

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет      Магістерської підготовки

Кафедра      Інформаційних технологій

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему: Розробка інформаційної системи управління фінансовою діяльністю  
житлових комплексів

Виконав студент 2 курсу групи МІС-19  
спеціальності 122 Комп'ютерні науки  
Симоненко Олександр Євгенійович

Керівник к.т.н., доцент  
Гнатовська Ганна Арнольдівна

Рецензент к.геогр.н., доцент  
Кузніченко Світлана Дмитрівна

Одеса 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської підготовки  
Кафедра Інформаційних технологій  
Рівень вищої освіти магістр  
Спеціальність 122 Комп'ютерні науки  
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри \_\_\_\_\_

“ 26 ” жовтня 2020 р.

З А В Д А Н Н Я  
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Симоненко Олександр Свгенійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інформаційної системи управління фінансовою діяльністю житлових комплексів

керівник роботи Гнатовська Ганна Арнольдівна, к.т.н., доцент,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “ 16 ” жовтня 2020р. №194 «С»

2. Строк подання студентом роботи 7 грудня 2020р.

3. Вихідні дані до роботи Ресурси мережі Інтернет, програмна платформа Angular, CSS-фреймворк Bootstrap3, Module bundler – Webpack, програмна платформа Node.js, технологія Single Page Application

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз та визначення функціональних вимог і технологій розробки інформаційної системи. Вирішення завдання вибору архітектурних рішень для реалізації системи. Проектування та вибір програмних засобів реалізації системи. Програмна реалізація ІС управління фінансовою діяльністю житлових комплексів

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

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 26 жовтня 2020р.

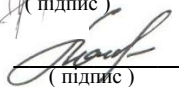
## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1	Дослідження особливостей управління фінансовою діяльністю житлових комплексів	26.10.20 – 30.10.20	92%	відмін.
2	Вирішення завдання вибору архітектурних рішень для реалізації системи	30.10.20 – 02.11.20	92%	відмін.
3	Дослідження та вибір програмної платформи реалізації системи засобами теорії прийняття рішень	03.11.20 – 05.11.20	92%	відмін.
4	Проектування системи за методологією функціонального моделювання SADT	06.11.20 – 08.11.20	94%	відмін.
5	Проектування бази даних системи	08.11.20 – 10.11.20	92%	відмін.
6	Програмна реалізація ІС управління фінансовою діяльністю житлових комплексів	10.11.20 – 02.12.20	92%	відмін.
7	Рубіжна атестація	19.11.20	92%	відмін.
8	Оформлення пояснювальної записки та презентації	03.12.20 – 06.12.20	95%	відмін.
9	Подання роботи на кафедру	07.12.20		
10	Перевірка на плагіат	08.12.20		
11	Рецензування	16.12.20		
	<b>Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)</b>		<b>92%</b>	<b>відмін.</b>

Студент

  
(підпис)

Керівник роботи

  
(підпис)

Симоненко О.Є.

(прізвище та ініціали)

Гнатовська Г.А.

(прізвище та ініціали)

## АНОТАЦІЯ

Тема магістерської роботи «Розробка інформаційної системи управління фінансовою діяльністю житлових комплексів».

Актуальність магістерської роботи полягає в необхідності розробки програмного забезпечення, яке забезпечить співвласників житлових комплексів зручним засобом для управління та відстеженням всіх фінансових операцій, що надасть користувачам можливість здійснення обліку фінансово-господарських операцій, що відбуваються у житловому комплексі.

Об'єкт дослідження – процеси проектування та розробки інформаційної системи управління фінансовою діяльністю, головним завданням якої є надання користувачам можливості здійснення послуг з оплати та відстеження всіх фінансових операцій співвласників житлових комплексів.

Предмет дослідження – інформаційні технології розробки та програмної реалізації системи управління фінансовою діяльністю житлових комплексів.

Мета роботи – проектування та розробка інформаційної системи управління фінансовою діяльністю житлових комплексів, що надає користувачам актуальну фінансову інформацію щодо здійснення нарахування і оплати отриманих комунальних послуг, планування майбутніх витрат та фінансові звіти про фінансово-господарську діяльність.

Практична цінність магістерської роботи полягає в тому, що розроблена система може бути використана для управління фінансовою діяльністю будь-якого зацікавленого об'єднання співвласників житлового комплексу.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, ФІНАНСОВА ДІЯЛЬНІСТЬ, ЖИТЛОВИЙ КОМПЛЕКС.

Магістерська робота містить 77 сторінок, 11 таблиць, 19 рисунків, 15 посилань.

## ANOTATION

The theme of the master's work is "Development of an information system for managing the financial activities of residential complexes."

The relevance of the master's thesis is the need to develop software that will provide co-owners of residential complexes with a convenient tool for managing and tracking all financial transactions, which will give users the opportunity to record financial and economic transactions in the residential complex.

The object of research is the processes of design and development of information management system of financial activities, the main task of which is to provide users with the ability to make payment services and track all financial transactions of co-owners of residential complexes.

The subject of research is information technology development and software implementation of the financial management system of residential complexes.

The purpose of the work is to design and develop an information system for managing the financial activities of residential complexes, which provides users with up-to-date financial information on the calculation and payment of utilities, planning future expenses and financial reports on financial and economic activities.

The practical value of the master's work is that the developed system can be used to manage the financial activities of any interested association of co-owners of a residential complex.

Keywords: INFORMATION SYSTEM, FINANCIAL ACTIVITY, HOUSING COMPLEX.

The master's thesis contains 77 pages, 11 tables, 19 figures, 15 references.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ .....	8
ВСТУП .....	9
1 АНАЛІЗ ТА ВИЗНАЧЕННЯ ФУНКЦІОНАЛЬНИХ ВИМОГ І ТЕХНОЛОГІЙ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	11
1.1 Дослідження особливостей управління фінансовою діяльністю житлових комплексів.....	11
1.2 Огляд програмних платформ для реалізації системи.....	13
1.3 Переваги застосування технології Single Page Application.....	18
1.4 Постановка завдання.....	21
2 ВИРІШЕННЯ ЗАВДАННЯ ВИБОРУ АРХІТЕКТУРНИХ РІШЕНЬ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ .....	23
2.1 Дослідження та вибір програмної платформи реалізації системи засобами теорії прийняття рішень .....	23
2.2 Огляд та аналіз методів для рішення багатокритеріальних завдань .....	26
2.3 Розв’язання завдання вибору програмної платформи.....	33
2.4 Формування множини допустимих варіантів та рангів критеріїв .....	35
2.5 Одержання найкращого варіанту вибору програмної платформи методом головного критерію.....	38
3 ПРОЕКТУВАННЯ ТА ВИБІР ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ СИСТЕМИ .....	40
3.1 Порівняльний аналіз технологій SPA та MPA.....	40
3.2 Визначення програмних засобів розробки системи .....	43
3.3 Використання технологій управління процесами збірки проекту.....	46
3.4 Засоби тестування системи .....	48
3.5 Проектування системи за методологією функціонального моделювання SADT .....	51

3.6	Проектування бази даних системи.....	57
4	ПРОГРАМНА РЕАЛІЗАЦІЯ ІС УПРАВЛІННЯ ФІНАНСОВОЮ ДІЯЛЬНІСТЮ ЖИТЛОВИХ КОМПЛЕКСІВ.....	62
4.1	Розробка додатку для користувача-клієнта.....	62
4.2	Реалізація функції авторизації та реєстрації у системі .....	69
4.3	Розробка додатку для користувача-адміністратора .....	71
	ВИСНОВКИ.....	75
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	76

## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

### Скорочення

БД	– база даних
ІС	– інформаційна система
ЖК	– житловий комплекс
ПЗ	– програмне забезпечення
API	– Application Programming Interface – програмний інтерфейс
DOM	– Document Object Model – програмний інтерфейс для доступу до документів;
JSON	– JavaScript Object Notation – форма представлення даних, в який самі дані представляються у вигляді об'єкта мови JavaScript, виду «Ключ»:«значення»
JWT	– JSON Web Tokens – відкритий стандарт (RFC 7519) для передачі так званих «запитів» (англ. claims) між двома сторонами, взаємодіючими з допомогою веб-додатків
HTML	– HyperText Markup Language – мова гіпертекстової розмітки
MVC	– Model-View-Controller
SPA	– Single Page Application – односторінкові додатки це веб-додатки
URL	– Uniform Resource Locator.

### Терміни

Хостинг	– послуга з надання простору для розміщення інформаційних систем в Інтернеті
REST	– стиль архітектури програмного забезпечення, при якому вся інформація про запит зберігається в самому запиті, відсутня зберігання сервером станів.



## ВСТУП

Житлово-комунальне господарство – це ланка управління фінансовою діяльністю житлових комплексів. Житлове господарства є однією з важливих і пріоритетних галузей національного господарського комплексу, яка забезпечує життєдіяльність населених пунктів та суттєво впливає на розвиток різноманітних взаємовідносин у державі. Житлова сфера включає управління фінансовою діяльністю житлових комплексів, житловим фондом і об'єктами комунального господарства, їх утримання, будівництво й ремонт. Головною умовою та сенсом розвитку соціально спрямованої економіки є формування гідних умов життя людини. Житлово-комунальне господарство (ЖКГ) значною мірою формує середовище життєдіяльності людини – комфортність міста, району, мікрорайону, житла; одночасно виступає найважливішим елементом як соціального, так і економічного середовища. Житлово-комунальне господарство має низку специфічних особливостей, що характеризують його як складну, багатоелементну, динамічну організаційно-економічну систему [1]<sup>1)</sup>.

В умовах ринкової економіки житлово-комунальні послуги є товаром, який реалізується на ринкових засадах, тому створення інформаційної системи управління фінансовою діяльністю житлових комплексів, головним завданням якої є надання послуг з оплати та відстеження всіх фінансових операцій співвласників житлових комплексів є актуальним завданням.

Специфікою діяльності житлових комплексів, які надають житлово-комунальні послуги, є те, що послуги надаються незалежно від стану платежів за них окремими споживачами. Застосування інформаційної системи управління фінансовою діяльністю житлових комплексів спрямована на ліквідування нестабільності роботи фінансових структур житлових комплексів, яке пояснюєть-

---

<sup>1)</sup> [1] Димченко О. В. Житлово-комунальне господарство в реформаційному процесі: аналіз, проектування, управління: монографія. Х., ХНАМГ, 2009. 356 с.

ся в основному неплатежами споживачів за надані послуги, несвоєчасністю та проведенням не в повному обсязі розрахунків за отримані послуги.

Завданням інформаційних систем для управління фінансовою діяльністю житлових комплексів є перш за все автоматизований збір та оброблення платежів за житлово-комунальні послуги, а також інтеграція та розподіл інформаційних та фінансових потоків між житлово-комунальними видами доходів та витрат. При створенні і впровадженні інформаційних систем для управління фінансовою діяльністю житлових комплексів, що забезпечують можливість автоматизованого збору платежів необхідно вирішити ряд проблем, пов'язаних з унікальністю реалізації системи автоматизації збору платежів для кожного житлового комплексу, а також доступність інформації для кожного з членів житлового комплексу для отримання своєчасної та достовірної фінансової інформації.

Метою магістерської роботи є проектування та розробка інформаційної системи управління фінансовою діяльністю житлових комплексів, що надає користувачам системи актуальну фінансову інформацію щодо здійснення нарахування і оплати отриманих комунальних послуг, планування майбутніх витрат та фінансові звіти про фінансово-господарську діяльність.

Розроблена інформаційна система управління фінансовою діяльністю повинна задовольняти наступним вимогам: надійність, простота використання та доступність для користувачів, застосування сучасних програмних засобів розробки, конфіденційність даних. Використання такої системи надасть можливість користувачам можливість здійснення обліку фінансово-господарських операцій, що відбуваються у житловому комплексі.

# **1 АНАЛІЗ ТА ВИЗНАЧЕННЯ ФУНКЦІОНАЛЬНИХ ВИМОГ І ТЕХНОЛОГІЙ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ**

## **1.1 Дослідження особливостей управління фінансовою діяльністю житлових комплексів**

Житлово-комунальні послуги – результат господарської діяльності, спрямованої на забезпечення умов проживання та перебування осіб у жилих і нежилых приміщеннях, будинках і спорудах, житлових комплексах будинків і споруд відповідно до нормативів, норм, стандартів, порядків і правил.

Для управління господарською та фінансовою діяльністю житлових комплексів власники квартир у кожному з таких комплексів утворюють об'єднання співвласників багатоквартирних будинків, які є юридичною особою, що створюється для представлення спільних інтересів співвласників. Ця організація є неприбутковою за статутом. Основна мета створення такого об'єднання – спільне користування, утримання та управління житловим комплексом та територією, яка належить цьому комплексу, а також для юридичного оформлення майнових прав як на житловий комплекс так і на навколишню територію. Це дозволяє співвласникам житлового комплексу здійснювати ефективне управління як господарською так і фінансовою діяльністю. Мета постачальників житлово-комунальних послуг для житлових комплексів – отримати максимальний прибуток, надаючи більше послуг за максимальною вартістю і мінімальними витратами. Але і мета споживачів, які є власниками квартир у житловому комплексі – отримати якісні послуги при мінімальних витратах. Ці дві мети знаходяться в суперечці. Тому серед мешканців житлових комплексів виникають завади, які потрібно вирішувати для здійснення ефективного управління житловим комплексом: які організації будуть обслуговувати будинок, величина внесків на утримання будинку і території, кому здавати в оренду нежитлові при-

міщення тощо, куди витратити гроші і т.п. Ефективне використання коштів, прозорість нарахування платежів за отримані послуги, можливість ознайомлення з кожною з фінансових операцій, які здійснені, надають кожному з співвласників можливість здійснення контролю і приводять до якіснішого та раціональнішого управління всіма фінансовими та господарськими операціями, що здійснюються у конкретному житловому комплексі [2]<sup>1)</sup>.

Така створена юридична особа, створена співвласниками житлових комплексів відповідає лише за управлінські повноваження, а не стає власником житлового будинку і його територій. А власники квартир стають співвласниками нежитлових приміщень, допоміжних нежитлових приміщень та земельної ділянки навколо будинку межі якої встановлені згідно плану. Ефективне управління житловим комплексом стає метою кожного з власників квартири і дозволяє здійснювати контроль як за фінансовою так і за господарською діяльністю житлового комплексу. Юридична неприбуткова установа співвласників житлового комплексу має право мати розрахункові рахунки, розміщувати депозити в банках, купувати/продавати майно, здавати в оренду нежитлові приміщення, територію або їх частину. При цьому, вільні (або не використані) грошові кошти об'єднання не можуть бути виплачені співвласникам, а повинні бути використані виключно на здійснення поліпшення житлових умов власників квартир.

Для ефективного функціонування організації співвласників житлового комплексу необхідно використовувати інформаційну систему управління фінансовою діяльністю, що надасть можливість здійснювати кожному з власників квартири контроль за нарахуваннями за отриманням комунальних послуг, своєчасне отримання рахунків на оплату, надасть можливість більш детально ознайомитися з здійсненими господарськими і фінансовими операціями, що буди

---

<sup>1)</sup> [2] Торкатюк В. І, Усенко Ю. Ю. Оценка экономической эффективности решения задач управления жилищно-коммунальной структурой города. URL: [https://pidru4niki.com/79173/finansy/finansova\\_diyalnist\\_zhitlovo-ekspluatatsiynogo\\_kompleksu](https://pidru4niki.com/79173/finansy/finansova_diyalnist_zhitlovo-ekspluatatsiynogo_kompleksu) (дата звернення: 06.04.2020)

здійсненні житловим комплексом, бути у колі сучасної інформації, яка є актуальною для конкретного житлового комплексу.

Реалізація у інформаційній системі функції оплати за комунальні платежі полягає в тому, що вона надає можливість для інформаційно-аналітичного оброблення отриманих даних, оперативного формування звітів про рівень збору платежів за комунальні послуги і стан заборгованості за послуги, аналізу, прогнозування і планування розвитку житлово-комунального господарства. Крім того системи дозволяє вести бухгалтерський облік всіх операцій і формувати фінансові звіти про всі види діяльності за обраний період, що стає зручним механізмом управління житловим комплексом як для осіб що здійснюють керівництво так і для кожного з власників.

## **1.2 Огляд програмних платформ для реалізації системи**

Для здійснення розробки інформаційної системи управління фінансовою діяльністю житлових комплексів необхідно обрати фреймворк або програмний каркас, який є основою для розробки. Фреймворк є програмною платформою, яка визначає структуру розроблюваної програмної системи. Використання такого програмне забезпечення, значно полегшує розробку інформаційної системи забезпечує зручний засіб для об'єднання різних компонентів в один програмний проект. На сьогоднішній день актуальними є декілька фреймворків, які задовольняють вимогам розробників, але для обрання програмної платформи для розробки необхідно проаналізувати всі переваги використання. Розглянемо наступні програмні платформи: Angular, React, Backbone, Knockout, Vue.

В основі фреймворків лежать певні елементи. Елемент отримує команду введення і після деяких внутрішніх поводжень, обчислень він повертає запитуваний UI-шаблон (наприклад, область вводу, виводу або елемент списку) в якості результату. При цьому не повинно виникати труднощів при повторному ви-

користанні певних елементів. Наприклад, є grid-елемент, що складається з header і декількох row-елементів, з різними параметрами (колонками таблиць, header, row і т.д.), який може бути повторно використаний з іншими параметрами для іншої веб-сторінки. Розглянемо Фреймворк Backbone. Backbone – JavaScript фреймворк, який є зручним інструментом при створенні односторінкових додатків здатний вирішити певний перелік завдань, що виникають при розробці саме таких веб-додатків. Перелічимо основні можливості програмної платформи Backbone [3]<sup>1)</sup>:

- моделі зі зв'язування за ключем й користувацькими подіями;
- відкритий сирцевий код (розповсюджується за MIT ліцензією);
- колекції з великим набором методів для роботи з перерахованими сутностями;
- представлення з декларативною обробкою подій;
- інтеграція за існуючим RESFUL JSON API;
- клієнтська маршрутизація.

