МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА ПО ДИСЦИПЛИНЕ

«ПРОЕКТНАЯ ДЕЯТЕЛЬНОСТЬ»

Лаборатория Тестирования

Куратор проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Яковлев Станислав Игоревич, Преподаватель/

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Фурман Кирилл Владимирович, 181-331/ подпись ФИО, группа

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Киличева Дилфуза Голомовна , 194-321/ подпись ФИО, группа

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Великанов Иван Вадимович, 181-331/

подпись ФИО, группа

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Махмадзиеев Али Олимович, 181-331/

подпись ФИО, группа

Москва, 2021

**Оглавление**

[**ВВЕДЕНИЕ** 3](#_Toc75908490)

[**ПЛАН РАБОТЫ** 5](#_Toc75908491)

[**Участники и их роли.** 6](#_Toc75908492)

[**1** **ТЕСТИРОВАНИЕ API** 6](#_Toc75908493)

[**1.1** **Программное обеспечение для автоматизации развёртывания и управления приложениями** 6](#_Toc75908494)

[**1.2** **Контейнеры и что это такое** 8](#_Toc75908495)

[**1.3** **Основные команды docker** 8](#_Toc75908496)

[**2** **БАЗЫ ДАННЫХ** 9](#_Toc75908497)

[**2.1** **Системы управления базами данных (MongoDB, PostgreSQL, MySQL)** 11](#_Toc75908498)

[**2.2** **GUI (Robo 3T, pgAdmin, PHPMyAdmin)** 14](#_Toc75908499)

[**3** **ОТЧЕТ ПО ТЕСТИРОВАНИЮ ПРОЕКТА** 15](#_Toc75908500)

[**РЕЗУЛЬТАТЫ** 1](#_Toc75908501)

[**ЗАКЛЮЧЕНИЕ** 1](#_Toc75908502)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 1](#_Toc75908503)

# **ВВЕДЕНИЕ**

Проект по тестированию мобильного приложения «Юла» от разработчиков «mail.ru» представляет собой поиск ошибок в работе этого приложения, которые могут помешать пользователю комфортно использовать данный программный продукт.

Данный курс предназначен для ознакомления с таким понятием как тестирование.

Цель курса: ознакомиться с базовыми понятиями и методиками, применяемыми в тестировании, изучить тестируемое приложение, написать чек-листы, тест-кейсы и рассмотреть выявленные ошибки.

Для учета обнаруженных ошибок будут использоваться такие программные средства как:

TestReil – инструмент для управления основными процессами тестирования, начиная от регистрации и заканчивая сроками решения и выполнения работы.

Google Таблицы – для ведения учёта основных аспектов работы.

Trello – программа для менеджмента и учёта сроков выполнения основных работ.

Данное исследование очень актуально, так как тестирование приложения напрямую влияет на выявление ошибок и критических сбоев в работе приложения, а также помогает обезопасить приложение, сохранить деньги компании и представить пользователю хороший качественный продукт.

Аннотация

В данной работе представлены методологии и методы исследования предметной области тестирования мобильных приложений. Также представлены тест-кейсы и чек-листы протестированного мобильного приложения vk mini app, раздел объявления, с представлением найденных дефектов.

# **ПЛАН РАБОТЫ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Тема занятия** | **Кол-во теоретических часов** | **Кол-во практических**  **часов** | **Кол-во часов самостоятельной работы** |
| Тестирование |  |  |  |
| Введение в задание | 3 | - | - |
| Знакомство с Jira, Redmine, Trello. | 2 | 2 | - |
| Тестирование в жизненном цикле разработки ПО. | 2 | 2 | - |
| Тестовая документация.  Test Rail. | 2 | 2 | - |
| Составление чек-листов и тест-кейсов. | 2 | 6 | 2 |
| Правила составления отчетов об ошибках. | 2 | 2 | - |
| Техники тест-дизайна. | 2 | 2 | - |
| Тестирование API |  |  |  |
| Изучение методов работы с API | 2 | - | - |
| Клиент-серверная архитектура  Протокол HTTP | 2 | 2 | - |
| Работа с API при помощи «Charles proxy» | 2 | 2 | - |
| Тестирование API с помощью Postman | 2 | 4 | - |
| Тестирование WEB-приложений |  |  |  |
| Основы HTML, CSS, JS для тестировщиков | 2 | 2 | - |
| Особенности тестирования WEB-приложений | 2 | 2 | 2 |
| DEV-Tools Google Chrome | 1 | 1 | 1 |
| Тестирование мобильных приложений |  |  |  |
| Особенности тестирования мобильных приложений. | 2 | 2 | - |
| Тестирование Android приложений | 2 | 2 | 1 |
| Тестирование мобильных приложений iOS | 1 | 1 | - |
| Сбор и анализ трафика (Charles Proxy, Fiddler) | 3 | 3 | - |
| Подготовка отчета |  |  |  |
| Создание отчета | - | - | 6 |
| Создание сайта | - | - | 8 |
| Подготовка постера | - | - | 3 |
| Создание презентации | - | - | 2 |
| Запись и монтаж видео презентации | - | - | 2 |
| Подготовка Git-репозитория | - | - | 1 |

Таблица 1 «Общий план работ».

**Участники и их роли.**

* Дилфуза Голомовна №группы 194-321 - тимлид, тестировщик iOS.
* Фурман Кирилл №группы 181-331 - разработка сайта, тестировщик iOS.
* Махмадзиеев Али №группы 181-331 - разработка постера, тестировщик iOS.
* Великанов Иван №группы 181-331 - тестировщик Android, разработка сайта.

1. **ТЕСТИРОВАНИЕ API** 
   1. **Программное обеспечение для автоматизации развёртывания и управления приложениями**

Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой [контейнеризации](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D0%B5%D0%B9%D0%BD%D0%B5%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), контейнеризатор приложений. Позволяет «упаковать» приложение со всем его [окружением](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%BA%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5&action=edit&redlink=1) и зависимостями в контейнер, который может быть развёрнут на любой [Linux](https://ru.wikipedia.org/wiki/Linux)-системе с поддержкой [cgroups](https://ru.wikipedia.org/wiki/Cgroups" \o "Cgroups) в [ядре](https://ru.wikipedia.org/wiki/%D0%AF%D0%B4%D1%80%D0%BE_Linux), а также предоставляет набор команд для управления этими контейнерами. Изначально использовал возможности [LXC](https://ru.wikipedia.org/wiki/LXC), с 2015 года начал использовать собственную библиотеку, абстрагирующую виртуализационные возможности ядра Linux — libcontainer. С появлением Open Container Initiative начался переход от монолитной к модульной архитектуре.

Разрабатывается и поддерживается одноимённой компанией-[стартапом](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%80%D1%82%D0%B0%D0%BF), распространяется в двух редакциях — общественной (*Community Edition*) по [лицензии Apache 2.0](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D1%86%D0%B5%D0%BD%D0%B7%D0%B8%D1%8F_Apache) и для организаций (*Enterprise Edition*) по [проприетарной](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D1%80%D0%B8%D0%B5%D1%82%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) лицензии. Написан на языке [Go](https://ru.wikipedia.org/wiki/Go_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)).

Vagrant — продукт компании HashiCorp, специализирующейся на инструментах для автоматизации разработки и эксплуатации. Он позволяет создавать и конфигурировать легковесные, повторяемые и переносимые окружения для разработки.Vagrant — продукт компании HashiCorp, специализирующейся на инструментах для автоматизации разработки и эксплуатации. Он позволяет создавать и конфигурировать легковесные, повторяемые и переносимые окружения для разработки.

Vagrant позволяет легко создать несколько виртуальных машин и управлять ими. Он построен поверх VirtualBox и VMWare, и он обеспечивает множество интересных возможностей. Вы можете создавать изолированные среды разработки, экспериментировать с новыми продуктами и технологиями, устанавливать новые версии существующих пакетов, создавать свой собственный центр обработки данных на своем ноутбуке и запускать различные операционные системы. Все это доступно в простом в управлении и полностью безопасном бункере, что в итоге не может помешать вашим основным действиям и операционной системе.

* 1. **Контейнеры и что это такое**

Каждый контейнер строится на основе Docker-образов. Контейнеры запускаются напрямую из ядра операционной системы Linux. Благодаря этому, они потребляют гораздо меньше ресурсов, чем при аппаратной виртуализации. Изоляция рабочей среды осуществляется при помощи технологии namespace. Для каждого изолированного пространства (контейнера) создается уникальное пространство имен, которое и обеспечивает к нему доступ. Любой процесс, выполняемый внутри контейнера, ограничивается namespace.

* 1. **Основные команды docker**

В таблице 1 представлены основные команды docker и их значение.

Таблица 1 - Значение команд docker

|  |  |
| --- | --- |
| Команда | Значение команды |
| docker create | создание контейнера |
| docker ps | просмотр контейнеров |
| docker rename | изменение имени контейнера |
| docker rm | удаление контейнера |
| docker run | запуск контейнера |
| docker stop | остановка контейнера |
| docker restart | перезагрузка контейнера |
| docker build | создание образа |
| docker network ls | просмотр сетей докера |

1. **БАЗЫ ДАННЫХ**

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базой данных (СУБД). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных, или, для краткости, просто базой данных.

Данные в наиболее распространенных типах современных баз данных обычно хранятся в виде строк и столбцов формирующих таблицу. Этими данными можно легко управлять, изменять, обновлять, контролировать и упорядочивать. В большинстве баз данных для записи и запросов данных используется язык структурированных запросов (SQL).

Типы баз данных:

* Реляционные базы данных.

Данные в реляционной базе организованы в виде таблиц, состоящих из столбцов и строк. Реляционная СУБД обеспечивает быстрый и эффективный доступ к структурированной информации.

* Объектно-ориентированные базы данных.

Информация в объектно-ориентированной базе данных представлена в форме объекта, как в объектно-ориентированном программировании.

* Распределенные базы данных.

Распределенная база данных состоит из двух или более частей, расположенных на разных серверах. Такая база данных может храниться на нескольких компьютерах.

* Хранилища данных.

Будучи централизованным репозиторием для данных, хранилище данных представляет собой тип базы данных, специально предназначенной для быстрого выполнения запросов и анализа.

* Базы данных NoSQL.

База данных NoSQL, или нереляционная база данных, дает возможность хранить и обрабатывать неструктурированные или слабоструктурированные данные (в отличие от реляционной базы данных, задающей структуру содержащихся в ней данных). Популярность баз данных NoSQL растет по мере распространения и усложнения веб-приложений.

* Графовые базы данных.

Графовая база данных хранит данные в контексте сущностей и связей между сущностями.

* Базы данных OLTP.

База данных OLTP — это база данных предназначенная для выполнения бизнес-транзакций, выполняемых множеством пользователей.

* 1. **Системы управления базами данных (MongoDB, PostgreSQL, MySQL)**

MongoDB - документоориентированная система управления базами данных, не требующая описания схемы таблиц. Считается одним из классических примеров NOSQL-систем, использует JSON-подобные документы и схему базы данных. Применяется в веб-разработке, в частности, в рамках JavaScript-ориентированного стека MEAN.

MongoDB подходит для следующих применений:

* хранение и регистрация событий;
* системы управления документами и контентом;
* электронная коммерция;
* игры;
* данные мониторинга, датчиков;
* мобильные приложения;
* хранилище операционных данных веб-страниц (например, хранение комментариев, рейтингов, профилей пользователей, сеансы пользователей).

PostgreSQL — СУБД с открытым исходным кодом, основой которого был код, написанный в Беркли. Она поддерживает большую часть стандарта SQL и предлагает множество современных функций:

* сложные запросы
* внешние ключи
* триггеры
* изменяемые представления
* транзакционная целостность
* многоверсионность

Сильные стороны PostgreSQL:

* высокопроизводительные и надежные механизмы транзакций и репликации;
* расширяемая система встроенных языков программирования: в стандартной поставке поддерживаются PL.pgSQL, PL/Perl, PL/Python и PL/TCL; дополнительно можно использовать PL/Java, PL/PHP, PL/Py, PL/R, PL/Ruby, PL/Scheme, PL/sh и Pl/V8, а также имеется поддержка загрузки модулей расширения на языке C;
* наследование;
* возможность индексирования геометрических (в частности, географических) объектов и наличие базирующегося на ней расширения PostGIS;
* встроенная поддержка слабоструктурированных данных в формате JSON с возможностью их индексации;
* расширяемость (возможность создавать новые типы данных, типы индексов, языки программирования, модули расширения, подключать любые внешние источники данных).

MySQL — это реляционная система управления базами данных с открытым исходным кодом. В настоящее время эта СУБД одна из наиболее популярных в веб-приложениях — подавляющее большинство CMS использует именно MySQL (часто только её, без альтернатив), а почти все веб-фреймворки поддерживают MySQL уже на уровне базовой конфигурации (без дополнительных модулей).

Чаще всего MySQL используют при разработке веб-решений, что объясняется тесной интеграцией с популярными языками программирования, высокими показателями скорости и открытым исходным кодом. В основном MySQL применяют в следующих проектах:

* CMS (Content Management System). Именно в системах управления содержимым сайта MySQL (в связке с PHP/Apache) обгоняет всех своих конкурентов.
* Корпоративные системы, ERP/CRM-приложения. У малого и среднего бизнеса обычно нет возможности приобретать такие продукты от ведущих игроков (SAP, Microsoft, Oracle), но подобное ПО бывает необходимым для успешного ведения бизнеса. Часто компании разрабатывают собственные решения с применением MySQL, которая покрывает все потребности разработчиков и пользователей такого рода систем.
* Groupware — системы групповой работы, планировщики и подобные системы. Частично они входят в состав CMS, но в большинстве случаев являются самостоятельными независимыми программами.
* Форумы и чаты. Примеры — phpBB, yaBB, Invision Power Board и другие.

MySQL — это реляционная система управления базами данных с открытым исходным кодом. В настоящее время эта СУБД одна из наиболее популярных в веб-приложениях — подавляющее большинство CMS использует именно MySQL (часто только её, без альтернатив), а почти все веб-фреймворки поддерживают MySQL уже на уровне базовой конфигурации (без дополнительных модулей).

Чаще всего MySQL используют при разработке веб-решений, что объясняется тесной интеграцией с популярными языками программирования, высокими показателями скорости и открытым исходным кодом. В основном MySQL применяют в следующих проектах:

* CMS (Content Management System). Именно в системах управления содержимым сайта MySQL (в связке с PHP/Apache) обгоняет всех своих конкурентов.
* Корпоративные системы, ERP/CRM-приложения. У малого и среднего бизнеса обычно нет возможности приобретать такие продукты от ведущих игроков (SAP, Microsoft, Oracle), но подобное ПО бывает необходимым для успешного ведения бизнеса. Часто компании разрабатывают собственные решения с применением MySQL, которая покрывает все потребности разработчиков и пользователей такого рода систем.
* Groupware — системы групповой работы, планировщики и подобные системы. Частично они входят в состав CMS, но в большинстве случаев являются самостоятельными независимыми программами.
* Форумы и чаты. Примеры — phpBB, yaBB, Invision Power Board и другие.
  1. **GUI (Robo 3T, pgAdmin, PHPMyAdmin)**

Robo 3T (ранее Robomongo) - это кросс-платформенный инструмент управления MongoDB, ориентированный на оболочку. В отличие от большинства других инструментов пользовательского интерфейса администратора MongoDB, Robo 3T встраивает фактическую оболочку монго в интерфейс с вкладками с доступом к командной строке оболочки и взаимодействием с графическим интерфейсом.

pgAdmin – это открытая платформа администрирования и разработки для PostgreSQL и связанных с ней систем управления базами данных. Написанная на Python и jQuery, эта платформа поддерживает все функции PostgreSQL. Вы можете использовать pgAdmin для выполнения любых задач: от написания базовых SQL-запросов до мониторинга ваших баз данных и настройки расширенных архитектур БД.

phpMyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PhpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд.

Приложение распространяется под лицензией GNUGeneral Public License и поэтому многие другие разработчики интегрируют его в свои разработки, например XAMPP, Denwer, AppServ, Open Server.

Проект на данный момент времени локализован на более чем 62 языках.

1. **ОТЧЕТ ПО ТЕСТИРОВАНИЮ ПРОЕКТА**

В результате тестирования приложения “юла mini app” были составлены чек-листы и тест-кейсы. Далее показан тест-кейс. Каждый элемент тест-кейса содержит в себе предусловия, важность кейса, шаги и ожидаемый результат.

Тест-кейсы “своя”-карточка IOS

**General:**

**Запчасти и авто товары:**

**Недвижимость:**

**Услуги исполнителей:**

Для примера использовался Iphone XR IOs 14.5.1

General:

* Проверка перехода назад с карточки в профиль (тап "назад")

Предусловия: открыта своя карточка товара

Шаги:

1. Тап “Назад”

Ожидаемый результат: Отображение экрана “Мой профиль”

Макет: <https://yadi.sk/i/JDFMBk5AQqPVBQ>

* Проверка перехода назад с карточки в профиль (скролл вправо системная функция)

Предусловия: открыта своя карточка товара

Шаги:

1. Скролл вправо

Ожидаемый результат: Отображение экрана “Мой профиль”

Макет: <https://yadi.sk/i/JDFMBk5AQqPVBQ>

* Открытие картинки на весь экран

Предусловия: открыта своя карточка товара,  у товара есть картинка

Шаги:

1. Тап по картинке

Ожидаемый результат: Корректное отображение картинки на весь экран

Макет: <https://yadi.sk/i/noeo_lMvzhEx_g>

* Скролл картинок объявления (не полное отображение)

Предусловия: у товара есть несколько картинок

Шаги:

1. Скролл горизонтальный картинок влево (картинка 1)

Ожидаемый результат: Корректное отображение следующей картинки  (картинка 2)

Макет: <https://yadi.sk/i/XDpCVIpCTdRmWQ>

2. Скролл горизонтальный картинок вправо  (картинка 2)

Ожидаемый результат: Корректное отображение предыдущей картинки  (картинка 1)

Макет: <https://yadi.sk/i/PdH9CeBYCk0BVA>

* Скролл картинок объявления (отображение на весь экран)

Предусловия: у товара есть несколько картинок, картинка открыта на весь экран

1. Скролл горизонтальный картинок влево (картинка 1)

Ожидаемый результат: Корректное отображение следующей картинки  (картинка 2)

Макет: <https://yadi.sk/i/rf2c0OhAwvwUJA>

2. Скролл горизонтальный картинок вправо  (картинка 2)

Ожидаемый результат: Корректное отображение предыдущей картинки  (картинка 1)

Макет: <https://yadi.sk/i/noeo_lMvzhEx_g>

* Проверка нажатие на троеточие в правом верхнем углу

Предусловия: открыта своя карточка товара

Макет: <https://yadi.sk/i/iJ1vao6X_RxCng>

1. Тап кнопка «Изменить»

Макет: <https://yadi.sk/i/PyW9_XnlPSzTgA>

1. Тап кнопка «Поделиться»

Макет: <https://yadi.sk/i/EeKiTa5TxHTctA>

1. Тап кнопка «Снять с публикации»

Макет: <https://yadi.sk/i/oibFSOvF_OP0eg>

* Проверка перехода на экран "Поделится в сообществах"

Предусловия: карточка опубликована

Шаги:

1. Тап “Поделится”

Ожидаемый результат: Отображение экрана “Поделится в сообществах”

Макет: <https://yadi.sk/i/6V_whsKjAtbzCg>

* Проверка кнопки «Снять с публикации»

Предусловия: карточка опубликована

Шаги:

1. Тап “Снять с публикации”

Ожидаемый результат: Отображение баннера с причинами снятия с публикации

Макет: <https://yadi.sk/i/oibFSOvF_OP0eg>

* Проверка перехода на экран с гео-меткой

Предусловия: указан адрес

Шаги:

1. Тап по разделу с адресом

Ожидаемый результат: Отображение экрана с картой

Макет: <https://yadi.sk/i/szfNOnBRqBqmcA>

* Скролл карточки товара

Предусловия: карточка открыта

Шаги:

1. Скролл карточки вниз

Ожидаемый результат: Разделы отображается корректно

Макет: <https://yadi.sk/i/a7NLRH8VVL5Aow>

* Проверка отображение информации

Предусловия: Длинное описание, скрыто за троеточием

Шаги:

1. Тап “Читать дальше”

Ожидаемый результат: Описание отображается полностью

Макет: <https://yadi.sk/i/r9lf8gFwximozQ>

* Проверка перехода на экран пользователя

Предусловия: карточка открыта

Шаги:

1. Тап на раздел с пользователем

Ожидаемый результат: Корректно отображается экран профиля пользователя

Макет: <https://yadi.sk/i/B_NrEhdkHn6HoA>

* Проверка отображение блока «Рекомендуемые сообщества»

Предусловия: карточка опубликована

Шаги:

1. Проверка отображения блока поделится

Ожидаемый результат: Раздел отображается корректно (соц. сети, копировать ссылку, троеточие)

Макет: <https://yadi.sk/i/Qv5xJI8x4_AXqQ>

* Проверка отображения «Похожие объявления»

Предусловия: открыта карточка товара

Шаги:

1. Проверка отображения раздела похожие объявления

Ожидаемый результат: Раздел отображается корректно, отображаются похожие объявления

Макет: <https://yadi.sk/i/LRAG-zUInaizaA>

* Багов не обнаружено

Для примера использовался Iphone XR IOs 14.5.1

Запчасти и авто товары:

* Проверка перехода назад с карточки в профиль (тап "назад")

Предусловия: открыта своя карточка товара

Шаги:

1. Тап “Назад”

Ожидаемый результат: Отображение экрана “Мой профиль”

Макет: <https://yadi.sk/i/C-PCsIPPFGGU3Q>

* Проверка перехода назад с карточки в профиль (скролл вправо системная функция)

Предусловия: открыта своя карточка товара

Шаги:

1. Скролл вправо

Ожидаемый результат: Отображение экрана “Мой профиль”

Макет: <https://yadi.sk/i/7vPWlt47-_Vr6w>

* Открытие картинки на весь экран

Предусловия: открыта своя карточка товара,  у товара есть картинка

Шаги:

1. Тап по картинке

Ожидаемый результат: Корректное отображение картинки на весь экран

Макет: <https://yadi.sk/i/2k3M12hFBUlmcw>

* Скролл картинок объявления (не полное отображение)

Предусловия: у товара есть несколько картинок

Шаги:

1. Скролл горизонтальный картинок влево (картинка 1)

Ожидаемый результат: Корректное отображение следующей картинки  (картинка 2)

Макет: <https://yadi.sk/i/Q3F5QomA6bJtkA>

2. Скролл горизонтальный картинок вправо  (картинка 2)

Ожидаемый результат: Корректное отображение предыдущей картинки  (картинка 1)

Макет: <https://yadi.sk/i/eZiYKdLHooscTA>

* Скролл картинок объявления (отображение на весь экран)

Предусловия: у товара есть несколько картинок, картинка открыта на весь экран

1. Скролл горизонтальный картинок влево (картинка 1)

Ожидаемый результат: Корректное отображение следующей картинки  (картинка 2)

Макет: <https://yadi.sk/i/aLZL4aGttoQCQQ>

2. Скролл горизонтальный картинок вправо  (картинка 2)

Ожидаемый результат: Корректное отображение предыдущей картинки  (картинка 1)

Макет: <https://yadi.sk/i/ejtSStfPZg4S4A>

* Проверка нажатие на троеточие в правом верхнем углу

Предусловия: открыта своя карточка товара

Макет: <https://yadi.sk/i/KPhJ9NCjLrUYbg>

1. Тап кнопка «Изменить»

Макет: <https://yadi.sk/i/DNVQqaVp7uVEQQ>

1. Тап кнопка «Поделиться»

Макет: <https://yadi.sk/i/tBpU-Y8GPdRbbA>

1. Тап кнопка «Снять с публикации»

Макет: <https://yadi.sk/i/schbIi-s8o4t9Q>

* Проверка перехода на экран "Поделится в сообществах"

Предусловия: карточка опубликована

Шаги:

1. Тап “Поделится”

Ожидаемый результат: Отображение экрана “Поделится в сообществах”

Макет: <https://yadi.sk/i/E3cRD0oHOP8ZkA>

* Проверка кнопки «Снять с публикации»

Предусловия: карточка опубликована

Шаги:

1. Тап “Снять с публикации”

Ожидаемый результат: Отображение баннера с причинами снятия с публикации

Макет: <https://yadi.sk/i/Pv25zqmxm39aTA>

* Проверка перехода на экран с гео-меткой

Предусловия: указан адрес

Шаги:

1. Тап по разделу с адресом

Ожидаемый результат: Отображение экрана с картой

Макет: <https://yadi.sk/i/uuD4Qr0plKgfgw>

* Скролл карточки товара

Предусловия: карточка открыта

Шаги:

1. Скролл карточки вниз

Ожидаемый результат: Разделы отображается корректно

Макет: <https://yadi.sk/i/hZmFVzyD45bewA>

* Проверка отображение информации

Предусловия: Длинное описание, скрыто за троеточием

Шаги:

1. Тап “Читать дальше”

Ожидаемый результат: Описание отображается полностью

Макет: <https://yadi.sk/i/TY7LpQBWxlfGYw>

* Проверка перехода на экран пользователя

Предусловия: карточка открыта

Шаги:

1. Тап на раздел с пользователем

Ожидаемый результат: Корректно отображается экран профиля пользователя

Макет: <https://yadi.sk/i/BzT0yh1zrfoO8Q>

* Проверка отображение блока «Рекомендуемые сообщества»

Предусловия: карточка опубликована

Шаги:

1. Проверка отображения блока поделится

Ожидаемый результат: Раздел отображается корректно (соц. сети, копировать ссылку, троеточие)

Макет: <https://yadi.sk/i/11PneSciJA8nnQ>

* Проверка отображения «Похожие объявления»

Предусловия: открыта карточка товара

Шаги:

1. Проверка отображения раздела похожие объявления

Ожидаемый результат: Раздел отображается корректно, отображаются похожие объявления

Макет: <https://yadi.sk/i/gAZHgA7haLmcdw>

Для примера использовался Iphone XR IOs 14.5.1

Недвижимость:

* Проверка перехода назад с карточки в профиль (тап "назад")

Предусловия: открыта своя карточка товара

Шаги:

1. Тап “Назад”

Ожидаемый результат: Отображение экрана “Мой профиль”

Макет: <https://yadi.sk/i/qkz5NUwPRmZgzQ>

* Проверка перехода назад с карточки в профиль (скролл вправо системная функция)

Предусловия: открыта своя карточка товара

Шаги:

1. Скролл вправо

Ожидаемый результат: Отображение экрана “Мой профиль”

Макет: <https://yadi.sk/i/qSuguoyjYBSiAA>

* Открытие картинки на весь экран

Предусловия: открыта своя карточка товара,  у товара есть картинка

Шаги:

1. Тап по картинке

Ожидаемый результат: Корректное отображение картинки на весь экран

Макет: <https://yadi.sk/i/_0Ys5T64PPi4Gw>

* Скролл картинок объявления (не полное отображение)

Предусловия: у товара есть несколько картинок

Шаги:

1. Скролл горизонтальный картинок влево (картинка 1)

Ожидаемый результат: Корректное отображение следующей картинки  (картинка 2)

Макет: <https://yadi.sk/i/kSGgAONxPsvgfw>

2. Скролл горизонтальный картинок вправо  (картинка 2)

Ожидаемый результат: Корректное отображение предыдущей картинки  (картинка 1)

Макет: <https://yadi.sk/i/k7FBa1t26lCchg>

* Скролл картинок объявления (отображение на весь экран)

Предусловия: у товара есть несколько картинок, картинка открыта на весь экран

1. Скролл горизонтальный картинок влево (картинка 1)

Ожидаемый результат: Корректное отображение следующей картинки  (картинка 2)

Макет: <https://yadi.sk/i/_0Ys5T64PPi4Gw>

2. Скролл горизонтальный картинок вправо  (картинка 2)

Ожидаемый результат: Корректное отображение предыдущей картинки  (картинка 1)

Макет: <https://yadi.sk/i/x4e1GCmxx7NwTw>

* Проверка нажатие на троеточие в правом верхнем углу

Предусловия: открыта своя карточка товара

Макет: <https://yadi.sk/i/nJXjQ_IkXz27QQ>

1. Тап кнопка «Изменить»

Макет: <https://yadi.sk/i/-Wy9GtIRI4US4A>

1. Тап кнопка «Поделиться»

Макет: <https://yadi.sk/i/cNxptKoTP8QAKg>

1. Тап кнопка «Снять с публикации»

Макет: <https://yadi.sk/i/GrF03_aqBp2oVg>

* Проверка перехода на экран "Поделится в сообществах"

Предусловия: карточка опубликована

Шаги:

1. Тап “Поделится”

Ожидаемый результат: Отображение экрана “Поделится в сообществах”

Макет: <https://yadi.sk/i/x4e1GCmxx7NwTw>

* Проверка кнопки «Снять с публикации»

Предусловия: карточка опубликована

Шаги:

1. Тап “Снять с публикации”

Ожидаемый результат: Отображение баннера с причинами снятия с публикации

Макет: <https://yadi.sk/i/hFwi0BA-RPo6sg>

* Проверка перехода на экран с гео-меткой

Предусловия: указан адрес

Шаги:

1. Тап по разделу с адресом

Ожидаемый результат: Отображение экрана с картой

Макет: <https://yadi.sk/i/6iRTl-1B7YLKgQ>

* Скролл карточки товара

Предусловия: карточка открыта

Шаги:

1. Скролл карточки вниз

Ожидаемый результат: Разделы отображается корректно

Макет: <https://yadi.sk/i/rCvQVvmcUmvkVA>

* Проверка отображение информации

Предусловия: Длинное описание, скрыто за троеточием

Шаги:

1. Тап “Читать дальше”

Ожидаемый результат: Описание отображается полностью

Макет: <https://yadi.sk/i/Ri8yym2BfhKQ7Q>

* Проверка перехода на экран пользователя

Предусловия: карточка открыта

Шаги:

1. Тап на раздел с пользователем

Ожидаемый результат: Корректно отображается экран профиля пользователя

Макет: <https://yadi.sk/i/aKZj3H-qRgnwdg>

* Проверка отображение блока «Рекомендуемые сообщества»

Предусловия: карточка опубликована

Шаги:

1. Проверка отображения блока поделится

Ожидаемый результат: Раздел отображается корректно (соц. сети, копировать ссылку, троеточие)

Макет: <https://yadi.sk/i/335AgDtPuR6LEw>

Для примера использовался Iphone X IOS 14.5

Услуги исполнителей:

* Проверка перехода назад с карточки в профиль (тап "назад")

Предусловия: открыта своя карточка товара

Шаги:

1. Тап “Назад”

Ожидаемый результат: Отображение экрана “Мой профиль”

Макет: <https://yadi.sk/i/8WDPAHO5hu9l_A>

* Проверка перехода назад с карточки в профиль (скролл вправо системная функция)

Предусловия: открыта своя карточка товара

Шаги:

1. Скролл вправо

Ожидаемый результат: Отображение экрана “Мой профиль”

Макет: <https://yadi.sk/i/IfTfyQhLYjR7SQ>

* Открытие картинки на весь экран

Предусловия: открыта своя карточка товара,  у товара есть картинка

Шаги:

1. Тап по картинке

Ожидаемый результат: Корректное отображение картинки на весь экран

Макет: <https://yadi.sk/i/h8y027p3InGBPQ>

* Скролл картинок объявления (не полное отображение)

Предусловия: у товара есть несколько картинок

Шаги:

1. Скролл горизонтальный картинок влево (картинка 1)

Ожидаемый результат: Корректное отображение следующей картинки  (картинка 2)

Макет: <https://yadi.sk/i/0uE5JWJLPjS-wg>

2. Скролл горизонтальный картинок вправо  (картинка 2)

Ожидаемый результат: Корректное отображение предыдущей картинки  (картинка 1)

Макет: <https://yadi.sk/i/RoI7Dc1Sdw2H4Q>

* Скролл картинок объявления (отображение на весь экран)

Предусловия: у товара есть несколько картинок, картинка открыта на весь экран

1. Скролл горизонтальный картинок влево (картинка 1)

Ожидаемый результат: Корректное отображение следующей картинки  (картинка 2)

Макет: <https://yadi.sk/i/h8y027p3InGBPQ>

2. Скролл горизонтальный картинок вправо  (картинка 2)

Ожидаемый результат: Корректное отображение предыдущей картинки  (картинка 1)

Макет: <https://yadi.sk/i/yI-NafuJoqxSkQ>

* Проверка нажатие на троеточие в правом верхнем углу

Предусловия: открыта своя карточка товара

Макет: <https://yadi.sk/i/Xui8GI6tNUf_jg>

1. Тап кнопка «Изменить»

Макет: <https://yadi.sk/i/GtCkghZ8GRsejA>

1. Тап кнопка «Поделиться»

Макет: <https://yadi.sk/i/-b3B7SSTPQO9mQ>

1. Тап кнопка «Снять с публикации»

Макет: <https://yadi.sk/i/0yNgqBjyjxvC-Q>

* Проверка перехода на экран "Поделится в сообществах"

Предусловия: карточка опубликована

Шаги:

1. Тап “Поделится”

Ожидаемый результат: Отображение экрана “Поделится в сообществах”

Макет: <https://yadi.sk/i/-b3B7SSTPQO9mQ>

* Проверка кнопки «Снять с публикации»

Предусловия: карточка опубликована

Шаги:

1. Тап “Снять с публикации”

Ожидаемый результат: Отображение баннера с причинами снятия с публикации

Макет: <https://yadi.sk/i/0yNgqBjyjxvC-Q>

* Проверка перехода на экран с гео-меткой

Предусловия: указан адрес

Шаги:

1. Тап по разделу с адресом

Ожидаемый результат: Отображение экрана с картой

Макет: <https://yadi.sk/i/E2DqHAjfKcKj5g>

* Скролл карточки товара

Предусловия: карточка открыта

Шаги:

1. Скролл карточки вниз

Ожидаемый результат: Разделы отображается корректно

Макет: <https://yadi.sk/i/Se87tuTUsUoIRQ>

* Проверка отображение информации

Предусловия: Длинное описание, скрыто за троеточием

Шаги:

1. Тап “Читать дальше”

Ожидаемый результат: Описание отображается полностью

Макет: <https://yadi.sk/i/iySjclZO8kx7Qg>

* Проверка перехода на экран пользователя

Предусловия: карточка открыта

Шаги:

1. Тап на раздел с пользователем

Ожидаемый результат: Корректно отображается экран профиля пользователя

Макет: <https://yadi.sk/i/ZUeyHqK4HE-A5g>

* Проверка отображение блока «Рекомендуемые сообщества»

Предусловия: карточка опубликована

Шаги:

1. Проверка отображения блока поделится

Ожидаемый результат: Раздел отображается корректно (соц. сети, копировать ссылку, троеточие)

Макет: <https://yadi.sk/i/EVlSO8ecEKWtJQ>

* Проверка отображения «Похожие объявления»

Предусловия: открыта карточка товара

Шаги:

1. Проверка отображения раздела похожие объявления

Ожидаемый результат: Раздел отображается корректно, отображаются похожие объявления

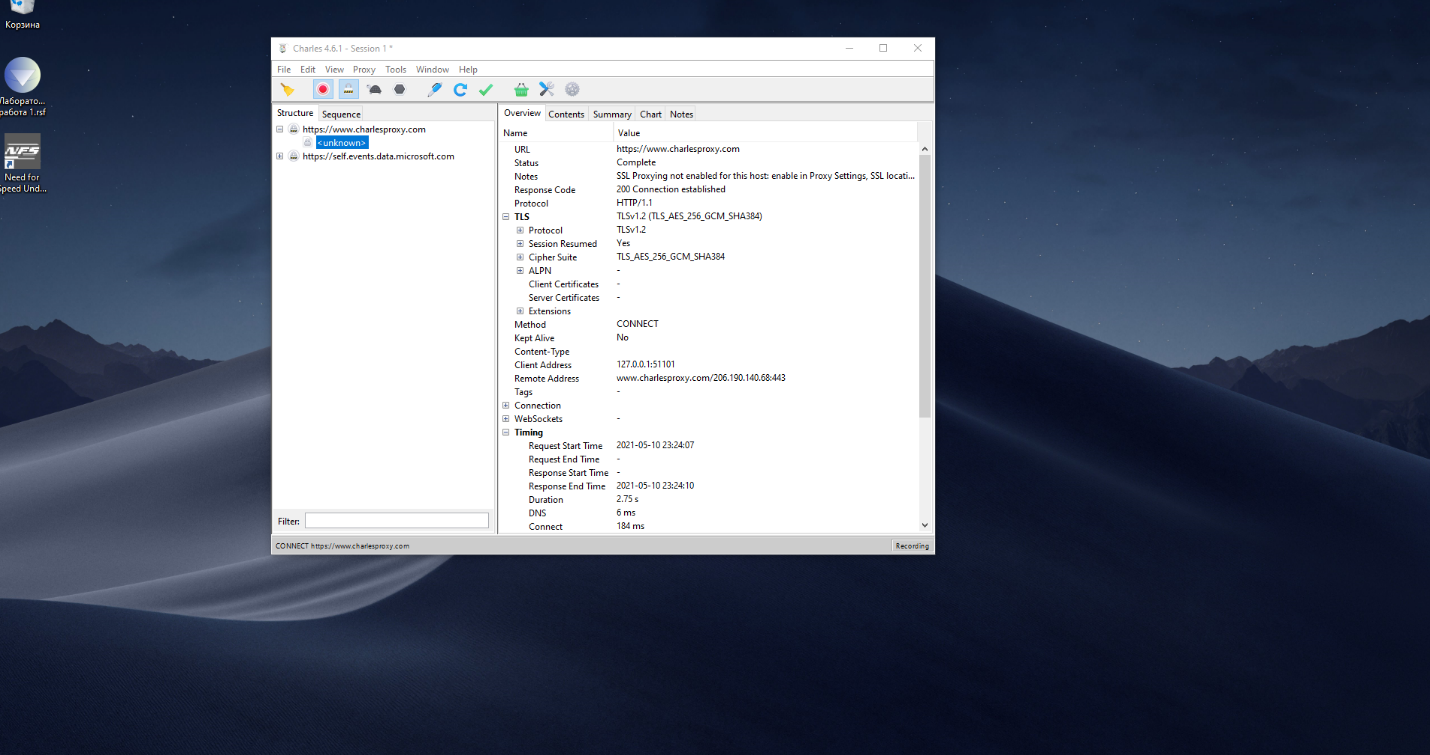
Макет: <https://yadi.sk/i/6XGJTvAKmb_01g>

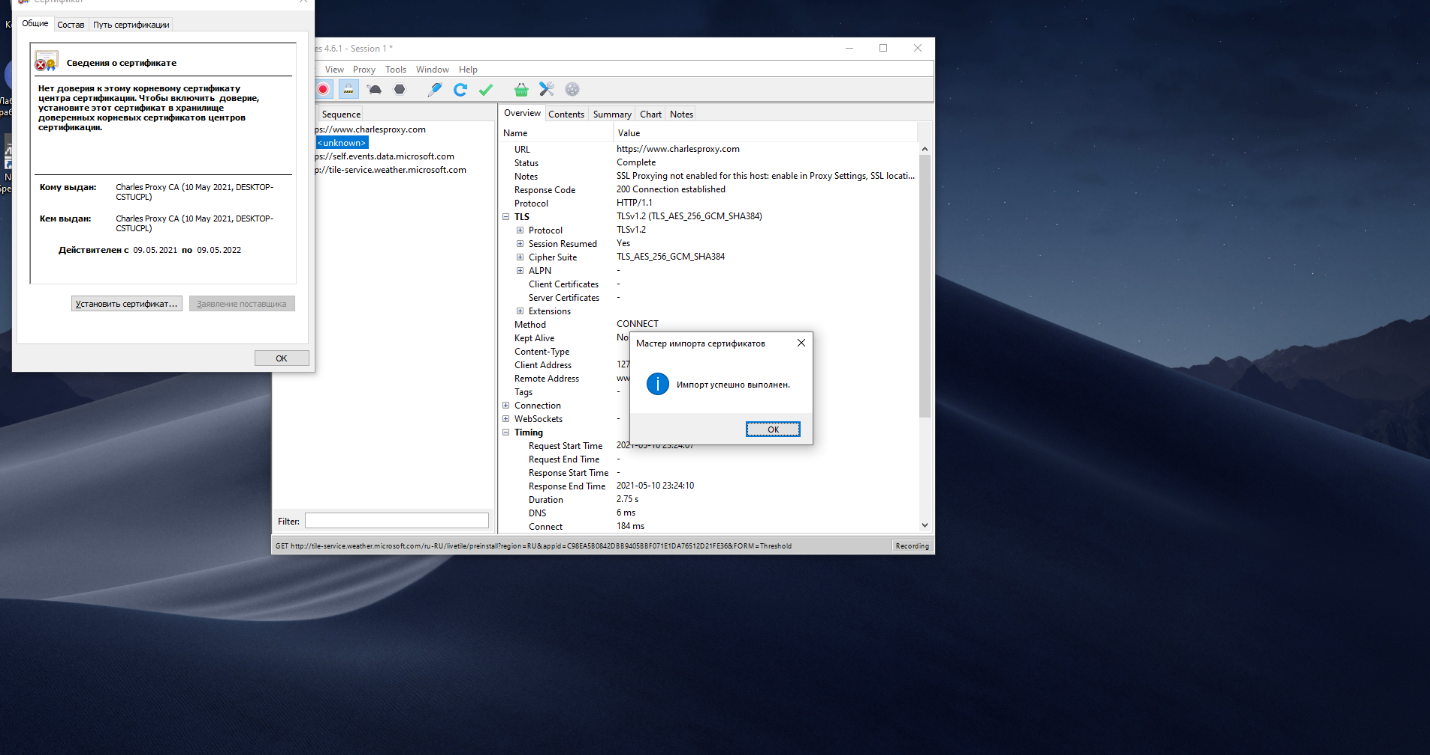
**Мини-проект подмена информации о бургере в приложении Макдоналдс**

1) Charles Proxy

Установка программы

<https://www.charlesproxy.com/download/latest-release/>

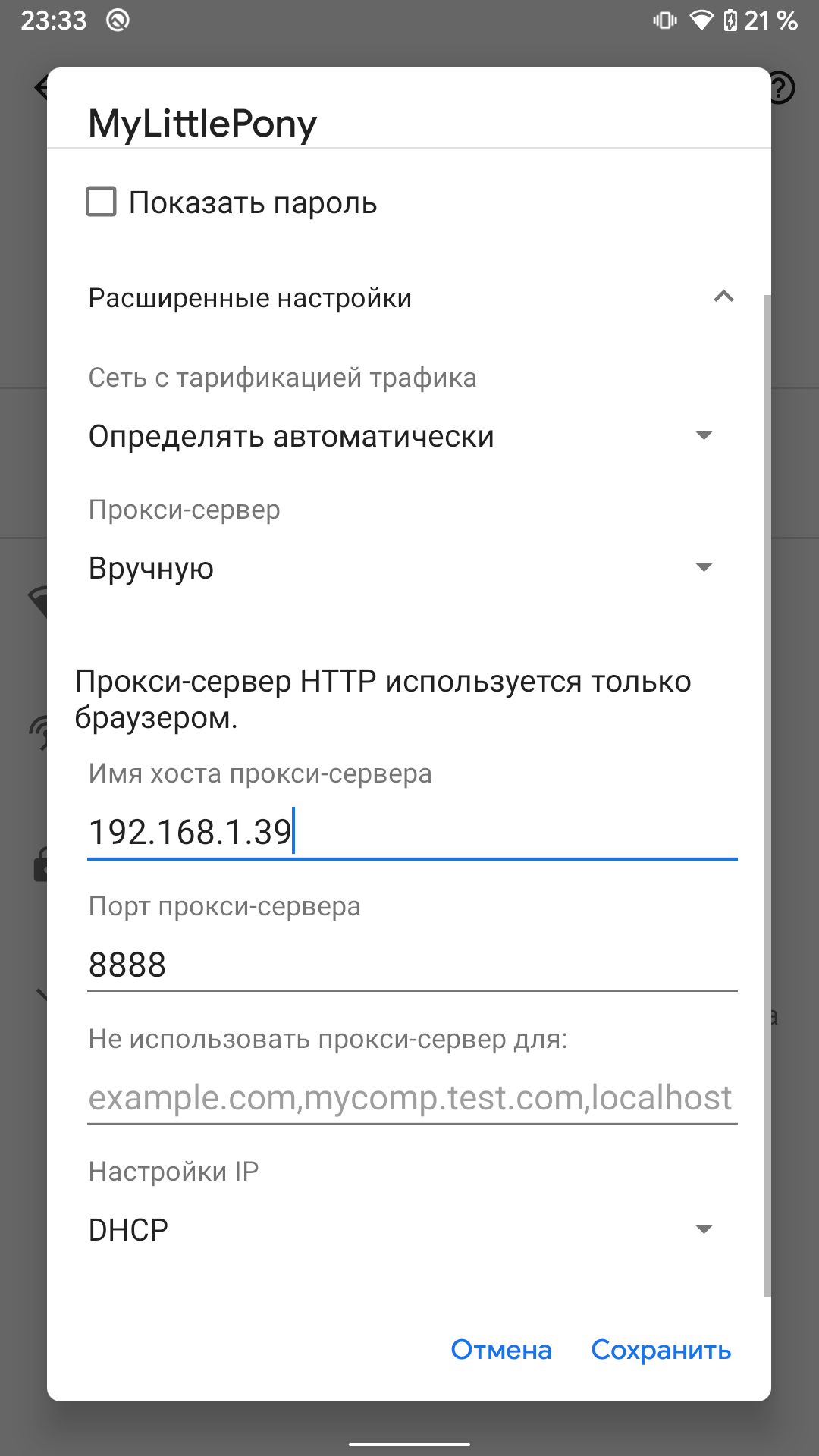
Установка сертфиката.



Сохраненный сертификат:



Настройки на телефоне:

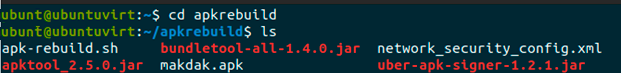


После настройки был установлен сертификат (chls.pro/ssl)

2) Apk Rebuild

Ссылка на apk: https://apps.evozi.com/apk-downloader/?id=com.apegroup.mcdonaldsrussia

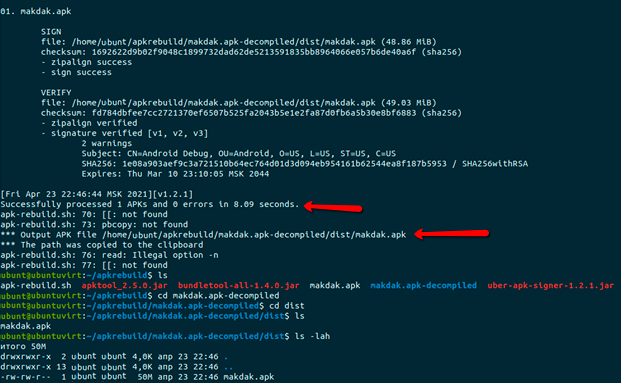
Необходимые файлы (исключение – xml-файл не использовался, но его наличие в директории обусловлено неудачной попыткой пересборки apk)  

Изменения в sh-скрипте (в соответствии со статьёй в https://telegra.ph/Perehvat-SSL-trafika-s-Android-prilozheniya-01-26)



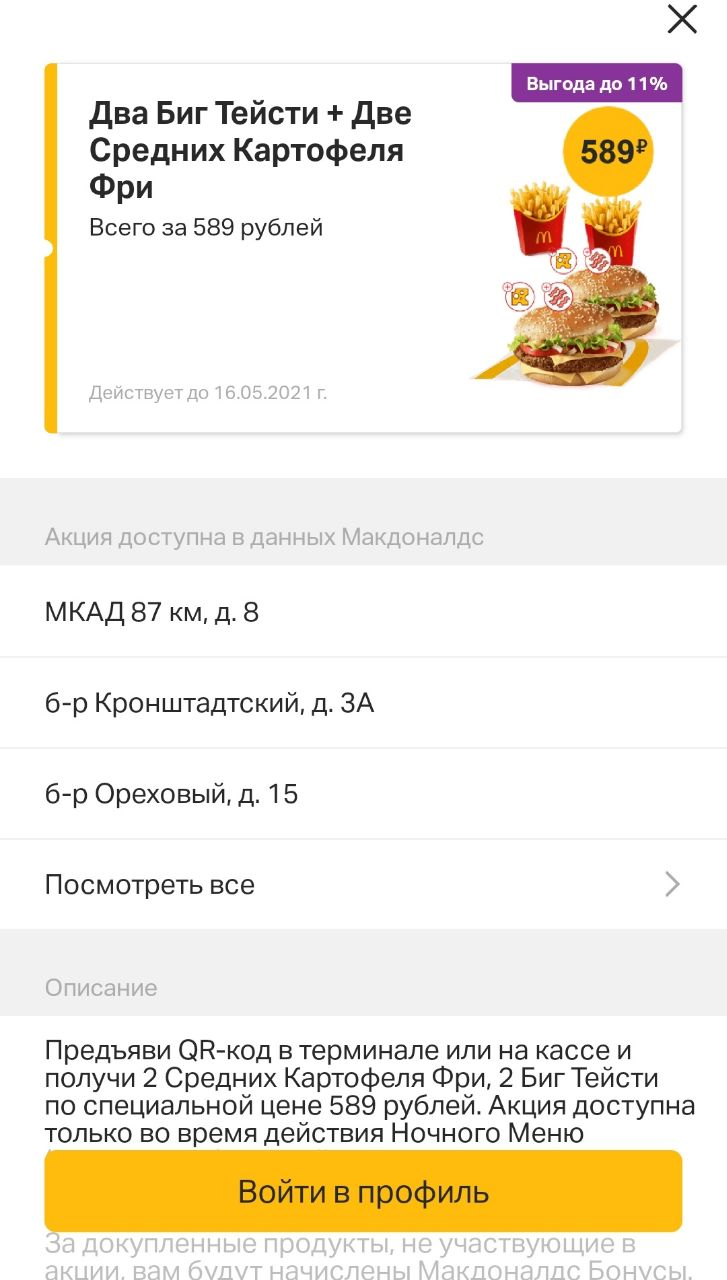
Пересборка



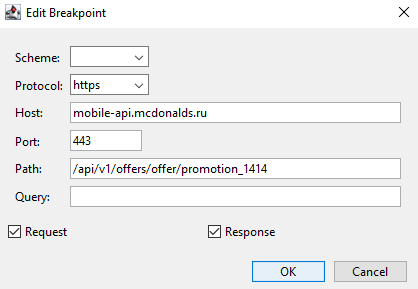
В ~/apkrebuild/makdak.apk-decompiled/dist находится пересобранный apk.

3) Применение Charles на пересобранном apk

Выбранное предложение



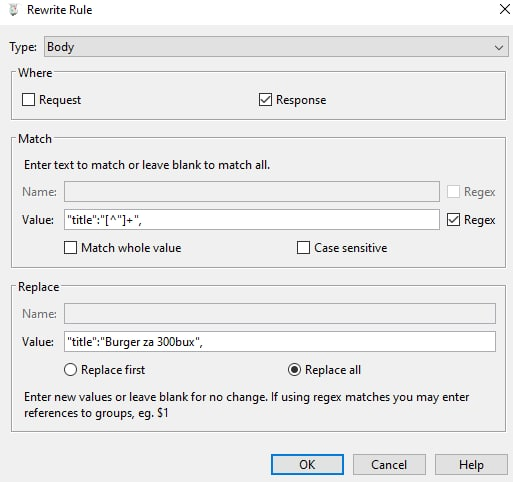
Часть 1. Breakpoint.

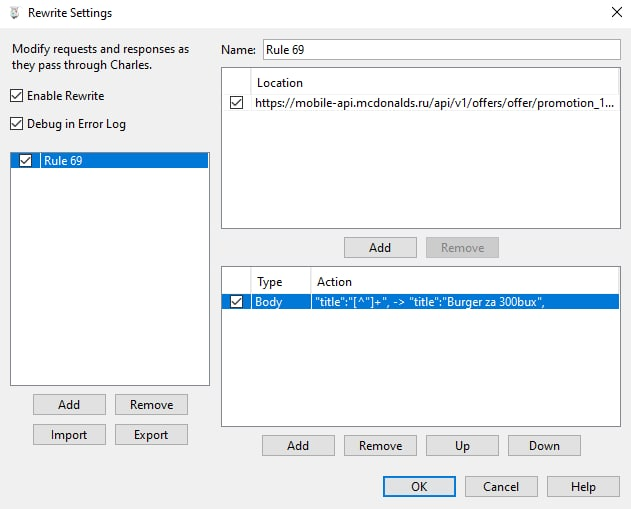


Замена данных производилась в полном соответствии со статьёй на портале Habr.

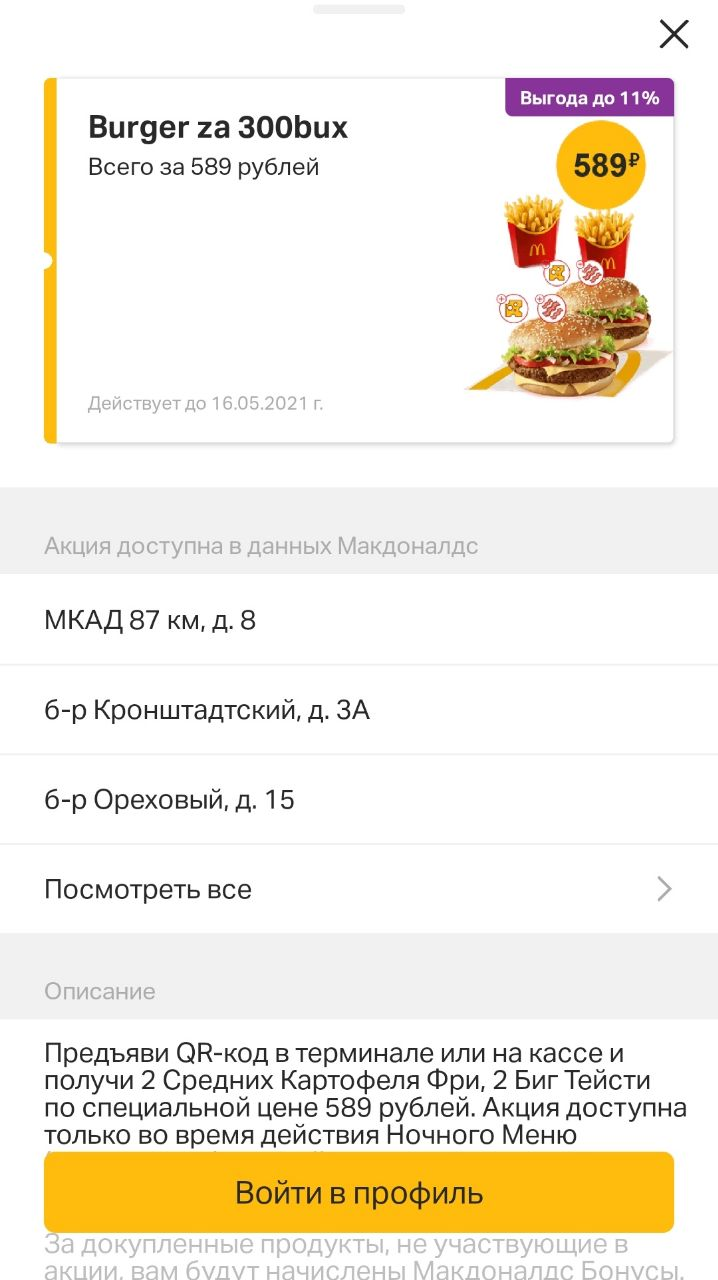
Часть 2. Rewrite.

*Индивидуальная ситуация. Использование регулярных выражений.*





Результат:



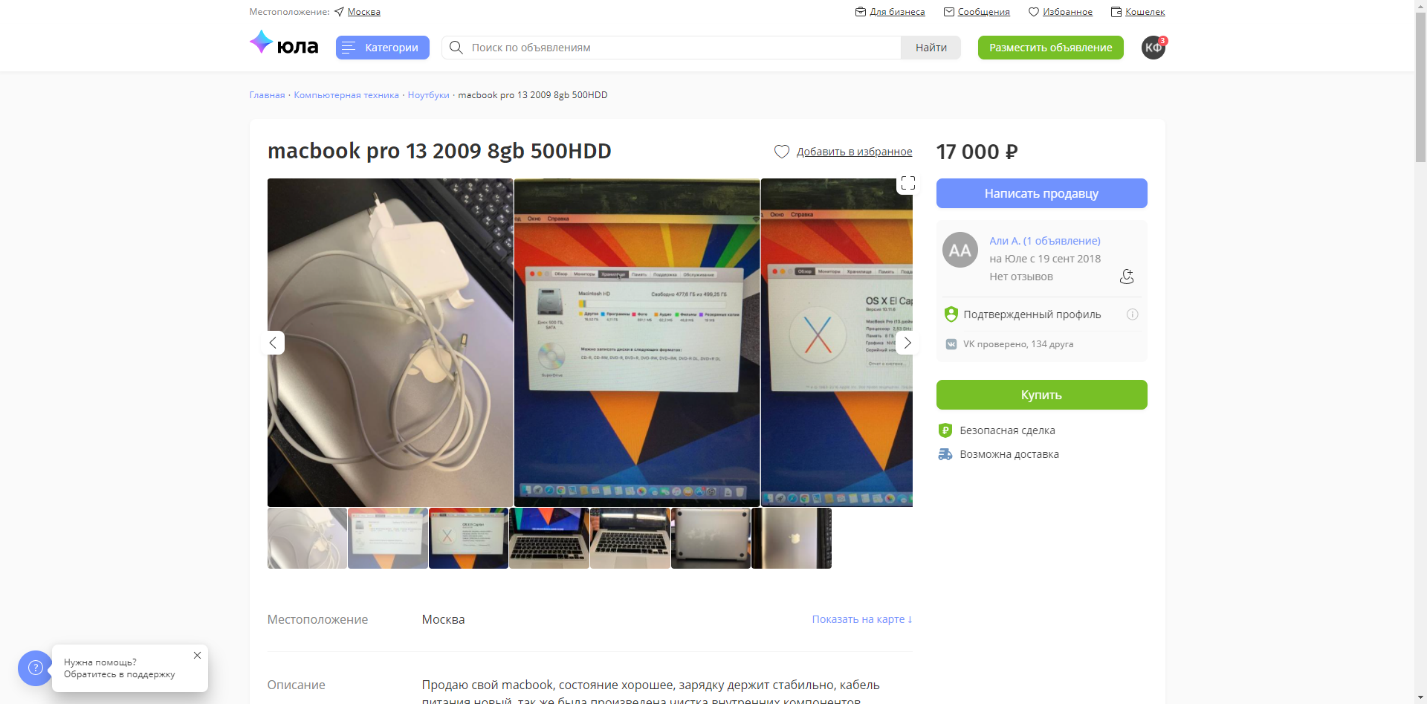
**Мини-проект отправка сообщения через Postman**

Использовано:

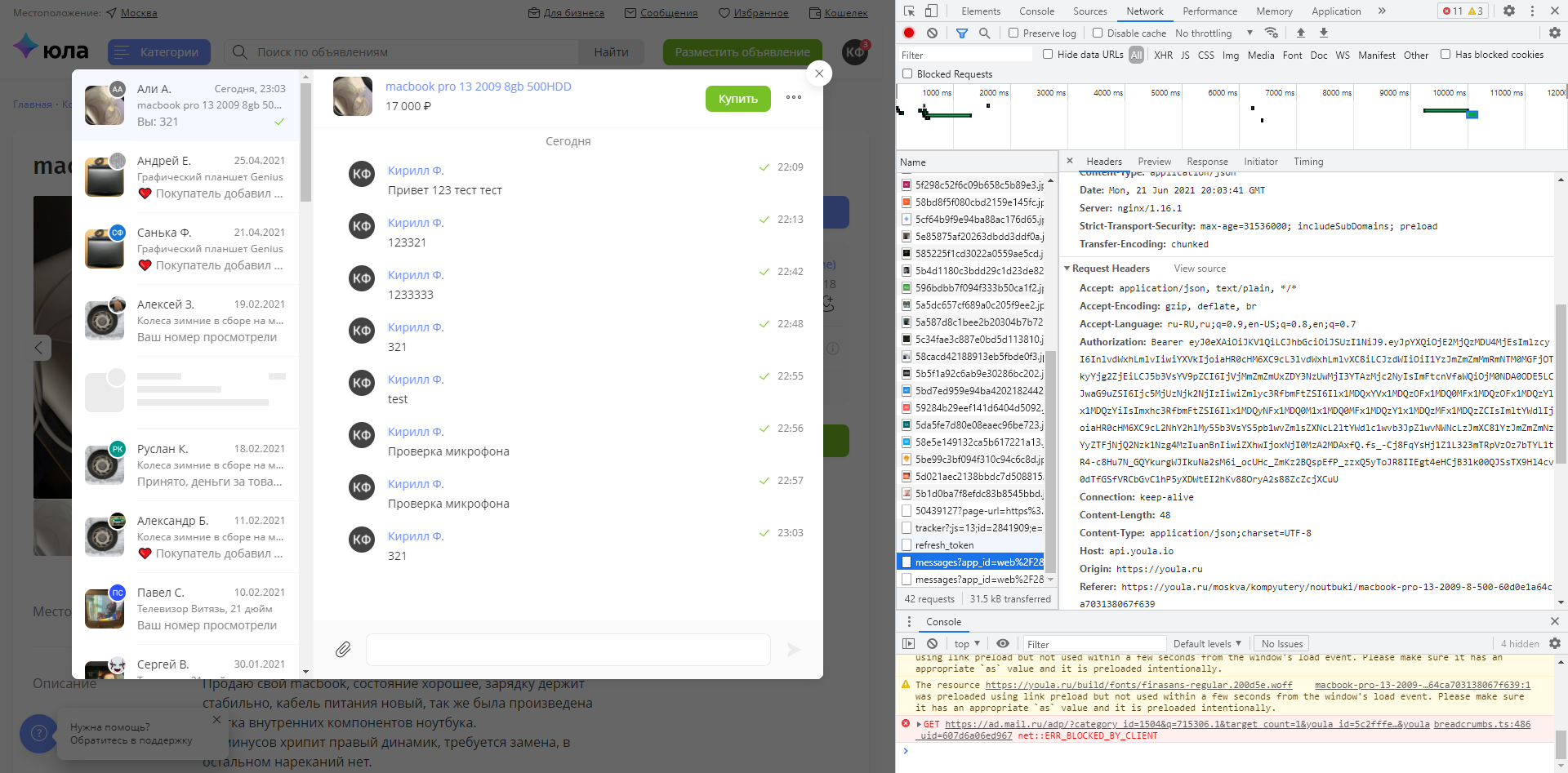
* 2 аккаунта в сервисе «Юла», 1 объявление на одном из них
* Postman
* Chrome

Выполнение:

1. Необходимо авторизоваться на двух аккаунтах сервиса «Юла».
2. На первом аккаунте необходимо создать объявление.
3. На втором аккаунте найти объявление от первого аккаунта, открыть объявление и написать сообщение. В этот момент необходимо нажать F12 и открыть вкладку “Network”.



1. Открывается чат между двумя аккаунтами. Написать тестовое сообщение. Во вкладке “Network” отображается запись, начинающаяся с “message”. Для просмотра текста сообщения необходимо выбрать вторую запись.

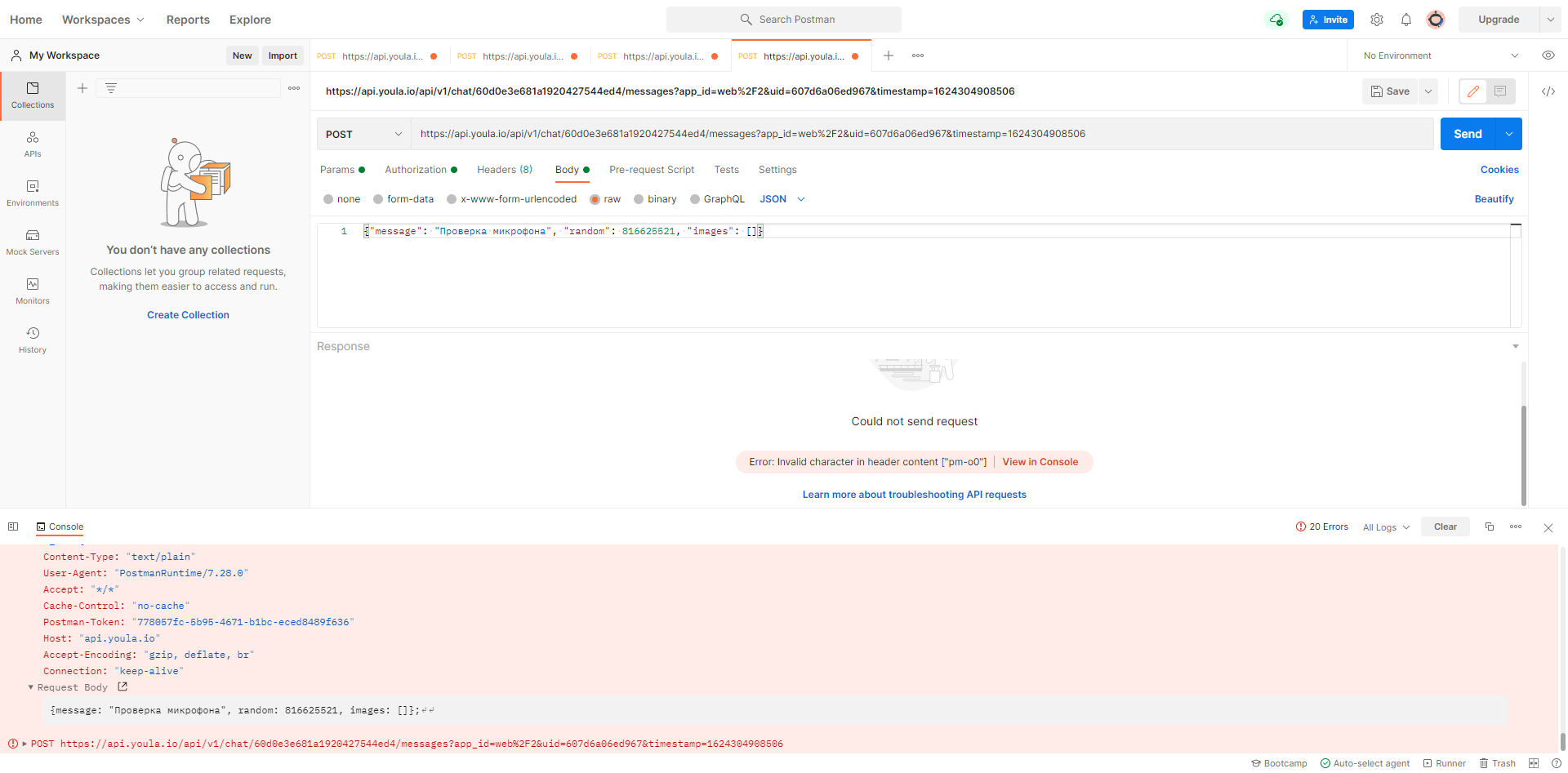


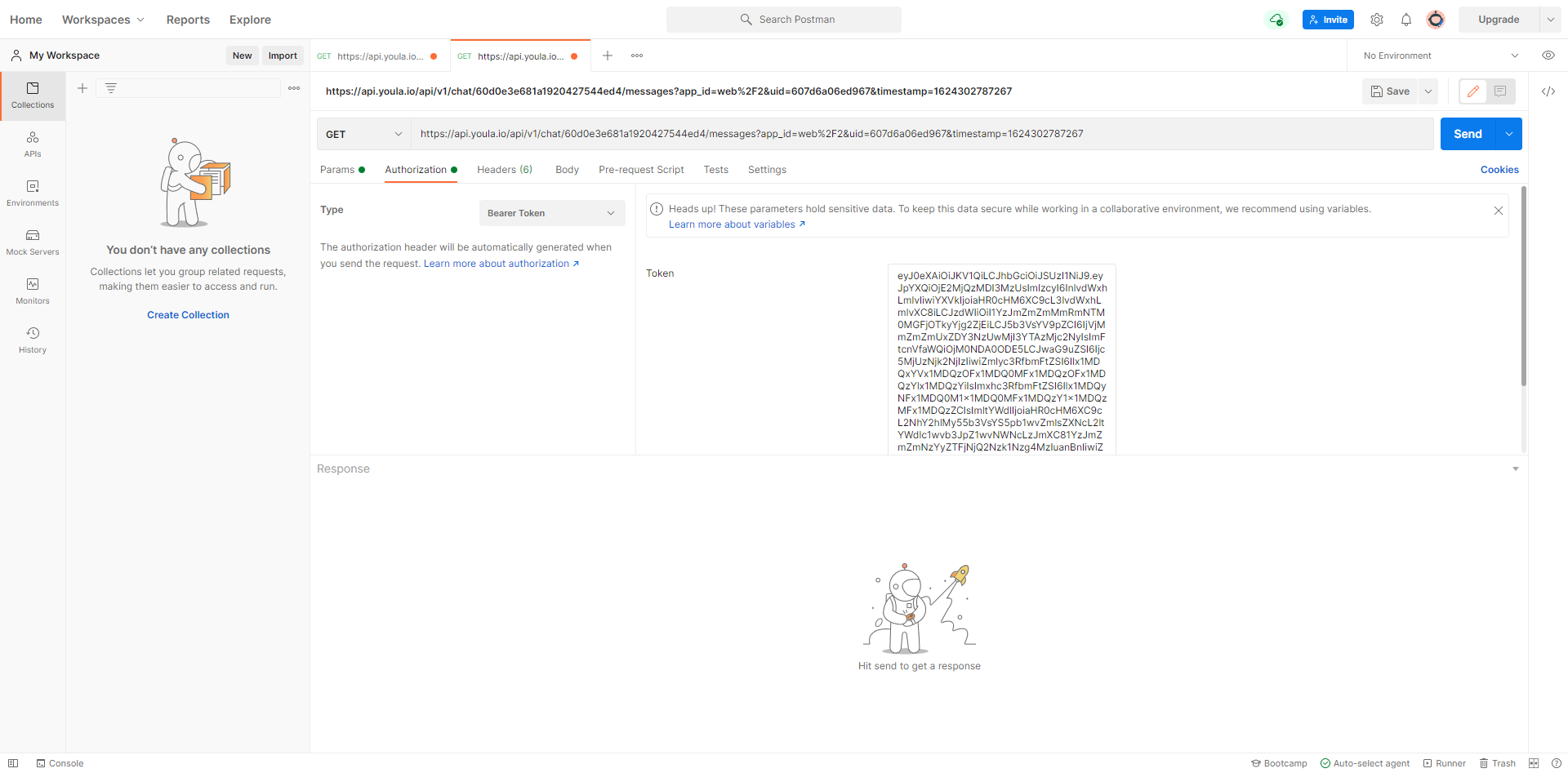
1. Во вкладке headers скопировать bearer token. С этой же вкладки скопировать ссылку в начале вкладки (эта ссылка будет использована, как action для запроса). В нижней части данной вкладки скопировать блок “data”.

Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9..fs\_-Cj8FqYsHj1Z1L323mTRpVzOz7bTYL1tR4-c8Hu7N\_GQYkurgWJIkuNa2sM6i\_ocUHc\_ZmKz2BQspEfP\_zzxQ5

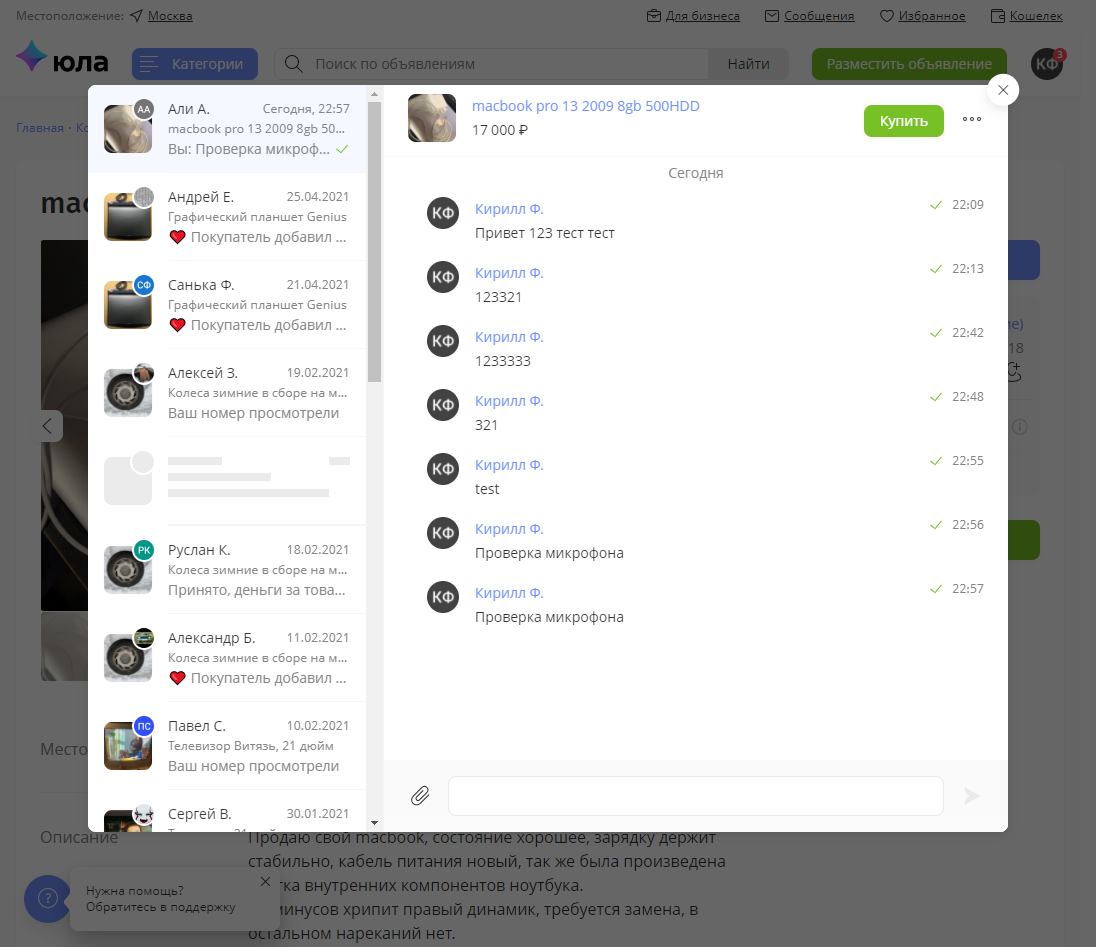
{message: "321", random: 388885842, images:[]}

1. Скопированное вставить в postman (выбрать тип POST, справа вставить ссылку, в тело запроса вставить блок “data” и изменить сообщение, во вкладке Authorization выбрать в dropdown-элементе Bearer и вставить токен в поле справа). Скриншоты ниже.





Результат:



# **РЕЗУЛЬТАТЫ**

* Чек-листы:

<https://checkvist.com/checklists/800390>

<https://checkvist.com/checklists/800391>

* Репозиторий гит:

<https://github.com/Drawishe/pd2021>

* Ссылка на сайт проекта:

<http://pd2021.std-816.ist.mospolytech.ru/>

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения работ были приобретены базовые знания о тестировании. Были изучены термины, рассмотрены инструменты, применяющиеся в данной сфере. Команда приобрела и углубила знания в области функционирования мобильных приложений на различных операционных системах. Были рассмотрены методы тестирования, способы ведения учета о дефектах.

Была проанализирована работа систем управления тестированием (менеджмента тестирования), которые используются для хранения информации, для получения информации в виде отчета о стадии тестирования и качестве тестируемого продукта. Следует отметить также высокую бизнес-ценность тестирования в современном профессиональном мире и существенную важность управления дефектами.

Выполняя практические задачи по тестированию, команда получила практические навыки тестирования мобильных приложений, составления тестовой документации.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Борис Бейзер «Тестирование черного ящика» от 04.01.2004 под редакцией издательства «Питер» https://codernet.ru/books/QA/testirovanie\_chernogo\_yashchika\_boris\_beyzer/
2. Центр подготовки IT специалистов. Дата обращение: 10.01.2021 [Ссылка](https://qalight.com.ua/baza-znaniy/ruchnoe-i-avtomatizirovannoe/#:~:text=%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BF%D1%80%D0%B5%D0%B4%D0%BF%D0%BE%D0%BB%D0%B0%D0%B3%D0%B0%)
3. Канер Сэм, Фолк Джек, Нгуен Енг Кек «Тестирование программного обеспечения» от 13.05.2001 под редакцией издательства «ДиаСофт». https://codernet.ru/books/QA/testirovanie\_programmnogo\_obespecheniya\_sem\_kaner\_2001/
4. Лиза Криспин, Джанет Грегори «Гибкое тестирование» от 12.11.2016 под редакцией издательства «Signature Series»
5. Александр Пыркин «Автоматизированное тестирование сайта за и против». Дата обращение: 10.01.2021 <https://www.uplab.ru/blog/automated-testing-of-website/>
6. Роберт Калбертсон, Крис Браун, Гэри Кобб — «Быстрое тестирование» от 16.02.2007 под редакцией издательства «Вильямс» https://codernet.ru/books/QA/bystroe\_testirovanie\_robert\_kalbertson/
7. Святослав Куликов «Тестирование программного обеспечения. Базовый курс» от 29.09.2015 под редакцией «Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International» https://fktpm.ru/file/113-svyatoslav-kulikov-testirovanie-po-bazovyi-kurs.pdf
8. Рекс Блэк «Ключевые процессы тестирования» от 17.04.2014 под редакцией издательства «Лори» https://codernet.ru/books/QA/klyuchevye\_protsessy\_testirovaniya\_reks\_blek/
9. Список обучающих лекций по работе над тестирование под редакцией «Lime Lab»