План тестирования “NDA”

Спринт №1

Основная информация

|  |  |
| --- | --- |
| Заказчик | Яндекс |
| Создано (автор) | QA Engineer - Бардюков Данил |
| Дата |  |
| Версия | 2.0 |
| Статус | Актуален |

История редактирования

|  |  |  |  |
| --- | --- | --- | --- |
| Версия | Описание | Автор | Дата |
|  | Тест-план | Бардюков Данил |  |
|  |  |  |  |

Оглавление

|  |  |
| --- | --- |
| 1. Введение | 4 |
| 1.1. Основная информация | 4 |
| 1.2. Цель | 4 |
| 2. Рамки проекта | 4 |
| 2.1. Область тестирования веб-сайта | 4 |
| 3. План работы | 4 |
| 4. План и стратегия тестирования | 5 |
| 4.1. Уровни и стратегия тестирования | 5 |
| 4.2. Функциональное тестирование | 5 |
| 4.3 Нефункциональное тестирование | 6 |
| 4.4. Глубина тестирования | 6 |
| 4.5. Процедура тестирования | 6 |
| 4.6. Сообщения об ошибках | 8 |
| 5. Ресурсы | 9 |
| 5.1. Инструменты | 10 |
| 5.2. Список браузеров | 11 |
| 5.3. Список устройств | 11 |
| 6. Критерии качества | 11 |
| 7. Риски процесса тестирования | 11 |
| 8. Ожидания команды тестирования | 12 |
| 9. Обязанности участников тестовой группы | 12 |
| 10. Результаты | 12 |

# 1. Введение

## 1.1. Основная информация

Документ описывает методы и подходы к тестированию, которые будут использоваться командой QA, для тестирования сайта “NDA”.

План тестирования может использоваться как тестировщиками, так и менеджерами и разработчиками.

Объект тестирования — это деятельность, направленная на проверку работоспособности функций веб-приложения в браузерах Google Chrome, Yandex Browser, Safari.

## 1.2. Цель

Тест-план проекта преследует следующие цели:

● Определить существующую информацию о проекте и программных компонентах, подлежащих тестированию.

● Описать стратегии тестирования, которые будут использоваться.

● Определить необходимые ресурсы для проведения работ по тестированию.

● Привести результаты тестирования.

Результаты будут оформлены в виде отчетов. Все найденные ошибки будут отслеживаться с помощью баг-трекинговой системы.

# 2. Рамки проекта

## 2.1. Область тестирования сайта

В объем работ по тестированию сайта входит тестирование следующих

компонентов и функций:

1. Профиль сотрудника;

# 3. План работы

План работ будет состоять из:

1. Подготовка плана тестирования.

2. Согласование тест-плана.

3. Проведение тест-анализа

4. Составление тестовой документации (чек-листы, тест-кейсы)

5. Исполнение функциональных и нефункциональных тестов. Написание отчетов об ошибках.

# 4. План и стратегия тестирования

## 4.1 Уровни и стратегия тестирования

В процессе тестирования будут выполнятся тесты на уровне интеграционного и системного тестирования используя стратегию чёрного ящика и серого ящика.

Способы генерации тестовых сценариев на основании технического задания, декомпозиции требований и используя методы разбиения требований на классы эквивалентности и граничных значений. Также, генерация тестовых случаев на основе исследовательского тестирования.

## 4.2. Функциональное тестирование

Цель функционального тестирования состоит в том, чтобы убедиться, что весь программный продукт (веб-приложение) работает в соответствии с требованиями, и в приложении не появляется существенных ошибок. Функциональное тестирование является наиболее существенной частью тестирования программного обеспечения, включающее в себя проверку различных аспектов системы. Программный продукт должен пройти все запланированные тесты. Только в этом случае можно быть уверенным в его качестве.

Цель тестирования: Обеспечение надлежащего качества целевой функциональности.

|  |  |
| --- | --- |
| Техника: | Выполнение каждого сценария, используя допустимые и недопустимые данные:  - При использовании заранее подготовленных валидных тестовых данных возникают ожидаемые результаты.  - При использовании невалидных данных отображаются соответствующие сообщения об ошибках или предупреждения.  - Каждая функциональность проверена. |
| Входные критерии: | - ТЗ первого спринта полностью сформировано;  - Разработчики реализовали весь запланированный на 1 спринт функционал;  - Составлена вся необходимая тестовая документация;  - Подготовлено тестовое окружение, приложение готово к тестированию на тестовой площадке. |
| Критерий приемки: | - Все критические тесты пройдены;  - Нет блокирующих багов;  - Все баги с средним приоритетом и выше, устранены;  - Тесты пройдены;  - Пройдено приемочное тестирование. |

## 4.3 Нефункциональное тестирование

Целью нефункционального тестирования является оценка и обеспечение качества аспектов, которые не относятся непосредственно к функциональности системы, но имеют важное значение для общего опыта пользователя. Нефункциональное тестирование в проекте “NDA” включает проверку графического интерфейса и его адаптивности, удобства использования, кроссбраузерности и кроссплатформенности.

Тестирование графического интерфейса:

Цель тестирования графического интерфейса состоит в проверке визуального представления веб-приложения, правильном отображении элементов пользовательского интерфейса и соответствии дизайну и стилю оформления при использовании популярных разрешений экрана, таких как 1920x1080, 2560x1600 и 2880x1800. Графический интерфейс является первым впечатлением пользователя о приложении и должен быть интуитивно понятным и привлекательным.

Тестирование удобства использования:

Цель тестирования удобства использования заключается в проверке, насколько легко и эффективно пользователь может взаимодействовать с веб-приложением. В процессе тестирования оценивается уровень интуитивности интерфейса, доступность функциональности, понятность инструкций и общая удобство работы с приложением. Это помогает обеспечить положительный пользовательский опыт и повысить удовлетворенность пользователя.

Тестирование кроссбраузерности:

Цель тестирования кроссбраузерности заключается в проверке, как приложение отображается и функционирует на различных веб-браузерах. Веб-приложение должно работать одинаково хорошо на популярных браузерах, таких как Google Chrome, Yandex Browser и Safari. Тестирование кроссбраузерности помогает обнаружить и исправить любые несовместимости или проблемы, связанные с отображением и функциональностью на разных браузерах.

Тестирование кроссплатформенности:

Цель тестирования кроссплатформенности заключается в проверке работы веб-приложения на разных платформах, таких как Windows и MacOS. Приложение должно быть совместимо и функционировать стабильно на различных платформах, обеспечивая единый пользовательский опыт независимо от выбранной платформы. Тестирование кроссплатформенности помогает выявить и устранить возможные проблемы, связанные с отличиями в платформенных особенностях.

Нефункциональное тестирование графического интерфейса, удобства использования, кроссбраузерности и кроссплатформенности является важной частью общего процесса тестирования. Правильное функционирование и соответствие этим аспектам помогают создать качественное приложение с высоким уровнем удовлетворенности пользователей. Описанный тест-план предоставляет основу для проведения тестирования и обосновывает необходимость проверки каждого из этих аспектов для достижения желаемого качества приложения.

## 

## 4.4. Глубина тестирования

В рамках проведения функционального и нефункционального тестирования запланированы проверки не только фронтенда, но и бэкенда - тестирование API, тестирование БД на корректную работу.

## 4.5. Процедура тестирования

Необходимо проверить различные аспекты тестируемого программного обеспечения, для этого требуется выполнение следующих типов тестирования:

По видам тестирования:

- Функциональное тестирование;

- Нефункциональное тестирование:

- Тестирование пользовательского интерфейса;

- Юзабилити-тестирование;

- Тестирование кроссбраузерности;

- Тестирование кроссплатформенности.

По стадии тестирования:

- Смоук тестирование;

- Общее тестирование;

- Повторное тестирование;

- Приёмочное тестирование.

## 4.6. Сообщения об ошибках

Отчеты об ошибках создаются для того, чтобы предоставить команде разработчиков и менеджерам проекта исчерпывающую информацию об обнаруженных ошибках. Они должны быть полезны при определении причин ошибок и их исправлении.

Степень приоритета ошибок можно разделить на четыре категории:

- Critical. Наивысший приоритет. Назначается экстренным ситуациям, которые очень отрицательно влияют на продукт или даже бизнес компании. Такие баги нужно устранять немедленно.

- High. Высокий приоритет. Назначается багам, которые должны быть устранены в первую очередь.

- Normal. Обычный приоритет, назначается по умолчанию. Эти баги устраняются во вторую очередь, в штатном порядке.

- Low. Низкий приоритет. Назначается багам, не влияющим на функционал. Исправление таких багов происходит в последнюю очередь, если есть время и ресурсы.

Приоритет бага сперва определяет инициатор, но в дальнейшем он корректируется менеджером проекта. Именно менеджер имеет общее представление о тестируемой системе и понимает, насколько срочно нужно исправить тот или иной баг.

Каждый баг-репорт будет содержать следующую информацию об ошибке:

- ID номер;

- Заголовок, представляющий собой краткое описание проблемы;

- Описание (если требуется);

- Шаги для воспроизведения ошибки;

- Ожидаемый результат;

- Фактический результат;

- Окружение, операционные системы, браузеры или версии приложений, в которых возникает ошибка.

- Критичность;

- Приоритет её исправления;

- Дополнительная информация о дефекте в виде прикрепленных скриншотов, видеозаписей, логов.

Стороннее программное обеспечение будет использоваться для отчетности и поддержки обнаруженных ошибок. Команда тестирования зарегистрирует в этом программном обеспечении все дефекты, которые будут обнаружены в процессе тестирования.

# 5. Ресурсы.

Человеческие ресурсы (Команда тестирования в составе 1 человека)

|  |  |
| --- | --- |
| **Роль / ФИО** | **Обязанности** |
| QA Engineer - Бардюков Данил | - Оценка задач спринта на полноту и непротиворечивость;  - Оценка трудозатрат на тестирование;  - Актуализация плана тестирования;  - Тест-анализ требований;  - Формирование чек-листов, тест-кейсов;  - Исполнение тестов и документирование багов.  - Приёмочное тестирование; |

## 5.1. Инструменты

Будут использованы следующие инструменты:

|  |  |
| --- | --- |
| **Наименование процесса** | **Инструмент** |
| Баг-трекинговая система | qase.io |
| Система управления тестированием | qase.io |
| Захват изображений/видео | Изображение - Joxi.  Видео - OBS Studio. |
| Инструменты для проверки API | Postman, Swagger |
| Инструменты для проверки UX/UI | Figma, DevTools |
| Инструменты для проверки функциональности | DevTools, Charles Proxy |

## 

## 

## 5.2. Список браузеров

|  |  |
| --- | --- |
| **Наименование браузера** | **Версия браузера** |
| Yandex Browser | Последняя |
| Google Chrome | Последняя |
| Safari | Последняя |

## 5.3. Список устройств

|  |  |
| --- | --- |
| **Устройство** | **Операционная система** |
| ПК (настольный, ноутбук) - 1 шт | Windows 10 |
| ПК (настольный, ноутбук) - 2 шт | Mac OS 13 Ventura |

# 6. Критерии качества

Продукт должен работать в соответствии с требованиями и техническим заданием.

Продукт не должен содержать критических и блокирующих дефектов в окончательной версии проекта.

# 7. Риски процесса тестирования

Следующие проблемы могут повлиять на результаты тестирования:

- Изменения и модификации программного продукта, которые не были запланированы и не обсуждались заранее с командой тестирования.

- Изменения в требованиях к программному обеспечению, которые не были предварительно обсуждены с командой тестирования.

- Задержки в исправлении/исправлении ошибок.

- Задержки в предоставлении новых сборок команде тестирования.

- Изменение состава команды тестирования (замена, отпуск, болезнь)

- Необходимость повторной отработки практических навыков для работы с инструментами.

# 8. Ожидания команды тестирования

- Команда тестирования должна быть обеспечена действительными, обновленными документами в течение всего процесса тестирования.

- Все критические ошибки должны быть исправлены как можно скорее.

- Разработчики должны исправить все ошибки в программных модулях перед выпуском новой версии.

# 9. Обязанности участников тестовой группы

Менеджер проекта

- Управление процессом тестирования.

- Предоставление всех необходимых ресурсов для проведения испытаний.

Тестировщики

- Процесс контроля качества, регистрация обнаруженных ошибок в утвержденной системе отслеживания ошибок.

# 10. Результаты

Результатом работы команды QA будут являться:

- План тестирования;

- Количество написанных проверок;

- Количество проведённых тестов;

- Количество успешно пройденных тестов;

- Количество исправленных и закрытых ошибок;

- Отчеты об ошибках и отчеты о ходе тестирования.

- Ссылка на qase.io;

- Вывод по результатам работы.