

**Вакалюк Т.А.**

Державний університет «Житомирська політехніка»

**Болотіна В.В.**

Державний університет «Житомирська політехніка»

**Чижмотря О.Г.**

Державний університет «Житомирська політехніка»

**Чижмотря О.В.**

Державний університет «Житомирська політехніка»

**Генвальдт А.С.**

Державний університет «Житомирська політехніка»

## РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ FREELANCE БІРЖІ

*У статті проаналізовано та досліджено найбільш важливу та актуальну інформацію щодо розробки додатка, а саме основні принципи створення вебдодатків, їх структури та функціональні можливості та взаємодію основних компонентів. Аналіз наявних web-орієнтованих систем freelance біржі дав змогу встановити, що основними рисами розроблюваного продукту мають бути: сучасний мінімалістичний дизайн, наявність зручного інтерфейсу, особистий кабінет, де користувач зможе налаштувати власний профіль, можливість для створення портфоліо виконавців та проектів замовниками. Були розглянуті сучасні технології створення сайтів, які використовуються розробниками у всьому світі. Для збереження створених записів та персональних даних користувачів було вибрано СУБД MySQL, оскільки вона проста у використанні, має практично всі необхідні інструменти, які можуть знадобитися в реалізації будь-якого проекту, є безпечною, масштабованою та швидкою. Під час проектування web-орієнтованої системи було проаналізовано функціональні вимоги до вебсайту, розроблено діаграму сутностей класу та побудовано діаграму використання користувачами додатка. Для реалізації серверної частини вибрано популярний фреймворк Symfony, який реалізує архітектурний шаблон MVC, клієнтська частина реалізована за допомогою javascript-бібліотеки – ReactJS. Були вибрані оптимальні архітектурні шаблони для реалізації системи. Варто зазначити, що взаємодія між бекенд та фронтенд частинами відбувається за допомогою архітектурного стилю – RESTAPI. Під час розробки сайту було проведено тестування функціональної частини та зручності використання системи користувачами. Реалізований програмний продукт готовий для використання користувачами. Надалі систему можливо розширювати, створюючи новий функціонал та покращувати наявний.*

**Ключові слова:** розробка, проектування, web-орієнтована система.

**Постановка проблеми.** Останніми роками Інтернет чинить величезний вплив на розвиток світових компаній, змінюючи засоби ведення бізнесу на світовому ринку праці у режимі онлайн, наприклад, служба технічної підтримки користувачів 24/7, а також створює нові умови конкуренції між світовими лідерами. Натепер із розвитком малого, середнього та великого бізнесу за допомогою всесвітньої мережі економічний стан країни дещо покращився: з'явилися нові робочі місця та посади.

Проаналізувавши зацікавленість користувачів у сфері зайнятості та пасивний дохід від реклами,

сучасний світ створює все більше веб та мобільних додатків, які допомагають так званим «фрілансерам» (особи, які шукають роботу самостійно) та замовникам (ті, хто пропонують роботу) укласти між собою неофіційну угоду про виконання проекту та вчасну виплату узгодженої заздалегідь оплати праці. Об'єднавши критерій масового маркетингу та інтерес людей у отриманні/наданні роботи, ми створюємо потужну вебсистему, яка надає послуги зручного, швидкого та надійного пошуку замовлень для фрілансерів та їх створення безпосередньо роботодавцями.

**Аналіз останніх досліджень і публікацій.** Аналіз наявних web-орієнтованих системи freelance біржі [1] (FL [2], Freelancer [3], Doit.ua [4]) дав змогу встановити, що основними рисами розроблюваного продукту мають бути: сучасний мінімалістичний дизайн, наявність зручного інтерфейсу, особистий кабінет, де користувач зможе налаштувати власний профіль, можливість для створення портфоліо виконавців та проектів замовниками.

**Постановка завдання.** Метою роботи є розроблення вебдодатка для пошуку роботи.

**Виклад основного матеріалу дослідження.** Такий програмний продукт повинен працювати у двох режимах: режим замовника та режим «фрілансера» (вільнонайманця). Отже, у такий спосіб користувачі зможуть як заохочувати нових працівників до проекту та створювати вакансії на посади, так і заробляти, виконуючи відповідні замовлення. Кожен зможе вибрати той режим роботи, який більше до вподоби.

Для розробки бекенд частини вебдодатка був вибраний архітектурний патерн “MVC”, який ідеально підходить для вищезазначеного сценарію написання сайту. Цей шаблон проєктування поділяє такі програми, призначені для користувацького інтерфейсу, та керуючу логіку на три окремі компоненти, а саме: контролер, модель та представлення таким чином, що зміна кожного компонента може здійснюватися незалежно.

Опишемо основні компоненти шаблону проєктування “MVC”:

1. Контролер (Controller) зазвичай виконує роль сполучення між запитом користувача, моделлю та представленням. Саме контролер є точкою входу на сайт. Його основна функція – викликати і координувати дію необхідних ресурсів і об'єктів, потрібних для виконання дій, що задаються користувачем, наприклад, зберегти дані до бази даних чи, навпаки, відправити дані у представлення. Зазвичай контролер викликає відповідну модель для задачі і вибирає відповідний вид.

2. Модель (Model) – це той компонент, який працює з даними. Цей компонент вміє додавати записи в базу даних, видаляти їх, змінювати або ж просто отримувати їх звітти. Завдання моделі – отримати дані і передати їх у такому разі до контролера. Таким чином, контролер може отримувати дані від моделі або ж передавати ці дані в модель.

3. Представлення (View) – частина програми, що забезпечує різні способи представлення даних, які отримані з моделі. Цей компонент може бути шаблоном, який заповнюється даними. Може бути

кілька різних представлень, і контролер вибирає, який підходить якнайкраще для поточної ситуації.

