

Итоговая работа

Согласно ТЗ, следует определить наборы проверок, необходимых для покрытия всех основных положительных сценариев, для этого проведём Smoke-тестирование системы по флоу.

Тестирование проводится поэтапно:

- модульное тестирование клиентского приложения
- интеграционное тестирование Клиент - Внешняя БД
- интеграционное тестирование Клиент - Локальная БД
- интеграционное тестирование Локальная БД - Внешняя БД
- системное тестирование
- нефункциональное тестирование (тестирование производительности и отказоустойчивости)
- UI/UX тестирование
- тестирование локализации
- тестирование безопасности

Загрузка приложения на мобильное устройство

На данном этапе необходимо уточнить, в рамках каких операционных систем доступно приложение. Будем считать, что приложение доступно для самых популярных мобильных ОС (Android, iOS).

Проведем модульное Monkey-тестирование клиентского приложения методом Black Box.

Позитивные проверки:

Проверка	Ожидаемый результат
Доступность для скачивания приложения в AppStore и Google Play	Приложение доступно для скачивания в AppStore и Google Play
Загрузка приложения при достаточном объеме памяти на мобильном устройстве	Приложение загружено на мобильное устройство

<p>Загрузка приложения при недостаточном объеме памяти на мобильном устройстве.</p> <p><i>Для реализации данного этапа в Android можно воспользоваться одним из эмуляторов (например, Android Studio)</i></p>	<p>Приложение не загружено на мобильное устройство, на экране появилось сообщение о недостаточном объеме памяти</p>
<p>После скачивания приложение появилось на мобильном устройстве</p>	<p>Приложение появилось в списке всех приложений мобильного устройства</p>
<p>Доступность открытия и закрытия приложения</p>	<p>Приложение открывается и закрывается без ошибок</p>
<p>Доступность сворачивания приложения и возврат в приложение после сворачивания</p>	<p>Приложение сворачивается и разворачивается без ошибок</p>
<p>Доступность удаления приложения</p>	<p>Приложение удалено</p>

Функциональное тестирование клиентского приложения

Позитивные проверки:

Проверка	Ожидаемый результат
Наличие поля ввода ИНН на форме	Поле ввода ИНН присутствует на форме
<p>Ввести существующий валидный ИНН ИП в поле ввода (12 цифр).</p> <p>Предварительные шаги: запросить валидные данные у аналитика или</p>	ИП найден в системе

попросить администратора создать их для возможности проверки данного шага.	
Ввести существующий валидный ИНН физического лица в поле ввода (12 цифр)	ИП не найден в системе, представлено сообщение об ошибке «Нет открытых ИП»
Проверить на наличие грамматических и орфографических ошибок в названии поля ввода, подсказок (при их наличии) и в тексте ошибок. Уже на данном этапе в ТЗ замечен дефект в документации	<i>«Нет открытых ИП»</i> Фактический результат: <i>«Нету открытых ИП»</i>
Ввести несуществующий ИНН ИП в поле ввода в правильном формате (12 цифр).	ИП не найден в системе, представлено сообщение об ошибке «Нет открытых ИП»
Работа приложения через стабильный мобильный Интернет	Приложение работает стабильно, все ЭФ прогружаются без задержек
Работа приложения через нестабильный мобильный Интернет	Приложение работает, время отклика увеличено, сообщение о нестабильном соединении
Работа приложения при стабильном Wi-Fi	Приложение работает стабильно, все ЭФ прогружаются без задержек
Работа приложения при слабом Wi-Fi	Приложение работает, время отклика увеличено, информационное сообщение о нестабильном соединении
Работа адаптивной верстки при смене ориентации экрана	Верстка работает корректно

Так как по условию ТЗ для этого модуля системы негативное тестирование проводить не требуется, негативные тестовые сценарии не прописаны.

При необходимости составления негативных тест-кейсов следует опираться на такие техники тест-дизайна, как Граничные значения (логическая, технологическая и произвольная границы для поля ввода), Классы эквивалентности, Ноль / Не ноль.

Интеграционное тестирование Клиент - Внешняя БД

Прямого доступа к базе данных УФНС нет, поэтому для интеграционного тестирования приложения с внешней БД необходимо провести тестирование API посредством HTTP запросов.

Проводить тестирование на релизной версии нельзя, поэтому для возможности проведения API-тестирования организовывается заглушка. С помощью SoapUI генерируется Mock-service с ответами, идентичными ответам от реального сервера УФНС.

БД УФНС доступна только для чтения, поэтому для позитивного тест-кейса достаточным будет только GET-запроса HTML с указанием URL заглушки.

С целью проверки наличия записей с введенными данными в БД УФНС воспользуемся GET-запросом. Для этих целей подойдет Swagger, SoapUI или Postman.

Для покрытия негативных сценариев в тесте (ИНН не найден в базе) необходимо указать ожидаемый код ошибки 404 (ошибка клиента) и сообщение «Нет открытых ИП» в возвращаемом json.

Также проверим доступность внесения изменений во внешнюю БД посредством остальных CRUD-операций (POST, PUT, DELETE requests). При положительном результате негативной проверки ожидается код ошибки сервера 501 (Not Implemented).

Негативные проверки	Ожидаемый результат
ИНН не найден в базе	Статус-код 404, сообщение об ошибке «Нет открытых ИП»
Запрос на ввод данных (POST)	Статус-код 501, ввод данных недоступен
Запрос на изменение данных (PUT)	Статус-код 501, ввод данных недоступен

Запрос на удаление данных из БД (DELETE)	Статус-код 501, ввод данных недоступен
--	--

Если при соединении с БД УФНС происходит какая-то ошибка/задержка/проблема, то пользователю должно выводиться сообщение о технической проблеме с просьбой проверить соединение попозже.

Проверки	Ожидаемый результат
При соединении с БД УФНС происходит ошибка/задержка/проблема	Наличие информационного сообщения об ошибке
Наличие орфографических и грамматических ошибок	Орфографических и грамматических ошибок в тексте информационного сообщения нет

Для покрытия сценария с попыткой повторного соединения при коде ошибки сервера 500 реализуется автотест с повторной попыткой соединения через таймаут.

При положительном ответе сервера (статус-код 200 ОК) проверяются:

Проверка	Ожидаемый результат
Открытие экрана с проверкой номера ОГРНИП	Экран с проверкой ОГРНИП открылся
Совпадение ОГРНИП	Открылся доступ к открытию расчетного счета в банке
Наличие открытого счета в локальной БД, используя SQL запрос SELECT в локальную БД	Результат запроса с записью в локальной БД

ОГРНИП не совпадает	На ЭФ присутствует кнопка для отправки отчета об ошибке
Кликабельность кнопки для отправки отчета об ошибке	Кнопка для отчета об ошибке кликабельна
Отправка отчета об ошибке	Отчет об ошибке отправлен
Проверка доставки отчета об ошибке в техподдержку	Отчет доставлен

Интеграционное тестирование Локальная БД - БД УФНС

Если в БД УФНС для пользователя произошли какие-то изменения, то в локальной БД они будут видны в течение суток. Для этих целей раз в сутки (в 23:59 по МСК) отработывает job, которая сравнивает данные в БД.

Прямого доступа в БД УФНС нет, поэтому для этого этапа тестирования также используется заглушка.

Производятся следующие проверки:

Проверка	Ожидаемый результат
Сервер внешней БД доступен (GET запрос)	Статус-код 200, база доступна для интеграции
Сервер локальной БД доступен	Статус-код 200, база доступна для интеграции
Сервер внешней БД недоступен (GET запрос)	Статус-код 500, сообщение об ошибке, повторная попытка соединения через таймаут
Данные в БД УФНС обновились.	Изменения найдены, job отработала

Сравнение данных в БД через job. Для проверки отработанной job можно изменить системное время, уведомив администратора	
Проверка отработанной хранимой процедуры (INSERT) через SELECT	Появились новые записи в локальной БД
Наличие внесенных данных в локальной БД через SELECT	Новые записи есть в локальной БД
Данные в БД УФНС не обновились.	Job отработала, изменения не найдены
Сервер локальной БД недоступен	Job не отработала, следующая попытка внесения изменения через сутки
Новые данные найдены, изменения в БД не добавлены	Сообщение об ошибке
Внесение уже добавленных ранее данных	Сообщение об ошибке "Данная запись уже внесена"
Внесение пустой записи (is null)	Сообщение об ошибке "Запись не содержит данные"
Создать уже существующий в БД счет	Сообщение об ошибке "Для данного ИП счет уже открыт"

Нефункциональное тестирование системы

При проведении тестов на производительность следует провести:

- **Нагрузочное тестирование** (при нормальных условиях).

Для тестирования нагрузки на сервер на протокольном уровне можно использовать Fiddler, jMeter, Load Runner. Для записи и перенаправления HTTP(S) запросов клиента на сервер необходимо настроить Proxy-сервер.

- **Тестирование стабильности** (длительная работа приложения)
- **Стресс-тестирование** (в условиях повышенной нагрузки): работа большого количество пользователей одновременно, создание большого количества записей и проверок.
- **Объемное тестирование** (увеличенный объем входящих данных).

Также необходимо проверить нагрузку системы на ресурсы мобильного устройства (память, энергопотребление) и на отказоустойчивость и восстановление.

Тестирование UI/UX

На данном этапе проводятся проверки, насколько удобно пользоваться приложением и системой в целом, насколько удобный для пользователя интерфейс приложения.

Необходимо также провести проверку, корректно ли отображается окно приложения на разных устройствах (кроссплатформенность) с разной диагональю экрана.

Тестирование локализации

Проверка локализации элементов форм, текста в ошибках.

Тестирование безопасности

Так как система работает с реальными пользовательскими данными и расчетным счетом, необходимо уделить особое внимание безопасности, в частности на устойчивость системы к SQL-инъекциям.

При тестировании безопасности нужно также провести проверку, что приложение не запрашивает доступ к контактам, фото, геопозиции и работе других приложений.