

## 1. Область тестирования

В объем работ по тестированию Системы входит тестирование новых версий, готовящихся к выпуску и уже существующего функционала на площадках «Test» и «Prod».

Тестирование осуществляется в рамках следующих областей:

1. Регистрация пользователя, авторизация в Системе, восстановление пароля;
2. Заполнение профиля, личных данных и данных о компании, настройка нотификации;
3. Внесение данных о банковских картах, оплата, настройка тарифного плана;
4. Подписание документов;
5. Создание, удаление и работа с загруженными документами в различных статусах;
6. Создание, работа и удаление шаблонов, шаблонов API;
7. Создание, работа и удаление форм многократного использования;
8. Приглашение, удаление и работа с приглашенными участниками команды;
9. Интеграции с различными системами работы с документами;
10. Прочие области.

Ручное тестирование осуществляется методом Черного ящика<sup>i</sup>. Для планирования процессов разработки и регистрации ошибок используется система регистрации ошибок (Mantis/Kaiten/Jira например).

Тестирующий приступает к созданию сценариев и написанию тест-кейсов по функционалу предыдущей версии системы и, если это возможно, по ближайшей будущей. Тест-кейсы создаются в QASE (например) на ту часть функционала, которая нуждается в постоянной проверке в последующих релизах. В случаях, если написание тест-кейсов до разработки функционала невозможно, они создаются в процессе непосредственной проверки интерфейса.

### 1.1. Основные методы ручного тестирования и виды

При проведении ручного тестирования воссоздаются сложные последовательности действий пользователей, которые невозможно воспроизвести с помощью автотестов. Также при ручном тестировании проверяется удобство интерфейса для реального пользователя. С помощью ручного тестирования проверяются нетривиальные кейсы и дополнительные сценарии, используя исследовательский и ad-hoc<sup>ii</sup> методы.

В рамках тестирования всегда выполняется:

- Компонентное (с точки зрения архитектуры)<sup>iii</sup>
- Функциональное тестирование.<sup>iv</sup>
- Тестирование пользовательского интерфейса.
- Юзабилити-тестирование<sup>v</sup> и тестирование локализации.
- Регрессионное тестирование<sup>vi</sup>
- Тестирование адаптивной верстки.<sup>vii</sup>
- Кросс браузерное тестирование

В рамках ручного тестирования не выполняется:

- Тестирование безопасности.
- Тестирование нагрузки.

Применяемые виды тестирования:

- Позитивные и негативные тесты
- Smoke-тесты, тесты критического пути, E2E
- Регрессионное тестирование и тестирование новых функциональностей

### 1.2. Применяемые техники и технологический стек

Для тестирования областей функционала используются следующие техники тест-дизайна:

1. Эквивалентное разбиение и граничные значения
2. Причинно-следственный анализ
3. Прогнозирование ошибок
4. Парное тестирование

### **Подробнее о тестировании каждого выделенного модуля:**

Общий функционал тестируется на соответствие макетам, переключение между табами, наличие опечаток, работоспособность табов переключения модулей и тд. Также тестируются (если известны) запросы API без использования UI.

#### **1. Регистрация пользователя, авторизация в Системе, восстановление пароля;**

Тестирование этой области проводится в первую очередь при первом знакомстве с системой. Тестируются соответствие макету, верстка при различном расширении/размере окна, поля логин/пароль, все имеющиеся баттоны и ссылки, содержание писем, точность ссылок, безопасность паролей, в целом соответствие формы требованиям, корректность валидации полей (размер вводимого значения, валидные/невалидные символы, корректность алертов и т.д).

Заполнение профиля, личных данных и данных о компании, настройка нотификации;

#### **2. Заполнение профиля, личных данных и данных о компании, настройка нотификации, внесение данных о банковских картах, оплата, настройка тарифного плана;**

После авторизации осуществляется переход к тестированию профиля и настроек. Тестируется соответствие макету, верстка при различном расширении/размере окна и адаптивность интерфейса, если это необходимо. Поля, селекты, чек-боксы и баттоны в блоке профиля (с помощью описанных выше техник тест-дизайна), загрузка аватара/логотипа компании (загрузка валидных/невалидных файлов по размеру, типу файла), нотификация (настройка, корректность работы триггеров и содержания приходящих уведомлений), корректность алертов (точность кодов ошибки и обработка их, содержание текста алерта).

Блок с банковскими реквизитами и банковской картой кроме всего прочего тестируется особенно внимательно на соответствие требованиям безопасности. Так же тестируется привязка, отвязка банковской карты, проведение оплаты и тд.

Тестирование тарифных планов заключается в проверке корректности начисления стоимости тарифа, его изменения, изменения параметров внутри тарифа, проверяется соответствие доступных функций выбранному тарифному плану, и, наконец, отмена подписки. Кроме того тестируются тарифные планы Signaturely API, создание ключей, работа с ними и удаление.

Во всех разделах профиля тестируются элементы графического интерфейса и корректная их работа.

#### **3. Подписание документов;**

Так как это является основной функциональностью системы подписание документов нуждается в пристальной проверке. Баги в этом модуле могут стать блокирующими.

Тестируется загрузка файлов/документов разными способами (перетащить, кликнуть на «Upload file», использовать импорт из других систем), различных расширений и размеров. Тестируются ограничения по количеству файлов для загрузки, а так же удаление загруженного документа. Проверяется корректность выбора шаблона, если он есть и его функциональность. Проверяется доступность баттона «Sign the document» при различных условиях. И непосредственно подписание документа.

Тестируется создание подписи, инициалов. Проверяется работа всех элементов модального окна и элементов управления (копи-паст, назад вперед, изменение масштаба, выбор/загрузка подписи, установка штампов – дат, инициалов, чек боксов и тд).

При тестировании применяются описанные выше техники, виды и методы тест-дизайна.

#### **4. Создание, удаление и работа с загруженными документами в различных статусах;**

Во вкладке «Documents» проверяется наличие загруженных документов, соответствие статусов и корректный переход между ними, функциональность всех доступных опций и их корректная отработка (изменение наименования файла/папки, перемещение, редактирование, изменение, просмотр истории активности, удаление). Тестируется корректность работы поиска и сортировки, создание папок и работа с ними.

Проверяется доступность и отклик всех элементов управления, баттонов, скроллов и тд.

#### **5. Создание, работа и удаление шаблонов, шаблонов API;**

В этой области тестируется аналогично другим элементы графического интерфейса. Проводится их верификация и валидация.

Тестируется создание шаблона, создание шаблона API (если это предусмотрено тарифным планом, если нет проверяется корректность уведомления об этом и переход к обновлению тарифного плана). В форме создания нового шаблона тестируются поля (валидные/невалидные данные, обработка ошибок и тд), добавление ролей, их удаление, загрузка файлов/документов (возможно элемент загрузки переиспользуется). Проверяется доступность баттона «Fill Template» при различных условиях.

В модальном окне тестируется вся доступная функциональность, корректность отображения документов и т.д.

Тестируются взаимосвязанные области – компоненты системы, где используются шаблоны (проверяется что шаблон отображается в выпадающих списках, что он активен, что его можно применить, проверяется что после удаления шаблон не отображается, не активен и его нельзя применить и тд)

#### **6. Создание, работа и удаление форм многократного использования;**

*(Чем форма от шаблона отличается не очень поняла)*

Данный модуль тестируется аналогично шаблонам.

#### **7. Приглашение, удаление и работа с приглашенными участниками команды;**

В модуле приглашения участников тестируется непосредственно функциональность отправки приглашения (самой формы, корректности отправки письма куда надо, проверка того, что сумма за каждого участника плюсуется к тарифу и тд) и принятие приглашения (содержание письма, корректность ссылки, открытие ссылки другим пользователем, вход приглашенного участника и тд).

Тестируется возможность смены роли приглашенного участника, наличие ограничений функционала в зависимости от роли и тд. Проверяется возможность удаления участника и наличие доступа к кабинету этого участника после удаления из команды.

