

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавра

(назва освітнього ступеня)

на тему: Проектування CRM системи для оренди квартир з використанням Google Appsheet

Виконав: студент IV курсу, групи СП-42
спеціальності 121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

_____ Мельник О.Р.
(підпис) (прізвище та ініціали)

Керівник _____ Бревус В.М.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ Стоянов Ю.М.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Петрик М.Р.
(підпис) (прізвище та ініціали)

Рецензент _____ Луцик Н.С.
(підпис) (прізвище та ініціали)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності та основи охорони праці			

7. Дата видачі завдання 18.04.2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Ознайомлений з завданням до кваліфікаційної роботи	18.04.2025	Виконано
2	Підбір та опрацювання літературних джерел по темі кваліфікаційної роботи	19.04.2025-23.04.2025	Виконано
3	Використання дослідження щодо розробки CRM системи	24.04.2025-28.04.2025	Виконано
4	Оформлення розділу «Аналіз стану проблеми та постановка задачі дослідження»	29.04.2025-05.05.2025	Виконано
5	Оформлення розділу «Обґрунтування вибору інструментів і методів дослідження»	06.05.2025-10.05.2025	Виконано
6	Оформлення розділу «розробка CRM-системи для оренди квартир»	11.05.2025-25.05.2025	Виконано
7	Виконання до підрозділу «Безпека життєдіяльності»	26.05.2025-30.05.2025	Виконано
8	Виконання до підрозділу «Основи охорони праці»	31.05.2025-04.06.2025	Виконано
9	Оформлення кваліфікаційної роботи	05.06.2025-15.06.2025	Виконано
10	Нормоконтроль	16.06.2025-18.06.2025	Виконано
11	Перевірка на антиплагіат	19.06.2025-20.06.2025	Виконано
12	Попередній захист кваліфікаційної роботи	21.06.2025	Виконано
13	Захист кваліфікаційної роботи	23.06.2025	Виконано

Студент

(підпис)

Мельник Олег Романович

(прізвище та ініціали)

Керівник роботи

(підпис)

Бреус В. М.

(прізвище та ініціали)

АНОТАЦІЯ

Атестаційна робота магістра містить: 66 с., 34 рис., 6 табл., 38 джер.

Ключові слова: CRM, оренда квартир, Google AppSheet, мобільний додаток, база даних, хмарні сервіси, автоматизація, цифрова трансформація, інтерфейс користувача, управління нерухомістю.

У даній атестаційній роботі розглянуто процес проектування та реалізації CRM-системи (системи управління взаємовідносинами з клієнтами) для автоматизації процесів оренди житлової нерухомості. Як основну платформу для реалізації рішення було обрано Google AppSheet – хмарний сервіс без необхідності програмування, який дозволяє створювати повнофункціональні мобільні та веб-додатки на основі електронних таблиць Google Sheets або інших джерел даних.

Метою роботи є спрощення процесу управління орендою квартир, зокрема обліку клієнтів, договорів, термінів оренди, платежів, технічного обслуговування та комунікації між орендодавцем і орендарем. У межах проєкту проаналізовано сучасні підходи до цифровізації управління нерухомістю, розроблено структуру бази даних, створено інтерфейси для зручної взаємодії користувачів, впроваджено автоматичні сповіщення, фільтри та форми для введення й редагування даних.

Розроблена CRM-система дає змогу значно підвищити ефективність роботи з клієнтами, мінімізувати людський фактор у веденні обліку, зменшити витрати часу на рутинні операції, а також підвищити якість обслуговування клієнтів завдяки швидкому доступу до актуальної інформації.

У роботі також проаналізовано можливості масштабування рішення, його інтеграції з іншими сервісами Google та зовнішніми інструментами. Наведені приклади інтерфейсів, таблиці з базовою структурою даних, результати тестування додатку та рекомендації щодо його впровадження у реальних умовах.

ABSTRACT

The master's thesis includes: 66 pages, 34 figures, 6 tables, 38 sources.

Keywords: CRM, apartment rental, Google AppSheet, mobile application, database, cloud services, automation, digital transformation, user interface, real estate management.

This thesis examines the process of designing and implementing a CRM system (Customer Relationship Management system) for automating apartment rental processes. Google AppSheet was chosen as the main platform for this solution—a cloud-based no-code service that allows the creation of fully functional mobile and web applications based on Google Sheets or other data sources.

The aim of this work is to simplify the management of apartment rentals, particularly the tracking of clients, contracts, rental periods, payments, maintenance, and communication between landlords and tenants. The project includes an analysis of modern approaches to the digitalization of real estate management, the development of a database structure, the creation of user-friendly interfaces, and the implementation of automated notifications, filters, and forms for data input and editing.

The developed CRM system significantly improves customer management efficiency, minimizes the human factor in recordkeeping, reduces time spent on routine operations, and enhances customer service quality by providing quick access to up-to-date information.

The thesis also analyzes the scalability of the solution and its integration with other Google services and external tools. Examples of user interfaces, basic data structure tables, application testing results, and recommendations for real-world implementation are provided.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ СТАНУ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	10
1.1 Аналіз сучасних CRM систем у сфері оренди нерухомості	10
1.2 Виявлення невирішених задач і протиріч в існуючих рішеннях	14
1.3 Постановка науково-технічної задачі та визначення цілей дослідження .	16
1.4 Висновки до 1 розділу	18
РОЗДІЛ 2 ОБҐРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТІВ І МЕТОДІВ ДОСЛІДЖЕННЯ	19
2.1 Порівняльний аналіз платформ для створення CRM систем	19
2.2 Обґрунтування вибору Google AppSheet як інструмента реалізації.....	24
2.3 Архітектура low-code/ no-code систем та їх роль у сучасній розробці.....	26
2.4 Методика розробки CRM-системи на базі Google AppSheet.....	28
2.5 Висновки до 2 розділу	31
РОЗДІЛ 3 РОЗРОБКА CRM-СИСТЕМИ ДЛЯ ОРЕНДИ КВАРТИР.....	33
3.1 Проєктування бази даних системи в Google Sheets.....	33
3.2 Створення додатку в Google AppSheet і налаштування основного функціоналу	38
3.3 Тестування функціоналу додатку	47
3.4 Висновки до 3 розділу	52
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОСНОВИ ОХОРОНИ ПРАЦІ.....	53
4.1 Інженерно-психологічні принципи професійного добору.....	53
4.2 Заходи, що покращують умови праці оператора	54
ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
ДОДАТКИ.....	65

ПЕРЕЛІК ПОЗНАЧЕНЬ

CRM – Customer Relationship Management.

UI – User Interface.

DB – Database.

ВСТУП

У сучасних умовах цифрової трансформації бізнес-процесів особливого значення набуває ефективне управління інформаційними потоками, зокрема в сфері оренди нерухомості. Оперативний доступ до даних про об'єкти, клієнтів, фінансові операції та стан укладених угод є необхідною умовою стабільної роботи компаній, що надають послуги з оренди квартир.

Традиційні методи ведення обліку за допомогою паперових журналів, або розрізнених електронних таблиць уже не задовольняють потреб сучасного ринку, оскільки не дозволяють своєчасно оновлювати інформацію, контролювати угоди в режимі реального часу та забезпечувати якісний сервіс для клієнтів. У зв'язку з цим, усе більшої популярності набувають CRM-системи (Customer Relationship Management), які автоматизують управління відносинами з клієнтами, об'єктами та орендними процесами.

Особливої актуальності набуває використання low-code та no-code платформ, що дозволяють створювати гнучкі та функціональні CRM-рішення без необхідності глибоких знань у програмуванні. Однією з найпоширеніших і доступних платформ такого типу є Google AppSheet, яка забезпечує швидке створення додатків на основі Google Sheets та інших сервісів, з можливістю налаштування бізнес-логіки, створення форм, звітів і зручного доступу до додатка з будь-яких пристроїв. Розробка CRM-системи для обліку орендованих квартир із використанням Google AppSheet дозволяє вирішити актуальні завдання галузі, такі як централізація даних, автоматизація обліку угод та контроль фінансових розрахунків.

Вибір теми дипломної роботи пов'язаний з актуальністю впровадження сучасних інформаційних технологій у сферу оренди житлової нерухомості та потребою в доступних рішеннях для невеликих агентств та приватних орендодавців. Розробка такої CRM-системи дозволяє не лише покращити якість управління, а й скоротити витрати на впровадження складного програмного

забезпечення, використовуючи можливості low-code технологій. Робота виконується у межах напряму наукових досліджень кафедри програмної інженерії, пов'язаних із розробкою та впровадженням інформаційних систем для автоматизації управлінських процесів.

Метою даної дипломної роботи є розробка CRM-системи для обліку орендованих квартир на базі платформи Google AppSheet, що забезпечить автоматизацію обліку об'єктів нерухомості, клієнтів, орендних угод, платежів та комунікацій між сторонами. Для досягнення цієї мети необхідно виконати низку завдань: проаналізувати сучасні CRM-системи у сфері оренди нерухомості, визначити їх переваги та недоліки, обґрунтувати доцільність використання low-code платформ, здійснити проєктування структури даних системи, створити додаток у Google AppSheet та протестувати його функціональні можливості.

Об'єктом дослідження є процес управління відносинами з клієнтами в агентствах оренди квартир, а предметом – методи та засоби розробки CRM-системи для обліку орендованих квартир на основі Google AppSheet. У ході роботи використано методи аналізу сучасних CRM-рішень, порівняльного аналізу інструментів low-code/ no-code розробки, методику проєктування баз даних у Google Sheets, а також метод експериментальної розробки та тестування програмного забезпечення на платформі Google AppSheet.

Наукова новизна роботи полягає в адаптації можливостей платформи Google AppSheet для створення CRM-системи у сфері оренди квартир, що дозволяє централізовано вести облік об'єктів нерухомості, клієнтів і орендних угод без використання традиційних складних програмних продуктів. Запропоноване рішення відрізняється від відомих аналогів спрощеною інтеграцією з хмарними сервісами, гнучкістю налаштувань і доступністю для малих підприємств та приватних орендодавців.

Практичне значення одержаних результатів полягає в можливості застосування розробленої CRM-системи в реальній діяльності агентств нерухомості для автоматизації обліку орендних об'єктів, клієнтів і угод.

РОЗДІЛ 1 АНАЛІЗ СТАНУ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Аналіз сучасних CRM систем у сфері оренди нерухомості

У сучасних умовах ринку нерухомості ефективне управління відносинами з клієнтами та об'єктами оренди неможливе без впровадження спеціалізованих CRM-систем [1]. Такі системи дозволяють автоматизувати основні бізнес-процеси агентств нерухомості, забезпечуючи централізований облік інформації про клієнтів, об'єкти нерухомості, угоди, фінансові операції та комунікації.

Завдяки використанню CRM-систем підвищується якість обслуговування клієнтів, зростає швидкість обробки заявок та зменшується кількість помилок, пов'язаних із людським фактором [2]. На сучасному ринку представлено чимало рішень, які мають різну функціональність, ціну та технічні особливості, тому для обґрунтування вибору платформи доцільно провести аналіз існуючих CRM-систем, що застосовуються у сфері оренди нерухомості [3].

Pipedrive – це популярна європейська CRM-система, яка широко застосовується малим та середнім бізнесом, у тому числі в нерухомості. Головна особливість Pipedrive – це простота налаштування і гнучкість, а також візуалізація процесів у вигляді воронки продажів чи угод. CRM дозволяє створювати необмежену кількість воронки для різних типів угод, включно з орендою (рис. 1.1).

Інтегрується з поштовими сервісами, календарями, сервісами для обробки документів і телефонії. Є можливість додавати власні поля для об'єктів оренди, фіксувати терміни договорів, платежі та нагадування. Проте, специфічних модулів для оренди (як от облік договорів чи історія оплат) немає – їх можна реалізувати через налаштування власних полів [4].

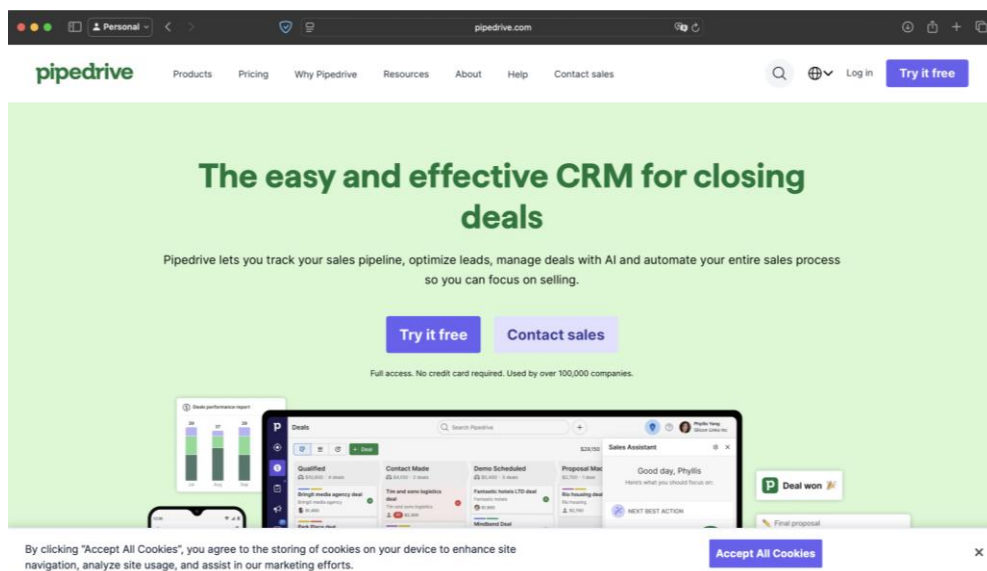


Рисунок 1.1 – Pipedrive CRM

Rentila – це спеціалізована CRM-система для обліку оренди нерухомості, яка підходить для використання як приватними орендодавцями, так і малими агентствами. Rentila дозволяє здійснювати облік орендарів, платежів, договорів та витрат, а також автоматизувати нагадування про терміни платежів і виконання зобов’язань. Платформа надає можливість легко налаштувати базу даних оренди без потреби у глибоких технічних знаннях.

