

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»

Магистерская программа: «Вычислительные машины, комплексы, системы и сети»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРА

на тему

Система функционального автотестирования

интернет-сервиса для образовательных
организаций

Студент


03.06.2020
(подпись, дата)

Пучкова Ульяна Николаевна
(ФИО полностью)

Руководитель


03.06.2020
(подпись, дата)

Синев М.П.
(фамилия, инициалы)

Нормоконтролёр


(подпись, дата)

Кучин А.В.
(фамилия, инициалы)

Рецензент

Доцент ПензГТУ  22.06.20
(должность, место работы) (подпись, дата) (фамилия, инициалы)

Печерский С.В.
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 03.06.2020 № 12)

Заведующий кафедрой


(подпись)

Митрохин М.А.
(фамилия, инициалы)

Работа защищена с отметкой

Отлично

(протокол заседания ГЭК от 16.06.20 № 14)

Секретарь ГЭК


(подпись)

Гурин Е.И.
(фамилия, инициалы)

Пенза 2020

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
 М.А. Митрохин
« » 20 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
МАГИСТРА

1. Студент Пучкова Ульяна Николаевна гр. 18ВВм1 факультета
вычислительной техники, направления 09.04.01

2. Тема работы «Система функционального автотестирования интернет-сервиса
для образовательных организаций».

Тема утверждена приказом ПГУ № 1039/о от "30" сентября 2019 г.

3. Руководитель магистерской диссертации к.т.н., доцент Синева М. П.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Назначение разработки: осуществление функциональной
автоматизированной проверки интернет-сервиса для образовательных
организаций.

Разработать программную систему для реализации функциональных
проверок интернет-сервиса для образовательных организаций.

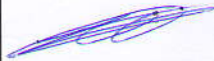
5. Перечень подлежащих разработке вопросов

1. Анализ предметной области автоматизированного функционального
тестирования и фреймворков автоматизации тестирования.

2. Разработка системы функционального автоматизированного тестирования на основе проведенного анализа.

3. Отладка системы функционального автоматизированного тестирования на интернет-сервисе.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области функционального тестирования	13%	30.09.2019-31.10.2019	
2.	Анализ предметной области автоматизированного тестирования	13%	01.11.2019-22.11.2019	
3.	Анализ фреймворков для автоматизации тестирования	20%	23.11.2019-19.01.2019	
4.	Разработка системы функционального автоматизированного тестирования на основе проведенного анализа	29%	20.01.2019-17.04.2019	
5.	Отладка системы функционального автоматизированного тестирования	15%	18.04.2019-9.05.2020	
6.	Оформление выпускной работы	10%	10.05.2020-10.06.2020	

Дата выдачи задания "26" сентября 2019 г.

Руководитель магистерской диссертации  Синева М. П.
(подпись, дата)

Задание к исполнению принял студент  Пучкова У. Н.
(подпись, дата)

Работу к защите допустить
Декан факультета  Фионова Л.Р.
(подпись, дата)

РЕФЕРАТ

Магистерская работа содержит 135 листов, 28 рисунков, 2 таблицы, 28 источников, 5 приложения.

ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ, АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ, WEB-СЕРВИСЫ, CODECEPTION, ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ, ОТЧЁТЫ ТЕСТИРОВАНИЯ

Цель работы – изучение систем автоматизированного тестирования и разработка программной системы для реализации функциональных проверок интернет-сервиса образовательных организаций.

Объектом разработки является система автоматизированного тестирования, осуществляющая функциональную проверку интернет-сервиса. Данная система позволяет выполнять виртуально действия пользователя и генерирует отчет о проведенных проверках.

В результате проведенной работы разработана система автоматизированного тестирования на базе фреймворка Codeception на языке PHP с применением паттернов проектирования Page Object, Steps, Value Object.

Результаты данной работы применены в научно-исследовательской работе ООО «Ай-Теко ВЦ» для организации рабочего процесса.

Содержание

Введение.....	6
1 Анализ предметной области	8
1.1 Определение функционального тестирования	8
1.2 Направления функционального тестирования	9
1.2.1 Ручное тестирование	10
1.2.2 Регрессионное тестирование	10
1.2.3 Smoke-тестирование	10
1.2.4 Интеграционное тестирование	11
1.2.5 Тестирование мобильных приложений	11
1.2.6 Тестирование безопасности	11
1.2.7 Системное тестирование	12
1.2.8 Тестирование установки	12
1.2.9 Конфигурационное тестирование	12
1.2.10 Обеспечение тестового покрытия	12
1.2.11 Тестирование удобства использования	13
1.3 Определение автоматизации тестирования	13
1.4 Направления автоматизации тестирования	14
1.4.1 GUI-тестирование	14
1.4.2 Тестирование веб-сервисов	14
1.4.3 Автоматизация тестирования мобильных приложений	14
1.4.4 Тестирование баз данных	15
1.4.5 Тестирование веб-приложений	15
1.5 Обзор фреймворков для автоматизации тестирования	16
1.5.1 Selenium	16
1.5.2 Codeception	16
1.5.3 Katalon Studio	17
1.5.4 QTP / UFT	18
1.5.5 Watir	19

1.5.6	IBM Rational Functional Tester.....	19
1.5.7	TestComplete.....	20
1.5.8	TestPlant eggplant	21
1.5.9	Tricentis Tosca	22
1.5.10	Ranorex.....	23
1.5.11	Robot framework.....	23
1.6	Выводы	24
2	Проектирование системы автотестирования	25
2.1	Развёртка системы тестирования	25
2.1.1	RHP	25
2.1.2	Codeception.....	25
2.1.3	PhpStorm	26
2.2	Реализация бизнес-логики и функциональности интернет-сервиса посредством реализации структуры проекта с использованием паттернов проектирования	28
2.2.1	Основные разделы проекта	29
2.2.2	Использование паттерна StepObjects	32
2.2.3	Использование паттерна PageObjects.....	34
2.2.4	Использование паттерна Value Object.....	39
2.3	Реализация функционального тестирования	41
2.3.1	Требования к тест-кейсам для автоматизации тестирования	41
2.3.2	Пример адаптированного тест-кейса для автоматизации	42
2.3.3	Пример реализации сценария «Зачисление».....	43
2.4	Выводы	48
3	Результаты работы системы автотестирования	49
3.1	Взаимодействие ручных тестировщиков и инженеров-автоматизаторов на проекте.....	49
3.2	Результаты выполнения автотестирования и структура отчета	50
3.1	Выводы	53
	Заключение	54

Список используемых источников	55
Приложение А Листинг StepObjects	58
Приложение А.1 Файл AcceptanceTester.php	58
Приложение А.2 Файл AdministrationSectionTester.php	70
Приложение А.3 Файл GroupsTester.php	72
Приложение В Листинг PageObjects	84
Приложение В.1 Файл HeaderComponent.php	84
Приложение В.2 Файл LoginClass.php	89
Приложение В.3 Файл RolesClass.php	93
Приложение В.4 Файл ServerClass.php	94
Приложение С Листинг Value Object	106
Приложение D Тест «Зачисление»	123
Приложение E Презентация	125

Введение

Информационные технологии уже давно стали неотъемлемой частью нашей жизни. Ежедневно каждый из нас пользуется различными информационными и программными системами. В таких условиях очень важно, чтобы каждая такая система работала корректно.

В условиях современной мировой рыночной экономики в сфере IT-индустрии больше шансов на успех имеют те компании, чьи решения быстрее выходят на рынок. Это означает, в частности, что программисты во всём мире работают так, чтобы их код как можно скорее был воплощён в жизнь. Но человеку свойственно ошибаться, а в таких условиях гонки – особенно. Как результат, в системах остаются различного рода ошибки.

Для проверок работоспособности информационных систем уже на этапах их разработки тестируется каждый разработанный модуль или компонент; тестируется перед релизом и вся информационная система. Специально для этих целей наряду с разработчиками есть тестировщики, выполняющие комплекс проверок системы в целом и её модулей, суть и цель которых – выявление и устранение ошибок в работе программного комплекса.

Одна из наиболее важных проверок – проверка функционала, то есть проверка правильности выполнения определенных задач, указанных в постановке как функциональные требования, – или функциональное тестирование. Для сложных информационных систем таких требований очень много, они отличаются и объёмом, и разнообразием используемого функционала, и потому возникает необходимость автоматизации процесса тестирования. Мировые тенденции в данной области также показывают, что автоматизация тестирования является на сегодняшний день трендом.

Интернет-сервисы образовательных организаций – это информационные системы, к которым предъявляется широкий спектр требований, и тестировать их без автоматизации процесса – весьма трудоёмкий и затратный не только в плане времени процесс.

В процессе выполнения данной работы были изучены концепции, применяемые в системах автоматизации функционального тестирования. Результатом работы стала система автоматизированного функционального тестирования интернет-сервиса для образовательных организаций. Полученные результаты были применены в научно-исследовательской работе ООО «Ай-Теко ВЦ» для организации рабочего процесса.

1 Анализ предметной области

1.1 Определение функционального тестирования

Под функциональным тестированием понимается процесс установки соответствия разработанного программного обеспечения функциональным требованиям, поставленными заказчиками, для решения определенных задач пользователей.

Данный вид тестирования основан в первую очередь на пошаговом анализе и изучении особенностей определенного приложения, как небольшого компонента, так и целой системы.

Каждой разработанному компоненту, системе или приложению ставятся определенные задачи – функциональные требования, включающие в себя: пригодность, точность, способность к взаимодействию, соответствие правилам и стандартам, а также защищенность.

Главными преимуществами использования функционального тестирования является то, что оно: полностью имитирует функционал системы; позволяет на каждом шаге выявлять системные ошибки ПО, тем самым, позволяет избежать множества проблем в дальнейшем; экономит время на более ранних этапах жизненного цикла ПО, путем исправления ошибок.

К основным этапам функционального тестирования относятся:

- подготовка;
- проведение;
- отчет.

На этапе подготовки сначала проводится анализ системы: технического задания, паспорта проекта, функциональных и бизнес-требований. Затем разрабатывается и согласовывается план тестирования, тест-кейсы, проектные

сроки, число итераций, оценки возможных рисков. Выполнение задач происходит совместно с заказчиком, либо его представителями.

На втором этапе, тестирование ведется вручную на основе заранее подготовленных тестовых сценариев с учетом всех найденных ошибок. Задачей теста, кроме выявления ошибок, ограничивающих функционал программ, является отсутствие проблем в работе тестировщика. Идеальный тест не оставляет вопросов к заказчику или разработчику.

На третьем этапе, разрабатываются и согласовываются отчеты о проведенном тестировании со списками отклонений и рекомендаций по улучшению системы. При необходимости проводится обучение пользователей.

1.2 Направления функционального тестирования

В настоящее время существует ряд направлений функционального тестирования помогающих выявить определенные закономерности поведения той или иной среды, программы или компонента. Направления функционального тестирования сведены в рисунке 1.

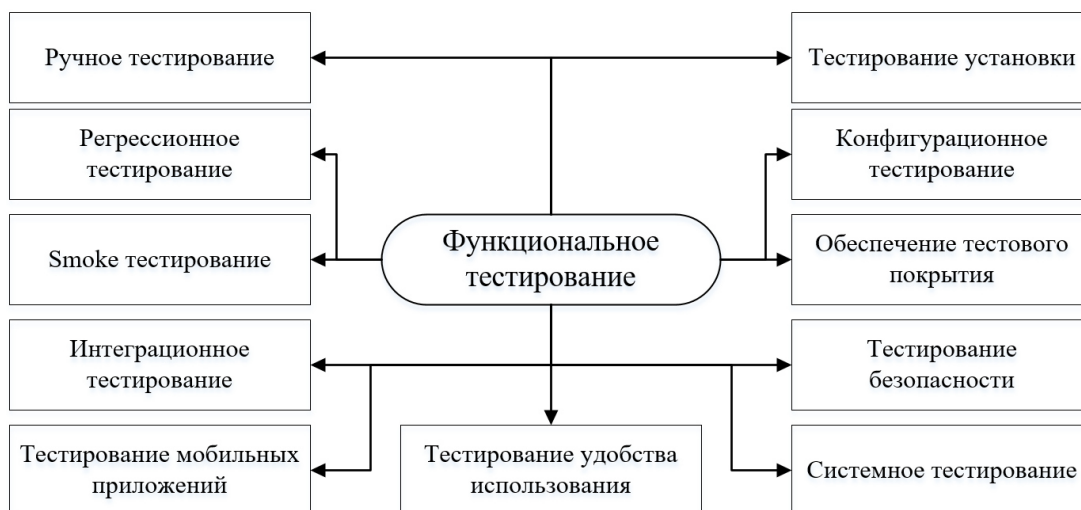


Рисунок 1 – Виды функционального тестирования

Рассмотрим каждое из них.

1.2.1 Ручное тестирование

При ручном функциональном тестировании проверка функций программного обеспечения осуществляется с помощью тест-кейсов. Основная цель данного вида тестирования заключается в определении способности программы при определенных условиях решать задачи, нужные пользователям.

В большей степени для функционального тестирования используются именно ручные тесты, потому что их легче адаптировать под нужные цели и задачи. Ручные тестировщики могут выявлять дефекты, которые не предполагались ранее, т.е. могут сами увеличить тестовое покрытие, и рассмотреть то, что могло быть не упущено в тестовых сценариях. Однако, на масштабных проектах, где возможно множество повторяющихся тестов, целесообразнее использовать автотесты.

1.2.2 Регрессионное тестирование

Регрессионное тестирование – тестирование основных функций данного продукта после выявления и исправления ошибок или реализации новых функциональных возможностей.

После каждого внесения изменений в систему существует вероятность того, что данные изменения повлияют на работоспособность функционала или системы в целом, которое было разработано ранее. Регрессионное тестирование позволяет проверить насколько корректны дополнения и удостовериться в том, что система после изменений продолжит соответствовать установленным требованиям и будет также успешно взаимодействовать с другими модулями.

1.2.3 Smoke-тестирование

Smoke-тестирование – малый цикл тестов для выявления правильности работы основных функций приложения.

Частая сборка программного обеспечения не всегда проходит качественно, из-за чего программный продукт может включать в себя ошибки

работы критичные для функционирования бизнеса. Именно поэтому проверку главного функционала системы следует проводить перед передачей ПО на тестирование и сразу после сборки.

1.2.4 Интеграционное тестирование

Интеграционное тестирование представляет собой тестирование взаимодействий не только между компонентами одной системы, но и между несколькими системами.

Большинство современных IT-систем взаимодействуют друг с другом, поэтому крайне важно иметь представление об их взаимосвязи, а также важно проверять их работоспособность. Интеграционное тестирование позволяет обнаружить дефекты систем, возникающие при их взаимодействии. Для этого возможно использование как ручных, так и автоматизированных тестов.

1.2.5 Тестирование мобильных приложений

Тестирование мобильных приложений – важнейший этап разработки мобильных приложений, основанный на выявлении дефектов в работе графического интерфейса. Этот вид тестирования позволяет проверить работоспособность приложения на различных устройствах и операционных системах в соответствии с заданными требованиями.

1.2.6 Тестирование безопасности

Тестирование безопасности – это анализ уязвимостей программного обеспечения к различным атакам и попыткам несанкционированного доступа к данным. Оно проводится с целью оценки устойчивости системы к противоправным действиям: хакерским атакам, вирусам, попыткам доступа к корпоративной информации. Тестирование безопасности крайне актуально для web-приложений, приложений с коммерческой или персональной информацией, платежных систем, приложений, требующих целостности

информации, социальных приложений. В основном для проверки безопасности таких систем данного тестирования оказывается недостаточно.

1.2.7 Системное тестирование

Системное тестирование – тестирование, основанное на проверке соответствия программного обеспечения с требованиями, заявленными в спецификации. Оно предназначено для тестирования готового ПО, готового для внедряться в опытно-промышленную эксплуатацию.

1.2.8 Тестирование установки

Тестирование установки состоит из проведения тестов процесса инсталляции или деинсталляции программного обеспечения.

Тестирование установки или инсталляционное тестирование позволяет убедиться в том, что ПО корректно устанавливается и настраивается, обновлений новых версий происходит без ошибок, а также предусмотрена ли возможность деинсталлировать и удалить данное ПО. Тестирование инсталляции необходимо проводить во время создания ПО, после выхода новой версии и при изменении конфигураций стенда.

1.2.9 Конфигурационное тестирование

Конфигурационное тестирование заключается в проверке работы программного обеспечения на различных программных и аппаратных средствах. Данный вид тестирования применяется в том случае, если известно, что информационный продукт будет использоваться на разных платформах, в различных браузерах или будет поддерживать разные версии драйверов.

1.2.10 Обеспечение тестового покрытия

Тестовым покрытием называется величина, выражающая процентное соотношение функциональности, проверяемой тестами, к полной функциональности системы. Оценку тестового покрытия рекомендуется проводить при подготовке плана и методики испытаний, для того чтобы

тестирование смогло обеспечить требуемый уровень тестового покрытия. Данный вид тестирования может проводиться как вручную, так и автоматизировано.

1.2.11 Тестирование удобства использования

Тестирование удобства использования определяет степень понятности и привлекательности разрабатываемого продукта. Оно проводится во время создания нового ПО, а также во время его доработки. Главной целью такого тестирования является поиск решений наиболее дружелюбного интерфейса программы, заключающего в минимизации времени на решение задач пользователей.

1.3 Определение автоматизации тестирования

Под автоматизированным тестированием понимается написание кода сравнения ожидаемых сценариев с получаемыми от пользователя и ведется учет расхождений. Данное тестирование играет важную роль в тяжёлых приложениях с большим функционалом.

Плюсы автоматизированного тестирования:

- возможность достаточно быстрого моделирования большого количества пользователей;
- экономия времени;
- повышение качества;
- возможность повторного использования.

Основные этапы автоматизации тестирования:

1. Подготовка: выбор бизнес-операций, подлежащих автоматизации, определение требований к системе, согласование проектных сроков, выбор инструментария автоматизации, оценка возможных рисков.

2. Проведение: запуск автоматизированных тестов и, если необходимо, проведение регрессионного автоматизированного тестирования.
3. Отчет: составление итогового документа с результатами тестирования, создаются руководства пользователя и инструкции по настройке и сопровождению системы

1.4 Направления автоматизации тестирования

1.4.1 GUI-тестирование

Автоматизированное тестирование GUI позволяет эмулировать работу реальных пользователей и интерфейса системы, при этом осуществляется проверка работы системы на соответствие техническому заданию благодаря современным инструментам автоматизированного тестирования.

Данный тестирование предназначено для проверки работы приложения через графический интерфейс пользователя, потому что они нуждаются в тестировании GUI. Примеры: web-приложения, клиентские и банковские приложения, а также платежные системы.

1.4.2 Тестирование веб-сервисов

В основном все веб-сервисы применяются как промежуточное звено между клиентской частью приложения и данными заказчика. Веб-сервисы могут использовать сразу множество пользователей, поэтому надежность его работы зависит от доступности решений клиента, что позволяет выявить возможные дефекты работы веб-сервиса и проверить его реализацию на соответствие отраслевым стандартам.

1.4.3 Автоматизация тестирования мобильных приложений

Автоматизированное тестирование мобильных приложений применяется для оперативного выявления дефектов приложения после дополнений или обновлений. Компания заказчика в данном случае получит возможность

проверки работы мобильного приложения без существенного увеличения трудозатрат. Данный тип тестирования применяется для проверки работоспособности приложения на большом перечне мобильных устройств и операционных систем.

1.4.4 Тестирование баз данных

Тестирование баз данных дает возможность проверить корректность функционирования базы данных (БД) и минимизировать риски при внедрении системы в промышленную эксплуатацию. Также данный вид тестирования позволяет проверить работу БД приложения на предмет соответствия функциональным и нефункциональным требованиям. Любые приложения, включающие в свою архитектуру базы данных, нуждаются в тестировании самой БД.

1.4.5 Тестирование веб-приложений

Тестирование веб-приложений – сложный комплексный процесс, состоящий из множества взаимозависимых элементов и требующий профессионального подхода. Команда опытных экспертов производит полный анализ системы с целью выявления существующих дефектов и проблем, а также обеспечивает контроль полноты их исправления на дальнейших этапах разработки. В процессе проверки приложения специалисты учитывают особенности конкретного проекта и механику взаимодействия серверной части, веб-серверов, базы данных, пользовательского интерфейса и сторонних компонентов.

1.5 Обзор фреймворков для автоматизации тестирования

1.5.1 Selenium

Selenium – инструментарий, специально разработанный для проведения автоматизированного тестирования веб-приложений в различных браузерах и на различных платформах, а также набор средств для автоматизированного тестирования [16]. Вид пользовательского интерфейса представлен на рисунке 2.

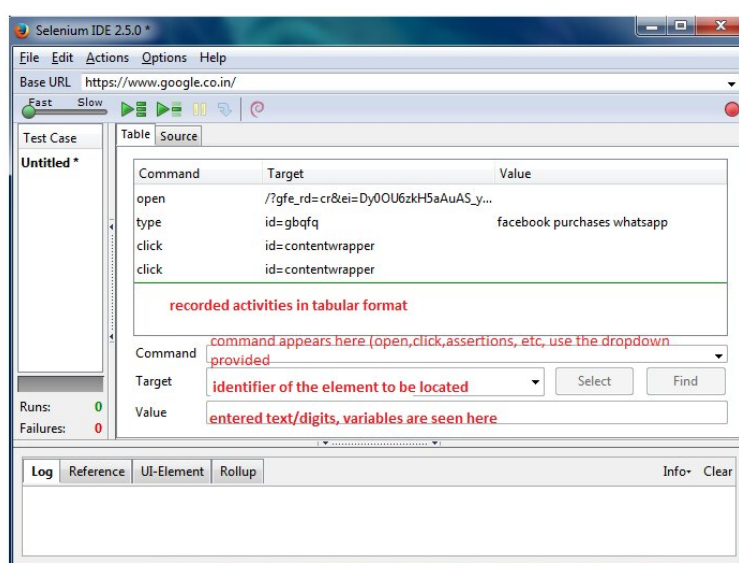


Рисунок 2 – Интерфейс Selenium

1.5.2 Codeception

Codeception – отличный PHP фреймворк для обеспечения проекта авто-тестами (Unit, Behaviour, Acceptance). Codeception использует два драйвера для покрытия тестами веб-интерфейса - guzzlehttp/guzzle и facebook/webdriver (через Selenium2) [17]. В отличие от не менее популярного фреймворка для тестирования по BDD методологии - Behat, Codeception не использует промежуточный слой абстракции - Behat/Gherkin - для обеспечения человеко-понятного описания шагов тестирования. Ключевой концепцией Codeception является представление тестов, как набор действий человека.

Вид пользовательского интерфейса представлен на рисунке 3.

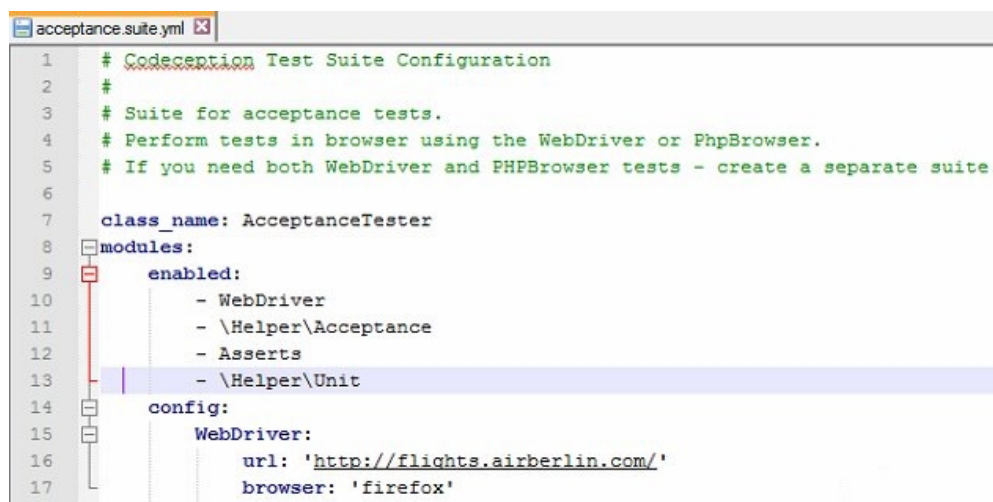


Рисунок 3 - Интерфейс Codeception

1.5.3 Katalon Studio

Katalon Studio это набор инструментов для автоматизированного тестирования веб и мобильных приложений. Он включает в себя полный пакет мощных средств, которые решают часто возникающие проблемы при автоматизации UI тестов, таких как pop-up, iFrame, и wait-time. Это гибкое и удобное решение помогает тестировщикам тестировать лучше, работать быстрее и запускать высококачественное программное обеспечение благодаря интеллекту, которым оно обеспечивает весь процесс автоматизации тестирования. Каждый элемент предоставляет множество возможностей для упрощения обслуживания и увеличения масштаба проекта [18]. Вид пользовательского интерфейса представлен на рисунке 4.