Окрім шаблону проєктування “MVC” для виконання відповідних задач та написання якісного та розширюваного програмного продукту, було вибрано ще декілька патернів, а саме: «Одинак» (Singleton), «Наслідування з єдиною таблицею» (SingleTableInheritance), «Репозиторій» (Repository).

Варто зазначити, що бекенд та фронтенд частини вебдодатка працюють окремо, особливо коли розміщені під різними доменами. Тому для їх коректної взаємодії між собою використовується архітектура “REST”. “REST” (Representational state transfer) – це стиль архітектури програмного забезпечення для розподілених систем, таких як World Wide Web. У загальному випадку “REST” є дуже простим інтерфейсом передачі інформації у тому вигляді, у якому вона представлена.

Для створення web-орієнтованої системи було вибрано такі програмні продукти та середовища розробки: PHP фреймворк Symfony 5.1, JavaScript бібліотека React, фреймворк для роботи з api – APIPlatform, середовище розробки PhpStorm 2019.3, середовище розробки IntelliJIDEA 2019.3, середовище тестування REST, SOAP та GraphQL запитів Postman.

Однією з найголовніших цілей створення web-орієнтованої системи freelance-біржі є насамперед залучення нових користувачів всесвітньої мережі до працевлаштування у режимі онлайн. Постає завдання створити зручний та максимально зрозумілий для користування сервіс, який би надав певний аналітичний функціонал для статистики прибутку для фрілансерів та замовників, а також кількість зайнятих у роботі користувачів. Таким чином, реалізувавши такий вебдодаток, зможемо дослідити зацікавленість людей у залученні до праці на біржах фрілансу.

Визначимо низку вимог до такого програмного продукту. *Бізнес-вимоги:*

1. Основні цілі: проєкт створюється з метою долучення людей до працевлаштування через Інтернет.

2. Можливості: однією з можливостей web-орієнтованої системи є аналітика прибутку та виконаних робіт за певний проміжок часу для фрілансера та замовника, а також статистика для адміністрації сайту, наприклад, яка кількість нових зареєстрованих користувачів та скільки фрілансерів отримали замовлення.

3. Представлення проєкту: проєкт буде реалізовано у вигляді вебсайту, який має зручний та привабливий інтерфейс, а також містить необхідні відомості з цієї тематики.

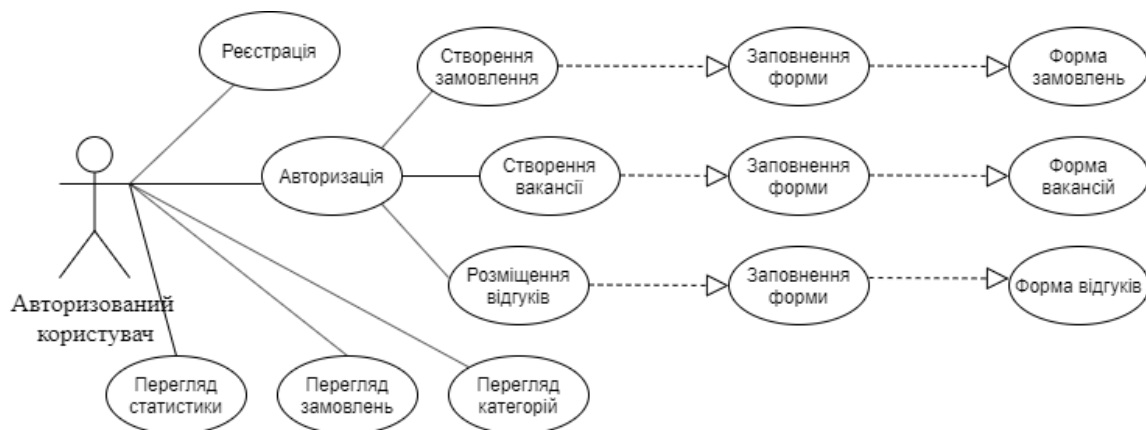


Рис. 1. Варіанти використання користувачем web-орієнтованої системи freelance біржі

