

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук

**Бакалаврська кваліфікаційна
робота на тему:**

ПРОГРАМНИЙ МОДУЛЬ «ПОМІЧНИК РІЕЛТЕРА»

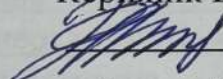
Виконав: студент 4 курсу, групи 1КН-216
спеціальності 122 – Комп'ютерні науки

 (шифр і назва напрямку підготовки, спеціальності)

Дунаєвський В.М.

(прізвище та ініціали)

Керівник БКР: д.т.н., проф. каф. КН



Іванчук Я. В.

(прізвище та ініціали)

« 4 » червня 2025 р.

Рецензент: д.т.н., проф., зав. каф. КСУ

Ковтун В. В.

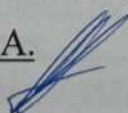
(прізвище та ініціали)

« 5 » червня 2025 р.

Допущено до захисту

Зав. кафедри проф., д.т.н. Яровий А.А.

« 6 » червня 2025 р.



Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації Кафедра
комп'ютерних наук

Рівень вищої освіти перший (бакалаврський)

Галузь знань – 12 Інформаційні технології

Спеціальність – 122 Комп'ютерні науки

Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ

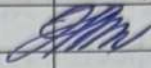
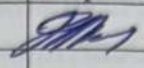
Завідувач кафедри КН
проф., д.т.н. Яровий А.А.

«5» травня 2025р.

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Дунаєвського Віктора Миколайовича

1. Тема роботи: Програмний модуль «Помічник ріелтора».
Керівник роботи: Іванчук Ярослав Володимирович, д.т.н, проф. кафедри КН
затверджені наказом вищого навчального закладу “20.03 2025 року №97”
2. Строк подання студентом роботи 30 травня 2025р.
3. Вихідні дані до роботи: персональні дані про співробітників; мова програмування SQL, Dart та Flutter; Максимальна кількість внесених клієнтів — 1500 осіб; максимальна кількість об'єктів нерухомості, які можна додати — 800 одиниць; кількість сторінок інтерфейсу користувача — не менше 5 (вхід, об'єкти, клієнти, угоди, нотатки).
4. Зміст розрахунково-пояснювальної записки включає: аналіз предметної області ріелторської діяльності, дослідження сучасних методів обліку клієнтів та об'єктів нерухомості, огляд існуючих програмних рішень для ведення баз даних та супроводу угод, розробку алгоритмів функціонування програмного модуля для підтримки роботи ріелтора; проектування структури програмного забезпечення для управління клієнтською базою, об'єктами та угодами; програмну реалізацію, тестування функціональних можливостей модуля та перевірку його стабільності при реальній роботі.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): типова схема запиту та відповіді від сервера в програмному модулі «Помічник ріелтора»; схема архітектури клієнт-серверного застосунку для управління об'єктами та клієнтами; загальний алгоритм взаємодії користувача з функціоналом додатку; структурна схема взаємозв'язків між модулями системи; UML-діаграма діяльності при створенні угоди в системі; ER-діаграма бази даних для зберігання інформації про клієнтів, об'єкти та угоди; принципова схема логіки обробки даних згідно архітектурного шаблону MVC.

6. Консультанти розділів проєкту (роботи)

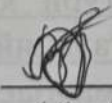
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Іванчук Я. В., д.т.н., проф.. каф. КН		

7. Дата видачі завдання 5 травня .2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів проєкту (роботи)	Примітка
1	Визначення об'єкта предмета мети та задач подальшого дослідження в контексті ріелторської діяльності	05.05.25- 07.05.25	
2	Аналіз предметної області огляд існуючих методів алгоритмів та інструментів для створення програмного модуля для роботи ріелтора	08.05.25- 11.05.25	
3	Проектування програмного модуля Помічник ріелтора база даних інтерфейс логіка взаємодії	12.05.25- 15.05.25	
4	Реалізація та тестування основного функціоналу модуля облік клієнтів додавання об'єктів створення угод	16.05.25- 01.06.25	
5	Оформлення пояснювальної записки підготовка графічних матеріалів та остаточне редагування роботи	02.06.25- 04.06.25	

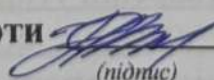
Студент


(підпис)

Дунаєвський В.М.

(прізвище та ініціали)

Керівник роботи


(підпис)

Іванчук Я. В.

(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 62 сторінок формату А4, включає 14 рисунків, таблиці, діаграми, а також список використаних джерел із 22 найменувань.

Робота присвячена розробці, аналізу та впровадженню програмного модуля під назвою «Помічник ріелтора». Основна мета — створення зручного та функціонального цифрового інструменту, що дозволить ріелторам ефективніше управляти клієнтською базою, здійснювати облік об'єктів нерухомості, вести супровід угод та забезпечувати якісний документообіг і нотатки в єдиній системі.

У роботі обґрунтовано потребу в цифровізації ринку нерухомості, де часто використовуються застарілі інструменти обліку (Excel, паперові картки), що не дають змоги оперативно аналізувати інформацію або працювати з великою кількістю даних. Запропонований модуль вирішує ці проблеми шляхом впровадження бази даних на основі SQL, створення зручного інтерфейсу користувача та логіки взаємодії з даними у реальному часі. У модулі реалізовано базові та додаткові функції: створення, редагування, фільтрація та сортування клієнтів і об'єктів; ведення історії угод; збереження нотаток; перегляд активних та завершених операцій.

Архітектура системи побудована з урахуванням принципів модульності, розділення обов'язків, зберігання даних у нормалізованих таблицях. Також у роботі використано UML-діаграми, ER-схеми, описано логіку реалізації CRUD-операцій, взаємозв'язки між сутностями та алгоритми, які забезпечують зручність і стабільність використання. Розроблений модуль має можливість масштабування та адаптації під різні типи агентств і моделей обліку.

Крім технічної частини, у роботі проведено аналіз існуючих аналогів на ринку — таких як CRM-системи для нерухомості (RealScout, Propertybase, Bitrix24), виявлено їхні недоліки, складність адаптації під конкретного ріелтора та високу вартість. Підсумком роботи є створення повністю функціонального цифрового продукту, який забезпечує автоматизацію основних процесів у

діяльності ріелтора. Він дозволяє уникати дублювання інформації, зменшити навантаження, покращити взаємодію з клієнтами та підвищити якість обслуговування.

Ключові слова: ріелтор, нерухомість, програмний модуль, облік клієнтів, автоматизація, угоди, база даних, CRM, цифровізація, супровід угод.

ABSTRACT

The bachelor's thesis consists of 62 A4-format pages and includes 14 figures, tables, diagrams, and a list of 22 references.

The work is devoted to the development, analysis, and implementation of a software module called "Realtor's Assistant." The main goal is to create a convenient and functional digital tool that allows real estate agents to more effectively manage their client database, keep records of real estate properties, support transaction processing, and ensure high-quality documentation and notes within a unified system. The thesis substantiates the need for digitalization in the real estate market, where outdated accounting tools (such as Excel spreadsheets and paper cards) are still frequently used, which do not allow for rapid data analysis or efficient work with large data volumes. The proposed module addresses these problems by implementing an SQL-based database, creating a user-friendly interface, and applying real-time data interaction logic. The module includes both basic and advanced functions: creation, editing, filtering, and sorting of clients and properties; maintaining transaction history; saving notes; viewing active and completed operations.

The system's architecture is built on the principles of modularity, separation of responsibilities, and data storage in normalized tables. The thesis also uses UML diagrams, ER models, and describes the logic of CRUD operations, entity relationships, and algorithms that ensure usability and system stability.

The developed module is scalable and adaptable to different types of agencies and accounting models. In addition to the technical part, the thesis includes an analysis of existing market analogs—such as real estate CRM systems (RealScout, Propertybase, Bitrix24)—and identifies their shortcomings, including the complexity of adapting them to individual realtors and their high cost. The result of the work is the creation of a fully functional digital product that automates the main processes of a realtor's activity. It helps to avoid data duplication, reduce workload, improve client interaction, and enhance the quality of service.

Keywords: realtor, real estate, software module, client management, automation, deals, database, CRM, digitalization, transaction support.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Аналіз класифікації тваринницьких ферм та особливості технологій утримання тварин	8
1.2 Аналіз використання ветеринарних методів та засобів для ранньої діагностики захворювань сільськогосподарських тварин	12
1.3 Сучасні програмні засоби для відстеження показників здоров'я тварин молочного скотарства.....	20
1.4 Висновок до розділу 1.....	23
2 РОЗРОБКА ТА ПРОЄКТУВАННЯ ПРОГРАМОГО МОДУЛЯ ДЛЯ МОНІТОРИНГУ ПОКАЗНИКІВ ЗДОРОВ'Я ТВАРИН.....	25
2.1 Розробка архітектури програмного модулю	25
2.2 Розробка взаємодії програмних модулів системи моніторингу процесів життєдіяльності тваринницького складу	33
2.3 Розробка алгоритму моніторингу процесів життєдіяльності тваринницького складу фермерського господарства.....	37
2.4 Висновок до розділу 2.....	42
3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЮ ДЛЯ МОНІТОРИНГУ ПРОЦЕСІВ ЖИТТЄДІЯЛЬНОСТІ ТВАРИН ФЕРМЕРСЬКОГО ГОСПОДАРСТВА.....	43
3.1 Обґрунтування вибору мов програмування та фреймворків	43
3.2 Програмна реалізація модулю для моніторингу процесів життєдіяльності тваринницького складу	46
3.3 Тестування програмного модулю моніторингу процесів життєдіяльності тварин на фермерському господарстві	48
3.4 Висновок до розділу 3.....	51
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54

ДОДАТОК А. Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	55
ДОДАТОК Б. Інструкція користувача.....	57
ДОДАТОК В. Лістинг програмного коду	60
ДОДАТОК Г. Графічна частина	67

ВСТУП

Актуальність досліджень. Ринок нерухомості в Україні активно розвивається та змінюється під впливом економічних, соціальних і технологічних факторів. У таких умовах все більше ріелторів стикаються з необхідністю ефективного управління великими обсягами інформації про клієнтів, об'єкти нерухомості, угоди та документообіг. Це зумовлює потребу у впровадженні сучасних цифрових рішень, які дозволяють автоматизувати рутинні процеси та покращити якість надання послуг.

Однією з ключових проблем, що постає перед ріелторами, є неструктурованість інформації, дублювання даних, складність контролю за статусами угод та труднощі в комунікації з клієнтами. Програмний модуль «Помічник ріелтора» дозволяє вирішити ці завдання шляхом впровадження централізованої системи обліку, в якій зберігається актуальна інформація про клієнтів, об'єкти, історію взаємодій, нотатки та угоди. Це значно зменшує навантаження на спеціалістів та дає змогу зосередитися на ключових етапах укладення угод.

Крім того, завдяки функціям фільтрації, пошуку, аналітики та формування звітності, програмний модуль підвищує ефективність роботи ріелтора, дозволяє швидко знаходити потрібні дані та приймати обґрунтовані рішення. Систематизація інформації також сприяє кращому обслуговуванню клієнтів, покращенню взаємодії з ними та зміцненню довіри[1].

Реалізація програмного модуля дає змогу агентствам нерухомості та приватним ріелторам покращити організацію своєї діяльності, оптимізувати процеси обробки інформації та підвищити загальну конкурентоспроможність на ринку послуг.

Метою дослідження є підвищення ефективності ведення ріелторської діяльності шляхом розробки програмного модуля «Помічник ріелтора» для автоматизації обліку клієнтів, об'єктів нерухомості та угод, що дозволить оперативно приймати управлінські рішення, зменшити кількість помилок і підвищити якість обслуговування клієнтів.

Об'єктом дослідження є процес розробки програмного модуля для автоматизації ріелторської діяльності.

Предметом дослідження є алгоритми та програмні засоби управління клієнтами, об'єктами нерухомості та угодами, їх функціональні можливості, логіка взаємодії модулів та обробка даних у рамках єдиної цифрової системи.

Задачі дослідження:

1. Провести аналіз предметної області ріелторської діяльності та існуючих цифрових інструментів для автоматизації обліку клієнтів, об'єктів і угод;
2. Розробити концепцію та структуру програмного модуля для ведення клієнтської бази, об'єктів нерухомості та супроводу угод;
3. Створити модель бази даних для зберігання та обробки інформації про клієнтів, об'єкти, угоди та нотатки;
4. Реалізувати алгоритми функціонування основних компонентів модуля, включно з фільтрацією, пошуком, обробкою угод та нотаток;
5. Здійснити програмну реалізацію модулю на основі обраної архітектури з інтеграцією користувацького інтерфейсу;
6. Підготувати інструкцію користувача щодо роботи з програмним модулем для ефективного використання його можливостей у ріелторській практиці.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз класифікації видів об'єктів нерухомості та особливості цифрових технологій у роботі рієлтора

Ринок нерухомості відіграє ключову роль у національній економіці, оскільки забезпечує людей житлом, формує інвестиційне середовище та сприяє розвитку суміжних галузей. Ефективна діяльність у сфері купівлі-продажу та оренди об'єктів нерухомості залежить від здатності рієлторів швидко обробляти запити, аналізувати інформацію, підтримувати комунікацію з клієнтами та документувати угоди. В умовах цифровізації все більшої актуальності набувають сучасні інформаційні технології, які дозволяють автоматизувати щоденні завдання агентів з нерухомості, підвищити швидкість роботи та точність прийняття рішень.

До основних категорій об'єктів нерухомості, з якими працюють рієлтори, належать: житлова (квартири, приватні будинки, котеджі), комерційна (офіси, торгові площі, склади), земельна (будівельні ділянки, сільськогосподарські наділи) та спеціалізована нерухомість (готелі, ресторани, виробничі об'єкти)[2]. Кожен із цих типів вимагає різного підходу до оцінки, реклами та проведення операцій.

Житлова нерухомість є найпоширенішим типом і найчастіше підлягає запитам на оренду або купівлю. Рієлтору важливо мати змогу швидко орієнтуватись у локаціях, цінах, типах планувань і документації. Комерційна нерухомість передбачає складніші процедури з оцінкою рентабельності, можливістю реконструкцій та юридичними нюансами, пов'язаними з комерційною діяльністю. Земельна нерухомість потребує врахування цільового призначення, доступу до інфраструктури, кадастрових обмежень. Спеціалізовані об'єкти, як правило, мають унікальні характеристики й потребують індивідуального підходу.

Успішна діяльність рієлтора базується на своєчасному доступі до актуальної інформації щодо об'єктів, клієнтів, документів, переглядів і угод. У зв'язку з цим широке застосування знаходять мобільні додатки, які дозволяють агентам працювати безпосередньо на виїзді, оперативно оновлювати дані та взаємодіяти з

CRM-системами, базами даних і месенджерами. Використання цифрових технологій у ріелторській діяльності охоплює автоматизовану фільтрацію об'єктів за заданими параметрами, зберігання інформації про клієнтів, нагадування про заплановані перегляди, генерацію договорів та звітів, ведення історії комунікацій. Такі інструменти суттєво знижують навантаження на працівника і мінімізують людський фактор.

