# Цель работы

Автоматизация тестирования приложений.

## **Список тест-кейсов из прошлой работы**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Описание (тип) | Предусловия | Шаги | Ожидаемый результат |
| 1.1. | Редактирование персональных данных  (негативный) | 1. Пользователь авторизован  2. Пользователь находится на странице личного кабинета | 1. Нажать кнопку «редактировать»  2. Ввести в поле ввода телефона новое значение, не соответствующее формату (например - +7999000) | 1. Рядом с полем ввода появится надпись «Введите данные в указанном формате»  2. Персональные данные не будут обновлены |
| 1.2. | Просмотр истории посещений  (позитивный) | 1. Пользователь авторизован  2. Пользователь находится на странице личного кабинета | 1. Прокрутить экран вниз, чтобы таблица истории появилась в окне.  2. Отсортировать записи по возрастанию поля «Врач»  3. Применить фильтр к любому из полей | 1. В таблице будут перечислены предыдущие приемы у врача в формате (дата-время, специализация, врач услуга, стоимость) |
| 2.1. | Просмотр информации  (негативный) | 1. Пользователь НЕ авторизован при помощи учетной записи администратора | 1. Перейти на страницу admin/patients | 1. Доступ к информации о пациентах будет запрещен  2. Появится сообщение «Необходимо авторизоваться» и форма ввода логина и пароля |
| 2.2. | Редактирование информации  (позитивный) | 1. Пользователь авторизован при помощи учетной записи администратора  2. Пользователь находится на странице admin/patients | 1. Выбрать в таблице любого пациента и нажать на соответствующую запись  2. Изменить в форме ввода фамилию на Иванова / Иванов  3. Нажать кнопку «Сохранить» | 1. Внизу страницы появится сообщение «Успех» в зеленой рамке  2. Фамилия будет изменена |
| 2.3. | Добавление нового объекта  (негативный) | 1. Пользователь авторизован при помощи учетной записи администратора  2. Пользователь находится на странице admin/patients | 1. Заполнить обязательные поля в форме ввода внизу страницы валидными данными  2. Оставить поле «Телефон» незаполненным  3. Нажать кнопку «Сохранить» | 1. Рядом с полем ввода телефона появится сообщение «Вы пропустили это поле»  2. Новая запись не будет создана |
| 2.4. | Просмотр статистики (позитивный) | 1. Пользователь авторизован при помощи учетной записи администратора | 1. Перейти на страницу admin/statistics  2. Изменить просматриваемый месяц на предыдущий, нажав стрелку влево | 1. На экране появится два графика – число посещений и общий доход клиники за прошлый месяц по дням |
| 3.1. | Выбор специализации врача (позитивный) | 1. Пользователь авторизован  2. Пользователь находится на странице make\_appointment | 1. Нажать на выпадающий список специализации врачей  2. Выбрать специализацию (например, «аллергология») и нажать на выбранный элемент | 1. В поле «специализация» появится выбранное значение  2. Поля «услуга» и «врач» станут доступными для ввода |
| 3.2. | Выбор конкретного врача  (негативный) | 1. Пользователь авторизован  2. Пользователь находится на странице make\_appointment | 1. Нажать на поле выбора врача | 1. Ничего не произойдет. Поле неактивно, так как не выбрана специализация |
| 3.3. | Создание записи  (позитивный) | 1. Пользователь авторизован  2. Пользователь находится на странице make\_appointment | 1. Выбрать из выпадающего списка специализацию (например, «аллергология»)  2. Выбрать услугу («ингаляционные пробы»)  3. Выбрать врача («Беляев Корнилий Анатольевич»)  4. Выбрать дату и время (текущие)  5. Нажать кнопку «сохранить» | 1. На экране появится сообщение «Запись создана», имя врача и время записи |
| 4.1. | Получение информации о клинике  (позитивный) |  | 1. Перейти на главную страницу | 1. На экране будет отображена общая информация о клинике, блоки «Наши преимущества» и «Технологии и оборудование» |
| 4.2. | Получение информации о врачах и услугах  (позитивный) |  | 1. Перейти на страницу /services | 1. На экране появится заголовок «Наши услуги» и список услуг, сгруппированных по врачебным специализациям |
| 4.3. | Получение контактов (позитивный) |  | 1. Перейти на страницу /contacts | 1. На экране появится заголовок «Контакты», блок информации о клинике и карта Москвы с геометкой |