Backbone – це простий фреймворк, який вписується в один JavaScript файл. Backbone був розроблений Джеремі Ашкенасом за допомогою CoffeeScript. Фреймворк базується на шаблоні проектування MVP (Model-ViewPresenter). Моделі дозволяють зв'язати ключ-значення і події для обробки зміни даних. Backbone особливо популярний серед розробників, які шукають просту структуру для невеликих веб-додатків, без застосування великих фреймворків як Angular або Ember. Backbone найкраще використовувати при побудові простих односторінкових додатків. При виборі програмного каркаса слід враховувати те, що в якості загального паттерна розробки програми був обраний SPA у вигляді товстого клієнта, при цьому інтерфейс клієнт-серверної взаємодії –

---

<sup>1)</sup> [3] Болотіна В. В. Огляд популярних JAVASCRIPT фреймворків. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2019/06/42.pdf> (дата звернення: 18.07.2020)

REST. Крім цього, слід також спиратися на наступні важливі не функціональних вимог до додатка [3]<sup>1)</sup>:

- гнучкість і масштабованість;
- високий рівень інтерактивності призначеного для користувача інтерфейсу;
- швидкість роботи програми повинна бути досить комфортною для роботи користувача з додатком;
- користувач повинен мати можливість працювати з додатком за допомогою ПК і планшета.

Спираючись на перераховані вимоги, а також з огляду на плюси і мінуси різних фреймворків, як програмного каркаса для front-end частини програми буде використаний Angular фреймворк.

Розглянемо фреймворк Ember. Ember – JavaScript фреймворк який спрощує створення масштабованих односторінкових додатків. Представляє з себе каркас що реалізує MVC шаблон. Перелічимо основні переваги: [4]<sup>2)</sup>.

- клієнтська маршрутизація;
- відкритий сирцевий код (розповсюджується за MIT ліцензією);
- моделі які містять данні що відповідають поточному стану додатку;
- контролери використовуються щоб надати моделі логіку відображення;
- бібліотека Ember Data для зв'язку з сервером й отримання JSON даних;
- шаблони (прототипування) на мові HTMLBars.

У цьому досить популярному фреймворку упор робиться на підвищення продуктивності програміста. Ember теж відноситься до MVC фреймворків. У ньому використовується шаблонізація і вбудований рендерер відображення, який автоматично оновлює дані так само, як і Angular, Backbone і React. Ember

---

<sup>1)</sup> [3] Болотіна В. В. Огляд популярних JAVASCRIPT фреймворків. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2019/06/42.pdf> (дата звернення: 18.07.2020)

<sup>2)</sup> [4] Популярные фреймворки JavaScript. URL: <https://vc.ru/dev/147263-populyarnye-freymvorki-javascript> (дата звернення: 18.07.2020)

підтримує технологію веб-компонентів, що дозволяє розширювати HTML за допомогою власних тегів, в цьому він схожий на Angular. Він також відмінно масштабується і може використовуватися для великих проєктів. У Ember є відмінний інструмент для збірки, запозичений з багатьох інших середовищ SPA, званий Ember CLI. Цей інструмент збірки має все необхідне для початку роботи: вбудований розтер, модуль тестування, внутрішні данні, які є на Ембер. Ember.js слід багатьом тим же принципам, що і Ruby в Rails. Він дуже продуманий, гнучкий. Однак, розробники фреймворка зробили великий акцент на його самодостатність, при цьому програвши в гнучкості. Ember надає інструменти контролю розробнику, вирішуючи багато речей самостійно. Як наслідок, Ember не підходить для розробки веб додатків, в яких однією з вимог є гнучкість [4]<sup>1)</sup>.

Наступним був розглянутий фреймворк Knockout – це JavaScript фреймворк для створення візуального інтерфейсу який використовує шаблон MVVM. Основна концепція бібліотеки полягає в тому, що бібліотека дозволяє автоматично відстежувати зміни в інтерфейсі і відповідно до змін в елементах інтерфейсу синхронізувати використовуваний джерело даних. Інакше кажучи KnockoutJS дозволяє простий підхід до використання прив'язки даних. KnockoutJS має такі особливості [5]<sup>2)</sup>:

- відстеження залежностей: бібліотека автоматично оновлює елементи призначеного для користувача інтерфейсу при змінах в прив'язаному джерелі даних;
- має декларативний спосіб прив'язки: бібліотека дозволяє декларативним способом в коді HTML прив'язувати елементи призначеного для користувача інтерфейсу до джерела даних;

---

<sup>1)</sup> [4] Популярные фреймворки JavaScript. URL: <https://vc.ru/dev/147263-populyarnye-freymvorki-javascript> (дата звернення: 18.07.2020)

<sup>2)</sup> [5] Введение в KnockoutJS. URL: <https://metanit.com/web/knockout/1.1.php> (дата звернення: 24.07.2020)



- має можливість розширення: при необхідності можна легко додати кастомними поведінку до вже наявних можливостей бібліотеки.

Зазначимо основні переваги: безкоштовний; відкритий код (розповсюджується за MIT ліцензією); написаний на чистому JavaScript; легкий (54 Кб мінімізований); не має залежностей; підтримується всіма сучасними браузерами; повністю задокументований.

Наступним був проаналізований фреймворк Angular, який являє собою найпотужніше середовище JavaScript. Google використовує цю платформу для розробки односторінкових додатків. Популярність цього середовища зумовлене тим, що надає розробникам найкращі умови для об'єднання JavaScript з HTML і CSS. Більш ніж півмільйона сайтів, таких як google.com, youtube.com і т.п., використовують саме Angular. Angular користується великою популярністю, оскільки пропонує повноцінну інфраструктуру додатка та зручні рішення. Серед його можливостей можливо визначити наступні: зв'язування; шаблони (прототипування); компонента архітектура додатка; впровадження залежностей; клієнтська маршрутизація. Angular найновіший й найбільший за можливостями фреймворк, він не тільки допомагає в розробці односторінкового додатку, а й пропонує його цілу інфраструктуру з використанням багатьох інших нових додаткових технологій таких як TypeScript й ECMAScript6. Також Angular інтегрував в собі багато дрібних бібліотек які вирішили проблеми першої версії, наприклад Zone.js [6]<sup>1)</sup>.

Проаналізуємо фреймворк React, який не відноситься до фреймворків в чистому вигляді, а саме являє собою модифіковану бібліотеку, яка підтримує шаблон MVC, де модель відповідає за надання даних, вид – відображає дані моделі користувача, а Контролер – інтерпретує дії користувача і змушує модель вносити зміни. ReactJS має відкритий вихідний код і розвивається в основному

---

<sup>1)</sup> [6] Сравнение JavaScript фреймворков Vue.js, React и Angular (2019). Статті о программировании mkdev. URL: <https://mkdev.me/posts/sravnenie-javascript-freymvorkov-vue-js-react-i-angular-2019> (дата звернення: 02.08.2020)

за допомогою Facebook за участю інших великих технологічних компаній. React описує себе як JavaScript бібліотека для створення користувацьких інтерфейсів. При цьому спосіб зберігання даних цілком лягає на розсуд розробника. Однак, React – це перш за все уявлення (вид) додатку, але у складі фреймворка відсутня будь-яка готова система події, або шар даних. При розробці додатків на React велику кількість часу йде на підбір і зв'язку відповідного технологічного стека. Тому React застосовують при розробці відносно простих веб-додатків [6]<sup>1)</sup>.

Проведений аналіз сучасних програмних платформ розробки веб-систем довів, що дуже багато з'явилося вдалих рішень для розробників, які використовують JavaScript-фреймворки для веб-розробки. Фреймворки JS являють собою бібліотеки програмування JavaScript з великою кількістю написаного коду для використання в стандартних функціях і завданнях програмування, що забезпечує добре підґрунтя для створення веб-додатків. Вдало вибраний фреймворк значно полегшити роботу розробнику, який отримає безкоштовні і з відкритим вихідним кодом потужні інструменти розробки. В першу чергу це підвищить скорочення часу і складності розробки. Деякі компоненти веб-системи можуть бути запозичені і розширені.

### 1.3 Переваги застосування технології Single Page Application

Інформаційна система для управління фінансовою діяльністю житлових комплексів має бути розроблена як веб-додаток, який повинен задовольняти потребам численної і різноманітної аудиторії користувачів (власників квартир), які використовують різноманітні пристрої і мають різні навички роботи з сучасними додатками в мережі Інтернет. Тому доцільним буде для розробки такої системи

---

<sup>1)</sup> [6] Сравнение JavaScript фреймворков Vue.js, React и Angular (2019). Статті о програмуванні mkdev. URL: <https://mkdev.me/posts/sravnenie-javascript-freymvorkov-vue-js-react-i-angular-2019> (дата звернення: 02.08.2020)

використовувати технологію SPA, що надасть можливість створити інформаційний web-додаток побудований за принципами односторінкового сайту.

SPA (single page application) – це web-додаток, який розміщено на одній web-сторінці, яка для забезпечення роботи завантажує весь необхідний код разом із завантаженням самої сторінки. Додатки, виконані з застосуванням технології SPA відмінно працюють на пристроях як стаціонарних, так і мобільних. Комп'ютери, планшети, смартфони можуть безперешкодно працювати з сайтами побудованих за принципом SPA. Існує три основні підходи до розробки веб-додатків: односторінкові (SPA), багатосторінкові (MPA) і прогресивні (PWA). Вони виділяються серед інших підходів простотою розробки, зручністю для користувачів і широкими можливостями для розвитку бізнесу. Існують два види SPA додатків: додаток, який вбудовує контент декількох сторінок в одну сторінку-оболонку, і додаток, який використовує Ajax-завантаження контенту, коли користувач клацає по посиланню. Таким чином, на сторінці оновлюється тільки та частина контенту, яка змінюється, а інша частина сторінки залишається без змін і перезавантажень. Односторінковий додаток SPA представляє собою веб-сторінку або додаток, розміщений на одній веб-сторінці., що забезпечує більш гнучкий і зручний призначений для користувача інтерфейс, схожий з інтерфейсом десктопного додатка, крім того, односторінковий додаток має більш високу продуктивність та забезпечує більш природний і контрольований досвід взаємодії (UX), приховуючи складні переходи (запити, відгуки та ін.). SPA відображає тільки той контент, який запитує користувач, завантажуючи HTML дані частинами. Для реалізації інформаційної системи за технологією SPA існує декілька популярних фреймворків [7]<sup>1)</sup>.

Для здійснення програмної реалізації інформаційної системи управління фінансовою діяльністю житлового комплексу, на підставі проведеного аналізу

---

<sup>1)</sup> [7] Разработка одностраничных web-приложений – Создание SPA-приложений. URL: <https://avada-media.ua/services/single-page-application-development/> (дата звернення: 14.09.2020)

сучасних фреймворків, обраний JavaScript-фреймворк з відкритим ісходним кодом – Angular, який стане зручним інструментом для розробки системи за технологією SPA. Це найбільш популярний і зручний фреймворк від корпорації Google, який є перспективним для подальшого розвитку і використання, а проекти, зроблені з його допомогою, через декілька років будуть сучасними програмними продуктами. Завдяки використанню різних елементів управління, система, реалізована за допомогою технології SPA, має привабливий, функціональний, зручний і інтуїтивно-зрозумілий інтерфейс користувача з можливістю швидкої модернізації у разі потреби. А завдяки тому, що вона реалізована за допомогою технології Single Page Application, дана система являється дуже швидкою в користуванні і виділяється на фоні інших систем саме унікальною швидкістю роботи. На сьогоднішній день існує багато різноманітних технологій, які дозволяють розробникам зробити з веб-сайту систему, дуже схожу на нативні додатки для операційних систем. Щоб це реалізувати, потрібно всі зміни обробляти за допомогою скриптів і запити на сервер робити за допомогою AJAX. Відкриваючи подібний веб-сайт користувач має лише один раз завантажити сторінку цього веб-сайту і далі все, що користувач буде робити на сайті (перехід по сторінкам, внесення якихось даних, тощо) буде відбуватись без перезавантаження сторінки і відбуватись все буде дуже швидко, майже миттєво. Це і є концепція технології SPA (односторінкового додатку). Сьогодні дуже багато нових проектів створюються саме за допомогою цієї технології. Принципи будь-якого фреймворка, який реалізує парадигму SPA повинні дотримуватися наступних понять і визначень [8]<sup>1)</sup>:

- SPA підтримує клієнтську навігації. Всі переходи користувача за модулями-сторінок однозначно фіксуються в історії навігації, причому навігація при цьому є «глибокою», тобто якщо користувач скопіює і відкриє

---

<sup>1)</sup> [8] Что такое SPA или одностраничный портал. URL: <http://www.codenet.ru/webmast/js/spa/> (дата звернення: 14.09.2020)

- посилання на внутрішню модуль-сторінку в іншому браузері або вікні, він потрапить на відповідну сторінку;
- SPA розміщується на одній web-сторінці, значить все необхідне для роботи сайту (порталу) скрипти і стилі повинні бути визначені в одному місці проекту – на єдиній web-сторінці;
- SPA зберігає постійно стан (важливі змінні) роботи клієнта (клієнтського скрипта) в кеші браузера або в Web Storage;
- SPA завантажує всі скрипти потрібні для старту програми при ініціалізації web-сторінки;
- SPA поступово завантажує модулі на вимогу.

Розглянута технологія SPA –це принцип архітектури за яким створюється додаток. Технологія SPA значно скорочує завантаження сторінок додатку при оновленні, тобто завантаження контенту.

#### **1.4 Постановка завдання**

Завданням магістерської роботи є розробка інформаційної системи управління фінансовою діяльністю житлових комплексів, яка є зручним інструментом для використання власниками квартир житлового комплексу та керівництвом житлового комплексу для здійснення оплат за отримані комунальні послуги та відстеження всіх фінансово-господарських операцій у житловому комплексі.

Інформаційна система повинна забезпечувати доступ до актуальної інформації наступним категоріям користувачів: адміністратору системи, в якості якого може виступати керівництво і бухгалтер житлового комплексу та користувачі-клієнти, в якості яких будуть виступати власники квартир житлового комплексу. Така система повинна забезпечувати зручний користувацький інтерфейс для всіх категорій користувачів та надання здійснення перегляду фінансової інформації, а саме нарахування оплати за отримані комунальні послуги.

ги, а також забезпечувала формування фінансової звітності по всім здійсненим господарсько-фінансовим операціям, які буди здійсненні житловим комплексом. В системі передбачена обов'язкова реєстрація та подальша авторизація всіх користувачів.

Інформаційна система управління фінансовою діяльністю житлових комплексів може буди використана будь-яким з житлових комплексів для управління та відстеження фінансової діяльності всіма співвласниками. Система забезпечує користувачів актуальною інформацією щодо кожної фінансової операції, ознайомлюватись з витратною частиною, яка здійснена у житловому комплексі більш детально, нарахуванню оплати за спожиті комунальні послуги та дозволяє формувати фінансову звітність за обраний період та інше.

Користувачу-адміністратору інформаційної системи управління фінансовою діяльністю житлових комплексів надаються наступні функції в системі: можливість здійснення внесення даних щодо витрат, які заплановані, а також виконані; здійснювати реєстрацію користувачів-власників квартир у системі, а також видаляти; додавати та видаляти інформацію щодо здійснення фінансових та господарських операцій в системі.

Користувачу-клієнту інформаційної системи управління фінансовою діяльністю житлових комплексів (власникам квартир) надаються наступні права у системі: можливість перегляду переліку всіх виконаних або запланованих фінансових або господарських операцій в системі; ознайомлення з деталями проведення кожної з фінансових операцій (кошторис на виконані роботи, акт прийняття робіт і т.п. ); можливість обрання необхідного періоду для формування переліку фінансових операцій; можливість відновлення або зміни паролю; можливість перегляду списку нарахувань за отримані комунальні послуги та їх оплати, а також пошуку в системі.

## **2 ВИРІШЕННЯ ЗАВДАННЯ ВИБОРУ АРХІТЕКТУРНИХ РІШЕНЬ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ**

### **2.1 Дослідження та вибір програмної платформи реалізації системи засобами теорії прийняття рішень**

Для здійснення розробки інформаційної системи управління фінансовою діяльністю житлових комплексів необхідно вирішити завдання вибору програмної платформи для реалізації системи, а саме зручного фреймворку для подальшої програмної розробки. Саме вдалий вибір архітектурних рішень для розробки системи забезпечить швидкість та спрощення розробки.

Засоби та методи теорії прийняття рішень є зручним інструментом для вибору підмножини альтернатив, якими є сучасні фреймворки, з вихідної їх безлічі. Вибір тієї чи іншої альтернативи варіанту фреймворку з початкової множини повинен бути так чи інакше обумовлений за деякими критеріями, які повинні бути значущими. Особою, яка приймає рішення (ОПР) – фахівцем предметної області, обираються найважливіші критерії. Розробник інформаційної системи є фахівцем, який обирає архітектурні рішення, які будуть застосовані при розробці системи. Для можливості вибору тієї чи іншої альтернативи з початкової множини у розробника-фахівця повинна бути, можливість здійснити порівняльний аналіз а також можливість оцінювати ці альтернативи.