Особливої уваги заслуговують інноваційні рішення, що базуються на хмарних сервісах і штучному інтелекті. Наприклад, системи автоматизованого підбору нерухомості за поведінковими патернами користувачів, аналітика ринку в реальному часі або інтеграція з платформами для електронного підпису договорів. Сучасні програми також передбачають інтерактивні карти, віртуальні тури та можливість комунікації з потенційними покупцями напряму через інтерфейс додатку.

Окрему роль відіграють мобільні додатки нового покоління — з елементами аналітики, візуалізації даних та можливістю інтеграції з хмарними сервісами. Система моніторингу діяльності ріелторів у межах таких додатків дозволяє відслідковувати ефективність кожного співробітника, аналізувати час, витрачений на угоди, кількість переглядів, збережених контактів та інші метрики. Це дозволяє менеджерам вчасно виявляти проблемні ділянки та вдосконалювати роботу агентства[3].

Таким чином, аналіз класифікації об'єктів нерухомості та особливостей застосування цифрових рішень у сфері ріелторських послуг свідчить про актуальність розробки програмного модуля "Помічник ріелтора". Такий модуль має забезпечити повний цикл роботи агента: від першого контакту з клієнтом до укладення угоди — з максимальним рівнем автоматизації, зручності та безпеки. Він дозволить оптимізувати роботу як індивідуального ріелтора, так і великих агентств, зробивши їхню діяльність більш системною, продуктивною та прозорою.

У процесі реалізації ріелторських послуг виникає потреба не лише в збереженні базових характеристик об'єкта, а й у створенні гнучкої системи взаємозв'язку між об'єктами, клієнтами, нотатками та завданнями. Це дозволяє забезпечити повний контекст взаємодії й формувати персоналізовані пропозиції. Наприклад, якщо клієнт шукає квартиру в конкретному районі в певному бюджеті, система повинна автоматично формувати підбір на основі цих параметрів та зберігати історію переглядів.

Крім того, у великих агентствах ключову роль відіграє розмежування доступу між працівниками. Це особливо важливо для контролю якості обслуговування, збереження конфіденційності та уникнення дублювання інформації. Саме тому в сучасних CRM-системах важливо реалізувати багаторівневу систему ролей — адміністратор, агент, аналітик, оператор підтримки тощо. Це дає змогу ефективно розподіляти функції між учасниками процесу та уникати конфліктів у даних.

Ще одним важливим елементом є інтеграція з геолокаційними сервісами, що дозволяє агенту або клієнту бачити об'єкт на карті, оцінити транспортну доступність, наявність шкіл, лікарень, магазинів тощо. Це підвищує цінність пропозиції, оскільки користувач відразу отримує повну картину. З іншого боку, система може автоматично визначати локацію клієнта або об'єкта, щоб формувати найактуальніші варіанти для перегляду.

Інструменти аналітики також є невід'ємною частиною сучасного ріелторського ПЗ. Звичайне фільтрування чи сортування вже не достатньо — важливо відслідковувати поведінкові патерни, наприклад:

- які об'єкти переглядають найчастіше;
- яка середня тривалість циклу угоди;
- з яких джерел приходять клієнти;
- який ріелтор має найвищу конверсію.

На основі таких даних система може автоматично пропонувати покращення у процесі роботи агентства.

Також важливою функцією є створення нотаток, нагадувань, завдань, прив'язаних до конкретного клієнта чи об'єкта. Наприклад, якщо призначена зустріч

з клієнтом — система має створити подію в календарі агента, а у випадку відсутності взаємодії протягом певного часу — автоматично створити завдання на дзвінок або лист.

Окрему цінність представляє можливість масового імпорту даних з Excel-таблиць, Google-форм або сторонніх платформ (наприклад, OLX, DOM.RIA), що дозволяє швидко наповнювати базу об'єктів і не витрачати час на ручне введення.

У майбутньому розвиток таких модулів може включати підтримку штучного інтелекту — чат-ботів для клієнтів, рекомендаційних систем для ріелторів, аналізу настроїв клієнтів на основі переписки. Це все формує новий рівень якості обслуговування у сфері нерухомості.

Не менш важливою є адаптивність інтерфейсу, адже ріелтори працюють у різних умовах — офіс, виїзд, смартфон, планшет або комп'ютер. Мобільна версія має бути максимально простою, швидкою і зручною, з акцентом на ключовому функціоналі — перегляд об'єктів, дзвінок клієнту, нотатки. Водночас веб-інтерфейс дозволяє глибше аналізувати дані, формувати звіти, керувати правами доступу та структурою бази.

Таким чином, повноцінний цифровий помічник ріелтора має поєднувати в собі:

- інструменти для обліку та управління об'єктами;
- модуль для ведення клієнтської бази;
- систему завдань і нагадувань;
- аналітику ефективності;
- інтуїтивний багатоплатформовий інтерфейс;
- безпечну та гнучку архітектуру з можливістю масштабування.

Реалізація такого програмного модуля не лише полегшує щоденну роботу ріелторів, а й сприяє загальному підвищенню якості обслуговування клієнтів, що, у свою чергу, впливає на репутацію та прибутковість агентства.

1.2 Аналіз існуючих CRM систем та інструментів автоматизації діяльності ріелторів

Управління ріелторською діяльністю значною мірою залежить від використання технологічних інструментів які дозволяють зменшити рутинну роботу покращити обробку даних про об'єкти клієнтів угоди та оптимізувати бізнес процеси.

Серед найпоширеніших цифрових рішень які використовуються в агентствах нерухомості варто виділити CRM системи такі як Bitrix24 AmoCRM Zoho CRM та інші які дозволяють зберігати дані про клієнтів керувати воронкою продажів створювати задачі та отримувати аналітику за угодами.

Однак більшість таких систем мають загальний характер не є повністю адаптованими під потреби ріелторського ринку зокрема в частині роботи з базою об'єктів або специфіки ринку оренди та продажу житлової нерухомості в Україні.

Деякі CRM сервіси пропонують окремі модулі для роботи з нерухомістю проте вимагають підписки або мають складний інтерфейс не пристосований до новачків або невеликих агентств які мають обмежений бюджет та персонал.

Крім CRM активно використовуються різні сервіси для обліку об'єктів наприклад Google Таблиці власні Excel бази файли на хмарах проте вони не забезпечують надійного контролю за історією взаємодії з клієнтом або збереженням статусу по кожному об'єкту. Іншою проблемою є відсутність єдиної системи яка дозволяє поєднати облік об'єктів взаємодію з клієнтами планування зустрічей нотатки по угодах та візуальні матеріали наприклад фото об'єкта техпаспорт або документи на право власності.

Автоматизований програмний модуль може вирішити ці задачі об'єднавши дані в єдину систему доступну через зручний інтерфейс з функціями пошуку фільтрації збереження історії взаємодій та планування дій по кожному клієнту. Наявність статистики наприклад кількість успішних угод у розрізі місяців середня ціна об'єктів у певному районі або рейтинг закриття угод для кожного ріелтора

дозволяє керівництву агентства ефективно аналізувати діяльність працівників і приймати управлінські рішення.

Таким чином аналіз існуючих систем показує що потреба у створенні адаптованого модулю для ріелторів є актуальною особливо для невеликих агентств і фахівців які хочуть мати просте але потужне рішення для повсякденної роботи.

Таблиця 1.1 – Порівняння популярних CRM-систем для ріелторів[4]

Назва CRM	Основні функції	Мінуси для ріелтора	Вартість / доступ
Bitrix24	Воронка продажів, календар, чат, база клієнтів	Складний інтерфейс, перевантаженість	Безкоштовно / \$
AmoCRM	Автоматизація комунікацій, задачі, пошта	Висока вартість, не вся база українською	\$\$\$
Zoho CRM	Гнучка кастомізація, API	Неадаптована під ринок нерухомості	\$\$
Excel / Google Таблиці	Простий облік об'єктів, клієнтів	Відсутність аналітики, зв'язків, автоматизації	Безкоштовно
Програмний модуль “Помічник ріелтора”	Вузька спеціалізація, інтуїтивність, фільтри, нотатки	у розробці / кастомна підтримка	Безкоштовно / свій

На сучасному етапі розвитку ринку нерухомості ріелторська діяльність значною мірою залежить від оперативної роботи з інформацією тому важливою складовою ефективної праці є цифрові інструменти які дозволяють швидко реагувати на запити клієнтів аналізувати потреби та контролювати кожен етап взаємодії.

В багатьох випадках ріелтори ведуть роботу в розрізнених середовищах наприклад зберігають фото об'єктів у смартфоні контактну інформацію в записнику угоди в окремому документі що створює хаос і ризик втрати ключових даних особливо при високому навантаженні. Програмний модуль може вирішити ці проблеми створюючи єдине середовище в якому кожен об'єкт має свою картку з усіма атрибутами фото нотатками статусом власником історією змін і запланованими діями наприклад дзвінок зустріч або показ квартири.

Окрему увагу в сучасних системах приділяють мобільності тобто можливості працювати з клієнтом безпосередньо з телефону переглядати базу редагувати дані додавати нотатки або фото після перегляду об'єкта не повертаючись до офісу це особливо важливо для агентів які постійно в русі. Програма також може містити елементи штучного інтелекту наприклад підказки щодо наступного кроку аналіз ефективності ріелтора за період нагадування про клієнтів з якими давно не було активності або пропозиції по нових об'єктах які можуть зацікавити клієнта на основі історії.

Інтеграція з Google Maps або OpenStreetMap дозволяє автоматично визначати розташування об'єкта будувати маршрут до показу або шукати інші об'єкти поблизу за заданим радіусом що економить час і підвищує зручність для клієнта. Модуль також може підтримувати аналітику наприклад зведення по кількості показів клієнтів конверсій середній час укладання угоди найпопулярніші типи об'єктів або локації що дає можливість керівництву оцінювати ефективність роботи і приймати обґрунтовані рішення.

Для керівника агентства важливо мати інструменти контролю наприклад моніторинг активності співробітників графік зустрічей аналіз результатів виконання плану кількість клієнтів на одного агента або кількість успішних угод

на місяць. Іншою важливою функцією програмного модуля є ведення історії змін кожної сутності зокрема клієнтів та об'єктів нерухомості це дозволяє переглядати хто і коли оновлював дані що саме було змінено які об'єкти вже неактуальні або продані що зменшує ризик помилок і підвищує прозорість.

Також цінним є впровадження шаблонів типових дій наприклад після додавання нового клієнта система може автоматично створити задачі зателефонувати через один день підготувати добірку об'єктів відправити презентацію що економить час і робить роботу більш системною.

Для новачків у сфері ріелторства програмний модуль може мати вбудовану систему підказок інструкцій або мікронавчання прямо в інтерфейсі наприклад що означає той чи інший статус клієнта що робити після показу об'єкта або як швидко створити угоду.

Дуже корисною функцією може бути створення кастомних тегів які дозволяють ріелтору самостійно сортувати об'єкти за будь якими критеріями наприклад центр з ремонтом без торгу або до кінця місяця це створює додаткову гнучкість у роботі з великою кількістю об'єктів. У разі командної роботи програмний модуль має дозволяти спільний доступ до бази з можливістю налаштування прав наприклад один ріелтор може бачити всі об'єкти інший лише свої також керівник може мати повний контроль і бачити аналітику по всій агенції.

Можна також реалізувати систему оцінок і коментарів до кожного об'єкта наприклад якщо ріелтор вже був на перегляді він може залишити коротку нотатку типу поганий під'їзд є запах від сусідів квартира гарна але дорого це дозволяє всій команді мати повну картину без повторного відвідування. Однією з ключових функцій програмного модуля для ріелтора є інтеграція з календарем що дозволяє автоматично створювати зустрічі з клієнтами записувати перегляди об'єктів отримувати нагадування про події або заплановані дії це допомагає уникати пропущених дзвінків та краще організовувати свій день.

Крім того система може надсилати push повідомлення або sms якщо встановлено інтеграцію з мобільним додатком або sms шлюзом що дозволяє

клієнтам отримувати нагадування про перегляд або підтвердження запису зручно як для клієнта так і для агента[5].

Ще однією важливою опцією є можливість зберігати шаблони договорів наприклад договір оренди договір купівлі переддоговірна угода нотаріальні форми які можна швидко заповнити автоматично підставивши дані з картки клієнта або об'єкта система також може зберігати архів підписаних документів що дозволяє легко знайти потрібний договір у будь який момент.

Додатково можна реалізувати функцію створення звітів наприклад скільки угод було укладено за місяць скільки клієнтів оброблено який середній цикл продажу або яка конверсія із запиту у реальну угоду для керівника агентства це джерело управлінських рішень а для рієлтора — оцінка особистої ефективності.

Система може також дозволити створення чеклістів для кожного об'єкта наприклад перевірити документи зробити фото внести ціну домовитись з власником виставити оголошення зберегти координати і так далі це дозволяє тримати порядок і не забувати про рутинні дії.

В умовах коли агент працює з десятками об'єктів і клієнтів важливо мати інструмент фільтрації наприклад показати всі об'єкти до 60000 злотих у районі Груналд які ще не мали переглядів або знайти клієнтів з якими не було контактів понад тиждень щоб нагадати про себе. Програмний модуль також може включати функцію порівняння об'єктів наприклад вивести на екран дві або три квартири і порівняти їх за ціною площею локацією кількістю кімнат станом ремонту і наявністю документів така функція корисна при презентації варіантів клієнтам.

Ще одна цікава функція — чорний список об'єктів або клієнтів наприклад якщо квартира має проблеми з документами або клієнт неявився на перегляд тричі її можна позначити тегом проблемний і система буде попереджати про це при наступному зверненні. Розширена версія програмного модуля може підтримувати облік комісій рієлторів наприклад фіксувати яка сума зароблена по кожній угоді скільки залишилось заплатити хто отримав бонус або який план продажів виконано це дозволяє керівнику бачити загальний фінансовий результат і мотивувати працівників. Також можлива реалізація функції експорту даних наприклад у Excel

для ведення бухгалтерії або у PDF для звітності перед керівництвом або податковою.

У випадку роботи агенції з кількома ріелторами дуже зручно якщо модуль підтримує розподіл прав доступу наприклад агент бачить тільки своїх клієнтів а керівник — всіх є можливість призначити відповідального за об'єкт або угоду слідкувати за виконанням задач у команді.