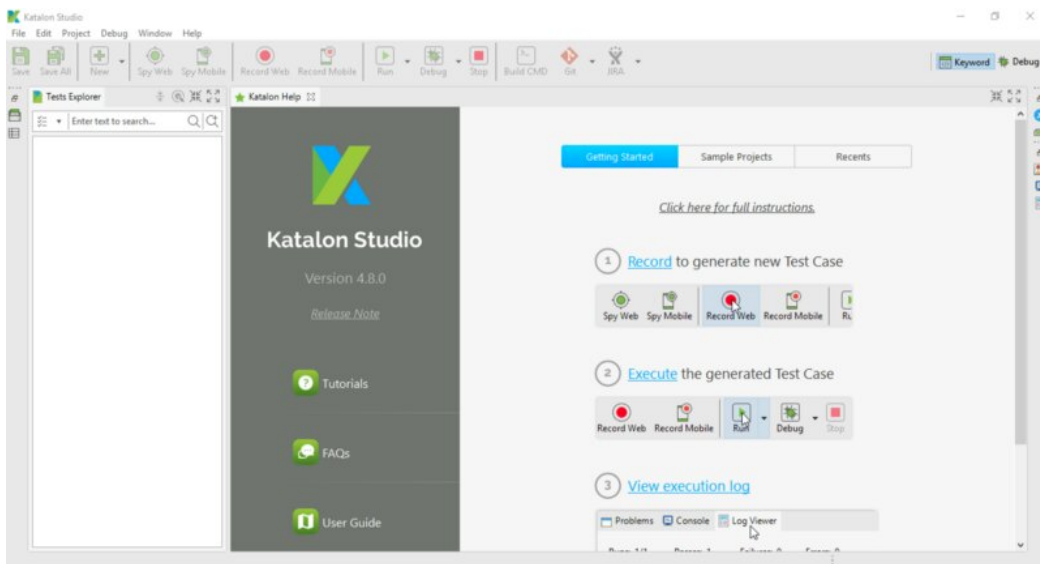


Рисунок 4 - Интерфейс Katalon Studio

1.5.4 QTP / UFT

QTP – это автоматизированный инструмент функционального тестирования, который помогает тестировщикам выполнять автоматизированные тесты для выявления любых ошибок, дефектов или пробелов в отличие от ожидаемых результатов тестируемого приложения. Он был разработан Mercury Interactive и позже приобретен HP, а затем – MicroFocus. Полная форма аббревиатуры QTP – QuickTest Professional, а UFT означает единое функциональное тестирование (Unified Functional Testing). Вид пользовательского интерфейса представлен на рисунке 5.

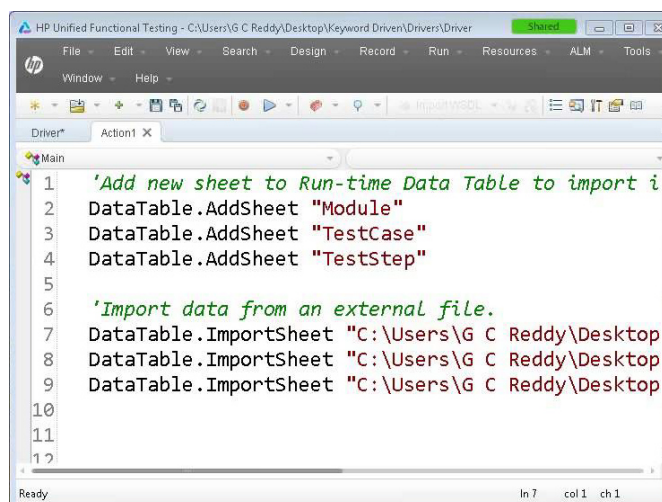


Рисунок 5 - Интерфейс UFT

1.5.5 Watir

Watir – это библиотека Ruby с открытым исходным кодом для автоматизации тестов. Watir взаимодействует с браузером так же, как люди: щелкают ссылки, заполняют формы и так далее [20]. Вид пользовательского интерфейса представлен на рисунке 6.

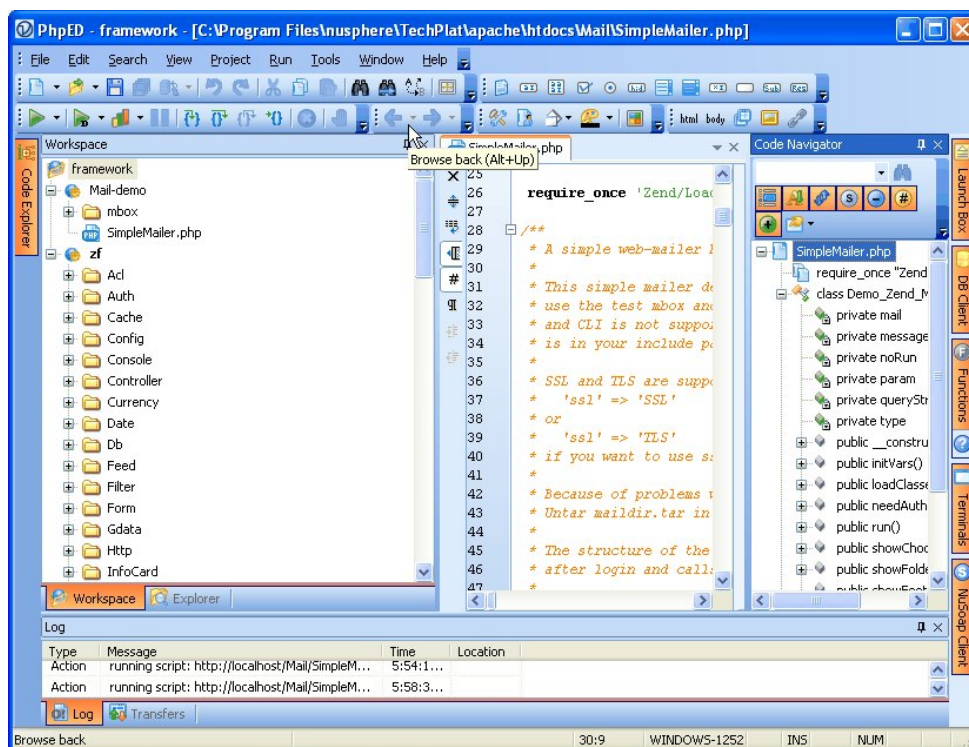


Рисунок 6 - Интерфейс Watir

1.5.6 IBM Rational Functional Tester

IBM Rational Functional Tester предоставляет тестировщикам средства автоматизированного тестирования, позволяющие выполнять функциональное тестирование, регрессивное тестирование, тестирование пользовательского интерфейса и тестирование, управляемое данными. Он поддерживает целый ряд приложений, таких как web-based, .Net, Java, Siebel, SAP, консольные приложения, PowerBuilder, Ajax, Adobe Flex, Dojo Toolkit, GEF, Adobe PDF documents, zSeries, iSeries и pSeries [21]. Вид пользовательского интерфейса представлен на рисунке 7.

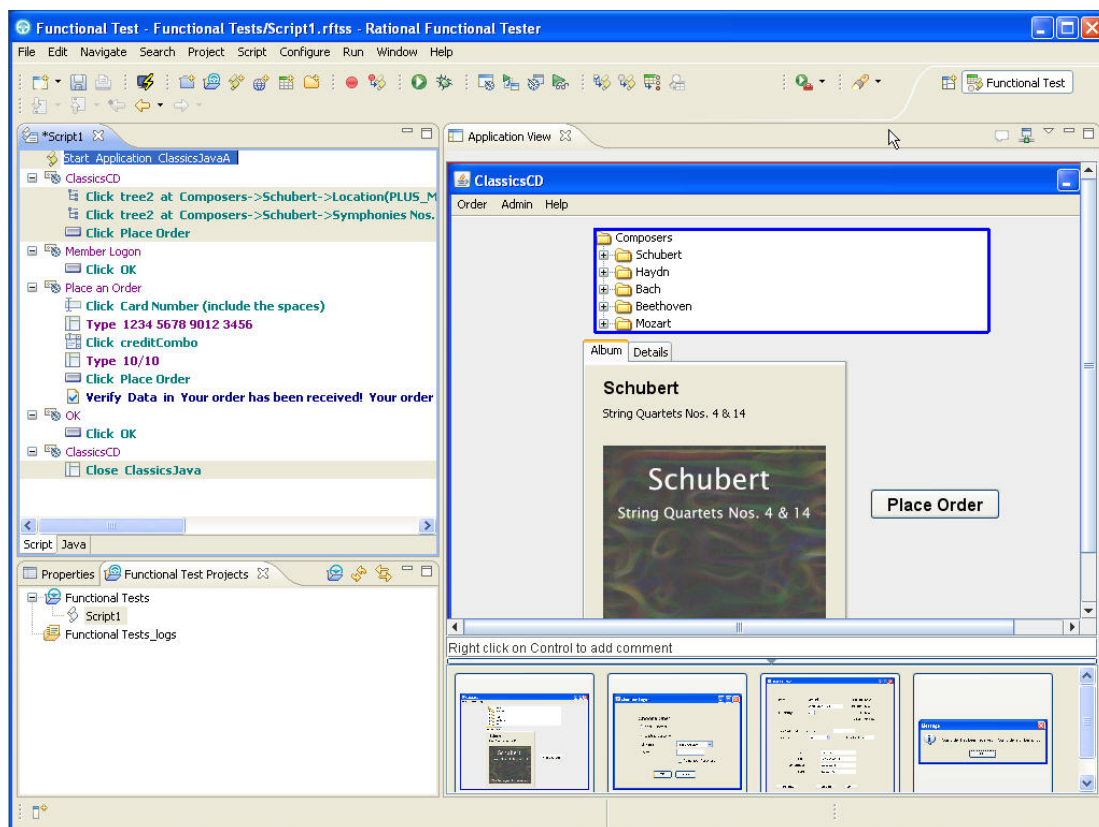


Рисунок 7 - Интерфейс Rational Functional Tester

Является лицензионным ПО, стоимость лицензии составляет 6000\$.

1.5.7 TestComplete

TestComplete – это графический инструмент автоматизации тестирования на ОС Windows. Модель распознавания объектов задействует базовые средства искусственного интеллекта. Предназначен для тестирования десктопных, веб и мобильных приложений [22]. Вид пользовательского интерфейса представлен на рисунке 8.

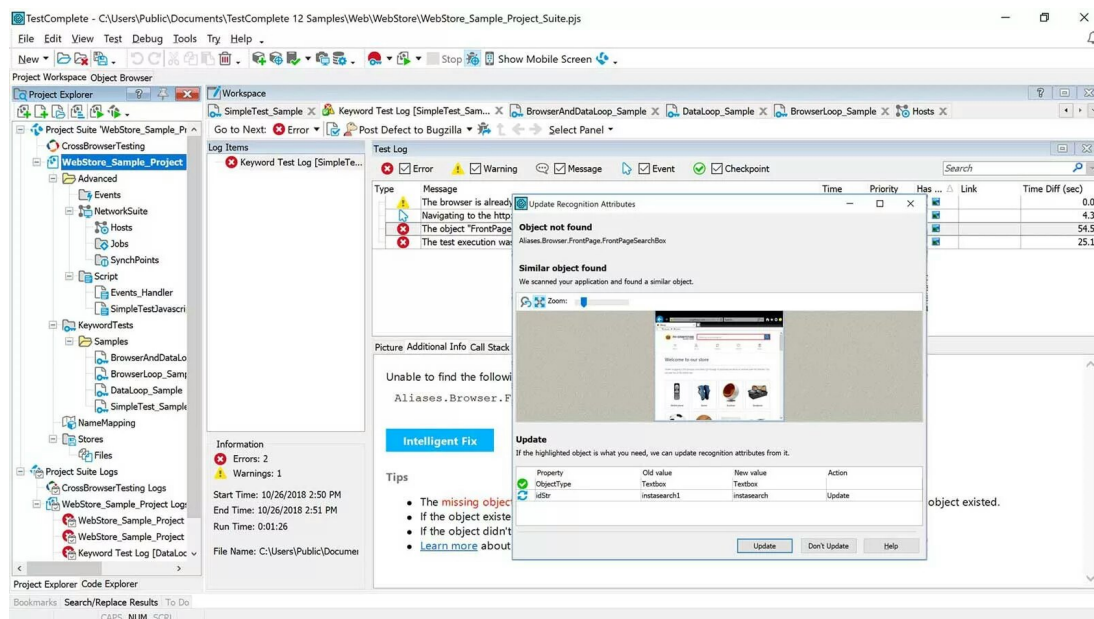


Рисунок 8 - Интерфейс TestComplete

1.5.8 TestPlant eggplant

Это инструмент автоматизации тестирования функционала приложений. Основа тестирования — это изображения, поэтому система может быть использована для запуска подготовленных тестов практически на любой платформе с графическим интерфейсом. Такой подход предлагает большую простоту использования тестировщикам. Будучи основанным на анализе изображений, инструмент распознает объекты на экранах по их внешнему виду и синхронизируется с ними путем сопоставления подготовленных фрагментов изображения с текущим окном приложения. Также имеется функция распознавания текста, которая позволяет находить текст на экранах [23]. Вид пользовательского интерфейса представлен на рисунке 9.

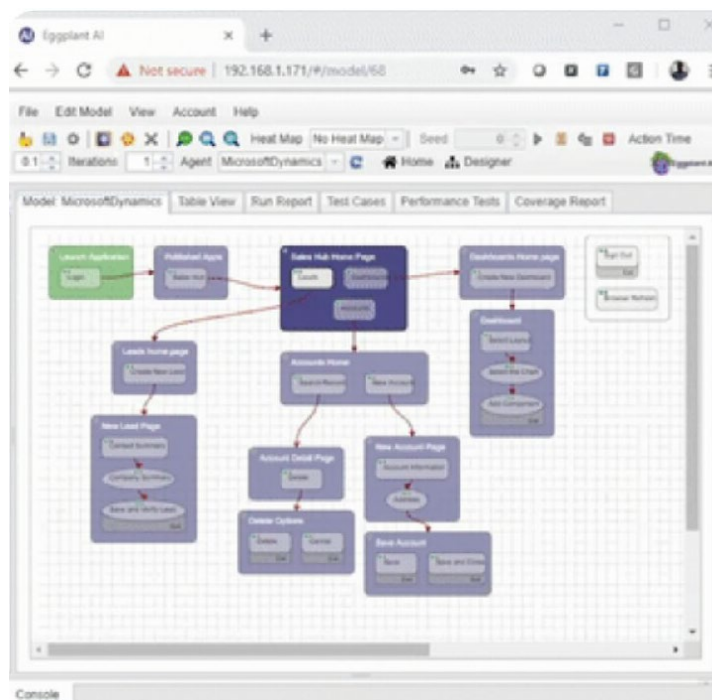


Рисунок 9 - Интерфейс TestPlant eggPlant

1.5.9 Tricentis Tosca

Tosca Test suite by Tricentis – это корпоративный гибкий инструмент тестирования программного обеспечения, используемый для автоматизации сквозных тестовых случаев и обеспечения комплексного управления тестированием для программных приложений. Тестовый инструмент TOSCA основан на методологии LinearQ(sm) [24]. Вид пользовательского интерфейса представлен на рисунке 10.

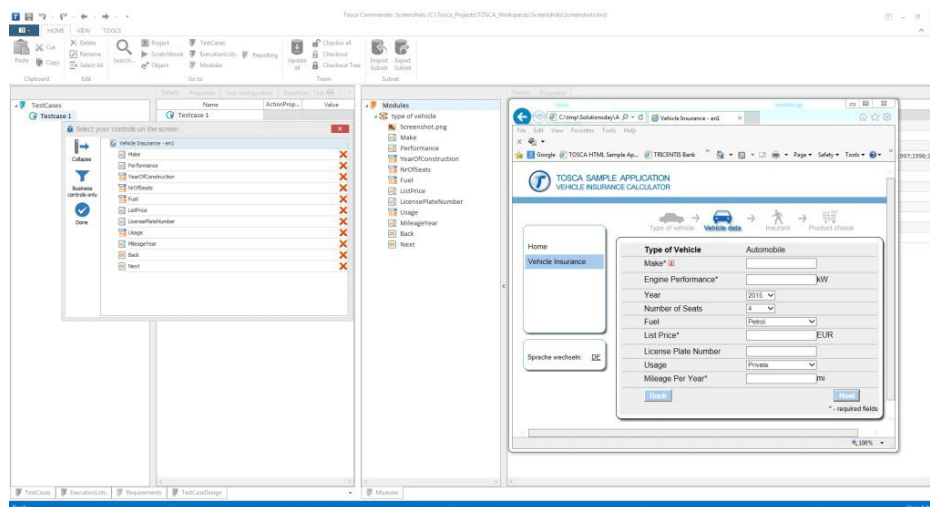


Рисунок 10 - Интерфейс Tricentis Tosca

1.5.10 Ranorex

Ranorex поставляется с полным набором инструментов для сквозного тестирования десктопных, веб и мобильных приложений в одиночной лицензии под операционными системами Windows, iOS или Android или на симуляторах/эмуляторах. Поддерживается как локальный, так и удаленный запуск тестов на реальных устройствах [25]. Вид пользовательского интерфейса представлен на рисунке 11.

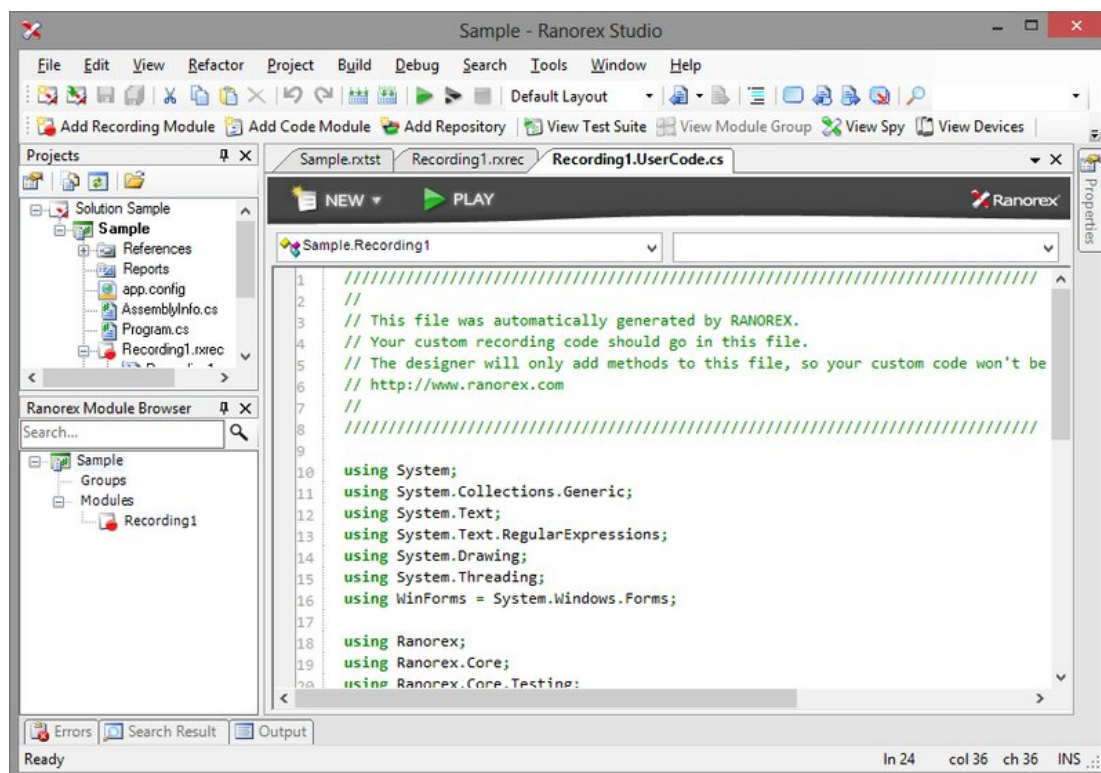


Рисунок 11 - Интерфейс Ranorex Studio

1.5.11 Robot framework

Robot Framework – это универсальная среда автоматизации тестирования с открытым исходным кодом для приемочных испытаний и разработки на её основе. Для написания тестовых примеров используются разные стили тестовых наборов – управляемые ключевыми словами, поведением и данными. Robot Framework обеспечивает поддержку внешних библиотек и инструментов с открытым исходным кодом, которые можно использовать для автоматизации.

Фреймворк построен на Python и работает на Jython (JVM) и IronPython (.NET) [26]. Вид пользовательского интерфейса представлен на рисунке 12.

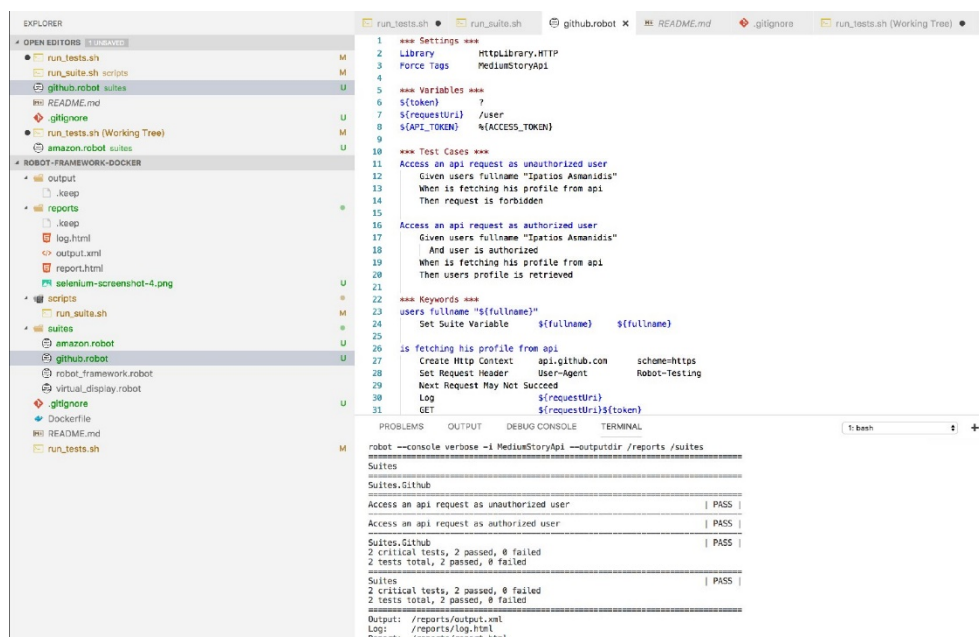


Рисунок 12 - Интерфейс Robot Framework

1.6 Выводы

В данной главе рассмотрены основные направления функционального тестирования, а также направления автоматизации тестирования. Для реализации поставленной задачи с учётом особенностей тестируемого интернет-сервиса образовательной организации в качестве основы выбран фреймворк Codesception.

В следующей главе подробно излагаются теоретические и практические аспекты тестирования применительно к поставленной задаче, а именно: рассматривается развертка системы тестирования, анализируется бизнес-логика тестируемой системы и определяются основные концепции разрабатываемого комплекса тестов.

2 Проектирование системы автотестирования

2.1 Развёртка системы тестирования

Для выполнения всех действий, связанных с тестированием сервиса в среде Codeception, необходимо установить некоторые программные компоненты.

2.1.1 PHP

Ядро PHP скачать из источника [27], распаковать, например, в папку “C:/php-x.y.z”, где x, y и z – это версия PHP. Для PHP версии 7.2.1 путь к папке установки получится “C:/php-7.2.1”.

В папке установки создать файл ‘php.ini’. Он понадобится в дальнейшем для работы с использованием https и Xdebug, процесс загрузки и установки которого подробно изложен в [28].

В переменную среды PHPRC вписать путь к созданному файлу, для данного примера это “C:/php-7.2.1/php.ini”.

Успешное завершение установки PHP проверить запросом информации о PHP, для чего следует выполнить команду:

```
php -r "phpinfo(INFO_GENERAL);"
```

2.1.2 Codeception

Для установки фреймворка Codeception с помощью Composer [14] в выбранном каталоге установки, например, “C:/Codeception”, выполнить скрипт из [17]:

```
composer require codeception/codeception --dev  
php vendor/bin/  
codecept bootstrap
```

Успешное завершение установки Codeception проверить запросом информации о Codeception, для чего следует выполнить команду:

```
codecept -version
```

2.1.3 PhpStorm

Установка PhpStorm стандартная и хорошо описана в [15], но в части Codeception необходимы дополнительные настройки. Необходимо указать расположение компонентов PHP и Codeception во вкладке языков. Соответствующие скриншоты приведены на рисунках 13-35.

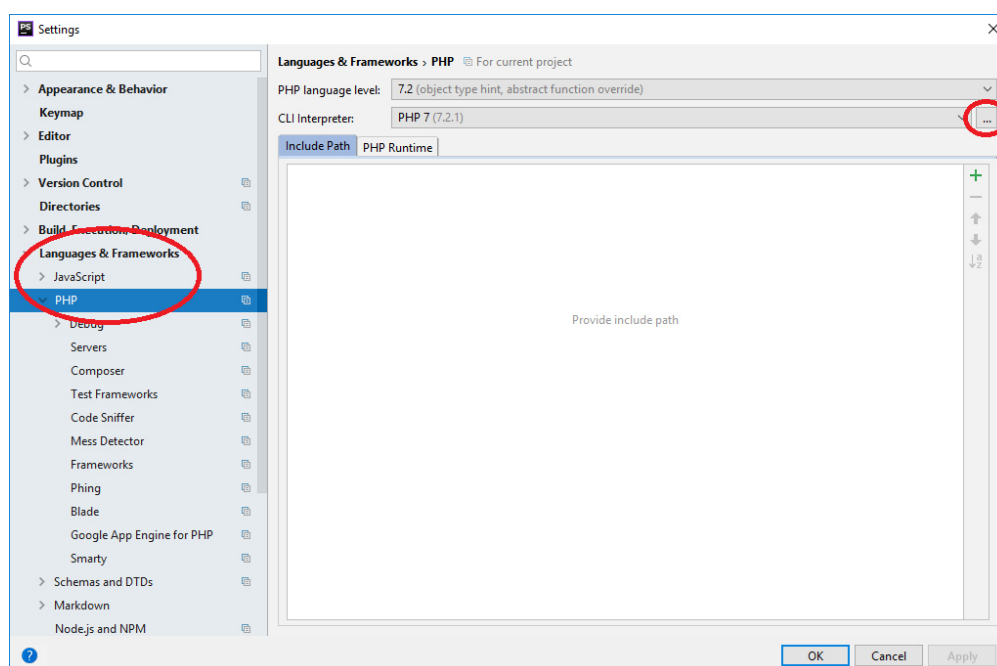


Рисунок 13 - Настройки PHP в окне настроек PhpStorm

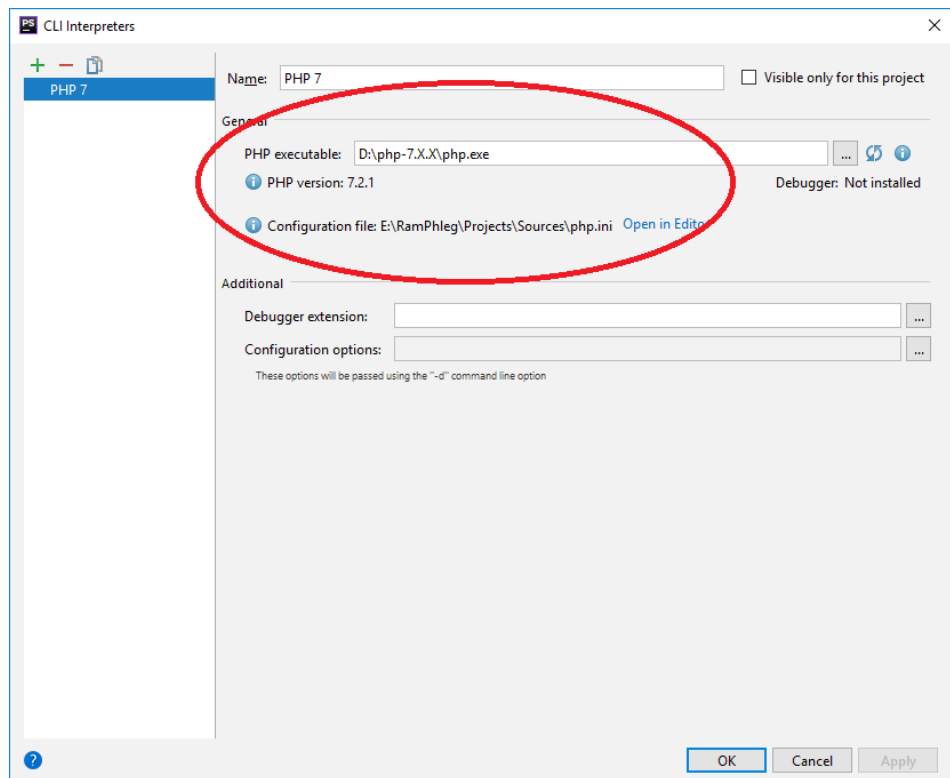


Рисунок 14 - Путь к интерпретатору PHP в настройках

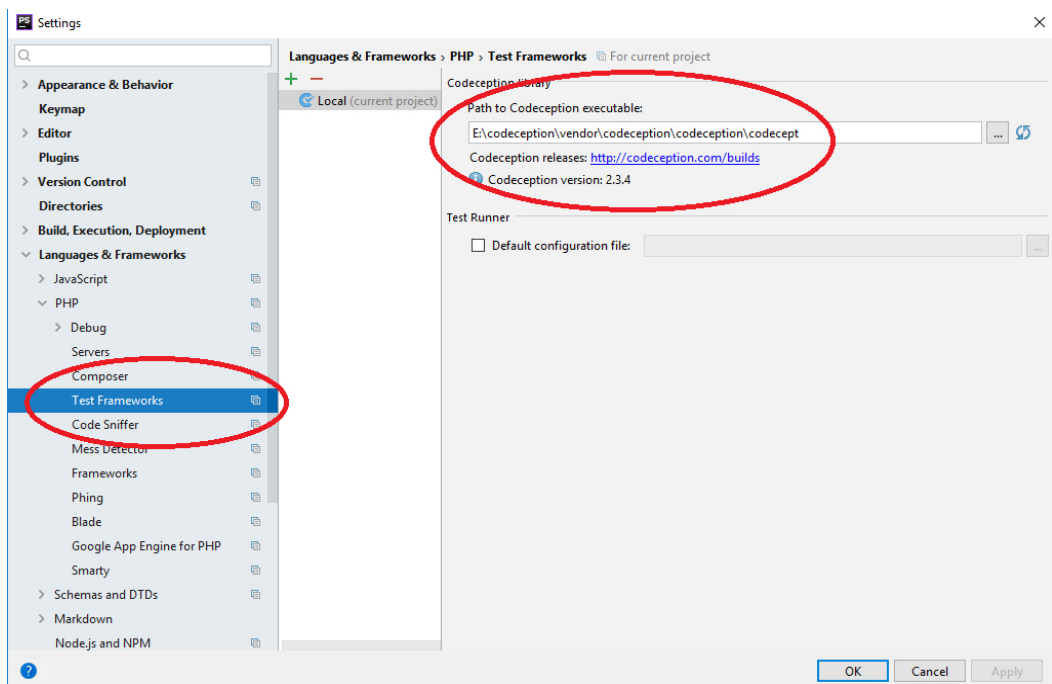


Рисунок 15 - Путь к Codeception в настройках

После выполнения указанных действий PhpStorm готов к работе.

2.2 Реализация бизнес-логики и функциональности интернет-сервиса посредством реализации структуры проекта с использованием паттернов проектирования

Тестируемый ресурс представляет собой интернет-сервис, главной целью которой является учет записи в учреждения дополнительного образования. Функции системы заключаются в следующем:

- предоставление информационной поддержки департаментам, организациям и учреждениям, связанным с дополнительным образованием;
- создание и ведение предоставляемых учреждениями образовательных услуг;
- электронный учет учащихся и педагогов в образовательных организациях.

На данный момент тестируются 24 роли, и 3 контура: 1 предпродуктивный и 2 dev контура.

В тестируемой системе существует 8 разделов интернет-сервиса. В зависимости от роли, прав, департамента и организации, к которой принадлежит пользователь, отображается разное количество этих разделов, а также возможны различия в отображении каждого конкретного раздела.

В системе предполагается 2 вида ролей – внутренний пользователь и администратор, которые создаются на 2 разных подразделах. Для работы администратора используется адрес администратора приложения, а для внутреннего пользователя обычное приложение, контур один и тот же.

Наглядности и понимания на рисунке 16 представлены подразделения ролей и их различия в функциональности.

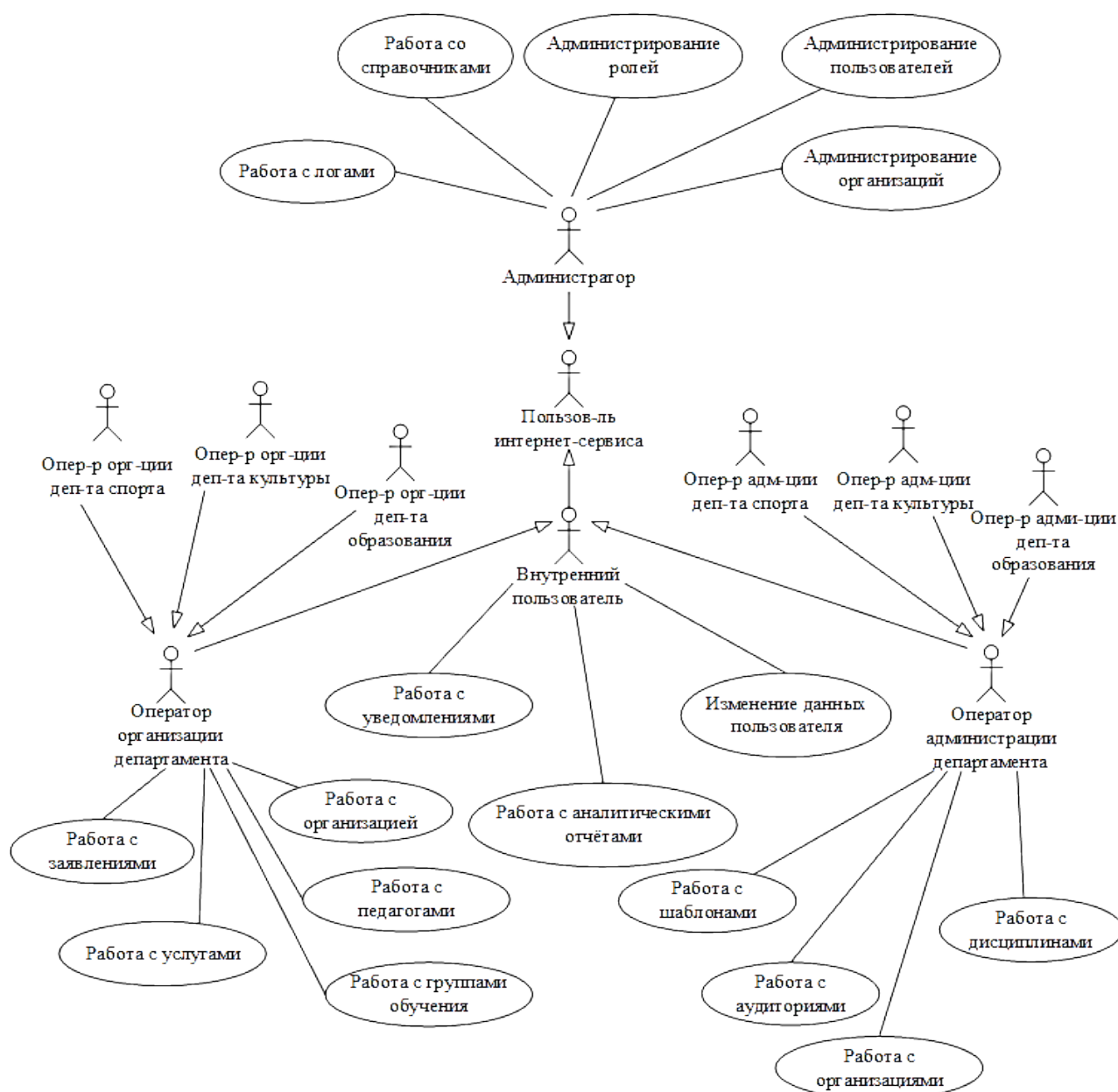


Рисунок 16 – UML-диаграмма прецедентов различий в функциональности ролей

2.2.1 Основные разделы проекта

Тестируемые роли системы хранятся в одном из Page файлов (о использовании паттерна PageObject будет написано далее), как и список тестируемых контуров. Это сделано с целью удобного подключения этих файлов в необходимые тесты.

Так как в проекте переход по разделам осуществляется через шапку сайта, то в отдельный файл были помещены функции, осуществляющие

переход по разделам посредством использования header-a (HeaderClass.php) и посредством прямого перехода на нужную страницу, а также массивы, которые позволяют при написании тестов сразу вставлять готовые названия разделов.

Также в отдельный файл было перемещено все, что относится к авторизации в системе (класс LoginClass.php), так как эта функциональность требуется в каждом файле теста, удобно переместить ее в отдельный класс, минуя AcceptanceTester, который доступен также в любом из файлов.

Разделение на уровне бизнес-логики реализовано с помощью паттерна StepObjects (о нем пойдет речь далее). Таким образом структурное разделение системы соответствует классам реализующим функциональность бизнес-логики.

Наконец, в проекте потребовалось реализовать паттерн Value Object, так как возникла потребность использовать несколько наборов данных для одних и тех же сущностей, плюс отдельные методы, относящиеся к данному классу. В основном для покрытия автотестами системы не требуется подобного, так как сущность создается с определенными свойствами и параметрами всего 1 раз для последующего использования.

На рисунке 17 показана структура проекта с делением по паттернам проектирования.

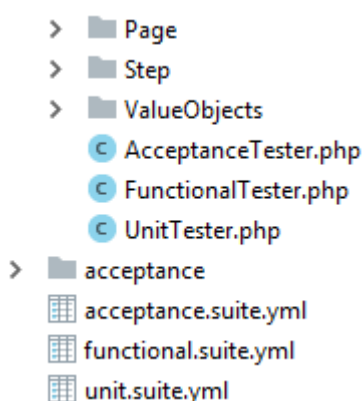


Рисунок 17 – Паттерны проектирования, используемые в проекте

Так как пользователи делятся на администраторов и внутренних пользователей, то в каждом логическом блоке проекта есть разделение на две части, которое и реализует подобное разделение ролей, которое влечет за собой серьезные различия в функциональности.

В разделе acceptance находятся сами тесты. Они в свою очередь также делятся на 2 категории: поверхностная проверка страниц и реализация функциональных сценариев.

В первом случае дальнейшее разделение осуществляется по ролям, во втором по дереву функциональных сценариев, в котором хранятся сценарии для покрытия автотестами.

На рисунке 18 показана структура проекта с учетом вышеописанного

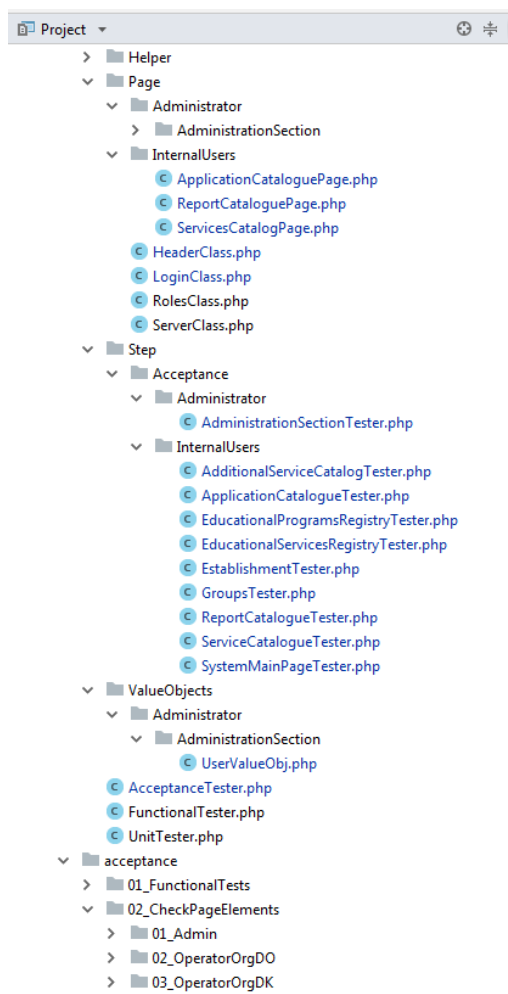


Рисунок 18 – Структура проекта, включающая содержимое основных каталогов

2.2.2 Использование паттерна StepObjects

В результате анализа бизнес-логики тестируемой системы, а также паттернов проектирования для автоматизированного тестирования было принято решение использовать паттерн StepObjects.

Суть данного паттерна заключается в том, что он позволяет разделить в проекте схожую функциональность для групп тестов. В данном случае этот паттерн прекрасно подойдет для того чтобы разделить функциональность разделов системы.

Во фреймворке Codeception существует так называемый главный актер – AcceptanceTester. Он необходим для подключения в каждый тест, потому что через него можно подключиться к библиотеке, которую предоставляет фреймворк помимо прочих возможностей. Отличным решением является наследование этого класса при создании тематического класса для одного из разделов. Например, в системе есть каталог заявлений, значит будет создан класс ApplicationCatalogueTester, который полностью наследует методы AcceptanceTester.

Подобное разделение позволяет реализовать необходимую бизнес-логику проекта при этом сохранив такое же разделение и в системе автотестирования, что позволяет лучше ориентироваться в проекте, писать более чистый код и всегда понимать, насколько проект покрывает разделение бизнес-логики интернет-системы.

Для наглядного представления реализации данного паттерна на рисунке 19 представлена диаграмма классов проекта.

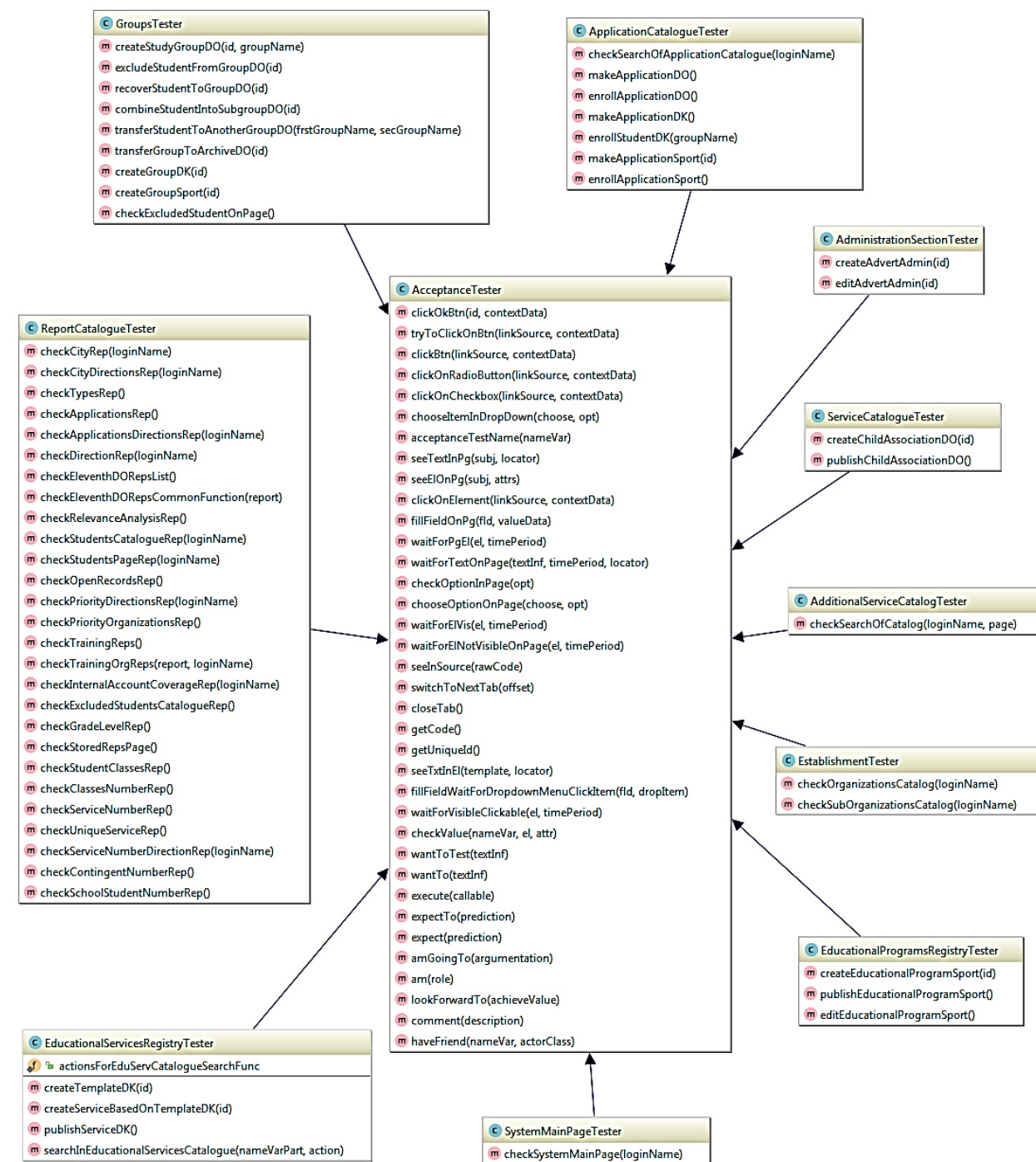


Рисунок 19 – UML-диаграмма классов, наследуемых от AcceptanceTester

Листинги классов StepObjects на рис. 19 представлены в приложении А, за исключением тех, которые не могут быть публично представлены, чтобы не нарушить коммерческую тайну; в представленных листингах все параметры и имена изменены.

2.2.3 Использование паттерна PageObjects

Данный паттерн был специально придуман для использования в автоматизированном тестировании и позволяет сделать код более чистым и понятным, разделить функциональность и вынести методы и переменные в отдельный класс.

В проекте, как уже было сказано ранее, есть основные файлы PageObject:

- HeaderComponent
- LoginComponent
- RolesClass
- ServerClass

HeaderClass содержит функции, переменные и массивы, позволяющие осуществлять переход по Header-у системы.

LoginClass содержит функциональность авторизации в систему. Метод подходят как для пользователей с ролью «Администратор», так и для внутренних пользователей.

RolesClass хранит все тестируемые на данный момент роли.

ServerClass хранит тестируемые адреса тестируемых контуров.

Для классов проверяющих функциональность разделов бизнес-логики интернет-сервиса созданы Page-объекты, в которых хранятся переменные, содержащие локаторы (Названия классов, CSS Selector-ы, ID, Name, XPath) элементов веб-страниц, которые используются при тестировании.

Для администраторов существует один Page-класс, содержащий наборы данных для создания сущности «Пользователь» в разделе, который ответственен в системе за администрирование.

На рисунке 20 представлена структура Page-классов проекта.

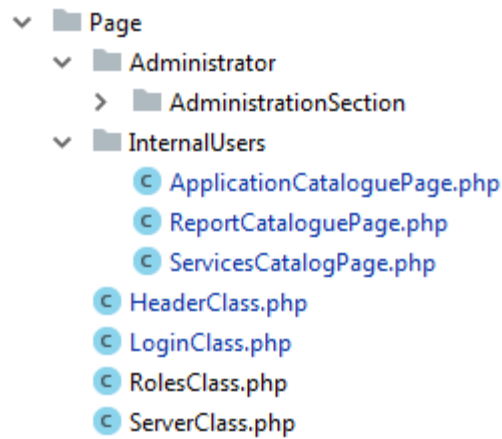


Рисунок 20 – Структура Page-классов

Листинги перечисленных Page-объектов представлены в приложении В.

При проектировании функций были использованы конечные детерминированные автоматы Мура. По данному автомату впоследствии была написана функция.

Код функции логина:

```
public function logIn($login, $server =
ServerClass::TEST_USER)
{
    $I = $this->actor;

    $I->comment('Авторизация в системе');
    $password = "qwertyQWERTY";
    $errFl = 0;
    foreach (RolesClass::$roleNames as $key => $value) {
        if ($key == $login) {
            $I->am(RolesClass::$roleNames[$login]);
        }
    }
    $I->maximizeWindow();
    $I->amOnUrl($server);
    try {
        $I->see('Единый сервис записи');
    } catch (Exception $ex) {
        ␣␣
    }
}
```

```

        $I->reloadPage();
    }

    //Проверка наличия элементов на странице авторизации
    $errFl += $I->seeTextOnPage('Единый сервис записи');
    $errFl += $I->seeTextOnPage('Логин:');
    $errFl += $I->seeElementOnPage('[name="login"]');//Поле "Логин"
    $errFl += $I->seeTextOnPage('Пароль:');
    $errFl += $I->seeElementOnPage('[name="password"]');//Поле "Пароль"
    $errFl += $I->seeTextOnPage('Запомнить');
    $errFl += $I->seeElementOnPage('[name="createPersistenCookie"]');//Чекбокс
    "Запомнить"
    $errFl += $I->seeTextOnPage('Войти', '[class="btn
btn-warning"]');
    $errFl += $I->seeTextOnPage('Войти с помощью ЭП',
'[id="btnSign"]');

    $this->typeLogin($login);
    $this->typePassword($password);
    $this->clickLogInButton();

    $errFl += $I->waitForTextOnPage('Единый сервис
записи', 5, HeaderComponent::$headerWords);
    $errFl += $I->waitForElementOnPage(HeaderClass::$exitButton, 5);
    $I->dontSee('Логин:');
    $I->dontSee('Пароль:');
    $I->tryToClick(HeaderClass::$menuButton);
    return $errFl;
}

```

На рисунке 21 представлен автомат функции авторизации.

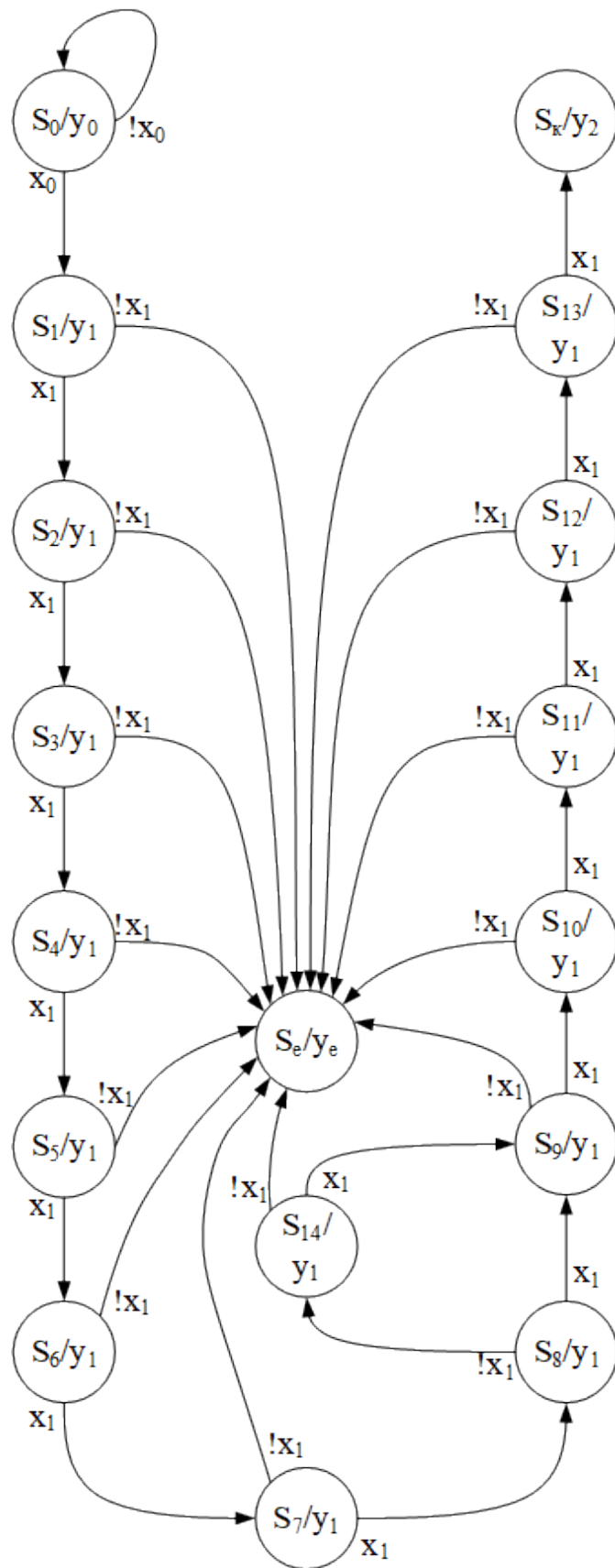


Рисунок 21 - Автомат функции авторизации

Множество состояний автомата:

S_0 – начальное состояние;

S_1 – создание переменной главного актёра;

S_2 – создание комментария: «...»;

S_3 – создание переменной пароля;

S_4 – создание флага ошибки;

S_5 – запись роли в комментарий;

S_6 – расширение окна;

S_7 – переход к серверу;

S_8 – проверка названия проверяемой системы;

S_9 – проверка элементов и надписей на странице авторизации с установкой флага ошибки;

S_{10} – авторизация;

S_{11} – ожидание загрузки страницы с установкой флага ошибки;

S_{12} – проверка отсутствия надписей страницы авторизации на текущей странице;

S_{13} – клик по хедеру страницы;

S_{14} – перезагрузка страницы;

S_s – состояние успешного завершения теста;

S_e – состояние ошибки в ходе выполнения алгоритма;

Множество входных сигналов:

x_0 – сигнал начала алгоритма;

x_1 – сигнал успешного завершения этапа алгоритма;

Множество выходных сигналов:

y_0 – пустой сигнал;

y_1 – сигнал работы алгоритма;

y_2 – флаг ошибки;

y_e – сигнал завершения алгоритма ошибкой;

На основе введенных выше событий система канонических уравнений, описывающих данные события, будет иметь вид:

$$\begin{aligned}
 S_0 &= S_0x_0 \\
 S_1 &= S_0x_1 \\
 S_2 &= S_1x_1 \\
 S_3 &= S_2x_1 \\
 S_4 &= S_3x_1 \\
 S_5 &= S_4x_1 \\
 S_6 &= S_5x_1 \\
 S_7 &= S_6x_1 \\
 S_8 &= S_7x_1
 \end{aligned} \tag{1}$$

$$S_9 = S_7x_1VS_{14}x_1$$

$$S_{10} = S_9x_1$$

$$S_{11} = S_{10}x_1$$

$$S_{12} = S_{11}x_1$$

$$S_{13} = S_{12}x_1$$

$$S_{14} = S_8\overline{x_1}$$

$$S_s = S_{13}x_1$$

$$S_e = S_1\overline{x_1}VS_2\overline{x_1}VS_3\overline{x_1}VS_4\overline{x_1}VS_5\overline{x_1}VS_6\overline{x_1}VS_7\overline{x_1}VS_9\overline{x_1}VS_{10}\overline{x_1}VS_{11}\overline{x_1}VS_{12}\overline{x_1}VS_{13}\overline{x_1}$$

2.2.4 Использование паттерна Value Object

Сценарии функционального тестирования на проекте, как правило, не требуют использования различных наборов данных для одних и тех же сущностей. Но в случае сценариев по проверке создания пользователей администратором необходимо было разрешить вопрос об использовании различных наборов данных. Данное обстоятельство усложнило проектирование системы, так как пришлось прибегнуть к использованию дополнительного

шаблона проектирования. После изучения паттернов проектирования для автоматизированного тестирования для решения подобной задачи был выбран паттерн, имеющий название Value Object.

Суть паттерна состоит в том, что для определенного объекта может быть использована разная информация для его свойств. Также объект может обладать определенным функционалом. И преимущество данного паттерна в том, что для использования какого-либо функционала необходимо передать сам объект, вместо перечня параметров, как это может осуществляться в методах, где не требуется различного набора данных, а только лишь один или несколько отличительных признака. В данном случае используется уникальный, генерируемый при выполнении каждого теста код, который и передается в качестве параметра.

Создать пользователя можно двумя путями:

- передать в конструктор уникальный ID и набор данных, вызвать функцию создания пользователя и передать в нее сущность «пользователь»;
- создать «пустого» пользователя, вызвать функцию формирования свойств сущности «пользователь», передав в нее объект и набор данных, затем вызвать функцию создания пользователя и передать в нее сущность «пользователь».

Наборы данных для пользователя представлены массивами, которых хранится в отдельном Page-файле.

Пример набора данных:

```
public $userDO = array(  
    'org' => 'ТЕСТ орг',  
    'fioData' => 'тестовый пользователь ',  
    'loginName' => 'test',  
    'emailAddress' => '@user.ru',
```

```
'userRole' => Role::OperatorOrgDO  
);
```

Листинг Value Object класса представлен в приложении С.

2.3 Реализация функционального тестирования

2.3.1 Требования к тест-кейсам для автоматизации тестирования

Для написания автотестов не подходят тест-кейсы для ручного тестирования. Это обусловлено тем, что при ручной проверке тестирущик сам ориентируется на интернет-ресурсе (или при тестировании любого другого продукта или ПО), и в случаях, когда, например, необходимо «выбрать любое заявление из результатов поиска», ручной тестирущик легко справится с этой задачей, а вот для того, чтобы автотест реализовал аналогичный сценарий, необходимо задать точные параметры для выбора заявления. Усложняет данную ситуацию ещё и тот факт, что в подобных списках (результаты поиска, реестры, справочники) элементы, как правило, не имеют уникального локатора элемента (например класс или ID элемента), что усложняет процесс идентификации фреймворком искомого объекта.

Для того чтобы инженеры-автоматизаторы быстрее и качественнее писали автотесты, необходимо адаптировать сценарии ручной проверки или написать заново сценарий для автотестирования.

В случае с выбором заявления в результатах поиска, оно точно должно быть в результатах поиска. Поэтому написание автоматизированных тестов по сценариям невозможно без соответствующих предусловий, которые должны быть выполнены перед началом самого сценария проверки.

Разумеется, необходимым атрибутом сценария для автоматизации является однозначность в использовании данных для него. Иными словами, для тест-кейсов по автоматизации точно также не подходят расплывчатые формулировки в стиле «создать заявление, заполнив его данными».

Необходимо не только указать какие данные должны быть внесены, но и куда. Только в этом случае может идти речь о хороших автоматизированных тестах, которые за счет предусловий становятся более отказоустойчивыми, так как не зависят от внезапных изменений в тестируемой среде.

Такие же требования предъявляются и к ожидаемому результату, который прописывается для каждого шага. Абстрактные формулировки недопустимы. Также не должно быть ветвлений в ожидаемых результатах. Например, два варианта отображения одной и той же страницы, при разных исходных данных. При наличии недостаточных предусловий тест не будет самодостаточным, отказоустойчивым и независимым.

2.3.2 Пример адаптированного тест-кейса для автоматизации

Для успешной работы по автоматизации тестирования и написания тест-кейсов мною был адаптирован один из сценариев программы и методики испытаний для выпуска релиза по проекту. Сценарий данного тест-кейса представлен в таблице 1.

Таблица 1– сценарий «ДО. Создание базовой услуги»

№ п/п	Шаг	Тестовые данные	Ожидаемый результат
1.	Войти в систему. Роль: «Оператор Организации ДО»	Логин: OperatorOrgDO	Пользователь авторизован в системе.
2.	Перейти в раздел «Каталог услуг». Нажать на кнопку «Добавить Услугу».		Открылась страница создания услуги: статус услуги – «Не опубликовано».
3.	Ввести данные в следующие поля: название услуги, описание услуги, описание деятельности, продолжительность обучения, возраст, расписание занятий. Нажать на кнопку "Сохранить".	Использовать данные из набора «Создание базовой услуги #service_base_DO_1»	Данные успешно сохранились, услуг создана. Статус услуги «Не опубликовано».

Пример тестового набора данных представлен в таблице 2.

Таблица 2 – Набор данных «Создание базовой услуги #service_base_DO_1»

№ п/п	Название поля	Данные для заполнения
1.	Название услуги	Тестовая услуга *уникальный номер*
2.	Описание услуги	Тестовое описание
3.	Описание деятельности	Художественная направленность
4.	Продолжительность обучения	12 месяцев
5.	Возраст	10 – 15 лет
6.	Расписание занятий	Ежедневно, кроме выходных

2.3.3 Пример реализации сценария «Зачисление»

Реализацию сценария рассмотрим на примере сценария зачисления на услугу.

Зачисление на какую-либо программу дополнительного образования является наиболее важным этапом для студента, подающего заявления на программу. Оператор какого-либо департамента не взаимодействует лично с абитуриентами, подающими заявки, но отвечает за корректность и правильность ввода данных в интернет-сервис. Поэтому в случае ошибки при вводе данных, либо ошибке на стороне заявителя (предоставил неточные сведения) заявление будет архивировано с указанием причины архивации. Но в случае если все данные внесены корректно, то дальнейшее развитие событий зависит в основном от абитуриента (если не принимать во внимание внешние обстоятельства, такие как: закрытие организации, отмена проведения обучения и т.д.). Абитуриент впоследствии должен явиться и заключить договор. В случае неявки заявителя, заявление переходит в архив. Схема опосредованного взаимодействия заявителя на услугу и оператора показана на рисунке

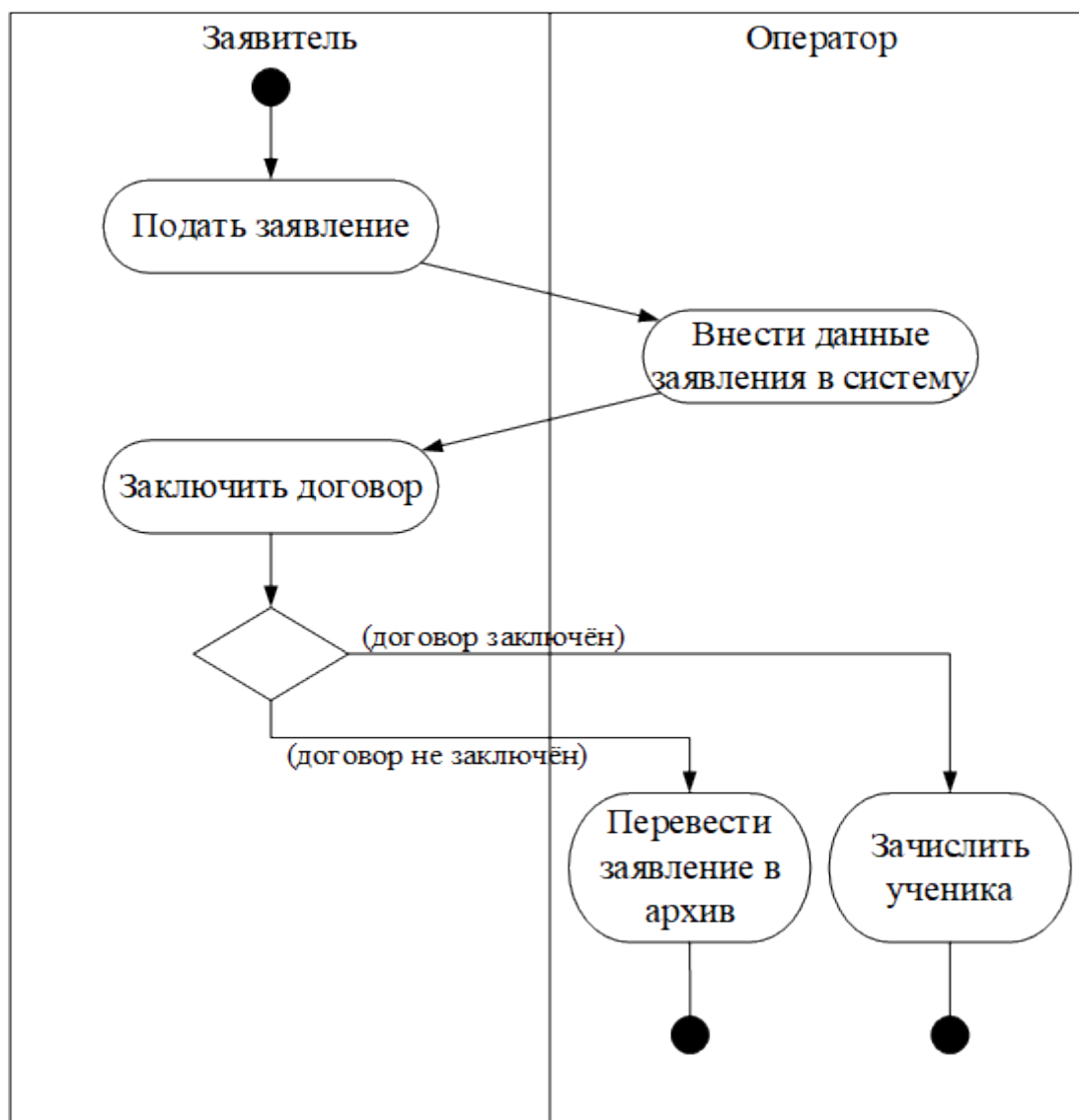


Рисунок 22 – UML-диаграмма деятельности взаимодействия заявителя и оператора сервиса

Но прежде чем подача заявления на какую-либо программу обучения может быть осуществлена необходимо сначала создать услугу и опубликовать ее. Данные об услуге точно также передаются оператору организации департамента руководством организации. Оператор создает услугу, а затем осуществляет публикацию услуги на портале услуг (сторонний, связанный с интернет-сервисом ресурс). Только после публикации на портале услуг на программу (услугу) можно подавать заявления. А для того, чтобы зачислить учеников на программу необходимо создать группы обучения (столько сколько

потребуется). Далее заявление можно перевести в статус «Зачислен» при заключении договора заявителем.

Сценарий зачисления на программу обучения вместе со всеми предусловиями показан на рисунке 23.

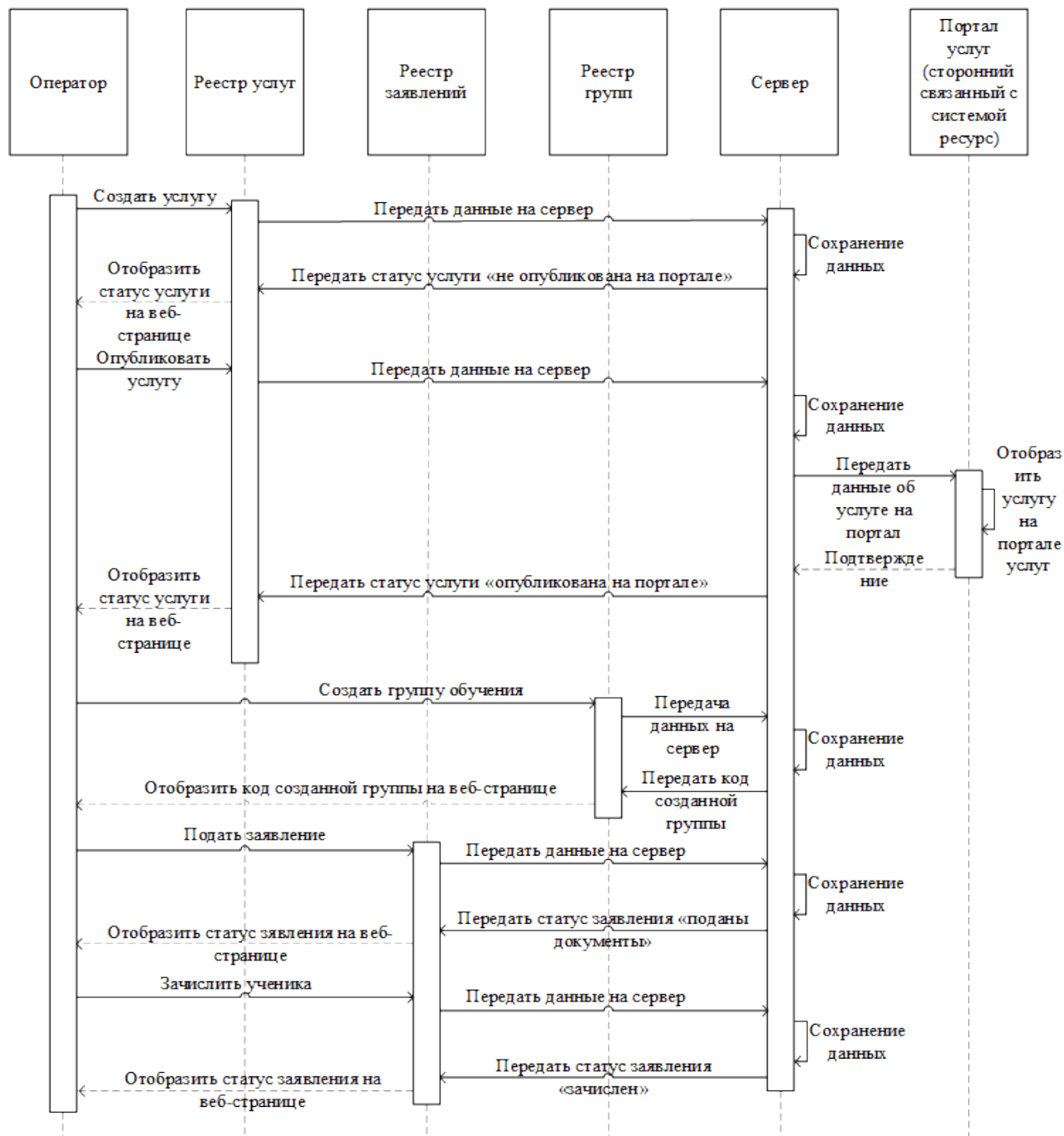


Рисунок 23 UML-диаграмма последовательности полного сценария «Зачисление»

Для успешного проектирования теста по зачислению был создан автомат реализующий данный сценарий. Автомат сценария «Зачисление» представлен

на рисунке 24. Листинг теста по сценарию «Зачисление» представлен в приложении D.

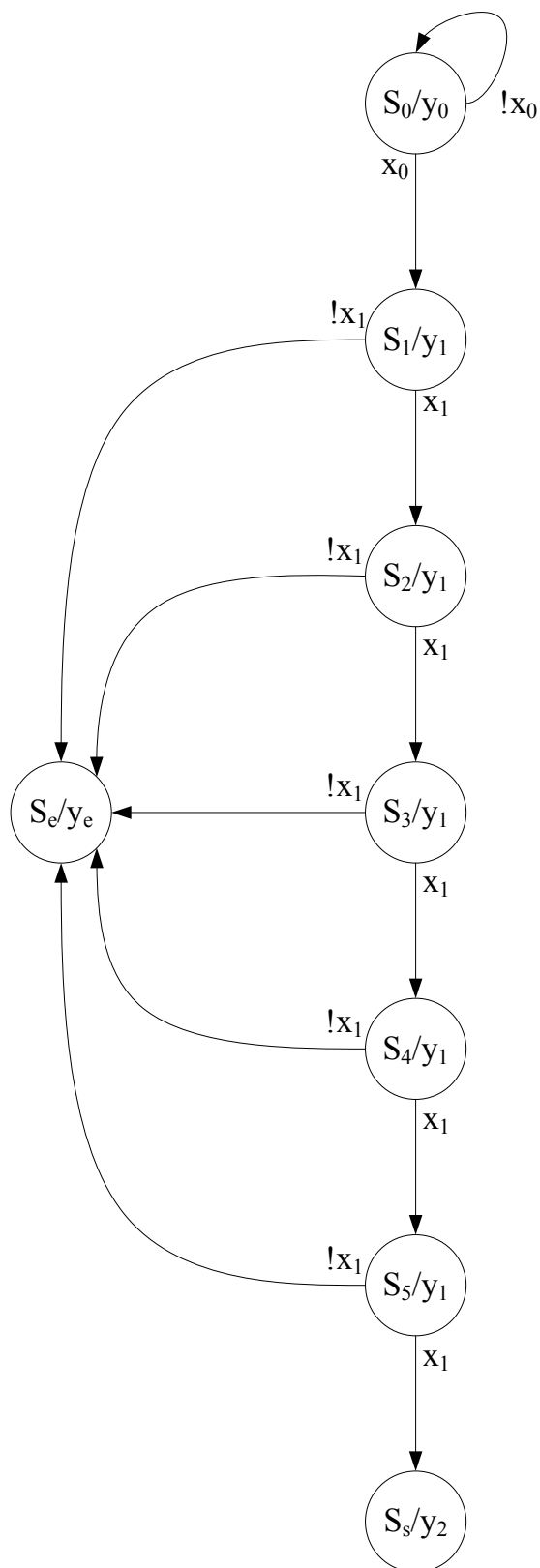


Рисунок 24 – Автомат сценария «Зачисление»

Множество состояний автомата:

S_0 – начальное состояние;

S_1 – создание программы обучения;

S_2 – публикация программы обучения;

S_3 – создание группы;

S_4 – подача заявления на ранее созданную и опубликованную программу;

S_5 – зачисление абитуриента;

S_s – состояние успешного завершения теста;

S_e – состояние ошибки в ходе выполнения теста;

Множество входных сигналов:

x_0 – сигнал начала тестирования;

x_1 – сигнал успешного завершения этапа теста;

Множество выходных сигналов:

y_0 – пустой сигнал;

y_1 – сигнал работы теста;

y_2 – сигнал успешного завершения тестирования;

y_e – сигнал завершения тестирования с ошибкой;

На основе введенных выше событий система канонических уравнений, описывающих данные события, будет иметь вид:

$$\begin{aligned}S_0 &= S_0 \overline{x_0} \\S_1 &= S_0 x_1 \\S_2 &= S_1 x_1 \\S_3 &= S_2 x_1 \\S_4 &= S_3 x_1 \\S_5 &= S_4 x_1 \\S_s &= S_5 x_1 \\S_e &= (S_1 V S_2 V S_3 V S_4 V S_5) \overline{x_1}\end{aligned}\tag{2}$$

2.4 Выводы

Во второй главе были рассмотрены следующие вопросы:

- развертка системы автоматизированного тестирования;
- структура проекта автотестирования интернет-сервиса;
- использование паттернов проектирования PageObjects, Value Object и StepObject в проекте;
- требования к тест-кейсам для автоматизации, был приведен пример тест-кейса;
- описан сценарий зачисления на услугу с помощью интернет-сервиса с помощью детерминированного автомата Мура и UML-диаграмм деятельности и последовательности.

3 Результаты работы системы автотестирования

3.1 Взаимодействие ручных тестировщиков и инженеров-автоматизаторов на проекте

Ручные тестировщики и автоматизаторы на проекте тесно взаимодействуют для более продуктивной работы по автоматизации тестирования.

Совместная работа тестировщиков и автоматизаторов по тестированию интернет-сервиса имеет следующие особенности:

Прогон тестов осуществляется на удаленной виртуальной машине;

- запуск тестов на удаленной машине могут осуществлять как ручные тестировщики, так и автоматизаторы;
- автоматизатор может отлаживать свои тесты на удаленной машине в случае, когда не требуется проверка интернет-сервиса для ручных тестировщиков;
- перед каждым тестированием релиза автоматизатор обновляет версию проекта автотестирования на удаленной машине;
- ручные тестировщики занимаются анализом сгенерированного отчета о прохождении тестов, и затем сообщают автоматизатору, что необходимо актуализировать, а что относится к возникшим ошибкам на интернет-сервисе;
- анализ ошибок после прогона автотестов по тестированию релиза не является обязательным для автоматизатора
- отчет об ошибках прилагается к документам по сдаче релиза по проекту.

Схема взаимодействия ручных тестировщиков и автоматизаторов показана на рисунке 25

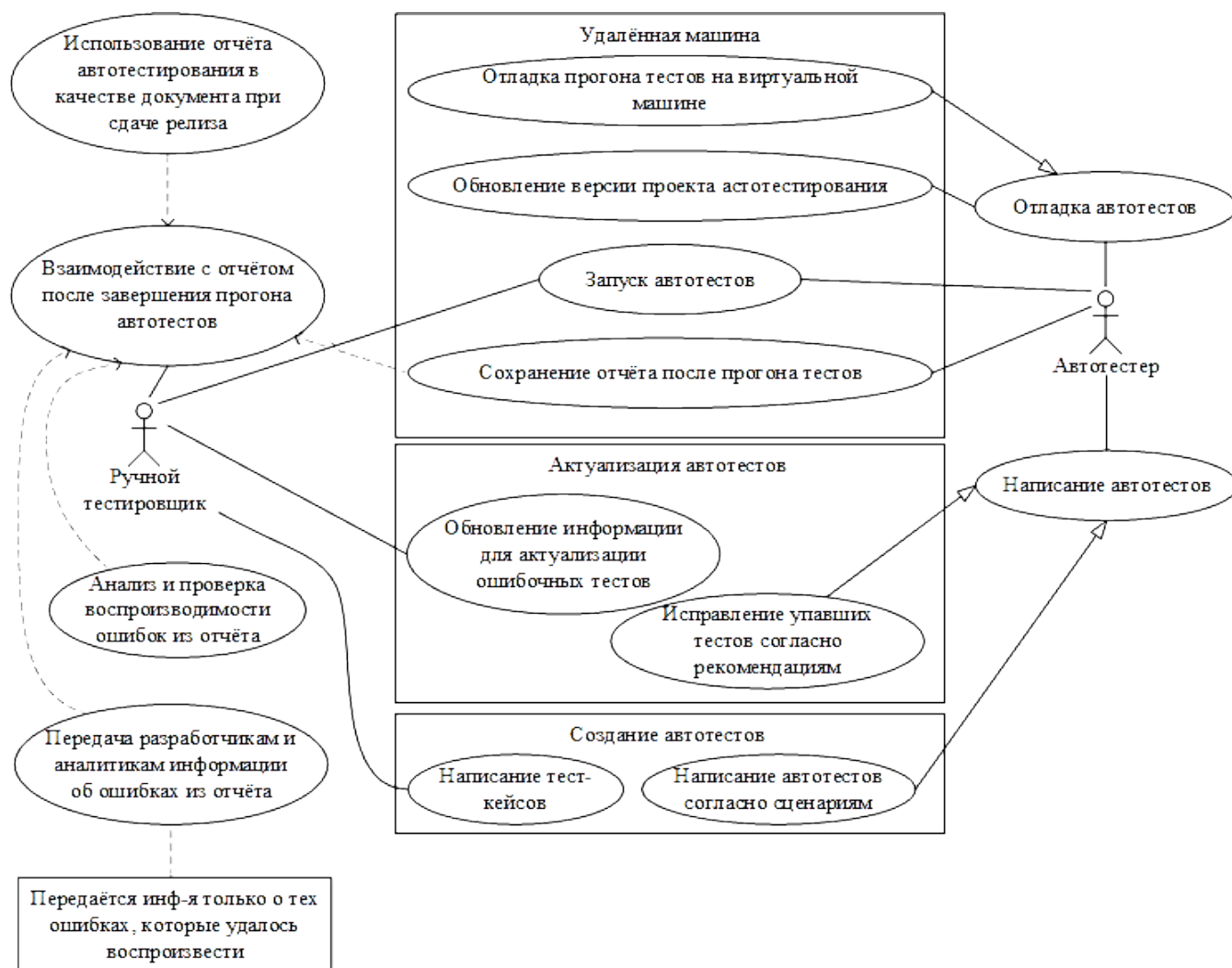


Рисунок 25 – UML-диаграмма прецедентов взаимодействия ручных тестировщиков и автоматизаторов

3.2 Результаты выполнения автотестирования и структура отчета

При тестовом прогоне формируется отчет о выполнении тестов. Фреймворк Codeception позволяет формировать html-отчет, в котором наглядно показаны результаты тестирования.

Html-отчет имеет следующие параметры:

- общее время тестирования;
- общее количество тестов
- количество ошибочно завершенных тестов;
- количество успешно отработанных тестов;
- количество тестов, отработавших не полностью по какой-либо причине;
- тестовые сценарии, представленные в виде строк, и содержащие в себе полное описание шагов тестирования.

На рисунке 26 представлен html-отчет вместе со статистической информацией.

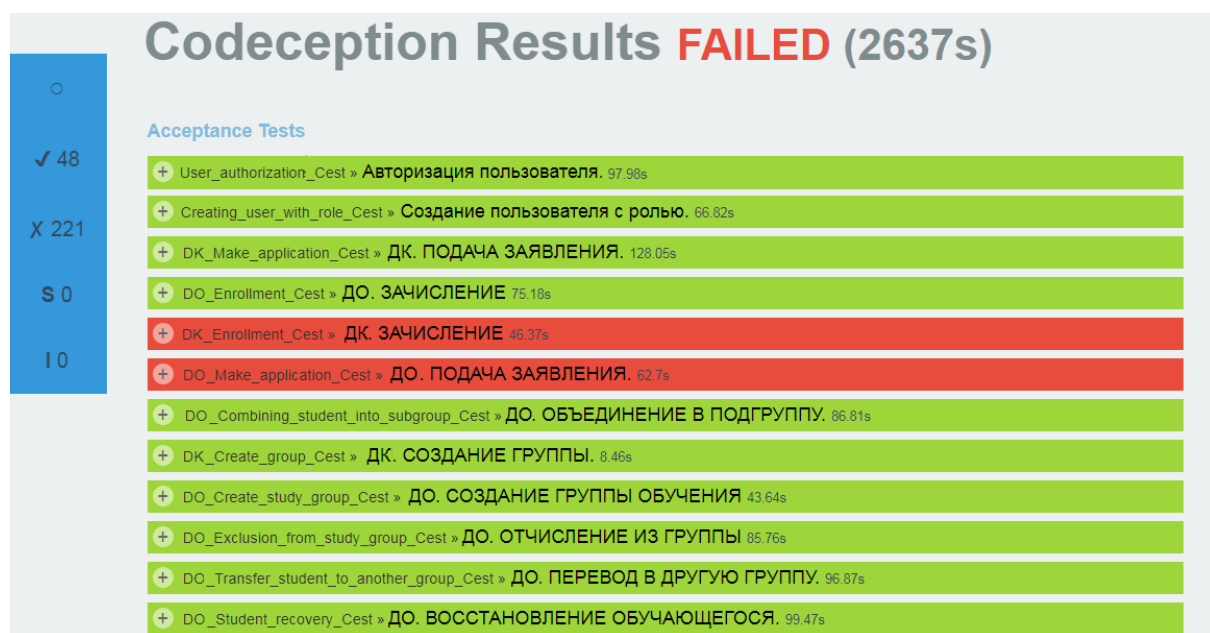


Рисунок 26 - Html-отчет о тестовом прогоне

Успешно заверченный тест выделен зеленым цветом. В раскрытом состоянии отчет об успешном прохождении теста представлен на рисунке 27.

DK_Create_group_Cest » ДК. СОЗДАНИЕ ГРУППЫ.8.46s	
Название теста: ДК. СОЗДАНИЕ ГРУППЫ.	
Авторизация в системе	
As a Оператор Организации ДК	
+ Page\LoginClass: log in "OperatorOrgDK"	
Переход на страницу: "Каталог групп обучения -> Каталог групп обучения"	
+ Page\HeaderClass: go to page "Каталог групп обучения Каталог групп обучения"	
I click "Добавить группу"	
ДК. СОЗДАНИЕ ГРУППЫ.	
+ I create group dk "147541"	
Выход из системы	
- Page\HeaderClass: exit	
I click "#logoutbtn"	
Завершение теста	

Рисунок 27 – Успешно заверченный сценарий

Функции, использующиеся в тесте, показаны в свернутом состоянии, чтобы посмотреть, какие шаги входят в ту или иную функцию, необходимо раскрыть ее, нажав на строку с ней.

В случае неуспешного завершения теста, полоса сценария выделяется красным цветом. При раскрытии подробного описания будет указан шаг, при выполнении которого возникла ошибка. Также фреймворк создает скриншот экрана в момент падения теста, данный скриншот включается в описание сценария. На рисунке 28 представлен отчет по ошибочно завершённому сценарию.

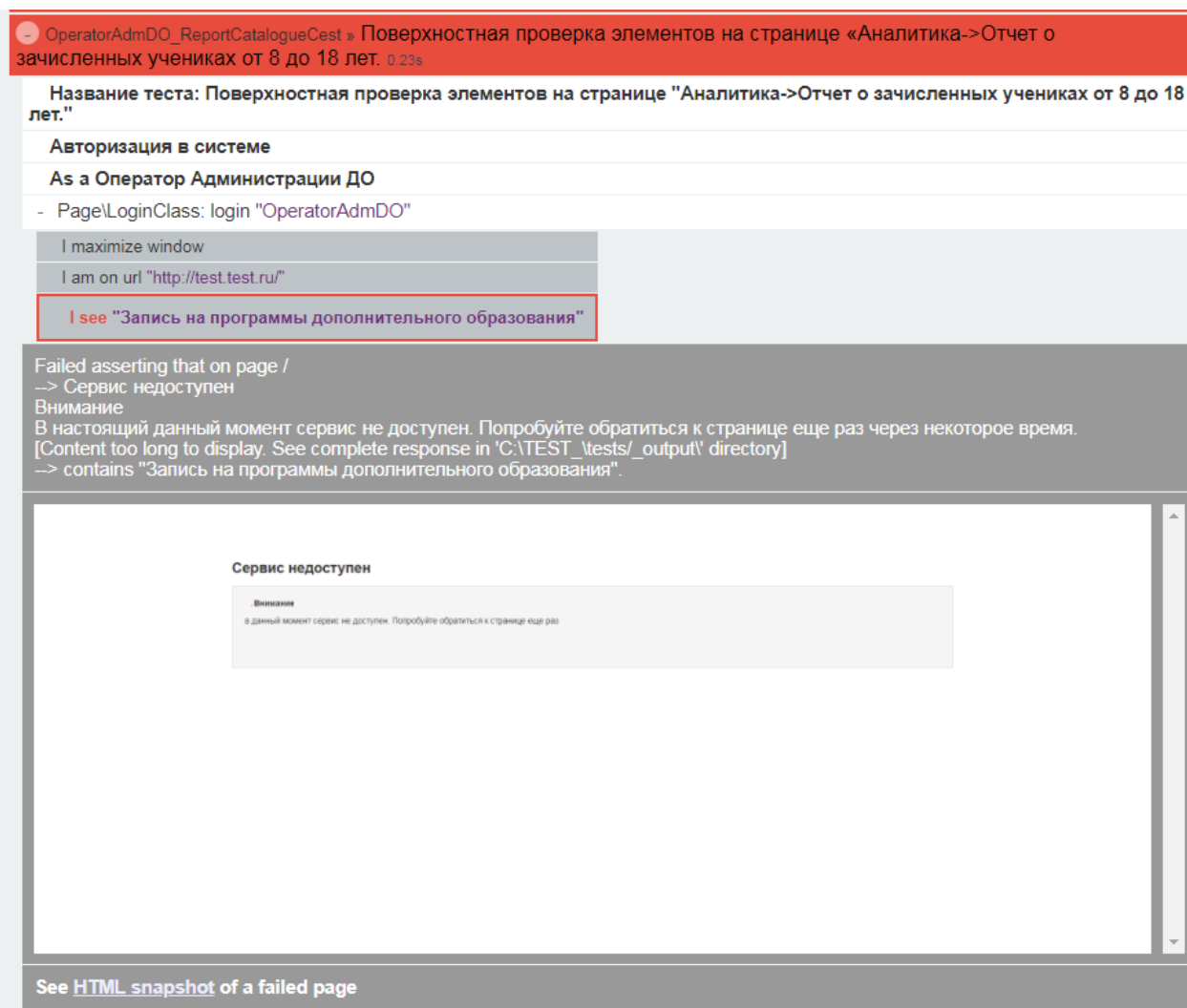


Рисунок 28 – Сценарий, завершившийся с ошибкой

3.1 Выводы

В третьей главе было представлено следующее:

- схема взаимодействия ручных тестировщиков и автоматизаторов;
- результаты тестирования в виде сгенерированного отчета о прогоне тестов;
- пример успешного теста в html-отчете о прогоне;
- пример теста, завершившегося с ошибкой, в html-отчете о прогоне.

Заключение

В рамках данной выпускной квалификационной работы была разработана система функционального автоматизированного тестирования интернет-сервиса для образовательных организаций.

Итогом проведенной работы являются следующие результаты: исследован материал, касающийся следующих предметных областей:

- функциональное тестирование, направления функционального тестирования;
- автоматизированное тестирование, направления автоматизированного тестирования;

Изучены паттерны проектирования для автоматизированного тестирования и проанализированы основные фреймворки для реализации автотестирования, представленные на рынке.

В ходе проектирования разработаны диаграммы классов, прецедентов, последовательности, деятельности и автоматы Мура алгоритмов.

К особенностям проекта можно отнести полное соответствие внутренней структуры проекта автотестирования бизнес-логике интернет-сервиса, а также его простоту конфигурации.

Полученные результаты обладают практической и научной ценностью. Они были применены в научно-исследовательской работе ООО «Ай-Теко ВЦ» для организации рабочего процесса, а также могут быть использованы для тестирования похожих по бизнес-логике интернет-сервисов.

Дальнейшее развитие разработанной системы может включать в себя разработку тестирования REST запросов, БД интернет-сервиса, а также внедрение функционала, позволяющего тестировать взаимодействие данного интернет-сервиса с другими внешними интернет-системами.

Список используемых источников

1. Дастин Э., Рэшка Дж., Пол Дж. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. — Лори, 2019. — С. 54-230.
2. Канер Сэм, Фолк Джек, Нгуен Енг Кек. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. — ДиаСофт, 2001. — С. 120- 301.
3. Куликов С. Тестирование программного обеспечения. Базовый курс. —Четыре четверти, 2015. — С. 108-155.
4. Роман С. Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах. — Издательские решения, 2017. — С. 98-148.
5. Борис Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. — Питер, 2004. — С. 38-120.
6. Гленфорд Майерс, Том Баджетт, Кори Сандлер. Искусство тестирования программ. — Вильямс, 2012. — С. 205-280.
7. Джеймс Уиттакер, Джейсон Арбон, Джефф Каролло. Как тестируют в Google. — Питер, 2014. — С.123-199.
8. Лайза Криспин, Джанет Грегори. Гибкое тестирование. Практическое руководство для тестировщиков ПО и гибких команд. — Вильямс, 2016. — С.112-187.
9. Рекс Блэк. Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование. — Лори, 2011. — С.99-135.

10. Вашкевич Н.П. Недетерминированные автоматы и их использование для реализации систем параллельной обработки информации / Н.П.Вашкевич, Р.А. Бикташев. – Пенза: Изд-во ПГУ, 2016. - 394с.
11. Вашкевич Н.П. Недетерминированные автоматы в проектировании систем параллельной обработки [текст]/ Н.П.Вашкевич: учеб. Пособие. - Пенза: изд-во Пенз.гос.ун-та, 2004. – 280 с.
12. Кларк Э.М. Верификация моделей программы: Model Checking [текст]. Перевод с англ./ Э.М.Кларк, О.Грамберг, Д.Пелед. – М.: МЦНМО, 2002. – 416 с.
13. Пранявичус Г.И. Модели и методы исследования вычислительных систем. – Вильнюс: Мокслас, 1982. – 227 с.
14. Composer [Электронный ресурс]. – Режим доступа: <https://getcomposer.org/>
15. PhpStorm: The Lightning-Smart IDE for PHP [Электронный ресурс]. – Режим доступа: <https://www.jetbrains.org/ru-ru/phpstorm/>
16. SeleniumHQ Browser Automation [Электронный ресурс]. – Режим доступа: <https://www.selenium.dev/>
17. Codeception – PHP testing framework [Электронный ресурс]. – Режим доступа: <https://www.codeception.com>
18. Katalon Automated Tests [Электронный ресурс]. – Режим доступа: <https://www.katalon.com>
19. Katalon Automated Tests [Электронный ресурс]. – Режим доступа: <https://www.katalon.com>
20. Watir Project [Электронный ресурс]. – Режим доступа: <http://www.watir.com>
21. Rational Functional Tester Overview [Электронный ресурс]. – Режим доступа: <https://www.ibm.com/us-en/marketplace/rational-functional-tester>

22. TestComplete: Automated UI testing tool [Электронный ресурс]. – Режим доступа: <https://smartbear.com/product/testcomplete/overview/>
23. Eggplant Functional Documentation [Электронный ресурс]. – Режим доступа: <http://docs.eggplantsoftware.com/eFP/eggplant-functional-documentation-home.htm>
24. Tricentis Automated Software Testing Tools [Электронный ресурс]. – Режим доступа: <https://www.tricentis.com/>
25. Ranorex Test Automation for GUI Testing Tools [Электронный ресурс]. – Режим доступа: <https://www.ranorex.com/>
26. Robot Framework [Электронный ресурс]. – Режим доступа: <https://robotframework.org/>
27. PHP [Электронный ресурс]. – Режим доступа: <http://windows.php.net/download#php-7.2>
28. Xdebug – Debugger and Profiler Tool for PHP [Электронный ресурс]. – Режим доступа: <https://xdebug.org/download.php>

Приложение А

Листинг StepObjects

Приложение А.1

Файл AcceptanceTester.php

```
<?php

use Page\HeaderClass;
use Page\RolesClass as Role;
use Page\ServerClass;

/**
 * Inherited Methods
 * @method void wantToTest($textInf)
 * @method void wantTo($textInf)
 * @method void execute($callable)
 * @method void expectTo($prediction)
 * @method void expect($prediction)
 * @method void amGoingTo($argumentation)
 * @method void am($role)
 * @method void lookForwardTo($achieveValue)
 * @method void comment($description)
 *
 * @method \Codeception\Lib\Friend haveFriend($nameVar,
$actorClass = NULL)
 *
 * @SuppressWarnings(PHPMD)
 */
class AcceptanceTester extends \Codeception\Actor
{
```

```

use _generated\AcceptanceTesterActions;

public function clickOkBtn($id, $contextData = null)
{
    $I = $this;
    $I->comment('Нажатие на кнопку "OK" в модальном
окне');

    $I->wait('2');
    $I->waitForVisibleClickable($id, 7);
    $I->wait('2');
    $I->clickBtn($id, $contextData = null);
    $I->wait('2');
    if (!$I->seeElOnPg($id)) {
        $I->clickBtn($id, $contextData = null);
        if (!$I->seeElOnPg($id)) {
            $I->clickBtn($id, $contextData = null);
        }
    }
    $I->waitForElementNotVisible($id, 15);
}

public function tryToClickOnBtn($linkSource, $contextData
= null)
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('click', func_get_args()));
    } catch (Exception $ex) {
        return 1;
    }
}

```

```

        }
    }

    public function clickBtn($linkSource, $contextData =
null)
    {
        return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('click', func_get_args()));
    }

    public function clickOnRadioBtn($linkSource, $contextData
= null)
    {
        return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('click', func_get_args()));
    }

    public function clickOnChk($linkSource, $contextData =
null)
    {
        return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('click', func_get_args())); //dropdown
    }

    public function chooseItemInDropDown($choose, $opt)
    {
        return $this->getScenario()->runStep(new
\Codeception\Step\Action('selectOption', func_get_args()));
    }

    public function acceptanceTestName($nameVar)

```

```

{
    $this->wantTo($nameVar);
    $this->comment("Название теста: " . $nameVar);
}

public function seeTextInPg($subj, $locator = null)
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('see', func_get_args()));
    } catch (Exception $e) {
        $msg = "Надпись - " . $subj . " - не найдена на
странице. (Функция seeTextInPg) \r\n";
        echo $msg;
        $this->comment($msg);
        return 1;
    }
}

public function seeElOnPg($subj, $attrs = null)
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('seeElement', func_get_args()));
    } catch (Exception $ex) {
        $msg = "Элемент " . $subj . " не найден на
странице. (Функция seeElOnPg) \r\n";
        echo $msg;
        $this->comment($msg);
        return 1;
    }
}

```

```

    }
}

public function clickOnElement($linkSource, $contextData
= null)
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Action('click', func_get_args()));
    } catch (Exception $ex) {
        $msg = "Ошибка при нажатии на элемент " .
$linkSource . ". (Функция clickOnElement) \r\n";
        echo $msg;
        $this->comment($msg);
        return 1;
    }
}

public function fillFieldOnPg($fld, $valueData)
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Action('fillField', func_get_args()));
    } catch (Exception $ex) {
        $msg = "Невозможно заполнить поле " . $fld . ".
(Функция fillFieldOnPg)\r\n";
        echo $msg;
        $this->comment($msg);
        return 1;
    }
}

```

```

    }

    public function waitForPgEl($el, $timePeriod = null)
    {
        try {
            return $this->getScenario()->runStep(new
\Codeception\Step\Action('waitForElement', func_get_args()));
        } catch (Exception $ex) {
            $msg = "Элемент " . $el . " отсутствует на
странице, время ожидания истекло. (Функция waitForPgEl) \r\n";

            echo $msg;

            $this->comment($msg);

            return 1;
        }
    }

    public function waitForTextOnPage($textInf, $timePeriod =
null, $locator = null)
    {
        try {
            return $this->getScenario()->runStep(new
\Codeception\Step\Action('waitForText', func_get_args()));
        } catch (Exception $ex) {
            $msg = "Текст " . $textInf . " отсутствует на
странице, время ожидания истекло. (Функция waitForTextOnPage)
\r\n";

            echo $msg;

            $this->comment($msg);

            return 1;
        }
    }

```

```

    }

    public function checkOptionInPage($opt)
    {
        try {
            return $this->getScenario()->runStep(new
\Codeception\Step\Action('checkOption', func_get_args()));
        } catch (Exception $ex) {
            $msg = "ошибка при установке радиобаттона или
чекбокса" . $opt . ". (Функция checkOptionInPage) \r\n";
            echo $msg;
            $this->comment($msg);
            return 1;
        }
    }

    public function chooseOptionOnPage($choose, $opt)
    {
        try {
            return $this->getScenario()->runStep(new
\Codeception\Step\Action('selectOption', func_get_args()));
        } catch (Exception $ex) {
            $msg = "ошибка при установке значения в
выпадающем списке или радиобаттона" . $opt . ". (Функция
chooseOptionOnPage) \r\n";
            echo $msg;
            $this->comment($msg);
            return 1;
        }
    }
}

```

```

        public function waitForElVis($el, $timePeriod = null)
        {
            try {
                return $this->getScenario()->runStep(new
\Codeception\Step\Action('waitForElementVisible',
func_get_args()));
            } catch (Exception $ex) {
                $msg = "возникла ошибка при ожидании появления
элемента " . $el . " на странице. (Функция waitForElVis) \r\n";
                echo $msg;
                $this->comment($msg);
                return 1;
            }
        }

        public function waitForElNotVisibleOnPage($el,
$timePeriod = null)
        {
            try {
                return $this->getScenario()->runStep(new
\Codeception\Step\Action('waitForElementNotVisible',
func_get_args()));
            } catch (Exception $ex) {
                $msg = "возникла ошибка при ожидании исчезновения
элемента " . $el . " со страницы. (Функция
waitForElNotVisibleOnPage) \r\n";
                echo $msg;
                $this->comment($msg);
                return 1;
            }
        }

```

```

    }

    public function seeInSource($rawCode)
    {
        try {
            return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('seeInSource', func_get_args()));
        } catch (Exception $ex) {
            $msg = "возникла ошибка при поиске строки " .
$rawCode . " в коде страницы. (Функция seeInSource) \r\n";
            echo $msg;
            $this->comment($msg);
            return 1;
        }
    }

    public function switchToNextTab($offset = null)
    {
        try {
            return $this->getScenario()->runStep(new
\Codeception\Step\Action('switchToNextTab', func_get_args()));
        } catch (Exception $ex) {
            $msg = "возникла ошибка при переключении на
другую вкладку в браузере (Функция switchToNextTab) \r\n";
            echo $msg;
            $this->comment($msg);
            return 1;
        }
    }
}

```

```

public function closeTab()
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Action('closeTab', func_get_args()));
    } catch (Exception $ex) {
        $msg = "возникла ошибка при закрытии вкладки в
браузере (Функция closeTab) \r\n";
        echo $msg;
        $this->comment($msg);
        return 1;
    }
}

public function getCode()
{
    $fst = range('a', 'z');
    $sec = range('a', 'z');
    $thd = range('a', 'z');
    $frth = range('a', 'z');
    shuffle($fst);
    shuffle($sec);
    shuffle($thd);
    shuffle($frth);
    for ($i = 0; $i < count($fst); $i++) {
        $result[$i] = mt_rand(100, 999) . $fst[$i] .
$sec[$i] . $thd[$i] . $frth[$i];
    }
    return end($result);
}

```

```

public function getId()
{
    return mt_rand(100, 999) . mt_rand(100, 999);
}

public function seeTxtInEl($template, $locator)
{
    $this->comment('Проверка наличия маски "' . $template
. '"');

    $this->comment('в элементе "' . $locator . '"');
    $subj = $this->grabTextFrom($locator);
    preg_match('/' . $template . '/', $subj, $matches);
    if (count($matches) <= 0) {
        throw new Exception('Текст "' . $template . '" не
найден в указанном элементе.');
```

```

    }
}

public function
fillFieldWaitForDropDownMenuClickItem($fld, $dropItem)
{
    $I = $this;
    $I->fillField($fld, $dropItem);
    $I->waitForText($dropItem, '25', '.dropdown-menu');
    $I->click($dropItem, '.dropdown-menu');
}

public function waitForVisibleClickable($el, $timePeriod
= null)

```

```

    {
        $I = $this;
        $I->wait(1);
        $I->waitForElement($el, $timePeriod);
        $I->waitForElementVisible($el, $timePeriod);
        $I->waitForElementClickable($el, $timePeriod);
    }

    public function checkValue($nameVar, $el, $attr = null)
    {
        $I = $this;

        $I->comment('Проверка сгенерированного значения
параметра "' . $nameVar . '"');

        if ($attr == null) {
            $valueData = $I->grabTextFrom($el);
        } else {
            $valueData = $I->grabAttributeFrom($el, $attr);
        }

        if (strlen($valueData) == null) {
            throw new Exception("\"$nameVar\" не
отображается");
        } else {
            $I->comment("Значения параметра \"$nameVar\":
\"$valueData\"");
        }

        return $valueData;
    }

```

```
}
```

Приложение А.2

Файл AdministrationSectionTester.php

```
<?php
namespace Step\Acceptance\Administrator;

class AdministrationSectionTester extends \AcceptanceTester
{
    public function createAdvertAdmin($id)
    {
        $I = $this;
        $I->comment(' "Объявления" ');

        $I->comment(' "Настройки".
        $I->scrollTo('#btnChanecurity');
        $I->clickBtn('#btnCacurity');
        $I->waitForVisibleClickable('#dlFEditModal');
        $I-
>waitForVisibleClickable('*[@id=\'3\']*ains(@class,      \'entity-
action\']/input');
        $I->clickOnChk('*[@id=\'3\']*co(@class,      \'entity-
action\']/input');
        $I-
>waitForVisibleClickable('*[@id="tabDep/div/div[2]/button");
        $I->clickBtn('*[@id="tartment"]/div/div[2]/button');
        $I->clickOkBtn('#filter-n-ok');

        $I->comment(' Публикация объявления ');
```

```

$I->wait(5);

$I->clickBtn('Опубликовать');


$I->see('Статус: Опубликовано');
}

public function editAdvertAdmin($id)
{
    $I = $this;

    $I->comment('Редактирование "Объявления"');

    $I->comment('"Настройки доступа".');
    $I->scrollTo('#btnChangeSerity');
    $I->clickBtn('#btnChangecurity');
    $I->waitForVisibleClickable('#dlgFilterEditModal');

    $I-
>waitForVisibleClickable('*[@id=\'3\']*[con(@class,      \'entity-
action\')] /input');

    $I->clickOnChk('*[@id=\'3\']*[containsss,      \'entity-
action\')] /input');

    $I-
>waitForVisibleClickable('*[@id="tabDement"] /div/div[2] /button');

    $I->clickBtn('*[@id="tabDepartment"] /div[2] /button');

    $I->clickOkBtn('#filter-btn-ok');

    $I->click('.mce-menu-item.mce-menu-itormal.mce-stack-
layout-item.mce-first');

    $I->waitForVisibleClickable('input[ass="mce-textbox"
and @hidefocus="1" and @spellcheck="false"]');

    $I->fillField('input[@class="mce-textb@hidefocus="1"
and @spellcheck="lse"]', "Для организаций департамента доработка
по группам {$id}. ");

```

```

        $I->clickOkBtn('span[text()=\'\']');

        $I->see('Статус: Опубликовано');
    }
}

```

Приложение А.3

Файл GroupsTester.php

```

<?php

namespace Step\Acceptance\InternalUsers;

use Page\HeaderClass;

class GroupsTester extends \AcceptanceTester
{
    public function createStudyGroupDO($id, $groupName =
null)
    {
        $I = $this;
        $I->comment('ДО. СОЗДАНИЕ ГРУППЫ ОБУЧЕНИЯ');

        $date1 = date('d.m.Y');
        $date4 = date('d.m.Y', strtotime("+12 months"));

        $I->fillField('#ServClass_UnionCatalogServsOne_Name',
$id);

        $I->waitForText($id);
        $I->click($id);
    }
}

```

```

        if ($groupName == null) {
            $groupName = $id;
        }

        $I->fillField('#ServClass_Name', 'Автотестирование_'
. $groupName);

        $I->fillField('#ServClass_TrainStartDate', $date1);
        $I->fillField('#ServClass_TrainEndDate', $date4);
        $I->fillField('#ServClass_Capacity', '100');
        $I->fillField('#ServClass_Description', 'Это
автотест');

        $I->click('*[@id="btAddTeacher"]/i');
        $I->fillField('input[@class="tbTeacher span12"]',
'Дякова Елена Викторовна');

        $I->waitForText('Дякова Елена Викторовна');
        $I->click('Дякова Елена Викторовна');
        $I-
>click('*[@id="tableTeachers"]/tbody/tr/button[1]/i');

        $I->clickOnRadioBtn('input[@name="tGroup" and
@class="rbTeacherIsSupervisor"]');

        $I->click('#SavBtn');
    }

    public function excludeStudentFromGroupDO($id)
    {
        $I = $this;

        $I->comment(' ДО. ОТЧИСЛЕНИЕ ИЗ ГРУППЫ');

        $date1 = date('d.m.Y');

        $I->fillField('#Search_Query', $id);
    }

```

```

        $I->clickBtn('#servClassSearchBtn');

        $I-
>waitForElement('*[@id="divResultList"]/div[2]/div/table/tbody/tr[
1]/td[2]/a', 30);

        $I-
>click('*[@id="divResultList"]/div[2]/div/table/tbody/tr[1]/td[2]/
a');

        $I-
>waitForElement('*[@id="tablePupilList"]/tbody/tr/td[1]/label/inpu
t', 10);

        $I-
>waitForElementVisible('*[@id="tablePupilList"]/tbody/tr/td[1]/lab
el/input', 10);

        $I-
>clickOnChk('*[@id="tablePupilList"]/tbody/tr/td[1]/label/input');

        $I->waitForElementVisible('#btnExclude', 10);
        $I->clickBtn('#btnExclude');
        $I->waitForElement('#scWinExcludePupil', 10);
        $I->waitForElementVisible('#scWinExcludePupil', 10);
        $I->fillField('#scWinExcludePupilDocNumber', '45');
        $I->fillField('#scWinExcludePupilDocDate', $date1);
        $I->clickBtn('#scWinExcludePupilBtnOk');
        $I->clickOkBtn('#_globalAlertModalBtnOK');

        $I->scrollTo('#cbShowExcluded');
        $I->checkExcludedOrTransferredStudentOnForm();
    }

    public function recoverStudentToGroupDO($id)
    {
        $I = $this;

```

```

$I->comment(' ДО. ВОССТАНОВЛЕНИЕ ОБУЧАЮЩЕГОСЯ. ');

$date1 = date('d.m.Y');
$date2 = date('d.m.Y', strtotime("+1 year"));

$I->fillField('#Search_Query', $id);
$I->clickBtn('#servClassSearchBtn');
$I-
>waitForElement('*[@id="divResultList"]/div[2]/div/table/tbody/tr[
1]/td[2]/a', 10);
$I-
>click('*[@id="divResultList"]/div[2]/div/table/tbody/tr[1]/td[2]/
a');

$I->wait(3);
$I->scrollTo('#cbShowExcluded');
$I->clickOnChk('#cbShowExcluded');
$I-
>waitForElement('*[@id="tablePupilList"]/tbody/tr/td[1]/label/inpu
t', 10);
$I-
>waitForElementVisible('*[@id="tablePupilList"]/tbody/tr/td[1]/lab
el/input', 10);
$I-
>clickOnChk('*[@id="tablePupilList"]/tbody/tr/td[1]/label/input');
$I->waitForElementVisible('#btnRecover', 5);
$I->clickBtn('#btnRecover');
$I->waitForElement('#scWinRecoverPupil', 10);
$I->waitForElementVisible('#scWinRecoverPupil', 10);
$I->fillField('#tbDateLearnStart', $date1);
$I->fillField('#tbDateLearnEnd', $date2);

```

```

        $I->fillField('#scWinRecoverPupilComment', 'Тестовая
причина');

        $I->clickBtn('#scWinRecoverPupilOk');

        $I->wait(1);

        $I->waitForElement('#CheckAll', 5);

        $I->scrollTo('#CheckAll');

        $I-
>waitForElement('*[@id="tablePupilList"]/tbody/tr', 15);
    }

    public function combineStudentIntoSubgroupDO($id)
    {
        $I = $this;

        $I->comment(' ДО. ОБЪЕДИНЕНИЕ В ПОДГРУППУ. ');

        $I->fillField('#Search_Query', $id);

        $I->clickBtn('#servClassSearchBtn');

        $I-
>waitForElement('*[@id="divResultList"]/div[2]/div/table/tbody/tr[
1]/td[2]/a', 10);

        $I-
>click('*[@id="divResultList"]/div[2]/div/table/tbody/tr[1]/td[2]/
a');

        $I-
>waitForElement('*[@id="tablePupilList"]/tbody/tr/td[1]/label/inpu
t', 10);

        $I-
>waitForElementVisible('*[@id="tablePupilList"]/tbody/tr/td[1]/lab
el/input', 10);

        $I-
>clickOnChk('*[@id="tablePupilList"]/tbody/tr/td[1]/label/input');

```

```

$I->waitForElementVisible('#btnGroup', 10);
$I->clickBtn('#btnGroup');
$I->waitForElement('#scGroups', 10);
$I->waitForElementVisible('#scGroups', 10);
$I->fillField('#scGroupsNewName', 'Тестовое
наименование');
$I->clickBtn('#scGroupsBtnGroup');
$I->waitForElementNotVisible('#scGroups', 10);
$I-
>waitForElement('*[@id="tablePupilList"]/tbody/tr/td[3]/span',
10);

$I->see('Тестовое наименование',
'*[@id="tablePupilList"]/tbody/tr/td[3]/span');
}

public function
transferStudentToAnotherGroupDO($fstGroupName, $secGroupName)
{
    $I = $this;
    $I->comment(' ДО. ПЕРЕВОД В ДРУГУЮ ГРУППУ. ');

    $date1 = date('d.m.Y');

    $I->fillField('#Search_Query', $fstGroupName);
    $I->clickBtn('#servClassSearchBtn');
    $I-
>waitForElement('*[@id="divResultList"]/div[2]/div/table/tbody/tr[
1]/td[2]/a', 10);
    $I-
>click('*[@id="divResultList"]/div[2]/div/table/tbody/tr[1]/td[2]/
a');

```

```

        $I-
>waitForElement('*[@id="tablePupilList"]/tbody/tr/td[1]/label/input', 10);

        $I-
>waitForElementVisible('*[@id="tablePupilList"]/tbody/tr/td[1]/label/input', 10);

        $I-
>clickOnChk('*[@id="tablePupilList"]/tbody/tr/td[1]/label/input');

        $I->waitForElementVisible('#btnOtherClass', 10);
        $I->clickBtn('#btnOtherClass');
        $I->waitForElement('#scWinOtherClass', 10);
        $I->waitForElementVisible('#scWinOtherClass', 10);
        $I->chooseItemInDropDown('#scWinOtherClassId',
$secGroupName);

        $I->fillField('#scWinOtherClassDocNumber', '123');
        $I->fillField('#scWinOtherClassDocDate', $date1);
        $I->clickBtn('#scWinOtherClassBtnOk');
        $I->waitForElementNotVisible('#scWinOtherClass', 10);

        $I->waitForVisibleClickable('#cbShowExcluded');
        $I->checkExcludedOrTransferredStudentOnForm();
        $I->see('Переведен',
'*[@id="tablePupilList"]/tbody/tr/td[7]');
    }

```

```

public function transferGroupToArchiveDO($id)
{

```

```

    $I = $this;

```

```

    $I->comment(' ДО. ПЕРЕВОД ГРУППЫ В АРХИВ');

```

```

    $header = new \Page\HeaderClass($I);

```

```

        $I->fillField('#Search_Query', $id);
        $I->clickBtn('#servClassSearchBtn');
        $I-
>waitForElement('*[@id="divResultList"]/div[2]/div/table/tbody/tr[
1]/td[2]/a', 10);
        $I-
>click('*[@id="divResultList"]/div[2]/div/table/tbody/tr[1]/td[2]/
a');

        $I->waitForElement('#btnArchive', 10);
        $I->waitForElementVisible('#btnArchive', 10);
        $I->clickBtn('#btnArchive');
        $I->clickOkBtn('#_globalAlertModalBtnOK');

        $header-
>goToHeaderSection(HeaderClass::$headerSecName['Каталог групп
обучения']['Каталог групп обучения']);

        $I->fillField('#Search_Query', $id);
        $I->clickBtn('#servClassSearchBtn');
        $I->wait(3);
        $I-
>dontSeeElement('*[@id="divResultList"]/div[2]/div/table/tbody/tr[
1]/td[2]/a');

        $I->click('Расширенный поиск');
        $I->wait(1);
        $I->waitForElement('#Search_ShowArchive', 10);
        $I->waitForElementVisible('#Search_ShowArchive', 10);
        $I->clickOnChk('#Search_ShowArchive');
        $I->clickBtn('#servClassSearchBtn');

```

```

        $I->waitForElement('*[@id="divResultList"]/div[2]/div/table/tbody/tr[1]/td[2]/a', 10);
    }

    public function createGroupDK($id)
    {
        $I = $this;

        $I->comment(' ДК. СОЗДАНИЕ ГРУППЫ. ');

        $groupName = 'Группа_Автотестирование_' . $id;
        $I->waitForElement('#ServClass_Name');
        $I->fillField('#ServClass_Name', $groupName);
        $I->fillFieldWaitForDropdownMenuClickItem('#ServClass_Tea_FullName',
        'Музыкантов Сергей Петрович');
        $I->fillField('#ServClass_Capacity', '50');
        $I->fillField('#ServClass_Description', 'Это
автотест');
        $I->clickBtn('#SaveBtn');

        return $groupName;
    }

    public function createGroupSport($id)
    {
        $I = $this;
        $I->comment(' ОП. СОЗДАНИЕ ГРУППЫ. ');
    }

```

```

    $studyPeriodFrom = date('d.m.Y');
    $studyPeriodTo = date('d.m.Y', strtotime("+1 year - 1
day"));

    $applicationAdmissionDateFrom      =      date('d.m.Y',
strtotime("- 1 day"));
    $applicationAdmissionDateTo        =      date('d.m.Y',
strtotime("+ 11 months"));

    $I-
>fillFieldWaitForDropDownMenuClickItem('#ServClass_UnionCatalogSer
vsOne_Name', $id);

    $I->wait(2);

    $I-
>waitForElementClickable('#ServClass_ProgrammModuleId', 10);

    $I-
>chooseItemInDropDown('#ServClass_ProgrammModuleId', 'Основной');

    $I->fillField('#ServClass_Name', 'Автотестирование '
. $id);

    $I-
>chooseItemInDropDown('#ServClass_TrainingGroupId',
$applicationAdmissionDateFrom      .      '      -      '      .
$applicationAdmissionDateTo);

    $I->fillField('#ServClass_TrainStartDate',
$studyPeriodFrom);

    $I->fillField('#ServClass_TrainEndDate',
$studyPeriodTo);

    $I->fillField('#ServClass_Capacity', '50');

```

```
$I->fillField('#ServClass_Description',  
'Автотестирование');
```

```
$I->  
>clickOnRadioBtn('*[@id="tableTeachers"]/tbody/tr[1]/td[2]/input[1]');  
];
```

```
$I->clickBtn('#SaveBtn');
```

```
$I->wait(1);
```

```
$I->checkGeneratedValue('Код группы обучения',  
'input[@id="ServClass_Code" and @name="ServClass.Code" and  
@readonly="readonly"]', 'value');  
}
```

```
public function checkExcludedStudentOnPage()  
{  
    $I = $this;  
    $I->comment('Проверка отображения в блоке  
отчисленных/переведенных');
```

```
$showExcludedOrTransferredChk = '#cbExcluded';  
$shownStudents =  
'*[@id="tablePupilList"]/tbody/td[2]/a';
```

```
try {
```

```
    $I->wait('3');
```

```
    $I->clickOnChk($showExcludedOrTransferredChk);
```

```
    $I->waitForElement($shownStudents, 10);
```

```

        $I->seeElement($shownStudents);
    } catch (\Exception $ex) {

        $I->wait('6');
        $I->clickOnChk($showExcludedOrTransferredChk);
        $I->clickOnChk($showExcludedOrTransferredChk);
        $I->waitForElement($shownStudents, 10);
        $I->seeElement($shownStudents);
    }
}
}

```

Приложение В

Листинг PageObjects

Приложение В.1

Файл HeaderComponent.php

```
<?php
namespace Page;
class HeaderComponent
{
    protected $actor;
    public function __construct(\AcceptanceTester $I)
    {
        $this->actor = $I;
    }
    public static $paths = array(
        'Каталог заявлений' => '/Req/Srch',
        'Каталог услуг' => '/GroupCatalogServices/Srch',
        'Каталог образовательных услуг' =>
'/GroupCatalogServices/SrchP',
        'Каталог программ дополнительного образования' =>
'/GrCatalogServices/Srch',
        'Каталог занятий продленного дня' =>
'/GroupCatalogServices/Srch',
        'Каталог групп обучения' =>
'/ServiceGroup/ServiceGroupList',
        'Каталог организаций' => '/Org',
        'Каталог подведомственных организаций' =>
'/Org/VedOrgs',
```

```

        'Каталог аналитических отчетов' => array(
            'Сводка по городу' => '/Rep/CommR3',
            'Сводка по городу с учетом организаций' =>
'/Rep/RCity',
            'Сводка по типам программ обучения' =>
'/Rep/CommRe33',
            'Сводка по заявлениям на зачисление' =>
'/Rep/Retype=3',
            'Сводка по направленностям' => '/Rep/Comm4'
        )
    );

    public static $footerNameGroup = '.footerGroup';
    public static $headerNames = '[class="row header"]';
    public static $orangeBtn = '[id="btnainer"]';
    public static $errInformBlock = '[class="spanet1 well"]';
    public static $systemHeader = '[class="na-inner"]';
    public static $dropdownMenu = '[class="dropdown main-
menu"]';

    public static $container = '[id="container"]';
    public static $smallBtn = '[class="warn-btn"]';
    public static $menuBtn = 'button[class = "btn btn-navbar
topMenuBtn"]';

    public static $breadcrumbsName = '[class="breadcr"]';
    public static $exitBtn = '[id="exitbtn"]';

    public static $headline =
'/html/body/div[1]/div[2]/div/h3';

    public static $analyticsRepList =
'/html/body/div[1]/div[1]/div/ul/li[5]/ul';

    public static $admAnalyticsRepList =
'/html/body/div[1]/div/ul/li[8]/ul';

```

```

public static function
getHeadlineNameXPath($headlineName)
{
    return "h[text()='{$headlineName}]";
}
public function goToHeaderSection($headerSecName)
{
    $I = $this->actor;
    $pathArr = $this::$$paths;
    $msgBegin = 'Переход на страницу: "';
    $char = ' -> ';
    $msg = '""';
    if ($headerSecName) {
        $sec = explode("|", $headerSecName);
        $secNumber = count($sec);
        if ($secNumber == 1) {
            $hdrSec = $sec[0];

            $msg = $msgBegin . $hdrSec . $msg;
            $I->comment($msg);
            $I->amOnPage($pathArr[$hdrSec]);
        }
        if ($secNumber == 2) {
            $hdrSec = $sec[0];
            $hdrSubSection = $sec[1];
            $msg = $msgBegin . $hdrSec . $char .
$hdrSubSection . $msg;
            $I->comment($msg);
            $I->
amOnPage($pathArr[$hdrSec][$hdrSubSection]);

```

```

    }

    if ($secNumber == 3) {
        $hdrSec = $sec[0];
        $hdrSubSection = $sec[1];
        $subSubSection = $sec[2];

        $msg = $msgBegin . $hdrSec . $char .
$hdrSubSection . $char . $subSubSection . $msg;

        $I->comment($msg);

        $I-
>amOnPage($pathArr[$hdrSec][$hdrSubSection][$subSubSection]);
    }

    } else {
        throw new Exception('Указано некорректное
значение');
    }

}

public function goToPage($headerSecName)
{
    $I = $this->actor;
    $msgBegin = 'Переход на страницу: "';
    $char = ' -> ';
    $msg = '""';
    if ($headerSecName) {
        $sec = explode("|", $headerSecName);
        $secNumber = count($sec);
        if ($secNumber == 1) {
            $hdrSec = $sec[0];
            $msg = $msgBegin . $hdrSec . $msg;
            $I->comment($msg);
        }
    }
}

```

```

        if ($secNumber == 2) {
            $hdrSec = $sec[0];
            $hdrSubSection = $sec[1];
            $msg = $msgBegin . $hdrSec . $char .
$hdrSubSection . $msg;
            $I->comment($msg);
        }
    } else {
        throw new Exception('Указано некорректное
значение');
    }
    if ($secNumber == 2) {
        $I->wait(2);
        $I->click($hdrSec);
        $I->wait(2);
        $I->click($hdrSubSection);
    }
}

public function exit()
{
    $I = $this->actor;
    $I->comment('Выход из системы');
    $I->clickBtn('#btexit');
}

public static function selectRep($repName)
{
    return 'a[contains(text(), \'' . $repName . '\')]' ;
}

```

```

        public static $AdministrationSectionList =
'/html/body/div[1]/div[1]/div/div/div/ul';

        public static function getClickableRepName($repName)
        {
            return 'a[contains(text(),\'\' . $repName . '\')]';
        }

        public function toUserCataloguePage()
        {
            $I = $this->actor;

            $I->waitForTextOnPage('Администрирование
пользователями', 2);

            $I->click('Администрирование пользователями');

            $I->waitForTextOnPage('Пользователи системы', 3,
self::$AdministrationDropList);

            $I->clickBtn('Пользователи системы');

            $I->waitForTextOnPage('Управление пользователями
системы', 3);

            $I->waitForTextOnPage('Добавить пользователя
системы', 3);
        }
    }
}

```

Приложение В.2

Файл LoginClass.php

```

<?php

namespace Page;

class LoginClass
{
    protected $actor;
}

```

```

public function __construct(\AcceptanceTester $I)
{
    $this->actor = $I;
}

public function login($loginName, $serverLink =
ServerClass::TEST)
{
    $I = $this->actor;
    $I->comment('Авторизация в системе');
    $pwd = "qwertyQWERTY";
    $errorFlag = 0;
    foreach (RolesClass::$roleNamesArray as $key =>
$valueData) {
        if ($key == $loginName) {
            $I->
>am(RolesClass::$roleNamesArray[$loginName]);
        }
    }
    $I->maximizeWindow();
    $I->amOnUrl($serverLink);
    try {
        $I->see('Запись на программы дополнительного
образования');
    } catch (Exception $ex) {
        $I->reloadPage();
    }
    //Проверка наличия элементов на странице авторизации
    $errorFlag += $I->seeTextInPg('Запись на программы
дополнительного образования');
}

```

```

$errorFlag += $I->seeTextInPg('Логин:');
$errorFlag += $I->seeElOnPg('[name="loginName"]');
$errorFlag += $I->seeTextInPg('Пароль:');
$errorFlag += $I->seeElOnPg('[name="pwd"]');
$errorFlag += $I->seeTextInPg('Запомнить');
$errorFlag += $I->seeElOnPg('[name="createPersnCookie"]');
$errorFlag += $I->seeTextInPg('Войти',
'[class="warning-btn"]');
$errorFlag += $I->seeTextInPg('Войти с помощью ЭЦП',
'[id="Signbtn"]');
//Авторизация
$this->typeLoginName($loginName);
$this->typePwd($pwd);
$this->clickLogInBtn();
$errorFlag += $I->waitForTextOnPage('Запись на
программы дополнительного образования', 5,
HeaderClass::$headerWords);
$errorFlag += $I->waitForPgEl(HeaderClass::$exitButton, 5);
$I->dontSee('Логин:');
$I->dontSee('Пароль:');
$I->tryToClickOnBtn(HeaderClass::$menuButton);
return $errorFlag;
}

public function changepwd($loginName, $pwd, $newPwd =
'qwertyQWERTY')
{
    $I = $this;
    $actor = $this->actor;
    $I->typeLoginName($loginName);

```

```

        $I->typePwd($pwd);
        $I->clickLogInBtn();
        $I->typeNewPwd($newPwd);
        $I->typeNewPwdConfirmation($newPwd);
        $I->clickLogInBtn();

        $actor->waitForTextOnPage('Запись на программы
дополнительного образования', 10, HeaderComponent::$headerWords);

        $actor->dontSee('Логин:');
        $actor->dontSee('Пароль:');
        $actor->clickBtn('#btnexit');
        return $newPwd;
    }

    public function typeLoginName($loginName)
    {
        $I = $this->actor;
        $I->fillField('login', $loginName);
    }

    public function typePwd($pwd)
    {
        $I = $this->actor;
        $I->fillField('pwd', $pwd);
    }

    public function typeNewPwd($newPwd = 'qwertyQWERTY')
    {
        $I = $this->actor;
        $I->fillField('newpwd', $newPwd);
    }

    public function typeNewPwdConfirmation($newPwd =
'qwertyQWERTY')
    {

```

```

        $I = $this->actor;

        $I->fillField('confirmNewpwd', $newPwd);
    }

    public function clickLogInBtn()
    {
        $I = $this->actor;

        $I->clickBtn('Войти');
    }
}

```

Приложение В.3

Файл RolesClass.php

```

<?php
namespace Page;
class RolesClass
{
    const Admin = 'Admin';
    const AdmWorkWithSportDO = 'AdmWorkWithSportDO';
    const OperatorAdmDK = 'OperatorAdmDK';
    const OperatorAdmDO = 'OperatorAdmDO';
    const OperatorAdmSport = 'OperatorAdmSport';
    const OperatorOrgDK = 'OperatorOrgDK';
    const OperatorOrgDO = 'OperatorOrgDO';
    const OperatorOrgSport = 'OperatorOrgSport';

    public static $roleNamesArray = array(
        'Admin' => 'Администратор',
        'AdmWorkWithSportDO' => 'Администрации по работе с ОП
(ДО)',

```

```

        'OperatorAdmDK' => 'Оператор Администрации ДК',
        'OperatorAdmDO' => 'Оператор Администрации ДО',
        'OperatorAdmSport' => 'Оператор Администрации Деп.
спорта',

        'OperatorOrgDK' => 'Оператор Организации ДК',
        'OperatorOrgDO' => 'Оператор Организации ДО',
        'OperatorOrgSport' => 'Оператор Организации Деп.
спорта',

    );
}

```

Приложение В.4

Файл ServerClass.php

```

<?php
use Page\HeaderClass;
use Page\RolesClass as Role;
use Page\ServerClass;
/**
 * Inherited Methods
 * @method void wantToTest($textInf)
 * @method void wantTo($textInf)
 * @method void execute($callable)
 * @method void expectTo($prediction)
 * @method void expect($prediction)
 * @method void amGoingTo($argumentation)
 * @method void am($role)
 * @method void lookForwardTo($achieveValue)
 * @method void comment($description)

```

```

        * @method \Codeception\Lib\Friend haveFriend($nameVar,
$actorClass = NULL)
    *
    * @SuppressWarnings(PHPMD)
*/
class AcceptanceTester extends \Codeception\Actor
{
    use _generated\AcceptanceTesterActions;

    public function clickOkBtn($id, $contextData = null)
    {
        $I = $this;

        $I->comment('Нажатие на кнопку "OK" в модальном
окне');

        $I->wait('2');
        $I->waitForVisibleClickable($id, 7);
        $I->wait('2');
        $I->clickBtn($id, $contextData = null);
        $I->wait('2');
        if (!$I->seeElOnPg($id)) {
            $I->clickBtn($id, $contextData = null);
            if (!$I->seeElOnPg($id)) {
                $I->clickBtn($id, $contextData = null);
            }
        }
        $I->waitForElementNotVisible($id, 15);
    }

    public function tryToClickOnBtn($linkSource, $contextData
= null)
    {
        try {

```

```

        return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('click', func_get_args()));
    } catch (Exception $ex) {
        return 1;
    }
}

public function clickBtn($linkSource, $contextData =
null)
{
    return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('click', func_get_args()));
}

public function clickOnRadioButton($linkSource,
$contextData = null)
{
    return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('click', func_get_args()));
}

public function clickOnCheckbox($linkSource, $contextData
= null)
{
    return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('click', func_get_args()));//dropdown
}

public function chooseItemInDropDown($choose, $opt)
{
    return $this->getScenario()->runStep(new
\Codeception\Step\Action('selectOption', func_get_args()));
}

public function acceptanceTestName($nameVar)
{

```

```

        $this->wantTo($nameVar);

        $this->comment("Название теста: " . $nameVar);
    }

    public function seeTextInPg($subj, $locator = null)
    {
        try {
            return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('see', func_get_args()));
        } catch (Exception $e) {
            $msg = "Надпись - " . $subj . " - не найдена на
странице. (Функция seeTextInPg) \r\n";

            echo $msg;

            $this->comment($msg);

            return 1;
        }
    }

    public function seeElOnPg($subj, $attrs = null)
    {
        try {
            return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('seeElement', func_get_args()));
        } catch (Exception $ex) {
            $msg = "Элемент " . $subj . " не найден на
странице. (Функция seeElOnPg) \r\n";

            echo $msg;

            $this->comment($msg);

            return 1;
        }
    }
}

```

```

        public function clickOnElement($linkSource, $contextData
= null)
        {
            try {
                return $this->getScenario()->runStep(new
\Codeception\Step\Action('click', func_get_args()));
            } catch (Exception $ex) {
                $msg = "Ошибка при нажатии на элемент " .
$linkSource . ". (Функция clickOnElement) \r\n";
                echo $msg;
                $this->comment($msg);
                return 1;
            }
        }

        public function fillFieldOnPg($fld, $valueData)
        {
            try {
                return $this->getScenario()->runStep(new
\Codeception\Step\Action('fillField', func_get_args()));
            } catch (Exception $ex) {
                $msg = "Невозможно заполнить поле " . $fld . ".
(Функция fillFieldOnPg)\r\n";
                echo $msg;
                $this->comment($msg);
                return 1;
            }
        }

        public function waitForPgEl($el, $timePeriod = null)
        {
            try {

```

```

        return $this->getScenario()->runStep(new
\Codeception\Step\Action('waitForElement', func_get_args()));
    } catch (Exception $ex) {
        $msg = "Элемент " . $el . " отсутствует на
странице, время ожидания истекло. (Функция waitForPgEl) \r\n";

        echo $msg;

        $this->comment($msg);

        return 1;
    }
}

public function waitForTextOnPage($textInf, $timePeriod =
null, $locator = null)
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Action('waitForText', func_get_args()));
    } catch (Exception $ex) {
        $msg = "Текст " . $textInf . " отсутствует на
странице, время ожидания истекло. (Функция waitForTextOnPage)
\r\n";

        echo $msg;

        $this->comment($msg);

        return 1;
    }
}

public function checkOptionInPage($opt)
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Action('checkOption', func_get_args()));
    } catch (Exception $ex) {

```

```

        $msg = "ошибка при установке радиобаттона или
чекбокса" . $opt . ". (Функция checkOptionInPage) \r\n";
        echo $msg;
        $this->comment($msg);
        return 1;
    }
}

public function chooseOptionOnPage($choose, $opt)
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Action('selectOption', func_get_args()));
    } catch (Exception $ex) {
        $msg = "ошибка при установке значения в
выпадающем списке или радиобаттона" . $opt . ". (Функция
chooseOptionOnPage) \r\n";
        echo $msg;
        $this->comment($msg);
        return 1;
    }
}

public function waitForElVis($el, $timePeriod = null)
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Action('waitForElementVisible',
func_get_args()));
    } catch (Exception $ex) {
        $msg = "возникла ошибка при ожидании появления
элемента " . $el . " на странице. (Функция waitForElVis) \r\n";
        echo $msg;

```

```

        $this->comment($msg);
        return 1;
    }
}

public function waitForElNotVisibleOnPage($el,
$timePeriod = null)
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Action('waitForElementNotVisible',
func_get_args()));
    } catch (Exception $ex) {
        $msg = "возникла ошибка при ожидании исчезновения
элемента " . $el . " со страницы. (Функция
waitForElNotVisibleOnPage) \r\n";
        echo $msg;
        $this->comment($msg);
        return 1;
    }
}

public function seeInSource($rawCode)
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Assertion('seeInSource', func_get_args()));
    } catch (Exception $ex) {
        $msg = "возникла ошибка при поиске строки " .
$rawCode . " в коде страницы. (Функция seeInSource) \r\n";
        echo $msg;
        $this->comment($msg);
        return 1;
    }
}

```

```

    }
}

public function switchToNextTab($offset = null)
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Action('switchToNextTab', func_get_args()));
    } catch (Exception $ex) {
        $msg = "возникла ошибка при переключении на
другую вкладку в браузере (Функция switchToNextTab) \r\n";
        echo $msg;
        $this->comment($msg);
        return 1;
    }
}

public function closeTab()
{
    try {
        return $this->getScenario()->runStep(new
\Codeception\Step\Action('closeTab', func_get_args()));
    } catch (Exception $ex) {
        $msg = "возникла ошибка при закрытии вкладки в
браузере (Функция closeTab) \r\n";
        echo $msg;
        $this->comment($msg);
        return 1;
    }
}

public function getCode()
{

```

```

$first = range('a', 'z');
$sec = range('a', 'z');
$thd = range('a', 'z');
$frth = range('a', 'z');
shuffle($first);
shuffle($sec);
shuffle($thd);
shuffle($frth);

for ($i = 0; $i < count($first); $i++) {
    $result[$i] = mt_rand(100, 999) . $first[$i] .
$sec[$i] . $thd[$i] . $frth[$i];
}

return end($result);
}

public function getId()
{
    return mt_rand(100, 999) . mt_rand(100, 999);
}

public function seeTxtInEl($template, $locator)
{
    $this->comment('Проверка наличия маски "' . $template
. '"');

    $this->comment('в элементе "' . $locator . '"');
    $subj = $this->grabTextFrom($locator);
    preg_match('/' . $template . '/', $subj, $matches);
    if (count($matches) <= 0) {
        throw new Exception('Текст "' . $template . '" не
найден в указанном элементе.');
```

```

public function
fillFieldWaitForDropDownMenuClickItem($fld, $dropItem)
{
    $I = $this;
    $I->fillField($fld, $dropItem);
    $I->waitForText($dropItem, '25', '.dropdown-menu');
    $I->click($dropItem, '.dropdown-menu');
}

public function waitForVisibleClickable($el, $timePeriod
= null)
{
    $I = $this;
    $I->wait(1);
    $I->waitForElement($el, $timePeriod);
    $I->waitForElementVisible($el, $timePeriod);
    $I->waitForElementClickable($el, $timePeriod);
}

public function checkValue($nameVar, $el, $attr = null)
{
    $I = $this;
    $I->comment('Проверка сгенерированного значения
параметра "' . $nameVar . '"');
    if ($attr == null) {
        $valueData = $I->grabTextFrom($el);
    } else {
        $valueData = $I->grabAttributeFrom($el, $attr);
    }
    if (strlen($valueData) == null) {

```

```

        throw new Exception("\"$nameVar\" не
отображается");
    } else {
        $I->comment("Значения параметра \"$nameVar\":
\"$valueData\");
    }
    return $valueData;
}
}

```

Приложение С

Листинг Value Object

```
<?php

namespace ValueObjects\Administrator\AdministrationSection;

use Page\HeaderClass;

use
Enums\Administrator\AdministrationSection\CreateUser\RightsEnum as
Rights;

use
Enums\Administrator\AdministrationSection\CreateUser\RolesEnum as
Roles;

use
Enums\Administrator\AdministrationSection\CreateUser\StatusEnum as
Status;

use
Enums\Administrator\AdministrationSection\CreateUser\EmailNotifica
tionsEnum as EmailNotifications;

use
Enums\Administrator\AdministrationSection\CreateUser\TypeEnum as
Type;

Class UserValueObj
{
    protected $actor;

    public $userId;
    public $org;
    public $fioData;
    public $loginName;
    public $emailAddress;
```

```

public $userStatus;
public $userType;
public $pwd;
public $newPwd;
public $userRights = [];
public $userRole;
public $emailAddressNotifications = [];

function __construct(\AcceptanceTester $I, $userId =
null, $userInfo = null)
{
    $this->actor = $I;
    $this->userId = $userId;
    if ($userInfo != null) {
        foreach ($userInfo as $key => $value) {
            if ($key == 'fio' || $key == 'loginName') {
                $this->$key = $value . $this->userId;
            } elseif ($key == 'emailAddress') {
                $this->$key = $this->userId . $value;
            } else {
                $this->$key = $value;
            }
        }
    }
}

public function clickSaveBtn()
{
    $I = $this->actor;
    $I->comment(' + Нажатие на кнопку "Сохранить"');
    $I->wait(1);
    $I->waitForPageEl('#SaveBtn');

```

```

        $I->waitForElVis('#SaveBtn');
        $I->clickBtn('Сохранить');
    }
    public function typeOrg($org)
    {
        $I = $this->actor;
        $I->comment(' + Заполнение поля "Организация"');
        $I->fillField('#Organization', $org);
        $I->waitForTextOnPage($org, 35);
        $I->click($org);
    }
    public function typeFioData($fioData)
    {
        $I = $this->actor;
        $I->comment(' + Заполнение поля "ФИО"');
        $I->fillField('#User_Name_Field', $fioData);
    }
    public function typeLoginName($loginName)
    {
        $I = $this->actor;
        $I->comment(' + Заполнение поля "Логин"');
        $I->fillField('#User_Login_Field', $loginName);
    }
    public function typeEmailAddress($emailAddress)
    {
        $I = $this->actor;
        $I->comment(' + Заполнение поля "Email"');
        $I->fillField('#User_Email_Field', $emailAddress);
    }
}

```

```

public function chooseStatus($userStatus)
{
    $I = $this->actor;
    if ($userStatus != null) {
        $I->comment(' + Выбор статуса в выпадающем списке
"Статус");
        $I->chooseItemInDropDown('#User_StatusId_Field',
$userStatus);
    }
}

public function chooseType($userType)
{
    $I = $this->actor;
    if ($userType != null) {
        $I->comment(' + Выбор вида пользователя в
выпадающем списке "Вид пользователя");
        $I->chooseItemInDropDown('#User_TypeId_Field',
$userType);
    }
}

public function setNewPwd($newPwd)
{
    $I = $this;
    $I->typeNewPwd($newPwd);
    $I->typeNewPwdConfirmation($newPwd);
    $I->clickSaveBtn();
}

public function typeNewPwd($newPwd)
{
    $I = $this->actor;

```

```

        $I->fillField('NewPwd', $newPwd);
    }
    public function typeNewPwdConfirmation($newPwd)
    {
        $I = $this->actor;
        $I->fillField('NewPwdConfirm', $newPwd);
    }
    public function generateNewPwd()
    {
        $I = $this->actor;
        $I->comment(' + Генерация нового пароля');
        $I->clickBtn('Сгенерировать пароль');
        $I->waitForPgEl('#alertModal');
        $I->waitForElVis('#alertModal');
        $I->waitForPgEl('#alertModalBtnOK');
        $I->waitForElVis('#alertModalBtnOK');
        $I->wait(2);
        $pwd = $this->copyPwdFromModal();
        $I->clickBtn('#alertModalBtnOK');
        $I->wait(1);
        return $pwd;
    }
    public function copyPwdFromModal()
    {
        $I = $this->actor;
        $text = $I->grabTextFrom('//*[@userId="_AlertModal"]');
        list($begin, $pwd, $end) = explode('"', $text);
        return $pwd;
    }

```

```

public function chooseUserRights($userRights)
{
    $I = $this->actor;
    if ($userRights != null) {
        $I->comment('      +      Выбор      необходимых      прав
пользователя');
        if (is_array($userRights)) {
            foreach ($userRights as $value) {
                $textInfId          =          $I->
>grabAttributeFrom('//input[@class="accessName" and @value="' .
$value . '"]', 'Id');
                $chkboxId    =    substr_replace($textInfId,
'Checked', -11);
                $I->clickOnCheckbox('#' . $chkboxId);
            }
        } else {
            throw new \Exception('Передан не массив');
        }
    }
}

public function chooseUserRole($userRole)
{
    $I = $this->actor;
    $I->comment(' + Выбор роли пользователя');
    $I->clickBtn('#btnAdd');
    $I->waitForPgEl('#userRoleWin');
    $I->waitForElVis('#userRoleWin');
    $I->chooseItemInDropDown('#userRoleDropId',
$userRole);
}

```

```

        $I->wait(1);

        $I->clickBtn('#btnInsert');

        $I->wait(1);

    }

    public function
typeNotifications($emailAddressNotifications)
    {

        $I = $this->actor;

        if ($emailAddressNotifications != null) {

            $I->comment(' + Выбор необходимых параметров в
разделе "Управление почтовыми уведомлениями"');

            if (is_array($emailAddressNotifications)) {

                foreach ($emailAddressNotifications as
$value) {

                    $I->clickOnCheckbox($value);

                }

            } else {

                throw new \Exception('Передан не массив');

            }

        }

    }

    public function formUserData($userObj, $userInfo = null)
    {

        $actor = $this->actor;

        $userObj->userId = $actor->getId();

        if ($userInfo == null)
        {

            $userObj->org = 'тестовая орг-ция';

```

```

        $userObj->fio    =    'Автотестирование    '.$userObj->
>userId;

        $userObj->loginName    =    'testUser'.$userObj->
>userId;

        $userObj->emailAddress    =    $userObj->
>userId.'@test.ru';
    }
    else
    {
        if ($userInfo != null)
        {
            foreach ($userInfo as $key => $value)
            {
                if ($key == 'fio' || $key == 'loginName')
                {
                    $userObj->$key    =    $value.$userObj->
>userId;

                }
                elseif ($key == 'emailAddress')
                {
                    $userObj->$key    =    $userObj->
>userId.$value;

                }
            }
        }
    }
}

```

```

        return $userObj->userId;
    }
    public function createUserWithRole($userObj)
    {
        $I = $this;
        $A = $this->actor;
        if ($userObj->org && $userObj->fio && $userObj->loginName && $userObj->emailAddress != null) {
            $A->comment(' + Нажатие на кнопку "Добавить пользователя"');
            $A->clickBtn('Добавить пользователя');
            $I->typeOrg($userObj->org);
            $I->typeFioData($userObj->fio);
            $I->typeLoginName($userObj->loginName);
            $I->typeEmailAddress($userObj->emailAddress);
            $I->chooseStatus($userObj->userStatus);
            $I->chooseType($userObj->userType);
            $pwd = $I->generateNewPwd();
            $I->chooseUserRights($userObj->userRights);
            $I->chooseUserRole($userObj->userRole);
            $I->typeEmailAddressNotifications($userObj->emailAddressNotifications);
            $I->clickSaveBtn();
            return $pwd;
        } else {
            throw new \Exception('Не заполнены переменные для обязательных полей: Организации ($org), ФИО ($fioData), Логина ($loginName), Email ($emailAddress)');
        }
    }
}

```

```

public function checkUserCreationPgElements()
{
    $errorFlag = 0;

    $I = $this->actor;

    $I->comment('      -- Проверка наличия хлебных крошек,
названия раздела на странице и кнопок');

    $errorFlag      +=      $I->seeTextInPg('Главная',
HeaderClass::$breadcrumbs);

    $errorFlag      +=      $I->seeTextInPg('Администрирование',
HeaderClass::$breadcrumbs);

    $errorFlag      +=      $I->seeTextInPg('Редактирование
пользователя', HeaderClass::$breadcrumbs);

    $errorFlag      +=      $I->seeTextInPg('Пользователь',
HeaderClass::$headlineName);

    $errorFlag += $I->seeTextInPg('История изменений');
    $errorFlag += $I->seeTextInPg('Сохранить');
    $errorFlag += $I->seeTextInPg('Удалить');
    $errorFlag += $I->seeTextInPg('Вернуться');

    $I->comment('      -- Проверка раздела "Основные
данные пользователя"');

    $errorFlag      +=      $I->seeTextInPg('Основные      данные
пользователя');

    $errorFlag += $I->seeTextInPg('Организация*');
    $errorFlag += $I->seeElOnPg('#Organization');
    $errorFlag += $I->seeTextInPg('ФИО*');
    $errorFlag += $I->seeElOnPg('#User_Name_Field');
    $errorFlag += $I->seeTextInPg('Логин*');
    $errorFlag += $I->seeElOnPg('#User_Login_Field');
    $errorFlag += $I->seeTextInPg('Email*');
    $errorFlag += $I->seeElOnPg('#User_Email_Field');

```

```

$errorFlag += $I->seeTextInPg('Статус:');
$errorFlag += $I->seeElOnPg('#User_StatusId_Field');
$errorFlag += $I->seeTextInPg('Вид пользователя:');
$errorFlag += $I->seeElOnPg('#User_TypeId_Field');
$I->comment('      -- Проверка раздела "Пароль"');
$errorFlag += $I->seeTextInPg('Пароль:');
$errorFlag += $I->seeTextInPg('Новый пароль*:');
$errorFlag += $I->seeElOnPg('#NewPwd');
$errorFlag      +=      $I->seeTextInPg('Подтверждение
пароля*:');

$errorFlag += $I->seeElOnPg('#NewPwdConfirm');
$errorFlag += $I->seeElOnPg('#passStrength');
$errorFlag      +=      $I->seeTextInPg('Сгенерировать
пароль');

$errorFlag      +=      $I->seeTextInPg('Пароль      должен
соответствовать правилам:');

$errorFlag      +=      $I->seeTextInPg('должен      содержать
только буквы латинского алфавита');

$I->comment('      -- Проверка раздела "Права
пользователя" и проставление всех чекбоксов');

$errorFlag += $I->seeTextInPg('Права пользователя');
$errorFlag += $I->seeElOnPg('#chooseAccess');
$errorFlag += $I->seeTextInPg('№', '#userAccessId');
$errorFlag      +=      $I->seeTextInPg('Наименование',
'#userAccessId');

$I->comment('      -- Проверка раздела "Роль
пользователя" и списка прав');

$errorFlag += $I->seeTextInPg('Роль пользователя');
$errorFlag      +=      $I->seeTextInPg('№',
'#userRoleTableId');

