

## Письменные ответы на вопросы

### Вопрос №1.

Для начала определимся с техникой тест-дизайна. В данном случае нам приходится комбинировать много различных вариантов входных данных. Входные данные связаны друг с другом. Поэтому, на мой взгляд подойдет техника попарного тестирования.

Плюс такой техники – она позволяет сократить количество тестов, и при этом сохранить качественное покрытие.

№	Название ЖК	Тип квартиры	Этаж	Отделка	Балкон
1	Золотое сияние	Студия	1	Черновая	есть
2	Небесный полет	1-комнатная	2	Чистовая	нет
3	Золотое сияние	2-комнатная	3	Черновая	нет
4	Небесный полет	3-комнатная	4	Чистовая	есть
5	Золотое сияние	4-комнатная	5	Черновая	нет
6	Небесный полет	Студия	6	Чистовая	есть
7	Золотое сияние	1-комнатная	7	Черновая	нет
8	Небесный полет	2-комнатная	8	Чистовая	есть
9	Золотое сияние	3-комнатная	9	Черновая	нет
10	Небесный полет	4-комнатная	1	Чистовая	есть
11	Золотое сияние	Студия	2	Черновая	нет
12	Небесный полет	1-комнатная	3	Чистовая	есть

13	Золотое сияние	2-комнатная	4	Черновая	нет
14	Небесный полет	3-комнатная	5	Чистовая	есть
15	Золотое сияние	4-комнатная	6	Черновая	нет
16	Небесный полет	Студия	7	Чистовая	есть
17	Золотое сияние	1-комнатная	8	Черновая	нет
18	Небесный полет	2-комнатная	9	Чистовая	есть
19	Золотое сияние	3-комнатная	1	Черновая	нет
20	Небесный полет	4-комнатная	2	Чистовая	есть
21	Золотое сияние	Студия	3	Черновая	нет
22	Небесный полет	1-комнатная	4	Чистовая	есть
23	Золотое сияние	2-комнатная	5	Черновая	нет
24	Небесный полет	3-комнатная	6	Чистовая	есть
25	Золотое сияние	4-комнатная	7	Черновая	нет
26	Небесный полет	Студия	8	Чистовая	есть
27	Золотое сияние	1-комнатная	9	Черновая	нет
28	Небесный полет	2-комнатная	1	Чистовая	есть
29	Золотое сияние	3-комнатная	2	Черновая	нет
30	Небесный полет	4-комнатная	3	Чистовая	есть

31	Золотое сияние	Студия	4	Черновая	нет
32	Небесный полет	1-комнатная	5	Чистовая	есть
33	Золотое сияние	2-комнатная	6	Черновая	нет
34	Небесный полет	3-комнатная	7	Чистовая	есть
35	Золотое сияние	4-комнатная	8	Черновая	нет
36	Небесный полет	Студия	9	Чистовая	есть
37	Золотое сияние	1-комнатная	1	Черновая	нет

Итого получается 37 уникальных комбинаций.

## **Вопрос №2.**

Хороший тест-кейс обязательно включает в себя: ID, Заголовок, Шаги, Ожидаемый и Фактический результат, Приоритет.

Опциональные элементы: Предусловие/постусловие, Окружение, Краткое описание, Ссылка на требования, Теги.

В данном тест-кейсе можно поправить:

- 1) Добавить ID теста (если это не делается автоматически);
- 2) Заголовок – можем убрать упоминание «Яндекс Практикума», так как из самого теста понятно, что тестируется. Оставить просто «Отображение главной страницы для неавторизованного пользователя»;
- 3) Добавить фактический результат;
- 4) Указать окружение;
- 5) Добавить скриншот или скринкаст;
- 6) Поставить приоритет.

## **Вопрос №3.**

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

1. Открыть страницу с формой создания заказа в браузере.
2. Открыть Devtools (например, в Chrome: Ctrl + Shift + I или F12).
3. Перейти на вкладку **Network**.
4. Отметить чек-бокс «Заплатить сразу» и отправьте форму.

5. Найти запрос, связанный с отправкой формы, в списке запросов на вкладке **Network**.
6. Проверить тело запроса (вкладка **Payload**). Убедится, что значение для чек-бокса передано как true.

Если значение передано правильно, проблема может быть на стороне бэкенда или базы данных. Если значение передано неправильно, проблема может быть во фронтенде.

## 2. Проверка работы бэкенда

1. Если значение в запросе передано правильно, проверить логи бэкенда. Убедится, что запрос успешно обработан, и данные сохранены в базе данных.
2. Если у есть доступ к коду бэкенда, добавьте логирование или отладочные точки, чтобы проверить, как обрабатываются данные чек-бокса.
3. Проверьте, правильно ли значение передается в базу данных.

## Проверить работу фронтенда

### Console Tab:

1. Перейти на вкладку **Console**.
2. Проверить наличие ошибок JavaScript, связанных с чек-боксом.
3. Убедится, что событие нажатия на чек-бкс триггерит нужные обработчики.

Для этого:

Использовать вкладку **Event Listeners** для элемента чек-бокса.

Проверить, вызывается ли нужный обработчик события (например, onClick).

4. Использовать метод **debugger** или поставите точки останова, чтобы убедиться, что правильные данные передаются в запрос.

## 3. Проверка взаимодействия с базой данных

1. Если логи бэкенда показывают, что данные переданы правильно, проверить базу данных.
2. Использовать SQL-запрос для проверки записи, созданной после отправки формы. Убедится, что поле, соответствующее чек-боксу, содержит true.
3. Если значение в базе данных false, проблема может быть в логике сохранения данных на стороне бэкенда.

## 4. Визуальная проверка

1. Перейти на вкладку **Elements** в Devtools.
2. Найти элемент чек-бокса в HTML-коде.
3. Убедится, что атрибут checked установлен, когда чек-бкс отмечен.
4. Проверить, что обработчик события чек-бокса (например, onChange) корректно обновляет значение в форме.

## Вопрос №4.

## 1. GET-запросы предназначены для получения данных

GET-запросы в HTTP предназначены для получения данных от сервера без изменения состояния данных на сервере. Это значит, что GET-запросы должны быть безопасным **GET-запросы не должны изменять данные на сервере.**

Таким образом, использование GET-запроса для удаления данных противоречит этим принципам.

## 2. Проблемы безопасности

Если GET-запрос может удалять данные, это создает серьезные проблемы с безопасностью:

- **Атаки методом сканирования:** Злоумышленники могут легко отправлять GET-запросы для удаления данных, просто перебирая URL-адреса.
- **Кэширование:** GET-запросы могут кэшироваться браузерами и прокси-серверами. Это может привести к случайному удалению данных, если кэшированный запрос будет выполнен повторно.

## 3. HTTP-методы для изменения данных

Для изменения данных, включая удаление, в HTTP предусмотрены другие методы, такие как: **DELETE, POST, PUT, PATCH.**

Эти методы не кэшируются и не считаются безопасными, что делает их более подходящими для операций, изменяющих данные.

## 4. Рекомендации по использованию HTTP-методов

Согласно спецификации HTTP, для операций, изменяющих данные, должны использоваться методы, отличные от GET. Это помогает предотвратить случайные или злонамеренные изменения данных.

## 5. Практические примеры

В реальных веб-приложениях GET-запросы обычно используются для получения данных, например:

- Получение списка товаров.
- Получение информации о пользователе.
- Получение страницы с контентом.

Операции, изменяющие данные, такие как удаление, добавление или обновление, выполняются с использованием методов POST, PUT, PATCH или DELETE.

## Заключение

Хотя теоретически возможно реализовать удаление данных с помощью GET-запроса, это противоречит стандартам и лучшим практикам разработки веб-приложений.

Использование GET-запросов для изменения данных создает проблемы с безопасностью и нарушает принципы безопасности и идемпотентности HTTP.

### Вопрос №5.

#### 1. Первичные ключи:

Поля id в обеих таблицах являются уникальными идентификаторами, которые обеспечивают целостность данных и позволяют быстро находить записи. Они должны быть первичными ключами, чтобы гарантировать уникальность каждой записи.

#### 2. Внешние ключи:

Поле position\_id в таблице employee должно быть внешним ключом, чтобы обеспечить связь между таблицами и гарантировать, что в этом поле всегда будет корректное значение, соответствующее существующей должности в таблице position. Это также помогает поддерживать целостность данных и предотвращает удаление записей из таблицы position, которые еще используются в таблице employee.

### Вопрос №6.

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

```
FROM employee
```

```
JOIN position ON employee.position_id = position.id;
```

### Вопрос №7.

В данном коде отсутствует утверждение, которое проверяет, что результат деления действительно равен ожидаемому значению. Для проверки в теста необходимо использовать функцию assert, которая позволяет проверить условие и выдать сообщение об ошибке, если условие не выполняется.

Вот исправленный код:

```
def test_integer_division():
```

```
    a = 5 // 2
```

```
    assert a == 2, f"Ожидалось, что 5 // 2 будет равно 2, но получилось {a}"
```

**Что было добавлено:**

- `assert a == 2`: Это утверждение проверяет, что результат деления a действительно равен 2. Если это условие не выполняется, тест завершится с ошибкой.
- `f"Ожидалось, что 5 // 2 будет равно 2, но получилось {a}"`: Это сообщение об ошибке, которое будет выведено, если утверждение не выполнено. Оно помогает понять, что именно пошло не так.

Таким образом, тест будет корректно проверять, что оператор // действительно выполняет целочисленное деление, и выдавать сообщение об ошибке, если результат не соответствует ожиданиям.

## **Вопрос №8.**

**Ситуация, когда классы эквивалентности могут существовать по отдельности:**

- Когда входные данные имеют четкие и понятные границы, но нет необходимости в детальном тестировании граничных значений. Например, если программа принимает возраст пользователя и на основе этого возраста принимает решения, можно разбить возраст на классы эквивалентности: дети (0-17 лет), взрослые (18-64 года), пожилые (65+ лет). В этом случае можно протестировать программу, используя один возраст из каждого класса, не заботясь о граничных значениях.

**Ситуация, когда граничные значения могут существовать по отдельности:**

- Когда входные данные имеют четкие границы, но нет необходимости в разбиении на классы эквивалентности. Например, если программа проверяет, что введенное число находится в определенном диапазоне, можно протестировать только граничные значения этого диапазона.

## **Вопрос №9.**

Проверка только на границах может быть более эффективной, так как она позволяет охватить ключевые точки, где ошибки наиболее вероятны, без необходимости тестирования всех возможных значений внутри диапазона. Это может сократить количество тестов и время, затрачиваемое на тестирование.

1. **Фокус на критических точках:** Ошибки часто возникают на границах диапазонов, так как именно там происходят переходы между различными состояниями программы. Проверка на границах позволяет убедиться, что программа корректно обрабатывает эти переходы.

2. **Простота и ясность:** Проверка только на границах делает тесты более простыми и понятными. Это может облегчить их поддержку и модификацию в будущем.

1. исключая проверку в середине диапазона, можно пропустить ошибки, которые могут возникнуть внутри диапазона. Например, если программа содержит логическую ошибку, которая проявляется только при определенных значениях внутри диапазона, проверка на границах может не выявить эту ошибку.

2. **Сложные зависимости:** в некоторых случаях поведение программы может зависеть от значений внутри диапазона, и проверка только на границах может не охватить все возможные сценарии. Например, если программа использует сложные алгоритмы или математические вычисления, ошибки могут возникать при определенных промежуточных значениях.

3. **Неопределенность требований:** если требования к программе нечеткие или недостаточно определены, проверка только на границах может не обеспечить полное покрытие всех возможных случаев использования.

## **Заключение**

Исключение проверки в середине диапазона в пользу проверок на границах может быть оправданным в случаях, когда границы диапазона являются критическими точками, и ошибки наиболее вероятны именно там. Однако это может привести к пропуску ошибок, которые могут возникнуть внутри диапазона. В идеале, следует использовать комбинацию проверок на границах и в середине диапазона, чтобы обеспечить более полное покрытие тестирования и минимизировать риск пропуска ошибок.

## **Вопрос №10.**

Для воспроизведения бага и сбора логов необходимо сделать несколько шагов:

### **Шаг 1: подготовить устройство**

Убедитесь, что ваше мобильное устройство настроено правильно для сбора логов:

Включить режим разработчика:

- Android: перейти в настройки устройства → Система → О телефоне → Нажать 7 раз на номер сборки, чтобы включить режим разработчика.
- iOS: подключить устройство к компьютеру с macOS через USB-кабель и открыть Xcode. Перейти в раздел Devices and Simulators, выбрать устройство и включить режим разработчика.

### **Шаг 2: воспроизвести баг**

1. Открыть приложение Яндекс Аренда на устройстве.
2. Перейдите в раздел, где возникает проблема («Показать на карте»).
3. Попробуйте воспроизвести шаги, которые приводят к появлению сообщения «Непредвиденная ошибка».
4. Запишите точные шаги, которые привели к возникновению проблемы.

### **Шаг 3: собрать логи**

#### **Для Android:**

1. Подключить устройство к компьютеру через USB-кабель.
2. Открыть Android Studio.
3. Перейти в раздел Logcat и отфильтровать вывод по приложению Яндекс Аренда.
4. Повторить шаги, которые вызывают проблему, пока она снова не возникнет. В логе появятся данные, которые относятся к сбою приложения.
5. Скопировать логи и отправить разработчику.



### Для iOS:

1. Подключить устройство к компьютеру с macOS через USB-кабель.
2. Открыть Xcode.
3. Перейдите в раздел Devices and Simulators и выбрать наше устройство.
4. Открыть вкладку Console и повторить ошибку, чтобы увидеть логи.
5. Скопировать логи и отправить разработчику.

### Шаг 4: приложить логи и описание проблемы

После сбора логов логи и записи шагов для воспроизведения проблемы, отправить их разработчику вместе с описанием проблемы. Указать:

- Версию операционной системы устройства.
- Версию приложения Яндекс Аренда.
- Точные шаги для воспроизведения бага.
- Логи, собранные во время воспроизведения проблемы.

Это поможет разработчикам быстрее диагностировать и исправить проблему.

## Отчёт о тестировании

### Функциональное тестирование веб-приложения

Приложение проверено на стенде ([стенд](#)). Все известные требования были покрыты чек-листом: ([чек-лист](#)).

Результаты выполнения тестов можно посмотреть здесь: ([чек-лист](#)). Из **119** успешно прошло **97**, не прошло — **20** и **2** невозможно проверить.

Список багов, найденных при тестировании, разбит по приоритетам:

#### 1. Блокирующие:

[https://docs.google.com/spreadsheets/d/1IqYyeAFPfa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A13:K13](https://docs.google.com/spreadsheets/d/1IqYyeAFPfa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A13:K13)

[https://docs.google.com/spreadsheets/d/1IqYyeAFPfa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A21:K21](https://docs.google.com/spreadsheets/d/1IqYyeAFPfa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A21:K21)

#### 2. Критичные:

[https://docs.google.com/spreadsheets/d/1IqYyeAFPfa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A3:K3](https://docs.google.com/spreadsheets/d/1IqYyeAFPfa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A3:K3)

[https://docs.google.com/spreadsheets/d/1IqYyeAFPfa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A5:K5](https://docs.google.com/spreadsheets/d/1IqYyeAFPfa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A5:K5)

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A6:K6](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A6:K6)

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A7:K7](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A7:K7)

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A8:K8](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A8:K8)

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A9:K9](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A9:K9)

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A10:K10](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A10:K10)

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A12:K12](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A12:K12)

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A15:K15](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A15:K15)

### 3. Средний приоритет:

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A2:K2](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A2:K2)

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A4:K4](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A4:K4)

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A11:K11](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A11:K11)

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A14:K14](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A14:K14)

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A16:K16](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A16:K16)

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A17:K17](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A17:K17)

### 4. Низкий приоритет:

[https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A18:K18](https://docs.google.com/spreadsheets/d/1lgYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A18:K18)

[https://docs.google.com/spreadsheets/d/1IqYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A19:K19](https://docs.google.com/spreadsheets/d/1IqYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A19:K19)

[https://docs.google.com/spreadsheets/d/1IqYyeAFPFa8kuMts-EZdhwAYmkY\\_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A20:K20](https://docs.google.com/spreadsheets/d/1IqYyeAFPFa8kuMts-EZdhwAYmkY_58LEaxYCQBnypt8/edit?gid=1186534874#gid=1186534874&range=A20:K20)

Заключение:

Какой баг оказался самым критичным? -

Блокирующий баг, который возникает в браузере Google Chrome. Баг не дает клиенту оформить заказ.

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

В требованиях указано, что можно вводить символы. Например, точки, запятые, тире. Но не указано можно ли вводить только символы.

2. Проверенная тобой функциональность готова к релизу? Почему?

Анализ текущего состояния позволяет сделать вывод о том, что веб-приложение не готово к релизу. Из-за выявленного бага пользователи не могут оформить заказ в браузере Google Chrome. Учитывая, что около четверти пользователей предпочитают использовать именно этот браузер, существует высокий риск потери аудитории, что негативно отразится на репутации и финансовом положении компании.

## Ретест багов в мобильном приложении

Был проверен фикс багов. Из них не исправлено **1**, исправлено — **3**. Также при проверке был обнаружен **1** новый баг.

Список багов можно посмотреть здесь:

[https://docs.google.com/spreadsheets/d/1S3OSATPPzEXV\\_B6TsMRYadX5bVoaZe-QjwCFOF5Vfql/edit?gid=1490883462#gid=1490883462&range=A3:J3](https://docs.google.com/spreadsheets/d/1S3OSATPPzEXV_B6TsMRYadX5bVoaZe-QjwCFOF5Vfql/edit?gid=1490883462#gid=1490883462&range=A3:J3)

[https://docs.google.com/spreadsheets/d/1S3OSATPPzEXV\\_B6TsMRYadX5bVoaZe-QjwCFOF5Vfql/edit?gid=1490883462#gid=1490883462&range=A4:J4](https://docs.google.com/spreadsheets/d/1S3OSATPPzEXV_B6TsMRYadX5bVoaZe-QjwCFOF5Vfql/edit?gid=1490883462#gid=1490883462&range=A4:J4)

[https://docs.google.com/spreadsheets/d/1S3OSATPPzEXV\\_B6TsMRYadX5bVoaZe-QjwCFOF5Vfql/edit?gid=1490883462#gid=1490883462&range=A5:J5](https://docs.google.com/spreadsheets/d/1S3OSATPPzEXV_B6TsMRYadX5bVoaZe-QjwCFOF5Vfql/edit?gid=1490883462#gid=1490883462&range=A5:J5)

[https://docs.google.com/spreadsheets/d/1S3OSATPPzEXV\\_B6TsMRYadX5bVoaZe-QjwCFOF5Vfql/edit?gid=1490883462#gid=1490883462&range=A6:J6](https://docs.google.com/spreadsheets/d/1S3OSATPPzEXV_B6TsMRYadX5bVoaZe-QjwCFOF5Vfql/edit?gid=1490883462#gid=1490883462&range=A6:J6)

[https://docs.google.com/spreadsheets/d/1S3OSATPPzEXV\\_B6TsMRYadX5bVoaZe-QjwCFOF5Vfql/edit?gid=1490883462#gid=1490883462&range=A7:J7](https://docs.google.com/spreadsheets/d/1S3OSATPPzEXV_B6TsMRYadX5bVoaZe-QjwCFOF5Vfql/edit?gid=1490883462#gid=1490883462&range=A7:J7).

## Регрессионное тестирование мобильного приложения по готовым тест-кейсам

Результаты выполнения регрессионных тестов можно посмотреть здесь: ([Тест-ран](#)).

Из 10 успешно прошло 1, не прошло — 9.

Список багов, найденных при тестировании, разбит по приоритетам:

1. Блокирующие:

[https://docs.google.com/spreadsheets/d/11Vt7xbW\\_Sy\\_Wr4H1dSCyaX0KjdPK90\\_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A4:J4](https://docs.google.com/spreadsheets/d/11Vt7xbW_Sy_Wr4H1dSCyaX0KjdPK90_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A4:J4)

2. Критичные:

[https://docs.google.com/spreadsheets/d/11Vt7xbW\\_Sy\\_Wr4H1dSCyaX0KjdPK90\\_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A2:J2](https://docs.google.com/spreadsheets/d/11Vt7xbW_Sy_Wr4H1dSCyaX0KjdPK90_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A2:J2)

[https://docs.google.com/spreadsheets/d/11Vt7xbW\\_Sy\\_Wr4H1dSCyaX0KjdPK90\\_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A3:J3](https://docs.google.com/spreadsheets/d/11Vt7xbW_Sy_Wr4H1dSCyaX0KjdPK90_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A3:J3)

[https://docs.google.com/spreadsheets/d/11Vt7xbW\\_Sy\\_Wr4H1dSCyaX0KjdPK90\\_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A5:J5](https://docs.google.com/spreadsheets/d/11Vt7xbW_Sy_Wr4H1dSCyaX0KjdPK90_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A5:J5)

[https://docs.google.com/spreadsheets/d/11Vt7xbW\\_Sy\\_Wr4H1dSCyaX0KjdPK90\\_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A10:J10](https://docs.google.com/spreadsheets/d/11Vt7xbW_Sy_Wr4H1dSCyaX0KjdPK90_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A10:J10)

[https://docs.google.com/spreadsheets/d/11Vt7xbW\\_Sy\\_Wr4H1dSCyaX0KjdPK90\\_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A11:J11](https://docs.google.com/spreadsheets/d/11Vt7xbW_Sy_Wr4H1dSCyaX0KjdPK90_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A11:J11)

3. Средний приоритет:

[https://docs.google.com/spreadsheets/d/11Vt7xbW\\_Sy\\_Wr4H1dSCyaX0KjdPK90\\_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A6:J6](https://docs.google.com/spreadsheets/d/11Vt7xbW_Sy_Wr4H1dSCyaX0KjdPK90_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A6:J6)

[https://docs.google.com/spreadsheets/d/11Vt7xbW\\_Sy\\_Wr4H1dSCyaX0KjdPK90\\_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A8:J8](https://docs.google.com/spreadsheets/d/11Vt7xbW_Sy_Wr4H1dSCyaX0KjdPK90_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A8:J8)

[https://docs.google.com/spreadsheets/d/11Vt7xbW\\_Sy\\_Wr4H1dSCyaX0KjdPK90\\_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A9:J9](https://docs.google.com/spreadsheets/d/11Vt7xbW_Sy_Wr4H1dSCyaX0KjdPK90_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A9:J9)

4. Низкий приоритет:

[https://docs.google.com/spreadsheets/d/11Vt7xbW\\_Sy\\_Wr4H1dSCyaX0KjdPK90\\_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A7:J7](https://docs.google.com/spreadsheets/d/11Vt7xbW_Sy_Wr4H1dSCyaX0KjdPK90_BtBZBavE4bj0/edit?gid=452637660#gid=452637660&range=A7:J7)

Закключение:

1. Какой баг показался самым критичным?

На мой взгляд, основные проблемы связаны с ошибкой на сервере при попытке принять отменённый заказ и с отсутствием уведомления об ошибке, если вы пытаетесь принять заказ, который уже был доставлен другим курьером.

2. Такой продукт можно выпускать в релиз? Почему?

Такой продукт нельзя выпускать в релиз, так как только 1 из 10 тест-кейсов был успешно пройден. Остальные тесты с багами, которые являются критическими и приводят к сбою приложения.

## Выводы о проделанной работе

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

Первая практическая часть прошла достаточно активно, старался покрыть проверками максимальную функциональность веб-приложения. Понимаю, что проверок можно сделать больше. При ретесте и регрессе помимо проверок по тест-кейсам, находил дополнительные Баги и заводил баг-репорты.

Пришлось уделить время чтобы вспомнить, как тестируют веб-приложения на эмуляторе Android Studio. С самим Android Studio тоже возникли проблемы и пришлось подключаться к удаленному рабочему столу, чтобы осуществить проверки. Были вопросы к тест-кейсам для регресса, часть проверок блокировалась из-за багов из предыдущих заданий. Много времени потратил чтобы понять что делать, но в итоге разобрался.