[ВИДЕО ГАЙД НА ПЕРВУЮ ЧАСТЬ](https://drive.google.com/file/d/1-kIgeQ9_vAavRjt7u7cBWvZdNFU9r5O-/view?usp=sharing)

\*Скопируй этот документ и работай в копии. В названии напиши свое имя, фамилию, номер когорты, проект. Например: «Мария Петрова, 1-я когорта, дипломный проект».

# Письменные ответы на вопросы

\*помести сюда ответы на 10 вопросов из теоретического блока

Вопрос 1

Какую технику тест-дизайна ты используешь, чтобы составить тест-кейсы для задачи ниже? Обоснуй свой выбор.

Задача

На сайте застройщика нужно реализовать следующую функциональность: поиск квартиры для покупки по определенным параметрам. Все фильтры — с одиночным выбором. Варианты такие:

Название ЖК: «Золотое сияние», «Небесный полет».

Тип квартиры: студия, 1-комнатная, 2-комнатная, 3-комнатная, 4-комнатная.

Этаж: 1, 2, 3, 4, 5, 6, 7, 8, 9.

Отделка: черновая, чистовая.

Балкон: есть/нет.

Выбор техник тест-дизайна можно представить в виде алгоритма:

1.Определи объекты тестирования, опираясь на требования.

2.Задай контрольные вопросы:

* Здесь есть множество тестовых значений, которые можно разделить на группы?
* Объект тестирования может находиться в нескольких состояниях?
* Существуют ли переходы между состояниями объекта при выполнении определенных действий?
* Ожидаемый результат зависит от набора входных данных?
* У объекта тестирования есть несколько параметров с несколькими значениями?

В данном случае лучше всего подходит техника тест-дизайна - Метод попарного тестирования (pairwise testing), инструменты Teremokgames, Pict – используется для сокращения количества тестов при сохранении их эффективности, особенно при наличии множества входных параметров.

[Получится 36 уникальных комбинаций.](https://docs.google.com/spreadsheets/d/1WfcMPqdaMJILDzfYpwAUQjUxla3FhZtk/edit?usp=sharing&ouid=109067516353572920065&rtpof=true&sd=true)

Вопрос 2

Посмотри на тест-кейс ниже. Что стоит изменить и почему?

Заголовок:

Отображение главной страницы Яндекс Практикума для неавторизованного пользователя

Предусловие:

Пользователь не авторизован в системе Яндекс Практикума

Шаги воспроизведения:

Открой главную страницу Яндекс Практикума https://practicum.yandex.ru/

Ожидаемый результат:

Открылась главная страница для неавторизованного пользователя

Хорошо написанный тест-кейс включает в себя:

* Название тест-кейса: краткое описание теста.
* Идентификатор: уникальный номер или код для отслеживания.
* Предварительные условия (Описание): любые состояния или данные, необходимые для выполнения теста.
* Приоритет.
* Шаги выполнения: пошаговое руководство по проведению теста.
* Ожидаемый результат: описание того, что должно произойти при выполнении шагов.
* Фактический результат (после выполнения теста): фиксируется только при выполнении.
* Окружение: информация о программном и аппаратном окружении.

1. Изменим заголовок - “Отображение главной страницы для неавторизованного пользователя” убираем Яндекс Практикум так как это и так понятно из названия проекта и названия ссылки.
2. Добавить ID Тест-кейса, в дальнейшем тест-кейсы можно будет привязать к матрице-трассируемости.
3. В предусловия добавить пункт:

Очистить куки и кеш, если это влияет на состояние авторизации.

1. Добавить приоритет High/Low/Medium (важно при регрессе, если мало времени)
2. Добавить скриншот (после выполнения теста)
3. Подробнее описать Ожидаемый результат
4. Указать Окружение.
5. Подробнее описать Ожидаемый результат:

**Ожидаемый результат:**

1. Открылась главная страница Яндекс Практикума.
2. Отображаются разделы: "Курсы", "Отзывы", "Помощь".
3. Кнопка "Войти" видна и активна.
4. Кнопка "Выход" отсутствует.

### Вопрос 3

### Ты тестируешь чек-бокс «Заплатить сразу» в форме создания заказа. Ожидаемый результат — чек-бокс отмечен, в базе данных это поле содержит true. Но на самом деле в базе почему-то стоит false. Это ошибка — её нужно локализовать. Тебе известно, что архитектура приложения включает фронтенд, бэкенд и базу данных. Опиши, что нужно сделать в Devtools, чтобы локализовать ошибку.

### **1. Проверить взаимодействие фронтенда с бэкендом**

# **Network Tab:**

# Перейдите на вкладку **Network** в DevTools.

# Найдите запрос, который отправляется при нажатии на чек-бокс.

# Проверьте:

# Тип запроса (например, POST, PATCH, PUT).

# Корректность данных в теле запроса (payload).

# Ответ от сервера (response). Посмотрите, подтверждает ли сервер, что значение чек-бокса установлено в true.

# Если запрос отсутствует, это указывает на проблему на уровне фронтенда.

### **2. Проверить работу фронтенда**

# **Console Tab:**

# Перейдите на вкладку **Console**.

# Проверьте наличие ошибок JavaScript, связанных с чек-боксом.

# Убедитесь, что событие нажатия на чек-бокс триггерит нужные обработчики. Для этого:

# Используйте вкладку **Event Listeners** для элемента чек-бокса.

# Проверьте, вызывается ли нужный обработчик события (например, onClick).

# Используйте метод **debugger** или поставьте точки останова, чтобы убедиться, что правильные данные передаются в запрос.

### **3. Проверить работу бэкенда**

# **Response в Network Tab:**

# Проверьте, возвращает ли сервер корректный ответ (например, статус 200 OK).

# Проверьте содержимое ответа. Если сервер возвращает ошибку или некорректное значение, это указывает на проблему на стороне бэкенда.

### **4. Проверить взаимодействие бэкенда с базой данных**

# Если Network показывает, что запрос от фронтенда корректный, но база данных содержит false, вероятная причина — ошибка на стороне бэкенда:

# Логика обработки данных может быть неверной.

# Запрос в базу данных может быть неправильно сформирован.

# Запросите логи сервера у разработчиков, чтобы увидеть, какие данные отправляются в базу.

### **5. Дополнительные дейс[[1]](#footnote-1)твия**

# **Elements Tab:** Проверьте атрибуты чек-бокса (например, checked, value). Возможно, фронтенд не устанавливает значение checked, что приводит к ошибке.

# **Local Storage или Cookies:** Убедитесь, что данные, связанные с состоянием чек-бокса, не записываются в хранилище локально и корректно отправляются на сервер.

# Вопрос 4

# Возможна ли такая ситуация: отправленный GET-запрос удаляет всю информацию в базе данных? Обоснуй свой ответ.

# GET-запрос может удалить данные в базе только в случае ошибки или нарушения стандартов. По спецификации HTTP, метод GET должен быть "безопасным" и использоваться только для получения данных, не изменяя состояние сервера. Однако, если сервер настроен неправильно или разработчики ошиблись в логике, GET-запрос может вызвать удаление, например, из-за прямой привязки к функции удаления.

# Это нарушение REST-принципов и указывает на недочеты в архитектуре или реализации API. Чтобы избежать такого поведения, операции, изменяющие данные, должны выполняться через POST, PUT или DELETE, а также проходить строгую проверку.

### **Как избежать:**

1. Убедиться, что операции изменения данных реализуются через POST, PUT или DELETE.
2. Проверить логику сервера: GET должен только возвращать данные.
3. Настроить строгую авторизацию и защиту от CSRF для всех методов API.

Вопрос 5

В базе данных есть две таблицы. Какие поля стоит сделать первичными ключами, а какие — внешними? Обоснуй своё решение.

Таблица сотрудников employee с полями:

id — идентификатор сотрудника,

fio — ФИО сотрудника,

position\_id — идентификатор должности.

Таблица должностей position с полями:

id — идентификатор должности,

name — название должности,

salary — зарплата на данной должности.

**Первичные ключи**: Делает запись уникальной и позволит избежать полных дубликатов в таблице. Поля id в обеих таблицах уникальны и идентифицируют записи, поэтому они идеально подходят для ролей первичных ключей.

**Внешние ключи**: Это поле, которое отсылает к первичному ключу другой таблицы. Другими словами, внешний ключ указывает на запись другой таблицы. Связь между сотрудниками и их должностями должна быть четкой и контролируемой. Внешний ключ на position\_id гарантирует, что сотрудник может быть привязан только к существующей должности.

ФИО/name/salary не может быть первичным ключом так как могут быть однофамильцы, одинаковые имена, одинаковые зарплаты. Не уникальные поля.

Вопрос 6

Посмотри информацию из пятого вопроса и напиши запрос, который выведет данные в таком формате: ФИО сотрудника, название должности и зарплата.

SELECT employee.fio, position.name, position.salary

FROM employee

INNER JOIN position ON employee.position\_id = position.id;

Вопрос 7

Код ниже должен проверять, что оператор // действительно выполняет деление на целое число. Чего здесь не хватает и почему?

def test\_integer\_division():

a = 5//2;

a == 2

test\_integer\_division();

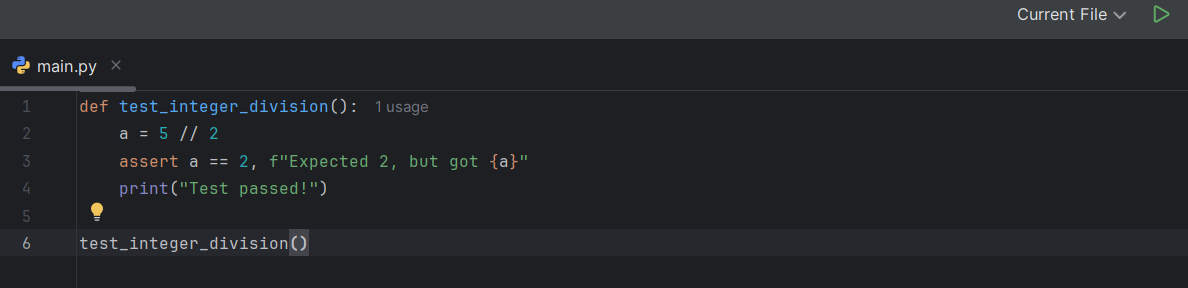
def test\_integer\_division():

a = 5 // 2

assert a == 2, f"Expected 2, but got {a}"

test\_integer\_division()

В данном коде пропущена проверка, которая должна убедиться, что результат деления соответствует ожидаемому значению. В Python для таких целей обычно используется ключевое слово **assert**, которое проверяет, выполнено ли условие, и если нет, выбрасывает ошибку.



Вопрос 8

В какой ситуации классы эквивалентности и граничные значения могут существовать по отдельности? Аргументируй свой ответ и приведи примеры.

Классы эквивалентности не имеющие числовых значений, не требуют проверки граничных значений. Например, статусы заказов могут включать: "новый", "в обработке", "отправлен", "доставлен", "отменён". Эти значения не имеют числовой последовательности, и их невозможно представить как линейный ряд, так как они не отражают какого-то количественного измерения. Вместо этого можно выделить два класса эквивалентности: валидный статус (например, "новый", "в обработке", "доставлен") и невалидный статус (например, "неизвестный", "ошибка", "отменён", если он не допустим на определённой стадии процесса).

В таком случае, граничные значения не применимы, так как мы не работаем с количественными или числовыми величинами, а с категориями, которые просто делятся на валидные и невалидные.

Вопрос 9

Можно ли исключить проверку в середине диапазона в пользу проверок на границах, входящих в диапазон? Аргументируй свой ответ.

Да, можно исключить проверку в середине диапазона, сосредоточив внимание на граничных значениях. Обычно ошибки валидации данных чаще происходят именно на границах диапазона, а не в его середине. Таким образом, проверка только крайних значений (например, минимального и максимального значения диапазона) уже предоставляет достаточную информацию для выявления большинства ошибок.

Пример: Для диапазона от 1 до 100 проверка на 1 и 100 уже будет эффективной, тогда как тестирование на 50 не всегда даст дополнительные результаты, если нет специфической логики, работающей именно с этим значением.

Такой подход экономит время и ресурсы, так как тесты на границах часто выявляют те же ошибки, что и тесты внутри диапазона.

Вопрос 10

Ты тестируешь мобильную версию Яндекс Аренды. Там обнаружился баг: при переходе в раздел «Показать на карте» появляется сообщение «Непредвиденная ошибка», и приложение закрывается. Разработчик просит воспроизвести баг и прислать логи. Как это сделать?

Сбор логов (для **Android**):

* Подключите ваше устройство к компьютеру и откройте **Android Studio**.
* Перейдите в раздел **Logcat** и отфильтруйте вывод по приложению Яндекс Аренда.
* Повторите шаги для воспроизведения ошибки, и в логе появятся данные, относящиеся к сбою приложения.
* Скопируйте логи и отправьте разработчику.

Сбор логов (для **iOS**):

* Подключите устройство к компьютеру с macOS и откройте Xcode.
* Перейдите в раздел **Devices and Simulators** и выберите ваше устройство.
* Откройте вкладку Console и повторите ошибку, чтобы увидеть все логи.
* Скопируйте информацию о сбое и отправьте разработчику.

Дополнительные шаги:

* Проверьте, включены ли в настройках приложения дополнительные логи или режим отладки.
* Перед отправкой логов, убедитесь, что вы отправили все данные с момента запуска приложения до сбоя.

Использование специальных инструментов для тестирования:

* **Firebase Crashlytics**: это инструмент, интегрируемый в мобильные приложения, который автоматически собирает логи с ошибками и сбоями. Разработчики могут легко просматривать информацию о сбоях, включая стек вызовов, устройство и версии приложения.
* **Sentry**: ещё одно популярное решение для отслеживания ошибок в реальном времени. Sentry поддерживает Android, iOS и другие платформы, позволяя отправлять подробные отчеты о сбоях.

# Отчёт о тестировании

\*заполни отчёт по образцу ниже

## Функциональное тестирование веб-приложения

Приложение проверено на стенде ([ссылка на стенд](https://4fd2ead4-7fbc-4cc6-8f78-1ee577918589.serverhub.praktikum-services.ru/)). Все известные требования были покрыты чек-листом: ([ссылка на google sheets с чек-листом](https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?usp=sharing)).

Результаты выполнения тестов можно посмотреть здесь: ([ссылка на google sheets с чек-листом](https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?usp=sharing)). Из 240 тестов в двух окружениях успешно прошло 198, не прошло — 42.

Список багов, найденных при тестировании, разбит по приоритетам:

1. Блокирующие (1):  
<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=14:14>

2. Критичные (2):  
<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=13:13>

3. Средний приоритет (13):  
<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=3:3>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=4:4>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=5:5>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=6:6>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=7:7>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=8:8>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=9:9>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=10:10>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=11:11>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=12:12>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=15:15>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=17:17>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=18:18>

4. Низкий приоритет (4):  
<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=19:19>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=20:20>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=21:21>

<https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=22:22>

Заключение:  
\*ответь на несколько вопросов ниже в свободной форме

1. Какой баг показался самым критичным?

[Блокирующий баг](https://docs.google.com/spreadsheets/d/1kxIWjTXBmoGGagHzfuAoEY6jKiyxgKnRTTQkvjRF1g4/edit?gid=1954882095#gid=1954882095&range=14:14) нельзя оформить заказ в Chrome = можно увольнять разрабов.

1. На твой взгляд, какая самая «хитрая» серая зона есть в требованиях?

Если указать станцию метро на Севере МСК, а адрес на юге МСК, куда повезет курьер самокат? Почему у этих двух полей нету связи. Как вернуть самокат, если самокат сломан, что делать ?

1. Проверенная тобой функциональность готова к релизу? Почему?

Не совсем понял вопрос, постарайтесь формулировать правильно. Проверенная? Мной? Релиз? Может быть после исправления багов и регресса имелось ввиду. Считаю вопрос немного не в тему. Я проверял часть функциональности, а приложение более масштабное, нужно смотреть в целом чтобы думать о релизе.

## Ретест багов в мобильном приложении

Был проверен фикс багов. Из них не исправлено 1, исправлено — 3, обнаружено 2 новых бага.

Список багов можно посмотреть здесь:

<https://docs.google.com/spreadsheets/d/1Yv7EkUSd9GNA1dtL0Oky9S7hev4JmN5CJzGXR3Hk7JQ/edit?gid=686196528#gid=686196528&range=3:3>

<https://docs.google.com/spreadsheets/d/1Yv7EkUSd9GNA1dtL0Oky9S7hev4JmN5CJzGXR3Hk7JQ/edit?gid=686196528#gid=686196528&range=4:4>

<https://docs.google.com/spreadsheets/d/1Yv7EkUSd9GNA1dtL0Oky9S7hev4JmN5CJzGXR3Hk7JQ/edit?gid=686196528#gid=686196528&range=5:5>

<https://docs.google.com/spreadsheets/d/1Yv7EkUSd9GNA1dtL0Oky9S7hev4JmN5CJzGXR3Hk7JQ/edit?gid=686196528#gid=686196528&range=6:6>

<https://docs.google.com/spreadsheets/d/1Yv7EkUSd9GNA1dtL0Oky9S7hev4JmN5CJzGXR3Hk7JQ/edit?gid=686196528#gid=686196528&range=7:7>

<https://docs.google.com/spreadsheets/d/1Yv7EkUSd9GNA1dtL0Oky9S7hev4JmN5CJzGXR3Hk7JQ/edit?gid=686196528#gid=686196528&range=8:8>

## Регрессионное тестирование мобильного приложения по готовым тест-кейсам

Результаты выполнения регрессионных тестов можно посмотреть здесь: ([ссылка](https://docs.google.com/spreadsheets/d/1YZzy45XUrQ-WvXQtHXNHpaYQzH-5sgT-0qHXxrSNARo/edit?gid=0#gid=0)).

Из 10 успешно прошло 1, не прошло — 9. Было обнаружено ещё 3 критических бага.

Список багов, найденных при тестировании, разбит по приоритетам:

1. Блокирующие:  
<https://docs.google.com/spreadsheets/d/1YZzy45XUrQ-WvXQtHXNHpaYQzH-5sgT-0qHXxrSNARo/edit?gid=404811733#gid=404811733&range=5:5>

2. Критичные:  
<https://docs.google.com/spreadsheets/d/1YZzy45XUrQ-WvXQtHXNHpaYQzH-5sgT-0qHXxrSNARo/edit?gid=404811733#gid=404811733&range=3:3>

<https://docs.google.com/spreadsheets/d/1YZzy45XUrQ-WvXQtHXNHpaYQzH-5sgT-0qHXxrSNARo/edit?gid=404811733#gid=404811733&range=4:4>

<https://docs.google.com/spreadsheets/d/1YZzy45XUrQ-WvXQtHXNHpaYQzH-5sgT-0qHXxrSNARo/edit?gid=404811733#gid=404811733&range=9:9>

<https://docs.google.com/spreadsheets/d/1YZzy45XUrQ-WvXQtHXNHpaYQzH-5sgT-0qHXxrSNARo/edit?gid=404811733#gid=404811733&range=11:11>

<https://docs.google.com/spreadsheets/d/1YZzy45XUrQ-WvXQtHXNHpaYQzH-5sgT-0qHXxrSNARo/edit?gid=404811733#gid=404811733&range=12:12>

3. Средний приоритет:  
<https://docs.google.com/spreadsheets/d/1YZzy45XUrQ-WvXQtHXNHpaYQzH-5sgT-0qHXxrSNARo/edit?gid=404811733#gid=404811733&range=6:6>

<https://docs.google.com/spreadsheets/d/1YZzy45XUrQ-WvXQtHXNHpaYQzH-5sgT-0qHXxrSNARo/edit?gid=404811733#gid=404811733&range=8:8>

<https://docs.google.com/spreadsheets/d/1YZzy45XUrQ-WvXQtHXNHpaYQzH-5sgT-0qHXxrSNARo/edit?gid=404811733#gid=404811733&range=10:10>

4. Низкий приоритет:  
<https://docs.google.com/spreadsheets/d/1YZzy45XUrQ-WvXQtHXNHpaYQzH-5sgT-0qHXxrSNARo/edit?gid=404811733#gid=404811733&range=7:7>

Заключение:  
\*ответь на несколько вопросов ниже в свободной форме

1. Какой баг показался самым критичным?

Баг который обнаружил не из тест-кейса, сервер перестает корректно взаимодействовать с мобильным приложением если в POST запросе передать значение null в необязательном поле color.

1. Такой продукт можно выпускать в релиз? Почему?

Нельзя, из 10 тест-кейсов 1 прошел, остальные с багами, баги критические связанные с крашем приложения и ошибками на стороне сервера.

Выводы о проделанной работе  
\*ответь на несколько вопросов ниже в свободной форме

Как для тебя прошла первая практическая часть проекта? С какими сложностями пришлось столкнуться? Что получилось хорошо, а что не очень? Какие мысли остались?

Прошла достаточно активно, старался покрыть максимально функциональность веб-приложения чек-листами, понимаю что можно ещё добавить проверок, используя попарное тестирование. Время = деньги. Расставляем приоритеты. При ретесте и регрессе помимо проверок по тест-кейсам, находил дополнительные Баги и фиксировал их баг-репортами.  
Пришлось уделить время чтобы вспомнить как тестируют веб-приложения на эмуляторе Android Studio. Был ряд вопросов к тест-кейсам для регресса, часть проверок блокировалось багами/требованиями из предыдущих зада. Много времени потратил чтобы понять что делать. Не ошибается тот - кто ничего не делает!

1. [↑](#footnote-ref-1)