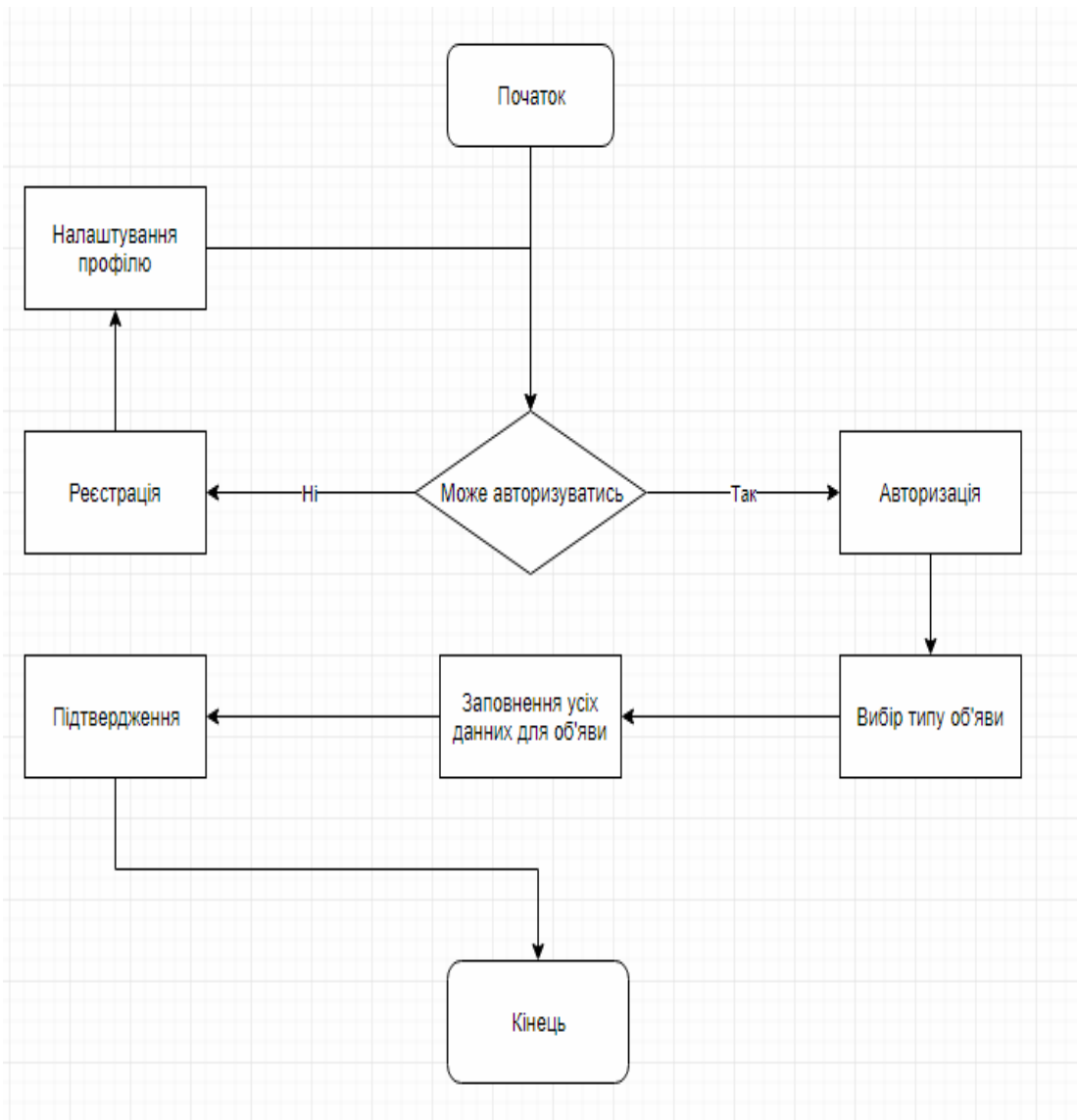


Додаток А

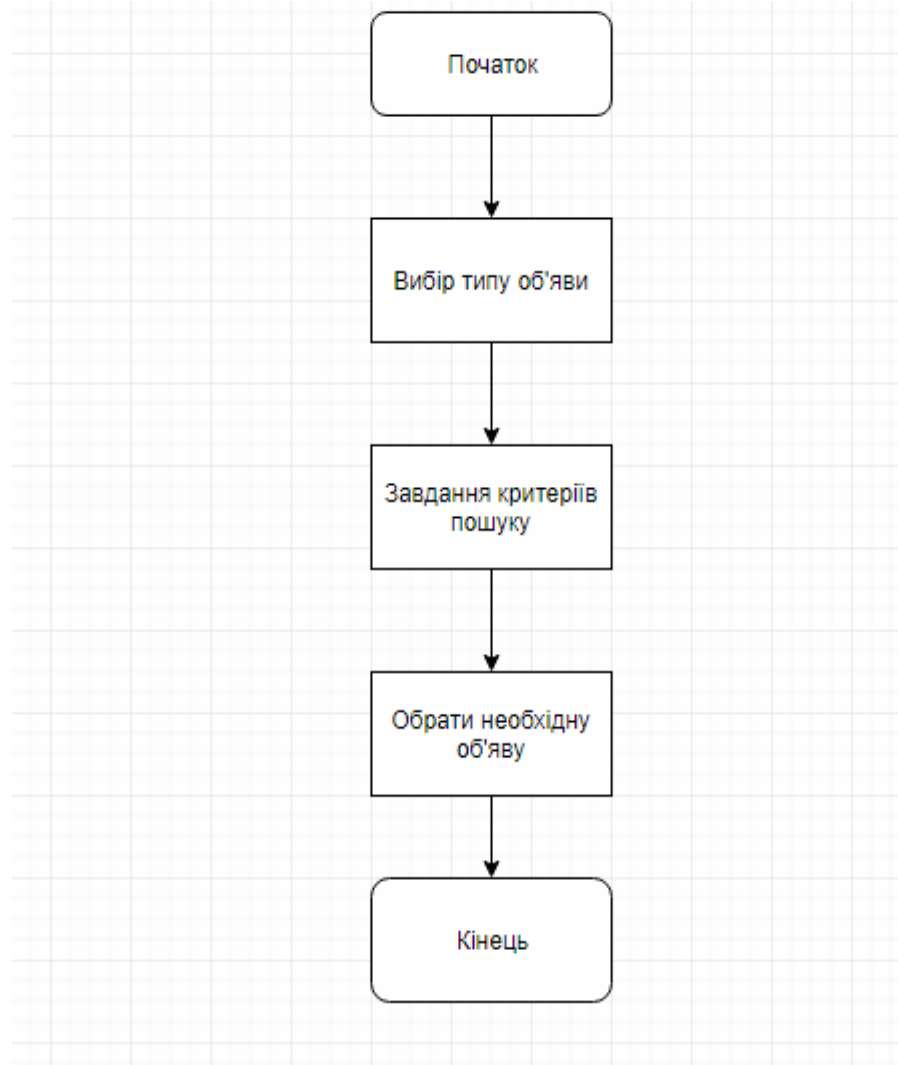
«Графічні матеріали атестаційної роботи»

Схема алгоритму створення об'яви



Розроб.	Образков В. Є.			Розробка та дослідження рекомендаційної системи для сервісу з підбору нерухомості	
Перевір.	Імангулова З.А.				
Н. Контр.	Імангулова З.А.				
				ІТПМ-18-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

Схема алгоритму пошуку об'яв



Розроб.	Образков В. Є.			Розробка та дослідження рекомендаційної системи для сервісу з підбору нерухомості	
Перевір.	Імангулова З.А.				
Н. Контр.	Імангулова З.А.				
				ІТПМ-18-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

Схема алгоритму роботи РС



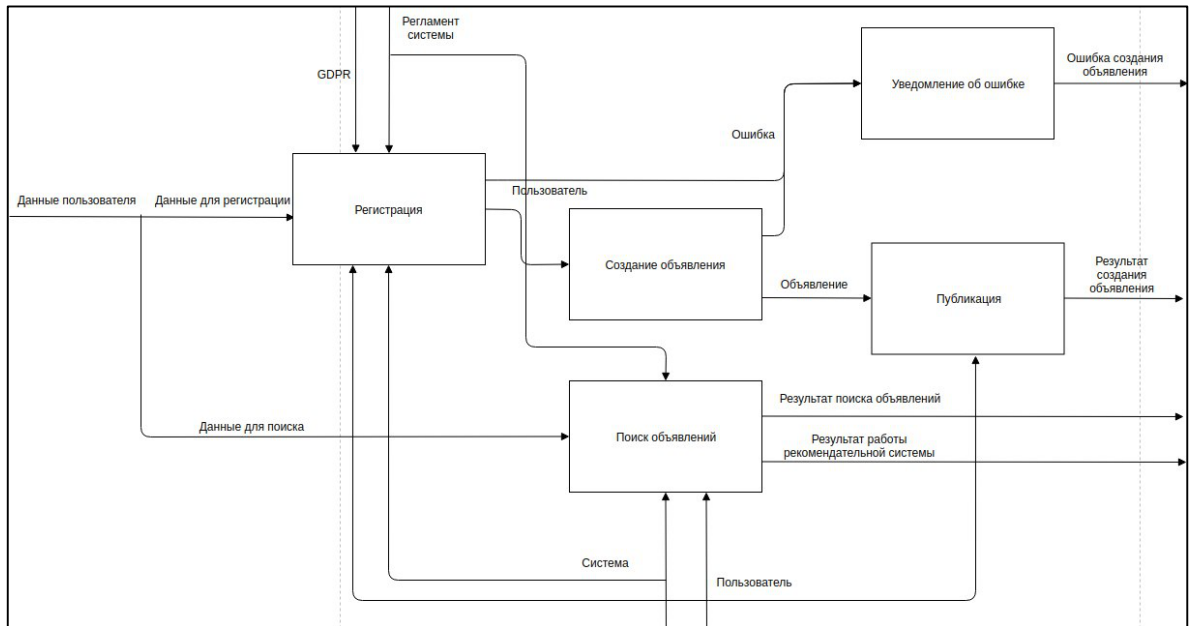
Розроб.	Образков В. Є.			Розробка та дослідження рекомендаційної системи для сервісу з підбору нерухомості	
Перевір.	Імангулова З.А.				
Н. Контр.	Імангулова З.А.				
				ІТПм-18-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

Концептуальна діаграма інформаційної системи



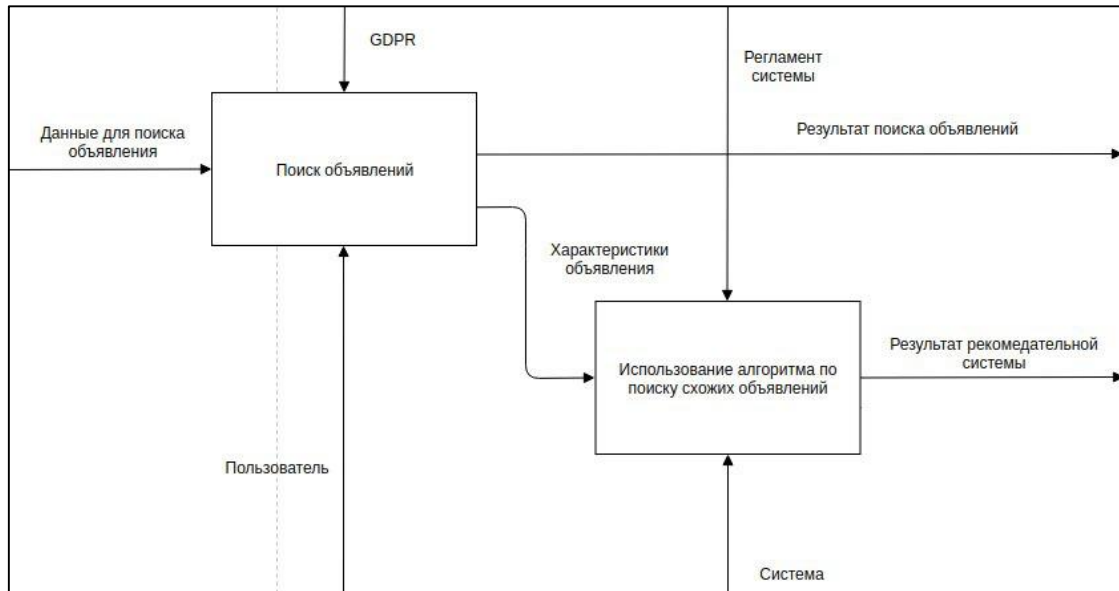
Розроб.	Образков В. Є.			Розробка та дослідження рекомендаційної системи для сервісу з підбору нерухомості	
Перевір.	Імангулова З.А.				
Н. Контр.	Імангулова З.А.				
				ІТПм-18-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

Діаграма декомпозиції функції «Система публікації оголошень з пошуку-продажу нерухомості»



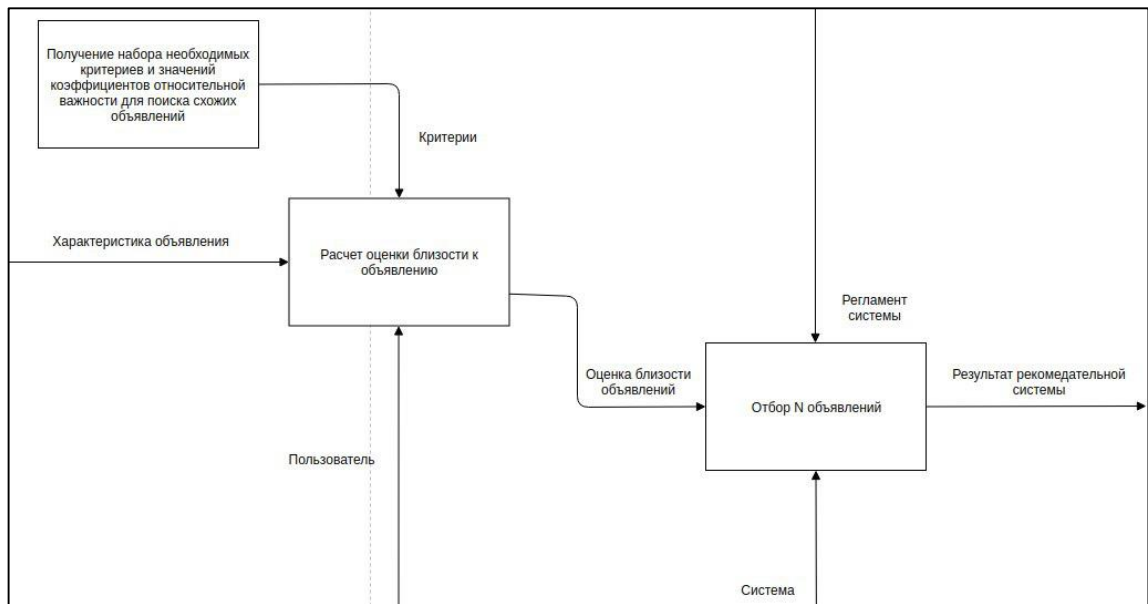
Розроб.	Образков В. Є.			Розробка та дослідження рекомендаційної системи для сервісу з підбору нерухомості	
Перевір.	Імангулова З.А.				
Н. Контр.	Імангулова З.А.				
				ІТПм-18-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

Діаграма декомпозиції функції «Пошук оголошень»




Розроб.	Образков В. Є.			Розробка та дослідження рекомендаційної системи для сервісу з підбору нерухомості	
Перевір.	Імангулова З.А.				
Н. Контр.	Імангулова З.А.				
				ІТПМ-18-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1


Діаграма декомпозиції функції «Застосування рекомендаційної системи»



Розроб.	Образков В. Є.			Розробка та дослідження рекомендаційної системи для сервісу з підбору нерухомості	
Перевір.	Імангулова З.А.				
Н. Контр.	Імангулова З.А.				
				ІТПм-18-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

Знімок екрану «Активне оголошення та рекомендації до нього»





Дубина Василій
Телефонний номер : 095-329-12-21
Почта : vasya.dubina82@gmail.com
[Перейти к профилю](#)

Сдам современную 2ком.кв


Современная квартира в пешей доступности от метро Алексеевка. Комфортная планировка с отдельными комнатами на две стороны. Шкафы для одежды в обеих комнатах. Для верхней одежды в прихожей панель с крючками при входе и раздвижной шкаф. Бытовая техника вся есть. Большой холодильник, водонагреватель, стиральная машина. Дом в парковой зоне, рядом лес и всегда свежий воздух. Квартира не сдавалась. Ждём

Характеристики

Местоположение: Харьков, Дзержинский район, Людвиг Свободы	Почтовый ящик: vasya.dubina82@gmail.com
Телефон: 095-329-12-21	Этаж: 6
Тип недвижимости: Квартира	Площадь: 45м²
Тип объявления: Аренда долгосрочная	Рядом с метро: +
Добавлено: 59 минут назад	Предоплата: -
Евроремонт: +	Ежемесячная оплата: 8500 грн.
Количество комнат: 2	
Рейтинг : 96%	

Вас также может заинтересовать


Добавлено: 58 минут назад



Сдам 2-ком кв

Адрес: г. Харьков, Дзержинский район, Победы
Цена: 9000 грн. Рейтинг : 94%


Добавлено: 1 час 47 минут назад



Сдам 2-х комн.

Адрес: г. Харьков, Дзержинский район, Победы
Цена: 8000 грн. Рейтинг : 94%

Добавлено: 5 часов назад



Срочно сдам квартиру на Алексеевке.

Адрес: г. Харьков, Дзержинский район, ул. Целиноградская
Цена: 8000 грн. Рейтинг : 92%

2019 © Недвижимость в Харькове "Харьков-Риэлтер"

Розроб.	Образков В. Є.			Розробка та дослідження рекомендаційної системи для сервісу з підбору нерухомості	
Перевір.	Імангулова З.А.				
Н. Контр.	Імангулова З.А.				
				ІТПм-18-1	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

№	Позначення				Найменування	Додаткові Відомості	
					Текстові документи		
1.	ГЮИК.506500.006 ПЗ				Пояснювальна записка	115 стор.	
2.	ГЮИК.506500.006 - 01 12 01				Текст програми	21 стор.	
3.	ГЮИК.506500.006 - 34 01 – ИЗ				Посібник користувача	11 стор.	
					Графічні документи		
4.					Схема алгоритму створення об'яви	1 аркуш	
5.					Схема алгоритму пошуку об'яв	1 аркуш	
6.					Схема алгоритму роботи рекомендаційної системи	1 аркуш	
7.					Концептуальна діаграма ІС	1 аркуш	
8.					Діаграма декомпозиції функції «Система публікації оголошень з пошуку-продажу нерухомості»	1 аркуш	
9.					Діаграма декомпозиції функції «Пошук оголошень»	1 аркуш	
10.					Діаграма декомпозиції функції «Використання РС»	1 аркуш	
11.					Знімок екрану «Активне оголошення та рекомендації до нього»	1 аркуш	
					ГЮИК.506500.006 ДЗ		
Змін	Арк	№ докум	Підп.	Дата			
<i>Розроб.</i>		<i>Образков В. Є.</i>			<i>Розробка та дослідження рекомендаційної системи для сервісу з підбору нерухомості</i>	Аркуш	Аркушів
<i>Перевір.</i>		<i>Імангулова З.А.</i>				1	1
<i>Н.контр.</i>		<i>Імангулова З.А.</i>				<i>ХНУРЕ</i>	
<i>Затв.</i>		<i>Гребеннік І. В.</i>				<i>Кафедра СТ</i>	

Додаток Б

«Посібник користувача»

ЗАТВЕРДЖЕНО

ГЮИК. 506500.006 ИЗ – ЛЗ

РОЗРОБКА ТА ДОСЛІДЖЕННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЛЯ
СЕРВІСУ З ПІДБОРУ НЕРУХОМОСТІ

Посібник користувача

ГЮИК. 506500.006 ИЗ

Аркушів 7

2019

Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»

Керівник атестаційної роботи,
доц. Імангулова З.А.

РОЗРОБКА ТА ДОСЛІДЖЕННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЛЯ
СЕРВІСУ З ПІДБОРУ НЕРУХОМОСТІ

Посібник користувача

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК. 506500.006 ИЗ – ЛЗ

РОЗРОБИВ:

Ст. гр.ІТПм-18-1

Образков В. Є.

ЗМІСТ

ВСТУП	87
Б.1 Підготовка до роботи.....	88
Б.2 Призначення та умови застосування.....	88
Б.3 Опис операцій	88
Б.4 Аварійні ситуації.....	93
Б.5 Рекомендації з освоєння	93

ВСТУП

Одним з результатів виконання атестаційної роботи є розроблений web-додаток для створення та пошуку об'яв з продажу та оренди нерухомості. Він може використовуватись для надання кінцевому користувачеві змоги розміщувати або проглядати об'яви для пошуку нерухомості.

В можливості системи входять наступні функції:

- реєстрація нових користувачів;
- перевірка введених даних;
- створення нових оголошень за типом нерухомості;
- редагування оголошень;
- написання відгуків у оголошенні;
- редагування профілю користувача;
- рейтинг у профілю;
- відображення місцезнаходження у оголошенні;
- відправка повідомлення на пошту користувачеві;
- фільтрація оголошень за різними критеріями;
- надання рекомендацій користувачеві;
- візуалізація статистики оголошень у вигляді графіків.

Перед початком роботи з системою, необхідно ознайомитися з керівництвом користувача, щоб повністю зрозуміти, як використовувати ту чи іншу функцію.

Б.1 ПІДГОТОВКА ДО РОБОТИ

Для початку роботи із клієнтським додатком необхідний комп'ютер під управлінням будь-якої операційної системи. Для роботи клієнтського додатка необхідне з'єднання з мережею інтернет. Користувач клієнтського додатку має бути ознайомлений з правилами користування програмою, а також з інструкцією користувача.

Б.2 ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ

Кожен користувач має можливість:

- шукати та переглядати оголошення;
- використовувати конвертор валют;
- переглядати статистику об'яв;
- зареєструватися;
- отримувати рекомендації, щодо шуканих оголошень.

Кожен авторизований користувач має можливості звичайного відвідувача, а також:

- створення нових об'яв;
- редагування оголошень;
- налаштування власного профілю;
- написання відгуків до об'яв.

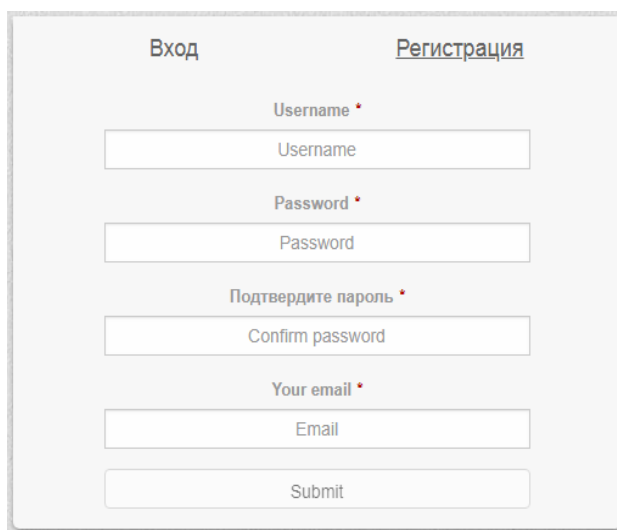
Додаток є веб-сайтом, тому щоб ним користуватись потрібно мати підключення до інтернету.

Б.3 ОПИС ОПЕРАЦІЙ

В даному розділі описується покрокове керівництво про те, як користуватися веб-сайтом і як використовувати всі доступні функції.

Для початку роботи користувач повинен зареєструватися в системі.

Для цього користувач повинен пройти реєстрацію на сайті при вході в систему. Форма реєстрації наведена на рисунку Б.1.

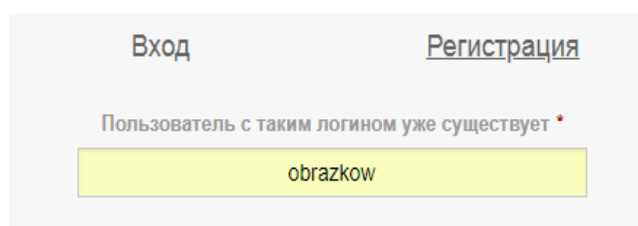


The image shows a registration form with two tabs: 'Вход' (Login) and 'Регистрация' (Registration). The 'Регистрация' tab is active. The form contains the following fields and labels:

- Username * (input field)
- Password * (input field)
- Подтвердите пароль * (Confirm password) (input field)
- Your email * (input field)
- Submit (button)

Рисунок Б.1 – Реєстрації користувача

У тому випадку, якщо користувач вводить некоректні дані при реєстрації, наприклад, якщо такий логін вже існує, то система демонструє користувачеві повідомлення з описом неправильних дій, які він зробив. Повідомлення про помилку зображено на рисунку Б.2.



The image shows the same registration form as in Figure B.1, but with an error message displayed. The error message is: 'Пользователь с таким логином уже существует *' (User with this login already exists *). Below the error message, the text 'obrazkow' is displayed in a yellow box, indicating the username that caused the error.

Рисунок Б.2 – Невірна поштова адреса

Якщо користувач вже має аккаунт, то він повинен авторизуватись, натиснувши на кнопку «Вход», зображену на рисунку Б.3.

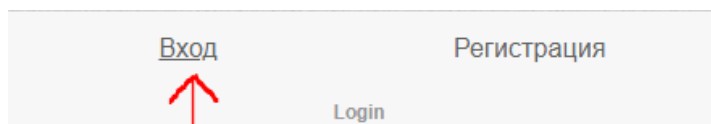


Рисунок Б.3 – Кнопка перехода на форму авторизації

Після натискання на кнопку «Вход» користувач потрапляє на форму авторизації, яка зображена на рисунку Б.4.

Рисунок Б.4 – Форма авторизації

Якщо користувач авторизується, то він потрапить на сторінку свого профілю (рисунок Б. 5).

Рентер Главная Продажа - Аренда - Обмен - О нас obrazkow - **НОВОЕ ОБЪЯВЛЕНИЕ +**

obrazkow

Имя: Владислав
 Телефон: 0955858604
 Рейтинг: 1

Фамилия: Образков
 Почта: Obraz_Vlad96@ukr.net
 Редактировать профиль

Удалить Обновить

Вперед →

Добавлено: 20 Июле в 7:29	Добавлено: 17 Июле в 12:28	Добавлено: 16 Июле в 11:04
Продажа	Обменяю квартиру	Сдам в аренду участок земли
Адрес: г. Харьков, Московский район, пер. Автострадный	Адрес: г. Харьков, Московский район, наб. Автострадная	Адрес: г. Харьков, Коминтерновский район, ул. Анапская
Тел.: 0955858604 Недвижимость: Гараж	Тел.: 0955858604 Недвижимость: Квартира	Тел.: 0955858604 Недвижимость: Участок
Объявление: Продажа	Объявление: Обмен	Объявление: Аренда
Цена: 25000 грн. Рейтинг: 87.5%	Цена: 20000 грн. Рейтинг: 37.5%	Цена: 3000 грн. Рейтинг: 37.5%

Рисунок Б.5 – Профіль користувача

Для додавання нового оголошення користувач повинен бути авторизованим та натиснути на кнопку «Новое оголошення», яка зображена на рисунку Б.6, заповнити та зберегти форму, яка зображена на рисунку Б.7.

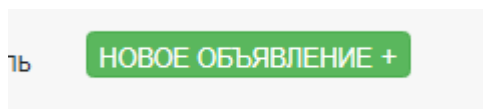


Рисунок Б.6 – Кнопка «Новое оголошення»

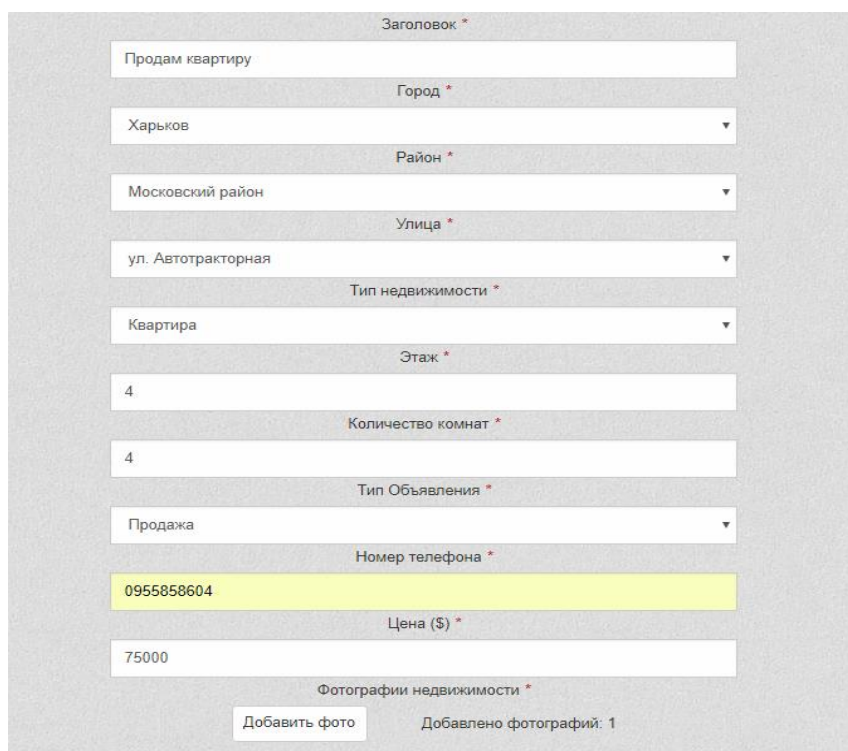


Рисунок Б.7 – Форма створення об'яви

Для пошуку об'яви про продаж квартири необхідно перейти у розділ «Продажа» та вибрати «Подобрать квартиру» (рисунок Б. 8).

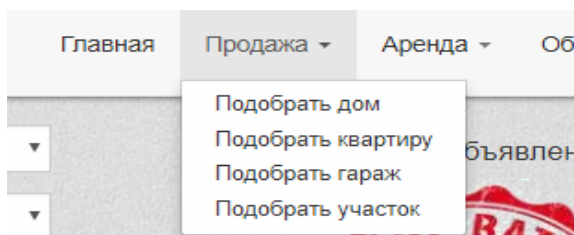


Рисунок Б.8 – Розділ продаж

Для більш детального пошуку можливо використовувати фільтри, які зображені на рисунку Б. 9

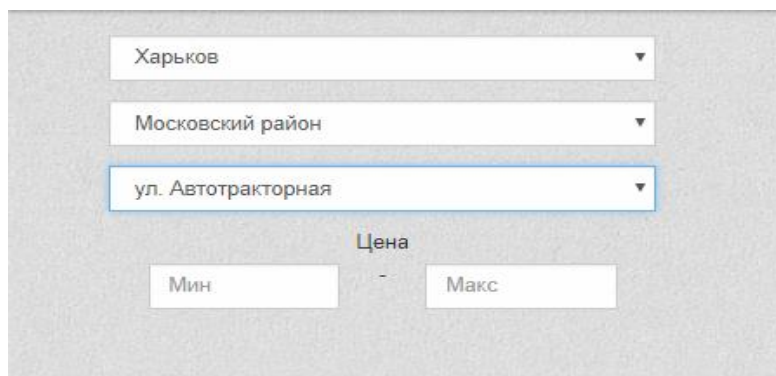


Рисунок Б.9 – Фільтри пошуку об'яв

Якщо перейти на об'яву, то можливо отримати рекомендаційну панель, яка зображена на рисунку Б. 10



Рисунок Б.10 – Рекомендаційна панель

Б.4 АВАРІЙНІ СИТУАЦІЇ

Аварійна ситуація можлива при виході з ладу апаратного забезпечення, яке використовується для роботи клієнтського додатку. Для вирішення проблеми необхідно замінити елемент причину непрацездатності.

Б.5 РЕКОМЕНДАЦІЇ ПО ОСВОЄННЮ

Для успішної взаємодії користувача з системою потрібно мати навички роботи з браузерними додатками.

Додаток В

«Текст програми»

Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»

Керівник атестаційної роботи,
доц. Імангулова З.А.

РОЗРОБКА ТА ДОСЛІДЖЕННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЛЯ
СЕРВІСУ З ПІДБОРУ НЕРУХОМОСТІ

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.506500.006 – 01 12 01 – ЛЗ

РОЗРОБИВ:

Ст. гр. ІТІм-18-1

Образков В. Є.

2019

ЗАТВЕРДЖЕНО

ГЮИК. 506500.006 - 01 12 01 – ЛЗ

РОЗРОБКА ТА ДОСЛІДЖЕННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЛЯ
СЕРВІСУ З ПІДБОРУ НЕРУХОМОСТІ

Текст програми

ГЮИК. 506500.006 - 01 12 01 – ЛЗ

Аркушів 17

ЗМІСТ

V.1 Обробка даних для об'яви.....	98
V.2 Видалення об'яви.....	99
V.3 Клас, у який завантажуються критерій.....	99
V.4 Клас для оперування з критеріями типу «дата».....	100
V.5 Клас для оперування з критеріями типу «число».....	101
V.6 Побудова об'єкту user з даних при реєстрації.....	101
V.7 Метод для відправки листа з підтвердженням на пошту.....	102
V.8 Метод для підтвердження електронної адреси.....	103
V.9 Метод для перевірки на можливість.....	104
V.10 Метод для додавання рейтингу.....	104
V.11 Метод для оновлення даних профілю.....	105
V.12 Клас для обробки картинок.....	106
V.13 Метод для завантаження фото об'яви.....	108
V.14 Метод для пошуку останніх об'яв у бд.....	109
V.15 Метод для отримання об'яв за заданими параметрами.....	109
V.16 Встановлення результату критеріїв користувача.....	111
V.17 Конфігураційний файл для критеріїв.....	111
V.18 Метод для додавання даних у відповідь користувачу, якщо він успішно авторизувався.....	112
V.19 Клас для видалення користувачей, які не підтвердили електронну адресу.....	112
V.20 Створення запиту для отримання рекомендацій.....	113

В.1 ОБРОБКА ДАНИХ ДЛЯ ОБ'ЯВИ

```

public function build(Request $request, Ad $ad) : Ad
{
    $token = $this->container-
>get('security.token_storage')->getToken();
    if(!$token instanceof JWTUserToken) {
        throw new InvalidTokenException('Invalid JWT');
    }

    $this->data = $request->request->all();
    $user = $token->getUser();
    $photos = $request->files->get('photos');

    $ad->setUser($user);
    $ad->setCity($this->em->getRepository('AppBundle:City')-
>find($this->data['city']));
    $ad->setArea($this->em->getRepository('AppBundle:Area')-
>find($this->data['area']));
    $ad->setStreet($this->em-
>getRepository('AppBundle:Street')->find($this-
>data['street']));
    $ad->setTypeProperty($this->em-
>getRepository('AppBundle:TypeProperty')->find($this-
>data['typeProperty']));
    $ad->setNumberOfStoreys(isset($this-
>data['numberOfStoreys']) ? $this->data['numberOfStoreys'] :
null);
    $ad->setPhone($this->data['phone']);
    $ad->setName($this->data['name']);
    $ad->setMail($user->getMail());
    $ad->setCost((float)$this->data['cost']);
    $ad->setTypeOffer($this->data['typeOffer']);
    $ad->setTerm(isset($this->data['term']) ? $this-
>data['term'] : null);
    $ad->setSpace($this->data['space'] != '' ? $this-
>data['space'] : null);
    $ad->setFloor(isset($this->data['floor']) ? $this-
>data['floor'] : null );
    $ad->setRooms(isset($this->data['rooms']) ? $this-
>data['rooms'] : null );
    $ad->setInfo($this->data['info'] != '' ? $this-
>data['info'] : null);
    $ad->setPrepayment($this->data['prepayment'] == 'false'
? 0 : 1);
    $ad->setMetro($this->data['metro'] == 'false' ? 0 : 1);
    $ad->setRenovation($this->data['renovation'] == 'false'
? 0 : 1);
    foreach ($photos as $photo) {
        $this->container->get('app.photo_manager')-
>savePhotoAd($photo, $ad);
    }
}

```

```

        $ad->calculateScore($this->container-
>get('app.rules_config'));

        return $ad;
    }

```

В. 2 ВИДАЛЕННЯ ОБ'ЯВИ

```

public function delete($id)
{
    $fileDeleter = $this->container-
>get('app.file_deleter');
    $ad = $this->em->getRepository('AppBundle:Ad')-
>find($id);
    $photos = $this->em->getRepository('AppBundle:PhotoAd')-
>findBy(['Ad' => $ad]);
    if(!$ad){
        throw new NotFoundHttpException('Ad not found');
    }
    if(!$photos){
        throw new NotFoundHttpException('Photos of Ad
'."$id"." not found');
    }

    foreach ($photos as $photo){
        $fileDeleter->deletePhotoAd($photo->getFilename());
        $this->em->remove($photo);
    }
    $this->em->remove($ad);
    $this->em->flush();
}

public function validate(Ad $ad)
{
    $validator = $this->container->get('validator');
    $errors = $validator->validate($ad);

    if(count($errors) > 0){
        return $errors;
    }

    return null;
}

```

В. 3 КЛАС, У ЯКИЙ ЗАВАНТАЖУЮТЬСЯ КРИТЕРІЙ

```

class ConfigRule
{
    private $config;
}

```

```

public function __construct(array $config)
{
    $this->config = $config;
}

public function get($path)
{
    $keys = explode('.', $path);

    $data = $this->config;
    foreach ($keys as $key) {

        if (isset($data[$key])) {
            $data = $data[$key];
        } else {
            return null;
        }

    }
    return $data;
}
}

```

В. 4 КЛАС ДЛЯ ОПЕРУВАННЯ З КРИТЕРІЯМИ ТИПУ «ДАТА»

```

class DateRule
{
    protected $config;
    protected $data;

    public function __construct(array $config)
    {
        $this->config = $config;
    }

    public function setData(array $data)
    {
        $this->data = $data;
    }

    public function calculate()
    {
        $calculatedData = [];
        foreach ($this->data as $id => $date) {
            $from = $this->config['from'] ?? time();
            foreach ($this->config['dates'] as $seconds =>
$value) {
                if ($date <= $from && $date > ($from -
$seconds)) {
                    $calculatedData[$id] = $value;
                    break;
                }
            }
        }
    }
}

```

```

    }
    if (isset($this->config['default'])) {
        $calculatedData[$id] = $this->
>config['default'];
    }
}

return $calculatedData;
}
}

```

В.5 КЛАС ДЛЯ ОПЕРУВАННЯ З КРИТЕРІЯМИ ТИПУ «ЧИСЛО»

```

class NumberRule
{
    protected $config;

    public function __construct(ConfigRule $config)
    {
        $this->config = $config;
    }

    public function calculate($data, $path)
    {
        $config = $this->config->get($path);
        if (!is_array($config) || !isset($config['numbers'],
$>config['default'])) {
            throw new \Exception();
        }

        foreach ($config['numbers'] as $number => $value) {
            if ($data < $number) {
                return $value;
            }
        }
        return $config['default'] ?? 0;
    }
}

```

В.6 ПОБУДОВА ОБ'ЄКТУ USER З ДАНИХ ПРИ РЕГІСТРАЦІЇ

```

public function build($data) : User
{
    $isExist = $this->em->getRepository('AppBundle:User')->
>findOneByUsername($data['login']);
    if($isExist){
        throw new HttpException(400, 'Пользователь с таким
ЛОГИНОМ уже существует');
    }
}

```

```

    }

    $encoder = $this->container-
>get('security.password_encoder');
    if(mb_strlen($data['password']) < 6 ||
mb_strlen($data['password']) > 16){
        throw new Exception();
    }
    $password = $encoder->encodePassword($this->user,
$data['password']);
    $this->user->setUsername($data['login']);
    $this->user->setPassword($password);
    $this->user->setMail($data['mail']);
    $this->user->setStatus(false);

    return $this->user;
}

```

В.7 МЕТОД ДЛЯ ВІДПРАВКИ ЛИСТА З ПІДТВЕРДЖЕННЯМ НА ПОШТУ

```

public function sendConfirmationMail(User $user)
{
    $mailer = $this->container->get('app.mailer');
    $twig = $this->container->get('templating');

    $month = ['Января', 'Февраля', 'Марта', 'Апреля', 'Мая',
'Июня',
        'Июля', 'Августа', 'Сентября', 'Октября', 'Ноября',
'Декабря'];
    $date = date('g') . ' ' . $month[date('n')-1] . ' в
'.date('h:m');
    $datePart = null;
    if(date('G')*60 + date('i') >= 405 && date('G')*60 +
date('i') < 720 ) {
        $datePart = 'Доброе утро';
    }
    if(date('G')*60 + date('i') >= 720 && date('G')*60 +
date('i') < 1080 ) {
        $datePart = 'Добрый день';
    }
    if(date('G')*60 + date('i') >= 1080 && date('G')*60 +
date('i') <= 1439 ) {
        $datePart = 'Добрый вечер';
    }
    if(date('G')*60 + date('i') >= 0 && date('G')*60 +
date('i') < 405 ) {
        $datePart = 'Доброй ночи';
    }
}

```

```

        $link = $this->em-
>getRepository('AppBundle:ConfirmRegistration')-
>findOneByUser($user);
        if(!$link instanceof ConfirmRegistration){
            return new Response(
                new Exception(),
                Response::HTTP_BAD_REQUEST,
                ['content-type', 'application/json']
            );
        }

        $body = $twig-
>render('AppBundle:Emails:confirmRegistration.html.twig', array(
            'name' => $user->getName(),
            'lastName' => $user->getLastName(),
            'link' => $link->getHash(),
            'date' => $date,
            'datePart' => $datePart,
        ));

        $mailer->buildMessage('Подтверждение регистрации',
        $user->getEmail(), $body)
            ->send();

        return null;
    }

```

В.8 МЕТОД ДЛЯ ПІДТВЕРДЖЕННЯ ЕЛЕКТРОННОЇ АДРЕСИ

```

public function confirmRegistration($hash):bool
{
    $verify = $this->em-
>getRepository('AppBundle:ConfirmRegistration')-
>findOneByHash($hash);

    if($verify instanceof ConfirmRegistration){
        $user = $verify->getUser();
        $user->setStatus(true);
        $this->em->remove($verify);

        $config = $this->container->get('app.rules_config');
        $profileScore = $this->em-
>getRepository(ProfileScores::class)->entityByUser($user);
        $profileScore->setProfileEmailScore($config-
>get('profile.email'));
        $this->em->getRepository(Ad::class)-
>updateProfileScore($user, $profileScore->getAll());
        $this->em->persist($profileScore);
        $this->em->flush();

        return true;
    }
}

```

```

    }
    return false;
}

```

B.9 МЕТОД ДЛЯ ПЕРЕВІРКИ НА МОЖЛИВІСТЬ

```

public function IsCanVote($username, User $user) : bool
{
    $userProfile = $this->em-
>getRepository('AppBundle:User')->findOneBy(['username' =>
$username]);
    if(!$userProfile instanceof User){
        return false;
    }

    $reputation = $this->em-
>getRepository('AppBundle:UserVotes')->findOneBy(['userId' =>
$userProfile->getId()]);
    if(!$reputation){
        return true;
    }

    if(in_array(['from' => $user->getId()], $reputation-
>getVotedUsers()) === true){
        return false;
    }

    return true;
}

```

B.10 МЕТОД ДЛЯ ДОДАВАННЯ РЕЙТИНГУ

```

public function addVote($username, $type)
{
    $fromUserId = $this->container-
>get('security.token_storage')->getToken()->getUser()->getId();
    $user = $this->em->getRepository('AppBundle:User')-
>findOneBy(['username' => $username]);
    if(!$user instanceof User){
        throw new UserNotFound();
    }

    $reputation = $this->em-
>getRepository('AppBundle:UserVotes')->findOneBy(['userId' =>
$user->getId()]);
    if(!$reputation) {
        $reputation = new UserVotes();
        $reputation->setUserId($user);
    } else if(!$reputation->canUserVote($fromUserId)) {

```

```

        return false;
    }

    $reputation->addVotedUsers(['from' => $fromUserId]);
    $user->setReputation($type);

    $rule = $this->container->get('app.rule_number');
    $profileScore = $this->em-
>getRepository(ProfileScores::class)->entityByUser($user);
    $profileScore->setProfileReputationScore($rule-
>calculate($user->getReputation(), 'profile.reputation'));
    $this->em->getRepository(Ad::class)-
>updateProfileScore($user, $profileScore->getAll());
    $this->em->persist($profileScore);
    $this->em->persist($reputation);
    $this->em->flush();

    return true;
}

```

V.11 МЕТОД ДЛЯ ОНОВЛЕННЯ ДАНИХ ПРОФІЛЮ

```

public function profileUpdate(User $user, Request $data)
{
    $config = $this->container->get('app.rules_config');
    $profileScore = $this->em-
>getRepository(ProfileScores::class)->entityByUser($user);

    $user
        ->setLastName($data->get('lastName') ?? null)
        ->setName($data->get('firstName') ?? null)
        ->setPhone($data->get('phone') ?? null);

    if ($user->getLastName() !== NULL && strlen($user-
>getLastName()) &&
        $user->getName() !== NULL && strlen($user-
>getName())
        ) {
        $profileScore->setProfileAboutUsScore($config-
>get('profile.aboutUs'));
    } else {
        $profileScore->setProfileAboutUsScore(0);
    }
    if ($user->getPhone() !== NULL && strlen($user-
>getPhone())) {
        $profileScore->setProfilePhoneScore($config-
>get('profile.phone'));
    } else {
        $profileScore->setProfilePhoneScore(0);
    }
}

```

```

    $this->em->persist($profileScore);
    $this->em->persist($user);
    $this->em->getRepository(Ad::class)-
>updateProfileScore($user, $profileScore->getAll());
    $this->em->flush();
}

```

B.12 КЛАС ДЛЯ ОБРОБКИ КАРТИНОК

```

class ImageCompress
{
    private $imgOriginal;
    private $imgNew;
    private $height;
    private $width;
    private $ratio;
    private $widthSrc;
    private $heightSrc;
    private $fileName;
    private $targetDir;
    private $ext;

    public function __construct($targetDir, $fileName, $ext){
        $this->fileName = $fileName;
        $this->targetDir = $targetDir;
        $this->ext = $ext;
    }

    public function createImgProportional($height = null, $width
= null)
    {
        switch ($this->ext){
            case 'jpeg':{
                $this->imgOriginal = imagecreatefromjpeg($this-
>targetDir.'/'.$this->fileName);
                break;
            }
            case 'png':{
                $this->imgOriginal = imagecreatefrompng($this-
>targetDir.'/'.$this->fileName);
                break;
            }
            case 'bmp':{
                $this->imgOriginal = imagecreatefromwbmp($this-
>targetDir.'/'.$this->fileName);
                break;
            }
            case 'gif':{
                $this->imgOriginal = imagecreatefromgif($this-
>targetDir.'/'.$this->fileName);

```

```

        break;
    }
    default: break;
}

$this->heightSrc = imagesy($this->imgOriginal);
$this->widthSrc = imagesx($this->imgOriginal);

if($this->heightSrc < 500){
    $this->width = $this->widthSrc;
    $this->height = $this->heightSrc;
} else {
    if($height !== null){
        $this->ratio = $this->heightSrc/$height;
    } else {
        $this->ratio = $this->widthSrc/$width;
    }
    $this->width = $this->widthSrc/$this->ratio;
    $this->height = $this->heightSrc/$this->ratio;
}

$this->imgNew = imagecreatetruecolor($this->width,
$this->height);

return $this;
}

public function saveImgAd()
{
    imagecopyresampled($this->imgNew, $this->imgOriginal, 0,
0, 0, 0, $this->width, $this->height, $this->widthSrc, $this-
>heightSrc);
    if(is_dir($this->targetDir.'../../compressed') == false){
        mkdir($this->targetDir.'../../compressed/');
    }

    switch ($this->ext){
        case 'jpeg':{
            imagejpeg($this->imgNew, $this-
>targetDir.'../../compressed/'.$this->fileName, 60);
            break;
        }
        case 'png':{
            imagepng($this->imgNew, $this-
>targetDir.'../../compressed/'.$this->fileName, 6);
            break;
        }
        case 'bmp':{
            imagewbmp($this->imgNew, $this-
>targetDir.'../../compressed/'.$this->fileName, 60);
            break;
        }
        case 'gif':{

```

```

        imagegif($this->imgNew, $this-
>targetDir.'../../compressed/'.$this->fileName);
        break;
    }
    default: break;
}

imagedestroy($this->imgNew);
}

public function saveImgAvatar()
{
    imagecopyresampled($this->imgNew, $this->imgOriginal, 0,
0, 0, 0, $this->width, $this->height, $this->widthSrc, $this-
>heightSrc);

    switch ($this->ext){
        case 'jpeg':{
            imagejpeg($this->imgNew, $this-
>targetDir.'/'.$this->fileName, 60);
            break;
        }
        case 'png':{
            imagepng($this->imgNew, $this-
>targetDir.'/'.$this->fileName, 6);
            break;
        }
        case 'bmp':{
            imagewbmp($this->imgNew, $this-
>targetDir.'/'.$this->fileName, 60);
            break;
        }
        case 'gif':{
            imagegif($this->imgNew, $this-
>targetDir.'/'.$this->fileName);
            break;
        }
        default: break;
    }

    imagedestroy($this->imgNew);
}
}

```

V.13 МЕТОД ДЛЯ ЗАВАНТАЖЕННЯ ФОТО ОБ'ЯВИ

```

public function uploadPhotoAd(UploadedFile $file)
{
    $ext = $file->guessExtension();
    $fileName = md5(uniqid()).'.'.$file->guessExtension();
    $file->move($this->targetDir.'/original/', $fileName);
}

```

```

        $imgCompressed = new ImageCompress($this->targetDir.'/original', $fileName, $ext);
        $imgCompressed->createImgProportional(500)->saveImgAd();

        return $fileName;
    }

```

В.14 МЕТОД ДЛЯ ПОШУКУ ОСТАННІХ ОБ'ЯВ У БД

```

public function getLatestOffer($page, $typeOffer = null,
    $username = null, $limit = 6)
    {
        $qb = $this->createQueryBuilder('post')
            ->select('post')
            ->setFirstResult($page*6-6)
            ->orderBy('post.updated', 'DESC');

        if($typeOffer == 'sell'){
            $qb->andWhere('post.term is null');
        }
        if($typeOffer == 'rent'){
            $qb->andWhere('post.term is not null');
        }

        if (false === is_null($limit)) {
            $qb->setMaxResults($limit);
        }
        if($username !== null){
            $qb->andWhere('post.user = :username');
            $qb->setParameter('username', $username);
        }

        return $qb->getQuery()
            ->getResult();
    }

```

В.15 МЕТОД ДЛЯ ОТРИМАННЯ ОБ'ЯВ ЗА ЗАДАНИМИ ПАРАМЕТРАМИ

```

public function findAllByParams(Request $request, $page,
    $limit=6)
    {
        $qb = $this->createQueryBuilder('Ad')
            ->select('Ad' )
            ->Where('Ad.typeOffer = :typeOffer')
            ->andWhere('Ad.typeProperty =
:typeProperty')

```

```

        ->orderBy('Ad.updated', 'DESC')
        ->setFirstResult($page*6-6)
        ->setParameter('typeOffer', $request-
>request->get('typeOffer'))
        ->setParameter('typeProperty', $request-
>request->get('typeProperty'));

        if (false === is_null($limit)) {
            $qb->setMaxResults($limit);
        }

        if($request->request->has('city') && $request->request-
>get('city') !== ''){
            $qb->andWhere('Ad.city = :city');
            $qb->setParameter('city', $request->request-
>get('city'));
        }
        if($request->request->has('area') && $request->request-
>get('area') !== ''){
            $qb->andWhere('Ad.area = :area');
            $qb->setParameter('area', $request->request-
>get('area'));
        }
        if($request->request->has('street') && $request-
>request->get('street') !== ''){
            $qb->andWhere('Ad.street = :street');
            $qb->setParameter('street', $request->request-
>get('street'));
        }
        if($request->request->has('min') && $request->request-
>get('min') !== ''){
            $qb->andWhere('Ad.cost > :min');
            $qb->setParameter('min', $request->request-
>get('min'));
        }
        if($request->request->has('max') && $request->request-
>get('max') !== ''){
            $qb->andWhere('Ad.cost < :max');
            $qb->setParameter('max', $request->request-
>get('max'));
        }

        if($request->request->has('rooms') && $request->request-
>get('rooms') !== ''){
            $qb->andWhere('Ad.rooms = :rooms');
            $qb->setParameter('rooms', $request->request-
>get('rooms'));
        }

        if($request->request->has('prepayment') && $request-
>request->get('prepayment') == 1 ){
            $qb->andWhere('Ad.prepayment = :prepayment');

```

```

        $qb->setParameter('prepayment', $request->request-
>get('prepayment'));
    }
    if($request->request->has('metro') && $request->request-
>get('metro') == 1 ){
        $qb->andWhere('Ad.metro = :metro');
        $qb->setParameter('metro', $request->request-
>get('metro'));
    }
    if($request->request->has('renovation') && $request-
>request->get('renovation') == 1 ){
        $qb->andWhere('Ad.renovation = :renovation');
        $qb->setParameter('renovation', $request->request-
>get('renovation'));
    }

    return $qb->getQuery()
        ->getResult();}

```

В.16 ВСТАНОВЛЕННЯ РЕЗУЛЬТАТУ КРИТЕРІВ КОРИСТУВАЧА

```

public function updateProfileScore(User $user, $score)
{
    $qb = $this->createQueryBuilder('p')
        ->update()
        ->set('p.profileScore', $score)
        ->where("p.user={$user->getId()}");

    $qb->getQuery()->execute();
}

```

В.17 КОНФІГУРАЦІЙНИЙ ФАЙЛ ДЛЯ КРИТЕРІВ

```

parameters:
  rules :
    criteria:
      address: 0.3
      rooms: 0.15
      price: 0.20
      floor: 0.025
      area: 0.03
      kitchen_area: 0.05
      type_of_apartment: 0.025
      type_of_walls: 0.025
      is_from_owner: 0.05
      floorsCount: 0.05
      house_area: 0.1
      isNearWithMetro: 0.025

```

```
prePayment: 0.025
```

В.18 МЕТОД ДЛЯ ДОДАВАННЯ ДАНИХ У ВІДПОВІДЬ КОРИСТУВАЧУ, ЯКЩО ВІН УСПІШНО АВТОРИЗУВАВСЯ

```
public function
onAuthenticationSuccessResponse (AuthenticationSuccessEvent
$event)
{
    $data = $event->getData();
    $user = $event->getUser();

    if (!$user instanceof UserInterface) {
        return;
    }

    if ($user->getStatus() !== true) {
        throw new UserEmailNotConfirmed(401, "User didn't
confirm email");
    }

    $data['data'] = array(
        'role' => $user->getRoles(),
        'username' => $user->getUsername(),
        'status' => $user->getStatus(),
    );

    $event->setData($data);
}
```

В.19 КЛАС ДЛЯ ВИДАЛЕННЯ КОРИСТУВАЧЕЙ, ЯКІ НЕ ПІДТВЕРДИЛИ ЕЛЕКТРОНУ АДРЕСУ

```
class DeleteUserFromBdCommand extends ContainerAwareCommand
{
    protected function configure()
    {
        $this->setName('db:user-delete')
            ->setDescription('Delete user not confirm profile');
    }

    protected function execute(InputInterface $input,
OutputInterface $output)
    {
        $userManager = $this->getContainer()-
>get('app.user_manager');
    }
}
```

```

        $userManager->deleteNotConfirmedUser();
    }
}

```

В.20 СТВОРЕННЯ ЗАПИТУ ДЛЯ ОТРИМАННЯ РЕКОМЕНДАЦІЙ

```

$params = [];
$params['index'] = $index;
$params['type'] = $type;
$params['body'] = $this->_getAggregationMaxRelation($args,
    $relationType);

    $size = isset($args['maxPanelProducts']) &&
    is_numeric($args['maxPanelProducts']) ?
    $args['maxPanelProducts'] : 100;
    $params['body'] = [
        '_source' => false,
        'size' => $size,
        'query' => [
            'bool' => [
                'filter' => [
                    [
                        'term' => [
                            'status' => 1
                        ]
                    ]
                ],
                'must_not' => [
                    [
                        'terms' => [
                            '_id' => is_null($idsFilter) ? [] :
$idsFilter
                        ]
                    ]
                ],
                'should' => [],
                'must' => []
            ]
        ]
    ];

    $params['body']['query']['bool']['should'][] = $this-
    >_getQueryByRelations(
        $args['productId'],
        $args['rates']['standard_rule_relations'],
        $maxScore,
        $relationType
    );
    $params['body']['query']['bool']['should'][] = $this-
    >_getQueryByPrice(

```

```

    $args['price'],
    $args['rates']['standard_rule_price'],
    $currency,
    $deltaPrice
);
$params['body']['query']['bool']['should'][] = $this-
>_getQueryByPageViews($args['rates']['standard_rule_pageviews'])
;
$params['body']['query']['bool']['should'][] = $this-
>_getQueryByPageViewsLinear($args['rates']['standard_rule_pagevi
ews_by_position'], $maxPageViewsByPosition);

foreach ($args['categories'] as $category) {
    $params['body']['query']['bool']['should'][] = $this-
>_getQueryByCategories($category,
$args['rates']['standard_rule_categories'],
count($args['categories']));
}

$response = $this->getClient()->search($params);

$result = [];
foreach ($response['hits']['hits'] as $product) {
    $result[] = $product['_source']['sku'];
}

```