

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**ФАКУЛЬТЕТ ЕКОНОМІЧНОЇ ІНФОРМАТИКИ**

**КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ**

## **Пояснювальна записка**

до дипломного проекту

бакалавра

на тему: Розроблення модуля обліку роботи з бронюванням житла на основі  
CRM-платформи Salesforce

**Виконала:** студентка другого року навчання за  
скороченою програмою,  
групи 6.04.121.012.18.1,  
спеціальності 121  
«Інженерія програмного забезпечення»  
Федорченко Р. В.

Керівник: к.т.н., проф. кафедри ІС ХНЕУ  
Щербаков О. В.

Харків – 2020 рік

## РЕФЕРАТ

Пояснювальна записка до дипломної роботи бакалавра містить 188 сторінок, 15 таблиць, 48 ілюстрацій, 1 додаток та список використаної літератури з 50 найменувань.

Об'єктом дослідження дипломного проєкту є засоби для автоматизації діяльності агентств з пошуку нерухомості.

Мета роботи: Розробка математичного, алгоритмічного та програмного забезпечення програмного модулю для ведення обліку роботи з бронюваннями на основі CRM-платформи Salesforce.

При написанні роботи використовувалися наступні методи дослідження: аналітичний, програмний.

У ході виконання даної дипломної роботи було розглянуто проблеми автоматизації діяльності співробітників агентств з пошуку нерухомості з використанням сучасних засобів обробки інформації.

В роботі зроблена якісна постановка задачі, створено гнучку онлайн програму. На підставі бізнес правил були розроблені функціональні та нефункціональні вимоги до програмного забезпечення, діаграми варіантів використання ПЗ, модель даних.

Детально описані можливості програмного продукту, обрано цільовий варіант архітектури, було описано обрані технічні засоби. Розроблено програмне забезпечення на платформі Salesforce для задачі автоматизації діяльності співробітників ріелторських агентств – програмне рішення для автоматизації всього процесу пошуку клієнтом нерухомості та надання співробітниками агенції нерухомості під оренду. Реалізовано побудову електронних звітів на підставі існуючих даних.

Область застосування. Розроблене програмне забезпечення може бути використано будь-якими комерційними та некомерційними організаціями, що специфікуються на роботі з нерухомістю та бронюваннями.

Значення роботи та висновки. Програмне забезпечення описане у дипломній роботі надає співробітникам ріелторських компаній комплект інструментів, що автоматизують основні бізнес-процеси підприємства тим самим заощаджуючи час працівників та підвищуючи ефективність їх роботи.

Список ключових слів: АВТОМАТИЗАЦІЯ, АГЕНТСТВО З ПОШУКУ НЕРУХОМОСТІ, SALESFORCE, СИСТЕМА ВЕДЕННЯ ОБЛІКУ, ОПЕРАЦІЙНІ СИСТЕМИ, ПРОГРАМНО-АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ.

## ABSTRACT

The explanatory note to the bachelor's work contains 118 pages, 15 tables, 48 illustrations, 1 appendix, 50 sources.

The subject of research of the diploma project is the means to automate the activities of real estate agencies.

The purpose of the project: Development of mathematical, algorithmic and software module for bookkeeping based on Salesforce CRM-platform.

Methods of research. The following research methods were used in writing the work: analytical, software.

During the implementation of this thesis was considered the problem of automating the activities of employees of real estate agencies using modern information processing tools.

Based on business rules, functional and non-functional software requirements, software usage diagrams, and data model were developed.

The practical value of work is to develop software on the Salesforce platform for the task of automating the activities of employees of real estate agencies - a software solution to automate the entire process of finding real estate by the client and the staff of the real estate agency for rent. The construction of electronic reports based on existing data is implemented.

The scope. The developed software can be used by any commercial and non-commercial organizations that specialize in real estate and reservations.

The value of the work and conclusions. The software described in the thesis provides employees of real estate companies a set of tools that automate the main business processes of the enterprise thereby saving employees time and increasing their efficiency.

Key words list: AUTOMATION, REAL ESTATE SEARCH AGENCY, SALESFORCE, ACCOUNTING SYSTEM, OPERATING SYSTEMS, SOFTWARE AND HARDWARE, CRM.

## ЗМІСТ

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ .....	8
ВСТУП .....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ .....	11
1.1 Коротка характеристика об'єкта управління .....	11
1.2 Опис предметної області .....	12
1.2.1 Призначення CRM-платформи .....	14
1.2.2 Функції та задачі, які повинна вирішувати система для агентства з пошуку нерухомості .....	15
1.3 Огляд і аналіз наявних аналогів, що реалізують функції предметної області .....	16
1.3.1 Сайт «Booking.com» .....	16
1.3.2 Сайт airbnb.com» .....	17
1.3.3 Сайт «Аренда квартир в Харькове» .....	18
1.3.4 Порівняння розглянутих рішень .....	18
2 РОЗРОБКА МАТЕМАТИЧНОГО ТА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ ВИРІШЕННЯ ЗАДАЧІ АВТОМАТИЗАЦІЇ ДІЯЛЬНОСТІ АГЕНТСТВА З ПОШУКУ НЕРУХОМОСТІ .....	19
2.1 Глосарій .....	21
2.2 Розроблення варіантів використання .....	27
2.2.1 Діаграма варіантів використання .....	27
2.2.2 Специфікація варіантів використання .....	31
2.3 Специфікація функціональних та нефункціональних вимог .....	37
2.3.1 Модель IDEF0 .....	37
2.3.2 Модель DFD .....	41
2.4 Нефункціональні вимоги .....	44
2.5 Функціональні вимоги .....	45
2.6 Проектування інтерфейсу користувача .....	46
3 ПРОЄКТНІ ТА ТЕХНІЧНІ РІШЕННЯ .....	49
3.1 Математична (логічна) та фізична моделі БД .....	49
3.1.2 Визначення типів зв'язків .....	54
3.1.3 Визначення атрибутів .....	56
3.1.4 Діаграма класів .....	66
3.1.5 Діаграма діяльності .....	69
3.1.6 Діаграма послідовності дій .....	71
3.2 Обґрунтування вибору засобів розробки .....	73
3.2.1 Apex .....	73
3.2.2 JavaScript .....	74

3.2.3 Мова розмітки гіпертексту HTML та Visualforce .....	74
3.2.4 SOQL .....	75
3.2.5 Lightning Component Framework .....	76
3.2.6 Вибір цільового варіанту архітектури програмного забезпечення .....	76
3.3 Розробка функціоналу програмного продукту .....	78
3.4 Розробка функціоналу входу до системи .....	86
3.5 Тестування програмної системи .....	87
<b>ВИСНОВКИ</b> .....	89
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	90
Додаток А. Лістинг програми для компоненту пошуку .....	94

## ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

CRUD (англ. create read update delete) - 4 базові функції управління даними «створення, зчитування, зміна і видалення».

CRM (англ. Customer relationship management.) - поняття, що охоплює концепції, котрі використовуються компаніями для управління взаємовідносинами зі споживачами, включаючи збір, зберігання й аналіз інформації про споживачів, постачальників, партнерів та інформації про взаємовідносини з ними.

SOQL – (англ. structured query language) – мова структурованих запитів у платформі Salesforce;

ПЗ – програмне забезпечення;

DFD – Data Flow Diagrams;

ПП – програмний продукт;

ПК – персональний комп'ютер;

ТЗ – технічне завдання;

CSS (англ. Cascading Style Sheets) – каскадні таблиці стилів;

HTML (англ. HyperText Markup Language) – мова розмітки гіпертекстових документів;

ОС – операційна система;

СУБД – система управління базами даних;

ІС – інформаційна система;

UML (англ. Unified Modeling Language) – уніфікована мова моделювання;

UI (англ. User Interface) – інтерфейс користувача;

UX (англ. User Experience) – досвід користування;

LCF (англ. Lightning component framework) – фреймворк для роботи з Salesforce;

USER – клієнт, користувач системи.

## ВСТУП

Об'єктом дослідження дипломного проєкту є система для автоматизації ведення обліку діяльності агентств з нерухомості. Метою роботи є розробка алгоритмічного та програмного забезпечення для вирішення задачі автоматизації ведення обліку діяльності співробітників та клієнтів агентств зі здачі в оренду різних видів нерухомості.

У ході виконання даної дипломної роботи було розглянуто проблеми автоматизації діяльності подібного типу компаній з використанням сучасних засобів обробки інформації. На підставі бізнес правил були розроблені функціональні та нефункціональні вимоги до програмного забезпечення, діаграми варіантів використання ПЗ, модель даних. Детально описані можливості програмного продукту, обрано цільовий варіант архітектури, було розглянуто обрані технічні засоби.

Програмне забезпечення для вирішення задачі автоматизації діяльності агентств з пошуку апартаментів реалізовано на базі CRM платформи Salesforce – програмне рішення призначене для автоматизації всього процесу пошуку клієнтом апартаментів. Буде реалізована побудова графічних звітів прогнозування кількості створених та сплачених клієнтами бронювань та побудова графіків на підставі існуючих даних та звітах.

Виконання дипломного проєкту передбачає під собою розробку web-застосування «Служба управління та ведення обліку бронювань». Якісне виконання дипломного проєкту є важливим етапом вивчення спеціальності, на якому студент має показати свою здатність застосовувати вивчений теоретичний матеріал на практиці.

Розроблене прикладне програмне забезпечення має два режими: адміністратор та User. Клієнт має можливість переглядати, сортувати, шукати потрібну йому інформацію про наявні апартаменти компанії та оформлювати бронювання. Окремі права є у адміністратора: він може переглядати інформацію про бронювання, змінювати їх статус чи видаляти. Також завдяки своїм привілеям адміністратор може видаляти та додавати інформацію про поставників, типи апартаментів, створювати контракти та платіжні вимоги, вносити інформацію щодо рахунків підприємства. Важливим аспектом у роботі системи є те, що адміністратор має можливість вести облік прибутку компанії за допомогою електронної звітності.

З позиції сучасності автоматизація – це один з напрямків науково-технічного прогресу, спрямований на застосування саморегульованих технічних засобів,

автоматичних й автоматизованих систем організації будь-якої діяльності. Зростання обсягів оброблюваних даних висуває на перший план проблему ефективності засобів організації обробки даних і доступу до них.

Оскільки агентство з нерухомості зазвичай працює з великою кількістю людей, то до нього, як і для будь-якого підприємства є актуальними проблеми обробки великої кількості інформації. Тому в даний час особлива увага приділяється впровадженню нових інформаційних технологій з обробки даних та автоматизації. Цілком очевидно, що в діяльності будь-якої організації є багато рутинної роботи, в тому числі і в ріелторському агентстві.

Зберігання одних тільки документів в електронному вигляді на диску або у хмарі, безумовно, більш доцільно, ніж зберігання їх у вигляді паперів. Ефективна організація управління та автоматизації діяльності співробітників цих агентств — непросте завдання. Система повинна найбільш повно автоматизувати рутинні дії для полегшення роботи робітників та зручності для клієнтів. Таким чином, висновки щодо актуальності створення даного програмного продукту були зроблені після аналізу предметної області.

Проєкт розробляється на базі платформи Salesforce з використанням мов HTML5, CSS3, JavaScript, SOQL та APEX. Робота складається з вступу, трьох розділів і висновків. Дипломна робота Містить 118 сторінок друкованого тексту, в тому числі 89 сторінок тексту основної частини з 48 рисунками, список з 50 використаних джерел з найменуваннями на 4 сторінках, один додаток на 20 сторінках.



# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

## 1.1 Коротка характеристика об'єкта управління

Програмний продукт що розроблявся в ході дипломного проектування призначений для підприємств які займаються пошуком вільних приміщень на ринку нерухомості з метою подальшої їх здачі в оренду, наданням житла для бронювання. Загалом схему роботи компаній такого типу можна описати наступним чином:

- компанія знаходить фізичних чи юридичних осіб, тобто поставників, які бажають надати вільне приміщення під оренду;
- вкладається договір на отримання прав компанією розпоряджатися майном поставника;
- приміщення стає доступним для бронювання орендаторами, тобто потенційними клієнтами компанії;
- клієнти бронюють вподобане житло після чого компанія складає з ними договір на користування приміщенням;
- за рахунок різниці у ціноутворенні для угод між поставниками та клієнтами ріелтори отримують прибуток;

Для того щоб уникнути кризових явищ, топ-менеджмент має здійснювати ефективне керівництво процесом розвитку підприємства, насамперед через систему управління, створюючи ефективну організаційну структуру. Організаційна структура управління підприємством є засобом сприяння досягнення менеджером своїх цілей. Оскільки цілі є похідними від загальної стратегії підприємства, тісний зв'язок стратегії та структури цілком логічний. Відповідно організаційна структура має дотримуватися певної стратегії. І якщо менеджери здійснюють вагомі зміни у своїй організаційній стратегії, вони мають модифікувати організаційну структуру задля пристосування до цих змін та їх підтримання. Відповідно до організаційної структури підприємства розробляється система управління підприємством.

Результатом розроблення схеми організаційної структури управління підприємством є наступний опис основних структурних елементів:

**КЕРІВНИК** – голова організації, координує роботу менеджерів з фінансовими контрактами, офіційний представник компанії;

**МЕНЕДЖЕР** – підпорядковується керівникові, контролює роботу адміністратора, працює з клієнтами, займається оформленням контрактів, працює над налагодженням відносин з клієнтами компанії, складає фінансові звіти.

АДМІНІСТРАТОР – слідкує за актуальністю об'єктів нерухомості, що здаються в оренду компанією, координує процес бронювання нерухомості, спілкується з клієнтами у разі виникнення конфліктів, звітує менеджеріві.

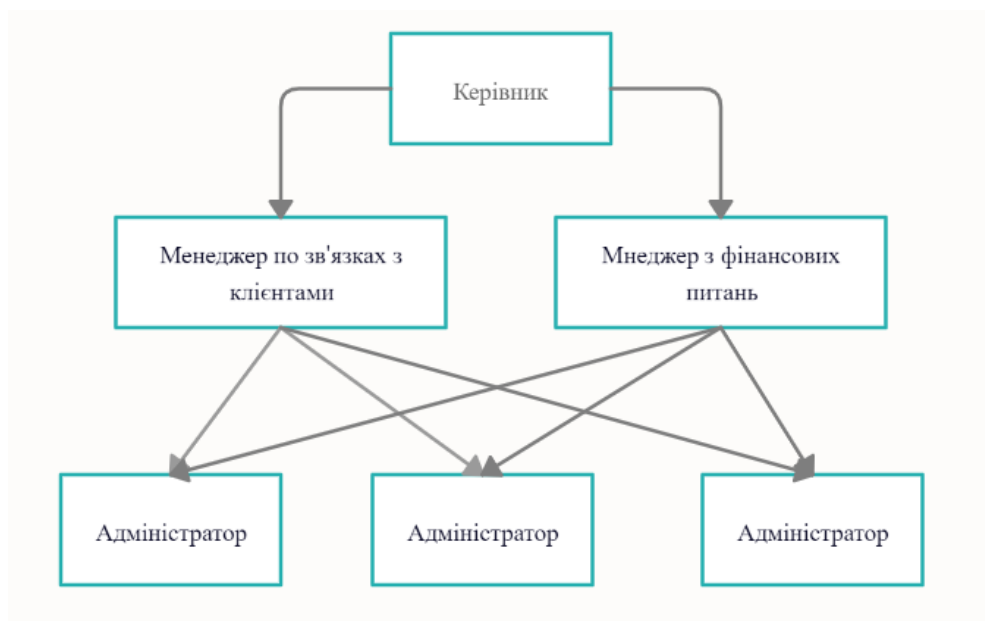


Рисунок 1.1 – Схема організаційної структури компанії

Таким чином, можна зробити висновок, що під час повсякденної роботи організації даного типу усі бізнес процеси є взаємопов'язаними: від пошуку потенційних поставників, які надають приміщення для оренди; розміщення пропозицій для потенційних клієнтів; бронювання апартаментів; до оформлення платіжної вимоги і укладання квартальної звітності.

Ґрунтуючись на цьому, у розроблюваному дипломному проєкті було створено електронну систему, що зосереджує у собі можливості автоматизації більшості вищеперерахованих процесів для всіх структурних елементів системи.

Даний програмний продукт буде корисний керівникам, які зможуть наочно переглянути результати роботи персоналу своєї компанії. Менеджерам та адміністраторам буде надано можливість дистанційно вести облік наявних апартаментів, співпрацювати з електронними документами, поставниками та клієнтами, генерувати електронні звіти та здійснювати модерацію створених клієнтами букінгів.

## 1.2 Опис предметної області

Сфера «online-пошуку» як соціальне явище весь час знаходиться в динаміці. На сучасному етапі розвитку суспільства нові можливості використання і передачі

інформації, вдосконалення відносин у діловому житті відносяться до ряду глобальних явищ.

Об'єктивною є необхідність вдосконалення методів роботи агентств з пошуку нерухомості. Система для компаній даного типу повинна охоплювати: легкий та зручний пошук і управління бізнес процесами; ефективне управління часом співробітників; швидку комунікацію клієнтів і агентів; поліпшення та полегшення процесу роботи користувачів з електронним додатком. Для досягнення поставленої мети і відповідності результату сучасним технологіям процес взаємодії користувачів з системою має бути автоматизовано. Здійснений аналіз існуючого ринку додатків, дає змогу дійти висновку, що найбільш доцільним є одночасна автоматизація декількох сфер діяльності підприємства [36].

Так, наприклад, звітність є одним з найважливіших бізнес процесів у будь-якій компанії. Звітність являє собою функцію управління. Сутність цього процесу полягає в логічному визначенні розвитку підприємства, постановці цілей для будь-якого сектору його діяльності, що необхідно в умовах сучасної конкуренції. При здійсненні ведення обліку ставляться задачі, визначаються матеріальні, трудові і фінансові кошти для їх досягнення та терміни виконання, а також послідовність їх реалізації. Оцінка досягнень компанії являє собою етап, від якого багато в чому залежить розвиток підприємства. Великий вплив на планування надає грамотність керівництва, кваліфікація фахівців, залучених до цього процесу, достатність ресурсів, необхідних для здійснення процесу (комп'ютерна техніка та ін.), інформативна база.

Не менш важливішим є планування робочого дня співробітників компанії. В даний час ефективність роботи організації багато в чому визначається ефективністю повсякденної роботи персоналу управління. Планування та управління робочим часом дозволяє вирішити наступні задачі:

- 1) підвищення рівня ефективності діяльності співробітників;
- 2) контроль дій і витрат часу на них;
- 3) оптимізація робочого графіку;
- 4) відсів неефективних видів діяльності;
- 5) раціональне планування на виконання найважливіших справ.

У підсумку, можна сказати, що управління бронюваннями та ведення обліку роботи як функції для автоматизації управління означають прагнення враховувати завчасно все зовнішні і внутрішні чинники, які забезпечують сприятливі умови для нормального функціонування і розвитку підприємства.

Дане ПЗ розробляється для інформаційної та аналітичної підтримки діяльності агентства з пошуку та найму нерухомості. Користувачами цієї системи мають бути клієнти, які шукають житло, та менеджери і адміністратори системи.

Програмне забезпечення, що імплементується, містить в своїй основі систему розроблену на платформі світового лідера з надання CRM платформи - Salesforce.

Тож розглянуте у даній дипломній роботі ПЗ користується системою захисту та безпеки компанії з світовим досвідом та визнанням однойменної компанії Salesforce.

Також програмне забезпечення має мобільний додаток з простим але зручним інтерфейсом. У даній дипломній роботі представлено технічне обґрунтування проєкту по розробці ПЗ для автоматизації діяльності агентства з найму та здачі нерухомості.

### 1.2.1 Призначення CRM-платформи

Оскільки одним з основних програмних засобів, що використовуються в даному проєкті є CRM система важливо описати ключові її характеристики, а також основну концепцію роботи з наявним програмним комплексом системи Salesforce.

У сучасному стрімко прогресуючому світі інформаційних систем та технологій величезна кількість підприємств, компаній, та приватних установ надає перевагу ведення бізнесу в електронному просторі. Для поліпшення роботи бізнес установ в електронному просторі було розроблено так звані CRM платформи, що містять у собі вбудовані засоби для автоматизації різних бізнес процесів. Сучасний бізнес вимагає автоматизації різних процесів, що суттєво спрощує роботу. Так, вже складно уявити компанію без налаштованого складського або бухобліку, а також використання спеціального програмного забезпечення.

Тобто, за замовченням система надає користувачам базовий набір різноманітного функціоналу. Вона повністю візьме на себе виконання рутинної роботи, яка зазвичай забирає багато часу у співробітників. Після впровадження менеджери зможуть більше часу приділяти своїм безпосереднім обов'язкам, а система буде займатися:

- веденням клієнтської бази;
- створенням історії взаємовідносин з клієнтами;
- швидким оформленням необхідних документів;
- організацією, плануванням роботи з клієнтами;

- розсиланням рекламних матеріалів з вигідними пропозиціями для покупців;
- створенням звітів про виконану роботу;

Вбудований функціонал у подальшому можна розширити за допомогою реалізації окремих програмних блоків, сторінок, сервісів і т.д. Саме таку можливість і надає своїм клієнтам Salesforce, адже з використанням мови програмування та розмітки електронних сторінок можна налаштувати програмний комплекс саме під вимоги кожної окремої організації.

1.2.2 Функції та задачі, які повинна вирішувати система для агентства з пошуку нерухомості

Для того, що виділити, які функції та задачі повинна вирішувати система, необхідно спочатку сформувавши перелік основних функцій та завдань які стоять перед співробітниками агентства з пошуку нерухомості. Автоматизація усієї діяльності є дуже великою та комплексною задачею, тому доцільно буде спочатку автоматизувати роботу функцій, які клієнт буде використовувати самостійно:

- Клієнт на зовнішньому сайті (призначений лише для клієнтів) може за допомогою фільтрів та пошуку знайти відповідні позиції, вибрати їх, заповнити форму з особистими даними та відправити її на сервер для обробки. Фільтри уявляють собою багатокритеріальний пошук.
- Система повинна завантажувати, перевіряти усі нові заяви на бронювання та зберігати їх у БД.
- Адміністратори системи повинні мати можливість переглядати та редагування заяви на бронювання, які були створені користувачами.
- Адміністратори системи повинні мати можливість заповнювати БД інформацією щодо наявної нерухомості, а безпосередньо система буде автоматично вивантажувати нові пропозиції на віддалений сервіс.
- Адміністратори також повинні мати можливість вносити в систему інформацію про платіжні вимоги, оплату платіжних вимог та на основі цих даних генерувати електронні звіти.
- Система повинна автоматично вивантажувати нові позиції житла, створені адміністратором, на компонент доступний для клієнтів.
- Система повинна здійснювати валідацію створюваних записів та виводити повідомлення для користувачів у разі виникнення помилок.
- Однією з вимог до системи є автоматична відправка email-підтверджень клієнтам щодо створених ними бронювань.
- Важливою вимогою до системи є генерація «PDF» версій електронних документів чи платіжних вимог, що зберігаються та генеруються системою.

- Адміністратор повинен мати можливість завантажувати у систему файли з локального ПК та подальшого використання

- Адміністратор повинен мати можливість створювати та запускати звіти щодо фінансових показників системи. Звіти можуть бути скомпоновані у dashboards за бажанням користувача.

Відносно безпеки даних та її доступності – у системі кожен користувач має мати можливість проглядати лише дані, які йому дозволено відповідно до вимог безпеки. Також система повинна автоматизовано займатися обробкою та збором інформації та статистики, створенням звітів.

1.3 Огляд і аналіз наявних аналогів, що реалізують функції предметної області

#### 1.3.1 Сайт «Booking.com»

Система, яка має в собі стандартні функції з пошуку та бронювання житла. Кожен користувач має можливість швидко знайти позиції, які підпадають під його вимоги. Орендодавці мають можливість додати свої пропозиції до VIP розділу для того, щоб ці апартаменти частіше були запропоновані клієнтам. Сервіс дозволяє орендодавцям зробити свої позиції більш популярними шляхом відображення їх на головній сторінці пошуку. Система також має ряд додаткових функцій:

- перегляд мапи з усім доступним житлом конкретного міста;
- детальна класифікація апартаментів;
- можливість отримати консультацію адміністратора ресурсу з найпоширеніших питань;
- відповідні нормативні документи, що регулюють роботу інтернет-ресурсу;
- загальна інформація про правила та умови;
- можливість залишити відгук щодо наданих послуг;

Інтерфейс сайту «Booking.com» зображено на рисунку 1.2. Знайти сайт можна за посиланням [https://booking.com /URL](https://booking.com/URL) [35].

The screenshot displays the Booking.com website interface. At the top, there is a navigation bar with the Booking.com logo, the Ukrainian flag, and buttons for 'Зареєструвати своє помешкання', 'Зареєструватися', and 'Увійти в акаунт'. Below this is a menu with icons for 'Помешкання', 'Переліт', 'Оренда автомобілів', 'Дозвілля', and 'Таксі з/до аеропорту'. A yellow banner contains a message about COVID-19 and a link to 'Зв'яжіться з адміністрацією заброньованого помешкання'. The main search area is on the left, with a yellow background, containing filters for location (Kyiv), dates (Monday, June 1, 2020 to Wednesday, June 30, 2020), number of nights (29), and number of guests (2 adults). The search results on the right show a list of accommodations, with the first one being 'Апартаменти BeRest Inn' in Solomianskyi, Kyiv, priced at UAH 23,600. The interface includes a 'На карті' button and a 'Переглянути наявність місць' button.

Рисунок 1.2 – Інтерфейс системи «Booking.com»

### 1.3.2 Сайт «airbnb.com»

Сайт [airbnb.com](https://www.airbnb.com/) є одним з найбільш популярних сайтів пошуку нерухомості в Україні і світі [41]. Доступні можливості:

- багатокритеріальний пошук;
- через велику популярність, сайт дає можливість вибирати місто, в якому ви хотіли б знайти житло;
- система надає статистику найпопулярніших позицій і власників з найбільшою кількістю позитивних відгуків ;
- після кількох переглянутих вакансій система "розумно" підбирає вам схожі пропозиції;
- структурована система оцінки житла за декількома основними характеристиками:
- наявний мобільний додаток;

Знайти сайт можна за наступним посиланням: <https://ru.airbnb.com/>. Інтерфейс системи представлено на рисунку 1.3.