На першому етапі процесу прийняття рішень необхідно здійснити формалізацію завдання. Визначимо основні завдання узагальненої процедури прийняття рішення. На першому етапі необхідно сформулювати мету, здійснити її аналіз та формалізацію. Подальшим кроком є визначення безлічі можливих шляхів досягнення мети, що сформує безліч рішень, а після проведення оцінювання, отримаємо можливість проведення ранжування отриманих рішень між

собою. Всі перелічені кроки призводять до вибору найкращого з альтернатив якісного єдиного рішення.

В теорії прийняття рішень сукупність перелічених завдань утворює загальну проблему прийняття рішень. Кінцевою метою вирішення загального завдання прийняття рішень є вибір з допустимої множини рішень  $X$  єдиного найкращого, тобто екстремального за обраними окремим критерієм рішення [9]<sup>1)</sup>.

$$x^o = \arg \underset{x \in X}{extr} \{ k_i(x) \}, i = \overline{1, n} \quad (2.1)$$

Коли задача є однокритеріальною, при умові що  $n = 1$ , задача має завжди єдине рішення і не являє собою складності вибору. А коли задача має умову, що  $n > 1$ , то така задача не має однозначного рішення і є багатокритеріальною, що дозволяє отримати єдине рішення тільки при окремих умовах, у всіх інших випадках задача не буде мати однозначної відповіді. Спираючись на вираз 2.1. можливо стверджувати, що задача багатокритеріальної оптимізації є некоректною, оскільки в загальному випадку не забезпечує визначення єдиного оптимального рішення з допустимої множини  $X$ . Але задача може бути вирішена за умови здійснення регуляризації задачі, що передбачає залучення додаткової інформації, такої як введення математичних правил, засобами яких здійснюється вибір єдиного вірного рішення. При здійсненні реалізації неконструктивний підходу ОПР не може формалізувати інформацію, вона розуміє її спираючись на досвід в предметній області, здійснюючи вибір на інтуїтивному рівні.

Загальний підхід до вирішення цієї проблеми полягає в трансформації багатокритеріальної задачі в однокритеріальну зі скалярним критерієм. Це обумовлено наступними двома причинами. По-перше, значення скалярного кількісно-

---

<sup>1)</sup> [9] Приймак В. Прийняття управлінських рішень: Навчальний посібник. К., Атіка, 2008, 240 с.



го критерію можна інтерпретувати як точку на числовій осі, і ранжування таких точок не становить труднощів, так як відносини переваги і еквівалентності перетворюються відповідно в нерівність і рівність. По-друге, всі методи пошуку екстремуму орієнтовані на скалярну функцію [10]<sup>1)</sup>.

Аналіз літературних джерел свідчить, що існує декілька методів, що дозволяють зводити багатокритеріальні оптимізаційні задачі в однокритеріальні. Найбільш поширеним з таких методів є метод головного критерію, засобами якого можливо віднести завдання вибору фреймворку для програмної реалізації інформаційної системи до рішення завдання прийняття рішень в умовах багатокритеріальності. Сформулюємо багатокритеріальне завдання вибору фреймворку. Нам відомо деяка безліч допустимих альтернатив – програмних платформ (фреймворків), кожен з яких можливо оцінювати деякою кількістю критеріїв, які є значущими для розробника. Необхідно знайти найкращу альтернативу. При вирішенні такого завдання основною проблемою є неоднозначність вибору найкращого рішення.

Для її усунення цієї суперечливої умови можливе застосування однієї з двох груп існуючих методів. Перша група методів передбачає ранжирування критеріїв і порівняння альтернатив засобами введення додаткових припущень, що дозволяє скоротити число значущих критеріїв. Застосування другої групи методів передбачає скорочення кількості альтернатив у вихідній множині, позбавляючись від невдалих. До методів першої групи відносяться метод згортки, метод головного критерію, метод граничних критеріїв, метод відстані. Обґрунтування однозначної області їх застосування відсутнє та обрання саме цих методів зумовлюється, перш за все, умовами завдання і перевагами якими керується особа, яка приймає рішення.

---

<sup>1)</sup> [10] Ларичев О.И. Теория и методы принятия решений. М., Логос, 2000, 354 с.

## 2.2 Огляд та аналіз методів для рішення багатокритеріальних завдань

Розглянемо завдання багатокритеріальної оптимізації. Припустимо, що дана деяка безліч допустимих рішень  $D$  та деяка множина критеріїв оцінки цих допустимих рішень  $f_1(x), f_2(x) \dots f_m(x)$ , що можуть бути визначені на всіх елементах множини. Завданням є необхідність знайти рішення  $x \in D$ , при умовах, що значення окремих критеріїв максимально можливі. Це завдання можливо надати, як гру  $m$  осіб, для яких безліч випадків в грі розглядається як множина  $D$ , тоді виграш гравців можливо надати у вигляді функції  $f_1(x)_i = \overline{1, m}$ . В наведений грі гравці прагнуть кожен сам для себе збільшити свій виграш залучивши кожен свою стратегію. А задача багатокритеріальної оптимізації передбачає здійснення вибору найкращого результату для особи що приймає рішення. Очевидно, що при здійсненні гри кожен гравець прагне досягти найкращого результату, а для вирішення завдання багатокритеріальної оптимізації рішення має приймати тільки одна людина, фахівець, що приймає рішення.

Розглянемо низку методів, спрямованих на вирішення завдань багатокритеріальної оптимізації, практичного характеру. Метод «ідеальної» точки надає можливість розглядати  $m$ -мірний простір з деякою кількістю локальних критеріїв  $m$ . Метод «ідеальної» точки передбачає обрання вектору, який і буде що відображати найкраще («ідеальне») рішення. Тобто «ідеальна» точка – це визначена координатами мінімальні або максимальні значення, що є найкращими значеннями для обраних локальних критеріїв. Для цього простори визначається метрика, яка передбачає обчислення відстані між вектором, який відображає рішення, що розглядається як «ідеальне». Найкращім рішення вважається таке, векторна оцінка якого найбільш наближена до найкращої («ідеальної») точці. Використання цього методу стикається з проблемою здійснення довільного вибору ідеальної точки і обов'язкове введення метрики.

Розглянемо приклад застосування методу ідеальної точки. По-перше, знайдемо максимально можливі значення окремих критеріїв, тобто для кожного з обраних критеріїв знайдемо максимальне значення функції  $f_i(x)$  на множині  $D$ . Оптимальні значення функції знайдемо за формулою:

$$w_i = \max_{x \in D} f_i(x), \quad i = 1, m.$$

Таким чином,  $w_i$  є максимально можливим значенням  $i$ -го критерію  $f_i(x)$  на множині  $D$ . Точка  $x_i \in D$ , така, що  $w_i = f_i(x) = w_i$ , є рішенням звичайної задачі однокритеріальної оптимізації. Припустимо, що  $w^* = (w_1, w_2, \dots, w_m)$ . Точка  $w$  називається ідеальною, оскільки в ній всі критерії мають максимально можливі значення і отримати більші значення ні одним критерієм без зменшення значень інших критеріїв, неможливо. Наступним вирішуємо завдання знаходження такого значення вектора  $w$ , для якого відстань до ідеальної точки  $p(w^*, w)$  є мінімальною. Для визначення функції відстані  $p(x, y)$  оберемо різні метрики, наприклад, середньоквадратичне відхилення, яке представлене наступним виразом [10]<sup>1)</sup>:

$$p(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}.$$

В подальшому представляється можливим визначення точки на множені, яка є найбільш близькою до значення ідеальної точки за допомогою виразу:

$$W = \{(f_1(x), f_2(x), \dots, f_m(x)), x \in D\}.$$

---

<sup>1)</sup> [10] Ларичев О.И. Теория и методы принятия решений. М., Логос, 2000, 354 с.

Така знайдена точка і є саме тим рішенням завдання багатокритеріальної оптимізації.

При розгляді лексикографічної постановки завдання формулюються задачі оптимізації складних систем, що складається з взаємозв'язаних підсистем, що відносяться до різних ієрархічних рівнів. Лексикографічне упорядкування часто використовується для встановлення правил старшинства і пріоритету. Розглянемо наступний приклад. Припустимо, що окремі критерії різняться за значимістю. Якщо здійснити упорядкування їх так, щоб найбільш значущий критерій має  $f_1(x)$ , наступним за значимістю є критерій  $f_2(x)$ , і т.д. Спочатку вирішується завдання максимізації найважливішим критерієм  $f_1(x)$  на всій множині допустимих рішень  $D$ . Тоді максимальне значення в цьому завданні знаходиться за формулою [11]<sup>1)</sup>:

$$\bar{\mu}_1 = \max_{x \in D} f_1(x),$$

тоді множина оптимальних рішень  $D_1$  для цього завдання знаходиться за формулою:

$$D_1 = \{x \in D : f_1(x) \geq \bar{\mu}_1\},$$

тобто множина всіх  $x \in D$ , на яких досягається максимальне значення першого критерію. В подальшому необхідно визначити максимальне значення наступного за важливістю другого критерію  $f_2(x)$ , використовуючи множину отриманих оптимальних рішень для першого критерію. А оптимальне значення в цьо-

---

<sup>1)</sup> [11] Штойер Р. Многокритериальная оптимизация. Теория вычислений и приложения. М., Наука, 1992, 204 с.

му завданні розрахуємо за формулою:  $\bar{\mu}_{21} = \max_{x \in D_1} f_2(x)$ , тоді для знаходження множини оптимальних рішень  $D_2$  можливо отримати через вираз:

$$D_2 = \{x \in D_1 : f_2(x) \geq \bar{\mu}_2\}.$$

Наступним кроком є вирішення завдань максимізації наступних за важливістю критеріїв на попередньо знайдений множині оптимальних рішень аж до завдання максимізації останнього критерію  $f_m(x)$ . Розглянутий метод можливо проаналізувати та визначити деякі недоліки використання такого методу, які полягають у тому, що якщо на якомусь етапі рішення відповідної задачі виявилося єдиним, то всі наступні кроки стають недоречними, а решта критеріїв фактично не беруть участі і не враховуються при визначенні рішення вихідної задачі.

Визначимо основні засади та алгоритми дії методу згортки критеріїв. Основна концепція методу полягає визначенні єдиного інтегрального критерію якості рішення, який враховує і агрегує в собі декілька окремих критеріїв, що дозволяє вирішити завдання багатокритеріальної оптимізації за рахунок визначення максимізації цього інтегрального критерію. Метод згортки окремих критеріїв можливо визначити по-кроково наступною низькою дій: по-перше визначається функція згортки  $\varphi(u_1, u_2, \dots, u_m)$ , де кількість змінних у функції згортки дорівнює кількості окремих критеріїв. Єдиний інтегральний критерій у вихідній задачі формується як суперпозиція функцій окремих критеріїв і функції згортки [10]<sup>1)</sup>:

$$\Phi(x) = \varphi(f_1(x), f_2(x), \dots, f_m(x)).$$

---

<sup>1)</sup> [10] Ларичев О.И. Теория и методы принятия решений. М., Логос, 2000, 354 с.

Тоді вихідне багатокритеріальне завдання зводиться до вирішення задачі максимізації інтегрального критерію  $\Phi(x)$  на множині  $D$ . Отримувати різні конкретні інтегральні критерії якості стає можливим, коли задаються різні функції згортки. Розглянемо деякі найбільш поширені способи згортання окремих критеріїв. Припустимо, що  $\varphi(u) = \sum_{i=1}^m a_i(u_i - \beta_i)$ , де  $a_i > 0$ ,  $\beta_i \geq 0$ ,  $i = 1, \overline{m}$ , при умові, що єдиний критерій ефективності надано лінійною комбінацією окремих критеріїв, які мають значення  $\varphi(u) = \sum_{i=1}^m a_i(f_i(x) - \beta_i)$  на деякій визначеній множині  $D$ , що дозволяє вирішити завдання максимізації цього критерію. Цей метод застосовують в економічних завданнях, де вагові коефіцієнти  $a_i, \beta_i$  часто виступають в ролі цін на  $i$ -й товар, а критерій  $\Phi(x)$  представляє собою сумарний прибуток. Призначення конкретних значень коефіцієнтів  $a_i, \beta_i$  викликає труднощі, якщо функцію згортки взяти у вигляді  $\varphi(u) = \min_{i=1, \overline{m}} a_i(u_i - \beta_i)$ , то інтегральний критерій буде мати вигляд [11]<sup>1)</sup>:

$$\Phi(x) = \min_{i=1, \overline{m}} a_i(f_i(x) - \beta_i).$$

При визначенні досягнення цілі користуються твердженням, при якому якісні цілі можуть бути або досягнуті, або не досягнуті, тому що часткове досягнення мети неприпустимо для цього завдання. Всі дії, що призводять до досягнення мети, однаково хороші, але якщо результати не призводять до досягнення мети вважаються незадовільними.

Значення критерію ефективності досягнення цілей може приймати значення при досягненні успіху – одиниця, або нуль, при поразці. В подальшому

---

<sup>1)</sup> [11] Штойер Р. Многокритериальная оптимизация. Теория вычислений и приложения. М., Наука, 1992, 204 с.

множину значень окремих критеріїв розбиваємо на задовільні і незадовільні. Визначаються числа  $\bar{\delta}_i$ , при  $i = 1, m$ , для яких вірним є умова  $f_i(x) \geq \bar{\delta}_i$  при виконанні якої числа визначаються як задовільні. Тоді визначення критерію згортки відбувається за формулою:

$$V(x) = 1 \text{ при } f_i(x) \geq \bar{\delta}_i$$

$$V(x) = 0 \text{ в інших випадках}$$

Такий алгоритм визначення єдиного критерію задовольняє оперуючи сторону і достатньо прозоро визначає вимоги, які визначені для значень окремих критеріїв.

Але виникає ряд труднощів для визначення граничних значень  $\bar{\delta}_i$ : є можливість втрати найбільш ефективних способів дій та опинитися в області недовсяжних значень. Такі умови роблять процедуру визначення граничних значень суперечливою і неоднозначною разом зі спрямованістю всього алгоритму дій до одночасного збільшення значень всіх окремих критеріїв.

Розглянемо метод головного критерію, який засновано на виділенні головного критерію та одночасне переведення всіх інших критеріїв в обмеження. Для визначення головного критерію проводиться аналіз особливостей багатокритеріальної задачі, враховуючі всю специфіку задачі і оцінку ваги всі критеріїв. З проаналізованої безлічі окремих критеріїв обирається один найважливіший критерій, який визначається в подальшому як єдиний критерій оптимізації. Решта переглянутих критеріїв підпадає під процедуру призначення граничних значень, нижче яких критерій не може зміщуватись.

Такі критерії і визначаються у цьому методі як обмеження в області допустимих рішень  $X$ , за винятком одного критерію – головного. Визначення головного(оптимізаційного) критерію з всієї множини критеріїв здійснюється осо-

бою, яка приймає рішення. Рівні обмежень також підлягають суб'єктивній оцінці експертами з обраної предметної області. Але у ОПР є можливість розгляду декількох різних варіантів та порівняти отримані результати. Важливим є умова не припущення деякої безлічі рішень, яка задана окремими критеріями, залишитися пустою.

Метод головного критерію передбачає вибір одного з функціоналів  $f_i$ , наприклад  $f_1$ , який найбільш точно і вдало відображає мету прийняття рішення на погляд ОПР. Всі інші критерії, що описані функціоналами  $f_1, \dots, f_m$ , враховують умови необхідних обмежень, що дозволяє вирішити отримане однокритеріальне завдання за виразами [11]<sup>1)</sup>:

$$f_1(x) \rightarrow \max_{x \in D}; \quad D' \subseteq D \subseteq R^n;$$

$$D' = \left\{ x \in \frac{D}{f_i(x)} \geq t_i, i = 2, \dots, m \right\}.$$

Якщо проаналізувати вирази, то можливо стверджувати, що отримано достатньо проста задача, що здійснює пошук максимуму функціонала  $f_1$ , на новій визначеній множині  $D'$ .

При умові дотримання обмеження виду  $f_i(x) \geq t_i$ , неможливе отримання максимальних значень для функціоналів  $f_2, \dots, f_m$ , що забезпечує їх обмеження на прийнятних рівнях. Застосування цього методу завжди натикається на суперечку, яка з'являється при визначенні одного найголовнішого критерію з декількох.

---

<sup>1)</sup> [11] Штойер Р. Многокритериальная оптимизация. Теория вычислений и приложения. М., Наука, 1992, 204 с.



Ще одним суперечливим фактом є нижніх меж  $t_i$ , який ґрунтується лише на інтуїтивних засадах.

Виконавши аналіз та огляд існуючих методів для здійснення прийняття рішення в умовах багатокритеріальності, при вирішенні завдання вибору програмної платформи для реалізації інформаційного системи управління фінансовою діяльністю житлових комплексів, було обрано метод головного критерію, який надає змогу з існуючої безлічі наявних альтернатив, обрати в якості головного критерію, критерій, який є найважливішим з точки розробника інформаційної системи.

### **2.3 Розв’язання завдання вибору програмної платформи**

Після проведеного вибору методу рішення багатокритеріального завдання вибору програмної платформи реалізації ІС та обрання методу головного критерію необхідно визначити деяку сукупність показників (критеріїв), які б були значущими при виборі програмної платформи (фреймворку).

Крім того необхідно визначити обмеження, які будуть враховуватися при рішенні завдання та найбільш вдало висвітлять індивідуальні переваги кожного з розглянутих фреймворків, що забезпечить оцінити переваги вибору той чи іншої програмної платформи особою, яка приймає рішення (ОПР) в якості якої виступає у даному випадку розробник.

Сформуємо таблицю вхідних варіантів, де обов’язково повинні буди: список назв різних JavaScript-фреймворків і декілька різних значущих для здійснення розробки показників, а саме: наявність MVC архітектури; складність проекту; модульне тестування; продуктивність; безпека; витрати.

