Видео-ответы

Вопрос 7. В чем разница между SOAP и REST?
REST - это архитектурный стиль построения распределенных приложений, ориентированный на использование HTTP в качестве транспортного протокола.
В отличие от SOAP, каждый запрос в REST имеет свой URL, то есть запросы клиента обслуживаются сервером на разных ресурсах (адресах), причем URL этих адресов обычно содержат указания на тип принимаемых запросов, а также идентификаторы обрабатываемых объектов. А для SOAP сервер обрабатывает запросы на одном и том же сетевом адресе (или на ряде заранее известных ресурсов, адреса которых не зависят от типа обрабатываемого запроса).
В REST нет ограничений на формат данных. Чаще всего это JSON, но может быть и XML, и любой другой. Но JSON является самым популярным форматом в REST, т.к. его легче обрабатывать и анализировать, а программисты имеют простые инструменты для работы с JSON во всех программных платформах на всех современных языках программирования.

SOAP - это формат (протокол) обмена данными, который строго специфицирован. В SOAP имеется стандартизованное описание сообщений прикладного протокола, где указываются все возможные названия операций запроса информации от сервера и ответов от сервера клиенту - WSDL, Web Service Document Layer. Кроме того, в протоколе SOAP есть стандартизованное описание схемы документа, то есть указание полей запроса и ответа с их типизацией - данная информация в SOAP фиксируется в документе XSD - XML Schema Definition и этот документ позволяет программистам строить программные структуры для работы с описанным в XSD прикладным протоколом. Для архитектурного стиля REST не существует стандартизованных форматов аналогичных XSD и WSDL, хотя есть аналогичные им общепринятые подходы - YAML и Open API.
SOAP может использовать в качестве транспортного протокола не только HTTP, это может быть MQ (с помощью брокеров Message Queue) и даже передача посредством файлов.

Вопрос 8. Можно ли отправить SOAP-запрос с помощью Postman?

Postman - это инструмент для тестирования API. Хотя изначально он создавался для работы с REST API, в нем можно оправить о SOAP запрос.
Для этого нужно:

- создать новый запрос
- выбрать тип запроса POST
- указать адрес запроса во вкладке URL
- в теле запроса, во вкладке “body” выбрать формат данных “row”
- в поле для ввода данных выбрать формат XML
- Postman не позволяет автоматически формировать структуру SOAP запроса, поэтому запрос надо писать вручную (или вставить запрос, заранее подготовленный и соответствующий формату протокола из другого инструмента)
- отправить запрос

- получить ответ

Письменные ответы на вопросы

Вопрос 1

Для тестирования функциональности по поиску квартир для покупки по определенным параметрам на сайте застройщика я использую таблицу принятия решений. Составляя тест-кейсы обычным способом можно упустить какие-либо комбинации.
Нам предстоит протестировать поиск по набору параметров, интересующих потенциального покупателя. Комбинации различных условий удобно представить в виде таблицы. Каждая строка таблицы - готовый тест-кейс.
Этот вид тест-дизайна удобно использовать при небольшом количестве параметров для тестирования.
Таблица принятия решений

Вопрос 2

ID теста: отсутствует
Мой вариант: “ GH-12”, например, лучше использовать английские буквы
Заголовок: “Отображение главной странице Яндекс Практикума для неавторизованного пользователя” сформулирован неточно, относительно ожидаемого результата, который подразумевает открытие главной страницы для неавторизованного пользователя.
Мой вариант: “Открывается главная страница Яндекс Практикума для неавторизованного пользователя при переходе по ссылке Яндекс Практикум”

Предусловие: “Пользователь не авторизован в системе Яндекс Практикума”.
Нужно уточнить, так как Яндекс Практикум имеет приложения, как десктопное, так и для мобильных устройств. Можно указать наличие/отсутствие приложения Яндекс Практикума и наличие/отсутствие Яндекс Почты.

Шаги воспроизведения: “Открой главную страницу Яндекс Практикума https://practicum.yandex.ru/”
Заголовок “Шаги воспроизведения” лучше переименовать в **“Шаги”**
Мой вариант: “Кликнуть на ссылку https://practicum.yandex.ru/” или “Открыть приложение Яндекс Практикум на главном экране устройства”

Ожидаемый результат: “Открылась главная страница для неавторизованного пользователя”. Не указано, какая главная страница и где она открылась.
Мой вариант: “Открылась главная страница Яндекс Практикума (ссылка на страницу)”

Окружение: не указано, какая операционная система и ее версия; не указан браузер, в котором открыт сайт; не указано разрешение экрана и тип устройства, на котором происходит тестирование.

Вопрос 3

- Открыть форму создания заказа.
- Запустить DevTools способом, подходящим для используемого браузера.
- Открыть вкладку Network.
- Обновить страницу.
- Вновь отметить чек-бокс “Заплатить сразу”
- Во вкладке Network просмотреть запросы фронтенда и ответы бэкенда.
- Кликнуть на строку запроса и просмотреть детальную информацию о запросе и ответе:
    1. Во вкладке General содержится основная информация о запросе: адрес, по которому отправился запрос, метод запроса (DEL,POST,GET) код состояния и текст, например, с сообщением об ошибке и ее коде
    2. Во вкладке Query String Parameters содержатся параметры запроса в виде ключ=значение
    3. Во вкладке Preview содержится ответ на запрос
- Сделать скриншоты и приложить их к баг-репорту

Вопрос 4

Ситуация с удалением всех данных в БД хоста очень маловероятна, но возможна.

Обоснование: GET-запрос по протоколу HTTP используется для получения данных с сервера, но прикладные протоколы поверх HTTP могут быть устроены так, что с помощью GET-запросов удаляются те или иные объекты в БД хоста, обычно по заданным идентификаторам или фильтрам, указанным в запросе.

Дополнение:
При использовании подобных протоколов, если идентификаторы/фильтры для удаления объектов в БД не заданы, программное обеспечение на хосте, отрабатывая такой GET-запрос, может повести себя так, что удалит все объекты определенного типа.

Уточнение:
Прикладные API устроены так, что работают с конкретными объектами определенных типов, а не со всей информацией в БД.
Но если сервер приложения взаимодействует с БД под суперпользователем (владельцем всех данных приложения в схеме БД), а программный код в обработчике какого-нибудь GET-запроса содержит намеренные инструкции для удаления всех

данных в схеме этого пользователя, то удаление всех данных приложения в БД возможно.

Вопрос 5

Таблица сотрудников employee с полями:

- id - идентификатор сотрудника в таблице employee - стоит сделать первичным ключом, т.к. это идентификатор записи в этой таблице, то есть, заведомо уникальное значение
- fio - ФИО сотрудника - обычные строковые данные по сотруднику в этой таблице
- position_id - идентификатор должности - внешний ключ для таблицы должностей position

Таблица должностей position с полями:

- id - идентификатор должности - первичный ключ, как уникальный ид.
- name - название должности - обычные строковые данные по сотруднику в этой таблице
- salary - зарплата на данной должности - обычные числовые данные по сотруднику в этой таблице

Каждая строка (запись) в таблице должна быть уникальной. Эту уникальность обеспечивает первичный ключ (Primary key), который представляет собой набор данных, уникальных для каждой строки. А внешний ключ (Foreign key) обеспечивает связи между разными таблицами в одной базе данных.

Вопрос 6

SELECT employee.fio, position.name, position.salary FROM employee JOIN position ON employee.position_id = position.id

Вопрос 7

Во 2-ой строке функции `test_integer_division()` не хватает служебного слова assert, эта строка на самом деле должны выглядеть так:

```
    assert a == 2
```

Именно оператор assert проверяет, что следующее за ним логическое выражение истинно, а если это вдруг не так, исполнение функции в этом месте прерывается с генерацией об ошибке с текстом о том, что ожидалась истинность проверки “a == 2”, но по факту эта проверка оказалась ложной.

Вопрос 8

Для того, чтобы определить граничные значения, нужно сначала выделить классы эквивалентности. Классы эквивалентности - это набор определенных данных, при котором тестируемое приложение работает одинаково, а граничные значения - это место, где один класс эквивалентности переходит в другой. Чаще всего, друг без друга они не применяются.

Но есть исключения, например, дата и время. Чаще всего они указываются в определенном формате, например, ДД.ММ.ГГГГ или ДД.ММ.ГГ
Таким образом, класс эквивалентности для формата даты, представленного, к примеру, двумя числами, будет заключаться в количестве этих цифр и не учитывать граничные значения самой даты. Проверка будет идти на наличие или отсутствие двух цифр в формате. То же самое верно для формата отображения времени.
Кроме того, есть есть тип классов эквивалентности, в котором отсутствуют граничные значения - набор. В наборе нельзя выделить начало или конец класса эквивалентности. К нему можно отнести, например, набор языков в приложении. Здесь мы просто не сможем выделить граничные значения и проверять придется все.

Вопрос 9

Существует мнение, что ошибки чаще всего случаются именно на границах, и проверять эти границы нужно обязательно. Таким образом, поведение на границе класса эквивалентности может отличаться от поведения внутри класса. Для проверки правильности поведения внутри класса эквивалентности нужно провести дополнительную проверку по середине (один дополнительный тест)
Например, тестируем возраст получателя кредита, который по условиям находится в периоде от 18 до 50 лет включительно. Логично будет проверить данные на нижней границе диапазона (17, 18 и 19 лет) и данные на верхней границе диапазона (49, 50 и 51 год). И дополнительно проверить данные в середине №30 лет).
Если диапазон состоит из сложных условий (или функций), то тогда вероятность ошибки внутри самого диапазона резко увеличивается, и отказываться от проверок в этом случае нельзя.

Вопрос 10

Тестирование мобильного приложения можно провести с помощью:

- Android Studio (эмулятор для Windows)
- Физического мобильного устройства с подключением его к Android Studio
- Apple iOS Simulator для Mac

Рассмотрим воспроизведение багов и снятие логов для разработчика на примере использования Android Debug Bridge (ADB, Android Studio).

- Запустить Android Studio
- Создать мобильное устройство для тестирования на андроид-платформе
- установить на него мобильное приложение “Яндекс Аренда”
- Очистить логи в приложении
- Открыть консоль Logcat
- Выбрать нужный эмулятор из списка
- В строке поиска ввести название приложения (Яндекс Аренда) или можно ввести level: ERROR
- Открыть приложение Яндекс Аренда
- Выбрать квартиру в приложении
- Нажать “Смотреть на карте”
- Появится сообщение “Непредвиденная ошибка” и приложение

- Вернуться в Android Studio и в консоли Logcat остановить запись логов
- Скопировать текст в консоли Logcat, сохранить в текстовый файл и приложить к баг-репорту
- Или через консоль Terminal:
  adb logcat - для вывода логов устройства
  adb logcat > c:\adb\log.txt - запись логов в отдельный файл
  ctrl+c - остановить выведение логов
- Выбрать из файла логи с ошибкой и прикрепить к баг-репорту

Отчёт о тестировании

Функциональное тестирование веб-приложения

Приложение проверено на стенде :

Тестовый стенд

Все известные требования были покрыты чек-листом:

Чек-лист.

Результаты выполнения тестов можно посмотреть здесь:

Чек-лист

Из 206 тестов успешно прошло 174 тестов,

не прошло — 32 тестов.

Список багов, найденных при тестировании, разбит по приоритетам:

1. Блокирующие:
- не найдено

2. Критичные:
Не оформляется заказ

Баг с длиной поля "Адрес"

Не подсвечивается пустое поле "Адрес"

Открывается форма при незаполненном обязательном поле

Некорректный выбор даты доставки

Не активна кнопка "Заказать" в шапке лендинга

3. Средний приоритет:
Ошибка с длиной поля "Фамилия"

Баг с корректностью в поле "Комментарий"

Баг с длиной поля "Комментарий"

4. Низкий приоритет:
Неверное название поля

Баг по цвету поля

Заключение:

Самым критичным оказался баг с оформлением заказа. На финальном этапе подтверждения оформления заказа кнопка “Да” нажимается, а оформления заказа не происходит. Также не появляется окно с номером оформленного заказа. Приложение, по сути, не работает.

Самая хитрая серая зона - выбор цвета самоката. В требованиях указано, что можно отметить сразу два чек-бокса, а если пользователь заказал один самокат, как быть? Здесь нужно доработать выбор цвета в зависимости от количества заказанных самокатов.

На данный момент функциональность “Оформить заказ” к релизу не готова по двум главным причинам:

1. Есть серьезный баг с полем “Адрес: куда привезти самокат”. Это поле обязательное, если оно не заполнено, то далее процесс оформления заказа идти не может. На деле при пустом поле “Адрес” открывается следующая форма для оформления аренды “Про аренду”.
2. Сам процесс оформления аренды самоката на последнем этапе прерывается и заказа самоката не происходит, то есть приложение, по сути, не выполняет свою основную функцию.

Ретест багов в мобильном приложении

Ретест багов в мобильном приложении
Был проверен фикс багов - 4 багов.
Из них не исправлено - 1 баг
исправлено — 3 бага

Список багов можно посмотреть здесь:
При клике на нотификацию приложение ломается
Баги дублировались, поэтому была проведена оптимизация: дубли были удалены.

Регрессионное тестирование мобильного приложения по готовым тест-кейсам

Результаты выполнения регрессионных тестов можно посмотреть здесь:

Ретест багов в мобильном приложении по готовым тест-кейсам

Из 10 тестов успешно прошло 2 теста, не прошло — 8 тестов.

Список багов, найденных при тестировании, разбит по приоритетам:

1. Блокирующие:
не найдено

2. Критичные:
Дублирование заказов

Вылетает приложение при принятии удаленого заказа

Приложение вылетает, если попытаться взять уже принятый кем-то заказ

Дублирование заказа при принятии полной карточки заказа

3. Средний приоритет:
Ошибка анимации

Синяя точка

Пустое поле "Комментарий

4. Низкий приоритет:
Пустое поле "Цвет"

Заключение:

Самыми критичными мне показались баги, связанные с ошибкой при принятии заказа:

- при принятии удаленного ранее заказа
- при принятии уже кем-то принятого заказа

По требованиям, должно появляться сообщение об ошибке «Ты не можешь принять заказ. Его взял другой курьер либо пользователь отменил доставку». Оно не просто не появляется, но приложение ломается и вылетает. Это серьезно ограничивает его применение и удобство использования для курьеров.

Тот факт, что удаленный или принятый уже кем-то заказ какое-то время остается в общей ленте заказов, это недоработка разработчиков. И это недостаток надо устранить перед выпуском приложения в релиз.

Сейчас приложение к релизу не готово. Основная функциональность “Принятие заказа курьером” полноценно не функционирует.

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?

В начале была некоторая растерянность. Дипломный проект заключает в себе все то, что мы проходили на курсе, и потребовалось время на то, чтобы собраться с мыслями. Но с другой стороны, это дало возможность еще раз повторить основные моменты теории и закрепить их на практике.
Теперь знаю, на что мне нужно обратить больше внимания, и куда двигаться дальше в плане дальнейшего развития и обучения.

Некоторые затруднения вызвала теория по тестированию API. Учту это на будущее.