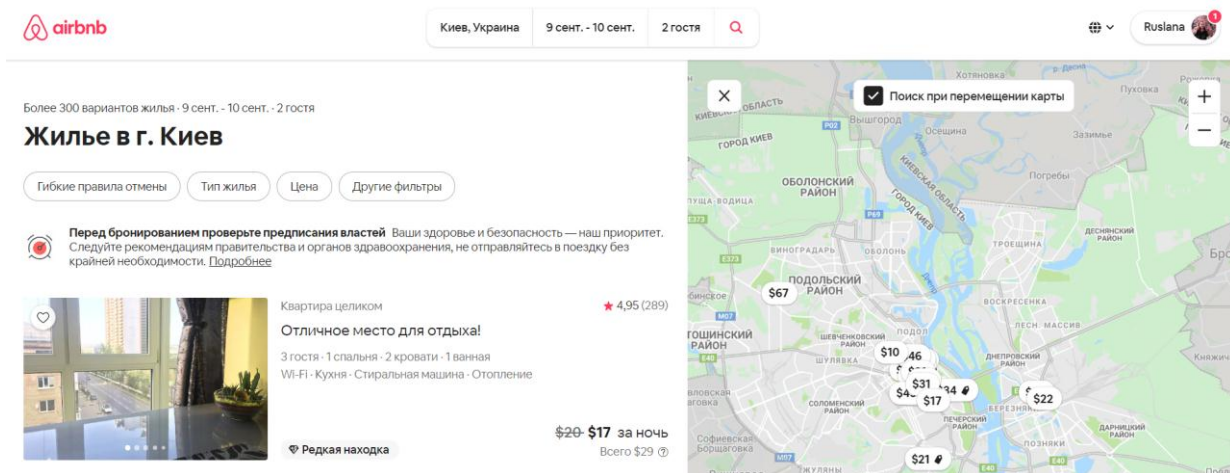


Рисунок 1.3 – Интерфейс системы «airbnb.com»

### 1.3.3 Сайт «Аренда квартир в Харькове»

<https://real-estate-agency-kharkiv.business.site/> - типовий міський сайт з пошуку житла. Має функцію перегляду наявних апартаментів, можна прокласти маршрут до вподобаного житла та замовити зворотній дзвінок з приводу якогось питання, зокрема вартості оренди.

Интерфейс системы представлено на рисунку 1.4. Знайти сайт можна за посиланням: <https://real-estate-agency-kharkiv.business.site/> [38].

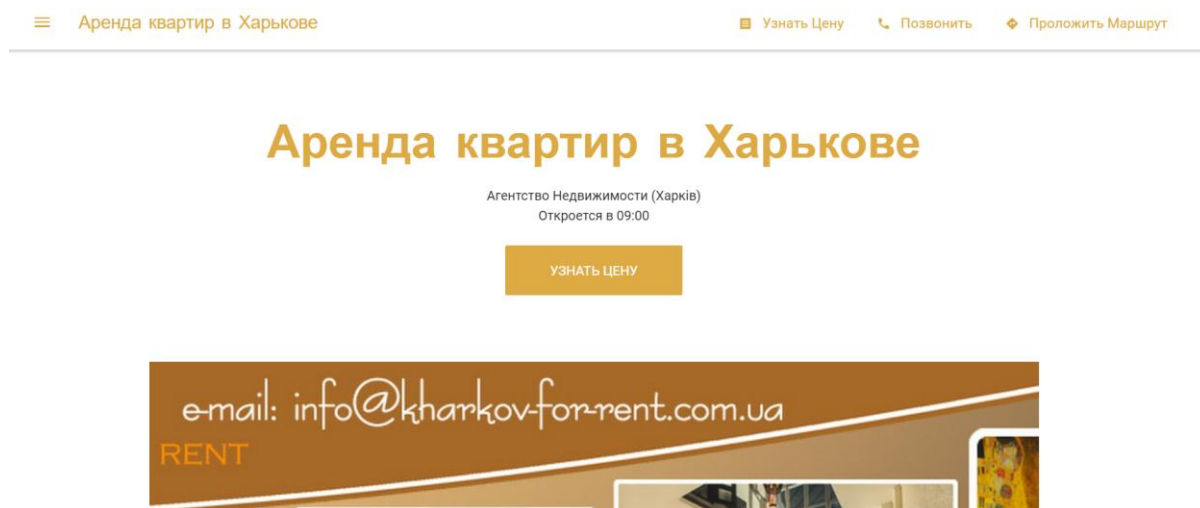


Рисунок 1.4 – Интерфейс системы «Аренда квартир в Харькове»

### 1.3.4 Порівняння розглянутих рішень

У таблиці 1.1 представлено порівняльний аналіз розглянутих систем, що дозволяють автоматизувати роботу агентств з пошуку нерухомості. У якості продуктів для порівняння при розробці програмної компоненти було взято декілька систем: booking.com, airbnb.com та real-estate-agency-kharkiv.business.site.



Ці програмні додатки були обрані у якості аналогів на основі таких факторів: схожий профіль; відповідність вимогам технічного завдання проекту; доступність для дослідження аналогу у зв'язку з його відкритим доступом.

Таблиця 1.1 – Порівняльний аналіз розглянутих рішень

Категорія	Сайт «Booking.com»	Сайт «airbnb.com»	Сайт «Аренда квартир в Харькове»
1	2	3	4
Функція сортування за вартістю житла	Присутнє	Присутнє	Відсутнє
Функція пошуку за розташуванням	Присутнє	Присутнє	Присутнє
Наявність додаткової інформації відносно пошуку житла	Є статті, статистика та зразки популярних пропозицій	Перелік популярних позицій, відгуки	Google-відгуки
Наявність мобільного додатку	Є	Є	Відсутнє
Платність	Так, до вартості бронювання входять податки	Так, у вартість бронювання входить збір за послуги сайту	Ні
Наявність обробки статистики	Є	Є	Відсутнє
Можливість залишать відгуки	Присутнє	Присутнє	Присутнє

Після аналізу вимог що до процесів, які система має автоматизувати можна зробити висновок, що системою користуються декілька типів користувачів і потрібно автоматизувати процеси кожного з них.

Основною характерною рисою розроблюваного ПП є те, що функціонування системи допомагає збирати статистику, на основі якої можливо виконати аналіз кількості отриманих запитів на бронювання.

Підсумовуючи перегляд та оцінку приведених вище систем, можна відзначити, що всі вони мають наступні характеристики: це є система, яка містить в собі стандартні функції з пошуку житла та подання заяви на його бронювання. Кожен користувач має можливість швидко знайти апартаменти, які підпадають під його вимоги. Орендодавці мають можливість додати свої позиції до VIP розділу для того, щоб ці апартаменти частіше були запропоновані клієнтам. Клієнти у свою чергу мають можливість якісно та кількісно оцінити окремі апартаменти. Інтерфейс програм не завжди є інтуїтивно зрозумілим та доступним новачку.

Порівняно з представленими аналогами вагомим плюсом є те, що описане у даній роботі ПЗ - це цілий комплекс для автоматизації бізнес-процесів. Адже, співробітники агенції зі здачі житла чи просто власники нерухомості можуть відстежити весь процес бронювання апартаментів, згенерувати звітність для ефективного відображення фінансових показників.

Розроблюваний продукт виступає як єдина та цілісна система, що буде зберігати у собі повний об'єм інформації, що пов'язана з бронюванням: починаючи від даних про постачальника і закінчуючи сплаченим рахунком за бронювання.

На відміну від багатьох існуючих онлайн-ресурсів, що надають різні можливості переважно своїм клієнтам, розроблювана система орієнтована саме на ріелторські агенції та компанії, що матимуть змогу вести облік своєї діяльності використовуючи єдину онлайн платформу, яка зосередить у собі всю важливу інформація щодо основних бізнес процесів підприємства.

Основною метою даного проєкту є концепція єдності, аби вся необхідна для роботи інформація зберігалася на єдиному ресурсі і користувачі оброблювали її за допомогою єдиного додатку.

З одного боку, використання даної системи надасть змогу всім співробітникам та клієнтам компаній швидко та зручно вести свою діяльність на спільній платформі, а з іншого боку автоматизація основних процесів полегшить процес роботи за оцінки економічних та статистичних результатів.

## 2 РОЗРОБКА МАТЕМАТИЧНОГО ТА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ ВИРІШЕННЯ ЗАДАЧІ АВТОМАТИЗАЦІЇ ДІЯЛЬНОСТІ АГЕНТСТВА З ПОШУКУ НЕРУХОМОСТІ

Кожна розробка програмного модулю передбачає також створення специфікації вимог, до якої входять глосарій проекту та діаграма варіантів використання. Метою такого проектування є надання усіх деталей вимог до застосунку задля спрощення процесу розробки та тестування програмного продукту. В цьому розділі також будуть визначені функціональні та нефункціональні вимоги.

Глосарій – це словник основних використовуваних термінів. Цей документ є найпершим результатом концептуального аналізу предметної області [7]. Глосарій можна розглядати як документ, що засвідчує спільне розуміння основної термінології Замовником і Розробником. Мета створення глосарія – покращити розуміння термінів та поглибити сприйняття функцій проекту. Саме на ньому лежить основа побудови більш розгорнутих моделей предметної області. Глосарій до даного дипломного проекту поданий у табл. 2.1.

Таблиця 2.1 – Глосарій

Термін	Опис терміну
1	2
1. Основні поняття і категорії предметної області та проекту	
Credentials	логін та пароль користувача для авторизації у системі;
Посилання, гіперпосилання, лінк	вказівка порталу здійснити перехід на головну сторінку будь-якого розділу, або на одну з внутрішніх;
Контент	інформаційний зміст сайту (тексти, графічна, звукова інформація та ін.);
Відправка документів	можливість автоматично відправити наявний у системі документ на пошту клієнту;
Апартаменти	запис об'єкту БД, що представляє собою інформацію щодо житла, його опис та характеристику;

## Продовження таблиці 2.1

1	2
1. Основні поняття і категорії предметної області та проекту	
Локалізація	переклад призначеного для користувача інтерфейсу на потрібну мову;
Популярні апартаменти	незаміщені житлові пропозиції, відсортовані за датами та кількістю вільного місця;
Дашборд/Dashboard	це наочне представлення інформації у вигляді декількох електронних звітів про поточний прибуток підприємства, зосереджене на електронній дошці чи календарі;
Підтверджене бронювання	букінг, який було схвалено системою та адміністратором;
Бронювання/Booking	форма заповнена клієнтом компанії, котра містить у собі інформацію щодо бронювання нерухомості, а саме: кількість гостей, кількість кімнат, дати заїзду і виїзду, коментарі і т.д.;
Вподобані записи	апартаменти, на яких була поставлена позначка «Мені подобається»;
Сток/Stock	об'єкт БД, що містить у собі інформацію про наявність вільного місця у певних апартаментах;
Табка/Tab	спеціальні вкладки що дозволяють відображати об'єктні дані чи інший веб-вміст у Salesforce. Відображається як елемент на панелі навігації програми;
Setup	головне меню налаштування Salesforce організації;

## Продовження таблиці 2.1

1	2
1. Основні поняття і категорії предметної області та проєкту	
Detail page	системна сторінка окремого запису сутності у Salesforce;
Блокування користувача	заборона адміністратором системи взаємодіяти іншому користувачеві з системою;
Інтерфейс	засіб зручної взаємодії користувача з інформаційною системою. Сукупність засобів для обробки та відбиття інформації, якнайбільше пристосованих для зручності користувача, тобто елементи сторінок порталу та їх розташування;
Іконка	(письмовий знак, риска, лінія), значок (зменшене від знак) — елемент графічного інтерфейсу, невелике зображення, що репрезентує застосунок, файл, теку, вікно, компонент ОС, пристрій тощо;
Портал, організація	комплекс апаратно-програмних засобів, що дозволяє користувачам комп'ютерної мережі Інтернет обмінюватися повідомленнями за темами;

## Продовження таблиці 2.1

1	2
1. Основні поняття і категорії предметної області та проєкту	
Ідентифікатор користувача (логін)	це ім'я користувача, яке буде відображатися тільки на сайті. Логін є псевдонімом та не має обов'язково повторювати справжнє ім'я людини;
Ідентифікаційний номер/Id	унікальний набір літер та цифр що закріплюється системою за кожним записом у системі, для подальшого використання адміністратором;
Авторизація	керування рівнями та засобами доступу до різних об'єктів (наприклад, автоматизована система контролю доступу) та ресурсів системи залежно від ідентифікатора і пароля користувача;
Service Type	об'єкт БД призначений для зберігання записів про наявну нерухомість у Поставника, її місткість;
Visualforce page	сторінки Visualforce - це основні будівельні блоки для розробників додатків Salesforce. Сторінка Visualforce схожа на звичайну веб-сторінку, але включає потужні функції доступу, відображення та оновлення даних вашої організації. Visualforce використовує мову розмітки на основі тегів, аналогічну HTML;

Продовження таблиці 2.1

1	2
1. Основні поняття і категорії предметної області та проєкту	
Lightning Component	це фреймворк для розробки інтерфейсу односторінкових програм для мобільних та настільних пристроїв;
Process Builder/ Конструктор процесів	допомагає автоматизувати бізнес-процеси, такі як оновлення записів, оповіщення електронною поштою та будь-які завдання, які призначить розробник і все це без єдиного рядка коду;
Головна сторінка/Home page	початкова сторінка порталу, яка, зазвичай, надає відомості про тематику веб-сайту та матеріали, які можна побачити на подальших сторінках (дозволяє переглянути зміст та меню порталу);
Затверджені заяви на бронювання	заяви, які створені користувачем, та вже затверджені системою на адміністратором;
Поставник	безпосередній власник нерухомості, інформація про якого зберігається у системному об'єкті Account;
Web-service	технологія, що реалізує автоматичне завантаження набору даних між різними організаціями у Salesforce;
Сутність/Об'єкт БД	визначена структура об'єкту для подальшого створення, редагування та видалення записів даного типу у системі Salesforce. Кожен об'єкт має свої поля різних типів;

Продовження таблиці 2.1

1	2
1. Основні поняття і категорії предметної області та проекту	
Record Type	типи записів, які дозволяють пропонувати різним користувачам різні бізнес-процеси, значення списку вибору та макети сторінок;
2. Користувачі системи	
Адміністратор порталу	користувач, який керує всім наповненням порталу і його контентом;
Юзер	клієнт/користувач портал, інформація про якого зберігається у системному об'єкті User;
Користувачі/Клієнти	всі користувачі порталу, які мають доступ до функціоналу системи;
3. Вхідні та вихідні документи	
Електронний документ	електронний документ, завантажений до порталу, що зберігається на ПК чи у хмарі, інформація в якому зафіксована у вигляді електронних даних;
Платіжна вимога/ Invoice	рахунок який має сплатити клієнт за бронювання;
Платіж/Payment	запис у системі, що є підтвердженням оплати бронювання;
Шаблон/Template	«PDF» версія запису, яка генерується за заздалегідь визначеною формою;
Вхідні повідомлення	підтвердження бронювань, які були отримані користувачем по електронній пошті;



## Закінчення таблиці 2.1

1	2
3. Вхідні та вихідні документи	
Відправлені повідомлення	підтвердження бронювань, у вигляді email, які були відправлені користувачем по електронній пошті, у чаті, або на форумі;
Форма бронювання/форма для заповнення	форма створення запису букінгу апартаментів, електронна форма для введення необхідних параметрів замовлення;
Звіт про дохід агенції	електронний документ чи таблиця, що містить інформацію щодо оцінки темпів росту чи падіння прибутків;
Контракт	запис у системі, який створюється адміністратором і містить у собі інформацію про ціни на апартаменти на певний період їх здачі;

## 2.2. Розроблення варіантів використання

UML – уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML моделлю.

UML було створено для визначення, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація, наприклад Rational Rose [27]. Це мова для візуалізації, специфікації, конструювання. Тому наступним етапом для створення курсового проекту є розробка UML - діаграм.

## 2.2.1. Діаграма варіантів використання

Перш за все, даному розділі дипломного проекту, потрібно було зробити діаграму варіантів використання (англ. use case diagram), яка описує функціональне призначення системи або, іншими словами, те, що система буде робити в процесі свого функціонування. Діаграма варіантів використання є

графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проєктована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (англ. use-case) використовують для описання послуг, які система надає актору.

Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

Оскільки, в даному web-застосунку можливості для користувачів з різними правами доступу відрізняються, окремо були прописані різні варіанти використання для двох акторів: клієнта та адміністратора.

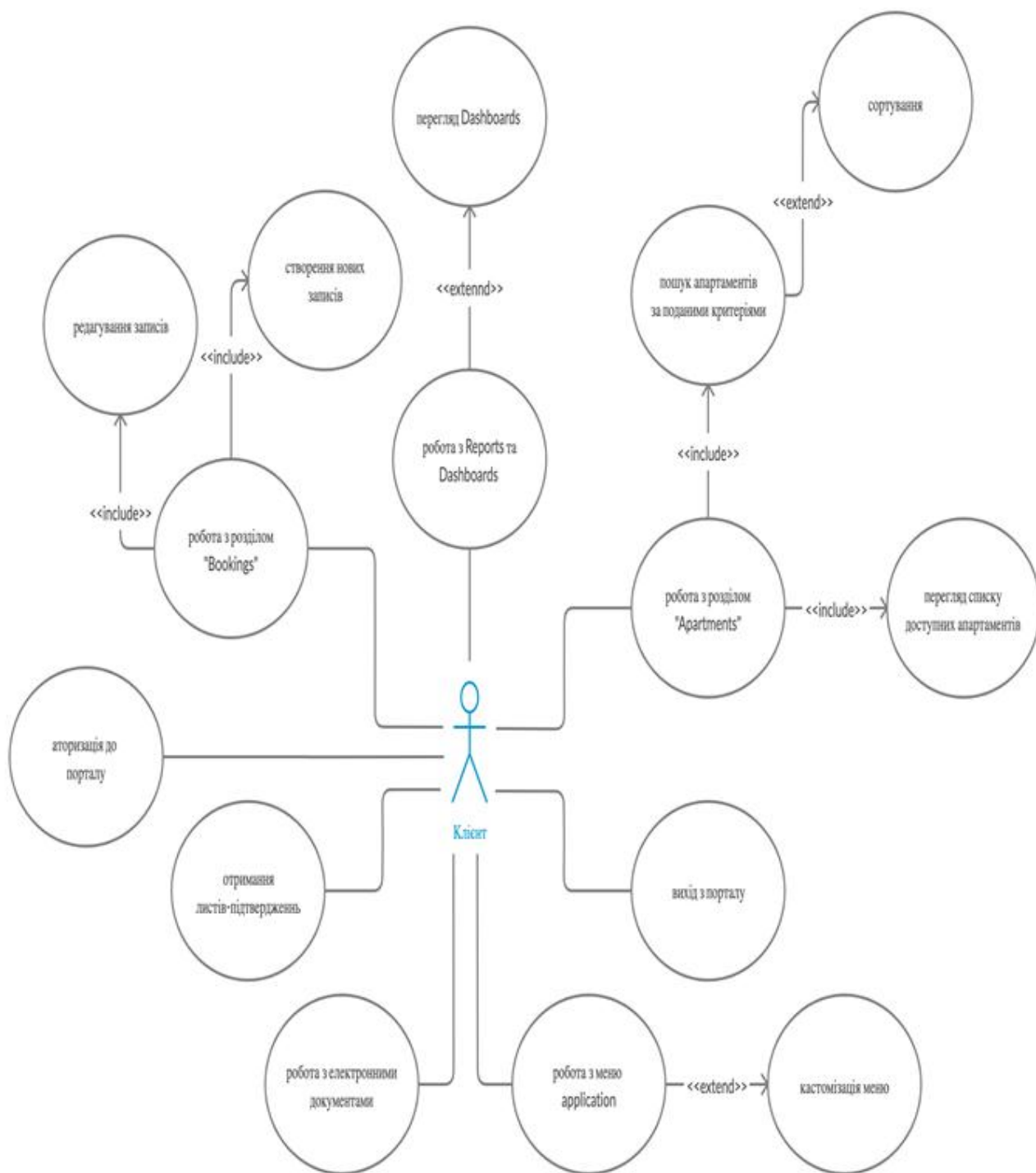


Рисунок 2.1 – Діаграма варіантів використання для Клієнта порталу

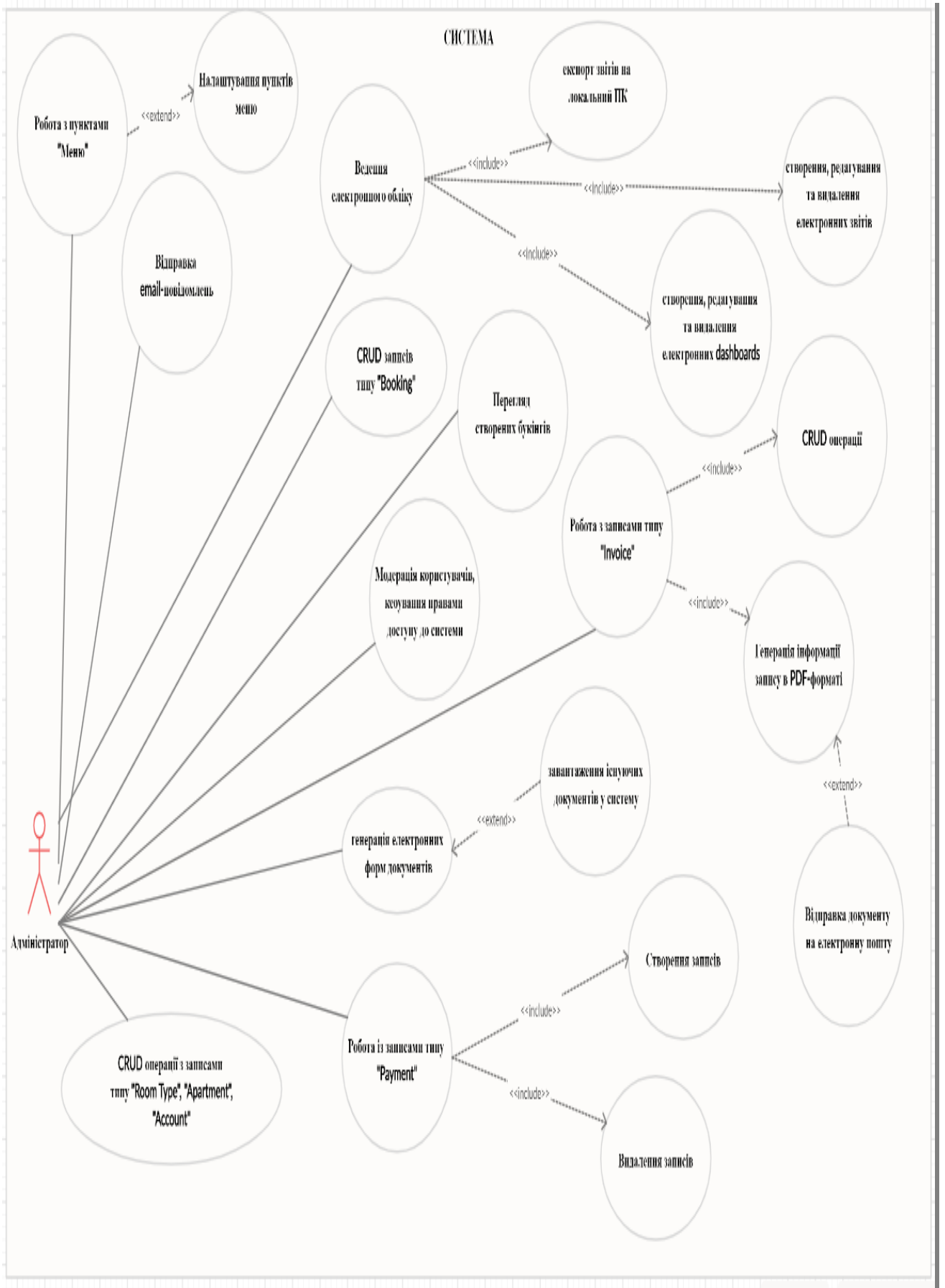


Рисунок 2.2 – Діаграма варіантів використання для Адміністратора

## 2.2.2. Специфікація варіантів використання

Таблиця 2.2 – Функція «Керування правами доступу»

Назва варіанту використання	Керування правами доступу користувачів до порталу.
Дійові особи	Системний адміністратор
Стислий опис	Надає адміністраторові можливість змінювати ролі користувачів порталу, а також блокувати доступ клієнтів до порталу.
Передумова	Необхідно авторизуватися як адміністратор, вибрати пункт меню «Setup» та розділ «User».
Основний потік подій	Так як кожний клієнт порталу має свій унікальний ідентифікаційний номер адміністратору необхідно спочатку авторизуватися, вибрати пункт меню «User» і у списку всіх користувачів та їх ідентифікаційних номерів за необхідністю змінити роль клієнта у порталі чи заблокувати йому доступ до порталу.
Альтернативний потік подій	Якщо, у разі негараздів, з'єднання з сервером неможливо, то виводиться відповідне повідомлення, налаштування не збережуться.
Постумови	Виконання даного варіанту використання означає, що користувач с конкретним ідентифікаційним номером не матиме можливість зайти до порталу, або його права доступу та роль буде змінено.

Таблиця 2.3 – Функція «Робота з розділом Букінги»

Назва варіанту використання	Робота з розділом «Букінги», подача форми на бронювання.
Дійові особи	Системний адміністратор, Клієнт

## Продовження таблиці 2.3

Стислий опис	Надає користувачеві можливість відправляти заяву на бронювання потрібних йому апартаментів.
Назва варіанту використання	Робота з розділом «Букінги», подача форми на бронювання.
Передумова	Необхідно авторизуватися у системі, вибрати пункт меню «Нерухомість».
Основний потік подій	<p>Необхідно авторизуватися, вибрати пункт меню «Нерухомість», далі користувач матиме змогу здійснювати пошук чи сортування апартаментів за потрібними йому параметрами.</p> <p>Після того як користувач визначився з записом житла за допомогою посилання він може перейти на детальну сторінку вподобаних апартаментів та натиснути кнопку «Забронювати». Система покаже форму, яку необхідно заповнити інформацією щодо проживання далі юзер має натиснути кнопку «Зберегти». Після цього, система автоматично розрахує вартість бронювання виходячи з введених користувачем параметрів, перевірить чи є вільне місце на обрані дати та чи відповідає місткість апартаментів бажаній кількості гостей і т.д.</p>
Альтернативний потік подій	Якщо з'єднання з сервером неможливо, то виводиться відповідне повідомлення, додане бронювання не буде збережено у системі, клієнт також може натиснути кнопку «Cancel» і запис також не буде збережено.
Постумови	Запис бронювання автоматично з'являється у розділі «Букінги» у клієнта та адміністратора, для можливості його подальшого редагування. Після валідації бронювання клієнтові на пошту автоматично відправиться лист-підтвердження.

Таблиця 2.4 – Функція «Налаштування меню»

Назва варіанту використання	Налаштування меню.
Дійові особи	Адміністратор, Клієнт
Стислий опис	Надає користувачеві можливість змінити послідовність відображення пунктів меню.
Передумова	Необхідно авторизуватися і перейти до домашньої сторінки порталу, далі вибрати пункт "Налаштування" натиснувши на іконку олівця в крайньому правому куті меню.
Основний потік подій	Після відкриття відповідної сторінки налаштувань порталу користувач має можливість додавати і приховувати необхідні пункти меню.
Альтернативний потік подій	Користувачеві виводиться відповідне повідомлення у разі виникнення проблем з'єднанням із сервером, інтерфейс меню не буде змінено на обраний.
Постумови	Результатом є зміна відображуваних пунктів меню у інтерфейсі користувача відповідно до налаштувань.

Таблиця 2.5 – Функція «Генерація електронних документів»

Назва варіанту використання	Генерація електронних документів.
Дійові особи	Адміністратор
Стислий опис	Дає можливість адміністратору порталу згенерувати «PDF» версію запису, що зберігається у системі.
Передумова	Необхідно авторизуватися у якості адміністратора і перейти до пункту меню «Контракти» чи «Платіжні вимоги».

## Закінчення таблиці 2.5

Основний потік подій	Завдяки можливості роботи з електронними документами у системі, адміністратор має змогу натиснувши кнопку «Template» на Detail page запису («Контракт» чи «Платіжна вимога»), після відкриття відповідної сторінки переглянути та завантажити обраний запис у вигляді «PDF» документу. Також даний документ може бути відправлений на пошту клієнтові, для цього потрібно натиснути кнопку «Send via email» та прикріпити до листа потрібний електронний документ.
Альтернативний потік подій	Користувачеві виводиться відповідне повідомлення у разі виникнення проблем з'єднанням із сервером, документ не буде відображено чи завантажено.
Постумови	В результаті виконання даного варіанту використання адміністратор може побачити форми типових договорів. Електронний лист з прикріпленим електронним документом буде знаходитися на пошті у Клієнта.

Таблиця 2.6 Функція «Оформлення звіту»

Назва варіанту використання	Оформлення звіту про квартальний дохід агенції.
Дійові особи	Системний адміністратор
Стислий опис	Створення електронного звіту з відомостями щодо квартального прибутку агенції з нерухомості.
Передумова	Необхідно авторизуватися у ролі адміністратора і далі вибрати пункт меню «Оформити звіт».



## Закінчення таблиці 2.6

Основний потік подій	Після відкриття відповідної сторінки адміністратор за допомогою вбудованих функцій системи Salesforce може створювати звіти з різними критеріями та ґрунтуючись на певному типі об'єктів БД. Крім того Salesforce надає можливість створювати пересічні звіти, що включають в себе поля декількох сутностей. Після завершення налаштувань потрібно натиснути кнопку «Зберегти». За бажанням декілька звітів можна об'єднати в один dashboard, обрати кольорову гаму.
Альтернативний потік подій	Користувачеві виводиться відповідне повідомлення у разі виникнення проблем з'єднанням із сервером, звіт не буде згенеровано та збережено.
Постумови	В результаті звіт буде відображатись на табці «Reports».

Таблиця 2.7 – Функція «Авторизація на порталі»

Назва варіанту використання	Авторизація користувача у системі.
Дійові особи	Адміністратор, Клієнт
Стислий опис	Надає користувачеві порталу можливість використовувати функціональність порталу.
Передумова	Зареєстрований у системі логін та пароль від порталу.
Основний потік подій	Після відкриття відповідної сторінки користувач має можливість ввести свій логін та пароль, після чого система здійснить його авторизацію.
Альтернативний потік подій	Користувачеві виводиться відповідне повідомлення у разі виникнення проблем с сервером або невірних credentials.
Постумови	Автоматичне перенаправлення користувача на домашню сторінку порталу.

Таблиця 2.8 – Функція «Перехід до компоненту відображення апартаментів»

Назва варіанту використання	Перехід до компоненту відображення апартаментів.
Дійові особи	Адміністратор, Клієнт
Стислий опис	Надає можливість перейти до компоненту відображення апартаментів для перегляду інформації про них.
Передумова	Авторизуватися у системі і обрати пункт меню «Нерухомість». Потім натиснути на назву апартаментів в рядку з необхідною позицією.
Основний потік подій	Після відкриття відповідної табки користувач бачить сторінку зі списком доступних позицій, може сортувати їх, виконувати пошук за умови введення параметрів або виставляти оцінку апартаментам.
Альтернативний потік подій	Користувачеві виводиться відповідне повідомлення у разі виникнення проблем з'єднанням із сервером чи невірно введених параметрів для пошуку.
Постумови	Перенаправлення користувача на детальну сторінку апартаментів, на якій він може переглядати повну інформацію про житло, оформити запит на його бронювання.

Таблиця 2.9 – Функція «Створення запису сутності «Нерухомість»»

Назва варіанту використання	Створення запису сутності «Нерухомість».
Дійові особи	Адміністратор
Стислий опис	Створення запису апартаментів у БД
Передумова	Необхідно авторизуватися у ролі адміністратора і далі вибрати пункт меню «Нерухомість». Попередньо повинні бути створені відповідні записи поставників, контракт, Stock та Room Type – до яких буде відноситися житло.

## Закінчення таблиці 2.9

Основний потік подій	Після відкриття відповідної сторінки адміністратор за допомогою вбудованих функцій системи Salesforce та кастомного Lightning компоненту може створювати записи апартаментів. Для цього потрібно натиснути кнопку «New», заповнити форму необхідною інформацією за бажанням завантажити фото та натиснути кнопку «Save».
Альтернативний потік подій	Користувачеві виводиться відповідне повідомлення у разі виникнення проблем з'єднанням із сервером, запис апартаменту не буде збережено у системі.
Постумови	БД система наповнена записами типу «Нерухомість», код компоненту створення записів покривається Unit-тестами більш ніж на 75%.

## 2.3. Специфікація функціональних та нефункціональних вимог

## 2.3.1 Модель IDEF0

Світ навколо нас сповнений системами — механізмами, живими істотами, спільнотами людей, тобто системами механічними, біологічними, соціальними тощо. У широкому розумінні інформаційною можна назвати будь-яку систему опрацювання інформації.

Інформаційна система забезпечує приймання інформації, її перетворення, опрацювання, збереження і передачу результатів опрацювання споживачу: людині, машині, іншій інформаційній системі. Прикладом сучасної інформаційної системи може бути редакція газети або журналу, оснащена комп'ютерною технікою чи CRM платформа, що автоматизує бізнес процес.

Інформаційна система – це сукупність засобів збору, зберігання, передачі, оброблення інформації в певній предметній області для досягнення поставленої мети у процесі управління [42].

Інформаційна система (у загальному розумінні) — це система, яка здійснює або в якій відбуваються інформаційні процеси: пошук, збирання, зберігання, передавання й опрацювання інформації.

В інформаційній системі можуть відбуватися одночасно один, два чи кілька процесів.

Опрацювання інформації залежить від змісту вхідної інформації, але під час самого опрацювання інформація не осмислюється, а лише перетворюється згідно з попередньо розробленими алгоритмами.

Інформаційна система (у вузькому розумінні) — це комплекс інформаційних, технічних, програмних та організаційних засобів, необхідних для автоматизованого опрацювання інформації.

IDEF0 – це методологія графічного опису систем і процесів діяльності організації як безлічі взаємозалежних функцій. Особливістю IDEF0 є її акцент на ієрархічне представлення об'єктів, що значно полегшує розуміння предметної області. Вона дозволяє досліджувати функції організації, не пов'язуючи їх з об'єктами, що забезпечують їх реалізацію. В IDEF0 розглядаються логічні зв'язки між роботами, а не послідовність їх виконання в часі.

Тому перед безпосередньою розробкою програмного продукту, для того, щоб зрозуміти та проаналізувати процес якнайкраще в інструменті для візуалізації бізнес-процесів Ramus Educational були побудовані два типи схем: IDEF0 та DFD.

Перш за все, перед побудовою IDEF0-діаграми була сформульована задача: показати процес приймання та ведення обліку бронювань.

У процесі аналізу предметної області, була складена контекстна діаграма моделі у стандарті IDEF0 що показана на рисунку 2.3:

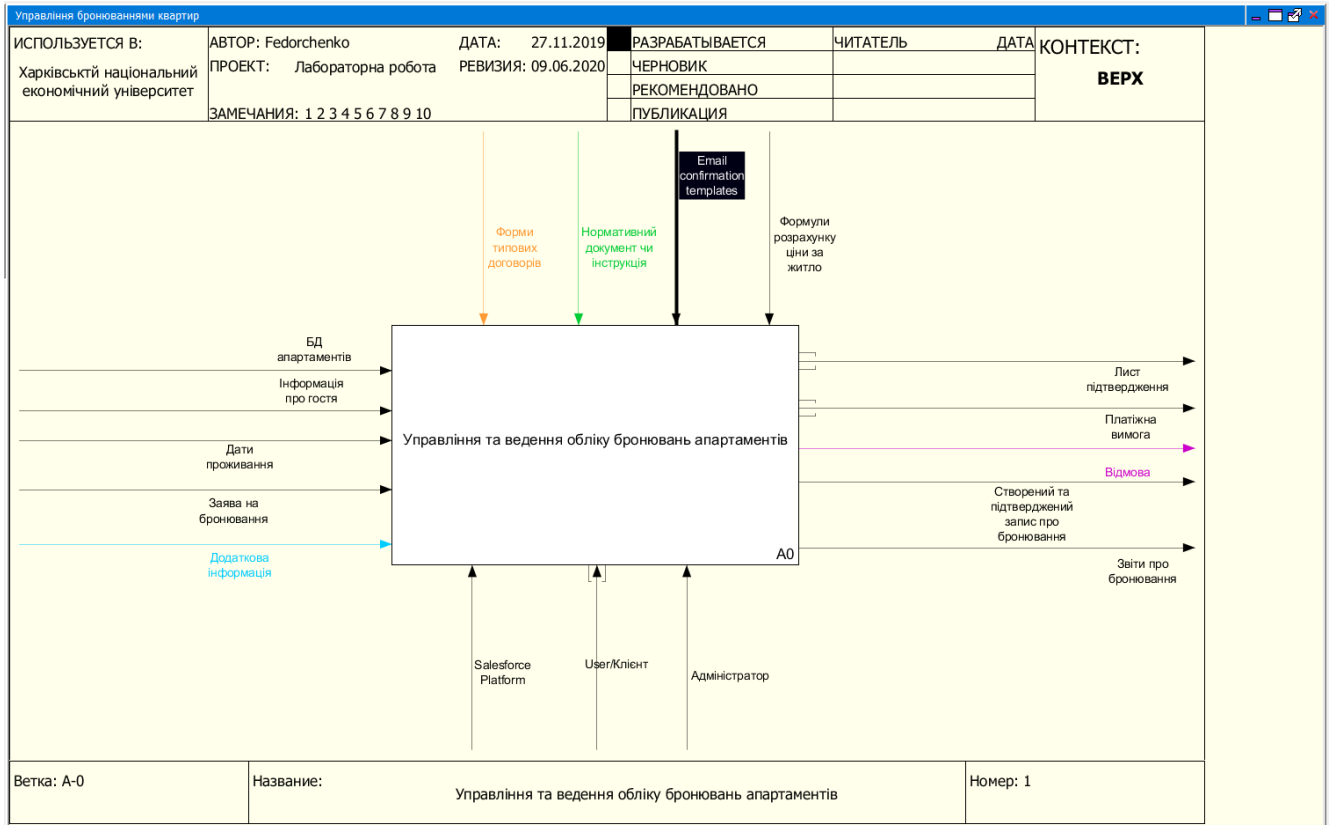


Рисунок 2.3 – Контекстна діаграма моделі у стандарті IDEF0

Декомпозиція контекстної діаграми реалізована на виділенні наступних робіт: пошук апартаментів, створення заявки на бронювання [5], визначення наявності вільного Stock, створення Invoice, створення Payment.

Таким чином, була побудована наступна діаграма 1-го рівня декомпозиції (рисунок 2.4):

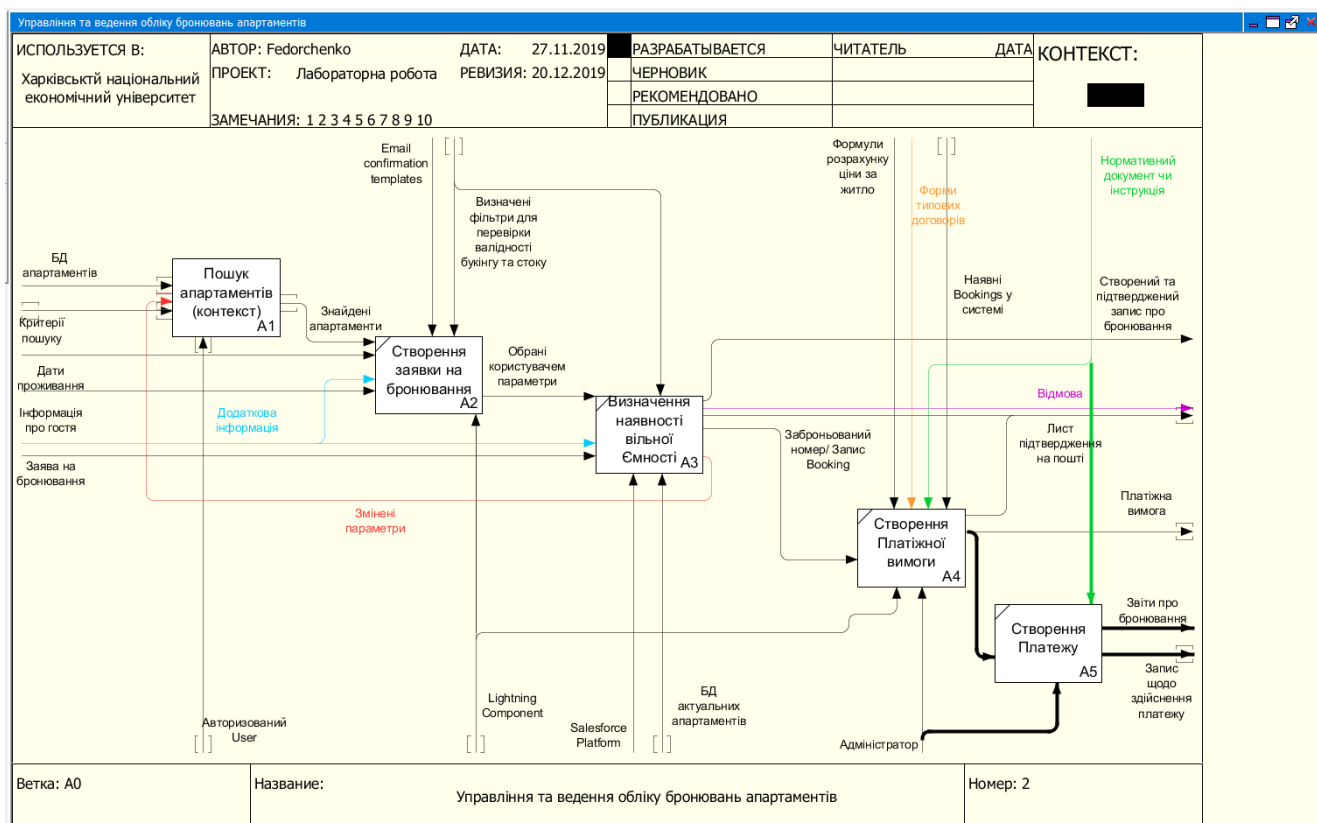


Рисунок 2.4 – Декомпозиція контекстної діаграми у стандарті IDEF0

Робота 1-го рівня декомпозиції має такі інтерфейсні дуги:

вхід: БД апартаментів, критерії пошуку, дати проживання, заява на бронювання;

вихід: звіт про сплачені бронювання, лист-підтвердження, платіжна вимога, БД наповнена даними;

управління: внутрішні правила організації, Цивільний кодекс України, розрахункові формули;

механізми: адміністратор, користувач, система Salesforce.

Декомпозиція роботи 1-го рівня – «Пошук апартаментів» вказана на рисунку 2.5:

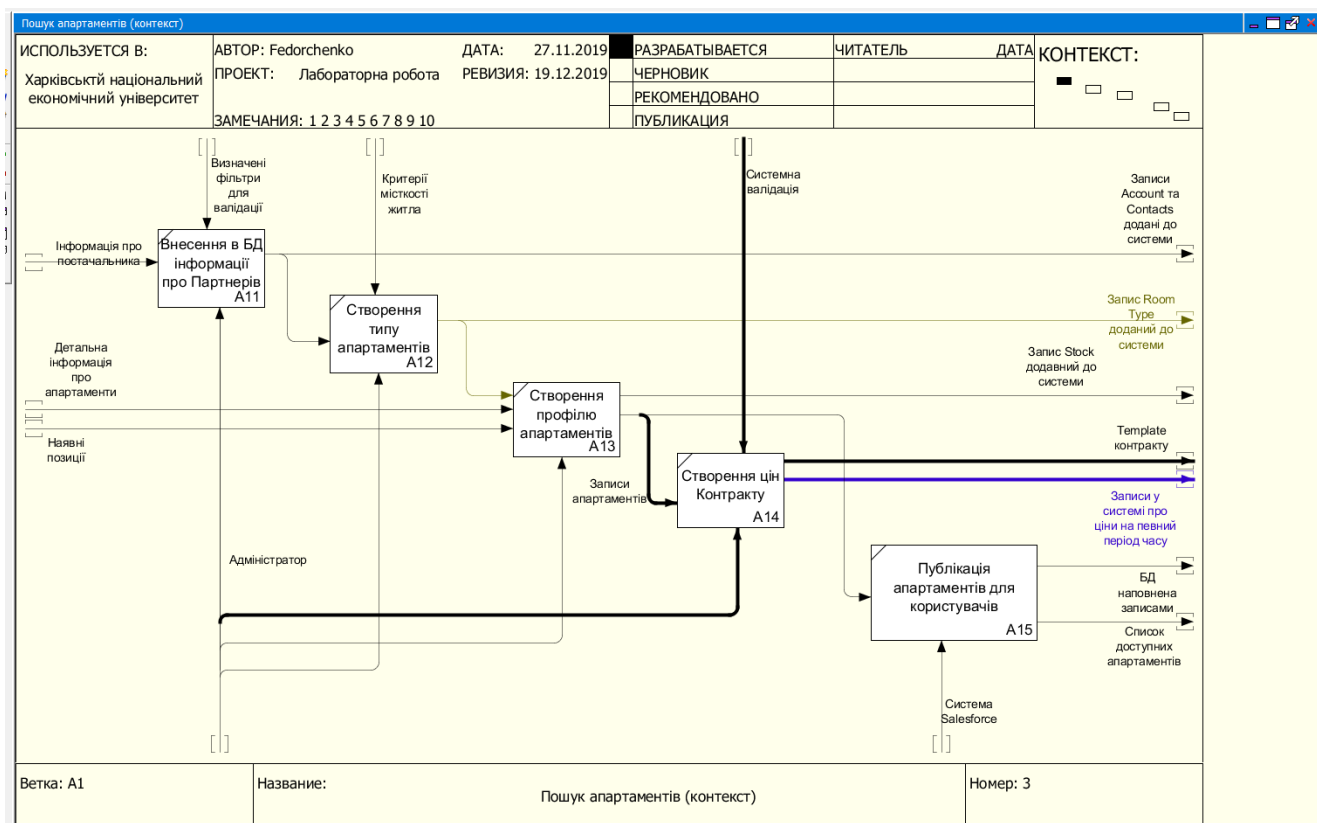


Рисунок 2.5 – Декомпозиція роботи «Пошук апартаментів» у стандарті IDEF0

Декомпозиція роботи 1-го рівня – «Пошук апартаментів» реалізована на виділенні наступних робіт: створення записів Account та Contact, створення різних типів апартаментів, створення безпосередньо профілю апартаментів, ціноутворення за допомогою контракту, публікація апартаментів для користувачів.

### 2.3.2 Модель DFD

Відображення структури системи, у якій зв'язками є інформаційні потоки, можна здійснити за допомогою діаграм потоків даних DFD (data flow diagrams). Ці діаграми використовують для аналізу та моделювання інформаційних систем з метою мінімізації потоків даних та зменшення їх об'єму, виявлення як дублювання інформації, так і дублювання шляхів її передавання.

DFD відображають джерела та споживачів інформації, вид та напрямок передачі інформації, елементи накопичення та процеси перетворення, при цьому використовуються різні засоби відображення елементів (нотації).

Діаграми потоків даних (DFD) є основним засобом моделювання функціональних вимог проєктованої системи. З їх допомогою ці вимоги розбиваються на функціональні компоненти (процеси) і представляються у вигляді мережі, пов'язаної потоками даних. Головна мета таких засобів -

продемонструвати, як кожен процес перетворить свої вхідні дані у вихідні, а також виявити відносини між цими процесами.

Потоки даних є механізмами, що використовуються для моделювання передачі інформації (або навіть фізичних компонент) з однієї частини системи в іншу. Важливість цього об'єкта очевидна: він дає назву цілому інструменту. Потоки на діаграмах зазвичай зображаються іменованими стрілками, орієнтація яких вказує напрямок руху інформації.

Діаграми потоків даних (DFD) є основним засобом моделювання функціональних вимог проєктованої системи. З їх допомогою ці вимоги розбиваються на функціональні компоненти (процеси) і представляються у вигляді мережі, пов'язаної потоками даних. Головна мета таких засобів - продемонструвати, як кожен процес перетворить свої вхідні дані у вихідні, а також виявити відносини між цими процесами.

В стандарті DFD контекстна діаграма модулю «Управління та ведення обліку бронювання апартаментів» та зовнішні об'єкти, з якими цей модуль взаємодіє (ці взаємодії позначені за допомогою вхідних і вихідних інформаційних потоків) наведена на рисунку 2.6:

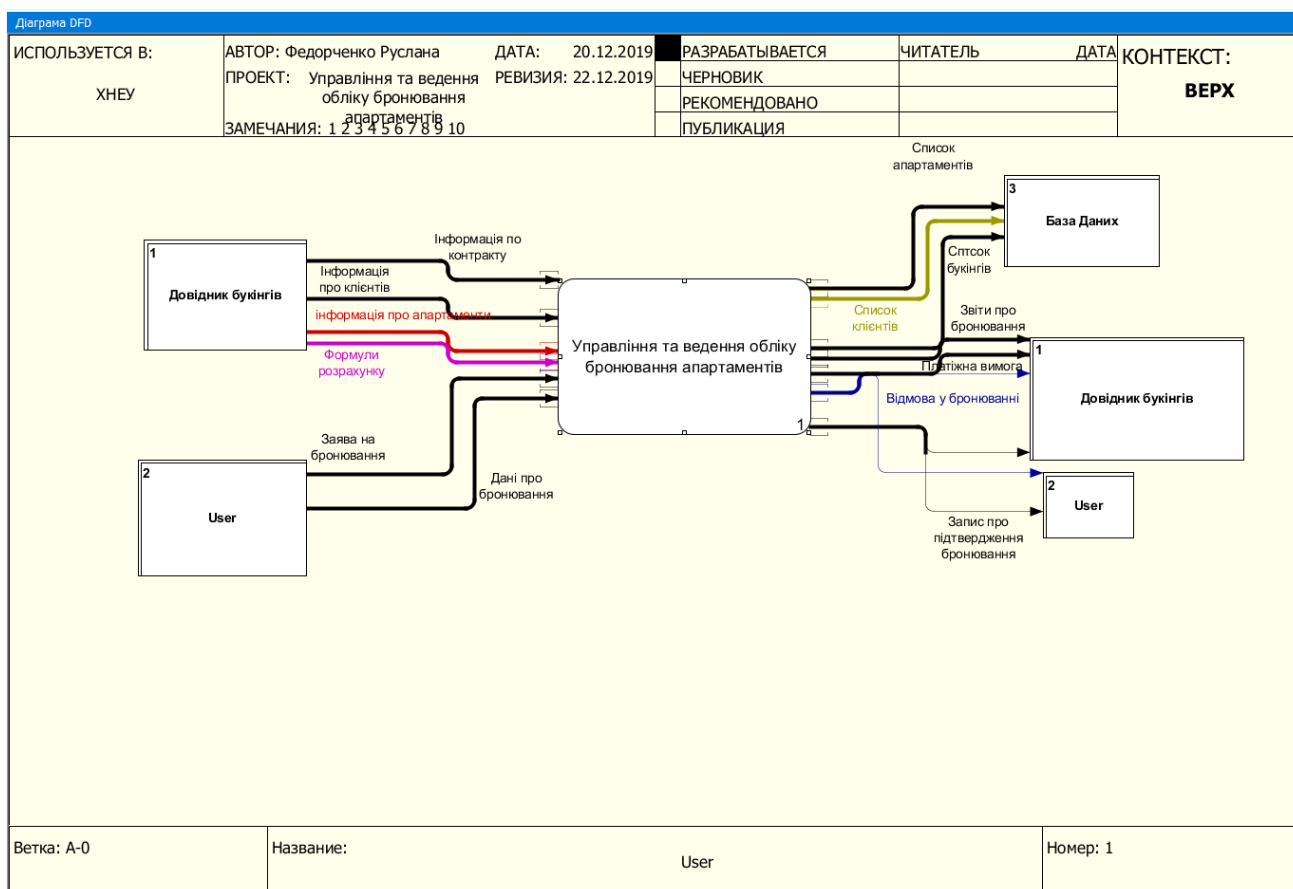


Рисунок 2.6 – Контекстна діаграма в стандарті DFD



Як видно з рисунку, зовнішня сутність Клієнт або User надає дані для бронювання. Дані про поставників, апартаменти та контракти надходять від Адміністратора системи.

У зовнішню сутність БД після закінчення бізнес-процесу передається на зберігання список виконаних замовлень, список клієнтів, апартаментів, платіжних вимог та список зроблених клієнтами платежів.

На рисунку 2.7 наведена діаграма потоків даних, що деталізує модуль «Управління та ведення обліку бронювання апартаментів» за основними роботами. Функціонально модуль декомпозується на такі процеси: формування списку клієнтів та апартаментів, створення контрактів, обробка бронювань, формування платіжних вимог та рахунків.

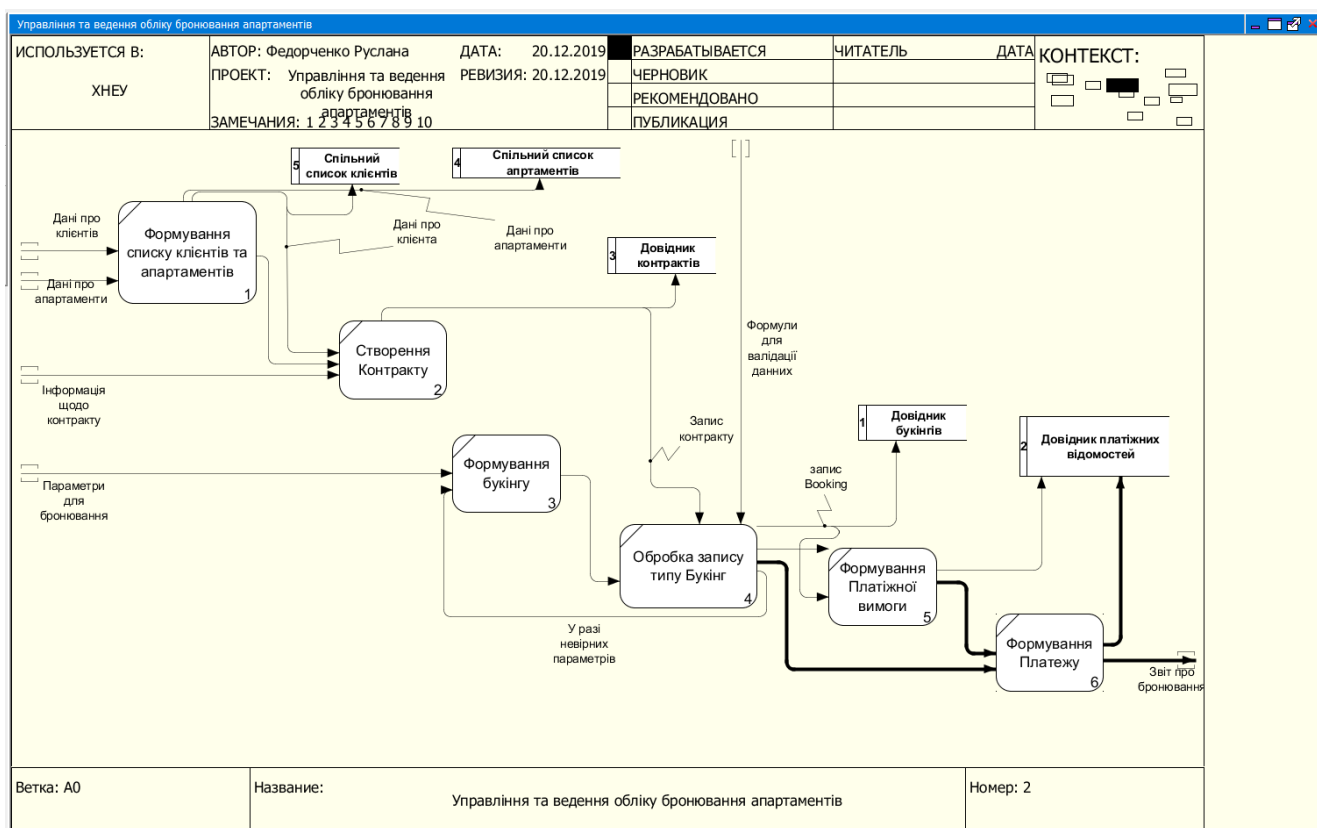


Рисунок 2.7 – Декомпозиція контекстної діаграми в стандарті DFD

На даному рівні також вводяться накопичувачі даних, які використовуються при вирішенні задач модуля. До них відносяться:

- довідник контрактів – призначений для зберігання даних за укладеними контрактами;
- список партнерів – призначений для зберігання даних про поставників;

- довідник апартаментів – містить інформацію про класи апартаментів, що здаються в агенції;
- букінг – призначений для зберігання інформації про замовлення, що включає в себе: дані про клієнта, місце та період бронювання, статус бронювання, інформація про апартаменти;
- довідник платіжних вимог та рахунків що в подальшому використовується для формування звітності з організаційної діяльності.

#### 2.4. Нефункціональні вимоги

Нефункціональні вимоги – описують «як» повинна працювати система або програмний продукт, та які властивості або характеристики повинна мати. Як правило, говорячи про нефункціональні вимоги, найчастіше говорять про атрибути якості (тобто вимоги, що визначають якісні характеристики розроблюваного ПЗ або системи – продуктивність, надійність, масштабованість) [18].

Було виділено 2 атрибути якості, а саме: продуктивність (performance) та зручність використання (usability). Система, що розробляється, пов'язана з автоматизацією діяльності агенств з пошуку нерухомості, отже виходячи з цього повинна виконувати задані функції з певним рівнем швидкодії, наприклад, обробляти запити до серверної частини в діапазоні від 300 до 500 мс.

Також, система повинна забезпечити зручність використання для початкового навчання та ефективної роботи, і для подальшої її експлуатації. Для сформованих функціональних та нефункціональних вимог процесу розроблена діаграма вимог в нотатції UML [10] та представлена на рисунку 2.8.

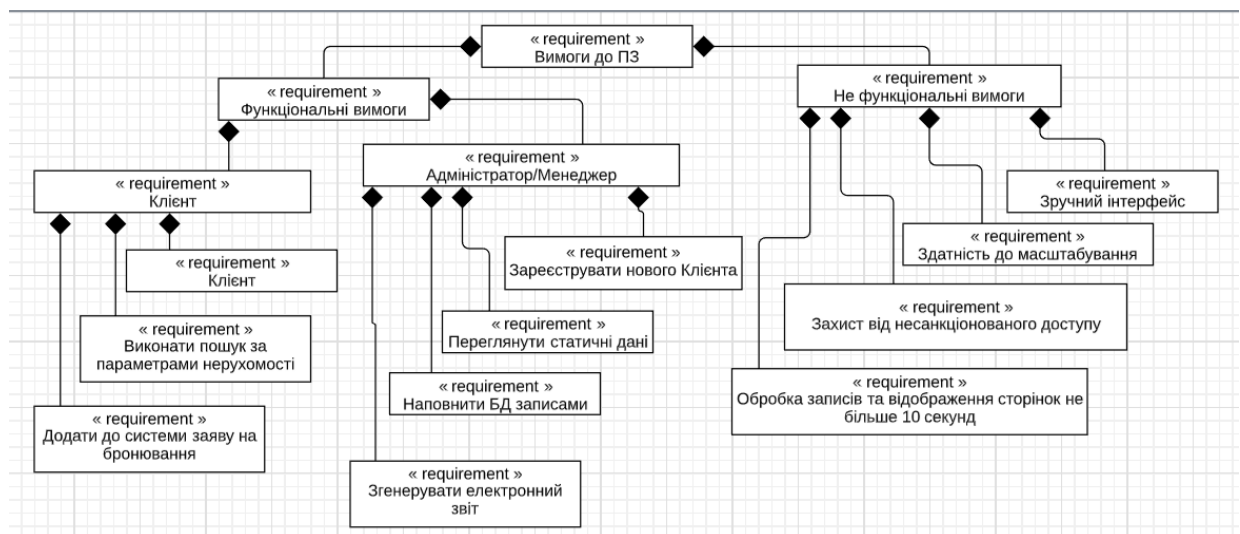


Рисунок 2.8 – Діаграма вимог

## 2.5. Функціональні вимоги

Перед розробкою вимог необхідно створити спочатку систему бізнес правил. Система бізнес правил – логічний опис стандартів, процедур і образу дій, якими керується підприємство у своїй роботі. Бізнес-правила використовуються для визначення взаємозв'язків між сутностями, діями, котрі можуть виконуватися над сутностями, проміжних контрольних точок і робочих процесів [27].

Бізнес правила також описують обмеження на образ дій, котрі повинні урахуватися системою. Правила слід складати, щоб вони були зрозумілі як керівникам системи так і розробникам, незалежно від мети або мови програмування. Під час аналізу предметної області процесу розроблена наступна система бізнес-правил:

- 1) користувачі в системі характеризуються ім'ям, прізвищем, по батьковим, електронною поштою, ідентифікаційним кодом, датою народження, адресом, телефоном;
- 2) ролі користувача можуть бути: адміністратор, клієнт та менеджер. Кожна роль має свої права доступу;
- 3) в системі не повинно існувати користувачів з однаковою поштою та ідентифікаційним номером;
- 4) адміністратор має привілеї доступу – він має доступ до кожної одиниці в системі;
- 5) клієнти не повинні мати повний доступ до системи, лише до стороннього сайту;
- 6) користувача можливо активувати, чи деактивувати. Активний користувач має доступ в систему, дезактивований – не має;
- 7) вхід до системи виконується через різні унікальні імена юзерів;
- 8) статус позиції нерухомості може бути: відкрита, заброньована або заповнена;
- 9) на одного Клієнта можливо призначити декілька бронювань.

Функціональні вимоги – описують внутрішню роботу системи, або, іншими словами, визначають «що» повинен робити програмний продукт. Прикладом може бути: калькулювання даних, маніпулювання даними, опрацювання даних та інші специфічні функції, які повинна виконувати система.

Нижче приведені функціональні вимоги системи, що розробляється:

- відстежувати позиції нерухомості на всіх етапах процесу, від тих, які відкриті до тих, які були заповнені бронюваннями або скасовані;

Пріоритет: Рекомендовано; Трудність: Висока;

- відстежувати всіх клієнтів системи, які претендують на певне житло, включаючи статус їхньої заяви на бронювання (були відхилені або прийняті);

Пріоритет: Рекомендовано; Трудність: Середня;

- відстежувати розміщення нерухомості на зовнішньому веб-сайті;

Пріоритет: Обов'язкове; Трудність: Середня;

- дозволити клієнтам публікувати відгуки на позиції нерухомості, які вони бронювали;

Пріоритет: Опційно; Трудність: Середня;

- забезпечувати безпеку даних для адміністраторів та клієнтів таким чином, щоб їх помилково не переглядали, редагували або видаляли користувачі, які не мають доступу до певних видів інформації;

Пріоритет: Обов'язкове; Трудність: Низька;

- автоматично інформувати всіх клієнтів про нові посади, які були розміщені;

Пріоритет: Опційно; Трудність: Висока;

- автоматично інформувати клієнтів про результат розгляду їх заяв на бронювання;

Пріоритет: Обов'язкове; Трудність: Середня;

- включити генерацію звітів, які надають дані про роботу системи, залежно від наявних у ній даних про платіжні вимоги, платежі та бронювання;

Пріоритет: Обов'язкове; Трудність: Середня;

- надати адміністраторам можливість відобразити місця розташування всіх апартаментів, щоб краще зрозуміти витрати на переселення;

Пріоритет: Рекомендовано; Трудність: Висока;

- імплементувати полегшене виконання декількох подібних завдань одночасно, наприклад, редагування декількох заяв на бронювання;

Пріоритет: Рекомендовано; Трудність: Висока;

- автоматично публікувати відкриті позиції на загальнодоступному вебсайті;

Пріоритет: Обов'язкове; Трудність: Висока;

- надати адміністраторові можливість наповнювати базу даних системи необхідними даними щодо поставників, нерухомості, платіжних вимог і платежів;

Пріоритет: Обов'язкове; Трудність: Висока;

## 2.6. Проектування інтерфейсу користувача

Інтерфейс - механізм, що забезпечує взаємодію користувача з програмним продуктом. Він, насамперед, повинен враховувати потреби кінцевих користувачів.

В рамках даного дипломного проекту стояло завдання розробити мокапи сторінок таким чином, щоб працювати з системою було зручно, зрозуміло та легко, як клієнту, так і адміністратору.

Вагомим фактором при розробці інтерфейсу компонентів було те, що він повинен бути спільним із нативним інтерфейсом самої системи Salesforce, а зокрема режимом Lightning Experience.

Акцент при розробці панелі адміністратора був зроблений на компактність подачі інформації, використання таблиць, мінімалістичний інтерфейс, що робить web-застосування комфортним для роботи та дозволяє новим співробітникам компанії швидко пристосуватися до роботи з професійним ПЗ.

Нижче, на рисунках 2.9 – 2.10, приведені приблизні мокапи для сторінок адмінського та клієнтського інтерфейсу.

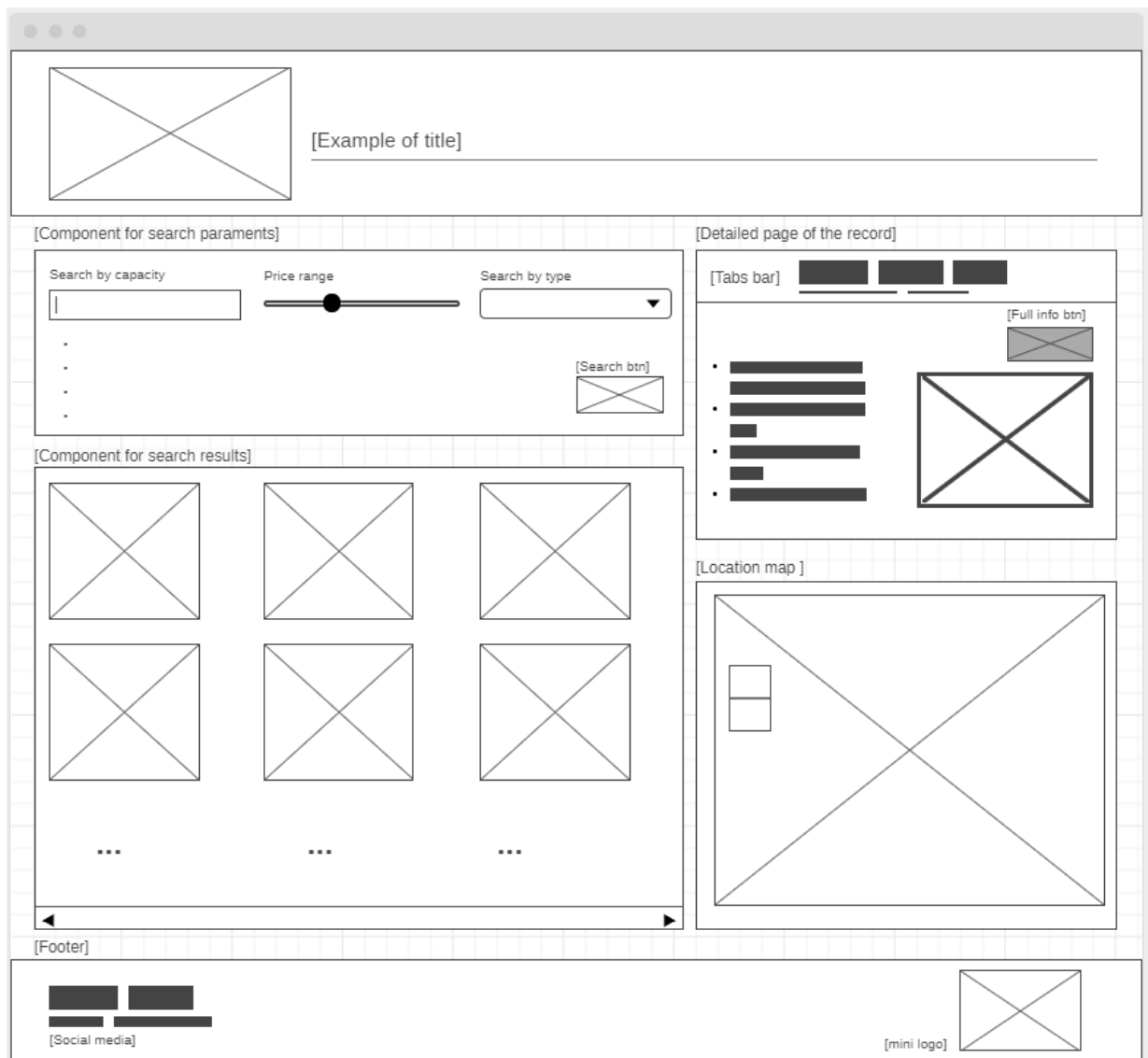


Рисунок 2.9 – Макет компоненту для пошуку апартаментів

Your Salesforce organization

[company logo] [global search in org] Org Setup

[global navigation items]

File Edit View Help

### Invoice creation

[search for Account] [date of Invoice] [Invoice Name] Search for Bookings

12 May 2016

		[Invoiced price]	[Price to Pay]
<input checked="" type="checkbox"/> Booking One	Client information	1000	670
<input checked="" type="checkbox"/> Booking Two	Client information	1000	1000
<input checked="" type="checkbox"/> Booking Three	Client information	1000	

[Cancel] [Create]

[Salesforce footer]

Full Banner Ad - 468x60

Рисунок 2.10 – Макет компоненту створення Invoice

## 3 ПРОЄКТНІ ТА ТЕХНІЧНІ РІШЕННЯ

### 3.1 Математична (логічна) та фізична моделі БД

Як правило, процес проектування бази даних складається з трьох основних етапів: концептуального, логічного та фізичного проектування. Розглянемо кожний із цих етапів окремо.

Концептуальне проектування бази даних — це процес створення моделі (схеми) предметної області, яка б не залежала від будь-яких фізичних аспектів її представлення; це процес, метою якого є виявлення, аналіз та характеристика виявлених сутностей предметної області, взаємозв'язків між ними, обмежень, накладених на ці сутності, а також їх поведінку.

Концептуальне проектування (моделювання) бази даних не залежить від таких подробиць її реалізації, як:

- Тип обраної цільової СУБД.
- Набір прикладних програм, що створюються.
- Мови програмування, що застосовуються.
- Типу обраної обчислювальної платформи.
- Будь-яких інших особливостей фізичної реалізації.

На даному етапі концептуальна модель предметної області повинна містити в собі виявлення та характеристику: типів сутностей; типів зв'язків; первинних та альтернативних ключів; обмежень цілісності.

Відповідно до обмежень проектування на даному етапі, маємо: мінімальна робоча модель інтегрованої бази даних системи обліку нерухомості буде містити п'ять взаємопов'язаних сутностей, що представляють кожен з етапів процесу бронювання нерухомості, на основі яких проводиться облік даних, що розглядаються у предметній області.

Атрибутом, що представляє первинний ключ стане атрибут `recordId`, який створюється автоматично для кожного збереженого екземпляру сутності та є унікальним на всьому просторі бази даних.

Базова логічна схема описуваної бази даних представлена ER-діаграмою на рис. 3.1:

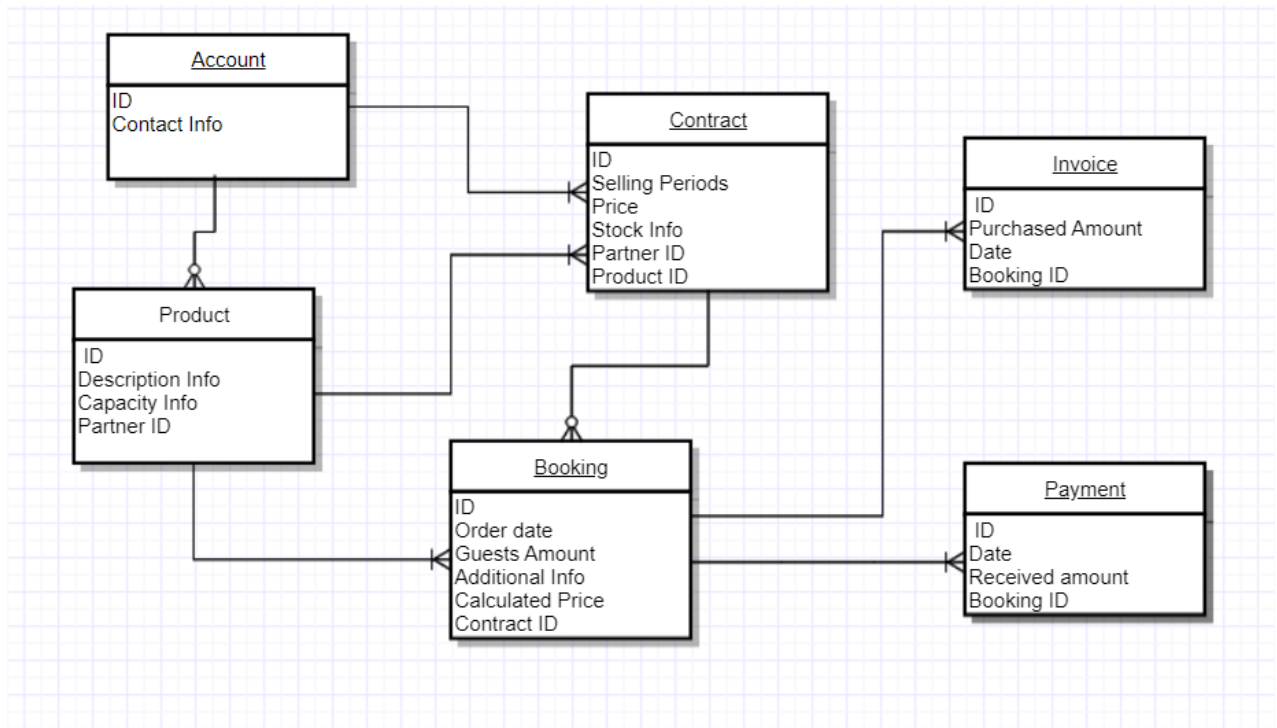


Рисунок 3.1 – ER-діаграма

Оскільки під час роботи над дипломним проектом були розглянуті різні варіанти процесу пошуку на бронювання нерухомості, основні етапи та параметри даного напрямку діяльності були виділені як сутності архітектури БД.

Після виділення кожної сутності, їй варто присвоїти певне осмислене ім'я, яке має бути зрозумілим для користувача. Обране ім'я та описання сутності включається до словника даних.

Якщо це можливо, варто встановити та внести в документацію інформацію про очікувану кількість екземплярів кожної сутності. Якщо сутність відома користувачам під різними іменами, всі додаткові імена рекомендовано виділити як синоніми або псевдоніми і також занести до словника даних. В таблиці 3.1 представлено словник даних, де описано сутності.



Таблиця 3.1 – Словник даних

Ім'я сутності	Описання	Псевдоніми	Коментарі
1	2	3	4
Партнер/Контакт	Загальне позначення для характеристики орендодавців, які надають нерухомість під оренду та їх потенційних представників. А також контактів, що відображають клієнтів підприємства, які будуть створювати заяви на бронювання нерухомості.	Account/Contact	Об'єкти, що беруть участь у формуванні та представленні статичних даних щодо наявних партнерів підприємства. Кожний об'єкт Account може мати декілька посилань на пов'язаний із ним запис сутності Product2 та декілька посилань на пов'язані із ним записи сутностей: Contract, Contract_Lines, Product2, Supplier_Invoice, Supplier_Payment, Contact

Продовження таблиці 3.1

1	2	3	4
Букінг	Загальне позначення для характеристики запитів на бронювання нерухомості	Booking	Може містити безліч посилань на об'єкти Contract_Lines, Supplier_Invoice, Supplier_Payment (зв'язок "один до багатьох") та лише одне посилання на запис типу Account, Contract
Нерухомість/Ємність	Загальне позначення для безпосередньої характеристики представленої до оренди нерухомості. Об'єкт Ємність використовується для контролю вільного місця у апартаментах під бронювання	Product2/Stock	Може містити безліч посилань на об'єкти Booking__c, Contract_Lines__c (зв'язок "один до багатьох"). З об'єктом Account існує зв'язок «один до одного»
Контракт/Ціноутворення	Загальне позначення для характеристики цін за нерухомість на певний період здачі її в оренду	Contract/Contract_Lines	Contract може мати лише одне посилання на об'єкт Account та безліч посилань на записи Contract_Lines. Ціноутворення містить лише одне посилання на об'єкти Product2 та Stock та безліч посилань на записи букінгів

Закінчення таблиці 3.1

1	2	3	4
Платіжна вимога/Платіж	Парні характеристики, що використовуються для обліку інформації щодо платежів за бронювання нерухомості	Supplier_Invoice/Supplier_Payment	Дані сутності можуть містити безліч посилань на об'єкт Booking__c, та лише одне посилання на об'єкт Account та Product2, адже за одну операцію можна здійснити оплату нерухомості одного конкретного Поставника
Daily Stock Data	Ця сутність використовується для описання інформації про кожен день ціноутворення і контролює кількість зайнятого гостями місця у нерухомості, а також використовується для підрахунку вартості Букінга	Daily Stock Data	Дана сутність може містити безліч посилань на записи букінгів (зв'язок «один до багатьох»), та лише одне посилання на такі об'єкти як Ємність, Ціноутворення та Нерухомість

Приклад документування типів сутностей в середовищі Salesforce Object Manager представлено на рисунках 3.2 та 3.3:

Enable Reports	✓
Track Activities	✓
Track Field History	✓
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

Рисунок 3.2 – Документування типів сутностей в Salesforce Object Manager

Description	Сутність для відстеження заяв клієнтів на бронювання наявної у компанії нерухомості.
API Name	Booking__c
Custom	✓
Singular Label	Букінг
Plural Label	Букінги

Рисунок 3.3 – Документування типів сутностей в Salesforce Object Manager

### 3.1.2 Визначення типів зв'язків

Для забезпечення коректної роботи мінімальної робочої моделі потрібно виявити лише ті зв'язки, що можуть допомогти вирішити поставлену задачу. Таким чином, між об'єктами, що розглядаються, може існувати величезна кількість зв'язків різного типу.

В процесі обробки даних було виявлено залежності між записами контрактів та ціноутворення з букінгами нерухомості та об'єктами, що відповідають за платіжні вимоги та платежі. Типи зв'язку було запроєктовано таким чином, аби відобразити залежності між сутностями найбільш наглядно та зручно для кінцевого користувача.

Отже, для подальшої обробки інформації було потрібно виділити одну з сутностей як головну. Це треба для компонування інформації та максимальної інкапсуляції відомостей про кожен з об'єктів нерухомості в одному місці.

Таким об'єктом став об'єкт типу Booking, що представляє основну інформацію з бази, оскільки саме дані з цього об'єкту є найбільш повними та різносторонніми, а що особливо важливо, відображають зв'язки з усіма іншими об'єктами системи. На мою думку, це добре для характеристики, порівняння та аналітики обліку нерухомості. Дані з об'єкту типу Booking майже не потребують доповнення та форматування.

Після прийняття цього рішення, виявити можливі зв'язки стало набагато простіше, адже мені потрібно було отримати якнайбільшу кількість інформації з однієї із можливих реляційних моделей для представлених сутностей.

Отже, на підставі аналізу предметної області була розроблена фізична модель даних. В роботі використовувався вбудований в систему Salesforce інтерфейс для створення бази даних. Створена БД є сукупністю зв'язаних таблиць. Побудована концептуальна модель приведена на рисунку 3.1. Фізична модель бази даних представлена рисунку 3.2.

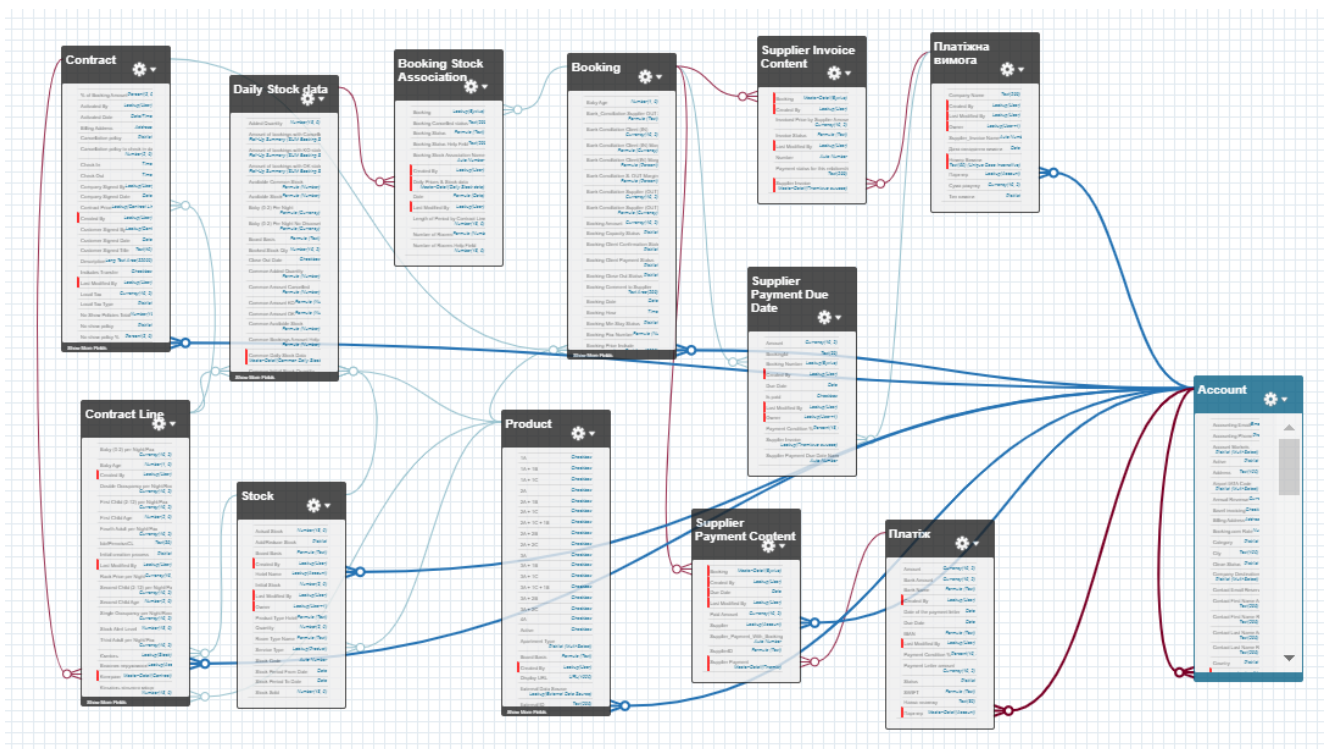


Рисунок 3.2 - Фізична модель бази даних

### 3.1.3 Визначення атрибутів

На даному етапі важливо визначити усі атрибути, які повинні бути представлені в концептуальній моделі бази даних. В моєму проекті усі атрибути об'єктів є стандартними характеристиками приведених сутностей, а також допоміжними полями (для валідації та обробки даних), окрім атрибутів, що створюються автоматично для кожного об'єкту і надають системну інформацію.

Кожний із створених об'єктів автоматично отримує наступні системні поля:

- Name – поле, що є обов'язковим для всіх об'єктів. Містить ім'я конкретного запису об'єкту.

- CreatedBy – містить інформацію про користувача, який створив конкретний запис.

- LastModifiedBy – містить інформацію про користувача, який редагував (змінював/видаляв) конкретний запис. На момент створення нового запису поля CreatedBy та LastModifiedBy містять однакову інформацію.

- CreatedDate – містить інформацію про дату та час створення конкретного запису.

- LastModifiedDate – містить інформацію про дату та час останнього внесення змін до конкретного запису. На момент створення нового запису поля CreatedDate та LastModifiedDate містять однакову інформацію.

- Owner – містить інформацію про користувача, до якого прикріплено конкретний запис (поля CreatedBy та Owner не завжди мають містити однакову інформацію, все залежить від налаштованих автоматизованих процесів).

Розглянемо кожну із створених сутностей окремо, у таблицях представлених нижче показані обмеження атрибутів сутностей [13].

Таблиця 3.2 - Сутність Booking.

Межі / допустимі значення	Структура (формат)	Умова	Значення за замовчуванням
1	2	3	4
Ім'я атрибута або агрегату: Ім'я орендатора, Рах_First_Name			
[A-Za-z]{3,15}	Text(50)	Обов'язкове поле	-
Ім'я атрибута або агрегату: Ім'я орендатора, Рах_First_Name			
[A-Za-z]{3,15}	Text(50)	Обов'язкове поле	-

## Продовження таблиці 3.2

Ім'я атрибута або агрегату: Основні умови проживання, Board_Basis			
Room Only Bed & Breakfast Half Board Full Board All Inclusive Ultra All Inclusive	Picklist	Обов'язкове поле Має відповідати аналогічному полю на Contract Line	-
Ім'я атрибута або агрегату: Орендодавець, Account			
-	Text(50), Lookup(Партнер)	Автоматично заповнюється системою на основі обраної нерухомості	-
Ім'я атрибута або агрегату: Назва букінгу, Name			
-	Auto Number	Автоматично заповнюється системою на основі обраної нерухомості	-
Ім'я атрибута або агрегату: Мобільний телефон, Pax_Phone_Number			
^[0-9]{3}-[0-9]{3}-[0-9]{4}\$	Phone	Обов'язкове поле	-
Ім'я атрибута або агрегату: Кількість малюків. Number_of_Babies			
Не повинно бути менше нуля	Number(1, 0)	-	0
Ім'я атрибута або агрегату: Кількість дітей, Number_of_Childs			
Не повинно бути менше нуля	Number(1, 0)	-	0

## Продовження таблиці 3.2

1	2	3	4
Ім'я атрибута або агрегату: Кількість дорослих, Number_of_Adults			
Не повинно бути менше чи дорівнювати нулю	Number(1, 0)	Обов'язкове поле	1
Ім'я атрибута або агрегату: Контракт, Contract			
-	Lookup (Контракт)	Автоматично заповнюється системою на основі обраної нерухомості та дат перебування	-
Ім'я атрибута або агрегату: Додаткова інформація, Additional_information			
-	Rich Text Area(32768)	-	-
Ім'я атрибута або агрегату: Дата Check out, Booking_Check_out_Date			
Має бути більша за поточну дату на більше за дату check-in	Date	Обов'язкове поле	-
Ім'я атрибута або агрегату: Дата Check in, Booking_Check_in_Date			
Має бути більша за поточну дату	Date	Обов'язкове поле	-
Ім'я атрибута або агрегату: Бажана кількість кімнат, Number_of_Rooms			
-	Number(1, 0)	Обов'язкове поле	-
Ім'я атрибута або агрегату: Supplier Invoice Status, Supplier_Invoice_Status			
Received Received Less Conflict	Picklist	Автоматично присвоюється системою залежно від суми за букінг у платіжній вимозі	-



## Продовження таблиці 3.2

1	2	3	4
Ім'я атрибута або агрегату: Supplier Paid Amount, Supplier_Paid_Amount			
-	Roll-Up Summary (SUM Supplier Payment Content)	Автоматично присвоюється системою залежно від оплаченої суми за букінг	-
Ім'я атрибута або агрегату: Invoiced Price by Supplier Amount			
-	Roll-Up Summary (SUM Supplier Invoice Content)	Сума всіх платіжних вимог за конкретний букінг	-
Ім'я атрибута або агрегату: Email, Рах_Email__с			
-	Email	Обов'язкове поле	-
Ім'я атрибута або агрегату: Check in/out Status, Check_in_out_Status			
OK check in/out KO check in/out	Picklist	Автоматично розраховується системою на основі дат заїзду і виїзду	-
Ім'я атрибута або агрегату: Calculated Purchasing Price			
-	Currency(16, 2)	Система автоматично розраховує вартість букінгу залежно від кількості гостей та терміну проживання	-
Ім'я атрибута або агрегату: Booking Stock Status			
-	Picklist	Автоматично калькулюється системою на основі дат заїзду і виїзду, а також бажаної кількості кімнат	-
Ім'я атрибута або агрегату: Booking Capacity Status			
OK capacity KO capacity Not Controlled	Picklist	Автоматично калькулюється системою на основі дат кількості гостей та ємності обраної нерухомості	-

## Закінчення таблиці 3.2

1	2	3	4
Ім'я атрибута або агрегату: Booking Supplier Payment Status			
Conflict Paid Consolidated Upcoming Paid Less	Picklist	Автоматично присвоюється системою залежно від оплаченої суми за букінг	Upcoming

Таблиця 3.3 - Обмеження атрибутів сутності Stock

Межі / допустимі значення	Структура (формат)	Умова	Значення за замовчуванням
1	2	3	4
Ім'я атрибута або агрегату: Stock Period From Date			
-	Date	Обов'язкове поле	Початок дії ціноутворення
Ім'я атрибута або агрегату: Stock Period To Date			
-	Date	Обов'язкове поле	Кінець дії ціноутворення
Ім'я атрибута або агрегату: Service Type			
-	Lookup (Нерухомість)	Обов'язкове поле	Автоматично заповнюється під час створення Ціноутворення
Ім'я атрибута або агрегату: Initial Stock			
Має бути більше нуля	Number(3, 0)	Обов'язкове поле	Автоматично заповнюється під час створення Ціноутворення

Таблиця 3.4 - Обмеження атрибутів сутності Contract Line

Межі / допустимі значення	Структура (формат)	Умова	Значення за замовчуванням
1	2	3	4
Ім'я атрибута або агрегату: Ємність, Stock__с			
-	Lookup(Stock)	Створюється автоматично та приєднується до того ж Поставника, що й обраний Контракт	-
Ім'я атрибута або агрегату: Початок дії, Travel_Period_From			
-	Date	Має бути в рамках обраного Контракту Обов'язкове поле	Початок дії Контракту
Ім'я атрибута або агрегату: Основні умови проживання, Board_Basis			
Room Only Bed & Breakfast Half Board Full Board All Inclusive Ultra All Inclusive	Picklist	Створюваний у подальшому букінг має мати аналогічне значення у відповідному полі Обов'язкове поле	-
Ім'я атрибута або агрегату: Нерухомість, Room_Type			
-	Lookup (Нерухомість)	Має належати до того ж Поставника, що й обраний Контракт Обов'язкове поле	-
Ім'я атрибута або агрегату: Назва			
-	Auto Number	-	CL-{0001}
Ім'я атрибута або агрегату: Мінімальний термін проживання, Length_of_Stay_From			
Повинно бути більше 0, та не повторюватися з іншими Contract Line у рамках одного контракту	Number(2, 0)	Використовується для відстеження тривалості відповідних букінгів Обов'язкове поле	-

## Закінчення таблиці 3.4

1	2	3	4
Ім'я атрибута або агрегату: Максимальний термін проживання, Length_of_Stay_To			
Повинно бути більше 0 і мінімального строку проживання	Number(2, 0)	Використовується для відстеження тривалості букінгів. Не має межувати з іншими Contract Line у рамках одного контракту Обов'язкове поле	-
Ім'я атрибута або агрегату: Кінець дії, Travel_Period_To			
-	Date	Має бути в рамках обраного Контракту Обов'язкове поле	Кінець дії Контракту
Ім'я атрибута або агрегату: Кількість вільного місця, Stock_period_quantity			
-	Number(18, 0)	Кількість запропонованого вільного місця на період дії Ціноутворення Обов'язкове поле	-
Ім'я атрибута або агрегату: Контракт, Contract			
-	Master-Detail (Контракт)	Заповнюється автоматично при створення запису Ціноутворення Обов'язкове поле	-
Ім'я атрибута або агрегату: Власник нерухомості, Supplier			
-	Text(50), Lookup(Партнер)	Заповнюється автоматично при створення запису Ціноутворення	Постачальник вказаний у відповідному контракті

Варто зазначити, що даний об'єкт додатково з'єднаний з допоміжним об'єктом, який називається Daily Stock Data. Ця додаткова, але важлива сутність, відповідає за відстеження створених букінгів та вільного простору у апартаментах, розподіляючись на кожний окремий день Ціноутворення.

Створювані букінги за допомогою junction objects (Booking Daily Stock Association, використовуючи зв'язок «один до багатьох» та посилаються на запис Букінг та Daily Stock Data) утворюють зв'язок з Daily Stock Data, завдяки даному зв'язку та створеному автоматичному процесу в інструменті Process Builder, система підраховує кількість бронювань на кожен день та віднімає кількість зайнятих букінгом кімнат від загальної ємності нерухомості.

Сутність Contract Line містить інформацію щодо ціноутворення, яка у подальшому використовується для підрахунку вартості бронювання. Дана інформація зберігається у полях типу Currency, обов'язковою для зазначення є ціна за одного дорослого на добу, та за двох дорослих на добу.

Таблиця 3.5 - Обмеження атрибутів сутності Нерухомість/Product2

Межі / допустимі значення	Структура (формат)	Умова	Значення за замовчуванням
1	2	3	4
Ім'я атрибута або агрегату: Назва, ApartmentName			
-	Text Area(255)	Обов'язкове поле	-
Ім'я атрибута або агрегату: Власник, Supplier			
-	Lookup(Партнер)	Обов'язкове поле	-
Ім'я атрибута або агрегату: WiFi, Кондиціонер, Пральна машина, Чи можна з тваринами?			
-	Checkbox	-	-
Ім'я атрибута або агрегату: Кількість кімнат, Rooms_Amount			
	Number(18, 0)	Обов'язкове поле	
Ім'я атрибута або агрегату: Місто, City			
	Text(40)	Обов'язкове поле	-
Ім'я атрибута або агрегату: Область, State			
-	-	Обов'язкове поле	-
Ім'я атрибута або агрегату: Основні умови проживання, Board_Basis			
Room Only Bed & Breakfast Half Board Full Board All Inclusive Ultra All Inclusive	Picklist	Обов'язкове поле	-

## Закінчення таблиці 3.5

1	2	3	4
Ім'я атрибута або агрегату: Поштовий код, Postal_Code			
-	Text(20)	-	-
Ім'я атрибута або агрегату: Тип апартаментів, Type_of_apartment			
Studio Loft Junior 1 Bedroom Basement apartment Officetel Luxury apartment House	Picklist	Обов'язкове поле	-
Ім'я атрибута або агрегату: Країна, Country			
-	Text(80)	Обов'язкове поле	-
Ім'я атрибута або агрегату: PictureID, PictureID			
-	URL(255)	Автоматично присвоюється системою у момент завантаження фото нерухомості	-
Ім'я атрибута або агрегату: Iframe Посилання, Iframe_Link			
-	URL(255)	-	-

Приведена вище сутність містить додаткові поля для контролю кількості постояльців, яку може прийняти даний вид нерухомості (1A\_\_с, 1A\_1B\_\_с, \_\_1C\_\_с, 2A\_\_с, 2A\_1B\_\_с, 2A\_1C\_\_с, 2A\_1C\_1B\_\_с, A\_2B\_\_с, 2A\_2C\_\_с, 3A\_\_с, 3A\_1B\_\_с, 4A\_\_с). У якості назв полів використовуються аббревіатури А- adults, тобто дорослі, С- children це діти та В- baby, дана аббревіатура відповідає за малюків.

Дані поля це характеристика типу Checkbox, що містить значення true чи false. При подальшій роботі система перевіряє чи може кількість гостей зазначена у букінгу «розміститися» у обраній нерухомості. Залежно від результатів даної перевірки на Букінгу змінюється поле Capacity Status.

## Об'єкт Контракт/Contract

Даний об'єкт використовується для систематизації інформації про договори укладені з постачальником нерухомості. Він слугує базою для подальшого

ціноутворення на різні періоди задачі нерухомості в оренду. Основні атрибути об'єкту **Контракт**:

- 1) **Contract\_Date** - дата укладання контракту;
- 2) **Contract\_Target** - очікуваний прибуток, використовується для генерування звітності;
- 3) **AccountId Lookup (Партнер)** – партнер, з яким укладаються договір;
- 4) **StartDate** - початок дії контракту;
- 5) **EndDate** - кінець дії контракту;
- 6) **Статус Status** - ціни на нерухомість можна створювати базуючись лише на підписаних контрактах;

#### Об'єкт **Відгук/Rating**

Даний об'єкт використовується для збору інформації щодо вражень клієнтів від нерухомості, слугує допоміжним об'єктом для генерування звітів. Основні атрибути об'єкту **Відгук**:

- 1) **Apartment** - нерухомість на яку користувач залишає відгук **Lookup (Нерухомість)**;
- 2) **Comment** - **Comment Rich Text Area(32768)**, розгорнутий опис вражень від нерухомості;
- 3) **RatingRating** - числове відображення рейтингу, від 1 до 5ти, тип поля **Number (1, 0)**;

#### Платіжна вимога/**Invoice**

Даний об'єкт використовується для збору інформації щодо висунутих Клієнтам платіжних вимог за створені бронювання. На основі інформації з даного об'єкту менеджер системи матиме змогу генерувати звітність за платіжками та поточними букінгами. Варто зазначити, що плата за один Букінг може бути знята за декілька платіжних вимог. Система розмежує стан платіжним вимог на Букінгу за допомогою поля статусу. Основні атрибути даної сутності:

- 1) Дата складання вимоги **Invoice\_date**;
- 2) Номер чи .назва Вимоги - **Supplier\_Invoice\_Number**, тип поля **Text(50) (Unique Case Insensitive)**;
- 3) Партнер **Supplier**, тип поля **Lookup** зв'язок з об'єктом **Партнер**;
- 4) Сума виставленого рахунку - **Invoiced\_Amount** поле типу **Currency(16, 2)**;

З сутністю **Букінг Платіжна вимога** з'єднується за допомогою **junction object** під назвою **Supplier Invoice Content**. Завдяки цьому зв'язку система автоматично рахує суму всіх Платіжних вимог створених для конкретного запису типу Букінгу.

## Платіж/Payment

Даний об'єкт використовується для збору інформації щодо оплачених Клієнтами платіжних вимог за створені бронювання. На основі інформації з даного об'єкту менеджер системи матиме змогу генерувати звітність за платіжками та поточними букінгами, це буде корисним для відстеження тенденцій розвитку підприємства.

Варто ще раз зазначити, що за аналогією із Платіжною вимогою плата за один Букінг може бути знята за декілька платежів. Система розмежує стан платежів на Букінгу за допомогою поля статусу. Основні атрибути даної сутності:

- 1) Назва платежу, Name - заповнюється системою автоматично;
- 2) Партнер Supplier, тип поля Master-detail зв'язок з об'єктом Партнер;
- 3) Status Status;
- 4) Payment Letter amount - характеристика, що показує сплачену суму за букінги Currency (16, 2);
- 5) поля IBAN, SWIFT та Bank Name використовуються в якості реквізитів для платежу;

З сутністю Букінг Платіжна вимога з'єднується за допомогою junction object під назвою Supplier Invoice Content. Завдяки цьому зв'язку система автоматично рахує суму всіх Платіжних вимог створених для конкретного запису типу Букінг.

Основна інформація щодо реквізитів компанії для генерації шаблонів документів та платіжних вимог статично зберігається у системі та знаходиться у стандартному об'єкті Custom Labels.

### 3.1.4 Діаграма класів

Class diagram (діаграма класів) основна діаграма для створення коду програми. За допомогою діаграми класів створюється внутрішня структура системи, описується спадкування й взаємне положення класів друг щодо друга. Тут описується логічне представлення системи. Саме логічне, тому що класи - це лише заготовки, на основі яких потім будуть визначені фізичні об'єкти. Клас в мові UML служить для позначення множини об'єктів, які мають однакову структуру, поведінку і відносини з об'єктами інших класів.

Нижче, на рисунках 3.3 – 3.5, зображені безпосередньо діаграми класів до даного web-проекту.



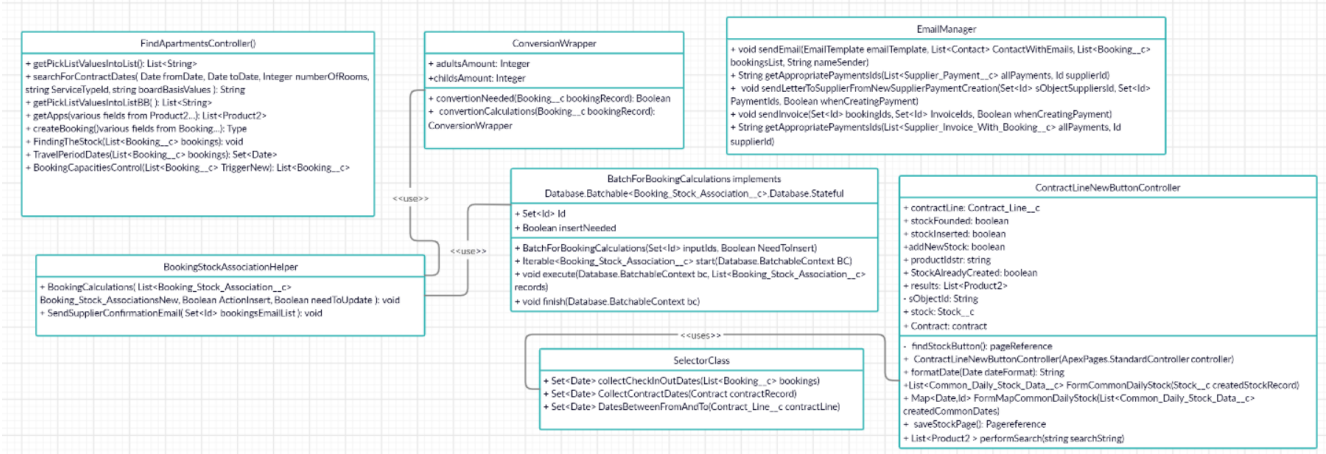


Рисунок 3.3 – Перша частина діаграми основних класів системи

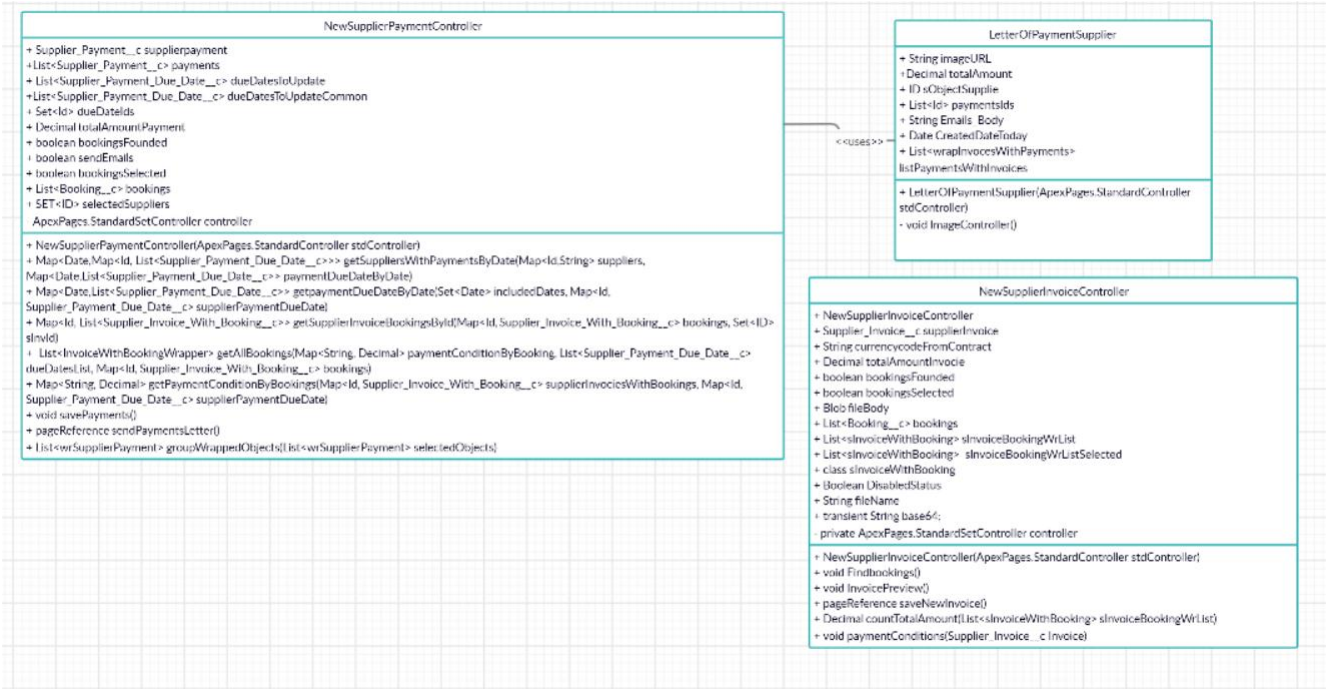


Рисунок 3.4 – Продовження діаграми основних класів системи

Користувальницькі контролери були розроблені для автоматизації всіх основних бізнес процесів, починаючи від пошуку нерухомості (за це відповідає клас Findapartmentcontroller()), закінчуючи внесенням інформації про платежі за бронювання (LetterOfPayment()).

Також Арех код використовується системою для автоматичної відправки email повідомлень, наповнення Календарю букінгами та відображення сторінок із сторонніх сервісів, що використовуються системою.

Однією з основних задач, які вирішуються за допомогою кодування є розрахунок вартості бронювання та автоматичний трекінг вільного місця для майбутніх букінгів (BookingStockAssosiationHelper()).

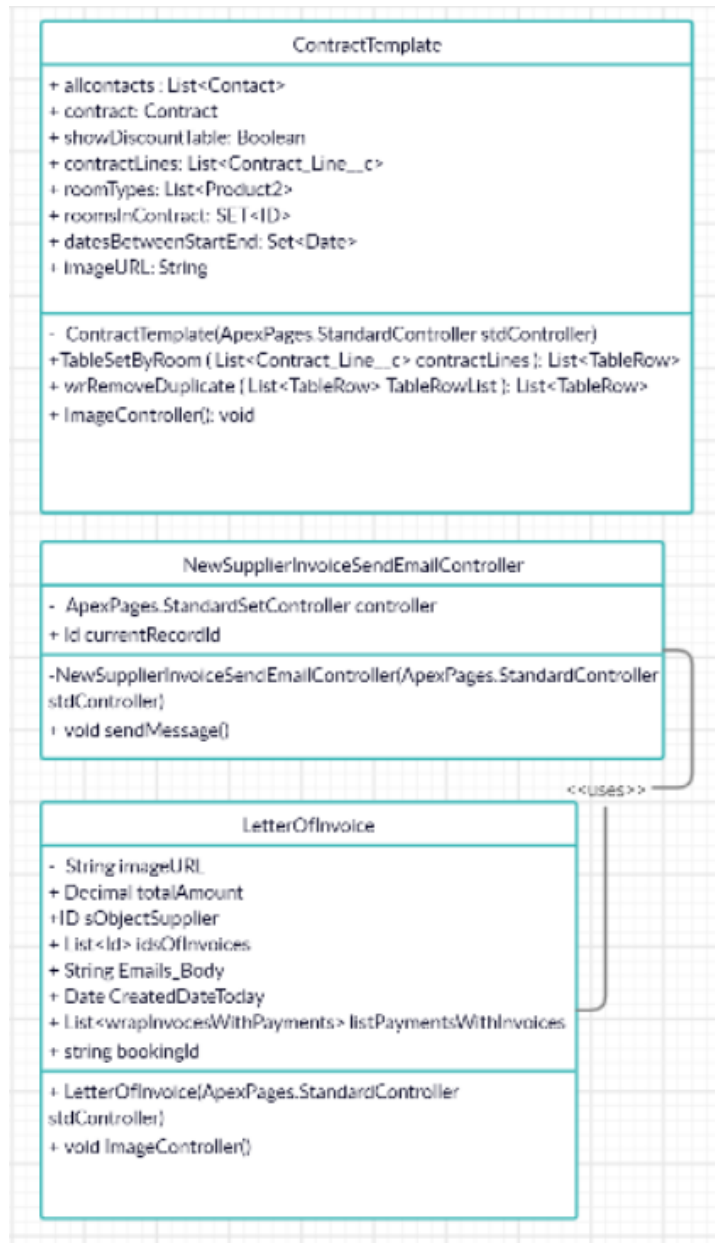


Рисунок 3.5 – Додаткові класи представлені на діаграмі класів

Окрім того, при розробці програмного забезпечення для дипломного проектування, мною був використаний Lightning Component Framework. За його допомогою були реалізовані компоненти пошуку та відображення нерухомості. Данні компоненти складаються зі зв'язки декількох файлів, а саме: component.cmp, controller.js, helper.js, style (містить кастомні css) класи.

### 3.1.5 Діаграма діяльності

При моделюванні поведінки системи виникає необхідність деталізувати особливості алгоритмічної та логічної реалізації операцій. Традиційно з цією метою використовують блок-схеми або структурні схеми алгоритмів. Кожна така схема акцентує увагу на послідовності виконання певних дій (або елементарних операцій), які в сукупності спричиняють отримання бажаного результату.

Для моделювання процесу виконання операцій у мові UML використовують діаграму діяльності, яка зображується графом, вершинами якого є стани (дій і/або видів діяльності), а дугами – переходи від одного стану (дій/виду діяльності) до іншого стану (дій/виду діяльності) [37].

Головним напрямом використання діаграм діяльності є візуалізація особливостей реалізації операцій класів, коли необхідно зобразити алгоритми їхнього виконання. Оскільки, в даному web-застосунку можливості для користувачів з різними правами доступу відрізняються, окремо була розроблена діаграма діяльності для двох ролей: клієнт та адміністратор.

Діаграма діяльності до дипломного проекту для адміністратора зображена на рисунках 3.6- 3.7:

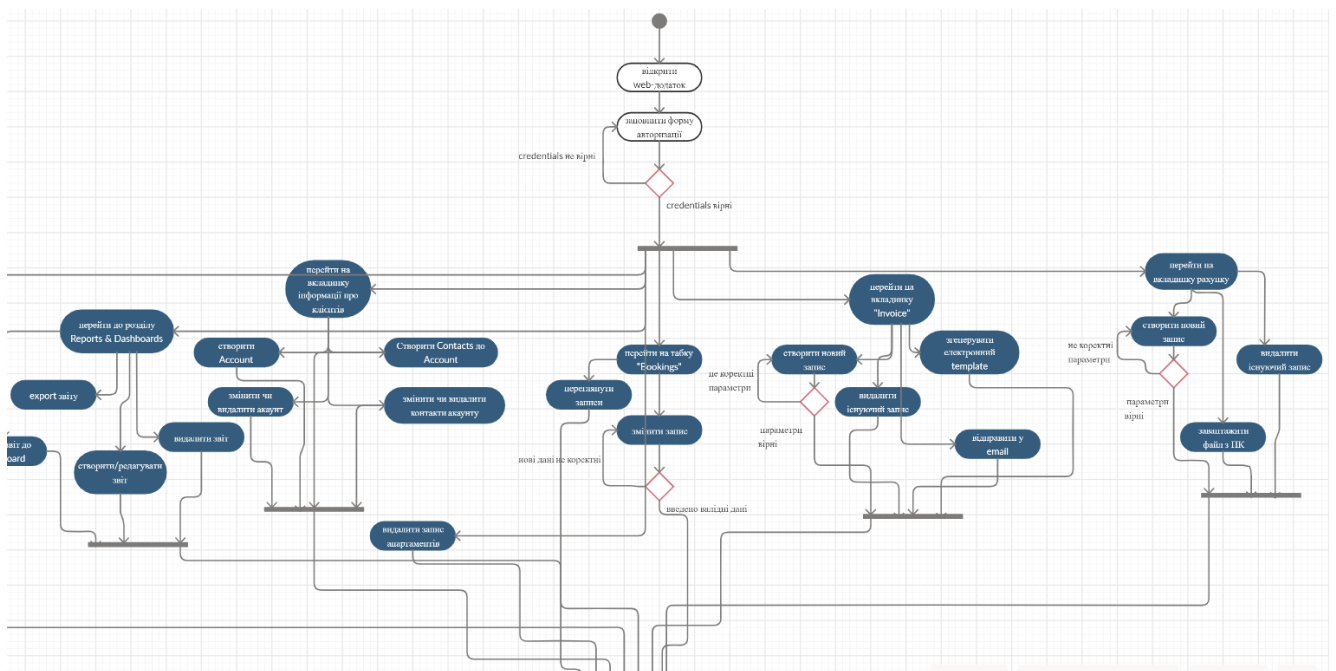


Рисунок 3.6 – Початок діаграми діяльності для адміністратора системи

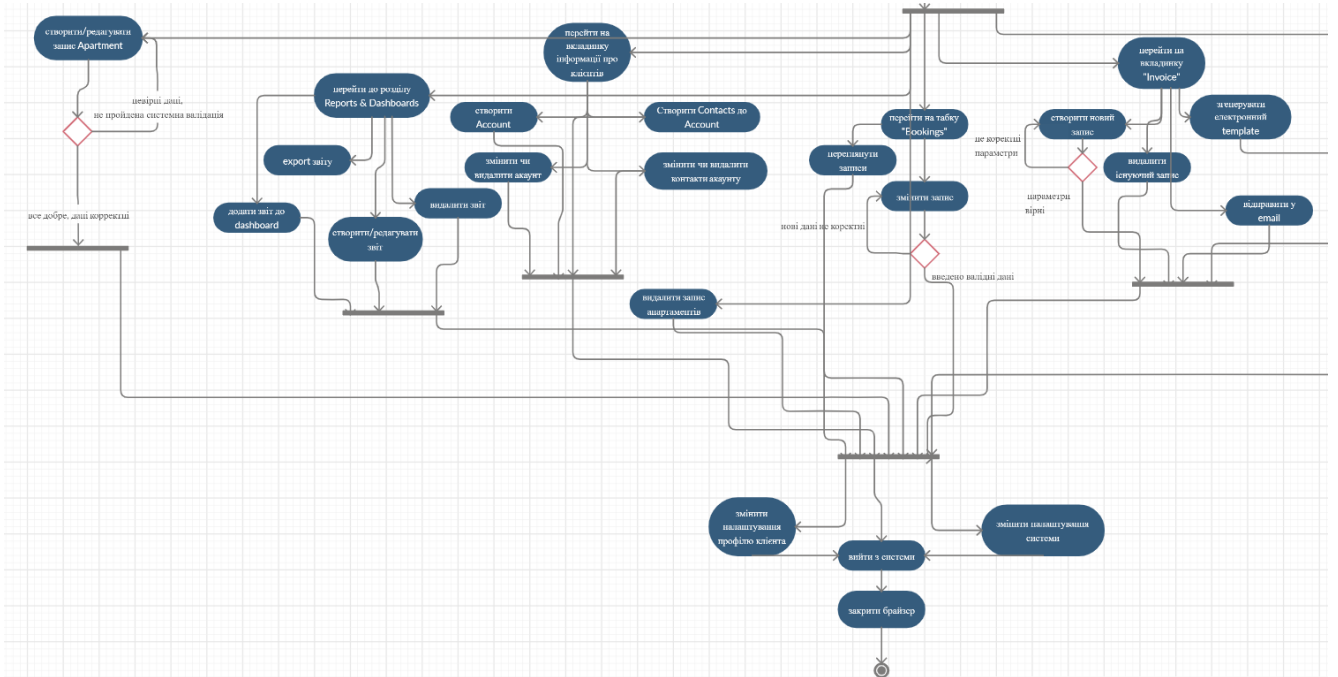


Рисунок 3.7 – Продовження діаграми діяльності для адміністратора системи

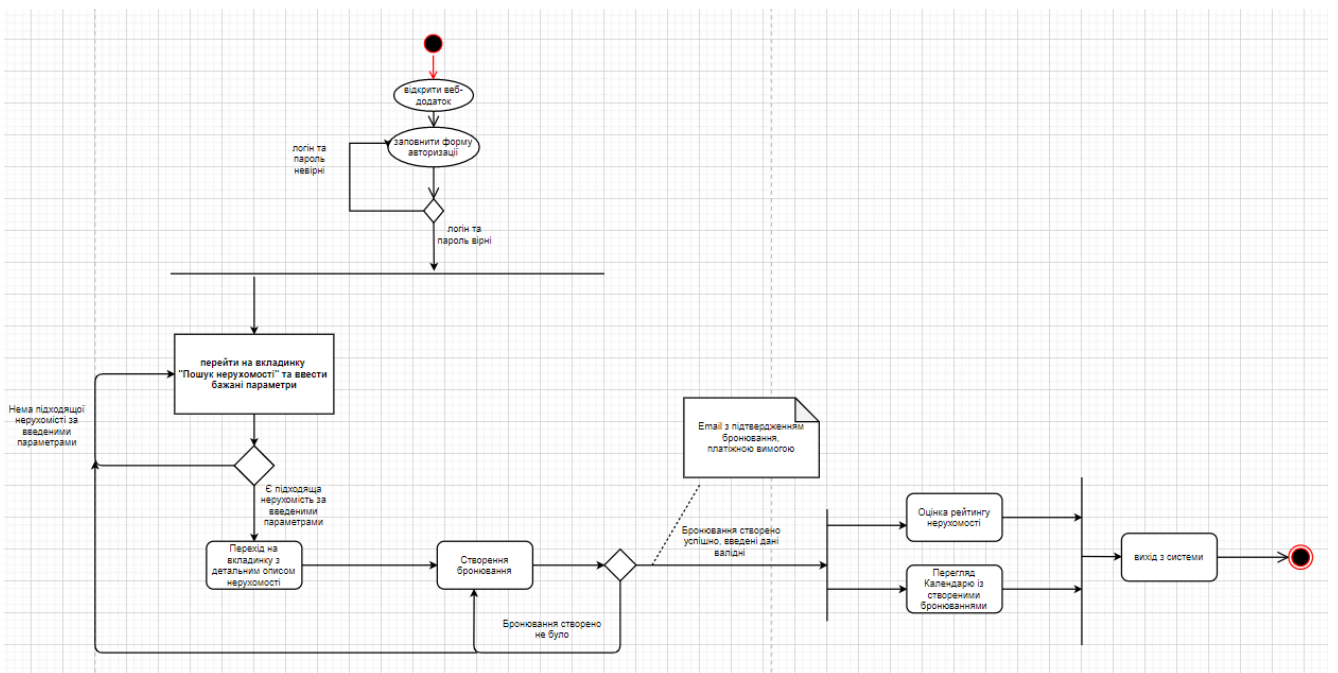


Рисунок 3.8 – Діаграма діяльності для клієнта системи

### 3.1.6 Діаграма послідовності дій

Діаграма послідовності дій (англ. sequence diagram) — це різновид діаграми в UML, що відображає взаємодії об'єктів впорядкованих за часом [49].

На діаграмі послідовності зображуються виключно ті об'єкти, які безпосередньо беруть участь у взаємодії і не показуються можливі статичні асоціації з іншими об'єктами. Для діаграми послідовності ключовим моментом є саме динаміка взаємодії об'єктів в часі.

Діаграми послідовностей можна використовувати для уточнення діаграм прецедентів, більш детального опису логіки сценаріїв використання. Це відмінний засіб документування проекту з точки зору сценаріїв використання. Діаграми послідовностей зазвичай містять об'єкти, які взаємодіють у рамках сценарію, повідомлення, якими вони обмінюються, і які повертають результати, що пов'язані з повідомленнями.

Для діаграми послідовності ключовим моментом є саме динаміка взаємодії об'єктів в часі. При цьому діаграма послідовності має як би два виміри. Одне — зліва направо у вигляді вертикальних ліній, кожна з яких зображує лінію життя окремого об'єкта, який бере участь у взаємодії. Графічно кожен об'єкт зображується прямокутником і розташовується у верхній частині своєї лінії життя.

Другий вимір діаграми послідовності – вертикальна тимчасова вісь, спрямована зверху вниз. Початкового моменту часу відповідає сама верхня частина діаграми. При цьому взаємодії об'єктів реалізуються за допомогою повідомлень, які надсилаються одними об'єктами іншим. Повідомлення зображуються у вигляді горизонтальних стрілок з ім'ям повідомлення.

На рисунку 3.9 зображена діаграма послідовності дій для адміністратора.

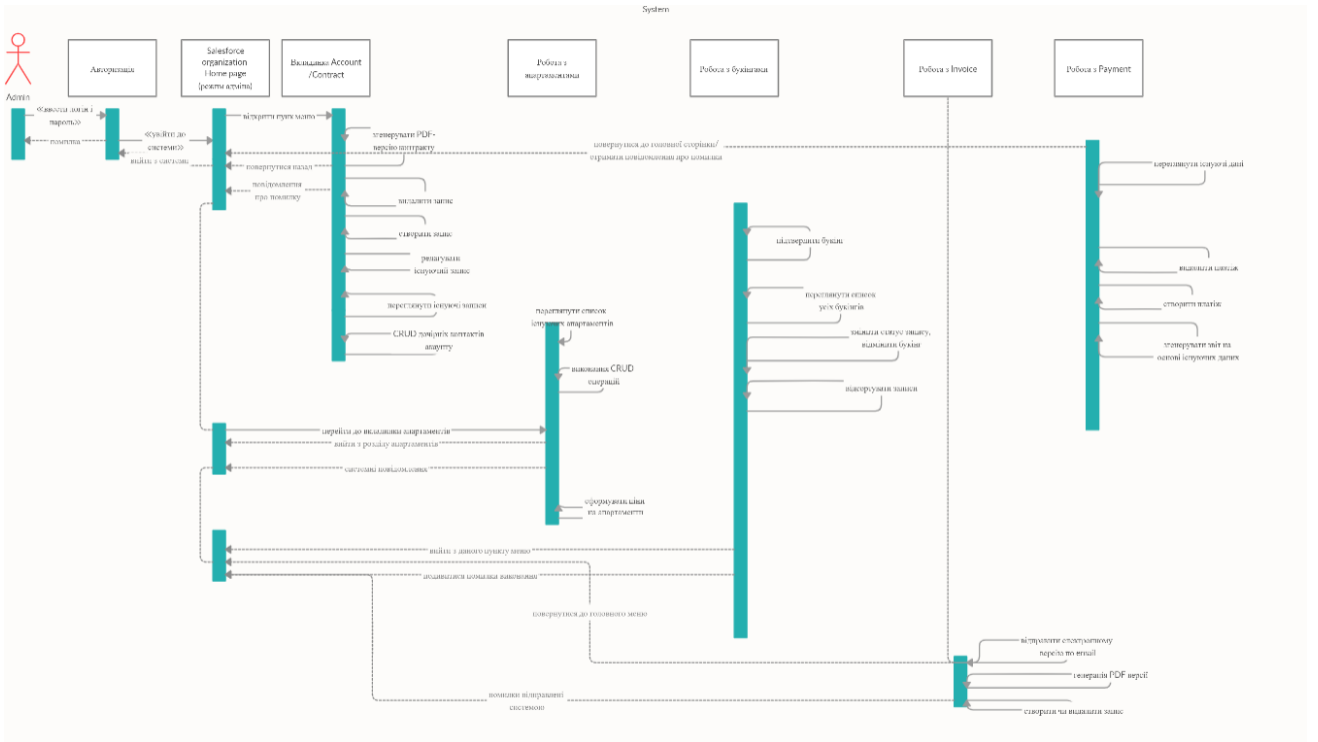


Рисунок 3.9 – Діаграма послідовності дій для адміністратора

На ній видно, що відкривши веб-застосування, адміністратор має спочатку авторизуватися у пункті меню «Авторизація», а потім вже в режимі адміністратора, він зможе працювати в одній з багатьох запропонованих вкладок, наприклад:

Робота з вкладинкою «Партнери», де він може заповнити форму створення записів, редагувати їх чи генерувати електронні документи;

«Букінги», де він може переглянути інформацію про наявні бронювання та редагувати її.

На діаграмі послідовності дій для клієнта (рисунок 3.10) зображено, що відкривши веб-застосування, клієнт також може працювати в одній з декількох запропонованих табок, наприклад:

«Пошук нерухомості», де користувач може обрати один з декількох пунктів: «Компонент пошуку апартаментів», «Компонент перегляду детальної інформації про апартаменти», «Компонент створення букінгу» і в них створювати бронювання, сортувати та шукати потрібну інформацію в залежності від обраного пункту меню;

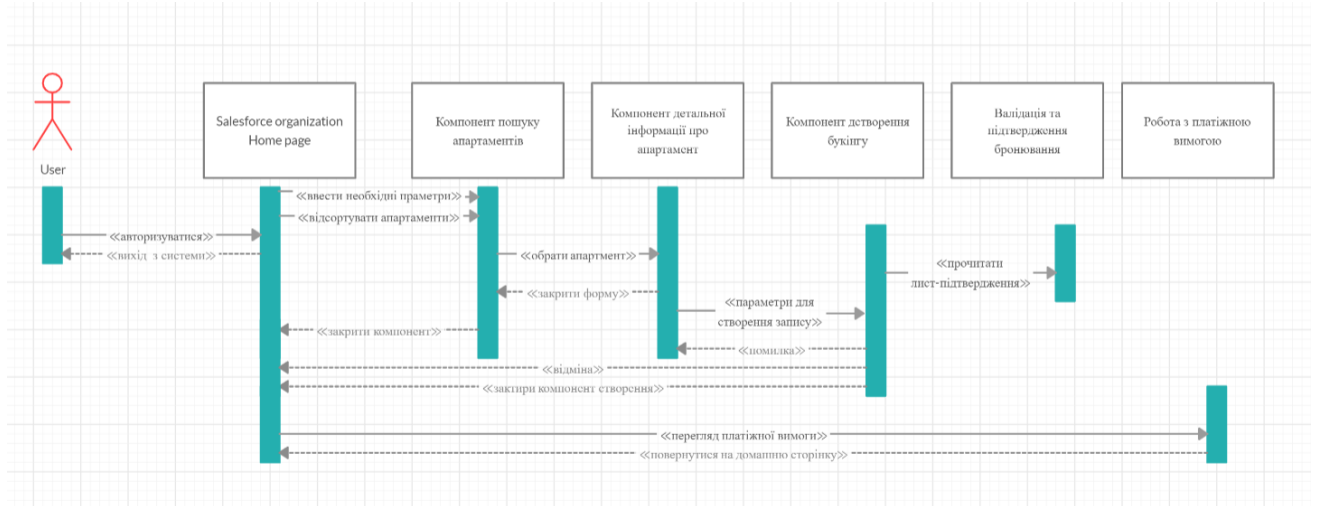


Рисунок 3.10 – Діаграма послідовності дій для клієнта

## 3.2 Обґрунтування вибору засобів розробки

### 3.2.1 Apex

Apex – це мова програмування, яка використовує синтаксис, схожий на Java та C#, і діє як процедури, збережені в базі даних. Apex дозволяє розробникам отримувати доступ до back-end бази даних Salesforce.com та інтерфейсів клієнт-сервера для створення сторонніх програм SaaS [48].

Apex дозволяє розробникам автоматизувати бізнес-логіку та додати її до системних подій, таких як натискання кнопок, оновлення відповідних стандартних і кастомних записів і сторінок Visualforce. Як мова програмування, Apex:

- hosted, Apex зберігається, компілюється і виконується на сервері Lightning Platform.

- є об'єктно-орієнтованою, адже Apex підтримує основні парадигми об'єктно-орієнтованого програмування: класи, інтерфейси та успадкування.

- строго типізована, бо Apex перевіряє посилання на об'єкти під час компіляції.

- Multi Tenant Aware – тому що Apex виконується на багатопрофільній платформі, на якій декілька користувачів можуть одночасно працювати над розробкою функціоналу. Виходячи з цього система повинна розмежувати використання середовища розробки різними користувачами, застосовуючи обмеження, які перешкоджають монополізації коду спільними ресурсами.

- підтримує процес інтеграції з базою даних – це простий доступ до записів і маніпулювання ними. Apex забезпечує прямий доступ до записів та їх полів, а також надає оператори та декларативну мову програмування запитів SOQL для керування цими записами.

Орієнтована на дані – Apex забезпечує транзакційний доступ до хмарної бази даних, що дозволяє відхиляти запущені операції. Проста у використанні – мова Apex базується на знайомих для всіх ідіомах Java. Легко пересвідчитися, що Apex надає вбудовану підтримку для створення, виконання та покриття клієнтського коду за допомогою Unit-тестів.

Salesforce гарантує, що весь користувальницький код Apex працює як слід, виконуючи всі модульні тести перед будь-якими оновленнями платформи.

### 3.2.2 JavaScript

JavaScript – мультипарадигменна мова програмування. Підтримує об'єктно-орієнтований, імперативний і функціональний стилі. Є реалізацією мови ECMAScript.

Основні архітектурні риси: динамічна типізація, слабка типізація, керування пам'яттю, прототипне програмування, функції інтерпретуються в якості об'єктів.

JavaScript зазвичай використовується в клієнтській частині веб-додатків: клієнт серверних процесів, в яких клієнтом є браузер, а сервером – веб-сервер, що мають розподілену між сервером і клієнтом логіку. Обмін інформацією в веб рішеннях відбувається по мережі [43]. Одним з переваг такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому веб-додатки є кроссплатформенними сервісами. Мова JavaScript має C-подібний синтаксис.

Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення односторінкових веб-додатків (особливо фреймворки ReactJS, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- імплементації стаціонарних додатків (Electron, NW.js);
- реалізації мобільних додатків (React Native, Cordova);
- опису сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite);

Хоча ця мова програмування з'явилася ще в 1995 році, вона є дуже популярною і у наш час.

### 3.2.3 Мова розмітки гіпертексту HTML та Visualforce

HTML (від англ. HyperText Markup Language - «мова гіпертекстової розмітки») – стандартизована мова розмітки документів у Всесвітній павутині. Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML).

Мова HTML інтерпретується браузерами; отриманий в результаті інтерпретації форматований текст відображається на екрані монітора комп'ютера



або мобільного пристрою. Мова HTML була розроблена британським вченим Тімом Бернерс-Лі приблизно в 1986-1991 роках в стінах ЦЕРНу в Женеві, Швейцарія. HTML створювалася як мова для обміну науковою і технічною документацією, придатна для використання людьми, які не є фахівцями в області верстки.

Мова розмітки web-сторінок, перш за все, потрібна для:

- створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- отримання інформації із Всесвітньої мережі через гіперпосилання;
- створення інтерактивних форм;
- включення зображень, звуку, відео, та інших об'єктів до тексту.

Для гарного оформлення web-додатку потрібно додавати стилі. Для цього можна застосовувати мову CSS.

CSS (каскадні таблиці стилів) – формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки. Переважно використовується як засіб опису, оформлення зовнішнього вигляду веб-сторінок, написаних за допомогою мов розмітки HTML і XHTML, але може також застосовуватися до будь-яких XML-документах, наприклад, до SVG або XUL.

CSS використовується творцями веб-сторінок для задання кольорів, шрифтів, розташування окремих блоків і інших аспектів представлення зовнішнього вигляду веб-сторінок. Основною метою розробки CSS було розділення опису логічної структури веб-сторінки (яке проводиться за допомогою HTML або інших мов розмітки) від опису зовнішнього вигляду цієї веб-сторінки (яке тепер проводиться за допомогою формальної мови CSS) [21].

Для розробки під платформу Salesforce спільно з описаними вище технологіями необхідно використовувати технологію компонентів Visualforce, які представляють собою невеликі функціональні фрагменти, придатні для багаторазового використання (інтелектуальні віджети, панелі, елементи призначеного для користувача інтерфейсу і т. п.),

Вбудовані компоненти застосовуються при розмітці сторінок Visualforce. Стандартні компоненти Visualforce можна використовувати для створення власних програм та веб-сторінок, що настроюються на розробляються виключно на базі існуючих Visualforce компонентів.

### 3.2.4 SOQL

Щоб «прочитати» запис з бази даних Salesforce, потрібно написати відповідний запит до back-end сховища системи. Salesforce надає своїм

розробникам та користувачам вбудовані мови запитів об'єктів Salesforce SOSL або SOQL, які можна використовувати для читання збережених у систему записів.

SOQL подібна до стандартної мови SQL [16], але налаштована для платформи Lightning. Оскільки Apex має прямий доступ до записів Salesforce, які зберігаються в базі даних, ви можете вбудувати SOQL-запити у ваш код Apex і отримувати результати у простий та зручний спосіб. SOQL вбудований в Apex, називається Inline SOQL [50]

### 3.2.5 Lightning Component Framework

Даний фреймворк відносно нещодавно був представлений на ринку платформи Salesforce [29]. Це UI framework призначений для розробки одно сторінкових застосунків для десктопних і мобільних пристроїв. Станом на реліз Spring'19 (API версії 45.0) можна будувати компоненти Lightning за допомогою двох моделей програмування: моделі Lightning Web Components та оригінальної моделі Aura Components. Lightning web компоненти це кастомні HTML-елементи, побудовані за допомогою HTML та сучасного JavaScript.[46] Lightning web components та Aura components можуть співіснувати та взаємодіяти на одній сторінці.

Налаштувати розроблені веб-компоненти Lightning та Aura компоненти для роботи можна в Lightning App Builder та Experience Builder. Адміністратори та кінцеві користувачі не знають, яка модель програмування використовується для розробки компонентів. Для кінцевих користувачів всі програмні модулі виглядають та функціонують однаково.

Використання інфраструктури Lightning описаної вище для розробки компонентів і застосунків надає цілий ряд переваг, зокрема: стандартизація, спільна екосистема з безліччю інших компонентів, прискорення процесу розробки, сумісність із різними пристроями та сумісність між веб-браузерами.

### 3.2.6 Вибір цільового варіанту архітектури програмного забезпечення

За для вибору архітектури, спочатку необхідно розглянути існуючі варіанти. «Товстий» клієнт (richclient) – це додаток, що забезпечує повну функціональність та незалежність від центрального сервера [44].

Як правило, сервер в цьому випадку є лише сховищем даних, а вся робота по обробці цих даних переноситься на машину клієнта. Товстий клієнт має повну функціональність для процесу обробки даних серверу, забезпечує режим роботи багатьох користувачів, надає можливість роботи навіть при обривах зв'язку з сервером, має високу швидкодію.

Проте, при роботі можуть виникати проблеми з віддаленим доступом до даних, що виражаються в складності оновлення даних, їх узгодження з іншими

клієнтами і, як результат, пов'язаним з цим неактуальністю даних. Також, «товстий» клієнт має досить складний процес установки та налаштування; має низьку масштабованість.

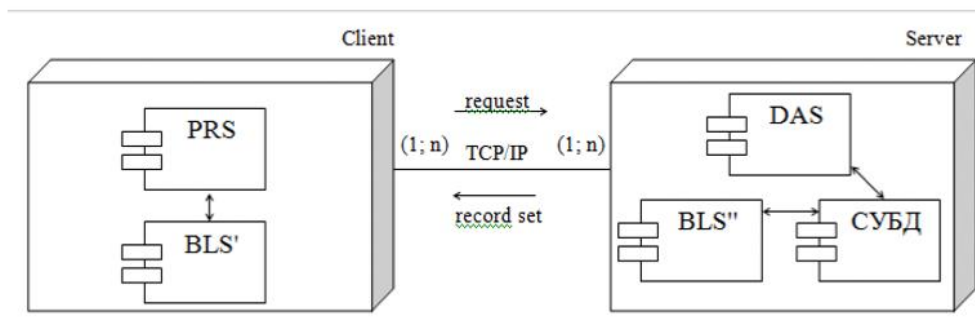


Рисунок 3.11 – Системна архітектура «товстий» клієнт

«Тонкий» клієнт (thinclient) – комп'ютер або програма-клієнт в мережах з клієнт-серверною архітектурою, де велика частина завдань по обробці інформації перенесена на сервер, права доступу клієнта строго обмежені [45]. Прикладом тонкого клієнта може служити комп'ютер з браузером, який використовується для процесу веб-додатками.

Така архітектура дозволяє зменшити затрати на супроводження, зростає масштабованість системи. З недоліків можна відмітити велику завантаженість серверу, що призводить до зниження продуктивності.

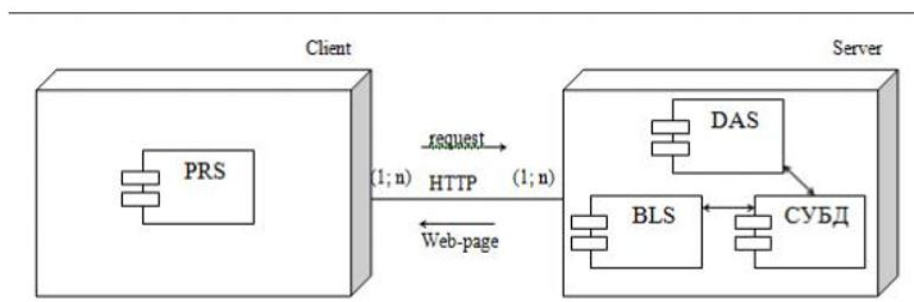


Рисунок 3.12 – Системна архітектура «тонкий» клієнт

Трирівнева еталонна системна архітектура забезпечує повне фізичне розділення програмних сервісів і необхідних для них обчислювальних ресурсів. Така системна архітектура дозволяє досягти максимального масштабування системи, розвантажити сервер БД та збільшити продуктивність.

До недоліків можна віднести складність в розгортанні та адмініструванні такої системи, високі вимоги до продуктивності серверів додатків й серверу бази

даних. Базова архітектура трирівневий «клієнт-сервер» з виділеним сервером додатків зображена на рисунку

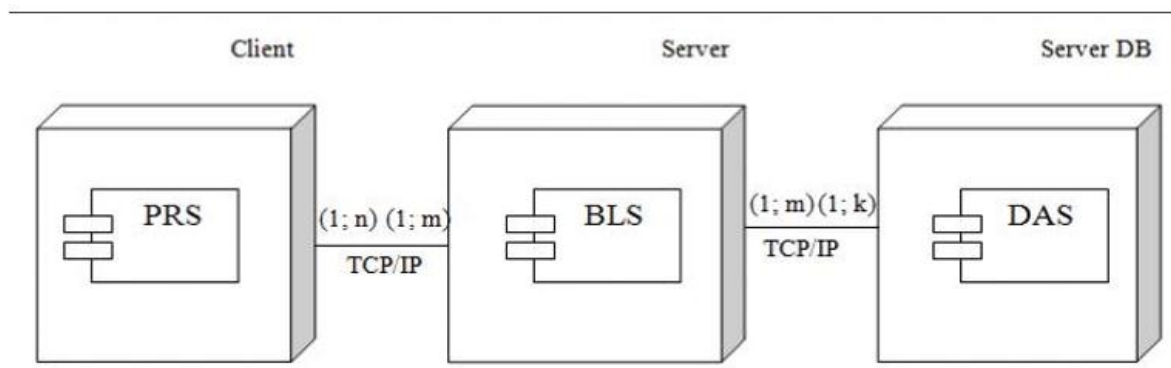


Рисунок 3.13 – Системна архітектура трирівневий «клієнт-сервер»

Програмне забезпечення проектується та розробляється для роботи в браузері на персональних комп'ютерах. Система повинна виконувати багато дій поза очима користувача: перевірка прав доступу, надсилання листів на пошту, сортування та фільтрація значних об'ємів даних.

При цьому система повинна бути розподіленою та відказостійкою. Тому для цієї задачі доцільно використати архітектуру «тонкий» клієнт, щоб перенести значну частину обробки даних на сервер, але занадто не збільшувати складність розробки та підтримки системи.

### 3.3 Розробка функціоналу програмного продукту

Основними одиницями в системі – є нерухомість, букінги та платежі. Об'єкт нерухомість включає все необхідні поля: назва, місце знаходження, характеристики приміщення. Також додатково були розроблені окремі вкладинки для більш зручного та зручного відображення інформації. Вікно створеного запису типу Нерухомість зображено на рисунку 3.14.

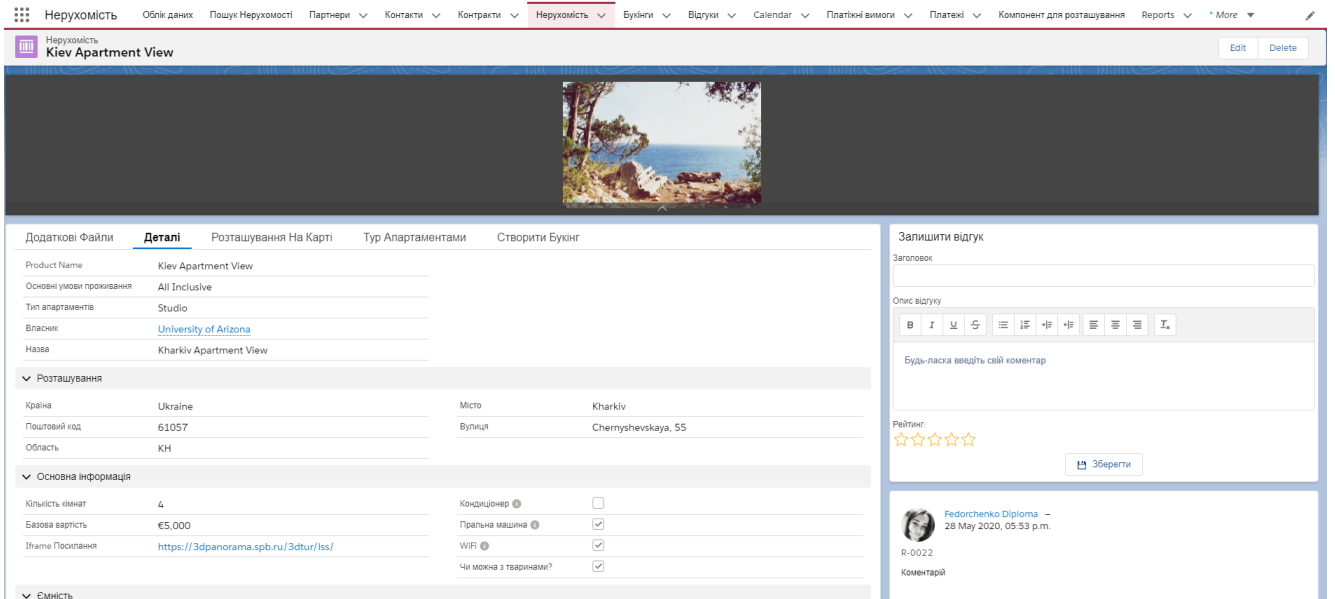


Рисунок 3.14 – Форма відображення запису типу Нерухомість/Product

Вікно, для відображення розташування нерухомості на карті зображено на рисунку 3.15:

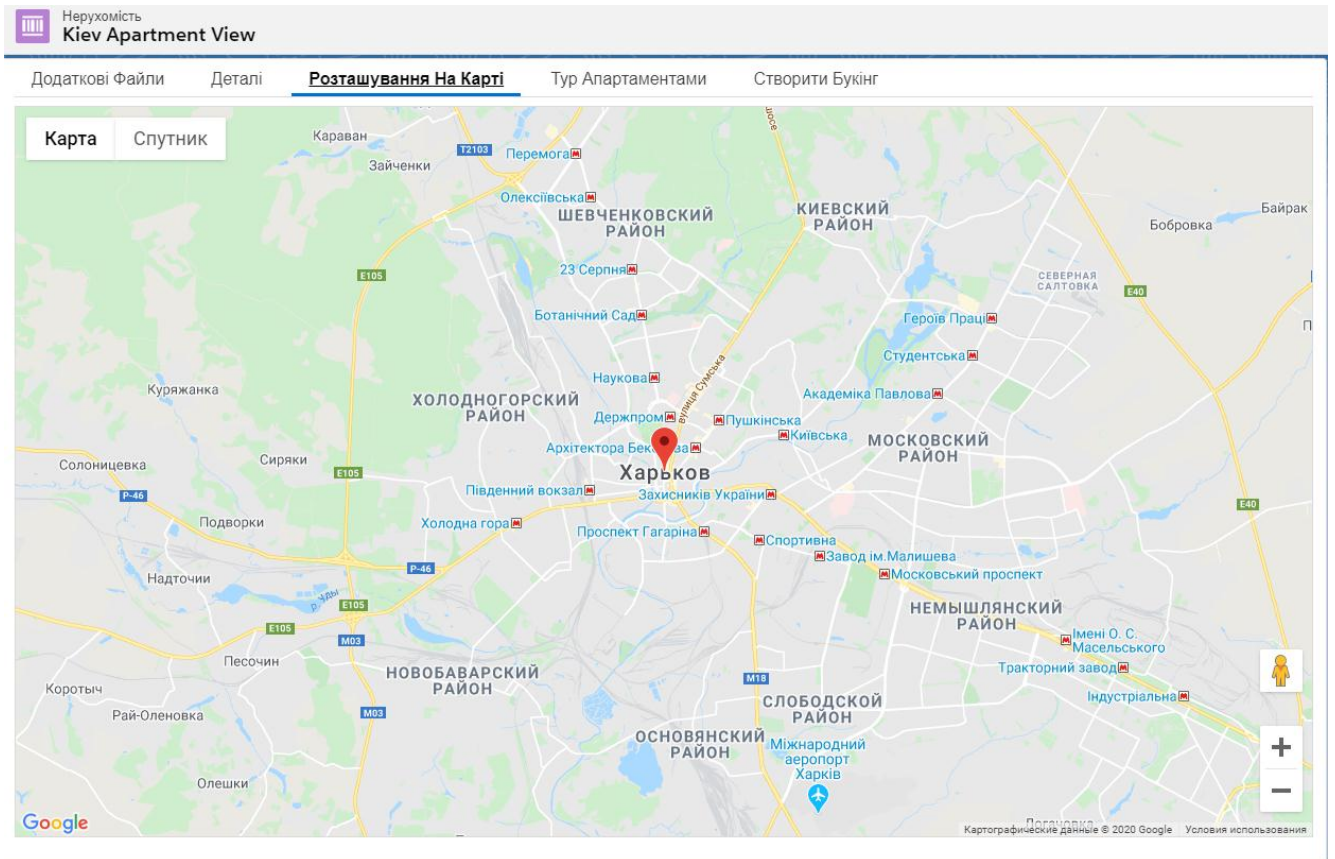


Рисунок 3.15 – Відображення розташування нерухомості на Google map

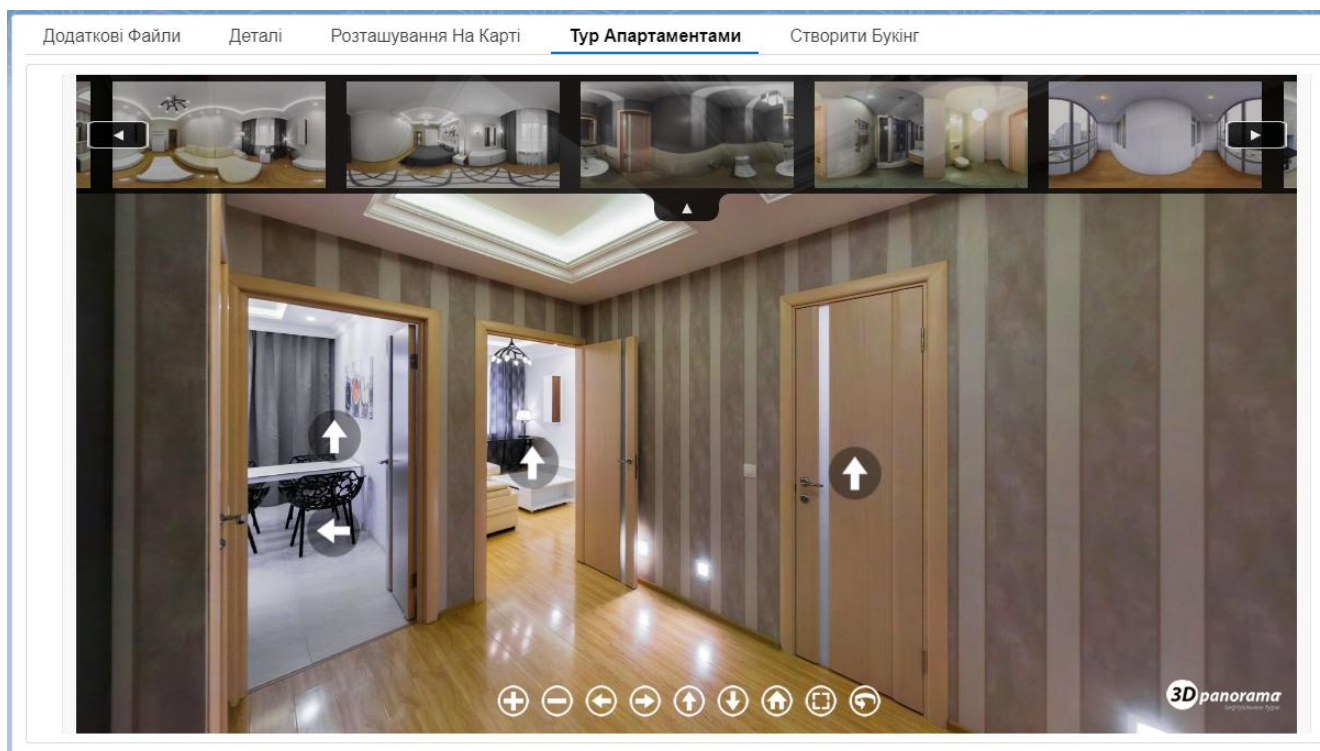


Рисунок 3.16 – Вкладка для проходження 3D туру апартаментами

Для того аби створити запис про нерухомість, адміністратор спочатку має увійти у систему. Необхідними складовими для створення нерухомості також є наявність існуючого запису про орендодавця.

Аби внести до системи інформацію про ціни за оренду тієї чи іншої позиції адміністратор має створити запис типу контракт, а вже після цього він матиме можливість вводити інформацію про ціни на певні періоди створеного контракту. Інтерфейс даного функціоналу зображений на рисунках нижче:

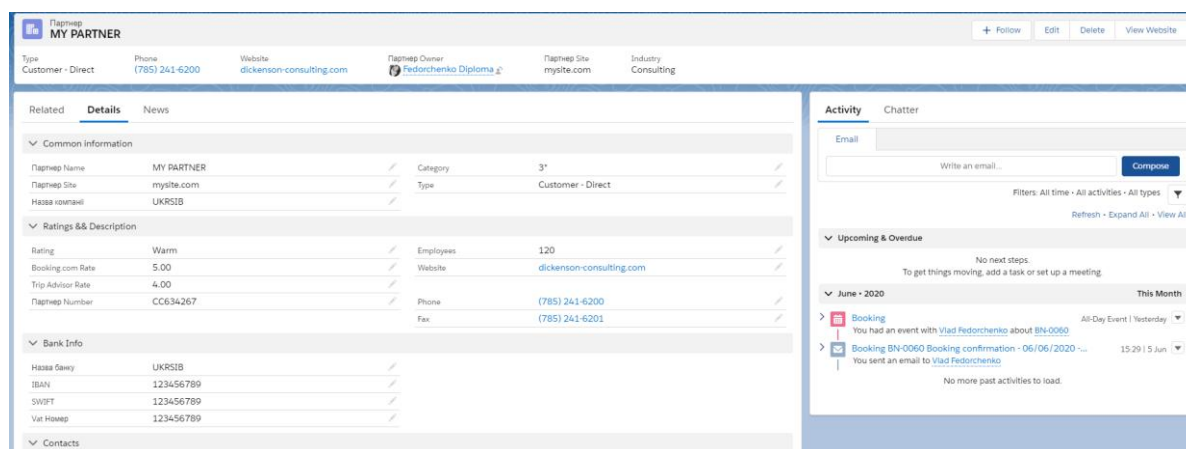


Рисунок 3.17 – Сторінка редагування запису типу Account

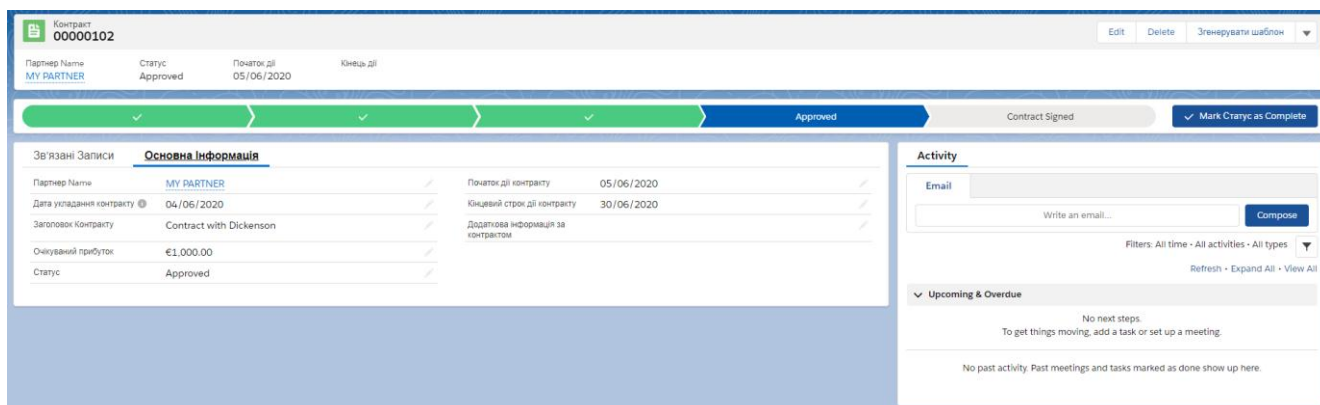


Рисунок 3.18 – Сторінка для перегляду та редагування Контракту

Як було зазначено раніше, із кожним Партнером компанія складає Контракт. На Detail сторінці Контракту можна редагувати запис, та відстежувати статус Контракту. Також система відображає всі зв'язки певного контракту із іншими записами, існуючими у системі:

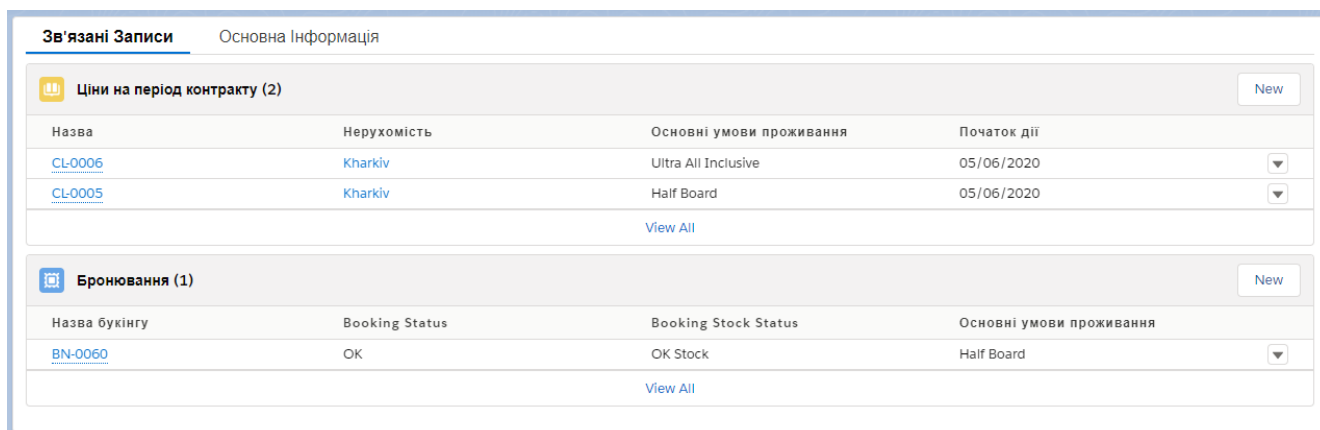


Рисунок 3.19 – Сторінка для перегляду Related Lists для Контракту

Вартно зазначити, що безпосередньо на сторінці контракту існує дві кастомні кнопки: «Задати вартість» (для створення цін на певний період дій контракту) та «Згенерувати шаблон» (для створення .pdf версій поточного запису).

У подальшому згенерований шаблон користувач може зберегти на свій ПК або ж роздрукувати.

Information

Контракт: 00000102  
Нерухомість: Kharkiv

Власник нерухомості: MY PARTNER  
Основні умови проживання: Room Only

Prices

Мінімальний термін проживання: 15  
Початок дії: 05/06/2020

Double Occupancy per Night/Room: 100.00

Single Occupancy per Night/Room: 70.00

Baby (0-2) per Night/Pax: 0.00

Rack Price per Night: 50.00

Максимальний термін проживання: 22  
Кінець дії: 30/06/2020

First Child (2-12) per Night/Pax: 10.00

Second Child (2-12) per Night/Pax: 25.00

Third Adult per Night/Pax: 35.00

Fourth Adult per Night/Pax: 35.00

Information

Ємність: 5TK5  
Кількість вільного місця: 5

Close Save

Рисунок 3.20 – Сторінка для створення Contract Line

Після того, як вся необхідна інформація про нерухомість була занесена до БД, кінцеві користувачі зможуть, користуючись компонентом пошуку, знайти нерухомість яка найбільше відповідатиме їх потребам. Інтерфейс форми пошуку представлений на рисунку 3.21

Нерухомість Облік даних Пошук Нерухомості Партнери Контакти Контракти Нерухомість Бюкінг Відгуки Calendar Платіжні вимоги Платіжі Компонент для розташування Reports More

Знайти нерухомість

Studio

Базові умови проживання

Введіть кількість кімнат...

Введіть місто...

Введіть вулицю...

Шукати

Кандіціонер  Наявний

Пральня машина  Є

Wi-Fi  Є

Ліфт  Наявний

Ціна 1000-5000

Результат пошуку

Рисунок 3.21 – Вкладка «Пошук нерухомості»

Одразу після вибору вподобаної нерухомості користувач може перейти безпосередньо до процесу створення бронювання. Система перевіряє всі поля вводу даних, та повідомляє користувача у разі виникнення будь-яких помилок. Також варто зазначити, що була імплементована кнопка, яка за заданими параметрами перевіряє чи є у системі підходяще вільне місце і виводить користувачеві повідомлення про результат.

Завдяки цьому у системі не зберігаються помилкові записи, а користувач одразу може отримати необхідну інформацію щодо зробленої ним заяви на бронювання. Інтерфейс створення букінгу представлений на рисунку 3.22



Рисунок 3.22 – Вікно створення заявки на бронювання

На наступних скріншотах представлений процес створення адміністратором системи платіжних вимог. Якщо якісь параметри були введені помилково, система відобразить повідомлення про помилку.

Також варто зазначити, що на Detail page запису платіжної вимоги є кнопка, за допомогою якої адміністратор може відправити .pdf версію платіжної вимоги на пошту орендаторові.

Information				
Партнер		COMMON PARTNER		Company Name
Дата складання вимоги		07/06/2020		Номер Вимоги
Тип вимоги		Invoice		Платіжна вимога v2.1
SELECT	BOOKING NUMBER	PAX FIRST NAME	PAX LAST NAME	CALCULATED PURCHASING PRICE AMOUNT
<input checked="" type="checkbox"/>	BN-0056	Olga	Olga	210.00
<input type="button" value="Next"/> <input type="button" value="Close"/>				

Рисунок 3.23 – Вікно створення платіжної вимоги

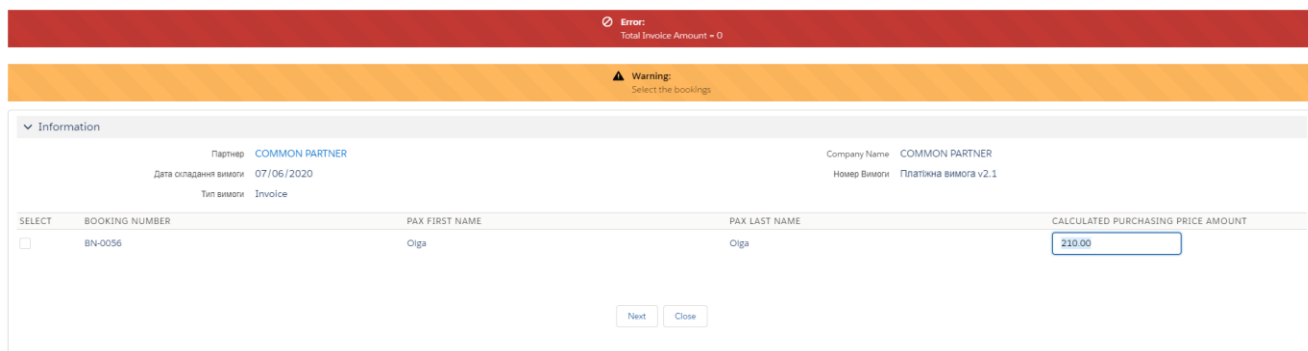


Рисунок 3.24 – Вікно відображення повідомлення про помилку під час створення платіжної вимоги

За аналогією адміністратор матиме змогу створити запис, що містить у собі інформацію про платежі, зроблені клієнтами. На основі усіх даних, що зберігаються у БД системи, було побудовано різноманітні звіти, та згруповано їх на одному Dashboard для зручності пошуку і сприйняття.

