

Техническое задание проекта

**«ЭЛЕКТРОННАЯ ДОСКА ОБЪЯВЛЕНИЙ»**

**СОДЕРЖАНИЕ**

[1. ЗАДАНИЕ 2](#_Toc78546405)

[1.1. Цель 2](#_Toc78546406)

[1.2. Краткая сводка возможностей доски объявлений 2](#_Toc78546407)

[1.3. Технологический стек 2](#_Toc78546408)

[1.4. Определения, акронимы и сокращения 2](#_Toc78546409)

[2. ОБЗОР СИСТЕМЫ 3](#_Toc78546410)

[2.1. Общие требования к системе 3](#_Toc78546411)

[2.2. Роли в системе 3](#_Toc78546412)

[2.3. Обзор функциональных возможностей системы 3](#_Toc78546413)

[2.4. Обзор пользовательского интерфейса 3](#_Toc78546414)

[3. ДЕТАЛЬНЫЕ ТРЕБОВАНИЯ 4](#_Toc78546415)

[3.1. Описание основных кейсов работы с системой 4](#_Toc78546416)

[4.2 Требования к экранным формам 5](#_Toc78546417)

[4 ДОПОЛНИТЕЛЬНЫЕ ТРЕБОВАНИЯ 6](#_Toc78546418)

[4.1 Список категорий 6](#_Toc78546419)

[4.2 Другие функциональные требования (TBD, П2) 6](#_Toc78546420)

[4.3 Нефункциональные требования 7](#_Toc78546421)

[5. ИСТОРИЯ ИЗМЕНЕНИЯ ДОКУМЕНТА 8](#_Toc78546422)

## ЗАДАНИЕ

### Цель

Разработка электронной доски объявлений для демонстрации практических навыков разработки, полученных во время прохождения курса программирования *.NET Full-stack Developer* компании *СоларЛаб*.

### Краткая сводка возможностей доски объявлений

Электронная доска объявлений должна предоставлять возможность создавать объявления с возможностью указания контактов продающего и категории товара. Система должна позволять добавлять фотографии к каждому объявлению. Система должна предоставлять возможность эффективного поиска по ряду параметров.

### Технологический стек

При разработке систем рекомендуется использовать следующие технологии:

- Микросервисная архитектура;

- С#, .Net (Net Core), ASP.Net;

- PostgreSQL \ Microsoft SQL Server;

- ORM (Entity Framework Core);

- Rest API;

- Angular;

- JavaScript;

- Redis;

- RabbitMQ;

- Docker;

- Nginx;

- Gitflow.

### Определения, акронимы и сокращения

**ТЗ** – Техническое задание (данный документ)

**Система** – веб-сайт «Электронная доска объявлений», требования к которому указаны в данном документе.

**TBD** – *To Be Defined*. Секция в ТЗ, которая не имеет жестких требований, детальное описание предоставлено на усмотрение команде разработки.

**ВИ** – Вариант Использования или Use Case, описание см. [Википедию](https://ru.wikipedia.org/wiki/%D0%A1%D1%86%D0%B5%D0%BD%D0%B0%D1%80%D0%B8%D0%B9_%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F).

**П2** – требования второго приоритета. Приступать к выполнению, только после реализации основного функционала.

## ОБЗОР СИСТЕМЫ

### Общие требования к системе

Система должна быть реализована как веб-приложение (с микросервисной архитектурой) с клиентским Angular-приложением (frontend) и сервером приложений (backend). Взаимодействие между клиентским приложением и сервером приложений должно осуществляться по Rest API.

Решение должно иметь авторизацию по логину и паролю, которые хранятся в базе (PostgreSQL \ Microsoft SQL Server). Работа с базой данных должна быть реализовано с помощью ОРМ.

Рекомендуется использовать Entity Framework Core. Если будет использована другая ОРМ, то

в комментариях к решению необходимо кратко описать, чем мотивирован выбор.

### Роли в системе

* *Неавторизированный пользователь* – все посетители системы, не прошедшие процедуру регистрации\логина.
* *Авторизированный пользователь* – зарегистрированный пользователь системы, прошедший процедуру логина.
* *Администратор* – роль с повышенными привилегиям (**TBD**).

### Обзор функциональных возможностей системы

Неавторизированный пользователь должен иметь возможность аутентифицироваться и авторизоваться в системе как путем регистрации, так и используя внешнюю аутентификацию через сервисы OAuth от Google, Facebook и VK (внешняя аутентификация - **П2**)

Авторизированный пользователь должен иметь возможность создавать, просматривать, редактировать и удалять свои объявления. А также просматривать контактные данные других пользователей и оставлять комментарии к объявлениям.

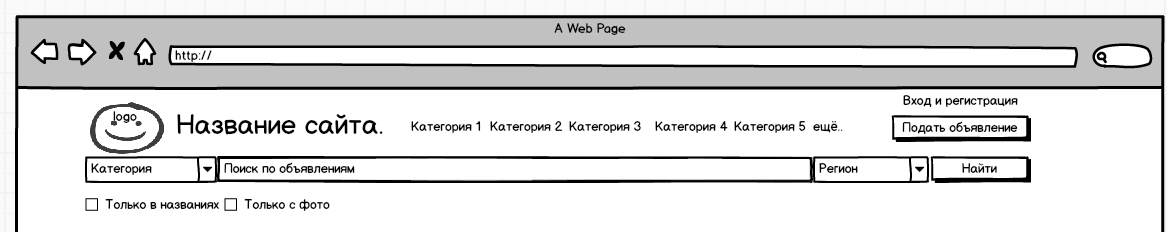
Все пользователи должны иметь возможность просматривать все объявления.

Администратор должен иметь возможность удалять все объявления.

### Обзор пользовательского интерфейса

Все страницы сайта системы должны быть структурно разделены на три блока: шапка, контентная часть и подвал.

Шапка дублируется на всех страницах. Содержит в себе логотип, название сайта, а также навигацию по сайту с возможность поиска. Также должна содержать кнопку «Подать объявление» и регистрационную информацию (либо ссылку на вход и регистрацию). Шапка должна дублироваться на всех страницах системы.



Контентная часть содержит динамически генерируемую информацию, основное содержимое сайта. Подвал сайта содержит служебную информацию, такую как copyright и счётчики для ведения статистики (TBD). Также в нем должны быть размещены ссылки на социальные сети компании и контакты службы поддержки (разработчиков).

## ДЕТАЛЬНЫЕ ТРЕБОВАНИЯ

### Описание основных кейсов работы с системой

#### «Регистрация в системе»

##### Описание ВИ

Незарегистрированный пользователь должен иметь возможность зарегистрироваться в системе.

##### Предусловия

Пользователь не зарегестрирован в системе (не создан его аккаунт с персональными данными)

##### Основной поток действий для пользователя

1. Незарегистрированный пользователь (НП) заходит на любую страницу сайта.
2. НП нажимает на ссылку «Вход и регистрация»
3. Система отображает форму «Вход», см п. 3.2.1
4. Пользователь нажимает на ссылку «Регистрация»
5. Система отображает форму «Регистрация», см. п. 3.2.2
6. Незарегистрированный пользователь вводит запрошенную информацию и подтверждает операцию. Если пользователь отменил операцию, то система отображает главную страницу сайта и сценарий завершается.
7. Система проверят, что необходимые поля введены корректно, Пользователя с таким электронным адресом нет еще в Системе. Если проверка прошла успешно, то поток продолжается, иначе сценарий переходит в п. д, причем все значения (правильные и неправильные) должны отображаться в заполненных ранее полях с выделением неправильного поля и комментариями о существовании такого аккаунта.
8. Система делает следующие действия:

* Регистрирует данного Пользователя с введёнными ранее параметрами.
* Отправляет e-mail вновь зарегистрированному Пользователю об активизации учетной записи с уникальной ссылкой, чтобы проверить правильность указания электронного адреса **П2**
* Пользователь нажимает на уникальную ссылку (ведущую на сайт) в полученном письме об активизации учетной записи на эл почте. **П2**
* Отображает главную страницу сайта с характеристиками личного кабинета: должно отображаться Имя на сайте.

#### ВИ «Войти в систему»

##### Описание ВИ

Пользователь должен иметь возможность пройти процедуру авторизации и войти в Систему.

##### Предусловия

Пользователь должен быть зарегистрирован в Системе, см. п. 3.1.1 ВИ «Регистрация в системе»

##### Основной поток действий

1. Незарегистрированный пользователь (НП) заходит на любую страницу сайта.
2. НП нажимает на ссылку «Вход и регистрация»
3. Система отображает форму «Вход», см п. 3.2.1
4. Пользователь нажимает на ссылку «Регистрация»

