

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ В.Ф. УТКИНА»

Факультет вычислительной техники  
Кафедра электронных вычислительных машин

К защите  
Руководитель проекта

---

дата, подпись

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К КУРСОВОМУ ПРОЕКТУ**

по дисциплине

«Администрирование информационных систем»

по теме

«Разработка автоматизированной системы для Web-приложения «Деканат в  
ВУЗе»

Выполнил:

студент группы 645

Гуськов П.А.

---

дата сдачи на проверку,  
подпись

Руководитель проекта:

Доцент кафедры ЭВМ

Свирина А.Г.

---

оценка

---

дата защиты, подпись

Рязань 2019 г



# СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	3
ВВЕДЕНИЕ .....	4
1 Обзор средств и методов автоматизации .....	5
2 Методологии автоматического тестирования .....	6
3 Практическая реализация проекта .....	8
3.1 Разработка прототипа веб-приложения.....	8
3.2 Разработка стратегии тестирования приложения.....	8
3.2.1 Типы и виды тестирования.....	8
3.2.2 Критерии качества.....	9
3.2.3 Инструменты тестирования.....	9
3.3 Тест-кейсы.....	9
3.4 Выбор тест-кейсов для автоматизации.....	18
3.5 Разработка автотестов по тест-кейсам .....	19
4 Отчет о тестировании.....	20
ЗАКЛЮЧЕНИЕ.....	21
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	22
ПРИЛОЖЕНИЕ 1 .....	23

## **ВВЕДЕНИЕ**

Целью данного курсового проекта является разработка автоматизированной системы для Web-приложения «Деканат в ВУЗе». В автоматизированную систему тестирования входит выбор стратегии тестирования, разработка критериев качества, тест-кейсов и автоматизация части тестовых вариантов.

Под автоматизацией подразумевается самостоятельное выполнение компьютером тестовых шагов с помощью некоторого инструмента автоматического тестирования, в нашем случае в качестве такого инструмента будет выступать Selenium, необходимый для автоматизации действий веб-браузера.

# 1 ОБЗОР СРЕДСТВ И МЕТОДОВ АВТОМАТИЗАЦИИ

Современное ПО является сложным многофункциональным объектом, и его проверка вручную может занимать достаточно большой отрезок времени. В связи с этим возникла потребность в возникновении средств для автоматизации процессов.

Средства автоматизации – специально разработанные приборы и устройства, функционирующие как в составе аппаратно-программного комплекса, так и автономно, выполняя ряд поставленных задач без вмешательства человека. В качестве средств автоматизации также можно рассматривать отдельные программные продукты, способные выполнять определенные действия над большими объемами данных в соответствие с поставленной задачей внутри какой-либо ЭВМ.

Определяющий фактор для успешного применения автоматизации тестирования программного обеспечения – выбор и использование правильного набора средств автоматизации тестирования. Это сложная задача, так как нет инструмента, который бы соответствовал всем требованиям автоматизированного тестирования. Это затрудняет поиск подходящего решения.

В курсовом проекте в качестве инструмента для автоматизации действий веб-браузера применяется система Selenium.

**Selenium** – это самая популярная система автоматизации, которая состоит из множества инструментов и плагинов для тестирования веб-приложений. Selenium известен возможностью тестирования производительности веб-приложений. Selenium - популярный выбор в области автоматизации тестирования с открытым исходным кодом, благодаря его активной разработке и большому сообществу пользователей.

## 2 МЕТОДОЛОГИИ АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ

В настоящее время существует большое количество типов и методов тестирования, которые применяются в зависимости от требований и условий при разработке стратегий тестирования.

Автоматизация тестов является одним из методов сокращения затрачиваемых усилий на проведение тестирования программного продукта, но при возникновении потребности в автоматизации руководствуются правилом – «Автоматизировать тесты нужно только в том случае, если затраты времени и ресурсов на автоматизацию не превысят затраты на ручное тестирование функционала данного приложения».

Указанное правило соблюдается в одном из направлений тестирования – регрессионном тестировании. Данный вид тестирования подразумевает проверку уже созданного функционала при добавлении в программный продукт новых возможностей. Это обусловлено тем, что необходимо проверять работоспособность старого функционала приложения при добавлении нового. Количество таких проверок достаточно велико, поэтому для экономии времени все связанные с этим тесты автоматизируют.

Также автоматизация тестов применяется при тестировании интерфейса пользователя, в частности графического. Такой вид тестирования получил название автоматического **GUI – тестирования**. Основная идея данного подхода состоит в проверке корректного функционирования интерфейса приложения. Автоматизация при данном подходе применяется в случае наличия сложных конструкций по выбору разнообразных вариантов деятельности. Для тестирования такого интерфейса потребуется большое количество времени, поэтому здесь к нам приходят на помощь различные инструменты для автоматизации выполняемых действий.

Иногда для проверки надежной работы приложения требуется создать для него существенную нагрузку, выраженную большим количеством пользователей, одновременно использующих данный программный продукт.

Для достижения таких требований необходимо большая группа людей, которая будет создавать требуемую нагрузку, но это экономически не выгодно и затратно по времени. Применительно к этому случаю используется вид тестирования, который называется **нагрузочным**. С помощью специальных программ достигается требуемая нагрузка на тестируемое приложение. Данный вид тоже относится к автоматизированному тестированию.

## **3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРОЕКТА**

### **3.1 Разработка прототипа веб-приложения**

Web-приложение «Деканат в ВУЗе» содержит 2 страницы с переходами между собой, а именно:

- Новости – содержит новостную ленту факультета, оставленные пользователями комментарии, а также форму для создания комментария;
- О факультете – содержит информацию кафедрах факультета.

Код приложения представлен в репозитории по ссылке:  
[https://github.com/nu17god/AIS\\_KP\\_GUSKOV\\_645](https://github.com/nu17god/AIS_KP_GUSKOV_645)

### **3.2 Разработка стратегии тестирования приложения**

#### **3.2.1 Типы и виды тестирования**

Для тестирования Web-приложения «Деканат в ВУЗе» применялись следующие подходы к функциональному тестированию:

1) Компонентное тестирование:

Проверка функционала отдельных элементов приложения с помощью набора пользовательских тестов.

2) Smoke-тесты:

Smoke-тесты представляют собой тесты работоспособности приложения, например, откроется ли оно в браузере.

3) Тесты критического пути:

Тесты критического пути подразумевают имитацию действий пользователя, например, переход по ссылке на другую страницу и заполнение формы на ней.

4) Расширенные тесты:



Расширенные тесты являются более обширными тестами критического пути и представляют собой длинную последовательность действий пользователя на всех страницах открытого приложения.

5) Автоматизированные и ручные тесты:

Автоматизированные тесты применяются тогда, когда нужно выполнить достаточно большую последовательность действий, используя при этом множество вариаций выбора в различных ситуациях.

6) Позитивные и негативные тесты:

Негативные тесты представляют собой проверку выполнения каких-либо ограничений на работу элементов приложения. Результатом такого теста становится не выполнение определенного действия над элементом страницы или появление ошибки, что и ожидается при заданных условиях.

### **3.2.2 Критерии качества**

Для корректной работы приложения необходимо, чтоб результат всех тестовых вариантов с приоритетом «А» соответствовал ожидаемому.

### **3.2.3 Инструменты тестирования**

В курсовом проекте для тестирования приложения используются:

- Среда разработки: Microsoft Visual Studio 2019;
- Язык программирования: C#
- Инструмент для автоматизации действий веб-браузера: Selenium;
- Фреймворк юнит тестов: N-Unit.

## **3.3 Тест-кейсы**

**Тест-кейс** (test case) – набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства.

Для тестирования приложения разработаны следующие тестовые варианты:

1) Модуль «Меню навигации»:

Таблица 3.1 – Модуль «Меню навигации»

Идентификатор	Приоритет	Подмодуль	Действия	Ожидаемый результат
1.1.1	А	Главное Меню	<b>Проверка переходов между страницами</b>	
			1) Находясь на странице "Новости" щелкнуть левой кнопкой мыши на ссылку "Новости"	Страница "Новости" обновится
			2) Щелкнуть левой кнопкой мыши на ссылку "О факультете" в главном меню страницы	Откроется страница "О факультете"
			3) Щелкнуть левой кнопкой мыши на ссылку "О факультете" в главном меню страницы	Страница "О факультете" обновится
			4) Щелкнуть левой кнопкой мыши на ссылку "Новости" в главном меню страницы	Откроется страница "Новости"
1.1.2	А	Главное Меню	<b>Проверка выпадающего меню пункта "О факультете"</b>	
			1) Подвести курсор к пункту меню "О факультете".	Появится выпадающее меню
			2) Увести курсор с пункта меню "О факультете"	Выпадающее меню пропадет
1.2.1	Б	Подменю	<b>Проверка ссылок на странице "О факультете"</b>	
			Щелкнуть в выпадающем списке на элемент	Откроется страница "О факультете" на выбранном заголовке.

2) Модуль «Поля формы»:

Таблица 3.2 – Модуль «Поля формы»

Идентификатор	Приоритет	Подмодуль	Действия	Ожидаемый результат
2.1.1	А	Поле "Фамилия"	<b>Проверка Поля "Фамилия"</b>	
			1) Щелкнуть левой кнопкой мыши по полю "Фамилия"	1) В поле "Фамилия" появится курсор
			2) Ввести любой текст в поле "Фамилия"	2) В поле "Фамилия" появится введенный текст
2.2.1	А	Поле "Имя"	<b>Проверка Поля "Имя"</b>	
			1) Щелкнуть левой кнопкой мыши по полю "Имя"	1) В поле "Имя" появится курсор
			2) Ввести любой текст в поле "Имя"	2) В поле "Имя" появится введенный текст
2.3.1	Б	Поле "Отчество"	<b>Проверка Поля "Отчество"</b>	
			1) Щелкнуть левой кнопкой мыши по полю "Отчество"	1) В поле "Отчество" появится курсор
			2) Ввести любой текст в поле "Отчество"	2) В поле "Отчество" появится введенный текст

Идентификатор	Приоритет	Подмодуль	Действия	Ожидаемый результат
2.4.1	А	Поле "Email"	<b>Проверка Поля "Email"</b>	
			1) Щелкнуть левой кнопкой мыши по полю "Email"	1) В поле "Email" появится курсор
			2) Ввести любой текст в поле "Email"	2) В поле "Email" появится введенный текст и граница поля изменит цвет на красный
			3) Ввести в поле "Email" текст в виде "<Текст на английском>@<Текст на английском>"	3) В поле "Email" появится введенный текст
2.6.1	Б	Поле "Дата рождения"	<b>Проверка поля "Дата Рождения"</b>	
			1) Щелкнуть левой кнопкой мыши по полю	1) Появится меню выбора даты
			2) Выбрать дату 01.01.1900	2) В поле отобразится дата 01.01.1900
			3) Выбрать дату 31.12.2019	3) В поле отобразится дата 31.12.2019
			4) Нажать на кнопку "Clear" в этом поле	4) Поле очистится

Идентификатор	Приоритет	Подмодуль	Действия	Ожидаемый результат
2.7.1	Б	Радио-Кнопки "Пол"	<b>Проверка радио-кнопок "Пол"</b>	
			1) Щелкнуть левой кнопкой мыши по значению "Мужской"	1) Значение "Мужской" станет активным
			2) Щелкнуть левой кнопкой мыши по значению "Женский"	2) Значение "Женский" станет активным
2.8.1	Б	Поле "Комментарий"	<b>Проверка Поля "Комментарий"</b>	
			1) Щелкнуть левой кнопкой мыши по полю "Комментарий"	1) В поле "Комментарий" появится курсор
			2) Ввести любой текст в поле "Комментарий"	2) В поле "Комментарий" появится введенный текст

Идентификатор	Приоритет	Подмодуль	Действия	Ожидаемый результат
2.5.1	А	Выпадающий список "Должность"	<b>Проверка выпадающего списка "Должность"</b>	
			1) Щелкнуть левой кнопкой мыши по списку	1) Появится список должностей
			2) Щелкнуть левой кнопкой мыши по элементу "Абитуриент"	2) В поле "Должность" отобразится "Абитуриент"
			3) Щелкнуть левой кнопкой мыши по элементу "Студент"	3) В поле "Должность" отобразится "Студент"
			4) Щелкнуть левой кнопкой мыши по элементу "Магистрант"	4) В поле "Должность" отобразится "Магистрант"
			5) Щелкнуть левой кнопкой мыши по элементу "Аспирант"	5) В поле "Должность" отобразится "Аспирант"
			6) Щелкнуть левой кнопкой мыши по элементу "Сотрудник"	6) В поле "Должность" отобразится "Сотрудник"
			7) Щелкнуть левой кнопкой мыши по элементу "Выпускник"	7) В поле "Должность" отобразится "Выпускник"

3) Модуль «Кнопки формы»:

Таблица 3.3 – Модуль «Кнопки формы»

Идентификатор	Приоритет	Подмодуль	Действия	Ожидаемый результат
3.1.1	А	Кнопка "Отправить"	<b>Проверка кнопки "Отправить"</b>	
			1) Нажать на кнопку "Отправить"	Форма потребует заполнить обязательные поля
3.1.2	А	Кнопка "Отправить"	<b>Проверка обязательных полей формы</b>	
			1) Заполнить поле Фамилия	1) В поле Фамилия появится введенный текст
			2) Заполнить поле Имя	2) В поле Имя появится введенный текст
			3) Заполнить поле Email	3) В поле Email появится введенный текст
			4) Подтвердить обработку персональных данных	4) Чек-бокс "license" перейдет в состояние "Checked"
			5) Нажать кнопку "Отправить"	5) Страница обновится

Идентификатор	Приоритет	Подмодуль	Действия	Ожидаемый результат
3.1.3	Б	Кнопка "Отправить"	<b>Проверка всех полей формы</b>	
			1) Заполнить поле Фамилия	1) В поле Фамилия появится введенный текст
			2) Заполнить поле Имя	2) В поле Имя появится введенный текст
			3) Заполнить поле Email	3) В поле Email появится введенный текст
			4) Заполнить поле Отчество	4) В поле Отчество появится введенный текст
			5) В поле дата рождения выбрать дату	5) В поле дата рождения появится выбранная дата
			6) Ввести текст в поле Комментарий	6) В поле Комментарий появится введенный текст
			7) Подтвердить обработку персональных данных	7) Чек-бокс "license" перейдет в состояние "Checked"
			8) Подтвердить email рассылку	8) Чек-бокс "spam" перейдет в состояние "Checked"
			9) Нажать кнопку "Отправить"	9) Страница обновится



Идентификатор	Приоритет	Подмодуль	Действия	Ожидаемый результат
3.2.1	А	Кнопка "Очистить"	<b>Проверка кнопки "Очистить"</b>	
			1) Нажать на кнопку "Очистить"	1) Кнопка нажмется и ничего не произойдет
3.2.2	А	Кнопка "Очистить"	<b>Проверка обязательных полей формы</b>	
			1) Заполнить поле Фамилия	1) В поле Фамилия появится введенный текст
			2) Заполнить поле Имя	2) В поле Имя появится введенный текст
			3) Заполнить поле Email	3) В поле Email появится введенный текст
			4) Подтвердить обработку персональных данных	4) Чек-бокс "license" перейдет в состояние "Checked"
			5) Нажать кнопку "Очистить"	5) Поля формы очистятся и чек-боксы перейдут в состояние "Unchecked"

Идентификатор	Приоритет	Подмодуль	Действия	Ожидаемый результат
3.2.3	Б	Кнопка "Очистить"	<b>Проверка всех полей формы</b>	
			1) Заполнить поле Фамилия	1) В поле Фамилия появится введенный текст
			2) Заполнить поле Имя	2) В поле Имя появится введенный текст
			3) Заполнить поле Email	3) В поле Email появится введенный текст
			4) Заполнить поле Отчество	4) В поле Отчество появится введенный текст
			5) В поле дата рождения выбрать дату	5) В поле дата рождения появится выбранная дата
			6) Ввести текст в поле Комментарий	6) В поле Комментарий появится введенный текст
			7) Подтвердить обработку персональных данных	7) Чек-бокс "license" перейдет в состояние "Checked"
			8) Подтвердить email рассылку	8) Чек-бокс "spam" перейдет в состояние "Checked"
			9) Нажать кнопку "Очистить"	9) Страница обновится

### 3.4 Выбор тест-кейсов для автоматизации

В качестве тест-кейсов для автоматизации были выбраны тест-кейсы с Идентификаторами 1.1.1, 2.1.1, 2.2.1, 2.4.1, 3.1.2 и 3.2.2. Данный выбор был сделан на основе затрат времени на выполнение данных тестов вручную.

### **3.5 Разработка автоматизированных тестов по тестовым-вариантам**

По выбранным тест-кейсам были разработаны автотесты, которые находятся в проекте TestKP. Автотесты были разработаны согласно требованиям, представленными в задании.

Код автотестов и веб-приложения размещен в репозитории по ссылке: [https://github.com/nu17god/AIS\\_KP\\_GUSKOV\\_645](https://github.com/nu17god/AIS_KP_GUSKOV_645), а также код автотестов представлен в приложении 1.

## **4 ОТЧЕТ О ТЕСТИРОВАНИИ**

Приложение было протестировано в браузерах Google Chrome 74.0 и Mozilla Firefox 67.0.

Приложение в ходе тестирования прошло полный набор тестовых вариантов. Часть тестов была выполнена автоматически.

При выполнении автоматизированных тестов, ошибок обнаружено не было. На последней версии сборки все автоматизированные тесты прошли успешно, в ходе проверок дефектов не обнаружено: приложение работает стабильно, основная функциональность работоспособна.

Всего было проведено более 20-25 проверок.

## **ЗАКЛЮЧЕНИЕ**

В ходе курсового проекта была разработана автоматизированная система для Web-приложения «Деканат в ВУЗе», разработана стратегия тестирования, составлены тестовые варианты и критерии качества, а также выполнено тестирование приложения по составленным тестовым-вариантам.

Были получены практические навыки работы с инструментом Selenium, а также написания тестов на языке C# с помощью фреймворка NUnit 3.0.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1) Албахари Джозеф , Албахари Бен С# 6.0. Справочник. Полное описание языка; Вильямс - М., 2016. - **623** с.
- 2) Свирина А.Г.: Конспект лекций по дисциплине «Администрирование информационных систем»; РГРТУ.
- 3) Шилдт, Герберт С# 4.0. Полное руководство / Герберт Шилдт. – М.: Вильямс, 2015. - **670** с.
- 4) Stack Overflow // [Электронный ресурс]. – URL:<http://stackoverflow.com>
- 5) Habr // [Электронный ресурс]. – URL:<https://habr.com/en/>.

# ПРИЛОЖЕНИЕ 1

## Листинг разработанных автотестов.

### TestNavigation.cs

```
using NUnit.Framework;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Firefox;

namespace TestKP
{
    [SetUpFixture]
    [TestFixture(typeof(FirefoxDriver))]
    [TestFixture(typeof(ChromeDriver))]
    public class TestNavigation<TWebDriver> where TWebDriver
: IWebDriver, new()
    {
        private NewsPage newsPage;
        private AboutPage aboutPage;
        private IWebDriver driver;

        [OneTimeSetUp]
        public void TestSetUp()
        {
            this.driver = new TWebDriver();
            driver.Url = "http://127.0.0.1:8080/";
            newsPage = new NewsPage(driver);
        }

        [Test, Order(1)]
        public void TestOpenNewsPageFromNewsPage()
        {
            newsPage = newsPage.OpenNewsPage();
            bool condition = newsPage.MainBlock.Displayed;

            Assert.True(condition);
        }

        [Test, Order(2)]
        public void TestOpenAboutPageFromNewsPage()
        {
            aboutPage = newsPage.OpenAboutPage();
            bool condition = aboutPage.MainBlock.Displayed;

            Assert.True(condition);
        }

        [Test, Order(3)]
```

```

public void TestOpenAboutPageFromAboutPage()
{
    aboutPage = aboutPage.OpenAboutPage();
    bool condition = aboutPage.MainBlock.Displayed;

    Assert.True(condition);
}

[Test, Order(4)]
public void TestOpenNewsPageFromAboutPage()
{
    newsPage = aboutPage.OpenNewsPage();
    bool condition = newsPage.MainBlock.Displayed;

    Assert.True(condition);
}

[OneTimeTearDown]
public void TestsTearDown()
{
    driver.Quit();
}
}
}

```

## TestFields.cs

```

using NUnit.Framework;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Firefox;

namespace TestKP
{
    [SetUpFixture]
    [TestFixture(typeof(FirefoxDriver))]
    [TestFixture(typeof(ChromeDriver))]
    public class TestFields<TWebDriver> where TWebDriver :
        IWebDriver, new()
    {
        private NewsPage newsPage;
        private IWebDriver driver;

        [OneTimeSetUp]
        public void TestSetUp()
        {
            this.driver = new TWebDriver();
            driver.Url = "http://127.0.0.1:8080/";
            newsPage = new NewsPage(driver);
        }

        [Test]

```



```

        public void TestLastNameFiled()
        {
            newsPage.LastNameField.SendKeys("Guskov");

            bool result =
newsPage.LastNameField.GetAttribute("value").Equals("Guskov");

            Assert.True(result);
        }

        [Test]
        public void TestFirstNameFiled()
        {
            newsPage.FirstNameField.SendKeys("Pavel");
            bool result =
newsPage.FirstNameField.GetAttribute("value").Equals("Pavel");

            Assert.True(result);
        }

        [Test]
        public void TestEmailFiled()
        {
            newsPage.EmailField.SendKeys("pagusskov62@gmail.com");
            bool result =
newsPage.EmailField.GetAttribute("value").Equals("pagusskov62@gmail.com");

            newsPage.ClearButton.Click();

            Assert.True(result);
        }

        [OneTimeTearDown]
        public void TestsTearDown()
        {
            driver.Quit();
        }
    }
}

```

## TestButtons.cs

```

using NUnit.Framework;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Firefox;

namespace TestKP
{
    [SetUpFixture]

```

```

[TestFixture(typeof(FirefoxDriver))]
[TestFixture(typeof(ChromeDriver))]
public class TestButtons<TWebDriver> where TWebDriver :
IWebDriver, new()
{
    private NewsPage newsPage;
    private IWebDriver driver;

    [OneTimeSetUp]
    public void TestSetUp()
    {
        this.driver = new TWebDriver();
        driver.Url = "http://127.0.0.1:8080/";
        newsPage = new NewsPage(driver);
    }

    [Test]
    public void TestSubmit()
    {
        newsPage.FillRequiredFields("Guskov", "Pavel",
"pagusskov@gmail.com");
        newsPage.SubmitButton.Click();

        bool result = newsPage.IsAlertDisplayed();
        newsPage.ClearButton.Click();

        Assert.True(result);
    }

    [Test]
    public void TestClear()
    {
        newsPage.FillRequiredFields("Guskov", "Pavel",
"pagusskov@gmail.com");
        newsPage.ClearButton.Click();

        bool result = newsPage.IsRequiredFieldsClear();

        Assert.True(result);
    }

    [OneTimeTearDown]
    public void TestsTearDown()
    {
        driver.Quit();
    }
}

```

## Page.cs

```
using OpenQA.Selenium;
```

```

using TestKP.InitElements;

namespace TestKP
{
    abstract class Page
    {
        protected IWebDriver Driver;
        public IWebElement News;
        public IWebElement About;
        public IWebElement SubMenu;

        protected Page(IWebDriver driver)
        {
            this.Driver = driver;
        }

        NavigationElements.InitNavigationElements(driver, this);

        public NewsPage OpenNewsPage()
        {
            News.Click();
            return new NewsPage(Driver);
        }

        public AboutPage OpenAboutPage()
        {
            About.Click();
            return new AboutPage(Driver);
        }
    }
}

```

## NewsPage.cs

```

using OpenQA.Selenium;
using TestKP.InitElements;

namespace TestKP
{
    class NewsPage : Page
    {
        public IWebElement MainBlock;
        public IWebElement SubmitButton;
        public IWebElement ClearButton;
        public IWebElement LastNameField;
        public IWebElement FirstNameField;
        public IWebElement SecondNameField;
        public IWebElement EmailField;
        public IWebElement RankDropBox;
        public IWebElement BirthDayField;
        public IWebElement MaleRadioButton;
        public IWebElement FemaleRadioButton;
    }
}

```

```

        public IWebElement CommentField;
        public IWebElement FirstCheckBox;
        public IWebElement SecondCheckBox;

        public NewsPage(IWebDriver driver): base(driver)
        {
            NewsPageElements.InitNewsPageElements(driver,
this);
        }

        public void FillRequiredFields(string lastName,
string firstName, string email)
        {
            LastNameField.SendKeys(lastName);
            FirstNameField.SendKeys(firstName);
            EmailField.SendKeys(email);
            FirstCheckBox.Click();
        }

        public bool IsRequiredFieldsClear()
        {
            if (string.IsNullOrEmpty(LastNameField.Text) &&
                string.IsNullOrEmpty(FirstNameField.Text) &&
                string.IsNullOrEmpty(EmailField.Text) &&
                !(FirstCheckBox.Selected))
            {
                return true;
            }

            return false;
        }

        public bool IsAlertDisplayed()
        {
            if (Driver.SwitchTo().Alert() != null)
            {
                Driver.SwitchTo().Alert().Accept();
                return true;
            }
            Driver.SwitchTo().Alert().Accept();
            return false;
        }
    }
}

```

## AboutPage.cs

```

using OpenQA.Selenium;
using TestKP.InitElements;

namespace TestKP
{

```

```

class AboutPage : Page
{
    public IWebElement MainBlock;
    public AboutPage(IWebDriver driver) : base(driver)
    {
        AboutPageElements.InitNewsPageElements(driver,
this);
    }
}

```

### NavigationElements.cs

```

using OpenQA.Selenium;

namespace TestKP.InitElements
{
    static class NavigationElements
    {
        public static void InitNavigationElements(IWebDriver
driver, Page page)
        {
            page.News
driver.FindElement(By.XPath("/html/body/div[1]/ul/li[1]/a"));
            page.About
driver.FindElement(By.XPath("/html/body/div[1]/ul/li[2]/a"));
            page.SubMenu
driver.FindElement(By.XPath("/html/body/div[1]/ul/li[2]/ul/li[1]
/a"));
        }
    }
}

```

### NewsPageElements.cs

```

using OpenQA.Selenium;

namespace TestKP.InitElements
{
    static class NewsPageElements
    {
        public static void InitNewsPageElements(IWebDriver
driver, NewsPage newsPage)
        {
            newsPage.MainBlock
driver.FindElement(By.Id("NewsPage"));
            newsPage.ClearButton
driver.FindElement(By.XPath("//div[2]/div[2]/div[2]/form/div[10]
/input[2]"));
        }
    }
}

```

```

        newsPage.SubmitButton
driver.FindElement(By.XPath("//div[2]/div[2]/div[2]/form/div[10]
/input[1]"));
        newsPage.LastNameField
driver.FindElement(By.XPath("//*[@id='lastname']"));
        newsPage.FirstNameField
driver.FindElement(By.XPath("//*[@id='name']"));
        newsPage.SecondNameField
driver.FindElement(By.XPath("//*[@id='secondname']"));
        newsPage.EmailField
driver.FindElement(By.XPath("//*[@id='email']"));
        newsPage.RankDropDown
driver.FindElement(By.XPath("/html/body/div[2]/div[2]/div[2]/for
m/div[5]/select"));
        newsPage.BirthdayField
driver.FindElement(By.XPath("//*[@id='date']"));
        newsPage.MaleRadioButton
driver.FindElement(By.XPath("//*[@id='Male']"));
        newsPage.FemaleRadioButton
driver.FindElement(By.XPath("//*[@id='Female']"));
        newsPage.CommentField
driver.FindElement(By.XPath("//*[@id='comment']"));
        newsPage.FirstCheckBox
driver.FindElement(By.XPath("//*[@id='license']"));
        newsPage.SecondCheckBox
driver.FindElement(By.XPath("//*[@id='spam']"));
    }
}
}

```

### AboutPageElements.cs

```

using OpenQA.Selenium;

namespace TestKP.InitElements
{
    class AboutPageElements
    {
        public static void InitNewsPageElements(IWebDriver
driver, AboutPage newsPage)
        {
            newsPage.MainBlock
driver.FindElement(By.Id("AboutPage"));
        }
    }
}

```