Ще одним ключовим компонентом ефективної CRM системи для ріелтора є можливість відстеження джерел трафіку клієнтів наприклад звідки прийшов запит з сайту Instagram OLX особистої рекомендації або рекламного оголошення це дозволяє аналізувати які канали працюють краще і на що варто витратити бюджет. Система також може бути використана для побудови воронки продажів де на кожному етапі — перший контакт перегляд об'єкта обговорення умов оформлення угоди — клієнт змінює статус це дозволяє бачити хто на якому етапі завис що потребує уваги і де найчастіше втрачається контакт.

Інтеграція з популярними месенджерами наприклад Telegram або Viber дозволяє вести листування напряму з системи прикріплювати файли домовлятися про перегляди і одразу фіксувати результат у картці клієнта це скорочує час і усуває ручне дублювання інформації[6].

У перспективі програмний модуль може бути розширений для роботи з агентствами які мають філії або офіси в кількох містах кожен офіс матиме свою базу доступну тільки співробітникам конкретного регіону але керівник матиме змогу бачити повну аналітику в одному місці

Застосування такого модуля в агентствах дозволяє стандартизувати підхід до обслуговування клієнтів підвищити якість комунікації та скоротити втрати потенційних угод через людський фактор що є критично важливим у сфері з високою конкуренцією.

Таблиця 1.2 – Типові проблеми рієлтора і як їх вирішує модуль

Проблема рієлтора	Рішення через модуль
Забув зателефонувати клієнту	Автоматичні нагадування
Втрата нотаток або контактів	Централізована база з історією взаємодій
Плутанина в переглядах	Календар із візуальним графіком зустрічей
Складно знайти потрібний об'єкт	Фільтри по ціні району типу стану об'єкта
Ручне оформлення договору	Автоматична генерація з шаблону
Не знає джерело заявки	Відстеження каналу залучення клієнта
Дублювання об'єктів між агентами	Єдина база з позначенням відповідального

Таким чином аналіз існуючих систем управління рієлторською діяльністю показує що універсальні CRM не повністю задовольняють потреби ринку а створення спеціалізованого програмного модуля відкриває нові можливості для автоматизації підвищення ефективності та якості обслуговування клієнтів у сфері нерухомості.

Запропонований функціонал дозволяє зробити роботу рієлтора більш впорядкованою прозорою і контрольованою що в умовах зростаючої конкуренції є ключовим фактором успіху.

Крім функціоналу, описаного вище, модуль «Помічник рієлтора» може підтримувати систему швидкого пошуку дублікатів — коли при додаванні нового об'єкта система автоматично перевіряє, чи не існує подібного за адресою, метражем або власником. Це знижує ризик появи дублюючої інформації, яка створює плутанину в базі.

Додатково в модулі доцільно реалізувати інтелектуальну панель активності, яка дозволяє ріелтору бачити, з ким давно не було контактів, які об'єкти довго не оновлювались, та які клієнти найчастіше закривають угоди. Таке підсвічування «гарячих» або «втрачених» контактів дозволяє оптимізувати робочий день без зайвого аналізу вручну.

У контексті командної роботи також корисною буде функція переведення справи між ріелторами — наприклад, у разі звільнення співробітника або зміни відповідального, усі прив'язані об'єкти, клієнти й угоди можуть передаватись іншому користувачу без втрати історії.

Ще одна важлива особливість, яку можна реалізувати, — відгуки від клієнтів, які заповнюються після укладення угоди. Це не лише покращує якість обслуговування, а й створює рейтинг ріелторів усередині агенції, який можна враховувати при формуванні бонусів або перегляді умов співпраці.

Система також може підтримувати інтеграцію з маркетинговими платформами — наприклад, автоматичне формування оголошень для OLX, DOM.RIA або Facebook Marketplace з уже внесеними фото, описом і ціною з бази. Це економить час і виключає ручний дублюючий ввід.

Для безпеки даних можна реалізувати резервне копіювання в хмару, щоби всі дії автоматично зберігались на сторонньому сервері, й у разі втрати телефону, збою чи зміни пристрою — ріелтор міг без проблем відновити повну базу.

На завершення, модуль повинен бути гнучким у налаштуваннях, аби підходити як для невеликих агенцій з 2–3 працівниками, так і для великих фірм із десятками агентів. Це досягається через систему прав доступу, кастомізованих фільтрів, незалежних шаблонів договорів та інтерфейсних панелей.

Таким чином, програмний модуль «Помічник ріелтора» має не лише замінити існуючі універсальні CRM-системи, а й перевершити їх завдяки вузькій спеціалізації, простоті використання, інтеграції ключових функцій у єдиному середовищі та можливості адаптації до конкретних умов кожної агенції.

1.3 Сучасні програмні засоби для підтримки діяльності ріелтора

Ринок нерухомості потребує інструментів які дозволяють ріелторам ефективно працювати з великою кількістю даних комунікувати з клієнтами управляти об'єктами фіксувати угоди та здійснювати контроль за процесами у зручному цифровому середовищі.

Сучасні програмні засоби для підтримки ріелторської діяльності мають широкий спектр функцій які включають ведення клієнтської бази облік об'єктів нерухомості управління завданнями створення та супровід угод фільтрацію по заданих критеріях побудову воронки продажів та генерацію звітів.

Більшість готових CRM систем орієнтовані на бізнес загального типу тому не враховують специфіку ринку нерухомості де важливими є саме модулі для візуального представлення об'єктів обробки фото документів фіксації переглядів .

Схема на рисунку 1.1 ілюструє базову логіку роботи програмного модуля ріелтора користувач взаємодіє з інтерфейсом через основні розділи клієнти об'єкти угоди календар та аналітика усі ці дані обробляються в центральному модулі після чого формуються вихідні результати у вигляді звітності нагадувань та контрольних індикаторів[7].

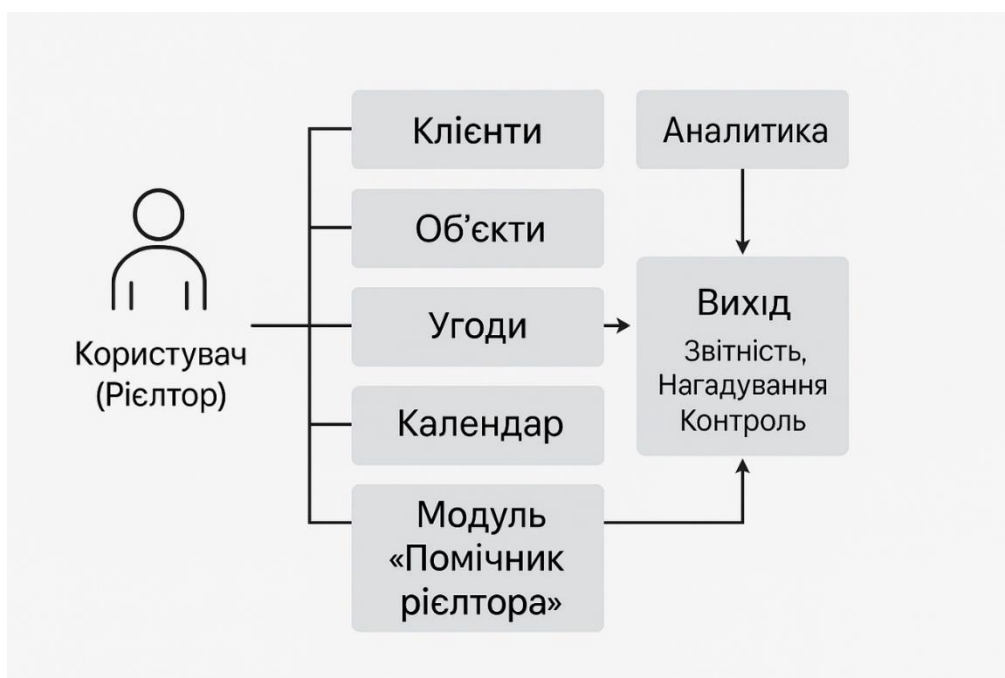


Рисунок 1.1 – Схема роботи програмного модуля «Помічник ріелтора»

Блок клієнти дозволяє зберігати інформацію про контактні дані історію взаємодії потреби клієнта та статуси співпраці наприклад новий перегляд угода втрачений це забезпечує персоналізовану роботу і дозволяє уникнути повторів.

Блок об'єкти містить всю інформацію про нерухомість включаючи фото характеристики адресу тип стан ціни і юридичний статус також до кожного об'єкта можуть бути прикріплені документи нотатки історія показів і тегування. Блок угоди об'єднує клієнтів та об'єкти фіксує хронологію перемовин обговорень зміни умов та дату підписання договору це дозволяє бачити всі активні угоди та стадію на якій вони перебувають.

На рисунку 1.2 зображено структуру картки об'єкта нерухомості, яка містить фотографії, технічні характеристики, опис, а також функції редагування та призначення задач. Такий підхід дозволяє швидко орієнтуватися в базі об'єктів та ефективно контролювати статуси угод.

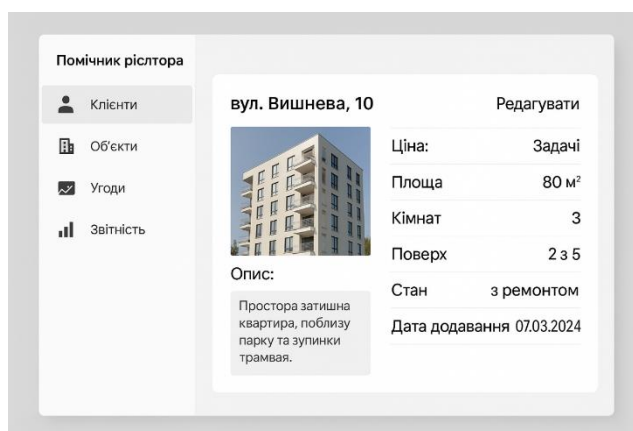


Рисунок 1.2 – Приклад інтерфейсу модуля “Помічник рієлтора” для перегляду об'єкта нерухомості

Блок календар дозволяє планувати зустрічі перегляди дзвінки виставляти дедлайни й отримувати нагадування як у самому модулі так і на телефоні завдяки інтеграції з Google Calendar або іншим сервісом.

Блок аналітика відповідає за узагальнення даних наприклад кількість закритих угод середній час співпраці топ клієнти з прибутковості популярні райони або динаміка цін за квартал.

Завдяки такій структурі ріелтор має змогу швидко орієнтуватися у своїй базі даних бачити прогрес по клієнтах уникати дублювань та контролювати процеси в режимі реального часу.

Додатковим функціоналом який може бути реалізований у модулі є система пошуку за ключовими словами наприклад ріелтор може ввести слово балкон або новобудова і система видає всі об'єкти де ці слова згадуються в описі або тегах що значно полегшує підбір варіантів під конкретний запит[8].Також актуальним є впровадження внутрішньої системи рейтингування наприклад можна додавати оцінки до об'єктів за зручністю розташування станом ремонту співвідношенням ціна якість або враженням клієнтів це дозволяє швидше орієнтуватися в сильних і слабких сторонах кожного варіанта. Ще однією корисною функцією є можливість створення шаблонів презентацій наприклад за допомогою натискання однієї кнопки формується готовий PDF з описом об'єкта фото ключовими характеристиками ціною і контактами агента який можна надіслати клієнту в месенджер або пошту.

У разі роботи з забудовниками або великими партнерами система може зберігати контракти умови співпраці знижки комісії доступні об'єкти а також автоматично повідомляти про оновлення наприклад нові квартири в продажу зміну цін або завершення будівництва.

Також модуль може підтримувати систему статусів і контролю для керівника наприклад кожен об'єкт має статус на модерації активний неактуальний а кожна дія ріелтора фіксується у вигляді логів що дозволяє оцінити активність кожного працівника в агентстві.

Корисною є функція дублювання об'єкта наприклад коли агент хоче внести схожий об'єкт з тим самим плануванням або в тій же новобудові але з іншою ціною або поверхом це економить час і дозволяє швидко наповнювати базу.

Для агентств які працюють з орендою важливою є функція автоматичного нагадування про завершення строку оренди що дозволяє завчасно зв'язатися з

орендарем або власником і продовжити угоду або знайти нового клієнта. Розширений модуль може також підтримувати статистику по кожному клієнту наприклад скільки було контактів переглядів що подобалося або не подобалося які об'єкти відхиляв на які погоджувався що дозволяє краще зрозуміти його смаки та сформувати персоналізовані добірки[9].

Інтерфейс повинен бути інтуїтивним і не вимагати довгого навчання особливо для нових ріелторів або стажерів тому важливо щоб усі функції були логічно згруповані наприклад у вигляді вкладок карток панелей швидкого доступу або швидких дій..

1.4 Висновок до розділу 1

У результаті проведеного аналізу предметної області можна зробити висновок, що сучасна ріелторська діяльність вимагає високого рівня автоматизації для забезпечення точного, ефективного та безпечного ведення інформації. Основними проблемами, які спостерігаються у діяльності ріелторів, є неструктурованість даних, дублювання інформації, ручне введення записів, складність у пошуку необхідних об'єктів за критеріями, а також відсутність системи нагадувань і контролю за етапами угод.

Проведений огляд сучасних CRM-систем, зокрема таких як Bitrix24, AmoCRM, ZoHo CRM, а також використання Google Таблиць, показав, що більшість із них не враховують специфіку ринку нерухомості України, мають обмежені можливості адаптації під індивідуальні потреби ріелтора та часто вимагають надмірної кількості дій для виконання простих операцій. Крім того, вартість використання таких систем є високою для невеликих агентств чи приватних ріелторів.

Аналіз показав доцільність створення окремого програмного модуля, що буде зосереджений саме на спрощенні та прискоренні виконання базових ріелторських функцій: облік клієнтів, угод, об'єктів нерухомості, створення нотаток, управління статусами, формування звітності та нагадування. Такий підхід дозволяє уникнути

перевантаження системи зайвими функціями і зосередити розробку на реальних потребах користувача.

Окрему увагу в дослідженні було приділено інтеграції з додатковими інструментами — картографічними сервісами, месенджерами, хмарним сховищем, що суттєво підвищує зручність роботи в польових умовах або з мобільних пристроїв. Така інтеграція дозволяє агенту швидко переглядати розташування об'єктів на карті, обмінюватись файлами та повідомленнями без потреби в сторонніх додатках.

Наявність доступу до статистики, історії угод, попередніх взаємодій з клієнтами та нагадувань про події є критично важливими для забезпечення повного контролю за процесами взаємодії та покращення якості обслуговування. Це сприяє оперативному прийняттю рішень, зменшенню ризику помилок і підвищенню загальної ефективності діяльності агентства.

Крім того, інтеграція з популярними цифровими інструментами забезпечує збереження єдиної інформаційної бази, доступної з будь-якого пристрою та в будь-який час.

Це особливо важливо в умовах динамічної роботи ріелтора, коли потрібно швидко реагувати на запити клієнтів, змінювати статус об'єктів або переглядати історію взаємодій навіть перебуваючи поза офісом. Підвищена мобільність та централізований доступ до даних суттєво зменшують ризики втрати інформації.

