

## Стратегия тестирования модуля «Собрать поездку» для мобильного приложения FUN&SUN

*user story: я, как QA инженер, хочу проанализировать имеющиеся требования от заказчика, дополнить их в соответствии с критериями качества требований и описать в документации. Составить тестовый сценарий и протестировать модуль на основании требований.*  
(Пункты стратегии будут дополняться по мере прохождения уроков).

- 1. Описание тестового объекта.** Новый модуль представляет собой конструктор поездок для самостоятельного конструирования пользователем выгодного пакета путешествия. Располагается вторым по счету в навигейшн-баре «Найти тур». Доступен авторизованным и неавторизованным пользователям. Основной функционал запускается по кнопке «Собрать поездку» и представляет собой выбор через два основных фильтра «Авиабилеты» и «Отели». Дополнительные фильтры доступны пользователю в разделе «Дополнительные услуги», их полный список может меняться, и поступает от back-end разработчиков.
- 2. Техническая документация.** Подробное описание тестовой задачи представлено: в файле «**Техническая задача.doc**», в графическом пользовательском сценарии «**Customer Journey Map FUN&SUN Mobile**», в двух mock-up верстках (export from Figma): **Fun&Sun\_1.pdf**, **TUI Plus 2020 из Sketch.pdf**. А также в диаграмме переходов между экранами: <https://miro.com/app/board/uXjVOcgMGxQ=/> (login: [elunyushkina@list.ru](mailto:elunyushkina@list.ru), password: 52qxc78Wu!5\$dUu).
- 3. Анализ необходимости того или иного уровня, и типа тестирования для нашего модуля.**

**Уровни тестирования** (решение: **да/нет/почему**):

**модульное (unit и контракт тестирование)** – нет, проводят разработчики.

**интеграционное:**

- **тестирование интеграции компонентов** – **да**, необходимо проверить совместимость нового модуля с приложением.
- **системное интеграционное (API)** – **да**, нужно перечислить все внешние системы, с которыми взаимодействует продукт, указать, какие данные передает и получает модуль, и проверить взаимодействие API.

**системное** – нет, т. к. на данном этапе нет ресурсов проверять работу всей системы целиком и проходить весь business flow приложения.

**приемочное** – **да**, можно выбрать пользовательское приемочное тестирование (**UAT**) и проверить – соответствует ли данный модуль ожиданиям группы конечных пользователей.

**Виды тестирования, которые будем применять к модулю (решение: да/нет/почему):**

**регрессионное** – нет. Поскольку в **Customer Journey Map** пользователя указано, что в конструкторе уже «дублируется функционал сайта», а значит, его проверяют при плановом тестировании всей системы, а также сроки и бюджет тестирования ограничены, то можно принять решение, что регрессионное тестирование проводить нецелесообразно. Его можно заменить регулярными проверками модуля с помощью чек-листов.

**функциональное:** частично да, **smoke** тестирование позволит проверить работу двух основных фильтров, тестирование **критического пути** – дополнительных фильтров. Расширенное функциональное тестирование модуля нецелесообразно.

**нефункциональное:**

**на отказ и восстановление** – нет, целесообразно проверять при плановом тестировании всей системы.

**performance testing** – частично да. Возможно, **нагрузочное** тестирование необходимо.

**usability** – нет, целесообразно ограничиться UAT тестированием.

**безопасности** – да, важно проверить сохранность пользовательских данных.

**инсталляционное** – нет.

**конфигурационное: кроссплатформенное и кросс браузерное** – да, на тех окружениях, которые уже используются в проекте.

**локализации/интернационализации** – нет, т.к. в конструкторе уже «дублируется функционал сайта», целесообразно проверять при плановом тестировании всей системы.

**доступности (accessibility)** – нет, т.к. в конструкторе уже «дублируется функционал сайта», целесообразно проверять при плановом тестировании всей системы.

#### **4. Роли и обязанности.**

Тестирование будет проводиться командой тестировщиков: Артем С., Виктор Кириллин, Елена Лун., Сергей Р., Павел Мурай, Юлия Ярмолович.

#### **5. Требования к окружению.**

Android Studio, снифферы траффика: Charles Proxy, Fiddler.

#### **6. Инструменты.**

Техника:

- Redmi Note 8 Pro (Android 11), MIUI Global 12.5.7.
- стационарный Asus (Windows 10), процессор AMD FX(tm)-8350 Eight-Core 4.00 GHz.

Проводится ручное тестирование, для этого используются инструменты: баг-трекинг-система (BTS) Jira и система управления тестированием (TMS) Qase.io.

- 7. Тестовая документация.** Тестовая стратегия. Чек-листы. Bug reports в Jira. Отчет о тестировании объекта командой тестировщиков.
- 8. Приоритеты тестирования.** Провести тестирование конструктора на smoke и critical path уровнях для проверки работы основного функционала. А также GUI тестирование для проверки графического интерфейса на соответствие требованиям по дизайну.

**9. Критерии начала и завершения тестирования.**

Для начала тестирования необходимо получить требования от заказчика в виде схемы переходов между экранами модуля, дизайн-макетов модуля. Тестирование на smoke и critical path уровнях должно иметь 100% результат положительного прохождения тест-кейсов. В противном случае процесс тестирования следует прекратить и отдать проект разработчикам для правки основного функционала. Тестирование GUI проводится параллельно и выявляет соответствие интерфейсов экранов конструктора дизайн-макетам. В случае несоответствия требованиям по дизайну модуль следует также отдать на доработку.

**10. Сроки.**

Сроки составления проектной документации (тестовой стратегии): 12-15 сентября (3 дня).

Сроки составления тестовой документации (чек-листов): 16-18 сентября (3 дня).

Сроки тестирования и составления отчетов о дефектах: 19.09.2022 – 24.09.2022 (6 дней).

**11. Оценка качества и мониторинг.**

В результате ручного тестирования функционала модуля на уровне smoke и critical path, а также проверки GUI в модуле найдены функционально-критичные дефекты и несоответствия интерфейса требованиям по дизайну. На текущий момент приложение не может быть отдано в релиз. Необходимы правки основного функционала. А также правки интерфейса экранов конструктора в соответствии с дизайн-макетами. Статистика: диаграмма Burndown, общая статистика по дефектам, статистика по новым дефектам, суммарная по тест кейсам, список функционально-критичных дефектов.

**13. Отчёт о результатах тестирования**

представлен в файле **FUN&SUN mobile app module testing report.pdf**.