**Программа и методика испытаний**

сервиса «Техрадар»

Листов \_

Москва 2024

# Цель теста

Целью тестирования является:

* проверка сервиса на соответствие техническому заданию (ТЗ)
* обнаружение отказов и дефектов
* предоставление заказчику информацию о состоянии продукта

# Условия тестирования

За отсутствием тестировочного стенда тестирование проводится на локальной машине ASUS Vivobook DESKTOP-T3PSIQH

операционная система Windows 10, браузер Google Chrome Версия 128.0.6613.85 (Официальная сборка), (64 бит).

# Навигация по таблицам

Для удобства навигации в документе таблицы были выделены цветами:

* Тест-дизайны. Для составления тест-кейсов был проведен тест-дизайн для полей форм авторизации пользователя, авторизации администратора, оценки технологии, добавления технологии. Основными методами проведения тест-дизайна стали методы “Анализ классов эквивалентностей” и “Анализ граничных значений” (см. приложение А).
* Тест-кейсы. Были составлены на основании тест-дизайнов полей форм.
* Тест-кейс API Для тестирования API был использован Swagger.

# Разрешенные символы полей

Данные символы являются корректными для полей форм

Авторизации пользователя и администратора: a-zA-Z \_ 1-9

Оценки технологии: 1-9

Добавления технологии (для имени технологии: a-zA-ZА-я0-9!@#$%^&\*()\_+-=[]{};':"\|,.<>/?, для ссылки на технологию: [-a-zA-Z0-9@:%.*+~#=]{1,256}.[a-zA-Z0-9()]{1,6}\b([-a-zA-Z0-9()@:%*+.~#?&//=]\*)

# Допустимые длины полей форм

Форма авторизации: логин 4-50 символов, пароль 3-50 символов

Форма добавления технологии: название технологии 1-100 символов, ссылка 2-200 символов

# Статусы тест-кейсов

* пройден
* не пройден

# Статусы баг-репортов дефектов

* зарегистрирован
* открыт
* отложен

# Степени важности дефектов

* Критический – дефект, блокирующий большую часть функциональности, полностью нерабочая функция, ветвь функциональности или дефект приводящий к потере информации;
* Серьезный – дефект приводящий к некорректной работе части функциональности, недоступности опций отдельной функции, но не блокирующий тестирование в целом
* Средний – ошибка характеризуется неправильно работающей неосновной части функциональности, не препятствующая проведению тестирования;
* Незначительный – приложение может успешно функционировать, исправление дефекта откладывается на последнюю очередь.

# Приоритет дефектов

* Уделить повышенное внимание  – нет необходимости в сверхсрочном исправлении дефекта, но устранение ошибки должно быть осуществлено непосредственно.
* Обычный  – дефолтный  приоритет при регистрации дефектов.
* Низкий – дефекты исправляются только после устранения всех других ошибок в программе.
* Улучшение – предложения тестовой группы по улучшению качества интерфейса.
* Исправить незамедлительно – исправление дефекта имеет наивысший приоритет, должно быть осуществлено в кратчайшие сроки.

# Термины и сокращения

Специальные термины, используемые в настоящем документе, приведены в таблице ниже. Прочая техническая терминология понимается в соответствии с действующими стандартами и рекомендациями международных органов, ответственных за вопросы стандартизации в сети Интернет.

Таблица 1

| Термин | Описание |
| --- | --- |
| ПМИ | Программа и методика испытаний |
| JSON | JSON (JavaScript Object Notation) — стандартный текстовый формат для хранения и передачи структурированных данных. Он основан на синтаксисе объекта в JavaScript, но не привязан к нему. |
| API | API (англ. application programming interface — программный интерфейс приложения) — это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными. |
|  |  |
|  |  |

# Тестирование формы авторизации

**Метод тест-дизайна: Анализ классов эквивалентностей (актуально только для дальнейшей интеграции с сервисом “Клубная карта”)**

Таблица 2

| Поле | Вариант ввода | Класс эквивалентности | Описание |
| --- | --- | --- | --- |
| Логин | Vova Kukushkin | (латиница) | пройден |
| Вова Кукушкин | (кириллица) | заблокирован |
| Vova\_Kukushkin123 | (латиница с разрешенными символами) | пройден |
| Vova Kukushkin!!! | (латиница с запрещенными символами) | заблокирован |
| Вова Кукушкин1234 | (кириллица с разрешенными символами) | заблокирован |
| Вова Кукушкин!!! | (кириллица с запрещенными символами) | заблокирован |
| ヴォヴァ・ククシュキン | (иные языки) | заблокирован |
|  | (пуст) | заблокирован |
| Vova Kukushkin | (пробел перед вводом) | заблокирован |

Таблица 3

| Поле | Вариант ввода | Класс эквивалентности | Описание |
| --- | --- | --- | --- |
| пароль | admin\_1234\_ | (латиница+разрешенные символы) | пройден |
| админ1234 | (кириллица+разрешенные символы) | заблокирован |
| админ | (кириллица) | заблокирован |
| admin1234!! | (латиница+запрещенные символы) | заблокирован |
| ヴォヴァ・ククシュキン | (иные языки) | заблокирован |
|  | (пуст) | заблокирован |
| admin | (пробел перед вводом) | заблокирован |

**Метод тест-дизайна: Анализ граничных значений (актуально только для дальнейшей интеграции с сервисом “Клубная карта”)**

Таблица 4

| Тестируемое поле | Кол-во вводимых символов | Ожидаемый результат | Находится в границе |
| --- | --- | --- | --- |
| Логин  (кол-во символов: мин 4, макс 50) | 4 символов | пройден | + |
| 50 символов | пройден | + |
| 3 символа | заблокирован | - |
| 51 символ | заблокирован | - |
| 46 символов | пройден | + |

Таблица 5

| Тестируемое поле | Кол-во вводимых символов | Ожидаемый результат | Находится в границе |
| --- | --- | --- | --- |
| Пароль  (кол-во символов: мин 3, макс 50) | 3 символов | пройден | + |
| 50 символов | пройден | + |
| 2 символа | заблокирован | - |
| 51 символ | заблокирован | - |
| 46 символов | пройден | + |

**Тест-кейсы формы авторизации пользователя (актуально только для дальнейшей интеграции с сервисом “Клубная карта”)**

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 01 | Авторизация пользователя (положительный флоу, верный логин и пароль) | 1) Убедиться, что логин и пароль соответствуют классам эквивалентностей из таблиц 2,3 (с полем “описание”=пройден) и таблицам 4,5 (с полем “Ожидаемый результат”=пройден) | 1) Открыт сайт <http://localhost:3000/>  2) Нажать кнопку “Авторизация” | 1) Отправлен POST запрос auth/controller/login  2) В таблице user в базе данных появились введенные логин и пароль  2) Пользователь авторизирован  3) Появится страница с Техрадаром и вкладками “Оценка”, “Радар”, “Выход” | - | - |

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 02 | Авторизация пользователя (негативный флоу, неверный логин и пароль) | 1) Убедиться, что логин соответствует классам эквивалентностей из таблиц 2,3 (с полем “описание”=заблокирован) и таблицам 4,5 (с полем “Ожидаемый результат”=заблокирован) и пароль соответствует классам эквивалентностей из таблиц 2,3 (с полем “описание”=пройден) и таблицам 4,5 (с полем “Ожидаемый результат”=пройден) | 1) Открыт сайт <http://localhost:3000/>  2) Нажать кнопку “Авторизация” | 1) Отправлен POST запрос auth/controller/login  2) В таблице user в базе данных не появились введенные логин и пароль  2) Пользователь не авторизован  3) Появится страница ошибкой “Неверный логин и пароль” | - | - |

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 03 | Авторизация пользователя (негативный флоу, неверный логин и верный пароль) | 1) Убедиться, что пароль соответствует классам эквивалентностей из таблиц 2,3 (с полем “описание”=заблокирован) и таблицам 4,5 (с полем “Ожидаемый результат”=заблокирован) и логин соответствует классам эквивалентностей из таблиц 2,3 (с полем “описание”=пройден) и таблицам 4,5 (с полем “Ожидаемый результат”=пройден) | 1) Открыт сайт <http://localhost:3000/>  2) Нажать кнопку “Авторизация” | 1) Отправлен POST запрос auth/controller/login запрос  2) В таблице user в базе данных не появились введенные логин и пароль  2) В Docker spring-boot-app есть обращение к базе данных (таблица в бд user)  2) Пользователь не авторизован  3) Появится страница ошибкой “Неверный логин” | - | - |

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 04 | Авторизация пользователя (негативный флоу, верный логин и неверный пароль) | 11) Убедиться, что логин и пароль соответствуют классам эквивалентностей из таблиц 2,3 (с полем “описание”=заблокирован) и таблицам 4,5 (с полем “Ожидаемый результат”=заблокирован) | 1) Открыт сайт <http://localhost:3000/>  2) Нажать кнопку “Авторизация” | 1) Отправлен POST запрос auth/controller/login запрос  2) В таблице user в базе данных не появились введенные логин и пароль  2) Пользователь не авторизован  3) Появится страница ошибкой “Неверный пароль” | - | - |

**Тест-кейсы API форма авторизации пользователя**

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 05 | Высокий | Авторизация пользователя (положительный флоу, верный логин и пароль) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) login = user  2) password = 123  Тело запроса:  {  "login": "user",  "password": "123"  } | Отправить запрос auth/controller/login | 2) В таблицу user в базу данных отправился запрос | Пройден |
| Проверить код состояния | Status: 200 OK | Пройден |
| Проверить тело ответа от сервера | Тело ответа в формате JSON возвращается от сервера и будет иметь следующий вид:  {  "login": "user",  "password": "123"  } | Пройден |

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 06 | Высокий | Авторизация пользователя (негативный флоу, неверный логин и пароль) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) login = user1  2) password = 1231  Тело запроса:  {  "login": "user1",  "password": "1231"  } | Отправить запрос auth/controller/login | 2) В таблицу user в базу данных отправился запрос | Пройден |
| Проверить код состояния | Error: response status is 401  (User not found) | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "login": "user1",  "password": "1231"  } | Пройден |

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 07 | Высокий | Авторизация пользователя (негативный флоу, неверный логин и верный пароль) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) login = user1  2) password = 123  Тело запроса:  {  "login": "user1",  "password": "123"  } | Отправить запрос auth/controller/login | 2) В таблицу user в базу данных отправился запрос | Пройден |
| Проверить код состояния | Error: response status is 401  (User not found) | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "login": "user1",  "password": "123"  } | Пройден |

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 08 | Высокий | Авторизация пользователя (негативный флоу, верный логин и неверный пароль) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) login = user  2) password = 1231  Тело запроса:  {  "login": "user",  "password": "1231"  } | Отправить запрос auth/controller/login | 2) В таблицу user в базу данных отправился запрос | Пройден |
| Проверить код состояния | Error: response status is 401  (Wrong password) | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "login": "user",  "password": "1231"  } | Пройден |

**Тест-кейсы формы авторизации администратора (актуально только для дальнейшей интеграции с сервисом “Клубная карта”)**

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 09 | Авторизация администратора (положительный флоу, верный логин и пароль) | 11) Убедиться, что логин и пароль соответствуют классам эквивалентностей из таблиц 2,3 (с полем “описание”=пройден) и таблицам 4,5 (с полем “Ожидаемый результат”=пройден) | 1) Открыт сайт <http://localhost:3000/>  2) Нажать кнопку “Авторизация” | 1) Отправлен POST запрос auth/controller/login запрос  2) В таблицу user в базу данных отправился запрос  2) Администратор авторизован  3) Появится страница с Техрадаром и вкладками “Технология”, “Радар”, “Выход” | - | - |

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | Авторизация администратора (негативный флоу, неверный логин и пароль) | 1) Убедиться, что логин соответствует классам эквивалентностей из таблиц 2,3 (с полем “описание”=заблокирован) и таблицам 4,5 (с полем “Ожидаемый результат”=заблокирован) и пароль соответствует классам эквивалентностей из таблиц 2,3 (с полем “описание”=пройден) и таблицам 4,5 (с полем “Ожидаемый результат”=пройден) | 1) Открыт сайт <http://localhost:3000/>  2) Нажать кнопку “Авторизация” | 1) Отправлен POST запрос auth/controller/login запрос  2) В таблицу user в базу данных отправился запрос  2) Администратор не авторизован  3) Появится страница ошибкой “Неверный логин и пароль” | - | - |

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 11 | Авторизация пользователя (негативный флоу, неверный логин и верный пароль) | 1) Убедиться, что логин и пароль соответствует классам эквивалентностей из таблиц 2,3 (с полем “описание”=заблокирован) и таблицам 4,5 (с полем “Ожидаемый результат”=заблокирован) | 1) Открыт сайт <http://localhost:3000/>  2) Нажать кнопку “Авторизация” | 1) Отправлен POST запрос auth/controller/login запрос  2) В таблицу user в базу данных отправился запрос  2) Администратор не авторизован  3) Появится страница ошибкой “Неверный логин” | - | - |

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | Авторизация пользователя (негативный флоу, верный логин и неверный пароль) | 1) Убедиться, что логин соответствует классам эквивалентностей из таблиц 2,3 (с полем “описание”=пройден) и таблицам 4,5 (с полем “Ожидаемый результат”=пройден) и пароль соответствует классам эквивалентностей из таблиц 2,3 (с полем “описание”=заблокирован) и таблицам 4,5 (с полем “Ожидаемый результат”=заблокирован) | 1) Открыт сайт <http://localhost:3000/>  2) Нажать кнопку “Авторизация” | 1) Отправлен POST запрос auth/controller/login запрос  2) В таблицу user в базу данных отправился запрос  2) Администратор не авторизован  3) Появится страница ошибкой “Неверный пароль” | - | - |

**Тест-кейсы API форма авторизации администратора**

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 13 | Высокий | Авторизация администратора (положительный флоу, верный логин и пароль) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) login = admin  2) password = 123  Тело запроса:  {  "login": "admin",  "password": "123"  } | Отправить запрос auth/controller/login | 2) В таблицу user в базу данных отправился запрос | Пройден |
| Проверить код состояния | Status: 200 OK | Пройден |
| Проверить тело ответа от сервера | Тело ответа в формате JSON возвращается от сервера и будет иметь следующий вид:  {  "login": "admin",  "password": "123"  } | Пройден |

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 14 | Высокий | Авторизация администратора (негативный флоу, неверный логин и пароль) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) login = admin1  2) password = 1231  Тело запроса:  {  "login": "admin1",  "password": "1231"  } | Отправить запрос auth/controller/login | 2) В таблицу user в базу данных отправился запрос | Пройден |
| Проверить код состояния | Error: response status is 401  (User not found) | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "login": "admin1",  "password": "1231"  } | Пройден |

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | Высокий | Авторизация пользователя (негативный флоу, неверный логин и верный пароль) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) login = admin1  2) password = 123  Тело запроса:  {  "login": "admin1",  "password": "123"  } | Отправить запрос auth/controller/login | 2) В таблицу user в базу данных отправился запрос | Пройден |
| Проверить код состояния | Error: response status is 401  (User not found) | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "login": "admin1",  "password": "123"  } | Пройден |

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 16 | Высокий | Авторизация пользователя (негативный флоу, верный логин и неверный пароль) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) login = admin  2) password = 1231  Тело запроса:  {  "login": "admin",  "password": "1231"  } | Отправить запрос auth/controller/login | 2) В таблицу user в базу данных отправился запрос | Пройден |
| Проверить код состояния | Error: response status is 401  (Wrong password) | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "login": "admin",  "password": "1231"  } | Пройден |

# Тестирование формы оценки технологии

**Тест-кейсы формы оценки технологии**

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 17 | Оценка технологии (положительный флоу) | 1) Открыт сайт <http://localhost:3000/>  2) Пользователь вошел в учетную запись  3) Нажать кнопку “Оценка” | 1) Ввести оценку от 1 до 10  2) Нажать на кнопку “Отправить” | 1) Отправлен запрос название запросов  2) В таблицу score в базу данных отправился запрос и появилось поле с оценкой  2) Статистика радара обновится  3) Снова появится страница с оценкой технологии | 1) В таблицу score в базу данных отправился запрос, появилось поле с оценкой | пройдено |

**Тест-кейсы API формы оценки технологии**

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 18 | Высокий | Добавление технологии (позитивный флоу, верная оценка технологии) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) techId = 2  2) scoreValue= 10  Тело запроса:  {  "techId": 2,  "scoreValue": 10  } | Отправить запрос auth/score-controller | 1) В таблицу tech в базу данных отправился запрос, появилось поле с оценкой | Пройден |
| Проверить код состояния | Status: 201 OK | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "techSolutionId": 2,  "scoreValue": 10  } | Пройден |

# Тестирование формы добавления технологии

**(доступно только из аккаунта администратора)**

**Метод тест-дизайна: анализ граничных значений**

**Таблица 6**

| Тестируемое поле | Кол-во вводимых символов | Ожидаемый результат | Фактический результат | Находится в границе |
| --- | --- | --- | --- | --- |
| Название технологии  (кол-во символов: мин 1, макс 100) | 1 символов | пройден | пройден | + |
| 100 символов | пройден | пройден | + |
| 0 символа | заблокирован | пройден | - |
| 101 символ | заблокирован | пройден | - |
| 46 символов | пройден | пройден | + |

**Метод тест-дизайна: Анализ классов эквивалентностей**

**Таблица 7**

| Поле | Вариант ввода | Класс эквивалентности | ожидаемый результат |
| --- | --- | --- | --- |
| Название технологии | python | (латиница) | пройден |
| питон | (кириллица) | заблокирован |
| python\_3.9 | (латиница с разрешенными символами) | пройден |
| “python 3.9” | (латиница с запрещенными символами) | заблокирован |
| питон 3.9 | (кириллица с разрешенными символами) | заблокирован |
| питон 3.9!!! | (кириллица с запрещенными символами) | заблокирован |
| ヴォヴァ・ククシュキン | (иные языки) | заблокирован |
|  | (пуст) | заблокирован |
| python | (пробел перед вводом) | заблокирован |

**Таблица 8**

| Поле | Вариант ввода | Класс эквивалентности | ожидаемый результат |
| --- | --- | --- | --- |
| Ссылка | https://www.python.org/ | (латиница) | пройден |
| https://кумир.рф | (кириллица) | пройден |
|  | (пуст) | заблокирован |
| hhttps://www.python.org/ | (пробел перед вводом) | заблокирован |

**Таблица 9**

| Поле | Вариант ввода | Класс эквивалентности | ожидаемый результат |
| --- | --- | --- | --- |
| Выберете категорию | https://www.python.org/ | (латиница) | пройден |
| https://кумир.рф | (кириллица) | пройден |
|  | (пуст) | пройден |
| hhttps://www.python.org/ | (пробел перед вводом) | пройден |

**Тест-кейсы формы добавления технологии**

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 19 | Добавление новой технологии (положительный флоу, технология и ссылка корректна) | 1) Открыт сайт <http://localhost:8080/swagger-ui/index.html#/auth-controller/login>  2) Администратор вошел в учетную запись  3) Нажата кнопка “Технология” | 1) Ввести корректное название технологии, ссылку на технологию, выбрать любую категорию  2) Нажать на кнопку “Добавить технологию” | 1) Отправлены запросы название запросов  2) В таблицу tech в базу данных отправился запрос и появилось поле с технологией  2) Технология добавлена на радар | - | пройдено |

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 20 | Добавление новой технологии (негативный флоу, технология корректна и ссылка не корректна) | 1) Открыт сайт ссылка  2) Администратор вошел в учетную запись  3) Нажата кнопка “Технология” | 1) Ввести название технологии на латинице, указать недействительную ссылку на технологию, выбрать любую категорию  2) Нажать на кнопку “Добавить технологию” | 1) Отправлены запросы название запросов  2) В таблицу tech в базу данных не отправился запрос и не появилось поле с технологией  2) Технология не добавлена на радар  3) Снова появится страница с добавлением технологии с сообщением “Некорректная ссылка” | - | пройдено |

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 21 | Добавление новой технологии (негативный флоу, технология некорректна и ссылка корректна) | 1) Открыт сайт ссылка  2) Администратор вошел в учетную запись  3) Нажата кнопка “Технология” | 1) Ввести название технологии на кириллице, указать действительную ссылку на технологию, выбрать любую категорию  2) Нажать на кнопку “Добавить технологию” | 1) Отправлены запросы название запросов  2) В таблицу tech в базу данных не отправился запрос и не появилось поле с технологией  2) Технология не добавлена на радар  3) Снова появится страница с добавлением технологии с сообщением “Некорректная технология” | - | пройдено |

| id | Название кейса | Предусловие | Шаги теста | Ожидаемые результаты | Фактические результаты | Статус |
| --- | --- | --- | --- | --- | --- | --- |
| 22 | Добавление новой технологии (негативный флоу, технология некорректна и ссылка некорректна) | 1) Открыт сайт ссылка  2) Администратор вошел в учетную запись  3) Нажата кнопка “Технология” | 1) Ввести название технологии на кириллице, указать недействительную ссылку на технологию, выбрать любую категорию  2) Нажать на кнопку “Добавить технологию” | 1) Отправлены запросы название запросов  2) В таблицу tech в базу данных не отправился запрос и не появилось поле с технологией  2) Технология не добавлена на радар  3) Снова появится страница с добавлением технологии с сообщением “Некорректная технология и некорректная ссылка” | - | пройдено |

**Тест-кейсы API формы добавления технологии**

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 23 | Высокий | Добавление технологии (положительный флоу верное название технологии, верная ссылка, любая технология) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) name = Swagger  2) documentationUrl= <https://swagger.io/>  3) category = TOOLS  Тело запроса:  {  "name": "Swagger",  "documentationUrl": "https://swagger.io/",  "category": "TOOLS"  } | Отправить POST запрос  <http://localhost:8080/swagger-ui/index.html#/tech-solution-controller/createSolution> | 1) В таблицу tech в базу данных отправился запрос, появилось поле с технологией | Пройден |
| Проверить код состояния | Status: 200 OK | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "name": "Swagger",  "documentationUrl": "https://swagger.io/",  "category": "TOOLS"  } | Пройден |

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 24 | Высокий | Добавление технологии (негативный флоу, неверное название технологии, верная ссылка, любая технология) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) name = Swagger № 2  2) documentationUrl= <https://swagger.io/>  3) category = TOOLS  Тело запроса:  {  "name": "Swagger № 2",  "documentationUrl": "https://swagger.io/",  "category": "TOOLS"  } | Отправить POST запрос  <http://localhost:8080/swagger-ui/index.html#/tech-solution-controller/createSolution> | 1) В таблицу tech в базу данных не отправился запрос, не появилось поле с технологией | Пройден |
| Проверить код состояния | Error: response status is 400 | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "name": "Swagger № 2",  "documentationUrl": "https://swagger.io/",  "category": "TOOLS"  } | Пройден |

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 25 | Высокий | Добавление технологии (негативный флоу, верное название технологии, неверная ссылка, любая технология) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) name = Swagger  2) documentationUrl= !!!<https://swagger.io/>  3) category = TOOLS  Тело запроса:  {  "name": "Swagger",  "documentationUrl": "!!!https://swagger.io/",  "category": "TOOLS"  } | Отправить POST запрос  <http://localhost:8080/swagger-ui/index.html#/tech-solution-controller/createSolution> | 1) В таблицу tech в базу данных не отправился запрос, не появилось поле с технологией | Пройден |
| Проверить код состояния | Error: response status is 500 | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "name": "Swagger",  "documentationUrl": "!!!https://swagger.io/",  "category": "TOOLS"  } | Пройден |

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 26 | Высокий | Добавление технологии (негативный флоу, неверное название технологии, неверная ссылка, любая технология) | Открыть документацию Swagger  <http://localhost:8080/swagger-ui/index.html#/auth-controller/login> | Параметры в случае POST передаются в теле запроса в JSON объекте.  1) name = Swagger № 2  2) documentationUrl= !!!<https://swagger.io/>  3) category = TOOLS  Тело запроса:  {  "name": "Swagger № 2",  "documentationUrl": "!!!https://swagger.io/",  "category": "TOOLS"  } | Отправить POST запрос  <http://localhost:8080/swagger-ui/index.html#/tech-solution-controller/createSolution> | 1) В таблицу tech в базу данных не отправился запрос, не появилось поле с технологией | Пройден |
| Проверить код состояния | Error: response status is 400 | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "name": "Swagger № 2",  "documentationUrl": "!!!https://swagger.io/",  "category": "TOOLS"  } | Пройден |

**Тест-кейсы API формы удаления технологии**

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 27 | Высокий | Добавление технологии (негативный флоу, неверное название технологии, верная ссылка, любая технология) | Headers request:  Content-Type:  application/json | Параметры в случае DELETE передаются в теле запроса в JSON объекте.  1) id = 2 | Отправить DELETE запрос  http://localhost:8080/swagger-ui/index.html#/tech-solution-controller/deleteTechSolution | 1) В таблицу tech в базу данных не отправился запрос, не появилось поле с технологией | Пройден |
| Проверить код состояния | Status: 200 OK | Пройден |

**Тест-кейсы API получение информации по категории**

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 28 | Высокий | Получение информации по технологии | Headers request:  Content-Type:  application/json | Параметры в случае GET передаются в теле запроса в JSON объекте.  1) category = LANGUAGES | Отправить GET запрос  http://localhost:8080/swagger-ui/index.html#/ | 1) В таблицу tech в базу данных отправился запрос, получен запрос | Пройден |
| Проверить код состояния | Status: 200 OK | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "entries": [  {  "id": 6,  "link": "https://ru.wikipedia.org/wiki/C%2B%2B",  "quadrant": 0,  "ring": 4,  "label": "C++",  "active": false,  "moved": 2  }  ]  } | Пройден |

**Тест-кейсы API получение всей доступной информации по категории**

| id | Приоритет | Название | Предусловия | Тестовые данные | Шаги | Ожидаемый результат | Статус прохождения |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 29 | Высокий | Получение всей доступной информации по технологии | Headers request:  Content-Type:  application/json | Параметры в случае GET не передаются | Отправить GET запрос  http://localhost:8080/swagger-ui/index.html#/tech-solution-controller/getAllTechSolution | 1) В таблицу tech в базу данных отправился запрос, получен запрос | Пройден |
| Проверить код состояния | Status: 200 OK | Пройден |
| Проверить тело ответа от сервера | Тело ответа возвращается от сервера и будет иметь следующий вид:  {  "entries": [  {  "id": 6,  "link": "https://ru.wikipedia.org/wiki/C%2B%2B",  "quadrant": 0,  "ring": 4,  "label": "C++",  "active": false,  "moved": 2  }  ]  } | Пройден |

**Приложение A**

### Техника анализа классов эквивалентности

### Техника анализа классов эквивалентности - это техника, при которой мы разделяем функционал (часто диапазон возможных вводимых значений) на группы эквивалентных по своему влиянию на систему значений. Такое разделение помогает убедиться в правильном функционировании целой системы — одного класса эквивалентности, проверив только один элемент этой группы. Эта техника заключается в разбиении всего набора тестов на классы эквивалентности с последующим сокращением числа тестов.

Техника рекомендует проведение тестов для всех классов эквивалентности, хотя бы по одному тесту для каждого класса. Д стремится *не только сокращать количество тестов*, но и *сохранять приемлемое тестовое покрытие*.

### *Признаки эквивалентности тестов*:

* направлены на поиск одной и той же ошибки;
* если один из тестов обнаруживает ошибку, другие скорее всего, тоже её обнаружат;
* если один из тестов не обнаруживает ошибку, другие, скорее всего, тоже её не обнаружат;
* тесты используют схожие наборы входных данных;
* для выполнения тестов мы совершаем одни и те же операции;
* тесты генерируют одинаковые выходные данные или приводят приложение в одно и то же состояние;
* все тесты приводят к срабатыванию одного и того же блока обработки ошибок;
* ни один из тестов не приводит к срабатыванию блока обработки ошибок.

## **Техника анализа граничных значений входных данных**

Это техника проверки поведения продукта на крайних (граничных) значениях входных данных. Граничное тестирование также может включать тесты, проверяющие поведение системы на входных данных, выходящих за допустимый диапазон значений. При этом система должна определённым (заранее оговоренным) способом обрабатывать такие ситуации. Например, с помощью исключительной ситуации или сообщения об ошибке.

### Граничные значения очень важны и их обязательно следует применять при написании тестов, т.к. именно в этом месте чаще всего и обнаруживаются ошибки.

На каждой границе диапазона следует проверить по три значения:

1. граничное значение;
2. значение перед границей;
3. значение после границы.

*Цель* этой техники — найти ошибки, связанные с граничными значениями.

**Ссылки на источники информации**

1. <https://qaevolution.ru/testovaya-dokumentaciya/test-dizajn/texnika-analiza-klassov-ekvivalentnosti/>
2. <https://qaevolution.ru/testovaya-dokumentaciya/test-dizajn/texnika-analiza-granichnyx-znachenij/>