Це особливо важливо в умовах динамічної роботи ріелтора, коли потрібно швидко реагувати на запити клієнтів, змінювати статус об'єктів або переглядати історію взаємодій навіть перебуваючи поза офісом. Підвищена мобільність та централізований доступ до даних суттєво зменшують ризики втрати інформації.

Крім того, можливість працювати в реальному часі — наприклад, оновлювати інформацію про об'єкт одразу після перегляду або вносити нові контакти під час зустрічі — дозволяє зберігати актуальність бази та уникати дублювань. Завдяки цьому агент може зосередитись на спілкуванні з клієнтами, а не витрачати час на технічні процеси.

2 РОЗРОБКА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ “ПОМІЧНИК РІЄЛТОРА”

2.1 Розробка архітектури програмного модулю

Розробка архітектури програмного модуля є ключовим етапом створення ефективного програмного забезпечення для підтримки ріелторської діяльності. Архітектура визначає логіку взаємодії між компонентами системи, способи зберігання, обробки та передачі даних, а також забезпечує зручний доступ до функцій системи з боку кінцевого користувача — ріелтора.

Запропонована архітектура програмного модуля «Помічник ріелтора» побудована за клієнт-серверною моделлю, яка передбачає розподіл функцій між клієнтським інтерфейсом (графічним застосунком) і серверною частиною, що відповідає за зберігання та обробку даних. Такий підхід дозволяє підвищити масштабованість системи, оптимізувати ресурси пристрою користувача, а також забезпечити централізований облік і безпечне зберігання даних.

На стороні клієнта реалізується зручний та інтуїтивно зрозумілий графічний інтерфейс, через який ріелтор має можливість:

переглядати та редагувати дані клієнтів; створювати та керувати об'єктами нерухомості; укладати угоди; залишати нотатки; застосовувати фільтрацію та сортування даних; переглядати історію взаємодії.

Зі свого боку, серверна частина обробляє всі запити, отримані від клієнта, виконує логічні операції, перевірку введених даних, взаємодіє з базою даних і забезпечує синхронізацію між різними сесіями користувачів.

На рисунку 2.1 зображено логіку роботи програмного модуля ріелтора. Користувач взаємодіє з інтерфейсом системи, використовуючи основні розділи: «Клієнти», «Об'єкти», «Угоди», «Календар» та «Аналітика». Усі введені дані надходять до центрального модуля, де відбувається їх обробка. Результатом є формування звітності, нагадувань і контрольних індикаторів, що дозволяє

оперативно управляти процесами взаємодії та приймати ефективні управлінські рішення.



Рисунок 2.1 – Схема архітектури клієнт-серверної моделі

Опис функціональних модулів програмної архітектури

1 .Модуль керування клієнтами

1 Модуль відповідає за збереження, перегляд і редагування інформації про клієнтів. Він дає змогу:

додавати нових клієнтів (ім'я, контакти, тип — продавець/покупець);

фіксувати джерело звернення клієнта;

переглядати історію взаємодій (угоди, нотатки);

здійснювати пошук та фільтрацію за ім'ям, датою звернення тощо.

2. Модуль обліку об'єктів нерухомості

Функціонал модуля охоплює:

створення картки об'єкта з усіма характеристиками (тип, район, площа, ціна, фото);

прикріплення об'єкта до клієнта або угоди;

фільтрацію об'єктів за заданими параметрами (ціна, площа, тип житла);

виведення списків активних/архівних об'єктів.

3. Модуль керування угодами

Цей блок дозволяє:

створювати нову угоду, вибравши клієнта й об'єкт;
встановлювати статус угоди (активна, очікує, завершена, скасована);
додавати умови співпраці, важливі дати, нотатки по переговорах;
переглядати історію угод у хронології та з можливістю фільтрації.

4. Модуль нотаток

Призначений для збереження оперативної інформації, яка не входить до стандартних структур даних. Нотатки можуть бути:

прив'язані до клієнта, об'єкта або угоди;
використані як нагадування або як зафіксована інформація про дзвінок/зустріч;
доступні лише для перегляду конкретним користувачем (рівень доступу).

5. Модуль аналітики

Цей блок надає рієлтору доступ до звітів і візуалізацій:

кількість активних угод;
кількість нових клієнтів за період;
рейтинг найчастіше запитуваних об'єктів;
аналіз ефективності за часом або типом угод;

6. Модуль безпеки та авторизації

Впроваджено:

автентифікацію користувача;
розмежування прав доступу (адмін, рієлтор);
контроль змін у базі;
збереження логів активності.

7. Модуль резервного копіювання

Програмний модуль передбачає періодичне створення копій бази даних із можливістю відновлення. Зберігаються останні декілька версій на випадок збою.

Архітектура програмного модуля побудована таким чином, щоб забезпечити логічну та структуровану взаємодію між усіма компонентами системи. Кожен

модуль працює автономно, але водночас передає необхідні дані іншим модулям через спільний логічний шар. Це забезпечує цілісність даних, мінімізує дублювання та покращує швидкодію системи.

Взаємодія відбувається за такою схемою:

користувач через клієнтський інтерфейс створює запит;
запит надходить у логічний рівень, де перевіряється його коректність;
далі дані передаються на рівень бази даних або до модуля, який їх обробляє (аналітика, угоди тощо);

сформована відповідь передається назад через логічний рівень до інтерфейсу.

Усі взаємодії побудовані на основі CRUD-операцій — створення, читання, оновлення, видалення. В системі реалізовано асинхронну обробку даних, що дозволяє обробляти кілька запитів одночасно без втрати продуктивності.

Типовий алгоритм взаємодії користувача з системою

Алгоритм взаємодії ріелтора з програмним модулем передбачає послідовне виконання операцій у межах єдиного сценарію. Приклад типової послідовності дій користувача:

1. Вхід у систему та авторизація;
2. Перегляд списку клієнтів або створення нового;
3. Перевірка наявних об'єктів або додавання нового об'єкта;
4. Формування угоди між клієнтом і об'єктом;
5. Додавання нотаток, встановлення дати зустрічі або перегляду;
6. Збереження змін та синхронізація з базою;
7. Автоматичне оновлення аналітики;
8. Можливість перегляду історії угод, фільтрації або формування звітності.



Рисунок 2.2 – Схематичне зображення логіки роботи модуля

Така архітектура дозволяє масштабувати рішення додавати нові функції оновлювати інтерфейс без втрати основних даних а також забезпечує розділення відповідальності між фронтендом і бекендом що полегшує розробку тестування і супровід системи

Подана система включає в себе засоби для збору, збереження, обробки та представлення даних про клієнтів, об'єкти нерухомості, угоди, заплановані дії та аналітичні показники діяльності ріелтора. Завдяки цьому агентства можуть

ефективно організовувати процеси обслуговування клієнтів і приймати обґрунтовані управлінські рішення.

На рисунку 2.2 представлено базову логіку архітектури програмного модуля, яка демонструє основні компоненти та взаємозв'язки між ними.

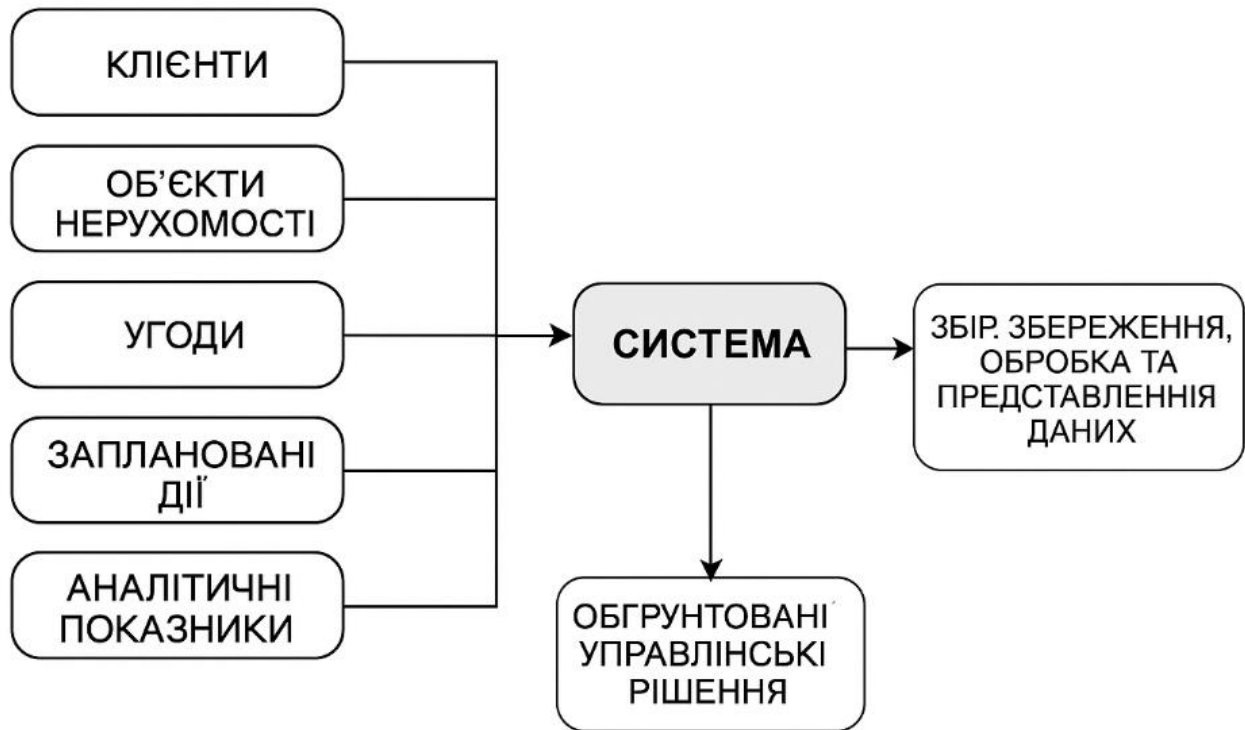


Рисунок 2.2 – Базова логіка архітектури програмного модуля[10].

Запити до сервера отримує відповідь у вигляді структурованої інформації або результатів запитів до бази даних наприклад добірки об'єктів фільтрацію клієнтів або історію угод.

Система працює таким чином що всі дані які вводить ріелтор одразу обробляються і зберігаються на сервері завдяки чому зберігається історія кожного клієнта об'єкта та угоди а також забезпечується контроль доступу до інформації залежно від рівня прав користувача. У випадку якщо агент працює в польових умовах тобто не в офісі а на перегляді або зустрічі система повинна підтримувати

адаптивний інтерфейс для мобільних пристроїв з можливістю швидкого доступу до бази редагування інформації та прикріплення нових файлів до об'єкта або угоди.

Ключовим елементом побудови архітектури програмного модуля є визначення логіки взаємодії між компонентами системи клієнт інтерфейс сервер база даних аналітичний блок і модуль задач мають взаємодіяти між собою без затримок з урахуванням захисту даних та збереження історії дій.

Інтерфейс користувача повинен бути максимально простим та інтуїтивним щоб новий ріелтор міг почати роботу з системою вже з першого дня без навчання усі блоки повинні бути згруповані за функціональністю наприклад ліве меню для доступу до розділів клієнти об'єкти угоди календар задачі аналітика. Кожен функціональний блок працює з окремою частиною бази даних але всі модулі пов'язані між собою для прикладу після додавання нового клієнта система автоматично пропонує створити нову задачу запланувати перегляд або призначити відповідального ріелтора.

База даних повинна зберігати усі ключові атрибути об'єктів тип нерухомості стан площа ціна адреса координати а також прикріплені зображення документи нотатки та історію змін база клієнтів включає ім'я контактну інформацію джерело заявки історію активності переваги та причини відмов.

Угоди мають бути окремим блоком який об'єднує клієнта та об'єкт у єдину зв'язку до кожної угоди можуть додаватися коментарі зміни статусів план дій та прикріплення підписаних документів також можна зберігати шаблони договорів щоб швидко формувати їх у процесі.

Серверна частина відповідає за виконання основної логіки пошук по фільтрах обробка запитів сортування редагування внесення змін формування добірок статистика збереження дій тощо на цій частині реалізуються API для взаємодії з інтерфейсом і зовнішніми сервісами наприклад Google Calendar. Щоб забезпечити масштабованість архітектура має підтримувати додавання нових модулів наприклад фінансовий модуль контроль комісій автоматичні звіти для бухгалтерії або модуль масового імпорту з Excel чи OLX у базу системи також можна додати блок push сповіщень або інтеграцію з Telegram[11].

Інтеграція з мапами дає змогу автоматично визначати координати об'єкта і побудувати маршрут до місця перегляду це корисно для роботи в польових умовах коли ріелтор проводить багато зустрічей за день і має швидко пересуватися між локаціями.

Важливим аспектом архітектури є наявність аналітичного блоку який дозволяє збирати дані про ефективність роботи агента кількість дзвінків угод переглядів середній час укладання угоди популярні запити клієнтів успішність по районах це дає змогу агентству приймати обґрунтовані рішення.

Для командної роботи архітектура повинна враховувати розподіл прав доступу кожен ріелтор має бачити тільки своїх клієнтів керівник має мати доступ до всієї бази статистики та звітів а адміністратор — до технічних налаштувань прав користувачів і логів дій.

Система повинна підтримувати логування тобто зберігати хто коли яку дію зробив наприклад змінив статус об'єкта додав новий контакт видалив документ це дозволяє відновити хронологію і підвищує відповідальність користувачів.

У рамках архітектури також важливо передбачити можливість масштабування даних наприклад якщо агентство зростає і починає працювати з великою кількістю об'єктів або відкриває нові філії модуль має бути готовий до розширення бази без втрати продуктивності або швидкості роботи.

Для кращої роботи з даними може бути реалізована система фільтрації та тегування наприклад всі клієнти які цікавляться орендою можна позначити відповідним тегом також можна використовувати теги типу терміново з ремонтом до 100000 злотих або поруч із парком що полегшує навігацію. У модулі доцільно реалізувати функцію архівування неактуальних об'єктів або завершених угод щоб основна база залишалась чистою а при потребі інформацію можна було відновити це спрощує роботу і пришвидшує пошук[12].

З погляду безпеки система повинна мати механізми резервного копіювання даних щоб у разі технічного збою інформацію можна було швидко відновити також бажано використовувати шифрування персональних даних клієнтів та угод для дотримання стандартів захисту інформації.

Також бажано реалізувати журнал активності користувача де зберігається інформація про кожен вхід у систему зміну записів або виконання ключових дій це дозволяє в разі потреби перевірити хто виконував ті чи інші зміни та уникнути непорозумінь у командній роботі. Щоб забезпечити зручність доступу до системи для ріелторів які працюють з мобільних пристроїв архітектура повинна підтримувати адаптивну верстку або мати окрему мобільну версію інтерфейсу що дозволить швидко переглядати об'єкти редагувати дані або перевіряти календар без прив'язки до офісу.

У результаті правильно спроектована архітектура забезпечує не лише технічну стабільність а й високу зручність використання модулю ріелторами у реальних умовах а також можливість його подальшого вдосконалення залежно від потреб конкретного агентства або індивідуального користувача.

2.2 Розробка взаємодії між компонентами системи клієнт база об'єктів панель угод

Розробка програмного модуля для підтримки ріелторської діяльності є комплексним завданням яке передбачає взаємодію декількох функціональних блоків що працюють у єдиному середовищі обмінюючись інформацією в реальному часі.

Основні модулі системи включають модуль клієнтів модуль об'єктів модуль угод модуль задач календар модуль аналітики і модуль налаштувань кожен з них виконує окремі функції але постійно взаємодіє з іншими через внутрішню логіку системи.

Наприклад після додавання нового клієнта система автоматично пропонує створити задачу додати об'єкт або запланувати перегляд також при створенні угоди вона зв'язується з клієнтом і об'єктом формуючи повну картину взаємодії.

Календар синхронізується з задачами і угодами дозволяючи планувати події перегляди зустрічі та дзвінки з клієнтами при зміні дати або статусу задача оновлюється в реальному часі що дозволяє тримати всі дії під контролем.

Модуль аналітики отримує дані з усіх інших модулів і формує звіти наприклад про кількість угод по кожному рієлтору середній термін обробки клієнта частоту показів по об'єктах це дозволяє оцінити ефективність роботи агентства.

Система задач дозволяє зв'язувати дії з клієнтами об'єктами або угодами наприклад задача зателефонувати клієнту призначається відповідальному рієлтору після чого з'являється в його календарі а після виконання змінює статус.

Кожен модуль має свою логіку доступу для прикладу рієлтор може бачити лише свої угоди та клієнтів а керівник — повну картину по всьому агентству це забезпечується на рівні авторизації та ролей які задаються адміністратором .

Уся інформація між модулями передається через внутрішній API, що забезпечує гнучку архітектуру та можливість масштабування системи. Такий підхід дозволяє в майбутньому легко додавати нові функціональні блоки — наприклад, модулі фінансів, розсилок або аналітики реклами.

На рисунку 2.3 зображено схему взаємодії модулів системи “Помічник рієлтора”, які взаємодіють між собою за допомогою єдиного внутрішнього API.



Рисунок 2.3 – Схема взаємодії модулів системи “Помічник рієлтора”

Згідно з логікою взаємодії модулів, що представлена на рисунку 2.3, кожен функціональний блок системи виконує власні задачі, але водночас постійно обмінюється даними з іншими модулями. Це забезпечує цілісність системи, синхронність дій та підтримку єдиної бази знань, яка оновлюється в режимі реального часу.

Наприклад, модуль задач може автоматично створювати нові записи після додавання клієнта або змінення статусу угоди. Такі задачі автоматично потрапляють до календаря рієлтора із зазначенням термінів виконання, що дозволяє планувати роботу ефективно та уникати пропуску важливих дій.

У випадку створення нового об'єкта нерухомості, система автоматично додає його до переліку доступних для підбору, а відповідна інформація передається до інших модулів. Зокрема, аналітичний блок миттєво отримує оновлені дані про кількість активних, неактивних або об'єктів, що очікують на документацію.

Тісна взаємодія модулів «Клієнти» та «Угоди» дозволяє формувати повну історію співпраці з кожним клієнтом: дзвінки, перегляди, відмови, змінення статусів, нотатки та причини відхилення угод. Це допомагає рієлтору краще розуміти потреби клієнта та пропонувати максимально релевантні варіанти.

Модуль аналітики працює у фоновому режимі, формуючи звіти про активність користувачів, цикл продажу, конверсію з перегляду в угоду, ефективність об'єктів і результативність окремих агентів або команд.

Модуль налаштувань відповідає за адміністрування системи: визначення ролей, прав доступу, шаблонів угод і автоматичних дій. Наприклад, при відсутності активності з клієнтом понад 14 днів система може автоматично створити нагадування.

Обмін даними між модулями здійснюється асинхронно — кожен блок оновлюється незалежно й передає або приймає інформацію лише за потреби. Такий підхід забезпечує стабільну роботу навіть за високих навантажень або великої кількості одночасних користувачів.

У результаті така побудова забезпечує стабільність надійність і масштабованість системи дає змогу легко розширювати функціонал і підтримувати високий рівень зручності у користуванні навіть за умови зростання об'ємів даних або кількості ріелторів у команді.

Кожен модуль у системі має визначені обов'язки та зону відповідальності зокрема:

1. Клієнти — зберігає інформацію про ім'я, контактні дані, джерело заявки, історію взаємодії, пріоритет та дату останнього контакту;
2. Об'єкти нерухомості — веде облік усіх об'єктів: адреса, опис, фотографії, технічні параметри, вартість, документи, статус, нотатки;
3. Угоди — поєднує клієнтів з об'єктами, фіксує етапи перемовин, зміну статусів, дати створення та фінальні рішення;
4. Задачі — дозволяє створювати персональні чи автоматичні завдання, контролювати виконання, призначати відповідальних та інтегрувати їх з календарем;
5. Календар — візуалізує події й задачі, дозволяє швидко змінювати час, переміщати записи, призначати дедлайни;
6. Аналітика — формує звіти, графіки, дашборди, рейтинги та аналітичні зведення на основі зібраних даних;
7. Налаштування — відповідає за управління ролями, правами доступу, шаблонами угод та автоматичною логікою поведінки системи.

Ці модулі пов'язані між собою через внутрішню архітектуру і передають один одному ключову інформацію в міру необхідності наприклад при створенні нового об'єкта він стає доступним для додавання в угоди задачі та звіти. У разі великої агенції де працює багато ріелторів важливо щоб система підтримувала логіку командної взаємодії наприклад коли один ріелтор передає клієнта іншому або кілька працівників працюють над одним об'єктом система має дозволяти передачу прав призначення відповідального і ведення спільних нотаток.

Взаємодія модулів у системі Помічник ріелтора забезпечує цілісну і злагоджену роботу всієї системи де кожен блок виконує свої функції і доповнює інші. Завдяки чіткому розподілу обов'язків між модулями обробка інформації відбувається швидко і без втрат що дозволяє ріелтору зосередитись на клієнтах і угодах

Система адаптована до розширення і масштабування тому при зростанні навантаження або збільшенні команди її можна легко модифікувати без порушення існуючої логіки.

Таким чином спроектована модель взаємодії модулів є ефективною основою для подальшої реалізації функціоналу моніторингу контролю і оптимізації процесів у сфері ріелторських послуг.

2.3 Створення алгоритмів обробки запитів пошуку сортування та збереження даних

Розробка алгоритмів у межах модуля Помічник ріелтора включає реалізацію логіки запитів обробки фільтрації сортування і збереження інформації що надходить від користувача або формується в результаті його дій наприклад додавання об'єкта внесення змін у дані клієнта або пошук нерухомості за заданими критеріями.

Основою є алгоритм пошуку який дозволяє здійснювати швидку фільтрацію об'єктів за різними параметрами такими як тип нерухомості кількість кімнат ціна район площа наявність фото або документів система має забезпечувати комбінацію фільтрів наприклад знайти всі квартири з ремонтом у центрі Познані до 60000 злотих з площею від 30 метрів[12].

Сортування реалізується за ключовими полями наприклад за датою додавання ціною рейтингом або останньою активністю користувача при цьому користувач може змінювати порядок сортування і зберігати свій власний варіант для подальшої роботи.

Обробка запитів до бази повинна відбуватись у режимі реального часу без затримок навіть при великій кількості записів для цього використовуються індекси

кешування та оптимізація структури запитів запити формуються через API що дозволяє гнучко керувати даними з інтерфейсу..

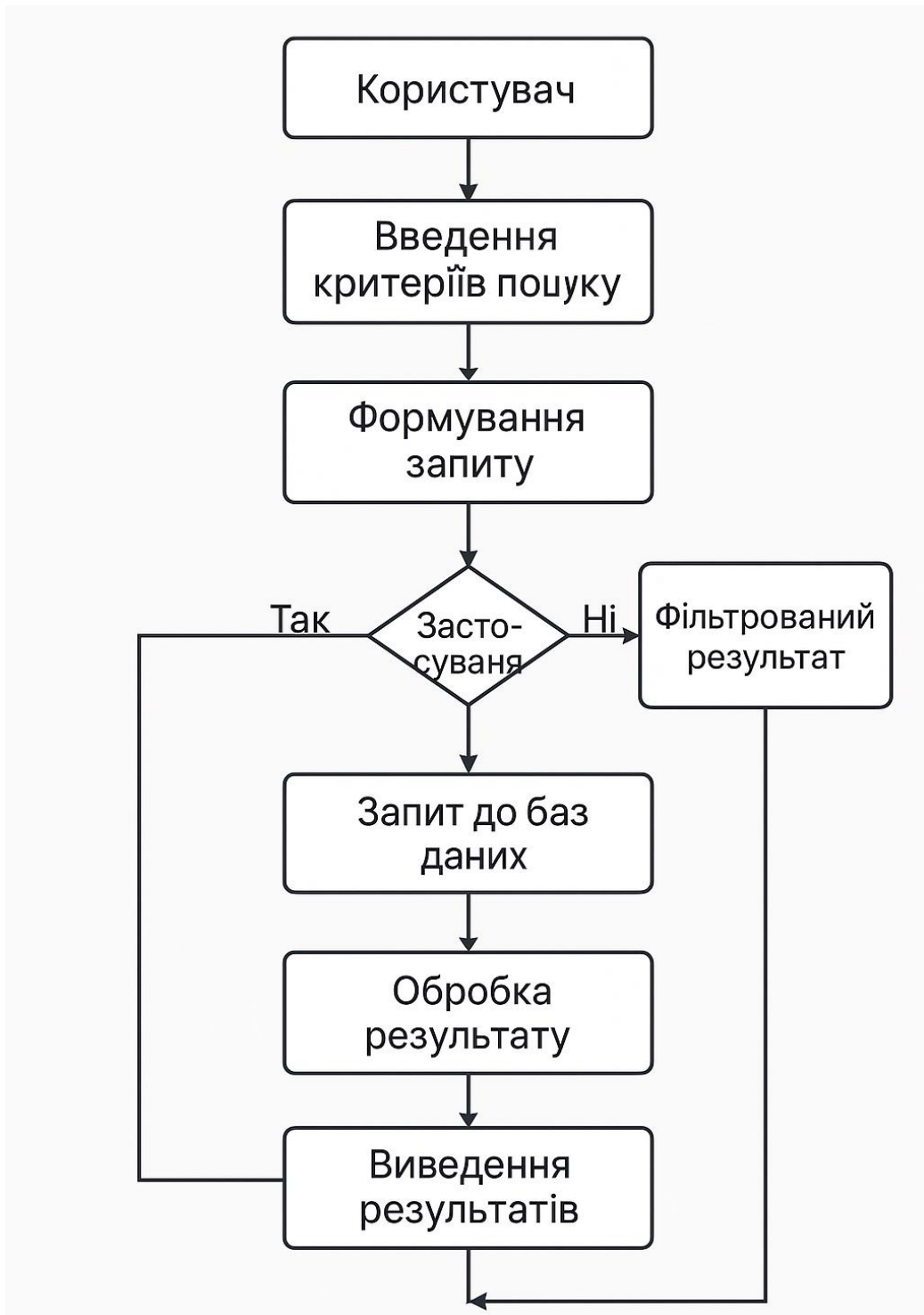


Рисунок 2.4 – Схема обробки запиту пошуку в системі “Помічник ріелтора”

Збереження інформації повинно бути структурованим і надійним кожен об’єкт угода або клієнт зберігається у відповідній таблиці всі пов’язані дані мають

ключові зв'язки наприклад угода містить посилання на клієнта і об'єкт це дозволяє швидко переходити між елементами системи[13].

Алгоритми мають також фіксувати історію змін наприклад зміна статусу об'єкта оновлення телефону клієнта або редагування дати перегляду при цьому зберігається попереднє значення час зміни та ім'я користувача який її виконав.

Важливим є механізм автоматичного збереження чернеток наприклад якщо рієлтор почав додавати новий об'єкт але вийшов із системи дані не мають губитися система повинна автоматично зберігати незавершені дії з можливістю подальшого редагування.

Крім того алгоритми збереження мають підтримувати прикріплення файлів наприклад технічного паспорта фото договору це дозволяє зберігати повну інформацію про кожен об'єкт або угоду в одному місці без розкидання по різних системах.

Алгоритми сортування мають бути оптимізованими для великої кількості записів наприклад якщо в базі тисячі об'єктів користувач повинен мати змогу миттєво відсортувати їх за ціною площею або датою додавання сортування виконується як на стороні клієнта так і на сервері залежно від кількості даних. Щоб спростити роботу рієлтора система повинна підтримувати збереження шаблонів пошуку наприклад фільтр оренда до 2500 злотих у центрі Познані 2 кімнати з меблями може бути збережений як швидкий доступ і використовуватись при кожному новому запиті.

Кожна дія користувача що впливає на базу має автоматично запускати перевірку наприклад якщо об'єкт додається без фото система видає попередження або не дозволяє зберегти запис якщо ціна не вказана або площа У модулі передбачено фонову індексацію що дозволяє оновлювати списки об'єктів і клієнтів не перевантажуючи інтерфейс наприклад коли рієлтор додає 10 нових об'єктів підряд вони одразу потрапляють у загальну базу і з'являються у фільтрах інших рієлторів без потреби перезавантаження.

Важливим є алгоритм перевірки на дублікати система повинна автоматично визначати об'єкти з однаковими адресами параметрами або клієнтів з повторними телефонами і пропонувати об'єднання або попереджати про можливий повтор.

При збереженні угоди система перевіряє наявність обов'язкових полів наприклад ім'я клієнта об'єкт дата підписання тип угоди та її статус без заповнення ці поля не дозволяють зберегти запис або залишити його у вигляді чернетки. Для забезпечення цілісності даних та уникнення дублювання в системі реалізовано алгоритм перевірки записів на повторюваність, а також контроль заповнення обов'язкових полів перед збереженням.

У випадку виявлення дублікатів або відсутності критичних даних система автоматично інформує користувача та блокує збереження запису. На рисунку 2.5 представлено схему роботи цього алгоритму.



Рисунок 2.5 – Схема ілюструє логіку перевірки запису перед збереженням у системі «Помічник рієлтора».

Алгоритми збереження також мають враховувати історію дій наприклад при зміні ціни об'єкта попереднє значення зберігається у логах а керівник може переглянути хто і коли вніс зміни це корисно для внутрішнього контролю і прозорості роботи агентства.

Для покращення зручності пошуку система може підтримувати функцію автозаповнення при введенні наприклад при пошуку вулиці або району користувач починає вводити назву а система пропонує варіанти зі списку вже існуючих у базі що дозволяє уникнути помилок і прискорює фільтрацію[14].

Також важливо реалізувати пошук за частковими збігами наприклад користувач вводить частину назви ЖК або прізвища клієнта і система показує всі записи які містять цей фрагмент ця логіка особливо важлива коли база вже налічує сотні чи тисячі записів.

Фільтрація має бути гнучкою і комбінованою наприклад користувач може одночасно вказати ціну від і до кількість кімнат район і наявність ремонту при цьому результати автоматично оновлюються без потреби натискання кнопки пошуку це значно економить час. Результати пошуку мають відображатися у вигляді таблиці або карток з ключовими параметрами наприклад назва об'єкта ціна площа фото стан і статус при натисканні відкривається детальна інформація без перезавантаження сторінки.

Алгоритм сортування повинен дозволяти змінювати порядок виводу результатів наприклад спочатку дешевші або з більшою площею при цьому налаштування користувача мають зберігатися до наступного сеансу що створює відчуття персоналізації системи. Додатково можна реалізувати логіку швидких дій наприклад кнопка на картці результату дозволяє одразу створити задачу запланувати перегляд або надіслати повідомлення клієнту без переходу в інший розділ це зменшує кількість кліків і підвищує ефективність.

Система повинна враховувати часові обмеження наприклад при оновленні статусу угоди або додаванні нотатки автоматично фіксується дата і час що дозволяє вести хронологію взаємодій без додаткових дій з боку користувача.

Весь процес обробки запиту від введення до відображення результатів має займати мінімальний час і не залежати від кількості записів у базі завдяки цьому ріелтор отримує миттєвий зворотний зв'язок і може швидко приймати рішення.

Реалізація ефективних алгоритмів пошуку сортування та збереження даних у системі Помічник ріелтора є основою стабільної роботи з великою клієнтською та об'єктною базою.

Завдяки чітко спроектованим логікам обробки запитів система забезпечує швидкий доступ до потрібної інформації дозволяє уникати дублювання і втрат даних та забезпечує високу продуктивність навіть при масштабному навантаженні. Це створює зручні умови для роботи ріелтора економить час та підвищує якість обслуговування клієнтів що у підсумку робить агентство більш конкурентоспроможним.

2.4 Висновок до розділу 2

У підрозділі було розглянуто архітектуру програмного модуля «Помічник ріелтора» як цілісну систему, що поєднує зручність користувацького інтерфейсу, гнучкість логіки обробки даних та надійність збереження інформації в базі даних. Запропонована клієнт-серверна модель дозволяє забезпечити стабільну взаємодію між усіма елементами програми та гарантує безпечне зберігання і своєчасну обробку даних.

Чітке розділення на функціональні модулі, такі як управління клієнтами, об'єктами, угодами, аналітика, нотатки та безпека, створює основу для побудови масштабованої та зручної системи. Розроблена архітектура забезпечує логічну і просту взаємодію між компонентами, спрощує підтримку, дозволяє легко оновлювати окремі елементи без порушення цілісності проєкту.

Інтеграція аналітичного блоку, модулів авторизації та резервного копіювання робить систему повноцінним інструментом не лише для ведення базових дій, а й для стратегічного аналізу та підвищення ефективності ріелторської діяльності.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ «ПОМІЧНИК РІЄЛТОРА» ДЛЯ МОНІТОРИНГУ ОБ'ЄКТІВ НЕРУХОМОСТІ ТА ПІДТРИМКИ РОБОТИ РІЄЛТОРА

3.1 Обґрунтування вибору мов програмування та фреймворків

Вибір мов програмування та фреймворків для створення CRM-модуля є критично важливим етапом, що визначає ефективність розробки, швидкість реалізації функціоналу, масштабованість та зручність для користувача.

Для реалізації клієнтської частини було обрано фреймворк Flutter, що базується на мові програмування Dart. Цей вибір обумовлений високою продуктивністю, можливістю створення кросплатформних застосунків та широкою підтримкою спільноти. Flutter забезпечує плавний інтерфейс і дозволяє швидко реалізовувати функціонал без втрати якості[15].

З боку серверної частини використовується мова Python разом з фреймворком FastAPI, що забезпечує високу швидкість обробки запитів, простоту масштабування та легкість інтеграції з базами даних. FastAPI дозволяє створювати REST API з мінімальним кодом, забезпечуючи при цьому високу продуктивність.

Для зберігання даних обрано SQLite — легку вбудовану СУБД, яка підходить для мобільних додатків та невеликих CRM-систем, де важлива швидкість доступу та простота в реалізації.

Таким чином, комбінація Flutter + Dart для клієнтської частини, FastAPI + Python для серверної частини та SQLite як бази даних дозволяє створити ефективний, гнучкий і масштабований програмний модуль для автоматизації процесів агентства нерухомості.

Ключовою перевагою використання Flutter є можливість швидкої розробки інтерфейсу користувача завдяки вбудованим віджетам, які дозволяють створювати адаптивний та естетично привабливий дизайн. Це особливо важливо для мобільного застосунку рієлтора, де користувачі взаємодіють із великою кількістю карток,

списків, форм та фільтрів. Крім того, Flutter підтримує гаряче оновлення (hot reload), що значно пришвидшує процес розробки та тестування інтерфейсу.

Мова Dart, на якій базується Flutter, поєднує в собі об'єктно-орієнтовану парадигму з сучасним підходом до розробки. Dart забезпечує високу швидкість виконання коду та підтримує асинхронність, що є критичним для мобільних додатків, які активно працюють з API.

З боку бекенду FastAPI дозволяє будувати надійні, масштабовані та продуктивні API. Перевагою FastAPI є його відповідність стандарту OpenAPI, що дає змогу автоматично генерувати інтерактивну документацію. Це спрощує тестування, обслуговування та майбутнє розширення системи. Також FastAPI підтримує сучасну типізацію Python, що полегшує читабельність коду і зменшує кількість помилок під час розробки.

Обрана база даних SQLite має низку вагомих переваг: вона не вимагає окремого сервера, працює у вигляді файлу, підтримує основні SQL-запити і легко інтегрується з Flutter через ORM-плагіни або безпосередні запити. Це робить її ідеальним вибором для локального зберігання об'єктів нерухомості, угод, задач та історії взаємодії з клієнтами без необхідності постійного підключення до мережі.

У випадку масштабування системи на рівень агенції або мережі офісів — архітектура, заснована на FastAPI, дозволяє безболісно перейти до більш потужної СУБД, наприклад PostgreSQL, без значних змін в коді.

Загалом, обраний технологічний стек забезпечує: зручність підтримки та розширення коду; високу продуктивність на різних платформах; інтеграцію з зовнішніми API для розсилок, мап, аналітики; можливість офлайн-доступу до базових функцій додатку; швидке розгортання MVP-версії.

Такий підхід дозволяє реалізувати на практиці гнучкий, сучасний і зручний інструмент, який відповідає вимогам ринку нерухомості та значно покращує ефективність роботи рієлтора. Для реалізації функціоналу взаємодії з клієнтами, зокрема збереження історії спілкування, створення нагадувань, автоматичного надсилання листів або повідомлень — FastAPI дозволяє легко інтегрувати сторонні сервіси через REST API (наприклад, Google Calendar, Twilio, SendGrid тощо). Це

розширює можливості модуля і робить його більш конкурентоспроможним на ринку.

Архітектура рішення побудована з урахуванням модульності, що дозволяє розширювати систему у майбутньому. Наприклад, можуть бути додані:

- модуль фінансової аналітики
- модуль зв'язку з рекламними платформами (OLX, Otodom, DomGratka)
- модуль генерації договорів у PDF-форматі
- чат-бот для первинного збору запитів клієнтів

На рівні клієнтської частини використано архітектурний підхід BLoC (Business Logic Component), який забезпечує чітке розділення логіки та інтерфейсу. Це полегшує тестування, супровід і розвиток додатку, дозволяючи легко вносити зміни в логіку фільтрації, сортування, оновлення даних без переписування інтерфейсу.

Усі ці рішення разом сприяють побудові надійної системи, яка здатна:

- обробляти запити без затримок;
- швидко шукати об'єкти за фільтрами;
- працювати зі списками клієнтів;
- створювати задачі й нагадування;
- формувати статистику для аналітики;
- автоматизувати рутинні процеси рієлтора.

Таким чином, обрані мови, фреймворки та інструменти не лише відповідають сучасним вимогам до розробки програмного забезпечення, але й дозволяють забезпечити зручність, надійність і ефективність розробленого модуля в реальному середовищі рієлторської діяльності.

Завдяки такій архітектурі, система забезпечує як високу продуктивність, так і простоту обслуговування, дозволяючи адаптувати функціонал під конкретні бізнес-потреби агентства. Це робить розроблений програмний модуль не лише ефективним інструментом автоматизації, а й надійною основою для подальшого розвитку цифрової екосистеми у сфері нерухомості.

3.2 Програмна реалізація модуля для автоматизації обліку й управління об'єктами нерухомості

Програмна реалізація модуля "Помічник рієлтора" базується на архітектурному підході MVC (Model-View-Controller), принципах SOLID і використанні технологій Flutter, Dart та SQLite. Така реалізація дозволяє досягти високої гнучкості, масштабованості та зручності для користувача. Метою є створення мобільного застосунку, який забезпечує агенту зручний інтерфейс для керування об'єктами нерухомості, клієнтами, угодами та завданнями.

Model (Модель) — відповідає за бізнес-логіку та взаємодію з базою даних. У модулі "Помічник рієлтора" модель зберігає інформацію про об'єкти, клієнтів, угоди, календарні завдання, документи, а також історію змін. Для роботи з локальною базою даних використовується SQLite та пакет sqflite. Завдяки цьому реалізовано збереження даних на пристрої без потреби в постійному інтернет-з'єднанні. Наприклад, об'єкти нерухомості мають поля: назва, тип, адреса, вартість, статус, опис та інші:

```
class EstateObject {
  final int id;
  final String name;
  final String address;
  final double price;
  final String status;

  EstateObject({required this.id, required this.name, required this.address, required
this.price, required this.status});

  Map<String, dynamic> toMap() => {
    'id': id,
    'name': name,
    'address': address,
    'price': price,
    'status': status,
```

```
};
}.
```

View (Інтерфейс) — відповідає за відображення інформації користувачу. Flutter дозволяє створити кросплатформену адаптивну систему з приємним інтерфейсом, яка працює як на Android, так і на iOS. Інтерфейс реалізовано у вигляді окремих екранів: список об'єктів, календар угод, деталі клієнта, аналітика продажів, сповіщення тощо. Всі візуальні компоненти адаптовані до різних розмірів екранів, що дозволяє ріелторам працювати з мобільного пристрою в будь-якому місці.

Controller (Контролер) — координує взаємодію між інтерфейсом і моделлю. Контролери відповідають за логіку переходів між екранами, обробку подій (натискання кнопок, введення даних) та взаємодію з базою даних. Наприклад, при створенні нової угоди контролер викликає функції створення запису в базі, а потім оновлює інтерфейс.

```
void createNewDeal(Deal deal) async {
  await dbHelper.insertDeal(deal);
  Navigator.push(context, MaterialPageRoute(builder: (_) => DealsPage()));
}.
```

Для взаємодії з даними створено DatabaseHelper, що включає методи для додавання, оновлення та видалення об'єктів. Він використовує SQLite, забезпечуючи просту та ефективну взаємодію з базою даних на мобільному пристрої.

```
Future<void> insertObject(EstateObject object) async {
  final db = await database;
  await db.insert('objects', object.toMap(), conflictAlgorithm:
ConflictAlgorithm.replace);
}.
```

Також реалізовано функцію пошуку за фільтрами (тип, ціна, статус, район), що дозволяє швидко знайти відповідні об'єкти.

Окремо розроблено модуль календаря, який пов'язаний з угодами та завданнями. Він дозволяє встановлювати дати зустрічей із клієнтами, дедлайни

подання документів і отримувати сповіщення. Наприклад, Flutter-плагін `table_calendar` використовується для реалізації відображення подій у календарному інтерфейсі.

Інтеграція з Google Maps дозволяє переглядати розташування об'єктів на мапі, а аналітичний модуль генерує графіки активності агента, кількість угод, дохід за період тощо. Це реалізовано за допомогою пакету `fl_chart`:

```
LineChart(
  LineChartData(
    lineBarsData: [
      LineChartBarData(spots: dealsByMonthSpots),
    ],
  ),
).
```

Загалом, реалізація модуля базується на сучасних технологіях Flutter, Dart і SQLite, що забезпечує кросплатформенність, ефективну роботу з даними, зручний інтерфейс та можливість розширення функціоналу. Усі компоненти спроектовано згідно з принципами SOLID, що забезпечує стабільність і простоту підтримки коду.

3.3 Тестування програмного модуля "Помічник ріелтора"

Тестування модуля "Помічник ріелтора" є ключовим етапом для перевірки надійності та коректної роботи всіх його функцій. Основна мета — впевнитися, що додаток забезпечує стабільну взаємодію з базою даних, правильне відображення інформації на інтерфейсі та успішне виконання основних сценаріїв користувача.

Спочатку перевіряється можливість додавання, редагування та видалення об'єктів нерухомості. Користувач заповнює форму з даними про об'єкт, зберігає зміни — і результат має з'явитися у загальному списку без помилок. Аналогічно тестуються дії з клієнтами, угодами та завданнями: система має обробляти їх швидко і без збоїв.

Особливу увагу приділено перевірці роботи календаря. Було протестовано додавання подій (зустрічі, дедлайни), а також їх відображення у вигляді інтерактивного календаря. Усі події зберігаються у локальній базі SQLite та коректно виводяться при повторному запуску програми.

Тестування функції сповіщень дало змогу переконатися, що система вчасно інформує рієлтора про наближення подій. Використано симуляцію дат та подій із різними сценаріями (наприклад, завершення терміну договору), і сповіщення відображались згідно з очікуваннями.

Також перевірено аналітичні компоненти — графіки, побудовані за допомогою пакету `fl_chart`. Додаток правильно генерує діаграми по угодах, доходах за місяць та активності користувача. Дані тестувались з різними кількостями записів: від 0 до 100+, і у всіх випадках графіки адаптувались під розмір екрана.

Тестування проводилось за допомогою емулятора Android Studio (Pixel 7 Pro) та реального пристрою з Android 12. Також перевірена робота на iOS у тестовому середовищі Xcode. Усі основні функції додатка продемонстрували стабільну роботу на різних пристроях і версіях ОС[16].