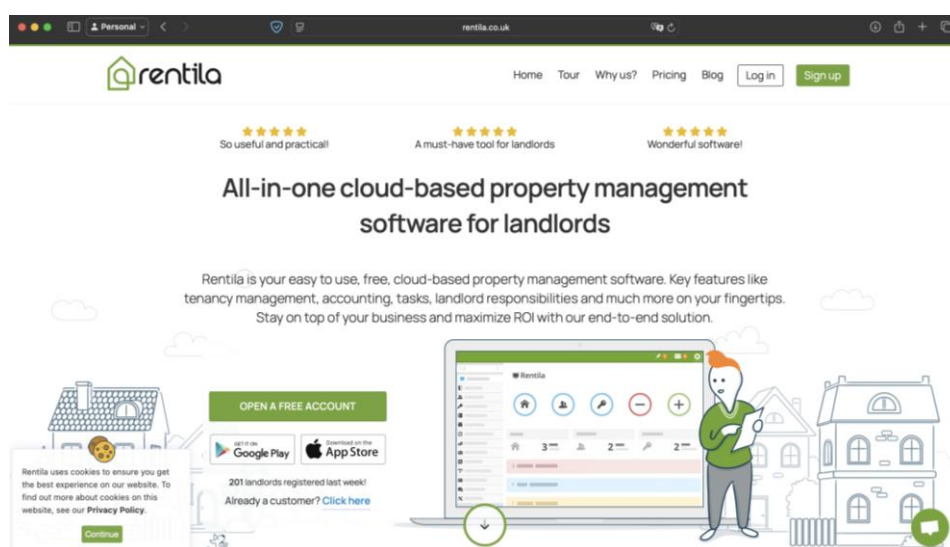


Рисунок 1.2 – Rentila CRM

Основним недоліком є обмежена функціональність щодо управління

великими обсягами даних, а також відсутність можливостей для інтеграції з іншими системами, такими як телефонія чи месенджери. Це робить Rentila менш гнучкою в порівнянні з більш потужними рішеннями для агентств з великими обсягами клієнтів та об'єктів нерухомості (рис. 1.2).

HubSpot CRM є ще однією потужною платформою для управління відносинами з клієнтами, яка може бути використана на ринку оренди нерухомості (рис. 1.3). Вона надає безкоштовний базовий пакет, що дозволяє малим агентствам нерухомості швидко почати використовувати систему без великих витрат. HubSpot CRM має зручний інтерфейс, підтримує автоматизацію процесів, зберігає історію взаємодії з клієнтами і дозволяє інтегруватися з іншими системами, такими як електронна пошта, телефонія та соціальні мережі [5].

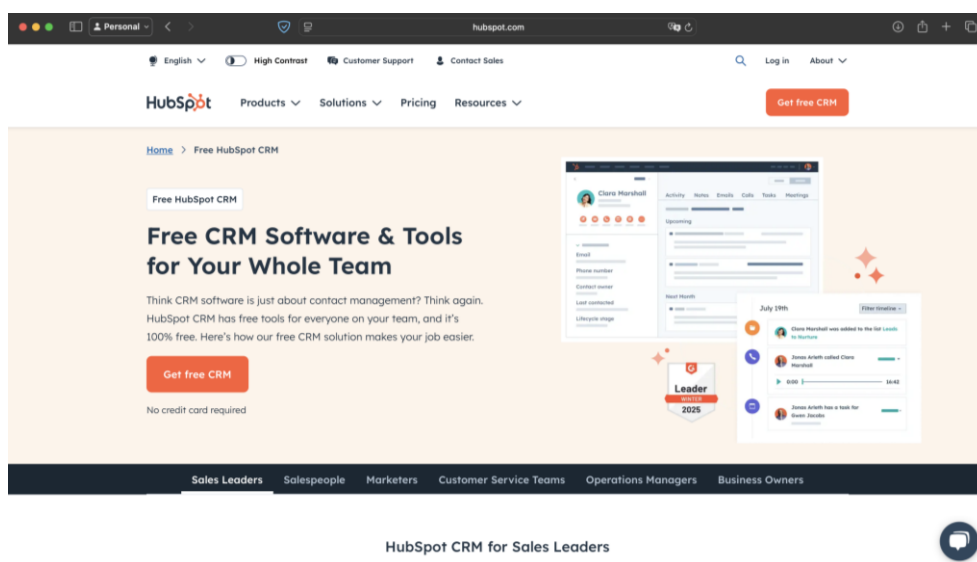


Рисунок 1.3 – HubSpot CRM

Одним з ключових переваг є масштабованість, що дозволяє компаніям легко розширювати функціональність за допомогою додаткових модулів і платних планів. Проте, відсутність спеціалізованих функцій для оренди нерухомості, таких як облік договорів і платежів, робить цю платформу менш привабливою для агентств, що спеціалізуються виключно на оренді.

Для зручності порівняння основних характеристик цих CRM-систем можна побудувати наступну таблицю (табл. 1.1).

Таблиця 1.1 – Порівняння обраних CRM

Характеристика	Pipedrive	Rentila	HubSpot CRM
Цільова аудиторія	Малі та середні агентства	Приватні орендодавці, малі агентства	Малі та середні бізнеси
Основні функції	Воронки угод, інтеграції, нагадування	Облік об'єктів, договорів, оплат, нагадування	Управління контактами, угодами, комунікаціями
Наявність модулів для оренди	Немає (можна налаштувати вручну)	Є (повний облік оренди)	Немає (можна адаптувати)
Інтеграції з іншими сервісами	Пошта, календарі, телефонія, документи	Мінімальні	Широкі (ел. пошта, месенджери, соцмережі)
Інтерфейс	Інтуїтивний, зрозумілий, візуалізація воронок	Простий, орієнтований на оренду	Інтуїтивно зрозумілий
Автоматизація процесів	Середня (можна налаштувати автоматизації)	Мінімальна	Висока (workflow, шаблони листів)
Ціна	Від \$14,90/місяць	Від €8/місяць	Безкоштовний базовий, платні розширення

З цього порівняння видно, що кожна з цих платформ має свої переваги та недоліки в залежності від потреб агентства нерухомості. amoCRM є потужним інструментом для великих агентств з комплексними вимогами, але має високу вартість. Rentila є спеціалізованою системою, що підходить для малих агентств і орендодавців, однак її функціональність є обмеженою. HubSpot CRM може бути хорошим вибором для малих і середніх агентств, які мають потребу в безкоштовному рішенні з можливістю масштабування, хоча вона не спеціалізується на оренді нерухомості [6].

1.2 Виявлення невирішених задач і протиріч в існуючих рішеннях

Аналіз сучасних CRM-систем, що використовуються у сфері оренди нерухомості, дозволяє виявити низку невирішених задач і суперечностей, які стримують ефективність їх застосування в умовах динамічного ринку. Попри наявність широкого функціоналу, більшість існуючих рішень не враховують специфіку саме орендного бізнесу, орієнтуючись переважно на продажі або загальне управління клієнтськими базами. У таких системах, як Pipedrive, HubSpot та Rentila, відсутні повноцінні модулі для ведення обліку об'єктів нерухомості, що унеможливує централізоване управління квартирами, їх станом, історією оренди та фінансовими операціями в межах єдиної системи без використання зовнішніх інструментів.

Ще однією проблемою є обмежені можливості інтеграції з сервісами для роботи з договорами та юридичною документацією. Більшість CRM дозволяють прикріплювати файли до карток клієнтів або угод, однак не мають вбудованих засобів для автоматичного створення, підписання та архівування договорів оренди, що особливо актуально за умов дистанційної співпраці з орендарями. Це змушує агентства нерухомості та приватних орендодавців використовувати сторонні сервіси для документообігу, що ускладнює ведення справ і підвищує ризики

помилки [7].

Окремою проблемою залишається обмежена автоматизація ключових бізнес-процесів. Такі операції, як автоматичне нарахування орендної плати, надсилання нагадувань про оплату, повідомлення про закінчення терміну дії договору, фіксація заборгованості та автоматичне формування звітності, часто реалізуються частково або взагалі не підтримуються. Це змушує користувачів здійснювати контроль вручну або розробляти складні бізнес-правила самостійно, що потребує додаткових ресурсів і технічних знань.

Також важливо відзначити проблему обмеженої адаптації до локальних ринків та правових вимог. Більшість популярних CRM створені з урахуванням західних стандартів і не враховують особливостей договорів, податкових норм і форм звітності, що діють в Україні. Це створює додаткові труднощі у веденні обліку й формуванні внутрішньої документації, змушуючи адаптувати або модифікувати готові рішення.

Ще одним суперечливим моментом є відсутність зручних мобільних додатків або їх обмежений функціонал у частині управління орендою. Багато CRM орієнтовані переважно на десктоп-версії, в той час як ринок оренди передбачає постійну мобільність користувачів, оперативну роботу з заявками, переглядом об'єктів, узгодженням умов і спілкуванням з клієнтами. Обмежений мобільний функціонал стримує ефективну роботу агентів на виїзді та поза офісом [8].

У підсумку, аналіз існуючих CRM-систем для оренди нерухомості виявив значний попит на інструменти, які дозволяли б комплексно вирішувати специфічні задачі орендного бізнесу, забезпечувати автоматизацію рутинних процесів, інтеграцію з сервісами документообігу та адаптацію до локальних умов. Водночас сучасні рішення залишають низку невирішених задач, що створює передумови для розробки нових спеціалізованих рішень у цій галузі.

1.3 Постановка науково-технічної задачі та визначення цілей дослідження

У сучасних умовах ринку нерухомості зростає потреба в цифрових рішеннях, які дозволяють систематизувати інформацію про об'єкти оренди, орендарів, фінансові операції, договори та технічний стан приміщень. Складність управління великою кількістю об'єктів, контролю укладених договорів та своєчасності платежів створює необхідність у впровадженні спеціалізованих CRM-систем, які б автоматизували ці процеси, підвищували прозорість роботи та мінімізували людський фактор.

На основі проведеного аналізу сучасних CRM-систем у сфері оренди нерухомості було виявлено ряд обмежень і недоліків, що свідчать про потребу у створенні гнучкої, доступної та адаптивної системи для комплексного управління даними про орендовані об'єкти. Це зумовлює актуальність розробки власного рішення на основі інструментів швидкої розробки додатків, яке забезпечить персоналізацію та оптимізацію функціоналу під конкретні потреби користувача [9].

Діаграма варіантів використання зображена на рисунку 1.4.

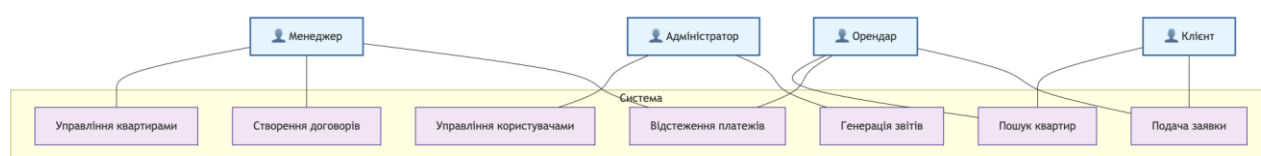


Рисунок 1.4 – Діаграма варіантів використання

У зв'язку з цим науково-технічна задача, яку необхідно вирішити в межах даної роботи, полягає у проєктуванні та створенні CRM-системи для обліку і управління процесами оренди житлових і комерційних приміщень з можливістю централізованого ведення об'єктів, реєстру клієнтів, контролю договорів, відстеження фінансових операцій та організації супутніх сервісних задач.

Для досягнення поставленої мети визначено такі основні цілі:

- Формування сторінки доступних приміщень, де відображатиметься список усіх вільних для оренди одиниць нерухомості. Приміщення повинні бути згруповані за об'єктами (будинками, житловими комплексами, бізнес-центрами тощо). Кожна одиниця повинна мати можливість відкривати детальну інформацію про себе та створювати новий запис про заселення.

- Реалізація сторінок для реєстрації заселення та виселення орендарів. Кожна така сторінка повинна містити чек-лист із переліком запитань щодо стану приміщення на момент заселення або виселення. Відповіді на ці запитання мають бути представлені у вигляді перелічувальних варіантів, щоб стандартизувати процес фіксації стану об'єкта. Важливо передбачити блок підпису, після внесення якого редагування чек-листа має бути заблоковано для запобігання змінам заднім числом.

- Розділ для управління об'єктами. Користувачі повинні мати можливість переглядати список усіх наявних об'єктів нерухомості та перелік приміщень у кожному об'єкті. Для цього необхідно реалізувати фільтрацію приміщень за назвою або ідентифікатором об'єкта.

- Організація структури даних таким чином, щоб усі чек-листи заселення/виселення мали посилання на відповідне приміщення, а також зберігали відповіді на контрольні запитання та дату внесення даних.

- Обмеження на редагування заповнених чек-листів після підпису відповідальної особи, що дозволить забезпечити цілісність даних та виключити можливість їх подальшого коригування без відповідного підтвердження.

- Наявність зрозумілого і доступного користувацького інтерфейсу, який дозволить швидко перемикатися між списками об'єктів, приміщень, доступних для заселення одиниць, а також відкривати форму для створення записів про заселення та виселення.

Таким чином, науково-технічна задача цієї роботи полягає у створенні сучасної CRM-системи для автоматизації обліку оренди нерухомості з повноцінним функціоналом, орієнтованим на централізоване управління інформацією, підвищення якості обслуговування клієнтів та оптимізацію бізнес-

процесів у сфері оренди житла й комерційних приміщень [10].

1.4 Висновки до 1 розділу

У першому розділі було проведено аналіз сучасного стану використання CRM-систем у сфері оренди нерухомості. Встановлено, що попит на автоматизовані рішення для управління орендою квартир зростає, проте більшість існуючих CRM-рішень орієнтовані на великі агентства та мають обмежену гнучкість для адаптації під конкретні потреби малого або середнього бізнесу.

У ході аналізу було виявлено низку невирішених задач, зокрема недостатню інтеграцію з мобільними платформами, складність налаштувань без спеціальних технічних знань, а також обмежені можливості кастомізації інтерфейсу для кінцевих користувачів.

У результаті було сформульовано науково-технічну задачу дослідження – розробити адаптивну, доступну та функціональну CRM-систему для оренди квартир на базі сучасних low-code/no-code платформ, зокрема Google AppSheet. Основною метою дослідження є створення прототипу системи, що задовольняє потреби користувачів у зручному, ефективному та безпечному управлінні процесами оренди нерухомості.

РОЗДІЛ 2 ОБҐРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТІВ І МЕТОДІВ ДОСЛІДЖЕННЯ

2.1 Порівняльний аналіз платформ для створення CRM систем

На сучасному ринку представлено велику кількість платформ, які дозволяють створювати CRM-системи з мінімальним залученням програмування. Найбільш популярними серед low-code та no-code рішень є Zoho Creator, Airtable та Google AppSheet. Кожна з цих платформ має свої особливості, переваги та обмеження, які слід враховувати під час вибору інструменту для реалізації CRM-системи у сфері оренди квартир.

Zoho Creator – це сучасна low-code платформа, яка дозволяє створювати бізнес-додатки, CRM-системи, системи обліку, автоматизовані сервіси та інші типи додатків без повноцінного програмування. Основною особливістю Zoho Creator є наявність власної мови сценаріїв Deluge, за допомогою якої користувач може налаштовувати автоматизації, перевірки даних і складні бізнес-правила (рис. 2.1). Платформа дозволяє створювати взаємопов'язані таблиці, форми для введення даних, звіти, графіки та інтегрувати систему з іншими сервісами, зокрема з екосистемою Zoho, Google Workspace, Zapier та сторонніми API [11].

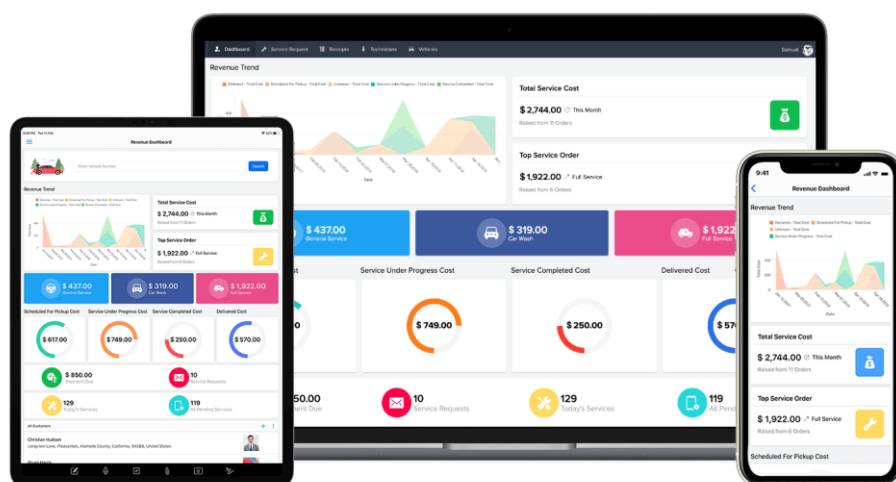


Рисунок 2.1 – Zoho Creator

Zoho Creator має зручний мобільний додаток для Android і iOS, що дозволяє користувачам працювати з CRM із будь-якого пристрою. До переваг цієї платформи відносять високу гнучкість структури додатків, розширені можливості автоматизації та підтримку інтеграцій, що дає змогу масштабувати систему за потреби. Недоліком є те, що для налаштування більш складних процесів і бізнес-логіки потрібно володіти базовими технічними знаннями та освоїти синтаксис Deluge. Крім того, попри наявність безкоштовного тарифу, повноцінна робота з більшістю функцій вимагає підписки від 10 доларів на місяць за користувача [12].

Airtable – це популярна no-code платформа, яка поєднує функціонал електронної таблиці та бази даних. Вона дозволяє створювати CRM-системи, системи обліку клієнтів, планування, управління проектами та інші бізнес-додатки без потреби в програмуванні. Airtable вирізняється простим і зрозумілим інтерфейсом, завдяки чому навіть користувачі без технічного досвіду можуть швидко налаштовувати таблиці, створювати взаємозв'язки між ними, формули та фільтри (рис. 2.2).

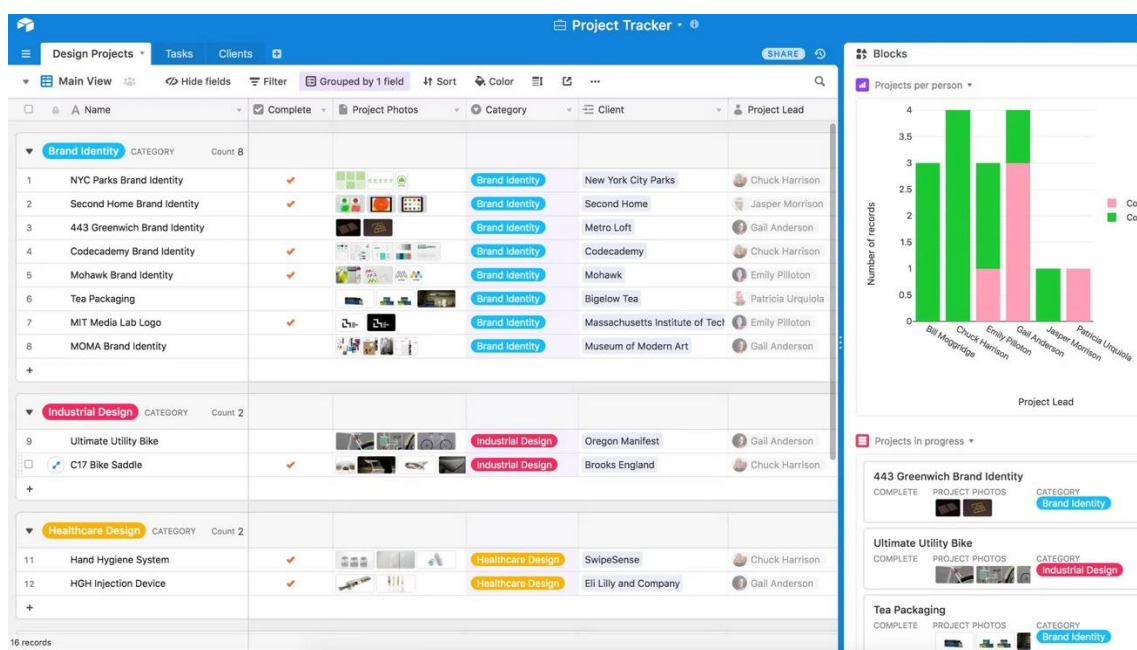


Рисунок 2.2 – Airtable

Платформа підтримує різні варіанти представлення даних: табличний, канбан-дошку, календар, галерею та інтерактивні форми. До її переваг належить

простота використання, доступність готових шаблонів та інтеграцій з популярними сервісами через Zapier, Make та API. Крім того, Airtable дозволяє швидко налаштовувати автоматизації та сповіщення на основі подій у таблицях. До недоліків платформи можна віднести певні обмеження в складності налаштування зв'язків між таблицями, а також обмежений функціонал безкоштовної версії. Для повноцінної роботи з великим обсягом даних та розширеними функціями автоматизації необхідно обрати платний тариф, що починається від 12 доларів на місяць за користувача [13].

Google AppSheet – це сучасна no-code платформа, яка дозволяє створювати повнофункціональні мобільні та веб-додатки без написання коду, використовуючи як основу таблиці Google Sheets, Excel, SQL-бази даних, Salesforce, Smartsheet та інші хмарні сервіси (рис. 2.3). Вона забезпечує гнучке налаштування бізнес-логіки, створення інтерактивних форм, дашбордів, списків, фільтрів і звітів, а також дозволяє задавати умови доступу для різних користувачів. Однією з головних переваг є автоматична генерація додатків за структурою таблиць, що значно пришвидшує розробку, а також інтеграція з екосистемою Google, що дозволяє легко зв'язувати додаток з Google Calendar, Google Drive, Google Maps і Gmail.

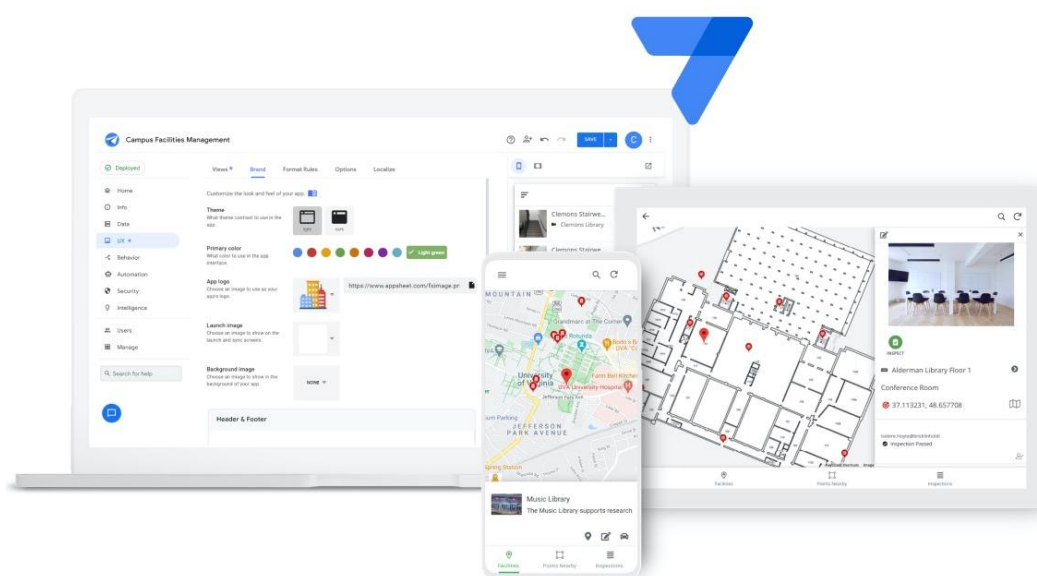


Рисунок 2.3 – Google AppSheet

Крім того, AppSheet підтримує офлайн-режим, що є цінним для користувачів, які працюють у польових умовах. Серед переваг також можна виділити наявність готових шаблонів для CRM, логістики, інвентаризації, трекінгу замовлень і технічного обслуговування, які можна адаптувати під власні потреби. До недоліків відносяться обмеження в продуктивності для великих обсягів даних (рекомендовано до 100 тис. рядків у Google Sheets для стабільної роботи), досить обмежена гнучкість для реалізації складних алгоритмів та бізнес-правил порівняно з класичними кодовими платформами, а також наявність платної ліцензії для використання додатків у комерційному середовищі чи при необхідності керувати правами доступу користувачів. Незважаючи на це, AppSheet залишається одним із найпопулярніших рішень для малого та середнього бізнесу завдяки простоті, швидкості запуску та широким можливостям інтеграції без потреби залучати розробників [14].

Можемо тепер цю інформацію узагальнити в порівняльній таблиці (табл. 2.2)

Таблиця 2.1 – Порівняння платформ для створення CRM

Характеристика	Google AppSheet	Zoho Creator	Airtable
Тип платформи	No-code	Low-code / No-code	No-code
База даних	Google Sheets, Excel, SQL, Cloud SQL	Власна база, інтеграції з іншими СУБД	Власна база, підтримка CSV, Excel
Інтерфейс	Автоматично генерується на основі даних	Гнучкі форми, звіти, панелі	Таблиці, форми, галереї, календарі

Продовження таблиці 2.1

Автоматизація	Боти, тригери, інтеграція з Google Workspace	Автоматизація через вебхуки, скрипти	Автоматизація через Airtable Automations
Мобільна версія	Автоматично генеруються мобільні додатки	Мобільні додатки для Android та iOS	Мобільний додаток з обмеженнями
Готові шаблони	Безліч галузевих шаблонів	Шаблони для бізнесу, CRM, сервісу	Шаблони для бізнесу, проектного менеджменту
Безпека та доступ	Ролі, доступ через Google акаунти	Політики доступу, шифрування	Обмежені налаштування доступу
Інтеграції	Google Workspace, Salesforce, SQL	Zoho CRM, Google Workspace, Zapier	Slack, Google Workspace, Zapier
Складність освоєння	Дуже проста	Потрібні знання базових логік	Дуже проста для початківців
Ціни (від)	Безкоштовний план, від \$5/користувача	Від \$8/користувача	Безкоштовний план, від \$12/користувача
Основні переваги	Інтеграція з Google, автоматична генерація	Гнучкість, розвинена автоматизація	Простота, зручні інтеграції

Продовження таблиці 2.1

Основні недоліки	Обмежена кастомізація інтерфейсу	Вища ціна для великих проектів	Обмежені автоматизації, слабкі доступи
---------------------	--	--------------------------------------	--

У підсумку, всі три платформи – Google AppSheet, Zoho Creator та Airtable – пропонують безкодову розробку CRM-систем для управління орендою нерухомості. Google AppSheet є ідеальним вибором для користувачів, які шукають швидке створення додатків з інтеграцією з Google продуктами та мінімальними технічними вимогами. Zoho Creator підійде для середніх і великих бізнесів, які потребують потужних можливостей для автоматизації та інтеграції, але готові інвестувати в навчання і налаштування платформи. Airtable – це зручний інструмент для невеликих команд, орієнтованих на простоту використання і швидке впровадження, однак з деякими обмеженнями у функціональності та кастомізації порівняно з іншими платформами [15].

2.2 Обґрунтування вибору Google AppSheet як інструмента реалізації

Вибір Google AppSheet як інструмента для реалізації CRM-системи для оренди квартир обумовлений рядом факторів, які роблять цю платформу ідеальним рішенням для розробки безкодових додатків, зокрема в сфері управління нерухомістю.

По-перше, Google AppSheet дозволяє швидко створювати функціональні додатки без необхідності програмування. Це особливо важливо для організацій, де немає можливості залучити розробників або де потрібен швидкий запуск продукту. Платформа підтримує інтеграцію з Google Sheets, що є популярним інструментом для зберігання даних і дозволяє безперешкодно працювати з базами даних у реальному часі. Для CRM-системи оренди квартир це дозволяє створити

централізовану базу даних для всіх об'єктів нерухомості, що легко оновлюється та доступна для всіх користувачів [16].

Крім того, Google AppSheet має просту візуальну настройку, що дозволяє налаштовувати форму введення даних, створювати списки, відображати інформацію та автоматизувати робочі процеси без необхідності писати код. Це забезпечує простоту використання і значно знижує витрати на розробку та підтримку системи, оскільки користувачі можуть самостійно налаштовувати додаток згідно з потребами бізнесу.

Платформа також підтримує різноманітні автоматизації, такі як автоматичні сповіщення, оновлення статусів, виконання перевірок даних на основі заданих правил. У контексті оренди квартир, це дозволяє автоматизувати процеси, такі як надання повідомлень орендарям про терміни закінчення оренди або про необхідність оновлення договору. Крім того, можливість інтеграції з іншими інструментами Google, такими як Google Calendar або Google Drive, дає змогу ефективно організовувати робочий процес і забезпечувати збереження документації, що супроводжує орендні угоди [17].

Ще однією перевагою є можливість мобільного доступу до системи через додаток, що особливо важливо для агентів з оренди нерухомості, які часто перебувають на виїзді. Мобільна версія додатку дозволяє управляти даними і виконувати необхідні дії з будь-якого місця, що підвищує ефективність роботи.

Не менш важливим аспектом є безпека. Оскільки Google AppSheet працює на основі інфраструктури Google, користувачі отримують доступ до високого рівня безпеки даних, зокрема через аутентифікацію Google та можливість використання додаткових заходів безпеки, таких як двофакторна аутентифікація.

Таким чином, вибір Google AppSheet для реалізації CRM-системи для оренди квартир зумовлений його зручністю, інтеграцією з популярними інструментами, можливістю автоматизації процесів, низькою вартістю впровадження і обслуговування, а також забезпеченням мобільності та безпеки для користувачів [18].

2.3 Архітектура low-code/ no-code систем та їх роль у сучасній розробці

Архітектура low-code та no-code систем відображає новий підхід до розробки програмного забезпечення, який дозволяє значно спростити процес створення додатків. Основною ідеєю таких платформ є можливість створення функціональних додатків без необхідності глибоких знань програмування. Користувачі можуть створювати складні бізнес-рішення, перетягуючи елементи інтерфейсу, налаштовуючи логіку за допомогою візуальних інструментів і використовувати вбудовані шаблони.

Low-code платформи надають більше свободи в налаштуваннях і дозволяють писати код, якщо потрібно реалізувати специфічні функції або виконати складні обчислення. Вони забезпечують основну основу для швидкої розробки додатків, але також пропонують можливість гнучкої кастомізації через написання коду для конкретних бізнес-вимог. Зазвичай, для роботи з такими платформами потрібен певний рівень технічних знань [19].

No-code системи, на відміну від low-code, зовсім не вимагають знання програмування. Вони призначені для людей без технічного досвіду, дозволяючи їм створювати додатки за допомогою простих інтерфейсів, таких як “перетягування” елементів, встановлення їхніх властивостей і створення робочих процесів без написання жодного рядка коду. Такі системи зазвичай орієнтовані на простоту і зручність, і їх часто використовують для створення інтерфейсів користувача, управлінських панелей, форм введення даних та інших рішень, які не вимагають складної логіки.

Основною архітектурною характеристикою цих платформ є їхня здатність об’єднувати кілька важливих компонентів у єдину систему: інтерфейс користувача, базу даних, серверні функції та бізнес-логіку. Такі платформи зазвичай мають інтеграцію з популярними базами даних та зовнішніми сервісами через API, що дозволяє користувачам об’єднувати їх із зовнішніми системами для покращення функціональності [20].

Крім того, сучасні low-code/no-code платформи надають інструменти для автоматизації робочих процесів, що дозволяє значно підвищити ефективність бізнес-процесів. Автоматизація включає в себе створення тригерів, які активують певні дії (наприклад, відправлення електронних листів, оновлення даних у базі, або інтеграція з іншими системами). Це дозволяє автоматизувати рутинні завдання, які раніше потребували участі людини, та значно зменшити кількість помилок, зберігаючи час та ресурси.

Найважливіше, що архітектура low-code/no-code платформ дозволяє бізнес-орієнтованим користувачам (наприклад, бізнес-аналітикам, менеджерам та маркетингологам) активно залучатися до процесу розробки додатків без потреби в програмуванні. Це дозволяє значно скоротити час розробки, адже бізнес-спеціалісти можуть самостійно створювати прототипи та рішення, що точно відповідають їх вимогам, не очікуючи на програмістів [21].

Ще однією важливою складовою є інтеграції з іншими інструментами, що дозволяє low-code/no-code платформам ефективно працювати в екосистемах, де вже використовуються інші програмні рішення. Це можуть бути платформи для аналітики, CRM-системи, платформи для керування проектами або навіть інструменти для роботи з великими даними. Більшість таких платформ підтримують вбудовані API та інші механізми інтеграції, що робить їх дуже гнучкими в роботі з різними джерелами даних.

На практиці використання low-code/no-code платформ дає можливість знижувати витрати на розробку та підтримку додатків, зменшувати час, необхідний для створення нових функціональностей, та дозволяє швидше реагувати на зміни на ринку або в вимогах користувачів. Такі платформи сприяють більш швидкому тестуванню нових ідей і відразу втілювати їх у життя, що є особливо важливим у динамічному світі бізнесу [22].

У результаті, архітектура таких систем дозволяє зберегти баланс між функціональністю та доступністю. Це дозволяє підприємствам швидко створювати та адаптувати додатки під свої потреби, що є важливою перевагою в умовах швидко змінюваного бізнес-середовища.

2.4 Методика розробки CRM-системи на базі Google AppSheet

Методика розробки CRM-системи на базі Google AppSheet включає кілька ключових етапів, які охоплюють від початкового планування та проєктування до створення кінцевого додатку. Завдяки візуальним інструментам і зручному інтерфейсу, цей процес значно спрощується, але для ефективної реалізації необхідно слідувати певним крокам. Далі розглянемо основні етапи розробки CRM-системи на платформі Google AppSheet [23].

Перед початком створення додатку важливо визначити структуру даних, які будуть зберігатися в CRM-системі. Google AppSheet використовує Google Sheets або інші сховища даних, такі як Excel, для зберігання інформації. Основним кроком є визначення таблиць, які необхідні для CRM-системи, а також створення відповідних полів для кожної таблиці. Наприклад, для системи оренди нерухомості це можуть бути таблиці для зберігання даних про об'єкти нерухомості, клієнтів, договори оренди, історію платежів і так далі.

На цьому етапі важливо також визначити зв'язки між таблицями, наприклад, один до багатьох або багато до багатьох, а також механізми фільтрації та сортування даних. Структура бази даних має бути такою, щоб забезпечити ефективну роботу додатку та зручний доступ до даних [24].

Після визначення структури бази даних, створюється Google Sheets документ або інша таблиця даних. Важливо, щоб усі необхідні таблиці були чітко організовані і мали зрозумілі назви та поля, що відповідають вимогам CRM-системи. Google AppSheet автоматично синхронізує дані з таблицями, і важливо, щоб дані були правильно структуровані для коректної роботи додатку.

Цей етап також включає налаштування типів даних для кожного поля: текст, числа, дата, посилання на інші таблиці тощо. Наприклад, для поля “Дата початку оренди” повинна бути встановлена дата як тип даних, а для “Сума платежу” – числовий формат.

Далі відбувається безпосереднє створення додатку в Google AppSheet.

Платформа дозволяє автоматично створювати додатки на основі вже налаштованих таблиць даних. Google AppSheet сканує таблиці та пропонує базові варіанти для створення додатку, такі як форми введення даних, таблиці для перегляду інформації, звіти тощо.

Користувач може налаштовувати вигляд і функціональність додатку, вибираючи серед готових шаблонів або створюючи власний інтерфейс. Платформа також дозволяє налаштувати поведінку додатку – наприклад, додавання нових записів, редагування наявних, відображення інформації, сортування та фільтрація даних [25].

Однією з ключових переваг Google AppSheet є можливість налаштування автоматизації процесів у CRM-системі. Це включає створення різних робочих процесів, тригерів та дій. Наприклад, можна налаштувати сповіщення для користувачів про наближення терміну закінчення договору оренди або автоматичне створення завдань для менеджера з оренди.

Google AppSheet дозволяє реалізувати логіку, за допомогою якої дані автоматично оновлюються або відправляються за певними умовами, що значно знижує обсяг рутинної роботи. Наприклад, можна налаштувати автоматичне оновлення статусу об'єкта нерухомості на “зайнято” після того, як орендар підписав договір, або генерувати квитанцію після здійснення платежу.

Діаграма діяльності зображена на рисунку 2.4.

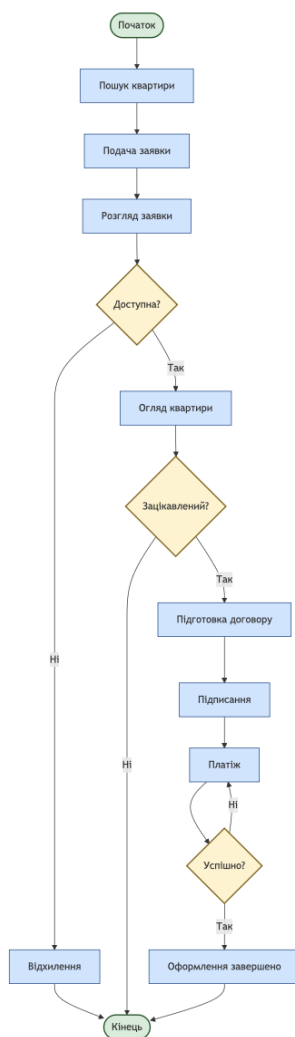


Рисунок 2.4 – Діаграма діяльності

Для будь-якої CRM-системи важливо налаштувати рівні доступу для різних користувачів. У Google AppSheet є вбудовані інструменти для контролю доступу, які дозволяють визначити, які користувачі мають доступ до яких функцій і даних. Наприклад, можна створити ролі для адміністратора, менеджера, орендаря та інших користувачів, обмеживши доступ до певних даних або функцій додатку [26].

Коли додаток створено, необхідно протестувати всі функціональності, переконатися, що дані коректно відображаються і можуть бути змінені або додані без помилок. Тестування включає перевірку всіх форм, таблиць, сповіщень та автоматизованих процесів. Після тестування можна налаштувати інтерфейс користувача, зробити його більш зручним і приємним для користування, додати логотип, вибрати кольори та шрифти, щоб додаток відповідав корпоративному

стилю.

Після того, як додаток протестовано, і всі функції працюють коректно, можна запустити його для користувачів. Важливо також налаштувати моніторинг роботи додатку, щоб відстежувати використання, виявляти можливі помилки і проводити необхідні оновлення та виправлення. Google AppSheet надає можливість аналізувати статистику використання додатку, а також отримувати звіти про помилки, що дозволяє ефективно підтримувати систему.

Після запуску додатку часто виникає необхідність в оновленнях або додаванні нових функціональностей на основі відгуків користувачів. Google AppSheet дозволяє швидко вносити зміни в додаток без необхідності його перевстановлення або виведення з експлуатації. Оновлення можна впроваджувати швидко, що дозволяє постійно покращувати CRM-систему [27].

Таким чином, методика розробки CRM-системи на базі Google AppSheet включає чітке планування, налаштування бази даних, створення додатку, реалізацію автоматизації, налаштування прав доступу, тестування та оптимізацію. Завдяки вбудованим інструментам платформи, можна швидко та ефективно створювати CRM-системи для управління орендою нерухомості без необхідності програмування.

2.5 Висновки до 2 розділу

У другому розділі було здійснено порівняльний аналіз сучасних платформ для створення CRM-систем, зокрема таких як Zoho Creator, Airtable, Microsoft Power Apps та Google AppSheet. За результатами аналізу встановлено, що Google AppSheet є оптимальним інструментом для реалізації CRM-системи у сфері оренди квартир, завдяки поєднанню простоти використання, широких можливостей інтеграції з Google-сервісами та підтримки мобільних і веб-платформ.

Також було досліджено архітектуру low-code/no-code систем, що дозволяє

скоротити терміни розробки, зменшити вартість впровадження та залучити до створення додатків користувачів без глибоких знань у програмуванні.

На основі обраного інструменту сформовано методику розробки CRM-системи, яка включає проєктування структури бази даних у Google Sheets, створення логіки додатку в AppSheet та налаштування функціоналу відповідно до вимог користувачів.

Таким чином, у розділі було обґрунтовано доцільність використання Google AppSheet для реалізації проєкту та визначено основні підходи до його розробки.

РОЗДІЛ 3 РОЗРОБКА CRM-СИСТЕМИ ДЛЯ ОРЕНДИ КВАРТИР

3.1 Проєктування бази даних системи в Google Sheets

У межах створення CRM-системи для управління орендою нерухомості на базі Google AppSheet одним із ключових етапів є проєктування структури бази даних. Оскільки AppSheet працює на основі зовнішніх джерел даних, зокрема Google Sheets, необхідно ретельно продумати організацію таблиць, структуру полів, зв'язки між ними та їхню подальшу інтеграцію у мобільний чи веб-застосунок. Правильно спроектована база забезпечує коректну роботу системи, зручний доступ до інформації, можливість фільтрації, сортування та взаємодії між даними [28].

Діаграма станів наведена на рисунку 3.1.