#### BИ «Добавить объявление в систему» (TBD)

#### ВИ «Отредактировать объявление» (TBD)

#### ВИ «Удалить объявление» (TBD)

#### ВИ «Найти объявление по требуемым характеристикам» (TBD)

#### ВИ «Оставить комментарий к объявлению» (TBD)

### 

### Требования к экранным формам

#### Форма «Вход»

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Электронная почта | Строка | Текстовое поле для ввода электронного адреса пользователя. |
| Пароль | Строка | Пароль пользователя в системе. |
| Забыли пароль | Ссылка | Ссылка на форму восстановления пароля **П2** |
| Регистрация | Ссылка | Ссылка на форму регистрации |
| Войти | Кнопка | По нажатию осуществляется проверка логина и пароля и вход пользователя в систему. |

#### Форма регистрации

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Ваше имя | Строка | Обязательное поле. Содержит имя пользователя на сайте. Все его объявления и комментарии будут подписаны этим именем. |
| Электронная почта | Строка | Строка с проверкой на допустимый e-mail адрес. Используется как логин пользователя в системе. |
| Номер телефона | Число | Необязательное поле. Содержит маску для ввода кода страны и семи цифр номера +x (xxx)xxx-xx-xx |
| Пароль | Строка | Обязательное поле. Строка не менее 6 символов |
| Подтверждения пароля | Строка | Обязательное поле. Строка не менее 6 символов, должна совпадать с паролем |
| Текст с картинки | Строка | Символы для проверки на то, что пользователь является человеком. |
| Зарегистрироваться | Кнопка | По нажатию проверяется корректность введенных данных и в случае успеха регистрация пользователя в системе. |

## ДОПОЛНИТЕЛЬНЫЕ ТРЕБОВАНИЯ

### Список категорий

Система должна содержать иерархический список категорий, например:

1. Транспорт
   1. Автомобили
   2. Мотоциклы и мототехника
   3. ..
2. Недвижимость
   1. Квартиры
   2. Комнаты
   3. ...

Необходимо использовать не менее 3 основных категорий, в каждой - не менее 2-х подкатегорий.

### Другие функциональные требования (TBD, П2)

При успешной реализации базовых кейсов работы системы предлагается (на выбор) реализовать дополнительные модули системы:

* + - *Сервис нотификации* – отдельный модуль площадки, интегрированный с почтовым клиентом, для рассылки важной информации на email пользователя. Например, информация по активности, связанной с объявлениями, рекомендации новых объявление, подтверждение \ обновление информации о пользователе в системе.
    - Реализация системы покупки лота авторизированным пользователем.
    - *Чат продавца с покупателем* – реализация online-чата на странице объявления \ добавление комментариев к объявлению
    - Создание *системы рейтингов* объявлений \ пользователей системы
    - *Добавление отзывов* о пользователях (добросовестных продавцах)
    - Реализация *системы рекомендаций* по новым объявлениям. Например, рекомендации объявлений из категории товаров, которой интересовался пользователь.
    - *Модуль торгов* – реализация проведения аукциона с возможностью подачи нескольких ценовых предложений от разных пользователей (с определением победителя).

Также, система может включать другой полезный функционал, предложенный и задокументированный командой.

### Нефункциональные требования

#### Интерфейс пользователя

Система должна корректно отображать интерфейс пользователя для различных (типовых) разрешений экрана.

#### Поддержка браузеров

Система должна корректно работать в различных браузерах. Например, поддерживать IE версии 11 и выше, FireFox версии 53 и более поздних, Opera версии 14 и более поздних, Chrome 2 последние версии.

#### Требования к производительности

Система должна отображать (время загрузки) любую форму не дольше, чем за 5 секунд.

#### Требования к безопасности

Система НЕ должна позволять доступ к интерфейсу администратора пользователям без роли «Администратор».

## ОПЫТ И НАВЫКИ

Опыт и навыки, на которые нацелено выполнение данной работы:

* + - Закрепление знаний по основным концепци разработки на платформе .NET (Net Core) и принципов объектно-ориентированного программирования;
    - Опыт работы с микросервисной архитектурой;
    - Опыт проектирования REST API;
    - Опыт разработки с использованием технологий: NET Core, RabbitMQ, PostgreSql, Redis;
    - Опыт работы с Docker;
    - Опыт работы по gitflow;
    - Умение писать unit-тесты;
    - Умение писать SOLID-ный чистый и понятный код, KISS, DRY;
    - Понимание шаблонов проектирования (design patterns);
    - Опыт командной разработки по гибким методологиям;

## ИСТОРИЯ ИЗМЕНЕНИЯ ДОКУМЕНТА

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Автор** | **Изменения** |
| 16.10.2017 | 0.1 | Новиков Е.Н. | Начало написания документа. Базовая структура. |
| 30.07.2021 | 1.0 | Дзицкий В.С. | Актуализация задания. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |