**Chains quizzes**

*Software Requirements Specification*

**Утверждение документа**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Фамилия** | **Дата** | **Подпись** |
| Подготовлен | Белокур Б. | 17.09.2017 |  |
| Подготовлен | Вергун А. | 17.09.2017 |  |
| Подготовлен | Полищук И. | 17.09.2017 |  |
| Проверен | Штогрина Е.С. | 25.09.2017 |  |
| Контроль качества |  |  |  |
| Утверждён |  |  |  |

**Список доступа**

В открытом доступе.

**История изменений**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Версия** | **Дата** | **Автор** | **Описание** | **Раздел** |
|  |  |  |  | **FUN** |

**Информация о документе**

|  |  |
| --- | --- |
| **Дата создания** | 17.09.2017 |
| **Имя файла** | **CQ\_SRS.doc** |
| **Местоположение** | Kiyv,Ukraine |
| **Кол-во страниц** |  |

## Методология разработки

Разработка ведется по методологии **Iterative model.**

## Стандарты написания кода

### **Front-end**

Код пишется на современном javascript используя возможности спецификаций ES6/ES7 (stage-1, stage-2, stage-3).

Основной стэк технологий состоит из React.js ^15.6.1, Redux ^5.\*.\*, React-router-v4.

Для сборки проекта используется Webpack ^3.\*.\*.

Для написание стилей для UI компонентов используется Sass, Css-modules.

Для тестирования кода используется Jest и Enzyme.

Используется линтер eslint.

JS код пишется следуя рекомендациям Airbnb.

Используется JSDoc подход для документирования кода, с помощью Dox генератора в будущем может быть сгенерирована html версия документации.

### **Back-end** Django-rest, postgresql Python рекомендации PEP8

## Введение

Данный документ описывает Веб приложение для проведения тестирования и создания тестов. Особенность этих тестов заключается в том, что каждый тест разбит на несколько уровней и ответы пользователя на вопросы одного из уровней будут влиять на вопросы следующих уровней, так же будет предусмотрена возможность регистрации пользователя при успешном прохождении теста пользователям(доля правильных ответов составляет более 95%) это пользователю будут надаваться права на создание и публикацию своих тестов.

### 

### **Сокращения**

* Оф. источники – официальные источники.

### **Соглашение о терминах**

* Личный кабинет - кабинет пользователя где хранятся его личные данные.
* Аккаунт – профиль пользователя сайта.
* Тест - способ изучения глубинных процессов деятельности системы, посредством помещения системы в разные ситуации и отслеживание доступных наблюдению изменений в ней.
* Цепочка – логическая связка вопросов.
* Уровень – список вопросов на который должен ответить пользователь.
* Тема – это описание ссылки на документацию, к которой закреплён данный вопрос.
* Бэкап - процесс создания копии данных на носителе (жёстком диске, дискете и т. д.), предназначенном для восстановления данных в оригинальном или новом месте их расположения в случае их повреждения или разрушения.

# Раздел 1: Требования

## Обзор

### **Введение**

Этот раздел описывает различные требования (функциональные и нефункциональные)

### **Содержание**

Этот раздел содержит следующие темы

|  |  |
| --- | --- |
| Тема | См. стр. |
| Типы требований | 5 |
| Список функциональных требований | 6 |
| Список не-функциональных требований | 9 |

# Список функциональных требований

|  |  |
| --- | --- |
| **Идентификатор функционального требования** | **Описание** |
| FUN-01 | Прохождения тестов |
| FUN-02 | Регистрация пользователя |
| FUN-03 | Управление личным кабинетом |
| FUN-04 | Нахождение тестов с помощью поиска и фильтров |
| FUN-05 | Создание тестов |
| FUN-06 | Просмотр личной страницы пользователя |
| FUN-07 | Вывод результата теста |
| FUN-08 | Техническая поддержка |
| FUN-09 | Рейтинговая система |
| FUN-10 | Управление пользовательским тестом |

**Прохождения тестов** Тесты в Веб приложении делятся на два вида:

1) Доступные зарегистрированным пользователям.  
2) Доступные незарегистрированным пользователям.

Каждый тест делится на уровни, на странице с уровнем есть список вопросов с ответами на них, пользователю необходимо выбрать ответ, который он считает правильным. Пользователь не может перейти на следующий уровень пока не ответит на все вопросы этого уровня. Следующий уровень будет генерироваться в зависимости от ответов который дал пользователь в этом уровне. То есть если пользователь дал неправильный ответ на один вопрос из цепочки, то в следующих уровнях ему будет задаваться меньше вопросов из этой цепочки, либо вообще не будут задаваться вопросы из данной темы.

**Регистрация на сайте**

Пользователь может зарегистрироваться в Веб приложении, для регистрации необходимо Имя пользователя и почта. На почту приходит код активации аккаунта, после этого ему будут доступны все тесты.

**Управление личным кабинетом**

Пользователь иметь возможность управлять кабинетом, чтобы:

* Добавлять контактную информацию.
* Изменять информацию.
* Удалять аккаунт.
* Восстанавливать аккаунт.
* Добавлять изображение пользователя.
* Смотреть результаты пройдённых тестов, и иметь возможность пройти их снова.
* Просматривать созданные тесты(если пользователь имеет на это право).
* Создание тестов.

**Поиск тестов**

Пользователи должны иметь возможность осуществить поиск определенного теста путем поиска по:

* Название.
* Автор.
* Популярность.

Фильтрация осуществляется по категориям: язык программирования, фреймворк, библиотека, core языка, тип сложности.ы

**Создание тестов**

Пользователи, которые имеют определенный рейтинг (более %95 процентов правильных ответов), имеют возможность создавать свои тесты, пользователь сам группирует свои вопросы по уровням и связывает их в цепочки, при этом он обязан к каждому вопросу дать ссылку на документацию и название темы вопроса или справку в оф. источнике, к каждому вопросу так же пользователь имеет возможность использовать один вопрос для более чем одной цепочки, менять доступность теста(для зарегистрированных или незарегистрированных пользователей). Прохождение своего же теста не меняет рейтинг пользователя, популярность теста. Так же возможно добавление изображения логотипа теста(не обязательно), название теста и краткое описание.

**Просмотр личной страницы пользователя**

Любому зарегистрированному пользователю сайта доступна такая информация другого пользователя:

1. Имя, фамилия.
2. Контактные данные.
3. Изображение профиля.
4. Пройденные тесты, созданные тесты.
5. Рейтинг пользователя(основан на среднем арифметическом успешности, пройдённых тестов)

# Вывод результата теста

После прохождения теста, пользователю выводятся следующие данные:

1. Процентное соотношение правильных ответов от всех.
2. Показываться цепочки, в которых были сделаны ошибки с темами и ссылками на документацию.

# Техническая поддержка

Каждому зарегистрированному пользователю будет доступна форма для связи с администрацией. В которой он может написать про любую проблему на веб-сайте.  
Каждый пользователь может просматривать эти вопросы и давать на них ответ, если ответ был дан раньше администрации, и он был правильный, то тема закрывается, если нет, то администрация сайта дает правильный ответ и закрывает тему.

# Рейтинговая система

Каждый пользователь сможет просматривать Рейтинг тестов по популярности.

# Управление пользовательским тестом

Каждый создатель теста имеет право:  
 1) Редактировать.

1. Порядок вопросов в цепочке.
2. Ссылку на документацию.
3. Название темы.
4. Ответы на вопрос.
5. Изображение к тесту.
6. Название теста.
7. Удалять вопросы.
8. Добавлять вопросы.
9. Удалять тесты.

# Список не-функциональных требований

## Требования удобства использования (USA)

В таблице ниже перечислены все требования удобства использования:

|  |  |
| --- | --- |
| **Идентификатор требования удобства** | **Описание** |
| USA-01 | Простота интерфейса |
| USA-02 | Справка и документация о использовании веб-приложения |

## Надежность (REL)

В таблице ниже перечислены все требования надежности:

|  |  |
| --- | --- |
| **Идентификатор требования надежности** | **Описание** |
| REL-01 | Гарантия работы серверов на 90% |
| REL-02 | Бэкап данных |

## Производительность (PER)

В таблице ниже перечислены все требования производительности:

|  |  |
| --- | --- |
| **Идентификатор требования**  **производительности** | **Описание** |
| PER-01 | Скорость передачи данных (>5Мбит/сек) |

## 

## Поддерживаемость (SUP)

В таблице ниже перечислены все требования поддерживаемости:

|  |  |
| --- | --- |
| **Идентификатор требования**  **поддерживаемости** | **Описание** |
| SUP-01 | Расширяемость функционала |
| SUP-02 | Удобство Администрирования |
| SUP-03 | Масштабируемость: географическая; клиент-ориентированная; |

## Конструктивные требования (DES)

В таблице ниже перечислены все требования к дизайну:

|  |  |
| --- | --- |
| **Идентификатор требования к дизайну** | **Описание** |
| DES-01 | Clean code |
| DES-02 | Unit testing |
| DES-03 | Архитектура Клиент-сервер |
| DES-04 | Используемые языки: Javascript, html, css, python, sql |
| DES-05 | Используемые технологии: React, Redux, React-router-v4, sass, jest, enzyme, babel, webpack, eslint, django, postgresql |
| DES-06 | Используемое ПО: Visual studio code, pycharm |

## Требования к интерфейсам (INT)

В таблице ниже перечислены все требования к интерфейсам:

|  |  |
| --- | --- |
| **Идентификатор требования к интерфейсам** | **Описание** |
| INT-01 | Понятность и логичность пользовательского интерфейса. |
| INT-02 | Гармоничная цветовая палитра пользовательского интерфейса. |

## Совместимость (COM)

В таблице ниже перечислены все требования совместимости:

|  |  |
| --- | --- |
| **Идентификатор требования**  **совместимости** | **Описание** |
| COM-01 | Совместимость с браузерами, в которых поддерживаются все стандарты es5, html5, css3 |

## Безопасность (SEC)

В таблице ниже перечислены все требования безопасности:

|  |  |
| --- | --- |
| **Идентификатор требования**  **безопасности** | **Описание** |
| SEC-01 | Разграничения прав доступа пользователей. |
| SEC-02 | Конфиденциальность данных о результатах прохождения тестирования. |
| SEC-03 | Регистрация пользователя обязательно должна быть подтверждена после верификации кода, который отправлен на указаную им почту. |

# 

# Варианты использования

Веб приложение нацелено на людей ,которые хотят проверить и укрепить свои знания в программировании, владении какой-либо технологией, языком программирования, платформой и т.д. Проект нацелен на людей с разным опытом в разработке. Пользователь сам может создавать тесты в какой-либо категории, если он подтвердил свой опыт или компетентность в знаниях соответвующей категории.

Для того чтоб начать тестирование необязательно регестрироватся, можно пройти тесты, которые доступны для всех.