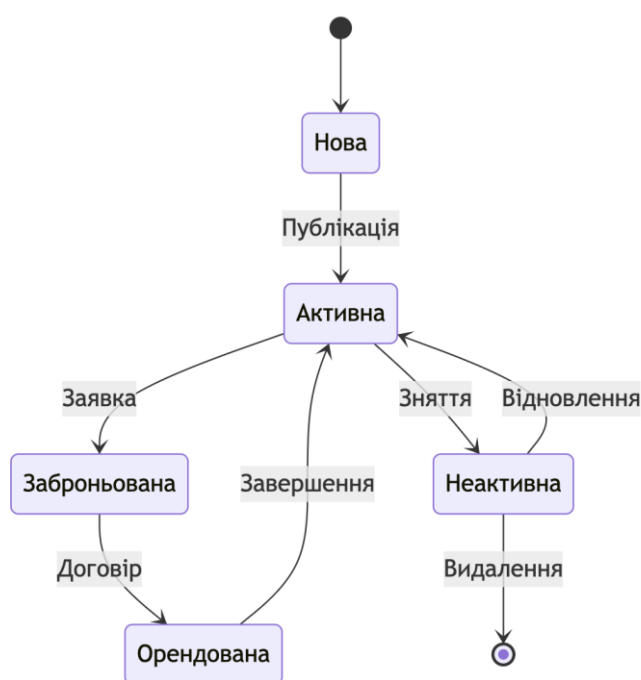


Рисунок 3.1 – Діаграма станів

На першому етапі було створено таблицю property, яка містить перелік

об'єктів нерухомості, доступних для оренди. Ця таблиця є однією з базових, оскільки кожен об'єкт може містити кілька орендних одиниць (квартир, апартаментів, офісів тощо) та пов'язаний із іншими таблицями. У ній зберігається ключова інформація про об'єкти: назва, адреса та зображення, що використовується для візуального представлення в інтерфейсі застосунку.

Структура таблиці property наведена в таблиці 3.1.

Таблиця 3.1 – Структура таблиці property

Property Name	Address	Image
Sunset View	200 Solar Ave WA	Properties_Images/Sunset View.Image.185801.png

Наступним етапом проєктування бази даних стало створення таблиці unit, яка містить детальну інформацію про окремі орендні одиниці (квартири, апартаменти, офісні приміщення), що входять до складу об'єктів нерухомості. Ця таблиця є однією з ключових у структурі CRM-системи, оскільки дозволяє зберігати інформацію про конкретні приміщення, їхню вартість оренди, статус зайнятості та зв'язок із орендарями [29].

У таблиці unit фіксується унікальний ідентифікатор кожного приміщення, до якого прив'язано інформацію про розташування, орендну ставку та статус доступності. Зв'язок із таблицею property забезпечується за допомогою колонки Property Name, що дозволяє AppSheet вибудовувати зв'язки між об'єктами нерухомості та їхніми орендними одиницями.

Структура таблиці unit наведена в таблиці 3.2.

Таблиця 3.2 – Структура таблиці unit

Unit ID	Property Name	Address	Unit No.	Monthly Rent (\$)	Tenant
SV100	Sunset View	200 Solar Ave WA	100	1500	Available
SV102	Sunset View	200 Solar Ave WA	102	1200	Ron
BW101	Beach Walk	300 Ocean Place WA	101	2400	Fred

Колонка Unit ID є унікальним ідентифікатором кожного приміщення та використовується як ключове поле для побудови зв'язків між таблицями в AppSheet. Поле Property Name вказує, до якого об'єкта нерухомості належить орендна одиниця. Address дублює адресу об'єкта для швидкого доступу до місцезнаходження. Поле Unit No. зберігає номер приміщення всередині об'єкта. Monthly Rent (\$) містить вартість місячної оренди, що може бути використана для розрахунків, звітності та автоматичних нагадувань про оплату. Поле Tenant показує поточний статус приміщення – або ім'я орендаря, або позначку Available, що дозволяє швидко бачити зайняті й доступні приміщення [30].

Наступним етапом проектування бази даних було створення таблиці move-in, яка є важливою частиною CRM-системи, що дозволяє фіксувати стан орендних одиниць на момент заселення орендаря. Ця таблиця допомагає детально зафіксувати інвентаризацію житлових та офісних приміщень на момент заселення, що є критичним для уникнення суперечок щодо стану майна під час та після оренди. Вона містить ключову інформацію про дату заселення, орендаря, а також подробиці про стан різних частин приміщення.

У таблиці move-in кожен запис має унікальний Move-in ID, який складається з комбінації ідентифікатора орендної одиниці та дати заселення, що дозволяє точно ідентифікувати кожен випадок заселення. Поле Move-in Date вказує дату заселення

орендаря, що забезпечує хронологічний облік подій. Unit ID є зв'язуючим елементом з таблицею units і дозволяє визначити, до якої саме орендної одиниці було заселено орендаря.

Детальніше, Tenant Name відображає ім'я орендаря, який заселяється, що дає змогу встановити зв'язок між заселенням і конкретним користувачем. Поля, що описують стан кожної зони приміщення (Living Room, Dining Room, Hallway, Kitchen), включають оцінки якості стану: Living Room Doors & Locks, Living Room Windows & Screens, Living Room Carpet or Flooring, і подібні для кожної кімнати. Ці оцінки (наприклад, "Good", "Fair", "Poor") дають змогу документувати фактичний стан приміщення на момент заселення, що важливо для подальшого використання в процесах ремонту чи обслуговування.

Окрім цього, важливою частиною таблиці є поле Tenant Signature, яке містить посилання на зображення підпису орендаря, що підтверджує згоди з описаним станом майна. Це є юридично важливим елементом, який дозволяє забезпечити точність і легітимність документування фактичного стану приміщення на момент заселення.

Структура таблиці unit наведена в таблиці 3.3.

Таблиця 3.3 – Структура таблиці move-in

Move-in ID	Move-in Date	Unit ID	Tenant Name	Living Room Doors & Locks	Living Room Windows & Screens
BW10 0: 07/06/2016	7/8/202 3	BW10 0	Harry	Good	Good

Останнім етапом у проєктуванні бази даних стала таблиця move-out, яка є логічним продовженням таблиці move-in та документує інформацію про виїзд

орендаря з орендної одиниці. Ця таблиця допомагає фіксувати стан майна при поверненні приміщення, порівнюючи його з початковим станом, що було зафіксовано на момент заселення. Таке документування необхідне для забезпечення прозорості в управлінні орендними відносинами та в разі потреби для врегулювання спорів щодо пошкоджень чи втрат.

У таблиці move-out ключовим полем є Move-out ID, яке являє собою унікальний ідентифікатор кожного виїзду, що може включати комбінацію Move-in ID (ідентифікатор заселення) та дати виїзду. Поле Move-out Date відображає дату виїзду орендаря, що дозволяє мати точний облік завершення оренди.

Unit ID є зв'язуючим елементом з таблицею units, що дозволяє ідентифікувати орендну одиницю, з якої виїжджає орендар. Tenant Name відображає ім'я орендаря, що виїжджає, що допомагає відслідковувати історію оренди кожної одиниці.

Таблиця move-out містить низку колонок, що описують стан приміщення при виїзді орендаря. Для кожної кімнати (Living Room, Dining Room, Hallway, Kitchen) є дві пари колонок: одна для стану приміщення на момент заселення (наприклад, Living Room Doors & Locks at Move-in), і інша – для стану на момент виїзду (наприклад, Living Room Doors & Locks). Такі поля дозволяють порівнювати зміни стану приміщення і фіксувати всі пошкодження або зміни, які відбулися за час оренди.

Крім того, є поле Tenant Signature, яке містить посилання на зображення підпису орендаря при виїзді. Це підпис підтверджує погодження орендаря з фіксованим станом майна на момент виїзду, що є важливим для юридичної сторони угоди.

Структура таблиці unit наведена в таблиці 3.4.

Таблиця 3.4 – Структура таблиці move-out

Move-out ID	Move-in ID	Move-out Date	Unit ID	Tenant Name	Page 2 Header
BW10 0: 07/06/2016	BW10 0: 07/06/2016	7/6/202 3	BW10 0	Harry	

3.2 Створення додатку в Google AppSheet і налаштування основного функціоналу

Першим кроком буде імпорт всіх створених таблиць в минулому пункті до створеного проекту. Імпортована таблиця unit зображена на рисунку 3.2.

Table: Unit					
Source: Properties Qualifier: Unit Data Source: google Source Type: Sheets Columns: 10					
NAME	TYPE	KEY?	LABEL?	FORMULA	
1 _RowNumber	Number	☐	☐	=	
2 Unit ID	Text	☒	☐	=	
3 Property Name	Ref	☐	☐	=	
4 Address	Address	☐	☐	=	
5 Unit No.	Number	☐	☐	=	
6 Monthly Rent (\$)	Price	☐	☐	=	
7 Tenant	Text	☐	☐	=	
8 Move in	List	☐	☐	= REF_I	
9 Unit Label	Text	☐	☒	= CONC	
10 Property Image	Image	☐	☐	= [Pro	

Рисунок 3.2 – Імпортована таблиця unit

Далі додаємо наступну таблицю – properties. Результат цього можна побачити на рисунку 3.3.

Table: Properties [View data source](#)    

Source: Properties Qualifier: Property Data Source: google Source Type: Sheets Columns: 5

	NAME	TYPE	KEY?	LABEL?	FORMULA
1	 _RowNumber	Number	☐	☐	=
2	 Property Name	Name	☒	☒	=
3	 Address	Address	☐	☐	=
4	 Image	Image	☐	☒	=
5	 Units	List	☐	☐	= REF_R

Рисунок 3.3 – Імпортована таблиця properties

Наступною на черзі йде таблиця move-in. Її доданою в проект можна побачити на рисунку 3.4.

Table: Move in [View data source](#)    

Source: Properties Qualifier: Move-in Data Source: google Source Type: Sheets Columns: 24

	NAME	TYPE	KEY?	LABEL?	FORMULA
1	 _RowNumber	Number	☐	☐	=
2	 Move-in ID	Text	☒	☐	=
3	 Move-in Date	Date	☐	☐	=
4	 Unit ID	Ref	☐	☐	=
5	 Tenant Name	Name	☐	☒	=
6	 Page 2 Header	Show	☐	☐	=
7	 Section 1 Header	Show	☐	☐	=
8	 Living Room Doors & Loc	Enum	☐	☐	=
9	 Living Room Windows & :	Enum	☐	☐	=
10	 Living Room Carpet or Flk	Enum	☐	☐	=

Рисунок 3.4 – Імпортована таблиця move-in

І останньою на черзі додамо таблицю move-out. Імпортована таблиця unit зображена на рисунку 3.5.

Table: Move out View data source

Source: Properti... Qualifier: Move-o... Data Source: goog... Source Type: Shee... Columns: 36

	NAME	TYPE	KEY?	LABEL?	FORMULA
1	_RowNumber	Number	☐	☐	=
2	Move-out ID	Text	☒	☐	=
3	Move-in ID	Ref	☐	☐	=
4	Move-out Date	Date	☐	☐	=
5	Unit ID	Text	☐	☐	= [Movi
6	Tenant Name	Name	☐	☒	= [Movi
7	Page 2 Header	Show	☐	☐	=
8	Section 1 Header	Show	☐	☐	=
9	Living Room Doors & Loc	Enum	☐	☐	= [Movi
10	Living Room Doors & Loc	Enum	☐	☐	=

Рисунок 3.5 – Імпортована таблиця move-out

Наступним кроком є додавання фільтрації даних для подальшого виводу їх в нашому застосунку. Почнемо з фільтрації таблиці unit. Нам потрібно буде колекція де поле [Tenant]="Available". Створену відфільтровану колекцію можна побачити на рисунку 3.6.

Data +

Move in

Active Move in

Move out

Properties

Unit

Available Units

Beach_Walk_Units

Sunset_View_Units

Wind_Valley_Units

OPTIONS

User settings

Slice: Available Units Preview data

Slice Name

The unique name for this slice

Available Units

Source Table

Which table to use as a source

Unit

Row filter condition

True/false expression that checks if a row should be included in the slice

= [Tenant]="Available"

Slice Columns

Columns to include in the slice

+ Add

_RowNumber

Unit ID Key

Property Name

Address

Unit No.

Monthly Rent (\$)

Tenant

...

Рисунок 3.6 – Відфільтрована по доступності таблиця unit

Також нам потрібні будуть апартаменти в яких [Property Name]="Beach Walk", [Property Name]="Sunset View" та [Property Name]="Wind Valley" для відповідних вкладок застосунку.

Також в таблиці move-in нам потрібно відфільтрувати апартаменти, в які вже заїхали ал еще не виїхали щоб в них не було змоги нікого іншого заселити. Для цього застосуємо умову ISBLANK([Move out]). Результат можна побачити на рисунку 3.7.

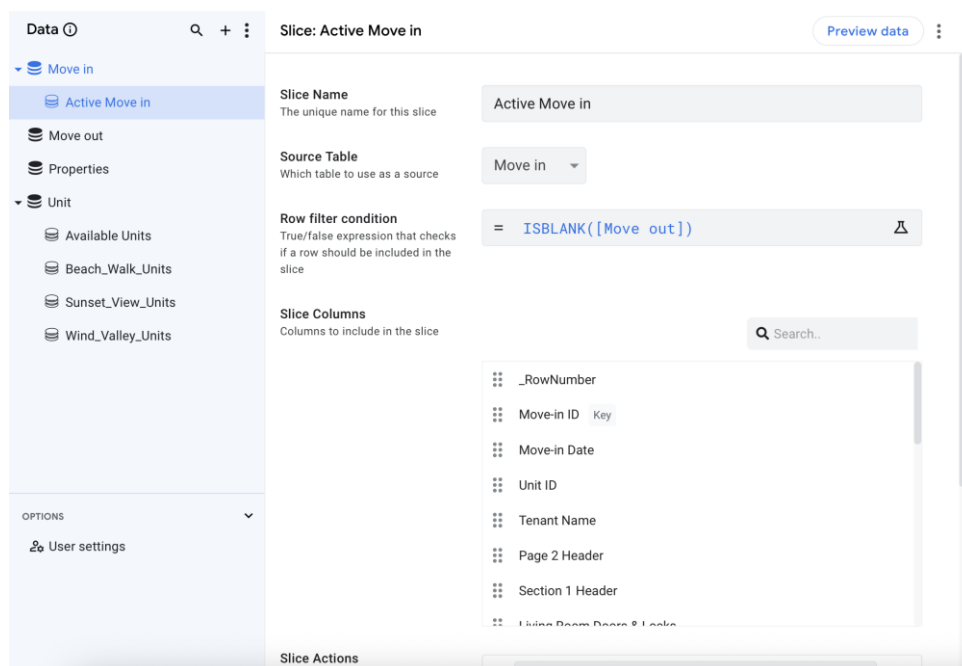


Рисунок 3.7 – Таблиця з апартаментами, які недоступні до заселення.

Далі перейдемо до створення сторінок з відображенням та додаванням інформації яка знаходиться в базі даних.

