

Лабораторная работа 1.4 Тестовая документация (ч1)

Вид тестовой документации также зависит от ситуации на проекте и ожиданий заказчика.

План тестирования (Test Plan) — документ или совокупность документов, расписывающих все тестовую активность пределах одного проекта, все работы проводимые командой тестирования или одним тестировщиком. Поверхностно можно выделить моменты описываемые в тест-плане: объект тестирования, расписания работ, критерии начала и окончания тестирования, стратегию, риски и список проводимых работ.

Тест дизайн (Test Design) — процесс проектирования и создания тестовых случаев, для проведения в дальнейшем проверки ПО с учетом спецификации проекта и требований.

Тестовый случай (Test-case) — строго сформированный документ (на каждой организации могут быть свои правила, в той или иной мере похожие) с подробным описанием шагов и действий, для проверки одной части функционала, со строго заданными критериями прохождения.

Юзе-кейс (Use-case) — документ для тестирования продукта или ПО менее официален и используется для предугадывания и построения на основе предполагаемых сценариев использования путей тестирования приложения. Часто разрабатывается на основе бизнес задач и проверки критического пути тестирования.

Баг-репорт (Bug report) — документ, в котором описывается ситуация, последовательность шагов приведшая к возникновению ошибки на тестируемом ПО. Чаще всего требуется описание ожидаемого результата для такой конкретной последовательности шагов который будет являться правильным.

Чек-лист (Check-list) — документ который содержит в себе краткое описание функциональности всего приложения, для проверки его (ПО). Создание и написание чек-листа обычно заменяет написание тестовых случаев, в большинстве случаев ускоряет подготовку к проведению тестирования.

Спецификация требований (Software Requirements Specification) — это полное, четкое описание разрабатываемой программы. Ключевыми особенностями является полное описание всей функциональности программы, четкость требований, отсутствие неточностей в их описании, простота понятий, и достаточно четкая детализации.

Задание

1. Построить схему декомпозиции функциональности и прописать чек-листы.

Теоретические сведения

Итоговые паттерны

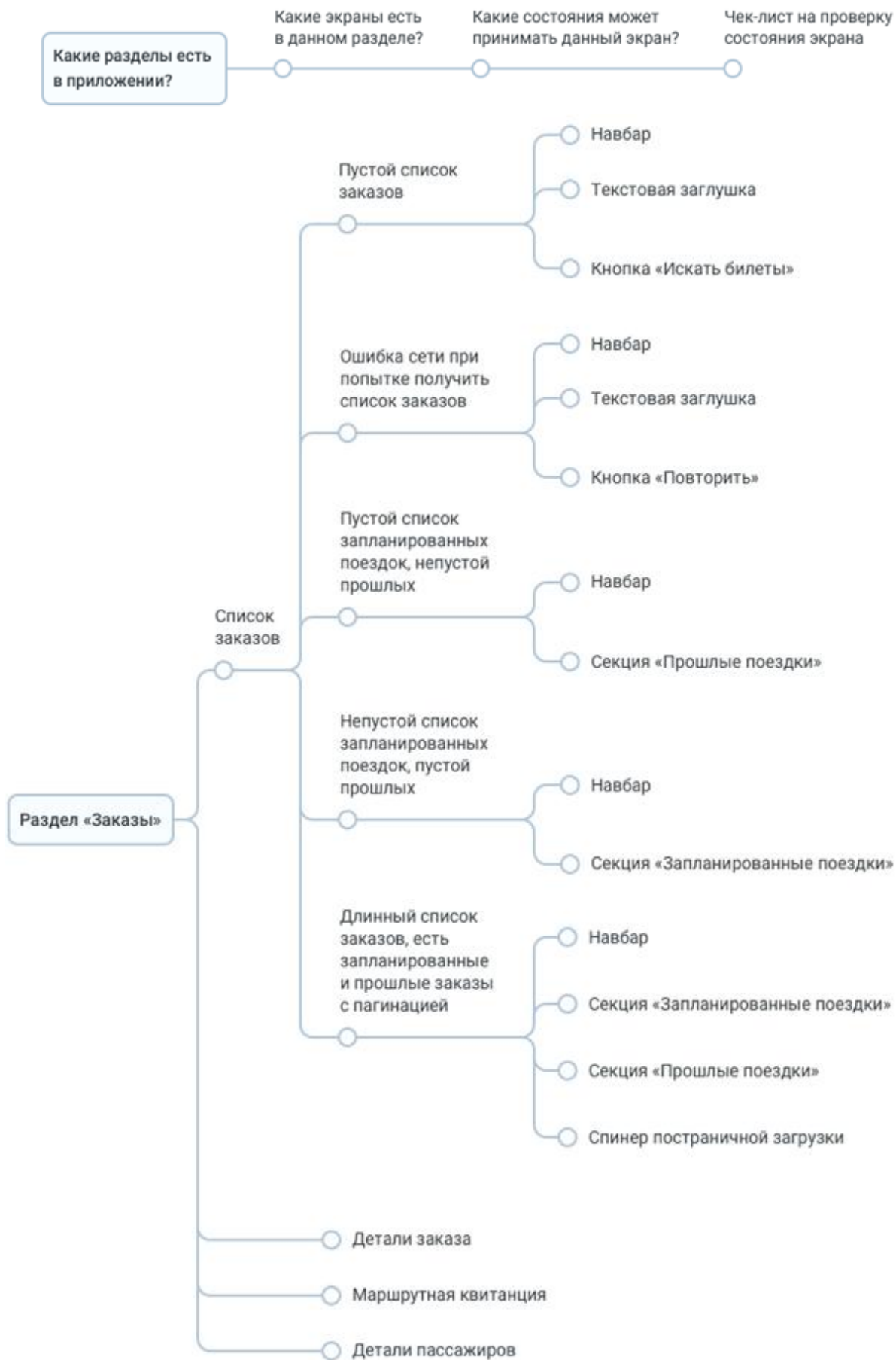
1. Экраны приложения

Чек-лист отличается от юз-кейса степенью подробности. В чек-листе вы не встретите подробных шагов кейса, для использования чек-листа при тестировании очень много информации нужно держать в голове в момент прогона тестов и знать логику работы приложения на отлично.

В целом идея была в том, чтобы прописывать те же самые пользовательские сценарии в виде дерева, в котором переход — это действие, а узел — это состояние, в котором оказывается приложение после этого действия.

Первый уровень дерева составляет список кейсов навсхемы, который обычно соответствует разделам приложения. Далее к ним подвешивается список экранов каждого раздела, к каждому экрану — список его состояний. В каждом узле дерева, начиная с третьего уровня, может содержаться чек-лист в табличном формате, описывающий каждый элемент дизайна и способы взаимодействия с ним. Если элементы дизайна сложные и имеют много состояний или на экране есть повторяющиеся элементы, можно декомпозировать еще глубже. Таким образом, *одна ветвь дерева описывает жизненный цикл одного экрана.*

В качестве примера приведена общая схема рассуждений при декомпозиции раздела заказов агрегатора авиабилетов.



К листьям этого дерева крепим короткие чек-листы. Так к каждому листу «навбар» линкуем чек-лист на элементы навбара для текущего экрана:

Навбар	
есть кнопка открытия бокового меню	по тапу -- открытие сайд-бара
есть название экрана "Мои заказы"	

А к каждому листу «секция запланированные поездки» линкуем чек-лист на проверку части списка с активными заказами:

Секция "Запланированные поездки"	
секция скроллится вместе со списком, не прилипает к навбару	
внутри секции список активных заказов	
переход к деталям заказа по тапу на элемент списка	
в списке у каждого заказа есть номер	
в списке у каждого заказа есть статус	"оплачен" или "необходима доплата"
если заказ оплачен, его статус отображается зеленым цветом	
если по заказу необходима доплата, он отображается красным цветом	
в списке у каждого заказа есть пункты назначения (города), через тире	
если названия городов длинные, происходит перенос текста на следующую строку	
в списке у каждого заказа есть даты поездки	-одна дата если заказ в одну сторону -две даты если заказ туда-обратно
возле каждой даты поездки есть двухбуквенное обозначение дня недели	
хотя бы одна из дат должна быть в будущем относительно текущей даты	
в списке у каждого заказа есть кол-во пассажиров	
в списке у каждого заказа есть цена	синим цветом

Критерии для выбора такого паттерна следующие:

- UI является приоритетным для заказчика
- минимум бизнес-логики на клиенте
- для приложения характерны кастомные элементы дизайна и анимации, сложные жесты
- на проекте нет других задокументированных требований кроме навсхемы

Лабораторная работа 1.5 Тестовая документация (ч2)

Задание

1. Декомпозировать функциональность на базе use-кейсам и составить чек-листы к ним

Теоретические сведения

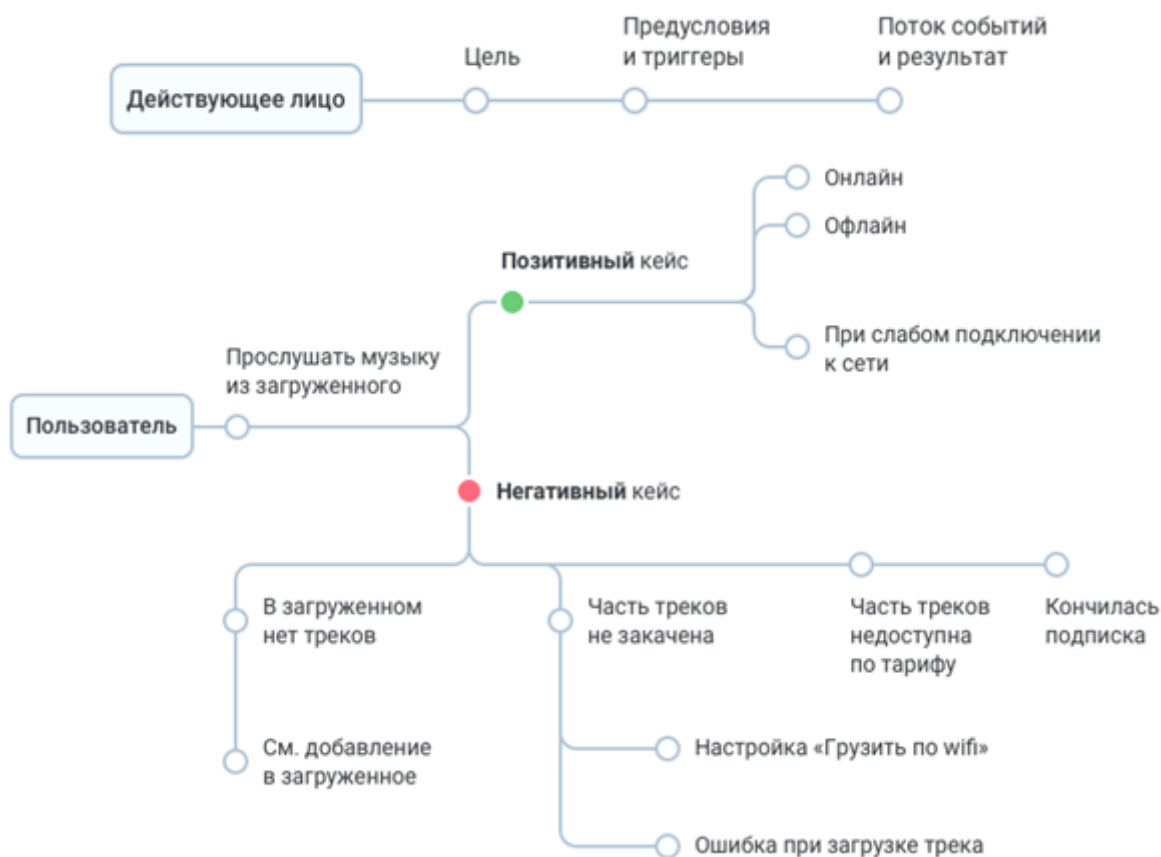
Итоговые паттерны

3. На базе Use cases

Бывают ситуации, в которых нерентабельно декомпозировать функциональность и проектировать тесты по двум описанным ранее схемам. Например, если мы хотим покрыть тестами длительную работу с приложением – как в случае с лентой соцсети или прослушиванием музыки в бекграунде. Или когда фича завязана на сценарии с малым количеством альтернатив – например, оформление подписки на контент. В таком случае пользуемся третьим паттерном, основанном на пользовательских сценариях.

Сначала декомпозируем функциональность по use-кейсам. Определяемся с тем, какие действующие лица могут участвовать в процессе работы с приложением и какие цели они могут перед собой ставить. За это будут отвечать первые два уровня нашего дерева. Далее пытаемся найти все возможные входные условия, которые могут повлиять на отработку сценария по достижению текущей цели, и структурируем их в дереве. Их так же удобнее всего делить на позитивные и негативные. Далее к каждому листу подвешиваем сценарный чек-лист на проверку функциональности, отвечающей за достижение цели.

В качестве примера ниже приведена схема для музыкального плеера с функцией загрузки треков для прослушивания офлайн:



Здесь ко всем листям позитивного сценария подвешиваем чек-лист, который нужно будет прогонять в условиях разных подключений к сети:

Прослушать несколько треков при активном приложении	треки переключаются, воспроизведение корректное
Скролл списка загруженных треков во время воспроизведения	плавный скролл, весь список загруженных треков доступен
Тапнуть на иконку текста на треке	отобразился и закэшировался текст трека
Залочить экран во время воспроизведения и прослушать несколько треков в бекграунде	треки переключаются, воспроизведение корректное
Свернуть приложение во время воспроизведения и прослушать несколько треков в бекграунде	треки переключаются, воспроизведение корректное
1. Выставить настройку don't keep activities 2. Свернуть приложение во время прослушивания музыки 3. Прослушать несколько треков в бекграунде	воспроизведение продолжается, треки корректно воспроизводятся и переключаются
Перемотать трек	состояние seekbar соответствует состоянию воспроизведения
Переключить на следующий трек	воспроизведение следующего трека в очереди воспроизведения
Переключить на предыдущий трек	воспроизведение предыдущего трека в очереди воспроизведения
Затапать кнопку переключения трека	воспроизведение n-го трека от последнего воспроизводимого, где n -- кол-во тапов
Переключить трек тапом по ячейке в очереди воспроизведения	воспроизведение выбранного трека
Включить шафл и переключить на следующий трек	формирование новой очереди воспроизведения по тапу на кнопку шафла, воспроизведение следующего трека в очереди
Отключить шафл и переключить на предыдущий трек	формирование новой очереди воспроизведения по тапу на кнопку шафла, соответствующей дефолтному порядку треков в списке загруженного; воспроизведение предыдущего трека в очереди
1. Включить последний трек в очереди воспроизведения 2. Включить заикливание 3. Переключить на следующий трек	воспроизводится первый трек в очереди воспроизведения
1. Включить последний трек в очереди воспроизведения 2. Отключить заикливание 3. Переключить на следующий трек	воспроизведение останавливается
Вытащить наушники во время воспроизведения	воспроизведение останавливается
Воткнуть наушники во время воспроизведения	переключение вывода аудиопотока в наушники
Изменить громкость воспроизведения	громкость корректно меняется
Запустить видео в другом приложении во время воспроизведения	отдача аудиопотока
Запустить аудио в другом приложении во время воспроизведения	отдача аудиопотока
Воспроизвести трек из загруженного во время воспроизведения аудио из другого приложения	перехват аудиопотока