

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**на тему «Інформаційна система взаємодії з клієнтами агентства
нерухомості»**

за спеціальністю 122 «Комп'ютерні науки та інформаційні технології»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студентка групи ІТ-61 Белка Ярослава Сергіївна

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «_____» _____ 2020 р.

Науковий керівник:

(підпис)

К. Т. Н., доц., Шендрик В.В.
(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії:

(підпис)

Шифрін Д. М.
(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2020

Сумський державний університет
 Факультет електроніки та інформаційних технологій
 Кафедра комп'ютерних наук
 Секція інформаційних технологій проектування
 Спеціальність 122 «Комп'ютерні науки»
 Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
 «__» _____ 2020 р.

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

Белка Ярославі Сергіївні

1 Тема роботи Інформаційна система взаємодії з клієнтами агентства нерухомості

керівник роботи Шендрик Віра Вікторівна, к.т.н., доцент,

затверджені наказом по університету від «14» травня 2020 р. № 0576-III

2 Строк подання студентом роботи «1» червня 2020 р.

3 Вхідні дані до роботи _____

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) Аналіз предметної області

2) Проектування інформаційної системи

3) Програмна реалізація системи

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Актуальність роботи, мета роботи, постановка задачі, аналіз систем-аналогів, порівняння систем аналогів, функціональні вимоги до інформаційної системи, моделювання роботи системи, діаграми варіантів використання, ERD-діаграма, засоби реалізації, архітектура системи, демонстрація системи, тестування системи, висновки, публікація результатів.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____ 01.10.2019 _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Оформлення планування робіт	До 18.03.2020	
2.	Оформлення технічного завдання	До 25.03.2020	
3.	Проведення аналізу предметної області	До 01.04.2020	
4.	Проведення проектування інформаційної системи	До 16.04.2020	
5.	Реалізація системи	До 18.05.2020	
6.	Проведення тестування	До 29.05.2020	
7.	Здача пояснювальної записки	До 01.06.2020	

Студент _____
(підпис)

Белка Я.С.

Керівник роботи _____
(підпис)

к.т.н., доц. Шендрик
В.В.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Інформаційна система взаємодії з клієнтами агентства нерухомості».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 19 найменувань, додатків. Загальний обсяг роботи – 85 сторінок, у тому числі 46 сторінок основного тексту, 2 сторінки списку використаних джерел, 36 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці інформаційної системи підтримки взаємодії співробітників агентства нерухомості з її клієнтами.

У роботі проведено аналіз предметної області та поставлено задачі до розробки інформаційної системи. Перед реалізацією системи було розроблено планування робіт, створено технічне завдання та обрано засоби реалізації.

У ході проектування інформаційної системи було розроблено діаграми варіантів використання та контекстні діаграма з точки зору користувачів. Розробка бази даних була проведена з використанням діаграми сутність-зв'язок та таблиці з описом атрибутів сутностей. Було розроблено два модулі для клієнтів та співробітників агентства, після чого було проведено тестування функціоналу системи для перевірки працездатності системи.

Результатом проведеної роботи є інформаційна система, що надає функціонал для управління інформацією, що використовується у діяльності продажу нерухомості.

Практичне значення роботи полягає у тому, що система дозволяє налагодити канал зв'язку між клієнтами агентства та ріелтора, що є важливою частиною ефективної співпраці.

Ключові слова: інформаційна система, система управління взаємовідносинами з клієнтами, агентство продажу нерухомості, фреймворк.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень.....	8
1.2 Аналіз програмних продуктів-аналогів	9
1.3 Постановка задачі.....	10
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	12
2.1 Структурно-функціональне моделювання інформаційної системи	12
2.2 Проектування моделі бази даних	17
3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ	24
3.1 Архітектура програмного додатку	24
3.2 Програмна реалізація	26
3.3 Використання програмного додатку	38
ВИСНОВКИ	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47
ДОДАТОК А	49
ДОДАТОК Б.....	50
ДОДАТОК В.....	61
ДОДАТОК Г.....	70
ДОДАТОК Д	75

ВСТУП

У наш час дуже швидко розвивається ринок нерухомості. Так, як рівень життя зростає порівняно з минулим, зростає і кількість населення. Люди тяжіють до забезпечення себе основними засобами комфортного життя, одним з яких є житло. У 2018 році по всій Україні було укладено 277230 договорів купівлі-продажу жилої нерухомості, що дорівнює приблизно 759 договорам щоденно [1]. Будь-то купівля земельної ділянки, приватного будинку чи квартири – це завжди відповідальний та трудомісткий процес. Саме тому, деякі люди віддають перевагу ріелторським організаціям для продажу чи купівлі нерухомості. Наявність вичерпної бази і систем підбору найкращого варіанту з представлених, грає вирішальну роль у налагодженні довіри клієнтів до ріелтора та фірми.

Актуальність проекту полягає у створенні каналу зв'язку між співробітниками агентства нерухомості та її клієнтами, що є необхідною складовою ефективної співпраці. Система збереження, обробки та редагування інформації знизить робочий час для пошуку серед сотень об'єктів потрібного, що дозволяє знизити завантаженість ріелторів та підвищити якість підбору пропозицій для продажу.

Об'єктом роботи є інформаційна підтримка процесу діяльності агентства нерухомості.

Предметом дослідження є засоби забезпечення інформаційної підтримки ріелторської діяльності.

Метою роботи є розробка інформаційної системи підтримки діяльності агентства продажу нерухомості.

Для досягнення поставленої мети потрібно вирішити наступні задачі:

- провести аналіз предметної області;
- проаналізувати існуючі системи підтримки продажу нерухомості;
- встановити головні вимоги до інформаційної системи та обрати засоби реалізації;
- розробити технічне завдання та виконати планування робіт;

- розробити систему;
- провести тестування.

Результат дипломного проекту – система інформаційної підтримки для агентства нерухомості. Її практичне значення полягає у можливості використанні в управлінні взаємовідносинами з клієнтами. Система та її окремі модулі можуть бути впроваджені у різні види діяльності.

Результати роботи, впроваджено у діяльність відділу розробки програмного забезпечення компанії ТОВ «СТРУКТУРА ОПТИМА», що засвідчує акт впровадження (додаток А).

Результати роботи були представлені на міжнародній науково-технічній конференції «Інформатика, математика, автоматизація – 2020» (м. Суми) [2].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень

Система управління взаємовідносинами з клієнтами автоматизує та організовує процеси в агентстві. Система використовується для планування та створення розкладів для контролю за передпродажною та після продажною діяльністю в організації. За допомогою інформаційної системи можна керувати процесами продажу, так як вона дозволяє зберігати та опрацьовувати великі масиви даних [3]. Саме тому, постає необхідність у розробці та впровадженні системи інформаційної підтримки.

Інформаційна система взаємодії з клієнтами має два основних компоненти: базу даних, яка дозволяє організації зібрати повний набір інформації про замовників і набір додатків, які дозволяють взаємодіяти з клієнтами, щоб задовольнити їх потреби. Використовуючи ІТ для збору інформації про побажання клієнтів, компанія може створити можливості для забезпечення отримання більших прибутків [4]. Розробка взаємопов'язаних між собою модулів для співробітників та клієнтами агентства нерухомості забезпечить канал зв'язку, який є важливою частиною ефективної співпраці.

Центральна база даних, яка доступна всім працівникам для перегляду та оновлення підвищить рівень швидкодії підприємства. У той же час, якщо надати клієнту веб-доступ до системи з використанням паролю, він зможе самостійно взаємодіяти з системою, що покращить націленість підприємства на клієнтів [5].

Найважливішою метою для підприємства, що працює на клієнтів – є забезпечення високого рівня задоволення її послугами. Система обробки та керування інформацією з автоматизацією пошуку серед сотень об'єктів потрібного знизить робочий час, а отже завантаженість ріелторів та підвищить якість підбору пропозицій для продажу. Саме тому, розробка інформаційної системи забезпечить досягнення мети підприємства.

1.2 Аналіз програмних продуктів-аналогів

Для огляду існуючих програм-конкурентів були взяті системи MegaRieltor [6] та Realsoft [7]. Системи мають зручний інтерфейс, надають функціонал планування та моніторингу задач, зберігають інформацію про об'єкти нерухомості та клієнтів. Порівняльна характеристика систем-аналогів наведена у таблиці 1.1.

Таблиця 1.1 – Аналіз розглянутих систем-аналогів

Критерії	MegaRieltor	Realsoft
1. Автоматизація пошуку помешкання відповідно до бажань клієнта	+	+
2. Створення бази помешкань	+	+
3. Введення бази клієнтів (власників та покупців)	+	+
4. Планування зустрічей та задач у календарі	+	+
5. Система сповіщень про події у календарі	+	+
6. Веб-додаток з доступом з будь-якого пристрою (ПК, планшет, смартфон) з наявністю мережі інтернет	+	+
7. Інтуїтивно зрозумілий інтерфейс	+	+
8. Привласнення задач співробітникам та контроль за виконанням	+	+
9. Доступ клієнта до власного кабінету зі зміною параметрів бажаного об'єкту	-	-
10. Можливість відправлення документів та сторінок з пропозицією об'єкта клієнту	-	-

Аналіз систем MegaRieltor та Realsoft виявив їх сильні сторони та недоліки. Жодна з систем не надає модуль взаємодії клієнта для зміни інформації про бажані характеристики помешкання. Тобто, не організований канал зв'язку клієнта з системою безпосередньо. Щоб клієнту змінити свої побажання до об'єкта, який він має намір придбати, він повинен зв'язатися з ріелтором, а той у свою чергу змінити дані. Такий спосіб зміни через посередництво, займає більше часу, а отже зменшує ефективність роботи. Надання клієнту власних логіну та паролю для входу в особистий кабінет, забезпечить зручний канал для швидкої зміни побажань щодо

об'єкта купівлі і виключить з процесу оновлення даних ріелтора. Співпрацівнику не доведеться після дзвінка клієнта самотійно змінювати інформацію, він зможе отримати повідомлення у той час, коли клієнт зробить це самотійно.

1.3 Постановка задачі

Мета дипломного проектування – це створення інформаційної системи для підтримки інформаційних процесів, які супроводжують процес продажу житлових приміщень. Впровадження комп'ютеризованих технологій у сферу діяльності продажу має на меті полегшити організацію роботи співробітників з клієнтами. Надання клієнтам доступу до системи дозволяє виключити ріелтора агентства з процесу внесення змін до побажання клієнта, тим самим, усунення посередництва зберігає час, який потрібний для забезпечення якісного надання послуг клієнтам.

Технічне завдання на розробку інформаційної системи наводиться у додатку Б. Планування робіт з деталізацією мети методом SMART, структурами робіт та організації, а також аналізом ризиків наведено у додатку В.

Система має на меті підвищення якості надання послуг клієнтам, скоротивши терміни підбору об'єктів серед сотень наявних, надаючи інтерфейс та можливості для додавання, редагування та видалення інформації.

Для досягнення поставленої мети потрібно вирішити наступні задачі:

- провести аналіз предметної області;
- проаналізувати існуючі системи підтримки продажу нерухомості розробивши порівняльну таблицю їх характеристик;
- встановити головні вимоги до інформаційної системи та обрати засоби реалізації;
- розробити технічне завдання та виконати планування робіт;
- провести проектування системи;

- розробити інформаційну систему;
- провести тестування, оформити звіт та виправити несправності.

Для реалізації інформаційної системи було обрано використання розділення даних додатку за принципом model-view-controller (MVC), Ієрархічна структура розробки дозволяє впроваджувати, використовувати та легко модифікувати кожен модуль продукту окремо від інших. У об'єктно-орієнтованому програмуванні MVC принцип використовується розробниками для забезпечення можливості повторного використання коду, що допомагає значно знизити час на реалізацію систем які мають інтерфейс користувача [8]. Тому, для реалізації веб-додатку був обраний фреймворк Yii2. Yii2 – це компонентний PHP-фреймворк, призначений для швидкої розробки сучасних веб-додатків. Завдяки його компонентній структурі і підтримці кешування, фреймворк особливо підходить для розробки таких великих проектів як портали, форуми, системи керування вмістом (CMS), інтернет-магазинів або RESTful-додатків [9].

Для реалізації системи також були обрані такі інструменти, як: PHP, HTML, CSS, JavaScript та Bootstrap. Bootstrap використовується для створення адаптивного веб-дизайну. Це підхід, який намагається адаптувати дизайн до відображення на будь-якому пристрої з можливою зміною вигляду, але з однаковою функціональністю [10].

Для реалізації бази даних обрано MySQL – систему керування реляційними базами даних від компанії Oracle [11]. Для адміністрування був обраний додаток phpMyAdmin.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Структурно-функціональне моделювання інформаційної системи

Головною метою проектування є забезпечення ефективного функціонування інформаційної системи у відповідності до поставлених вимог та при наявності заданих обмежень та технологій [12]. Для моделювання системи було використано методологію функціонального моделювання IDEF0, що дозволяє відобразити систему у вигляді взаємозалежних блоків [13]. Контекстна діаграма функціонування інформаційної системи у нотації IDEF0, виконана з точки зору співробітника агентства, наведена на рисунку 2.1.

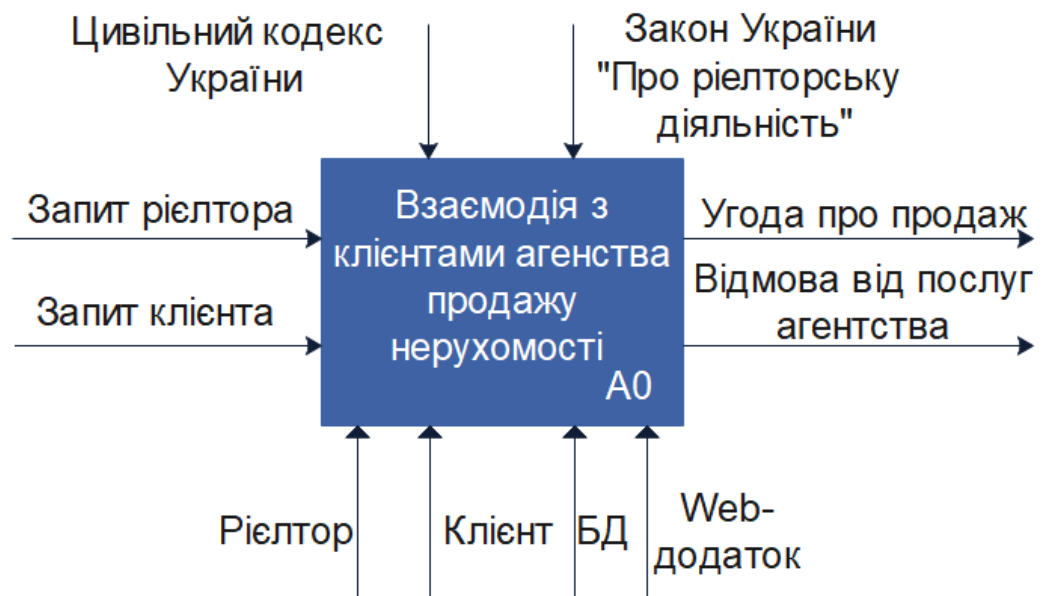


Рисунок 2.1 – Контекстна діаграма з точки зору співробітника

Головними складовими бізнес-процесу розроблюваної системи є:

- процес: взаємодія з клієнтами агенства продажу нерухомості;
- вхідні дані: запит ріелтора;
- вихідні потоки: угода про продаж або відмова від співпраці з агенством;

- нормативи управління: Цивільний кодекс України;
- ресурси: ріелтор, база даних, веб-додаток.

На рисунку 2.2 представлена декомпозиція контекстної діаграми A0 у нотації IDEF0. Робота починається з внесення необхідних даних про клієнта, продавця, об'єкт. Та продовжується взаємодією ріелтора з клієнтом через надсилання знайдених пропозицій.

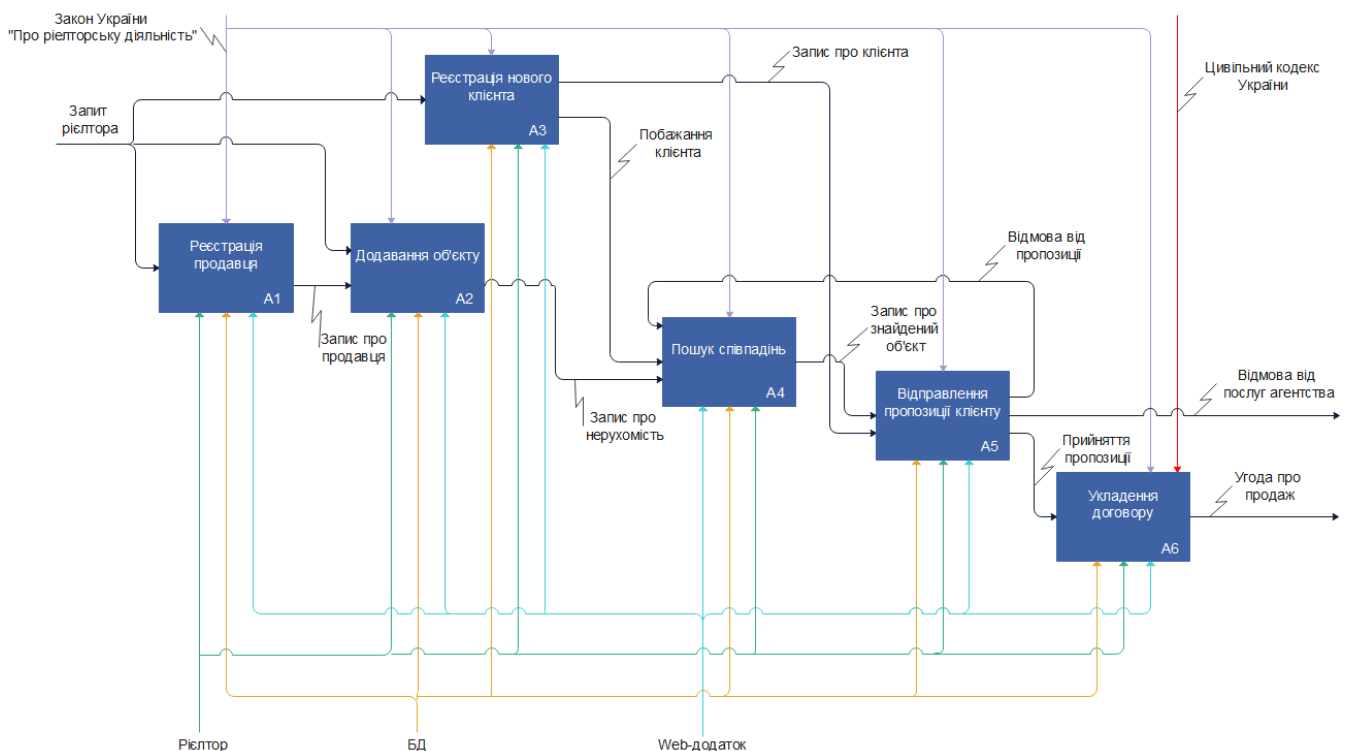


Рисунок 2.2 – Діаграма декомпозиції контекстної діаграми A0

Діаграма декомпозиції контекстної діаграми A0 містить 6 блоків. Вхідними даними до блоків «Реєстрація продавця», «Додавання об'єкту» та «Реєстрація нового клієнта» є «Запит ріелтора». Також на вхід «Додавання об'єкту» подаються вихідні дані отримані після реєстрації продавця. Запис про додану нерухомість та про побажання клієнта є вхідними даними до «Пошук співпадінь». Після відправки пропозиції клієнту, у разі відмови пошук повторюється. Процеси пошуку та надсилання пропозиції ітеративні, тому з'єднані стрілкою, що повертається з виходу блоку «Відправлення пропозиції клієнтові» на вхід блоку «Пошук співпадінь». Вихідні дані після виконання блоку «Відправлення пропозиції клієнтові» можуть

подаватися на вхід до «Укладення договору» або завершувати процес потоку даних у випадку відмови клієнта від подальшої співпраці з агентством. Вихідними даними після укладання договору є «Угода про продаж».

Для виділення основних можливостей системи також було створено діаграми варіантів використання для трьох акторів: ріелтора, менеджера та клієнта. Діаграми варіантів використання потрібні для відображення дій, які зовнішній користувач (актор) може виконувати у системі [14]. Діаграма варіантів використання для актора ріелтора наведена на рисунку 2.3.

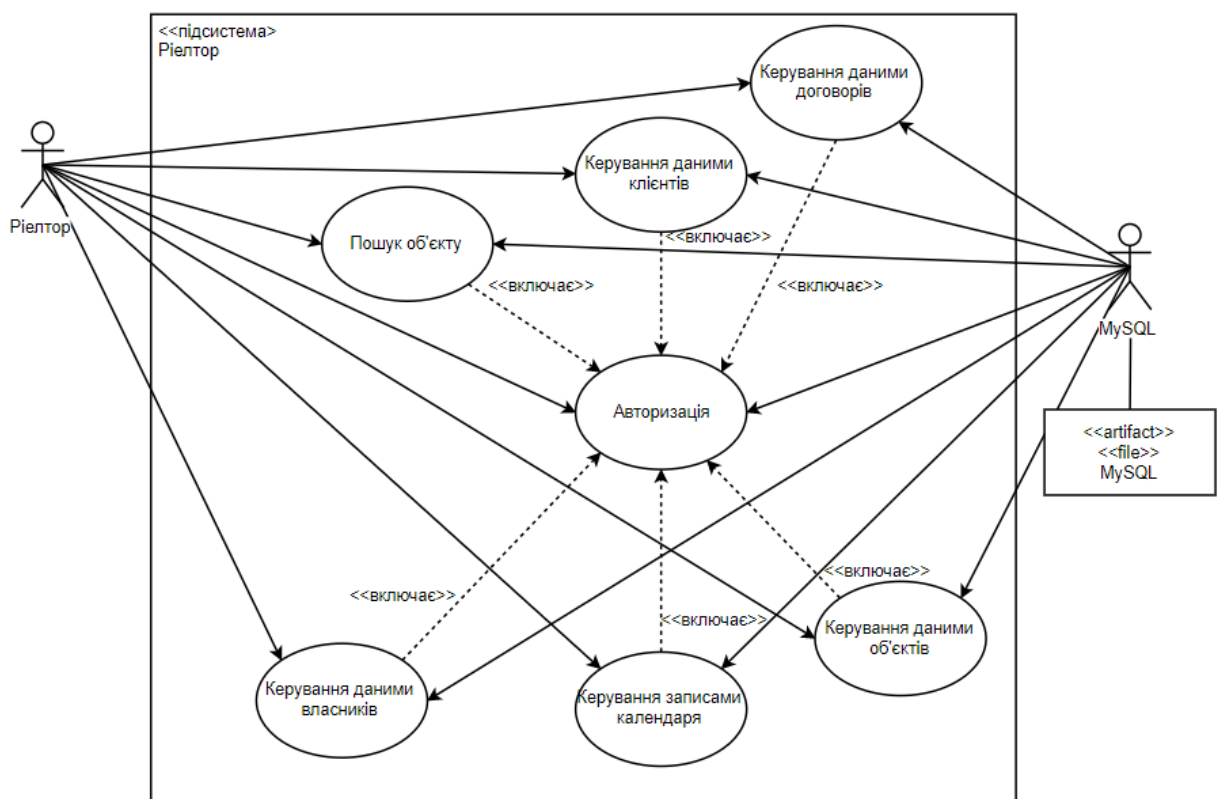


Рисунок 2.3 – Діаграма варіантів використання для актора ріелтора

На діаграмі виділено два актора: ріелтор та база даних mySQL. Варіанти використання для актора ріелтора на діаграмі:

- ВВ авторизація – використовуючи унікальний логін та пароль актор отримує доступ до системи;

- ВВ керування даними власників – дозволяє актору створювати, редагувати, переглядати та видаляти дані (create-read-update-delete (CRUD-функції)) про власників нерухомості;
- ВВ керування даними клієнтів – надає CRUD-функції для роботи з інформацією про клієнтів;
- ВВ керування даними об'єктів – надає CRUD-функції для роботи з інформацією про будинки та квартири;
- ВВ керування даними договорів – надає CRUD-функції для роботи з договорами;
- ВВ керування записами календаря – надає CRUD-функції для управління задачами та зустрічами у календарі;
- ВВ пошук об'єкта – дозволяє актору виконувати пошук пропозицій за побажанням клієнта.

Діаграма варіантів використання для актора ріелтора наведена на рисунку 2.4.

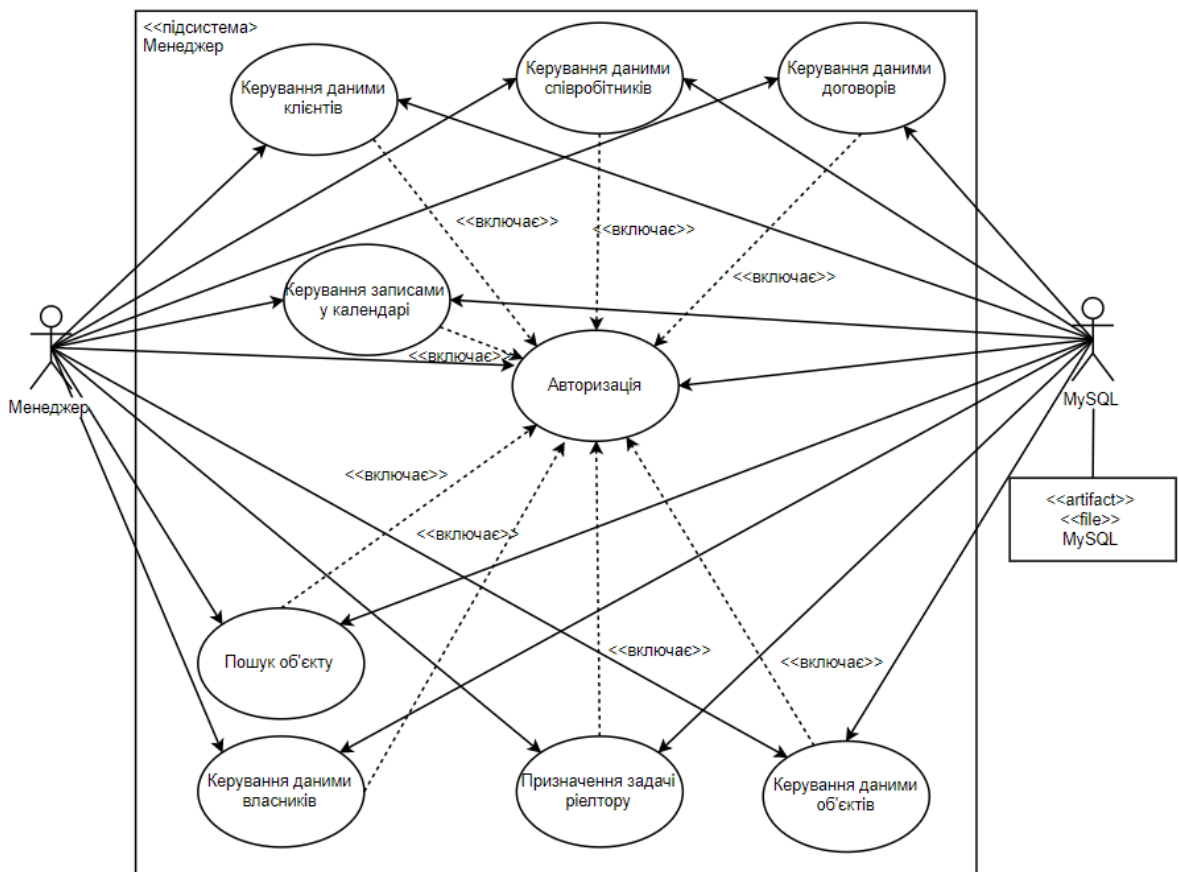


Рисунок 2.4 – Діаграма варіантів використання для актора менеджера

Діаграма має два актора: менеджер та база даних `mysql`. Повноваження менеджера розширюють можливості релтора двома блоками варіантів використання:

- ВВ керування даними співробітників – надає CRUD-функції для роботи з інформацією про співробітників;
- ВВ призначення задачі релтору – дозволяє менеджеру створювати та призначати задачі релтору.

Діаграма варіантів використання для актора клієнта наведена на рисунку 2.5.

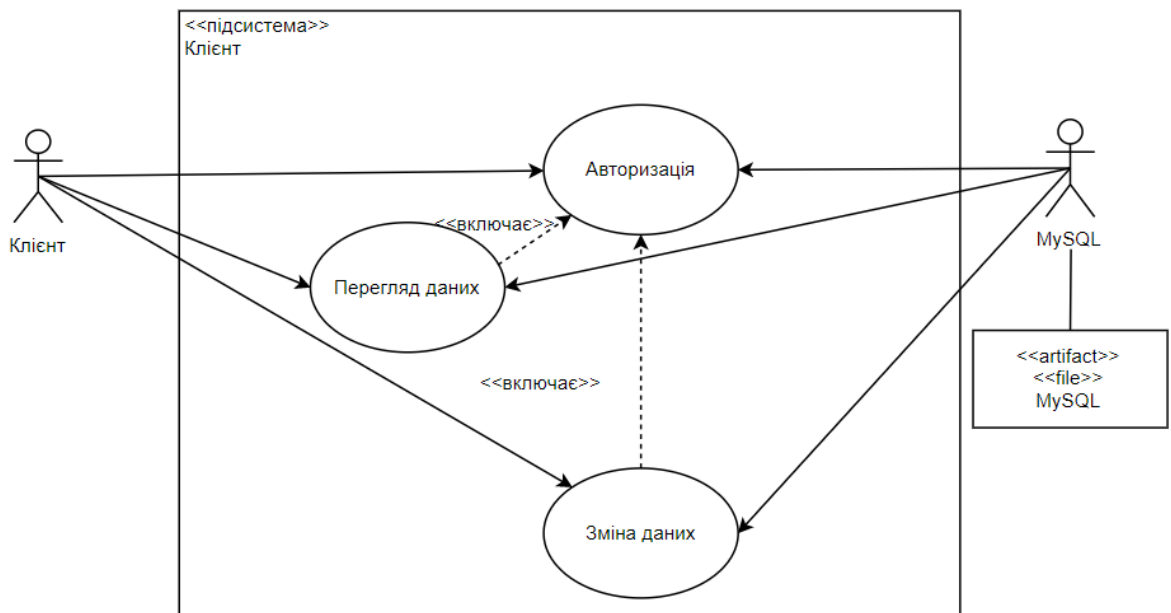


Рисунок 2.5 – Діаграма варіантів використання для актора клієнта

Головними акторами на діаграмі виступають: клієнт та база даних `mysql`. Варіанти використання наступні:

- ВВ авторизація – використовуючи унікальний логін та пароль клієнт отримує доступ до системи;
- ВВ перегляд даних – актор має можливість перегляд власних даних, інформацію про релтора, надіслані пропозиції та призначені зустрічі;
- ВВ зміна даних – актор може змінювати дані побажання та статус надісланих зустрічей та пропозицій.

2.2 Проектування моделі бази даних

Для проектування бази даних було використано модель сутність-зв'язок (entity relationship diagram (ER)), яка допомагає візуально представити відносини між сутностями, що зберігаються у базі даних. Виділення сутностей, їх атрибутів та показ взаємовідносин між ними ілюструє логічну структуру бази [15]. ER-діаграма бази даних показана на рисунку 2.6.

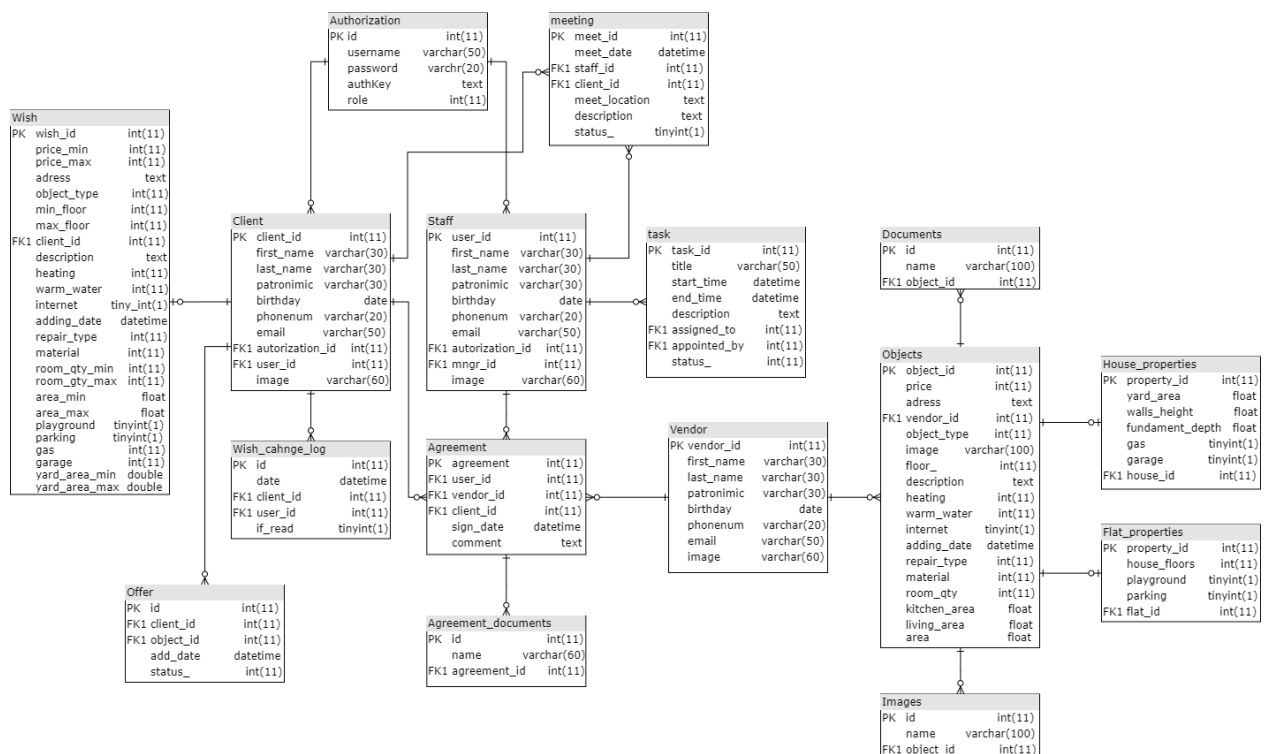


Рисунок 2.6 – ER-діаграма бази даних

На вищенаведеній діаграмі зображені зв'язки між виділеними сутностями. Так між таблицями клієнта (client) та побажання (wish) встановлений зв'язок один – нуль або один, так як клієнт у системі може мати, а може і не мати побажання. Такий же зв'язок пов'язує таблиці об'єкта (object) та характеристик будинку (house_properties), та об'єкта (object) та характеристик квартири (flat_properties). Усі інші таблиці з'єднані зв'язком один і тільки один – нуль або багато.

Отже, для реалізації функціоналу системи було розроблено базу даних яка має шістнадцять сутностей:

- інформація про співробітників агентства - таблиця staff;
- інформація про клієнтів-покупців - таблиця client;
- інформація про власників нерухомості - таблиця vendor;
- інформація про об'єкти продажу - таблиця objects;
- дані користувачів для входу до системи - таблиця authorization;
- параметри будинків - таблиця house_properties;
- параметри квартир - таблиця flat_properties;
- документи об'єктів нерухомості - таблиця documents;
- фотографії об'єктів нерухомості - таблиця images;
- записи про задачі - таблиця task;
- записи про зустрічі - таблиця meeting;
- інформація про побажання клієнта - таблиця wish;
- історія змін побажання клієнтом - таблиця wish_change_log;
- інформація про надіслані клієнту пропозиції - таблиця offer;
- записи про укладені договори - таблиця agreement;
- документи, прив'язані до договору - таблиця agreement_documents.

Кожна з вищеописаних сутностей складається з атрибутів, опис яких наведено у таблиці 2.1.

Таблиця 2.1 – Опис таблиць бази даних

	Поле	Зміст	Тип	Ключі	Обмеження
staff	user_id	Ідентифікатор	int(11)	РК	Не пустий, унікальний
	first_name	Ім'я	varchar(30)		Не пустий
	last_name	Прізвище	varchar(30)		Не пустий
	patronymic	По-батькові	varchar(30)		Не пустий
	birthday	Дата народження	date		Не пустий
	email	Імейл	varchar(50)		Не пустий
	phonenumber	Телефон	varchar(50)		Не пустий

Продовження таблиці 2.1 – Опис таблиць бази даних

	authorization_id	Ідентифікатор таблиці авторизації ^у	int (11)	FK	Не пустий
	mngr_id	Ідентифікатор менеджера	int(11)	FK	
	image	Назва фото	varchar(60)		
client	client_id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	first_name	Ім'я	varchar(30)		Не пустий
	last_name	Прізвище	varchar(30)		Не пустий
	patronimic	По-батькові	varchar(30)		Не пустий
	birthday	Дата народження	date		Не пустий
	email	Імейл	varchar(50)		Не пустий
	phonenumber	Телефон	varchar(50)		Не пустий
	authorization_id	Ідентифікатор	int (11)	FK	Не пустий
	user_id	Ідентифікатор ріелтора	int(11)	FK	Не пустий
image	Назва фото	varchar(60)			
vendor	vendor_id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	first_name	Ім'я	varchar(30)		Не пустий
	last_name	Прізвище	varchar(30)		Не пустий
	patronimic	По-батькові	varchar(30)		Не пустий
	birthday	Дата народження	date		Не пустий
	email	Імейл	varchar(50)		Не пустий
	phonenumber	Телефон	varchar(50)		Не пустий
	image	Назва фото	varchar(60)		
objects	object_id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	price	Ціна	int(11)		Не пустий
	address	Адреса	text		Не пустий
	vendor_id	Ідентифікатор власника	int(11)	FK	Не пустий
	object_type	Тип об'єкту	int(11)		Не пустий
	image	Назва головного зображення	varchar(100)		Не пустий
	floor_	Поверх на якому знаходиться квартира, чи к-ть поверхів будинку	int(11)		Не пустий
	description	Додатковий опис	text		Не пустий
	heating	Тип опалення	int(11)		Не пустий

Продовження таблиці 2.1 – Опис таблиць бази даних

	warm_water	Тип постачання гарячої води	int(11)		Не пустий
	internet	Наявність інтернету	bool		Не пустий
	adding_date	Дата додавання	datetime		Не пустий
	repair_type	Тип ремонту	int(11)		Не пустий
	material	Матеріал стін	int(11)		Не пустий
	room_qty	К-ть кімнат	int(11)		Не пустий
	kitchen_area	Площа кухні	float		Не пустий
	living_area	Жила площа	float		Не пустий
	area	Загальна площа	float		Не пустий
authorization	id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	username	Логін	varchar(50)		Не пустий
	password	Пароль	varchar(50)		Не пустий
	authKey	Cookies	text		
	role	Роль користувача	int(11)		Не пустий
house_properties	property_id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	yard_area	Площа ділянки	float		Не пустий
	walls_height	Висота стін	float		Не пустий
	fundament_dept h	Глибина фундаменту	float		Не пустий
	gas	Наявність газу	bool		Не пустий
	garage	Наявність гаражу	bool		Не пустий
	house_id	Ідентифікатор об'єкту	int(11)	FK	Не пустий
flat_properties	property_id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	house_floors	К-ть поверхів будинку	int(11)		Не пустий
	playground	Наявність дитячого майданчика	bool		Не пустий
	parking	Наявність парковки	bool		Не пустий
	flat_id	Ідентифікатор об'єкту	int(11)	FK	Не пустий
documents	id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	name	Назва документа	varchar(100)		Не пустий
	object_id	Ідентифікатор об'єкту	int(11)	FK	Не пустий

Продовження таблиці 2.1 – Опис таблиць бази даних

images	id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	name	Назва фото	varchar(100)		Не пустий
	object_id	Ідентифікатор об'єкту	int(11)		Не пустий
task	task_id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	title	Заголовок	varchar(50)		Не пустий
	start_time	Час та дата початку	datetime		Не пустий
	end_time	Час та дата закінчення	datetime		Не пустий
	description	Опис	text		Не пустий
	assigned_to	Кому призначена	int(11)	FK	Не пустий
	appointed_by	Ким призначена	int(11)	FK	
status_	Статус виконання задачі	int(11)		Не пустий	
meeting	meet_id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	meet_date	Дата та час зустрічі	datetime		Не пустий
	staff_id	Ідентифікатор ріелтора	int(11)	FK	Не пустий
	client_id	Ідентифікатор клієнта	int(11)	FK	Не пустий
	meet_location	Місце зустрічі	text		Не пустий
	description	Опис	text		Не пустий
	status_	Статус	tinyint(1)		Не пустий
wish	wish_id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	price_min	Мінімальна ціна	int(11)		Не пустий
	price_max	Максимальна ціна	int(11)		Не пустий
	address	Адреса	text		
	object_type	Тип	int(11)		Не пустий
	min_floor	Мін. поверх	int(11)		
	max_floor	Макс. поверх	int(11)		
	client_id	Ідентифікатор клієнта	int(11)	FK	Не пустий
	description	Опис	text		Не пустий
	heating	Тип опалення	int(11)		Не пустий
warm_water	Тип постачання гарячої води	int(11)		Не пустий	

Продовження таблиці 2.1 – Опис таблиць бази даних

	internet	Інтернет	bool		Не пустий
	adding_date	Дата додавання	datetime		Не пустий
	repair_type	Тип ремонту	int(11)		Не пустий
	material	Матеріал стін	int(11)		Не пустий
	room_qty_min	Мінімальна. кількість кімнат	int(11)		Не пустий
	room_qty_max	Максимальна кількість кімнат	int(11)		Не пустий
	area_min	Мін. площа	float		Не пустий
	area_max	Макс. площа	float		Не пустий
	playground	Дитячий майданчик	bool		
	parking	Парковка	bool		
	gas	Газ	int(11)		
	garage	Гараж	bool		
	yard_area_min	Мінімальна площа ділянки	float		
	yard_area_max	Максимальна площа ділянки	float		
wish_change_log	id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	date	Дата редагування	datetime		Не пустий
	client_id	Ідентифікатор клієнта	int(11)	FK	Не пустий
	user_id	Ідентифікатор рієлтора	int(11)	FK	Не пустий
	if_read	Статус	bool		Не пустий
offer	id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	client_id	Ідентифікатор клієнта	int(11)	FK	Не пустий
	object_id	Ідентифікатор об'єкта	int(11)		Не пустий
	add_date	Дата надсилання	datetime		Не пустий
	status_	Прийнята чи скасована пропозиція	int(11)		Не пустий
agreement	agreement	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	user_id	Ідентифікатор рієлтора	int(11)	FK	Не пустий
	vendor_id	Ідентифікатор продавця	int(11)	FK	Не пустий

Продовження таблиці 2.1 – Опис таблиць бази даних

	client_id	Ідентифікатор покупця	int(11)	FK	Не пустий
	sign_date	Дата підписання	datetime		Не пустий
	comment	Коментар	text		
agreement_ documents	id	Ідентифікатор	int(11)	PK	Не пустий, унікальний
	name	Назва договору	varchar(60)		Не пустий
	agreement_id	Ідентифікатор договору	int(11)	FK	Не пустий

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Архітектура програмного додатку

Реалізація системи була проведена з використанням об'єктно-орієнтованого фреймворку PHP - Yii2, який реалізує парадигму model-view-controller (MVC). MVC модель дизайну додатків, що включає в себе три взаємопов'язані частини: модель (інформація), вид (інтерфейс користувача) та контролер (логіка обробки вхідної інформації) [16]. Тим самим, структура додатку виконаного на фреймворку Yii2 має наступний вигляд, поданий на рисунку 3.1.

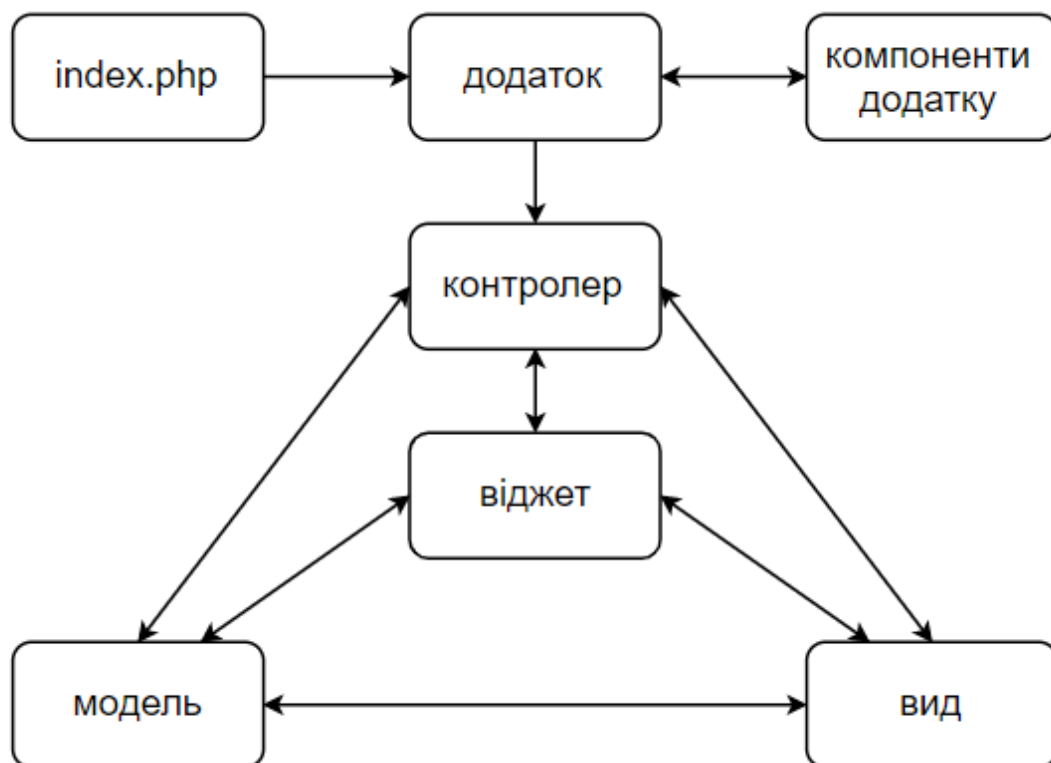


Рисунок 3.1 – Структура додатку Yii2

Моделі представляють дані додатку, тому для кожної таблиці з бази даних створено окрему модель. Види використовуються для відображення інформації, кожна сторінка системи має свій власний вид. Контролер оновлює і види, і моделі.

Контролер приймає вхідні дані та виконує їх обробку у відповідності до дій, описаних у ньому. Також Yii2 надає доступ до використання віджетів, багаторазових блоків коду для створення складних елементів інтерфейсу у об'єктно-орієнтованій формі [17]. До інших компонентів додатку відносяться шаблони сторінок, файли конфігурації, CSS файли та ін. Архітектура інформаційної системи наведена на рисунку 3.2.

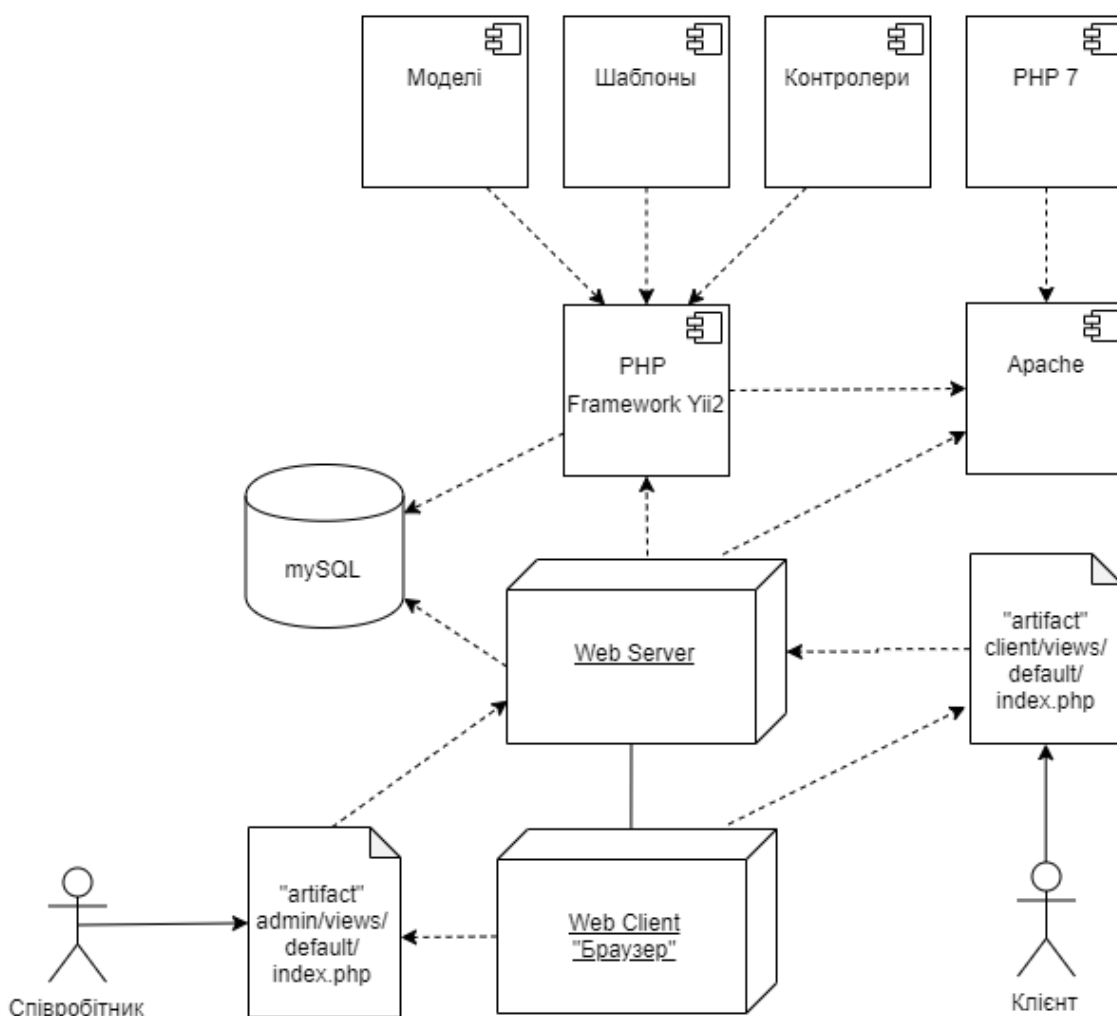


Рисунок 3.2 – Архітектура інформаційної системи

Система має два модулі для співробітників (admin) та клієнтів (client), тому у системі два index.php файли, до яких користувачі отримують доступ після авторизації у системі в залежності від ролі. Отримуючи запит від користувачів, сервер опрацьовує його, з'єднуючись з базою даних.

3.2 Програмна реалізація

3.2.1 Розміщення сайту та реалізація бази даних

Розробка інформаційної системи була розпочата з вибору хостинг провайдера. Був обраний провайдер hostpro, який надає послуги резервного копіювання даних, захисту від DDoS-атак, антивірус та зручну панель управління [18]. Сайт розміщений на Linux-хостингу з доменним ім'ям i.smartbuild.ml.

Після проектування бази даних за допомогою ER-діаграми та опису атрибутів сутностей було написано скрипти для створення таблиць. Приклад скрипта наведений на рисунку 3.3, повний код створення бази даних наведений у додатку Г. Для адміністрування бази даних використовується phpMyAdmin.

```
97 -- Структура таблицы `client`
98 --
99
100 CREATE TABLE `client` (
101   `client_id` int(11) NOT NULL,
102   `first_name` varchar(30) NOT NULL,
103   `last_name` varchar(30) NOT NULL,
104   `patronimic` varchar(30) NOT NULL,
105   `birthday` date NOT NULL,
106   `phonenum` varchar(20) NOT NULL,
107   `email` varchar(50) NOT NULL,
108   `authorization_id` int(11) DEFAULT NULL,
109   `user_id` int(11) DEFAULT NULL,
110   `image` varchar(60) DEFAULT NULL
111 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
112
113 --
114 -- Структура таблицы `documents`
115 --
116
117 CREATE TABLE `documents` (
118   `id` int(11) NOT NULL,
119   `name` varchar(100) NOT NULL,
120   `object_id` int(11) NOT NULL
121 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Рисунок 3.3 – Фрагмент коду створення бази даних

Доступ до бази даних було отримано прописавши у файлі db.php код підключення вказавши клас 'yii\db\Connection', ім'я бази, логін та пароль.

3.2.2 Розробка системи авторизації

Авторизація до системи описана у класі «User». Клас має ряд функцій для перевірки валідності введених даних, пошуку запису про користувача за логіном та генерації ключа авторизації cookies для реалізації можливості «Запам'ятати мене» на сайті. Фрагмент коду класу «User» представлений на рисунку 3.4.

```
public function getAuthKey()
{
    return $this->authKey;
}

public function validateAuthKey($authKey)
{
    return $this->authKey === $authKey;
}

public function validatePassword($password)
{
    return \Yii::$app->security->validatepassword($password, $this->password);
}

public function generateAuthKey(){
    $this->authKey = \Yii::$app->security->generateRandomString();
}
```

Рисунок 3.4 – Фрагмент коду класу авторизації

Форма авторизації виконана за допомогою yii\widgets\ActiveForm, її програмний код наведено на рисунку 3.5. Клас ActiveForm можна застосовувати, коли форма заснована на моделі. Модель містить правила за якими перевіряється правильність введених даних, а також перелік атрибутів з визначеними підписами для форми. У контролері створюється об'єкт model класу «User» і передається до виду для подальшого використання.

```

<?php $form = ActiveForm::begin([
    'id' => 'login-form',
    'layout' => 'horizontal',
    'fieldConfig' => [
        'template' => "{label}\n<div class=\"col-lg-3\">{input}</div>\n<div class=\"col-lg-8\">{error}</div>",
        'labelOptions' => ['class' => 'col-lg-1 control-label'],
    ],
]); ?>

<?= $form->field($model, 'username')->textInput(['autofocus' => true]) ?>
<?= $form->field($model, 'password')->passwordInput() ?>
<?= $form->field($model, 'rememberMe')->checkbox([
    'template' => "<div class=\"col-lg-offset-1 col-lg-3\">{input} {label}</div>\n<div class=\"col-lg-8\">{error}</div>",
]) ?>
<div class="form-group">
    <div class="col-lg-offset-1 col-lg-11">
        <?= Html::submitButton('Login', ['class' => 'btn btn-primary', 'name' => 'login-button']) ?>
    </div>
</div>
<?php ActiveForm::end(); ?>

```

Рисунок 3.5 – Форма авторизації

3.2.3 Розробка модуля для співробітників

Структура файлової системи для реалізації модуля співробітників наведена на рисунку 3.6. Модуль admin має папки для зберігання контролерів (controllers), видів (views/default) та шаблонів (views/layouts). Вони містять один файл шаблону, декілька контролерів та дев'ятнадцять видів, які включають сторінки для перегляду, редагування та додавання даних. Реалізація виконувалася з використанням текстового редактору Brackets, який має багато зручних плагінів для полегшення розробки [19].

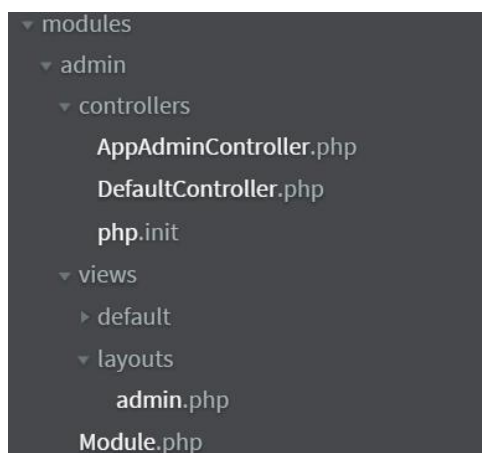


Рисунок 3.6 – Файлова система модуля співробітників

Для роботи з файлами на віддаленому сервері був обраний FTP-клієнт WinSCP (рис. 3.7). Папка моделей містить 28 файлів, які описують правила роботи з таблицями бази даних; класи, що описують завантаження файлів; моделі для пошуку у таблицях GridView.

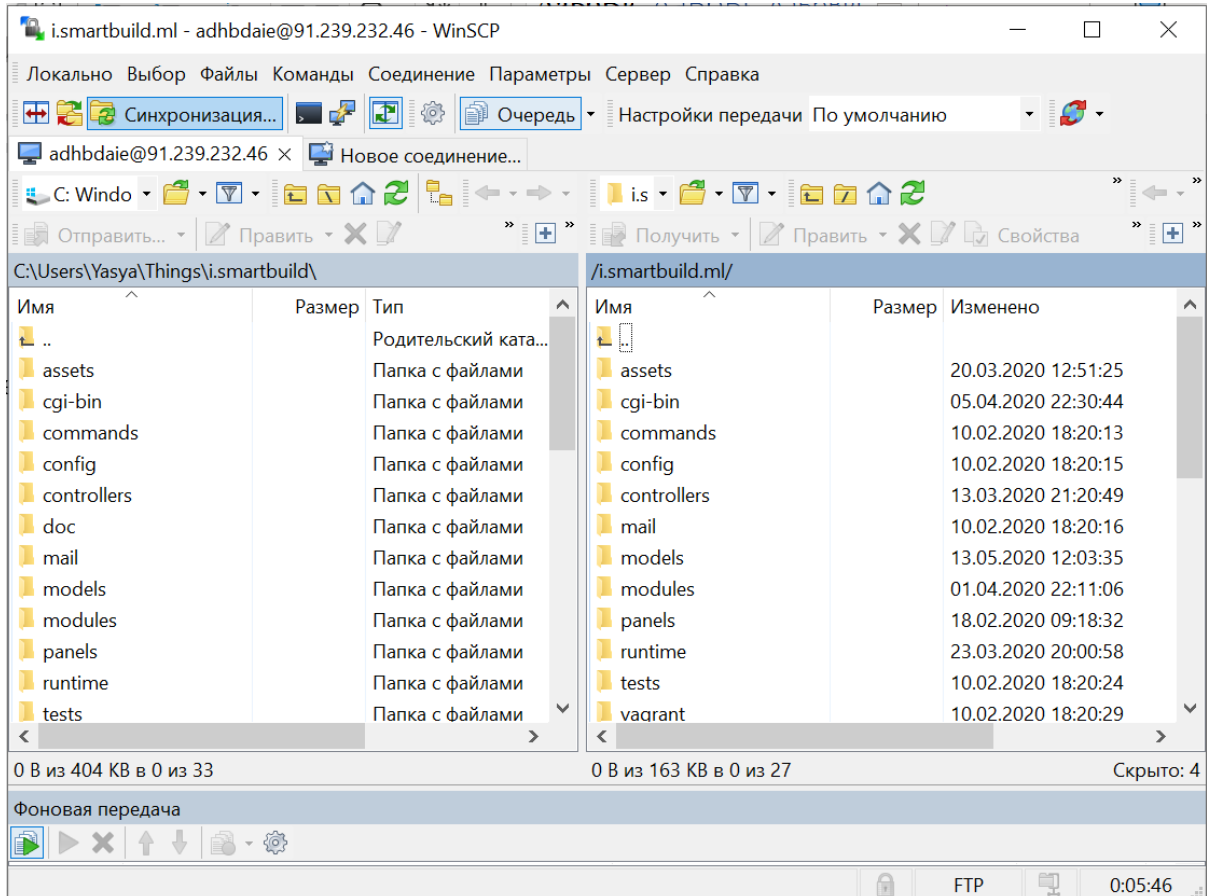


Рисунок 3.7 – Підключення через FTP-клієнт WinSCP

Шаблон сторінок був описаний у файлі `admin.php`. Усі сторінки зберігаються у папці `views/default`. Після створення сторінки потрібно описати дію (action) для її обробки у `DefaultController.php`. Контролер відповідає на запит, оброблює інформацію з моделі та повертає результат для відображення у виді. На рисунку 3.8 зображено приклад дії контролеру з обробки POST запитів до сторінки клієнтів.

```

public function actionClient()
{
    $searchModel = new ClientSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->get());

    if(Yii::$app->request->isPost && isset($_POST['del_client'])){
        $delete_client = $_POST['del_client'];

        $deleteClient = User::findOne($delete_client);
        $deleteClient->delete();

        $user = Client::find()->where(['authorization_id' =>
        $delete_client])->one();
        $user->authorization_id = -1;

        $deleteWish = Wish::find()->where(['client_id' => $user->
        client_id])->one();
        if(!is_null($deleteWish))
            $deleteWish->delete();
        $user->save();
        return $this->refresh();
    }
    return $this->render('client', compact('dataProvider', 'searchModel'));
}

```

Рисунок 3.8 – Опис дії у контролері для сторінки клієнтів

Функція контролеру `actionClient()` опрацьовує лише запит на видалення запису про клієнта. Для двох інших опцій «Змінити» та «Пошук» створені окремі сторінки, а також відповідні функції контролеру «`updateclient.php`» та «`search.php`».

Сторінки «Співробітники», «Клієнти» та «Власники» створені подібним способом з використанням віджету `GridView` для відображення контенту у таблиці. На прикладі реалізації сторінки «Клієнти» буде наведена логіка роботи з віджетом. Для забезпечення можливості реалізації пошуку за полями таблиці у папці `models` створюється окремий клас `ClientSearch`, який наслідує клас `Client` створений для роботи з таблицею бази даних, яка зберігає дані про клієнтів. Фрагмент коду класу `ClientSearch` наведений на рисунку 3.9.

```

public function search($params)
{
    $query = Client::find();
    $dataProvider = new ActiveDataProvider([
        'query' => $query,
    ]);
    if (!($this->load($params) && $this->validate())) {
        return $dataProvider;
    }
    $query->andWhere(['like', 'first_name', $this->first_name])
        ->andWhere(['like', 'last_name', $this->last_name])
        ->andWhere(['like', 'patronimic', $this->patronimic]);
    $staff_id = Staff::find()->where(['authorization_id' => Yii::$app->user-
    >identity->id])->one();

    if($this->user_id === self::STATUS_MY )
        $query->andWhere(['=', 'user_id', $staff_id->user_id]);
    return $dataProvider;
}

const STATUS_MY = 'Мої клієнти';
const STATUS_ALL = 'Усі клієнти';

public function clientOptions(){
    return [
        self::STATUS_ALL => 'Усі клієнти',
        self::STATUS_MY => 'Мої клієнти'
    ];
}

```

Рисунок 3.9 – Реалізація класу для роботи з GridView

У класі розроблений пошук за параметрами повного ім'я клієнта, а також описано логіку роботи спадного списку для вибору у таблиці відображення власних чи усіх клієнтів. Функція `clientOptions()` передається як параметр у спадний список з властивістю «filter». Результати пошуку записуються у об'єкт класу `ActiveDataProvider`, він повертає результати запиту до бази у масиві екземплярів класу `Active Record`. Саме `Active Record` надає об'єктно-орієнтований інтерфейс для доступу і керування даними, що зберігаються у базі. Функція повертає об'єкт класу `ActiveDataProvider` у контролер, який передає результат до виду у відповідь на запит. Фрагмент коду реалізації віджету `GridView` на сторінці «Клієнти» зображено на рисунку 3.10.

```

<?= GridView::widget([
    'dataProvider' => $dataProvider,
    'filterModel' => $searchModel,
    'columns' => [
        ['class' => 'yii\grid\SerialColumn'],
        'last_name',
        'first_name',
        'patronimic',
        [
            'label' => 'Рієлтор',
            'value' => function($data){
                return Staff::getName($data->user_id);
            },
        ],
        [
            'filter' => Html::activeDropDownList ( $searchModel , 'user_id' ,
            $searchModel->clientOptions() ) ,
            'attribute' => 'phonenum',
            'label' => Yii::t ( 'app' , 'Номер телефону' ) ,
        ],
        [
            'class' => 'yii\grid\ActionColumn',
            'template' => '{updateclient} {find} {client}',
            'visibleButtons' => [
                'updateclient' => true,
                'find' => function ($model) {
                    if($model->authorization_id===-1) return false;
                    else return true;
                },
                'client' => function ($model) {
                    if($model->authorization_id===-1) return false;
                    else return true;
                }
            ],
            'buttons' => [

```

Рисунок 3.10 – Використання GridView на сторінці клієнтів

До віджету з контролеру передаються об'єкти моделі постачання та фільтрування даних. У ньому описані атрибути для відображення та кнопки для управління інформацією. На сторінці клієнтів є три кнопки маніпулювання даними запису: видалення, перегляду з можливістю зміни, а також кнопка пошуку за інформацією про побажання клієнта. Отже, сторінки «Співробітники», «Клієнти» та «Власники» створені з використанням однієї технології. Додавання нового співробітника та власника реалізовано через виклик модального вікна. Через необхідність введення даних про побажання клієнта, додавання нового клієнта відбувається на новій сторінці `addclient.php` (рис. 3.11).


```

public function actionAddclient()
{
    $client = new Client();
    $wish = new Wish();
    $model = new AddUserForm();
    $model->role=2;
    if($model->load(Yii::$app->request->post())){
        if($model->validate()){
            $hash = Yii::$app->getSecurity()->generatePasswordHash($model->password);
            $model->password=$hash;
            $model->save();
        }
    }
    if($client->load(Yii::$app->request->post())){
        if($client->validate()){
            $client->birthday=date("Y-m-d");
            $client->authorization_id=$model->id;
            $staff_id = Staff::find()->where(['authorization_id' => Yii::$app->user->identity->id]) ->one();
            $client->user_id=$staff_id->user_id;
            $client->save();
        }
    }
    if($wish->load(Yii::$app->request->post())){
        if($wish->validate()){
            $wish->client_id=$client->client_id;
            $wish->save();
            return $this->refresh();
        }
    }
    return $this->render('addclient', compact('model', 'client', 'wish'));
}

```

Рисунок 3.11 – Реалізація дії додавання нового клієнта

При додаванні нового клієнта у контролері створюються три об'єкти класів «Client» для запису особистих даних, «Wish» для запису даних побажань та «AddUserForm» для запису логіну та паролю для авторизації клієнта у системі. Раніше була показана обробка запитів методом POST, на рисунку 3.12 наведено приклад обробки GET запитів.

```

public function actionUpdateclient($id=null)
{
    if($id==null)
        return Yii::$app->getResponse()->redirect(Url::to(['default/client']));
    $clientForm = Client::findOne($id);
    if($clientForm->load(Yii::$app->request->post())){
        if($clientForm->validate()){
            $clientForm->save();
            return $this->refresh();
        }
    }
}

```

Рисунок 3.12– Обробка GET запитів

Для обробки GET запитів ідентифікатор даних для редагування передається як вхідний параметр до дії у контролері.

Для завантаження фото профілю співробітників, клієнтів та власників у відповідній дії створюється об'єкт класу UploadImage. Функції класу наведені на рисунку 3.13.

```
public function rules(){
    return[
        [['image'], 'file', 'extensions' =>
        ['jpg'], 'checkExtensionByMimeType'=>false]
    ];
}

public function upload(){
    if($this->validate()){
        $this->image->saveAs("uploads/{$this->image->baseName}.
        {$this->image->extension}");
    }else{
        return false;
    }
}
```

Рисунок 3.13 – Функції класу завантаження фото

Форма завантаження фото приймає файли у форматі .jpg. У той же час для завантаження фото та документів для об'єктів продажу використовується інший клас MultipleUpload, який дозволяє завантажувати декілька файлів одночасно у форматах .jpg та .pdf.

GridView віджет дуже зручний для роботи з одною моделлю, а не коли потрібно мати справу з даними з декількох таблиць. Тому для реалізації сторінок «Задачі» та «Договори» було обрано стандартні засоби HTML <table>. Вибірка даних за запитом користувача у цьому випадку відбувається у самому контролері без використання класів надання та пошуку даних. Для запису параметрів пошуку використовувалися змінні сесії, що проілюстровано на рисунку 3.14 на прикладі реалізації сторінки задач.

```

<table id="customers">
  <tr>
    <th>Назва</th>
    <th class="dt">Початок</th>
    <th class="dt">Кінець</th>
    <th>Виконавець</th>
    <th>Опис</th>
    <th>Опції</th>
  </tr>
  <tr>
    <tr>
      <form action="" method="post">
        <td><input type="text" id="title" name="title" value="<?=$_SESSION['task_title']?>"></td>
        <td><input type="datetime-local" id="start_" name="start_" value="<?=$_SESSION['task_start']?>" min="2019-01-01" max="2030-12-31"></td>
        <td><input type="datetime-local" id="end_" name="end_" value="<?=$_SESSION['task_end']?>" min="2019-01-01" max="2030-12-31"></td>
        <td><input type="text" id="staff" name="staff" value="<?=$_SESSION['task_staff']?>"></td>
        <td><button type="submit" class="btn btn-primary">Пошук</button></td>
      </form>
    </tr>
    <?php foreach($tasks as $task){ ?>|
    <tr>
      <td><?=$_task->title ?></td>
      <td><?=$_task->start_time?></td>
      <td><?=$_task->end_time?></td>
      <td><?=$_Staff::getName($task->assigned_to)?></td>
      <td><?=$_task->description?></td>
      <td>
        <button style="margin-right:5px; margin-bottom:3px;" type="button" id="<?=$_task->task_id?>"
          class="btn btn-sm add-task btn-primary" data-toggle="modal" data-target="#showTaskModal">
          <span class="glyphicon glyphicon-pencil" aria-hidden="true"></span></button>
      </td>
    </tr>
  </tr>
</table>

```

Рисунок 3.14 – Використання session для пошуку у таблиці задач

Після розгляду альтернатив було вирішено розробити власний календар. Для цього було створено клас Calendar, фрагмент коду якого наведено на рисунку 3.15.

```

public static function get_task_preview($y,$m,$d){
    $result=" ";
    $dt=Calendar::get_dt_value($y,$m,$d);
    $user_id=Staff::find()->where(['authorization_id'=> Yii::$app->user->identity->id])->one()['user_id'];
    $task=Task::findBySql("SELECT title, left(right(start_time,8),5) start_time FROM task WHERE left(start_time,10) = \"\$dt\" and assigned_to = $user_id" )->one();
    $meeting=Meeting::findBySql("SELECT meet_location, left(right(meet_dt,8),5) meet_dt FROM meeting WHERE left(meet_dt,10) = \"\$dt\" and staff_id = $user_id")->one();
    if(!is_null($task)) {
        $result.="<div class='task-bg'>$task->start_time $task->title</div>";
    }
    if(!is_null($meeting)) {
        $result.="<div class='meeting-bg'>$meeting->meet_dt $meeting->meet_location</div>";
    }
    return $result;
}

public static function get_task_and_meeting_list($y,$m,$d){
    $result=" ";
    $dt=Calendar::get_dt_value($y,$m,$d);
    $user_id=Staff::find()->where(['authorization_id'=> Yii::$app->user->identity->id])->one()['user_id'];
    $result.=Meeting::get_meeting_list($dt, $user_id, $y, $m);
    $result.=Task::get_task_list($dt, $user_id, $y, $m);
    return $result;
}

```

Рисунок 3.15 – Фрагмент коду класу Calendar

Важливими функціями у класі календар, окрім власне відображення календаря, є отримання масиву даних задач та зустрічей для перегляду. Для цього у функції `get_task_and_meeting_list` викликаються функції класів `Meeting` та `Task` для отримання результатів запитів до таблиць.

Для реалізації кнопки відображення сповіщень було змінено шаблон сторінок модуля співробітників, а саме панель навігації `NavBar`, що показано на рисунку 3.16.

```

<?php
NavBar::begin([
    'brandLabel' => '',
    'brandUrl' => '/admin/default/index.html',
    'options' => [
        'class' => 'navbar-inverse navbar-fixed-top',
    ],
]);
$id = Yii::$app->user->identity->id;
$staff = Staff::find()->where(['authorization_id'=> $id])->one();
echo Nav::widget([
    'options' => ['class' => 'navbar-nav navbar-right'],
    'items' => [
        ['label' => '', 'url' => '/admin/default/index.html'],
        Yii::$app->user->isGuest ? (
            ['label' => 'Login', 'url' => ['/login']]
        ) : (
            '<li>'.WishChangeLog::getNotifications().'</li><li>'
            . Html::beginForm(['/site/logout'], 'post')
            . Html::submitButton(
                'Вийти (' . $staff->first_name . ')',
                ['class' => 'btn btn-link logout']
            )
            . Html::endForm()
            . '</li>'
        )
    ],
]);
NavBar::end();
?>

```

Рисунок 3.16 – Реалізація кнопки відображення сповіщень

Функція `getNotifications()` класу `WishChangeLog` повертає усі непрочитані сповіщення з бази даних, які адресовані користувачу у вигляді HTML коду.

3.2.4 Розробка модуля для клієнтів

Файлова система модуля клієнтів складається з одного шаблону та трьох сторінок (рис. 3.17).

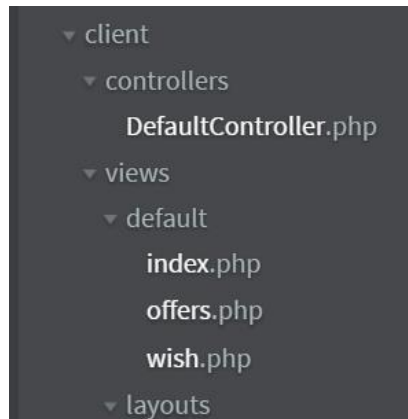


Рисунок 3.17 – Файлова система модуля клієнтів

На сторінці wish.php за допомогою форми Yii2 клієнт може змінити дані побажання. На головній сторінці спадні списки дозволяють змінити статус надісланої пропозиції чи зустрічі. Зокрема, на сторінці перегляду пропозиції було використано JavaScript для створення галереї зображень об'єкту. Програмна реалізація наведена на рисунку 3.18.

```

<?php foreach($object_images as $img){ ?>
<li>
    <a href="http://i.smartbuild.ml/web/uploads/<?= $img->name ?>"
      title="<?= $img->name ?>">name ?>" height="70"></a>
    </li>
<?php } ?>
</ul>
<?php endif; ?>
<script>
    thumbs.onclick = function(event) {
        let thumbnail = event.target.closest('a');
        if (!thumbnail) return;
        showThumbnail(thumbnail.href, thumbnail.title);
        event.preventDefault();
    }
    function showThumbnail(href, title) {
        largeImg.src = href;
        largeImg.alt = title;
    }
}
  
```

Рисунок 3.18 – Реалізація перегляду зображень об'єкту

3.3 Використання програмного додатку

Сторінка входу до системи зображена на рисунку 3.19. Якщо залишити поля пустими, з'являється повідомлення про необхідність введення даних для входу. Система є недоступною для неавторизованих користувачів, тому при спробі переходу до інших сторінок відбувається переадресація на сторінку входу.

← → ↻ Не захищено | i.smartbuild.ml/login.html

YOU REALTOR

Вхід

Будь-ласка заповніть дані для входу:

Логін Необхідно заповнити "Логін".

Пароль Необхідно заповнити "Пароль".

Запам'ятати

Login

Рисунок 3.19 – Форма входу

Модуль співробітника має 8 сторінок: «Головна», «Співробітники», «Клієнти», «Власники», «Об'єкти», «Календар», «Задачі» та «Договори». Бокове бургер меню зображене на рисунку 3.20. На відкритій сторінці показано календар з можливістю переглянути завдання та зустрічі заплановані на обраний місяць. Викликавши модельне вікно після натискання знаку додавання можна створити нову задачу або зустріч з клієнтом. Створена зустріч буде відображатися у кабінеті клієнта.

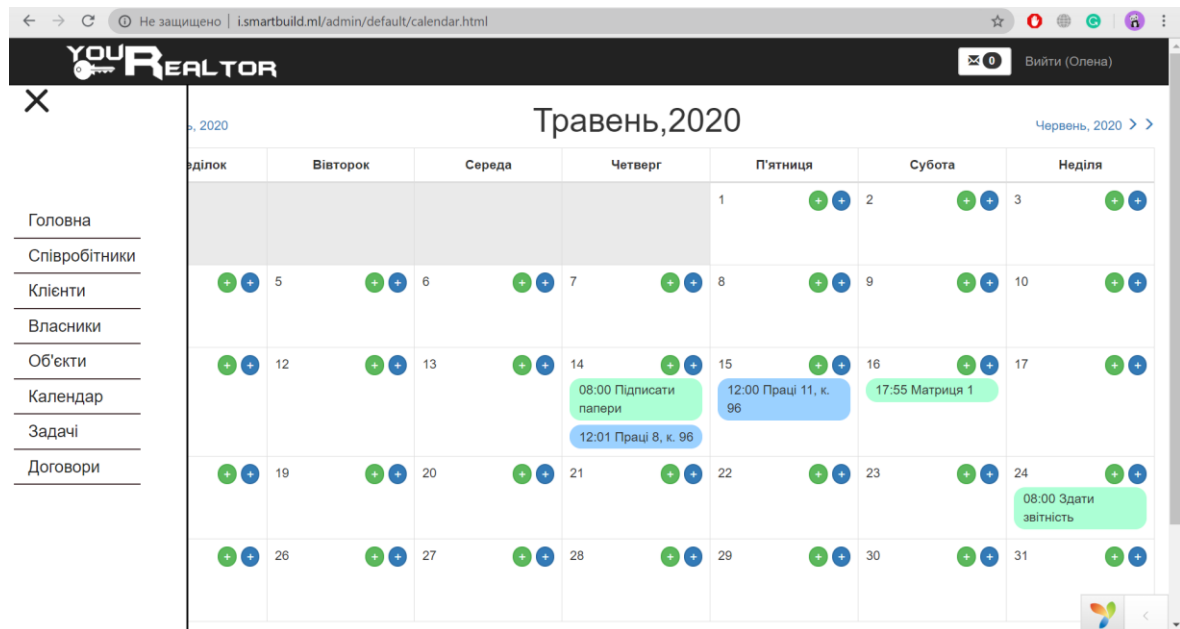


Рисунок 3.20 – Бокове меню та календар

Докладніше про задачі можна дізнатися клацнувши на обраний день, де співробітник має можливість відмітити завдання, як виконане чи ні (рис. 3.21).

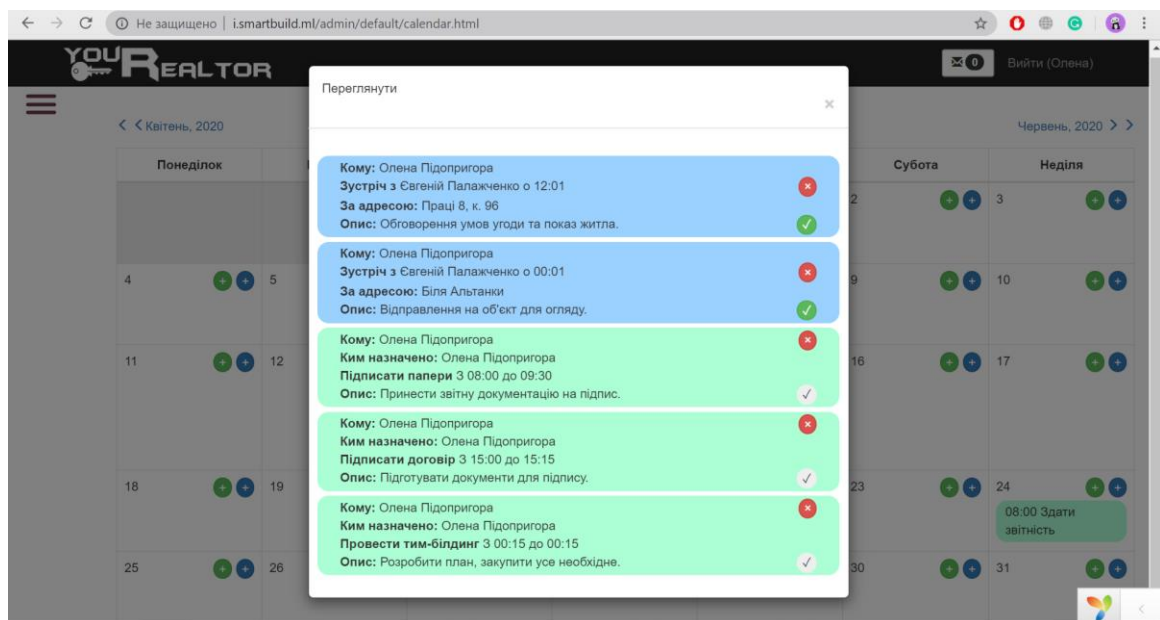


Рисунок 3.21 – Перегляд задач

На рисунку 3.22 наведено приклад пошуку за таблицею клієнтів, яка виконана за допомогою GridView. Пошук можна здійснити за полями прізвища, ім'я, по-батькові, також спадним списком можна обрати показ усіх чи тільки власних клієнтів.

GridView підтримує автоматичне сортування за алфавітом. Опції в останній колонці дозволяють переглянути повну інформацію з можливістю редагування, видалити запис та розпочати пошук за побажанням клієнта.

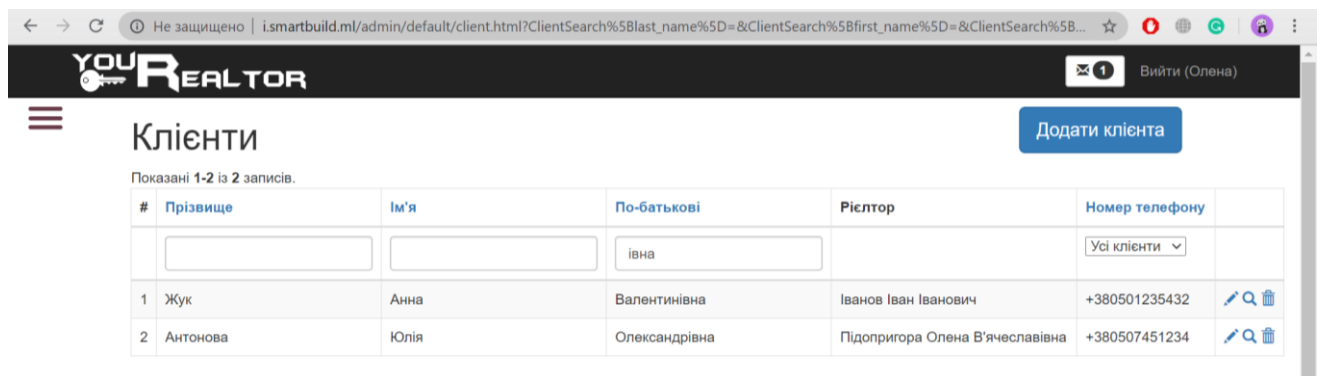


Рисунок 3.22 – Пошук за таблицею GridView

На прикладі форми сторінки власників наведено форму додавання нового запису (рис.3.23). У колонці опцій таблиці власників є можливість переглянути, видалити дані, а також додати нову квартиру чи будинок.

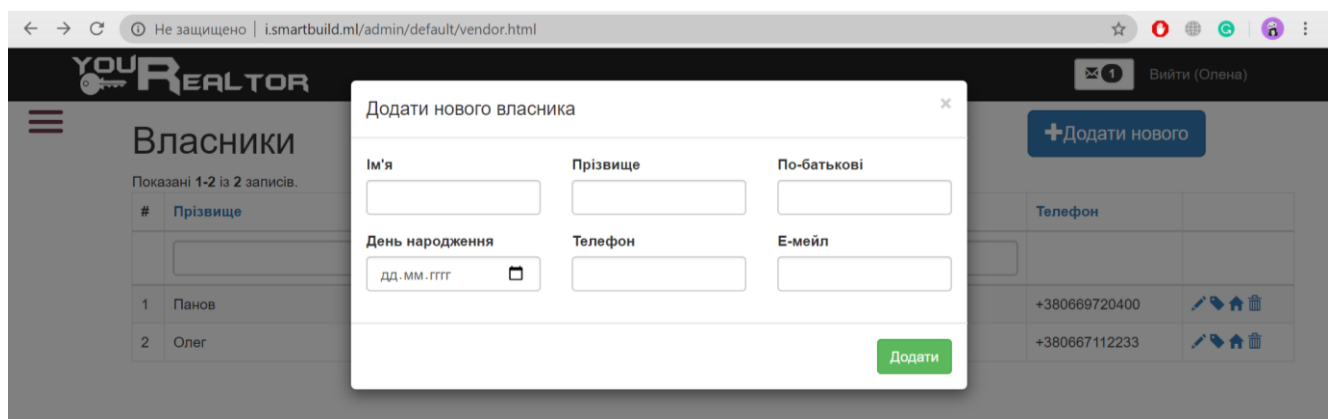


Рисунок 3.23 – Форма додавання

Зі сторінки перегляду усіх об'єктів є можливість перейти до сторінки редагування та до сторінки перегляду. На рисунку 3.24 зображено сторінку об'єкту у тому вигляді у якому вона відправляється до кабінету клієнта для перегляду пропозиції.

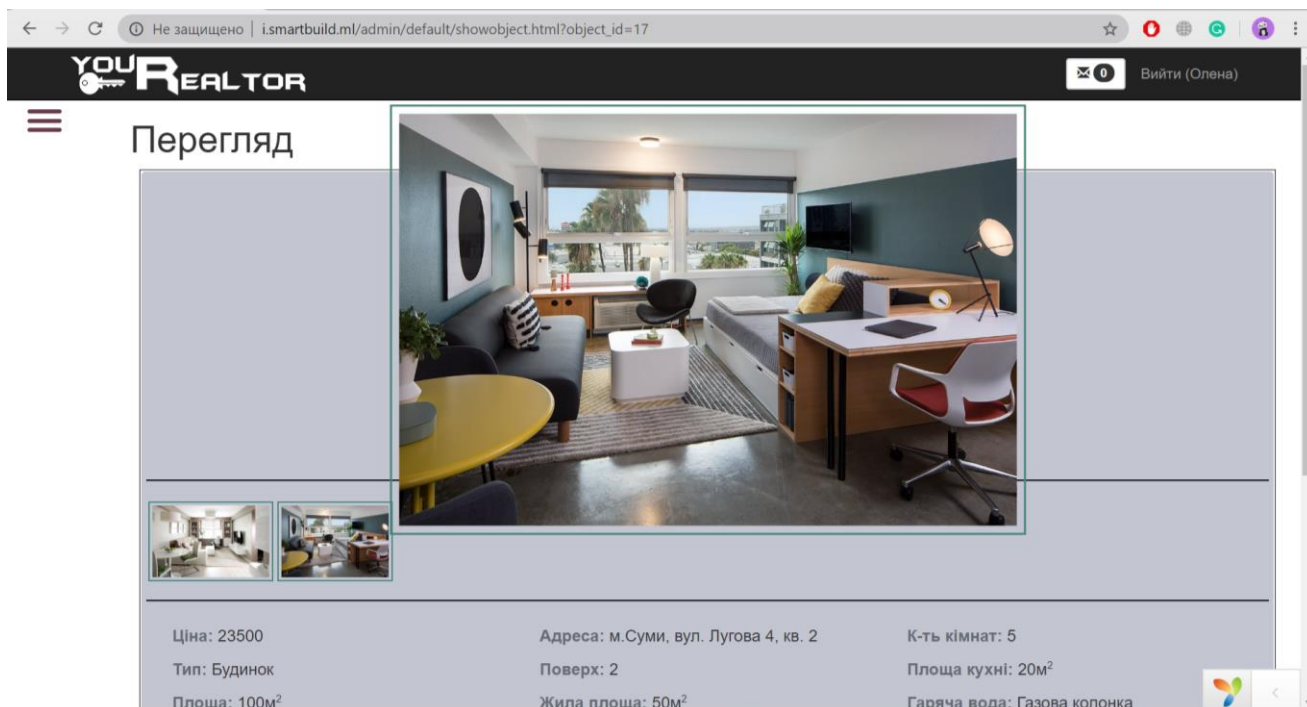


Рисунок 3.24 – Перегляд даних про нерухомість

На сторінці редагування даних про нерухомість окрім можливості зміни загальної інформації є функція додавання файлів у форматі .jpg та .pdf. Після завантаження файли відображаються у відведених блоках «Фотографії» та «Документи». Сторінка редагування даних зображена на рисунку 3.25.

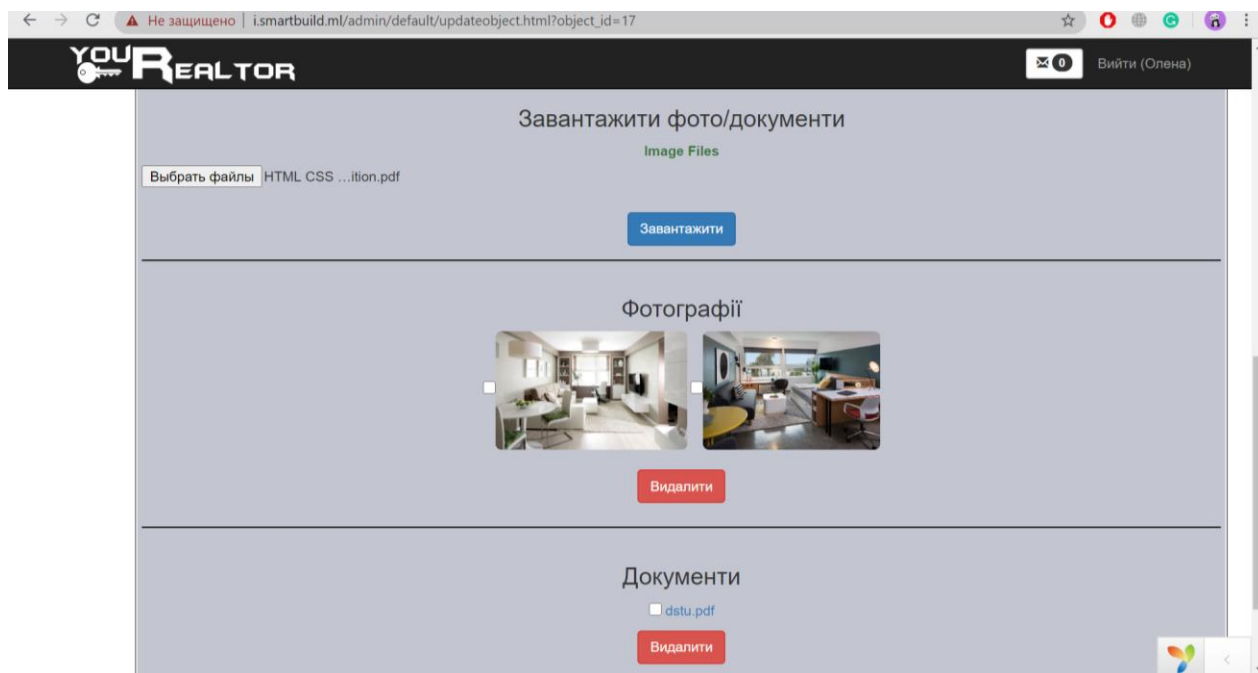


Рисунок 3.25 – Форма завантаження файлів

Якщо у календарі відображаються власні зустрічі та задачі авторизованого користувача, на сторінці «Задачі» менеджер має змогу переглянути задачі підлеглих йому ріелторів та призначити їм нові. У таблиці є можливість пошуку за назвою та датою виконання. Приклад використання сторінки «Задачі» з відкритим полем для вибору дати наведений на рисунку 3.26.

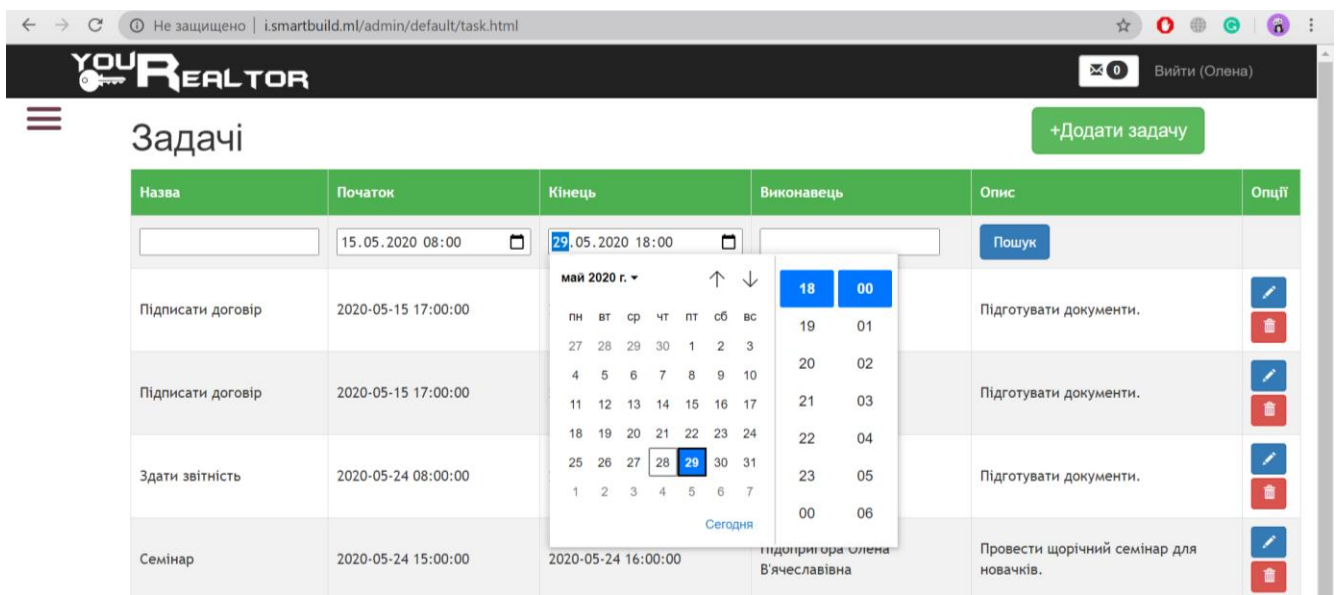


Рисунок 3.26 – Пошук у таблиці

Як вже згадувалося, зі сторінки клієнта можна розпочати пошук за його побажанням. Для цього достатньо натиснути кнопку «Пошук» з іконкою лупи у останній колонці опцій обраного клієнта. На сторінці, що відкриється з'явиться форма з уже введеними даними з перемикачами для виключення чи включення параметрів у пошук. Сторінка пошуку з результатами запиту зображена на рисунку 3.27. Кнопка «Перегляд» дозволяє відкрити знайдений запис на новій сторінці для перегляду більш докладної інформації, а кнопка з іконкою конверта надсилає знайдену пропозицію до кабінету клієнта.

YOU REALTOR Вийти (Олена)

Побажання

Мін. ціна	Макс. ціна	Мін. поверх	Макс. поверх
100	30000	1	15

Адреса
Суми

Мін. к-ть кімнат	Макс. к-ть кімнат	Мін. заг. площа	Макс. заг. площа
1	6	25	150

Ремонт: Без ремонту

Опалення: Автономне

Матеріал будівлі: Газоблок

Гаряча вода: Центр. постачання

Опис
Потрібна квартира біля великого універмагу або ринку.

Результати

Результат 1:
 Ціна: 28000
 Площа: 89м²
 Статус ремонту: Без ремонту
 Адреса: м. Суми, вул. Річна 5, кв. 2
 Поверх: 7
 К-ть кімнат: 4
 Жила площа: 40м²
 Площа кухні: 34м²
 Опалення: Центральне
 Гаряча вода: Центр. постачання
 Матеріал стін: Цегла

Результат 2:
 Ціна: 23000
 Площа: 70м²
 Статус ремонту: Авторський дизайн
 Адреса: м. Суми, вул. Річна 25, кв. 105
 Поверх: 7
 К-ть кімнат: 3
 Жила площа: 40м²
 Площа кухні: 20м²

Рисунок 3.27 – Пошук об'єктів

Усі надіслані пропозиції та зустрічі з'являються на головній сторінці у кабінеті клієнта (рис. 3.28). Обравши опцію спадного списку, користувач може змінити статус запису. На сторінці також відображається контактна інформація про ріелтора клієнта.

← → ↻ Не захищено | i.smartbuild.ml/client/default/index.html

YOU REALTOR Головна Побажання Вийти (Юлія)

Ласкаво просимо!
 Ми допоможемо вам отримати житло своєї мрії!
 Дякуємо, що користуєтеся нашими послугами

Ваш ріелтор

Підпригора Олена В'ячеславівна
 📅 1990-03-23
 📞 +380661112233
 ✉ o.pidoprygora@gmail.com

Призначені зустрічі

Де:	Опис:	Статус:	Дії:
Праці 11, к. 96 Коли: 2020-05-15 12:00:00	Відправлення на об'єкт для огляду.	Прийняти	Змінити

Пропозиції

Пропозиція №	Дії:	Відправлена	Статус:	Дії:
Пропозиція № 1	👁 Переглянути	2020-05-13 16:44:03	Відкрита	Змінити

Рисунок 3.28 – Головна сторінка у кабінеті клієнта

Друга сторінка модуля клієнта призначена для зміни побажання (рис. 3.29).

Побаження

Об'єкт: Будинок | Мін. ціна: 30000 | Макс. ціна: 50000 | Адреса: Суми

Мін. поверх: 1 | Макс. поверх: 2 | Мін. к-ть кімнат: 8 | Макс. к-ть кімнат: 9 | Мін. заг. площа: 124 | Макс. заг. площа: 234

Ремонт: Без ремонту | Матеріал будівлі: Цегла-дерево | Опалення: Центральне | Гаряча вода: Газова колонка

Опис: еее

Газ: € | Гараж: - | Мін. площа діл.: 0 | Макс. площа діл.: 0

[Змінити](#)

Рисунок 3.29 – Сторінка зміни побажання

Після зміни клієнтом запису до кабінету ріелтору надходить сповіщення, яке відображається у верхньому меню сторінки (рис. 3.30).

Співробітники

Показані 1-2 із 2 записів.

#	Прізвище	Ім'я	По-батькові	Телефон
1	Підпригора	Олена	В'ячеславівна	+380661112233
2	Іванов	Іван	Іванович	+380507411111

Рисунок 3.30 – Відображення сповіщень

Функціональність системи була перевірена під час тестування. Результати проведення тестування у вигляді тест-кейсів зі статусом проходження наведені у додатку Д.

ВИСНОВКИ

У ході реалізації дипломного проекту було опрацьовано літературні джерела за темою розробки систем управління взаємовідносинами з клієнтами та проаналізовано продукти аналогів, які надають підтримку процесів продажу. У ході аналізу було виявлено недолік систем, який виявляється у відсутності надання клієнтові доступу до системи. Тому, метою роботи стала розробка інформаційної системи для взаємодії з клієнтами агентства нерухомості, яка б налагодила канал зв'язку між клієнтами та співробітниками.

Після аналізу сучасного стану питання було розроблено технічне завдання та виконане планування робіт. Перед початком реалізації було проведено проектування архітектури системи, бази даних та виділення варіантів використання.

Розроблена система відповідає усім поставленим функціональним вимогам, а саме:

- надавати CRUD можливості для роботи з даними клієнтів, власників, співробітників, об'єктів продажу, задач та договорів;
- забезпечувати функціонал для пошуку об'єкта за побажанням клієнта;
- надавати можливість планування, призначення задач та зустрічей. Повинен бути присутній фільтр за виконавцем, датою та назвою задачі;
- забезпечувати різні рівні доступу до функціоналу модуль адміністратора для ріелтора та менеджера;
- надавати особистий кабінет клієнту для зміни побажань, перегляду інформації про ріелтора, пропозиції та призначені зустрічі.

Інформаційну систему було розроблено за допомогою об'єктно-орієнтованого PHP фреймворку Yii2 з використанням PHP, JS, HTML та CSS.

Отже, виконавши усі етапи дипломного проекту, було розроблено інформаційну систему підтримки діяльності продажу нерухомості, яка забезпечує

взаємодію між клієнтами та співробітниками агентства. Використання системи дозволяє скоротити час на пошук потрібного варіанту нерухомості серед сотень доступних для надання якісних послуг.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Де в Україні продали найбільше квартир у 2018: промовисті цифри – інфографіка [Електронний ресурс] // 24 канал. – 2019. – Режим доступу до ресурсу: https://24tv.ua/de_v_ukrayini_proдали_naybilshe_kvartir_u_2018_promovisti_tsifri__info_grafika_n1123299.
2. Белка Я.С. Інформаційна система взаємодії з клієнтами агентства нерухомості. / Белка Я.С., Шендрик В.В. // Інформатика, математика, автоматика: матеріали та програма науково-технічної конференції, м. Суми, 2020 р. – Суми : СумДУ, 2020. – 93 с.
3. Buttle F. Customer Relationship Management / Francis Buttle. – Burlington: Springer Berlin Heidelberg, 2012. – 523 с.
4. Payne A. Handbook of CRM: Achieving Excellence in Customer Management / Adrian Payne., 2005. – 459 с.
5. Customer relationship management [Електронний ресурс]. – 2013. – Режим доступу до ресурсу: [https://www.onebusiness.ca/sites/default/files/MEDI_Booklet_Customer_Relationship_Management_Accessible_E.pdf].
6. Mega Realtor [Електронний ресурс] – Режим доступу до ресурсу: [<http://mega-realtor.com.ua/>].
7. Realtsoft CRM [Електронний ресурс] – Режим доступу до ресурсу: [<http://realtsoft.net/>].
8. Model view controller [Електронний ресурс] – Режим доступу до ресурсу: [<https://whatis.techtarget.com/definition/model-view-controller-MVC>].
9. Українська спільнота Yii Framework [Електронний ресурс] – Режим доступу до ресурсу: [<https://yiiframework.com.ua/uk/doc/guide/2/>].
10. Lemay L. Sams Teach Yourself HTML, CSS & JavaScript Web Publishing in One Hour a Day / L. Lemay, R. Coburn, J. Kyrnin. – Indianapolis, 2016. – 1895 с.
11. Мулеса О. Інформаційні системи та реляційні бази даних / Оксана Мулеса. – Ужгород, 2018. – 118 с.

12. Карпенко М. Ю. Технології створення програмних продуктів та інформаційних систем : навч. посібник / М. Ю. Карпенко, Н. О. Манакова, І. О. Гавриленко ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2017. – 93 с.
13. Нетепчук В. В. Управління бізнес-процесами / В. В. Нетепчук. – Рівне: НУВГП, 2014. – 158 с.
14. UML Use Case Diagrams [Електронний ресурс] – Режим доступу до ресурсу: [<https://www.uml-diagrams.org/use-case-diagrams.html>]
15. What is an Entity Diagram (ERD)? [Електронний ресурс] – Режим доступу до ресурсу: [<https://medium.com/@soni.dumitru/what-is-an-entity-relationship-diagram-erd-13daee5b2a>]
16. MVC Definition [Електронний ресурс] – Режим доступу до ресурсу: [<https://techterms.com/definition/mvc>].
17. The Definitive Guide to Yii 2.0 Widgets [Електронний ресурс] – Режим доступу до ресурсу: [<https://www.yiiframework.com/doc/guide/2.0/en/structure-widgets>]
18. Hostpro Найкращий хостинг в Україні [Електронний ресурс] – Режим доступу до ресурсу: [<https://hostpro.ua/ua/hosting-linux.html>]
19. Brackets A modern, open source text editor that understands web design. [Електронний ресурс] – Режим доступу до ресурсу: [<http://brackets.io/>]

ДОДАТОК А

Акт впровадження

АКТ

впровадження результатів студентської наукової роботи

м. Київ

«28» травня 2020 р.

Впровадження результатів студентської наукової роботи «ІНФОРМАЦІЙНА СИСТЕМА ВЗАЄМОДІЇ З КЛІЄНТАМИ АГЕНТСТВА НЕРУХОМОСТІ» у діяльність відділу розробки програмного забезпечення компанії ТОВ "СТРУКТУРА ОПТИМА"

Комісія фахівців компанії ТОВ "СТРУКТУРА ОПТИМА" в м. Київ у складі: директора компанії Грицай Р. Ю. та технічного директора Ткаченка М. М. склала цей акт в тому, що результати студентської дипломної роботи Белка Я.С. на тему «Інформаційна система взаємодії з клієнтами агентства нерухомості» після періоду тестування та корекцій було впроваджено у діяльність відділу розробки програмного забезпечення компанії.

Впровадження результатів студентської наукової роботи Белка Я.С. дозволяє робітникам компанії Структура Оптима покращити виробничий процес шляхом зменшення часових витрат на виконання проектів та збільшується точність введених даних та полегшує подальше їх використання.

Голова комісії:



Грицай Р. Ю.

Комісія:

A handwritten signature in black ink, likely belonging to M. M. Tkachenko.

Ткаченко М. М.

ДОДАТОК Б

Технічне завдання

1 Призначення й мета створення інформаційної системи

1.1 Призначення інформаційної системи

Інформаційна система повинна надавати канал зв'язку між співробітниками ріелторської фірми та її клієнтами.

1.2 Мета створення інформаційної системи

Інформаційна підтримка діяльності продажу нерухомості.

1.3 Цільова аудиторія

У цільовій аудиторії інформаційної системи можна виділити дві групи:

1. Співробітники ріелторської фірми.
2. Клієнти-покупці.

2 Вимоги до інформаційної системи

2.1 Вимоги до структури й функціонування інформаційної системи

Інформаційна система повинна бути реалізована у вигляді сайту, доступного в мережі Інтернет під доменним іменем www.i.smartbuild.ml. Сайт повинен складатися із взаємозалежних розділів із чітко розділеними функціями. Детальніше про структуру розділів у пункті 3.1 Структура сайту.

2.2 Вимоги до персоналу

Для підтримки й експлуатації сайту від персоналу не вимагається спеціальних технічних навичок, знання технологій або програмних продуктів, за винятком загальних навичок роботи з персональним комп'ютером і стандартним веб-браузером (наприклад, Google Chrome 80.0.3987.149 або вище).

2.3 Вимоги до збереженні інформації

Збереження даних повинно бути реалізоване у реляційній базі MySQL.

2.4 Вимоги до розмежування доступу

Інформація, яка розташована на сайті, не є загальнодоступною. Незареєстровані користувачі та гості сайту мають доступ лише до сторінки входу. Доступ до сайту повинен здійснюватися з використанням унікального логіна та пароля.

Користувачів сайту можна розділити на 3 групи відповідно до прав доступу:

1. Менеджер.
2. Ріелтор.
3. Клієнт-покупець.

Менеджери мають доступ до всіх частин сайту та можуть користуватися всіма, передбачуваними у вимогах, функціями.

Ріелтори мають обмежений, порівняно з менеджерами, доступ до інформації та функціоналу.

Клієнти-покупці можуть переглядати власний кабінет та змінювати інформацію в ньому.

2 Основні вимоги

2.1 Структура сайту

Система повинна надавати два модулі «Адмін» та «Клієнт». Перший розроблений для співробітників, другий для клієнтів. Модулі відрізняються шаблоном, сторінками та їх контентом. Модуль «Адмін» повинен складатися з наступних розділів: «Головна», «Співробітники», «Клієнти», «Власники», «Об'єкти», «Календар», «Задачі», «Договори».

Структура сайту для менеджера:

- Головна – привітальна сторінка, містить загальну інформацію про підприємство.
- Співробітники – менеджер може додавати, редагувати та видаляти співробітників.
- Клієнти – менеджер може додавати, видаляти та редагувати інформацію про нового клієнта та переглядати сторінки побажань покупців. На сторінці доступна форма пошуку та фільтрації за власними та усіма клієнтами.
- Власники – менеджер може додавати, видаляти та редагувати інформацію про власників та додавати нерухомість через сторінку власника. На сторінці доступна форма пошуку.
- Об'єкти – менеджер може видаляти та редагувати інформацію про новий об'єкт та переглядати інформацію про нерухомість.
- Календар – надає графічне представлення у формі календаря всіх задач та зустрічей співробітників. Менеджер може не тільки додавати власні зустрічі та задачі, але й присвоювати задачі на виконання співробітникам.
- Задачі – на сторінці відображаються задачі співробітників офісу, на якій менеджер може редагувати, видаляти та додавати задачі. Присутній фільтр за датою, пошук за виконавцем та назвою задачі.
- Договори – доступна функція пошуку та перегляду усіх договорів.

Структура сайту для ріелтора:

- Головна – привітальна сторінка, містить загальну інформацію про підприємство.

- Співробітники – перегляд сторінки.

- Клієнти – ріелтор може додавати, видаляти та редагувати інформацію про нового клієнта. Він має доступ до листу побажань лише своїх клієнтів. На сторінці доступна форма пошуку та фільтрації за власними та усіма клієнтами.

- Власники – ріелтор може додавати, видаляти та редагувати інформацію про власників та додавати нерухомість через сторінку власника. На сторінці доступна форма пошуку.

- Об'єкти – ріелтор може видаляти та редагувати інформацію про об'єкт та переглядати інформацію. На сторінці доступна форма пошуку.

- Календар – у формі календаря представлені всі задачі та зустрічі ріелтора. Ріелтор може додавати власні зустрічі та задачі. Є можливість змінити статус на завершений або видалити запис.

- Задачі – на сторінці відображаються власні задачі та зустрічі ріелтора, де він може переглядати та змінювати задачі. Присутній фільтр за датою та пошук за назвою задачі.

- Договори – доступна функція додавання нового договору та перегляду договорів укладених за участю самого ріелтора.

Модуль «Клієнт» повинен складатися з розділів «Головна» та «Побажання».

Структура сайту для клієнта-покупця:

- Головна – користувач має доступ до перегляду своїх особистих даних. Відображаються надіслані пропозиції з даними про нерухомість, назначені зустрічі та інформація про ріелтора та відповідального менеджера.

- Побажання – користувач може змінювати побажання до об'єкту купівлі. Після відповідної зміни, на сторінку відповідного ріелтора надходить сповіщення.

2.2 Навігація

Інтерфейс сайту повинен бути дружнім до користувач, мати швидкі і логічні переходи до розділів і сторінок. Навігаційні елементи повинні містити контент та можливості вказані у пункті 3.1 Структура сайту.

Система повинна забезпечувати навігацію по всіх доступних користувачеві сторінках і відображати відповідну інформацію. Для навігації у модулі співробітників потрібно використовувати бокове меню. Верхнє меню має кнопку виходу та кнопку для перегляду непрочитаних сповіщень. Навігація у модулі клієнта повинна бути реалізована у верхньому меню.

2.3 Наповнення сайту (контент)

Сторінки всіх розділів сайту формуються безпосередньо під час роботи співробітників підприємства.

2.4 Система навігації (карта сайту)

Взаємозв'язок між розділами й підрозділами сайту (карта сайту) представлено на рисунку А.1 – для менеджера сайту; А.2 – для ріелтора; А.3 – для клієнта-покупця.

Концептуальна схема сайту (Менеджер)

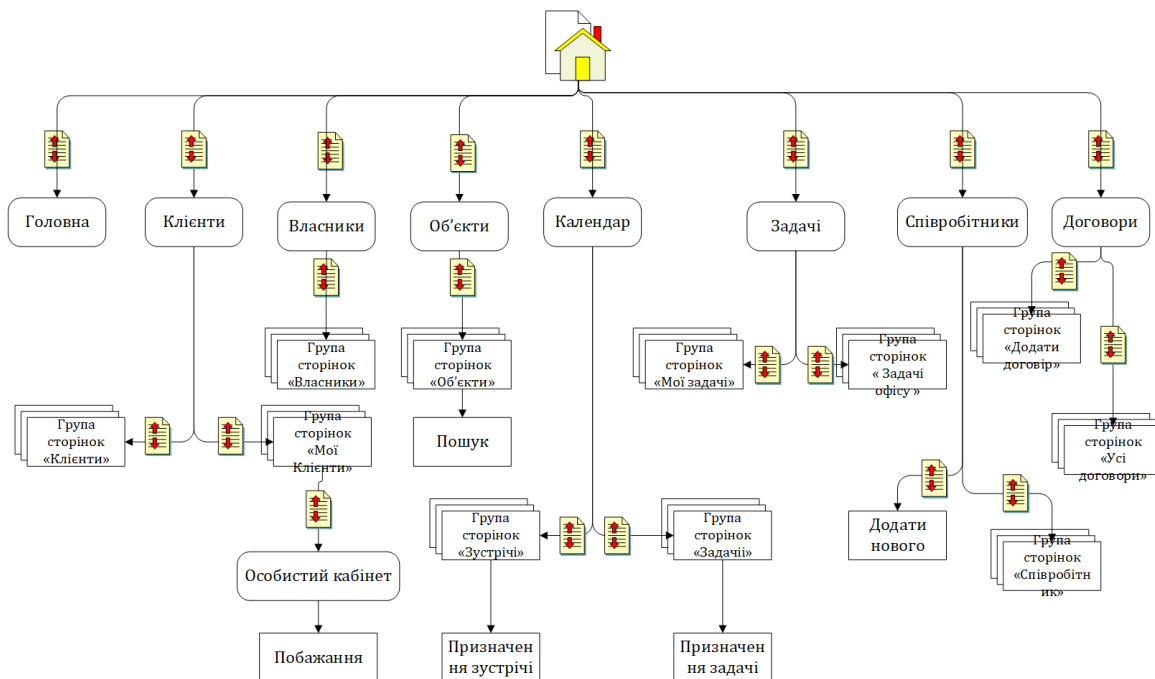


Рисунок А.1– Карта сайту для менеджера

Концептуальна схема сайту (Ріелтор)

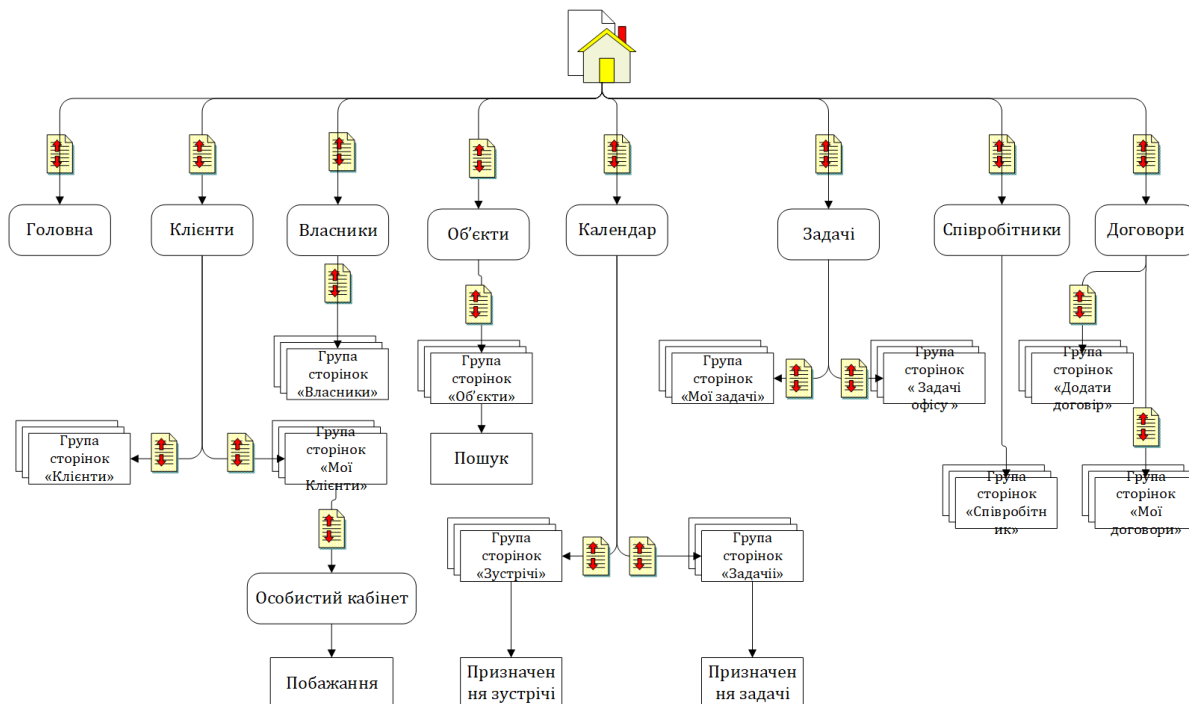


Рисунок А.2– Карта сайту для ріелтора

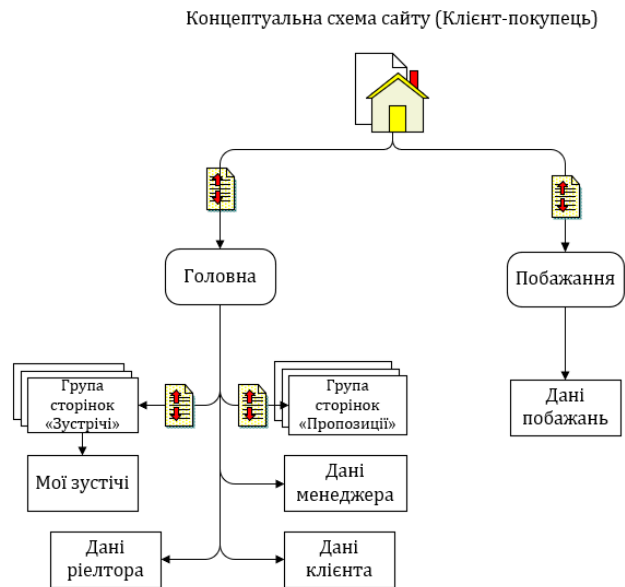


Рисунок А.3– Карта сайту для клієнта

2.5 Вимоги до функціональних можливостей

Система повинна надавати наступні можливості:

- надавати CRUD можливості для роботи з даними клієнтів, власників, співробітників, об'єктів продажу, задач та договорів;
- забезпечувати функціонал для пошуку об'єкта за побажанням клієнта;
- надавати можливість планування, призначення задач та зустрічей. Повинен бути присутній фільтр за виконавцем, датою та назвою задачі;
- забезпечувати різні рівні доступу до функціоналу модуль адміністратора для рієлтора та менеджера;
- надавати особистий кабінет клієнту для зміни побажань, перегляду інформації про рієлтора, пропозиції та призначені зустрічі.

2.6 Функціональні можливості розділів

Функціональні можливості розділів викладені раніше у розділі 3.1 Структура сайту для кожного з трьох типів користувачів.

На головній сторінці окрім розділів меню у шапці сайту будуть представлені наступні елементи:

- Кнопка виходу з системи.
- Іконка з зображенням листів та лічильником непрочитаних сповіщень.
При відкритті якої, спадним списком відображаються сповіщення з кнопкою «Помітити як прочитане».

2.7 Загальні вимоги

Стиль сайту можна описати як сучасний, діловий. У якості кольорової палітри рекомендується використовувати нейтральні пастельні відтінки.

Оформлення не повинне привертати до себе увагу, а лише підкреслювати розділення контенту на сайті.

Розташування елементів на сторінці перегляду об'єктів продажу з правами ріелтора схематично показано на рис. А.4.

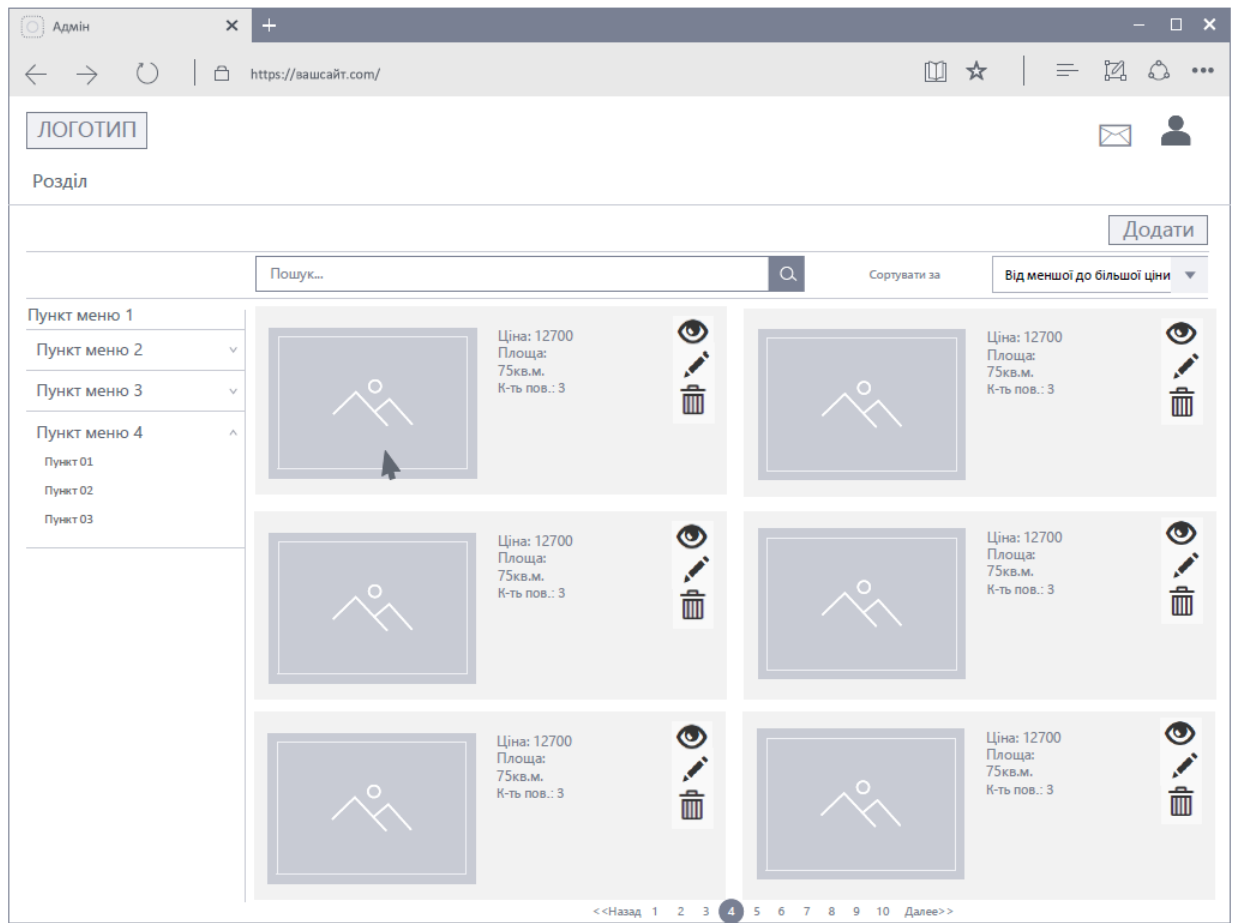


Рисунок А.4 – Типова сторінка

2.8 Типові навігаційні й інформаційні елементи

- Шапка сайту з панеллю меню
- Бокове-меню
- Основне поле контенту

2.8.1 Шапка сайту

Шапка сайту повинна містити логотип і назву сайту. Логотип є посиланням на головну сторінку сайту, а також категорії меню, викладені у структурі сайту.

2.8.2 Основне поле контенту

Основне поле контенту повинне розташовуватися в центрі сторінки по всій її ширині. У цьому полі відображається основний зміст обраного розділу. Стильове

оформлення матеріалів і їх елементів (посилань, заголовків, основного тексту, зображень, форм, таблиць і т.п.) повинне бути єдиним для всього веб-сайту.

3 Вимоги до видів забезпечення

3.1 Вимоги до інформаційного забезпечення

Реалізація сайту відбувається з використанням:

- PHP 7
- MySQL 5.6
- PHP Framework Yii2
- HTML
- CSS
- phpMyAdmin
- JavaScript
- Bootstrap

3.2 Вимоги до лінгвістичного забезпечення

Сайт повинен бути виконаний українською мовою.

3.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище;

4 Склад і зміст робіт зі створення сайту

Докладний опис етапів роботи зі створення сайту наведено на рис. А.5.

№	Назва задачі	Длительность	Начало	Окончание	Пред	Названия ресурсов
1	Розробка бази даних	5 днів	Ср 01.04.20	Вт 07.04.20		Белка Я.С.
2	Аналіз об'єктів	1 день	Ср 01.04.20	Ср 01.04.20		Белка Я.С.
3	Створення моделі бази	2 днів	Чт 02.04.20	Пт 03.04.20	2	Белка Я.С.
4	Написання скриптів для створення таблиць	1 день	Пн 06.04.20	Пн 06.04.20	3	Белка Я.С.
5	Інтеграція бази	1 день	Вт 07.04.20	Вт 07.04.20	4	Белка Я.С.
6	Розробка інтерфейсу	6 днів	Ср 08.04.20	Ср 15.04.20	1;5	Белка Я.С.
7	Розробка сценаріїв користувачів	1 день	Ср 08.04.20	Ср 08.04.20	5	Белка Я.С.
8	Розробка карти сайту	1 день	Чт 09.04.20	Чт 09.04.20	7	Белка Я.С.
9	Розробка інтелект-карти	1 день	Пт 10.04.20	Пт 10.04.20	8	Белка Я.С.
10	Вибір стилістики	1 день	Пн 13.04.20	Пн 13.04.20	9	Белка Я.С.
11	Прототипування	2 днів	Вт 14.04.20	Ср 15.04.20	10	Белка Я.С.
12	Реалізація	22 днів	Чт 16.04.20	Пт 15.05.20	6;11	Белка Я.С.
13	Верстка шаблону та сторінок модулю адміна	6 днів	Чт 16.04.20	Чт 23.04.20	11	Белка Я.С.
14	Верстка шаблону та сторінок модулю клієнта	3 днів	Пт 24.04.20	Вт 28.04.20	13	Белка Я.С.
15	Розробка модулю адміна	10 днів	Ср 29.04.20	Вт 12.05.20	14	Белка Я.С.
16	Розробка модулю клієнта	3 днів	Ср 13.05.20	Пт 15.05.20	15	Белка Я.С.
17	Тестування	10 днів	Пн 18.05.20	Пт 29.05.20	12	Белка Я.С.
18	Написання тест-кейсів до модулю адміна	2 днів	Пн 18.05.20	Вт 19.05.20	16	Белка Я.С.
19	Написання тест-кейсів до модулю клієнта	1 день	Ср 20.05.20	Ср 20.05.20	18	Белка Я.С.
20	Проведення тестувань та оформлення баг-репортів	2 днів	Чт 21.05.20	Пт 22.05.20	19	Белка Я.С.
21	Виправлення несправностей	5 днів	Пн 25.05.20	Пт 29.05.20	20	Белка Я.С.

Рисунок А.5 – Зміст роботи проекту

Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) - 43 дні.

5 Вимоги до складу й змісту робіт із введення сайту в експлуатацію

Для введення сайту в експлуатацію потрібно перенести його на хостинг разом з базою даних.

ДОДАТОК В

Планування робіт

Деталізація мети проекту методом SMART. Продуктом дипломного проекту є веб-додаток «Інформаційна система взаємодії з клієнтами агентства нерухомості», призначений для використання ріелторськими організаціями. Перед початком виконання проекту було проведено деталізацію мети методом SMART. Результати деталізації розміщені у табл. Б.1.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Розробити інформаційну систему у вигляді веб-додатку для підтримки взаємодії з клієнтами агентства нерухомості.
Measurable (вимірювана)	Головним критерієм результату є оцінка приймальної комісії.
Achievable (досяжна)	Реалізація системи здійснюється за допомогою PHP фреймворку Yii2, з використанням HTML, CSS, PHP та JS.
Relevant (реалістична)	Для розробки веб-додатку, достатньо кваліфікований розробник має усі необхідні програми, технічні засоби та доступ до Інтернет-мережі.
Time-framed (обмежена у часі)	Інформаційна система створюється у відповідності до дат, зазначених у календарному плані.

Планування змісту структури робіт. Планування змісту структури робіт здійснено за допомогою WBS діаграми, яка графічно представляє елементи проекту у вигляді пакетів робіт. Структура декомпозиції робіт є інструментом

розбиття великих задач на менші, що допомагає краще управляти проектом. Діаграма WBS зображена на рис. Б.1.

Планування структури організації, для впровадження готового проекту (OBS). Організаційна структура виконавців OBS забезпечує призначення команд чи окремих виконавців для виконання кожної задачі. Діаграма OBS представлена на рис. Б.2. Список виконавців в проекті наведений в табл. Б.2.

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Белка Я.С.	Виконує розробку бази даних, інтерфейсу користувача та реалізацію функціоналу системи.
Тестувальник	Белка Я.С.	Відповідає за тестування функціоналу та дизайну додатку.
Консультант проекту	Шендрик В.В.	Формує технічне завдання на розробку проекту. Затверджує модель бази даних та інтерфейс системи.
Менеджер проекту	Белка Я.С.	Відповідає за додержання термінів, розподіл ресурсів та завдань між учасниками. Виконує збір та аналіз даних.

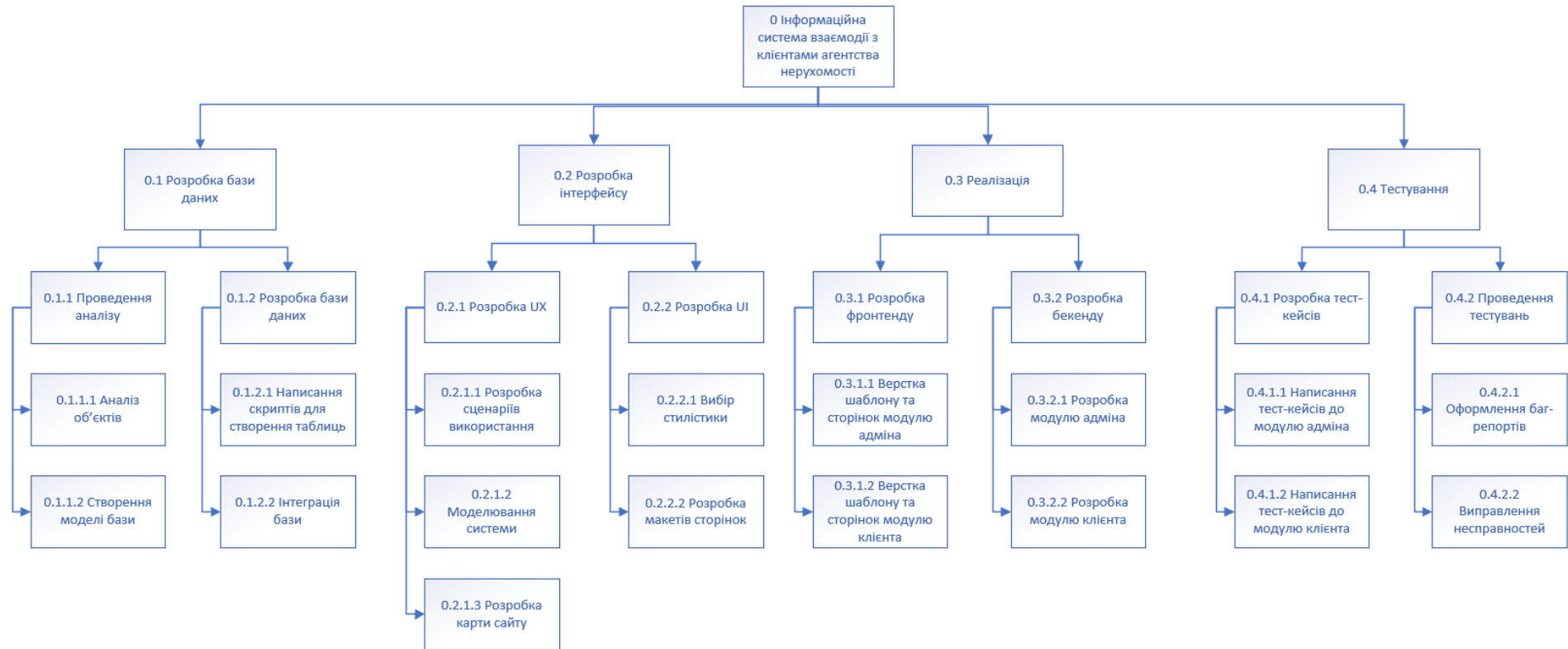


Рисунок Б.1 – WBS Структура робіт проекту

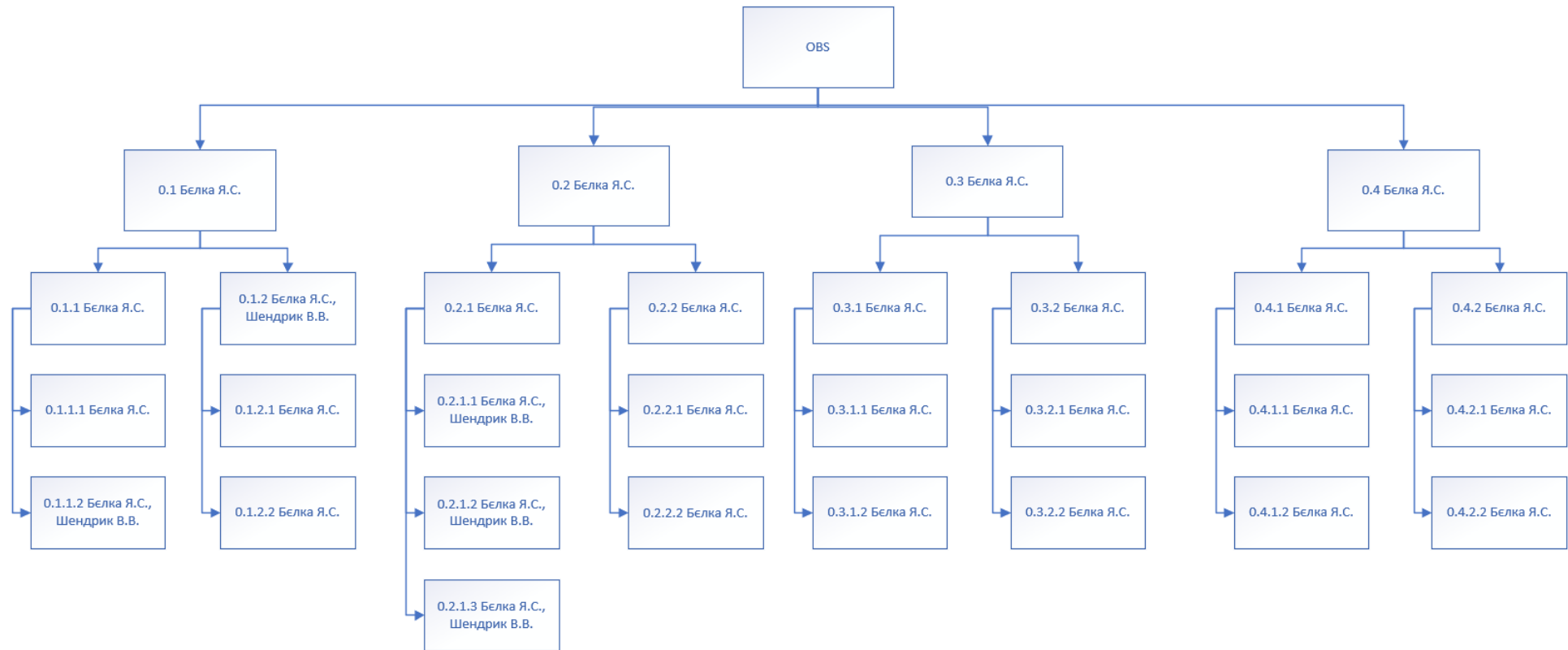


Рисунок Б.2 – Організаційна структура проекту (OBS)

Діаграма Ганта. Календарний план виконання проекту виконаний у формі діаграми Ганта. Діаграма візуалізує графіки часу і взаємозв'язок між окремими завданнями та етапами роботи над проектом. Тривалість виконання робіт зазначена в днях. Фактична тривалість виконання робіт дорівнює 6 годинам на день. Діаграма Ганта та список робіт зображені на рис. Б.3-Б.4.

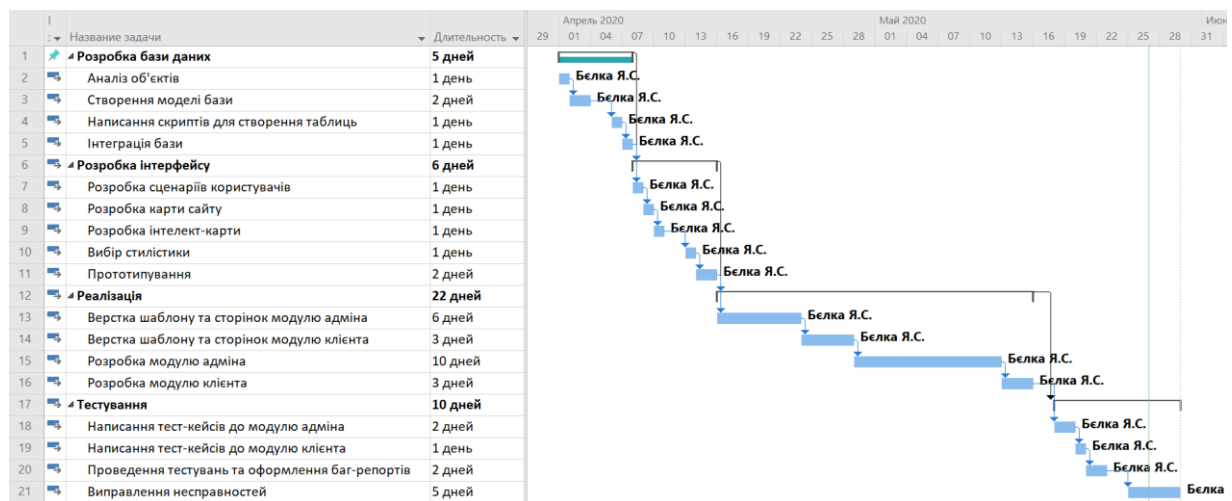


Рисунок Б.3 – Діаграма Ганта

	Назва задачі	Длительность	Начало	Окончание	Пред	Названия ресурсов
1	Розробка бази даних	5 днів	Ср 01.04.20	Вт 07.04.20		Белка Я.С.
2	Аналіз об'єктів	1 день	Ср 01.04.20	Ср 01.04.20		Белка Я.С.
3	Створення моделі бази	2 днів	Чт 02.04.20	Пт 03.04.20	2	Белка Я.С.
4	Написання скриптів для створення таблиць	1 день	Пн 06.04.20	Пн 06.04.20	3	Белка Я.С.
5	Інтеграція бази	1 день	Вт 07.04.20	Вт 07.04.20	4	Белка Я.С.
6	Розробка інтерфейсу	6 днів	Ср 08.04.20	Ср 15.04.20	1;5	Белка Я.С.
7	Розробка сценаріїв користувачів	1 день	Ср 08.04.20	Ср 08.04.20	5	Белка Я.С.
8	Розробка карти сайту	1 день	Чт 09.04.20	Чт 09.04.20	7	Белка Я.С.
9	Розробка інтелект-карти	1 день	Пт 10.04.20	Пт 10.04.20	8	Белка Я.С.
10	Вибір стилістики	1 день	Пн 13.04.20	Пн 13.04.20	9	Белка Я.С.
11	Прототипування	2 днів	Чт 14.04.20	Ср 15.04.20	10	Белка Я.С.
12	Реалізація	22 днів	Чт 16.04.20	Пт 15.05.20	6;11	Белка Я.С.
13	Верстка шаблону та сторінок модулю адміна	6 днів	Чт 16.04.20	Чт 23.04.20	11	Белка Я.С.
14	Верстка шаблону та сторінок модулю клієнта	3 днів	Пт 24.04.20	Вт 28.04.20	13	Белка Я.С.
15	Розробка модулю адміна	10 днів	Ср 29.04.20	Вт 12.05.20	14	Белка Я.С.
16	Розробка модулю клієнта	3 днів	Ср 13.05.20	Пт 15.05.20	15	Белка Я.С.
17	Тестування	10 днів	Пн 18.05.20	Пт 29.05.20	12	Белка Я.С.
18	Написання тест-кейсів до модулю адміна	2 днів	Пн 18.05.20	Вт 19.05.20	16	Белка Я.С.
19	Написання тест-кейсів до модулю клієнта	1 день	Ср 20.05.20	Ср 20.05.20	18	Белка Я.С.
20	Проведення тестувань та оформлення баг-репортів	2 днів	Чт 21.05.20	Пт 22.05.20	19	Белка Я.С.
21	Виправлення несправностей	5 днів	Пн 25.05.20	Пт 29.05.20	20	Белка Я.С.

Рисунок Б.4 – Список робіт для побудови діаграми Ганта

Аналіз ризиків. Аналіз ризиків розпочнемо з якісної оцінки, спочатку визначимо пріоритет ризиків для подальшого аналізу, а потім їх вірогідність виникнення та вплив. Це допоможе зменшити неоднозначність і зосередити увагу на пріоритетних ризиках. Після якісної оцінки проведемо кількісну – чисельний аналіз впливу виділених ризиків на цілі проекту, де ризику присвоюється числовий рейтинг. У табл. Б.5 наведена класифікація ризиків за показниками ймовірності виникнення ризику та величиною втрат.

Для зведення негативного впливу загроз на цілі проекту до мінімуму виконаємо планування реагування на ризики. Планування враховує ризики за їх пріоритетом, впливом на бюджет, графік. Ризики оцінюються за показниками, що наведені в табл. Б.3. На основі оцінки побудовано матрицю ймовірності виникнення та впливу ризиків, що зображена на рис. Б.5.

Таблиця Б.3 – Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

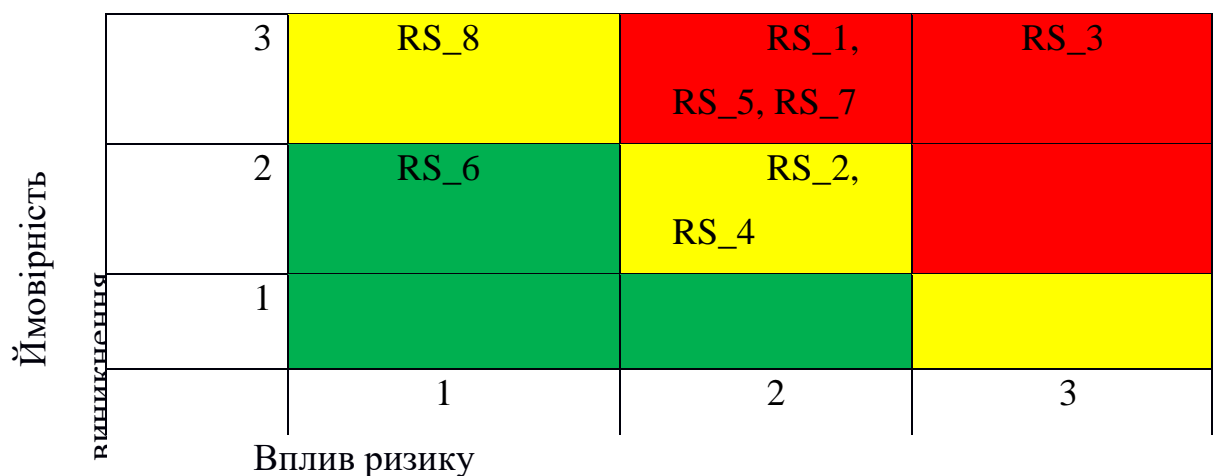


Рисунок Б.5 – Матриця ймовірності виникнення ризиків та впливу ризику

- зелений колір – прийнятні ризики;
- жовтий колір – виправданні ризики;
- червоний колір – недопустимі ризики.

На підставі отриманого значення індексу ризику класифікують за рівнем, що знаходиться в табл. Б.4.

Таблиця Б.4– Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	$1 \leq R \leq 2$	6
2	Виправдані	$3 \leq R \leq 4$	2, 4, 8
3	Недопустимі	$6 \leq R \leq 9$	1, 3, 5, 7

Таблиця Б.5 – Оцінка ймовірності виникнення, величини витрат та індексу ризику

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Нечітке технічне завдання	Середня	Високий	6	Перед початком виконання обговорити усі незрозумілості. Співпрацювати з замовником на усіх етапах роботи.	Попередження	При виявленні невідповідностей потрібно негайно оцінити межі впливу похибки та виправити її.
RS_2	Відкритий	Недостатня кваліфікація розробника	Середня	Середній	4	Використання наявних онлайн-ресурсів для підвищення кваліфікації.	Пом'якшення	Використовувати наявні відео-тutorіали та тематичну літературу.
RS_3	Відкритий	Неоптимальний розподіл часу	Висока	Високий	9	Створити структуру робіт і виділяти час виконання на найменші пакети, щоб зменшити можливість похибок розподілу часу. Дотримуватися термінів, визначених у календарному плані.	Пом'якшення	Оптимізувати роботу згідно створеному плану. Розглянути можливість внесення правок до термінів виконання.
RS_4	Відкритий	Часте внесення змін у ТЗ	Середня	Середній	4	Перед початком виконання узгодити з замовником можливість внесення змін та окреслити їх масштаб .	Попередження	Діяти за потребою, розглядаючи можливість та варіанти внесення змін.

Продовження таблиці Б.5

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_5	Відкритий	Вибір недоцільної технології розробки	Середня	Високий	6	Проаналізувати наявні фреймворки та технології виконання проекту.	Пом'якшення	Приділити час на ознайомлення з технологією, мінімізацію її слабких сторін.
RS_6	Відкритий	Недооцінений масштаб проекту	Низька	Середній	3	На етапі планування провести аналіз проекту. Визначити структуру робіт.	Пом'якшення	Перебудова стратегії реалізації проекту.
RS_7	Відкритий	Не правильно створена архітектура проекту	Середній	Високий	6	Детально продумати та створити діаграми структури бек-енду та фронт-енду на етапі проектування.	Пом'якшення	Моніторити процес реалізації для швидко реагування на несумісності.
RS_8	Відкритий	Відсутність резервних копій	Низька	Високий	3	Вибір хостинг провайдеру, який управляє здійснення резервних копій.	Попередження	Налаштувати збереження даних на стороні провайдера.

ДОДАТОК Г

Скрипти створення бази даних

```
CREATE TABLE `agreement` (  
  `agreement_id` int(11) NOT NULL,  
  `user_id` int(11) NOT NULL,  
  `vendor_id` int(11) NOT NULL,  
  `client_id` int(11) NOT NULL,  
  `sign_date` datetime NOT NULL DEFAULT current_timestamp(),  
  `comment` text DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `agreement_documents` (  
  `id` int(11) NOT NULL,  
  `name` text NOT NULL,  
  `agreement_id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `authorization` (  
  `id` int(11) NOT NULL,  
  `username` varchar(50) DEFAULT NULL,  
  `password` text DEFAULT NULL,  
  `authKey` text DEFAULT NULL,  
  `role` int(11) NOT NULL COMMENT '0-mngr, 1-realtor, 2-client'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `client` (  
  `client_id` int(11) NOT NULL,  
  `first_name` varchar(30) NOT NULL,  
  `last_name` varchar(30) NOT NULL,  
  `patronymic` varchar(30) NOT NULL,  
  `birthday` date NOT NULL,  
  `phonenum` varchar(20) NOT NULL,  
  `email` varchar(50) NOT NULL,  
  `authorization_id` int(11) DEFAULT NULL,  
  `user_id` int(11) DEFAULT NULL,  
  `image` varchar(60) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `documents` (  
  `id` int(11) NOT NULL,  
  `name` varchar(100) NOT NULL,  
  `object_id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `flat_properties` (  
  `id` int(11) NOT NULL,  
  `name` varchar(100) NOT NULL,  
  `object_id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```

`property_id` int(11) NOT NULL,
`house_floors` int(11) NOT NULL,
`playground` tinyint(1) NOT NULL,
`parking` tinyint(1) NOT NULL,
`flat_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `house_properties` (
  `property_id` int(11) NOT NULL,
  `yard_area` float NOT NULL,
  `walls_height` float NOT NULL,
  `fundament_depth` float NOT NULL,
  `gas` tinyint(1) NOT NULL,
  `garage` tinyint(1) NOT NULL,
  `house_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `images` (
  `id` int(11) NOT NULL,
  `name` varchar(100) NOT NULL,
  `object_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `meeting` (
  `meet_id` int(11) NOT NULL,
  `meet_dt` datetime NOT NULL,
  `staff_id` int(11) NOT NULL,
  `client_id` int(11) NOT NULL,
  `meet_location` text NOT NULL,
  `description` text NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=cp1251;

```

```

CREATE TABLE `objects` (
  `object_id` int(11) NOT NULL,
  `price` int(11) NOT NULL,
  `address` text DEFAULT NULL,
  `vendor_id` int(11) DEFAULT NULL,
  `object_type` int(11) NOT NULL COMMENT '0-flat, 1-house',
  `image` varchar(100) DEFAULT NULL,
  `floor` int(11) NOT NULL,
  `description` text DEFAULT NULL,
  `heating` int(11) NOT NULL,
  `warm_water` int(11) NOT NULL,
  `internet` tinyint(1) NOT NULL,
  `adding_date` datetime NOT NULL DEFAULT current_timestamp(),
  `repair_type` int(11) NOT NULL,
  `material` int(11) NOT NULL,
  `room_qty` int(11) NOT NULL,
  `kitchen_area` float NOT NULL,
  `living_area` float NOT NULL,
  `area` float NOT NULL
)

```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `offer` (
  `id` int(11) NOT NULL,
  `client_id` int(11) NOT NULL,
  `object_id` int(11) NOT NULL,
  `add_date` datetime NOT NULL DEFAULT current_timestamp(),
  `status_` int(11) NOT NULL DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `staff` (
  `user_id` int(11) NOT NULL,
  `first_name` varchar(30) NOT NULL,
  `last_name` varchar(30) NOT NULL,
  `patronymic` varchar(30) NOT NULL,
  `birthday` date DEFAULT NULL,
  `phonenum` varchar(20) NOT NULL,
  `email` varchar(50) NOT NULL,
  `image` varchar(60) DEFAULT NULL,
  `authorization_id` int(11) DEFAULT NULL,
  `mngr_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `task` (
  `task_id` int(11) NOT NULL,
  `title` char(50) NOT NULL,
  `start_time` datetime NOT NULL,
  `end_time` datetime NOT NULL,
  `description` text NOT NULL,
  `assigned_to` int(11) NOT NULL,
  `appointed_by` int(11) NOT NULL,
  `status_` tinyint(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `vendor` (
  `vendor_id` int(11) NOT NULL,
  `first_name` varchar(30) NOT NULL,
  `last_name` varchar(30) NOT NULL,
  `patronymic` varchar(30) NOT NULL,
  `birthday` date NOT NULL,
  `phonenum` varchar(20) NOT NULL,
  `email` varchar(50) NOT NULL,
  `image` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `wish` (
  `wish_id` int(11) NOT NULL,
  `price_min` double NOT NULL,
  `price_max` double NOT NULL,
  `address` text NOT NULL,
  `object_type` int(11) NOT NULL COMMENT '0-flat, 1-house',
  `min_floor` int(11) NOT NULL,
```



```

`max_floor` int(11) NOT NULL,
`description` text DEFAULT NULL,
`heating` int(11) NOT NULL,
`warm_water` int(11) NOT NULL,
`adding_date` datetime DEFAULT current_timestamp(),
`repair_type` int(11) NOT NULL,
`material` int(11) NOT NULL,
`room_qty_min` int(11) NOT NULL,
`room_qty_max` int(11) NOT NULL,
`area_min` float NOT NULL,
`area_max` float NOT NULL,
`internet` tinyint(4) NOT NULL,
`playground` tinyint(1) NOT NULL DEFAULT -1,
`parking` tinyint(1) NOT NULL DEFAULT -1,
`gas` tinyint(1) NOT NULL DEFAULT -1,
`garage` tinyint(1) NOT NULL DEFAULT -1,
`yard_area_min` double NOT NULL DEFAULT 0,
`yard_area_max` double NOT NULL DEFAULT 0,
`client_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `wish_change_log` (
  `id` int(11) NOT NULL,
  `date` datetime NOT NULL DEFAULT current_timestamp(),
  `client_id` int(11) DEFAULT NULL,
  `user_id` int(11) DEFAULT NULL,
  `if_read` tinyint(11) NOT NULL DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

ALTER TABLE `agreement` ADD PRIMARY KEY (`agreement_id`);
ALTER TABLE `agreement_documents` ADD PRIMARY KEY (`id`);
ALTER TABLE `authorization` ADD PRIMARY KEY (`id`);
ALTER TABLE `client` ADD PRIMARY KEY (`client_id`);
ALTER TABLE `documents` ADD PRIMARY KEY (`id`);
ALTER TABLE `flat_properties` ADD PRIMARY KEY (`property_id`);
ALTER TABLE `house_properties` ADD PRIMARY KEY (`property_id`);
ALTER TABLE `images` ADD PRIMARY KEY (`id`);
ALTER TABLE `meeting` ADD PRIMARY KEY (`meet_id`);
ALTER TABLE `objects` ADD PRIMARY KEY (`object_id`);
ALTER TABLE `offer` ADD PRIMARY KEY (`id`);
ALTER TABLE `staff` ADD PRIMARY KEY (`user_id`);
ALTER TABLE `task` ADD PRIMARY KEY (`task_id`);
ALTER TABLE `vendor` ADD PRIMARY KEY (`vendor_id`);
ALTER TABLE `wish` ADD PRIMARY KEY (`wish_id`);
ALTER TABLE `wish_change_log` ADD PRIMARY KEY (`id`);
ALTER TABLE `agreement`
  MODIFY `agreement_id` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=0;
ALTER TABLE `agreement_documents`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;
ALTER TABLE `authorization`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;

```

```
ALTER TABLE `client`
  MODIFY `client_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;
ALTER TABLE `documents`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;
ALTER TABLE `flat_properties`
  MODIFY `property_id` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=0;

ALTER TABLE `house_properties`
  MODIFY `property_id` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=0;
ALTER TABLE `images`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;
ALTER TABLE `meeting`
  MODIFY `meet_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;
ALTER TABLE `objects`
  MODIFY `object_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;
ALTER TABLE `offer`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;
ALTER TABLE `staff`
  MODIFY `user_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;
ALTER TABLE `task`
  MODIFY `task_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;
ALTER TABLE `vendor`
  MODIFY `vendor_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;
ALTER TABLE `wish`
  MODIFY `wish_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;
ALTER TABLE `wish_change_log`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=0;
COMMIT;
```

ДОДАТОК Д

Тестування системи

1. Авторизація користувачів у системі

Таблиця Д.1 – Тестування авторизації

ІД	Назва	Вхід/дії	Очікуваний результат	Статус (Pass/Fail)
1)	Позитивна перевірка авторизації співробітника	1. Відкрити сторінку авторизації http://i.smartbuild.ml/login.html . 2. Ввести логін співробітника. 3. Ввести пароль. 4. Натиснути кнопку Увійти.	Користувач успішно увійшов до модуля admin. На екрані з'явилася головна сторінка.	Pass
2)	Введення неіснуючого логіну під час входу.	1. Відкрити сторінку авторизації http://i.smartbuild.ml/login.html . 2. Ввести неіснуючий логін. 3. Ввести пароль. 4. Натиснути кнопку Увійти.	Сповіщення системи про те, що логін чи пароль введені неправильно.	Pass
3)	Пусте введення логіну користувача під час входу	1. Відкрити сторінку авторизації http://i.smartbuild.ml/login.html . 2. Не вводити логін. 3. Ввести пароль. 4. Натиснути кнопку Увійти.	Сповіщення про те, що поле обов'язкове для заповнення.	Pass
4)	Неправильне введення паролю користувача під час входу	1. Відкрити сторінку авторизації http://i.smartbuild.ml/login.html . 2. Ввести валідний логін. 3. Ввести неправильний пароль. 4. Натиснути кнопку Увійти.	Сповіщення системи про те, що логін чи пароль введені неправильно.	Pass
5)	Пусте введення паролю під час авторизації	1. Відкрити сторінку авторизації http://i.smartbuild.ml/login.html . 2. Ввести валідний логін. 3. Не вводити пароль. 4. Натиснути кнопку Увійти.	Сповіщення про те, що поле обов'язкове для заповнення.	Pass
6)	Позитивна перевірка авторизації клієнта	1. Відкрити сторінку авторизації http://i.smartbuild.ml/login.html . 2. Ввести логін клієнта. 3. Ввести пароль. 4. Натиснути кнопку Увійти.	Користувач успішно увійшов до модуля client. На екрані з'явилася головна сторінка.	Pass

2. Перевірка функціональності сторінки «Співробітники» модуля admin

Для виконання тестування потрібно повторити пункт 1 з таблиці Д.1 для входу у систему з правами менеджера.

Таблиця Д.2 – Тестування функціоналу сторінки співробітників

ID	Назва	Вхід/дії	Очікуваний результат	Статус (Pass/Fail)
1)	Позитивна перевірка додавання нового співробітника	1) Відкрити сторінку «Співробітники» 2) Натиснути кнопку «Додати нового» 3) Увести усі надані поля 4) Натиснути кнопку додати	Новий запис успішно доданий до бази даних.	Pass
2)	Введення неправильного паролю при реєстрації	1) Відкрити сторінку «Співробітники» 2) Натиснути кнопку «Додати нового» 3) Увести усі надані поля 4) У поле «Пароль» ввести значення менше за 8 символів. 5) Натиснути кнопку додати	Повідомлення «Пароль повинен містити не менше 8 символів»	Pass
3)	Введення неправильного імейлу при реєстрації	1) Відкрити сторінку «Співробітники» 2) Натиснути кнопку «Додати нового» 3) Увести усі надані поля 4) У поле «Логін» ввести значення менше за 2 символів. 5) Натиснути кнопку додати	Повідомлення «Логін повинен містити не менше 2 символів»	Pass
4)	Перевірка роботи пошуку у таблиці	1) Відкрити сторінку «Співробітники» 2) Сфокусувати обране поле та увести запит для пошуку в поля «Прізвище», «Ім'я» та «По-батькові» 3) Убрати фокус	У таблиці відобразилися лише ті записи, які відповідають запитові.	Pass

Продовження таблиці Д.2 – Тестування функціоналу сторінки співробітників

5)	Перевірка редагування запису	<ol style="list-style-type: none"> 1) Відкрити сторінку «Співробітники» 2) Біля обраного рядка запису натиснути кнопку «Редагувати» 3) Ввести зміни до запису 4) Натиснути кнопку «Змінити» 	Внесені зміни відобразились у відповідних полях.	Pass
6)	Перевірка видалення запису	<ol style="list-style-type: none"> 1) Відкрити сторінку «Співробітники» 2) Біля обраного рядка запису натиснути кнопку «Видалити» 3) Підтвердити видалення у вікні, що з'явилося. 	Обраний запис видалився з таблиці.	Pass

3. Перевірка функціональності сторінки «Клієнти» модуля admin

Таблиця Д.3 – Тестування функціоналу сторінки клієнтів

ID	Назва	Вхід/дії	Очікуваний результат	Статус (Pass/Fail)
1)	Позитивна перевірка додавання нового клієнта	<ol style="list-style-type: none"> 1) Відкрити сторінку «Клієнти» 2) Натиснути кнопку «Додати клієнта» 3) Увести усі надані поля 4) Натиснути кнопку додати. 	Новий запис успішно доданий до бази даних.	Pass
2)	Введення неправильного паролю при реєстрації	<ol style="list-style-type: none"> 1) Відкрити сторінку «Клієнти» 2) Натиснути кнопку «Додати клієнта» 3) Увести усі надані поля 4) У поле «Пароль» ввести значення менше за 8 символів. 5) Натиснути кнопку додати 	Повідомлення «Пароль повинен містити не менше 8 символів»	Pass

Продовження таблиці Д.3 – Тестування функціоналу сторінки клієнтів

3)	Введення не правильного імейлу при реєстрації	<ol style="list-style-type: none"> 1) Відкрити сторінку «Клієнти» 2) Натиснути кнопку «Додати клієнта» 3) Увести усі надані поля 4) У поле «Логін» ввести значення менше за 2 символів. 5) Натиснути кнопку додати 	Повідомлення «Логін повинен містити не менше 2 символів»	Pass
4)	Перевірка роботи пошуку у таблиці	<ol style="list-style-type: none"> 1) Відкрити сторінку «Клієнти» 2) Сфокусувати обране поле та увести запит для пошуку в поля «Прізвище», «Ім'я» та «По-батькові» 3) Убрати фокус 	У таблиці відобразилися лише ті записи, які відповідають запитові.	Pass
5)	Перевірка роботи фільтру за клієнтами	<ol style="list-style-type: none"> 1) Відкрити сторінку «Клієнти» 2) У спадному списку обрати опцію «Мої клієнти» 	У таблиці відобразилися лише ті записи, які відповідають запитові.	Pass
6)	Перевірка редагування запису	<ol style="list-style-type: none"> 1) Відкрити сторінку «Клієнти» 2) Біля обраного рядка запису натиснути кнопку «Редагувати» 3) Ввести зміни до запису 4) Натиснути кнопку «Змінити» 	Внесені зміни відобразились у відповідних полях.	Pass
7)	Перевірка видалення запису	<ol style="list-style-type: none"> 1) Відкрити сторінку «Співробітники» 2) Біля обраного рядка запису натиснути кнопку «Видалити» 3) Підтвердити видалення у вікні, що з'явилося. 	Обраний запис видалився з таблиці.	Pass
8)	Перевірка роботи пошуку за побажанням	<ol style="list-style-type: none"> 1) Відкрити сторінку «Клієнти» 2) Біля обраного рядка запису натиснути кнопку «Пошук» 3) Підтвердити видалення у вікні. 4) Ввести зміни до параметрів пошуку. 5) Натиснути кнопку «Пошук» 	У залежності від змінених параметрів, змінюються результати пошуку.	Pass
9)	Перевірка перегляду результатів пошуку	<ol style="list-style-type: none"> 1) Повторити пункт 8 у таблиці Д.3 2) Обрати опцію «Переглянути» 	Відкрилося нова вкладка з описом об'єкта.	Pass
10)	Перевірка надсилання пропозиції клієнтові	<ol style="list-style-type: none"> 1) Повторити пункт 8 у таблиці Д.3 2) Обрати опцію «Надіслати» 	Надіслана пропозиція з'явилася у кабінеті відповідного клієнта.	Pass

4. Перевірка функціональності сторінки «Власники» модуля admin

Таблиця Д.4 – Тестування функціоналу сторінки власників

ІД	Назва	Вхід/дії	Очікуваний результат	Статус (Pass/Fail)
1)	Позитивна перевірка додавання нового власника	<ol style="list-style-type: none"> 1) Відкрити сторінку «Власники» 2) Натиснути кнопку «Додати нового» 3) Увести усі надані поля 4) Натиснути кнопку додати 	Новий запис успішно доданий до бази даних.	Pass
2)	Введення неправильного паролю при реєстрації	<ol style="list-style-type: none"> 1) Відкрити сторінку «Власники» 2) Натиснути кнопку «Додати нового» 3) Увести усі надані поля 4) У поле «Пароль» ввести значення менше за 8 символів. 5) Натиснути кнопку додати 	Повідомлення «Пароль повинен містити не менше 8 символів»	Pass
3)	Введення неправильного імейлу при реєстрації	<ol style="list-style-type: none"> 1) Відкрити сторінку «Власники» 2) Натиснути кнопку «Додати нового» 3) Увести усі надані поля 4) У поле «Логін» ввести значення менше за 2 символів. 5) Натиснути кнопку додати 	Повідомлення «Логін повинен містити не менше 2 символів»	Pass
4)	Перевірка роботи пошуку у таблиці	<ol style="list-style-type: none"> 1) Відкрити сторінку «Власники» 2) Сфокусувати обране поле та увести запит для пошуку в поля «Прізвище», «Ім'я» та «По-батькові» 3) Убрати фокус 	У таблиці відобразилися лише ті записи, які відповідають запиту.	Pass
5)	Перевірка редагування запису	<ol style="list-style-type: none"> 1) Відкрити сторінку «Власники» 2) Біля обраного рядка запису натиснути кнопку «Редагувати» 3) Ввести зміни до запису 4) Натиснути кнопку «Змінити» 	Внесені зміни відобразились у відповідних полях.	Pass

Продовження таблиці Д.4 – Тестування функціоналу сторінки власників

6)	Перевірка видалення запису	<ol style="list-style-type: none"> 1) Відкрити сторінку «Власники» 2) Біля обраного рядка запису натиснути кнопку «Видалити» 3) Підтвердити видалення у вікні, що з'явилося. 	Обраний запис видалився з таблиці.	Pass
7)	Перевірка додавання квартири	<ol style="list-style-type: none"> 1) Відкрити сторінку «Власники» 2) Біля обраного рядка запису натиснути кнопку «Квартира» 3) На сторінці увести усі дані. 4) Натиснути кнопку «Додати». 	Об'єкт успішно створений.	Pass
8)	Перевірка додавання будинку	<ol style="list-style-type: none"> 1) Відкрити сторінку «Власники» 2) Біля обраного рядка запису натиснути кнопку «Будинок» 3) На сторінці увести усі дані. 4) Натиснути кнопку «Додати». 	Об'єкт успішно створений.	Pass

5. Перевірка функціональності сторінки «Об'єкти» модуля admin

Таблиця Д.5 – Тестування функціоналу сторінки об'єктів

ID	Назва	Вхід/дії	Очікуваний результат	Статус (Pass/Fail)
1)	Перевірка редагування запису	<ol style="list-style-type: none"> 1) Відкрити сторінку «Об'єкти». 2) Біля обраного рядка запису натиснути кнопку «Редагувати» 3) Ввести зміни до запису 4) Натиснути кнопку «Змінити» 	Внесені зміни відобразились у відповідних полях.	Pass

Продовження таблиці Д.5 – Тестування функціоналу сторінки об'єктів

2)	Перевірка завантаження файлів	<ol style="list-style-type: none"> 1) Відкрити сторінку «Об'єкти». 2) Біля обраного рядка запису натиснути кнопку «Редагувати» 3) Натиснути кнопку «Обрати файл» 4) Обрати декілька файлів у форматі pdf або jpg 5) Натиснути кнопку «Завантажити» 	Файли завантажені та відображаються у відповідній частині сторінки.	Pass
3)	Негативна перевірка завантаження файлів	<ol style="list-style-type: none"> 1) Відкрити сторінку «Об'єкти». 2) Біля обраного рядка запису натиснути кнопку «Редагувати» 3) Натиснути кнопку «Обрати файл» 4) Обрати декілька файлів не у форматі .pdf або .jpg 1) Натиснути кнопку «Завантажити» 	З'являється попередження про те, що файли для завантаження повинні бути у форматі .pdf або .jpg.	Pass
4)	Перевірка видалення запису	<ol style="list-style-type: none"> 1) Відкрити сторінку «Об'єкти» 2) Біля обраного рядка запису натиснути кнопку «Видалити» 3) Підтвердити видалення у вікні, що з'явилося. 	Обраний запис видалився з таблиці.	Pass
5)	Перевірка перегляду запису	<ol style="list-style-type: none"> 1) Відкрити сторінку «Об'єкти» 2) Біля обраного рядка запису натиснути кнопку «Переглянути» 3) Навести на фото. 4) Обрати інше фото з мініатюр та натиснути на нього. 	При наведенні фото збільшує розмір. При виборі іншої мініатюри, воно замінює головне у перегляді.	Pass

6. Перевірка функціональності сторінки «Календар» модуля admin

Таблиця Д.6 – Тестування функціоналу сторінки календаря

ІД	Назва	Вхід/дії	Очікуваний результат	Статус (Pass/Fail)
1)	Перевірка додавання задачі	1) Відкрити сторінку «Задачі» 2) Натиснути кнопку з зображенням знаку додавання на зеленому тлі. 3) У модальному вікні ввести всі дані. 4) Натиснути кнопку «Додати задачу».	Задача створена і відображається у списку роботи на день.	Pass
2)	Перевірка призначення зустрічі	1) Відкрити сторінку «Задачі» 2) Натиснути кнопку з зображенням знаку додавання на синьому тлі. 3) У модальному вікні ввести всі дані. 4) Натиснути кнопку «Додати зустріч».	Зустріч створена і відображається у списку роботи на день.	Pass
3)	Перевірка зміни статусу виконання задачі	1) Відкрити сторінку «Задачі» 2) Натиснути поле обраного дня. 3) У модальному вікні обрати задачу та натиснути кнопку з зображенням галочки для зміни статусу.	Зображення галочки змінило свій колір з сірого на зелений.	Pass
4)	Перевірка зміни статусу виконання зустрічі	1) Відкрити сторінку «Задачі» 2) Натиснути поле обраного дня. 3) У модальному вікні обрати зустріч та натиснути кнопку з зображенням галочки для зміни статусу.	Зображення галочки змінило свій колір з сірого на зелений.	Pass
5)	Перевірка навігації календарем	1) Відкрити сторінку «Задачі» 2) Натиснути перемикачі з назвами місяців.	Місяць змінюється відповідно обраній кнопці.	Pass

7. Перевірка функціональності сторінки «Задачі» модуля admin

Таблиця Д.7 – Тестування функціоналу сторінки задачі

ID	Назва	Вхід/дії	Очікуваний результат	Статус (Pass/Fail)
1)	Позитивна перевірка додавання задачі	1) Відкрити сторінку «Задачі» 2) Натиснути кнопку «Додати задачу» 3) Увести усі надані поля 4) Натиснути кнопку «Додати задачу»	Новий запис успішно доданий до бази даних.	Pass
2)	Перевірка роботи пошуку у таблиці за текстовими полями	1) Відкрити сторінку «Задачі» 2) Обрати поля та увести запит для пошуку в поля «Назва» та «Виконавець» 3) Натиснути кнопку «Пошук»	У таблиці відобразилися лише ті записи, які відповідають запитові.	Pass
3)	Перевірка роботи пошуку у таблиці за датою	1) Відкрити сторінку «Задачі» 2) Обрати поля та змінити інтервали початку та закінчення строку виконання задачі 3) Натиснути кнопку «Пошук»	У таблиці відобразилися лише ті записи, які відповідають часовому проміжку.	Pass
4)	Перевірка редагування запису	1) Відкрити сторінку «Клієнти» 2) Біля обраного рядка запису натиснути кнопку «Редагувати» 3) Ввести зміни до запису Натиснути кнопку «Змінити»	Внесені зміни відобразились у відповідних полях.	Pass
5)	Перевірка видалення запису	1) Відкрити сторінку «Співробітники» 2) Біля обраного рядка запису натиснути кнопку «Видалити» 3) Підтвердити видалення у вікні, що з'явилося.	Обраний запис видалився з таблиці.	Pass

8. Перевірка функціональності сторінки «Договори» модуля admin

Таблиця Д.8 – Тестування функціоналу сторінки договори

ID	Назва	Вхід/дії	Очікуваний результат	Статус (Pass/Fail)
1)	Позитивна перевірка додавання договору	<ol style="list-style-type: none"> 1) Відкрити сторінку редагування об'єкта для продажу 2) Натиснути кнопку «Продано» 3) Увести усі надані поля 4) Натиснути кнопку «Додати» 	Новий запис успішно доданий до бази даних.	Pass
2)	Перевірка роботи пошуку у таблиці	<ol style="list-style-type: none"> 1) Відкрити сторінку «Договори» 2) Обрати поля та увести запит для пошуку в поля «Ріелтор», «Продавець» та «Покупець» 3) Натиснути кнопку «Пошук» 	У таблиці відобразилися лише ті записи, які відповідають запитові.	Pass
3)	Перевірка редагування запису	<ol style="list-style-type: none"> 1) Відкрити сторінку «Договори» 2) Біля обраного рядка запису натиснути кнопку «Редагувати» 3) Ввести зміни до запису 4) Натиснути кнопку «Змінити» 	Внесені зміни відобразились у відповідних полях.	Pass

9. Перевірка функціональності модуля client

Таблиця Д.9 – Тестування функціоналу модуля client

ID	Назва	Вхід/дії	Очікуваний результат	Статус (Pass/Fail)
1)	Перевірка зміни статусу зустрічі	<ol style="list-style-type: none"> 1) Відкрити сторінку «Головна» 2) Вибрати зі спадного списку статус зустрічі 3) Натиснути кнопку змінити 	Статус змінився у кабінеті клієнта та відповідного ріелтора.	Pass

Продовження таблиці Д.9 – Тестування функціоналу модуля client

2)	Перевірка зміни статусу пропозиції	<ol style="list-style-type: none"> 1) Відкрити сторінку «Головна» 2) Вибрати зі спадного списку статус пропозиції 3) Натиснути кнопку змінити 	Статус змінився у кабінеті клієнта та відповідного ріелтора.	Pass
3)	Перевірка перегляду пропозиції	<ol style="list-style-type: none"> 1) Відкрити сторінку «Головна» 2) Вибрати зі списку пропозицій одну та натиснути кнопку «Переглянути» 	У новій вкладці відкрилася сторінка з інформацією про пропозицію.	Pass
4)	Перевірка редагування побажання	<ol style="list-style-type: none"> 1) Відкрити сторінку «Побажання» 2) Змінити поля 3) Натиснути кнопку «Змінити» 	Внесені зміни збереглися. Ріелтор отримав у кабінеті сповіщення.	Pass