Першою створимо сторінку Available на якій будуть доступні до заселення квартири. Для цього додамо ім'я, виберемо таблицю з даними які ми відфільтрували, виберемо режим відображення картка, також позицію кнопки для доступу до цієї сторінки і виберемо сортування за вартістю від найменш до найбільш дорогих (рис. 3.8).

Available

Show in preview

View name

The unique name for this view.

Available

Go to display options

For this data

Which table or slice to display.

Available Units

View type

What kind of view this is.

calendar

deck

table

gallery

detail

map

chart

dashboard

form

onboarding

card

Position

Where the button to access this view is located.

first

next

middle

later

last

menu

ref

View Options

Sort by

Sort the rows by one or more columns.

Monthly Rent (\$)

Ascending

Рисунок 3.8 – Створення сторінки Available

Далі перейдемо до створення сторінки Move-in для оформлення заявки на в'їзд в апартаменти. Виберемо ім'я, дані які будемо заповнювати, режим відображення форму і розташування кнопки по центру (рис. 3.9).

Move-in

Show in preview

View name

The unique name for this view.

Move-in

Go to display options

For this data

Which table or slice to display.

Move in

Use slices to filter your data

View type

What kind of view this is.

calendar

deck

table

gallery

detail

map

chart

dashboard

form

onboarding

card

Position

Where the button to access this view is located.

first

next

middle

later

last

menu

ref

Рисунок 3.9 – Створення сторінки Move-in

Також потрібно налаштувати форму для заповнення даних. Для цього виберемо формат сторінки – автоматичну, стиль форми також автоматичну і порядок колонок також автоматичний, тобто в такій самій черзі в якій вони йдуть в базі даних і також виберемо максимальну кількість вкладених рядків (рис. 3.10).

Move-in [Show in preview](#) ⋮

identified by this key.

Page style
How form pages are displayed.

Automatic Simple Page Count Tabs

Form style
How form inputs are displayed.

Automatic Default Side-by-side

Column order
Display columns in a different order than they appear in the original data.

Automatic Manual

Move-in Date
Unit ID
Tenant Name
Page 2 Header
Section 1 Header
Living Room Doors & Locks
Living Room Windows & Screens

Save / cancel position
How the save / cancel buttons are positioned.

Bottom Top

Max. nested rows
The maximum number of rows to show for related references lists.

– 5 +

Рисунок 3.10 – Налаштування форми для в'їзду в квартиру

І так само створимо сторінку з формою для виїзду з квартири. Введемо ім'я, виберемо данні з таблиці move-out, режим відображення також форму і розташування кнопки останньою (рис. 3.11).

Move-out [Show in preview](#) ⋮

View name
The unique name for this view.

Move-out

[Go to display options](#) ⓘ

For this data
Which table or slice to display.

Move out

✦ Use slices to filter your data ⓘ

View type
What kind of view this is.

calendar	deck	table	gallery	detail
map	chart	dashboard	form	onboarding
card				

Position
Where the button to access this view is located.

first	next	middle	later	last	menu	ref
-------	------	--------	--------------	------	------	-----

Рисунок 3.11 – Створення сторінки Move-out з формую

І тепер ми можемо додати посилання на ці сторінки в нижню панель навігації (рис. 3.12).

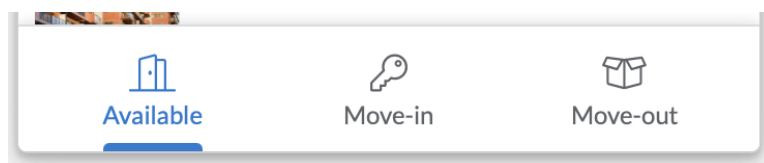


Рисунок 3.12 – Нижня панель навігації

Тепер перейдемо до створення сторінки з відображенням всіх квартир з усіх локацій. Виберемо дані з таблиці Unit, відповідну іконку і розташування в меню, так як кнопка для переходу на цю сторінку буде в боковому меню (рис. 3.13).

All Units
Show in preview

View name
The unique name for this view.

All Units

[Go to display options](#)

For this data
Which table or slice to display.

Unit

[Use slices to filter your data](#)

View type
What kind of view this is.

calendar

deck

table

gallery

detail

map

chart

dashboard

form

onboarding

card

Position
Where the button to access this view is located.

first
next
middle
later
last
menu
ref

Рисунок 3.13 – Створення сторінки All units

І також ще потрібно налаштувати відображення. Сортування буде по ціні від найменшого до найбільшого, групування буде по назві комплексу (рис. 3.14).

Available
Show in preview

View Options

Sort by
Sort the rows by one or more columns.

Monthly Rent (\$)
Ascending

Add

Group by
Group rows by the values in one or more of their columns.

Property Name
Ascending

Add

Group aggregate
Display a numeric summary of the rows in each group.

NONE

Рисунок 3.14 – Налаштування відображення даних на сторінці All Units

Далі створимо окремі сторінки для кожного комплексу. Розглянемо сторінку Beach Walk Units. Виберемо данні з таблиці Beach_Walk_Units, вид відображення таблиця, без сортування і групування (рис. 3.15). І аналогічно зробимо сторінки

Sunset View Units та Wind Valley Units.

Beach Walk Units

Show in preview

View name

The unique name for this view.

Beach Walk Units

Go to display options

For this data

Which table or slice to display.

Beach_Walk_Units

View type

What kind of view this is.

calendar

deck

table

gallery

detail

map

chart

dashboard

form

onboarding

card

Position

Where the button to access this view is located.

first

next

middle

later

last

menu

ref

Рисунок 3.15 – Сторінка Beach Walk Units

І на останок створимо сторінку Properties де відобразимо дані про кожен комплекс з кількістю доступних квартир в кожному. Виберемо дані з таблиці Properties, відображення таблицею, положення кнопки для відкриття в меню, без сортування і групування (рис. 3.16).

Properties

Show in preview

View name

The unique name for this view.

Properties

Go to display options

For this data

Which table or slice to display.

Properties

Use slices to filter your data

View type

What kind of view this is.

calendar

deck

table

gallery

detail

map

chart

dashboard

form

onboarding

card

Position

Where the button to access this view is located.

first

next

middle

later

last

menu

ref

Рисунок 3.16 – Сторінка Properties

І тепер ми можемо додати всі створені сторінки на бокове меню нашого застосунку (рис.3.17).

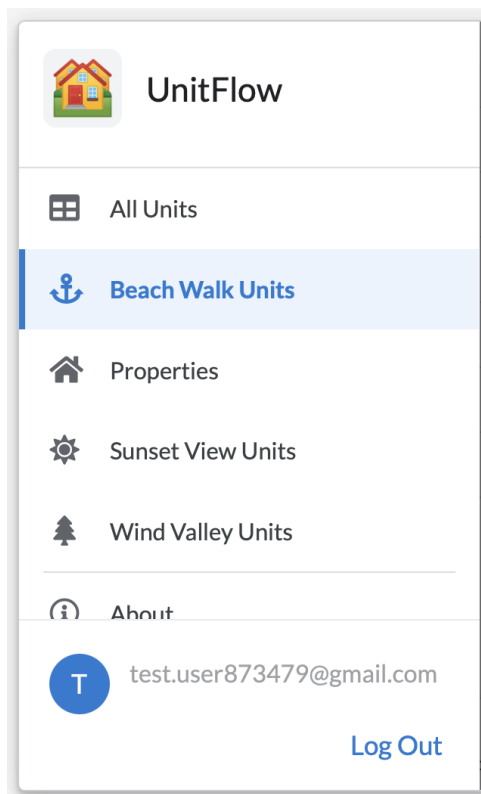


Рисунок 3.17 – Бокове меню застосунку

3.3 Тестування функціоналу додатку

При відкритті застосунку бачимо вкладку з доступними для оренди житловими комплексами, згрупованими по назві комплексу (рис. 3.18).

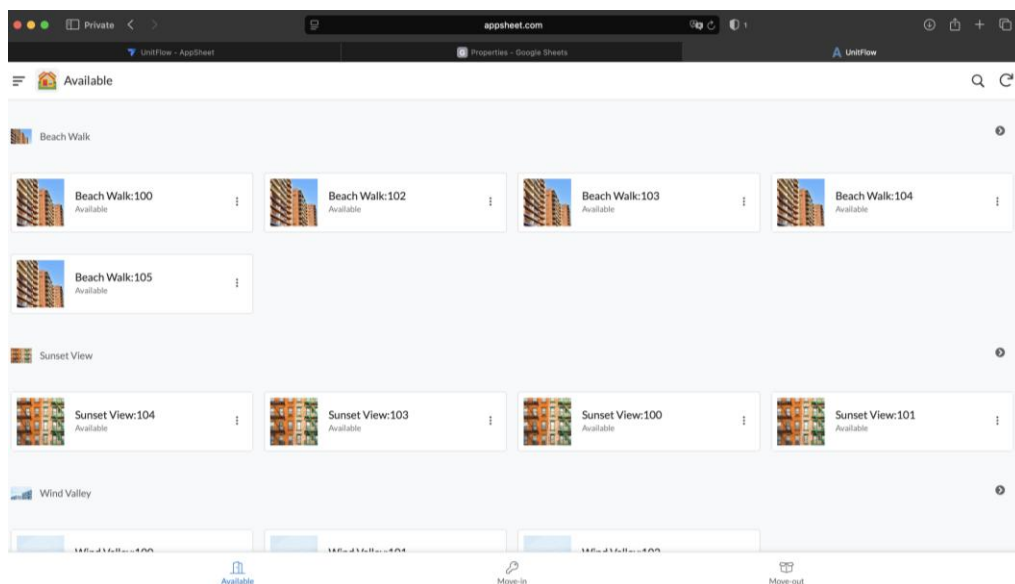


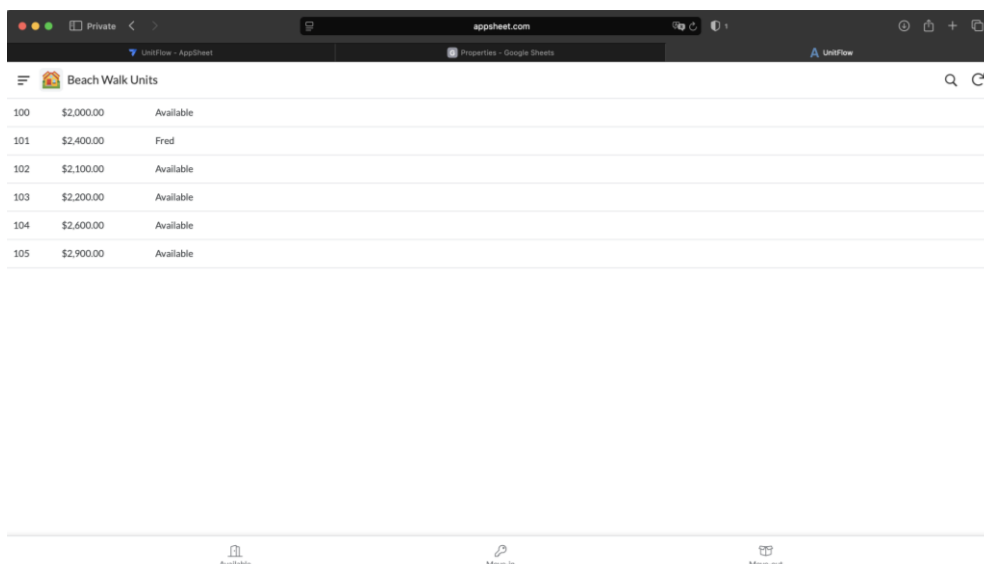
Рисунок 3.18 – Сторінка Available

Далі переглянемо всі житлові комплекси на вкладці All Units (рис. 3.19). Де можна побачити згруповані квартири по назві комплексу і ціну за оренду.

Property	Unit Number	Rent Price	Status
Beach Walk	100	\$2,000.00	Available
	101	\$2,400.00	Fred
	102	\$2,100.00	Available
	103	\$2,200.00	Available
	104	\$2,600.00	Available
	105	\$2,900.00	Available
Sunset View	100	\$1,500.00	Available
	101	\$1,700.00	Available
	102	\$1,200.00	Ron
	103	\$1,400.00	Available
	104	\$1,300.00	Available
Wind Valley	100	\$3,500.00	Available

Рисунок 3.19 – Сторінка All Units

Далі перейдемо на вкладку конкретного комплексу де можна побачити всі квартири які відносяться до нього з ціною за оренду і інформацію про те чи доступні вони чи ні (рис. 3.20). Ідентично виглядають сторінки Sunset View Units та Wind Valley Units.



Unit ID	Price	Status			
100	\$2,000.00	Available			
101	\$2,400.00	Fred			
102	\$2,100.00	Available			
103	\$2,200.00	Available			
104	\$2,600.00	Available			
105	\$2,900.00	Available			

Рисунок 3.20 – Сторінка Beach Walk Units

Далі протестуємо бронювання квартири для в'їзду. Після переходу на вкладку Move-in заповнимо дату в'їзду, виберемо назву комплексу і введемо ім'я (рис. 3.21).

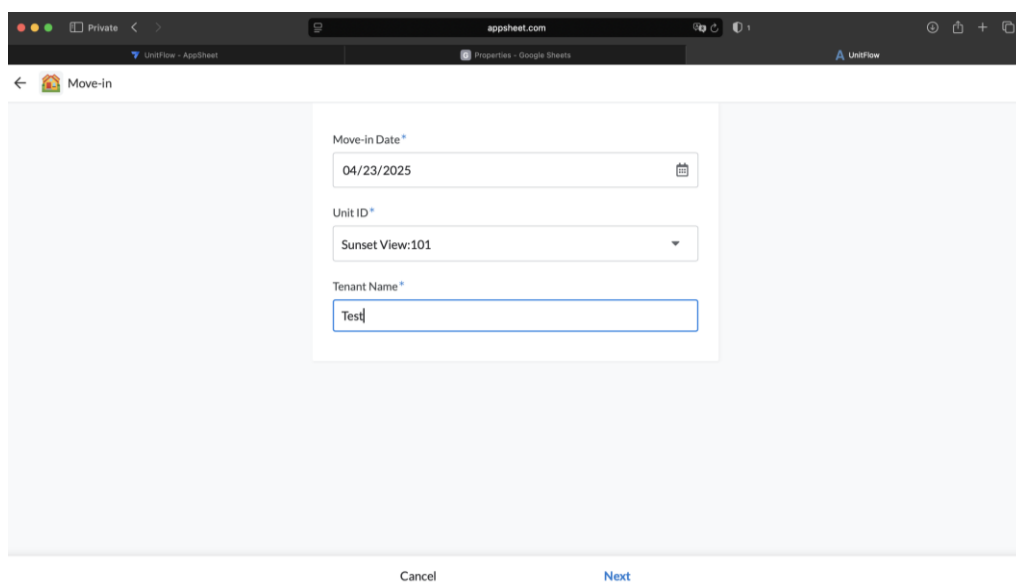


Рисунок 3.21 – Заповнення базової інформації для в'їзду

Далі виберемо який стан в кожній важливій речі в апартаментах щоб при виселенні звірити чи все в такому ж стані як було (рис. 3.22). І також затвердимо це підписом.

