# 

# 

# **План тестирования Телеграмм-бота “HackOrohezh\_bot” для подачи заявок в учебный центр Neoflex**

# 

# 

# 

# 

# 

# 

# **Основная информация**

| **Заказчик** | ООО “Неофлекс Консалтинг” |
| --- | --- |
| **Создано (автор)** | Гермоненко Е.А., Меркулов И.А. |
| **Дата** | 26.01.2025 |
| **Версия** | 1.0 |
| **Статус** |  |

# 

# **История редактирования**

| **Версия** | **Описание** | **Автор** | **Дата** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# 

[**1. Введение 4**](#_1v65np6cjrpp)

[1.1 Основная информация 4](#_2qpeqruzwf9i)

[1.2 Цель 4](#_zhss2n1aup2c)

[**2. Рамки проекта 4**](#_mmr8o61cl0hv)

[2.1 Область тестирования Telegram-бота 4](#_owo1qhcsqgvx)

[2.2 Область тестирования админ-панели 5](#_1dnnitsw4naa)

[2.3 Область тестирования микросервисов 5](#_3a2dgzvjb0fz)

[**3. План работы 5**](#_lt2r7a5pfg3a)

[**4. Функции, которые нужно протестировать 6**](#_62egrf5n5eew)

[4.1. Подача заявки пользователем: 6](#_n5iuxbm18u9m)

[4.2. Оставление предзаявки пользователем: 6](#_bnfov2us5xcu)

[4.3. Корректность работы кнопок «Подать заявку» и «Оставить предзаявку» в разные периоды времени: 6](#_6v6et45i3nsc)

[4.4. Админ-панель: 6](#_c02v7qywrmaw)

[4.5. Проверка взаимодействия front-end и back-end: 6](#_rv8jo01amnay)

[**5. Ресурсы 7**](#_cp39frsfo8cs)

[5.1. Инструменты: 7](#_qzwgb4a007k)

[5.2. Список браузеров: 7](#_3ii6qt57n67q)

[5.3. Список устройств: 7](#_4qmnlulzdzgf)

[5.4. Список приложений: 8](#_4c7uyttzu9zf)

[**6. Критерии качества 8**](#_27t3f4cv46sk)

[6.1. Критерии начала тестирования: 8](#_o7nx53gmo7os)

[6.2. Критерии окончания тестирования: 8](#_dvyran2msql8)

[6.3. Функциональные критерии: 8](#_hl51i2q8y4i4)

[6.4. Технические критерии: 8](#_cuy22eybzr2p)

[6.5. Критерии удобства использования: 9](#_tps3j4mjjfc)

[6.6. Критерии безопасности. 9](#_jnkorotxw4vu)

[**7. Проблемы и риски 9**](#_17t6tmmd1ovj)

[8. Ожидания команды тестирования 9](#_1ddxg9mtws5q)

[9. Участники команды тестирования и их обязанности 10](#_4k7lk1tg1b0w)

[10. Результаты 10](#_4ilc6e15uz9)

### 

# **1. Введение**

## **1.1 Основная информация**

Документ описывает методы и подходы к тестированию, которые будут использоваться тестировщиками команды для проверки функциональности Telegram-бота, предназначенного для автоматизации подачи заявок на обучение, а также для проверки работы связанных микросервисов и админ-панели.

План тестирования предназначен для использования тестировщиками, аналитиками, разработчиками и другими заинтересованными сторонами.

Объект тестирования — это функциональность Telegram-бота, взаимодействие с базой данных и API, а также корректная работа админ-панели. Тестирование будет проводиться с использованием Docker, Postman, Swagger, PostgreSQL, Telegram Web, Telegram App.

## **1.2 Цель**

Тест-план проекта преследует следующие цели:

* Определить ключевые сценарии и компоненты, подлежащие тестированию.
* Описать стратегии и методы тестирования, которые будут применяться для проверки функциональности системы.
* Определить ресурсы, необходимые для проведения тестирования.
* Составить подробный отчет о результатах тестирования.

Результаты тестирования будут представлены в виде отчетов, включающих описание найденных ошибок. Все баги будут отслеживаться в баг-трекинговой системе(excel), а также будут предоставлены рекомендации по их исправлению.

# **2. Рамки проекта**

## **2.1 Область тестирования Telegram-бота**

В объем работ по тестированию Telegram-бота входит проверка следующих компонентов и функций:

1. Подача заявки на обучение:
   * Проверка последовательного ввода данных (имя, фамилия, город, номер телефона, электронная почта).
   * Проверка выбора направления обучения.
   * Проверка смены ролей пользователя (visitor → candidate → external\_user).
   * Проверка регистрации новых пользователей в системе.
   * Проверка предоставления ссылки на страницу с входными испытаниями.
2. Оставление предзаявки:
   * Проверка последовательного ввода данных (имя, фамилия, город, номер телефона, электронная почта).
   * Проверка уведомления об открытии набора.
   * Проверка выбора направления обучения после открытия набора.
3. Проверка работы кнопок «Подать заявку» и «Оставить предзаявку» в зависимости от периода набора.
4. Проверка взаимодействия Telegram-бота с базой данных и микросервисами.
5. Проверка шаблонов сообщений (приветствия, уведомления, ошибки).

## **2.2 Область тестирования админ-панели**

В объем работ по тестированию админ-панели входит проверка следующих компонентов и функций:

1. Вход в админ-панель.
2. Настройка периодов активности кнопок «Подать заявку» и «Оставить предзаявку».
3. Просмотр списка всех заявок с возможностью фильтрации и сортировки.
4. Просмотр детальной информации о заявке (личные данные пользователя, выбранное направление).
5. Изменение статуса заявки (например, «На рассмотрении», «Принята», «Отклонена»).

## **2.3 Область тестирования микросервисов**

В объем работ по тестированию микросервисов входит проверка следующих компонентов и функций:

1. Взаимодействие микросервиса №1 с базой данных (создание, обновление и получение данных).
2. Взаимодействие микросервиса №2 с микросервисом №1.
3. Обработка ошибок и некорректных данных на уровне API.
4. Проверка соблюдения ограничений доступа к данным.

# **3. План работы**

Стороны договорились следовать следующему плану работы:

1. **Подготовка плана тестирования:**
   * Определение целей тестирования.
   * Разработка тест-плана.
   * Утверждение тест-плана с командой разработки.
2. **Разработка тестовой документации:**
   * Подготовка тест-кейсов на основе функциональных требований.
   * Утверждение тест-кейсов с разработчиками.
3. **Функциональное тестирование:**
   * Проведение тестирования Telegram-бота, админ-панели и микросервисов.
   * Документирование найденных ошибок и их регистрация в баг-трекинговой системе.
4. **Подготовка финального отчета:**
   * Обобщение результатов тестирования.
   * Составление отчета, включая статус исправленных/неисправленных дефектов.
   * Передача финального отчета заказчику.

# **4. Функции, которые нужно протестировать**

## **4.1. Подача заявки пользователем:**

* Проверить корректную подачу заявки пользователем.
* Убедиться, что подача новой заявки невозможна, при вводе уже существующей почты. Пользователю выдается ранее созданная учетная запись с выбранным направлением.
* Проверить, что при успешной подаче новой заявки пользователь получает ссылку на страницу с входными испытаниями.

## **4.2. Оставление предзаявки пользователем:**

* Проверить корректную подачу предзаявки пользователем.
* Убедиться, что подача новой заявки невозможна, при вводе уже существующей почты. Пользователю выдается ранее созданная учетная запись с выбранным направлением.
* Проверить, что при успешной подаче предзаявки пользователь получает ссылку на страницу с входными испытаниями.

## **4.3. Корректность работы кнопок «Подать заявку» и «Оставить предзаявку» в разные периоды времени:**

* Проверить, что кнопка «Подать заявку» активна только в даты набора.
* Проверить, что кнопка «Оставить предзаявку» активна, если набор закрыт.
* Убедиться, что обе кнопки не могут быть активны одновременно.

## **4.4. Админ-панель:**

* Проверить возможность просмотра всех поданных заявок с возможностью фильтрации и сортировки, а также возможность просмотра детальной информации о заявке.
* Убедиться, что функционал для изменения статуса заявки работает корректно.
* Проверить возможность настройки периодов активности кнопок «Подать заявку» и «Оставить предзаявку».

## **4.5. Проверка взаимодействия front-end и back-end:**

* Убедиться, что все описанные функции корректно обрабатываются в БД.

# **5. Ресурсы**

## **5.1. Инструменты:**

| **Наименование процесса** | **Инструмент** |
| --- | --- |
| Баг трекинговая система | Microsoft Word |
| Тест-кейсы | Microsoft Excel |
| Захват изображений | Screenshots |
| Тестирование API | Postman, Swagger |
| Создание и управление тестовым окружением | Docker |
| Работа с БД | PostgreSQL, pgAdmin4, DBVisualizer, DBeaver |

## **5.2. Список браузеров:**

| **Наименование браузера** | **Версия браузера** |
| --- | --- |
| Google Chrome | 132.0.6834.111 |
| Mozilla Firefox | 134.0.2 |

## **5.3. Список устройств:**

| **Устройства** | **Версия устройства** |
| --- | --- |
| Samsung Galaxy M51 | Android 12 |
| Iphone 14 | IOS 18.1.1 |

## **5.4. Список приложений:**

| **Приложения** | **Версия приложения** |
| --- | --- |
| Telegram Desktop | 5.10.5 |
| Telegram Messenger | 11.7 |
| Telegram Web | 2.1.0 |

# **6. Критерии качества**

## **6.1. Критерии начала тестирования:**

* Наличие технической документации.
* Завершение разработки ключевых функций.
* Подготовлены тестовые данные.
* Получение одобрения от разработчиков.

## **6.2. Критерии окончания тестирования:**

* Отсутствие критических и блокирующих дефектов.
* Завершение всех запланированных сценариев тестирования.
* Соответствие требованиям.
* Код системы должен быть покрыт тестами минимум на 50%

## **6.3. Функциональные критерии:**

* Все функции работают согласно требованиям.
* Нет критических и блокирующих дефектов, влияющих на работоспособность основных функций.
* Корректная работоспособность на всех заявленных платформах и устройствах.

## **6.4. Технические критерии:**

* Корректность типов данных в БД.
* Telegram-бот должен обрабатывать до 1000 заявок в сутки без задержек.
* Архитектура системы должна быть масштабируемой, с возможностью увеличения нагрузки без изменений базовых компонентов.

## **6.5. Критерии удобства использования:**

* Интерфейс системы должен быть интуитивно понятным и удобным для пользователя.

## **6.6. Критерии безопасности.**

* Данные пользователей, должны храниться в зашифрованном виде в БД.

# **7. Проблемы и риски**

Проблемы, которые могут повлиять на результаты тестирования:

* Изменение требований в процессе тестирования.
* Недостаток человеческих ресурсов, оборудования или программного обеспечения.
* Задержки в разработке ПО.
* Задержки в исправлении ошибок.

# **8. Ожидания команды тестирования**

1. **Обеспечение документацией:**Команда тестирования ожидает, что предоставленные материалы (технические задания, схемы бизнес-процессов, спецификации и описание функционала) будут актуальными и полными на протяжении всего хакатона.
2. **Подготовка инфраструктуры:**Все необходимые ресурсы, такие как доступ к репозиториям кода, тестовому серверу, базе данных и Telegram-боту, должны быть предоставлены до начала тестирования.
3. **Приоритетное исправление ошибок:**Участники команды разработки должны оперативно исправлять критические ошибки, найденные в процессе тестирования, чтобы минимизировать задержки и риски несоответствия функционала требованиям.
4. **Релизные заметки:**К каждому выпуску новой версии программного обеспечения должна быть добавлена заметка, содержащая:
   * Список добавленных, измененных или удаленных функций.
   * Информацию о том, как эти изменения влияют на общее поведение системы.
5. **Поддержка в процессе хакатона:**В рамках хакатона от учебного центра ожидается, что менторы и преподаватели будут доступны для консультаций по вопросам архитектуры и функциональности, а также по настройке и интеграции инструментов.

# **9. Участники команды тестирования и их обязанности**

Для успешной реализации проекта в рамках хакатона от учебного центра команда состоит из следующих специалистов и их ключевых задач:

**QA-инженеры (тестировщики, 2 человека):**

* + **Подготовка плана тестирования**, включающего:
    - Проверку работы всех сценариев, включая подачу заявки и предзаявки.
    - Проверку корректности работы кнопок «Подать заявку» и «Оставить предзаявку» в разные периоды активности.
    - Проверку взаимодействия Frontend и Backend.
  + **Разработка тест-кейсов** для каждого функционального требования.
  + **Проведение тестирования** (функционального, регрессионного, интеграционного).
  + Составление отчетов о найденных ошибках, описание проблем и предложений по их исправлению.
  + Подготовка итогового отчета о тестировании.

# **10. Результаты**

По итогам хакатона команда тестирования предоставляет следующие результаты:

1. **План тестирования:**
   * Подробный документ, описывающий цели тестирования, стратегии, подходы, инструменты, сценарии.
2. **Отчеты об ошибках:**
   * Список выявленных дефектов с описанием:
     + Статус ошибки (например, «Открыта», «Исправлена», «Закрыта»).
     + Приоритет и уровень критичности (например, «Критическая», «Средняя», «Незначительная»).
     + Шаги для воспроизведения.
     + Ожидаемый и фактический результаты.
     + Скриншоты, логи или другие доказательства, подтверждающие наличие ошибки.
     + Рекомендации по исправлению.
3. **Итоговый отчет о тестировании:**
   * Итоговый отчет с общими выводами по результатам тестирования, включая:
     + Статистику выполнения тестов.
     + Анализ качества разработанного функционала.
     + Рекомендации по дальнейшему улучшению продукта.