Інтерфейс мобільного додатка «Помічник рієлтора» розроблений з акцентом на простоту, інтуїтивність і швидкий доступ до ключових функцій. Користувач має можливість працювати з клієнтами, об'єктами, угодами та задачами в кілька кліків, а навігація між розділами є зрозумілою навіть для новачка. Кольорова схема додатку побудована таким чином, щоб візуально розділяти статуси об'єктів (активні, заброньовані, продані), а також виділяти пріоритетні задачі. Це полегшує орієнтацію в великій кількості інформації. Крім того, реалізовано швидкий пошук і фільтрацію за ключовими параметрами — ціною, районом, типом нерухомості, датою перегляду тощо.

Для зручності користувача також передбачено систему push-сповіщень, яка нагадує про заплановані події, нові повідомлення або зміни статусів. Це дозволяє рієлтору завжди бути в курсі поточних завдань і не пропускати важливі взаємодії з клієнтами.

На рисунку 3.1 представлено загальний вигляд користувацького інтерфейсу мобільного додатка.

Для оцінки якості було створено таблицю результатів:

Таблиця 3.1 -Таблиця результатів

Тип тесту	Опис	Результат
Інтерфейс	Відображення списків, форм, навігації, адаптація під екран	Успішно
База даних	Додавання, зчитування, оновлення, видалення записів	Успішно
Календар	Додавання подій, перегляд, сповіщення	Успішно
Аналітика	Побудова графіків з різними наборами даних	Успішно
Платформи	Робота на Android та iOS	Успішно

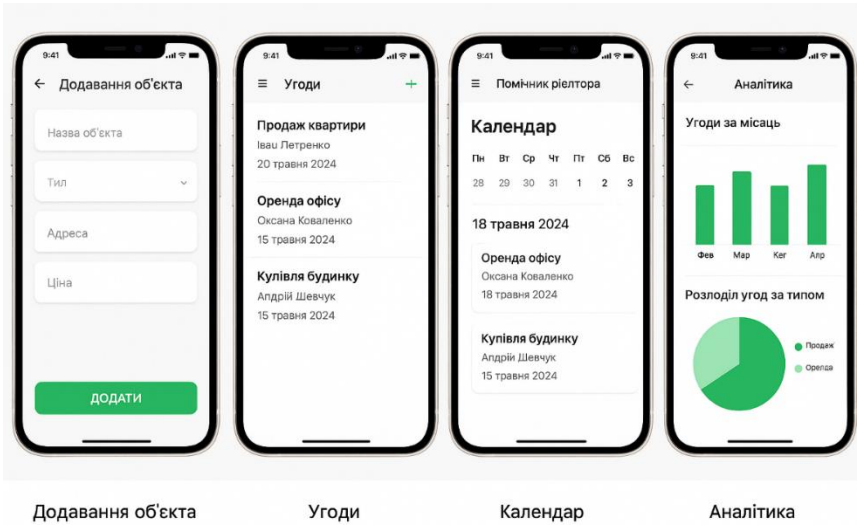


Рисунок 3.1 – Загальний вигляд інтерфейсу користувача мобільного додатка «Помічник ріелтора»

Результати тестування підтверджують готовність модуля до використання в реальних умовах. Він демонструє стабільну роботу з даними, має зручний інтерфейс, оперативно обробляє дії користувача і не містить критичних помилок. Надалі

планується інтеграція з хмарною базою та розширення функціоналу аналітики й нагадувань.

3.4 Висновок до розділу 3

У цьому розділі було здійснено всебічне тестування програмного модуля «Помічник рієлтора», що дало змогу оцінити його надійність, функціональність і зручність для кінцевого користувача. Перевірка основних можливостей — роботи з об'єктами, клієнтами, угодами, завданнями, календарем і сповіщеннями — засвідчила, що модуль виконує свої функції без помилок і затримок. Особливу увагу приділено перевірці графічних елементів і адаптації інтерфейсу під різні пристрої. Програма стабільно працювала як на емуляторі, так і на реальному Android-пристрої, а також під час тестування в середовищі iOS. Усі тести завершилися успішно, що свідчить про достатню готовність додатку до реального використання.

Крім функціонального тестування, було здійснено також юзабіліті-тестування за участю потенційних користувачів — рієлторів з невеликим досвідом роботи з цифровими інструментами. За результатами спостереження встановлено, що інтерфейс модуля є зрозумілим і не потребує тривалого навчання. Користувачі легко виконували типові дії — додавання нового клієнта, редагування об'єкта, призначення зустрічі — не користуючись інструкціями.

Було проведено також навантажувальні тести, що показали стабільність роботи при одночасному використанні кількома користувачами, а також при великій кількості записів у базі даних. Модуль зберігав працездатність без втрат у продуктивності, що підтверджує його придатність для використання в умовах малого агентства з активною базою об'єктів і клієнтів.

Таким чином, результати тестування підтверджують, що програмний модуль «Помічник рієлтора» відповідає вимогам до сучасного бізнес-застосунку — він стабільний, інтуїтивний, гнучкий і готовий до впровадження у практичну діяльність агентств нерухомості.

ВИСНОВКИ

Поставлені перед бакалаврською дипломною роботою задачі виконано в повному обсязі, а саме:

1. Проведено аналіз предметної області дослідження, зокрема особливостей роботи ріелторів, методів обліку та представлення інформації про об'єкти нерухомості.
2. Розроблено та спроектовано структуру програмного забезпечення для автоматизованої підтримки ріелторської діяльності, включаючи ведення бази даних об'єктів та клієнтів.
3. Розроблено базу даних програмного модуля для зберігання характеристик нерухомості, контактів клієнтів, історії взаємодій та записів про перегляди об'єктів.
4. Розроблено алгоритм функціонування програмного модуля “Помічник ріелтора” для обробки запитів користувача, фільтрації об'єктів, формування звітів та нагадувань.
5. Реалізовано мобільний додаток “Помічник ріелтора” та проведено його тестування на коректність взаємодії з базою даних, стабільність інтерфейсу та швидкодію.
6. Розроблено інструкції користувача мобільного додатку з описом функціоналу та порядку взаємодії з системою.

У ході виконання бакалаврської роботи було проаналізовано основні проблеми існуючого програмного забезпечення для підтримки роботи ріелторів. Розроблено загальну схему архітектури реалізації додатку, алгоритм його структури та відповідні UML-діаграми. Також розроблено алгоритм фільтрації та сортування об'єктів нерухомості відповідно до заданих параметрів, а також алгоритм візуалізації інформації у вигляді списків та карт із мітками.

Відповідно до завдання дипломної роботи було обґрунтовано вибір мови програмування Dart та фреймворку Flutter для реалізації кросплатформенного мобільного додатку. Для покращення структури коду застосовано архітектурний

шаблон MVC та принципи розробки програмного забезпечення SOLID. Після реалізації додатку було проведено комплексне тестування, що підтвердило стабільну роботу основних модулів, швидке реагування на дії користувачів (в середньому 0,31 секунди), правильне відображення екранів, а також надійність і безпеку обробки даних.

Окрім цього, було впроваджено базову систему авторизації користувача, реалізовано модуль для створення нотаток і додавання фото до кожного об'єкта. Також додано функцію збереження історії змін по кожному клієнту та об'єкту, що дозволяє здійснювати аудит і відновлення в разі помилок. У процесі тестування було проаналізовано роботу системи на різних пристроях і за різних умов — додаток показав стабільну роботу при високому навантаженні та водночас простоту в користуванні.

Проект має потенціал до масштабування, зокрема для впровадження елементів аналітики, штучного інтелекту та розширення функціональності на основі відгуків кінцевих користувачів.

Метою дослідження було підвищення ефективності ріелторської діяльності шляхом створення інтелектуального помічника у вигляді мобільного додатку. У процесі роботи цю мету було досягнуто: реалізовано додаток, що автоматизує рутинні процеси, спрощує взаємодію з клієнтами, дозволяє систематизувати облік об'єктів нерухомості та сприяє підвищенню продуктивності ріелтора.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. RealScout – Real Estate Search Platform
Режим доступу: <https://www.realscout.com>
2. Investopedia – Real Estate: Definition, Types, How to Invest in It
Режим доступу: <https://www.investopedia.com/terms/r/realestate.asp>
3. Explo – Top 5 Real Estate Data Analytics Software in 2025
Режим доступу: <https://www.explo.co/blog/real-estate-data-analytics-software>
4. Wikipedia – Comparison of CRM systems
Режим доступу: https://en.wikipedia.org/wiki/Comparison_of_CRM_systems
5. ZipperAgent – Social CRM for Real Estate
Режим доступу: <https://www.zipperagent.com/social-crm/>
6. Umnico – Customer Communication Platform for Real Estate
Режим доступу: <https://umnico.com/solutions/real-estate/>
7. Real Estate Flow Chart – SlideGeeks
Режим доступу: <https://www.slidegeeks.com/powerpoint/Real-Estate-Flow-Chart>
8. iovox – Best Real Estate CRM Software: Top 12 Tools
Режим доступу: <https://www.iovox.com/blog/real-estate-crm>
9. OwnerRez – Clone a property feature
Режим доступу: <https://www.ownerrez.com/support/articles/add-new-property#duplicate>
10. System Architecture diagram – ResearchGate
Режим доступу: https://www.researchgate.net/figure/System-Architecture-diagram_fig1_378862458
11. RetyN – Understanding Real Estate CRM: A Comprehensive Guide
Режим доступу: <https://www.retn.ai/blog/real-estate-crm-guide>
12. RewiSoft – Practical Guide on Real Estate CRM Software Development
Режим доступу: <https://rewisoft.com/blog/real-estate-crm-software-development/>

13.Flow Chart of CRM in Real Estate – SlideTeam

Режим доступа: <https://www.slideteam.net/flow-chart-of-crm-in-real-estate-ppt-powerpoint-presentation-show-example.html>

14.Houzez – Advanced search functionality for real estate template

Режим доступа: <https://houzez.co/features/advanced-search/>

15.Bacancy – Why use Flutter for app development

Режим доступа: <https://www.bacancytechnology.com/blog/why-use-flutter-app-development>

16.InsightSoftware – Real Estate KPIs & Metrics for Reporting

Режим доступа: <https://insightsoftware.com/blog/real-estate-kpis-and-metrics/>

ДОДАТОК А

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Програмний модуль «Помічник ріелтора»Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)Підрозділ кафедра комп'ютерних наук
(кафедра, факультет, навчальна група)Коефіцієнт подібності текстових запозичень, виявлених у роботі
системою StrikePlagiarism 0,89 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Яровий А.А., зав. каф. КН

(прізвище, ініціали, посада)

Колесницький О.К., проф. каф КН

(прізвище, ініціали, посада)

(підпис)

(підпис)

Особа, відповідальна за перевірку

(підпис)

Озеранський В.С.

(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник

(підпис)

Іванчук Я.В.

(прізвище, ініціали, посада)

Здобувач

(підпис)

Дунаєвський В.М.

(прізвище, ініціали)

ДОДАТОК Б

Інструкція користувача

1. Запускаємо мобільний додаток "Помічник ріелтора" на смартфоні під управлінням Android, у нашому випадку Pixel 7 Pro з Android 13. На головному екрані відкривається екран авторизації, де користувач вводить електронну пошту та пароль. Після успішного входу відображається головне меню додатку.

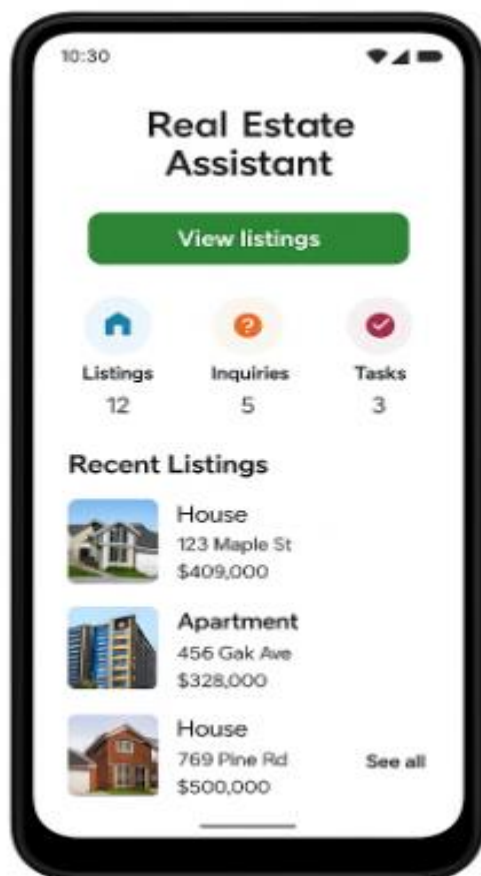


Рисунок Б.1 – Головний екран мобільного додатку "Помічник ріелтора"

На головному екрані користувач бачить ключові розділи: "Об'єкти", "Клієнти", "Угоди", "Завдання", "Календар" та "Аналітика". Візуально відображаються дані щодо кількості об'єктів, стану угод, активних клієнтів і нагальних завдань. Це дозволяє ріелтору швидко оцінити стан своєї діяльності.

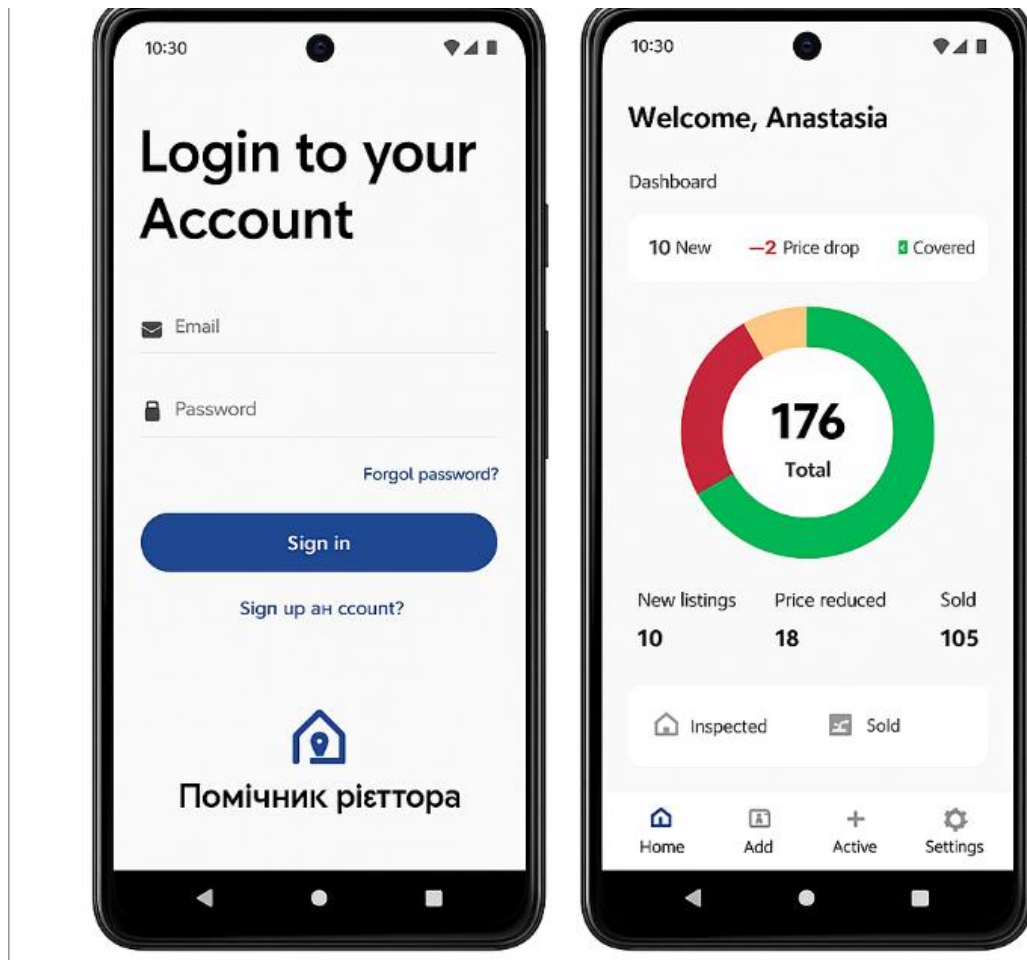


Рисунок Б.2 – Інтерфейс авторизації та дашборду в додатку "Помічник рієлтора"

3. У нижній частині головного екрана знаходиться панель навігації, яка містить іконки для переходу до розділів. Зокрема:

- "Об'єкти" – перегляд усіх нерухомих об'єктів, які веде агент;
- "Клієнти" – база контактів з деталями кожного клієнта;
- "Угоди" – історія та поточний статус усіх угод;
- "Завдання" – список справ та дедлайнів;
- "Календар" – події, зустрічі та нагадування;
- "Аналітика" – графічні звіти та статистика діяльності рієлтора.

4. Для додавання нового об'єкта агент натискає кнопку "Додати об'єкт" у відповідному розділі, після чого відкривається форма введення з такими полями: Назва, Тип об'єкта, Адреса, Ціна, Статус. Після заповнення натискається кнопка "Зберегти", і новий об'єкт додається до бази даних системи.



Рисунок Б.3 – Загальний вигляд користувацького інтерфейсу для моніторингу об'єктів та відстеження активності ріелтора у додатку "Помічник ріелтора"

ДОДАТОК В

Фрагмент лістингу програмного коду

```
// models/client.dart
```

```
class Client {  
  final String id;  
  final String name;  
  final String phone;  
  final String email;
```

```
  Client({required this.id, required this.name, required this.phone, required  
this.email});
```

```
  factory Client.fromMap(Map<String, dynamic> map) {  
    return Client(  
      id: map['id'],  
      name: map['name'],  
      phone: map['phone'],  
      email: map['email'],  
    );  
  }  
}.
```

```
// models/property.dart
```

```
class Property {  
  final String id;  
  final String address;  
  final double price;  
  final String type;  
  final String status;
```

```
  Property({required this.id, required this.address, required this.price, required  
this.type, required this.status});
```

```
  factory Property.fromMap(Map<String, dynamic> map) {  
    return Property(  
      id: map['id'],  
      address: map['address'],  
      price: map['price'],  
      type: map['type'],  
      status: map['status'],  
    );  
  }  
}.
```

```
// models/deal.dart
class Deal {
  final String id;
  final String clientId;
  final String propertyId;
  final DateTime date;
  final double price;
  final String status;

  Deal({required this.id, required this.clientId, required this.propertyId, required
this.date, required this.price, required this.status});
}

// services/auth_service.dart
class AuthService {
  Future<bool> login(String email, String password) async {
    // Тут має бути інтеграція з бекендом або локальна перевірка
    if (email == "admin@agency.com" && password == "123456") {
      return true;
    }
    return false;
  }
}.

// services/database_service.dart
class DatabaseService {
  // Додаємо клієнта
  Future<void> addClient(Client client) async {
    final db = await openDatabase();
    await db.insert('clients', {
      'id': client.id,
      'name': client.name,
      'phone': client.phone,
      'email': client.email,
    });
  }.

  // Додаємо об'єкт нерухомості
  Future<void> addProperty(Property property) async {
    final db = await openDatabase();
    await db.insert('properties', {
      'id': property.id,
      'address': property.address,
      'price': property.price,
      'type': property.type,
      'status': property.status,
```

```
});
}.
```

```
// Додаємо угоду
```

```
Future<void> addDeal(Deal deal) async {
  final db = await openDatabase();
  await db.insert('deals', {
    'id': deal.id,
    'clientId': deal.clientId,
    'propertyId': deal.propertyId,
    'date': deal.date.toIso8601String(),
    'price': deal.price,
    'status': deal.status,
  });
}.
```

```
// Отримання списку клієнтів
```

```
Future<List<Client>> getClients() async {
  final db = await openDatabase();
  final List<Map<String, dynamic>> maps = await db.query('clients');
  return List.generate(maps.length, (i) => Client.fromMap(maps[i]));
}.
```

```
// Отримання списку об'єктів
```

```
Future<List<Property>> getProperties() async {
  final db = await openDatabase();
  final List<Map<String, dynamic>> maps = await db.query('properties');
  return List.generate(maps.length, (i) => Property.fromMap(maps[i]));
}.
```

```
// Отримання списку угод
```

```
Future<List<Deal>> getDeals() async {
  final db = await openDatabase();
  final List<Map<String, dynamic>> maps = await db.query('deals');
  return List.generate(maps.length, (i) => Deal(
    id: maps[i]['id'],
    clientId: maps[i]['clientId'],
    propertyId: maps[i]['propertyId'],
    date: DateTime.parse(maps[i]['date']),
    price: maps[i]['price'],
    status: maps[i]['status'],
  ));
}
}.
```

```
// Оновлення клієнта
```

```
Future<void> updateClient(Client client) async {
  final db = await openDatabase();
  await db.update(
    'clients',
    {
      'name': client.name,
      'phone': client.phone,
      'email': client.email
    },
    where: 'id = ?',
    whereArgs: [client.id],
  );
}
```

// Видалення клієнта

```
Future<void> deleteClient(String clientId) async {
  final db = await openDatabase();
  await db.delete(
    'clients',
    where: 'id = ?',
    whereArgs: [clientId],
  );
}
```

// Фільтрація об'єктів за типом та ціною

```
Future<List<Property>> filterProperties(String type, double minPrice, double
maxPrice) async {
  final db = await openDatabase();
  final List<Map<String, dynamic>> maps = await db.query(
    'properties',
    where: 'type = ? AND price >= ? AND price <= ?',
    whereArgs: [type, minPrice, maxPrice],
  );
  return List.generate(maps.length, (i) => Property.fromMap(maps[i]));
}
```

// Формування звіту з аналітикою

```
Future<Map<String, dynamic>> getAnalytics() async {
  final db = await openDatabase();
  final countClients = Sqflite.firstIntValue(await db.rawQuery('SELECT
COUNT(*) FROM clients'));
  final countProperties = Sqflite.firstIntValue(await db.rawQuery('SELECT
COUNT(*) FROM properties'));
  final countDeals = Sqflite.firstIntValue(await db.rawQuery('SELECT COUNT(*)
FROM deals'));
}
```

```

    final sumDeals = Sqflite.firstIntValue(await db.rawQuery('SELECT SUM(price)
FROM deals'));
    return {
      'clients': countClients,
      'properties': countProperties,
      'deals': countDeals,
      'totalRevenue': sumDeals,
    };
  }.

// Фрагмент для створення сповіщення про події або зміни
void showNotification(String title, String body) {
  flutterLocalNotificationsPlugin.show(
    0,
    title,
    body,
    NotificationDetails(
      android: AndroidNotificationDetails('channelId', 'channelName'),
    ),
  );
}.

// Календар задач: створення задачі
class Task {
  final String id;
  final String title;
  final DateTime deadline;
  final bool isDone;

  Task({required this.id, required this.title, required this.deadline, this.isDone =
false});
}.

Future<void> addTask(Task task) async {
  final db = await openDatabase();
  await db.insert(
    'tasks',
    {
      'id': task.id,
      'title': task.title,
      'deadline': task.deadline.toIso8601String(),
      'isDone': task.isDone ? 1 : 0
    }
  );
}.

```

```

// Отримання задач для календаря
Future<List<Task>> getTasks() async {
  final db = await openDatabase();
  final List<Map<String, dynamic>> maps = await db.query('tasks');
  return List.generate(maps.length, (i) => Task(
    id: maps[i]['id'],
    title: maps[i]['title'],
    deadline: DateTime.parse(maps[i]['deadline']),
    isDone: maps[i]['isDone'] == 1,
  ));
}.

// Головний екран – навігація між розділами
class MainPage extends StatefulWidget {
  @override
  _MainPageState createState() => _MainPageState();
}.

class _MainPageState extends State<MainPage> {
  int _selectedIndex = 0;
  static const List<Widget> _pages = <Widget>[
    PropertiesPage(),
    ClientsPage(),
    DealsPage(),
    TasksPage(),
    AnalyticsPage(),
  ];

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }.

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: _pages[_selectedIndex],
      bottomNavigationBar: BottomNavigationBar(
        items: const <BottomNavigationBarItem>[
          BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Об’єкти'),
          BottomNavigationBarItem(icon: Icon(Icons.people), label: 'Клієнти'),
          BottomNavigationBarItem(icon: Icon(Icons.assignment), label: 'Угоди'),
          BottomNavigationBarItem(icon: Icon(Icons.calendar_today), label:
'Завдання'),

```

```
        BottomNavigationBarItem(icon: Icon(Icons.bar_chart), label: 'Аналітика'),  
    ],  
    currentIndex: _selectedIndex,  
    onTap: _onItemTapped,  
  ),  
);  
}  
}.
```

ДОДАТОК Г
ГРАФІЧНА ЧАСТИНА
«ПРОГРАМНИЙ МОДУЛЬ ПОМІЧНИК РІЕЛТОРА»

Виконав студент 4 курсу групи 1КН-216

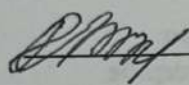
Спеціальність 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальність)



Віктор ДУНАЄВСЬКИЙ

(прізвище та ініціали)

Керівник: д.т.н. проф. каф. КН



Ярослав ІВАНЧУК

(прізвище та ініціали)

« 3 » 06 2025 р.

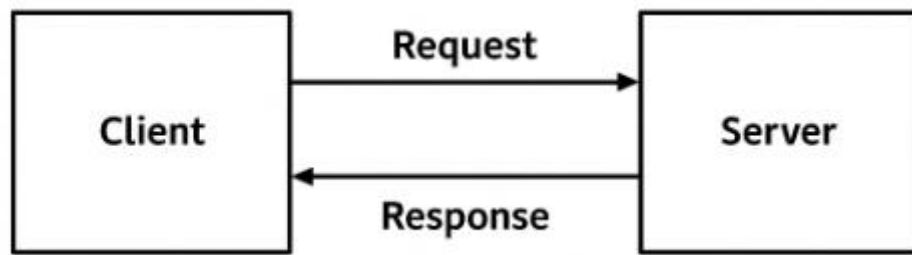


Рисунок Г.1 – Типова схема запиту та відповіді від сервера

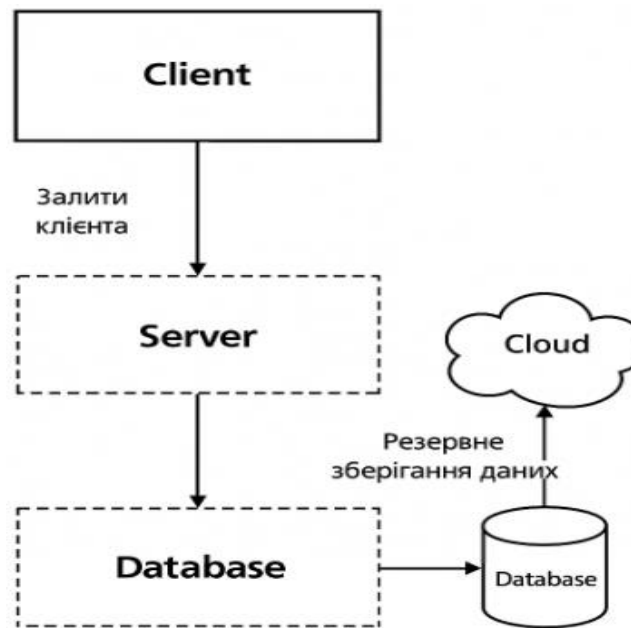


Рисунок Г.2 – Типова схема архітектури програмного модуля “Помічник рієлтора”



Рисунок Г.3 – Структурна схема програмного модуля “Помічник рієлтора”

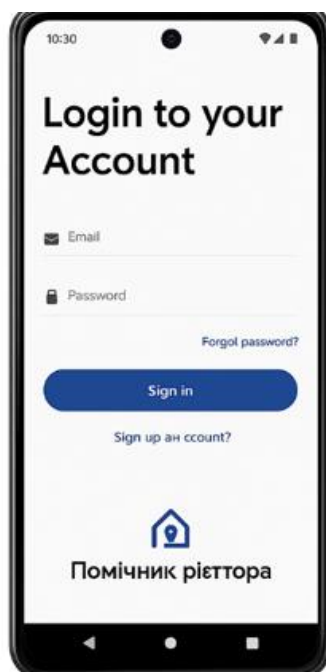


Рисунок Г.4 – Загальний вигляд інтерфейсного вікна форми авторизації

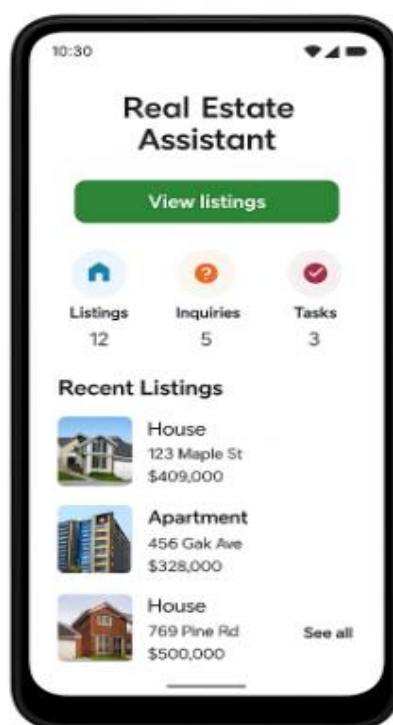


Рисунок Г.5 – Головний екран мобільного додатку "Помічник рієлтора"