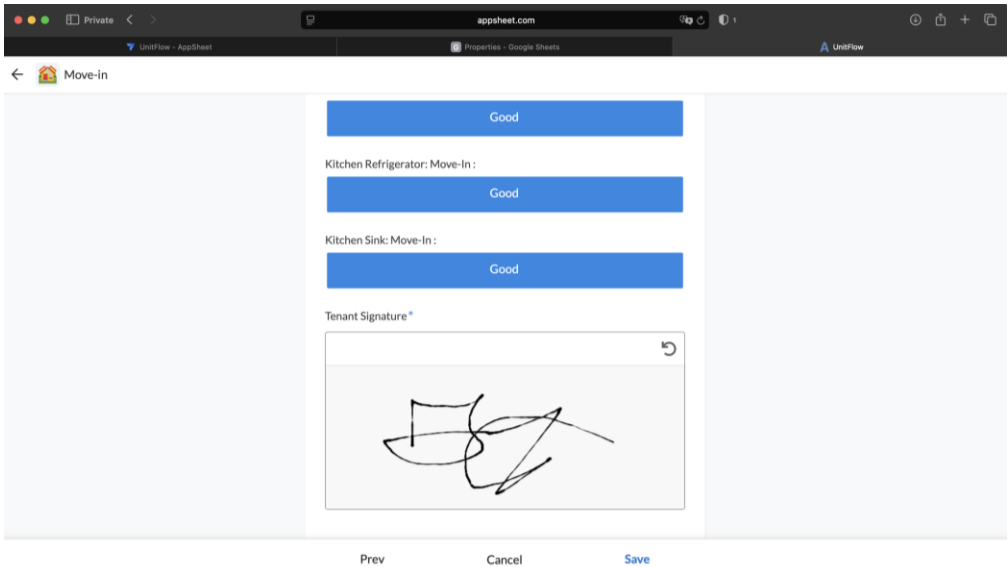


Рисунок 3.22 – Перевірка стану речей в апартаментах до в’їзду

І перевіримо чи додався запис про в’їзд до бази даних (рис. 3.23). Я ми бачимо що останнім рядком якраз додався запис про в’їзд з ім’ям Test.

The screenshot shows a Google Sheet with the following data:

Move-in ID	Move-in Date	Unit ID	Tenant Name	Page 2 Header	Section 1 Header	Living Room Door	Living Room W	Living Room Ca	Section 2 Head	Dining Room W	Dining Room Ca	Section 3 Head	Halfway Ci
BW100: 07/06/2016	7/6/2023	BW100	Harry			Good	Good	Good	Good	Good	Good	Good	Good
SV102: 07/06/2016	7/6/2023	SV102	Ron			Good	Good	Good	Good	Good	Good	Good	Good
SV100: 01/14/2021	1/14/2024	SV100	Joe			Good	Good	Good	Good	Good	Good	Good	Good
BW100: 01/14/2021	1/14/2024	BW100	Sue			Good	Good	Good	Good	Good	Good	Good	Good
SV104: 02/11/2021	2/11/2024	SV104	Bob			Good	Good	Good	Good	Good	Good	Good	Good
BW101: 02/21/2021	2/21/2024	BW101	Fred			Good	Good	Good	Good	Good	Good	Good	Good
SV100: 02/21/2021	2/21/2024	SV100	Sally			Good	Good	Good	Good	Good	Good	Good	Good
SV101: 04/23/2025	4/23/2025	SV101	Test			Good	Good	Good	Good	Good	Good	Good	Good

Рисунок 3.23 – Перевірка про в’їзд в базі даних

Далі протестуємо функціонал виїзду з квартири. При переході на відповідну вкладку виберемо жильця Test і автоматично побачимо що вся інформація про цього підтягнулась (рис. 2.24).

Move-out

Move-in ID
Test

Move-out Date *
04/23/2025

Unit ID *
SV101

Tenant Name *
Test

Cancel Next

Рисунок 3.24 – Вибір хто вижає з квартири

Далі так само заповнюємо дані про стан речей після перебування в квартирі і також додамо підпис який це підтверджує (рис. 2.25).

Move-out

Good

Kitchen Sink Move-In :
Good

Kitchen Sink Move-Out : *
Good

Tenant Signature *
[Handwritten Signature]

Prev Cancel Save

Рисунок 3.25 – Заповнені дані про виїзд з квартири

І тепер на перевіримо базу даних чи додався запис про виїзд людини з ім'ям Test (рис. 3.26). Як ми бачимо останнім записом є інформація про виселення даної особи.

Move-out ID	Move-in ID	Move-out Date	Unit ID	Tenant Name	Page 2 Header	Section 1 Header	Living Room	Do Living Room	Wi Living Room	Ca Living Room	Section 2 Header
BW100: 07/06/2016	BW100: 07/06/2016	7/6/2023	BW100	Harry			Good	Good	Good	Good	
BW105: 07/06/2016	BW105: 07/06/2016	7/6/2023	BW105	Hermione			Good	Good	Good	Good	
SV104: 02/11/2021	SV104: 02/11/2021	2/11/2024	SV104	Bob			Good	Good	Good	Good	
SV100: 02/18/2021	SV100: 01/14/2021	2/18/2024	SV100	Sally			Good	Good	Good	Good	
SV101: 04/23/2025	SV101: 04/23/2025	4/23/2025	SV101	Test			Good	Good	Good	Good	

Рисунок 3.26 – DB з інформацією про виселення

3.4 Висновки до 3 розділу

У третьому розділі було безпосередньо реалізовано CRM-систему для оренди квартир з використанням платформи Google AppSheet. Розробка розпочалася з проєктування структури бази даних у Google Sheets, яка включає інформацію про клієнтів, об'єкти нерухомості, договори оренди, оплати, звернення тощо. Особливу увагу було приділено зв'язкам між таблицями та забезпеченню цілісності даних.

На основі цієї бази даних було створено функціональний додаток у Google AppSheet. Реалізовано ключовий функціонал: перегляд і фільтрація об'єктів, додавання нових записів, редагування даних, відображення статистики й історії оренди. Система підтримує рольовий доступ, що дозволяє обмежити права.

У завершальній частині було проведено тестування додатку, яке підтвердило працездатність усіх основних функцій, а також зручність інтерфейсу для кінцевих користувачів. Виявлені недоліки були усунені на етапі налагодження.

Таким чином, результатом третього розділу є повноцінний прототип CRM-системи, який може бути використаний у реальному середовищі для автоматизації процесів управління орендою квартир.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Інженерно-психологічні принципи професійного добору

Професійний добір персоналу в цифровій сфері – це не лише кадрова процедура, а важливий елемент безпеки життєдіяльності. Особливо це актуально в умовах використання CRM-систем, зокрема побудованих на платформі Google Appsheet, де оператори виконують завдання в інтенсивному інформаційному середовищі. Надійність їхньої роботи безпосередньо впливає на якість послуг, а також на безпеку процесів управління орендною нерухомістю.

У сучасних умовах цифровізації трудової діяльності значення професійного добору персоналу зростає як з точки зору ефективності роботи, так і забезпечення безпеки життєдіяльності. Особливо це актуально для спеціалістів, які працюють із інформаційними системами, такими як CRM-системи, призначені для автоматизації управління орендою квартир. Такі системи передбачають взаємодію з великими обсягами інформації, швидке прийняття рішень, безперервну комунікацію з клієнтами та гнучке реагування на змінні ситуації. Це створює додаткове психоемоційне навантаження, з яким не всі користувачі можуть впоратися однаково ефективно. Тому інженерно-психологічний підхід до добору персоналу є не лише корисним, а й необхідним компонентом.

Професійний добір як елемент охорони праці ґрунтується на нормативній базі. Зокрема, відповідно до Закону України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ, роботодавець зобов'язаний організовувати робочий процес із урахуванням фізіологічних і психофізіологічних особливостей працівників [31]. Це включає їх добір, навчання, контроль стану здоров'я та адаптацію до умов праці.

Крім того, згідно з Постановою Кабінету Міністрів України № 77 від 03.02.2021 р., професії, пов'язані з ризиком, підпадають під обов'язкове психофізіологічне обстеження [32]. Це положення прямо стосується CRM-операторів, які працюють в умовах підвищеного інформаційного навантаження та стресу.

Психофізіологічна оцінка кандидатів – необхідний етап, що дозволяє попередити зниження продуктивності або помилки через надмірне навантаження.

Тож у CRM, створеній за допомогою Google Appsheet, можна реалізувати такі механізми: симуляційні вправи, відстеження кількості помилок, оцінка часу реакції, автоматичні сигнали про перевантаження або вигорання – з можливістю корекції навантаження або організації підтримки.

Крім технічних рішень, важливо створити інтерфейси з урахуванням ергономічних вимог (ДСТУ 7299:2013; ДСТУ 8604:2015), що значно знижує когнітивне навантаження на користувача. Це забезпечує комфортну та безпечну роботу, мінімізує кількість помилок та підвищує продуктивність.

Отже, інженерно-психологічні принципи – відповідність «людина–система», багаторівневий добір, облік когнітивних факторів, адаптація та психофізіологічний моніторинг – стають важливою складовою під час розробки CRM для оренди квартир. Вони забезпечують стабільність, безпеку праці та зменшення ризиків професійного вигорання й помилок.

4.2 Заходи, що покращують умови праці оператора

Ефективність використання CRM-системи для оренди квартир безпосередньо залежить від організації умов праці та дотримання норм охорони праці. Серед основних показників ефективності можна відзначити зменшення часу, необхідного на виконання рутинних операцій, таких як внесення даних про орендарів, формування звітів чи перегляд історії проживання. Водночас підвищується точність введення інформації, знижується ймовірність помилок, а також покращується швидкість доступу до необхідних даних, що дозволяє працівникам приймати більш обґрунтовані та своєчасні рішення. Усе це в сукупності сприяє підвищенню продуктивності праці й загальної задоволеності користувачів від роботи з системою.

Для забезпечення комфортної та безпечної роботи з CRM-системою необхідно створити ергономічне робоче місце, яке відповідатиме фізіологічним потребам працівника, а також сприятиме зменшенню навантаження на зір і опорно-руховий апарат. Використання сучасних моніторів із захистом очей, зручних меблів, правильного освітлення, а також впровадження практики регулярних перерв дозволяє мінімізувати вплив негативних факторів на здоров'я. Ефективне організаційне рішення для підвищення умов праці оператора – ергономічна організація робочого місця. Згідно з ДСТУ 7299:2013 та ДСТУ ISO 9241-5:2004 [33], компонування робочого місця має передбачати:

1. Регульоване крісло зі спинкою та підлокітниками (ДСТУ 7951:2015);
2. Стіл з можливістю регулювання висоти;
3. Правильне розташування монітора – на рівні очей, на відстані 50–70 см;
4. Достатнє робоче освітлення – не менше 300 лк [34].

Оператори повинні бути поінформовані про правила безпечної роботи з комп'ютером, а також мати доступ до інструктажів і навчальних матеріалів з охорони праці.

У межах реалізації системи на базі Google AppSheet важливою перевагою є хмарна архітектура, яка дозволяє працювати з додатком у віддаленому режимі. Це зменшує необхідність фізичної присутності в офісі, що позитивно впливає на організацію гнучкого графіка, зменшує стрес та сприяє дотриманню балансу між роботою та особистим життям. Інтерфейс системи має бути максимально простим, інтуїтивним і логічним, щоб зменшити когнітивне навантаження на оператора й забезпечити швидкий доступ до основних функцій. Крім того, доцільно періодично проводити моніторинг технічного стану обладнання, умов праці та стану здоров'я працівників, щоб своєчасно виявляти проблеми та впроваджувати коригувальні заходи.

Ергономічні вимоги до крісла оператора:

1. Крісло має забезпечувати оператору відповідну умовам праці фізіологічно раціональну робочу позу.

2. Крісло має забезпечувати тривалу підтримку основної робочої пози під час трудової діяльності. У той же час форма і розміри крісла не повинні вимагати від оператора фіксованої робочої пози.

3. У разі неможливості покинути робоче місце тривалий час конструкція крісла повинна забезпечувати умови для відпочинку оператора в кріслі.

4. Крісло має створювати умови для підтримки корпусу людини у фізіологічно раціональному положенні зі збереженням природних вигинів хребта.

5. Регулювання параметрів може бути плавне або східчасте. Крок східчастого регулювання для лінійних параметрів – 15–25 мм.

6. Вимоги до крісел робочих місць для роботи з відеотерміналами – згідно з ДСТУ ISO 9241-5.

7. В конструкції крісла має бути враховано загальні й окремі вимоги щодо безпеки. Загальні вимоги до безпечності сидіння рухомих і стаціонарних машин – згідно з ДСТУ EN 292-2. Вимоги до безпечності робочих стільців – згідно з ДСТУ ISO 9241-5. [35].

У підсумку, покращення умов праці та забезпечення відповідного рівня оператора є невід’ємною складовою ефективного впровадження та експлуатації CRM-системи. Це дозволяє не лише підвищити якість виконання завдань, а й сформувати здорове цифрове середовище, у якому оператори можуть почуватися комфортно та безпечно.