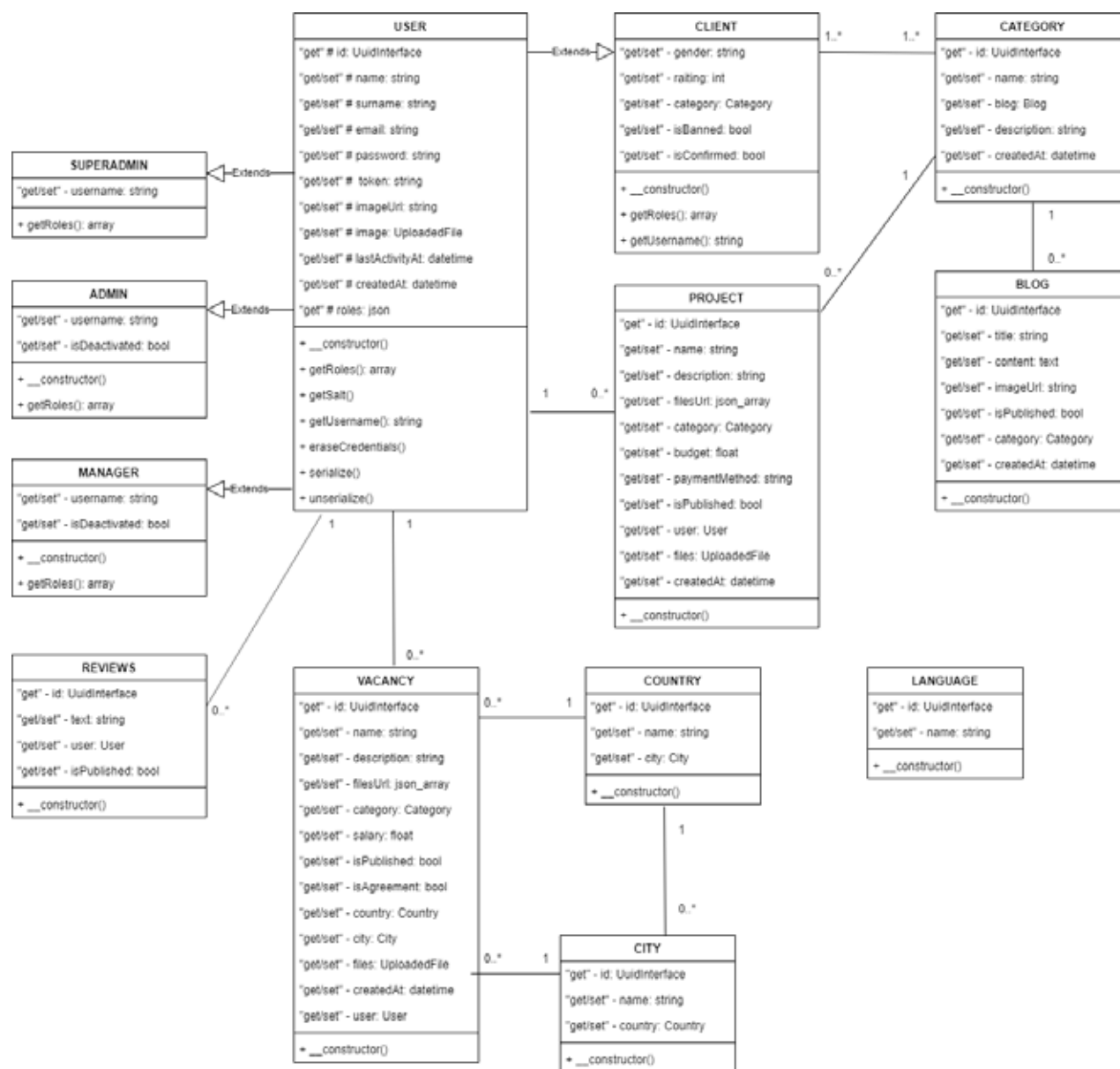


Рис. 2. Діаграма класів

### *Вимоги до дій користувачів:*

1. Можливість реєстрації на сайті користувачів з підтвердженням електронної пошти через унікальне посилання.
2. Можливість авторизації на сайті та доступ до особистої «адміністративної» панелі.
3. Можливість створення замовлення, а також редагування та видалення запису (доступ лише у тому разі, коли користувач авторизований як замовник).
4. Можливість створення вакансії, а також редагування та видалення запису (доступ лише у тому разі, коли користувач авторизований як замовник).
5. Можливість зміни пароллю та персонального фото профілю.
6. Можливість пошуку даних, фільтрація та сортування.
7. Можливість перегляду рейтингу фрілансера для користувачів та відвідувачів сайту.
8. Можливість перегляду статистики виконаних замовлень за відповідний проміжок часу.

### *Функціональні вимоги:*

1. Використання web-орієнтованої системи: у додатку повинна бути представлена можливість створення замовлень та вакансій замовниками, а також відгук на ці записи фрілансерами. Такий функціонал є однією з найважливіших реалізацій сайту біржі.
2. Розміщення даних: розміщення замовлень та вакансій відбувається безпосередньо за участю менеджера-модератора, який після перевірки коректності даних розміщує записи для публічного перегляду користувачами додатка.
3. Оплата за виконану роботу: оплата замовлення відбувається безпосередньо за угодою між замовником та фрілансером.

Аналіз функціональних вимог дав змогу виділити такі сутності, що забезпечать реалізацію програмного комплексу системи. На рис. 2 наведено діаграму класів, які використовуються користувачами web-орієнтованої системи freelance біржі.

За результатами UML діаграми можна виділити класи, які спроектовані за допомогою патерну «Наслідування з єдиною таблицею». Така архітектура реалізує одну з трьох основних парадигм ООП, а саме поліморфізм. Поліморфізм – це здатність об'єкта використовувати методи похідного класу, який не існує на момент створення базового.

У такому разі абстрактний клас User – це клас-шаблон. Він реалізує функціональність тільки на тому рівні, на якому вона відома на поточний момент. Похідні ж класи її доповнюють.

На діаграмі класів (рис. 2) зображені такі сутності:

– User – абстрактний клас, який реалізує логіку авторизованих користувачів вебдодатка. Клас наслідує інтерфейс Serializable для авторизації користувача у системі та збереження всіх даних у сесії сайту. Також абстрактний клас User наслідує та перевизначає п'ять основних методів інтерфейсу UserInterface, які необхідні для визначення користувача у системі та збереження особистих даних користувачів від потенційних зловмисників:

– getUsername() – такий метод повинен повернути будь-яке призначене для користувача унікальне поле, наприклад, адресу електронної пошти, логін чи ідентифікатор типу uuid. Функціонал реалізовано тільки для того, щоб показати розробнику додатка, хто увійшов у систему у разі налагодження;

– getRoles() – повертає роль авторизованого користувача у вигляді масиву даних;

– getPassword(), getSalt(), eraseCredentials() – реалізація таких методів необхідна тільки в тому разі, якщо додаток несе особисту відповідальність за зберігання і перевірку користувальницьких паролів;

– Superadmin – клас реалізований для збереження особистих даних користувачів, які мають найвищі привілеї над усіма ролями додатка. Такий користувач має доступ до всіх даних web-орієнтованої системи;

– Admin – зберігає дані адміністраторів додатка, які мають права суперадміністратора. Винятком є те, що користувач не має можливості реєструвати у системі адміністраторів;

– Manager – клас зберігає дані менеджерів сайту, які виконують модерацию даних на сайті та публікують узгоджені з правилами web-орієнтованої системи записи. Це стосується даних, які будуть відображатися на сторінках сайту, які не мають обмежень у разі перегляду контексту;

