Я работал в компании "Название компании" как сотрудник подряда на проект "Название твоего проекта", на проекте было WEB и Mobile приложения

Тестированием больше занимался мобилки т.к. на тот момент было много работы по тестированию релизов именно мобильного приложения, но и основным веб-сайтом занимался тоже

Сначала больше работал над фронт-ендом чем над бэк -ендом, пока погружался в проект и его техническую документацию, далее баланс нагрузки сместился больше в сторону бэк-енда так как там было больше работы для тестирования.  
  
Мы работали по гибкой методологии СКРАМ с двухнедельными спринтами, и каждый день у нас были дейлики (ежедневные созвоны) для обсуждения хода работы.  
Команда тестирования состояла из 4 человек (вместе с тимлидом).   
Команда разработки состояла из 2-х фронт разработчиков и 4-х бэк разработчиков. Так же было 3 аналитика(бизнес-аналитики и системные аналитики) и руководитель проекта (менеджер).  
Для управления процессом тестирования и отслеживания багов, использовали **ДЖИРУ**. Для ведения документации по проекту использовался **КОНФЛЮЕНС.(Требования по которым писались тест-кейсы находились в конфлюенсе, составляли требования аналитики, тестировщики в конфлюенсе составляли отчеты по итогам тестирования)**   
Перед спринтом проводилось планирование, где мы распределяли задачи на будущий спринт в зависимости от задач команды по предстоящим релизам. Это могут быть задачи такие как составление тестовых моделей для последующего тестирования(написание тест-кейсов), актуализация существующих тестовых моделей, и разумеется задачи на само тестирование какого-то функционала предстоящего релиза (Мобильного приложения или сайта)  
Перед началом процесса тестирования релиза мы писали тестовую документацию на основании требований(Тест-кейсы, чек-листы), после чего выполняли ручное функциональное тестирование. Сборками тестовых стендов у нас занимался в основном **ТИМЛИД  
  
Тест-кейсы создавались и хранились ... (название платформы, которую изучали)**  
  
Изначально взяли меня в команду еще совсем неопытным, так что сложность работ выполняемых мной повышалась постепенно, но где-то спустя год работы я уже выполнял полный объем функций моей позиции.

По основному сайту и мобильному приложению по началу работы у меня большинство задач было связано с фронтом. Я проверял соответствие интерфейса с макетами в **ФИГМЕ** и проводил кроссбраузерное, кроссплатформенное тестирование(Android, iOS) с помощью **БРАУЗЕРСТЭКА** и **ДЕВТУЛЗ(это только по Веб-сайту)** (вкладка **ЭЛЕМЕНТС**).   
Browserstack был неоценимым инструментом для кросс-браузерного и кросс-платформенного тестирования, позволяя эффективно проверять функциональность и отображение нашего веб-приложения в различных браузерах и на различных устройствах.  
С мобильным приложением я работал в том числе используя Anroid studio(Для андроида) и xCode(дляiOS) - для снятия логов и создания эмуляций(эмуляции в Anroid studio)

Также проверял взаимодействие фронта с бэком. В работе часто пользовался инструментами разработчика, например, во вкладке **НЕТВОРК** (Сеть) смотрел отработку тех или иных запросов.   
При проверке взаимодействия нашего веб-приложения с API, я использовал **ПОСТМАН**, а для понимания структуры API и его спецификаций - СВАГГЕР.  
  
Так же я наигрывал(Воспроизводил) фронт/бэк ошибки, изучал причины возникновения дефектов. Ошибки воспроизводил в том числе для проверки отказа и восстановления процессов, например ставил таймаут на сервис обработки платежа, смотрел как себя поведет система, например послей отмены таймаута сервис должен снова попытаться обработать платеж по заказу и поменять статус заказа на оплачен/не оплачен в зависимости от результата самой оплаты. Нельзя чтобы оплата по заказу пропала и обработка заказа просто зависла в воздухе, это негативный опыт для клиента и скорее всего дополнительная работа для службы поддержки. В реальном случае, если такое случится для клиента это конечно должно все проходить без ошибок и проблем.  
  
  
  
Так как мы работали над проектом с розничными продажами, важной частью нашей работы было интеграционное тестирование. Некоторые из ключевых интеграций включали:

- Интеграции с платежными системами, очень важно чтобы путь клиента от его корзины покупок до успешной покупки проходил без каких-либо проблем и ошибок

- Интеграция с системами управления заказами, автоматизация распределения, контроля и отслеживания статусов заказов в том числе находящихся в процессах доставки должны так же проходить без проблем и ошибок (Так же очень важно, чтобы заказы клиентов уезжали в правильные филлиалы для самовывоза/передачи в доставку)

- Интеграция с CRM-системой для улучшения взаимодействия с клиентами и персонализации предложений.  
  
  
У проекта очень плотная интеграция основного веб-сайта с мобильным приложением, очень важно чтобы присутствовала синхронизация личного кабинета(история покупок, настройки, введенные промокоды), корзины покупок и т.д. (Например пользователь ввел в мобильном приложении промокод на скидку - открыл сайт с компьютера, а скидки нет, должна быть полная синхронизация)  
  
Также работал с базами данных на уровне: просмотреть зарегистрированных пользователей и проверить подключенные услуги для них/приобретенные гарантии/статусы его заказов/доставок. На проекте была база данных **ПОСТГРЕС**, в качестве администратора данных использовался **dBeaver**.  
Активно использовал **КИБАНУ** для мониторинга и анализа логов, что помогало быстро выявлять и отслеживать ошибки.  
Для снятия мобильных логов еще дополнительно использовался - Android studio, xCdoe  
Android studio - андроид, в том числе создание эмуляций  
xCode - iOS  
  
Все дефекты, которые были найдены и заведены в ЖИРЕ (Jira) отслеживались и обязательно по исправлению мы их тестировали.

В случае возникновения каких-либо вопросов по техническим требованиям обращался за уточнениями к аналитику, по вопросам реализации к разработчикам. По всем релизам ВСЕГДА у нас проводилось регрессионное тестирование, тест-кейсы для которого выбирал наш **ТИМЛИД** (руководитель отдела тестирования). Весь регресс выполнялся вручную всей командой тестирования путем прохождения **ТЕСТОВОГО** **ПРОГОНА** (тест-рана). После регресса релиз проходил приемочное тестирование для команды сопровождения релизов и далее релиз уходил на основной сайт/мобильное приложение т.е. в прод. По функционалу, наверное, все. Если у вас возникли какие-то вопросы, то я готов на них ответить.   
  
По внутренней кухне:

Каждый раз когда к вам приходил новый патч или релиз для тестирования вы проводили тестирование нового функционала (что-то новое, что добавилось) и вдобавок к этому ВСЕГДА проводили регресс (проверить все остальное, после добавления нового, чтобы убедиться, что это новое ничего не сломало), регресс зачастую затрагивает только подсистему в которой были изменения. Допустим была добавлена новая кнопка в личный кабинет - вы протестировали кнопку( новый функционал), а затем вы тестируете весь личный кабинет, чтобы убедиться, что эта кнопка ничего не сломала(регресс).

Соответственно под каждый новый функционал ты писал новые тест-кейсы, в зависимости от критичности проверяемого функционала - те кейсы которые ты писал для нового функционала могут войти в регрессионую модель и войти в число регрессионых тестов которые ты будешь проводить на следующем релизе/патче

Тест-кейсы по каждому релизу/патчу ты писал и на фронт и на бэк

Фронт условно, что кнопка в личный кабинет добавилась, она есть и нажимается, она правильного цвета и т.д.

Бэк то что происходит когда ты нажимаешь кнопку, корректный ли уходит запрос, куда уходит и т.д.

**Как ты пришел в тестирование?**

Всегда хотел работать онлайн и не быть привязанным к одному месту. Смотря на то как развивается сфера IT очень хотел стать частью этого течения и решил себя попробовать в тестировании. Я начал изучать материалы из интернета, смотреть ролики на YouTube и вскоре смог найти работу (можно добавить что-нибудь от себя)  
Изучал книгу Романа Савина по тестированию, десятки изученных статей на хабре, бесконечные видео-уроки и прочий познавательный контент от толковых тестеров и конечно кладезь всея неформальной информации - форумы тестировщиков, там я много увидел какие проблемные кейсы встречаются у различных тестировщиков различных сегментов разработки (от геймдева до банковских разработок) и сответственно смотрел какие пути решения предлагали другие, увлекательно и полезно для меня было тогда.

Так же пользовалься многочисленными тренажерами для тестировщиков. Поломанные сайты или дэшборды где нужно было искать баги и понимать их природу.

Сертификатов никаких пока нет, попробовал вначале скиллбокс до моего первого опыта работы, фигня оказалась, намного больше смог выучить и узнать занимаясь сам и изучая экспертов по тестированию, очень быстро смог прокачаться на энтузиазме, очень понравилась сфера тестирования, чувствую, что делаю важную работу)  
  
Несоизмеримо больше конечно узнал о процессах тестирования уже в процессе самой работы, к изучению теории наконец добавилась постоянная практика и я чувствую свой рост как специалиста в этой сфере.

**Почему уволился?**

Отсутствовало личное развитие на проекте, как специалиста, хотелось развиваться дальше. Хотел бы попробовать свои силы в новой компании на новом интересном проекте и поднять свой зарплатный уровень.

**Как вы видите свое дальнейшее развитие?**

На данный момент хочу получить опыт на новом проекте, но в дальнейшем рассматриваю возможность перехода в автотестирование. Было бы прекрасно, если бы новая компания могла помочь с обучением, но это не будет являться основополагающим фактором в конечном выборе компании. (Если спросят: Писать автотесты хотел бы на Пайтоне, потому что сам язык программирования считаю перспективным в будущем и более понятным)

**ВОПРОСЫ ДЛЯ HR:**

1. Есть ли в вашей компании возможность полной удаленной работы?

2. Является ли ваша компания аутсорсинговой?

3. На какой проект требуется сотрудник? (если HR знает о проекте)

4. Сколько человек в команде тестирования? (если HR знает о проекте)

5. Во сколько этапов у вас проходит собеседование?