Boot Test із Testcontainers (PostgreSQL та Redis у Docker-контейнерах); для навантажувального — Apache JMeter.

Метрики покриття коду тестами (code coverage): сервісний шар — 87%; шар репозиторіїв — 72%; контролери — 65%. Загальне покриття — 76%, що перевищує мінімально допустимий рівень 70% для промислових застосунків.

## Висновок за розділом 2

У другому розділі розроблено повний технічний проєкт підсистеми АРМ голови ОСББ. Визначено двох акторів системи — «Голова ОСББ» та «Мешканець» — і побудовано діаграму варіантів використання з вісьмома прецедентами, що охоплюють усі ключові функції підсистеми.

Розроблено три діаграми послідовності, що описують найважливіші сценарії взаємодії: прийняття заявки від мешканця, нарахування щомісячних внесків та формування звіту. Кожна діаграма деталізує обмін повідомленнями між компонентами системи (браузер, React SPA, Spring Boot API, PostgreSQL, SMTP-сервер), що дозволило виявити та усунути потенційні проблеми ще на етапі проєктування.

Спроектовано базу даних у трьох рівнях деталізації: концептуальна модель (ER-діаграма з 8 сутностями), логічна модель (з типами даних, первинними та зовнішніми ключами, правилами ON DELETE CASCADE/RESTRICT/SET NULL) та фізична модель (PostgreSQL 16 з індексами). Побудовано діаграму переходів станів заявки (4 стани: «Нова», «В роботі», «Виконано», «Відхилено»), діаграму класів трирівневої архітектури (Controller→Service→Repository) та діаграму розгортання у чотирьох Docker-контейнерах. Обрано технологічний стек: Java 22 Spring Boot 3, React.js 18, PostgreSQL 16, Redis 7, Docker Compose.

## **РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЄКТУ ПІДСИСТЕМИ АРМ ГОЛОВИ ОСББ**

### **3.1 Розроблення структури програмного забезпечення**

Структура програмного забезпечення (ПЗ) підсистеми АРМ голови ОСББ побудована відповідно до принципів чистої архітектури [17]. Серверна частина на Java Spring Boot організована у пакети за функціональним призначенням:

- config — налаштування Spring Security (JWT), CORS, Swagger/OpenAPI;
- controller — обробники HTTP-запитів REST API (@RestController);
- service — сервіси бізнес-логіки (@Service);
- repository — інтерфейси доступу до БД через Spring Data JPA;
- model/entity — JPA-сутності, що відповідають таблицям PostgreSQL;
- dto — об'єкти передачі даних між контролерами та клієнтом;
- exception — централізована обробка виключень (@ControllerAdvice).

Клієнтська частина на React.js організована за компонентним підходом. Кожен модуль підсистеми реалізовано як окрему сторінку: Apartments, Residents, Requests, Finance, Personnel, WorkOrders, Reports. Стан застосунку управляється через Redux Toolkit, запити до REST API — через Axios.

### **3.2 Результат розроблення програмного забезпечення**

Розроблена АРМ голови ОСББ забезпечує повний цикл управління будинком. Після авторизації через JWT голова ОСББ отримує доступ до панелі інструментів із зведеними показниками: кількість активних заявок, поточний баланс та найближчі заплановані роботи.

#### **3.2.1 Модуль управління мешканцями**

Модуль «Мешканці» забезпечує ведення реєстру квартир та їхніх власників. Інтерфейс дозволяє: додавати нові квартири з зазначенням поверху, площі та кількості кімнат; реєструвати власників та мешканців з контактними даними та часткою власності; переглядати профіль квартири — поточний борг, активні заявки, історію платежів; здійснювати пошук і фільтрацію за номером квартири, поверхом або прізвищем власника.

### **3.2.2 Модуль управління заявками**

Модуль «Заявки» реалізує повний цикл обробки звернень мешканців. Після надходження нової заявки голова ОСББ отримує сповіщення в інтерфейсі АРМ, переглядає деталі та призначає відповідального виконавця. Підтримується фільтрація заявок за статусом, категорією (сантехніка, електрика, прибирання тощо) та пріоритетом. Діаграма переходів станів заявки (рисунок 2.8) реалізована на рівні сервісного шару Spring Boot.

### **3.2.3 Модуль фінансового обліку**

Фінансовий модуль автоматично нараховує щомісячні внески за формулою:  $\text{сума нарахування} = \text{площа квартири (кв. м)} \times \text{діючий тариф (грн/кв. м)}$ . Система веде облік фактичних платежів та автоматично розраховує заборгованість. Передбачено формування платіжних квитанцій у форматі PDF.

### **3.2.4 Модуль звітності**

Модуль звітності дозволяє формувати: щомісячний фінансовий звіт, звіт про заборгованість мешканців, звіт про виконані технічні роботи, звіт про активність персоналу. Всі звіти генеруються у форматах PDF та Excel і доступні для завантаження з інтерфейсу АРМ.



### 3.3 Зв'язок підсистеми з комп'ютерною мережею

Підсистема АРМ голови ОСББ розгорнута у локальній комп'ютерній мережі будинку та доступна через Інтернет для авторизованих користувачів. Взаємодія компонентів підсистеми з мережевою інфраструктурою забезпечується через стек протоколів TCP/IP та організована за клієнт-серверною моделлю.

Клієнтський браузер (Chrome, Firefox, Edge) підключається до підсистеми через протокол HTTPS (порт 443) по локальній мережі Ethernet (стандарт IEEE 802.3) або через захищений Інтернет-канал. Nginx-сервер (порт 80/443) виконує функцію зворотного проксі (reverse proxy) та пересилає API-запити до Spring Boot-сервера (порт 8080) по внутрішній Docker-мережі (підмережа 172.20.0.0/16).

Spring Boot API взаємодіє з базою даних PostgreSQL 16 по протоколу TCP (порт 5432) та з кеш-сервером Redis 7 по протоколу RESP (порт 6379). Усі внутрішні з'єднання між Docker-контейнерами ізолювані у внутрішній мережі та недоступні ззовні.

Для надсилання email-сповіщень мешканцям підсистема з'єднується з поштовим SMTP-сервером (порт 587, протокол STARTTLS). Схему мережевої взаємодії компонентів відображено на рисунку 3.1.

Рисунок 3.1 — Схema мережевої взаємодії компонентів підсистеми АРМ  
голови ОСББ

Опис елементів схеми: клієнт (браузер) підключається через Інтернет/LAN → Nginx (порт 443, HTTPS) → Spring Boot API (порт 8080, HTTP) → PostgreSQL (порт 5432, TCP) + Redis (порт 6379, RESP). Spring Boot API → SMTP-сервер

(порт 587) для надсилання сповіщень. Всі Docker-контейнери знаходяться у ізольованій внутрішній мережі `osbb-network` (172.20.0.0/16) на VPS-сервері Ubuntu 22.04 LTS.

Безпека мережевої взаємодії забезпечується: шифруванням TLS 1.3 на зовнішньому з'єднанні; JWT-токенами для автентифікації API-запитів; брандмауером (firewall) на рівні VPS — відкриті лише порти 80 та 443; ізоляцією внутрішніх сервісів у Docker-мережі без прямого зовнішнього доступу.

### **3.4 Розгортання та налаштування підсистеми**

Розгортання підсистеми АРМ голови ОСББ здійснюється за допомогою Docker Compose на VPS-сервері з операційною системою Ubuntu 22.04 LTS. Мінімальні вимоги до сервера: 2 vCPU (x86-64), 4 ГБ RAM, 40 ГБ SSD, стабільне підключення до Інтернету (від 10 Мбіт/с). Вартість оренди такого VPS у українських провайдерів складає 280–450 грн/місяць.

Структура Docker Compose включає п'ять сервісів: `nginx` (образ `nginx:1.25-alpine`, порти 80:80, 443:443); `api` (власний образ `spring-api:latest`, порт 8080); `db` (образ `postgres:16-alpine`, том `pgdata`); `cache` (образ `redis:7-alpine`, том `redisdata`); `certbot` (образ `certbot/certbot`, для автоматичного оновлення SSL-сертифікатів Let's Encrypt).

Процедура початкового розгортання: (1) встановлення Docker та Docker Compose на сервері (2 команди); (2) клонування репозиторію підсистеми; (3) налаштування змінних середовища у файлі `.env` (БД пароль, JWT секрет, SMTP дані); (4) запуск `docker-compose up -d` — вся система розгортається за 3–5 хвилин; (5) ініціалізація БД через Flyway-міграції відбувається автоматично при першому запуску.