– Client – такий клас реалізований для збереження даних про зареєстрованого користувача вебдодатка. Користувач має можливість створювати замовлення та вакансії у тому разі, якщо він авторизований як замовник, у іншому разі брати замовлення у роботу;

– Category – зберігає інформацію про категорії, наприклад, дизайн інтер'єру чи мобільну розробку, за якими створюються замовлення, вакансії та статті. Такий функціонал створений для зручної навігації по сайту та швидкого пошуку необхідної інформації користувачами вебдодатка;

- Blog – клас зберігає статті вебдодатка, які пишуть користувачі сайту за відповідною тематикою, за якою спеціалізуються та яку вибрали під час реєстрації на сайті. Після модерації менеджером пост блогу відображається разом з іншими статтями;
- Project – клас реалізує збереження даних про створені замовлення користувачами або здебільшого адміністрацією сайту. Запис містить назву, детальний опис про роботу, яку потрібно виконати разом з файлами (якщо наявні), вартість, спосіб оплати та автора проєкту;
- Vacansy – такий клас зберігає інформацію про вакансії, які розміщують на сайті замовники для пошуку нових співробітників. Вакансія містить детальний опис, запропоновану заробітну плату та інформацію про країну та місто, у якому проживає потенційний працівник;
- Country – містить інформацію про всі країни світу;
- City – такий клас зберігає назву міста та ідентифікатор країни, де розташований;
- Reviews – зберігає відгуки користувачів на сайті.

Таким чином, така система реалізує функціонал web-орієнтованої системи freelance біржі.

Серверна частина вебдодатка зберігає додані файли до відповідних замовлень та вакансій, а також зображення профілю зареєстрованих у системі користувачів. Файли зберігаються у відповідну директорию, яку розробник створює на початку проєктування. Для вакансій – це каталог “vacansy”, замовлення – “project”, а для зображення про-

філю – “profile”. Перед завантаженням файлів створюється підкаталог з унікальним іменем. У цьому разі береться унікальний ідентифікатор замовлення чи вакансії. Таким чином, час, витрачений на пошук конкретних файлів, значно мінімізується.

Вебдодаток, як і більшість подібних сайтів, розділений на загальнодоступну (клієнтську) та адміністративну частини. Адміністративна частина сайту є окремим його складником, вхід у який можливий тільки користувачам з відведеним доступом. По суті, це додатковий сайт зі своїм дизайном та функціоналом. Завдання до адміністративної частини ставляться, виходячи з функціоналу сайту.

Кожен співробітник вебсайту має відведений обліковий запис та роль. Нижче наведений детальний опис користувачів адміністративної панелі:

- Суперадміністратор є керівником над усіма співробітниками. Він не має обмежень щодо функціональності на сайті. Загалом такий користувач слідкує за порядком на сайті та реєструє адміністраторів. Але також може створювати замовлення та вакансії, банити клієнта та опублікувати статті.

- Адміністратор є заступником суперадміністратора. Як і керівник, він не має обмежень. Адміністратор може реєструвати менеджерів, слідкувати за статистикою і т. д.

- Менеджер-модератор має більш обмежені права ніж керівництво. Головним завданням співробітника є модерація замовлень, вакансій та постів блогу.

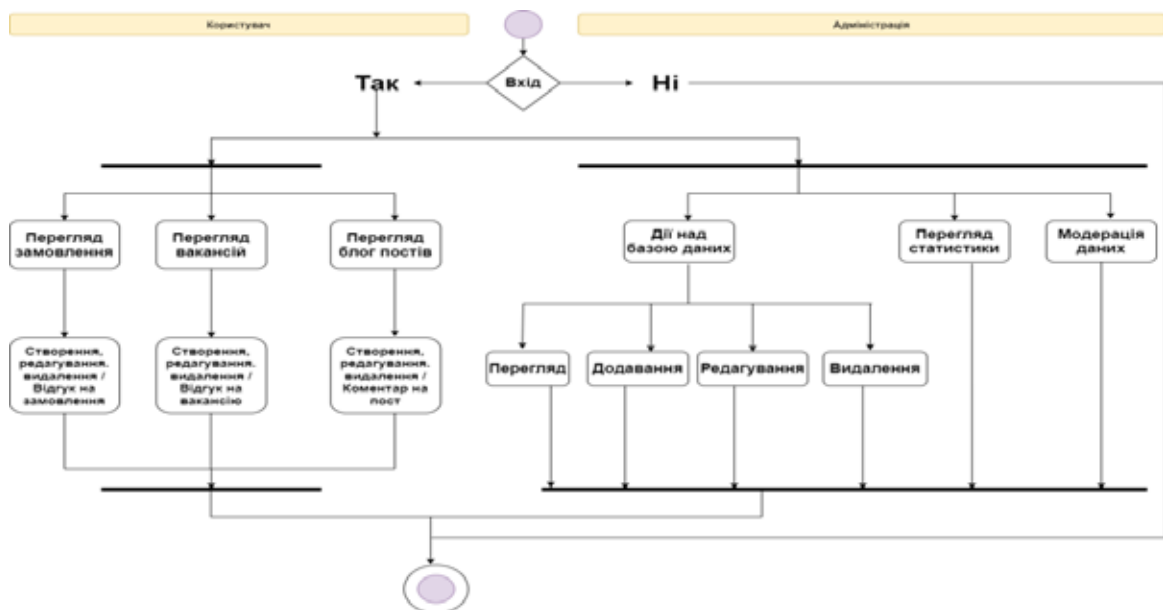


Рис. 3. Діаграма активності роботи програми

Після того як користувач потрапив на сайт (рис. 3), він має можливість продивлятися загальнодоступну інформацію, а саме: проекти для виконання замовниками, вакансії та статті блогу, які пишуть фрілансери за індивідуальною сферою спеціалізації. Функціонал створення відповідної інформації доступний лише для авторизованих користувачів, отже, наступним кроком є реєстрація на сайті. Після підтвердження електронної пошти та авторизації користувачу доступні вищеперелічені функції, а також редагування інформації особистого профілю та зміна паролю.