Перелік варіантів обраних критеріїв оцінки вибору JS-фреймворку наведено у табл. 2.1.

Таблиця 2.1 – Визначені критерії оцінки вибору JS-фреймворку

№ варіанту	Найменування JavaScript-фреймворків	Наявність MVC, SPA архітектури	Складність проекту	Модульне тестування	Продуктивність	Безпека	Кількість бібліотек
1	Angular	1	4	5	4	5	5
2	React	1	4	5	3	3	5
3	Ember	0	4	2	2	5	2
4	Backbone	1	5	4	3	3	3
5	Knockout	1	4	3	2	2	2
Обмеження	*	$\leq 1$	$\geq 4$	[2;5]	$\geq 3$	[2;5]	[3;5]

Для визначеного переліку критеріїв оцінки вибору програмної платформи реалізації ІС були враховані сучасні огляди фреймворків, наведені фахівцями з ІТ-технологій у літературі [3]. В результаті аналізу отримано декілька програмних платформ, визначені значущі характеристики, які є окремими критеріями вибору при застосуванні методу головного критерію. Під стовпцем кожного критерію розташовані обмеження для кожного з фреймворків, які відображають переваги того чи іншого показника.

Обмеження може мати одну з наступних форм: обмеження зверху ( $\leq$ ), що дозволяє використання обмеження типу «не більше зазначеного числа»; обмеження знизу ( $\geq$ ), що дозволяє використання обмеження типу «не менше зазначеного числа»; обмеження, які діють в інтервалі ( $[1;5]$ ) дозволяють обмежити в рамках «від 1 до 5 включно»; відсутність обмежень позначають символом «\*». Таким чином була сформована таблиця 2.1, в якій визначені критерії

оцінки вибору JS-фреймворку та накладені обмеження, спираючись на досвід розробника системи та аналіз сучасних літературних джерел.

## **2.4 Формування множини допустимих варіантів та рангів критеріїв**

Наступним кроком є завдання формування таблиці, яка повинна містити припустимі варіанти, які визначаються наступним правилом: з таблиці видаляються ті рядки, в яких хоча б один окремих критеріїв не задовольняє умовам обмеження. Наступним кроком є визначення в таблиці вхідних варіантів: у кожному рядку необхідно порівняти значення кожного окремого критерію зі значенням відповідного обмеження. При умові, що кожне з аналізованих значень окремого критерію не задовольняє зазначеному обмеженню, то в значення цього осередку потребує видалення (ознака видалення), а замість видаленого значення розміщуємо в осередку поточного рядка номер цього окремого критерію. Цю операцію порівняння треба здійснити для кожного з рядків по кожному з критеріїв. Всі ці номери, які проставлені в стовбці «Ознака видалення» і є умовою видалення поточного варіанту з подальшого розгляду. Але можлива ситуація, коли таблиця допустимих варіантів може залишитися пустою, тоді ОПР формується нова таблиця варіантів і відбувається знов порівняння значень окремих критеріїв з зазначеними обмеженнями. Такі дії виконуються до тих пір, поки в таблиці допустимих варіантів залишиться один рядок, що свідчить про отримання рішення завдання багатокритеріального вибору метод головного критерію. Коли таблиця допустимих варіантів складається має декількох рядків, то необхідно перейти до наступного кроку вирішення завдання. Після виконання перелічених дій таблиця припустимих варіантів для вибору JavaScript-фреймворку для розробки інформаційної системи (табл. 2.1) буде мати наступний вигляд (табл. 2.2).

Таблиця 2.2 – Припустимі варіанти для завдання вибору

Номер варіанта	Найменування	Наявність MVC, SPA архітектури	Складність проекту	Модульне тестування	Продуктивність	Безпека	Кількість бібліотек	Ознака видалення
1	Angular	1	4	5	4	5	5	
2	React	1	4	5	3	3	5	
3	Ember	0	4	2	2	5	2	0
4	Backbone	1	5	4	3	3	3	5
5	Knockout	1	4	3	2	2	2	2
Об- меже	*	$\leq 1$	$\geq 4$	[2;5]	$\geq 3$	[2;5]	[3;5]	

Отримана таблиця матиме більш ніж один рядок і не є порожньою, тому треба перейти к наступному кроку вибору припустимого варіанту.

Подальше рішення завдання вибору найкращого варіанту методом головного критерію вимагає встановлення рангів критеріїв за умови, чим менша важливість, тим менший номер згідно з визначенням особою, яка приймає рішення.

Визначена сукупність критеріїв, яка наведена у таблиці теж є деякою безліччю вербальних значень, тому ранги критеріїв встановлюються після здійснення процедури попарних порівнянь, після чого критеріям надаються бали переваг.

Після проведення попарних порівнянь зазначених у табл. 2.2 критеріїв, отримаємо деякі значення окремих критеріїв, які наведено у табл. 2.3.

Таблиця 2.3 – Таблиця рангів окремих критеріїв завдання вибору

Окремий критерій	Наявність MVC, SPA архітектури	Складність проекту	Модульне тестування	Продуктивність	Безпека	Кількість бібліотек	Сума балів	Ранг
Наявність MVC архітектури	0	1	-1	-1	1	1	1	3
Складність проекту	1	0	-1	1	1	1	2	2
Модульне тестування	-1	1	0	-1	1	-1	-1	4
Продуктивність	1	1	1	0	-1	1	3	1
Безпека	-1	-1	1	-1	0	1	-1	4
Кількість бібліотек	-1	-1	1	-1	-1	0	-3	5

Наступним кроком буде виконання ранжування. Результати ранжування наведені у табл. 2.4.

Таблиця 2.4 – Отримані ранги окремих критеріїв при виконанні завдання вибору

Окремий критерій	Продуктивність	Складність проекту	Наявність MVC, SPA архітектури	Модульне тестування	Безпека	Кількість бібліотек
Ранг	1	2	3	4	4	5
Сума балів	3	2	1	-1	-1	-3

Проведемо аналіз отриманих результатів у табл. 2.3 та табл. 2.4. Очевидно, що обраний ОПР окремий критерій «продуктивність» виявляється найбільш важливим, про що свідчить максимальна можлива сума балів, яку набрав саме

цей критерій і його перемога у всіх проведених парних порівняннях. На думку особи, яка приймає рішення критерій «Продуктивність» посів друге місце в переліку найважливіших критеріїв, про що свідчить отримання сума балів. Якщо здійснювати подальший аналіз отриманих результатів, то можливо зазначити, суми балів, які набрані при порівнянні інших критеріїв рівномірно зменшуються.

## **2.5 Одержання найкращого варіанту вибору програмної платформи методом головного критерію**

Отримані ранги окремих критеріїв при виконанні завдання вибору найкращого варіанту програмної платформи для реалізації інформаційної системи управління фінансовою діяльністю житлових комплексів (табл.2.4) дозволяють виконати вибір найкращого програмної платформи за методом головного критерію за умови, що найвищий ранг має найменше значення номеру місця. За умови, що в таблиці є наявним єдиний найкращий варіант, то рішення задачі отримано. Але при виявленні декількох варіантів однакових за головним критерієм, то здійснюється перехід до розгляду варіантів по головному з інших критеріїв – наступного за встановленим рангом. За умови, коли всі варіанти виявляються еквівалентними за всіма критеріями, то кращий варіант вибирається за жеребом.

У розглянутому в магістерської роботі випадку виконання вибору найкращої програмної платформи для реалізації інформаційної системи управління фінансовою діяльністю житлових комплексів методом головного критерію виявлено самим значущим показником якості критерій «Продуктивність». Найкращим варіантом вибору саме за цим критерієм являється сучасний, потужний та популярний JS-фреймворк Angular, що розміщувався під номером 1 в таблиці вхідних варіантів (табл. 2.1).

Angular – найшвидший з популярних фреймворків та найпотужніша програмна платформа для середовища JavaScript. Корпорація Google використовує саме цю платформу для розробки односторінкових додатків SPA. Angular надає розробникам найкращі умови для об'єднання JavaScript з HTML і CSS. Цей потужний інструмент надає все необхідне для створення і управління динамічними front end сторінками для створення сучасних веб-додатків, що забезпечено використанням модульного підходу до веб-дизайну. Angular значно спрощує розробку, завдяки наявності шаблонів (прототипування), фахівці з фронтенду можуть з легкістю використовувати Angular для створення динамічних односторінкових веб-додатків з безліччю змінюваних даних. Angular підтримує архітектуру Model-View-Controller, що значно спрощує процес тестування.

Отже, можливо підсумувати всі переваги Angular і зробити висновок, що цей фреймворк є дуже перспективним й тому стає найкращим вибором для подальшого використання для розробки інформаційної системи.

### **3 ПРОЕКТУВАННЯ ТА ВИБІР ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ СИСТЕМИ**

Сучасні інформаційні системи мають досить складну архітектуру, визначення якої дозволяє ще на етапі проектування правильно побудувати архітектурні рішення, обравши всі необхідні і сучасні технології розробки, що є запорукою подальшого вдалого розвитку системи і легкого розширення функціональних можливостей системи. Архітектурні рішення дозволяють сформулювати концепцію, визначити модель функціонування, обрати структуру, що забезпечить ефективне і надійне функціонування системи.

Інформаційна система управління фінансовою діяльністю житлових комплексів має клієнт-серверну архітектуру, яка характеризується застосуванням двох незалежно функціонуючих процесів, а саме клієнта та серверу, які можуть функціонувати і взаємодіяти засобами мережі, як на двох різних комп'ютерах, так і бути розташовані на одному комп'ютері. Додаток, функціонуючий на стороні клієнта відповідає за виконання функцій на стороні клієнта, інтерфейс користувача, логіку обробки даних. Виконання функції маніпулювання даними в системі може виконувати декілька незалежних процесів в системі, що обумовлено тою чи іншою технологіями розробки інформаційної системи.

#### **3.1 Порівняльний аналіз технологій SPA та MPA**

Сучасні додатки та інформаційні системи повинні забезпечувати надійне функціонування, тому ще на етапі проектування і розробки необхідно обирати сучасні технології, які дозволяють надавати найкращу продуктивність і швидкість роботи системи. На сьогодні існує два способи створення веб-додатків, які мають різні концепції та підходи, але обидва застосовуються в сучасних розробках. Це односторінкові додатки (SPA) і багаторічкові додатки (MPA).



Для додатків, створених за технологією Multi Page Application характерною особливістю є виконання значної частини логіки на стороні серверу, а для додатків, створених за технологією Single Page Application характерною рисою є виконання всієї логіки інтерфейсу користувача веб-браузером на стороні клієнта, що зумовлює здійснення взаємодії з веб-сервером переважно з застосуванням веб-API. Також для розробки сучасних систем може бути використано, так званий гібридний підхід, який передбачає використання у веб-системі розміщення одного або декілька повнофункціональних підлеглих додатків, які переважно створені за технологією одно сторінкових додатків.

Застосування технології МРА бажано використовувати в випадках, коли до клієнтської частини веб-додатка застосовуються мінімальні вимоги (наприклад, підтримка тільки функції читання інформації), або додаток буде функціонувати в браузерах без підтримки JavaScript, або кваліфікація команди розробників не достатня для роботи з JavaScript або TypeScript.

Застосування технології SPA виправдано в випадках, коли веб-додаток повинен забезпечити повнофункціональний інтерфейс користувача, або умовою функціонування додатку є наявність API для інших внутрішніх або загальнодоступних клієнтів та розробники веб-додатку є кваліфікованими розробниками на JavaScript, TypeScript або Blazor WebAssembly.

При виборі програмної платформи для роботи з односторінковими додатками розробник повинен багато уваги приділити побудові архітектури системи та забезпеченню безпечного функціонування системи. При порівняльному аналізі цих двох програмних платформ виявилось, що при роботі з SPA-додатками може ускладнитися настройка процесів автоматизованої побудови і розгортання, а також використання таких варіантів розгортання, як контейнери в порівнянні з веб-додатками створеними за технологією МРА. Технологію SPA доцільно застосовувати коли потрібно відображати на сторінці додатку лише змінну інформацію не змінюючи весь вигляд сторінки. Додаток, розроблений за SPA

технологією, підключає JavaScript бібліотеки, засобами якого відбувається змінення DOM сторінки, що забезпечує відображення сторінки в реальному часі. При застосуванні MPA технології сервер забезпечує виконання функцій уявлення, а на стороні клієнта відбувається розміщення і обробка статичного контенту додатка. Технологію SPA не готує готове представлення контенту сторінок, а обмінюється к клієнтом інформаційними пакетами, найчастіше це пакети у форматі JSON, що надає уявлення при роботі с клієнтом, як з товстим клієнтом. На рисунку 3.1 наведено схематично механізми роботи цих двох розглянутих технологій з боку виконання як на стороні сервера та і на стороні клієнта специфічних функцій, які характеризують ці технології (рис. 3.1).

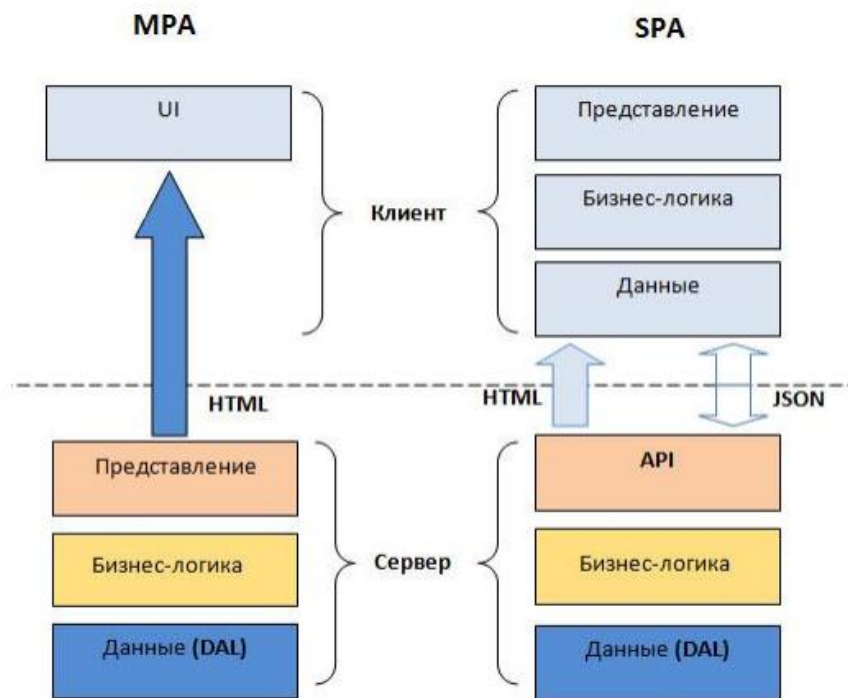


Рисунок 3.1 – Схеми розробки додатків за технологією MPA та SPA

Кожна з розглянутих в ході порівняльного аналізу архітектура має свої переваги та також і недоліки, а вибір той чи іншої концепції зумовлено конкре-

тними функціями проекту що розробляється. Додатки, створені за технологією SPA можуть пишатися своєю швидкістю і здатністю застосовувати готові модулі коду для швидкого створення мобільних програми. Але технологія SPA має погану SEO-оптимізацію в пошукових системах, що зумовлює область використання цієї технології: в додатках, де оптимізація пошукових систем не має значення SPA – найкраще рішення. Застосування MPA архітектури більш підходить для створення систем електронної комерції, каталогів і т.п., де саме SEO-оптимізація грає визначну роль для реалізації бізнес-логіки. Обидві технології можуть задовольняти потреби розробників для створення масштабованих веб-додатків.

На даному етапі розвитку WEB, односторінкові додатки швидко витісняють класичні додатки і є великим внеском у розробку масштабних, швидких, динамічних веб-систем. Однак технології реалізації front end частини web-додатків потребують додаткового дослідження в контексті вибору кращих альтернатив для реалізації систем під конкретні вимоги користувачів чи замовників продукту [12]<sup>1)</sup>.

### **3.2 Визначення програмних засобів розробки системи**

Для розробки інформаційної системи управління фінансовою діяльністю житлових комплексів обрано архітектурне рішення, що надає SPA-технологія, а програмною платформою обраний сучасний потужний фреймворк Angular. Розроблена ІС буде представляти собою односторінковий веб-додаток, який передбачає розподіл системи на дві взаємодіючі частини: на серверну – back-end і клієнтську front-end. Клієнтська частина буде виконуватися на комп'ютері користувачів, а серверна частина передбачає використання окремого фізичного

---

<sup>1)</sup> [12] Багатосторінкові та односторінкові додатки, їх переваги та недоліки . DOU URL: <https://dou.ua/forums/topic/25444/> (дата звернення: 16.09.2020)

вузла. Між клієнтом і сервером взаємодія здійснюється засобами REST API з використанням стандарту JSON в якості даних.

Відповідно до одного з функціональних вимог про те, що програма має бути здатна брати на себе деякі функції клієнта і взаємодіяти з сервером безпосередньо в реальному часі, back-end додатки будуть взаємодіяти з сервером за допомогою спеціального бінарного TCP протоколу для віддаленого управління сервером – Server Management API. Даний протокол дозволяє взаємодіяти з сервером безпосередньо в реальному часі. Server Management API не дозволяє виконувати всі можливі операції над сервером. Існують варіанти використання програми, наприклад, такі як «Реєстрація аккаунта», «Відображення статистики» і т.д., яким не обов'язково безпосередньо взаємодіяти з сервером. В цьому випадку з додатком може знадобитися з'єднання з СУБД MySQL для з'єднання з базою даних системи. Таким чином, back-end повинен вміти взаємодіяти з MySQL базою даних, при цьому back-end повинен мати доступ до відповідного вузла. Відповідно до одного з функціональних вимог про те, що дані, які не пов'язані з предметною областю, додаток повинен зберігати у власній MySQL базі даних, потрібен окремий фізичний вузол з розгорнутою на ній MySQL базою даних. Серверна частина веб-програми повинна вміти взаємодіяти з MySQL базою даних, при цьому повинна мати доступ до відповідного вузла [13]<sup>1)</sup>.

Написання повноцінного односторінкового додатку на чистому JavaScript можливо, однак отриманий вихідний код вийде дуже об'ємним, а отримана програма не буде мати достатню гнучкість і масштабованість, при цьому його буде складно тестувати. Саме тому виникає необхідність в програмному каркасі. Фреймворк Angular широко використовуваний для створення і підтримки складних веб-додатків. Angular має відкритий вихідний код (MIT ліцензія) за підтримки компанією Google. Angular це MVC фреймворк. Також у фреймворк

---

<sup>1)</sup> [13] Ярцев В.П. Організація баз даних та знань: навчальний посібник. К., ДУТ, 2018, 214 с.

вбудований движок маршрутизації і сервіс для зручної роботи з REST API. Фреймворк адаптує і розширює традиційний HTML, щоб забезпечити двосторонню прив'язку даних для динамічного контенту, що дозволяє автоматично синхронізувати модель і уявлення. В результаті Angular зменшує роль DOM-маніпуляцій, тим самим покращує підтримку коду і тестування. Двостороннє зв'язування даних в Angular зменшує кількість коду, звільняючи сервер від роботи з шаблонами [3]<sup>1)</sup>.

На рис. 3.2 наведена принципіальна схема взаємодії модулів програмної платформи Angular.

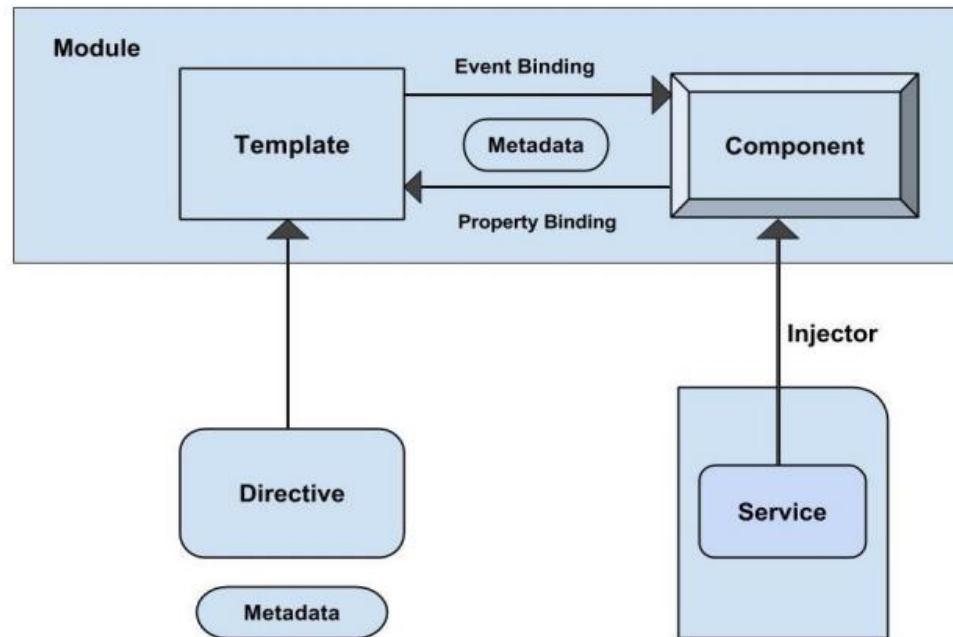


Рисунок 3.2 – Схема взаємодії модулів програмної платформи Angular

Фреймворк Angular складається з окремих модулів, кожен з яких має свою версію. Але розробка програми передбачає підтримку всіх модулів в актуаль-

<sup>1)</sup> [3] Болотіна В. В. Огляд популярних JAVASCRIPT фреймворків. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2019/06/42.pdf> (дата звернення: 18.07.2020)

ному стані. Але розробник не завжди має змогу відстежувати актуальні оновлення такої дуже великої кількості бібліотек, що зумовлює необхідність використання автоматичного пакетного менеджера. Для управління пакетами додатками доцільно використовувати пакетний менеджер NPM, що входить до складу Node.js. Node.js – програмна платформа, що перетворює JavaScript з вузькоспеціалізованої мови в мову загального призначення.

### **3.3 Використання технологій управління процесами збірки проекту**

У процесі складання проекту всі JavaScript модулі повинні бути об'єднані в єдиний модуль, який надалі буде завантажений клієнтом і підключений до єдиної html-сторінки, щоб в подальшому інтерпретуватися браузером.

Для управління процесами збірки, налагодження і тестування Front-end частини проекту обрана потокова система збирання проектів Gulp, написаних на JavaScript та інтегрована в багато середовищ розробки. Дозволяє гнучко управляти складанням проектів, уникнути трудомістких процесів складання, спрощує тестування, покращує структуру програми та зменшує кількість коду. Gulp поширюється через менеджер пакетів NPM під MIT ліцензією. Виконанням збирання також займається програмна платформа Node.JS. Gulp дозволяє створювати і гнучко налаштовувати окремі сценарії для різних завдань від збірки проекту до тестування і розгортання.

При розробці програми використовувався збиральник модулів Module bundler – Webpack, який розроблюється, складається з різних модулів, що містять код на JavaScript, а також різні модулі CSS стилів. Поточні браузери не мають підтримки модульності, а підключення кожного файлу окремо не має сенсу, тому що в загальному випадку завантаження необхідних ресурсів одним HTTP-запитом відбувається в кілька разів швидше, ніж завантаження за допо-

могою декількох окремих запитів. Тому виникає необхідність генерації статичних ресурсів програми.

Webpack надає всі перераховані раніше переваги. За замовчуванням webpack забезпечує тільки генерацію єдиного модуля додатка, однак він має підтримку розширень.

Для розробки інформаційної системи управління фінансовою діяльністю житлових комплексів були використані наступні бібліотеки разом з їх розширеннями для Webpack: babel і babel-loader, що забезпечує трансляцію коду зі стандарту ES6 в стандарт ES5; babel-preset-react, що забезпечує трансляцію коду на мові JSX, які були застосовані при здійсненні описання інтерфейсів користувача засобами Angular в код на JavaScript; style-loader і css-loader що забезпечують для завантаження модулів з CSS стилями.

В якості клієнта додатка для інформаційної системи управління фінансовою діяльністю житлових комплексів виступає браузер. Наявність численних браузерів для будь-якої платформи надає можливість розробнику реалізувати підтримку як для мобільних пристроїв, так і для звичайних комп'ютерів.

В якості CSS-фреймворку був обраний Bootstrap 3, який надає вільний набір інструментів для створення веб-додатків. Bootstrap має HTML і CSS шаблони оформлення для веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсів. Основні переваги Bootstrap [14]<sup>1)</sup>:

- дозволяє заощадити час і зусилля, використовуючи шаблони дизайну і класи;
- динамічні макети Bootstrap масштабуються на різні пристрої і дозволи екранів без будь-яких змін в розмітці, що забезпечує високу швидкість;
- простота у використанні та сумісність з різними браузерами;
- Bootstrap передбачає зручність використання за допомогою відкритості вихідних кодів і безкоштовного завантаження.

---

<sup>1)</sup> [14] Нільсен, Хоа Лоранжер. Технологія Angular 5. СПб., Символ-плюс, 2009, 512 с.

- Bootstrap надає можливості для створення адаптивного дизайну, що дозволяє оптимізувати додаток під різні варіанти ширини екрану пристрою.

Завдяки зазначених вище переваг, HTML-верстка буде відображатися по-різному в залежності від обраного для перегляду пристрою.

### 3.4 Засоби тестування системи

Тестування передбачає написання і запуску автоматичних ізольованих тестових модулів для будь-яких артефактів програми, написаної із застосуванням фреймворка Angular. Для автоматичного тестування front-end додатків застосовується зв'язка Karma-Jasmine. Karma – інструмент для запуску тестів безпосередньо з командного рядка, легко управляється за допомогою налаштувань, прописаних в файлі конфігурації, дозволяє виконувати покрокове налагодження тестів. Вміє взаємодіяти з безліччю фреймворків для написання тестів, а також є безкоштовним проектом з відкритим вихідним кодом. Jasmine – фреймворк Behavior-Driven Development, який застосовують для тестування JavaScript коду, має зрозумілий синтаксис, а також допомагає структурувати код тесту. Основними синтаксичними конструкціями Jasmine є [14]<sup>1)</sup>:

- `describe` – глобальна функція, яка об'єднує тести по будь-якою ознакою;
- `it` – функція, яка викликається для виконання конкретного тесту;
- `expect` – функція-обчислювач, яка визначає, що очікується від тесту, а також порівнює отриманий в ході тесту результат з очікуваним.

При розробці інтерфейсу інформаційної системи управління фінансовою діяльністю житлових комплексів основною вимогою є зручність роботи корис-

---

<sup>1)</sup> [14] Нільсен, Хоа Лоранжер. Технологія Angular 5. СПб., Символ-плюс, 2009, 512 с.



тувача з цим інтерфейсом. Таким чином, доступ до основних, найбільш часто використовуваних функцій користувач може отримати з головного меню на будь-якій сторінці програми. Деякі функції програми, такі як відображення актуальних записів, доступні за замовчуванням при першій завантаженні програму користувачем.

При великій кількості варіантів використання, інтерфейс завжди повинен залишатися компактним, при цьому він повинен продовжувати правильно працювати на будь-яких типах пристроїв, що підтримуються. Для певних складних варіантів використання додаток здатний динамічно створювати окремі модальні вікна. Крім операцій з очікуванням, анімація зустрічається і в інших функціях програми. Анімація в додатку досягається за допомогою спеціальної CSS бібліотеки «animate.css», також доступною через пакетний менеджер Gulp. Після авторизації, користувачеві доступний особистий кабінет. Однак, користувач в будь-який момент може переключити додаток на відображення головної сторінки, не залишаючи особистого кабінету. В особистому кабінеті так само є велика кількість функцій. Багато операцій особистого кабінету припускають початковий вибір даних зі спеціального списку. Для прискорення розробки інтерфейсу додатку застосовувалися SASS стилі. Оптимізація під мобільні пристрої досягається за рахунок використання фреймворка Bootstrap.

Програмна платформа Angular використовує архітектурну концепцію Model-View-Controller, однак, на відміну від стандартного MVC, тут також додається пакет «Service», який забезпечує взаємодію клієнта з сервером. Сервіси перетворюють отримані від сервера дані у вигляді JSON-об'єкта в модель, зрозумілу клієнту, і в зворотному порядку – роблять реалізацію моделі і контролера незалежної від представлення даних (рис. 3.3).

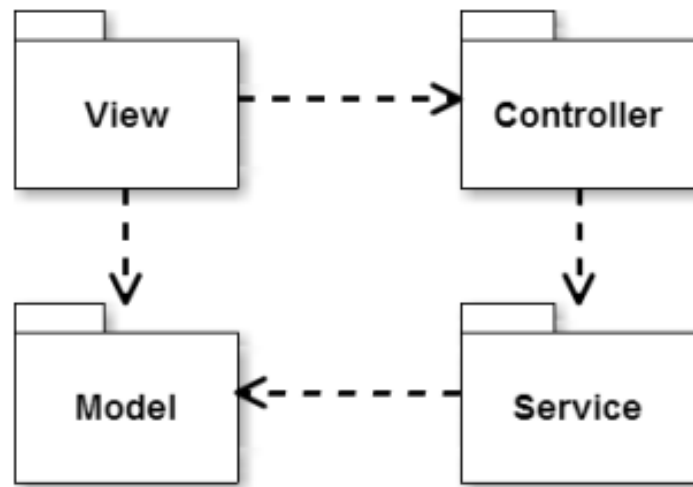


Рисунок 3.3 – Схема архітектури Model-View-Controller

У загальному випадку, в сервіс може бути інкапсульована будь-яка самостійна логіка, а сам сервіс реалізує патерн проектування Singleton. Controller бере на себе відстеження дій користувача і реакцію на них. Кожен контролер відповідає за реалізацію логіки роботи деякого елемента або всієї сторінки цілком. В даний пакет входять контролери сторінок і директив. Model є поданням сутностей системи. Як правило, кожна модель містить в собі поля, відповідні API, і деякі перетворюють функції, які використовуються тільки в коді для зручності розробника і не впливають на архітектуру. Пакет View містить в собі html-документи, що представляють відображення сторінки додатка, а також html-документи, що представляють відображення окремих директив [15]<sup>1)</sup>.

Структурні директиви дозволяють додавати, видаляти, повторювати або змінювати елементи DOM в залежності від значень певних змінних, властивостей компоненту або їх виразів:

- умовна директива – дозволяє додавати та прибирати елемент DOM в залежності від значення логічного виразу;

<sup>1)</sup> [15] Крістофер Косентіно. SPA технологія. Київ., BHV, 2016, 465с.

- директива перемикач – дозволяє змінювати елемент DOM в залежності від значення виразу;
- директива повтору – дозволяє створити елемент DOM для кожного елемента списку.

Можна зробити висновок, що фреймворк Angular є дуже гнучкий й зручним та володіє великою кількістю можливостей для створення сучасних односторінкових веб-додатків.

### **3.5 Проектування системи за методологією функціонального моделювання SADT**

При здійсненні проектування інформаційної системи управління фінансовою діяльністю житлових комплексів обрана методологія функціонального моделювання SADT (Structured Analysis and Design Technique) – стандарт IDEF0.

Засобами методології SADT, яка передбачає застосування деякої сукупності методів, правил і процедур, проведена побудова функціональної моделі об'єкта з обраної предметної області.

За допомогою графічної мови IDEF0, інформаційна система управління фінансовою діяльністю житлових комплексів відображається у вигляді набору взаємопов'язаних функціональних блоків, що надає перше уявлення про процеси у системі та взаємодію системи з зовнішнім оточенням.

Представлена у вигляді контекстної діаграми інформаційної системи управління фінансовою діяльністю житлових комплексів надає розробнику уявлення та загальний опис системи та її взаємодія з зовнішнім середовищем. Системи представлена єдиним функціональним блоком, а граничні стрілки виходять зовні, за межі, які простираються за межі аналізованої області (рис. 3.4).

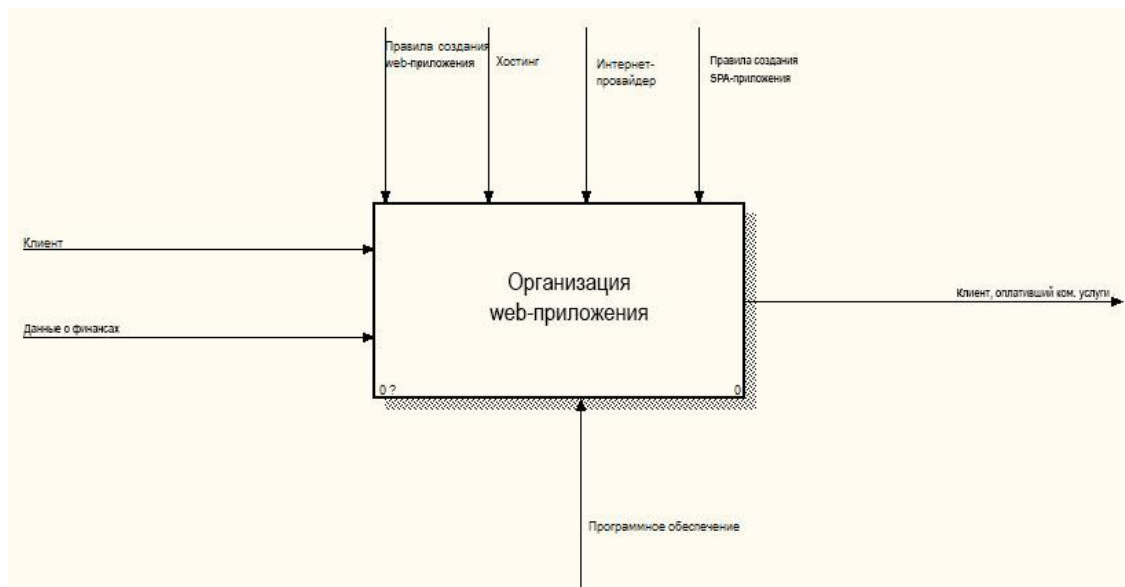


Рисунок 3.4 – Контекстна діаграма ІС управління фінансовою діяльністю ЖК

На контекстній діаграмі відображена головна робота системи «Організація web-додатку», що представляю собою повнофункціональну ІС управління фінансовою діяльністю житлових комплексів. На вхід подається інформація про клієнта (власників квартир ЖК і керівників ЖК) і дані, про господарсько-фінансові операції, що проводяться у житловому комплексі. Головна робота керується правилами створення web-системи, особливостями хостингу, можливостями, які надає Інтернет-провайдер та правилами створення додатків за допомогою SPA технології. На виході знаходиться клієнт, який оплатив послуги та отримав всю необхідну інформацію щодо фінансової діяльності ЖК.

Наступним кроком є виконання процесу функціональної декомпозиції, що передбачає розбиття єдиного блоку, яким представлена система на деяку кількість фрагментів. Схема, що надає уявлення про кожний фрагмент і взаємодію цих фрагментів і складає діаграму декомпозиції. В подальшому відбувається декілька ітерацій таких декомпозицій кожного з отриманих фрагментів системи на більш дрібні, до тих пір, поки не буде досягнуто потрібного рівня опису системи для проведення подальшого проектування. Кожна декомпозиція передба-

час по завершенні проведення процесу експертизи, де експерти в обраній предметній області, або замовники системи вказують на відповідність реальних процесів створеним діаграмам, якщо експертизою встановлені невідповідності, то вони виправляються. Таким чином досягається відповідність змодельованої системи реальним процесам, які повинна надавати система.

Після декомпозиції контекстної діаграми на рис. 3.4 отримано три блока основної роботи. Блок «Розробка web-системи» передбачає здійснення розробки ІС для житлових комплексів на локальній станції. Це блок має наступні види работ: розробка інтерфейсу, розробка скриптів, розробка баз даних. Даними, що знаходяться на вході у цей блок є: дані про фінансові операції, які проводились в ЖК, які будуть додавати у систему адміністратори. Управління виконується за допомогою правил створення Web-системи та правил проведення фінансово господарських операцій у житловому комплексі. А в якості механізму виступає програмне забезпечення, обране для проведення розробки ІС, а результат – розроблена і готова для використання ІС управління фінансами ЖК.

Блок з назвою «Розміщення Web-системи» виконує функцію, що передбачає отримання доменного ім'я з подальшим розміщенням на хостингу. Входом для цього блоку є готова Web-система для розміщення, а управління здійснюється особливостями хостингу та Інтернет-провайдером, в якості механізму обране програмне забезпечення для розробки системи.

Блок «Реалізація додатку» передбачає наявність всіх програмних і апаратних складових для проведення процесу розробки ІС управління фінансовою діяльністю ЖК. Цей блок має входи: дані про фінансово-господарські операції, дані про клієнтів системи та доменне ім'я, яке отримано для розміщення ІС на хостингу. Управління цими визначеними видами работ забезпечується правилами створення Web-системи за обраними технологіями та за вимогами Інтернет-провайдера. Механізмом виступає обране програмне забезпечення, а в результаті отримаємо клієнта системи, який здійснив засобами ІС оплату комуна-

льних послуг та отримав доступ до інформації по кожній з фінансових операцій. Діаграма декомпозиції наведена на рис. 3.5.

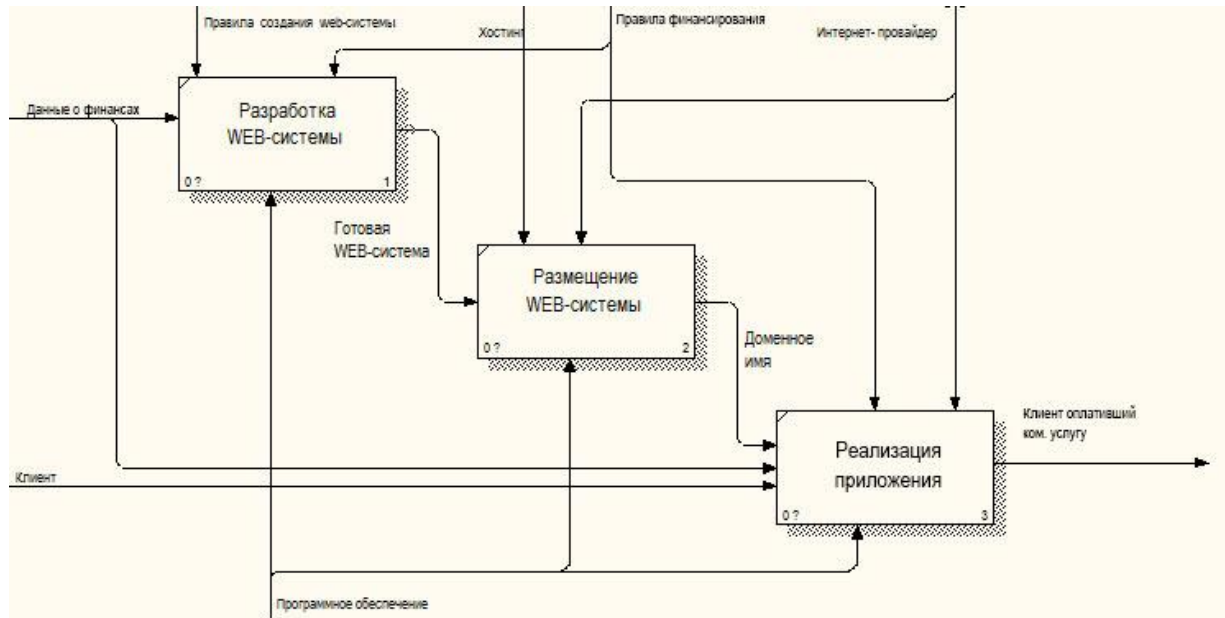


Рисунок 3.5 – Діаграма декомпозиції Блоку 1– «Організація веб-додатку»

При декомпозиції першого блоку «Розробка Web-системи» отримано блоки:

- «Створення інтерфейсу», управління – правила створення Web-системи; механізм – програмне забезпечення; вихід – інтерфейс Web-системи.
- «Розробка скриптів» – входом є інтерфейс; механізм, як визначено у всіх роботах – програмне забезпечення; вихід – готова ІС управління фінансовою діяльністю ЖК.
- «Створення БД» – даний блок управляється правилами розробки БД; робота виконується за допомогою програмного забезпечення; виходом є – готова БД.
- «Наповнення БД», ця робота має три входи: готова БД, робоча Web-система, дані про фінансові операції; керується правилами створення

Web-системи; механізмом є програмне забезпечення, вихід з даної роботи є готова ІС управління фінансовою діяльністю ЖК, яка переходить входом на наступну роботу.

Діаграма декомпозиції описаного першого блоку наведена на рис. 3.6.

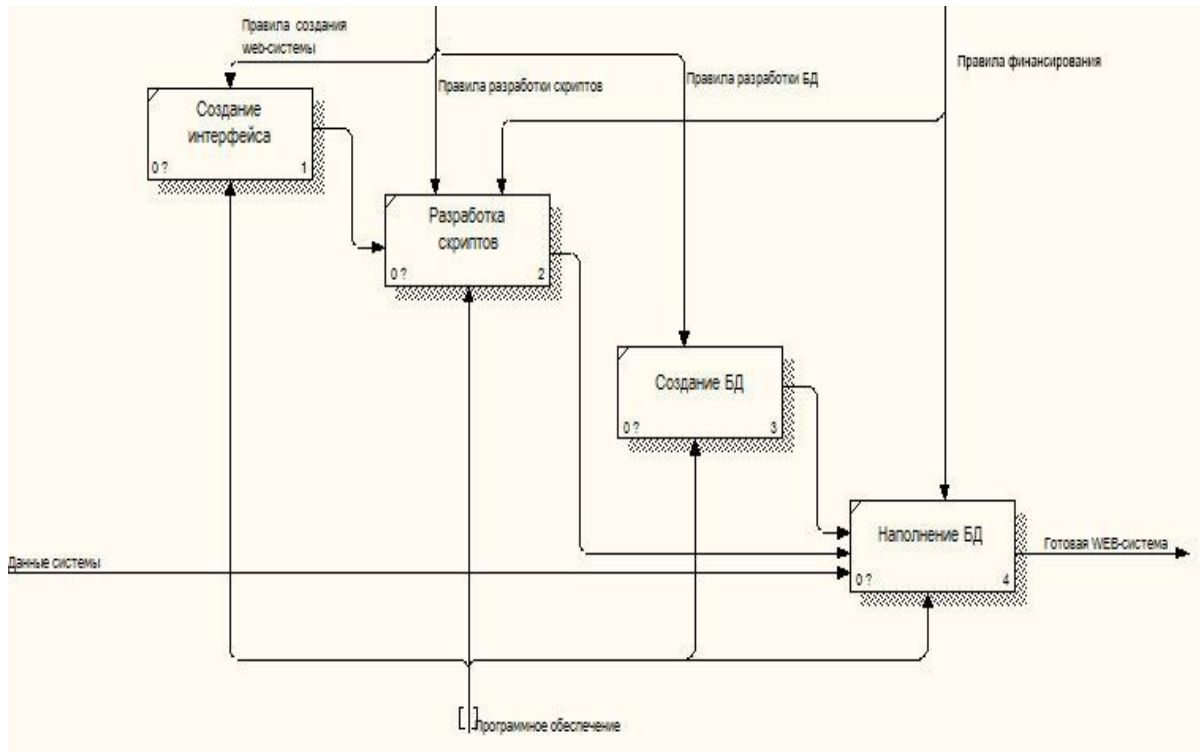


Рисунок 3.6 – Діаграми декомпозиції першого A1 блоку

При декомпозиції другого блоку «Розміщення Web-системи» виділені чотири роботи. Робота «Визначити доменне ім'я» керується бажаннями замовника та Інтернет-провайдером; виконується за допомогою програмного забезпечення, де виходом є ID системи, тобто саме доменне ім'я. Робота «Визначення зони реєстрації» передбачає на вході доменне ім'я, управляється бажаннями замовника та Інтернет-провайдером, а в якості механізму виступає програмне забезпечення, за допомогою якого здійснюється розробка; виходом роботи «Визначення зони реєстрації» є надане доменне ім'я ІС управління фінансовою діяльністю

ЖК. Робота під назвою «Реєстрація на хостингу» буде керуватися особливостями обраного хостингу та Інтернет-провайдером; в якості механізму обране програмне забезпечення, а виходом буде логін і пароль облікового запису на сервері обраного хостингу. Наступна робота «Перенесення сайту на хостинг» передбачає наявність логіну і паролю від облікового запису на хостингу-сервері і готова інформаційна система; результатом роботи є розміщена на хостингу ІС управління фінансовою діяльністю ЖК.

При декомпозиції третього блоку «Реалізація додатку» буди виділені п'ять робот. Опишемо кожну з робот. Робота «Організація реєстрації» передбачає наявність виділеного доменного ім'я та даних про кожного з клієнтів (власників квартир житлового комплексу); управління роботою здійснюється за правилами проведення реєстрації клієнтів у системі; механізмом виступає наявне програмне забезпечення; результатом роботи є зареєстрований користувач.

Робота «Організація оплати послуг» передбачає наявність механізму здійснення оплати комунальних послуг для зареєстрованих користувачів, який буде здійснювати оплату тієї чи іншої послуги; механізмом є програмне забезпечення. Робота «Розробка механізму додатку» керується API-системи та Інтернет-провайдером; механізм роботи є програмне забезпечення.

Наступна робота «Реалізація фінансової діаграми» дозволяє клієнтам системи отримувати актуальні дані про всі фінансові операції, які здійснені органами управління житловим комплексом системи; механізмом є програмне забезпечення; виходом будуть аналізовані і надані у вигляді діаграми фінансові дані житлового комплексу. Робота «Реалізація обліку фінансів» забезпечує виконання обліку фінансово-господарських операцій, що здійснені житловим комплексом та відображені інформаційною системою. Діаграма декомпозиції цього блоку наведено на рис. 3.7.



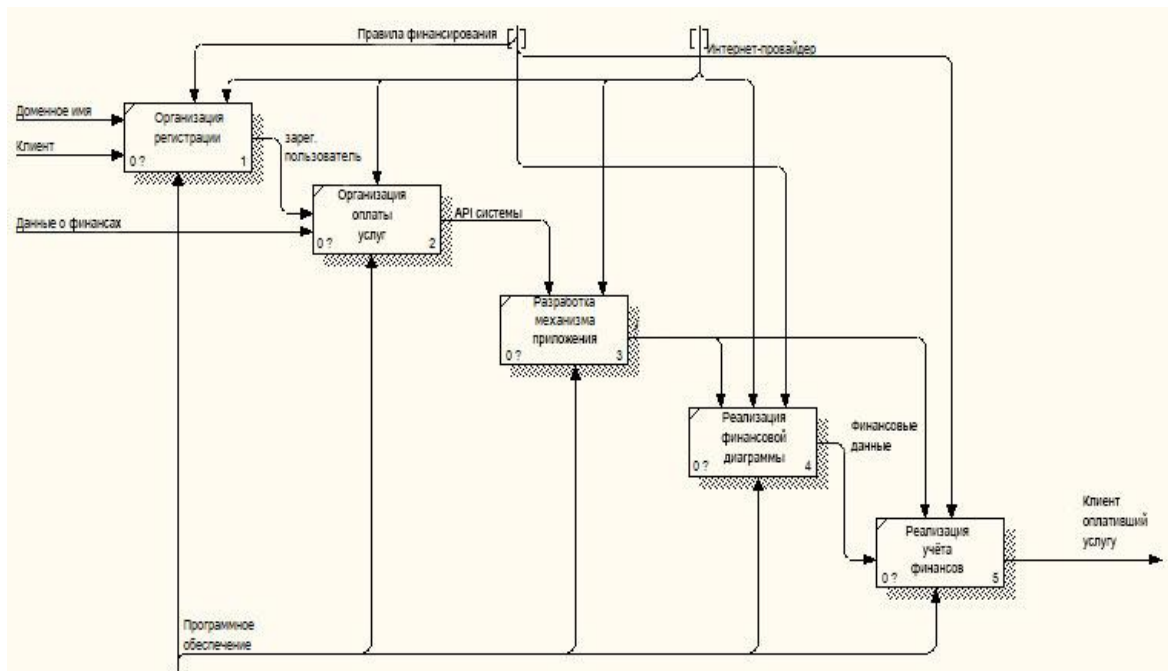


Рисунок 3.7 – Діаграми декомпозиції блоку «Реалізація додатку»

### 3.6 Проектування бази даних системи

Для виконання всіх зазначених функцій у інформаційній системі управління фінансовою діяльністю житлового комплексу необхідно здійснити проектування та реалізацію бази даних. Логічна структура спроектованої бази даних для інформаційної системи управління фінансовою діяльністю житлового комплексу наведена у вигляді моделі представлення даних «сутність-зв'язок», яка ґрунтується на важливій семантичній інформації про обрану предметну область для логічного представлення даних. Вона визначає значення даних в контексті їх взаємозв'язку з іншими даними. Діаграма «сутність-зв'язок» бази даних інформаційної системи управління фінансовою діяльністю житлових комплексів наведена на рис. 3.8.

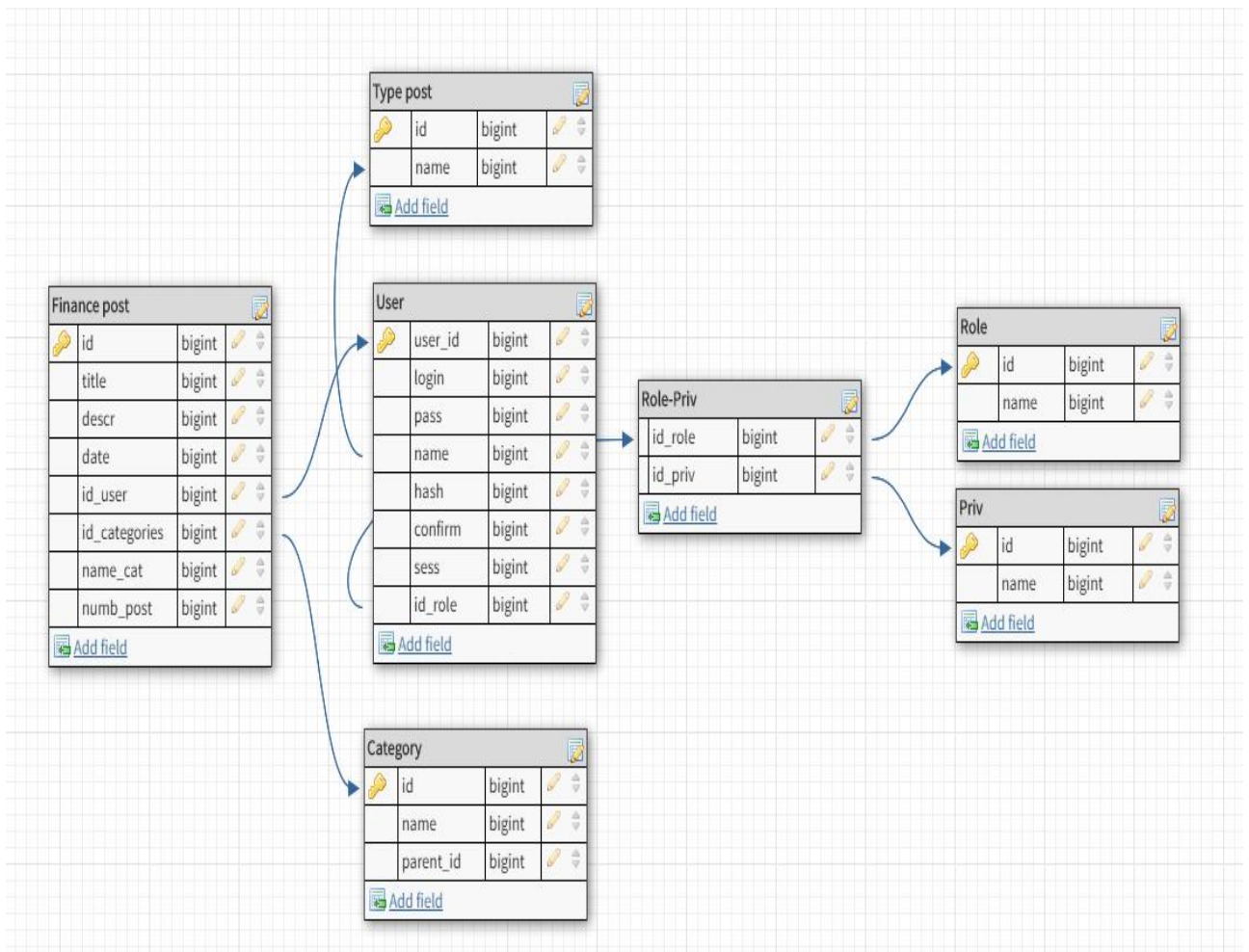


Рисунок 3.8 – Модель «сутність-зв'язок» ІС

Всі зазначені сутності являють собою визначену кількість атрибутів, які описують властивості всіх членів для кожної з сутностей.

Для бази даних ІС управління фінансовою діяльністю ЖК були виділені наступні сутності: наступні сутності: фінансові записи, користувачі, типи запису, категорії, ролі, привілеї та ролі-привілеї. Кожна сутність містить атрибут, який однозначно ідентифікують кожен з екземплярів сутності, такий атрибут і є первинним ключем.

Виходячи з постановки завдання до розробки ІС управління фінансовою діяльністю ЖК розроблена база даних, яка містить в собі наступні дані: про ко-

ристувачів; про привілеї для користувачів; про ролі користувачів; про кожну з фінансових операцій; про категорії; про перелік адміністраторів системи.

Основними об'єктами в системі визначені наступні: фінансовий запис та користувач. При проектуванні бази даних ІС управління фінансовою діяльністю житловими комплексами визначені сутності та їх атрибути.

Сутність «Фінансові записи» містить наступні атрибути: id\_запису, заголовок, опис запису, дата додавання, id\_користувача, id\_категорії, назва категорії, номер запису, які наведено у табл. 3.1.

Таблиця 3.1 – Сутність «Фінансові записи»

№	Атрибут	Тип
1	id	int
2	title	varchar(255)
3	descr	text
4	date	date
5	id_user	int
6	id_categories	int
7	name_cat	varchar(255)
8	numb_post	int

Сутність «Привілегії» містить наступні атрибути: id\_привілеї, назва привілеї, які наведено у табл. 3.2.

Таблиця 3.2 – Сутність «Привілегії»

№	Атрибут	Тип
1	id	int
2	name	varchar(255)

Сутність «Користувачі» містить наступні атрибути: id\_користувача, логін, пароль, ім'я, hash-код, підтвердження, сесія, id\_ролі, які наведено у табл. 3.3

Таблиця 3.3 – Сутність «Користувачі»

№	Атрибут	Тип
1	user_id	int
2	login	varchar(255)
3	password	varchar(32)
4	name	varchar(255)
5	hash	varchar(32)
6	confirm	enum('0','1')
7	sess	varchar(32)
8	id_role	tinyint(4)

Сутність «Типи запису» містить наступні атрибути: id\_типу, назва типу, які наведено у табл. 3.4.

Таблиця 3.4 – Сутність «Тип запису»

№	Атрибут	Тип
1	id	int
2	name	varchar(255)

Сутність «Категорії» містить наступні атрибути: id\_категорії, назва категорії, id\_батьківської категорії, які наведено у табл. 3.5.

Таблиця 3.5 – Сутність «Категорії»

№	Атрибут	Тип
1	id	int
2	name	varchar(255)
3	parent_id	tinyint(4)

Сутність «Ролі» містить наступні атрибути: id\_ролі, назва ролі, які наведено у табл. 3.6.

Таблиця 3.6 – Сутність «Ролі»

№	Атрибут	Тип
1	id	int
2	name	varchar(255)

Сутність «Ролі-привілеї» містить наступні атрибути: id\_ролі, id\_привілеї, які наведено у табл. 3.7.

Таблиця 3.7 – Сутність «Ролі-Привілеї»

№	Атрибут	Тип
1	id_role	int
2	id_priv	int

Зв'язки між всіма визначними сутностями бази даних ІС управління фінансовою діяльністю житлових комплексів наведено на діаграмі «сутність-зв'язок», а типом зв'язків є зв'язок «один-до-багатьох» для всіх сутностей.

## **4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІС УПРАВЛІННЯ ФІНАНСОВОЮ ДІЯЛЬНІСТЮ ЖИТЛОВИХ КОМПЛЕКСІВ**

Розроблена інформаційної системи управління фінансовою діяльністю житлових комплексів є зручним інструментом для використання власниками квартир житлового комплексу та керівництвом житлового комплексу для здійснення оплат за отримані комунальні послуги та відстеження всіх фінансово-господарських операцій у житловому комплексі.

Інформаційна система забезпечує доступ до актуальної інформації користувачу-адміністратору системи, в якості якого може виступати керівництво і бухгалтер житлового комплексу та користувачу-клієнту, в якості яких виступає власники квартир житлового комплексу. Система забезпечує зручний інтерфейс користувача для всіх категорій та надає можливість здійснення перегляду фінансової інформації, а саме нарахування оплати за отримані комунальні послуги, а також можливість формування фінансової звітності по всім здійсненим господарсько-фінансовим операціям, які буди здійсненні житловим комплексом. В системі передбачена обов'язкова реєстрація та подальша авторизація всіх користувачів.

### **4.1 Розробка додатку для користувача-клієнта**

Метою магістерської роботи є розробка інформаційної системи управління фінансовою діяльністю житлових комплексів, що надає користувачам системи актуальну фінансову інформацію щодо здійснення нарахування і оплати отриманих комунальних послуг, планування майбутніх витрат та фінансові звіти про фінансово-господарську діяльність. Інформаційна система надає можливість здійснення обліку бухгалтерських і господарських операцій, що відбуваються у житловому комплексі. Головна сторінка інформаційної системи, яка ві-

дображається при першому вході в систему користувача-клієнта, надає ліворуч розташоване меню з розділами системи, основна область – це область контенту, де буде відображатися вся інформація. У верхній частині сторінки розміщена назва системи з логотипом, що повідомляє користувачу поточну дату, профіль та призначення системи. В верхній частині сторінки відображається ім'я користувача, з можливістю редагування профілю (рис. 4.1).

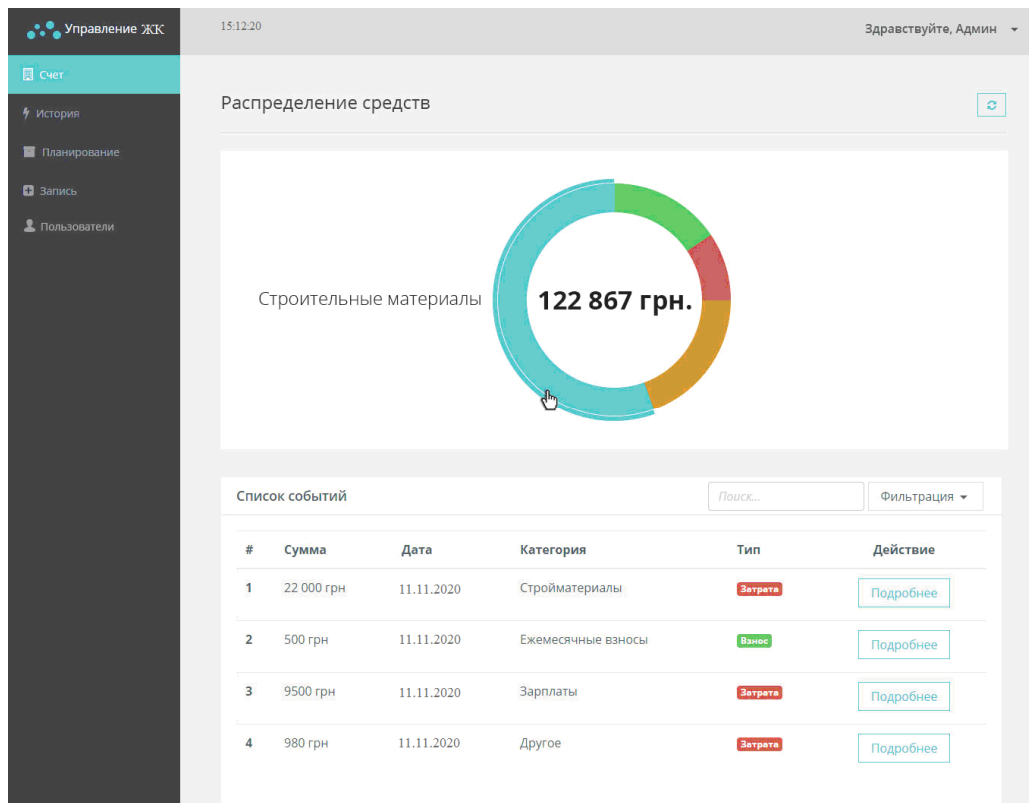


Рисунок 4.1 – Початкова сторінка інформаційної системи управління фінансовою діяльністю житлових комплексів

Головна сторінка ІС управління фінансовою діяльністю ЖК поєднана разом зі сторінкою «Распределение средств», що передбачає відображення всіх фінансових операцій, які здійснені на поточну дату житловим комплексом, де можливо детальніше ознайомитися з кожною з операцій, натиснувши кнопку

«Подробнее» напротив обраної операції. У верхній частині відображається діаграма витрачених коштів, розбита по категоріям витрат, що позначені кольором для зручності. При наведенні на конкретний сектор виводиться відповідна інформація: назва категорії та сума, витрачена на цю категорію (рис. 4.2).

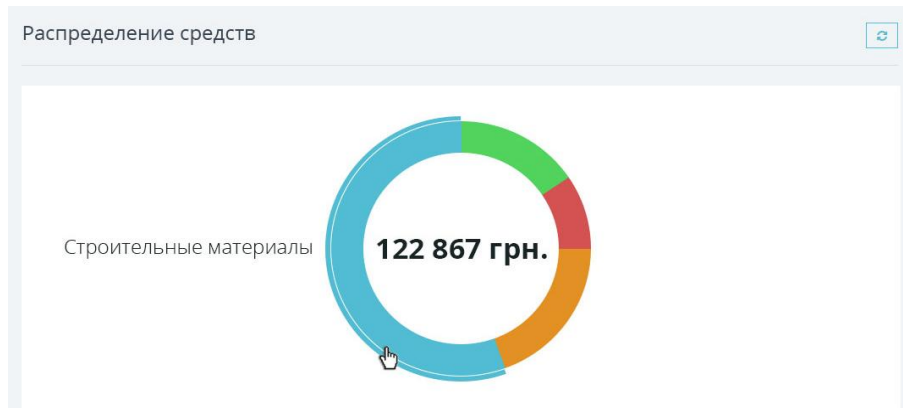


Рисунок 4.2 – Відображення в ІС діаграми розподілу коштів за витратами

В нижньої частині області основного контенту наведено перелік подій, а саме фінансових операцій, які відбувались на поточну дату у ЖК (рис. 4.3).

Список событий					
			Поиск...	Фильтрация ▾	
#	Сумма	Дата	Категория	Тип	Действие
1	22 000 грн	11.11.2020	Стройматериалы	Затрата	Подробнее
2	500 грн	11.11.2020	Ежемесячные взносы	Внос	Подробнее
3	9500 грн	11.11.2020	Зарплаты	Затрата	Подробнее
4	980 грн	11.11.2020	Другое	Затрата	Подробнее

Рисунок 4.3 – Область контента «Список событий»



Розділ ІС «Список событий» містить наступну інформацію, щодо кожної операції: номер у переліку записів, сума здійсненої фінансової операції, дата проведення фінансової операції, категорія фінансових витрат до якої віднесено кожну з фінансових операцій, що передбачає подальший пошук за категоріями. Кнопка «Подробнее» відображає більш детальну інформацію по кожній з операцій. Така інформація буде відображена на окремій сторінці. Наявність розділів «Поиск» і «Фильтрация» дозволяють знайти запис за будь-якою умовою, наприклад сума операції, або за типом операції. За допомогою реалізованої у системі функції фільтрації користувач може знайти записи за датою, або за типом операції до якої вона занесена в систему (рис.4.4).

The image shows a web interface for date selection. At the top, there is a search bar labeled "Поиск..." and a dropdown menu labeled "Фильтрация". Below this is a calendar for January 2019. The calendar grid shows days from 27 to 31. The 7th and 29th are highlighted in blue. Below the calendar, there are buttons for "Today" and "Clear". At the bottom, there is a dropdown menu labeled "Категория" and a button labeled "Фильтровать".

Рисунок 4.4 – Функція вибору інформації за умовами дата та тип операції

При виборі користувачем функції «Подробнее», відобразиться сторінка з детальною інформацією щодо обраної фінансової операції: сторінка з номером запису, якій надає адміністратор системи при внесенні цієї операції в систему. В залежності від типу до якого була віднесена ця операція (витрата або внесок), хедер запису набуває відповідного кольорового забарвлення ( витрата – черво-

ного, внесок –синього). Детальна інформація щодо обраної фінансовій операції містить наступну інформацію: сума операції, тип операції, дата проведення та детальний опис (рис. 4.5).

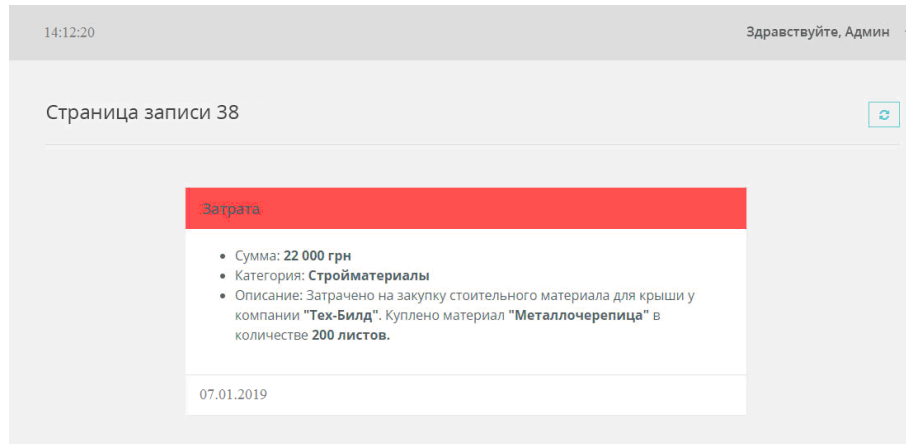


Рисунок 4.5 – Сторінка, що відображає детальну інформацію щодо обраної фінансової операції

Якщо у меню, розташованому на головній сторінці ліворуч обрано пункт «Планирование финансов», користувач отримує актуальну інформацію щодо стану фінансів на рахунку житлового комплексу, а саме: загальний залишок коштів, а також всі витрати, які відображаються системою за типами. В розділі «Расходы» наведена інформація щодо витрат за типами: заробітна плата, будівельні матеріали, інші види витрат.

Графічно відображено, для кожного типу витрат, витрачена сума та залишок коштів у житлового комплексу саме для цієї категорії витрат, що дозволяє кожному власнику квартири у житловому комплексі обстежувати рух коштів та їх витрати і мати актуальну інформацію в любий зручний час. А бухгалтеру житлового комплексу вчасно звітувати про витрати власників на ту ці іншу витратну статтю (рис. 4.6).

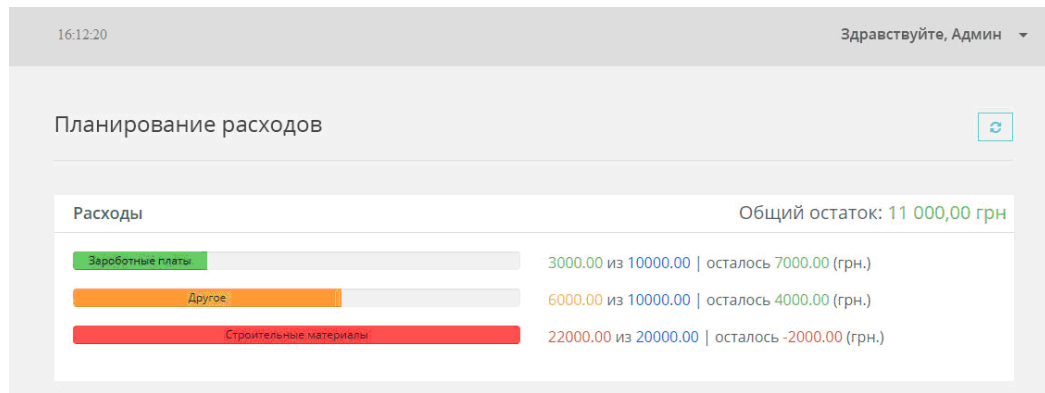


Рисунок 4.6 – Сторінка «Планирование расходов» при перегляді витрат ЖК

Планування, а також врахування оплат і інші фінансові операції здійснює у системі лише адміністратор системи, яким може бути як голова житлового комплексу, так і бухгалтер – саме в цієї категорії користувачів є ці повноваження. Адміністратор вносить дані у систему, розподіляє між категоріями (типами). Для адміністратора існує окремий одно сторінковий додаток для редагування та внесення таких даних.

Використовуючи ІС управління фінансовою діяльністю житлових комплексів, користувачі-клієнти мають можливість засобами системи здійснювати оплату всіх комунальних послуг, які власниками квартир були отримані.

Можливість здійснення оплати комунальних послуг реалізовано за допомогою програмного інтерфейсу додатку (API) від системи прийому платежів під назвою «EasyPay». Даний сервіс надійний і існує на нашому ринку України з 2007 року. Сервіс EasyPay API надає дуже багато можливостей, а саме зручною функцією сервісу є можливість для користувачів сплачувати послуги не виходячи з своєї системи. За допомогою даної системи користувачі мають змогу оплатити безліч комунальних послуг, таких як електропостачання, водопостачання, теплопостачання та багато іншого. Влаштувати API від сервісу EasyPay можливо тільки після того, як доменне ім'я та система пройдуть перевірку на

справжність адміністраторами платіжної системи EasyPay. Сторінка оплати комунальних послуг наведено на рис. 4.7.

