

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ

Лабораторная работа №3
по дисциплине
«Тестирование программного обеспечения»
Вариант 41

Выполнил:
Векшин Арсений Иванович Р3316

Преподаватель:
Ермаков Михаил Константинович

Текст задания	3
Ссылка на репозиторий	3
Use-case диаграмма	4
Описание тестовых сценариев	5
0. Авторизация	5
1. Создание проекта (обычный)	5
2. Создание проекта (шаблонный)	5
3. Удаление проекта	5
4. Редактирование файла проекта	6
5. Управление пайплайнами	6
Вывод	7

Текст задания

Привечание: Вариант изменен на gitlab.com

Лабораторная работа #3

Сформировать варианты использования, разработать на их основе тестовое покрытие и провести функциональное тестирование интерфейса сайта (в соответствии с вариантом).

Вариант 41

Вариант №41: lamoda.ru. Интернет-магазин одежды и обуви. - <http://www.lamoda.ru/>

Требования к выполнению работы:

1. Тестовое покрытие должно быть сформировано на основании набора прецедентов использования сайта.
2. Тестирование должно осуществляться автоматически - с помощью системы автоматизированного тестирования [Selenium](#).
3. Шаблоны тестов должны формироваться при помощи Selenium IDE и исполняться при помощи Selenium RC в браузерах Firefox и Chrome.
4. Предполагается, что тестируемый сайт использует динамическую генерацию элементов на странице, т.е. выбор элемента в DOM должен осуществляться не на основании его ID, а с помощью [XPath](#).

Требования к содержанию отчёта:

1. Текст задания.
2. UseCase-диаграмму с прецедентами использования тестируемого сайта.
3. CheckList тестового покрытия.
4. Описание набора тестовых сценариев.
5. Результаты тестирования.
6. Выводы.

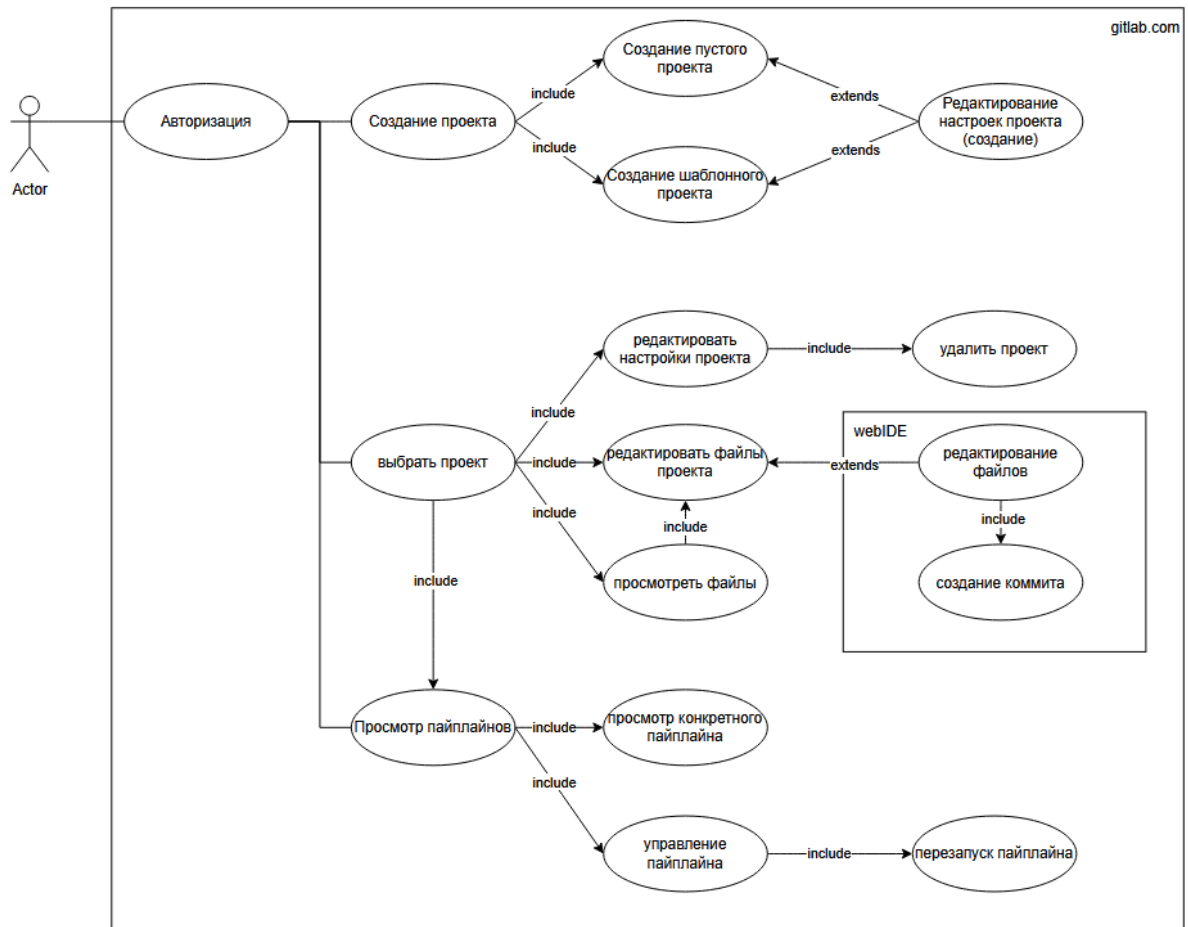
Вопросы к защите лабораторной работы:

1. Функциональное тестирование. Основные понятия, способы организации и решаемые задачи.
2. Система Selenium. Архитектура, принципы написания сценариев, способы доступа к элементам страницы.
3. Язык XPath. Основные конструкции, системные функции, работа с множествами элементов.

Ссылка на репозиторий

<https://github.com/ArsenyVekshin/ITMO/tree/master/Testing/lab3>

Use-case диаграмма



Описание тестовых сценариев

0. Авторизация

Предусловие: система функционирует

Шаги:

1. Открыть страницу авторизации
2. Ввод логина и пароля, установка флага remember-me
3. Пользователь нажимает кнопку "Войти"

Ожидаемый результат: переадресация на страницу проектов пользователя

1. Создание проекта (обычный)

Предусловие: пользователь авторизован, система функционирует

Шаги:

1. Открыть страницу "New project"
2. Пользователь нажимает кнопку "Create blank project"
3. Переадресация на страницу заполнения параметров проекта
4. Пользователь заполняет поля проекта (name, slug, visibility, readme)
5. Пользователь нажимает кнопку "create project"
6. Переадресация на страницу проекта.

Ожидаемый результат: проект создан

2. Создание проекта (шаблонный)

Предусловие: пользователь авторизован, система функционирует

Шаги:

1. Открыть страницу "New project"
2. Пользователь нажимает кнопку "Create template project"
3. Переадресация на страницу выбора шаблонов
4. Пользователь выбирает шаблон и нажимает кнопку "choose" выбранного шаблона
5. Переадресация на страницу заполнения параметров проекта
6. Пользователь заполняет поля проекта (name, slug, visibility, readme)
7. Пользователь нажимает кнопку "create project"
8. Переадресация на страницу проекта.

Ожидаемый результат: проект создан

3. Удаление проекта

Предусловие: пользователь авторизован, система функционирует, проект существует

Шаги:

1. Открыть страницу проекта
2. Пользователь нажимает кнопку "Settings"
3. Переадресация на страницу настроек проекта

4. Пользователь нажимает кнопку “expand” в разделе “Advanced settings”
5. Пользователь нажимает кнопку “delete project”
6. Открывается диалоговое окно удаления проекта
7. Пользователь вводит slug в контрольное поле
8. Пользователь нажимает кнопку “Yes, delete project”

Ожидаемый результат: проект удален

4. Редактирование файла проекта

Предусловие: пользователь авторизован, система функционирует, проект существует

Шаги:

1. Открыть страницу проекта
2. Пользователь нажимает кнопку “WebIDE”
3. Переадресация на страницу WebIDE
4. Пользователь выбирает файл в списке
5. Пользователь нажимает на выбранный файл
6. Файл открывается в редакторе
7. Пользователь вносит правки в файл
8. Пользователь нажимает кнопку “Commit”
9. Открывается панель создания коммита
10. Пользователь нажимает кнопку “commit to master branch”
11. Пользователь вводит имя коммита в поле “Commit message”
12. Пользователь нажимает кнопку “Commit”

Ожидаемый результат: Новый коммит создан, пайплайн автоматически запущен

5. Управление пайплайнами

Предусловие: пользователь авторизован, система функционирует, проект существует

Шаги:

1. Открыть страницу проекта
2. Пользователь нажимает кнопку “CI/CD”
3. Переадресация на страницу управления пайплайнами
4. Пользователь выбирает пайплайн в списке
5. Пользователь нажимает на кнопку “retry” выбранного пайплайна
6. Система перезапускает пайплайн

Ожидаемый результат: Выбранный пайплайн изменил параметры (контроль: время запуска)

Вывод