Для того щоб створити запис, користувач повинен заповнити відповідну форму та зберегти дані. Якщо вхідні дані були введені некоректно, то система видасть відповідну помилку для користувача.

Варто зазначити, що після створення замовлення, вакансії чи статті запис не відразу відображається для перегляду іншим. Для того щоб опублікувати інформацію, вона повинна пройти модерацию адміністрацією вебдодатка. Менеджер-модератор авторизується у системі та перевіряє текст. У разі якщо запис відповідає вимогам системи, то співробітник натискає на спеціально відведену кнопку й інформація автоматично з'являється на загальнодоступних сторінках вебсайту.

Для більш детального пошуку роботи супер-адміністратори та адміністратори можуть створювати категорії спеціалізації різних напрямів, наприклад, вебдизайн або переклад текстів з іноземних мов. Після того як вони будуть опубліковані для перегляду, на головній сторінці з'являється список усіх категорій, які є посиланнями на сторінку з усіма проектами від замовників. Фрілансер може вибрати собі роботу та домовитися персонально з власником щодо оплати праці та більш детальних рекомендацій.

Список вакансій, як і блог постів, розміщується на окремих сторінках сайту. На таких сторінках є фільтр по категоріях для зручного пошуку за спеціальностями, а також можливість продивлятися увесь список. Варто зазначити, що у разі першого перегляду даних для користувача відображаються тільки ті записи, які відповідають вибраній спеціалізації під час реєстрації на сайті. Це може бути як одна категорія, так і декілька. Також присутня пагінація для коректного відображення та зручної навігації.

Для користувачів у особистому кабінеті є окрема сторінка зі статистикою за певний період. Авторизований користувач може переглядати, наскільки його рейтинг покращився або погіршився, скільки замовлень було отримано і виконано.

Керівництво системи має можливість слідкувати за статистикою всього проекту для аналізу зацікавленості користувачів та прибутку. На сторінці продемонстровано кількість зареєстрованих користувачів за певний період, кількість нових замовлень та вакансій, найпопулярніші спеціалізації, а також скільки фрілансерів відгукнулися на проекти.

На сайті присутня можливість створювати відгуки для авторизованих користувачів. Клієнт може висловити власну думку щодо додатка та сервісу обслуговування. Коментар виводиться на сторінку тільки після модерації. Також можливо зазначити, що адміністрація не може створювати відгуки на сайт.

На рис. 4 та рис 5. зображені діаграми послідовності для варіантів використання web-орієнтованої системи freelance біржі.

Структура інтерфейсу та порядок роботи з web-орієнтованою системою freelance біржі

У разі переходу на сайт за відповідним URL з'являється домашня сторінка додатка. На рис. 6 наведено зображення головної сторінки з вибором відповідного пункту меню: «Головна», «Вакансії», «Блог», «Про нас», «Увійти», які є посиланнями. Неавторизований користувач має змогу переглядати замовлення, вакансії та статті блогу, але не може створювати персональні записи та писати коментарі.

Для того щоб почати повноцінну роботу з додатком, користувачеві, який вперше увійшов на сайт, потрібно зареєструватися. Перед тим як зареєструвати сайт пошуковими системами, адміністрація повинна наповнити динамічні записи вебдодатка для подальшого користування. Система автоматично створює обліковий запис для супер-адміністратора, основним завданням якого є створення адміністраторів. Для цього на адміністративній панелі у пункті «Адміністратори» потрібно натиснути на кнопку «Зареєструвати адміністратора». Відкриється форма, де потрібно ввести дані та натиснути на «Створити». Варто зазначити, перед полями вводу у формі присутній текст, який коротко описує як зареєструвати адміністратора. Після успішної операції надсилається лист на вказану електронну пошту з логіном та паролем, а у списку з'являються дані співробітників.

На кожній сторінці адміністративної панелі є можливість фільтрувати дані, таким чином пошук серед багатьох даних спрощується. У тому разі, якщо адміністратор більше не працює у компанії, його можна деактивувати, а після певного часу обліковий запис видаляється автоматично системою. Якщо під час створення адміністратора

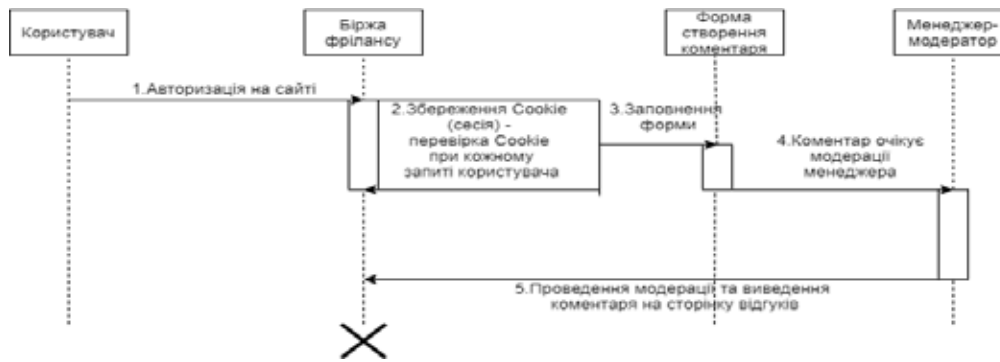


Рис. 4. Алгоритм роботи створення відгуку на сайт



Рис. 5. Алгоритм створення вакансії на сайті

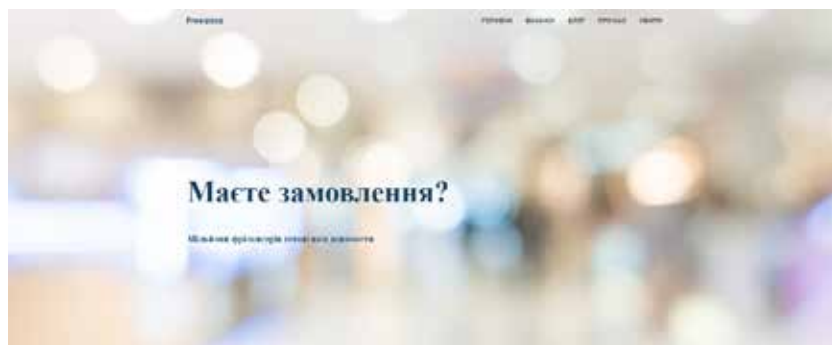


Рис. 6. Головна сторінка сайту

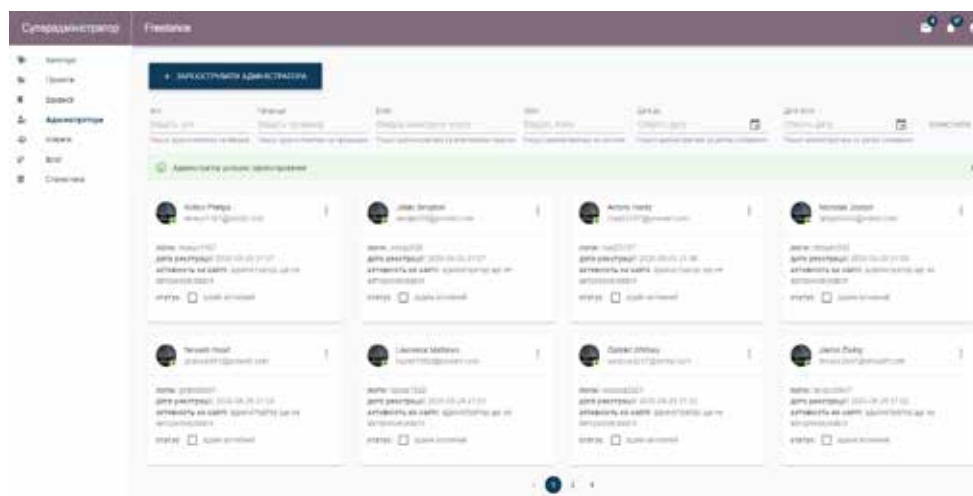


Рис. 7. Список усіх адміністраторів системи з пагінацією

суперадмін припускається помилки, він може у пункті меню вибрати «Редагувати адміністратора» та виправити помилку у формі (рис. 8).

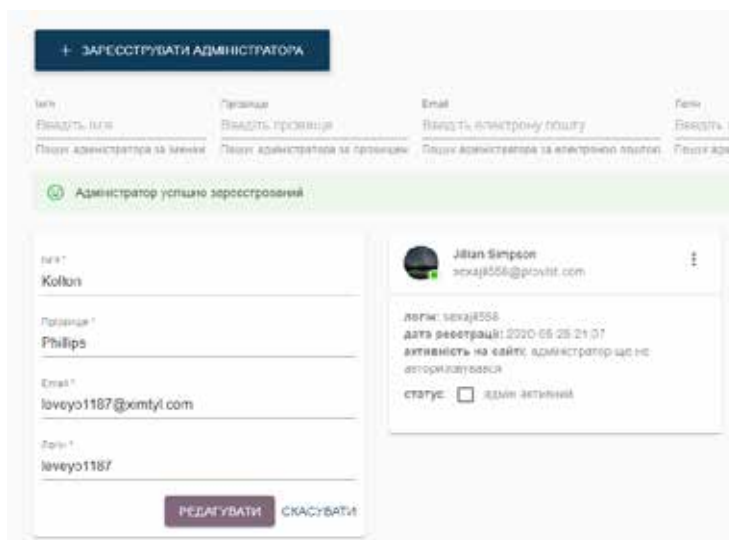
Перед тим як створювати нові замовлення, вакансії та статті блогу потрібно створити декілька категорій, з якими будуть працювати користувачі системи. Створення категорії відбувається суперадміном або адміном, менеджери-модератори не мають такої можливості. Для створення спеціалізацій потрібно натиснути на посилання «Категорії» у пункті меню панелі. Натиснувши на кнопку «Створити категорію», з'являється форма, де потрібно ввести назву та опис до спеціалізації (рис. 9). Після того як співробітник системи ввів усі дані, необхідно натиснути на кнопку «Створити», а у разі якщо потрібно закрити форму без збереження даних – «Скасувати». Якщо під час створення дані не коректні, виведеться повідомлення з помилкою.

Після створення всіх категорій користувач може створити замовлення або вакансію. У адмі-

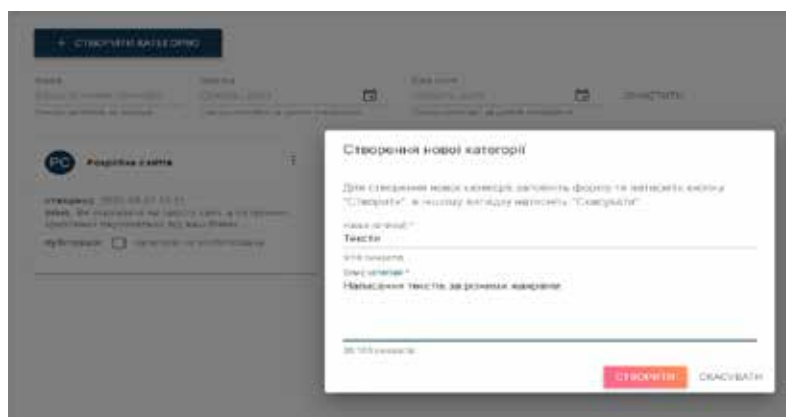
ністративній панелі для створення проекту замовник повинен перейти за посиланням «Проекти».