18:23:20 Здравствуйте, Админ

Оплата коммунальных услуг: ↻

### Выбор способа оплаты

Лицевой счет / телефон

Сумма

Услуга

Описание

грн.

[Войти или Зарегистрироваться](#)

---

☒ Платежная карта VISA/Mastercard

Номер карты

MM / ГГ

CVV код

Комиссия

грн.

---

☐ Банковский счет

грн.

---

☐ Получить оригинал квитанции почтой

Услуги предоставляются согласно публичного договора о предоставлении услуг

Email для получения квитанции

На данный email автоматически будет отправлена квитанция

Всего к оплате:  грн.

[Оплатить](#)

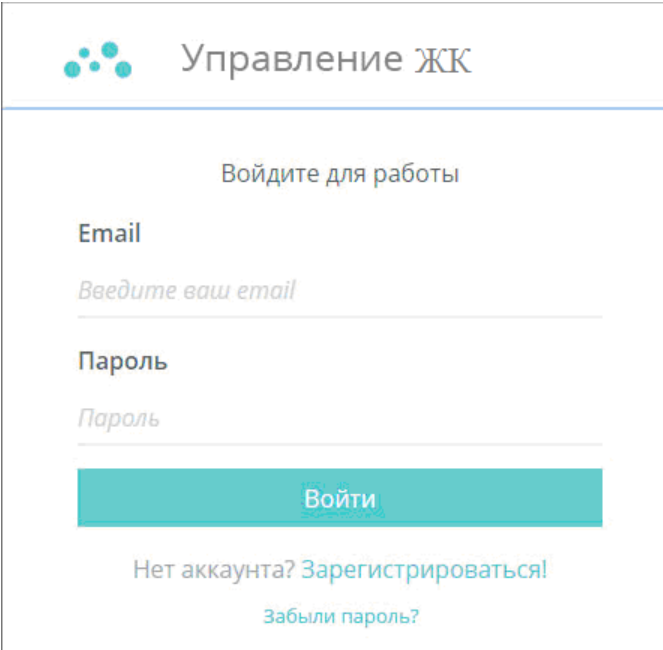
Рисунок 4.7 – Сторінка ІС «Оплата коммунальных услуг»

Основним завданням інформаційної системи управління фінансовою діяльністю житлових комплексів для клієнтів-користувачів, якими є власники квартир житлового комплексу, є забезпечення системою реалізації функцій оплати комунальних платежів та доступ до актуальної інформації щодо здійснення фінансової діяльності керівництвом житлового комплексу, а також прозоре планування розподілу фінансових потоків між житлово-комунальними видами доходів та витрат.

## 4.2 Реалізація функції авторизації та реєстрації у системі

Користувачам, для входу в інформаційну систему управління фінансовою діяльністю житлових комплексів, необхідно буде ввести унікальний логін і пароль, інакше система не надасть доступ клієнту-користувачу до фінансової інформації, змогу оплачувати послуги, редагувати свій профіль та інше. Користувач буде мати доступ лише до сторінки авторизації та відновлення паролю.

На сторінці авторизації є можливість перейти на сторінку реєстрації, а також відновити забутий пароль. Після того, як користувач перейде за посиланням «Забули пароль?», він попадає на сторінку відновлення паролю. Далі йому буде запропоновано ввести свою електронну адресу. Після цього на дану електронну адресу буде відісланий новий випадково згенерований пароль. Далі користувач зможе його відредагувати в особистому кабінеті системи (рис. 4.8).



Управление ЖК

Войдите для работы

Email

Введите ваш email

Пароль

Пароль

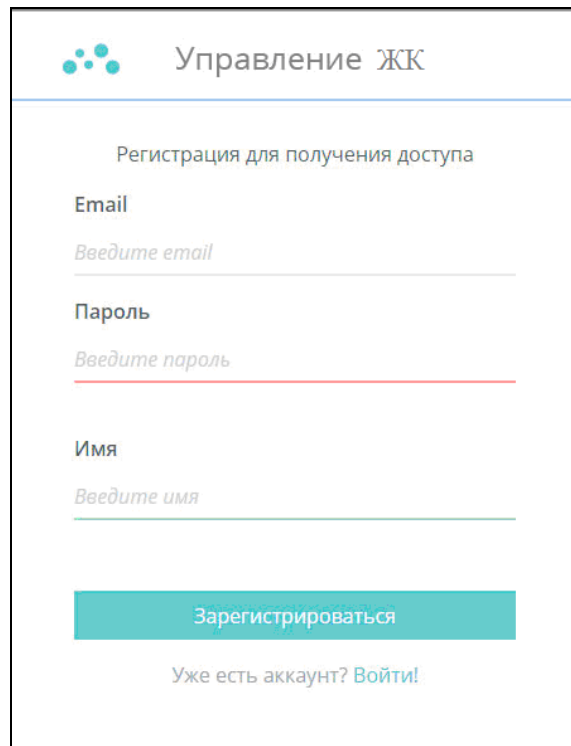
Войти

Нет аккаунта? Зарегистрироваться!

Забыли пароль?

Рисунок 4.8 – Сторінка авторизації користувачів в ІС

Якщо логін і пароль введені правильно, то користувачу дозволяється увійти в систему. Якщо ж користувач ввів невірний або логін або пароль, то користувачу надається повідомлення: «Неправильный логин или пароль». При вході в систему користувач, який буде авторизований матиме доступ до головної сторінки, а інакше користувачу буде запропоновано авторизуватись в системі засобами заповнення відповідної форми, яка має наступні обов'язкові поля для заповнення: електронної адреси, пароля, імені користувача. Для зручності заповнення на сторінці знаходяться деякі рекомендації (рис. 4.9).



Управление ЖК

Регистрация для получения доступа

Email

Введите email

Пароль

Введите пароль

Имя

Введите имя

Зарегистрироваться

Уже есть аккаунт? Войти!

Рисунок 4.9 – Сторінка реєстрації користувача в ІС

Якщо реєстрація пройшла успішно, користувач перенаправляється на сторінку, яка повідомляє йому про успішну реєстрацію та видає повідомлення «Реєстрація пройшла успішно! Активуйте ваш профіль через e-mail». Після цього користувач має перейти в свою електронну скриньку та підтвердити свій облі-

ковий запис. Така функція реалізована за допомогою хеш-функції, яка передається користувачу в листі на електронну адресу. Якщо користувач при реєстрації намагається ввести вже наявний в системі логін, то профіль не реєструється і нижче форми з'являється повідомлення «Користувач з таким логіном вже зареєстрований на сайті. Введіть інший логін!»

### **4.3 Розробка додатку для користувача-адміністратора**

При здійсненні розробки інформаційної системи управління фінансовою діяльністю житлових комплексів передбачено постановкою завдання наявності в системі користувача-адміністратора для якого був розроблений спеціальний одно сторінковий додаток, який реалізує всі необхідні функції. В даній системі адміністратор має доступ до всіх тих самих сторінок, до яких має користувач. Відмінність полягає лише в тому, що для адміністратору відкривається доступ до деяких окремих сторінок, на яких він має можливість вносити або виправляти (редагувати) інформацію, редагувати чи видаляти користувачів і таке інше.

Для певного класу систем інтерфейс адміністратора є обов'язковою частиною інфраструктури. Це вид інтерфейсу, доступний довіреним особам системи. Інтерфейс адміністратора, читаючи метадані з моделі, надає потужний і повністю готовий інтерфейс.

Для користувача-адміністратора в системі передбачена сторінка керування користувачами в системі: «Редактирование пользователей», що надає можливість адміністратору підтверджувати нових користувачів, або редагувати та видаляти за потреби. У першій половині цієї сторінки виводяться перелік нових користувачів. Натиснувши на кнопку «Подтвердить» адміністратор активує нового користувача і у нього з'явиться доступ до системи. Також адміністратор може видалити користувача, натиснувши на червоні хрестик біля потрібного запису (рис. 4.10).

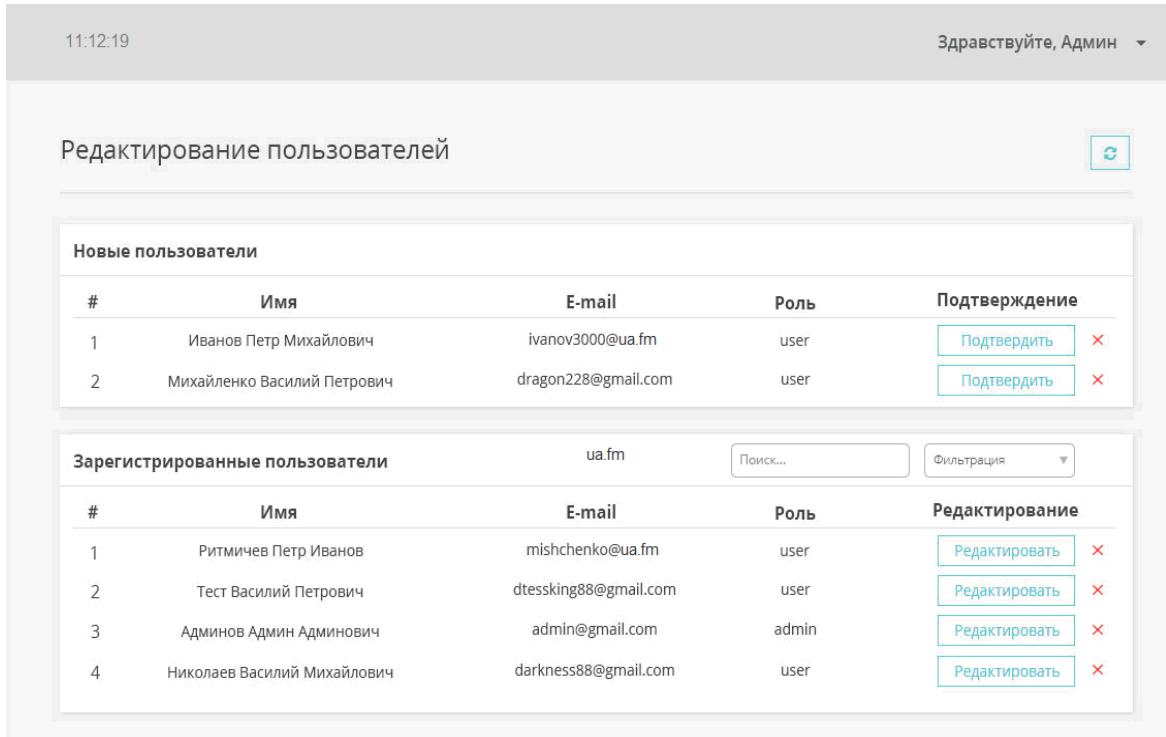


Рисунок 4.10 – Сторінка «Редактирование пользователей»

Таке підтвердження реалізовано для того, щоб ніхто сторонній, крім самих жителів комплексу (власників квартир), які користується даною системою, не зміг зареєструватись в системі і мати доступ до перегляду фінансової інформації. Також на цій сторінці можна редагувати ролі користувачів. Адміністратор має можливість пошуку конкретного користувача через строку пошуку або за допомогою фільтрів.

Наступна сторінка надає адміністратору системи змогу додавання нової події, або категорії в систему: «Добавление события или категории», що дозволяє додавати нові події або створювати нові категорії для занесення фінансових записів.

Для додавання нової події реалізовано форму у частині вікна, що розташована ліворуч, та підтверджується натисканням кнопки «Добавить». Спочатку



вибрати категорію запису, далі вибрати тип запису (внесок чи витрата), після цього вказати суму та опис запису, якщо потрібно. Праворуч розташована форма де адміністратору потрібно вказати тільки назву категорії та ліміт коштів для даної категорії. Після цього натиснути кнопку «Добавить» і нова категорія буде додана (рис. 4.11).

21:07:17

Здравствуйте, Админ

Добавление события или категории

Добавить событие

Выберите категорию

Стойматериалы

Выберите тип

☐ Внос

☒ Затрата

Введите сумму

0

Описание

Добавить

Добавить категорию

Введите название

Введите лимит

0

Валидация нет ошибки

Добавить

Рисунок 4.11 – Сторінка «Добавление события или категории»

Також дана система адаптована і для мобільних та портативних пристроїв, засобами сучасного CSS, що дозволяє досить просто реалізовувати адаптивні додатки. Приклад адаптивності сторінки для мобільного пристрою наведений на рисунку 4.12.

iPhone 375 x 667 87%

Управление ЖК

Регистрация для получения доступа

Email  
Введите email

Пароль  
Введите пароль  
Error message.

Имя  
Введите имя  
Success message.

Зарегистрироваться

Уже есть аккаунт? [Войти!](#)

Рисунок 4.12 – Пример адаптивной страницы регистрации на мобильном устройстве

В результате здійснена програмна реалізація інформаційної системи управління фінансовою діяльністю житлових комплексів, що дозволяє завдяки реалізованому функціоналу, використовувати систему будь-яким житловим комплексом для забезпечення власників квартир актуальною фінансовою інформацією щодо здійснення фінансової діяльності керівництвом житлового комплексу, та засобами системи проводити оплати комунальних послуг.

Система має простий і зрозумілий інтерфейс, орієнтований на користувачів різних вікових категорій, які з легкістю можуть оволодіти навичками роботи з системою. Застосування технології Single Page Application, дозволило реалізувати достатньо швидко у користуванні систему, що виділяє її з позиції ресурсоефективності та ресурсозбереження.

## ВИСНОВКИ

В результаті виконання магістерської роботи, була розроблена інформаційна система управління фінансовою діяльністю житлових комплексів, що являє собою інформаційну систему з Web-інтерфейсом, що надає користувачам актуальну фінансову інформацію щодо здійснення нарахування і оплати отриманих комунальних послуг, планування майбутніх витрат та фінансові звіти про фінансово-господарську діяльність житлового комплексу.

В умовах ринкової економіки житлово-комунальні послуги є товаром, який реалізується на ринкових засадах, тому створення інформаційної системи управління фінансовою діяльністю житлових комплексів, головним завданням якої є надання послуг з оплати та відстеження всіх фінансових операцій членів співвласників житлових комплексів є актуальним завданням.

В результаті проведеного проектування інформаційної системи виконано аналіз та визначення функціональних вимог і технологій розробки системи; проведено дослідження та вибір архітектурних рішень і програмної платформи реалізації системи засобами теорії прийняття рішень; розроблена база даних системи; здійснена програмна реалізація додатків користувачів системи.

Завдяки використанню різних елементів управління, система має привабливий, функціональний, зручний і інтуїтивно-зрозумілий інтерфейс користувача з можливістю швидкої модернізації у разі потреби. Застосування технології створення одно сторінкових додатків Single Page Application та сучасної програмної платформи Angular, дозволило розробити достатньо швидко у користуванні систему, що виділяє її з позиції ресурсоефективності та ресурсозбереження. Завдяки сучасним технологіям розробки інформаційна система управління фінансовою діяльністю житлових комплексів може бути легко розширена та доповнена новими функціями.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Димченко О. В. Житлово-комунальне господарство в реформаційному процесі: аналіз, проектування, управління: монографія. Х., ХНАМГ, 2009. 356 с.
2. В. І. Торкатюк, Ю. Ю. Усенко. Оценка экономической эффективности решения задач управления жилищно- коммунальной структурой города. URL: [https://pidru4niki.com/79173/ finans/finansova\\_diyalnist\\_ zhitlovo-ekspluatatsiynogo\\_kompleksu](https://pidru4niki.com/79173/finans/finansova_diyalnist_zhitlovo-ekspluatatsiynogo_kompleksu) (дата звернення: 06.04.2020).
3. Болотіна В. В. Огляд популярних JAVASCRIPT фреймворків. URL: [https://conf.ztu.edu.ua/wp-content/uploads/2019/06/ 42.pdf](https://conf.ztu.edu.ua/wp-content/uploads/2019/06/42.pdf) (дата звернення: 18.07.2020).
4. Популярные фреймворки JavaScript. URL: <https://vc.ru/dev/147263-populyarnye-freymvorki-javascript> (дата звернення: 18.07.2020).
5. Введение в KnockoutJS. URL: [https://metanit.com/web/ knockout/1.1.php](https://metanit.com/web/knockout/1.1.php) (дата звернення: 24.07.2020).
6. Сравнение JavaScript фреймворков Vue.js, React и Angular (2019). Статьи о программировании mkdev. URL: [https://mkdev.me/posts/ sravnenie-javascript-freymvorkov-vue-js-react-i-angular-2019](https://mkdev.me/posts/sravnenie-javascript-freymvorkov-vue-js-react-i-angular-2019) (дата звернення: 02.08.2020).
7. Разработка одностраничных web-приложений – Создание SPA-приложений. URL: <https://avada-media.ua/services/single-page-application-development/> (дата звернення: 14.09.2020).
8. Что такое SPA или одностраничный портал. URL: <http://www.codenet.ru/webmast/js/spa/> (дата звернення: 14.09.2020).
9. Приймак В. Прийняття управлінських рішень: Навчальний посібник. К., Атїка, 2008, 240 с.

10. Ларичев О.И. Теория и методы принятия решений. М., Логос, 2000, 354 с.
11. Штойер Р. Многокритериальная оптимизация. Теория вычислений и приложения. М., Наука, 1992, 204 с.
12. Багатосторінкові та односторінкові додатки, їх переваги та недоліки. DOU URL: <https://dou.ua/forums/topic/25444/> (дата звернення: 16.09.2020).
13. Ярцев В.П. Організація баз даних та знань: навчальний посібник. К., ДУТ, 2018, 214 с.
14. Нільсен, Хоа Лоранжер. Технологія Angular 5. СПб., Символ-плюс, 2009, 512 с.
15. Крістофер Косентіно. SPA технологія. Київ., BHV, 2016, 465с.