```

```

        $errorFlag += $I->seeTextInPg('Наименование',
'#userRoleTableId');

        $I->comment('        -- Проверка раздела "Сертификаты
подписей" и модального окна');

        $errorFlag += $I->seeTextInPg('Описание');

        $errorFlag += $I->seeTextInPg('Окончание действия');

        $I->comment('        -- Проверка раздела "Управление
уведомлениями" и проставление чекбоксов');

        $errorFlag += $I->seeTextInPg('Управление
уведомлениями (Показать/скрыть)');

        $I->clickBtn(' (Показать/скрыть) ');

        $errorFlag += $I->waitForTextOnPage('Получать
отмеченные уведомления:', 8);

        $errorFlag += $I->seeTextInPg('Получать отмеченные
уведомления:');

        $errorFlag += $I->seeElOnPg('#checkEvents');

        $errorFlag += $I->seeTextInPg('№', '.table.table-
striped.table-condensed.table-hover');

        $errorFlag += $I->seeTextInPg('Наименование',
'.table.table-striped.table-condensed.table-hover');

        $errorFlag += $I->seeElOnPg('#User_IsReceiveEvent1');

        $errorFlag += $I->seeTextInPg('1',
'.table.condensed.table-hover');

        $errorFlag += $I->seeTextInPg('Уведомление о новых
записях на тестирование', '.table.table-striped.table-
condensed.table-hover');

        $errorFlag += $I->seeTextInPg('В начало');

        $errorFlag += $I->seeTextInPg(HeaderClass::$footer,
HeaderClass::$footer);

        return $errorFlag;
    }

```

```

public function fullCheckUserCreationPageElements()
{
    $errorFlag = 0;

    $I = $this->actor;

    $I->comment('      -- Проверка наличия хлебных крошек,
названия раздела на странице и кнопок');

    $errorFlag      +=      $I->seeTextInPg('Главная',
HeaderClass::$breadcrumbs);

    $errorFlag      +=      $I->seeTextInPg('Администрирование
пользователей', HeaderClass::$breadcrumbs);

    $errorFlag      +=      $I->seeTextInPg('Редактирование
пользователя', HeaderClass::$breadcrumbs);

    $errorFlag      +=      $I->seeTextInPg('Пользователь',
HeaderClass::$headlineName);

    $errorFlag += $I->seeTextInPg('История изменений');
    $errorFlag += $I->seeTextInPg('Сохранить');
    $errorFlag += $I->seeTextInPg('Удалить');
    $errorFlag += $I->seeTextInPg('Вернуться');

    $I->comment('      -- Проверка раздела "Основные
данные пользователя"');

    $errorFlag      +=      $I->seeTextInPg('Основные      данные
пользователя');

    $errorFlag += $I->seeTextInPg('Организация*');
    $errorFlag += $I->seeElOnPg('#Organization');
    $errorFlag += $I->seeTextInPg('ФИО*');
    $errorFlag += $I->seeElOnPg('#User_Name_Field');
    $errorFlag += $I->seeTextInPg('Логин*');
    $errorFlag += $I->seeElOnPg('#User_Login_Field');
    $errorFlag += $I->seeTextInPg('Email*');
    $errorFlag += $I->seeElOnPg('#User_Email_Field');
    $errorFlag += $I->seeTextInPg('Статус');

```

```

$errorFlag += $I->seeElOnPg('#User_StatusId_Field');
$this->chooseStatus(Status::Blocked_Status);
$this->chooseStatus(Status::Active_Status);
$errorFlag += $I->seeTextInPg('Вид пользователя:');
$errorFlag += $I->seeElOnPg('#User_TypeId_Field');
$this->chooseType(Type::Administrator_User);
$this->chooseType(Type::External_type_User);
$this->chooseType(Type::Internal_type_User);
$I->comment('      -- Проверка раздела "Пароль"');
$errorFlag += $I->seeTextInPg('Пароль:');
$errorFlag += $I->seeTextInPg('Новый пароль*:');
$errorFlag += $I->seeElOnPg('#NewPwd');
$errorFlag      +=      $I->seeTextInPg('Подтверждение
пароля*:');

$errorFlag += $I->seeElOnPg('#NewPwdConfirm');
$errorFlag += $I->seeElOnPg('#passStrength');
$errorFlag      +=      $I->seeTextInPg('Сгенерировать
пароль');

$errorFlag      +=      $I->seeTextInPg('Пароль      должен
соответствовать правилам:');

$errorFlag      +=      $I->seeTextInPg('должен      содержать
только буквы латинского алфавита');

$I->comment('      -- Проверка заполнения полей "Новый
пароль" и "Подтверждение пароля" и генерации нового пароля');
$errorFlag += $I->fillFieldOnPg('#NewPwd', '12345');
$I->clearField('#NewPwd');
$errorFlag      +=      $I->fillFieldOnPg('#NewPwdConfirm',
'12345');

$I->clearField('#NewPwdConfirm');
$I->clickBtn('#genPwdBtn');

```

```

$errorFlag += $I->waitForElVis('#alertModal', 6);
$errorFlag += $I->waitForPgEl('#alertModalBtnOK');
$errorFlag += $I->waitForElVis('#alertModalBtnOK');
$I->wait(2);
$errorFlag += $I->seeTextInPg('Пароль сгенерирован');
$errorFlag += $I->seeElOnPg('.close_');
$errorFlag += $I->seeTextInPg('Сгенерирован новый
пароль :');

$errorFlag += $I->seeTextInPg('Пароль будет
скопирован в буфер обмена');
$I->clickBtn('#alertBtnOK');
$I->wait(1);
$I->comment('          -- Проверка раздела "Права
пользователя" и проставление всех чекбоксов');
$errorFlag += $I->seeTextInPg('Права пользователя');
$errorFlag += $I->seeElOnPg('#chooseAccess');
$errorFlag += $I->seeTextInPg('№', '#userTableId');
$errorFlag += $I->seeTextInPg('Наименование',
'#userTableId');
$I->clickOnCheckbox('#chooseAll');
foreach (Rights::$Rights as $nameVar => $value) {
    $textInfId = $I->grabAttributeFrom('//input[@class="accessName" and @value="' .
$value . '"]', 'userId');
    $checkboxId = substr_replace($textInfId,
'Chcked', -11);
    $I->clickOnCheckbox('#' . $checkboxId);
}
$I->comment('          -- Проверка раздела "Роль
пользователя" и списка прав');
$errorFlag += $I->seeTextInPg('Роль пользователя');

```

```

$ErrorFlag += $I->seeTextInPg('№',
'#userRoleTableId');

$ErrorFlag += $I->seeTextInPg('Наименование',
'#userRoleTableId');

$I->clickBtn('#btnAddRole');
$ErrorFlag += $I->waitForPgEl('#userRoleAddWin');
$ErrorFlag += $I->waitForElVis('#userRoleAddWin');
$ErrorFlag += $I->seeTextInPg('Добавление роли');
$ErrorFlag += $I->seeTextInPg('Роль:');
$ErrorFlag += $I->seeElOnPg('.close__');
foreach (Roles::$Roles as $nameVar => $value) {
    $I->chooseItemInDropDown('#userRoleDropId',
$value);
}
$ErrorFlag += $I->seeElOnPg('#btnInsertRole');

$I->clickBtn('#btnInsertRole');
$I->wait(1);
$I->comment('        -- Проверка раздела "Сертификаты
электронных подписей" и модального окна');
$ErrorFlag += $I->seeTextInPg('Описание');
$ErrorFlag += $I->seeTextInPg('Окончание действия');
$I->clickBtn('Добавить данные');
$ErrorFlag += $I->waitForPgEl('#editWin');
$ErrorFlag += $I->waitForElVis('#editWin');
$ErrorFlag += $I->waitForPgEl('#SaveBtn');
$ErrorFlag += $I->waitForElVis('#SaveBtn');
$ErrorFlag += $I->seeTextInPg('Добавление/редактирование ', '.modal-header');
$ErrorFlag += $I->seeElOnPg('.close__');

```

```

$errorFlag += $I->seeTextInPg('Выбрать файл ');
$I->clickBtn('Отмена');
$I->wait(1);
$I->comment('          -- Проверка раздела "Управление
уведомлениями" и проставление чекбоксов');
$errorFlag += $I->seeTextInPg('Управление
уведомлениями (Показать/скрыть) ');
$I->clickBtn(' (Показать/скрыть) ');
$errorFlag += $I->seeTextInPg('Получать отмеченные
уведомления: ');
$errorFlag += $I->seeElOnPg('#checkEvs');
$errorFlag += $I->seeTextInPg('№', '.tabletable-
condensed.table-hover');
$errorFlag += $I->seeTextInPg('Наименование',
'.tabletable-condensed.table-hover');
$errorFlag += $I->seeElOnPg('#User_IsReceive');
$errorFlag += $I->seeTextInPg('1', '.table.table-
condensed.table-hover');
$errorFlag += $I->seeTextInPg('Уведомление о новых
записях ', '.table.table-striped.table-condensed.table-hover');
$errorFlag += $I->seeTextInPg('В начало');
$errorFlag += $I->seeTextInPg(HeaderClass::$footer,
HeaderClass::$footer);
return $errorFlag;
}
}

```

Приложение D

Тест «Зачисление»

```
<?php

use Page\ServerClass;
use Page\RolesClass as Role;
use Page\HeaderClass;

class DO_Enroll_Cest
{
    public function enrollApplicationDO(AcceptanceTester $I,
\Page\HeaderClass $header, Page\LoginClass $loginNamePage,

\Step\Acceptance\InternalUsers\ServiceCatalogueTester
$serviceCatalogueTester,

\Step\Acceptance\InternalUsers\GroupsTester
$groupsCatalogueTester,

\Step\Acceptance\InternalUsers\ApplicationCatalogueTester
$applicationCatalogueTester)
    {
        $I->acceptanceTestName('ДО. ЗАЧИСЛЕНИЕ');
        $id = $I->getUniqueId();
        //ДО. СОЗДАНИЕ УСЛУГИ.
        $loginNamePage->login(Role::OperatorOrgDO);
        $header->
>goToPage(HeaderClass::$headerSectionNames['Каталог услуг']);
        $I->clickBtn('Добавить услугу');
```

```

        $serviceCatalogueTester-
>createChildAssociationDO($id);

        //ДО. ПУБЛИКАЦИЯ УСЛУГИ.

        $serviceCatalogueTester->publishChildAssociationDO();

        //ДО. СОЗДАНИЕ ГРУППЫ

        $header-
>goToPage(HeaderClass::$headerSectionNames['Каталог          группы
обучения'] ['Каталог группы обучения']);

        $I->clickBtn('Добавить группу');

        $groupsCatalogueTester->createStudyGroupDO($id);

        //ДО. ПОДАЧА ЗАЯВЛЕНИЯ.

        $header-
>goToPage(HeaderClass::$headerSectionNames['Каталог услуг']);

        $I->fillField('#Query_field', $id);

        $I->clickBtn('#btnSch');

        $I->click('//*[ @id="formSrch"]/div[2]/div/p/a/span');

        $request          =          $applicationCatalogueTester-
>makeApplicationDO();

        //ДО. ЗАЧИСЛЕНИЕ УЧЕНИКА

        $header-
>goToPage(HeaderClass::$headerSectionNames['Каталог заявлений']);

        $I->fillField('#SearchParam', $request);

        $I->click('#SearchBtn');

        $I->click('//*[ @id="requestsGr"]/tbody/td[1]/a');

        $applicationCatalogueTester->enrollApplicationDO();

        $header->exit();

        $I->comment('ЗАВЕРШЕНИЕ ПРОВЕРКИ');

    }

}

```

Приложение Е

Презентация

Система функционального автотестирования интернет-сервиса для образовательных организаций

Автор работы: магистрантка группы 18ВВм1 Пучкова У.Н.
Научный руководитель: к.т.н., доцент Синев М.П.

Пенза, 2020

Цель и объект работы

Цель работы: разработать программную систему для реализации функциональных проверок интернет-сервиса для образовательных организаций.

Объект работы: система автоматизированного тестирования, обеспечивающая следующие возможности:

- возможность реализовывать функциональное тестирование интернет-сервиса;
- возможность проведения приемочного тестирования интернет-сервиса с помощью автоматизированных проверок;
- возможность генерации отчета о прохождении тестов;

Функциональное тестирование

Под функциональным тестированием понимается процесс установки соответствия разработанного программного обеспечения функциональным требованиям, поставленными заказчиками, для решения определенных задач пользователей.

К основным этапам функционального тестирования относятся:

- подготовка;
- проведение;
- отчет.

3

Виды функционального тестирования



4

Стек разработки



PhpStorm



Codeception

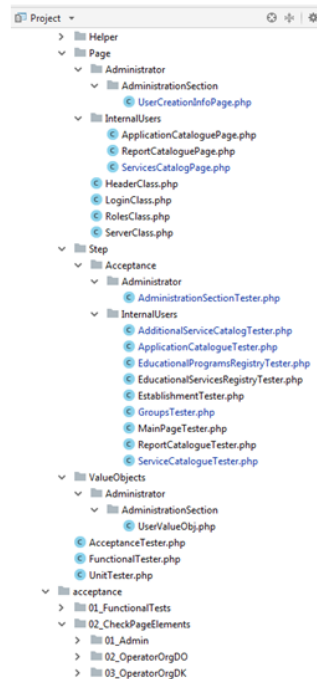
5

UML-диаграмма прецедентов различий в функциональности ролей



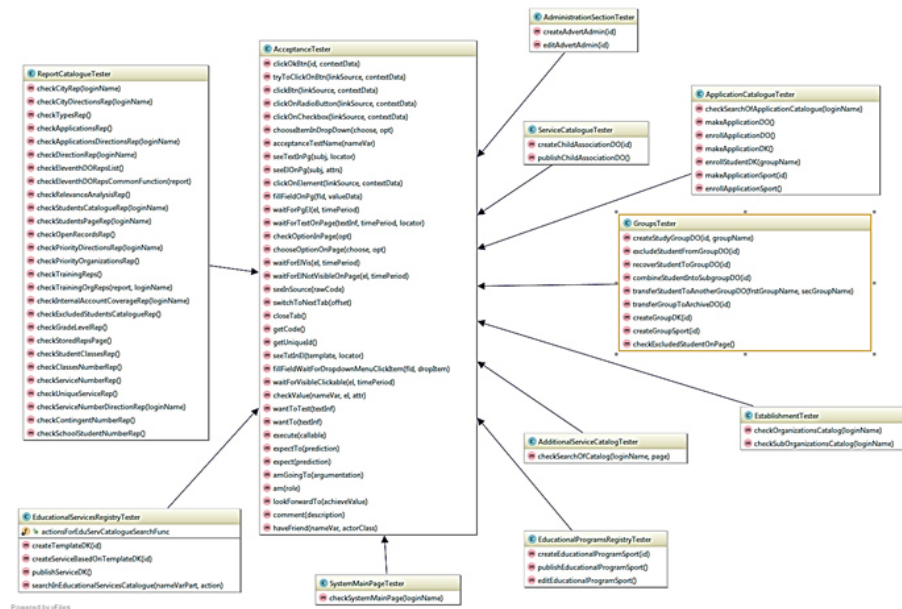
6

Структура проекта



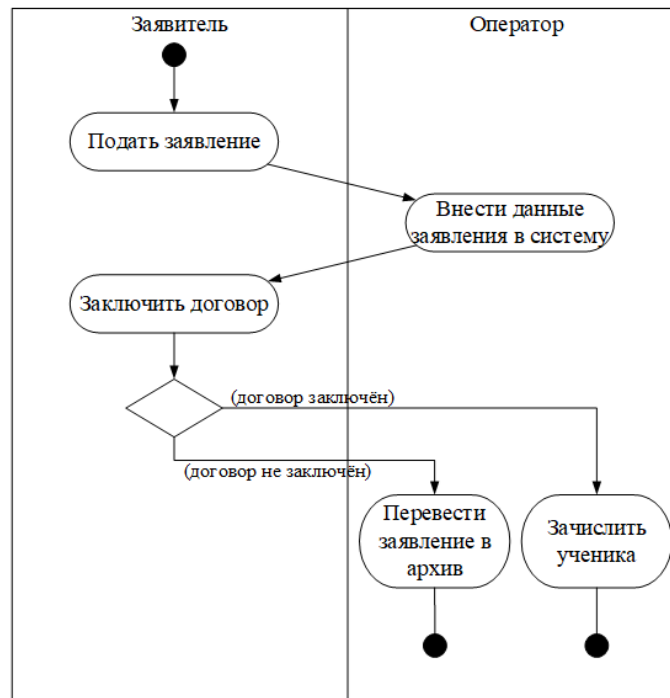
7

UML-диаграмма классов, наследуемых от AcceptanceTester



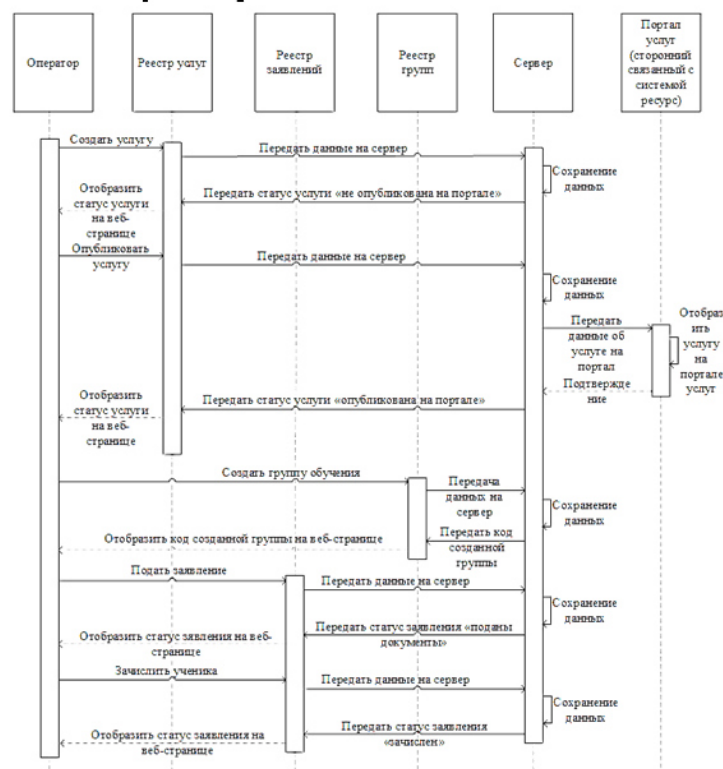
8

UML-диаграмма деятельности взаимодействия оператора и заявителя на услугу



9

UML-диаграмма последовательности сценария «Зачисление»



10

Автомат сценария «Зачисление»

Множество состояний:

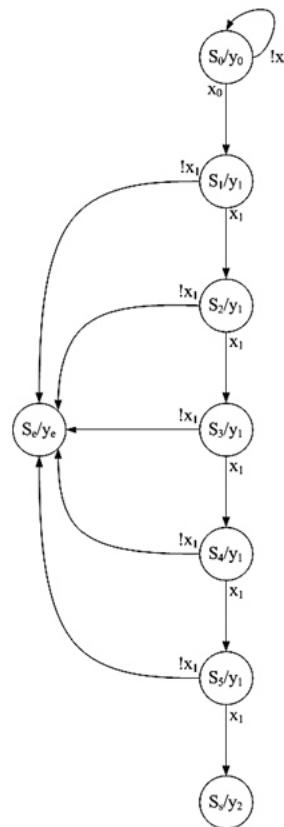
- S_0 – начальное состояние;
- S_1 – создание программы обучения;
- S_2 – публикация программы обучения;
- S_3 – создание группы;
- S_4 – подача заявления на ранее созданную и опубликованную программу;
- S_5 – зачисление абитуриента;
- S_e – конечное состояние;
- S_e – состояние ошибки в ходе выполнения теста;

Множество входных сигналов:

- x_0 – сигнал начала тестирования;
- x_1 – сигнал успешного завершения этапа теста;

Множество выходных сигналов:

- y_0 – пустой сигнал;
- y_1 – сигнал работы теста;
- y_2 – сигнал успешного завершения тестирования;
- y_e – сигнал завершения тестирования с ошибкой;

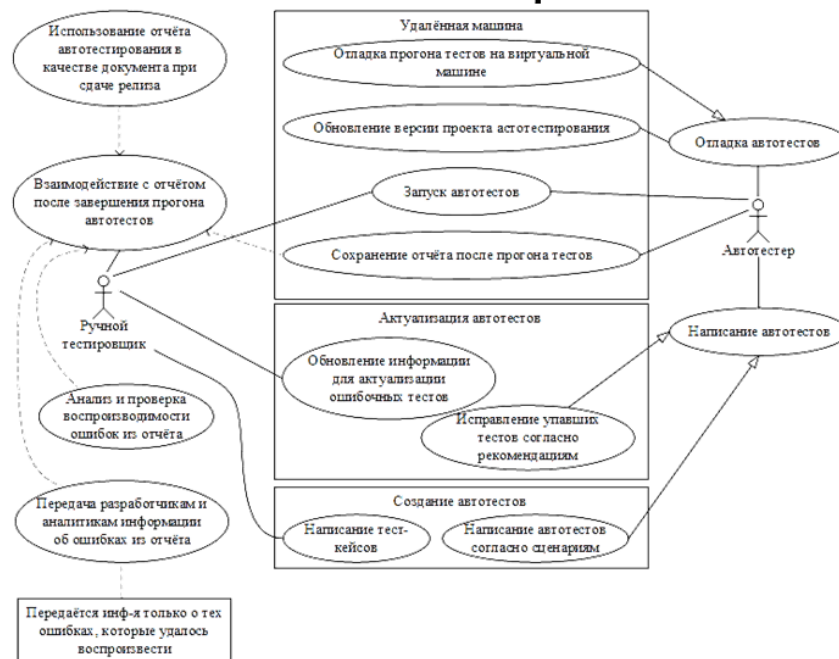


СКУ:

$$\begin{aligned}
 S_0 &= S_0 \overline{x_0} \\
 S_1 &= S_0 x_1 \\
 S_2 &= S_1 x_1 \\
 S_3 &= S_2 x_1 \\
 S_4 &= S_3 x_1 \\
 S_5 &= S_4 x_1 \\
 S_e &= S_5 x_1 \\
 S_e &= (S_1 V S_2 V S_3 V S_4 V S_5) \overline{x_1}
 \end{aligned}$$

11

UML-диаграмма прецедентов взаимодействия ручных тестировщиков и инженеров-автоматизаторов



12

Результаты тестирования интернет-сервиса образовательных организаций



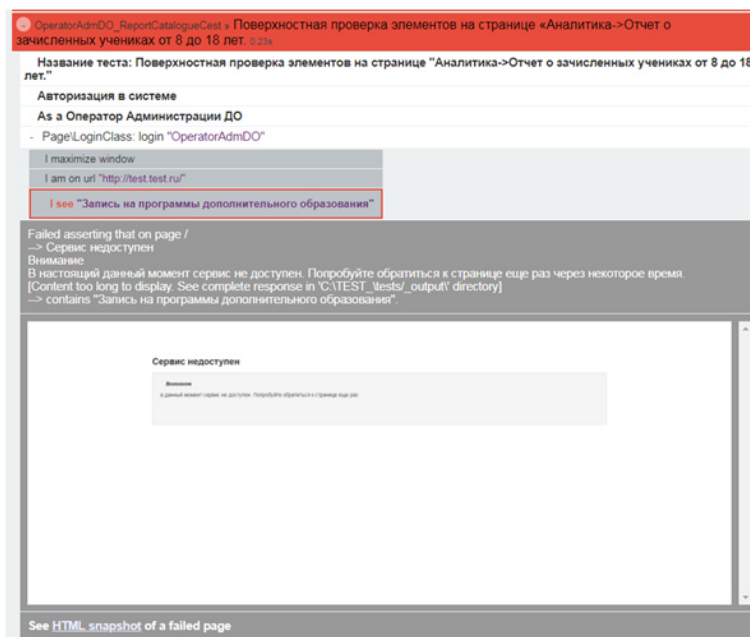
13

Успешно пройденный тест

- DK_Create_group_Cest » ДК. СОЗДАНИЕ ГРУППЫ.	8.46s
Название теста: ДК. СОЗДАНИЕ ГРУППЫ.	
Авторизация в системе	
As a Оператор Организации ДК	
+ Page\LoginClass: log in "OperatorOrgDK"	
Переход на страницу: "Каталог групп обучения -> Каталог групп обучения"	
+ Page\HeaderClass: go to page "Каталог групп обучения Каталог групп обучения"	
I click "Добавить группу"	
ДК. СОЗДАНИЕ ГРУППЫ.	
+ I create group dk "147541"	
Выход из системы	
- Page\HeaderClass: exit	
I click "#logoutbtn"	
Завершение теста	

14

Ошибочно пройденный тест



15

Заключение

Разработанная система автоматизированного функционального тестирования позволяет реализовывать сценарии функционального тестирования, производить их проверку на интернет-сервисе и выявлять ошибки.

Результатом выполнения работы программы являются отчеты, составленные автоматизированной системой функционального тестирования.

16

Благодарю за внимание!