Після натискання кнопки «Створити запис» у новому вікні відкривається форма. Користувачу потрібно заповнити всі обов'язкові поля, а саме: назву проекту, опис, спеціальність (категорія), бюджет та спосіб оплати (з банківської карти або незахищена угода).

Під полями вводу назви та опису є підказка у вигляді максимальної кількості символів та кількість введених. Файли не є обов'язковими для заповнення, але за бажанням можна додати для більш детальної інформації для фрілансера. У разі, якщо дані не є коректними, наприклад, файл не відповідає одному з форматів, прописаних системою або занадто коротка назва для проекту, система видасть повідомлення про помилку і не збереже дані у базі даних. Якщо потрібно повернутися до списку проектів, на сторінці є посилання у вигляді стрілки.



**Рис. 8.** Форма редагування адміністратора



**Рис. 9.** Форма створення категорій





Рис. 10. Сторінка проєктів

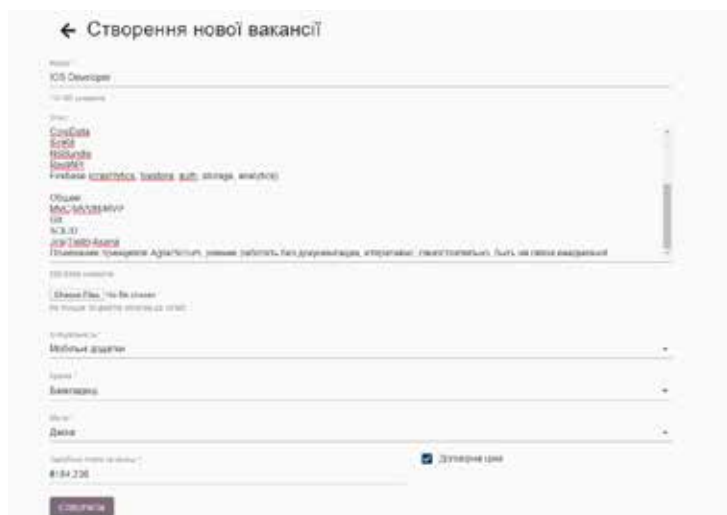


Рис. 11. Створення вакансії користувачем

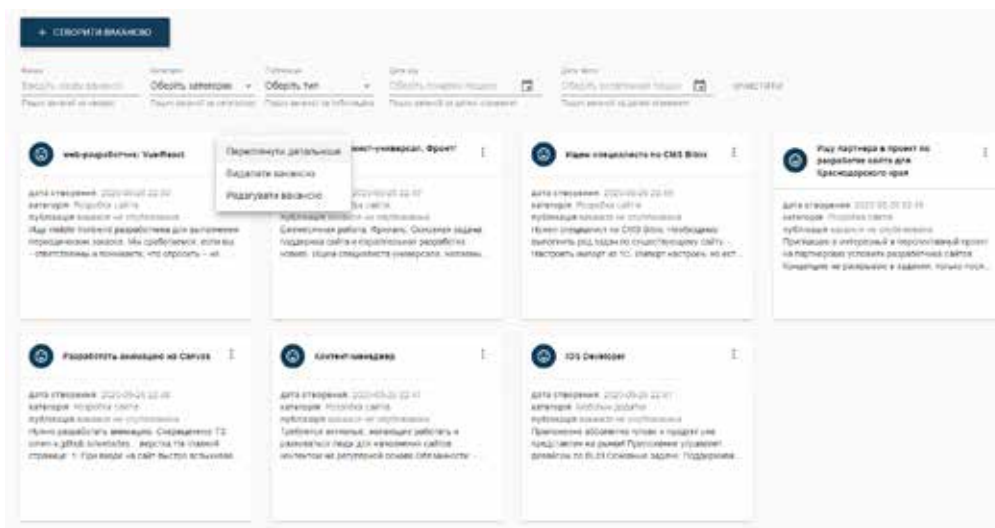


Рис. 12. Список вакансій

Надалі створений проєкт перевіряє менеджер-модератор та опубліковує для перегляду, якщо запис відповідає нормам системи. Для користувача доступний функціонал детального перегляду, видалення та редагування. Під час перегляду запису відкривається нова сторінка. Якщо до запису додані

файли, їх можна завантажити. Проєкти відображаються у вигляді списку. Для зручності додана пагінація. Проєкти можна сортувати за назвою, категорією, публікацією та датою створення.

Створення нової вакансії дещо ідентичне до створення проєкту. Однак користувачу обов'язково

