

#### МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

#### «МИРЭА – Российский технологический университет» РТУ МИРЭА

Институт международного образования Кафедра инструментального и прикладного программного обеспечения

#### ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**На тему:** «Веб-сервис для розыска потерянных домашних животных»

#### Выполнил студент группы ИКБО-13-18 Бугайчук Назар Сергеевич

Руководитель ВКР: к.т.н., доцент, доцент кафедры ИиППО К.Г. Кряженков

Консультант по экономической части ВКР: старший преподаватель

кафедры экономики И. В. Белоусова

# ЦЕЛЬ И ЗАДАЧИ

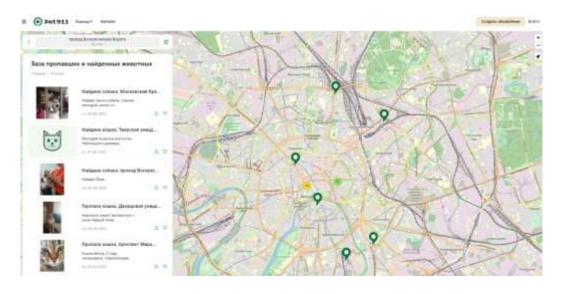


**Цель работы:** спроектировать и разработать веб-сервис для розыска потерянных домашних животных.

#### Задачи работы:

- 1. Провести анализ аналогов веб-сервисов для розыска потерянных домашних животных.
- 2. Спроектировать веб-сервис для розыска потерянных домашних животных.
- 3. Разработать веб-сервис для розыска потерянных домашних животных.
- 4. Провести тестирование разработанного продукта.
- 5. Рассчитать стоимость проведения работ.

# АНАЛИЗ АНАЛОГОВ

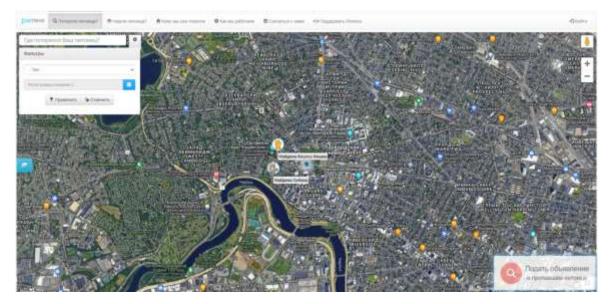






#### Сравнительная характеристика существующих аналогов

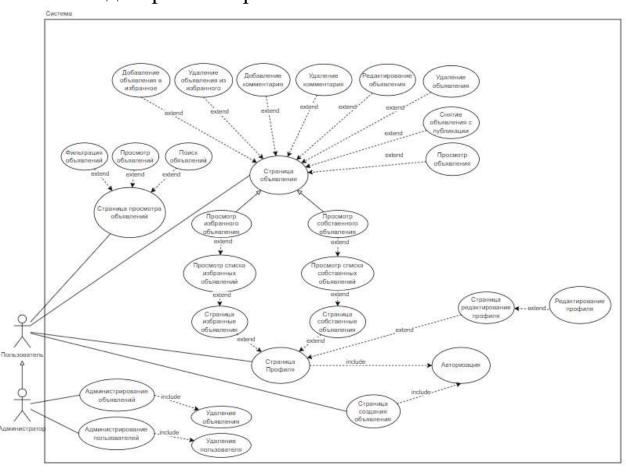
Критерии	Pet911.ru	«Хвост удачи»	Petrieve
Возможность работы с картой	+	-	+
Возможность фильтрации	-	+	-
объявлений			
Поиск объявления по ключевым	-	-	-
словам			
Возможность создания личного	+	+	+
кабинета			
Работа с животными любого вида	+	-	-
Полнотекстовый поиск	-	-	-



# ПРОЕКТИРОВАНИЕ UML-ДИАГРАММЫ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ



#### UML-диаграмма вариантов использования



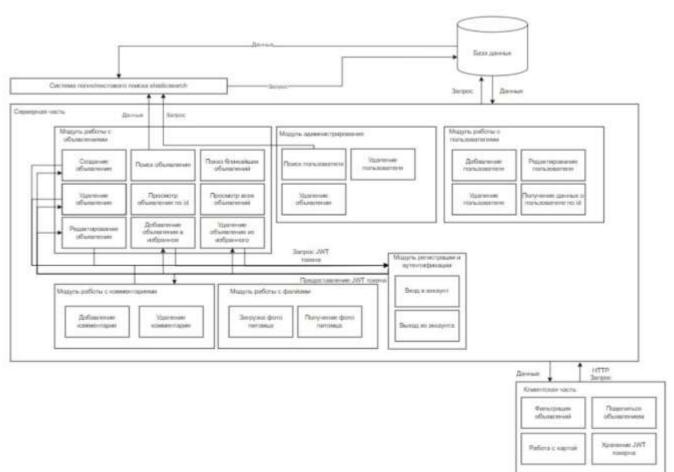
#### UML-диаграмма вариантов использования имеет:

- 2 группы пользователей.
- 19 отношений расширения.
- 4 отношения включения.
- 6 отношений ассоциации.
- 3 отношения обобщения.

# ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНОЙ СХЕМЫ



#### Функциональная схема



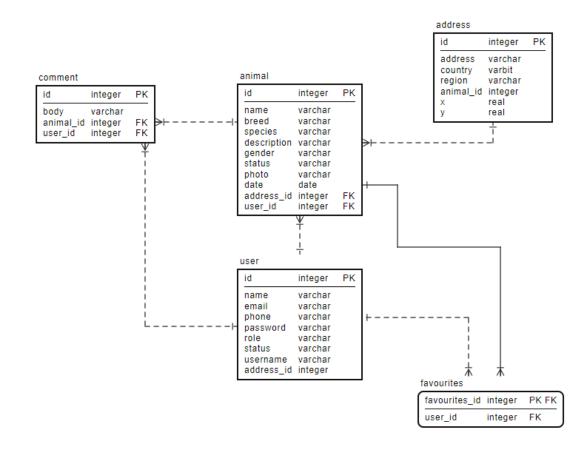
В функциональной схеме было спроектировано:

- 4 компонента.
- 6 модулей.
- 26 функций.

# ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ



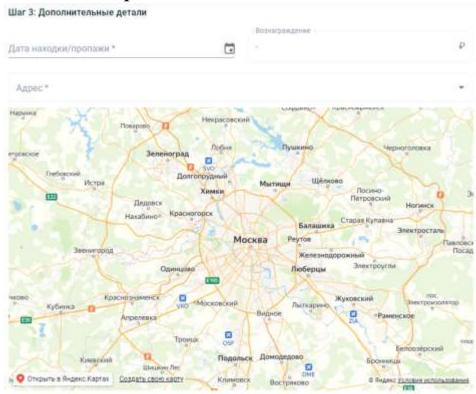
#### ER-диаграмма базы данных



#### ER-диаграмма базы данных имеет:

- 5 сущностей.
- 33 атрибута сущностей.
- 5 связей.
- 6 внешних ключей.
- 5 первичных ключей.

#### Работа с картой – создание объявления



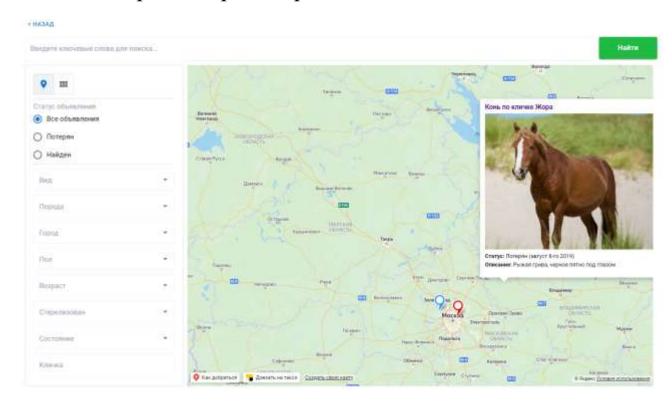
#### Обращение к АРІ Яндекс. Карт

```
const geocode = async (geocode) => {
    return await axios.get( wh 'https://geocode-maps.vandex.ru/l.x/Papiwey=$iprocess.env.REACT_APP_VANDEX_UEOCODER}&format=ison&geocode>$igeocode}')
}
const searchLocation = async (yalue) => {
    return await axios.get( wh 'https://search-maps.vandex.ru/vi/Papiwey=$iprocess.env.REACT_APP_VANDEX_SEARCH}&text=$(yalue)&Lang=ru_RU')
}
```

#### Функции поиска и геокодирования

```
function AddressInput({address, coordinates, setCoordinates, setFormData, formData}) {
   const [places, setPlaces] = useState( initialState []);
   const [locationDuplicate, setLocationDuplicate] = useState( unsubtwise '');
   const onPlacesChange = (e) => {
       e.target.value.length >= 3 && mapService.searchLocation(e.target.value)
            .then(res => setFlaces(res.data.features));
   useEffect( effect () => {
        coordinates && coordinates.length !== 0 && mopService.geocode(coordinates.reverse())
            .then(r =>
               setFormData({
                    ...formData,
                    address: {
                        address: "${r.data.response.GeoObjectCollection.featureMember[0].GeoObject.name},
                        ${r.data.response.GeoObjectCollection.featureMember[0].GeoObject.description},
                        x: coordinates[8].toString(),
                        y: coordinates[1].to5tring()
   1. deps [coordinates])
   const debouncedChangeHandler = useCallback(
        debounce(onPlacesChange, want 306)
        , dept []);
   useEffect( effect () => {
        locationDuplicate && mopService.geocode(locationDuplicate)
                setCoordinates(r.data.response.GeoObjectCollection.featureMember[8].GeoObject.Point.pos.split(' ')
                    .map(str => {
                    return Number(str)
               }).reverse()));
   }, deps [locationDuplicate])
   const onPlacesConfirm = async (e, value) => setLocationOuplicate(value);
```

#### Работа с картой – просмотр объявлений

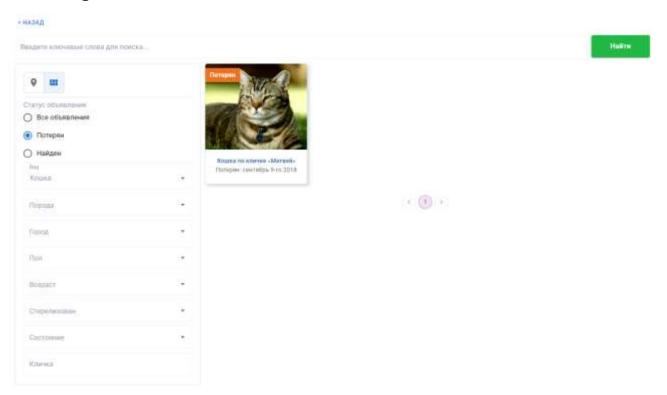


#### Компонент карты просмотра объявлений

```
import React from 'react';
import YMap from "../../ymap/YMap";
import {Clusterer} from "react-yandex-maps";
import MarkItem from "../mark-item/MarkItem";
import useStyles from './PostsMapStyles'
import {useSelector} from "react-redux";
function PostsMap() {
    const classes = useStyles();
    const {filtered} = useSelector( selector state => state.postsReducer);
    return (
            defaultState={{center: [55.75, 37.57], zoom: 9}}
            classes={classes.map}
            <Clusterer
                options={{
                    preset: 'islands#invertedVioletClusterIcons',
                    groupByCoordinates: false,
                }}
                    filtered.map((post) => (
                        <MarkItem key={post.id} post={post} />
                    ))}
            </Clusterer>
        </YMap>
export default PostsMap;
```



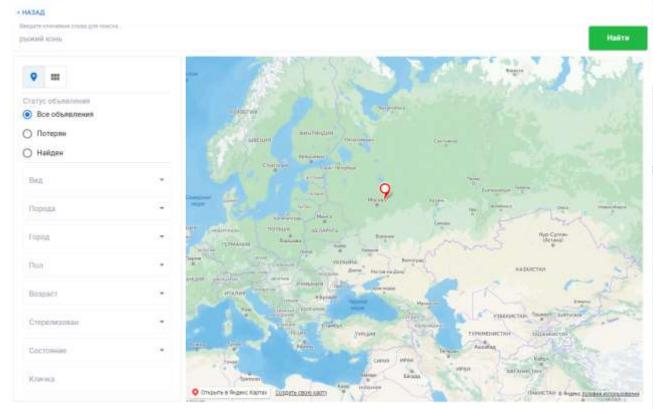
#### Фильтрация объявлений



#### Функция фильтрации объявлений

```
case FILTER_POSTS:
    return {
          ...state,
          filtered: _.filter(state.posts, action.payload),
          isFetching: false,
          error: false
};
```

#### Поиск объявления по ключевым словам

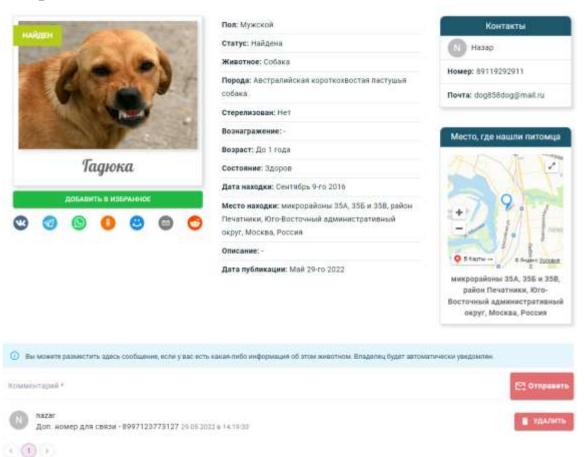


#### Обращение к АРІ поиска

#### Реализация метода поиска объявлений

```
public class AnimalSearch {
    @PersistenceContext
   private EntityManager entityManager;
   public List search(String text) {
        FullTextEntityManager fullTextEntityManager =
                Search.getFullTextEntityManager(entityManager);
        QueryBuilder queryBuilder =
                fullTextEntityManager.getSearchFactory()
                         .buildQueryBuilder().forEntity(Animal.class).get();
        Query query = queryBuilder.keyword().fuzzy() FuzzyContext
                .withEditDistanceUpTo(3)
                .onFields("name", "breed", "species", "description") TermMatchingContext
                .matching(text) TermTermination
                .createQuery();
        FullTextQuery jpaQuery =
                fullTextEntityManager.createFullTextQuery(query, Animal.class);
        List results = jpaQuery.getResultList();
        return results;
```

### Страница объявления



#### Конечная точка добавления объявления

### Обращение к конечной точке добавления объявления

#### Главная страница





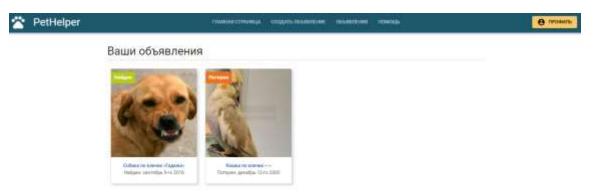






of bearings

#### Страница объявлений пользователя



### Страница избранных объявлений



#### Маршрутизация веб-сервиса

```
function App() {
    const {user} = useSelector( whether state => state.userReducer);
   return (
        <React.Fragment>
            «Header/»
            <Routes>
                <Route
                    element={<Navigate to="/unknown" replace />}
                <Route path="/login" element={user ? <Navigate replace to='/'/> : <Login/>}/>
                <Route path="/register" element={user ? <Navigate replace to="/"/> : <Register/>}/>
                <Route path="/forgetPass" element={user ? <Navigate replace to='/'/> ; <ForgetPassword/>}/>
                <Route path="/changePassword/:code" element={user ? <Navigate replace to='/'/> : <ChangePassword/>}/>
                <Route path="/" element=(<Home/>)/>
                <Route path="/posts" element={<PostsPage/>}/>
                <Route path="/report" element={user ? <Report/> : <Navigate replace to='/login'/>}/>
                <Route path="/posts/:post" element={<Post/>}/>
                <Route path="/activation/:code" element={<Activation/>}/>
                <Route path="/help" element={<Help/>}/>
                <Route path="/unknown" element={<Unknown/>}/>
                <Route path="/profile" element={user ? <Profile/> : <Navigate replace to='/'/>}/>
                <Route path="/profile/edit" element={user ? <ProfileEdit/> : <Navigate replace to='/'/>}/>
                <Route path="/profile/myPosts" element={user ? <MyPosts/> : <Navigate replace to='/'/>}/>
                <Route path="/admin-panel" element={user?.role === "ADMIN" && <AdminPanel/>}/>
                <Route path="/profile/favourites" element={user ? <Favourites /> :<Navigate replace to='/'/> }/>
            </Routes>
       </React.Fragment>
export default App;
```

#### Запись JWT токена в localStorage

```
const login = async (username, password) => {
    return await axios.post( url: '/api/auth/login', data: {
        "username": username,
        "password": password
}).then(response => {
        if (response.data.token) {
            localStorage.setItem("token", JSON.stringify(response.data.token));
            localStorage.setItem("user", JSON.stringify(response.data.user));
        }
        return response.data.user;
})
```

#### Проверка на наличие JWT токена

```
export default function authHeader() {
   const token = JSON.parse(localStorage.getItem( key: 'token'));
   if (token) {
      return {Authorization: token};
   } else {
      return {};
   }
}
```

Всего была разработана и отлажена 6901 строка кода.

# ТЕСТИРОВАНИЕ ВЕБ-СЕРВИСА



#### Тест-кейс поиска объявления

UID 01-9	Numb1	High			
IDEA: Поиск объявления с помощью полнотекстового поиска					
SETUP and ADDITIONAL INFO:					
Адрес сайта: http://localhost:3000/posts					
Браузер: Yandex					
Текст: черное пятно					
Вывод объявлений, у которых присутствуют слова «черное пятно».					
RevisionHistory					
Создан: Бугайчук Н.С. 2022.0:	5.02	Причина: Поиск объявления с помощью			
		полнотекстового поиска			
EXECUTION PART		·			
Шаги:					
Перейдите на страницу объявления сайта http://localhost:3000/posts					
Введите ключевые слова, по которым необходимо найти объявление					
Нажмите кнопку «Найти».					
Ожидаемый результат: «Объявления, в которых присутствуют ранее введенные					
ключевые слова»	ключевые слова»				

#### Тест-кейс добавления комментария

UID 01-7	Numb1	High		
IDEA: Добавление коммен	тария			
SETUP and ADDITIONAL INFO:				
Адрес сайта: http://localhos	t:3000/posts/5			
Браузер: Yandex				
Tekct: «tect.»				
Комментарий добавлен.				
RevisionHistory				
Создан: Бугайчук Н.С. 202	2.05.02	Причина: Добавление комментария		
EXECUTION PART		·		
Шаги:				
Перейдите на страницу объявления сайта http://localhost:3000/posts/5				
Введите комментарий, который хотели бы оставить.				
Нажмите кнопку «Отправ				
Ожидаемый результат: «Комментарий добавлен в БД и отобразился на сайте»				

Всего было разработано 10 тест-кейсов.

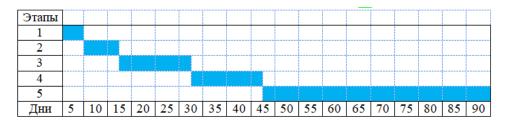
# ЭКОНОМИЧЕСКАЯ ЧАСТЬ



### Вычисления по расчету полной себестоимости проекта

№	Номенклатура статей расходов	Затраты (руб)
1	Материалы, покупные изделия и полуфабрикаты (за вычетом отходов)	5106
2	Специальное оборудование для научных (экспериментальных) работ	-
3	Основная заработная плата научного и производственного персонала	142490
4	Дополнительная заработная плата научного и производственного персонала	39897,2
5	Страховые взносы в социальные фонды	54716,16
6	Расходы на научные и производственные командировки	-
7	Оплата работ, выполненных сторонними организациями и предприятиями	-
8	Накладные расходы	356225
9	Прочие прямые расходы	-
	Итого	598 434,36

### Календарный график проведения работ



# ЗАКЛЮЧЕНИЕ



Результаты выпускной квалификационной работы:

- 1. Проведен анализ аналогов веб-сервисов для розыска потерянных домашних животных.
- 2. Спроектирован веб-сервис для розыска потерянных домашних животных.
- 3. Разработан веб-сервис для розыска потерянных домашних животных.
- 4. Протестирован веб-сервис для розыска потерянных домашних животных.
- 5. Рассчитана себестоимость проведения работ.

# СПАСИБО ЗА ВНИМАНИЕ