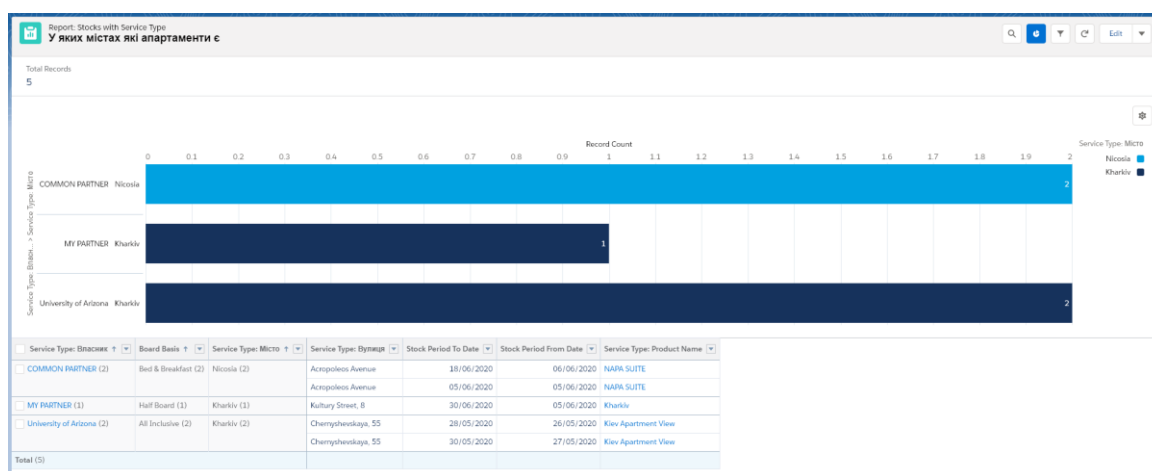


Рисунок 3.25 – Вікно відображення звіту з графіком для візуалізації нерухомості та основних її показників

Система генерує наступні звіти і розміщає її на вкладинці «Облік» під профілем адміністратора:

- З інформацією про неоплачені букінги
- З інформацією про нерухомість та її розташування (згруповано за Партнерами та містами)
- З інформацією про платіжні вимоги, що потрібно сплатити/надіслати Клієнтам протягом наступних 30ти днів
- Зі списком контрактів строк дії яких спливає за 30 днів

- З інформацією про вільну ємність всіх наявних апартаментів
- Звіт що відображає статуси букінгів зареєстрованих у системі
- Інформацію щодо платежів, чи були вони сплачені та кінцевий строк оплати
- Інформацію щодо загальної суми усіх наявних платежів, а також середнє, мінімальне та максимальне значення
- Дані про букінги розсортовані за місяцями для зручного сприйняття та візуалізації
- Рейтинг нерухомості за кількістю відгуків
- Суму, що залишилося отримати від здачі букінгів для того, аби отримати очікуваний прибуток по контракту

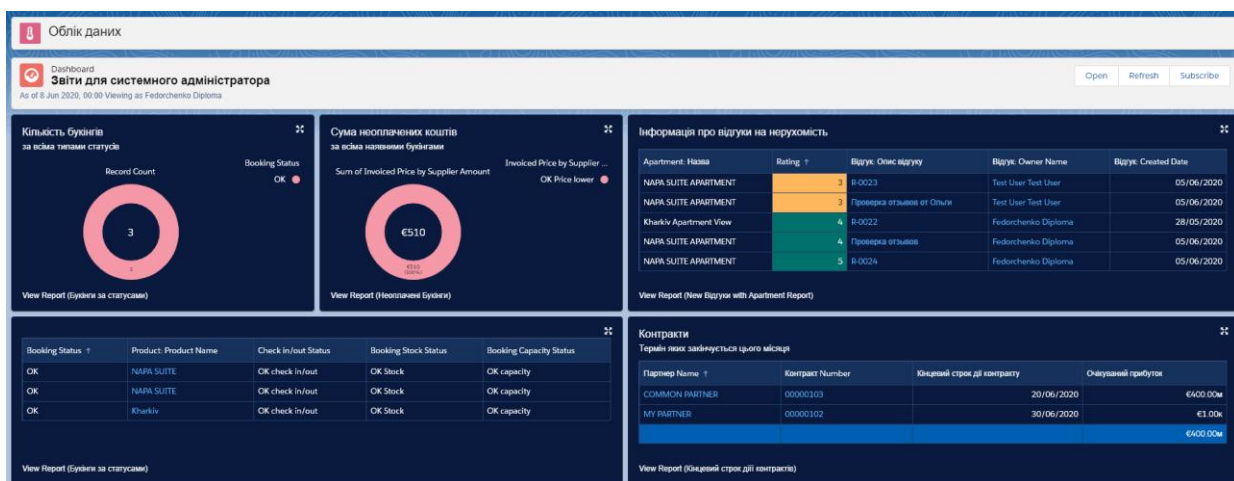


Рисунок 3.26 – Вікно відображення Dashboard зі згрупованими звітами

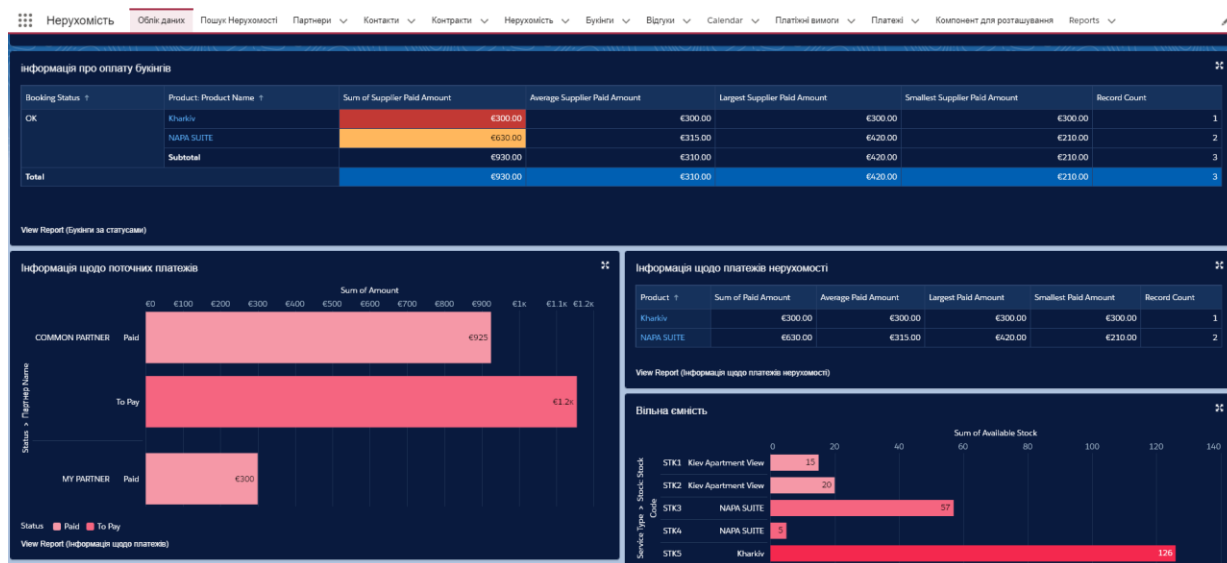


Рисунок 3.27 – Вікно відображення Dashboard зі згрупованими звітами

На скріншотах представлених нижче можна побачити застосунок для Клієнтів системи (вони не мають доступу до вкладинок із фінансовою інформацією).



Рисунок 3.28 – Вікно відображення пунктів меню для Клієнта

Сторінка Календар, на якій адміністратор системи може бачити всі створені клієнтами букінги: Клієнти системи мають доступ тільки до власних букінгів.

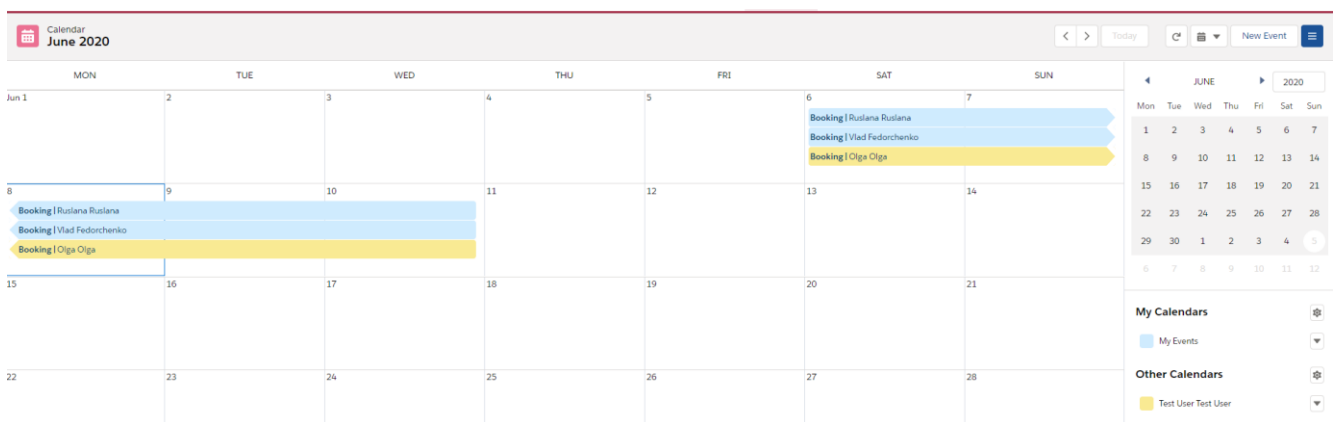


Рисунок 3.29 – Вікно відображення Календарю для адміністратора

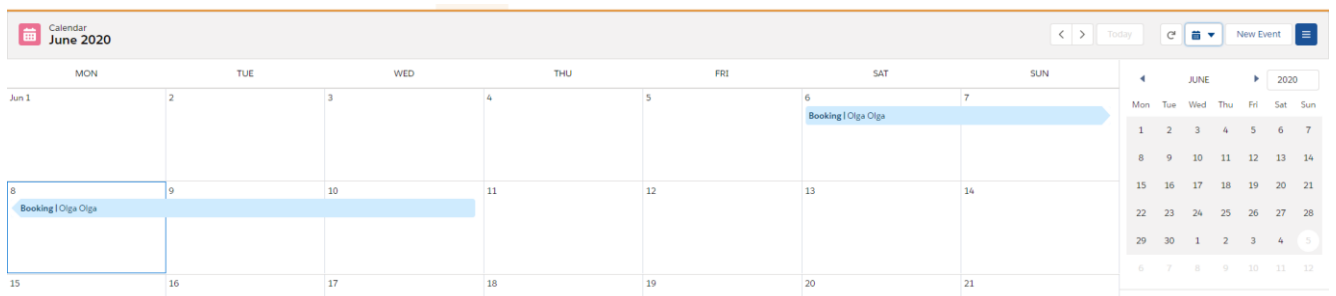
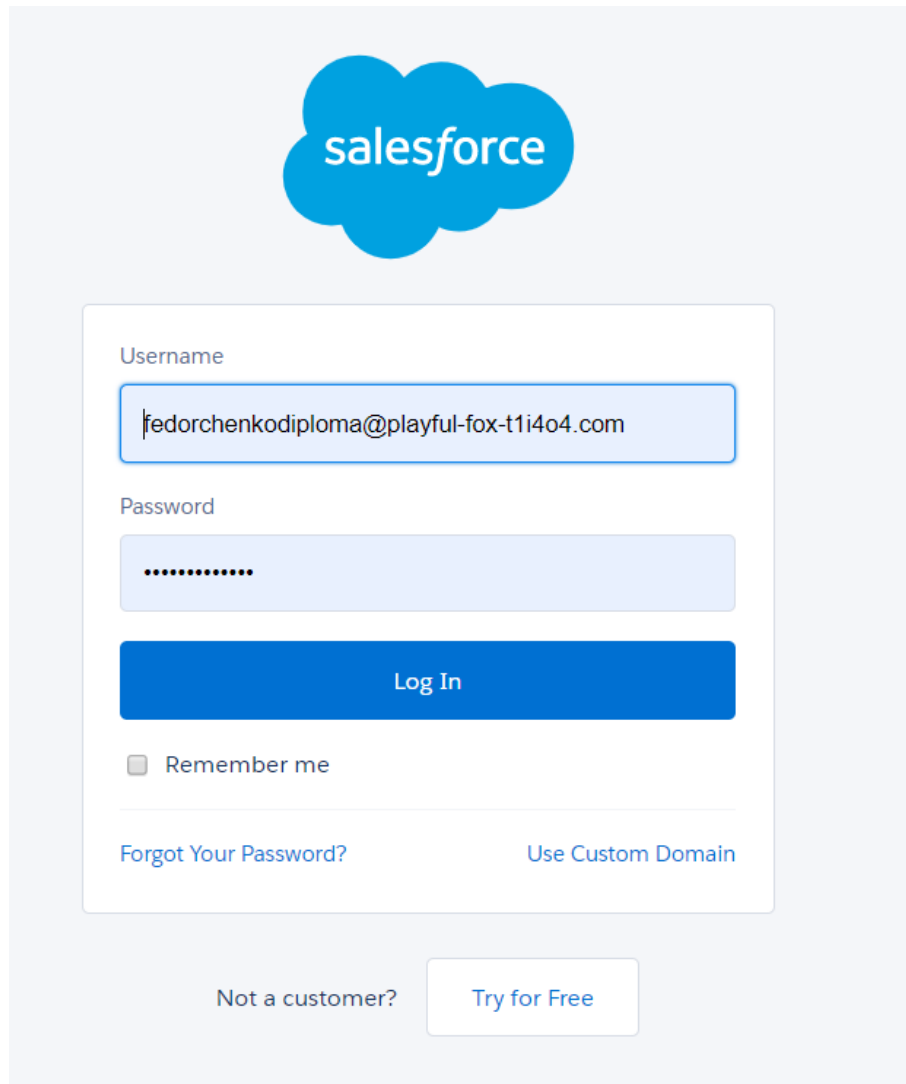


Рисунок 3.30 – Вікно відображення Календарю для Клієнта

### 3.4 Розробка функціоналу входу до системи.

Для того, щоб адміністратору системи або ріелтору зайти до порталу, йому потрібно пройти процес авторизації. Сторінка авторизації зображена на рисунку 3.31:

Кожному користувачу створюється особистий аккаунт, який відображається у списку користувачів Salesforce, який зображено на рисунку:



The image shows the Salesforce login interface. At the top center is the Salesforce logo, a blue cloud with the word 'salesforce' in white. Below the logo is a white login form with a light blue border. The form contains the following elements: a 'Username' label above a text input field containing 'fedorchenkodiploma@playful-fox-t1i4o4.com'; a 'Password' label above a password input field with masked characters; a prominent blue 'Log In' button; a 'Remember me' checkbox; two links: 'Forgot Your Password?' and 'Use Custom Domain'; and at the bottom, a 'Not a customer?' link and a 'Try for Free' button.

Рисунок 3.31 – Вікно входу до системи

У системі є два типу користувачів: адміністратори системи та клієнти. Відповідно до типу користувача кожен має різний інтерфейс обумовлений обмеженням доступу до різних об'єктів. Адміністратор має доступ до усіх об'єктів та записів [22].

### 3.5 Тестування програмної системи

В якості технології тестування для даного проєкту мою було обрано модульне тестування. У ході модульного тестування кожен модуль тестується як на відповідність вимогам, так і на відсутність проблемних ділянок програмного коду, які можуть викликати відмови й збої в роботі системи.

Модульне тестування (Unit testing) - тестування кожної атомарної функціональності програми окремо, в штучно створеному середовищі. Саме потреба в створенні штучної робочого середовища для певного модуля, вимагає від тестувальника знань в автоматизації тестування програмного забезпечення, навичок програмування.

Розглянемо переваги модульного тестування:

Модульне тестування мотивує програмістів писати код максимально оптимізованим, проводити рефакторинг (спрощення коду програми, не зачіпаючи її функціональність), так як за допомогою Юніт-тестування можна легко перевірити працездатність будь-якого компонента [9].

Необхідність відділення реалізації від інтерфейсу (зважаючи на особливості модульного тестування), що дозволяє мінімізувати залежності в системі.

Модульне тестування допомагає краще зрозуміти роль кожного класу на тлі всієї програмної системи.

Також, при «розробці через тестування», яка активно використовується в екстремальному програмуванні, модульне тестування є одним з основних інструментів, що дозволяє розробляти тестові модулі відповідно до вимог до кожного функціонального модулю [11].

Результати проведення запуску тестів приведені на рисунку 3.32:

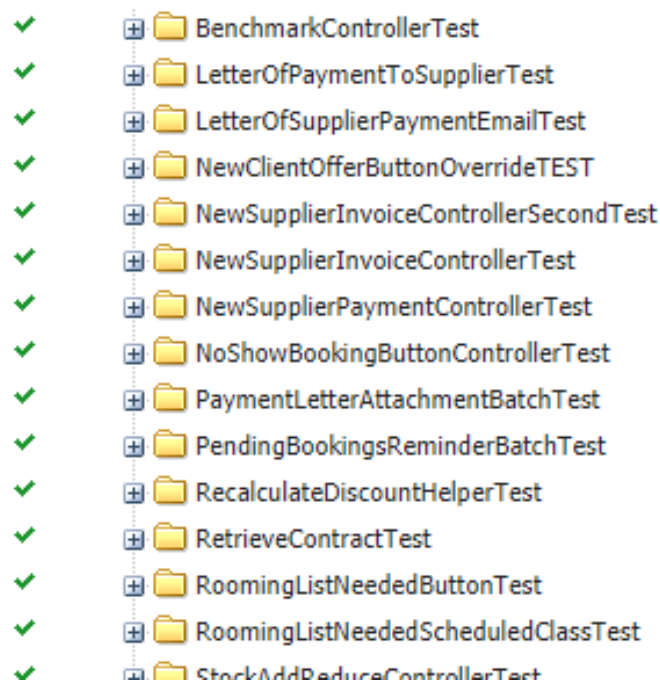


Рисунок 3.32 – Результат запуску Unit-тестів

## ВИСНОВКИ

У ході виконання даної дипломної роботи було розглянуто проблеми автоматизації діяльності агентства з пошуку нерухомості з використанням сучасних засобів обробки інформації. В роботі зроблена якісна постановка задачі, розглянуті методи проектування інформаційних систем. Розроблено алгоритмічне забезпечення для поставленої задачі.

У представленій дипломній була обґрунтована актуальність розробки програмного забезпечення, спроектовано модель користувальницького інтерфейсу. На підставі бізнес правил були розроблені функціональні та нефункціональні вимоги до програмного забезпечення, діаграми варіантів використання програмного забезпечення, модель даних.

Детально описані можливості програмного продукту, обрано цільовий варіант архітектури, розроблена архітектура системи, було описано обрані технічні засоби.

Даний програмний застосунок розроблений для інформаційної та аналітичної підтримки діяльності агенства з пошуку нерухомості. Користувачами цієї системи мають бути клієнти, які шукають житло, та адміністратори системи чи менеджери ріелторських агенцій. Розроблене програмне забезпечення розгорнуте на платформі світового лідера з надання CRM послуг, платформі для розробки Salesforce. Тож описане у даній роботі ПЗ користується системою захисту та безпеки компанії з світовим досвідом та визнанням однойменної компанії Salesforce.

Робота складається з вступу, трьох розділів і висновків. Містить 116 сторінок друкованого тексту, в тому числі 89 сторінок тексту основної частини з 48 рисунками, список з 50 використаних джерел з найменуваннями на 4 сторінках, один додаток на 20 сторінках.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Влиссидес Дж. Применение шаблонов проектирования. Дополнительные штрихи. Москва : Вильямс, 2003. 144с.
2. Дейт К. Дж. Введение в системы баз данных, 8-е издание / пер. с англ. Москва : Вильямс, 2005. 1328 с.
3. Джошуа Кериевски Рефакторинг с использованием шаблонов (паттернов проектирования). Москва : Вильямс, 2006. 400 с.
4. ДСТУ 8302:2015. Бібліографічне посилання. Загальні положення та правила складання. Київ : ДП «УкрНДНЦ», 2016. 29 с.
5. Житловий кодекс України : прийнятий Верховною Радою від 30.07.1983 р. Дата оновлення: 25.07.2018. URL: <https://zakon.gov.ua/laws/main/5464-10> (дата звернення: 23.04.2020)
6. Задачин В. М. Методичні рекомендації до лабораторних робіт для студентів галузі знань 12 "Інформаційні технології" першого (бакалаврського) рівня. Харків : Вид. ХНЕУ, 2019. 217 с.
7. Знахур С. В., Плоха О. Б. Методичні рекомендації до тренінгу з основ управління ІТ-проектами для студентів спеціальності 122 "Комп'ютерні науки та інформаційні технології" першого (бакалаврського) рівня. Харків : Вид. ХНЕУ, 2017. 110 с.
8. Илья Кантор Современный учебник Java Script в 3 книгах. Москва : Миркниг, 2019. 1518с. URL: <https://drive.google.com/open?id=OBw1tpOoY9MYrcmpOb2VoYWJWZ2>
9. Криспин Л., Грегори Дж. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд. Москва : Вильямс, 2010. 464 с.
10. Крэг Ларман Применение UML 2.0 и шаблонов проектирования. Москва : Вильямс, 2006. 36 с.
11. Кулаков К. А., Димитров В. М. Основы тестирования программного обеспечения : навч. посіб. Петрозаводск : Изд-во ПетрГУ, 2018. 57 с.
12. Лабберс П., Олберс Б., Салим Ф. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений. Москва : Вильямс, 2011. 272 с.
13. Левитин А. В. Алгоритмы. Введение в разработку и анализ. Москва : Вильямс, 2006. 257 с.
14. Лосев М. Ю., Тарасов О. В., Федько В. В. Організація баз даних та знань (ADO.NET) : консп. лекц. Харків : Вид. ХНЕУ, 2011. 108 с.



15. Марк Гранд Шаблоны проектирования в JAVA. Каталог популярных шаблонов проектирования, проиллюстрированных при помощи UML. Москва : Новое знание, 2004. 560 с.
16. Маркин А. В. Программирование на SQL : уч. пос. Москва : Юрайт, 2019. 435 с.
17. Мартин Фаулер Шаблоны корпоративных приложений. Москва : Вильямс, 2009. 544 с.
18. Парфьонов Ю. Е., Ушакова І. О. Методичні рекомендації до виконання кваліфікаційних робіт для студентів спеціальностей 121 "Інженерія програмного забезпечення", 122 "Комп'ютерні науки", 126 "Інформаційні системи та технології" першого (бакалаврського) рівня. Харків : Вид. ХНЕУ, 2020. 65 с.
19. Податковий кодекс України : Закон України від 02.12.2010 р. № 2755-VI. Дата оновлення: 20.04.2020 р. URL: <https://zakon.gov.ua/laws/main/2755-17> (дата звернення: 29.04.2020)
20. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Санкт-Петербург : Питер, 2007. 366 с.
21. Роббинс Дж. HTML5, CSS3 и JavaScript: исчерпывающее руководство. Москва : Эксмо, 2014. 528 с.
22. Скорін Ю. І. Якість програмного забезпечення та тестування. Методичні рекомендації до самостійної роботи студентів спеціальності 121 "Інженерія програмного забезпечення" першого (бакалаврського) рівня. Харків : Вид. ХНЕУ, 2019. 47 с.
23. Скотт В. Эмблер, Прамодкумар Дж. Садаладж Рефакторинг баз данных: эволюционное проектирование. Москва : Вильямс, 2007. 368 с.
24. Тарасов О. В., Федько В. В., Лосєв М. Ю. Використання мови SQL для роботи з сучасними системами керування базами даних. Практикум з навчальної дисципліни "Організація баз даних та знань" : навч. посібн. Харків : Вид. ХНЕУ, 2013. 349 с.
25. Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн Алгоритмы: построение и анализ. Москва : Вильямс, 2013. 320 с.
26. Уиттакер Дж., Арбон Дж., Каролло Дж. Как тестирует Google. Санкт-Петербург : Питер, 2014. 320 с.
27. Ушакова І. О. Основи системного аналізу об'єктів та процесів комп'ютеризації : навч. посіб. Ч. 1. Харків : Вид. ХНЕУ, 2007. 212 с.
28. Ушакова І. О. Основи системного аналізу об'єктів та процесів комп'ютеризації : навч. посіб. Ч. 2. Харків : Вид. ХНЕУ, 2008. 324 с.

29. Федорченко Р. В. Використання Salesforce Lightning Component Framework. *Інформаційні технології в сучасному світі: дослідження молодих вчених* : тези доп. міжнар. наук.-практ. конф. (м. Харків, 20-21 лют. 2020 р.). Харків, 2020. С. 29.

30. Федько В. В., Тарасов О. В., Лосев М. Ю. Організація баз даних та знань : навч.-практ. посібн. Харків : Вид. ХНЕУ, 2013. 200 с.

31. Федько В. В., Тарасов О. В., Лосев М. Ю. Сучасні засоби доступу до даних : навчальний посібник для самостійної роботи студентів з навчальної дисципліни «Організація баз даних та знань». Харків : Вид. ХНЕУ, 2014. 328 с.

32. Флэнаган Д. 13.8.1. Чего не может JavaScript. JavaScript. Подробное руководство / пер. з англ. А. Киселев. Санкт-Петербург : Символ-Плюс, 2008. С. 80 – 280, 281.

33. Larry Rockoff The Language of SQL. Cengage Learning ISBN, 2017. 457 p.

34. Thompson P., Paul D., Paul A. Developing Information Systems: Practical guidance for IT professional. London : BCS Learning & Development Limited, 2014. 206 p.

35. Бронювання житла : веб-сайт. URL: <https://www.booking.com> (дата звернення 25.05.2020)

36. Гостинність в соціокультурній сфері туризму як головний предмет філософського аналізу. URL: [http://tourlib.net/statti\\_ukr/garbar.htm](http://tourlib.net/statti_ukr/garbar.htm).

37. Діаграма прецедентів : веб-сайт. URL: [http://fitm.nusta.edu.ua/mediawiki/index.php?title=Діаграма\\_прецедентів](http://fitm.nusta.edu.ua/mediawiki/index.php?title=Діаграма_прецедентів). (дата звернення 13.05.2020)

38. Оренда квартир у Харкові : веб-сайт. URL: <https://real-estate-agency-kharkiv.business.site/> (дата звернення 17.05.2020)

39. Особливості Web застосунків: веб-сайт. URL: <http://sites.znu.edu.ua/webprog/lect/1191.ukr.html> (дата звернення 11.05.2020)

40. Особливості індустрії гостинності. URL: [com/1056112760340/turizm/osoblivosti\\_industriyi\\_gostinnosti#259](http://com/1056112760340/turizm/osoblivosti_industriyi_gostinnosti#259).

41. Сервіс з бронювання житла : веб-сайт. URL: <https://www.airbnb.com> (дата звернення 23.05.2020)

42. Студопедія: Програмне забезпечення інформаційних систем : веб-сайт. URL: <https://studopedia.info/1-112703.html> (дата звернення 11.05.2020)

43. 7 лучших JavaScript фреймворков и библиотек для изучения в 2018. URL: <http://proglib.io/> (дата звернення 28.05.2020)

44. Типи клієнтів у системі клієнт-сервер : веб-сайт. URL: <http://bourabai.kz/dbt/richclient.htm> (дата звернення 11.05.2020)