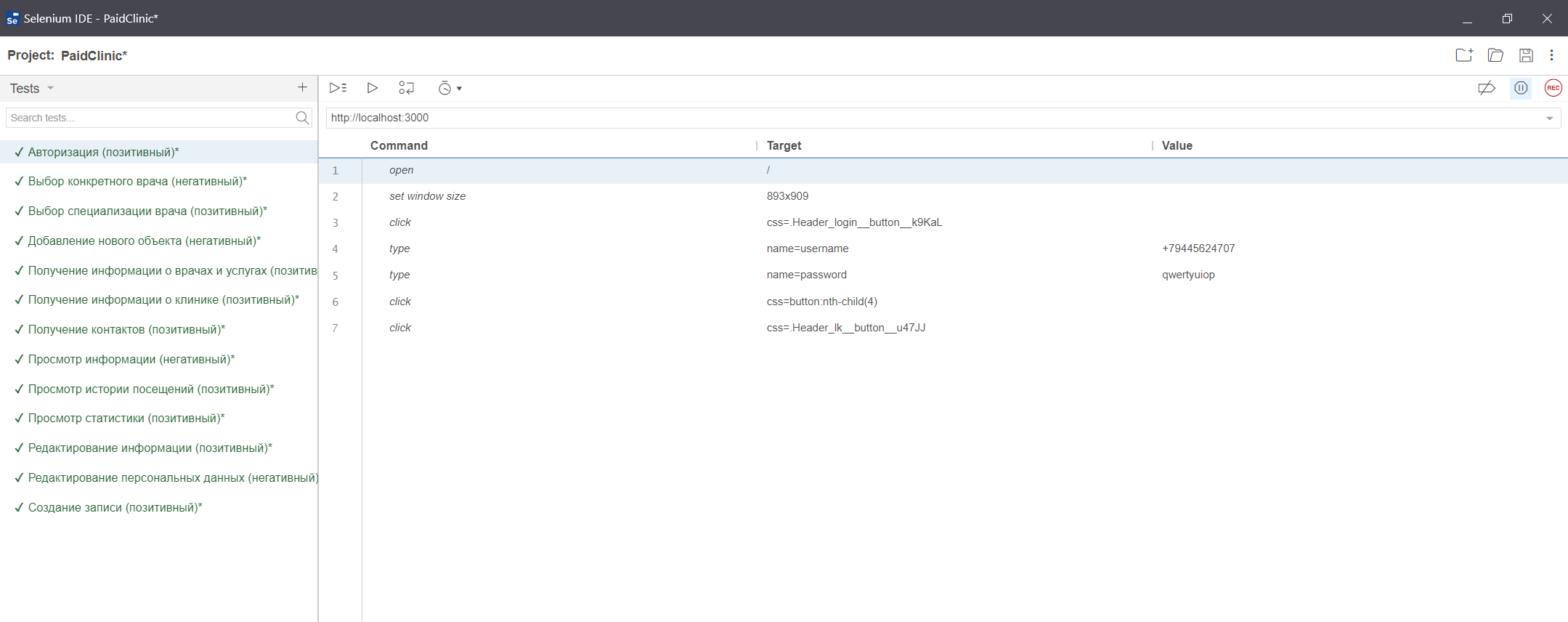
# Описание реализованных автотестов

Работа выполнена на примере веб-приложения «Платная клиника» - моего курсового проекта.

Все автотесты сделаны при помощи Selenium IDE.

В программе был создан автотест под каждый отдельный тест-кейс.

Программа прошла все поставленные тесты.



Selenium предоставляет интерактивную среду для создания автоматических тестов визуального интерфейса и сам конвертирует их в необходимый код на любом языке

Пример кода теста авторизации на Java (экспортировано средствами Selenium):

// Generated by Selenium IDE

import org.junit.Test;

import org.junit.Before;

import org.junit.After;

import static org.junit.Assert.\*;

import static org.hamcrest.CoreMatchers.is;

import static org.hamcrest.core.IsNot.not;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.remote.RemoteWebDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.Dimension;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.Alert;

import org.openqa.selenium.Keys;

import java.util.\*;

import java.net.MalformedURLException;

import java.net.URL;

public class Test {

private WebDriver driver;

private Map<String, Object> vars;

JavascriptExecutor js;

@Before

public void setUp() {

driver = new ChromeDriver();

js = (JavascriptExecutor) driver;

vars = new HashMap<String, Object>();

}

@After

public void tearDown() {

driver.quit();

}

@Test

public void () {

driver.get("http://localhost:3000/");

driver.manage().window().setSize(new Dimension(893, 909));

driver.findElement(By.cssSelector(".Header\_login\_\_button\_\_k9KaL")).click();

driver.findElement(By.name("username")).sendKeys("+79445624707");

driver.findElement(By.name("password")).sendKeys("qwertyuiop");

driver.findElement(By.cssSelector("button:nth-child(4)")).click();

driver.findElement(By.cssSelector(".Header\_lk\_\_button\_\_u47JJ")).click();

}

}

Весь проект Selenium экспортирует в xml файл. Вот все тесты:



Их можно открыть при помощи любого текстового редактора.

# Выводы по работе

В ходе работы были разработаны автотесты с использованием Selenium IDE, тесты успешно пройдены. Интерактивная среда создания тестов позволяет удобно проводить тестирование веб-интерфейса, сокращает время на их разработку и позволяет сделать тесты универсальными.

В ходе запуска тестов не было выявлено никаких дефектов визуального интерфейса.

**Список использованных источников**

* Мой проект с GitHub: <https://github.com/Redegit/Coursework-PaidClinic>
* Официальный сайт Selenium