#### **8. Интеграции с различными системами работы с документами;**

Тестируется сама форма (UI), активность баттонов. Тестируется активация интеграций, корректность переходов, содержание запросов, которыми обмениваются системы.

Технологический стек:

Web-браузеры	Yandex, Safari, Edge, Mozilla Firefox и Opera. Используются последние версии каждого браузера. (Можно использовать BrowserStack)
Операционные системы	Windows, IOS, Android
Настройки браузера и ОС	<ul style="list-style-type: none"><li>• Стандартный / крупный шрифт</li><li>• Масштабирование 100%, 75%, 50%</li><li>• Размер окна браузера – от 100% до 25%</li></ul>
Хранение документации (тест-кейсы, ТЗ)	Gitlab/QASE или др.
Фиксация ошибок	Mantis или др.

Основные параметры ручного тестирования представлены в таб. 1:

Цель:	Обеспечение надлежащего качества работы функционала Системы
Техника:	Проверка сценариев, описанных в тест-кейсах с допустимыми и недопустимыми данными: <ul style="list-style-type: none"> <li>• Ожидаемый результат возникает при использовании заведомо правильных (валидных) данных;</li> <li>• Соответствующие уведомления об ошибках возникают при использовании заведомо неправильных (невалидных) данных;</li> <li>• Каждый шаг воспроизводится.</li> </ul>
Входные критерии:	Подготовлено тестовое окружение, тестируемая задача помещена на тестовый стенд, тестовые данные подготовлены.
Критерии приемки:	<ul style="list-style-type: none"> <li>• Все дымовые тесты<sup>viii</sup> пройдены;</li> <li>• Блокирующих дефектов не найдено;</li> <li>• Все дефекты с высоким приоритетом исправлены;</li> <li>• Ручные тесты пройдены.</li> </ul>

Таблица 1 «Параметры ручного тестирования»

Тестирование функционала системы осуществляется ручным и автотестирующим. Их взаимодействие обусловлено следующими принципами:

- Ручные и автотесты не пересекаются по функционалу, соответственно не происходит дублирования.
- Ручное тестирование производится по тому функционалу, по которому невозможно произвести автотестирование.
- Длинные бизнес-процессы производятся также ручным способом.
- Стабильный функционал покрывается автотестами, новый тестируется в ручном режиме.
- Ручной тестирующий покрывает функционал тест-кейсами, автотестирующий автоматизирует их, в случае необходимости дорабатывает или корректирует их.

<sup>i</sup> Тестирование чёрного ящика или поведенческое тестирование — стратегия (метод) тестирования функционального поведения объекта (программы, системы) с точки зрения внешнего мира, при котором не используется знание о внутреннем устройстве (коде) тестируемого объекта.

<sup>ii</sup> Свободное тестирование (ad-hoc testing) - это вид тестирования, который выполняется без подготовки к тестированию продукта, без определения ожидаемых результатов, проектирования тестовых сценариев.

<sup>iii</sup> Компонентное тестирование - тип тестирования ПО, при котором тестирование выполняется для каждого отдельного компонента отдельно, без интеграции с другими компонентами.

<sup>iv</sup> Функциональное тестирование — это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи, нужные пользователям.

<sup>v</sup> Юзабилити-тестирование или тестирование удобства — это проверка удобства продукта для конечных пользователей.

<sup>vi</sup> Регрессионное тестирование – это повторное выполнение функциональных и нефункциональных тестов для обеспечения того, чтобы ранее разработанное и протестированное программное обеспечение все еще работало после изменения.

<sup>vii</sup> Тестирование адаптивной верстки - просмотр сайта на разных разрешениях экрана, в разных браузерах и устройствах.

<sup>viii</sup> Дымовое тестирование (smoke testing) – это тип тестирования, направленный на проверку главной функциональности, неработоспособность которой делает бессмысленной саму идею использования приложения (или иного объекта, подвергаемого дымовому тестированию).