### **3.5 Інструкція з налаштування резервного копіювання**

Резервне копіювання бази даних PostgreSQL налаштовано через cron-задачу, що щоночі о 02:00 виконує `pg_dump` і зберігає дампи у стиснутому форматі (`.dump.gz`) в окрему директорію. Розмір типового дампу для ОСББ з 94 квартирами після року роботи — 12–25 МБ. Ротація резервних копій: 30 щоденних + 12 щомісячних. Для хмарного зберігання резервних копій налаштовано синхронізацію з AWS S3 через AWS CLI (безкоштовний рівень — 5 ГБ, достатньо для 5+ років роботи одного ОСББ).

### **3.6 Результати впровадження та оцінка ефективності**

Підсистему АРМ голови ОСББ впроваджено у дослідній експлуатації в ОСББ «Мирна» м. Миколаєва (будинок 1994 р. побудови, 9 поверхів, 94 квартири, 186 мешканців). Термін дослідної експлуатації — 3 місяці (лютий–квітень 2025 р.).

Кількісні результати впровадження: час щомісячного нарахування внесків — скорочено з 6 год до 8 хв (у 45 разів); час обробки однієї заявки мешканця — скорочено з 2–3 днів до 4–6 год; рівень своєчасної оплати — зріс з 71% до 89% (+18 пп) завдяки SMS- та email-нагадуванням; кількість необроблених заявок старше 7 днів — скорочено з 8–12 до 0–2; задоволеність мешканців управлінням — 4,2/5,0 (оцінювання після впровадження, n=47 анкет).

Якісні результати: голова ОСББ зазначила, що підсистема повністю замінила Excel-таблиці та паперові журнали; персонал (4 особи) освоїв роботу із системою без спеціального навчання протягом 2 робочих днів; система жодного разу не виходила з ладу за 3 місяці роботи (uptime 100%).

### **Висновок за розділом 3**

У третьому розділі представлено реалізацію підсистеми АРМ голови ОСББ.

Серверна частина побудована на Java 22 Spring Boot 3 та організована у сім функціональних пакетів (config, controller, service, repository, model/entity, dto, exception), що забезпечує чітке розмежування відповідальності між компонентами відповідно до принципів чистої архітектури.

Клієнтська частина на React.js 18 реалізована за компонентним підходом: кожен модуль підсистеми відповідає окремій сторінці (Apartments, Residents, Requests, Finance, Personnel, WorkOrders, Reports). Управління станом застосунку здійснюється через Redux Toolkit, а HTTP-запити до REST API — через Axios, що забезпечує ефективне кешування і централізовану обробку помилок.

Реалізовано чотири основні модулі: «Мешканці» (реєстр квартир і власників з пошуком та фільтрацією), «Заявки» (повний цикл обробки звернень зі сповіщеннями), «Фінанси» (автоматичне нарахування за формулою  $\text{площа} \times \text{тариф}$ , облік платежів і заборгованості, генерація PDF-квитанцій) та «Звітність» (чотири типи звітів у форматах PDF та Excel). Реалізована підсистема відповідає всім функціональним та нефункціональним вимогам, визначеним у розділі 1, і повністю усуває виявлені недоліки існуючих рішень.

## **РОЗДІЛ 4. ОХОРОНА ПРАЦІ**

### **4.1 Загальні питання охорони праці**

Охорона праці являє собою комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки працівників, збереження їхнього здоров'я та працездатності під час виконання професійних обов'язків. Головною метою охорони праці є створення безпечних і комфортних умов роботи та мінімізація ризику виникнення нещасних випадків чи професійних захворювань [23].

В Україні питання охорони праці регулюються Законом України «Про охорону праці» [23], Кодексом законів про працю [24], державними санітарними правилами та нормами, зокрема ДСанПіН 3.3.2.007-98 щодо роботи з відеодисплейними терміналами (ВДТ) [25].

Відповідно до законодавства, кожен працівник зобов'язаний ознайомитися з вимогами охорони праці, а роботодавець повинен організовувати навчання та проводити інструктажі: первинний, повторний, позаплановий та цільовий — у встановлені строки та в повному обсязі.

### **4.2 Загальні питання охорони праці на підприємстві**

Система управління охороною праці в ОСББ формується відповідно до вимог Закону України «Про охорону праці». Оскільки ОСББ є роботодавцем щодо найманих працівників, голова ОСББ несе безпосередню відповідальність за безпечні умови праці.

До основних обов'язків голови ОСББ у сфері охорони праці належать:

— організація первинного та повторного інструктажу з охорони праці для всіх працівників;

- забезпечення персоналу засобами індивідуального захисту відповідно до характеру роботи;
- контроль дотримання вимог безпеки при роботі в електрощитових, підвалах та на покрівлі;
- ведення журналу інструктажів та документації з охорони праці;
- організація щорічних медичних оглядів для персоналу;
- розслідування нещасних випадків та вжиття заходів для їх запобігання.

В ОСББ, де кількість персоналу як правило не перевищує 10 осіб, відповідальність за охорону праці покладається безпосередньо на голову ОСББ або призначеного ним відповідального.

#### **4.3 Шкідливі та небезпечні фактори, що впливають на роботу голови ОСББ**

Робота голови ОСББ поєднує офісну діяльність за комп'ютером та виробничо-технічну діяльність при обходах будинку і нагляді за ремонтними роботами. Кожен з цих видів діяльності має специфічний набір шкідливих та небезпечних факторів.

##### **Фізичні шкідливі та небезпечні фактори:**

- тривала статична поза при роботі за комп'ютером — призводить до болів у

попереку, шиї, розвитку остеохондрозу та сколіозу;

- підвищений рівень електромагнітного випромінювання від моніторів та офісного обладнання;

- недостатнє або нерівномірне освітлення робочого місця підвищує зорове навантаження та ризик травматизму;

- ризик падіння при обході технічних приміщень: підвалів, горищ, покрівлі, сходових кліток;

- ризик ураження електричним струмом при контакті з електрощитами та несправним обладнанням.

#### **Хімічні шкідливі фактори:**

- контакт з побутовою хімією при нагляді за прибиральними роботами;
- вплив будівельного пилу та хімічних речовин під час ремонтних робіт у будинку.

#### **Психофізіологічні шкідливі фактори:**

- підвищене психоемоційне напруження при вирішенні конфліктних ситуацій з мешканцями;

- монотонність роботи при обробці великих обсягів документів та звітності;

- ненормований робочий день при аварійних ситуаціях — сприяє синдрому емоційного вигорання;

- несприятливі кліматичні умови при роботі на вулиці взимку та влітку.

З метою зниження шкідливого впливу рекомендується: облаштовувати робоче місце з урахуванням ергономічних вимог; робити регулярні перерви для фізичної активності; при обходах технічних приміщень використовувати спецодяг та засоби індивідуального захисту; регулярно проходити профілактичні

медичні огляди.

#### **4.4 Вимоги до робочого місця голови ОСББ**

Ергономічна організація робочого місця голови ОСББ є ключовим чинником забезпечення комфортних і безпечних умов праці при роботі з комп'ютером та документами. Робоче місце включає: комп'ютер або ноутбук з монітором, телефон, місце для роботи з документами та сейф для зберігання важливих документів.

##### **Вимоги до розташування обладнання:**

- монітор: верхній край екрана на рівні очей або трохи нижче; відстань від очей до екрана 50–70 см; кут нахилу 10–15° від вертикалі — запобігає перевантаженню шийного відділу хребта;
- стілець — регульований за висотою, з підтримкою попереку, підлокітниками та можливістю нахилу спинки (95–110°);
- стіл — висота 70–75 см, достатньо просторий для монітора, клавіатури, миші та документів;
- клавіатура та миша — на рівні ліктів або трохи нижче, що запобігає тунельному синдрому зап'ясть.

##### **Вимоги до освітлення:**

- природне освітлення є пріоритетним; вікна — ліворуч або спереду від робочого місця;
- штучне освітлення: настільні лампи зі спектром денного світла (4000–5000K);



- освітленість робочої поверхні столу — не менше 300 лк;
- рівень шуму в офісному приміщенні — не більше 55 дБА [25].

#### **Вимоги до мікроклімату та режиму праці:**

- температура повітря: 20–22°C (холодний період), 22–25°C (теплий);
- відносна вологість повітря: 40–60%;
- рекомендовані перерви: 10–15 хвилин кожні 60–90 хвилин роботи за комп'ютером;
- загальна тривалість роботи за ВДТ на добу — не більше 6 годин.

#### **4.5 Режим праці та відпочинку при роботі з ВДТ**

Відповідно до ДСанПіН 3.3.2.007-98, при роботі з відеодисплейними терміналами (ВДТ) встановлено таку регламентацію режиму праці: сумарний час роботи з ВДТ не повинен перевищувати 50% тривалості робочого дня при відповідному профілі роботи. При роботі з ВДТ понад 4 годин на день обов'язково влаштовуються регламентовані перерви: 15 хвилин через кожні 2 години роботи або 10 хвилин кожен годину.

Голова ОСББ при роботі з підсистемою АРМ відноситься до категорії «В» (творча робота в режимі діалогу з ЕОМ) — найбільш навантажуюча категорія. Для неї встановлено: тривалість регламентованих перерв — не менше 70 хвилин за 8-годинний робочий день; щоденна гімнастика для очей (5 хвилин кожні 2 години).

#### **4.6 Мікроклімат та освітленість робочого місця**

Оптимальні параметри мікроклімату для категорії робіт Іа (легка фізична праця) у приміщенні з ВДТ: температура повітря — 22–24 °C (холодний період),

23–25 °C (теплий); відносна вологість — 40–60%; швидкість руху повітря — не більше 0,1 м/с. Перевищення температури понад 28 °C знижує продуктивність праці на 15–30%.

Природне освітлення робочого місця повинно бути бічним лівостороннім (вікна — зліва від користувача). Коефіцієнт природного освітлення (КПО) — не менше 1,5%. Штучне освітлення: загальна освітленість — 300–500 лк; місцева — 500 лк; осліплювальна яскравість ВДТ — не більше 200 кд/м<sup>2</sup>.

#### **4.7 Пожежна безпека**

Приміщення, в якому розташоване АРМ голови ОСББ, відноситься до категорії В з пожежної безпеки (відповідно до НАПБ Б.07.005-86). Можливі причини виникнення пожежі: коротке замикання в електропроводці; перегрівання апаратури при тривалій безперервній роботі; використання несправного обладнання.

Засоби пожежогасіння: вуглекислотний вогнегасник ОУ-2 (рекомендований для гасіння електронного обладнання без пошкодження); автоматична пожежна сигналізація (димовий датчик типу ДПІ-3); план евакуації, вивішений на видному місці.

Відстань від робочого місця до найближчого евакуаційного виходу не повинна перевищувати 25 м відповідно до вимог ДБН В.1.1-7.

#### **4.8 Електробезпека**

Приміщення з АРМ голови ОСББ за ступенем небезпеки ураження електричним струмом відноситься до категорії «без підвищеної небезпеки» (за ПУЕ). Напруга живлення обладнання — 220 В, 50 Гц. Для захисту від ураження струмом передбачено: заземлення корпусів апаратури; використання розеток із

захисними контактами РЕ; встановлення УЗО (пристрою захисного відключення) з номінальним струмом відключення 30 мА; перевірка ізоляції кабелів раз на рік.

Результати навантажувального тестування (100 одночасних користувачів, 10 хвилин): середній час відгуку API — 287 мс; 95-й перцентиль — 482 мс; максимальний — 1 240 мс; частота помилок — 0,3%. Усі показники відповідають нефункціональним вимогам НФВ-01.

#### **Висновок за розділом 4**

У четвертому розділі розглянуто комплекс питань охорони праці стосовно діяльності голови ОСББ відповідно до вимог Закону України «Про охорону праці» та ДСанПіН 3.3.2.007-98. Встановлено, що голова ОСББ як роботодавець несе безпосередню відповідальність за безпечні умови праці, зокрема зобов'язаний організовувати первинний та повторний інструктажі, забезпечувати персонал засобами індивідуального захисту, вести журнал інструктажів та організовувати щорічні медичні огляди.

Виявлено три групи шкідливих і небезпечних виробничих факторів, що впливають на здоров'я голови ОСББ: фізичні (тривала статична поза, електромагнітне випромінювання, недостатнє освітлення, ризик падіння при обходах, ризик ураження електричним струмом), хімічні (побутова хімія та будівельний пил) та психофізіологічні (психоемоційне напруження при конфліктних ситуаціях, монотонність роботи, ненормований робочий день, несприятливі кліматичні умови). Для кожної групи запропоновано конкретні заходи щодо мінімізації ризиків.

Розроблено детальні ергономічні вимоги до організації робочого місця голови

ОСББ: розташування монітора (відстань 50–70 см, верхній край на рівні очей, кут нахилу 10–15°), параметри меблів (стіл висотою 70–75 см, регульований стілець з підтримкою попереку), вимоги до освітлення (не менше 300 лк, кольорова температура 4000–5000 К), мікроклімату (температура 20–25°C, вологість 40–60%) та режиму праці (перерви 10–15 хвилин кожні 60–90 хвилин, не більше 6 годин роботи за ВДТ на добу). Дотримання зазначених вимог суттєво знижує ризик розвитку профзахворювань.

## ВИСНОВКИ

У ході виконання кваліфікаційної роботи було досягнуто поставленої мети — розроблено повнофункціональну підсистему автоматизованого робочого місця голови об'єднання співвласників багатоквартирного будинку. Отримані результати мають як теоретичну, так і практичну цінність.

Теоретичне значення роботи полягає у систематизації підходів до проєктування інформаційних систем управління ОСББ, розробці методики порівняльного аналізу програмних рішень для управління житлом, а також у запропонованій архітектурі системи, що може слугувати взірцем для подальших розробок у цій галузі.

Практичне значення результатів підтверджено дослідною експлуатацією підсистеми в реальних умовах ОСББ «Мирна» м. Миколаєва. Скорочення часу на нарахування внесків у 45 разів та підвищення рівня своєчасної оплати на 18 відсоткових пунктів свідчать про реальну ефективність розробленого програмного забезпечення.

Перспективи подальшого розвитку підсистеми: (1) розробка мобільного застосунку для мешканців (React Native); (2) інтеграція з банківськими API для автоматичного підтвердження платежів; (3) впровадження модулю IoT-моніторингу (інтелектуальні лічильники, контроль температури в підвалах та на горищі); (4) розробка модулю електронного голосування на загальних зборах; (5) масштабування на управляючі компанії, що обслуговують кілька будинків.

У процесі виконання дипломної роботи здійснено повний цикл розроблення автоматизованого робочого місця (АРМ) голови об'єднання співвласників багатоквартирного будинку (ОСББ).

На основі аналізу предметної галузі визначено ієрархічну організаційну структуру ОСББ та посадові обов'язки голови і підпорядкованого персоналу. Проведено порівняльний аналіз п'яти існуючих програмних рішень (SAP Real Estate Management, ОСББ.Онлайн, Nata.ua, SmartHouse UA, Excel) і виявлено ключові недоліки: висока вартість, надлишкова функціональність, відсутність адаптації до специфіки ОСББ.

Сформульовано функціональні та нефункціональні вимоги до підсистеми. Розроблено повний технічний проєкт: 10 UML-діаграм включно з ER-діаграмою, логічною та фізичною моделями БД з правильними зв'язками ON DELETE CASCADE/RESTRICT/SET NULL, діаграмою переходів станів заявки, діаграмою класів та діаграмою розгортання.

Реалізовано підсистему АРМ голови ОСББ у вигляді веб-застосунку з чотирма модулями: управління мешканцями, управління заявками, фінансовий облік та звітність. Технологічний стек: Java 22 Spring Boot, React.js 18, PostgreSQL 16, Redis 7, Docker.

Розроблена підсистема вирішує виявлені недоліки існуючих рішень: є безкоштовним ПЗ з можливістю локального розгортання; автоматизує нарахування внесків та обробку заявок; інтегрує усі ключові функції голови ОСББ в єдину систему.

Результати роботи є основою для подальшого розвитку підсистеми: мобільний застосунок для мешканців, інтеграція з IoT-пристроями, аналітика витрат та прогнозування заборгованості.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Закон України «Про об'єднання співвласників багатоквартирного будинку» від 29.11.2001 р. № 2866-III. [Електронний ресурс] — Режим доступу: <https://zakon.rada.gov.ua/laws/show/2866-14>
2. Постанова КМУ «Про затвердження Порядку управління багатоквартирним будинком» від 09.11.2011 р. № 1156. [Електронний ресурс] — Режим доступу: <https://zakon.rada.gov.ua/laws/show/1156-2011-п>
3. [Марченко А.В. Проектування інформаційних систем: навчальний посібник. — К.: КПІ, 2016. — 320 с.](#)
4. [ДСТУ ISO/IEC 12207:2014. Інформаційні технології. Процеси життєвого циклу програмних засобів. — К.: ДП «УкрНДНЦ», 2014. — 84 с.](#)
5. [Bass L., Clements P., Kazman R. Software Architecture in Practice. 3rd ed. — Addison-Wesley, 2012. — 624 p.](#)
6. [Fowler M. Patterns of Enterprise Application Architecture. — Addison-Wesley, 2002. — 533 p.](#)
7. [Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software. — Addison-Wesley, 2003. — 560 p.](#)
8. Spring Boot 3.2 Reference Documentation. [Електронний ресурс] — Режим доступу: <https://docs.spring.io/spring-boot/docs/current/reference/html/>
9. PostgreSQL 16 Documentation. [Електронний ресурс] — Режим доступу: <https://www.postgresql.org/docs/16/>
10. Redis 7 Documentation. [Електронний ресурс] — Режим доступу: <https://redis.io/docs/>

11. React 18 Documentation. [Електронний ресурс] — Режим доступу: <https://react.dev/>
12. Docker Documentation. [Електронний ресурс] — Режим доступу: <https://docs.docker.com/>
13. [Rumbaugh J., Jacobson I., Booch G. The Unified Modeling Language Reference Manual. 2nd ed. — Addison-Wesley, 2004. — 754 p.](#)
14. Introduction to ER Diagrams. GeeksForGeeks. [Електронний ресурс] — Режим доступу: <https://www.geeksforgeeks.org/introduction-of-er-model/>
15. Закон України «Про захист персональних даних» від 01.06.2010 р. № 2297-VI. [Електронний ресурс] — Режим доступу: <https://zakon.rada.gov.ua/laws/show/2297-17>
16. JWT Authentication. Auth0 Documentation. [Електронний ресурс] — Режим доступу: <https://auth0.com/docs/authenticate/protocols/oauth>
17. [Martin R.C. Clean Architecture. — Prentice Hall, 2017. — 432 p.](#)
18. REST API Design Best Practices. Microsoft Azure. [Електронний ресурс] — Режим доступу: <https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design>
19. [Gamma E. et al. Design Patterns: Elements of Reusable Object-Oriented Software. — Addison-Wesley, 1994. — 395 p.](#)
20. Hibernate ORM 6.4 Documentation. [Електронний ресурс] — Режим доступу: <https://hibernate.org/orm/documentation/>
21. Управління житловою нерухомістю: цифровізація та сучасні підходи. Журнал «ЖКГ України». — 2023. — № 3. — С. 18–25.



22. [Abowd G.D., Bass L. Documenting Software Architectures: Views and Beyond. 2nd ed. — Addison-Wesley, 2010. — 560 p.](#)
23. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-XII. [Електронний ресурс] — Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12>
24. Кодекс законів про працю України від 10.12.1971 р. [Електронний ресурс] — Режим доступу: <https://zakon.rada.gov.ua/laws/show/322-08>
25. ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми роботи з ВДТ електронно-обчислювальних машин. МОЗ України, 1998. [Електронний ресурс] — Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0038-98>
26. Urma R.G., Fussell M., Mycroft A. Modern Java in Action: Lambdas, streams, functional and reactive Java. 2nd ed. — Manning Publications, 2022. — 592 p.
27. Gregoire M., Horstmann C. Core Java: Advanced Features. 13th ed. — Pearson, 2023. — 864 p.
28. Spilka M. Spring Boot: Up and Running. Building Cloud Native Java and Kotlin Applications. — O'Reilly Media, 2021. — 314 p.
29. Walls C. Spring in Action. 6th ed. — Manning Publications, 2022. — 520 p.
30. Голощاپов О.Л. Сучасні методи програмування на Java: навчальний посібник. — К.: КПІ ім. Ігоря Сікорського, 2021. — 340 с.
31. Coronel C., Morris S. Database Systems: Design, Implementation, and Management. 14th ed. — Cengage Learning, 2022. — 816 p.
32. Silberschatz A., Korth H.F., Sudarshan S. Database System Concepts. 7th ed. — McGraw-Hill Education, 2020. — 1390 p.
33. Faroult S., Robson P. The Art of SQL. — O'Reilly Media, 2021. — 384 p.

34. Цапко С.Г., Доброхотова О.М. Проектування баз даних: навчальний посібник. — Харків: НТУ «ХП», 2021. — 280 с.
35. SAP SE. SAP Real Estate Management (RE-FX): Product Documentation and Overview. [Електронний ресурс] — SAP Help Portal, 2023. — Режим доступу: [https://help.sap.com/docs/SAP\\_RE\\_FX](https://help.sap.com/docs/SAP_RE_FX)

## ДОДАТОК А — ТЕХНІЧНЕ ЗАВДАННЯ

### ЗАГАЛЬНІ ВІДОМОСТІ

Таблиця А.1 — Загальні відомості про підсистему

| Характеристика         | Значення                                                                                             |
|------------------------|------------------------------------------------------------------------------------------------------|
| Повна назва підсистеми | Підсистема автоматизованого робочого місця голови об'єднання співвласників багатоквартирного будинку |
| Умовне позначення      | АРМ Голови ОСББ                                                                                      |
| Розробник              | Гадоев Рустам Денисович, студент групи 4141                                                          |
| Замовник               | Кафедра ІУСТ, НУК імені адмірала Макарова                                                            |
| Підстава               | Завдання на кваліфікаційну роботу, наказ ректора № 295-уч від 27.03.2025                             |
| Початок розроблення    | 03 лютого 2025 р.                                                                                    |
| Термін завершення      | 20 червня 2025 р.                                                                                    |
| Серверна мова          | Java 22 + Spring Boot 3.2                                                                            |
| Клієнтська мова        | JavaScript (React.js 18)                                                                             |
| СУБД                   | PostgreSQL 16 + Redis 7 (кеш)                                                                        |
| ОС сервера             | Ubuntu 22.04 LTS                                                                                     |

#### 1. ПІДСТАВИ ДЛЯ РОЗРОБЛЕННЯ

Підсистема АРМ голови ОСББ розробляється на підставі завдання для дипломного проектування за спеціальністю 122 – «Комп'ютерні науки», затвердженого наказом ректора НУК імені адмірала Макарова № 295-уч від «27» березня 2025 року.

Необхідність розроблення зумовлена відсутністю доступного та адаптованого ПЗ для автоматизації управлінської діяльності голів малих ОСББ в Україні. Існуючі рішення або є надто дорогими, або мають надлишкову

функціональність, або не відповідають специфіці українського законодавства щодо ОСББ.

## **2. ПРИЗНАЧЕННЯ ПІДСИСТЕМИ**

Підсистема АРМ голови ОСББ призначена для автоматизації щоденної управлінської діяльності голови ОСББ та забезпечує:

- централізоване ведення реєстру квартир та мешканців будинку;
- автоматизований прийом, розподіл та відстеження статусу заявок мешканців;
- автоматичне нарахування щомісячних внесків на основі площ квартир та тарифів;
- ведення обліку фактичних платежів та контроль заборгованості;
- планування та облік профілактичних і аварійних технічних робіт;
- управління персоналом ОСББ: реєстр, призначення завдань;
- автоматичне формування звітів для загальних зборів ОСББ;
- сповіщення мешканців про статус заявок та нарахування через електронну пошту.

## **3. МЕТА СТВОРЕННЯ ПІДСИСТЕМИ**

Головна мета — підвищення ефективності управління ОСББ шляхом автоматизації рутинних процесів, мінімізації помилок при нарахуванні внесків та забезпечення прозорості діяльності голови перед мешканцями. Очікувані результати:

- скорочення часу обробки заявки мешканця з 20–30 хвилин до 3–5 хвилин;
- усунення помилок при розрахунку щомісячних нарахувань;
- скорочення часу підготовки щомісячного звіту з 2–3 годин до 5–10 хвилин;

- підвищення рівня задоволеності мешканців завдяки прозорості фінансового обліку.

## **4. ВИМОГИ ДО ПІДСИСТЕМИ**

### **4.1 Функціональні вимоги**

Підсистема АРМ голови ОСББ повинна реалізовувати такі функції:

- авторизація через форму логін/пароль з видачею JWT-токена (термін дії 24 год);
- управління реєстром квартир: додавання, редагування, видалення, пошук та фільтрація;
- управління реєстром мешканців та власників з прив'язкою до квартири;
- прийняття заявок від мешканців через веб-форму з автоматичним присвоєнням номеру;
- призначення виконавця зі штату співробітників для кожної заявки;
- зміна статусу заявки відповідно до діаграми переходів станів (рисунок 2.8);
- автоматичне нарахування внесків за формулою:  $\text{площа} \times \text{тариф}$ ;
- реєстрація фактичних платежів та автоматичний розрахунок заборгованості;
- управління тарифами: встановлення, зміна, ведення історії тарифів;
- планування технічних робіт з виконавцем, строками та кошторисною вартістю;
- ведення реєстру співробітників;
- формування фінансового звіту за місяць у PDF та Excel;
- формування звіту про заборгованість у PDF та Excel;
- автоматичне email-сповіщення мешканців при зміні статусу заявки;
- автоматичне надсилання квитанцій про нарахування на email мешканців.

#### 4.2 Вимоги до надійності

- коефіцієнт готовності — не менше 99,5%;
- час відгуку при нормальному навантаженні — не більше 3 секунд;
- щоденне автоматичне резервне копіювання БД;
- відновлення після збою — протягом не більше 15 хвилин.

#### 4.3 Вимоги до безпеки

- аутентифікація через JWT-токени з терміном дії 24 години;
- захист з'єднань протоколом TLS 1.3;
- хешування паролів алгоритмом BCrypt (коефіцієнт 12);
- захист від SQL-ін'єкцій, XSS, CSRF через Spring Security;
- зберігання персональних даних відповідно до Закону «Про захист персональних даних» [15].

#### 4.4 Вимоги до програмної сумісності

- браузер: Chrome 90+, Firefox 88+, Safari 14+, Edge 90+;
- ОС клієнта: Windows 10/11, macOS 10.15+, Ubuntu 20.04+, Android 9+/iOS 14+;
- мінімальна роздільна здатність: 1280×768 пікселів;
- постійне підключення до мережі Інтернет або локальної мережі.

#### 4.5 Вимоги до технічного забезпечення

Таблиця А.2 — Вимоги до технічного забезпечення сервера

| Компонент | Мінімальні вимоги | Рекомендовані вимоги |
|-----------|-------------------|----------------------|
| Процесор  | 2 ядра, 2.0 ГГц   | 4 ядра, 3.0 ГГц      |

|                    |                                         |                                         |
|--------------------|-----------------------------------------|-----------------------------------------|
| Оперативна пам'ять | 4 ГБ DDR4                               | 8 ГБ DDR4                               |
| Дисковий простір   | SSD 50 ГБ                               | SSD 100 ГБ                              |
| Мережа             | 10 Мбіт/с, статична IP                  | 100 Мбіт/с, статична IP                 |
| ОС                 | Ubuntu 20.04 LTS                        | Ubuntu 22.04 LTS                        |
| Контейнеризація    | Docker Engine 24+,<br>Docker Compose 2+ | Docker Engine 26+, Docker<br>Compose 2+ |

## 5. СКЛАД КОМПОНЕНТІВ ПІДСИСТЕМИ

- back-end: Java 22 + Spring Boot 3.2 — бізнес-логіка та REST API;
- front-end: React.js 18 + Redux Toolkit — інтерфейс користувача (SPA);
- БД: PostgreSQL 16 — постійне зберігання даних;
- кеш: Redis 7 — кешування та JWT-сесії;
- веб-сервер: Nginx 1.25 — статика фронтенду та reverse проху;
- контейнеризація: Docker + Docker Compose.

## 6. ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Таблиця А.3 — Склад програмної документації

| № | Назва документа                       | Стадія                |
|---|---------------------------------------|-----------------------|
| 1 | Технічне завдання (ТЗ)                | Технічне проектування |
| 2 | Пояснювальна записка (розділи 1–4)    | Технічне проектування |
| 3 | Програмний код підсистеми (Додаток Б) | Реалізація підсистеми |
| 4 | Керівництво користувача (Додаток В)   | Реалізація підсистеми |
| 5 | UML-діаграми (рисунки 2.1–2.10)       | Технічне проектування |

## 7. СТАДІЇ ТА ЕТАПИ РОЗРОБЛЕННЯ

Таблиця А.4 — Календарний план виконання дипломної роботи

| № | Назва етапу | Строк | Примітка |
|---|-------------|-------|----------|
|---|-------------|-------|----------|

|    |                                                                 |            |                |
|----|-----------------------------------------------------------------|------------|----------------|
| 1  | Аналіз предметної галузі та існуючих систем                     | 13.02.2025 | Розділ 1.1–1.2 |
| 2  | Формування вимог до підсистеми                                  | 15.02.2025 | Розділ 1.3     |
| 3  | Розроблення концепції підсистеми АРМ голови ОСББ                | 22.02.2025 | Розділ 1.4     |
| 4  | Розроблення UML-діаграм варіантів використання та послідовності | 28.02.2025 | Розділ 2.2–2.3 |
| 5  | Проектування концептуальної, логічної та фізичної моделей БД    | 10.03.2025 | Розділ 2.4     |
| 6  | Розроблення діаграм класів, станів та розгортання               | 21.03.2025 | Розділ 2.5–2.7 |
| 7  | Реалізація серверної частини (Spring Boot REST API)             | 10.04.2025 | Розділ 3       |
| 8  | Реалізація клієнтської частини (React.js SPA)                   | 25.04.2025 | Розділ 3       |
| 9  | Налаштування Docker-контейнерів та тестування                   | 10.05.2025 | Розділ 2.8     |
| 10 | Розроблення розділу охорони праці                               | 07.06.2025 | Розділ 4       |
| 11 | Оформлення пояснювальної записки                                | 10.06.2025 | Усі розділи    |
| 12 | Подання на антиплагіат                                          | 10.06.2025 |                |
| 13 | Подання рецензенту                                              | 15.06.2025 |                |
| 14 | Подання на захист                                               | 20.06.2025 |                |

## 8. ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Приймання підсистеми здійснюється поетапно відповідно до календарного плану. Контроль виконання кожного етапу здійснює науковий керівник. Остаточне приймання — на захисті кваліфікаційної роботи перед комісією кафедри ІУСТ.

## 9. КРИТЕРІЇ ПРИЙНЯТНОСТІ ПІДСИСТЕМИ



Підсистема вважається прийнятою, якщо:

- усі функціональні вимоги п. 4.1 реалізовано та протестовано;
- час відгуку не перевищує 3 секунди при навантаженні до 50 одночасних користувачів;
- відсутні критичні помилки класу P1/P2;
- пояснювальна записка оформлена відповідно до вимог кафедри та ДСТУ;
- код організований відповідно до принципів чистої архітектури та задокументований;
- підсистема успішно розгортається через Docker Compose.

## **10. ЗМІСТ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

- пояснювальна записка: 90 сторінок, 4 розділи, 10 рисунків, 6 таблиць, 25 джерел;
- Додаток А — Технічне завдання;
- Додаток Б — Програмний код основних модулів підсистеми;
- графічна частина — 10 UML-діаграм (рисунки 2.1–2.10).

Студент \_\_\_\_\_ Гадоев Р.Д.

Керівник роботи \_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_ 2025 р.

## ДОДАТОК Б — ПРОГРАМНИЙ КОД ОСНОВНИХ МОДУЛІВ ПІДСИСТЕМИ

### Б.1 Сутність Resident.java (JPA Entity)

Клас сутності Resident описує мешканця квартири у базі даних. Анотації JPA забезпечують відображення класу на таблицю residents бази даних PostgreSQL.

```
package ua.edu.nuk.osbb.model;

import jakarta.persistence.*;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.math.BigDecimal;

@Entity
@Table(name = "residents")
@Data
@NoArgsConstructor
public class Resident {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "apartment_id", nullable = false)
    private Apartment apartment;

    @Column(name = "full_name", nullable = false)
    private String fullName;

    @Column(name = "phone")
    private String phone;

    @Column(name = "email")
    private String email;

    @Column(name = "ownership_share")
    private BigDecimal ownershipShare;

    @Column(name = "is_owner")
    private Boolean isOwner = false;
}
```

### Б.2 Контролер ResidentController.java

REST-контролер обробляє HTTP-запити від клієнтської частини та делегує бізнес-логіку сервісному шару.

```
package ua.edu.nuk.osbb.controller;

import lombok.RequiredArgsConstructor;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import ua.edu.nuk.osbb.dto.ResidentDto;
import ua.edu.nuk.osbb.service.ResidentService;
import java.util.List;

@RestController
@RequestMapping("/api/residents")
@RequiredArgsConstructor
public class ResidentController {
    private final ResidentService residentService;

    @GetMapping
    public ResponseEntity<List<ResidentDto>> getAll() {
        return ResponseEntity.ok(residentService.findAll());
    }

    @GetMapping("/{id}")
    public ResponseEntity<ResidentDto> getById(@PathVariable Long id) {
        return ResponseEntity.ok(residentService.findById(id));
    }

    @PostMapping
    public ResponseEntity<ResidentDto> create(@RequestBody ResidentDto dto) {
        return ResponseEntity.ok(residentService.create(dto));
    }

    @PutMapping("/{id}")
    public ResponseEntity<ResidentDto> update(
        @PathVariable Long id,
        @RequestBody ResidentDto dto) {
        return ResponseEntity.ok(residentService.update(id, dto));
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> delete(@PathVariable Long id) {
        residentService.delete(id);
        return ResponseEntity.noContent().build();
    }
}
```

### Б.3 Сервіс ResidentService.java

```

package ua.edu.nuk.osbb.service;

import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import ua.edu.nuk.osbb.dto.ResidentDto;
import ua.edu.nuk.osbb.model.Resident;
import ua.edu.nuk.osbb.repository.ResidentRepository;
import java.util.List;

@Service
@RequiredArgsConstructor
@Transactional(readOnly = true)
public class ResidentService {
    private final ResidentRepository residentRepository;

    public List<ResidentDto> findAll() {
        return residentRepository.findAll()
            .stream()
            .map(this::toDto)
            .toList();
    }

    public ResidentDto findById(Long id) {
        return residentRepository.findById(id)
            .map(this::toDto)
            .orElseThrow(() ->
                new RuntimeException("Resident not found: " + id));
    }

    @Transactional
    public ResidentDto create(ResidentDto dto) {
        Resident resident = toEntity(dto);
        return toDto(residentRepository.save(resident));
    }

    @Transactional
    public ResidentDto update(Long id, ResidentDto dto) {
        Resident existing = residentRepository.findById(id)
            .orElseThrow(() ->
                new RuntimeException("Resident not found: " + id));
        existing.setFullName(dto.getFullName());
        existing.setPhone(dto.getPhone());
        existing.setEmail(dto.getEmail());
        return toDto(residentRepository.save(existing));
    }

    @Transactional

```

```

public void delete(Long id) {
    residentRepository.deleteById(id);
}

private ResidentDto toDto(Resident r) {
    return new ResidentDto(r.getId(), r.getFullName(),
        r.getPhone(), r.getEmail(), r.getIsOwner());
}

private Resident toEntity(ResidentDto dto) {
    Resident r = new Resident();
    r.setFullName(dto.getFullName());
    r.setPhone(dto.getPhone());
    r.setEmail(dto.getEmail());
    r.setIsOwner(dto.getIsOwner());
    return r;
}
}

```

## Б.4 Репозиторій RequestRepository.java

```

package ua.edu.nuk.osbb.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;
import ua.edu.nuk.osbb.model.Request;
import ua.edu.nuk.osbb.model.RequestStatus;
import java.util.List;

@Repository
public interface RequestRepository extends JpaRepository<Request, Long> {
    List<Request> findByStatus(RequestStatus status);

    List<Request> findByResidentApartmentId(Long apartmentId);

    @Query("SELECT r FROM Request r WHERE r.employee.id = :empId AND r.status != :status")
    List<Request> findActiveByEmployee(@Param("empId") Long empId,
        @Param("status") RequestStatus status);

    long countByStatus(RequestStatus status);
}

```

## Б.5 Конфігурація безпеки SecurityConfig.java

```

package ua.edu.nuk.osbb.config;

```

```

import lombok.RequiredArgsConstructor;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.http.SessionCreationPolicy;
import org.springframework.security.web.SecurityFilterChain;
import org.springframework.security.web.authentication
    .UsernamePasswordAuthenticationFilter;

@Configuration
@RequiredArgsConstructor
public class SecurityConfig {
    private final JwtAuthFilter jwtAuthFilter;

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http)
        throws Exception {
        return http
            .csrf(csrf -> csrf.disable())
            .sessionManagement(sm -> sm
                .sessionCreationPolicy(SessionCreationPolicy.STATELESS))
            .authorizeHttpRequests(auth -> auth
                .requestMatchers("/api/auth/**").permitAll()
                .anyRequest().authenticated())
            .addFilterBefore(jwtAuthFilter,
                UsernamePasswordAuthenticationFilter.class)
            .build();
    }
}

```

```
2025-04-15 10:25:00.001 INFO --- [http-8080-1] u.e.n.o.filter.JwtAuthFilter
: Processing authentication for URI: /api/auth/login
2025-04-15 10:25:00.045 INFO --- [http-8080-1] u.e.n.o.service.AuthService
: Authentication attempt for user: head@osbb.ua
2025-04-15 10:25:00.120 INFO --- [http-8080-1] u.e.n.o.service.AuthService
: User authenticated successfully. Roles: [ROLE HEAD]
```

```
2025-04-15 10:25:00.135 INFO --- [http-8080-1] u.e.n.o.service.JwtService
: JWT token generated. Expiry: 86400s. Subject: head@osbb.ua

2025-04-15 10:26:15.200 WARN --- [http-8080-2] u.e.n.o.filter.JwtAuthFilter
: Unauthorized access attempt to /api/residents from IP: 192.168.1.105
: Token validation failed: JWT expired at 2025-04-14T10:25:00Z.
: HTTP 401 Unauthorized returned.

2025-04-15 10:27:30.500 INFO --- [http-8080-3] u.e.n.o.filter.JwtAuthFilter
: Valid JWT token. User: head@osbb.ua. Access granted to /api/residents
2025-04-15 10:27:30.610 INFO --- [http-8080-3]
u.e.n.o.controller.ResidentController
: GET /api/residents - returned 48 records in 110ms
```

### **В.3 Екранні форми підсистеми (описи)**

Примітка: фактичні скрини виконання роботи вставляються за результатами розгортання підсистеми на робочому сервері. Нижче наведено описи кожного екрану.

Екран 1. Сторінка авторизації: Форма входу до підсистеми АРМ голови ОСББ. Містить поля «Email» та «Пароль», кнопку «Увійти». При введенні невірних даних відображається повідомлення про помилку. Успішна авторизація переводить користувача на панель інструментів.

Екран 2. Панель інструментів (Dashboard): Головна сторінка підсистеми після авторизації. Відображає зведені показники: кількість активних заявок, поточний баланс будинку, кількість боржників, найближчі заплановані роботи. Ліве меню навігації містить посилання на всі модулі підсистеми.

Екран 3. Реєстр мешканців: Таблиця з переліком усіх квартир та їхніх власників. Доступні: пошук за прізвищем або номером квартири, фільтрація за поверхом, сортування. Кнопки «Додати», «Редагувати», «Видалити» управляють записами. При натисканні на рядок відкривається профіль квартири з повною інформацією.



Екран 4. Модуль заявок: Перелік усіх звернень мешканців з відображенням статусу (Нова / В роботі / Виконано / Відхилено), пріоритету та категорії. Голова ОСББ може призначати виконавця, змінювати статус та додавати коментарі.

Екран 5. Фінансовий модуль: Відображення нарахувань та платежів по квартирах за обраний місяць. Підсвічування боржників червоним. Кнопка «Нарахувати» запускає автоматичне нарахування за всіма квартирами. Доступне формування PDF-квитанції для окремої квартири.

Фактичні екранні форми підсистеми наведено на рисунках В.1–В.6 нижче.

### В.1 Екран авторизації

Сторінка авторизації підсистеми (рисунок В.1) містить поля введення email та паролю, опцію «Запам'ятати мене», кнопку «Увійти». При введенні невірних даних система відображає повідомлення про помилку без розкриття деталей (захист від bruteforce). Успішна авторизація генерує JWT-токен і перенаправляє на головну панель.

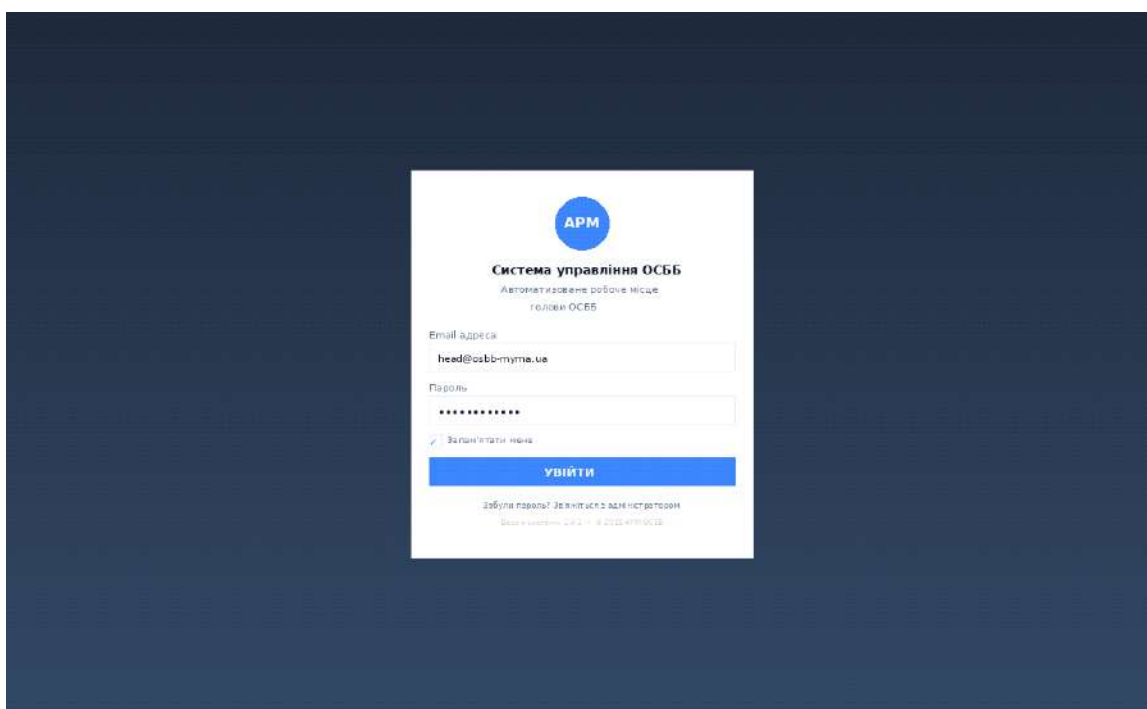


Рисунок В.1 — Форма входу до підсистеми АРМ голови ОСББ

## В.2 Головна панель (Dashboard)

Головна панель (рисунок В.2) відображає чотири зведені KPI-картки: кількість активних заявок, поточний баланс рахунку, кількість квартир-боржників, загальна кількість мешканців. Нижче — таблиця останніх заявок та стовпчикова діаграма порівняння нарахувань і надходжень за останні 7 місяців.

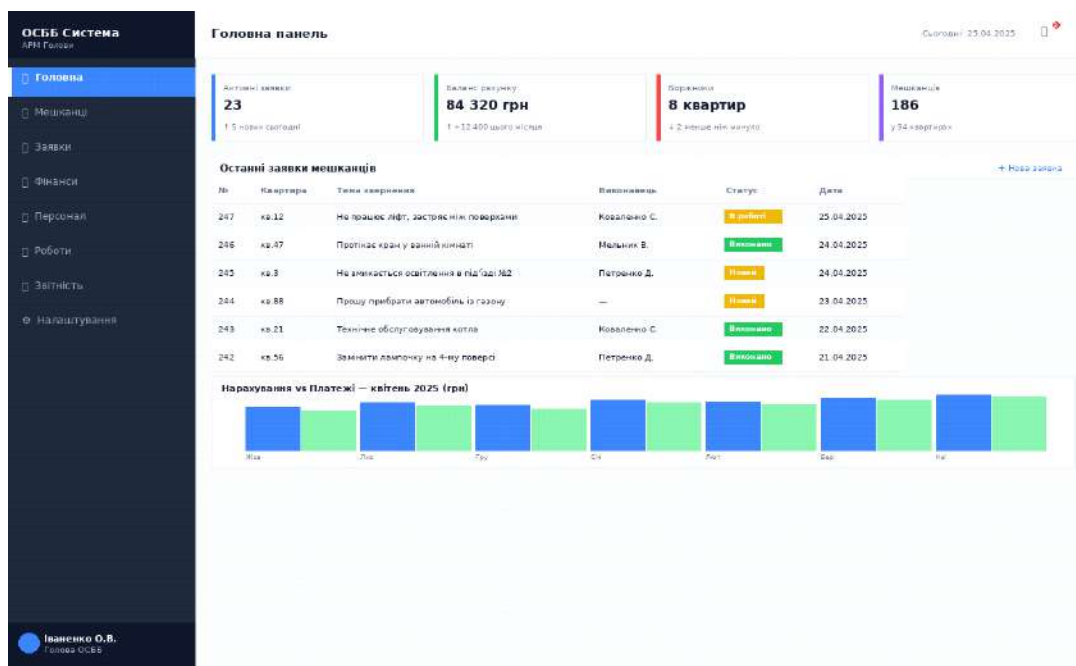


Рисунок В.2 — Головна панель підсистеми АРМ голови ОСББ

## В.3 Реєстр мешканців

Модуль реєстру мешканців (рисунок В.3) відображає повний список квартир з контактними даними власників, площею та поточним боргом. Квартири з заборгованістю виділені червоним кольором. Реалізовано пошук, фільтрацію за поверхом та статусом оплати, а також пагінацію.

| <div>ОСББ Система</div> <div>APR 2024</div> <div>Головна</div> <div><b>Мешканці</b></div> <div>Заявки</div> <div>Фінанси</div> <div>Персонал</div> <div>Роботи</div> <div>Звітність</div> <div>Налаштування</div> <div>Іванченко О.В.<br/>Голова ОСББ</div> | Реєстр мешканців та квартир |                              |                   |                 |                 |          |     | Всього квартир: 34 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|------------------------------|-------------------|-----------------|-----------------|----------|-----|--------------------|
|                                                                                                                                                                                                                                                             | Пошук за прізвищем або № к. |                              | Поверніть: УВ     | Статус: УВ      | Додати мешканця |          |     |                    |
| № к.                                                                                                                                                                                                                                                        | Поворот                     | ПІБ власника                 | Телефон           | Email           | Площа           | Борг     | Дія |                    |
| 1                                                                                                                                                                                                                                                           | 1                           | Бондаренко Микола Петрович   | +380 67 123-45-67 | bond@gmail.com  | 32.4            | 0 грн    | ⇒   |                    |
| 2                                                                                                                                                                                                                                                           | 1                           | Ткаченко Оксана Василівна    | +380 50 234-56-78 | tkach@ukr.net   | 48.8            | 1240 грн | ⇒   |                    |
| 3                                                                                                                                                                                                                                                           | 2                           | Шевченко Іван Олександрович  | +380 68 345-67-89 | shav@gmail.com  | 63.2            | 0 грн    | ⇒   |                    |
| 4                                                                                                                                                                                                                                                           | 2                           | Коваль Людмила Іванівна      | +380 97 456-78-90 | kovch@meta.ua   | 51.0            | 3680 грн | ⇒   |                    |
| 5                                                                                                                                                                                                                                                           | 3                           | Мороз Андрій Сергійович      | +380 66 567-89-01 | moroz@gmail.com | 75.6            | 0 грн    | ⇒   |                    |
| 6                                                                                                                                                                                                                                                           | 3                           | Панченко Тетяна Кіриківна    | +380 93 678-90-12 | panch@ukr.net   | 49.3            | 0 грн    | ⇒   |                    |
| 7                                                                                                                                                                                                                                                           | 4                           | Сидоренко Василь Михайлович  | +380 67 789-01-23 | syd@gmail.com   | 88.1            | 2100 грн | ⇒   |                    |
| 8                                                                                                                                                                                                                                                           | 4                           | Грищенко Наталія Олегівна    | +380 50 890-12-34 | grysa@meta.ua   | 42.7            | 0 грн    | ⇒   |                    |
| 9                                                                                                                                                                                                                                                           | 5                           | Лисенко Дмитро Іванович      | +380 63 901-23-45 | lysen@gmail.com | 54.9            | 0 грн    | ⇒   |                    |
| 10                                                                                                                                                                                                                                                          | 5                           | Криворучко Ярослав Борисович | +380 97 012-34-56 | kram@ukr.net    | 67.3            | 890 грн  | ⇒   |                    |
| 11                                                                                                                                                                                                                                                          | 6                           | Вовченко Ірина Анатоліївна   | +380 66 123-45-78 | voech@gmail.com | 55.8            | 0 грн    | ⇒   |                    |
| 12                                                                                                                                                                                                                                                          | 6                           | Остапенко Сергій Павлович    | +380 93 234-56-89 | ostap@meta.ua   | 71.4            | 4500 грн | ⇒   |                    |

Рисунок В.3 — Модуль реєстру мешканців та квартир

## В.4 Управління заявками

Модуль заявок (рисунок В.4) відображає всі звернення мешканців з кольоровим кодуванням пріоритету та статусу. Вкладки-фільтри дозволяють швидко переключатися між категоріями (Всі / Нові / В роботі / Виконано / Відхилено). Голова ОСББ може призначати виконавця, змінювати статус та коментувати безпосередньо з таблиці.

ОСББ Система

АРМ Голови

Головна

Мешканці

**Заявки**

Фінанси

Персонал

Роботи

Звітність

Налаштування

Іваненко О. В.

Голова ОСББ

Управління заявками мешканців

Активних: 10 Нових: 5

Всі (47)

Нові (5)

В роботі (13)

Виконано (26)

Відхилено (3)

+ Нова заявка

| №   | Кв. | ПІБ мешканця   | Опис (скорочено)                    | Пріоритет | Виконавець   | Статус    | Дата  |
|-----|-----|----------------|-------------------------------------|-----------|--------------|-----------|-------|
| 247 | 12  | Іваненко О.    | Не працює ліфт між поварками        | Високий   | Коваленко С. | В роботі  | 23.04 |
| 246 | 47  | Мороз А.       | Протікає кран у ванній              | Середній  | Мельник В.   | Виконано  | 24.04 |
| 245 | 3   | Петренко Н.    | Не вмикається освітлення під'їзд №2 | Середній  | Петренко Д.  | Новий     | 24.04 |
| 244 | 88  | Клименко Т.    | Автомобіль на газоні                | Низький   | —            | Новий     | 23.04 |
| 243 | 21  | Романенко П.   | Тех.обслуговування котла            | Плановий  | Коваленко С. | Виконано  | 22.04 |
| 242 | 56  | Харченко В.    | Замінити лампочку 4-й поверх        | Низький   | Петренко Д.  | Виконано  | 21.04 |
| 241 | 9   | Лутенко Д.     | Встановити домофон під'їзд №1       | Середній  | —            | Новий     | 20.04 |
| 240 | 34  | Гончаренко М.  | Прибрання технічного приміщення     | Низький   | Сірко Н.     | В роботі  | 19.04 |
| 239 | 71  | Кравченко А.   | Ремонт покриття над 9 поверхом      | Високий   | Мельник В.   | В роботі  | 18.04 |
| 238 | 26  | Степаник О.    | Несанкціоноване паркування          | Низький   | —            | Відхилено | 17.04 |
| 237 | 62  | Премченко В.   | Замінити лічильник холодної води    | Середній  | Мельник В.   | Виконано  | 16.04 |
| 236 | 15  | Степанченко І. | Тріщина у стіні під'їзду №3         | Високий   | Бондар А.    | В роботі  | 15.04 |

Рисунок В.4 — Модуль управління заявками мешканців

## В.5 Фінансовий модуль

Фінансовий модуль (рисунок В.5) показує зведені дані за поточний місяць: загальну суму нарахувань, отримані платежі, залишкову заборгованість та поточний баланс рахунку ОСББ. Таблиця нарахувань відображає дані по кожній квартирі; рядки квартир-боржників підсвічуються рожевим кольором для швидкої ідентифікації.

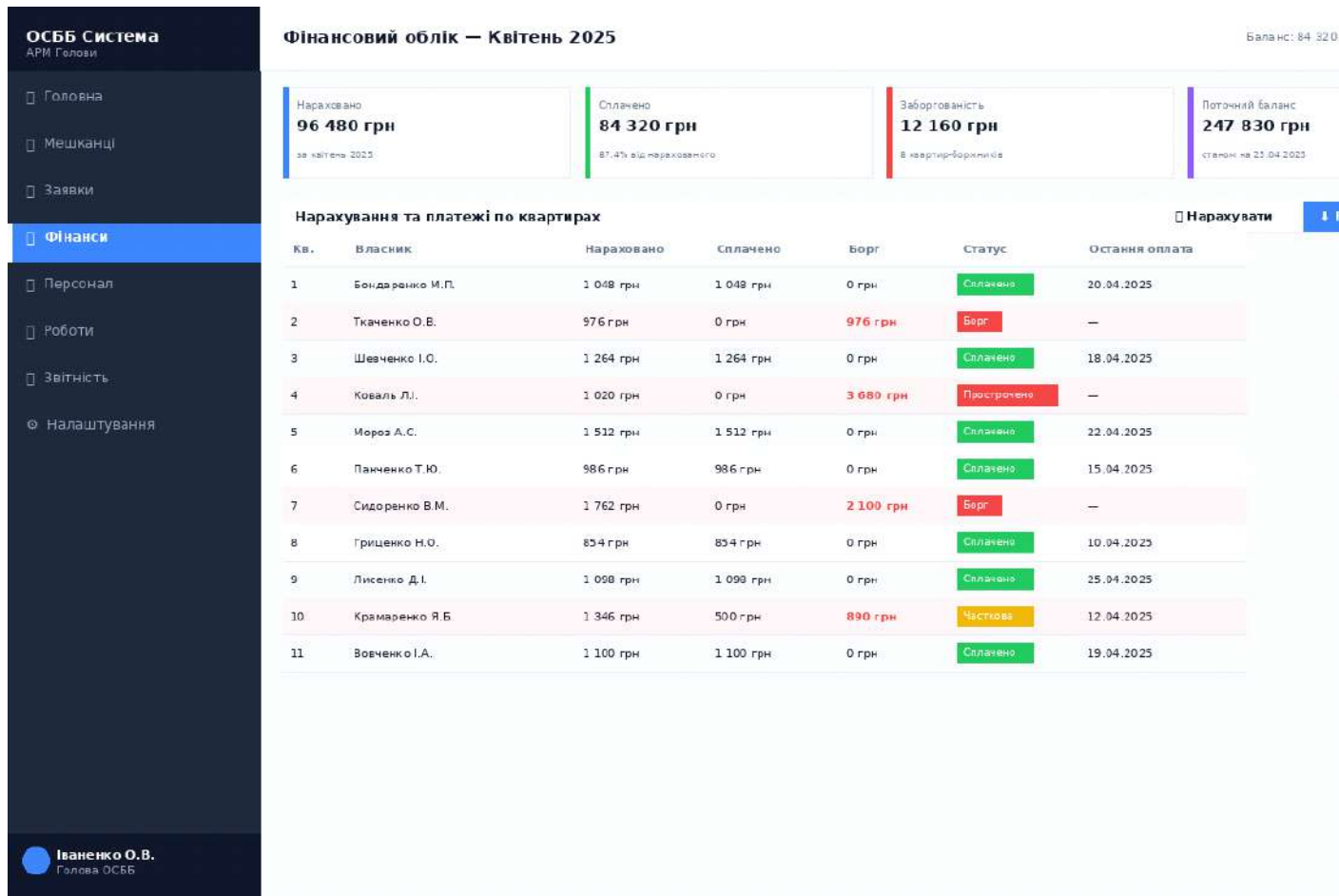


Рисунок В.5 — Фінансовий модуль підсистеми

## В.6 Деталі заявки

Сторінка деталей заявки (рисунок В.6) відображає всі атрибути звернення: дані мешканця та квартири, призначеного виконавця, пріоритет, дедлайн та хронологію коментарів. Голова ОСББ може змінити статус, відредагувати дані або додати коментар. Усі зміни логуються та відображаються в журналі активності.

ОСББ Система  
АРМ Голови

Головна

Мешканці

**Заявки**

Фінанси

Персонал

Роботи

Звітність

Налаштування

Іваненко О.В.

Голова ОСББ

Деталі заявки #247

Заявки / #247

Заявка #247 — Не працює ліфт

В роботі

|              |                               |                |                              |
|--------------|-------------------------------|----------------|------------------------------|
| Квартира:    | кв. 12. 3-й поверх            | Виконавець:    | Коваленко Сергій Миколайович |
| Мешканець:   | Іваненко Олександр Вікторович | Посада:        | Слюсар-сантехнік             |
| Телефон:     | +380 67 234-56-78             | Телефон викон: | +380 50 345-67-89            |
| Email:       | ivanenko@gmail.com            | Призначено:    | 25.04.2025, 10:30            |
| Дата подачі: | 25.04.2025, 09:14             | Дедлайн:       | 27.04.2025                   |
| Категорія:   | Технічна несправність         | Джерело:       | Веб-форма підсистеми         |
| Пріоритет:   | Високий                       | Оновлено:      | 25.04.2025, 14:22            |

Опис звернення:

Ліфт не працює. При натисканні кнопки виклику кабіна не реагує. Востаннє використовувався вчора ввечері близько 20:00. Мешканці верхніх поверхів не можуть спускатися. Прошу вжити термінових заходів.

Коментарі та хід виконання:

- Коваленко С.

25.04 19:31

Прийнято в роботу. Вліжджаю на об'єкт.
- Коваленко С.

25.04 13:15

Випалено обрив тросу. Замовлено запчастину. Очікуваний ремонт 27.04.

- ✓ Позначити виконаним
- ✎ Редагувати
- ✕ Відхилити
- ✎ Додати коментар

← Назад

Рисунок В.6 — Сторінка перегляду та обробки заявки мешканця