Ще одним важливим показником ефективності використання CRM-системи є рівень зниження кількості звернень до технічної підтримки з боку користувачів. Це свідчить про зрозумілий інтерфейс, належний рівень навчання персоналу та якість самої системи. У свою чергу, скорочення часу на вирішення технічних питань дозволяє зосередитись на основних функціональних обов’язках і сприяє кращому використанню трудових ресурсів підприємства.

Регулярне опитування працівників щодо рівня задоволеності умовами праці та функціоналом системи сприяє підвищенню їх залученості й лояльності до компанії.

З технічної точки зору, покращення умов оператора може також включати

використання засобів автоматизації, таких як гарячі клавіші, шаблони введення даних або автоматичне заповнення полів. Це не лише пришвидшує роботу, а й знижує ризик виникнення повторюваних рухів, які можуть спричинити перенавантаження м'язів чи розвиток професійних захворювань.

Крім того, важливою складовою охорони праці оператора є формування корпоративної культури, що орієнтована на безпеку, комфорт і психологічне благополуччя працівника. Роботодавець повинен не лише забезпечувати ергономічне робоче місце, належне освітлення та відповідні умови мікроклімату, а й проявляти системну турботу про персонал. Це включає участь у програмах профілактики зорових і опорно-рухових розладів, регулярне проведення тренінгів зі зниження рівня професійного стресу, а також заохочення здорового способу життя. Такий підхід сприяє створенню сприятливої робочої атмосфери, підвищує залученість персоналу та мотивує операторів відповідально ставитися до роботи з CRM-системою.

ВИСНОВКИ

У ході виконання дипломної роботи було реалізовано повний цикл розробки CRM-системи для управління процесами оренди квартир з використанням інструменту Google AppSheet. Основною метою дослідження було створення ефективного, доступного та зручного інструменту для ведення обліку орендарів, контроль за станом житлових об'єктів та автоматизація щоденних операцій, пов'язаних з управлінням житловим фондом.

Перший розділ було присвячено аналізу сучасного стану проблеми. Розглянуто приклади існуючих CRM-систем у сфері нерухомості, виявлено їхні переваги та недоліки, окреслено нерозв'язані завдання. Це дозволило чітко сформулювати дослідницьке завдання та визначити функціональні потреби, які має задовольняти розроблена система.

У другому розділі було проведено обґрунтування вибору Google AppSheet як платформи для реалізації CRM-системи. У результаті порівняльного аналізу альтернативних інструментів встановлено, що AppSheet є зручним low-code/ no-code рішенням, що дозволяє швидко та ефективно створювати додатки з доступом до хмарних таблиць Google Sheets. Також було розглянуто архітектуру подібних систем та визначено методику розробки, яка забезпечує гнучкість, масштабованість та простоту у впровадженні.

У третьому розділі здійснено практичну реалізацію системи. Було спроектовано структуру бази даних у Google Sheets, визначено основні таблиці та зв'язки між ними. На базі цієї структури створено застосунок у Google AppSheet, який включає основний функціонал: облік квартир, орендарів, дат заселення та виселення, перевірки стану житла при в'їзді та виїзді, завантаження фото, збереження підписів. Проведено тестування працездатності системи, що підтвердило її функціональність та придатність до використання у реальних умовах.

У четвертому розділі було розглянуто аспекти охорони праці та безпеки

життєдіяльності, що є важливим компонентом при впровадженні інформаційних технологій. Особливу увагу було приділено питанням ергономіки, організації безпечного цифрового середовища, а також заходам, що сприяють збереженню здоров'я працівників під час взаємодії з CRM-системою.

В результаті виконання дипломної роботи досягнуто поставленої мети. Було розроблено CRM-систему, яка відповідає вимогам сучасного управління орендою квартир, є зручною у використанні, не потребує спеціальної технічної підготовки та може бути легко адаптована під індивідуальні потреби користувачів. Отримані результати можуть бути впроваджені як в малий бізнес, так і в більші компанії, що займаються управлінням нерухомістю. У майбутньому систему можливо доповнити новими функціями, такими як автоматичні нагадування, інтеграція з платіжними сервісами або аналітичні панелі для оцінки ефективності управління житловим фондом.

Таким чином, розроблена система демонструє потенціал технологій low-code/ no-code у вирішенні прикладних задач та є прикладом практичного впровадження цифрової трансформації в сфері оренди житла.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Облік в хмарі чи Excel: що обираєте ви? [Електронний ресурс]. – Режим доступу: <https://miydimonline.com.ua/uk/blog/oblik-v-hmari-chy-excel>, вільний. – Дата звернення: 23.04.2025.
2. CRM для оренди нерухомості – Універсальна Система Обліку [Електронний ресурс]. – Режим доступу: https://ussoft.com.ua/uk/crm_dlya_arendy_nedvizhimosti.php, вільний. – Дата звернення: 23.04.2025.
3. CRM для агентства нерухомості – RemOnline [Електронний ресурс]. – Режим доступу: <https://remonline.ua/property-maintenance-software/>, вільний. – Дата звернення: 23.04.2025.
4. CRM для агентств нерухомості та ріелторів – Skylex [Електронний ресурс]. – Режим доступу: <https://skylex.com.ua/blog/crm-dlya-ahent-stv-nerukhomosti-ta-rieltoriv>, вільний. – Дата звернення: 23.04.2025.
5. CRM для агентства нерухомості, ріелторів, для забудовників – SalesDrive [Електронний ресурс]. – Режим доступу: <https://salesdrive.ua/by-industry/real-estate/>, вільний. – Дата звернення: 23.04.2025.
6. CRM для агентств нерухомості: застосунок, веб-сайт – Wezom [Електронний ресурс]. – Режим доступу: <https://wezom.com.ua/ua/blog/digital-reshenie-dlya-segmenta-nedvizhimosti>, вільний. – Дата звернення: 23.04.2025.
7. CRM для агенції нерухомості: все по полицкам – KeyCRM Blog [Електронний ресурс]. – Режим доступу: <https://blog.keycrm.app/uk/crm-dlya-agencii-neruhomosti-vse-po-polichkam/>, вільний. – Дата звернення: 23.04.2025.
8. CRM in Real Estate: Meaning, Features, Benefits – Inoxoft [Електронний ресурс]. – Режим доступу: <https://inoxoft.com/blog/what-is-crm-in-real-estate-and-what-business-needs-it-should-cover/>, вільний. – Дата звернення: 23.04.2025.
9. CRM for Real Estate: Benefits, Features, and Review of the Top 10 – FindMyCRM [Електронний ресурс]. – Режим доступу:

<https://www.findmycrm.com/blog/crm-industries/crm-for-real-estateurlipcitycountry>, вільний. – Дата звернення: 23.04.2025.

10. Low-code/ no-code real-life use cases – LowCode Agency [Електронний ресурс]. – Режим доступу: <https://www.lowcode.agency/blog/low-code-no-code-use-cases>, вільний. – Дата звернення: 23.04.2025.

11. Best Real Estate CRM Software: Top 12 Tools (Reviews and Pricing) – iovox [Електронний ресурс]. – Режим доступу: <https://www.iovox.com/blog/real-estate-crm>, вільний. – Дата звернення: 23.04.2025.

12. NoCode and LowCode Tools Revolutionizing Real Estate Development – Kreante [Електронний ресурс]. – Режим доступу: <https://www.kreante.co/post/nocode-and-lowcode-tools-revolutionizing-real-estate-development>, вільний. – Дата звернення: 23.04.2025.

13. How to Create a Simple CRM using a Google Sheet and AppSheet – LeadSync [Електронний ресурс]. – Режим доступу: <https://leadsync.me/blog/google-sheet-appsheet-app-crm/>, вільний. – Дата звернення: 23.04.2025.

14. Real Estate CRM: The Preferred CRM for Realtors – Salesforce India [Електронний ресурс]. – Режим доступу: <https://www.salesforce.com/in/crm/real-estate-crm/>, вільний. – Дата звернення: 23.04.2025.

15. AppSheet vs Power Apps vs Clappia: Which One is Best for You? [Електронний ресурс]. – Режим доступу: <https://www.clappia.com/blog/appsheet-vs-power-apps-vs-clappia>, вільний. – Дата звернення: 23.04.2025.

16. Best Cost Saving Low Code Development Platform Tools for Real Estate – The Storefront [Електронний ресурс]. – Режим доступу: <https://www.thestorefront.com/go/best-low-code-development-platform-tools-for-the-real-estate-industry/>, вільний. – Дата звернення: 23.04.2025.

17. How to build a real estate app for free—no code required – Softr [Електронний ресурс]. – Режим доступу: <https://www.softtr.io/blog/build-a-real-estate-app>, вільний. – Дата звернення: 23.04.2025.

18. No-Code / Low-Code Options in the Sugar Platform – SugarCRM [Електронний ресурс]. – Режим доступу: <https://www.sugarcrm.com/blog/no-code-low-code-sugar-platform/>, вільний. – Дата звернення: 23.04.2025.

19. 10 Best Real Estate CRM Software in 2025 – Forbes Advisor [Електронний ресурс]. – Режим доступу: <https://www.forbes.com/advisor/business/software/best-real-estate-crm/>, вільний. – Дата звернення: 23.04.2025.

20. Create a Custom CRM for Real Estate Agents Using No-Code Tools – NoCode MBA [Електронний ресурс]. – Режим доступу: <https://www.nocode.mba/articles/create-a-custom-crm-for-real-estate-agents-using-no-code-tools>, вільний. – Дата звернення: 23.04.2025.

21. What Does CRM Stand for in Real Estate? Benefits Explained – Blaze.tech [Електронний ресурс]. – Режим доступу: <https://www.blaze.tech/post/unraveling-the-mystery-what-is-a-crm-in-real-estate>, вільний. – Дата звернення: 23.04.2025.

22. How to Create a CRM Using Google Sheets & AppSheet [Електронний ресурс]. – Режим доступу: <https://leadsync.me/blog/google-sheet-appsheet-app-crm/>, вільний. – Дата звернення: 23.04.2025.

23. Building Your Own CRM With Google AppSheet [Електронний ресурс]. – Режим доступу: <https://www.thathaus.co.uk/post/building-your-own-crm-with-google-appsheet>, вільний. – Дата звернення: 23.04.2025.

24. Google AppSheet tutorial for non-technical citizen developers [Електронний ресурс]. – Режим доступу: <https://pretius.com/blog/google-appsheet-tutorial/>, вільний. – Дата звернення: 23.04.2025.

25. Build a custom CRM with AppSheet [Електронний ресурс]. – Режим доступу: <https://www.incahootsco.com.au/build-a-custom-crm-with-appsheet>, вільний. – Дата звернення: 23.04.2025.

26. Creating CRM in AppSheet Part 3 [Електронний ресурс]. – Режим доступу: <https://arisazhar.com/creating-crm-in-appsheet-part-3/>, вільний. – Дата звернення: 23.04.2025.

27. Google AppSheet Primer: What It Can Do and How to Use It [Електронний ресурс]. – Режим доступу: <https://www.reworked.co/digital-workplace/google>

appsheet-primer-what-it-can-do-and-how-to-use-it/, вільний. – Дата звернення: 23.04.2025.

28. Toki – Custom CRM build with AppSheet [Електронний ресурс]. – Режим доступу: <https://cloudoffice.max-media.io/g/success-stories/toki-appsheet>, вільний. – Дата звернення: 23.04.2025.

29. How to Build a CRM with Google Sheets [Електронний ресурс]. – Режим доступу: <https://www.driftrack.com/blog/how-to-build-a-crm-with-google-sheet>, вільний. – Дата звернення: 23.04.2025.

30. Free CRM Templates in Google Sheets & Excel [Електронний ресурс]. – Режим доступу: <https://www.onepagecrm.com/blog/crm-spreadsheet-template/>, вільний. – Дата звернення: 23.04.2025.

31. Закон України «Про охорону праці» № 2694-XII від 14.10.1992 [Електронний ресурс]. – База даних «Законодавство України» / Верховна Рада України. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12> . – Дата звернення: 15.06.2025.

32. Кабінет Міністрів України. Постанова № 77 від 03.02.2021 р. «Про затвердження переліку машин, механізмів, устаткування підвищеної небезпеки» [Електронний ресурс]. – База даних «Законодавство України» / Верховна Рада України. – Режим доступу: <https://zakon.rada.gov.ua/go/77-2021-п> . – Дата звернення: 15.06.2025.

33. ДСТУ ISO 9241-5:2004 Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 5. Вимоги до компонування робочого місця та до робочої пози (ISO 9241-5:1998, IDT). [Електронний ресурс]. – Режим доступу: https://online.budstandart.com/ua/catalog/doc-page?id_doc=53291 – Дата звернення: 15.06.2025.

34. Організація робочого місця Оператора ПК. [Електронний ресурс]. – Режим доступу: <https://studcon.org/organizaciya-robochogo-miscya-operatora-pk?page=5>. – Дата звернення: 15.06.2025

35. Дизайн і ергономіка КРІСЛЮ ОПЕРАТОРА Загальні ергономічні вимоги ДСТУ 7951:2015 [Електронний ресурс]. – Режим доступу:

https://www.ksv.biz.ua/GOST/DSTY_ALL/DSTU2/dstu_7951-2015.pdf. – Дата звернення: 15.06.2025

36. Дистанційний курс «Кваліфікаційні роботи бакалаврів» сайту дистанційного навчання ТНТУ [Електронний ресурс]. – Режим доступу: <https://dl.tntu.edu.ua/bounce.php?course=5329> – Дата звернення: 15.06.2025

37. Guide to the Software Engineering Body of Knowledge. [Електронний ресурс]. – Режим доступу: <https://ieeecsmedia.computer.org/media/education/swebok/swebok-v4.pdf> of access: – Дата звернення: 15.06.2025

38. F2-121-методичні вказівки до виконання КР бакалавра-2024. . [Електронний ресурс]. – Режим доступу: https://docs.google.com/document/d/e/2PACX1vTZ_e_W6gyWi3_duGZX22jkbhFXe3QxvjOYnbUtWmQ40lszuvLT0Z2P84IAWkjGgRIrIoAYPWPL7OJYA/pub?urp=gmail_link. – Дата звернення: 15.06.2025

ДОДАТКИ

ДОДАТОК А

Диск з роботою: