

**Київський національний торговельно-економічний
університет**

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Інформаційно-пошукова WEB-система підбору виконавців
та замовників послуг»**

Студента 2 курсу, 3м групи,
спеціальності
122 «Комп'ютерні науки»

Павлючика Дениса
Андрійовича

*підпис
студента*

Науковий керівник
доктор фізико-математичних наук,
професор

Пурський Олег
Іванович

*підпис
керівника*

Гарант освітньої програми
доктор фізико-математичних наук,
професор

Пурський Олег
Іванович

*підпис
керівника*

Київ 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра комп'ютерних наук та інформаційних систем

Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри _____

Затверджую

Пурський О.І.

«20» грудня 2020р.

**Завдання
на випускню кваліфікаційну роботу студенту**

Павлючику Денису Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи

«Інформаційно-пошукова WEB-система підбору виконавців та замовників послуг.»

Затверджена наказом ректора від «02» грудня 2020 р. № 4110

2. Строк здачі студентом закінченої роботи 25 листопада 2021 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: розробка інформаційно-пошукової WEB-системи підбору виконавців та замовників послуг

Об'єкт дослідження: процес проектування інформаційно-пошукової web-системи підбору виконавців та замовників послуг

Предмет дослідження: засоби створення інформаційно-пошукової web-системи підбору виконавців та замовників послуг

4. Перелік графічного матеріалу схема структурна варіантів використань, схема структурна класів програмного забезпечення, схема структурна компонентів програмного забезпечення, схема бази даних, схема структурна послідовності варіантів використань, креслення вигляду екранних форм

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Пурський О.І.	5.12.2020р.	5.12.2020р.
2	Пурський О.І.	5.12.2020р.	5.12.2020р.
3	Пурський О.І.	5.12.2020р.	5.12.2020р.

6. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. Загальні положення

1.1. Опис предметного середовища

1.2. Огляд наявних аналогів

1.3. Постановка задачі

Висновки до Розділу 1

РОЗДІЛ 2. Інформаційне та математичне забезпечення.

2.1. Вхідні та вихідні дані

2.2 Опис структури бази даних.

2.3. Змістовна та математична постановки задачі

2.4. Обґрунтування та опис методу розв'язання

Висновки до Розділу 2

РОЗДІЛ 3. Програмне та технічне забезпечення

3.1 Засоби розробки

3.2 Вимоги до технічного забезпечення

3.3 Архітектура програмного забезпечення

3.4 Керівництво користувача

3.5 Випробування програмного продукту

Висновки до Розділу 3

ВИСНОВОК

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	<i>01.11.2020</i>	<i>01.11.2020</i>
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	<i>05.12.2020</i>	<i>05.12.2020</i>
3	<i>Вступ</i>	<i>01.06.2021</i>	
4	<i>РОЗДІЛ 1. Загальні положення</i>	<i>25.06.2021</i>	
5	<i>РОЗДІЛ 2. Інформаційне та математичне забезпечення</i>	<i>02.09.2021</i>	
6	<i>РОЗДІЛ 3. Програмне та технічне забезпечення</i>	<i>21.10.2021</i>	
7	<i>Висновки</i>	<i>02.11.2021</i>	
8	<i>Підготовка статті у збірник наукових статей магістрів</i>	<i>18.06.2021</i>	
9	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	<i>05.11.2021</i>	
10	<i>Попередній захист випускної кваліфікаційної роботи</i>	<i>01.12.2021</i>	
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	<i>03.12.2021</i>	
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	<i>06.12.2021</i>	
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	<i>За розкладом роботи ЕК</i>	

8. Дата видачі завдання «5» грудня 2020 р.

Керівник випускного кваліфікаційного проекту

Пурський О.І.

(прізвище, ініціали, підпис)

Гарант освітньої програми

Пурський О.І.

(прізвище, ініціали, підпис)

Анотація

У випускній кваліфікаційній роботі розглянуто розробку інформаційно-пошукової WEB-системи підбору виконавців та замовників послуг(на прикладі будівництва). Основною ціллю розробки є створення інформаційної системи, що дозволить швидко знайти підходящого спеціаліста для виконання ремонтних робіт в квартирі, офісі, гаражі, будинку, тощо, з однієї сторони, та дозволить спеціалістам швидко знайти собі додатковий заробіток з іншої.

Ключові слова: інформаційна система, будівництво, спеціалісти, ремонт, замовники, робота, пошук роботи.

Annotation

The graduation qualification work is devoted to the development of an information retrieval WEB-system for selection of contractors and customers for services (on the example of construction). The main goal of the development is to create an information system that will allow to find quickly a suitable specialist to perform repairs in an apartment, office, garage, house, etc., on the one hand, and allow professionals to quickly find additional income on the other.

Keywords: information system, construction, specialists, repair, customers, work, job search.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ	10
1.1. Опис предметного середовища	10
1.2. Огляд наявних аналогів	19
1.3. Постановка задачі	19
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	21
2.1. Вхідні та вихідні дані	21
2.2. Опис структури бази даних	21
2.3. Змістовна та математична постановки задачі	23
2.4 Обґрунтування та опис методу розв’язання	24
РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	26
3.1. Засоби розробки	26
3.2. Вимоги до технічного забезпечення	28
3.3. Архітектура програмного забезпечення	28
3.4. Програмна реалізація і технологія використання	35
3.5. Випробування програмного продукту	43
ВИСНОВОК	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	46

ВСТУП

Актуальність теми. На сьогоднішній день людям важко знайти досвідчених будівельників, ремонтників, майстрів своєї справи, котрі зможуть зробити їм якісний ремонт. Це насправді дуже велика проблема, адже якщо ремонт буде зроблений погано, його потрібно буде перероблювати, а це понесе за собою великі затрати.

З другого боку, кваліфікованим фахівцям важко знайти чесних та відповідальних замовників, так як багато хто зараз вимагає спочатку зробити роботу, а потім по факту проводить оплату виконаних послуг. На останньому етапі багато людей відмовляються платити, адже знаходять якісь «видумані дефекти роботи», і видають це як причину того що не збираються платити.

На фоні всіх цих проблем виникає потреба в системі, що дозволить швидко знайти відповідальних, працьовитих, чесних професіоналів своєї справи з однієї сторони, та адекватних, чесних, порядних замовників з іншої.

Саме для цього і створена система пошуку виконавців або замовників надання послуг(на прикладі послуг будівництва).

Мета роботи: розробка інформаційно-пошукової WEB-системи підбору виконавців та замовників послуг

Об'єкт дослідження: процес проектування інформаційно-пшукової web-системи підбору виконавців та замовників послуг.

Предмет дослідження: засоби створення інформаційно-пшукової web-системи підбору виконавців та замовників послуг.

Інформаційна база дослідження: наукові публікації вітчизняних та зарубіжних фахівців, електронні ресурси Інтернет, навчальні посібники, видані Міністерством освіти та науки України.

Методи дослідження: загальнонауковий аналітичний метод, системний метод, методи теорії бази даних, метод алгоритмічного програмування.

Практичне значення одержаних результатів. Розроблену систему створення замовлень з будівництва можна використати як власний бізнес-проект із залученням інвесторів та подальшому використанні його для отримання прибутку.

Наукова новизна одержаних результатів полягає в ґрунтовному аналізі і дослідженні типів інформаційно пошукових Web-систем і удосконаленню механізмів їх ефективного використання у повсякденному житті.

Публікація. Результати дослідження опубліковано у збірнику наукових статей студентів, які здобувають освітній ступінь магістра за спеціалізацією «Комп'ютерні науки» КНТЕУ на тему: «Інформаційно-пошукова WEB-система підбору виконавців та замовників послуг», 2021 р.

У відповідності до поставлених мети та конкретних завдань дослідження, визначено структуру роботи. Вона складається зі вступу, трьох розділів, висновків та списку використаної літератури.

РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1. Опис предметного середовища

Діяльність людей, бізнесу та підприємств нерозривно пов'язана з виконанням різного роду задач та доручень. Але не завжди є час для того, щоб встигнути зробити все та вчасно. Тоді виникає потреба в пошуку виконавців, котрі могли б допомогти. Саме для цього мною і була створена інформаційна система пошуку виконавців або замовників надання послуг(в моєму випадку на прикладі будівництва).

**ПРОФЕСІЇ, ЯКІ ПОТРЕБУЮТЬСЯ НА РИНКУ ПРАЦІ М. КИЄВА
СТАНОМ НА 01.04.2019 РОКУ**

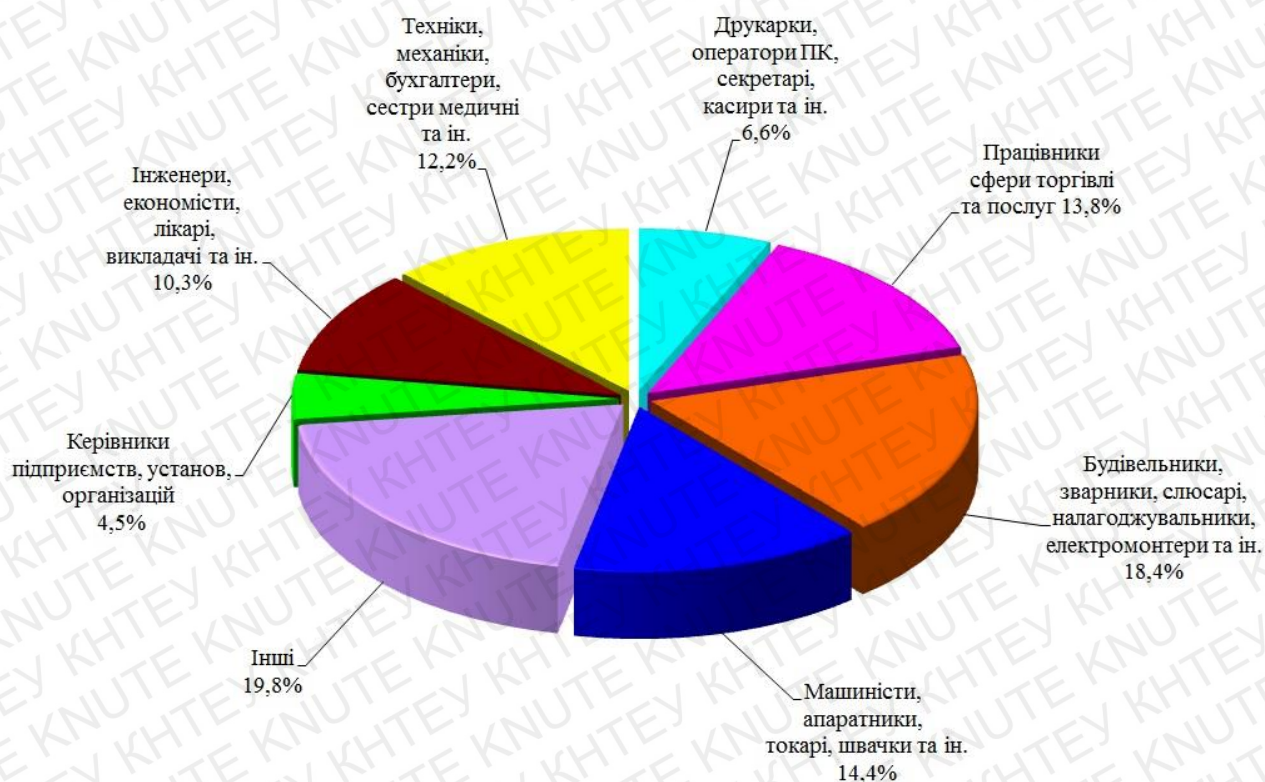


Рисунок 1.1 – Діаграма професій, які потребуються на ринку праці м. Києва

Як ми бачимо з рисунку 1.1, потреба в талановитих людях в сфері будівництва є дуже великою. Близько 20% від загальної кількості дефіцитних галузей займає будівництво. Основною ціллю розробки є інформаційна система, що дозволить швидко знайти підходящого спеціаліста для виконання ремонтних робіт в квартирі, офісі, гаражі, будинку. Також виконавці повинні мати змогу реєструватися для пошуку роботи або підробітку. Для цього необхідно реалізувати два види особистого кабінету: «Замовника» та «Виконавця». Для більш гнучкої роботи системи, мною було обрано створення одного профілю з можливістю переключатися між режимами у своєму кабінеті.

Для того, щоб у замовника був вибір між декількома виконавцями, на одне замовлення повинна бути можливість подати заявку у всіх бажаних працівників, таким чином буде створено конкурентне середовище, що покращить якість виконання робіт, адже після завершення роботи, замовник її перевіряє та може залишити відгук про виконавця

Для цього повинна бути створена рейтингова система. Рейтинг має формуватися на основі відгуків та оцінок виконаної праці, і коригуватись з кожним новим відгуком.

Для зручності системи, вона має бути розміщена на хостингу, що дозволить мати доступ до неї онлайн з будь-якого телефону, планшета, або комп'ютера.

Таким чином, інформаційна система, що розроблюється, має включати в себе чітке розмежування ролей у користувачів, можливість створення, редагування, зберігання та видалення замовлень, рейтингову систему для створення конкурентного середовища. Продукт має працювати на різних платформах для забезпечення користувачів інформацією незалежно від платформи користування. Оновлення дошки оголошень має відбуватися у режимі реального часу задля забезпечення актуальної інформації для всіх замовників та виконавців.

В інформаційній системі буде автоматизований пошук виконавців на конкретне замовлення з однієї сторони, та робітників на виконання певних робіт або задач з іншої.

Система дозволить працівникам сфери будівництва легше знаходити собі роботу, котру вони можуть виконати. Для цього в оголошенні, котре буде подавати замовник будуть вказані всі потрібні для виконання роботи деталі, об'єм необхідних робіт, навички, котрі хотів би бачити замовник у виконавці та інші деталі замовлення.

З іншого боку для замовника буде надана можливість обирати саме того виконавця, котрий йому потрібен, завдяки можливості отримувати декілька заявок на виконання замовлення з короткою характеристикою переваг кожного з робітників. Також замовник буде бачити рейтинг всіх виконавців, що дозволить обрати найдосвідченішого з-поміж них.

Ця система вирішує питання довгого пошуку роботи в сфері будівництва. Тим самим полегшує ведення робіт, пришвидшує терміни їх виконання та спрощує життя як роботодавцям так і найманим працівникам. Так як сфера будівництва буде завжди необхідна людству, я вирішив спростити та вдосконалити пошук необхідних людей. В майбутньому планується розширення системи, додавання туди інтернет-магазину з будівельними товарами, для того щоб ведення ремонтних робіт стало ще більш зручним та комфортним.

Основним актором у системі є користувач. Він може бути як замовником так і виконавцем, для цього було введено переключення між режимами в особистому кабінеті. Тому у користувача є два варіанти використання інформаційної системи – це пошук роботи, або створення замовлення та пошук виконавців. На рисунку 1.2 зображена модель роботи інформаційної системи пошуку виконавців або замовників надання послуг(на прикладі будівництва) для кожного варіанту використання системи.

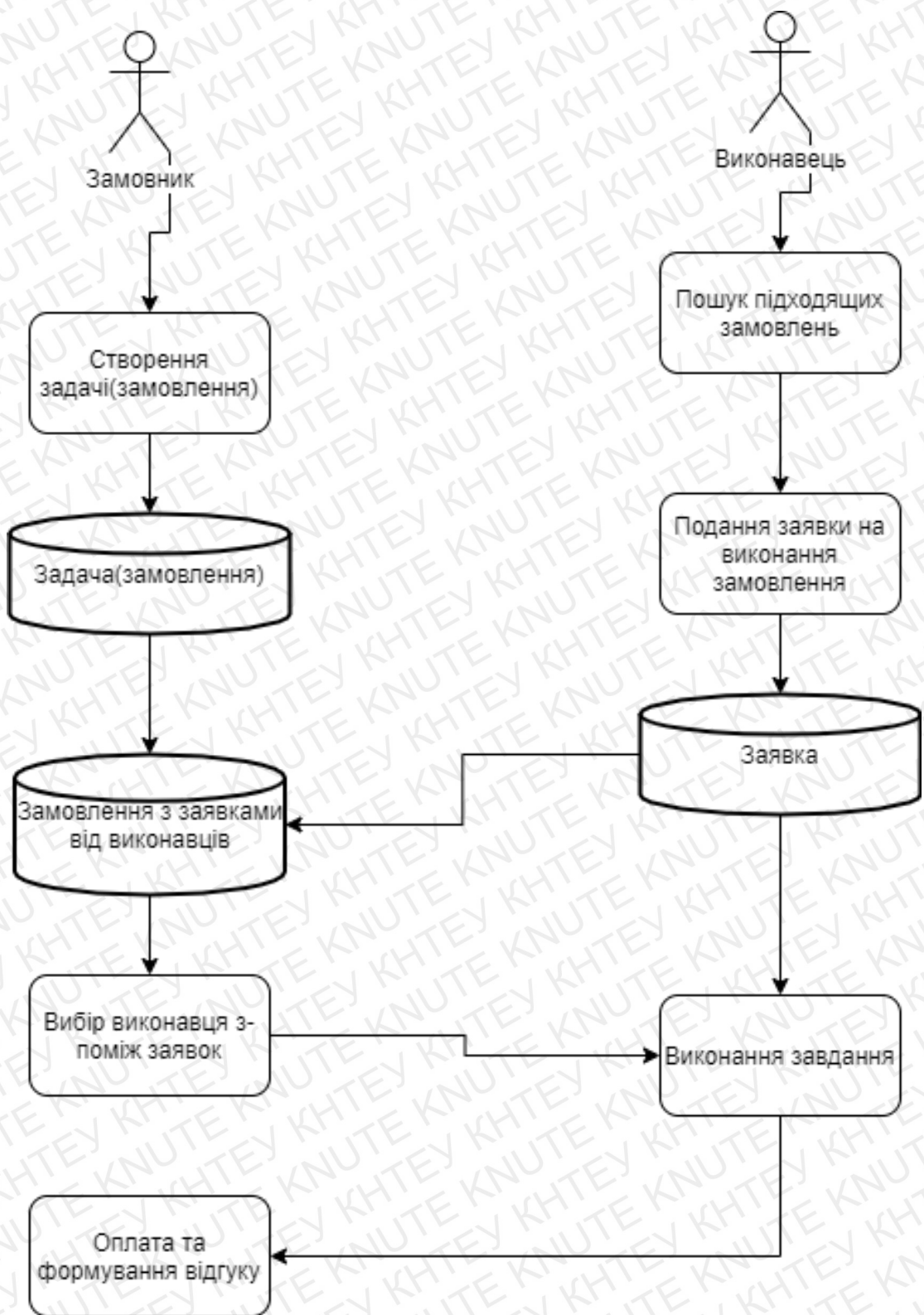


Рисунок 1.2 – UML-діаграма функціональної моделі системи

На діаграмі присутні два актори: замовник та виконавець.

В процесі пошуку виконавця наявні наступні етапи:

- створення замовлення замовником;
- подача заявки на виконання виконавцем;
- вибір одного виконавця з-поміж інших;
- виконання замовлення виконавцем;
- оплата та формування відгуку;

Створення замовлення

На даному етапі замовник описує спектр робіт, котрі повинні бути виконані з усіма додатковими уточненнями щодо їх виконання, такими як: заголовок, категорія робіт, деталі виконання завдання, місто, адреса, дата виконання, орієнтовна вартість(або договірна). Потім він публікує своє замовлення на дошку оголошень. Протягом деякого часу замовник отримує заявки від виконавців на виконання робіт. У заявці виконавець додає коментар з текстом, в якому може розповісти чому саме він має виконувати це замовлення, які в нього переваги, та інші можливі деталі. Потім замовник повинен обрати одного виконавця з-поміж інших для виконання робіт. Наступним кроком є обмін контактними даними, для подальшої комунікації між замовником та виконавцем. Надалі замовник та виконавець домовляються про саме виконання робіт, їх графік і т.д., це вони роблять поза межами інформаційної системи. Після того як виконавець виконує роботу, замовник її оплачує та формує відгук та оцінку в інформаційній системі. Цей відгук та оцінка дуже важливі, тому що вони потім формулюють рейтинг виконавця та допомагають іншим людям бачити, чи є він майстром своєї справи чи ні.

Також наглядний приклад використання системи з боку користувача можна побачити на рисунку 1.3.

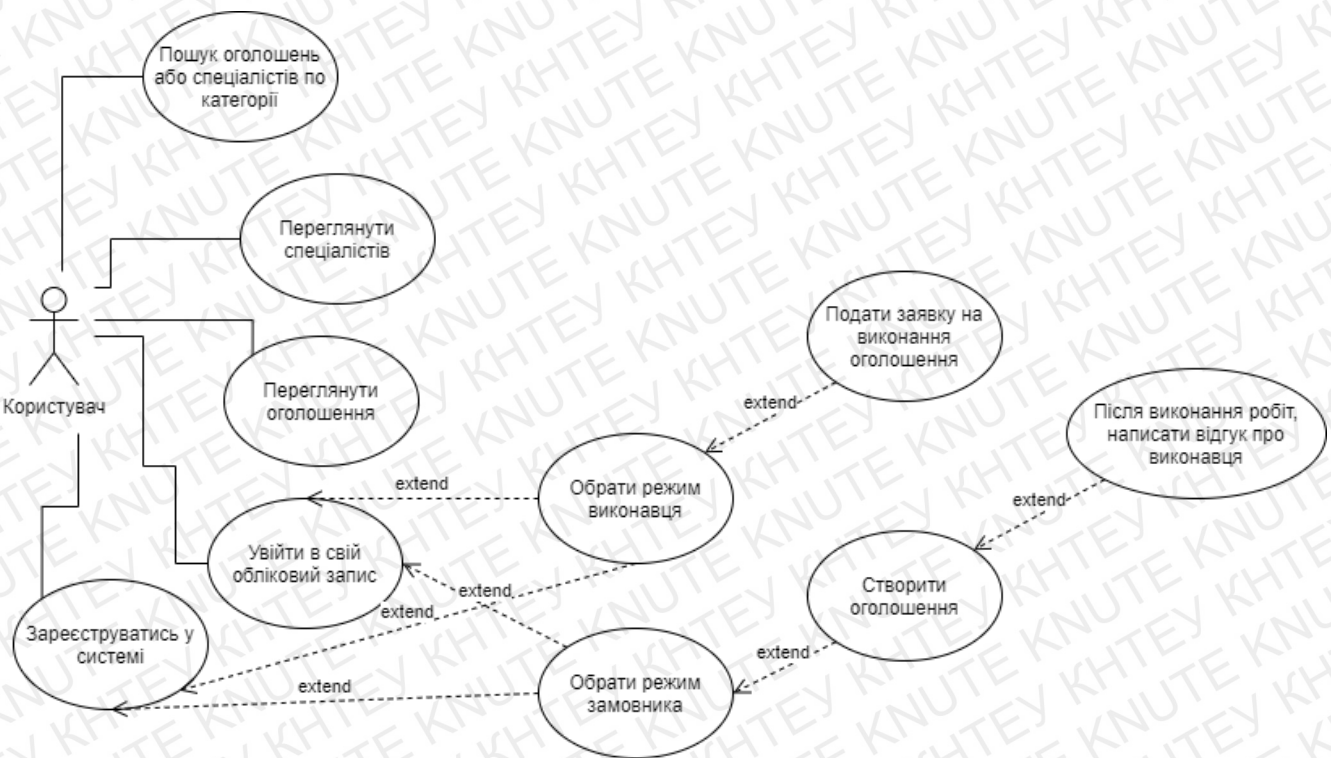


Рисунок 1.3 – Діаграма варіантів використання

Відповідно до визначених варіантів використання виявлено функціональні вимоги та встановлена їх пріоритетність, результат наведено у таблиці 1.1.

Таблиця 1.1 – Функціональні вимоги

Актор	Варіант використання	Функціональна вимога	Пріоритет
Користувач	Зареєструватись у системі	<p>1. Система надає клієнту реєстраційну форму для заповнення.</p> <p>1.1 Система потребує від клієнта заповнити такі поля: електронну пошту, номер телефону, пароль, обрати категорії, ім'я, прізвище, місто та завантажити особисту фотографію</p> <p>1.2 Система вимагає від користувача введення правильного номеру телефону та електронної пошти для реєстрації у системі.</p>	<p>Важливий</p> <p>Важливий</p> <p>Важливий</p>
	Увійти в свій обліковий запис	<p>2. Система надає користувачу можливість авторизуватись у системі.</p> <p>2.1 Система вимагає від користувача введення електронної пошти та паролю.</p> <p>2.2 Система надає можливість перейти на сторінку реєстрації, якщо користувач не зареєстрований.</p>	<p>Важливий</p> <p>Важливий</p> <p>Важливий</p>

Таблиця 1.1 – Функціональні вимоги – продовження

Актор	Варіант використання	Функціональна вимога	Пріоритет
	Переглянути оголошення	<p>3. Система надає можливість користувачеві переглянути список оголошень.</p> <p>3.1 При натисканні на оголошення з'являється спливаюче віконце з деталями оголошення, та кнопкою для початку реєстрації, якщо користувач не авторизований в системі</p>	<p>Важливий</p> <p>Важливий</p>
	Переглянути спеціалістів	<p>4. Система надає можливість користувачеві переглянути спеціалістів, котрі зареєстровані в системі.</p> <p>4.1 При натискання на кнопку відправити замовлення, система пропонує користувачеві зареєструватися, якщо він не зареєстрований, або переводить його на сторінку створення оголошення, якщо користувач авторизований</p>	<p>Важливий</p> <p>Важливий</p>

Таблиця 1.1 – Функціональні вимоги – продовження

Актор	Варіант використання	Функціональна вимога	Пріоритет
	Обрати режим «Виконавця» або «Замовника»	5. Система надає можливість в особистому кабінеті перемикались між режимами, для полегшення роботи з системою та розмежування функцій	Важливий
	Створити оголошення	6. Система надає можливість авторизованим користувачам створювати оголошення в режимі замовника	Важливий
	Подати заявку на оголошення	7. Система надає можливість авторизованим користувачам подавати заявку на виконання оголошення в режимі виконавця	Важливий
	Написати відгук	8. Система надає можливість авторизованим користувачам писати відгуки про виконавців після виконання ними замовлень в режимі замовника	Важливий

1.2. Огляд наявних аналогів

Ринок на даний момент має декілька аналогів інформаційної системи:

- kabanchik.ua – онлайн-сервіс пошуку приватних спеціалістів для вирішення побутових та бізнес-задач. Він об'єднує замовників послуг, котрим необхідно виконати яку-небудь роботу та компетентних спеціалістів, котрі шукають підробіток або додатковий заробіток;
- olx.ua - платформа онлайн-оголошень, яка об'єднує людей для покупки, продажу або обміну товарами та послугами;
- work.ua - сайт пошуку роботи в Україні. Зазвичай люди там шукають постійну роботу;
- rabota.ua - сайт пошуку роботи в Україні. Зазвичай люди там шукають постійну роботу;

Як ми можемо бачити з опису сервісів, вони є широко спеціалізованим і включають в себе найрізноманітніші сфери життя та бізнесу. І тому знайти вузького спеціаліста, в моєму випадку для будівництва, буде не так просто. Наявність інформаційної системи з пошуку виконавців або замовників послуг на прикладі будівництва допоможе легше та швидше знайти собі підходящих робітників, або у випадку виконавців, роботу чи підробіток.

1.3. Постановка задачі.

Призначенням розробки є реалізація інформаційної системи пошуку виконавців або замовників надання послуг(на прикладі будівництва) яка включає:

- Можливість створення, редагування, видалення замовлення

- Можливість подання одним виконавцем заявок на різні замовлення
- Створення рейтингової системи оцінки роботи виконавців
- Створення одного профілю з двома режимами: «Замовник» та «Виконавець»
- Формування чесного конкурентного середовища за рахунок можливості отримання замовником декількох заявок на виконання замовлення
- Актуальність даних в системі, за рахунок розміщення її на хостингу

Цілі та задачі розробки

1. Забезпечити швидкий пошук виконавців для замовників.
2. Забезпечити швидкий пошук роботи та замовлень зі сторони виконавця.
3. Забезпечити актуальність даних.

РОЗДІЛ 2.

ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1. Вхідні та вихідні дані

На вхід ми отримуємо два актори – замовник та виконавець

Замовник має можливість створювати, редагувати, видаляти замовлення, писати відгук та оцінювати роботу виконавця

Виконавець має можливість переглядати всі актуальні замовлення, подавати заявки на виконання, отримувати відгук від замовника, тим самим формувати свій рейтинг

На виході ми отримуємо список виконаних замовлень с позитивними або негативними відгуками, таким чином, оцінивши кількість позитивних замовлень відносно негативних ми можемо оцінити користь від роботи системи

2.2. Опис структури бази даних

Основними в базі даних є користувач та замовлення. Користувачі зберігаються в таблиці, що містить інформацію про ім'я, прізвище, опис, електронну пошту, номер телефону, пароль, основне зображення профілю, роль користувача, його рейтинг та місто.

Таблиця з замовленнями містить інформацію про заголовок, опис, оголошену вартість, адресу, дату початку замовлення, місто, де необхідно його виконати, номер користувача, котрий його створив та номер користувача котрий буде його виконувати.

Кожен користувач та замовлення пов'язані з категоріями зв'язком один до багатьох, про що свідчить існування таблиць виду Користувач-Категорія та Замовлення-Категорія відповідно. Поля цих таблиць містять інформацію про те,

які саме категорії обрав для себе виконавець та які категорії замовник вказав у замовленні.

Безпосередньо таблиця Категорій містить в собі інформацію про заголовок та опис категорій.

Також кожен користувач та замовлення мають такі поля як місто. В таблиці з містами вказані можливі варіанти країн та міст, де буде працювати інформаційна система.

Кожен виконавець має змогу подати заявку на виконання певного замовлення. Один виконавець може подавати заявки на будь-які замовлення. Тому в таблиці Заявок ми бачимо такі поля як ідентифікатор користувача, котрий подав заявку, ідентифікатор замовлення на котре була подана заявка, та коментар виконавця, де він вказує причини, чому саме він має виконувати це замовлення.

Для того щоб вірно обрати собі виконавця, замовник має бачити відгуки про нього. Вони зберігаються в таблиці Відгуків. Вона включає в себе такі поля: ідентифікатор виконавця, ідентифікатор замовлення, текст відгуку та булеву змінну, котра відображатиме чи це позитивний відгук, чи негативний.

Таблиці у базі даних та зв'язки між ними відображені у ER (Entity Relationship) діаграмі (Рисунок 2.1).

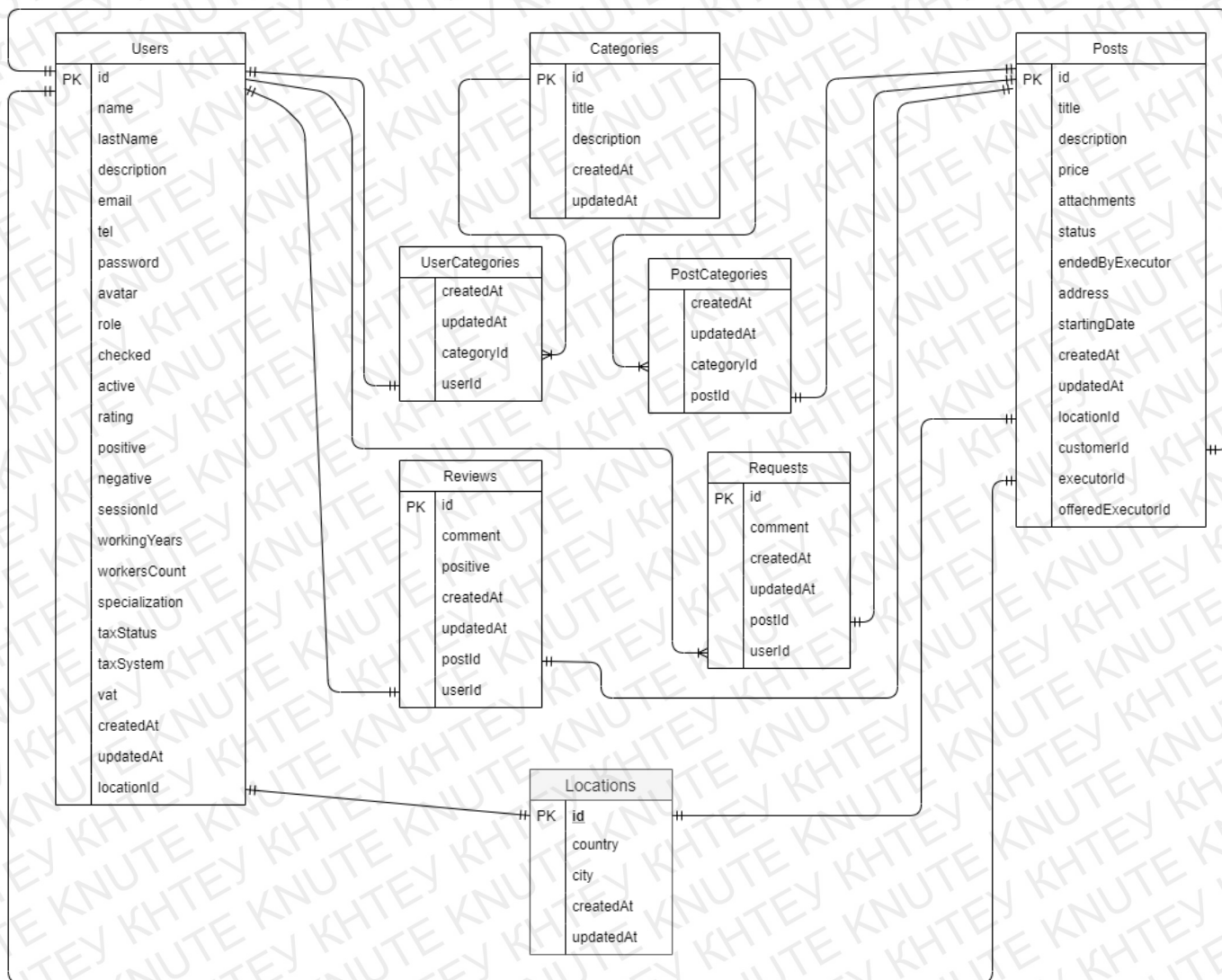


Рисунок 2.1 – ER діаграма

2.3. Змістовна та математична постановки задачі

Змістовна: на сьогоднішній день для створення та підтримання інформаційних систем використовуються безліч рішень, котрі мають свої переваги та недоліки. Інформаційну систему можна створити такими способами як:

- додаток на смартфон;
- додаток на персональний комп'ютер;
- локальна інформаційна система;

– веб-застосунок;

Якщо розглядати перші два варіанти розробки, вони матимуть найбільший недолік, це недоступність на всіх сучасних девайсах, адже додаток на смартфоні неможливо переглянути через персональний комп'ютер і навпаки.

Якщо розглядати локальну інформаційну систему, вона матиме такий недолік як недостовірність актуальних даних, тому що дані потрібно оновлювати в ручному режимі як тільки вони будуть змінені.

Тому найкращим варіантом для впровадження інформаційної системи буде створення та розміщення її на веб-ресурсах, що дозволить завжди мати актуальні дані, швидкий доступ до них, та можливість миттєвого редагування програмного забезпечення інформаційної системи.

Математична: нехай існує набір виконавців, котрі шукають роботу та набір замовників, котрим потрібні послуги виконавців. Також відомо що у виконавців є рейтинг, котрий формується на основі кількості та якості виконаних ними замовлень.

Завдання: створити інформаційну систему що дозволить замовникам створювати замовлення з деталізованим описом потрібних їм робіт, а виконавцям шукати дані замовлення та подавати заявки на їх виконання, тобто створити дошку оголошень.

2.4 Обґрунтування та опис методу розв'язання.

Для розв'язання задачі проектування системи було вирішено створити онлайн дошку оголошень, так як саме вона дозволить показувати користувачам актуальну інформацію в будь-який час, та вона не потребує постійного контролю та виправлень.

Для коректного та актуального відображення інформації мною було обрано створити інформаційну систему у вигляді веб-сайту, що дозволить швидко оновлювати дані, тим самим зменшити користувачам час перегляду та пошуку актуальної інформації.

Розміщення системи на веб-ресурсі має ряд переваг, таких як:

- не потрібні великі обчислювальні потужності ПК - по суті будь-який смартфон, планшет і т.д., при відкритті вікна браузера отримує всю потрібну інформацію;
- відмовостійкість;
- певний рівень безпеки;
- висока швидкість обробки даних;
- економія на покупці софта - всі необхідні програми вже є в сервісі, де буде працювати система;
- всі дані зберігаються в мережі;

Також потрібно врахувати певні ризики, такі як:

- хмарна послуга надається завжди якоюсь компанією, відповідно, збереження даних користувача залежить від цієї компанії;

небезпека хакерських атак на сервер;

В моїй інформаційній системі враховані всі ризики, всі дані валідуються, тим самим зменшується ризик потрапляння чужого коду на мій веб-ресурс.

РОЗДІЛ 3.

ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Засоби розробки

Для реалізації систем, що розмішуються на веб-ресурсах є доволі багато засобів розробки. Адже потрібно обирати на чому буде написаний front-end та back-end веб-сторінки. Фронтенд-розробка зосереджена на тих елементах сайту, які ви бачите у браузері і з якими безпосередньо взаємодієте. Бекенд-розробка відповідає за функціонал сайту і має справу з речами, яких ви не бачите, — такими як бази даних та сервери.

Є багато систем, котрі дозволяють використовувати готові шаблони для побудови веб-застосунку, такі як WordPress, Medium, Joomla, Drupal, WIX, Ghost, та інші. В них вже автоматично генерується код для front-end та back-end. Також можна самостійно розробляти веб-застосунок, або використовувати фреймворки.

Фреймворк — інфраструктура програмних рішень, що полегшує розробку складних систем. Спрощено дану інфраструктуру можна вважати своєрідною комплексною бібліотекою, але при цьому вона має ряд обмежень, що задають правила створення структури проекту та написання коду.

Часто використовують такі фреймворки як Angular, React та Vue.js. Для написання даної дипломної роботи було обрано фреймворк Angular для front-end та Node.js для back-end.

Angular - написаний на TypeScript front-end фреймворк з відкритим кодом, який розробляється під керівництвом Angular Team у компанії Google, а також спільноту приватних розробників та корпорацій. Переваги Angular:

- Angular є повноцінним MVC фреймворком, який надає розробникові всі можливості для програмування “з коробки”: шаблони, засновані на HTML; впровадження залежності (Dependency injection); запити Ajax; роутінг; інкапсуляція компонентів CSS; утиліти для тестування компонентів; можливості для створення форм тощо;

- Angular підтримує поняття компонент(створення власного компоненту в розмітці веб-сторінки з можливістю його подальшого використання на інших сторінках);
- керування станом. Будь-який веб-додаток характеризується певним станом. Інтерфейс користувача, який описується компонентом, відповідає певному стану компонента на даний час. Як тільки дані компонента змінюються, фреймворк змінює відповідно і інтерфейс користувача. Це гарантує, що дані завжди актуальні;
- прив'язка даних (Data Binding). Angular використовує двостороннє зв'язування. Наприклад, якщо ви змінюєте елемент інтерфейсу (введення користувача) на Angular, то також змінюється відповідний стан моделі. Крім того, якщо ви змінюєте стан моделі, то змінюється і елемент інтерфейсу — отже, двостороння прив'язка даних;

Для написання бекенд серверу мною було обрано Node.js.

Node.js — платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript.

Основними перевагами цієї платформи є асинхронність в поєднанні з подійним підходом. Тобто виконання програми залежить від дій користувача.

Таке рішення спрощує розробку інтерактивних веб-додатків, тобто основаних на роботі з використанням вводу-виводу.

Іншими перевагами платформи є: простий та відомий JavaScript, багата стандартна бібліотека та величезна кількість новостворених бібліотек та готових модулів від користувачів платформи. На сьогодні кількість пакетів у відкритому доступі більш ніж 500 тисяч.

Для роботи з базою даних було обрано Sequelize.

Sequelize - це ORM-бібліотека для додатків на Node.js, котра виконує співставлення таблиць в базі даних та відношень між ними з класами. При використанні Sequelize ми можемо не писати SQL-запити а працювати з даними

як зі звичайними об'єктами. В даному дипломному проєкті Sequelize працює в парі з СУБД Postgres.

Для додаткового захисту сервера та даних від неавторизованого доступу, був реалізований механізм автентифікації та авторизації з використанням JWT (JSON Web Token). Це стандартизований, в деяких випадках підписаний та/або зашифрований формат упаковки даних, котрий використовується для безпечної передачі інформації між двома сторонами.

Токен JWT складається з трьох частин: заголовок (header), вмісту (payload) і підпису. Заголовок це JSON об'єкт, який описує до якого типу токену належить даний і які методи шифрування використовувались. Вміст складається з об'єкта JSON який описує твердження. Щоб згенерувати підпис заголовок та вміст кодуються в Base64, записуються в один рядок через крапку, а потім цей рядок хешується визначеним методом.

3.2. Вимоги до технічного забезпечення

Користувач повинен мати смартфон, ПК, або ноутбук котрий матиме доступ до мережі інтернет. Швидкість з'єднання впливатиме лише на швидкість завантаження вмісту веб-сторінки, тому бажано щоб вона була не менш ніж 35 мбіт/с. та діагональ екрану не менш ніж 4.0 ", так як відображений контент може бути спотворений.

3.3. Архітектура програмного забезпечення

Нижче наведена діаграма класів (рис. 3.1).

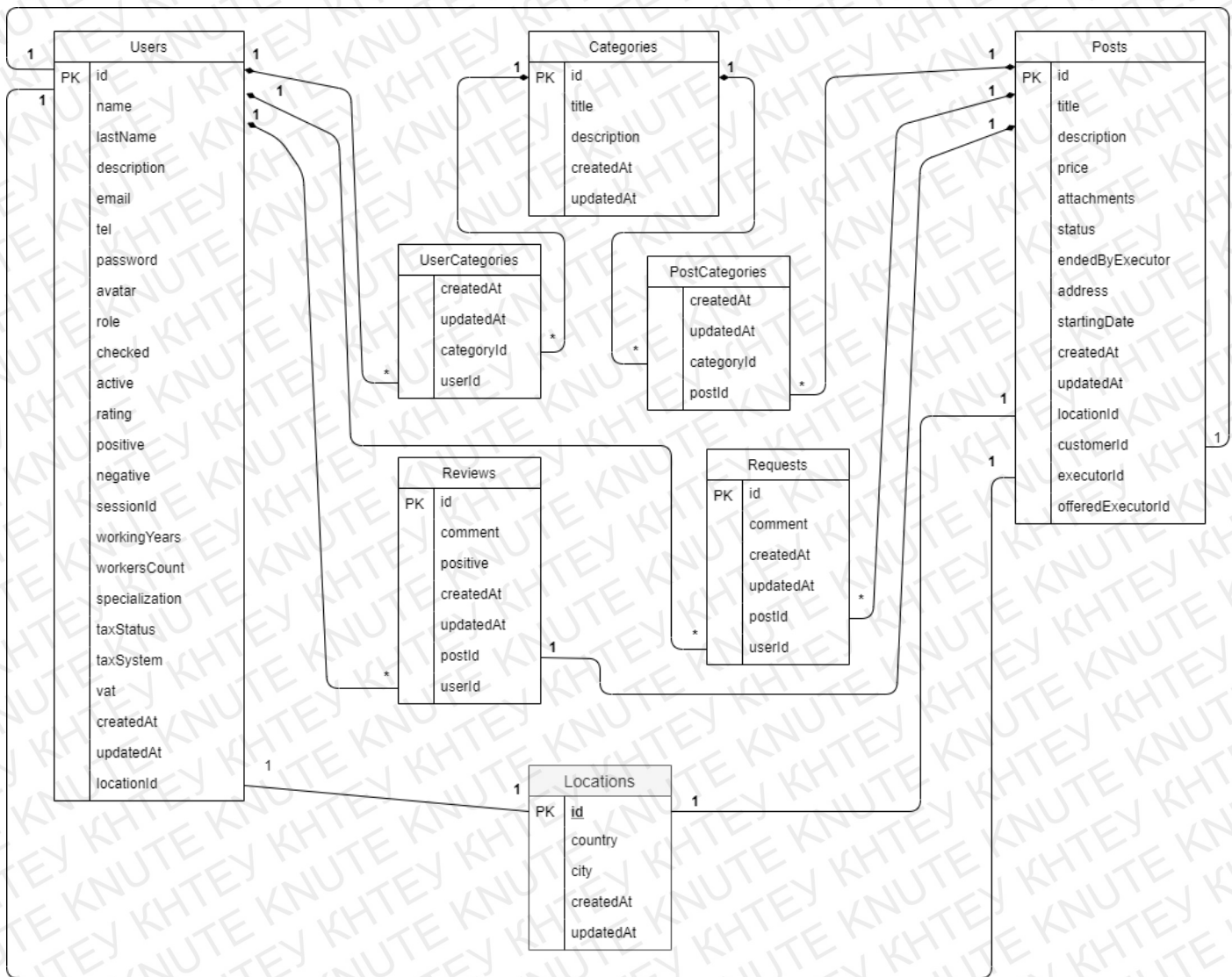


Рисунок 3.1 – Діаграма класів

Основними класами є користувач та замовлення. Користувачі мають такі атрибути: ім'я, прізвище, опис, електронну пошту, номер телефону, пароль, основне зображення профілю, роль користувача, його рейтинг та місто.

Клас замовленнями містить такі атрибути: заголовок, опис, оголошену вартість, адресу, дату початку замовлення, місто, де необхідно його виконати, номер користувача, котрий його створив та номер користувача котрий буде його виконувати.

Кожен користувач та замовлення пов'язані з категоріями зв'язком один до багатьох, про що свідчить існування класу виду Користувач-Категорія та

Замовлення-Категорія відповідно. Атрибути цих класів містять інформацію про те, які саме категорії обрав для себе виконавець та які категорії замовник вказав у замовленні.

Безпосередньо клас Категорій містить в собі такі атрибути як заголовок та опис категорій.

Також кожен користувач та замовлення мають такі атрибути як місто. В класі Локація вказані атрибути: країна та місто.

Кожен виконавець має змогу подати заявку на виконання певного замовлення. Один виконавець може подавати заявки на будь-які замовлення. Тому в класі Заявок ми бачимо такі атрибути як ідентифікатор користувача, котрий подав заявку, ідентифікатор замовлення на котре була подана заявка, та коментар виконавця, де він вказує причини, чому саме він має виконувати це замовлення.

Для того щоб вірно обрати собі виконавця, замовник має бачити відгуки про нього. Вони зберігаються в класі Відгуків. Він включає в себе такі атрибути: ідентифікатор виконавця, ідентифікатор замовлення, текст відгуку та булеву змінну, котра відображатиме чи це позитивний відгук, чи негативний.

Нижче наведена діаграма послідовності (рис. 3.2).

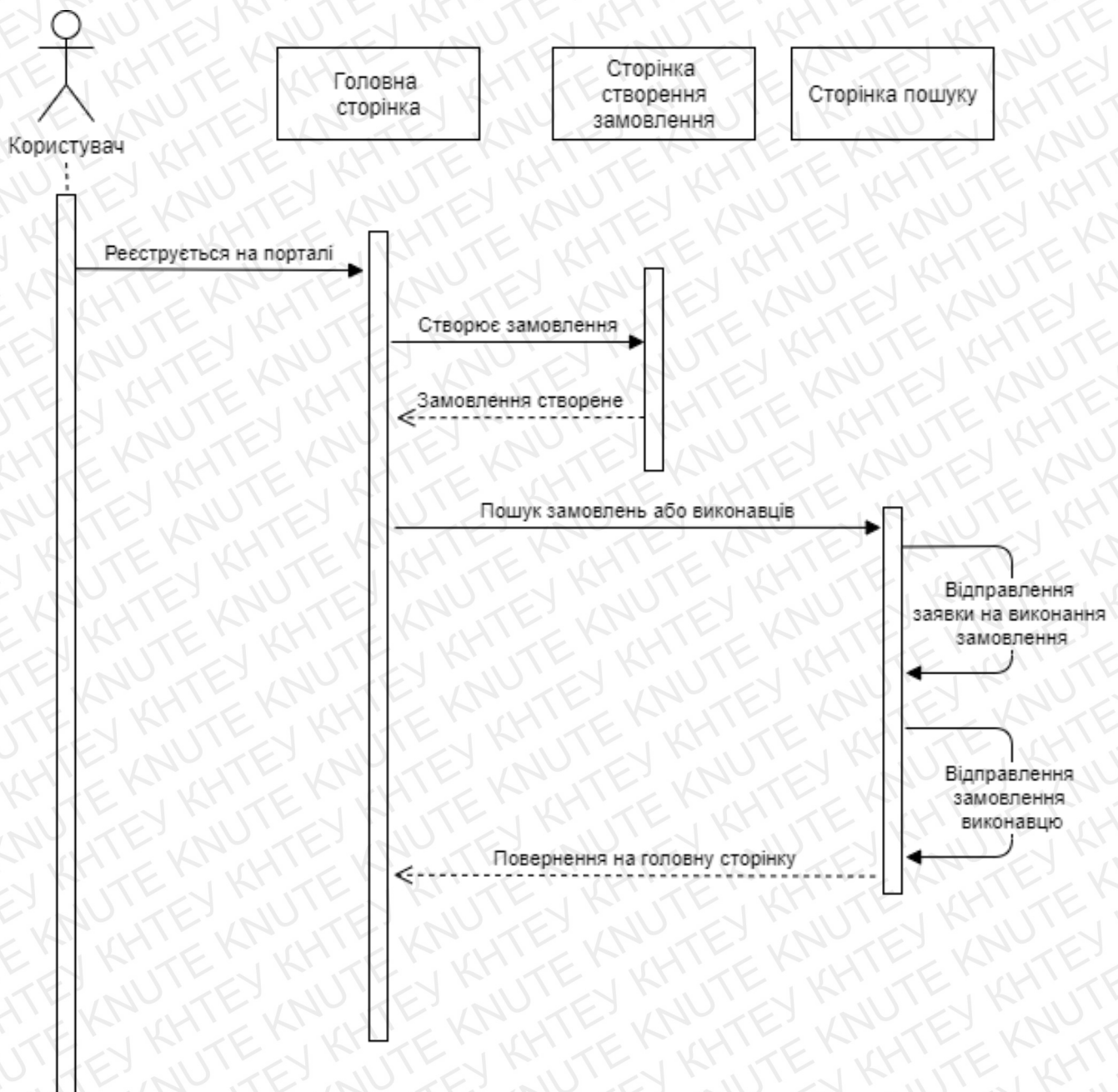


Рисунок 3.2 – Діаграма послідовності

Як ми бачимо на діаграмі, для доступу до всіх можливостей системи користувачу потрібно в ній зареєструватись. Після реєстрації, користувач має змогу бути виконавцем та замовником одночасно. Таким чином полегшується використання системи для різних задач, адже користувачу не потрібно реєструвати два профіля, один для замовника, інший для виконавця.

Користувач має можливість створювати власні замовлення на виконання ремонтних робіт, отримувати на них заявки від виконавців та обирати потрібного йому майстра своєї справи.

З іншого боку користувач може виступати у ролі виконавця, тим самим шукаючи підходящі для нього замовлення та залишати свою заявку на його виконання.

Нижче наведена діаграма компонентів (рис. 3.3).



Рисунок 3.3 – Діаграма компонентів

Як ми бачимо з діаграми компонентів, система складається з двох ключових складових – це фронтенд та бекенд.

В бекенд входять такі складові як: файл `Server.js`, котрий відповідає за те, що відбувається на стороні сервера; моделі бази даних, в котрих прописані всі характеристики, котрі матимуть таблиці в базі даних та відношення між ними; контролери авторизації, об'яв, часто повторюваних питань та локацій, що відповідають за операції, котрі проходять з відповідними таблицями в базі даних; конфігураційний файл сервера; папка з головними фото користувачів; папка з завантаженнями, куди будуть потрапляти всі завантажені користувачами файли.

У фронтенд входить дві великі складові – це сервіси та компоненти. Сервіси відповідають за зв'язок між фронтенд-частиною та бекенд-частиною застосунку, тобто виконують функцію передачі введеної користувачем інформації на бекенд, а звідти вже вона піде в базу даних. Компоненти відповідають за правильне графічне відображення на моніторі всіх складових сайту. Тобто в компоненти входять файли, в котрих прописана розмітка сторінки, варіанти взаємодії кожного компонента з користувачем та файли стилів, що відповідають за правильне розміщення елементів на сторінці у вікні браузера.

Нижче наведена таблиця зі специфікацією функцій(таблиця 3.1).

Таблиця 3.1 – Специфікація функцій

Назва функції	Опис функції	Вхідні дані	Вихідні дані
sequelize.define()	Створює в базі даних таблицю с визначеними полями та атрибутами	Назва таблиці, назви полів та їх атрибути	На виході отримуємо таблицю в базі даних
router.get()	Обробляє дані на сервері, при певному запиті	Посилання, при котрому функція спрацьовує, GET-параметри з посилання	Дані, взяти з таблиці по певним критеріям
router.post()	Додає нові записи в базу даних	Посилання, при котрому функція спрацьовує, параметри з посилання	Додавання в базу даних нових значень відповідно до запиту
router.put()	Змінює записи в базі даних	Посилання, при котрому функція спрацьовує, параметри з посилання	Зміна записів в базі даних

3.4. Програмна реалізація і технологія використання

При потраплянні на сайт, користувач одразу бачить головну сторінку

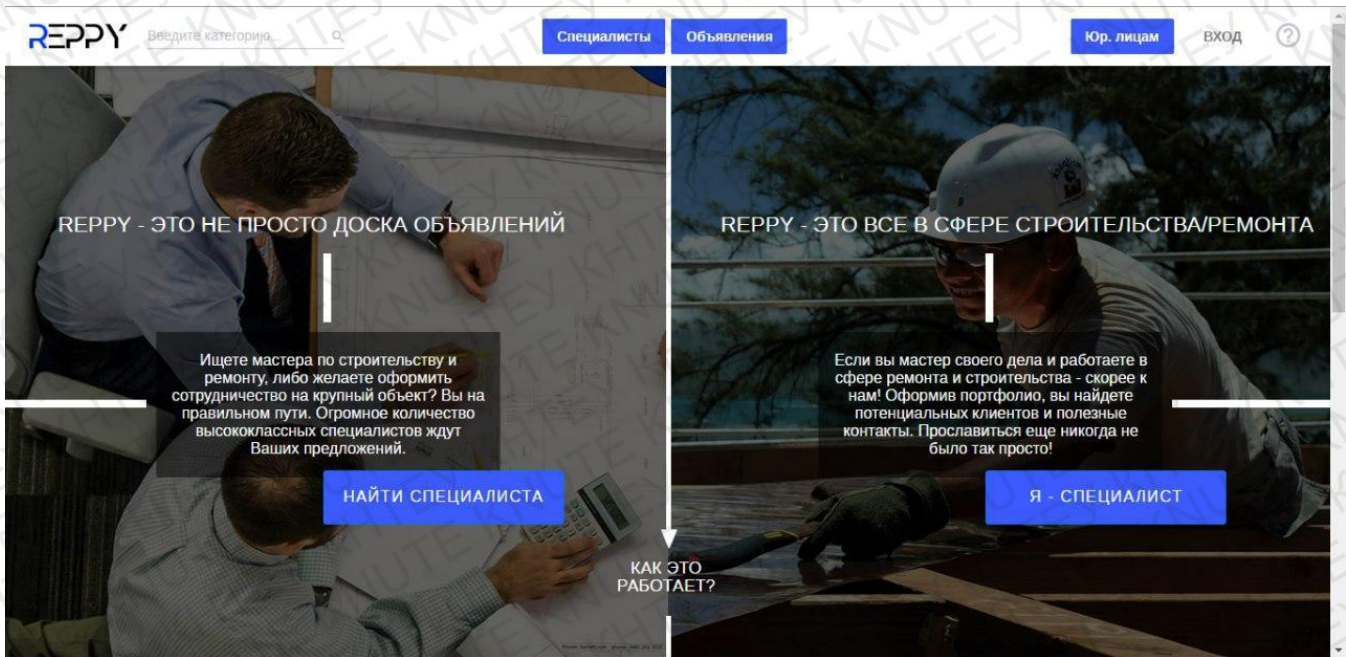


Рисунок 3.4 – Головне меню

На ній користувач може переглянути як працює сервіс, перейти за посиланнями та переглянути всіх спеціалістів, всі об'яви, зробити пошук по категорії та зареєструватись або авторизуватись у системі.

При натисканні на кнопку «Як це працює» сторінка прокручується вниз і користувач бачить опис взаємодії з сервісом, посилання на соціальні мережі внизу сторінки та версію інформаційної системи.



Рисунок 3.5 – «Як це працює?»

При натисканні на кнопку «Спеціалісти» користувач переходить на сторінку з усіма спеціалістами.

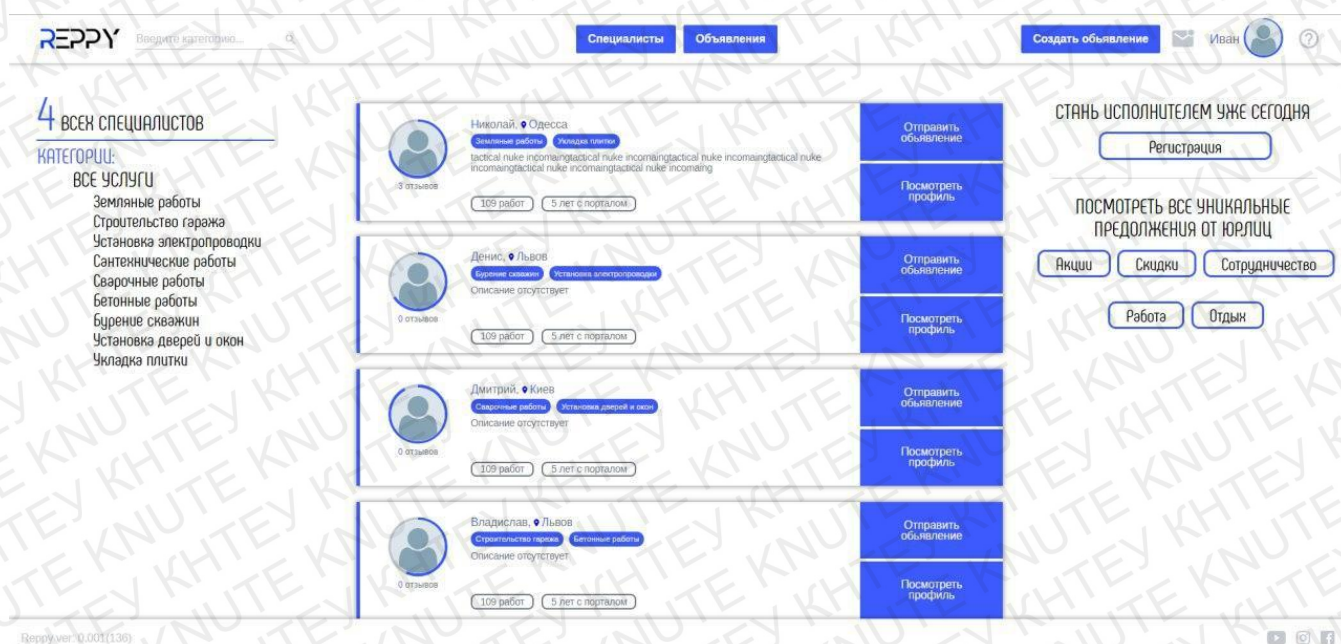


Рисунок 3.6 – Розділ «Спеціалісти»

Тут він може переглянути профіль спеціаліста та надіслати йому своє замовлення. Також є можливість відсортувати їх за категоріями.

При натисканні на кнопку «Оголошення» користувач переходить на сторінку з усіма оголошеннями.

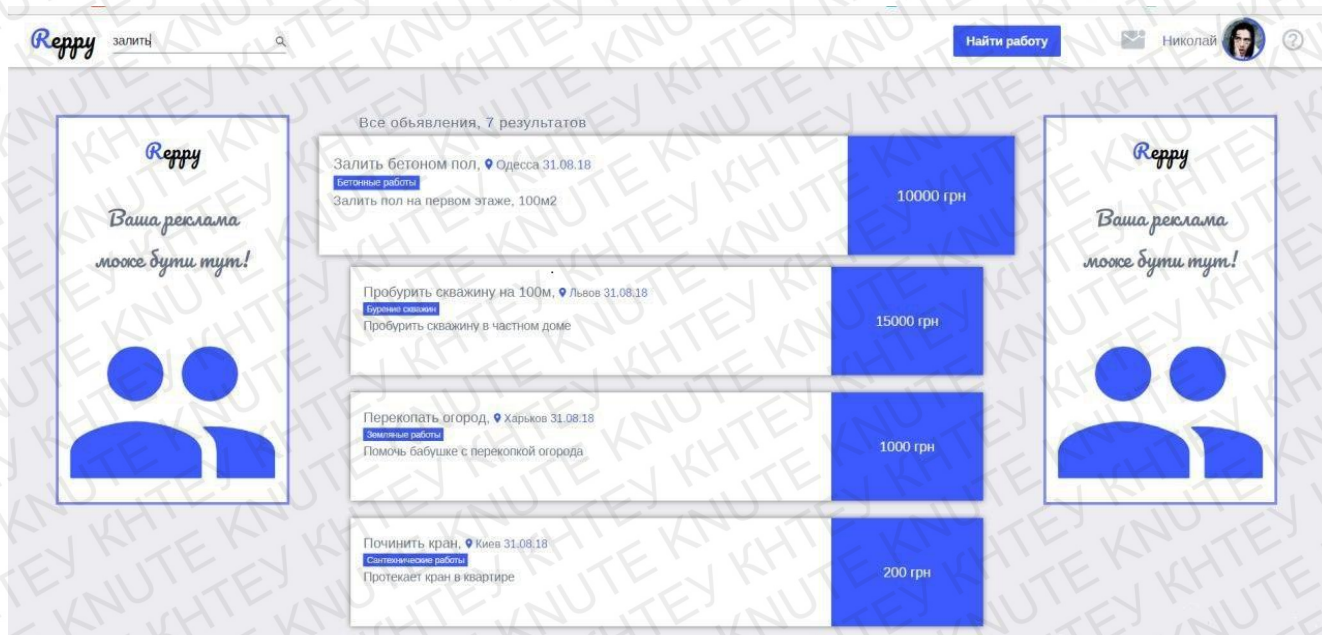


Рисунок 3.7 – «Оголошення»

Тут користувач може переглянути доступні оголошення, виконати пошук, або знайти роботу/створити оголошення в залежності від того в якому режимі він знаходиться, переключенням режимів відбувається у особистому кабінеті. Для доступу в свій кабінет користувач має зареєструватись.

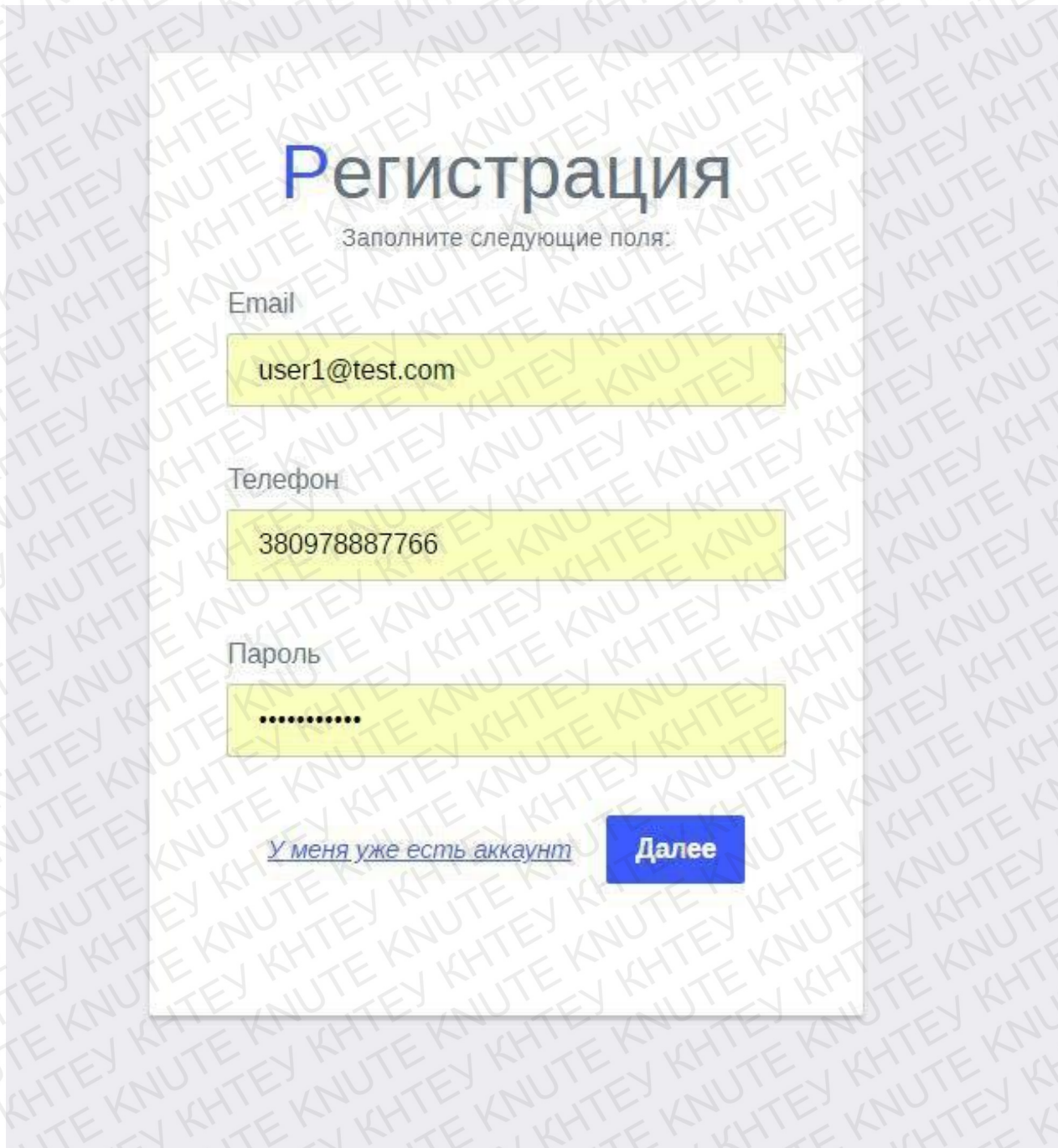
A screenshot of a registration form titled "Регистрация" (Registration). The form is set against a light gray background. At the top, the word "Регистрация" is written in a large, blue, sans-serif font. Below it, the instruction "Заполните следующие поля:" (Fill in the following fields:) is displayed in a smaller, dark gray font. There are three input fields, each with a yellow background and a thin black border. The first field is labeled "Email" and contains the text "user1@test.com". The second field is labeled "Телефон" (Phone) and contains the number "380978887766". The third field is labeled "Пароль" (Password) and contains a series of ten black dots. Below the input fields, there is a blue button with the white text "Далее" (Next). To the left of the button, there is a link that says "У меня уже есть аккаунт" (I already have an account) in a dark gray font, underlined.

Рисунок 3.8 – Сторінка реєстрації, 1 етап

На першому етапі реєстрації користувач має ввести такі дані як: електронну пошту, телефон, пароль.

Регистрация

Заполните следующие поля:

Имя

Фамилия

Локация

Ваше фото :




Рисунок 3.9 – Сторінка реєстрації, 2 етап

На другому етапі реєстрації користувач повинен ввести своє ім'я, прізвище, місто, в якому він знаходиться та завантажити фото профілю.

Иван Петров

Заказчик Исполнитель

Активные(3) | Выполненные(3) | Предложения(3)

Залить бетоном пол, Киев 29.08.2018
Залить пол на первом этаже, 100м2 [Посмотреть](#)

Предложения:

- Сделаю быстро и качественно [Нанять](#)
- Работаю в этой сфере очень долго [Нанять](#)
- Занимаю заливкой бетона более 10 лет [Нанять](#)


Пробурить скважину на 100м, Киев 29.08.2018
Пробурить скважину в частном доме [Посмотреть](#)

Рисунок 3.10 – Профиль замовника

У профілі замовника користувач бачить свій рейтинг, активні, виконані оголошення та пропозиції до виконання його оголошень, він може переглянути своє оголошення, а також прийняти на роботу одного з виконавців. У користувача є можливість редагування свого профілю.

Николай Васильев [Редактировать](#)

Заказчик Исполнитель



73%
ReppuRate

Активные(5) | Выполненные(3) | Мои заявки(1) | Предложения(1)

Пробурить скважину на 100м, 📍 Львов 12.11.2018
380991234567

[Бурение скважин](#)

Пробурить скважину в частном доме

[Посмотреть](#)

Ожидает ревью .

Сварить систему выхлопа, 📍 Харьков 12.11.2018
380991234567

[Сварочные работы](#)

Сварить паука для ЗАЗ-968М

[Посмотреть](#)

Ожидает ревью .

Скважина для питьевой воды, 📍 Львов 12.11.2018
380991234567

[Бурение скважин](#)

[Посмотреть](#)

Ожидает

Рисунок 3.11 – Профиль виконавця

У профілі виконавця користувач бачить свій рейтинг, свої відправлені заявки, виконані замовлення та пропозиції від користувачів щодо роботи.

При використанні профілю замовника у користувача є можливість створювати замовлення. У користувача є можливість редагування свого профілю.

Имя
Иван

Фамилия
Петров

Локация
Харьков

Ваше фото :

Выбрать файл

Сохранить

Рисунок 3.12 – Сторінка редагування профілю

Під час редагування профілю користувач може змінити своє ім'я, прізвище, місто та фото профілю.

REPPY Введите категорию...

Специалисты Объявления Создать объявление Иван

СОЗДАТЬ ОБЪЯВЛЕНИЕ 20%

Укажите как можно больше информации:

Чем больше информации Вы укажете в Вашем объявлении, тем больше вероятность того, что объявление будет в ТОП рейтинга и Вы очень быстро найдете нужного Вам мастера.

Укажите больше информации, Вы сделаете поиск нужных специалистов для Вашей работ

Напишите оглавление:

Установка электророзеток × Бетонные работы ×

Выберите нужное:

Укажите детали:

Отсутствует или некорректное описание

Адрес и дата выполнения:

Выберите город:

30% Возможно нужен для Вас специалист живет в 5 шагах от Вашего объекта. Упростите поиск, как себе, так и специалистам

Рисунок 3.13 – Створення замовлення, 1 етап

Для початку при створенні замовлення користувач повинен ввести назву оголошення, обрати категорію, розписати деталі замовлення.

и Вы очень быстро найдете нужного Вам мастера.

Выберите нужное: Категория

Укажите детали: Например: На данный момент в квартире ремонт 90-х годов, я бы хотел сделать более современный стиль с элементами Лофта

30%

Адрес и дата выполнения:

Выберите город: Город

Адрес объекта: Например: Ярослава Мудрого д.3

Дата выполнения:

Пон	Вт	Ср	Чт	Пт	Сб	Вс
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Рисунок 3.14 - Створення замовлення, 2 етап

На другому етапі створення замовлення користувач повинен обрати місто та адресу, де його замовлення буде виконувати виконавець, та дату виконання.

REPPY Введіть категорію...

Специалисты **Объявления** Создать объявление Иван

Дата выполнения:

Пон	Вт	Ср	Чт	Пт	Сб	Вс
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

80%

Оплата работы:

Ориентировочная стоимость работ:

Цена обсуждается лично

Необходимо выбрать цену или выбрать пункт "Договорная цена"

Создать объявление

Рисунок 3.15 - Створення замовлення, 3 етап

На останньому етапі користувач повинен вказати вартість робіт, також є можливість поставити галочку, що означатиме що вартість буде оговорюватись.

У інформаційній системі присутній пошук.

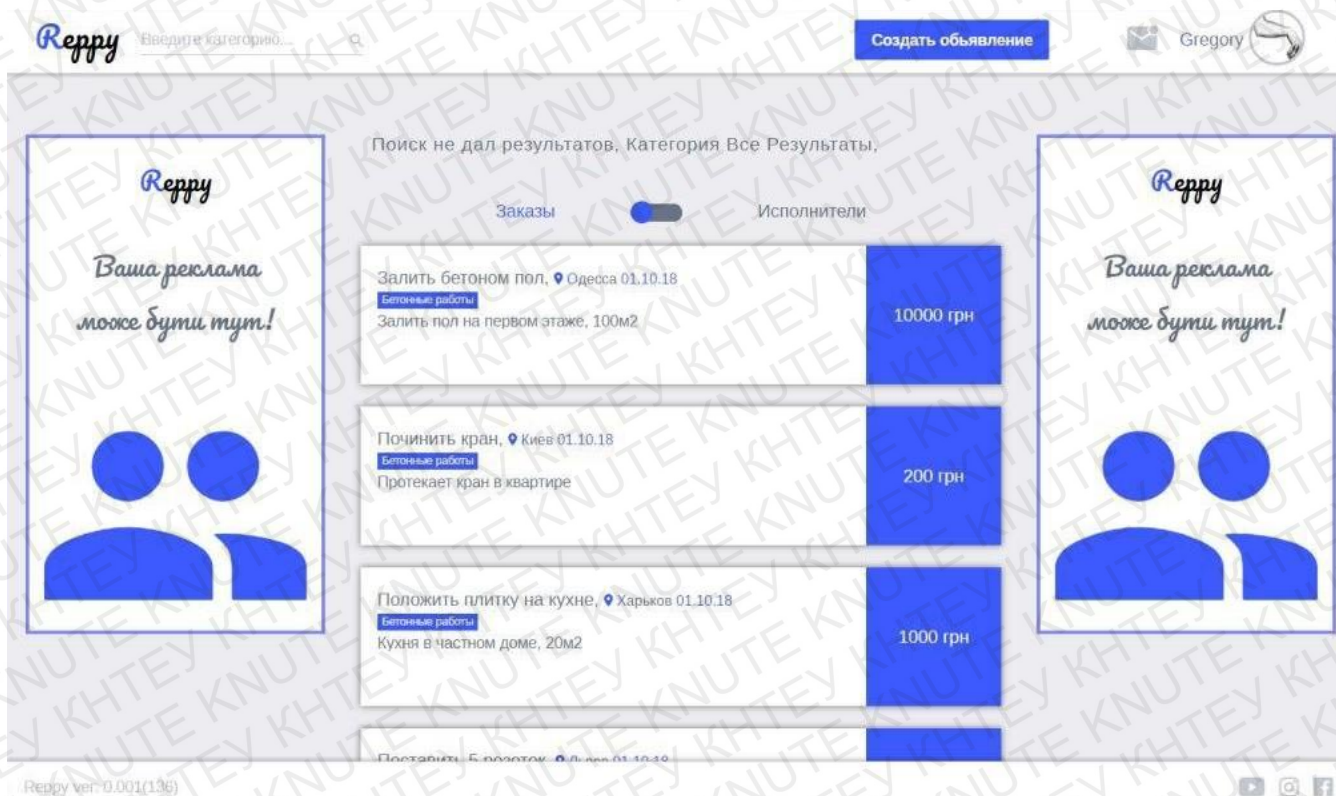


Рисунок 3.16 – Сторінка пошуку

Користувач має можливість шукати по категорії як виконавця, так і замовлення, переключачоючи світч вгорі сторінки.

3.5. Випробування програмного продукту

Інформаційна система проходила ряд тестів. Перевірялись валідність вхідних даних, правильність відпрацювання запитів на сервері, неможливість подати заявку на виконання користувачу, котрий створив замовлення, неможливість подати 2 заявки від одного користувача на одне замовлення та неможливість введення неправильних даних під час реєстрації

Метою випробувань являється перевірка відповідності функцій комплексу задач інформаційної системи пошуку замовників та виконавців послуг вимогам технічного завдання.

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

Система пройшла випробування успішно, працездатність системи не порушена

ВИСНОВОК

В розділі «Загальні Положення» детально описані причини вибору даного предметного середовища, також вказано опис процесу діяльності та функціональної моделі, був проведений огляд та аналіз аналогів існуючих рішень та вказане призначення та основні цілі та задачі розробки.

В розділі «Інформаційне забезпечення» детально описані вхідні та вихідні дані системи та показана структура бази даних. Вони повністю відповідають вимогам технічного завдання, що показує повну відповідність системи до заявлених вимог.

В розділі «Математичне забезпечення» було розглянуто змістовну постановку задачі та математичну постановку задачі. Також було розглянуто декілька альтернативних методів розв'язання. Був обраний, обґрунтований та описаний найбільш підходящий метод.

В розділі «Програмне та технічне забезпечення» було описане програмне забезпеченні інформаційної системи. Були продемонстровані засоби розробки системи, вимоги до технічного забезпечення, загальні вимоги до користування системою. Була детально описана архітектура програмного забезпечення, відображені діаграми класів, послідовностей та компонентів. Була проведена специфікація функцій та опис звітів. Було розписано детальне керівництво користувача з додаванням зображень готової інформаційної системи. Також були описані випробування програмного продукту, їх мета та результати. Система пройшла всі випробування на відмінно, тобто ми можемо сказати, що вона є відмовостійкою.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Web Technologies: A Journey From HTML To Web 3.0. [Електронний ресурс].
– Режим доступу: <https://www.goodcore.co.uk/blog/web-technologies/>
2. Ангуляр [Електронний ресурс] – Режим доступу до ресурсу:
<https://angular.io/docs>.
3. Node.js [Електронний ресурс] – Режим доступу до ресурсу:
<https://nodejs.org/uk/>.
4. Sequelize [Електронний ресурс] – Режим доступу до ресурсу:
<https://metanit.com/web/nodejs/9.1.php>.
5. PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.postgresql.org/>.
6. Переваги ангуляру [Електронний ресурс] – Режим доступу до ресурсу:
<http://fpm.dnu.dp.ua/2019/12/04/angular-i-react-u-chomu-vidminnist/>.
7. Фронтенд і бекенд: у чому різниця? [Електронний ресурс] – Режим доступу до ресурсу:
<https://internetdevels.ua/blog/front-end-development-vs-back-end-development>.
8. JWT [Електронний ресурс] – Режим доступу до ресурсу:
<https://proglib.io/p/json-tokens>.
9. Як використовувати JSON Web Tokens (JWT) для автентифікації [Електронний ресурс] – Режим доступу до ресурсу:
<https://codeguida.com/post/1567>.
10. Кабанчик.юа [Електронний ресурс] – Режим доступу до ресурсу:
<https://kabanchik.ua/>.
11. Ворк.юа [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.work.ua/>.
12. Работа.юа [Електронний ресурс] – Режим доступу до ресурсу:
<https://rabota.ua/ua>.

13. ОЛХ [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.olx.ua/uk/>.
14. All About Web Technologies. [Електронний ресурс]. – Режим доступу:
<https://www.lnssoft.com/all-about-web-technologies/>
15. Робота із браузером Internet Explorer (Mozilla Firefox). [Електронний ресурс]. – Режим доступу: <http://lib.mdpu.org.ua/e-book/oi/pract1.htm>
16. Що таке база даних? | Кафедра АПЕПС ТЕФ КПІ — програмна інженерія. [Електронний ресурс]. – Режим доступу: <http://apeps.kpi.ua/shcho-take-basa-danykh>
17. Веб-програмування | Кафедра АПЕПС ТЕФ КПІ — програмна інженерія. [Електронний ресурс]. – Режим доступу: <http://apeps.kpi.ua/web-programuvania>
18. HTML — мова розмітки гіпертексту. [Електронний ресурс]. – Режим доступу: <http://www.znannya.org/?view=html>
19. JavaScript - Клієнтською мовою програмування, що робить сторінки сайту інтерактивними. [Електронний ресурс]. – Режим доступу:
<https://astwellsoft.com/uk/blog/tehnology/javascript.html>
20. Посібник: знайомство з React. [Електронний ресурс]. – Режим доступу:
<https://uk.reactjs.org/tutorial/tutorial.html>
21. jQuery. [Електронний ресурс]. – Режим доступу:
<https://astwellsoft.com/uk/blog/tehnology/jquery.html>
22. Технологія Ажах і сайти в концепції Web 2.0. [Електронний ресурс]. – Режим доступу: <http://www.znannya.org/?view=AJAX-technology>
23. What is PHP? Write your first PHP Program. [Електронний ресурс]. – Режим доступу: <https://www.guru99.com/what-is-php-first-php-program.html>
24. What is ASP.NET? [Електронний ресурс]. – Режим доступу:
<https://umbraco.com/knowledge-base/aspnet/>
25. Що таке модель? Учасники процесу моделювання. Види моделей та вимоги до них. [Електронний ресурс]. – Режим доступу:
<https://www.kursak.com/shcho-take-model-uchasnyky-protsesu-modeliuvannia/>

26. Системний Аналіз. Розділи 9-16. [Електронний ресурс]. – Режим доступу: <https://core.ac.uk/download/pdf/11320265.pdf>
27. Visual Studio Code - Code Editing. Redefined. [Електронний ресурс]. – Режим доступу: <https://code.visualstudio.com/>
28. Beautiful Soup Documentation. [Електронний ресурс]. – Режим доступу: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>