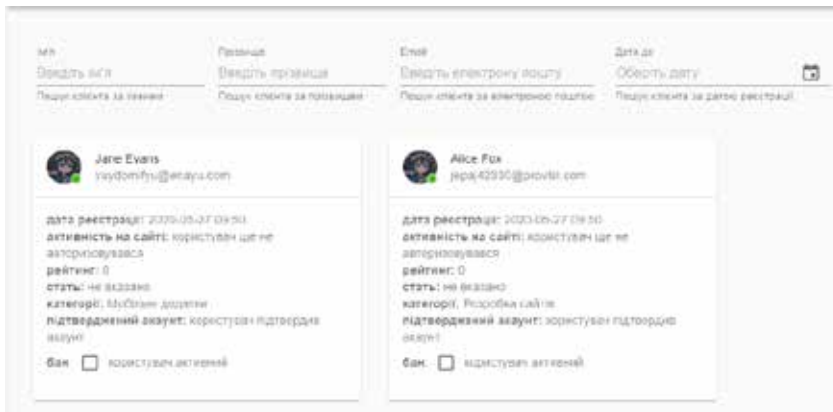


Рис. 13. Список зареєстрованих клієнтів у системі

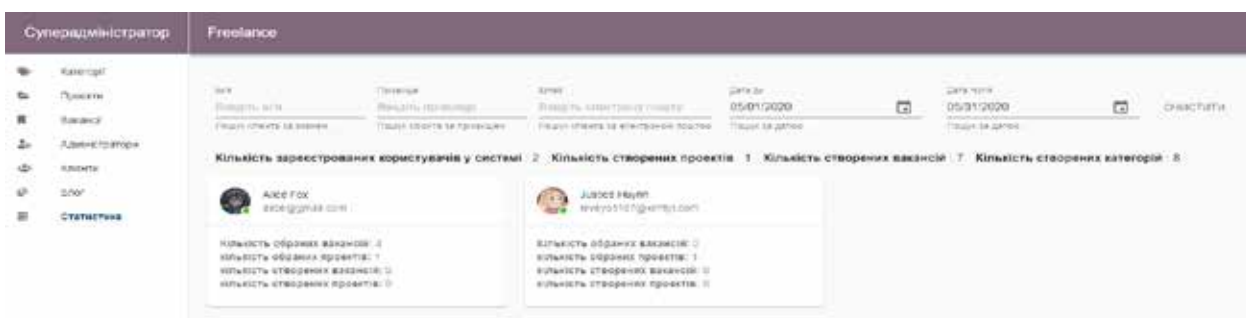


Рис. 14. Сторінка зі статистикою

потрібно вказати місто та країну, а замість «бюджету» виступає «заробітна плата», яка може бути договірною.

Під час створення нового запису система видає користувачу відповідне повідомлення. Усі вакансії виводяться у списку (рис. 12). Як і проекти, інформація про вакансію не є доступною для перегляду, оскільки повинна пройти модерацію. На сторінці присутня пагінація та фільтрування даних. Записи можна переглядати, видаляти та редагувати.

Усі зареєстровані у системі клієнти у вигляді списку відображаються на окремій сторінці (рис. 13).

Для перегляду загальнодоступних проектів на головній сторінці є посилання за категоріями. Далі відкривається сторінка зі списком проектів за вибраною тематикою. Назва запису є посиланням на сторінку з детальною інформацією. Для того щоб узяти проект у роботу, потрібно натиснути на відповідну кнопку у деталях проекту.

Адміністрація сайту має можливість продивлятися статистику за певний період. На панелі є відповідне меню. На сторінці відображена кількість зареєстрованих користувачів та створених записів, а також інформація за відповідним клієн-

том (рис. 14). Присутня фільтрація для зручного пошуку та пагінація.

**Висновки.** Проаналізовано та досліджено найбільш важливу та актуальну інформацію щодо розробки додатка, а саме основні принципи створення вебдодатків, їх структури та функціональні можливості та взаємодії основних компонентів. Під час проєктування web-орієнтованої системи було проаналізовано функціональні вимоги до вебсайту, розроблено діаграму сутностей класу та побудовано діаграму використання користувачами додатка. Для реалізації серверної частини вибрано популярний фреймворк Symfony, який реалізує архітектурний шаблон MVC, клієнтська частина реалізована за допомогою javascript-бібліотеки – ReactJS. Під час розробки сайту проведено тестування функціональної частини та зручності використання системи користувачами. Усі виявлені недоліки були усунені, система працює коректно та виконує поставлені на початку розробки завдання.

Реалізований програмний продукт готовий для використання користувачами. Надалі систему можна розширювати, створюючи новий функціонал, та покращувати наявний.

**Список літератури:**

1. Вакалюк Т.А., Болотіна В.В., Байлюк Є.М., Покотило О.А., Генвальдт А.С. Аналіз наявних web-орієнтованих систем freelance біржі. The 11th International scientific and practical conference “*Scientific achievements of modern society*” (June 24–26, 2020). Cognum Publishing House, Liverpool, United Kingdom. 2020. С. 206–212.
2. Фриланс – сайт удаленной работы. URL: <https://www.fl.ru/>.
3. Фриланс. URL: [www.freelancer.com.ua](http://www.freelancer.com.ua).
4. Услуги фрилансеров. URL: [Doit.ua](http://Doit.ua).
5. Генвальдт А.С., Болотіна В.В. Розробка web-орієнтованої системи freelance біржі. Тези доповідей XI Міжнародної науково-технічної конференції «*Інформаційно-комп'ютерні технології – 2020 (ІКТ-2020)*», м. Житомир, 09–11 квітня 2020 р. Житомир : Житомирська політехніка, 2020. С. 6–7.

**Vakaliuk T.A., Bolotina V.V., Chyzhmotria O.G., Chyzhmotria O.V., Henvaldt A.S.**

**DEVELOPMENT OF A FREELANCE EXCHANGE INFORMATION SYSTEM**

*The article analyzes and explores the most important and relevant information on application development, namely: the basic principles of creating web applications, their structure and functionality, and the interaction of the main components. Analysis of the existing web-oriented freelance exchange system revealed that the main features of the developed product should be: modern minimalist design, user-friendly interface, the personal account where the user can customize their profile, the ability to create a portfolio of performers and projects by customers. Modern technologies for creating sites used by developers around the world were considered. MySQL was chosen to save the created records and personal data of users, because it is easy to use, has almost all the necessary tools that may be needed in the implementation of any project, is secure, scalable, and fast. During the design of the web-oriented system, the functional requirements for the website were analyzed, a diagram of the essence of the class was developed and a diagram of the use of the application by users was built. The popular Symfony framework, which implements the MVC architectural template, was chosen to implement the server part, and the client part was implemented using the javascript library – ReactJS. The optimal architectural templates for the system implementation were selected. It is worth noting that the interaction between the backend and frontend parts is through the architectural style – RESTAPI. During the development of the site, the functional part and usability of the system were tested by users. The implemented software product is ready for use by users. In the future, the system can be expanded, creating new functionality, and improving existing ones.*

**Key words:** development, design, web-oriented system.