

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
вычислительной техники

Кафедра
МОиПЭВМ

Направление подготовки 09.03.04 «Программная инженерия»

БАКАЛАВРСКАЯ РАБОТА
на тему
Web-приложение для контроля исполнения
поручений в системе электронного
документооборота вуза

Студент	_____	_____
	(подпись, дата)	Копылова Ольга Юрьевна (ФИО полностью)
Руководитель	_____	_____
	(подпись, дата)	Князев В. Н. (фамилия, инициалы)
Нормоконтролёр	_____	_____
	(подпись, дата)	Такташкин Д. В. (фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой	_____	_____
	(подпись)	Макарычев П. П. (фамилия, инициалы)

Работа защищена с отметкой _____ (протокол заседания ГЭК от _____ № _____)

Секретарь ГЭК	_____	_____
	(подпись)	Попова Н. А. (фамилия, инициалы)

Пенза, 2020

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

МО и ПЭВМ

«Утверждаю»
Заведующий кафедрой
П.П. Макарычев
«30» декабря 2019 г

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА**

1. Студент Копылова Ольга Юрьевна гр. 16ВП1 факультета ВТ
<Фамилия, имя, отчество полностью>
2. Тема работы Web-приложение для контроля исполнения поручений в системе электронного документооборота вуза

Тема утверждена приказом ПГУ № 1488/о от " 30 " декабря 2019

3. Руководитель работы к.т.н., доцент кафедры «МОиПЭВМ» Князев В.Н.

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

Назначение: web-приложение предназначено для контроля исполнительской дисциплины сотрудников вуза, позволяющее автоматизировать процессы, связанные с работой с поручениями и документами.

Среда выполнения: операционные системы семейства Windows и Unix

Языковые средства разработки: PHP, JavaScript, HTML, SQL, CSS, UML.

Инструментальные средства разработки: PhpStorm JetBrains 2019.2.4, MySQL, ArgoUML, Erwin Data Modeler.

Технологии разработки: RUP, ООП.

Основные функции:

а) Создание поручения, документа

б) Назначение заместителя

в) Подписание документа невалифицированной электронной подписью

г) Продление выполнения поручения

д) Завершение выполнения поручения

е) Ответ на запрос (одобрение или отклонение)

ж) Отображение списков поручений, документов, замещений

5. Перечень подлежащих разработке вопросов

1) Анализ предметной области и постановка задачи

2) Проектирование

3) Реализация

4) Тестирование

5) Оформление пояснительной записки

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	<i>Анализ предметной области постановка задачи</i>	15%	13.04.2020– 21.04.2020	
2	<i>Проектирование</i>	25%	22.04.2020– 05.05.2020	
3	<i>Реализация</i>	40%	06.05.2020– 30.05.2020	
4	<i>Тестирование</i>	5%	31.05.2020– 02.06.2020	
5	<i>Оформление пояснительной записки</i>	15%	03.06.2020– 13.06.2020	

Дата выдачи задания "30" _____ декабря _____ 2019

Руководитель бакалаврской работы _____
(подпись, дата)

Задание к исполнению принял студент _____
(подпись, дата)

Работу к защите допустить
Декан факультета _____
(подпись, дата)

Реферат

Пояснительная записка содержит 123 листа, 73 рисунка, 26 таблиц, 30 использованных источников, 8 приложений.

КОНТРОЛЬ ИСПОЛНЕНИЯ ПОРУЧЕНИЙ, СИСТЕМА ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА ВУЗА, АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ ВУЗОМ, КОНТРОЛЬ ИСПОЛНИТЕЛЬСКОЙ ДИСЦИПЛИНЫ, WEB-ПРИЛОЖЕНИЕ, КЛИЕНТ-СЕРВЕРНАЯ АРХИТЕКТУРА, MYSQL, PHP

Объектом разработки являются программные средства автоматизации контроля исполнения поручений в электронном документообороте вуза.

Цель работы – разработка web-приложения для контроля исполнительской дисциплины сотрудников вуза, которое позволяет автоматизировать процессы, связанные с работой с поручениями и документами.

Результатом работы является web-приложение для контроля исполнения поручений в системе электронного документооборота вуза.

Разработка серверной части проводилась на языке программирования PHP и SQL. Клиентская часть разрабатывалась на языке разметки документов HTML и языке описания внешнего вида документов CSS, а также Javascript с использованием технологии Ajax. В качестве среды разработки использовалась IDE PhpStorm 2019.2.4. Для работы с данными была выбрана реляционная система управления базами данных MySQL.

Web-приложение было внедрено в систему электронного документооборота ФГБОУ ВО «Пензенский государственный университет».

					ПГУ 09.03.04 – 08BP161.17 ПЗ			
Изм.	Лист	№ докум.	Подп.	Дата	Web – приложение для контроля исполнения поручений в системе электронного документооборота вуза Пояснительная записка	Лит.	Лист	Листов
Разраб.		Копылова О.Ю.					4	123
Пров.		Князев В.Н				гр. 16ВП1		
Реценз.								
Н. контр.		Такташкин Д.В						
Утв.		Макарычев П.П						

Содержание

Введение.....	7
1 Анализ предметной области и требований	9
1.1 Анализ предметной области	9
1.2 Анализ программного обеспечения аналогичного назначения	11
1.2.1 Анализ системы электронного документооборота «Directum»	11
1.2.2 Анализ системы электронного документооборота «Мотив»	14
1.2.3 Анализ системы электронного документооборота «Тезис».....	16
1.3 Постановка задачи	18
1.4 Анализ требований.....	19
1.4.1 Анализ функциональных требований.....	20
1.4.2 Анализ нефункциональных требований.....	26
2 Проектирование Web-приложения.....	29
2.1 Выбор технологий и инструментальных средств разработки.....	29
2.2 Планирование разработки и оценка бюджета.....	30
2.3 Проектирование логической модели базы данных	37
2.4 Проектирование пользовательского интерфейса	44
2.5 Проектирование архитектуры программного обеспечения	53
3 Реализация web-приложения	56
3.1 Реализация базы данных	56
3.2 Реализация клиентской части web-приложения	57
3.3 Реализация серверной части web-приложения	58
4 Тестирование web-приложения	63
4.1 Функциональное тестирование.....	63
4.2 Структурное тестирование.....	64
4.2.1 Покрытие операторов	66
4.2.2 Покрытие условий.....	67
4.2.3 Покрытие маршрутов	67
4.3 Метрики кода.....	68
4.3.1 Цикломатическая метрика Мак-Кейба	68

4.3.2 Метрика Холстеда.....	68
4.3.3 Метрика Чепина	69
4.4 Автоматизированное тестирование	70
4.4.1 Модульное тестирование	70
4.4.2 Тестирование с использованием Katalon Recorder.....	71
Заключение	73
Список использованных источников	74
Приложение А. Глоссарий	77
Приложение Б. Спецификация прецедентов.....	79
Приложение В. Исходный код создания базы данных	83
Приложение Г. Исходный код web-приложения.....	86
Приложение Д. Результаты тестирования.....	101
Приложение Е. Блок-схема метода «Создать поручение»	117
Приложение Ж. Граф потока данных	120
Приложение И. Акт о внедрении	122

Введение

Информационные технологии и достижения вычислительной техники стремительно вторгаются в социальные, экономические и политические процессы общества. Поэтому в настоящее время их внедрение во все сферы деятельности человека является одной из основных задач. Для Российской Федерации это направление является также приоритетным, поэтому в 2017 году была принята Государственная программа «Цифровая экономика Российской Федерации», которая помогает «...созданию необходимых условий для развития цифровой экономики Российской Федерации, в которой данные в цифровой форме являются ключевым фактором производства во всех сферах социально-экономической деятельности, что повышает конкурентоспособность страны, качество жизни граждан, обеспечивает экономический рост» [1].

Кроме принятой программы, в том же году, был подписан документ «Стратегия развития информационного общества в Российской Федерации на 2017 – 2030 годы», в котором были сформулированы основные «...цели, задачи и меры по реализации внутренней и внешней политики Российской Федерации в сфере применения информационных и коммуникационных технологий, направленные на развитие информационного общества, формирование национальной цифровой экономики...» [2]. Это означает, что программное обеспечение будет становиться более доступным и востребованным, что поспособствует его внедрению в различные сферы деятельности человека.

Люди стремятся с помощью новых технологий упростить свою жизнь, поэтому в настоящее время растет спрос на автоматизацию и внедрение программного обеспечения во все внутренние процессы организации. Сфера высшего образования также не остается в стороне.

В вузах ведется активное внедрение информационных технологий в учебный процесс [3]. Это помогает улучшить качество образования, что влияет на рейтинг и востребованность вуза. Однако стоит отметить, что на качество и успешность вуза также влияет эффективно построенный аппарат управления и документооборота. Поэтому внедрение программного обеспечения в эту сферу

деятельности вуза также важно, как и внедрение в процесс обучения. Поэтому в эпоху развития информационных технологий огромное внимание уделяется разработке сайтов, мобильных приложений и программного обеспечения для персональных компьютеров для организации совместной работы сотрудников, постановки и контроля поручений, задач, распоряжений, а также для работы с документами.

Исходя из вышесказанного, актуальность данной выпускной квалификационной работы определяется тем, что внедрение информационных технологий во все сферы деятельности человека в настоящее время является приоритетной задачей для Российской Федерации. Поэтому уделяется большое внимание организации эффективной работы аппарата управления и документооборота в организациях различной направленности, в том числе и в вузах, с использованием новых технологий.

Целью данной квалификационной работы является разработка web-приложения, которое позволит организовать комфортную совместную работу сотрудников вуза и контроль исполнения поручений внутри организации, что обеспечит повышение показателя исполнительской дисциплины.

Для достижения поставленной цели были поставлены и решены следующие задачи:

- анализ предметной области;
- анализ программного обеспечения аналогичного значения;
- проектирование серверной части;
- проектирование пользовательского интерфейса;
- разработка клиентской и серверной частей;
- тестирование и отладка.

1 Анализ предметной области и требований

1.1 Анализ предметной области

Система электронного документооборота (далее СЭД) – это система, которая позволяет организовать работу с электронными документами, а также взаимодействие между сотрудниками и контроль выполнения работы [4]. В целом, этот термин служит для обозначения разного рода автоматизации процессов в государственных, коммерческих и других организациях, в том числе в высших учебных заведениях.

Вузы имеют четкую иерархическую структуру и состоят из множества подразделений, в которые входят сотрудники. Эффективность исполнения поручений каждого конкретного сотрудника сказывается на эффективности работы подразделения в целом. Таким образом, важнейшим показателем эффективности работы организации, на который надо обращать особое внимание, является исполнительская дисциплина [5].

На определенном этапе в организацию системы постановки и контроля исполнения поручений, задач, распоряжений начинают вписываться документы, которые имеют не только служебную, но и юридическую и экономическую силу. На примере вузов это могут быть служебные записки на выделение средств, акты на закупки или приказы ректора. В таком случае задачи контроля исполнительской дисциплины тесно связываются с задачами документооборота [5, 6].

По оценке TAdviser динамика развития российского рынка систем электронного документооборота растет с каждым годом. Например, в 2019 году рост рынка составил порядка 7%, при этом его объем составлял 52 млрд. рублей. Из этого следует, что спрос на внедрение инновационных сервисов в работу организации продолжает расти. Главными факторами, способствующими их развитию, является реализация национальной программы «Цифровая экономика», необходимость улучшения существующих систем и импортозамещение [7].

Распределение доли проектов СЭД по отраслям по данным базы TAdviser за период наблюдений с 2005 по декабрь 2019 года представлено на рисунке 1.

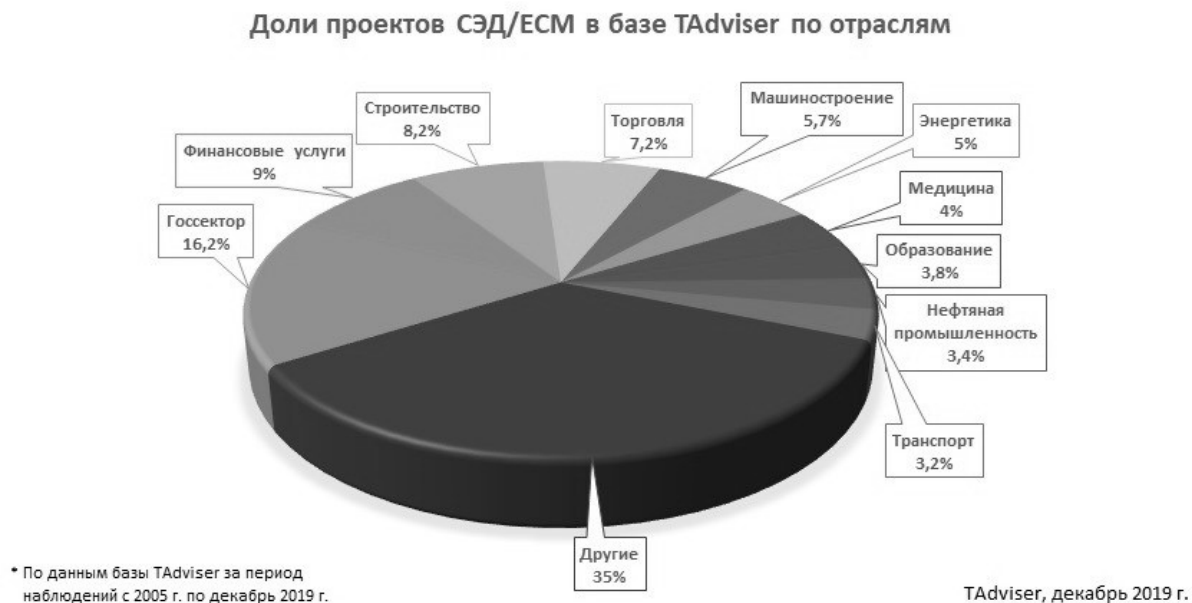


Рисунок 1 – Доля проектов СЭД по отраслям по данным TAdviser

Из рисунка, представленного выше, видно, что лидирует госсектор, а в первую пятерку входят отрасли финансовых услуг, строительство, торговля и машиностроение, в то время как сфере образования процессам автоматизации документооборота и контроля исполнительской дисциплины не уделяется должного внимания.

Согласно данным исследовательской компании Gartner к функциям СЭД можно отнести следующее:

- организация управления документами;
- поддержка совместной работы сотрудников;
- преобразование и управление бумажными документами;
- управление записям, которое осуществляется в соответствии с ГОСТ Р ИСО 15489-1-2007;
- организация бизнес-процессов, перемещение контента по строго определенным маршрутам, возможность назначения задач и контроль их исполнения;

– управление динамическим контентом и взаимодействием пользователя.

Чтобы система могла быть классифицирована как СЭД, она должна поддерживать как минимум 3 функции из рассмотренных выше. На основе того, какие функции реализует СЭД, их можно разделить на следующие виды [8, 9]:

- системы делопроизводства;
- электронные архивы;
- системы управления потоками работ;
- комплексные СЭД или ЕСМ-системы.

Как правило системы электронного документооборота имеют клиент-серверную архитектуру. В качестве клиента может выступать desktop-приложение или web-браузер. Через клиентское приложение осуществляется доступ к данным. По сети интернет происходит обмен между клиентом и web-сервером, который в свою очередь отвечает за обработку запросов и передачу данных между клиентом и базой данных.

Актуальность выбранной темы подчеркивается тем, что вузы, как и любая большая организация, нуждаются в контроле исполнительской дисциплины, однако они только начинают процесс автоматизации аппарата управления. Таким образом решения, которые будут удовлетворять запросам вузов будут пользоваться спросом.

1.2 Анализ программного обеспечения аналогичного назначения

Существует множество решений для автоматизации электронного документооборота организации. Проведем анализ программного обеспечения (ПО) аналогичного назначения.

В качестве аналогов были выбраны клиент-серверные приложения «Directum», «Мотив», «Тезис».

1.2.1 Анализ системы электронного документооборота «Directum»

СЭД Directum предоставляет возможность управления цифровыми

документами и поручениями. Эта система может использоваться в любой организации, где необходима работа с множеством документов, в том числе и в вузе. Например, данную СЭД используют в ФГБОУ ВО «Пензенский государственный университет», Финансовый Университет при правительстве РФ [10].

Интерфейс главной страницы системы «Directum» представлен на рисунке 2.

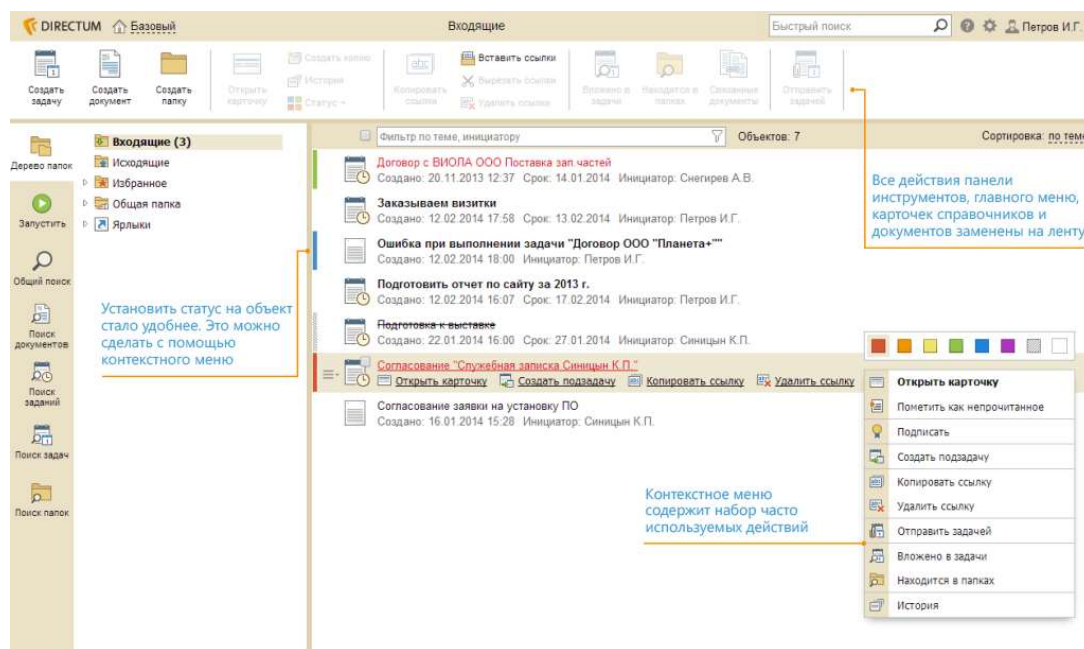


Рисунок 2 – Интерфейс СЭД «Directum»

Система электронного документооборота «Directum» состоит из множества модулей, которые подключаются отдельно. Необходимым функционалом обладают модули «Управление деловыми процессами» и «Управление электронными документами». Они предоставляют следующие основные возможности:

- выполнение основных задач делопроизводства (создание, ведение документа, хранение и т.д.);
- создание и отслеживание выполнения поручений;
- разграничение прав доступа к документам, что обеспечивает конфиденциальность и безопасность;

- построение маршрутов в графическом редакторе,
- мониторинг эффективности сотрудника,
- поиск необходимого документа по его атрибутам и другие.

Для организации электронного документооборота в вузе функционал, которые предоставляет данная СЭД, является избыточным. Также система «Directum» имеет перегруженный и усложненный интерфейс, что не позволяет использовать систему без предварительного обучения.

Стоимость базовых лицензий представлена в таблице 1.

Таблица 1 – Стоимость ПО «Directum»

Критерий	Расширенная версия, рублей	Стандартная версия, рублей
Базовая серверная лицензия	36 300	20 700
Базовая клиентская лицензия	14 500	8 400
Пакет 20 базовых клиентских лицензий	275 500	159 600
Пакет 50 базовых клиентских лицензий	652 500	378 000
Пакет 100 базовых клиентских лицензий	1 160 000	672 000
Пакет 200 базовых клиентских лицензий	2 175 000	—

После покупки базовой лицензии возможно отдельное приобретение дополнений. Однако, рассмотренные ранее компоненты «Управление деловыми процессами» и «Управление электронными документами» входят в состав базовой лицензии, поэтому стоимость дополнительных компонентов не рассматривалась.

Также, стоит отметить, что обновление приобретенного ПО осуществляется только при наличии действующего абонемента. Расчет его стоимости представлен на рисунке 3.

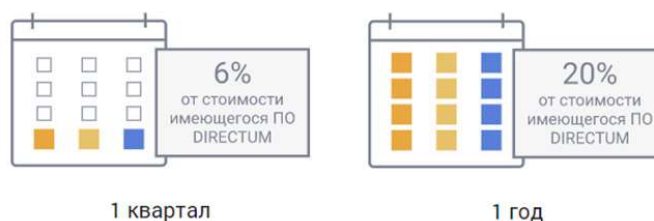


Рисунок 3 – Стоимость поддержки и обновления ПО «Directum»

Таким образом можно сделать вывод, что внедрение, поддержка и использование этой системы требует больших финансовых затрат. Также из-за сложности интерфейса сотрудникам требуется постоянное прохождение курсов, что также требует дополнительных финансовых вложений и больших затрат времени.

1.2.2 Анализ системы электронного документооборота «Мотив»

Система «Мотив» обеспечивает возможность совместной работы сотрудников и управление электронными документами. Пользователями этой системы являются следующие вузы: Институт отраслевого менеджмента РАНХ и ГС при Президенте РФ, Казахский национальный технический университет, Санкт-Петербургский университет Министерства внутренних дел Российской Федерации [11].

Система «Мотив» обладает следующими возможностями:

- создание иерархической структуры взаимодействия руководящего персонала с подчиненными;
- коллективная работа над проектами и документами;
- получение регулярных отчетов о ходе выполнения задач по проектам;
- эффективный контроль за процессом исполнения задач;
- контроль сроков исполнения;
- мониторинг занятости персонала;
- значительное ускорение поиска и выборки документов (по различным атрибутам);
- улучшение контроля над исполнением документов;
- повышение безопасности документов и т.д.

Стоимость лицензий на использование СЭД мотив представлена в таблице 2.

Таблица 2 – Стоимость СЭД «Мотив»

Тип	Модуль для работы с задачами, рублей	Модуль для работы с документами рублей	Объединенные возможности
Лицензия на конкретного сотрудника	5000	5000	6500
Лицензия на одно подключение к системе	10000	10000	11000

Таким образом, стоимость использования данной системы для организации с большим количеством сотрудников требует больших финансовых затрат, так как лицензия действует либо на одного конкретного человека, либо на одно подключение.

Интерфейс системы «Мотив» представлен на рисунке 4.

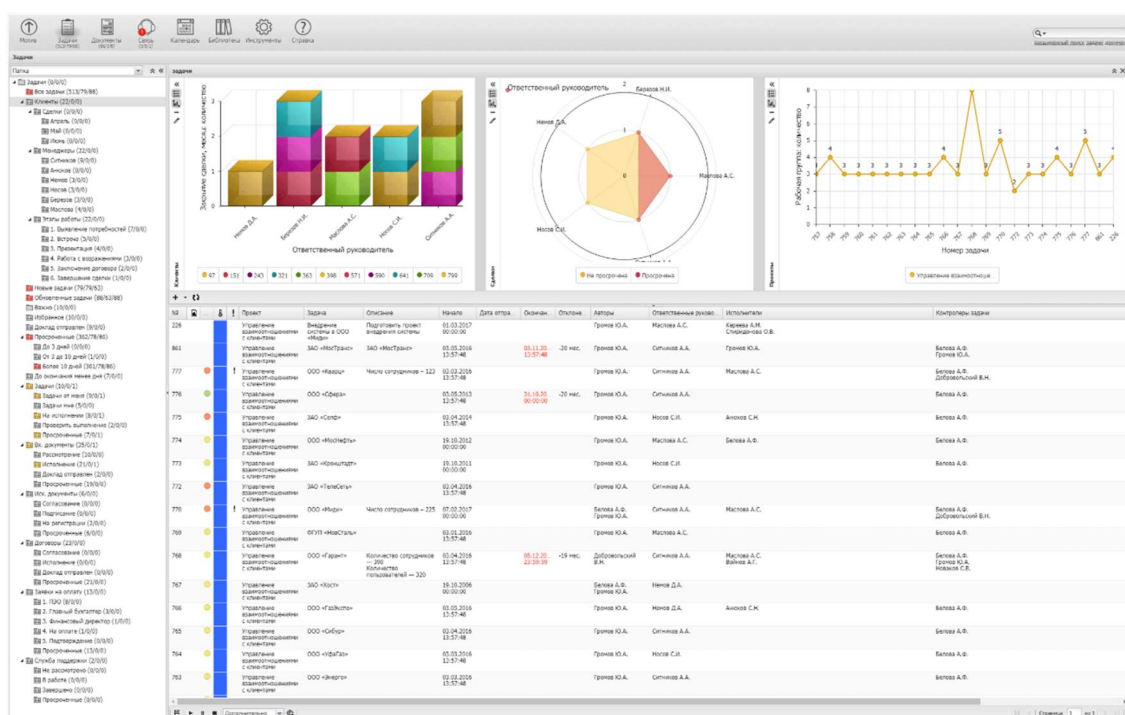


Рисунок 4 – Интерфейс СЭД «Мотив»

Интерфейс системы «Мотив» является более простым и понятным, чем интерфейс СЭД «Directum». Однако недостатком данной системы является избыточный функционал, который затрудняет ее использование без предварительного обучения. Помимо этого, для использования необходима покупка лицензий на каждого человека, а обучение администратора, внедрение и техподдержка оплачивается дополнительно, что существенно увеличивает

финансовые затраты, если внедрять эту систему в большую организацию.

1.2.3 Анализ системы электронного документооборота «Тезис»

СЭД «Тезис» является готовым решением для автоматизации классического документооборота. Данную систему используют во многих высших учебных заведениях, например, ФГБОУ ВО «Воронежский государственный университет», СПбГПУ «Санкт-Петербургский государственный политехнический институт», Медицинский университет «Реавиз» и т.д. [12].

Существует 3 версии программного продукта: базовая, стандартная и расширенная. В зависимости от редакции система предоставляет следующие функции:

- контроль исполнительской дисциплины;
- автоматизация документооборота;
- подписание документов неквалифицированной электронной подписью (ЭП);
- дизайнер бизнес-процессов и другие.

Интерфейс системы электронного документооборота «Тезис» представлен на рисунке 5.

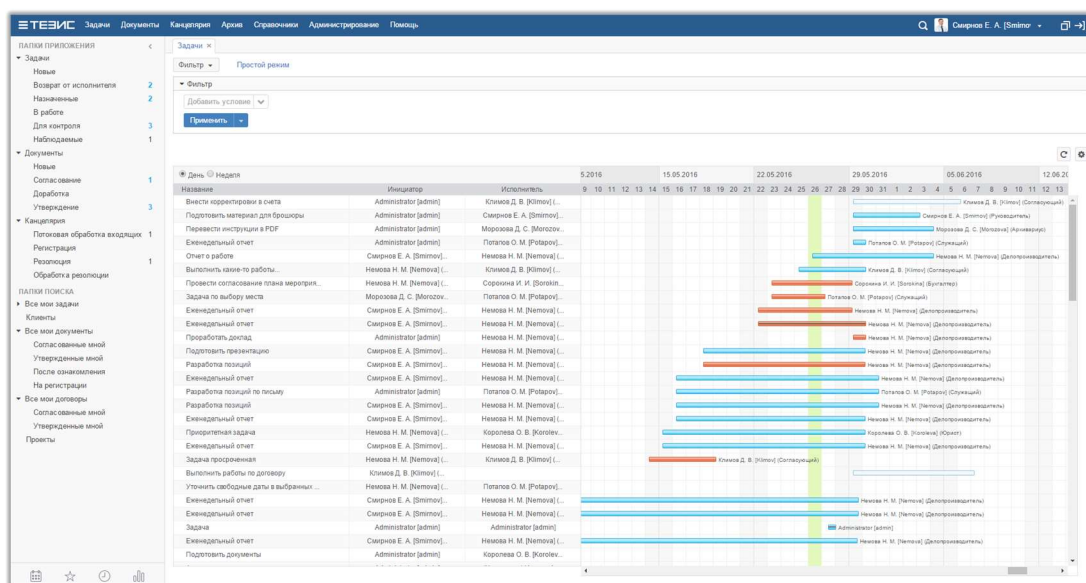


Рисунок 5 – Интерфейс СЭД «Тезис»

Стоимость системы определяется количеством одновременных подключений и выбранной версией программного продукта, что является существенным недостатком для большой организации, такой как вуз. Также стоит отметить, что при необходимости расширения функциональных возможностей системы необходимо приобретать соответствующую лицензию заново, расширение уже имеющегося функционала невозможно.

Стоимость системы «Тезис» представлена в таблице 3.

Таблица 3 – Стоимость СЭД «Тезис»

Пакет одновременных подключений, единиц	Базовая, рублей	Стандартная, рублей	Расширенная, рублей
5	35 000	-	-
10	69 000	115 000	160 000
25	140 000	200 000	290 000
50	-	320 000	470 000
75	-	470 000	700 000
100	-	620 000	920 000
Более 100	-	-	По запросу

Из представленной выше таблицы видно, что использование СЭД «Тезис» для большой организации будет стоить свыше 320 000 руб., что делает ее слишком дорогой для внедрения в вуз.

Таким образом, в ходе анализа программных продуктов аналогичного назначения было принято решение разработать собственную систему, которая не будет обладать рассмотренными выше недостатками, а именно: дороговизна, недружественный для пользователя интерфейс, избыточный функционал, затрудняющий использование системы работниками вуза.

Очевидное преимущество всех рассмотренных систем заключается в том, что это web-приложения с клиент-серверной архитектурой. Это обеспечивает возможность использования программного продукта с любого устройства. Поэтому было принято решение разрабатывать систему также в виде web-приложения.

1.3 Постановка задачи

В результате анализа предметной области и имеющихся готовых решений было принято решение разработать web-приложение для контроля исполнения поручений в системе электронного документооборота вуза.

Система должна предоставлять возможность полного отслеживания исполнения поручений и организации совместной работы сотрудников с документами.

В системе у пользователя может быть 3 роли: инициатор, исполнитель и наблюдатель. В разных работах каждый пользователь может принимать разные роли.

Инициатором является пользователь, создавший задачу. Он может назначить одного или нескольких исполнителей и одного или нескольких наблюдателей. Если было назначено несколько исполнителей, то один из них обязательно должен быть выбран как ответственный. Исполнителем является тот пользователь, кому было отправлено поручение или запрос на подписание. Наблюдатель не может выполнять никаких действий с работой, на которую он назначен. Его задача состоит в наблюдении за ходом выполнения работы.

Система должна предоставлять следующие основные возможности работы с поручениями:

- продление выполнения (от исполнителя к инициатору направляется запрос на изменение даты окончания работы);
- закрытие поручения (от исполнителя отправляется отчет о проделанной работе, затем инициатор подтверждает этот запрос);
- отправка поручения на исполнение другому пользователю (исполнитель отправляет свое поручение на исполнение другому человеку), таким образом обеспечивается дальнейшая маршрутизация работы.

Система должна обладать следующими основными функциями для работы с документами:

- внесение документа в систему;

- подписание документа неквалифицированной ЭП;
- выдача поручений на основе документа.

Исполнитель может принять или отклонить запрос подписания. Документ считается подписанным, если ответственный исполнитель принял запрос подписания. Подписание документа производится с помощью неквалифицированной ЭП.

Система контроля исполнения поручений предоставляет возможность назначения заместителей. Заместитель может выполнять все ранее описанные действия для роли инициатора, но от имени пользователя, которого он замещает. Замещение может быть временным (указываются дата начала действия и дата окончания действия) и постоянным (бессрочное).

Доступ к системе предоставляется после авторизации пользователя в личном кабинете ЭИОС ПГУ с ролью сотрудник. Безопасность информации в системе обеспечивается за счет разграничения прав. Пользователь может видеть поручения и документы, для которых он является инициатором, исполнителем или наблюдателем.

Для использования системы пользователь может использовать смартфон, планшетный компьютер или персональный компьютер с выходом в интернет и с установленным браузером. Это предоставляет сотрудникам возможность использовать web-приложение контроля исполнения поручений в системе электронного документооборота вуза и удаленно, и непосредственно на рабочем месте.

1.4 Анализ требований

Анализ требований – это неотъемлемая часть разработки программного обеспечения. Этот этап включает в себя сбор требований, которые выдвигаются к программному обеспечению, выявление взаимосвязей между ними и документирование [13].

Качество и полнота проведения анализа требований непосредственно влияют на успешное завершение проекта. Требования можно разделить на две

категории: функциональные и нефункциональные. Функциональные требования представляют собой перечень сервисов, которые система должна выполнять, и описание их поведения. Нефункциональные требования описывают характеристики системы и ее окружения. Также в них могут быть описаны ограничения, которые должны накладываться на действия [14].

1.4.1 Анализ функциональных требований

Главной задачей разрабатываемого web-приложения является выдача и отслеживание исполнения задач, что поможет повысить эффективность работы сотрудников вуза.

В ходе анализа предметной области были выделены следующие функции, которые должны быть реализованы в web-приложении:

- создание поручения;
- создание документа;
- назначение заместителя;
- создание поручения на основе документа;
- создание поручения на основе другого поручения;
- получение от сервера списка поручений, замещений пользователя и внесенных в систему документов;
- подписание документа;
- продление даты окончания выполнения поручения;
- завершение выполнения поручения;
- контроль инициатором выполнения поручения (ответ на запросы от исполнителя).

На рисунке 6 представлена диаграмма вариантов использования, разработанная на основе выделенных требований, описывающая функциональное назначение системы.

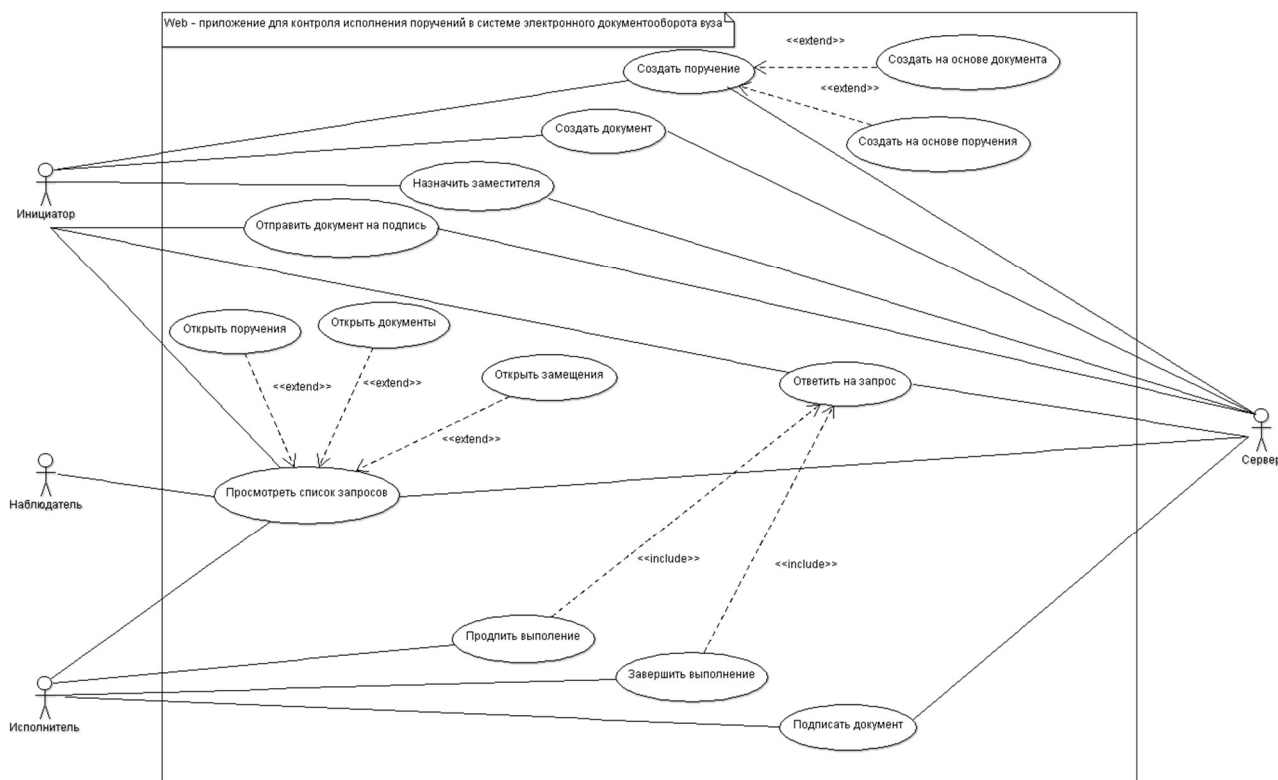


Рисунок 6 – Диаграмма вариантов использования web-приложения

В ходе проектирования системы было выделено 4 сущности (актера):

- инициатор отправляет задачу на исполнение и имеет возможность назначения своего заместителя;
- наблюдатель отслеживает выполнение работы без возможности совершения какого-либо действия;
- исполнитель выполняет направленную ему задачу;
- сервер выполняет обработку запросов (выполнение поручений, подписание документов и т.д.).

Спецификация является основным источником для выполнения анализа и проектирования системы, поэтому для более наглядного представления функционирования прецедентов были описаны спецификации прецедентов «Создать поручение» и «Завершить выполнение» представлены на рисунках 7 и 8.

Прецедент: Завершить выполнение
ID: 7
Краткое описание: Исполнитель создает запрос на завершение поручения
Главные актёры: Исполнитель
Второстепенные актёры: Сервер
Предусловия: 1. Исполнитель вошел в систему. 2. Исполнителю было отправлено на исполнение поручение
Основной поток: 1. Исполнитель перешел на страницу поручения 2. Исполнитель нажал на кнопку «Завершить» 3. Исполнитель вводит данные 3.1. Исполнитель вводит описание 3.2. Если исполнитель прикрепляет файлы 3.2.1. Исполнитель выбирает один или несколько файлов 4. Исполнитель нажимает кнопку «Сохранить» 5. Если ошибок нет 5.1. Сервер выполняет сохранение запроса 6. Иначе переход на действие 2 7. Include «Ответить на запрос»
Постусловия: 1. Создается запрос на завершение и ответ на него
Альтернативные потоки: Нет.

Рисунок 7 – Спецификация прецедента «Завершить выполнение»

Прецедент: Отправить документ на подпись
ID: 4
Краткое описание: Инициатор создает запрос на подписание документа
Главные актёры: Инициатор
Второстепенные актёры: Сервер
Предусловия: 1. Инициатор вошел в систему.
Основной поток: 1. Инициатор открывает страницу создания подписания 2. Инициатор вводит данные 2.1. Если инициатор является заместителем другого пользователя и поручение должно быть создано от его имени 2.1.1. Подписание создается от имени того, кого замещает инициатор 2.2. Инициатор выбирает одного или нескольких согласующих 2.3. Инициатор выбирает утверждающую подпись из выбранных согласующих 2.4. Если инициатор назначает наблюдателей 2.4.1. Инициатор выбирает одного или нескольких наблюдателей 2.5. Инициатор вводит название и описание подписания 2.6. Если инициатор прикрепляет файлы 2.6.1. Инициатор выбирает один или несколько файлов 3. Инициатор нажимает кнопку «Сохранить» 4. Если ошибок нет 4.1. Сервер выполняет сохранение поручения 5. Иначе переход на действие 2
Постусловия: 1. Документ отправлен на подписание
Альтернативные потоки: Нет.

Рисунок 8 – Спецификация прецедента «Отправить документ на подпись»

Спецификация прецедента «Отправить документ на подпись» представлена на рисунке 9. Спецификация остальных прецедентов представлена в приложении Б.

Прецедент: Создать поручение
ID: 1
Краткое описание: Инициатор создает запрос на исполнение поручения
Главные актёры: Инициатор
Второстепенные актёры: Сервер
Предусловия: 1. Инициатор вошел в систему.
Основной поток: 1. Инициатор открывает страницу создания поручения 2. Инициатор вводит данные 2.1. Extend «Создать на основе документа» 2.2. Extend «Создать на основе поручения» 2.3. Если инициатор является заместителем другого пользователя и поручение должно быть создано от его имени 2.3.1. Поручение создается от имени того, кого замещает инициатор 2.4. Инициатор выбирает одного или нескольких исполнителей 2.5. Инициатор выбирает ответственного исполнителя из выбранных исполнителей 2.6. Если исполнитель назначает наблюдателей 2.6.1. Инициатор выбирает одного или нескольких наблюдателей 2.7. Инициатор вводит название и описание поручения 2.8. Если инициатор прикрепляет файлы 2.8.1. Инициатор выбирает один или несколько файлов 3. Инициатор нажимает кнопку «Сохранить» 4. Если ошибок нет 4.1. Сервер выполняет сохранение поручения 5. Иначе переход на действие 2
Постусловия: 1. Поручение создано и отправлено на исполнение
Альтернативные потоки: Нет.

Рисунок 9 – Спецификация прецедента «Создать поручение»

Более подробное описание функционирования прецедентов можно представить в виде диаграммы деятельности. Данный вид UML-диаграммы показывает последовательность действий, которая необходима для достижения конечной цели рассматриваемого прецедента. На рисунке 10 представлена диаграмма деятельности для варианта использования «Завершить выполнение».

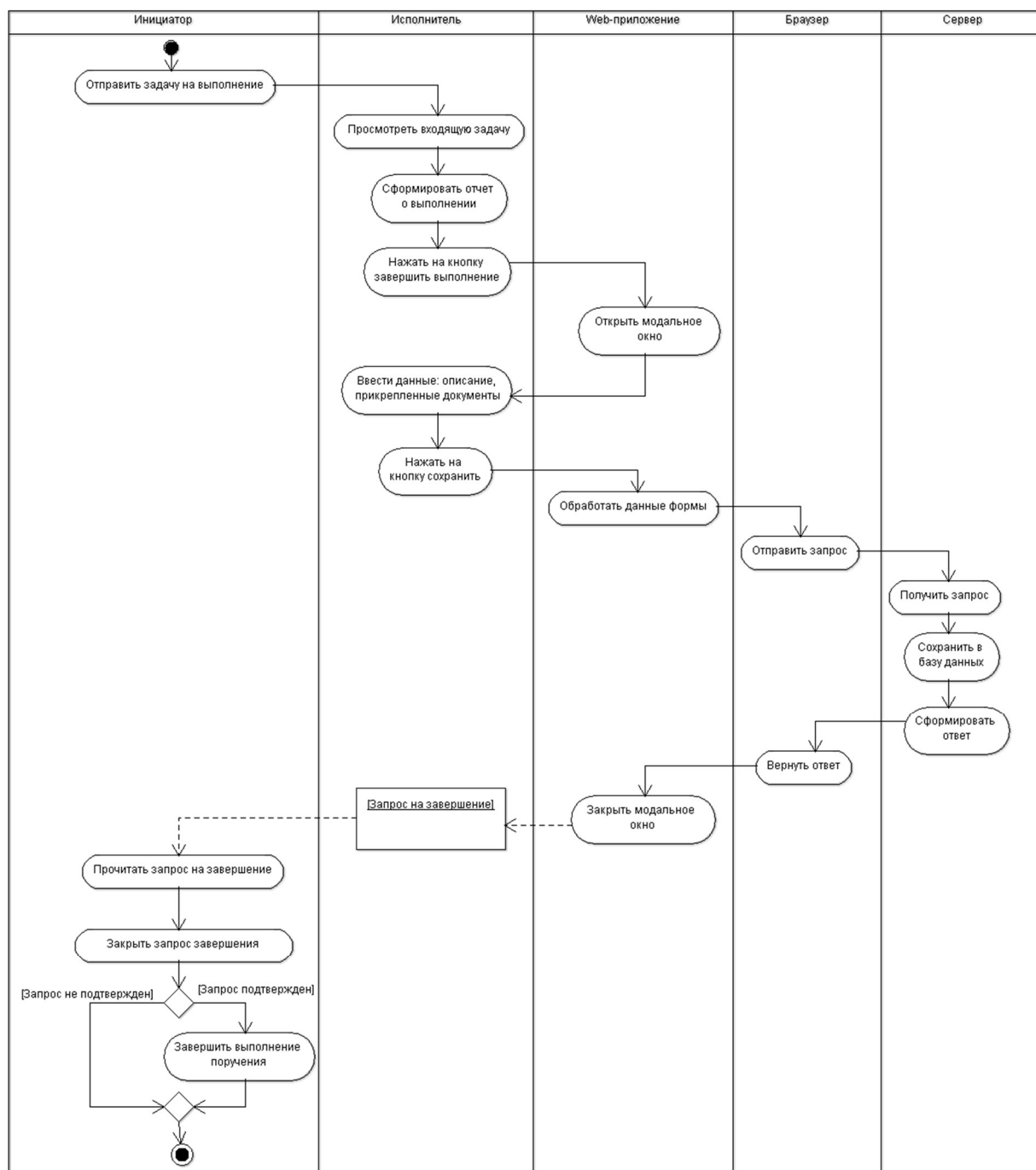


Рисунок 10 – Диаграмма деятельности для прецедента «Завершить выполнение»

На рисунке 11 представлена диаграмма деятельности для варианта использования «Отправить документ на подпись».

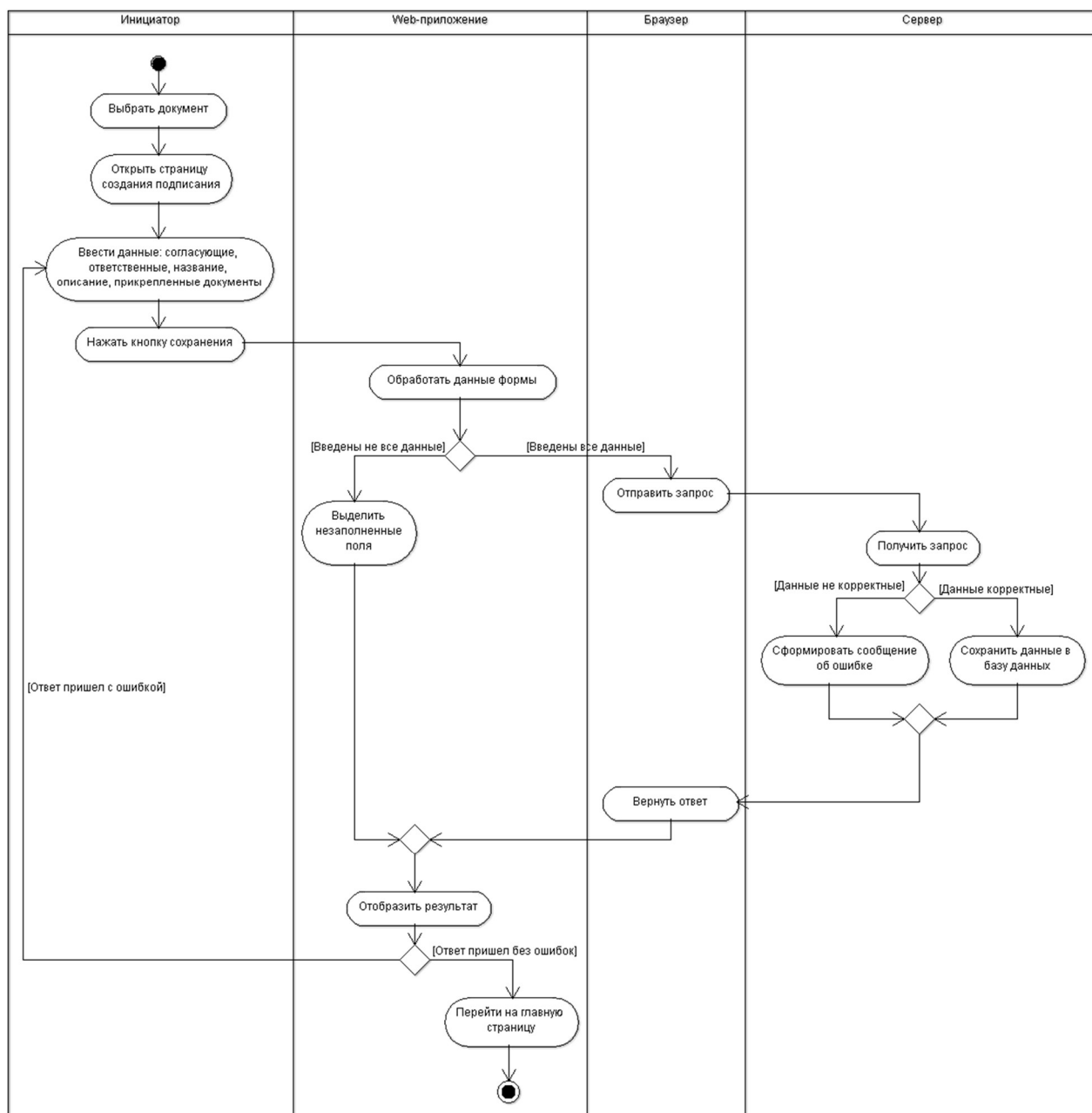


Рисунок 11 – Диаграмма деятельности для прецедента «Отправить документ на подпись»

На рисунке 12 представлена диаграмма последовательности для прецедента «Отправить документ на подписание», которая отображает более подробное взаимодействие объектов во времени, а также показывает обмен сообщениями между объектами в рамках рассматриваемого сценария.

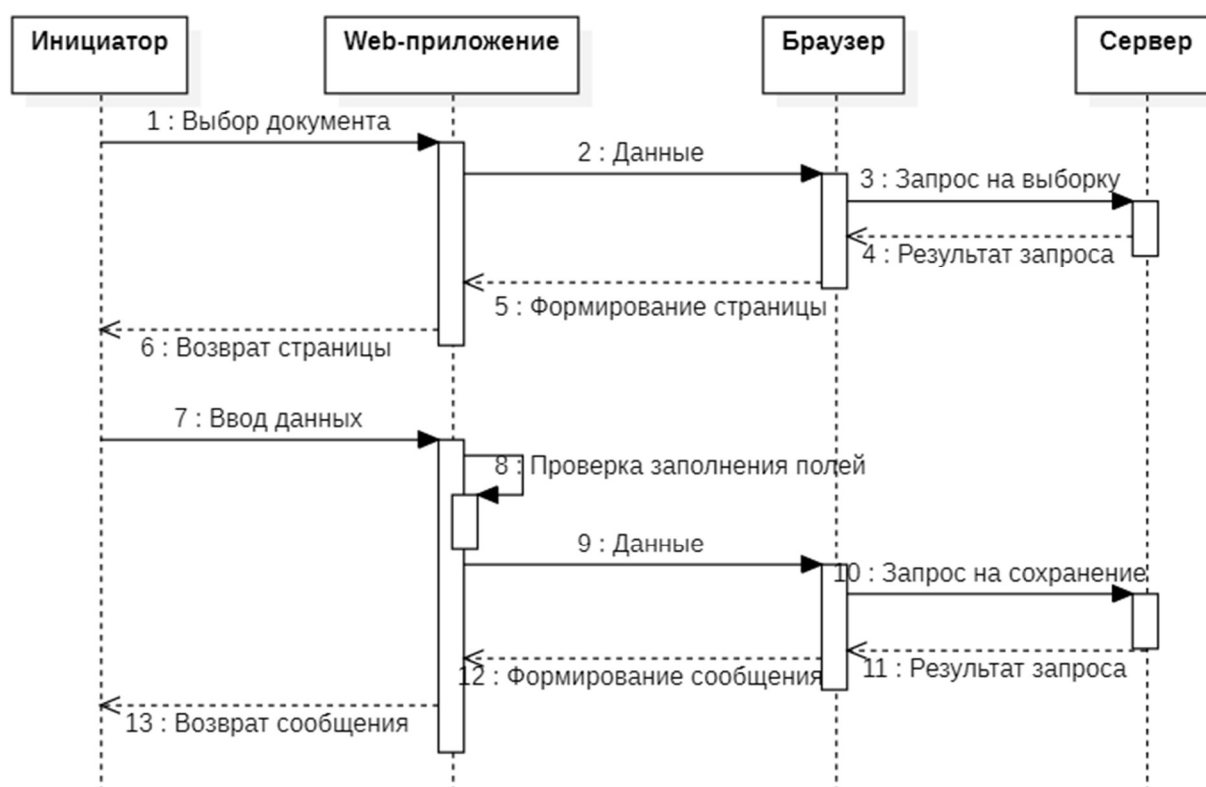


Рисунок 12 – Диаграмма последовательности для прецедента «Отправить документ на подпись»

На диаграмме последовательности (смотри рисунок 12) изображены следующие компоненты системы: инициатор, web-приложение, браузер и сервер. Данный вид UML диаграмм позволяет рассмотреть особенности приема и отправки сообщений объектами во времени.

1.4.2 Анализ нефункциональных требований

Нефункциональные требования определяют ограничения, которые должна соблюдать система, и свойства, которые она демонстрирует. Они не относятся к поведению системы [14].

Программное обеспечение должно иметь клиент-серверную архитектуру. Серверная часть программного обеспечения должна быть разработана на языке PHP, работа с базой данных должна осуществляться на языке SQL. Для обеспечения простоты поддержки и развития программного кода должны быть соблюдены принципы ООП. В качестве системы управления базами данных

(СУБД) должна использоваться MySQL.

Клиентская часть должна быть разработана с использованием языка разметки документов HTML и языка описания внешнего вида документов CSS, а также язык Javascript.

Разрабатываемое web-приложение должно обеспечивать корректную работу и отображение на наиболее используемых разрешениях экрана и в наиболее популярных браузерах. Статистика StatCounter по использованию браузеров и разрешений экрана за период с апреля 2019 года по апрель 2020 года представлена на рисунках 13 и 14 соответственно [15].

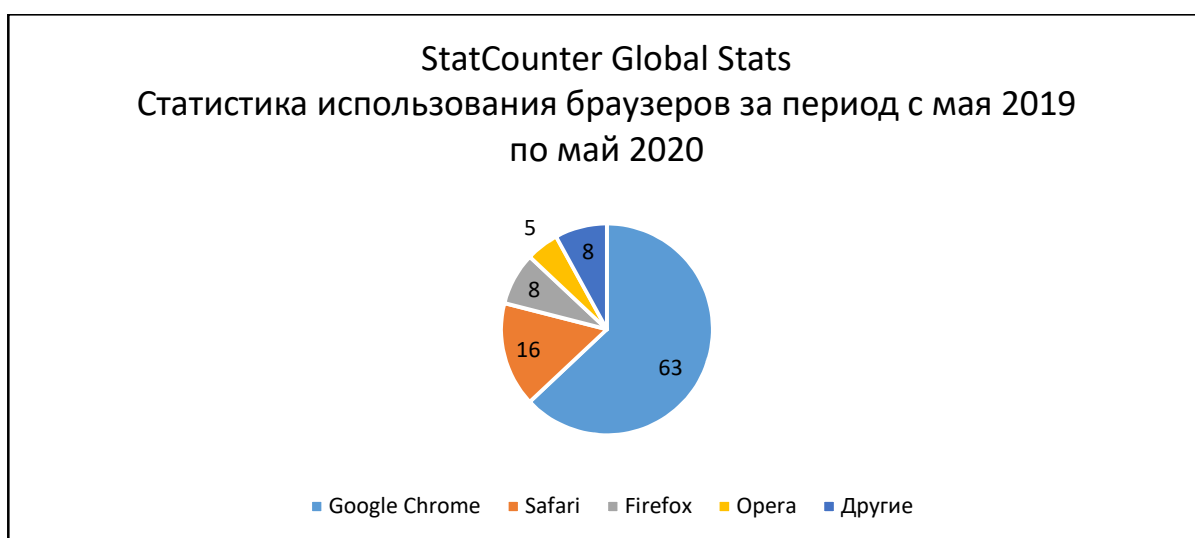


Рисунок 13 – Статистика использования браузеров

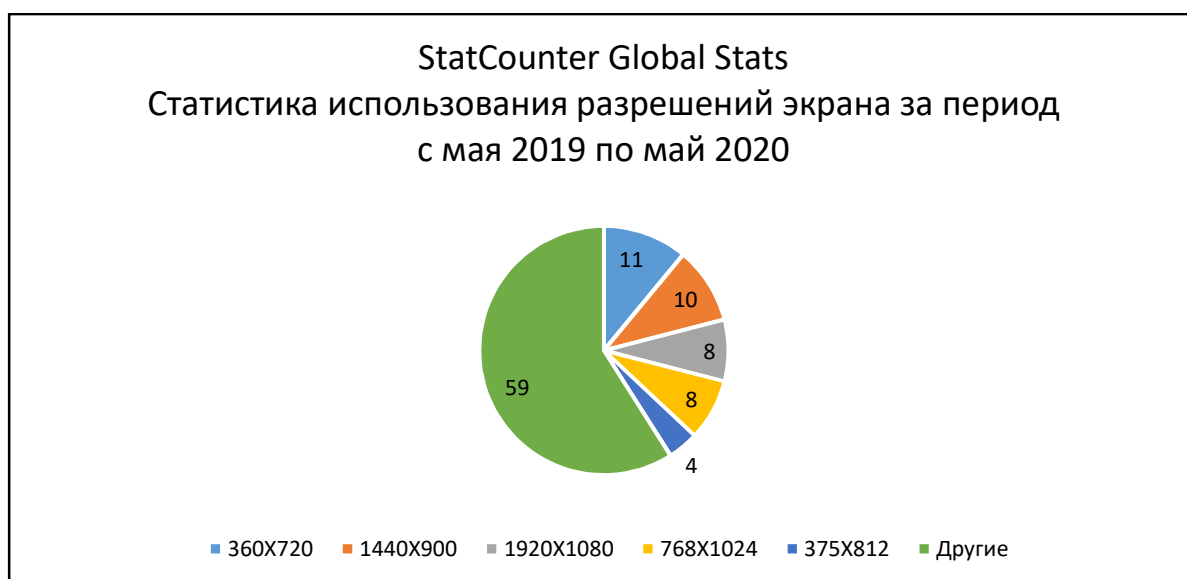


Рисунок 14 – Статистика использования разрешений экрана

Таким образом необходимо предусмотреть одинаковое отображение и корректную работу на наиболее распространенных браузерах (Chrome, Safari, Firefox, Opera) и разрешениях экрана (360x720, 1440x900, 1920x1080, 768x1024, 375x812). Также интерфейс должен быть интуитивно понятным и удобным.

Передача данных между клиентской и серверной частями должна производиться с использованием технологии Asynchronous Javascript and XML (AJAX), что позволяет менять содержимое страницы без ее перезагрузки.

Разрабатываемое программное обеспечение должно иметь механизм защиты от ошибок и отказов.

2 Проектирование Web-приложения

2.1 Выбор технологий и инструментальных средств разработки

Реализация web-приложения проводилась с использованием технологии Rational Unified Process (RUP). Это универсальная технология, которая подходит для разработки любых систем. Ее отличительной чертой является итеративный подход к проектированию и разработке программного продукта. Она основана не на классической водопадной модели жизненного цикла проекта, а на спиральной. Таким образом, на выходе с очередного витка спирали должен быть готов протестированный прототип. Итеративный подход к разработке позволяет на ранних стадиях отслеживать и устранять риски, оперативно реагировать на меняющиеся требования и постоянно следить за качеством разрабатываемого программного продукта [16].

Технология RUP подразумевает использование языка визуального моделирования Unified Modeling Language (UML). UML – унифицированный графический язык для объектного моделирования в области разработки программных продуктов. Он используется для моделирования программного обеспечения и бизнес-процессов, системного проектирования и отображения организационных структур [17].

UML предоставляет возможности описания всех этапов жизненного цикла программного продукта. Этот язык создавался специально для оптимизации процесса разработки программных систем. Это позволяет увеличить эффективность реализации проекта и улучшить качество конечного продукта.

Для работы с базой данных была выбрана система управления базами данных (СУБД) MySQL Community Server версии 5.6.47 от компании Oracle. На сегодняшний день она является наиболее популярной при разработке web-приложений и свободно распространяемой (с некоторыми ограничениями). СУБД MySQL обладает следующими преимуществами [18]:

- простота развертывания системы;
- обширный функционал;

- безопасность;
- подходит для работы с малыми и большими объемами данных;
- высокая производительность.

Для администрирования СУБД MySQL использовался web-интерфейс phpMyAdmin 4.4.15.10.

Разработка серверной части web-приложения проводилась на языке программирования PHP, который активно поддерживается большинством хостинг-провайдеров и является лидером при разработке динамических web-сайтов. Для написания скриптов на выбранном языке программирования использовалась IDE PhpStorm 2019.2.4 – коммерческая кросс-платформенная среда разработки, разработанная компанией JetBrains на основе платформы IntelliJ IDEA [19].

Современные web-приложения должны быть адаптивными, производительными, с интуитивно понятным интерфейсом. Для создания визуальной составляющей страниц приложения использовался стандартизированный язык разметки документов HTML и формальный язык описания внешнего вида документа CSS. Поведение компонентов страницы будет описываться на языке JavaScript. Для обеспечения кросс-браузерности и адаптивности использовалась библиотека Bootstrap v4.0. Он включает в себя HTML и CSS-шаблоны для оформления компонентов страницы (кнопок, форм, ссылок и т.д.) [20].

Обмен данными между браузером и сервером будет организован с использованием Asynchronous Javascript and XML (AJAX), что позволяет обновлять данные на странице без ее перезагрузки. Благодаря этому web-приложение становится быстрее и удобнее.

2.2 Планирование разработки и оценка бюджета

Управление проектами позволяет достигать высокой эффективности по показателям производительности, оптимизации сроков и затрат на выполнения отдельных работ или проекта в целом.

Для оценки стоимости и трудоемкости проекта необходимо провести правильное планирование. Для этого были определены задачи и ресурсы, которые необходимы для создания web-приложения [21, 22].

В таблице 4 представлены задачи, которые представляют из себя этапы создания проекта.

Таблица 4 – Перечень задач проекта

№	Название задачи	Длительность	Предшественник и
1	Начало проекта	-	-
2	Анализ требований	-	-
3	Анализ аналогичных решений	7 дней	1
4	Анализ движения поручений и документов в ВУЗе	10 дней	3
5	Консультация с документоведами	2 дня	4
6	Принятие окончательного решения	1 день	5
7	Анализ требований завершен	-	6
8	Выбор средств реализации проекта	-	-
9	Анализ возможных программных средств	4 дня	1
10	Выбор программного средства для реализации	3 дня	9
11	Выбор средств завершен	-	10
12	Реализация проекта	-	-
13	Разработка структуры данных	2 дня	7; 11
14	Разработка пользовательского интерфейса	10 дней	7; 11; 13
15	Разработка модулей обработки данных	-	-
16	Программирование сопутствующих и обслуживающих классов	20 дней	14
17	Программирование логики движения поручений и документов	25 дней	16
18	Реализация проекта завершена	-	14; 17
19	Отладка	-	-
20	Тестирование пользовательского интерфейса	7 дней	18
21	Тестирование логики движения поручений и документов	10 дней	18
22	Исправление ошибок	10 дней	20; 21
23	Тестирование завершенно	-	22
24	Обучение работников ВУЗа	-	-
25	Принципы работы системы	3 дня	23
26	Изучение интерфейса	4 дня	25
27	Обучение завершенно	-	26
28	Внедрение программного средства в систему документооборота ВУЗа	-	-
29	Внедрение программного средства	5 дней	27
30	Поиск и исправление ошибок при внедрении	5 дней	29
31	Внедрение завершенно	-	30
32	Конец проекта	-	31

Из таблицы, представленной выше видно, что для реализации проекта необходимо выполнить 32 задачи.

В таблице 5 представлено описание затрачиваемых ресурсов.

Таблица 5 – Список ресурсов проекта

Название ресурса	Тип	Стандартная ставка	Ставка сверхурочных
Web-разработчик	Трудовой	200 Р/ч	250 Р/ч
Бизнес аналитик	Трудовой	200 Р/ч	250 Р/ч
Программист-тестировщик	Трудовой	150 Р/ч	200 Р/ч
Оператор	Трудовой	100 Р/ч	150 Р/ч
Дизайнер	Трудовой	150 Р/ч	200 Р/ч
Интернет	Затраты		
Компьютер	Материальный	500 Р	

Распределение ресурсов по задачам представлено в таблице 6.

Таблица 6 – Назначение ресурсов

№	Название задачи	Ресурс	Единицы
1	Начало проекта		
2	Анализ требований		
3	Анализ аналогичных решений	Web-разработчик Бизнес аналитик Компьютер Интернет	50% 50% 2
4	Анализ движения поручений и документов в ВУЗе	Web-разработчик Бизнес аналитик Программист-тестировщик Компьютер Интернет	50% 50% 50% 3 шт
5	Консультация с документоведами	Бизнес аналитик Компьютер Интернет	20% 1 шт
6	Принятие окончательного решения	Web-разработчик Бизнес аналитик	100% 100%
7	Анализ требований завершен		
8	Выбор средств реализации проекта		
9	Анализ возможных программных средств	Web-разработчик Компьютер Интернет	100% 1 шт
10	Выбор программного средства для реализации	Web-разработчик Компьютер Интернет	100% 1 шт
11	Выбор средств завершен		
12	Реализация проекта		
13	Разработка структуры данных	Web-разработчик Бизнес аналитик Компьютер Интернет	75% 25% 2 шт
14	Разработка пользовательского интерфейса	Web-разработчик Бизнес аналитик Дизайнер Компьютер Интернет	10% 50% 50% 3 шт
15	Разработка модулей обработки данных		

Продолжение таблицы 6

№	Название задачи	Ресурс	Единицы
16	Программирование сопутствующих и обслуживающих классов	Web-разработчик Компьютер Интернет	50% 1 шт
17	Программирование логики движения поручений и документов	Web-разработчик Компьютер Интернет	100% 1 шт
18	Реализация проекта завершена		
19	Отладка		
20	Тестирование пользовательского интерфейса	Бизнес аналитик Программист-тестировщик Компьютер Интернет	25% 75% 2 шт
21	Тестирование логики движения поручений и документов	Web-разработчик Бизнес аналитик Программист-тестировщик Компьютер Интернет	25% 50% 100% 3 шт
22	Исправление ошибок	Web-разработчик Бизнес аналитик Дизайнер Компьютер Интернет	75% 25% 50% 3 шт
23	Тестирование завершено		
24	Обучение работников ВУЗа		
25	Принципы работы системы	Web-разработчик Бизнес аналитик Оператор Компьютер	5% 10% 100% 1 шт
26	Изучение интерфейса	Бизнес аналитик Оператор Компьютер	10% 100% 1 шт
27	Обучение завершено		
28	Внедрение программного средства в систему документооборота ВУЗа		
29	Внедрение программного средства	Бизнес аналитик Web-разработчик Оператор	75% 25% 100%
30	Поиск и исправление ошибок при внедрении	Web-разработчик Бизнес аналитик Программист-тестировщик Компьютер Интернет	75% 25% 50% 3 шт
31	Внедрение завершено		
32	Конец проекта		

Из представленных таблиц видно, что реализацией проекта будут заниматься web-разработчик, бизнес аналитик, программист-тестировщик, дизайнер и оператор, которым для работы нужны компьютеры и доступ к интернету.

На рисунке 15 представлен список задач, внесенный в Microsoft Project. Началом проекта была принята дата 02.12.2019.

	🔍	Название задачи	Длительность	Начало	Окончание
1	📅	Начало проекта	1 день?	Пн 02.12.19	Пн 02.12.19
2	📅	Анализ требований	20 дней	Вт 03.12.19	Пн 30.12.19
3	📅	Анализ аналогичных решений	7 дней	Вт 03.12.19	Ср 11.12.19
4		Анализ движения поручений и документов в ВУЗе	10 дней	Чт 12.12.19	Ср 25.12.19
5		Консультация с документоведами	2 дней	Чт 26.12.19	Пт 27.12.19
6		Принятие окончательного решения	1 день	Пн 30.12.19	Пн 30.12.19
7	📅	Анализ требований завершен	0 дней	Пн 30.12.19	Пн 30.12.19
8	📅	Выбор средств реализации проекта	7 дней	Вт 03.12.19	Ср 11.12.19
9	📅	Анализ возможных программных средств	4 дней	Вт 03.12.19	Пт 06.12.19
10		Выбор программного средства для реализации	3 дней	Пн 09.12.19	Ср 11.12.19
11		Выбор средств завершен	0 дней	Ср 11.12.19	Ср 11.12.19
12	📅	Реализация проекта	57 дней	Вт 31.12.19	Ср 18.03.20
13		Разработка структуры данных	2 дней	Вт 31.12.19	Ср 01.01.20
14		Разработка пользовательского интерфейса	10 дней	Чт 02.01.20	Ср 15.01.20
15	📅	Разработка модулей обработки данных	45 дней	Чт 16.01.20	Ср 18.03.20
16		Программирование сопутствующих и обслуживающих классов	20 дней	Чт 16.01.20	Ср 12.02.20
17		Программирование логики движения поручений и документов	25 дней	Чт 13.02.20	Ср 18.03.20
18		Реализация проекта завершена	0 дней	Ср 18.03.20	Ср 18.03.20
19	📅	Отладка	20 дней	Чт 19.03.20	Ср 15.04.20
20		Тестирование пользовательского интерфейса	7 дней	Чт 19.03.20	Пт 27.03.20
21		Тестирование логики движения поручений и документов	10 дней	Чт 19.03.20	Ср 01.04.20
22		Исправление ошибок	10 дней	Чт 02.04.20	Ср 15.04.20
23		Тестирование завершено	0 дней	Ср 15.04.20	Ср 15.04.20
24	📅	Обучение работников ВУЗа	7 дней	Чт 16.04.20	Пт 24.04.20
25		Принципы работы системы	3 дней	Чт 16.04.20	Пн 20.04.20
26		Изучение интерфейса	4 дней	Вт 21.04.20	Пт 24.04.20
27		Обучение завершено	0 дней	Пт 24.04.20	Пт 24.04.20
28	📅	Внедрение программного средства в систему документооборота ВУЗа	10 дней	Пн 27.04.20	Пт 08.05.20
29		Внедрение программного средства	5 дней	Пн 27.04.20	Пт 01.05.20
30		Поиск и исправление ошибок при внедрении	5 дней	Пн 04.05.20	Пт 08.05.20
31		Внедрение завершено	0 дней	Пт 08.05.20	Пт 08.05.20
32		Конец проекта	0 дней	Пт 08.05.20	Пт 08.05.20

Рисунок 15 – Задачи проекта в среде Microsoft Project

На основе введенных данных в среде Microsoft Project была построена диаграмма Ганта, которая представлена на рисунке 16.

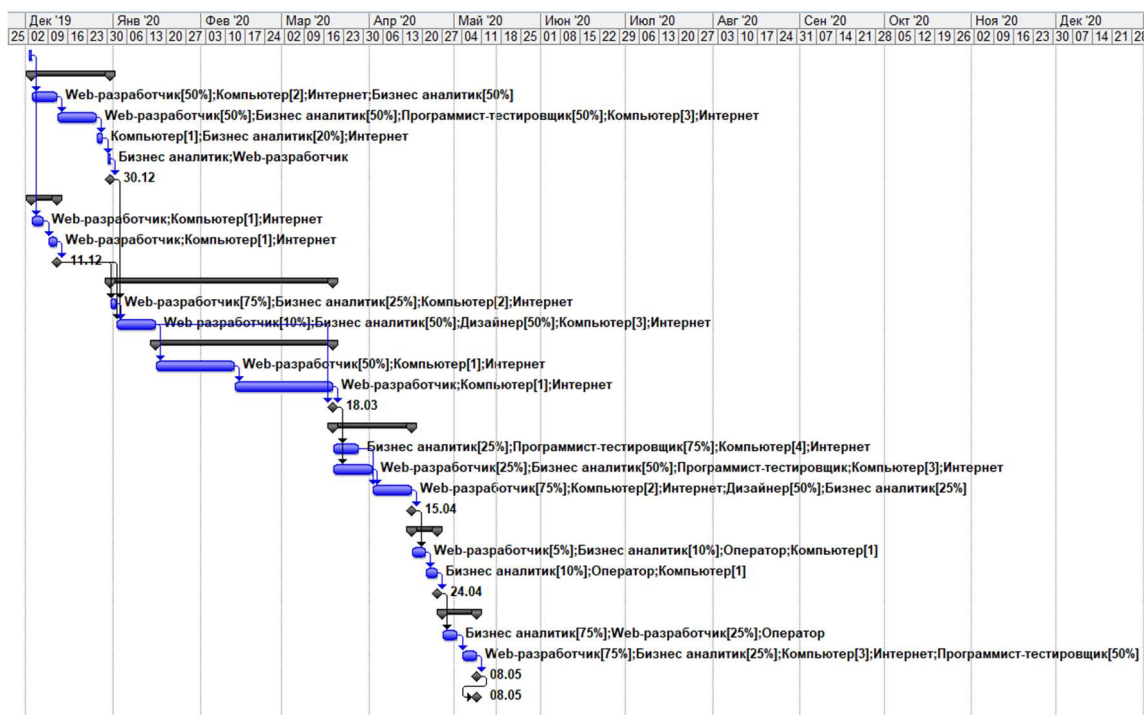
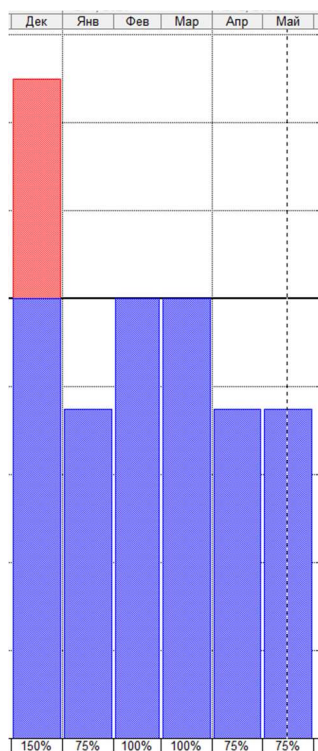
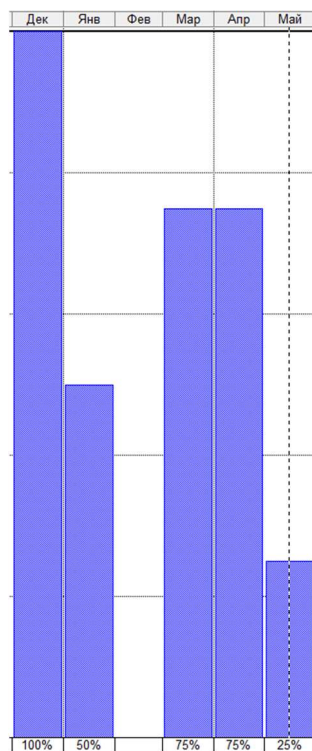


Рисунок 16 – Диаграмма Ганта проекта

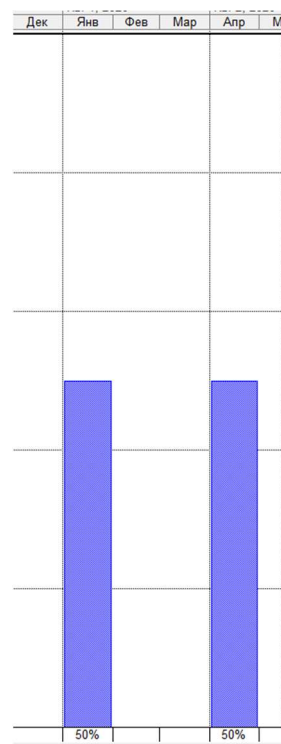
На рисунке 17 представлены графики трудовых ресурсов «Web-разработчик», «Бизнес аналитик», «Программист-тестировщик», «Оператор», «Дизайнер».



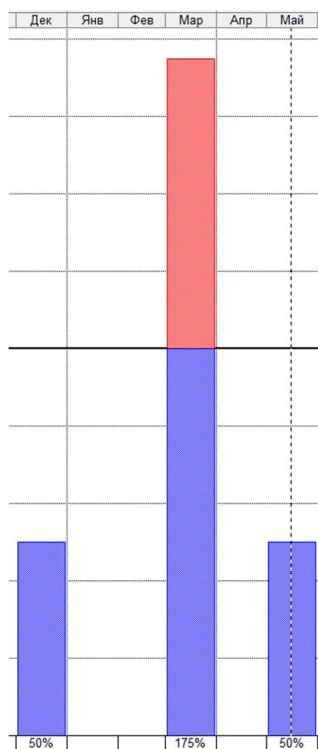
а) «Web-разработчик»



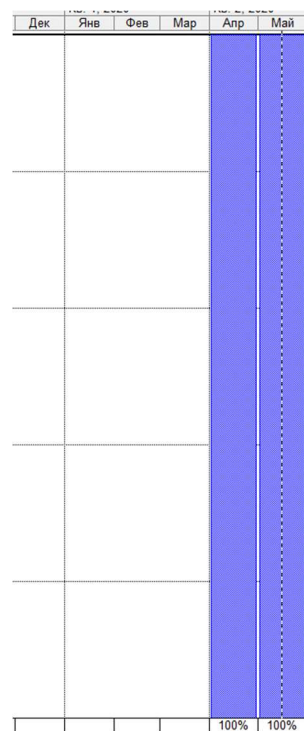
б) «Бизнес аналитик»



в) «Дизайнер»



г) «Программист-тестировщик»



д) «Оператор»

Рисунок 17 – Графики загрузки трудовых ресурсов

На рисунках видно, что у двух трудовых ресурсов во время проведения проекта нагрузка превышает нормальную. У ресурса «Web-разработчик» перегрузка наблюдается в декабре, а у ресурса «Программист-тестировщик» в марте.

На рисунке 18 представлен график загрузки материального ресурса «Компьютер».

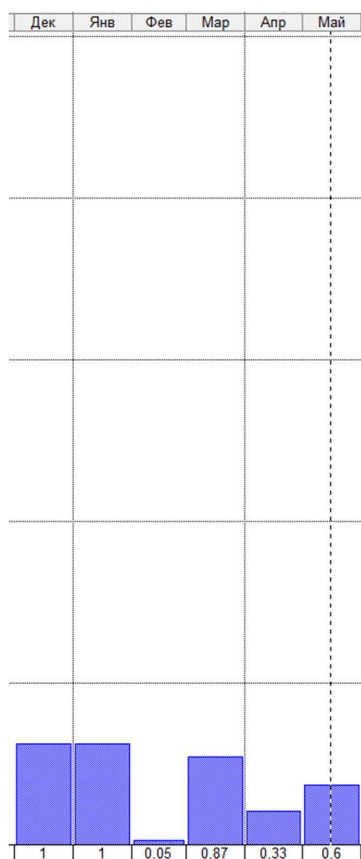


Рисунок 18 – График загрузки материального ресурса «Компьютер»

Из рисунка видно, что на протяжении всего проекта загрузка материального ресурса «Компьютер» не превышает нормальную.

На рисунке 19 представлен обзор проекта. Из рисунка видно, что общая длительность проекта составляет 115 дней, а предполагаемые затраты – 194 929,48 руб.

Даты			
Начало:	Пн 02.12.19	Окончание:	Пт 08.05.20
Баз. начало:	НД	Баз. окончание:	НД
Факт. начало:	НД	Факт. окончание:	НД
Отклон. начала:	0 дней	Отклон. окончания:	0 дней
Длительность			
Текущий план:	115 дней	Оставшаяся:	115 дней
Базовый план:	0 дней	Фактическая:	0 дней
Отклонение:	115 дней	Процент завершения:	0%
Трудозатраты			
Текущий план:	986,65 ч	Оставшиеся:	986,65 ч
Базовый план:	0 ч	Фактические:	0 ч
Отклонение:	986,65 ч	Процент завершения:	0%
Затраты			
Текущий план:	194 929,48 Р	Оставшиеся:	194 929,48 Р
Базовый план:	0,00 Р	Фактические:	0,00 Р
Отклонение:	194 929,48 Р		
Состояние задачи		Состояние ресурса	
Еще не начатые задачи	32	Трудовые ресурсы:	4
Выполняющиеся задачи	0	Трудовые ресурсы с прев. дос:	2
Завершенные задачи:	0	Материальные ресурсы:	1
Итого задач:	32	Итого ресурсов:	7

Рисунок 19 – Обзорный отчет проекта

Таким образом можно сделать вывод, что разработка собственного web-приложения является экономически целесообразной, поскольку стоимость разработки и внедрения значительно ниже аналогичных программных продуктов и функционал будет соответствовать непосредственным запросам вуза.

2.3 Проектирование логической модели базы данных

Проектирование логической модели служит для графического представления структуры базы. Оно производится с учетом выбранной модели данных, но не учитывает особенности конкретной СУБД. Перед непосредственным проектированием логической модели необходимо выделить основные сущности предметной области, их взаимосвязи и атрибуты, которые описывают их отдельные характеристики [23].

На основе анализа предметной области были выделены типы сущностей, представленные в таблице 7.

Таблица 7 – Типы основных сущностей предметной области

№ п/п	Наименование	Описание	Категория	Количество экземпляров	Режим обновления
1	Пользователь	Содержит сведения о пользователях	Стержневая	60000	8% в год
2	Исполнение пользователя	Содержит сведения о занимаемой должности пользователя	Стержневая	100000	5% в год
3	Подразделения	Содержит информацию о подразделениях вуза	Справочная	400	0% в год
4	Должность	Содержит информацию о должности пользователя	Справочная	3500	1% в год
5	Замещения	Содержит информацию о замещениях пользователей	Стержневая	20000	80% в год
6	Действия с документами	Содержит информацию о действиях с поручениями и документами	Стержневая	50000	100% в год
7	Тип документа	Содержит информацию о типах документов	Справочная	5	0% в год
8	Файл	Содержит информацию о файлах, прикрепленных к поручениям и документам	Стержневая	60000	100% в год
9	Текстовое сообщение	Содержит отдельные положения квитанции	Стержневая	50000	100% в год
10	Процесс	Содержит идентификатор уникального процесса	Справочная	25000	100% в год

Отношения между сущностями представлены в таблице 8.

Таблица 8 – Типы связей между сущностями

№ п/п	Тип сущности 1	Тип сущности 2	Описание	Мощность	Обязательность	Ограничения
1	2	3	4	5	6	7
1	Пользователь	Роль пользователя	Пользователь <i>Имеет</i> Роль пользователя Роль пользователя <i>Относится к</i> Пользователю	1:M	Обязательная (пользователь не может не иметь роль)	Без ограничений
2	Исполнение пользователя	Подразделение	Исполнение пользователя <i>Имеет</i> Подразделение Подразделение <i>Относится к</i> Исполнению пользователя	M:1	Обязательная (исполнение пользователя не может не иметь подразделения)	Без ограничений
3	Исполнение пользователя	Должность	Исполнение пользователя	M:1	Обязательная (исполнение)	Без ограничений

Продолжение таблицы 8

1	2	3	4	5	6	7
			<i>Имеет</i> Должность Должность <i>Относится</i> к Исполнению пользователя		пользователя не может не иметь должность)	
4	Подразделение	Подразделение	Подразделение <i>Имеет</i> Подразделение Подразделение <i>Относится</i> к Подразделени ю	1:M	Необязательная (подразделение может не меть родительского подразделения)	Без ограничений
5	Действия с документами	Действия с документами	Действия с документами <i>Имеет</i> Действия с документами Действия с документами <i>Относится</i> к Действиям с документами	1:M	Необязательная (действие с документом может не иметь родительского действия)	Без ограничений
6	Действия с документами	Тип документа	Действия с документами <i>Имеет</i> Тип документа Тип документа <i>Относится</i> к Действиям с документами	M:1	Обязательная (действие с документом не может не иметь тип документа)	Без ограничений
7	Файл	Текстовое сообщение	Файл <i>Имеет</i> Текстовое сообщение Текстовое сообщение <i>Относится</i> к Файлу	M:1	Обязательная (файл не может не иметь текстовое сообщение)	Без ограничений
8	Действия с документами	Текстовое сообщение	Действия с документами <i>Имеет</i> Текстовое сообщение Текстовое сообщение <i>Относится</i> к Действиям с документами	1:1	Обязательная (действие с документом не может не иметь текстовое сообщение)	Без ограничений
9	Действия с документами	Процесс	Действия с документами <i>Имеет</i> Процесс Процесс <i>Относится</i> к Действиям с документами	M:1	Обязательная (действие с документом не может не иметь процесс)	Без ограничений
10	Пользователь	Замещения	Пользователь <i>Имеет</i>	1:M	Необязательная (пользователь	Без ограничений

Окончание таблицы 8

1	2	3	4	5	6	7
			Замещения Замещения <i>Относится к</i> Пользователю		может не иметь замещений)	
11	Пользователь	Действия с документами	Пользователь Имеет Действия с документами Действия с документами <i>Относится к</i> Пользователю	1:M	Необязательная (пользователь может не действий с документами)	Без ограничений

Подробное описание атрибутов сущностей «Пользователь», «Должность», «Исполнение пользователя» и «Подразделение» представлено в таблицах 9 – 12.

Таблица 9 – Атрибуты типа сущности «Пользователь»

Наименование	Описание	Тип данных	Размерность	Область допустимых значений	Возможность значения Null	Роль
ID_Пользователь	Идентификатор пользователя	Целый	11	00000000000-99999999999	Нет	РК
Фамилия	Фамилия пользователя	Символьный	40	‘А-я’, ‘-’, ‘0-9’	Нет	
Имя	Имя пользователя	Символьный	40	‘А-я’, ‘-’, ‘0-9’	Нет	
Отчество	Отчество пользователя	Символьный	40	‘А-я’, ‘-’, ‘0-9’	Да	
Дата создания	Дата создания пользователя	Целый	15	00000000000000-9999999999999999	Нет	
Дата окончания действия	Дата удаления пользователя	Целый	15	0000000000000000-9999999999999999	Да	
Флаг удаления	Флаг удаления	Целый	1	0-9	Нет	

Таблица 10 – Атрибуты типа сущности «Должность»

Наименование	Описание	Тип данных	Размерность	Область допустимых значений	Возможность значения Null	Роль
ID_Должности	Идентификатор пользователя	Целый	15	0000000000000000-9999999999999999	Нет	РК
Наименование	Наименование должности	Символьный	100	‘А-я’, ‘-’, ‘0-9’	Нет	
Флаг удаления	Флаг удаления	Целый	1	0-9	Нет	

Таблица 11 – Атрибуты типа сущности «Исполнение пользователя»

Наименование	Описание	Тип данных	Размерность	Область допустимых значений	Возможность значения Null	Роль
ID_ Исполнения	Идентификатор исполнения пользователя	Целый	15	000000000000 0000-99999 99999999999	Нет	PK
ID_Пользователя	Идентификатор пользователя	Целый	15	000000000000 0000-99999 99999999999	Нет	FK
ID_Подразделения	Идентификатор подразделения	Целый	8	00000000- 99999999	Нет	FK
ID_Должности	Идентификатор должности	Целый	10	0000000000- 9999999999	Да	FK
Дата создания	Дата начала действия роли	Целый	15	000000000000 0000-99999 99999999999	Нет	
Дата завершения	Дата окончания действия роли	Целый	15	000000000000 0000-99999 99999999999	Да	
Флаг удаления	Флаг удаления	Целый	1	0-9	Нет	

Таблица 12 – Атрибуты типа сущности «Подразделения»

Наименование	Описание	Тип данных	Размерность	Область допустимых значений	Возможность значения Null	Роль
ID_Подразделения	Идентификатор подразделения	Целый	15	000000000000 0000-99999 99999999999	Нет	PK
ID_Родительского подразделения	Идентификатор родительского подразделения	Целый	15	000000000000 0000-99999 99999999999	Нет	FK
Наименование	Наименование подразделения	Символьный	100	‘А-я’, ‘-’, ‘0-9’	Нет	
Флаг удаления	Флаг удаления	Целый	1	0-9	Нет	

Описание атрибутов сущностей «Файл», «Тип документа», «Замещения» и «Процесс» представлены в таблицах 13 – 16.

Таблица 13 – Атрибуты типа сущности «Файл»

Наименование	Описание	Тип данных	Размерность	Область допустимых значений	Возможность значения Null	Роль
ID_Файла	Идентификатор файла	Целый	15	000000000000 0000-99999 99999999999	Нет	PK
ID_Текстового сообщения	Идентификатор текста	Целый	15	000000000000 0000-99999 99999999999	Нет	FK
Наименование	Наименование файла	Символьный	40	‘А-z’, ‘-’, ‘0-9’	Нет	
Флаг удаления	Флаг удаления	Целый	1	0-9	Нет	

Таблица 14 – Атрибуты типа сущности «Тип документа»

Наименование	Описание	Тип данных	Размерность	Область допустимых значений	Возможность значения Null	Роль
ID_Типа документа	Идентификатор типа документа	Целый	15	000000000000 0000-999999 9999999999	Нет	PK
Наименование	Наименование типа документа	Символьный	40	‘А-я’, ‘-’, ‘0-9’	Нет	

Таблица 15 – Атрибуты типа сущности «Замещения»

Наименование	Описание	Тип данных	Размерность	Область допустимых значений	Возможность значения Null	Роль
ID_Замещения	Идентификатор замещения	Целый	10	0000000000- 9999999999	Нет	PK
ID_Начальника	Идентификатор пользователя, которого замещают	Целый	15	000000000000 0000-99999 9999999999	Нет	FK
ID_Заместителя	Идентификатор пользователя, который замещает	Целый	15	000000000000 0000-99999 9999999999	Нет	FK
Дата начала действия	Дата начала действия замещения	Целый	15	000000000000 0000-99999 9999999999	Да	
Дата окончания действия	Дата окончания действия замещения	Целый	15	000000000000 0000-99999 9999999999	Нет	
Режим замещения	Режим замещения	Целый	1	0-9	Нет	0
Тип замещения	Тип замещения	Целый	1	0-9	Нет	
Флаг удаления	Флаг удаления	Целый	1	0-9	Нет	

Таблица 16 – Атрибуты типа сущности «Текстовое сообщение»

Наименование	Описание	Тип данных	Размерность	Область допустимых значений	Возможность значения Null	Роль
ID_Текста	Идентификатор текста	Целый	15	000000000000 0000-99999 9999999999	Нет	PK
Наименование	Идентификатор текста	Символьный	100	000000000000 0000-99999 9999999999	Нет	FK
Содержание	Содержание текста	Символьный	255	‘А-z’, ‘А-я’, ‘-’, ‘0-9’, ‘<’, ‘>’, ‘/’	Нет	
Флаг удаления	Флаг удаления	Целый	1	0-9	Нет	

Описание атрибутов сущностей «Текстовое сообщение» и «Действия с документами» представлены в таблицах 17 и 18 соответственно.

Таблица 17 – Атрибуты типа сущности «Процесс»

Наименование	Описание	Тип данных	Размерность	Область допустимых значений	Возможность значения Null	Роль
ID_Процесса	Идентификатор процесса	Целый	15	000000000000 0000-99999 99999999999	Нет	PK
Флаг удаления	Флаг удаления	Целый	1	0-9	Нет	

Таблица 18 – Атрибуты типа сущности «Действия с документами»

Наименование	Описание	Тип данных	Размерность	Область допустимых значений	Возможность значения Null	Роль
ID_Действия	Идентификатор действия	Целый	15	000000000000 0000-99999 99999999999	Нет	PK
ID_Родительского движения	Идентификатор родительского движения	Целый	15	000000000000 0000-99999 99999999999	Нет	FK
ID_Типа документа	Идентификатор типа документа	Целый	15	000000000000 0000-9999 99999999999	Нет	FK
Номер	Номер документа	Символьный	100	‘А-я’, ‘-’, ‘0-9’	Да	
ID_Инициатора	Идентификатор пользователя инициатора	Целый	15	000000000000 0000-99999 99999999999	Нет	FK
ID_Заместителя	Идентификатор пользователя заместителя	Целый	15	000000000000 0000-99999 99999999999	Да	FK
ID_Исполнителя	Идентификатор пользователя исполнителя	Целый	15	000000000000 0000-99999 99999999999	Да	FK
ID_Наблюдателя	Идентификатор пользователя наблюдателя	Целый	15	000000000000 0000-99999 99999999999	Да	FK
ID_Текста	Идентификатор текста	Целый	15	000000000000 0000-99999 99999999999	Нет	FK
ID_Документа	Идентификатор документа	Целый	10	00000000000- 99999999999	Нет	FK
Дата создания	Дата создания	Целый	15	000000000000 0000-99999 99999999999	Нет	
Дата окончания	Дата окончания	Целый	15	000000000000 0000-99999 99999999999	Да	
Дата фактического окончания	Дата фактического окончания	Целый	15	000000000000 0000-99999 99999999999	Да	
Дата просмотра	Дата просмотра	Целый	15	000000000000 0000-99999 99999999999	Да	
Флаг отслеживания	Флаг отслеживания	Целый	1	0-9	Нет	
Флаг закрытия	Флаг закрытия	Целый	1	0-9	Нет	

Продолжение таблицы 18

Наименование	Описание	Тип данных	Размерность	Область допустимых значений	Возможность значения Null	Роль
Флаг подписания	Флаг подписания	Целый	1	0-9	Нет	
Хэш-код	Хэш-код пользователя, подписавшего документ	Символьный	40	'A-z', '0-9'	Да	
Флаг ответственного	Флаг ответственного	Целый	1	0-9	Нет	
Флаг удаления	Флаг удаления	Целый	1	0-9	Нет	

На основании выделенных сущностей, их атрибутов и связей между ними была построена логическая модель базы данных в нотации IDEF1X в среде ERwin Data Modeler. Логическая модель представлена на рисунке 20.

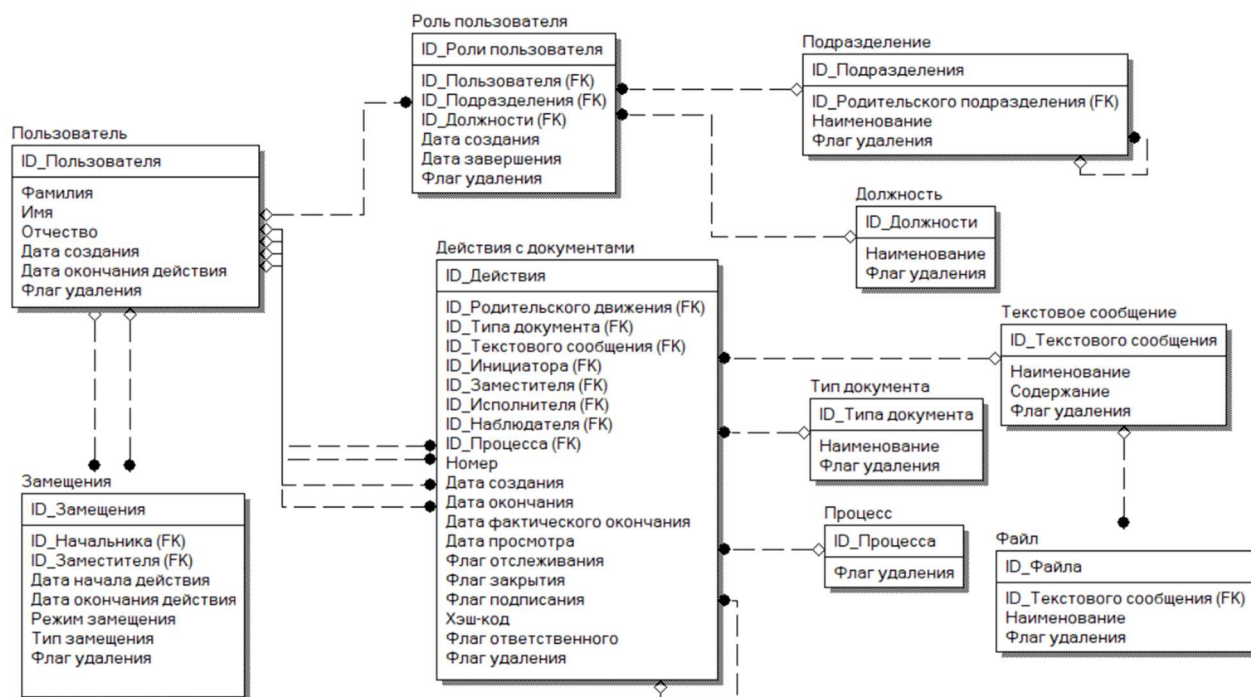


Рисунок 20 – Логическая модель базы данных

На основе полученной логической модели базы данных будет осуществлен переход к физической модели и ее реализации в целевой СУБД.

2.4 Проектирование пользовательского интерфейса

На показатели качества функциональности, производительности и эстетики внешнего вида конечного программного продукта оказывает влияние

дизайн пользовательского интерфейса. Поэтому при проектировании пользовательского интерфейса необходимо придерживаться основных правил [24, 25]:

- простой и понятный интерфейс;
- минимальное количество действий при решении задачи;
- привычные элементы управления;
- не более 3-х нажатий для перехода из одного раздела в другой;
- однородный функционал.

На основе анализа предметной области, требований к разрабатываемой системе и основных принципов создания дизайна были выделены следующие виды пользовательских интерфейсов:

- главная страница;
- страница документов;
- страница замещений;
- страница создания поручения/подписания;
- страница создания документа;
- страница создания замещения.

Прототип интерфейса главной страницы, представленный на рисунке 21, содержит большое количество элементов, которые некорректно масштабируются при изменении разрешения экрана, поэтому было разработано два прототипа интерфейса. Для устройств с разрешением экрана 480×800 и менее было принято решение вынести элемент, в котором осуществляется предварительный просмотр задачи или документа, в модальное окно, а на устройствах с разрешением более 480×800 информацию для предварительного просмотра размещать в боковом всплывающем окне.

На рисунке 21 представлен интерфейс для устройств с разрешением экрана более 480×800, а на рисунке 22 представлен интерфейс для устройств с разрешением экрана 480×800 и менее.

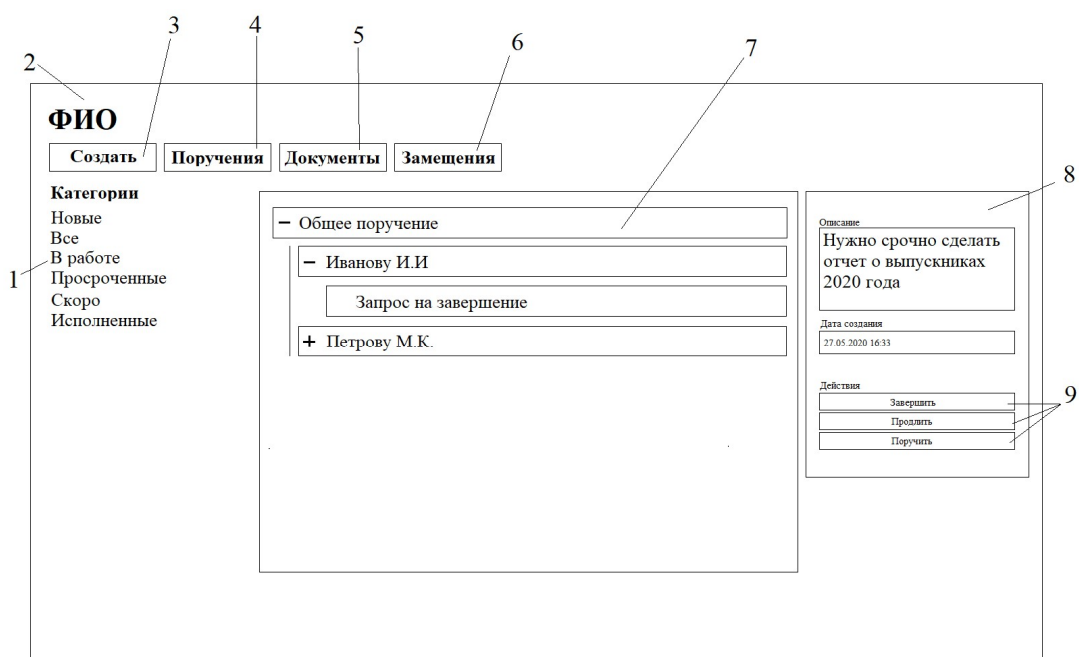


Рисунок 21 – Интерфейс главной страницы для экранов более 480×800

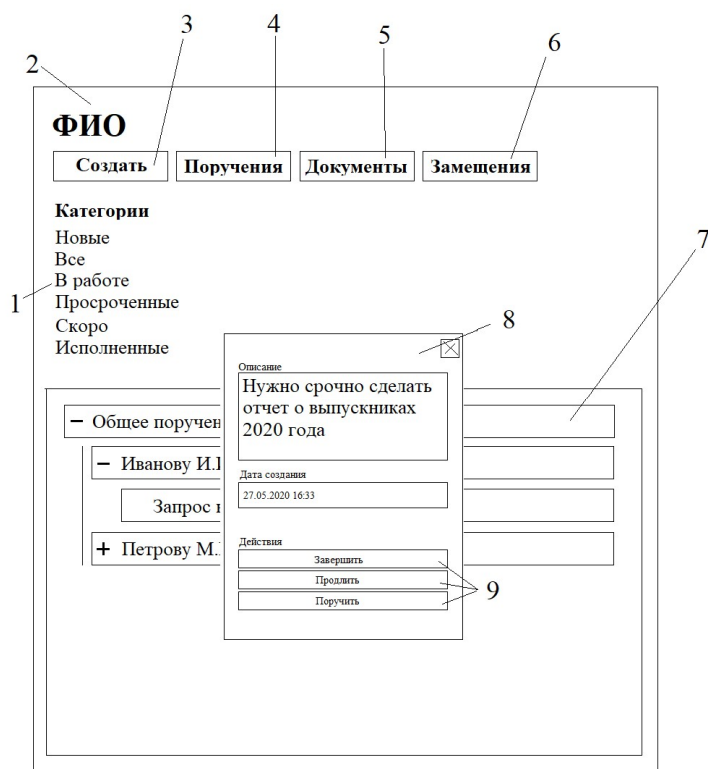


Рисунок 22 – Интерфейс главной страницы для экранов с разрешением 480×800 и менее

Цифрами на рисунках 21 и 22 отмечены:

1 – меню выбора категорий;

- 2 – ФИО пользователя;
- 3 – кнопка для перехода на страницу создания поручения;
- 4 – кнопка перехода на страницу поручений (главная страница);
- 5 – кнопка перехода на страницу документов;
- 6 – кнопка перехода на страницу заместителей;
- 7 – дерево запросов;
- 8 – окно предварительного просмотра запроса из дерева (смотри рисунок 21) или модальное окно (смотри рисунок 22);
- 9 – доступные действия с запросом.

Окно предварительного просмотра запроса открывается при нажатии на запрос в дереве и содержит краткую информацию о нем. Кнопки действий меняются в зависимости от типа запроса и роли пользователя по отношению к нему. Описание возможных действий представлено в таблице 19.

Таблица 19 – Возможные кнопки и описание их действий

Роль Тип	Инициатор	Исполнитель
Документ	«Поручить», по нажатию этой кнопки, инициатор создает новое поручение на основе документа.	Доступных действий нет.
Поручение	«Отменить запрос», по нажатию этой кнопки происходит отмена выданного поручения. Она доступна только если поручение еще не просмотрено исполнителем.	«Завершить», по нажатию этой кнопки создается запрос на завершение от исполнителя к инициатору. «Продлить», по нажатию этой кнопки создается запрос на продление от исполнителя к инициатору с новой датой завершения поручения. «Поручить», по нажатию этой кнопки, инициатор создает новое поручение на основе своего.
Подписание	«Отменить запрос», по нажатию этой кнопки происходит отмена выданного подписания. Она доступна только если запрос подписания еще не просмотрен исполнителем.	«Подписать», по нажатию этой кнопки подтверждается подписание. «Отклонить», по нажатию кнопки отклоняется подписание.
Запрос продления	«Принять», по нажатию этой кнопки запрос о продлении выполнения поручения подтверждается. «Отклонить», по нажатию этой кнопки запрос о продлении выполнения поручения отклоняется.	«Отменить запрос», по нажатию этой кнопки происходит отмена запроса продления.

Продолжение таблицы 19

Роль Тип	Инициатор	Исполнитель
Запрос завершения	«Принять», по нажатию этой кнопки запрос о завершении поручения подтверждается. «Отклонить», по нажатию этой кнопки запрос о завершении поручения отклоняется.	«Отменить запрос», по нажатию этой кнопки происходит отмена запроса завершения.

На главной странице при нажатии на кнопку «Создать» открывается страница с формой для создания нового поручения. Формы для создания поручения и подписания должны содержать одинаковые элементы, таким образом их интерфейсы совпадают.

Интерфейс страницы создания нового поручения и подписания представлен на рисунке 23.

Рисунок 23 – Интерфейс страницы для создания нового поручения или подписания

На этом рисунке цифрами отмечены следующие элементы:

- 1 – выпадающий список для выбора инициатора, в нем содержится сам пользователь и его замещения (пользователи, которых он замещает);
- 2 – выпадающий список для выбора исполнителя или согласующего;
- 3 – выпадающий список для выбора ответственного исполнителя или согласующего из уже выбранных в предыдущем выпадающем списке;
- 4 – выпадающий список для выбора наблюдателя;
- 5 – поля для общего описания запроса;
- 6 – поле выбора файлов;
- 7 – поле с датой окончания;
- 8 – кнопка сохранения запроса.

При нажатии на кнопку «Документы» открывается страница со списком документов. Интерфейс этой страницы представлен на рисунке 24.



Рисунок 24 – Интерфейс страницы документов

Цифрами на рисунке отмечены:

- 1 – кнопка создания нового документа;
- 2 – таблица со списком документов;

3 – кнопка просмотра списка поручений, выданных на основе документа;

4 – кнопка создания подписания на основе документа;

5 – кнопка создания поручения на основе документа.

На странице документов при нажатии на кнопку «Создать» открывается страница с формой для создания нового документа. Интерфейс этой страницы представлен на рисунке 25.

Рисунок 25 – Интерфейс страницы создания нового документа

Цифрами на рисунке 25 отмечены следующие элементы:

1 – выпадающий список с типами документов;

2 – выпадающий список с выбором создателя документа;

3 – поля для описания документа;

4 – поле даты присвоения номера;

5 – поле номера документа;

6 – поле выбора файлов;

7 – кнопка сохранения документа.

Поля под номерами 4 и 5 может заполнять и редактировать только

создатель документа после его успешного подписания.

При нажатии на кнопку «Замещения» происходит переход на страницу со списком заместителей и замещений пользователя. Интерфейс страницы представлен на рисунке 26.

ФИО

Создать | Поручения | Документы | Замещения

Мои заместители

ФИО	Дата начала	Дата окончания	Тип	Статус	
Иванов И.И.	-		Постоянное	Удалено	
Сидоров П.В.	27.05.2020	03.06.2020	Временное	Действует	☒

Мои замещения

ФИО	Дата начала	Дата окончания	Тип	Статус	

Рисунок 26 – Интерфейс страницы заместителей

Цифрами на этом рисунке отмечены:

- 1 – кнопка создания замещения;
- 2 – таблица со списком заместителей (кто замещает) пользователя
- 3 – кнопка удаления замещения;
- 4 – таблица со списком замещений (кого замещает) пользователя.

При нажатии на кнопку «Удалить» снимаются обязанности с заместителя, но в списке он будет отображаться со статусом «Удалено».

На странице заместителей при нажатии на кнопку «Создать» происходит переход на страницу создания нового замещения. Интерфейс этой страницы представлен на рисунке 27.

ФИО

Поручения Документы **Замещения**

1 Кого замещает
Выберите V

2 Кто замещает
Выберите V

3 Дата начала

4 Дата окончания

5 ☐ Постоянное замещение

6 Сохранить

Рисунок 27 – Интерфейс страницы создания нового замещения

Цифрами на рисунке отмечены следующие элементы:

- 1 – ФИО пользователя;
- 2 – выпадающий список для выбора заместителя;
- 3 – поле даты начала действия замещения;
- 4 – поле даты окончания действия замещения;
- 5 – поле выбора типа замещения;
- 6 – кнопка сохранения замещения.

При выборе «постоянного» типа замещения поля под номерами 3 и 5 становятся недоступными.

Интерфейсы, представленные на рисунках 23-27 подходят для устройств с различными разрешениями экранов, поэтому прототипы для отдельных групп разрешений не требуются.

Таким образом, пользовательский интерфейс, полученный в результате проектирования, является простым, удобным, интуитивно понятным и адаптивным под различные разрешения экранов.

2.5 Проектирование архитектуры программного обеспечения

При разработке программного обеспечения огромное значение имеет грамотное построение архитектуры программы, которую в дальнейшем будет легко поддерживать и расширять. Адаптивная разработка подразумевает использование принципов Single responsibility, Open-closed, Liskov substitution, Interface segregation и Dependency inversion (SOLID), поэтому проектирование архитектуры web-приложения производилось с соблюдением этих подходов [26].

Все классы, разработанные для web-приложения, можно разделить на 2 группы:

- классы, обеспечивающие работу с задачами (поручениями, подписаниями) и документами;
- сопутствующие (обслуживающие) классы, которые позволяют работать с данными о пользователе, разграничивать права доступа, проверять корректность введенных данных и т.д.

В системе есть алгоритмы работы с поручением, с запросом подписания и документом, которые имеют схожую направленность, но различную реализацию. Поэтому для разработки классов, относящихся к первой группе, использовался поведенческий шаблон проектирования «Стратегия». Он позволяет определить группу схожих алгоритмов и помещает каждый из них в отдельный класс, затем их можно менять непосредственно во время исполнения программы [27].

Классы, которые отвечают за работу с поручением, подписанием и документом должны иметь общий интерфейс. Он будет определять обязательный набор методов, а классы, реализующие его, будут содержать конкретное поведение объекта.

Доступ к реализациям интерфейса осуществляется через отдельный класс. Благодаря этому расширение или изменение алгоритмов работы с поручением, подписанием или документом будет производиться в конкретных реализациях, а исходный код, который обеспечивает доступ к ним затрагиваться не будет. Этот

подход соответствует одному из принципов SOLID – принципу открытости и закрытости [26].

К первой группе относятся следующие классы:

- «DocumentController» – класс, который хранит ссылку на объект конкретной стратегии;
- «IStrategy» – интерфейс стратегии, который является общим для всех вариантов реализаций;
- «Document» – реализация работы с документами;
- «Sign» – реализация работы с процессом подписания;
- «Errand» – реализация работы с поручением.

Разработка сопутствующих классов проводилась на основе еще одного принципа SOLID – принципа единственной ответственности. Это значит, что каждый класс должен отвечать за одну направленность. Этот подход позволяет избегать сложных взаимосвязей и снижать риски возникновения ошибок при модификации программного кода. К этой группе относятся следующие классы:

- «DBUtil» – класс, реализующий методы работы с записями в базе данных;
- «DocumentationHome» – класс, реализующий методы для получения информации о запросах, их статусе и типе;
- «PrintHTMLHome» – класс, реализующий методы для формирования HTML элементов для отображения на странице;
- «UserHome» – класс, реализующий методы для получения информации о пользователе;
- «RightsHome» – класс, реализующий методы для определения прав пользователей;
- «ValidationHome» – класс, реализующий методы для проверки корректности данных и действий пользователя.

На основе рассмотренных ранее принципов и в соответствии с предметной областью была разработана диаграмма классов для серверной части web-приложения, представленная на рисунке 28.

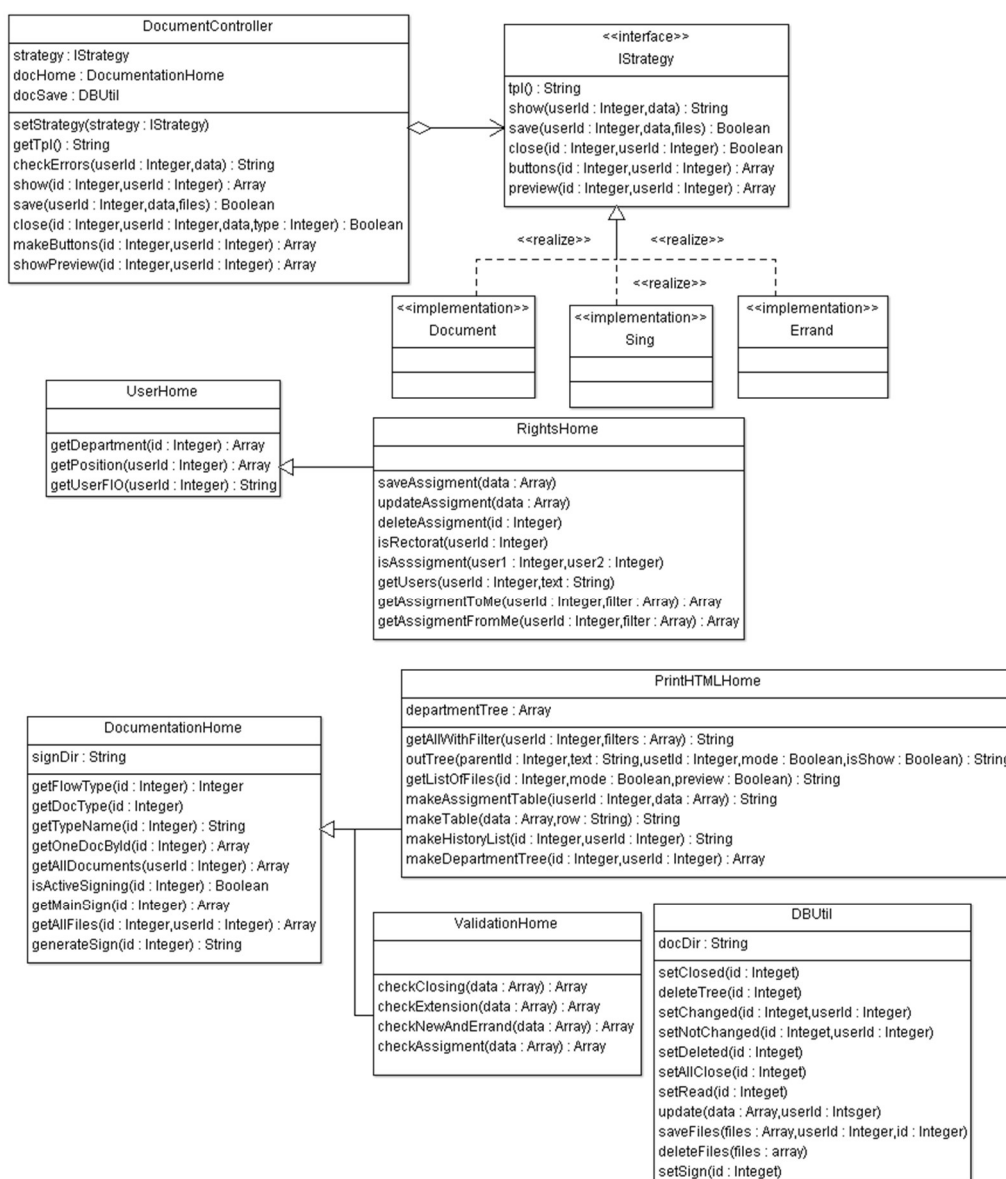


Рисунок 28 – Диаграмма классов для web-приложения

Таким образом, использование принципов SOLID и шаблона проектирования «Стратегия» дает возможность расширять систему путем добавления нового класса с необходимым функционалом и минимальными изменениями программного кода вне этого класса и упрощает дальнейшее сопровождение и понимание программного кода.

3 Реализация web-приложения

3.1 Реализация базы данных

Реализация базы данных производилась на основе разработанной логической модели базы данных. Развертывание выполнено путем запуска SQL-скрипта в web-приложении PhpMyAdmin, которое представляет собой web-интерфейсе для администрирования СУБД MySQL. Физическая модель, сгенерированная программным средством для работы с базой данных, представлена на рисунке 29.

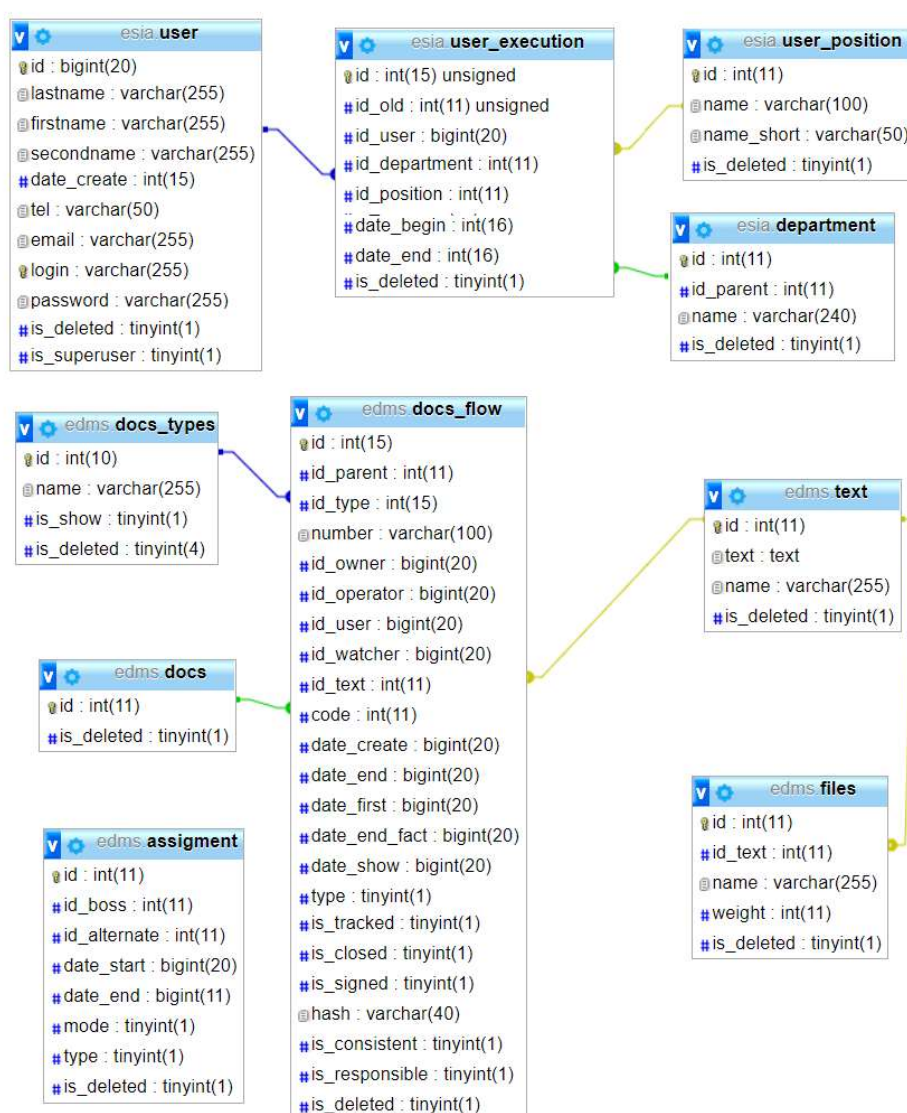


Рисунок 29 – Результат выполнения сгенерированного скрипта

Исходный код SQL-скрипта создания базы данных представлен в

приложении В.

Разрабатываемая система внедрялась в существующую корпоративную информационную систему вуза, поэтому все таблицы, кроме таблиц, содержащих данные о пользователях, были помещены в отдельную базу данных «edms». Остальные таблицы, которые были выделены в логической модели базы данных, такие как users, user_execution, department, position, были созданы до начала разработки системы и расположены в базе данных «esia». Поэтому целостность данных, которая должна быть реализована между таблицами из разных баз данных, поддерживается на уровне программного кода.

3.2 Реализация клиентской части web-приложения

Формирование страницы web-приложения производится на основе TPL-шаблонов. Их применение вместе с PHP-скриптами позволяет облегчить создание web-приложения.

TPL-шаблоны представляют собой файлы с HTML-разметкой, в которых изменяющиеся элементы заменены на переменные в фигурных скобках. Необходимая информация внедряется в шаблон с помощью функции parse. Она принимает 2 аргумента: путь к файлу TPL-шаблона и массив данных, которые необходимо разместить в шаблоне. Таким образом, функция parse выполняет замены переменных на соответствующие значения из переданного массива. Пример TPL-шаблона для модального окна представлен в листинге 1.

Листинг 1 – TPL-шаблон для модального окна

```
<div class="modal-header">
    <h5 id="title" class="modal-title">{#title}</h5>
</div>
<div class="modal-body">
    {#body}
</div>
<div class="modal-footer">
    <button type="button" id="close_doc" class="btn btn-secondary" data-
dismiss="modal">Отмена</button>
    <button type="submit" id="doc" onclick = "" class="btn btn-
primary">Сохранить</button>
</div>
```

Такой механизм позволяет избежать создания множества файлов с HTML-

разметкой, делает создаваемые шаблоны гибкими и позволяет повторное использование на разных страницах.

Адаптивность отображения элементов страницы на различных разрешениях экрана поддерживается с помощью библиотеки Bootstrap v4.0.

3.3 Реализация серверной части web-приложения

Для реализации серверной части использовался язык программирования PHP. Разрабатываемая система является частью личного кабинета пользователя ПГУ с ролью сотрудник. Она имеет полностью независимый от остальных модулей функционал. Серверная часть включает в себя 2 части:

- файла `index.inc.php` и `ajax.inc.php`, в которых обрабатываются http запросы от клиентской части web-приложения;
- отдельные классы, которые описывают бизнес логику работы web-приложения и обеспечивают работу с базой данных.

При начальной загрузке страниц web-приложения запросы для отображения соответствующих данных отправляются в файл `index.inc.php`. Он служит входной точкой на страницы разрабатываемой системы. Запросы, обрабатываемые в данном файле, представлены в таблице 20.

Таблица 20 – Запросы, обрабатываемые в файле `index.inc.php`

URL запроса	Назначение
<code>https://lk.pnzgu.ru/edms/{ID_USER}</code>	Генерация начальной страницы с поручениями пользователя.
<code>https://lk.pnzgu.ru/edms/{ID_USER}/documents</code>	Генерация страницы со списком документов, к которым имеет доступ пользователь.
<code>https://lk.pnzgu.ru/edms/{ID_USER}/assignments</code>	Генерация страницы со списками замещений и заместителей пользователя.
<code>https://lk.pnzgu.ru/edms/{ID_USER}/create/errand</code>	Генерация страницы создания поручения.
<code>https://lk.pnzgu.ru/edms/{ID_USER}/create/document</code>	Генерация страницы создания документа.
<code>https://lk.pnzgu.ru/edms/{ID_USER}/errand/{ID}</code>	Генерация страницы просмотра конкретного существующего поручения, к которому имеет доступ пользователь.
<code>https://lk.pnzgu.ru/edms/{ID_USER}/create/assignment</code>	Генерация страницы создания заместителя.

Продолжение таблицы 20

URL запроса	Назначение
https://lk.pnzgu.ru/edms/{ID_USER}/document/{ID}	Генерация страницы просмотра конкретного существующего документа, к которому имеет доступ пользователь.
https://lk.pnzgu.ru/edms/{ID_USER}/assignment/{ID}	Генерация страницы просмотра конкретного существующего замещения, к которому имеет доступ пользователь.

При выполнении каких-либо действий на клиентской части web-приложения происходит асинхронная отправка GET или POST запросов в файл `ajax.inc.php`, в котором формируется ответ на запросы. Ответ может представлять собой HTML-блок для модального окна, сообщение об ошибке или удачном выполнении и т.д. Запросы, обрабатываемые в файле `ajax.inc.php` представлены в таблице 21.

Таблица 21 – Запросы, обрабатываемые в файле `ajax.inc.php`

URL запроса	Метод	Назначение
https://lk.pnzgu.ru/ajax/edms/documentation/preview/{ID_USER}/d/{ID}	GET	Генерация блока предварительного просмотра поручения или документа.
https://lk.pnzgu.ru/ajax/edms/documentation/filter/user/{ID_USER}	GET	Генерация содержимого для блоков, находящихся на странице, на основе введенных фильтров.
https://lk.pnzgu.ru/ajax/edms/documentation/user/{ID_USER}/?view={TYPE}	GET	Генерация списка поручений по конкретной категории.
https://lk.pnzgu.ru/ajax/edms/documentation/assignment/{ID}	POST	Удаление замещения.
https://lk.pnzgu.ru/ajax/edms/documentation/updateable/user/{ID_USER}	GET	Генерация таблиц с замещениями и заместителями для их обновления без перезагрузки страницы.
https://lk.pnzgu.ru/ajax/edms/documentation/doc/user/{ID_USER}	POST	Сохранение запроса на создание поручения, подписания, документа, запроса завершения и т.д.
https://lk.pnzgu.ru/ajax/edms/documentation/close/id/{ID}	GET	Генерация модального окна для закрытия поручения.

Основная логика работы web-приложения и доступ к базе данных осуществляется в отдельных классах, структура которых была разработана в ходе проектирования архитектуры программного кода. Описание методов разработанных классов представлено в таблице 22.

Таблица 22 – Описание методов классов

Название	Методы
DocumentController.class.php	setStrategy – сеттер для выбора стратегии действий; getTpl – шаблон создания; checkDocument – проверка корректности данных; showDocument – полная информация для показа запроса; saveDocument – сохранение запроса; closeDocument – закрытие запроса; makeButtons – формирование кнопок управления; showPreview – данные для предварительного просмотра запроса.
IStrategy.php	tpl – шаблон формы; check – проверка корректности данных; show – данные для просмотра запроса; save – сохранение запроса; close – закрытие запроса; buttons – кнопку управления для конкретного типа запроса; preview – данные для формы предварительного просмотра для конкретного типа запроса.
Document.class.php	Имеют те же методы, которые заявлены в интерфейсе IStrategy.php.
Sign.class.php	
Errand.class.php	
ValidationHome.class.php	checkClosing – проверка данных по запросу закрытия; checkExtension – проверка данных по запросу продления; checkNewAndErrand – проверка данных по поручению; checkAssignment – проверка данных по замещению.
UserHome.class.php	getDepartment – получение подразделения к которому относится пользователь; getPosition – получение должности пользователя; getUserFIO – получение ФИО пользователя.
DocumentationHome.class.php	getFlowType – возвращает список типов запросов; getDocType – возвращает тип конкретного запроса; getTypeName – возвращает имя конкретного типа запроса; getOneDocById – возвращает все данные по запросу; getAllDocuments – возвращает список всех документов; isActiveSigning – проверка, находится ли документ в процессе подписания; getMainSign – возвращает ответственного согласующего в подписании.
PrintHTMLHome.class.php	getAllWithFilter – формирование списка запросов подходящих под выбранные фильтры; outTree – формирование дерева одного запроса; makeAssignmentTable – формирование таблицы замещений; makeTable – формирование любой таблицы; makeOption – формирование элементов выпадающего списка; makeHistoryList – формирование истории конкретного запроса; getListOfFiles – формирование списка прикрепленных файлов.
RightsHome.class.php	saveAssignment – сохранение нового замещения пользователя; updateAssignment – изменение замещения; deleteAssignment – удаление замещения; getOneAssignment – данные об одном замещении; isRectorat – проверка, входит ли пользователь в ректорат; isAssignment – проверка, является ли пользователь заместителем; checkResponsible – проверка, является ли пользователь ответственным по поручению или подписанию; getUsers – получение списка пользователей подразделения; getAssignmentToMe – список людей, которых замещает пользователь; getAssignmentFromMe – список людей, которые замещают пользователя.
DBUtil.class.php	setClosed – помечает запись как завершенная; deletedTree – помечает все записи, которые входят в дерево запроса, как удаленные; setChanged – помечает запись как измененная; setNotChanged – помечает запись как не измененная;

Продолжение таблицы 22

Название	Методы
	setDeleted – помечает одну запись как удаленная; setAllClose – помечает все записи, как завершенные; setRead – ставит записи статус прочтена; update – обновление записи; saveFiles – сохранение прикрепленных файлов; deleteFiles – удаление прикрепленных файлов; setSign – помечает запись как подписанная.

Диаграмма компонентов разработанного web-приложения представлена на рисунке 30.

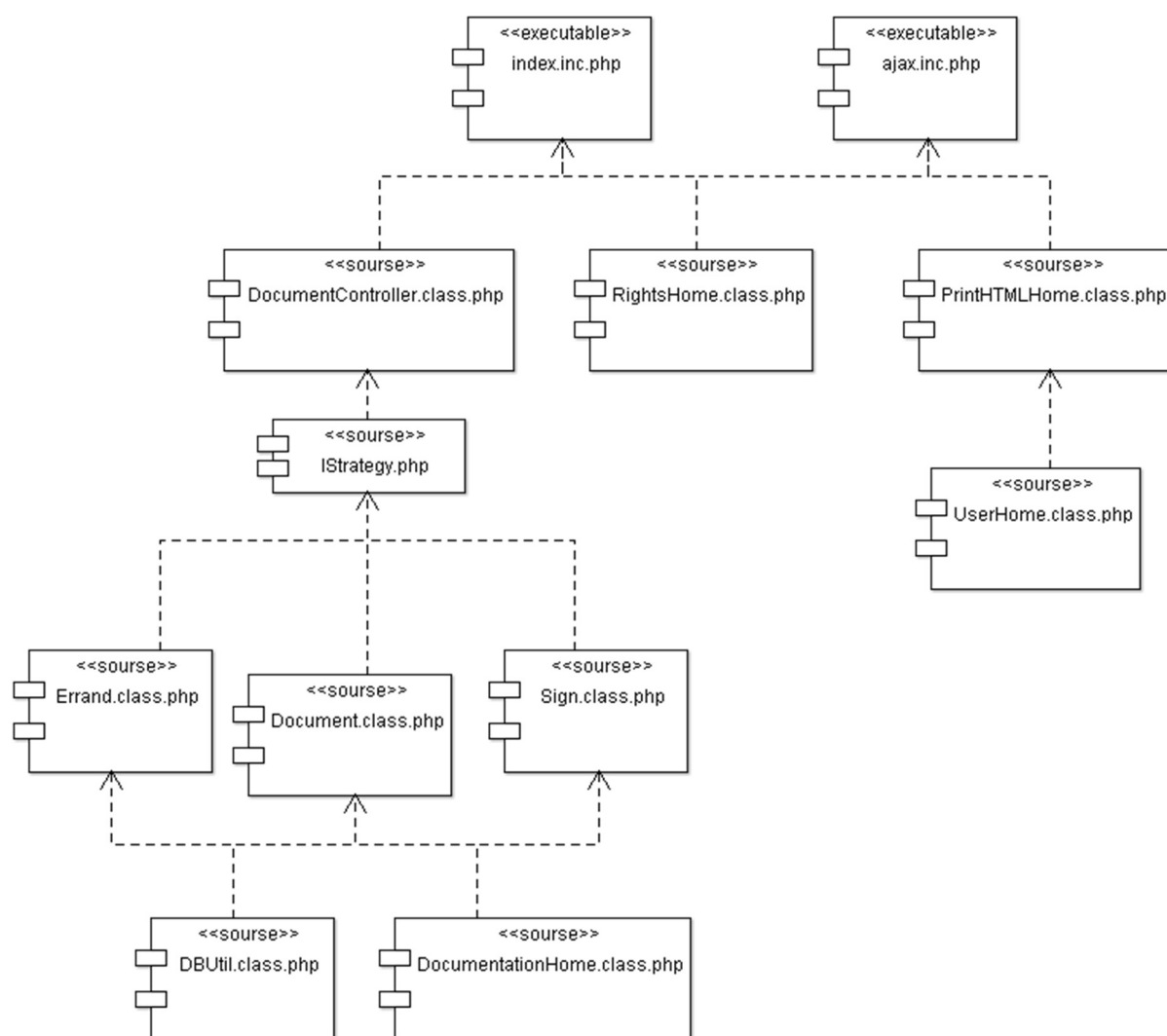


Рисунок 30 – Диаграмма компонентов web-приложения

Исходный код, разработанного web-приложения представлен в приложении Г.

Диаграмма развертывания, которая показывает распределение

компонентов системы по узлам и маршруты передачи информации между ними, представлена на рисунке 31.

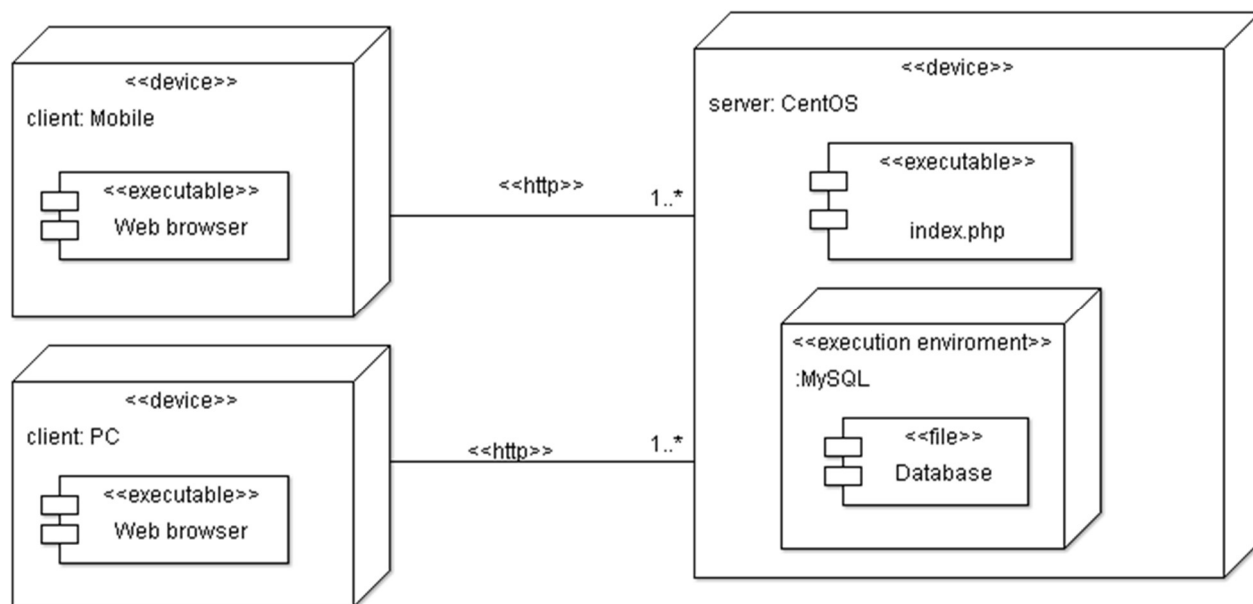


Рисунок 31 – Диаграмма развертывания разработанной системы

Согласно рисунку 31 клиентское приложение разворачивается на рабочих станциях пользователей (смартфон, планшетный компьютер или персональный компьютер), а база данных и исполняемый файл на отдельном сервере.

4 Тестирование web-приложения

Разработка программного обеспечения всегда включает в себя этап тестирования, что помогает обеспечивать контроль качества разрабатываемого программного продукта. Оно помогает выявить ошибки на раннем этапе и определить соответствия между реальным и ожидаемым поведением разработанного продукта [28 – 30].

Тестирование разработанного web-приложения проводилось в следующих браузерах: Google Chrome, Firefox и Opera.

4.1 Функциональное тестирование

Для проверки соответствия web-приложения требованиям технического задания было проведено функциональное тестирование на основе функций, которые были выделены на этапе анализа предметной области. Описание функциональных тестов представлено в таблице 23.

Таблица 23 – Функциональные тесты web-приложения

№ п/п	Назначение теста	Описание теста	Результат теста
1	Тестирование формы создания поручения.	Пользователь создает поручение на одного человека. Браузер – Firefox.	Тест выполнен. Рисунки Д.1 – Д.2.
2	Тестирование формы создания документа.	Пользователь создает документ с типом «Протокол». Браузер – Opera.	Тест выполнен. Рисунки Д.3 – Д.4.
3	Тестирование формы создания подписания.	На основе документа пользователь создает подписание Браузер – Google Chrome.	Тест выполнен. Рисунки Д.5 – Д.6.
4	Тестирование формы создания замещения.	Пользователь создает временное замещение (с 07.06.2020 по 30.06.2020). Браузер – Firefox.	Тест выполнен. Рисунки Д.7 – Д.8.
5	Тестирование формы создания замещения.	Данные введены с ошибкой (дата начала действия меньше текущей даты). Браузер – Opera.	Тест выполнен. Рисунок Д.9.
6	Создание запроса на завершение (от исполнителя поручения).	Пользователь выбирает поручение, у которого он назначен исполнителем, и нажимает на кнопку управления «Завершить». Создается запрос на завершение. Браузер – Google Chrome.	Тест выполнен. Рисунки Д.10.
7	Подтверждение запроса на завершение (от инициатора поручения).	Пользователь выбирает запрос на завершение, направленный ему от исполнителя, и нажимает на кнопку управления «Подтвердить». Выполнение поручения завершается. Браузер – Firefox.	Тест выполнен. Рисунки Д.11.
8	Отклонение запроса на завершение (от инициатора поручения).	Пользователь выбирает запрос на завершение, направленный ему от исполнителя, и нажимает на кнопку управления «Отклонить». Выполнение поручения не завершено. Браузер – Opera.	Тест выполнен. Рисунки Д.12.

Продолжение таблицы 23

№ п/п	Назначение теста	Описание теста	Результат теста
9	Создание запроса на продление (от исполнителя поручения).	Пользователь выбирает поручение, у которого он назначен исполнителем, и нажимает на кнопку управления «Продлить» и задает новую дату. Создается запрос на продление. Браузер – Google Chrome.	Тест выполнен. Рисунки Д.13.
10	Подтверждение запроса продления (от инициатора поручения).	Пользователь выбирает запрос на продление, направленный ему от исполнителя, и нажимает на кнопку управления «Подтвердить». Дата окончания выполнения поручения изменена. Браузер – Firefox.	Тест выполнен. Рисунки Д.14.
11	Отклонение запроса продления (от инициатора поручения).	Пользователь выбирает запрос на продление, направленный ему от исполнителя, и нажимает на кнопку управления «Отклонить». Дата не изменена. Браузер – Google Chrome.	Тест выполнен. Рисунки Д.15.
12	Подтверждение подписания документа (от исполнителя поручения).	Пользователь, который является ответственным исполнителем, выбирает запрос на подписание, направленный ему, и нажимает на кнопку управления «Подтвердить». Документ подписан. Браузер – Opera.	Тест выполнен. Рисунки Д.16.
13	Отклонение подписания документа (от исполнителя поручения).	Пользователь, который является ответственным исполнителем, выбирает запрос на подписание, направленный ему, и нажимает на кнопку управления «Отклонить». Документ не подписан. Браузер – Google Chrome.	Тест выполнен. Рисунки Д.17.

Результаты функционального тестирования показывают, что разработанное web-приложение корректно выполняет весь функционал в соответствии с техническим заданием.

4.2 Структурное тестирование

Основная задача структурного тестирования заключается в детальном изучении логики программы и разработке тестов, которые позволяют проверить максимальное число операторов условий и логических ветвлений [28].

Листинг тестируемого метода «Создать поручение» представлен на листинге 2.

Листинг 2 – Метод «Создать поручение»

```
function saveErrand($userId, $params, $files){
    $textId = $this->docSave->saveText($params);
    //сохранение нового поручения и перепоручения
    if ($params['from'] != $userId)
        if (!isset($params['ernd']) && $doc['id_user'] != $userId)
            $params['id_operator'] = $userId;
    $params['date_create'] = $params['date_first'] = time();
    $params['date_end'] = new DateTime($params['date_end']);
}
```

```

$params['date_end'] = $params['date_end']->getTimestamp();
$params['id_text'] = $textId;
if(isset($params['supp']))
    $params['id_resp']=$this->docHome->getUserFromExecution
        ($params['id_user']);
else
    $params['id_resp']=$this->docHome->getUserFromExecution
        ($params['id_resp']);
if (isset($params['new'])) {
    if($params['id_prot']!=null){
        $params['id_parent'] = $params['id_prot'];
        $params['code']=$this->docHome->getCode($params['id_prot'])
            ['code'];
    }else
        $params['code'] = $this->docSave->saveMainDoc();
    $params['id_owner'] = $params['from'];
    $users = $params['id_user'];
    $wathers = $params['id_watcher'];
    unset($params['id_user']);
    unset($params['id_watcher']);
} else {
    $params['id_type'] = $doc['id_type'];
    $params['id_owner'] = $userId;
    $params['is_tracked'] = $params['is_tracked1'];
    $params['id_parent'] = $doc['id'];
    $params['code'] = $doc['code'];
    $users = $params['id_user1'];
    $wathers = $params['id_watcher1'];
    $params['type']=4;
}
if (count($users) > 1) {
    $params['id_parent'] = $this->docSave->saveDocFlow($params);
    $params['type']=4;
}
if (is_array($users)) {
    foreach ($users as $key => $user) {
        $params['id_user'] = $this->docHome->getUserFromExecution
            ($user);
        if($params['id_user']==$params['id_resp'])
            $params['is_responsible']=1;
        else
            unset($params['is_responsible']);
        if (is_array($wathers)) {
            foreach ($wathers as $key2 => $wather) {
                $params['id_watcher']=$this->docHome->
                    getUserFromExecution($wather);
                $this->docSave->saveDocFlow($params);
            }
        }else
            $this->docSave->saveDocFlow($params);
    }
}
}
}

```

Далее рассмотрим подробно тесты, которые были составлены для этого метода.

4.2.1 Покрывтие операторов

Для наглядного представления функционирования метода «Создать поручение» была разработана блок-схема, которая представлена в приложении Е на рисунках Е.1 – Е.2.

Тесты для анализа покрытия операторов, составленные на основе блок-схемы, представлены в таблице 24.

Таблица 24 – Тесты для метода «Создать поручение»

№ п/п	Входные данные	Операторы	Условия	Ожидаемый результат	Полученный результат
1	Пользователь создает поручение как заместитель, на нескольких человек, без наблюдателей. Браузер – Орега.	2, 3, 5, 7, 9, 10, 11, 13, 14, 16, 20	1 (истина), 4 (ложь), 8 (истина), 12 (ложь), 12 (истина), 15 (ложь)	Создано поручение от заместителя на нескольких людей без наблюдателей	Создано поручение от заместителя на нескольких людей без наблюдателей. Рисунок Д.18.
2	Пользователь создает поручение на одного человека от своего имени, на основании документа, без наблюдателей. Браузер – Firefox.	3, 6, 7, 10, 11, 12, 14, 16, 20	1 (ложь), 4 (истина) 8 (ложь) 12 (истина) 15 (ложь)	От имени пользователя создано поручение на одного человека без наблюдателей на основании документа	От имени пользователя создано поручение на одного человека без наблюдателей на основании документа. Рисунок Д.19.
3	Пользователь создает поручение от своего имени, на нескольких людей, на основании документа, с одним наблюдателем. Браузер – Google Chrome.	3, 6, 7, 9, 10, 12, 13, 17, 18, 19, 20	1 (ложь), 4 (истина) 8 (истина) 12 (истина) 12 (ложь) 15 (истина)	Создано поручение от имени пользователя, на нескольких людей, на основании документа, с одним наблюдателем	Создано поручение от имени пользователя, на нескольких людей, на основании документа, с одним наблюдателем. Рисунок Д.20.
4	Пользователь создает поручение как заместитель, на одного человека, с несколькими наблюдателями. Браузер – Google Chrome.	2, 3, 5, 7, 10, 11, 14, 16	1 (истина) 4 (ложь) 8 (ложь) 12 (истина) 14 (истина)	Создано поручение от заместителя, на одного человека, с несколькими наблюдателями	Создано поручение от заместителя, на одного человека, с несколькими наблюдателями. Рисунок Д.21.
5	Пользователь создает поручение как заместитель, на нескольких людей, с несколькими наблюдателями, на основании документа. Браузер – Google Chrome.	2, 3, 6, 7, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20	1 (истина) 4 (истина) 8 (истина) 12 (истина), 12 (ложь) 15 (истина)	Создано поручение от заместителя, на нескольких людей, с несколькими наблюдателями на основании документа	Создано поручение от заместителя, на нескольких людей, с несколькими наблюдателями на основании документа. Рисунок Д.22.

Таким образом, в ходе тестирования, были покрыты все операторы метода «Создать поручение». Полученные результаты показали корректную работу рассматриваемого метода.

4.2.2 Покрытие условий

Покрытие условий было выполнено в предыдущем разделе в ходе проведения тестов по покрытию операторов.

4.2.3 Покрытие маршрутов

Для рассмотрения этого метода структурного тестирования был разработан граф потока данных, который представлен в приложении Ж.

Тесты на покрытие маршрутов представлены в таблице 25.

Таблица 25 – Тесты на покрытие маршрутов

№ п/п	Маршрут	Входные данные	Ожидаемый результат	Полученный результат
1	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 20	Пользователь создает поручение как заместитель, не на основании документа, на нескольких пользователей, без наблюдателей. Браузер – Google Chrome.	Создано поручение от заместителя, на нескольких пользователей, без наблюдателей	Создано поручение от заместителя, на нескольких пользователей, без наблюдателей. Рисунок Д.23 – Д.24.
2	1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20	Пользователь создает поручение от своего имени, не на основании документа, на нескольких пользователей, без наблюдателей. Браузер – Opera.	Создано поручение от имени пользователя, на нескольких пользователей, без наблюдателей	Создано поручение от имени пользователя, на нескольких пользователей, без наблюдателей. Рисунок Д.25 – Д.26.
3	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20	Пользователь создает поручение как заместитель, на основании документа, на нескольких пользователей, без наблюдателей. Браузер – Firefox.	Создано поручение от заместителя, на нескольких пользователей, без наблюдателей, на основании документа	Создано поручение от заместителя, на нескольких пользователей, без наблюдателей, на основании документа. Рисунок Д.27 – Д.28.
4	1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20	Пользователь создает поручение от своего имени, не на основании документа, на нескольких пользователей, с наблюдателями. Браузер – Google Chrome.	Создано поручение от имени пользователя, на нескольких пользователей, с наблюдателями	Создано поручение от имени пользователя, на нескольких пользователей, с наблюдателями. Рисунок Д.29 – Д.30.

Из представленной таблицы видно, что все маршруты метода протестированы успешно.

4.3 Метрики кода

Вычисление метрик кода позволяет оценить качество разработанного программного кода в виде численных значений его свойств. В ходе тестирования было проведено вычисление метрик кода для метода «Создать поручение».

4.3.1 Цикломатическая метрика Мак-Кейба

Для вычисления этой метрики необходимо проанализировать граф метода, представленный в приложении Ж. Метрика Мак-Кейба определяет цикломатическую сложность программы.

Для вычисления этой величины необходимо подсчитать количество дуг (e) и вершин (n) в графе:

$$e = 25,$$

$$n = 21.$$

Затем определяем число компонент связи (p) – это количество дуг, которое необходимо добавить, чтобы из каждой вершины графа можно было попасть в каждую вершину. В данном случае $p=1$.

Подставляем полученные значения и получаем:

$$V(G) = e - n + 2p = 6.$$

Исходя из полученного результата можно сделать вывод, что рассматриваемая функция достаточно понятна и обладает средней сложностью.

4.3.2 Метрика Холстеда

Для определения этой метрики необходимо подсчитать четыре характеристики: количество уникальных операторов ($n1$), число вхождений операторов ($N1$), количество уникальных операндов ($n2$), число вхождений операндов ($N2$). Уникальные операторы и операнды представлены в таблице 26.

Таблица 26 – Уникальные операторы и операнды

Операторы	Число вхождений	Операнды	Число вхождений
1	2	3	4
=	20	userId	5

Продолжение таблицы 26

Операторы	Число вхождений	Операнды	Число вхождений
.	2	params	5
->	24	users	3
>	1	watchers	2
&&	1	user	2
!=	1	watcher	2
{	8	docSave	6
}	8	saveDocFlow	4
if	8	saveText	1
else	6	saveMainDoc	1
foreach	2	getUserFromExecution	5
!	1	isset	3
		docHome	6
n1=12	N1=81	n2=13	N2=45

Рассчитаем длину программы (N) и ее словарь (n):

$$N = N1 + N2 = 126,$$

$$n = n1 + n2 = 25.$$

Подставим полученные значения в приведенную ниже формулу для определения метрики Холстеда:

$$V = N \times \log(2)n = 126 * 1,4 = 176,4.$$

Полученное значение показывает, что рассматриваемая функция является компактной.

4.3.3 Метрика Чепина

Эта метрика позволяет оценить информационную прочность с помощью анализа характера использования переменных. Для определения метрики необходимо определить вводимые переменные (P), переменные создаваемые внутри (M), управляющие переменные (C) и неиспользуемые переменные (T).

Для рассматриваемого метода характеристики равны:

$$P = 3,$$

$$M = 3,$$

$$C = 2,$$

$$T = 0.$$

Подставим полученные значения в формулу, представленную ниже:

$$Q = P + 2M + 3C + 0.5T = 15.$$

Полученное значение показывает, высокую прочность рассматриваемого метода.

4.4 Автоматизированное тестирование

Ручное тестирование занимает много времени и может оказаться не полным. Поэтому в настоящее время полноценное тестирование программного продукта не может проводится без проведения автоматизированных тестов.

4.4.1 Модульное тестирование

Для проверки правильности работы метода «Создать поручение» был написан unit-тест, который представлен в листинге 3.

Листинг 3 – Unit-тест для проверки метода «Создать поручение»

```
<?php
require_once($modules_root."edms/class/Errand.class.php");
use PHPUnit\Framework\TestCase;
class ErrandTest extends PHPUnit_Framework_TestCase
{
    private $errand;
    private $db_edms;
    protected function setUp()
    {
        $this->errand = new Errand();
        $this->db_edms = $GLOBALS ["db10"];
    }
    protected function tearDown()
    {
        $this->errand = NULL;
    }
    public function testSaveNew(){
        $userId=517649557;
        $testParams=array('new'=>1,
            'id_type'=>1,
            'id_prot'=>129,
            'from'=>517649557,
            'id_user'=>array(
                0=>2883,
                1=>189155),
            'id_resp'=>2883,
            'name'=>'Тестирование поручения на основе
                документа',
            'text'=>'<p>Поручение на двух исполнителей</p>',
            'date_end'=>'2020-04-19');
        $this->errand->save($userId,$testParams,null);
        $sql="SELECT COUNT(*) cn FROM docs_flow WHERE id_parent=" .
            $testParams['id_prot'] . " OR id_parent=(SELECT id FROM docs_flow
            WHERE id_parent=" . $testParams['id_prot'] . ")";
```

```

$this->assertEquals(3, $this->db_edms->getOneData($sql, ['cn']))
    ['cn']);
$sql="SELECT id_user FROM docs_flow WHERE id_parent=(SELECT id FROM
    docs_flow WHERE id_parent=".$testParams['id_prot'].")";
$users=$this->db_edms->getListData($sql, ['id_user']);
$this->assertEquals(31507538, $users[0]);
$this->assertEquals(250178987, $users[1]);
$sql="SELECT text.name FROM docs_flow LEFT JOIN text on text.id=
    docs_flow.Id_text WHERE id_parent=129";
$this->assertEquals("Тестирование тестирование поручения на основе
    документа", $this->db_edms->getOneData($sql, ['name']) ['name']);
    }
}

```

Результат проведения теста представлен на рисунке 32.

PHPUnit 3.7.32 by Sebastian Bergmann.

.

Time: 31ms, Memory: 2.25Mb

OK (1 test, 3 assertion)

Рисунок 32 – Результат проведения unit-теста

В результате проведения unit-тестирования метода «Создать поручение» ошибки не были выявлены. Таким образом можно сделать вывод, что он работает корректно.

4.4.2 Тестирование с использованием Katalon Recorder

Для проверки клиентской части web-приложения был использован плагин Katalon Recorder для web-браузера Google Chrome.

Для тестирования функции «Создать поручение» был создан сценарий действий, который представлен в листинге 4.

Листинг 4 – Программный код, сгенерированный плагином Katalon Recorder

```

WebUI.openBrowser('https://www.google.com/')
def driver = DriverFactory.getWebDriver()
String baseUrl = "https://www.google.com/"
selenium = new WebDriverBackedSelenium(driver, baseUrl)
selenium.open("https://lk.pnzgu.ru/edms")
selenium.click("link=Документы")
selenium.click("//button[@onclick='window.open(\"/edms/517649557/create/errand/77\")';]")
selenium.selectWindow("win_ser_1")
selenium.click("//input[@class='chosen-search-input default']")
selenium.click("//div[@id='id_user_chosen']/div/ul/li[2]")
selenium.click("id=name")

```

```
selenium.type("id=name", "Тест")
selenium.click("id=date_end")
selenium.click("id=date_end")
selenium.type("id=date_end", "2020-04-19")
selenium.click("id=doc")
```

Результат проведения теста представлен на рисунке 33.

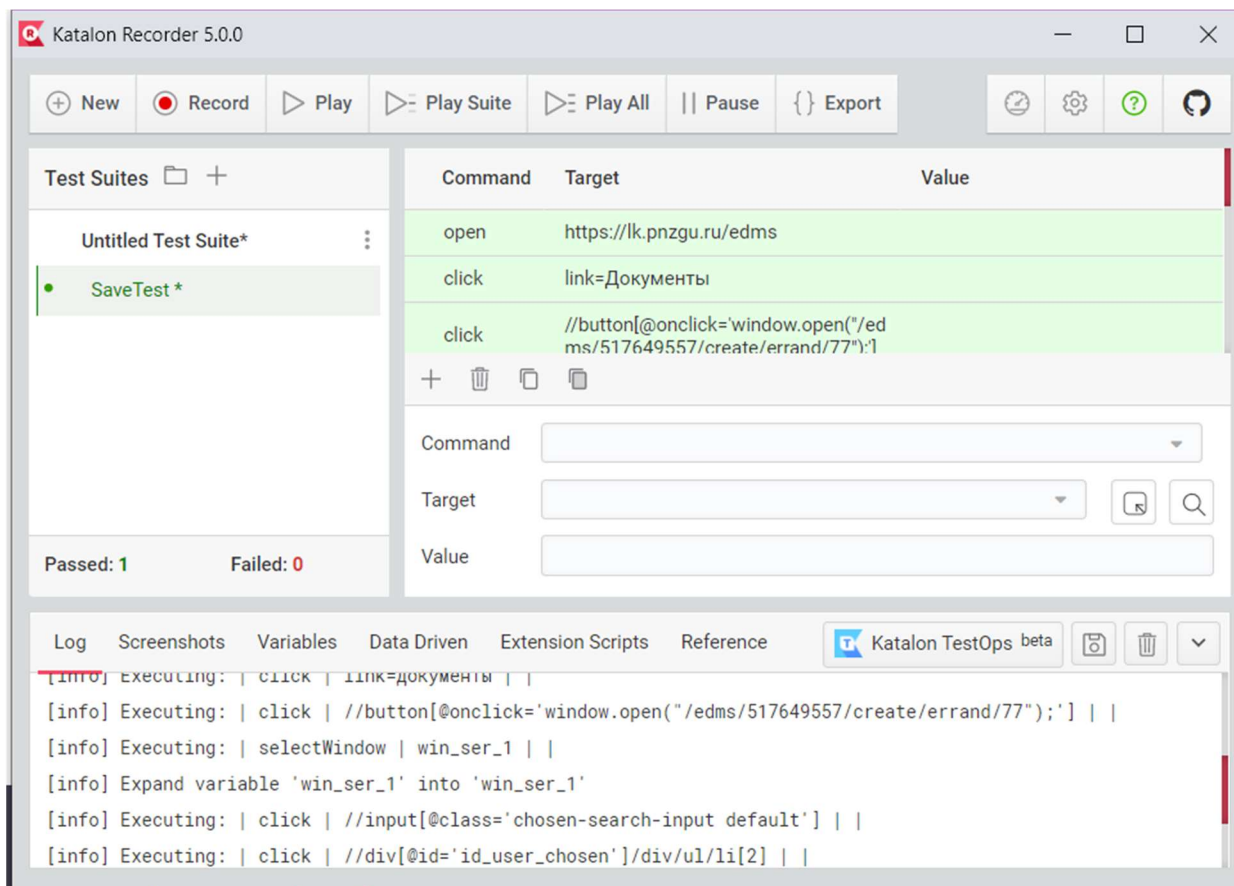


Рисунок 33 – Результат тестирования с использованием плагина Katalon Recorder

Результат тестирования с использованием плагина Katalon Recorder показал, корректную работу клиентской части разработанного web-приложения.

По результатам тестирования web-приложения для контроля исполнения поручений в системе электронного документооборота вуза можно сделать вывод, что оно функционирует корректно. Также, в ходе тестирования были рассчитаны метрики кода для отдельного метода «Создать поручение», которые показали, что она является компактной, прочной со средней сложностью. Результаты тестов представлены в приложении Д.

Заключение

В рамках выполнения выпускной квалификационной работы было разработано web-приложение для контроля исполнения поручений в системе электронного документооборота вуза.

Был проведен анализ предметной области и требований, в ходе которого было проанализировано программное обеспечение аналогичного назначения, рассмотрены их преимущества и недостатки, также разработаны функциональные и нефункциональные требования.

На этапе проектирования было проведено планирование разработки web-приложения и оценка бюджета. Длительность реализации проекта составляет 115 дней с бюджетом в 194 929,48 руб. Полученные, в ходе планирования, результаты показывают экономическую выгоду и целесообразность разработки. Также было проведено проектирование базы данных, графического интерфейса и архитектуры программного обеспечения.

Web-приложение реализовано с использованием языков PHP, SQL, JavaScript с использованием технологии Ajax, язык разметки документов HTML и язык описания внешнего вида документов CSS в среде разработки PhpStorm JetBrains 2019.2.4.

Было проведено функциональное, структурное и автоматизированное тестирование разработанного web-приложения. На основе результатов тестирования можно сделать вывод, что разработанное web-приложение работает корректно. Также были рассчитаны метрики кода, которые показывают, что программный код является прочным, компактным со средней сложностью.

В приложении И представлен акт о внедрении web-приложения для контроля исполнения поручений в системе электронного документооборота вуза в ФГБОУ ВО «Пензенский государственный университет».

Таким образом, в результате выполнения выпускной квалификационной работы разработано web-приложение, которое отвечает всем требованиям технического задания.

Список использованных источников

1. Программа «Цифровая экономика Российской Федерации». Утверждена распоряжением Правительства Российской Федерации от 28 июля 2017 г. № 1632-р. [Электронный ресурс]. – URL: <http://static.government.ru/media/files/9gFM4FHj4PsB79I5v7yLVuPgu4bvR7M0.pdf> (дата обращения: 02.05.2020).
2. Распоряжение Президента Российской Федерации от 9 мая 2017 г. № 203 «Стратегия развития информационного общества в Российской Федерации на 2017 – 2030 годы». [Электронный ресурс]. – URL: http://zmedu.hostedu.ru/files/ykaz_7668.pdf (дата обращения: 02.05.2020).
3. Федеральный закон от 29.12.2012 N 273-ФЗ (ред. от 24.04.2020) «Об образовании в Российской Федерации». [Электронный ресурс]. – URL: http://www.consultant.ru/document/cons_doc_LAW_140174/9ab9b85e5291f25d6986b5301ab79c23f0055ca4/ (дата обращения: 02.05.2020).
4. Бобылева, М. П. Управленческий документооборот: от бумажного к электронному. Вопросы теории и практики / М. П. Бобылева. – М.: Термика, 2016. – 360 с.
5. Исполнительская дисциплина как ключевой показатель эффективности. // Справочник секретаря [Электронный ресурс]. – URL: <https://www.sekretariat.ru/article/211183-ispolnitelskaya-distsiplina-pokazatel-effektivnosti-17-m12> (дата обращения: 01.05.2020).
6. Крюков, В. В. Развитие информационной инфраструктуры вуза для решения задач управления / В. В. Крюков, К. И. Шахгельдян // Университетское управление: практика и анализ. – 2004. – №4. – С. 66-77.
7. СЭД. Рынок России. // TAdviser. Государство. Бизнес. IT [Электронный ресурс]. – URL: <http://www.tadviser.ru/a/53815> (дата обращения: 26.04.2020).
8. Просто об электронном документообороте. // Журнал об электронном контенте, документах и бизнес-процессах [Электронный ресурс]. – URL: <https://ecm-journal.ru/mustknow> (дата обращения: 25.04.2020).

9. Жарков, А. А. Система электронного документооборота / А. А. Жарков // Наука, техника и образование. – 2014 – №3. – С. 65-71.
10. Directum. Управление бизнес-процессами // Официальный сайт Directum [Электронный ресурс]. – URL: <https://www.directum.ru/business-process-administration> (дата обращения: 01.05.2020).
11. Система электронного документооборота «Тезис» // Официальный сайт СЭД Тезис [Электронный ресурс]. – URL: <https://www.tezis-doc.ru/blog/plyusy-i-minusy-sed-v-raznyh-redaktsiyah/> (дата обращения: 30.04.2020).
12. Система электронного документооборота «Мотив». // Официальный сайт СЭД Мотив [Электронный ресурс]. – URL: <https://www.motiw.ru/solutions/> (дата обращения: 30.04.2020).
13. Коберн, А. Современные методы описания функциональных требований к системам / А. Коберн. – М.: Лори, 2012. – 264 с.
14. Маглинец, Ю. А. Анализ требований к автоматизированным информационным системам / Ю. А. Маглинец. – М.: Бином, 2014. – 200 с.
15. Статистика использования браузеров и разрешений экрана // Сервис сбора и обработки статистики web-трафика [Электронный ресурс]. – URL: <https://gs.statcounter.com/> (дата обращения: 30.05.2020).
16. Крачер, Ф. Введение в Rational Unified Process / Ф. Крачер. – М.: Вильямс, 2002. – 240 с.
17. Арлоу, Д. UML 2 и Унифицированный процесс / Д. Арлоу, И. Нейштадт. – СПб.: Символ-Плюс, 2007. – 617 с.
18. Талманн, Л. Обеспечение высокой доступности систем на основе MySQL / Л. Талманн, Ч. Белл. – СПб.:БХВ, 2012. – 624 с.
19. Николаева, Л. Э. Обзор средств создания сайтов / Л. Э. Николаева, А. Н. Ванютин // Состояние и перспективы развития ИТ-образования. – 2018. – С. 98 – 105.

20. Bootstrap: преимущества и недостатки // Портал Depix [Электронный ресурс]. URL: https://depix.ru/articles/bootstrap_preimuschestva_i_nedostatki (дата обращения: 25.04.2020).
21. Мазур И. И. Управление проектами. / В. Д. Шапиро, Н. Г. Ольдерогге – М.: Омега-Л, 2007. – 664 с.
22. Правдина, Н.В. Сетевое планирование и оценка проектных затрат / Н. В. Правдина. – Челябинск: Издательский центр ЮУрГУ, 2015. – 38 с.
23. Пирогов, В. Ю. Информационные системы и базы данных: организация и проектирование: учебное пособие / В. Ю. Пирогов. – СПб. – БХВ, 2009. – 529 с.
24. Филимоненкова, Т.Н. UX/UI дизайн и его роль в разработке пользовательского интерфейса сайта / Т.Н. Филимоненкова, А.С. Дунаевский. – 2017. – С. 158-160.
25. Статья на Хабр о принципах построения интерфейсов // Коллективный блог [Электронный ресурс]. URL: https://habr.com/ru/company/SECL_GROUP/blog/182208/ (дата обращения: 25.05.2020).
26. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб.: Питер, – 2001. – 368 с.
27. Маклин, Х. Г. Адаптивный код: гибкое кодирование с помощью паттернов проектирования и принципов SOLID / Х. Г. Маклин. – М.: Вильямс, – 2017. – 448 с.
28. Гленфорд, М. Искусство тестирования программ / М. Гленфорд, Т. Баджетт, К. Сандлер. – М.: Вильямс, – 2012. – 272 с.
29. Статья о тестировании ПО // Коллективный блог [Электронный ресурс]. URL: <https://habr.com/ru/post/279535/> (дата обращения: 23.05.2020).
30. Аббаров, Р. Д. Тестирование графического интерфейса (GUI testing) / Р. Д. Аббаров, В. О. Пак. — Текст : непосредственный / Техника. Технологии. Инженерия. — 2017. — № 3.1 (5.1). — С. 3-4.

Приложение А
ГЛОССАРИЙ
(справочное)

Web-приложение – клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера.

Неквалифицированная электронная подпись (НЭП) – это зашифрованная комбинация символов, которая подтверждает личность пользователя и позволяет обнаружить внесение изменений в документ после его подписания. Подходит для внутреннего документооборота

Система электронного документооборота (СЭД) – это система (компьютерная программа, программное обеспечение и т.п.), позволяющая организовать и автоматизировать работу с электронными документами (т.е. электронный документооборот) на протяжении всего их жизненного цикла.

Корпоративная информационная система (КИС) – это масштабируемая система, предназначенная для комплексной автоматизации всех видов хозяйственной деятельности компаний, а также корпораций, требующих единого управления.

AJAX (Asynchronous Javascript and XML) – подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в асинхронном обмене данными браузера с веб-сервером.

RUP (Rational Unified Process) – методология разработки программного обеспечения, в основе которой лежит использование итеративной модели разработки.

UML (Unified Modeling Language) – язык графического описания, с помощью которого осуществляется объектное моделирование в области разработки программного обеспечения, проектирования систем и отображения организационных структур.

Модульное тестирование (Unit testing) – тестирование каждой атомарной функциональности приложения отдельно, в искусственно созданной среде.

Приложение Б
СПЕЦИФИКАЦИЯ ПРЕЦЕДЕНТОВ
(рекомендуемое)

Прецедент: Просмотреть список запросов
ID: 5
Краткое описание: Инициатор, Исполнитель или Наблюдатель просматривает список поручений, замещений или документов
Главные актёры: Инициатор, Исполнитель, Наблюдатель
Второстепенные актёры: Сервер
Предусловия: 1. Пользователь вошел в систему.
Основной поток: 1. Если пользователь открывает страницу со списком поручений 1.1. Extend «Открыть поручения» 2. Если пользователь открывает страницу со списком документов 2.1. Extend «Открыть документы» 3. Если пользователь открывает страницу со списком замещений 3.1. Extend «Открыть замещения» 4. Сервер возвращает данные
Постусловия: 1. Пользователь получает список доступных данных
Альтернативные потоки: Нет.

Рисунок Б.1 – Спецификация прецедента «Просмотреть список запросов»

Прецедент: Назначить заместителя
ID: 2
Краткое описание: Исполнитель назначает своего заместителя
Главные актёры: Инициатор
Второстепенные актёры: Сервер
Предусловия: 1. Инициатор вошел в систему.
Основной поток: 1. Инициатор открывает страницу создания замещения 2. Инициатор вводит данные 2.1. Инициатор выбирает пользователя, который будет его замещать 2.2. Если инициатор выбирает временное замещение 2.2.1. Инициатор вводит дату начала и дату окончания действия замещения 2.3. Инициатор выбирает тип замещения 3. Инициатор нажимает кнопку «Сохранить» 4. Если ошибок нет 4.1. Сервер выполняет сохранение замещения 5. Иначе переход на действие 2
Постусловия: 1. Замещение создано
Альтернативные потоки: Нет.

Рисунок Б.2 – Спецификация прецедента «Назначить заместителя»

Прецедент: Создать документ
ID: 3
Краткое описание: Инициатор создает документ
Главные актёры: Инициатор
Второстепенные актёры: Сервер
Предусловия: 1. Инициатор вошел в систему.
Основной поток: 1. Инициатор открывает страницу создания поручения 2. Инициатор вводит данные 2.1. Если инициатор является заместителем другого пользователя и поручение должно быть создано от его имени 2.1.1. Документ создается от имени того, кого замещает инициатор 2.2. Инициатор выбирает тип документа 2.3. Инициатор вводит название и описание документа 2.4. Если инициатор прикрепляет файлы 2.4.1. Инициатор выбирает один или несколько файлов 3. Инициатор нажимает кнопку «Сохранить» 4. Если ошибок нет 4.1. Сервер выполняет сохранение документа 5. Иначе переход на действие 2
Постусловия: 1. Документ создан
Альтернативные потоки: Нет.

Рисунок Б.3 – Спецификация прецедента «Создать документ»

Прецедент: Продлить выполнение
ID: 6
Краткое описание: Исполнитель создает запрос на продление выполнения поручения
Главные актёры: Исполнитель
Второстепенные актёры: Сервер
Предусловия: 1. Исполнитель вошел в систему. 2. Исполнителю было отправлено на исполнение поручение
Основной поток: 1. Исполнитель перешел на страницу поручения 2. Исполнитель нажал на кнопку «Продлить» 3. Исполнитель вводит данные 3.1. Исполнитель выбирает новую дату окончания 3.2. Исполнитель вводит описание 4. Исполнитель нажимает кнопку «Сохранить» 5. Если ошибок нет 5.1. Сервер выполняет сохранение запроса 6. Иначе переход на действие 2 7. Include «Ответить на запрос»
Постусловия: 1. Создается запрос на продление и ответ на него
Альтернативные потоки: Нет.

Рисунок Б.4 – Спецификация прецедента «Продлить выполнение»

Прецедент: Подписать документ
ID: 8
Краткое описание: Исполнитель отвечает на запрос подписания документа
Главные актёры: Исполнитель
Второстепенные актёры: Сервер
Предусловия: 1. Исполнитель вошел в систему. 2. Исполнителю был отправлен запрос на подписание документа
Основной поток: 1. Исполнитель перешел на страницу запроса подписания 2. Если исполнитель нажал на кнопку «Подписать» 2.1. Сервер выполняет сохранение запроса 2.2. Сервер помечает документ как подписанный 3. Если исполнитель нажал на кнопку «Отклонить» 3.1. Сервер выполняет сохранение запроса
Постусловия: 1. Запрос на подписание закрывается
Альтернативные потоки: Нет.

Рисунок Б.5 – Спецификация прецедента «Подписать документ»

Прецедент: Ответить на запрос
ID: 9
Краткое описание: Инициатор отвечает на запрос продления или закрытия, который был направлен ему от Исполнителя
Главные актёры: Инициатор
Второстепенные актёры: Сервер
Предусловия: 1. Инициатор вошел в систему. 2. Инициатор создал поручение 3. Инициатор получил запрос от Исполнителя
Основной поток: 1. Инициатор перешел на страницу запроса 2. Если Инициатор нажимает кнопку «Подтвердить» 2.1. Если Инициатор получил запрос на продление 2.1.1. Если Инициатор не согласен с новой датой 2.1.1.1. Инициатор устанавливает новую дату 2.2. Инициатор подтверждает выполнение запроса 3. Если исполнитель нажал на кнопку «Отклонить» 3.1. Инициатор не подтверждает выполнение запроса 4. Сервер выполняет сохранение ответа на запрос
Постусловия: 1. Запрос, направленный Инициатору, закрыт
Альтернативные потоки: Нет.

Рисунок Б.6 – Спецификация прецедента «Ответить на запрос»

Приложение В
ИСХОДНЫЙ КОД СОЗДАНИЯ БАЗЫ ДАННЫХ
(рекомендуемое)

Листинг В.1 – Исходный код создания базы данных

```

CREATE TABLE IF NOT EXISTS `assignment` (
  `id` int(11) NOT NULL,
  `id_boss` int(11) NOT NULL,
  `id_alternate` int(11) NOT NULL,
  `date_start` bigint(20) DEFAULT NULL,
  `date_end` bigint(11) DEFAULT NULL,
  `mode` tinyint(1) NOT NULL DEFAULT '0',
  `type` tinyint(1) NOT NULL DEFAULT '0',
  `is_deleted` tinyint(1) NOT NULL DEFAULT '0'
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `docs` (
  `id` int(11) NOT NULL,
  `is_deleted` tinyint(1) NOT NULL DEFAULT '0'
) ENGINE=InnoDB AUTO_INCREMENT=77 DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `docs_flow` (
  `id` int(15) NOT NULL,
  `id_parent` int(11) NOT NULL DEFAULT '0',
  `id_type` int(15) NOT NULL,
  `number` varchar(100) DEFAULT NULL,
  `id_owner` bigint(20) DEFAULT NULL,
  `id_operator` bigint(20) DEFAULT NULL,
  `id_user` bigint(20) DEFAULT NULL,
  `id_watcher` bigint(20) DEFAULT NULL,
  `id_text` int(11) DEFAULT NULL,
  `code` int(11) NOT NULL,
  `date_create` bigint(20) DEFAULT NULL,
  `date_end` bigint(20) DEFAULT NULL,
  `date_first` bigint(20) NOT NULL,
  `date_end_fact` bigint(20) DEFAULT NULL,
  `date_show` bigint(20) DEFAULT NULL,
  `type` tinyint(1) NOT NULL DEFAULT '0',
  `is_changed1` tinyint(1) NOT NULL DEFAULT '0' COMMENT 'owner',
  `is_changed2` tinyint(1) NOT NULL DEFAULT '0' COMMENT 'user',
  `is_changed3` tinyint(1) NOT NULL DEFAULT '0' COMMENT 'operator',
  `is_changed4` tinyint(1) NOT NULL DEFAULT '0' COMMENT 'watcher',
  `is_tracked` tinyint(1) NOT NULL DEFAULT '1',
  `is_closed` tinyint(1) NOT NULL DEFAULT '0',
  `is_signed` tinyint(1) NOT NULL DEFAULT '0',
  `hash` varchar(40) DEFAULT NULL,
  `is_consistent` tinyint(1) NOT NULL DEFAULT '0',
  `is_responsible` tinyint(1) NOT NULL DEFAULT '0',
  `is_deleted` tinyint(1) NOT NULL DEFAULT '0'
) ENGINE=InnoDB AUTO_INCREMENT=234 DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `docs_types` (
  `id` int(10) NOT NULL,
  `name` varchar(255) CHARACTER SET utf8 NOT NULL,
  `is_show` tinyint(1) NOT NULL DEFAULT '0',
  `is_deleted` tinyint(4) NOT NULL DEFAULT '0'
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `files` (
  `id` int(11) NOT NULL,
  `id_text` int(11) NOT NULL,
  `name` varchar(255) CHARACTER SET utf8 NOT NULL,
  `weight` int(11) NOT NULL,
  `is_deleted` tinyint(1) NOT NULL DEFAULT '0'
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=latin1;

```

```

CREATE TABLE IF NOT EXISTS `text` (
  `id` int(11) NOT NULL,
  `text` text CHARACTER SET utf8,
  `name` varchar(255) CHARACTER SET utf8 NOT NULL,
  `is_deleted` tinyint(1) NOT NULL DEFAULT '0'
) ENGINE=InnoDB AUTO_INCREMENT=312 DEFAULT CHARSET=latin1;

ALTER TABLE `assignment`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `docs`
  ADD PRIMARY KEY (`id`),
  ADD KEY `id` (`id`);

ALTER TABLE `docs_flow`
  ADD PRIMARY KEY (`id`),
  ADD KEY `id` (`id`),
  ADD KEY `code` (`code`),
  ADD KEY `id_text` (`id_text`),
  ADD KEY `id_parent` (`id_parent`);

ALTER TABLE `docs_types`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `files`
  ADD PRIMARY KEY (`id`),
  ADD KEY `id_text` (`id_text`);

ALTER TABLE `text`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `assignment`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=20;

ALTER TABLE `docs`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=77;

ALTER TABLE `docs_flow`
  MODIFY `id` int(15) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=234;

ALTER TABLE `docs_types`
  MODIFY `id` int(10) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=6;

ALTER TABLE `files`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=20;

ALTER TABLE `text`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=312;

ALTER TABLE `docs_flow`
  ADD CONSTRAINT `docs_flow_ibfk_1` FOREIGN KEY (`code`) REFERENCES `docs`
    (`id`),
  ADD CONSTRAINT `docs_flow_ibfk_2` FOREIGN KEY (`id_text`) REFERENCES
    `text` (`id`);

ALTER TABLE `files`
  ADD CONSTRAINT `files_ibfk_1` FOREIGN KEY (`id_text`) REFERENCES `text`
    (`id`);

```

Приложение Г
ИСХОДНЫЙ КОД ПРИЛОЖЕНИЯ
(рекомендуемое)

Листинг Г.1 – ajax.inc.php

```

<?php
require_once($modules_root."edms/class/DocumentationHome.class.php");
require_once($modules_root."edms/class/PrintHTMLHome.class.php");
require_once($modules_root."edms/class/ref/DocumentController.class.php");
require_once($modules_root."edms/class/ref/Errand.class.php");
require_once($modules_root."edms/class/ref/Sign.class.php");
require_once($modules_root."edms/class/ref/Document.class.php");

if(!$docHome) $docHome = new DocumentationHome();
if(!$printHtml) $printHtml = new PrintHTMLHome();
if(!$controller) $controller = new DocumentController(new Errand());
if($loggedIn) {
    if ($request->hasValue('create')) {
        if ($request->hasValue('errand')){//создание нового поручения
            $controller->setStrategy(new Errand());
            $module['text'] .= $controller->showNewDocument($request->
                getValue('errand'),$request->getValue('edms'));
        }elseif ($request->hasValue('document')){//создание документа
            $controller->setStrategy(new Document());
            $module['text'] .= $controller->showNewDocument($request->
                getValue('document'),$request->getValue('edms'));
        }elseif ($request->hasValue('assignment')){//создание замещения
            $module['text'] .= $printHtml->showAssignment($request->
                getValue('edms'));
        }elseif ($request->hasValue('signing')){//создание согласования
            $controller->setStrategy(new Sign());
            $module['text'] .= $controller->showNewDocument($request->
                getValue('signing'),$request->getValue('edms'));
        }
    }elseif ($request->hasValue('errand')) {
        if ($request->getValue('errand') != null) {
            switch ($docHome->getDocType($request->getValue('errand'))){
                case 1:
                    $controller->set(new Errand());
                    break;
                case 5:
                    $controller->set(new Sign());
                    break;
                default:
                    if ($docHome->getShowType($docHome->getDocType(
                        $request->getValue('errand'))==0)
                        $controller->set(new Document());
                    break;
            }
            $module['text'] .= $controller->showDocument($request->
                getValue('errand'),$request->getValue('edms'));
        }
    } elseif ($request->hasValue('assignments'))
        $module['text'] .= $printHtml->getAllAssignments($request->
            getValue('edms'));
    elseif ($request->hasValue('assignment'))
        $module['text'] .= $printHtml->showAssignment($request->
            getValue('edms'),$request->getValue('assignment'));
    elseif ($request->hasValue('documents'))
        $module['text'] .= $printHtml->getAllDocuments($request->
            getValue('edms'));
    elseif ($request->hasValue('document')){
        $controller->setStrategy(new Document());
        $module['text'] .= $request->hasValue('errands')?$printHtml->
            getErrandsOfDocument($request->getValue('edms'),$request->

```

```

        getValue('document')):
        $controller->showDocument($request->getValue('document'),$request-
        >getValue('edms'));
    }
    elseif ($request->getValue('edms')== "")
        $module['text'] .= $printHtml->getAllErrands($GLOBALS
        ["current_user"]['id']);
}

```

Листинг Г.2 – IStrategy.php

```

interface IStrategy
{
    public function tpl();
    public function documentation($items,$userId);
    public function show($docId,$userId);
    public function save($userId, $items, $files);
    public function close($docId,$userId);
    public function buttons($doc,$userId);
    public function showCreate($docId,$userId);
    public function preview($docId,$userId);
}

```

Листинг Г.3 – Sing.class.php

```

<?php
require_once($modules_root."edms/class/DocumentationHome.class.php");
require_once($modules_root."edms/class/DocSaveHome.class.php");
require_once($modules_root."edms/class/ref/IStrategy.php");
require_once($modules_root."edms/class/ValidationHome.class.php");
require_once($modules_root."edms/class/PrintHTMLHome.class.php");
class Sign implements IStrategy
{
    private $check;
    private $printHtml;
    private $templateHome;
    private $modules_root;
    private $docHome;
    private $docSave;
    public function __construct(){
        $this->check=new ValidationHome();
        $this->printHtml=new PrintHTMLHome();
        $this->templateHome=$GLOBALS['templateHome'];
        $this->modules_root=$GLOBALS['modules_root'];
        $this->docHome=new DocumentationHome();
        $this->docSave=new DocSaveHome();
    }
    public function tpl(){
        return "signing.tpl";
    }
    public function documentation($data,$userId){
        if ($data['sign'])
            $result['errors_none']=true;
        elseif(!isset($data['mode'])) {
            $signHome = new SignHome();
            $sign_param = $signHome->checkSignCookie($userId);
            $cName = 'srfhh' . $userId;
            if ($sign_param && $data['pin']) {
                $params = array('pin' => $data['pin'],'id_user' => $userId,
                'hashmd' => $data['COOKIE'][$cName]);
                $opts = array('http'=>array('method'=>"POST",

```

```

        'content' => http_build_query($params)));
    $context = stream_context_create($opts);
    $param = file_get_contents('https://lk.pnzgu.ru/ajax/sign
        /getmd/', false, $context);
    $decode_param = json_decode($param);
    if($decode_param->{'hash'})
        $result['errors_none']=true;
    else
        $result['errors'] = "Введен неверный pin-код.
            Обратитесь к администратору";
    }elseif(!isset($data['COOKIE'][$cName]))
        $result['errors']="Не установлена подпись";
    else
        $result['errors_none']=true;
    $result['content'] = $this->templateHome->parse($this-
        >modules_root . "edms/tpl/modal.tpl", $param);
}
return $result;
}
public function show($docId,$userId){
    $doc=$this->docHome->getOneDocById($docId);
    $temp = array_merge($doc, $this->docHome->getText($docId));
    $param['mode'] = false;
    $result['user'] = $userId;
    $result['fio'] = UserHome::getFIO($result['user']);
    $t['user'] = $userId;
    if (!$this->docHome->isRead($docId) && $this->docHome->
        isItToMe($docId, $userId))
        $this->docSave->setRead($docId);
    if ($this->docHome->isItToMe($docId, $userId))
        $this->docSave->setNotChanged($docId,$userId);
    if($doc['id_parent']==0)
        $id=$doc['id'];
    elseif($doc['id_parent']!=0){
        $parent=$this->docHome->getOneDocById($this->docHome->
            getParent($doc['id']));
        if($parent['id_parent']==0)
            $id=$parent['id'];
        else
            $id=$this->docHome->getParent($parent['id']);
    }
    if (!isset($temp['id_user'])) {
        $temp['rows2'] = $this->printHtml->outListUser($this->
            docHome->checkOtherUsers($docId));
        $temp['rows3'] = $this->printHtml->outListUser($this->
            docHome->checkOtherWatchers($docId));
        $temp['rows4'] = $this->printHtml->outOneUser($this->
            docHome->checkResponsible($docId));
    } else {
        $temp['rows2'] = $this->printHtml->
            outOneUser($temp['id_user']);
        $temp['rows3'] = $this->printHtml->outListUser($this->
            docHome->checkOtherWatchers($docId));
        $temp['rows4'] = $this->printHtml->outOneUser($this->
            docHome->checkResponsible($docId));
    }
    if($doc['is_consistent']==1)
        $temp["checked"]=true;
    $temp['addFile'] = $this->templateHome->parse($this->modules_root.
        "edms/tpl/docs_card/add_file.tpl", []);
    $temp['type'] = 'sign';
    $temp['user']=$result['user'];
    $param['body']=$this->templateHome->parse($this->modules_root.
        "edms/tpl/docs/".$this->tpl(), $temp);
}

```

```

$param['buttons']=$this->buttons($doc,$userId);
$param['files']=$this->printHtml->getListOfFiles($docId,
    $param['mode']);
$temp['table'] = $this->printHtml->
    makeHistoryList($docId,$userId);
$param['history'] = $this->templateHome->parse($this->modules_root
    ."edms/tpl/docs_card/history.tpl", $temp);
$t['content'] = $this->templateHome->parse($this->modules_root .
    "edms/tpl/docs_card/show.tpl", $param);
$result['content']=$this->templateHome->parse($this->modules_root
    ."edms/tpl/start.tpl", $t);
$result['top'] = $this->templateHome->parse($this->modules_root .
    "edms/tpl/top/top.tpl", $result);
$result=$this->templateHome->parse($this->modules_root .
    "edms/tpl/all.tpl", $result);
return $result;
}

public function showCreate($docId, $userId){
    $result['user'] = $userId;
    $result['fio'] = UserHome::getFIO($result['user']);
    $temp['rows0'] = $this->printHtml->makeOption($this->docHome->
        getProtocol($docId),$docId);
    $temp['rows5'] = $this->printHtml->makeOption($this->docHome->
        getOneDocType(5),5);
    $temp['rows1'] = $this->printHtml->makeOption($this->right->
        getAssigment($result['user']), $result['user']);
    $temp['rows3'] = $temp['rows2'] = $this->printHtml->
        makeDepartmentTree($result['user']);
    $temp['addFile'] = $this->templateHome->parse($this->modules_root
        ."edms/tpl/docs_card/add_file.tpl", []);
    $t['body'] = $this->templateHome->parse($this->modules_root .
        "edms/tpl/docs/". $this->tpl(), $temp);
    $t['top'] = $this->templateHome->parse($this->modules_root .
        "edms/tpl/top/top.tpl", $result);
    $t['content'] = $this->templateHome->parse($this->modules_root .
        "edms/tpl/docs_card/create.tpl", $t);
    $result['content']=$this->templateHome->parse($this->modules_root
        ."edms/tpl/all.tpl", $t);
    $result= $this->templateHome->parse($this->modules_root .
        "edms/tpl/start.tpl", $result);
    return $result;
}

public function save($userId, $params, $files){
    if ($params['id']!='')
        $doc = $this->docHome->getOneDocById($params['id']);
    if($params['id']=="){
        $textId = $this->docSave->saveText($params);
        $this->docSave->saveFiles($files, $textId);
        if ($params['from'] != $userId)
            if($doc['id_user']!=$userId)
                $params['id_operator'] = $userId;
        $params['date_create']=$params['date_first'] = time();
        $params['id_text'] = $textId;
        $params['id_resp']=$this->docHome->
            getUserFromExecution($params['id_resp']);
        if($params['id_prot']!=null){
            $params['id_parent'] = $params['id_prot'];
            $params['code']=$this->docHome->getCode($params['id_prot'])
                ['code'];
        }else
            $params['code'] = $this->docSave->saveMainDoc();
        $params['id_owner'] = $params['from'];
    }
}

```

```

$temp = $params['id_user'];
$temp2 = $params['id_watcher'];
if (count($temp) > 1) {
    $params['id_parent'] = $this->docSave->saveDocFlow($params);
    $this->docSave->setChanged($params['id_parent'],$userId);
    $params['type']=4;
}
if (is_array($temp)) {
    foreach ($temp as $key => $id) {
        $params['id_user'] = $this->docHome->
            getUserFromExecution($id);
        if($params['id_user']==$params['id_resp'])
            $params['is_responsible']=1;
        else
            unset($params['is_responsible']);
        if (is_array($temp2)) {
            foreach ($temp2 as $key2 => $id2) {
                $params['id_watcher'] = $this->docHome->
                    getUserFromExecution($id2);
                $this->docSave->setChanged($this->docSave-
                    >saveDocFlow($params),$userId);
            }
        }
    }
}
}
public function buttons($doc,$userId){
    $result="";
    if($doc['is_closed']==0) {
        if ($doc['id_owner'] == $userId || $doc['id_user'] ==
            $userId || $doc['id_operator'] == $userId){
            if ($doc['id_user'] == $userId) {
                if (($doc['is_consistent']==1
                    &&$doc['is_responsible'] == 1&&$this->docHome->
                    allSignChildIsClose($doc['id_parent']))) ||
                    $doc['is_consistent']==0) {
                    $result .= "<div class=\"row\"><button
                        type=\"button\" class=\"btn btn-
                        outline-success btn-sm btn-block\"
                        onclick='Sign(1,\" . $doc['id'] .
                        \",this,\" . $userId .
                        \")'>Подписать</button></div>";
                    $result .= "<div class=\"row\"><button
                        type=\"button\" class=\"btn btn-outline-
                        danger btn-sm btn-block\"
                        onclick='Sign(0,\" . $doc['id'] . \",this,\"
                        . $userId . \")'>Отклонить</button></div>";
                    $result .= "<div class=\"row\"><button type=\"button\"
                        class=\"btn btn-outline-primary btn-sm btn-
                        block\" onclick='Errand(\" . $doc['id'] . \",\" .
                        $userId . \")'>Поручить</button></div>";
                }
            }
        }
        $result="<div class=\"alert alert-info\"
            role=\"alert\">Подписание возможно после остальных
            согласующих</div>";
    }
    elseif ($doc['id_owner']== $userId||$doc['id_operator']==
        $userId){//если документ отправлен от меня
        if(!isset($doc['date_show']))
            $result.="<div class=\"row\"><button type=\"button\"
                class=\"btn btn-outline-danger btn-sm btn-
                block\" onclick='CloseAll(\".$doc['id'].\",
                this)'>Отменить процесс</button></div>";
        }
    }
}

```

```

        return $result;
    }
    public function preview($docId, $userId){
        if (!$this->docHome->isRead($docId) && $this->docHome->isItToMe($docId,
            $userId))
            $this->docSave->setRead($docId);
        if ($this->docHome->isItToMe($docId, $userId))
            $this->docSave->setNotChanged($docId,$userId);
        $doc = $this->docHome->getOneDocById($docId);
        $temp=$this->docHome->getPreview($docId);
        if (!$this->docHome->isRead($docId) && $this->docHome->
            isItToMe($docId, $userId))
            $this->docSave->setRead($docId);
        $temp['btn'] = $this->buttons($doc, $userId);
        $temp['files'] = $this->printHtml->
            getListOfFiles($docId,true,true);
        $result['content']=$this->templateHome->parse($this->
            modules_root."edms/tpl/docs_card/preview.tpl",$temp);
        return $result;
    }
}

```

Листинг Г.4 – Document.class.php

```

<?php
require_once($modules_root."edms/class/DocumentationHome.class.php");
require_once($modules_root."edms/class/DocSaveHome.class.php");
require_once($modules_root."edms/class/ref/IStrategy.php");
require_once($modules_root."edms/class/ValidationHome.class.php");
require_once($modules_root."edms/class/RightsHome.class.php");
require_once($modules_root."edms/class/PrintHTMLHome.class.php");
class Document implements IStrategy
{
    private $check;
    private $right;
    private $printHtml;
    private $templateHome;
    private $modules_root;
    private $docHome;
    private $docSave;
    public function __construct(){
        $this->check=new ValidationHome();
        $this->right=new RightsHome();
        $this->printHtml=new PrintHTMLHome();
        $this->templateHome=$GLOBALS['templateHome'];
        $this->modules_root=$GLOBALS['modules_root'];
        $this->docHome=new DocumentationHome();
        $this->docSave=new DocSaveHome();
    }
    public function tpl(){
        return "task.tpl";
    }
    public function show($docId,$userId){
        $result['user']=$t['user'] = $userId;
        $result['id'] = $docId;
        $result['fio'] = UserHome::getFIO($result['user']);
        $t=$this->docHome->getOneProtocol($result['id']);
        if (($this->docHome->haveChild($result['id'])&&($t['id_owner']==
            $userId||$t['id_operator']==$userId))
            ||($t['id_owner']!=$userId&&$t['id_operator']!=$userId)){
            $t['disabled']=true;
            $t['files'] = $this->printHtml->getListOfFiles

```

```

        ($result['id'],true);
    }else{
        $t['files'] = $this->printHtml->getListOfFiles
            ($result['id'],false);
        $t['addFile'] = $this->templateHome->parse($this->
            modules_root . "edms/tpl/docs_card/add_file.tpl", []);
    }
    $doc=$this->docHome->getOneDocById($docId);
    if($t['is_signed']==1&&$this->docHome->isMainSign($userId,$docId))
        $name=UserHome::getFIO($t['id_owner']).', '.UserHome::
            getShortPosition($t['id_owner']);
    $t['rows1']=$this->printHtml->makeOption([[ 'id'=>$t['id_owner'],
        'name'=>$name]], $t['id_owner']);
    $t['show']=true;
    if($doc["id_owner"]==$userId||$doc["id_operator"]==$userId)
        unset($t['disabled']);
    $t['buttons']=$this->buttons($doc,$userId);
    $t['content'] = $this->templateHome->parse($this->modules_root .
        "edms/tpl/docs_card/show_3.tpl", $t);
    $result['content']=$this->templateHome->parse($this->modules_root
        ."edms/tpl/all.tpl", $t);
    $result=$this->templateHome->parse($this->modules_root .
        "edms/tpl/start.tpl", $result);
    return $result;
}
public function save($userId, $items, $files){
    if($items['id']!=""){
        $doc=$this->docHome->getOneDocById($items['id']);
        if(isset($items['name'])&&isset($items['text'])){
            $this->docSave->update(['name'=>$items['name'],
                'text'=>$items['text']], 'text', $doc['id_text']);
        }
        if($items['date_create']!=""){
            $items['date_create']=$items['date_create'].'
                '.date('H').':'.date('i').':'.date('s');
            $items['date_create']= new DateTime($items
                ['date_create']);
            $temp['date_create']=$items['date_create']->
                getTimestamp();
        }
        if(isset($items['number'])){
            $temp['number']=$items['number'];
        }
        if(count($temp)!=0)
            $this->docSave->update($temp, 'docs_flow', $doc['id']);
        $this->docSave->deleteFiles($items['scans'],
            $doc['id']);
        $this->docSave->saveFiles($files, $doc['id_text']);
    }else{
        $params=array('id_type'=>$items['id_type'],'code'=>
            $this->docSave->saveMainDoc(),
            'id_text'=>$this->docSave->saveText($items),
            'id_owner'=>$items['id_user'],
            'date_first'=>time());
        if($items['id_user']!= $userId)
            $params['id_operator']=$userId;
        $this->docSave->saveFiles($files, $params['id_text']);
        $this->docSave->saveDocFlow($params);
    }
}
public function showCreate($docId, $userId){
    $temp['addFile'] = $this->templateHome->parse($this->modules_root
        ."edms/tpl/docs_card/add_file.tpl", []);
    $temp['date_first']=date("Y-m-d",time());
    $temp['rows1'] = $this->printHtml->makeOption($this->right->

```

```

        getAssignment($userId, $userId);
$result['user'] = $userId;//поменять как то
$t['top'] = $this->templateHome->parse($this->modules_root .
    "edms/tpl/top/top.tpl", $this->printHtml->
        makeHeader($userId));
$t['content'] = $this->templateHome->parse($this->modules_root .
    "edms/tpl/docs_card/create.tpl", $t);
$result['content']=$this->templateHome->parse($this->modules_root
    ."edms/tpl/all.tpl", $t);
return $this->templateHome->parse($this->modules_root .
    "edms/tpl/start.tpl", $result);
}
public function buttons($doc,$userId){
    $result=" <div class=\"row\"><button type=\"button\" class=\"btn
        btn-outline-primary btn-sm btn-block\" onclick='Errand(\".
            $doc['id'].\", \"$userId.\")'>Поручить</button></div>";
    return $result;
}
public function preview($docId, $userId){
    $doc = $this->docHome->getOneDocById($docId);
    $temp=$this->docHome->getPreview($docId);
    if (!$this->docHome->isRead($docId) && $this->docHome->isItToMe($docId,
        $userId))
        $this->docSave->setRead($docId);
    if($doc['is_signed']==1){
        $temp['sgn']="Статус: <font color='#27AE60'>
            ПОДПИСАНО</font>";
        $temp['mainsgn']=$this->docHome->getMainSignTitle($docId);
        $temp['other']=$this->docHome->getOtherSign($docId);
        $temp['filesgn']="<a href=\"\"/edms/doc_sign/?id_doc=\".$docId.
            \"\" traget=\"\"_blank\"\" >Подписи</a>";
    }
    elseif($this->docHome->isActiveSigning($doc['id']))
        $temp['sgn']="Статус: НА ПОДПИСАНИИ";
    $temp['btn'] = $this->buttons($doc,$userId);
    $temp['files'] = $this->printHtml->getListOfFiles
        ($docId,true,true);
    $result['content']=$this->templateHome->parse($this->modules_root.
        "edms/tpl/docs_card/preview.tpl", $temp);
    return $result;
}
}
}

```

Листинг Г.5 – DocumentationHome.class.php

```

require_once($modules_root."edms/class/RightsHome.class.php");
require_once($modules_root."edms/class/UserHome.class.php");
require_once("core/lib/vendor/phpoffice/phpword/bootstrap.php");

use PhpOffice\PhpWord\IOFactory;
use PhpOffice\PhpWord\TemplateProcessor;
use PhpOffice\PhpWord\PhpWord;

class DocumentationHome
{
    protected $db_edms;
    protected $db_esia;
    protected $right;
    protected $templateHome;
    protected $modules_root;
    public static $docsDir="files/lk/docs/documentation/";
    protected static $sign="files/lk/docs/documentation/";
}

```

```

protected $doc=array('id','is_deleted');
protected $files=array(
    'id',
    'id_text',
    'name',
    'weight',
    'is_deleted',
);
protected $now_date;
public function __construct() {
    $this->templateHome=$GLOBALS['templateHome'];
    $this->modules_root=$GLOBALS['modules_root'];
    $this->db_edms = $GLOBALS ["db10"];
    $this->db_esia=$GLOBALS["db2"];
    $this->now_date=new DateTime("now");
    $this->now_date = $this->now_date->format("Y-m-d");
    $this->right=new RightsHome();
}
public function getDocType($docId){
    $sql="SELECT id_type FROM docs_flow WHERE id=".$docId;
    return $this->db_edms->getOneData($sql,['id_type'])['id_type'];
}
public function getShowType($id){
    $sql="SELECT is_show from docs_types WHERE id=".$id;
    return $this->db_edms->getOneData($sql,['is_show'])['is_show'];
}
public function getOneDocType($id){
    $sql="SELECT * FROM docs_types WHERE id=".$id;
    return $this->db_edms->getListData($sql,['id','name']);
}
public function getDoc($textId, $code){
    $sql = "SELECT id FROM docs_flow Where id_text=" . $textId . " AND
        code=" . $code;
    return $this->db_edms->getListData($sql, ['id']);
}
public function getAllTypes(){
    $sql="SELECT * FROM docs_types WHERE is_deleted=0 AND is_show=0";
    return $this->db_edms->getListData($sql,['id','name']);
}
public function getCode($id){
    $sql="SELECT code FROM docs_flow where id=".$id;
    return $this->db_edms->getOneData($sql,['code']);
}
public function getMainSignTitle($docId){
    $sql="SELECT id from docs_flow Where id_parent=".$docId." AND
        id_type=5 AND is_deleted=0";
    $ids= $this->db_edms->getListData($sql,['id']);
    $user="";
    if(isset($ids[0]['id']))
        foreach ($ids as $key=>$id){
            $sql="SELECT id_user from docs_flow Where id_parent=
                ".$id['id']." AND is_responsible=1 and id_type=5
                AND is_deleted=0 and is_signed=1 ORDER BY
                date_end";
            $uid=$this->db_edms->getOneData($sql,['id_user'])
                ['id_user'];
            if($uid!=null||$uid!=""){
                $user.=UserHome::getFIO($uid).' ('.UserHome::
                    getShortPosition($uid).")";
            }
            if($key<count($ids)-1)
                $user.=", ";
        }
    }
}

```

```

        return $user;
    }
    public function getOtherSign($docId){
        $sql="SELECT id from docs_flow Where id_parent=".$docId." AND
            id_type=5 AND is_deleted=0";
        $ids= $this->db_edms->getListData($sql,['id']);
        $user="";
        if(isset($ids[0]['id']))
            foreach ($ids as $key=>$id){
                $sql="SELECT id, id_user,code,id_text, is_signed from
                    docs_flow Where id_parent=".$id['id']." AND
                    is_responsible=0 and id_type=5 AND is_deleted=0 and
                    type=4 AND is_closed=1 ORDER BY is_signed DESC ";
                $uids=$this->db_edms->getListData($sql,['id',
                    'id_user','code','id_text','is_signed']);
                $uids=DocumentationHome::deleteRepeat($uids);
                foreach ($uids as $key=>$uid){
                    if($uid['id_user']!=null||$uid['id_user']!="") {
                        if($uid['is_signed']==1)
                            $status="ПОДПИСАНО";
                        else
                            $status="ОТКЛЮЧЕНО";
                        $user .= UserHome::getFIO($uid['id_user']) . '
                        (' . UserHome::getShortPosition(
                            $uid['id_user']).") ".$status;
                        if($key<count($uids)-1)
                            $user.="<br> ";
                    }
                }
            }
        return $user;
    }
    public static function getMainSignForFile($docId){
        $sql="SELECT id from docs_flow Where id_parent=".$docId." AND
            id_type=5 AND is_deleted=0";
        $ids= $GLOBALS ["db10"]->getListData($sql,['id']);
        $user['other']=array();
        $user['main']=array();
        return $user;
    }
    public function isMainSign($userId,$docId){
        $sql="SELECT id from docs_flow Where id_parent=".$docId." AND
            id_type=5 AND is_deleted=0";
        $ids= $this->db_edms->getListData($sql,['id']);
        if(isset($ids[0]['id']))
            foreach ($ids as $key=>$id){
                $sql="SELECT id_user from docs_flow Where
                    id_parent=".$id['id']." AND is_responsible=1 and
                    id_type=5 AND is_deleted=0 and is_signed=1";
                $user=$this->db_edms->getOneData($sql,['id_user'])
                    ['id_user'];
                if($user!=null||$user!="")
                    if($user==$userId||RightsHome::isAssignment($user,
                        $userId))
                        return true;
            }
        return false;
    }
    public function getOneProtocol($docId){
        $sql="SELECT df.id,t.name,df.number,t.text,df.date_create,
            df.id_owner, df.id_operator,df.date_first, df.id_type,
            df.is_signed FROM docs_flow df
            LEFT JOIN text t ON t.id=df.id_text WHERE df.id=".$docId;
    }

```

```

$result=$this->db_edms->getOneData($sql,['id','name','number',
    'text','date_create','id_owner','id_operator','date_first',
    'id_type','is_signed']);
if($result['date_create']!='01.01.1970 03:00'){
    $result['date_create'] = new DateTime($result
        ['date_create']);
    $result['date_create'] = $result['date_create']->
        format("Y-m-d");
}
$temp=$result['date_first'];
$result['date_first'] = new DateTime();
$result['date_first']->setTimestamp($temp);
$result['date_first'] = $result['date_first']->format("Y-m-d");
return $result;
}
public function getProtocol($docId){
    $sql="SELECT df.id,t.name FROM docs_flow df
        LEFT JOIN text t ON t.id=df.id_text WHERE df.id=".$docId;
    return $this->db_edms->getListData($sql,['id','name']);
}
public function getOneDocById($docId){
    $sql="SELECT * FROM docs_flow WHERE id=".$docId;
    $result=$this->db_edms->getOneData($sql,$this->docs_flow);
    return $result;
}
public function checkResponsible($docId){
    $doc=$this->getOneDocById($docId);
    if($doc['id_user']==null&&$this->getShowType($doc['id_type'])==1)
        $sql="SELECT id_user FROM docs_flow WHERE id_parent=
            ".$docId." AND is_responsible=1";
    else{
        parentId=$this->getParent($docId);
        if($parentId!=0)
            $parent=$this->getOneDocById($parentId);
        if(is_array($parent)&&$parent['id_user']==null&&$this->
            getShowType($parent['id_type'])==1)
            $sql="SELECT id_user FROM docs_flow WHERE id_parent=
                ".$parentId." AND is_responsible=1";
        else
            $sql="SELECT id_user FROM docs_flow WHERE id=
                ".$docId." AND is_responsible=1";
    }
    return $this->db_edms->getOneData($sql,['id_user'])['id_user'];
}
public static function generateSign($docId){
    $templateProcessor = new TemplateProcessor(realpath(self::
        $docsDir."template/sign.docx"));
    $users=self::getMainSignForFile($docId);
    if(count($users['main']) > 0) {
        $templateProcessor->cloneRow('main', count($users['main']));
        $n=1;
        foreach ($users['main'] as $user){
            $templateProcessor->setValue('main#'.$n, $user);
            $n ++;
        }
    }
    if(count($users['other']) > 0) {
        $templateProcessor->cloneRow('other', count($users
            ['other']));
        $n=1;
        foreach ($users['other'] as $user){
            $templateProcessor->setValue('other#'.$n, $user);
            $n ++;
        }
    }
}

```

```

    }
    }else
        $templateProcessor->setValue('other', 'Her');
    $filename = 'Подписи_'. $docId;
    $file = realpath(self::$sign) . '/' . $filename . '.docx';
    $templateProcessor->saveAs($file);
    return $file;
}

public function getUserFromExecution($execId){
    $sql="SELECT id_user FROM user_execution WHERE id=".$execId;
    return $this->db_esia->getOneData($sql, ['id_user'])['id_user'];
}

private static function getWatchers($textId,$userId){
    $sql="Select id_watcher from docs_flow WHERE is_deleted=0 and
        id_text=".$textId." AND id_user=".$userId;
    $ws=$GLOBALS ["db10"]->getListData($sql,['id_watcher']);
    $result=array();
    foreach ($ws as $w)
        array_push($result,$w['id_watcher']);
    return $result;
}

public function checkOtherUsers($id){
    $sql="SELECT id_user FROM docs_flow WHERE id_parent=".$id." AND
        type in (0,4) AND id_user IS NOT NULL GROUP BY id_user";
    $result = $this->db_edms->getListData($sql,['id_user']);
    if($result[0])
        return $result;
    else
        return false;
}

public function checkOtherWatchers($docId){
    $doc=$this->getOneDocById($docId);
    if(isset($doc['id_user']))
        $sql="SELECT id_watcher FROM docs_flow WHERE
            id_text=".$doc['id_text']." AND
            id_user=".$doc['id_user']." AND code=".$doc['code'];
    else
        $sql="SELECT id_watcher FROM docs_flow WHERE
            id_text=".$doc['id_text']." AND
            id_parent=".$doc['id']." AND code=".$doc['code'].
            GROUP BY id_watcher";
    $result = $this->db_edms->getListData($sql,['id_watcher']);
    if($result[0])
        return $result;
    else
        return false;
}

public function isRead($docId){
    $sql="SELECT date_show FROM docs_flow WHERE id=".$docId;
    if($this->db_edms->getOneData($sql, ['date_show'])['date_show'])
        return true;
    else
        return false;
}

public function isExtension($docId){
    $sql="SELECT COUNT(*) cn FROM docs_flow df LEFT JOIN text t ON
        t.id=df.id_text WHERE df.id=".$docId." AND (t.name='Запрос
        на продление' OR t.text LIKE '%Продление даты окончания%')";
    if($this->db_edms->getOneData($sql, ['cn'])['cn']!=0)
        return true;
    else
        return false;
}

```

```

public function getParent($docId){
    $sql="SELECT id_parent FROM docs_flow WHERE id=".$docId;
    return $this->db_edms->getOneData($sql, ['id_parent'])
        ['id_parent'];
}

public function isItToMe($docId, $userId){
    $sql="SELECT COUNT(*) cn FROM docs_flow df WHERE df.id=".$docId."
        AND (df.id_user=".$userId." OR (df.id_user IN (SELECT
            id_boss FROM assignement where id_alterate=".$userId.") AND
            df.id_operator=".$userId."))";
    if($this->db_edms->getOneData($sql, ['cn'])['cn']!=0)
        return true;
    else
        return false;
}

public function haveLevelsAbove($code,$docId){
    $sql="SELECT count(*) cn from docs_flow df WHERE df.id=(SELECT
        id_parent from docs_flow WHERE docs_flow.id=".$docId.") AND
        type IN (0,4) AND ((df.id_parent<>0 AND (SELECT id_type FROM
        docs_flow WHERE docs_flow.id=df.id_parent)=1) OR ((select
        id_parent from docs_flow WHERE docs_flow.id=df.id_parent AND
        docs_flow.id_type=1)<>0 AND (select id_user from docs_flow
        WHERE docs_flow.id=df.id_parent AND docs_flow.id_type=1) is
        null))";
    $result=$this->db_edms->getOneData($sql, ['cn']);
    if($result['cn']!=0)
        return true;
    else
        return false;
}

public function isItFromMe($docId, $userId){
    $sql="SELECT COUNT(*) cn FROM docs_flow df WHERE df.id=".$docId."
        AND (df.id_owner=".$userId." OR (df.id_owner IN (SELECT
            id_boss FROM assignement where id_alterate=".$userId.") AND
            df.id_operator=".$userId."))";
    if($this->db_edms->getOneData($sql, ['cn'])['cn']!=0)
        return true;
    else
        return false;
}

public function checkChildIsRead($docId){
    $sql="SELECT date_show FROM docs_flow df WHERE df.id_parent=
        ".$docId;
    $allCh=$this->db_edms->getListData($sql, ['date_show']);
    foreach ($allCh as $ch){
        if(isset($ch['date_show']))
            return true;
    }
    return false;
}

public function allChildIsClose($id){
    $sql="SELECT COUNT(*) cn FROM docs_flow WHERE id_parent=".$id."
        AND is_closed=0 AND is_deleted=0";
    if($this->db_edms->getOneData($sql,['cn'])['cn']==0)
        return true;
    else
        return false;
}

public function allSignChildIsClose($id){
    $sql="SELECT COUNT(*) cn FROM docs_flow WHERE id_parent=".$id."
        AND id_type=5 AND is_responsible=0 AND is_deleted=0";
    $max=$this->db_edms->getOneData($sql,['cn'])['cn'];
    $sql="SELECT COUNT(*) cn FROM docs_flow WHERE id_parent=".$id."

```

```

        AND id_type=5 AND is_responsible=0 AND is_closed=1 AND
        is_deleted=0";
    $real=$this->db_edms->getOneData($sql,['cn'])['cn'];
    if(($max-$real)==0)
        return true;
    else
        return false;
}

public function haveChild($id){
    $sql="SELECT COUNT(*) cn FROM docs_flow WHERE id_parent=".$id."
        AND is_deleted=0";
    if($this->db_edms->getOneData($sql,['cn'])['cn']>0)
        return true;
    else
        return false;
}

public function childIsRead($id){
    $sql="SELECT COUNT(*) cn FROM docs_flow WHERE id_parent=".$id."
        AND NOT date_show IS NULL";
    if($this->db_edms->getOneData($sql,['cn'])['cn']>0)
        return true;
    else
        return false;
}

public function getText($docId){
    $sql="SELECT text.id, text.name, text.text FROM text
    LEFT JOIN docs_flow ON docs_flow.id=".$docId."
    WHERE text.id=docs_flow.id_text";
    return $this->db_edms->getOneData($sql, ['id','name','text']);
}

public function getAllInformation($docId){
    $sql="SELECT df.id_owner, df.id_operator, df.id_user, df.date_end,
        t.name, t.text from docs_flow df
    LEFT JOIN text t ON t.id=df.id_text
    WHERE df.id=".$docId." ";
    $result=$this->db_edms->getOneData($sql, ["id_owner",
        "id_operator", "id_user", "name", "text", "date_start",
        "date_end"]);
    $result['date_end'] = new DateTime($result['date_end']);
    $result['date_end'] = $result['date_end']->format("Y-m-d");
    return $result;
}

public function isClosed($docId){
    $sql="SELECT is_closed FROM docs_flow WHERE id=".$docId;
    if($this->db_edms->getOneData($sql, ['is_closed'])
        ['is_closed']==1)
        return true;
    else
        return false;
}

public function isClosing($docId){
    $sql="SELECT COUNT(*) cn FROM docs_flow
    LEFT JOIN text ON text.id=docs_flow.id_text
    WHERE docs_flow.id=".$docId." AND text.name LIKE
    '%Заккрытие%'";
    if($this->db_edms->getOneData($sql, ['cn'])['cn']==0)
        return false;
    else
        return true;
}
}

```

Приложение Д
РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ
(рекомендуемое)

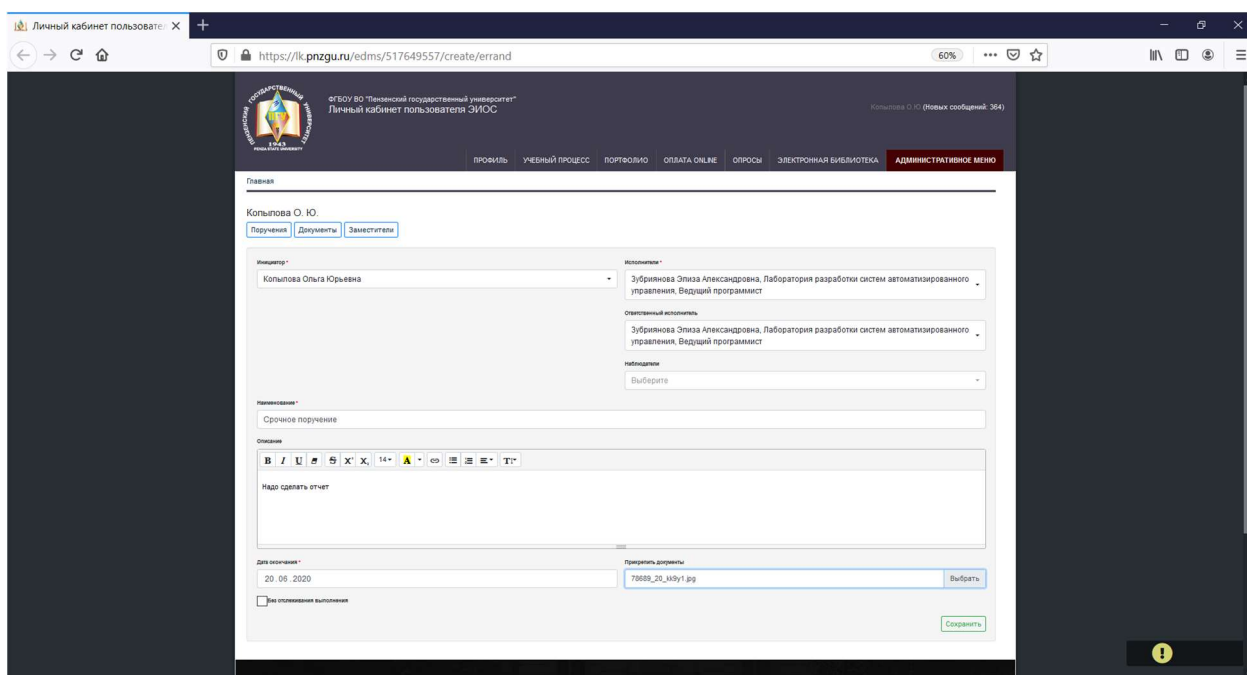


Рисунок Д.1 – Результат функционального теста №1 (заполнение формы создания поручения)

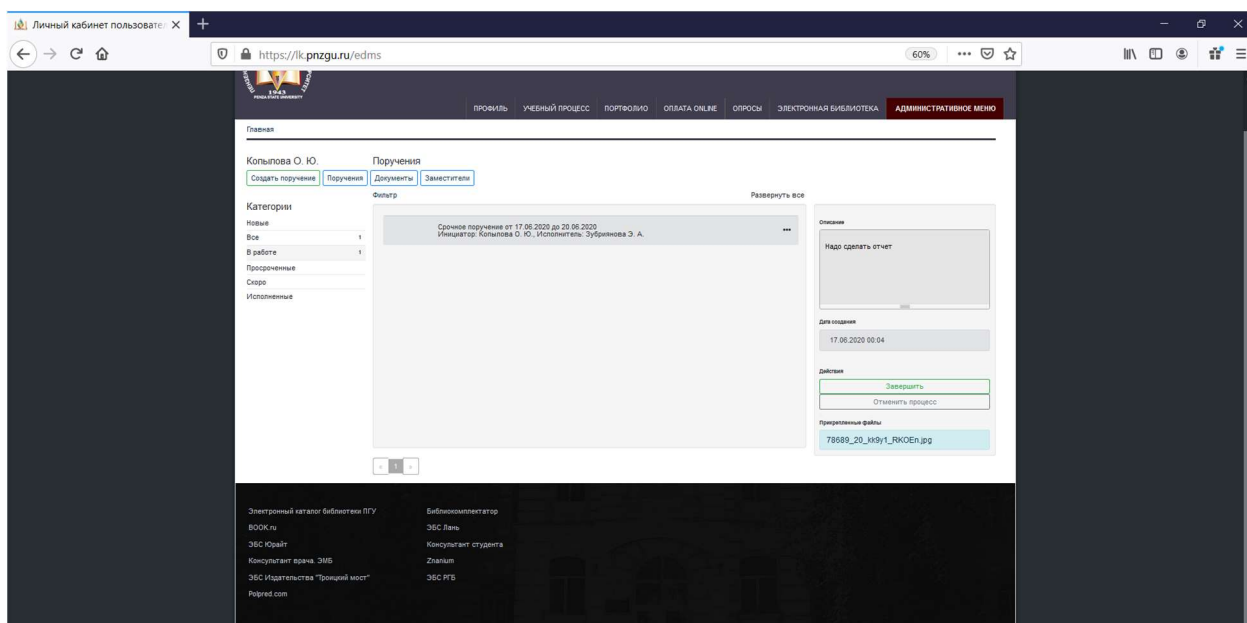


Рисунок Д.2 – Результат функционального теста №1 (сохранение поручения)

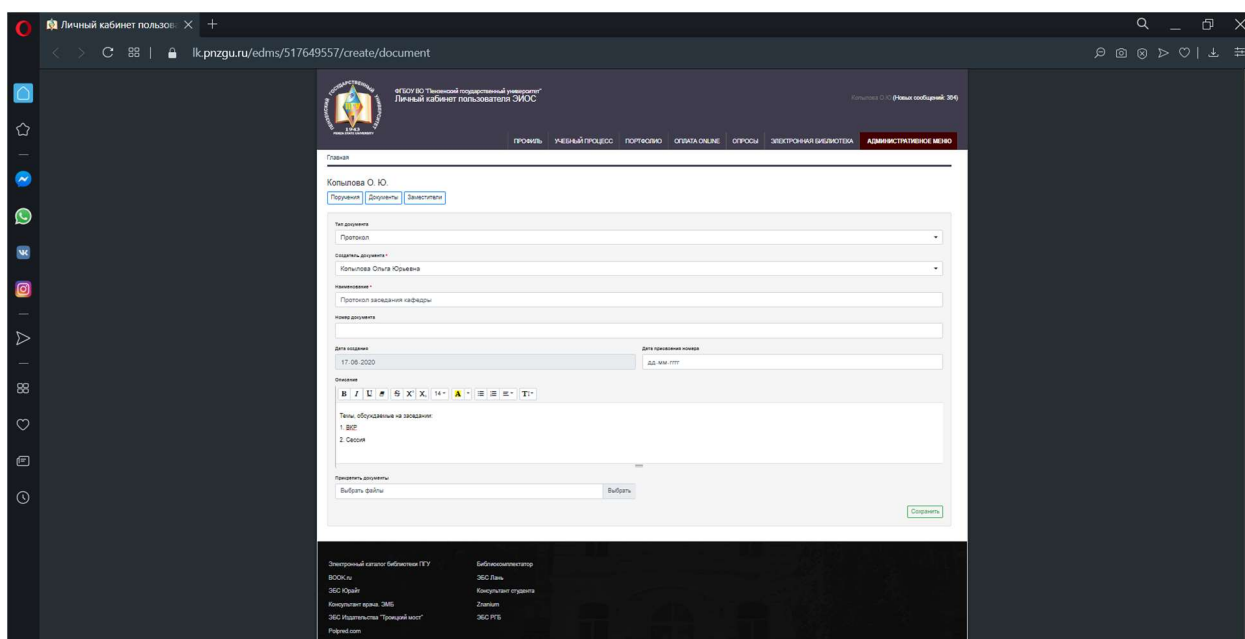


Рисунок Д.3 – Результат функционального теста №2 (заполнение формы создания документа)

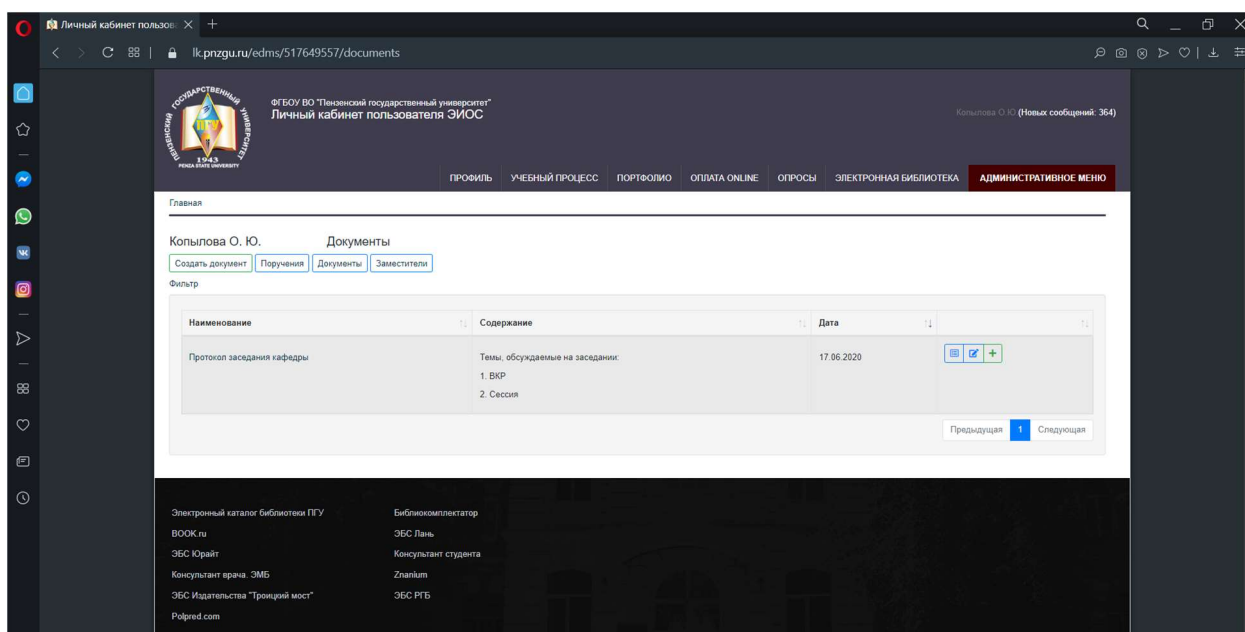


Рисунок Д.4 – Результат функционального теста №2 (сохранение документа)

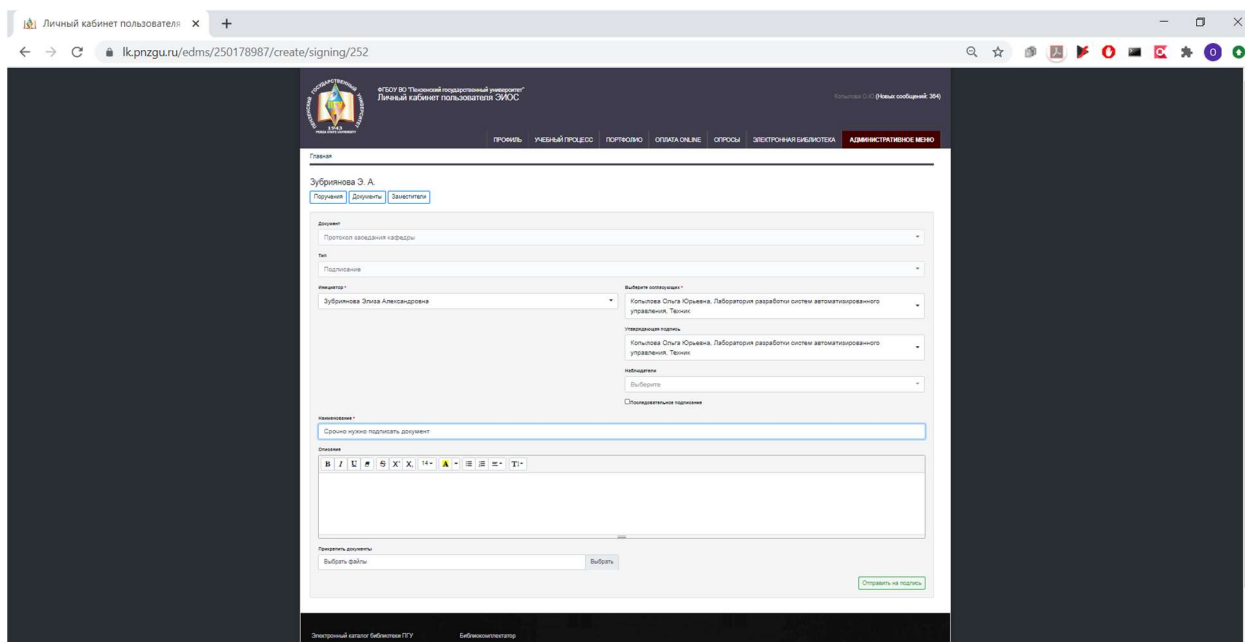


Рисунок Д.5 – Результат функционального теста №3 (заполнение формы создания подписания)

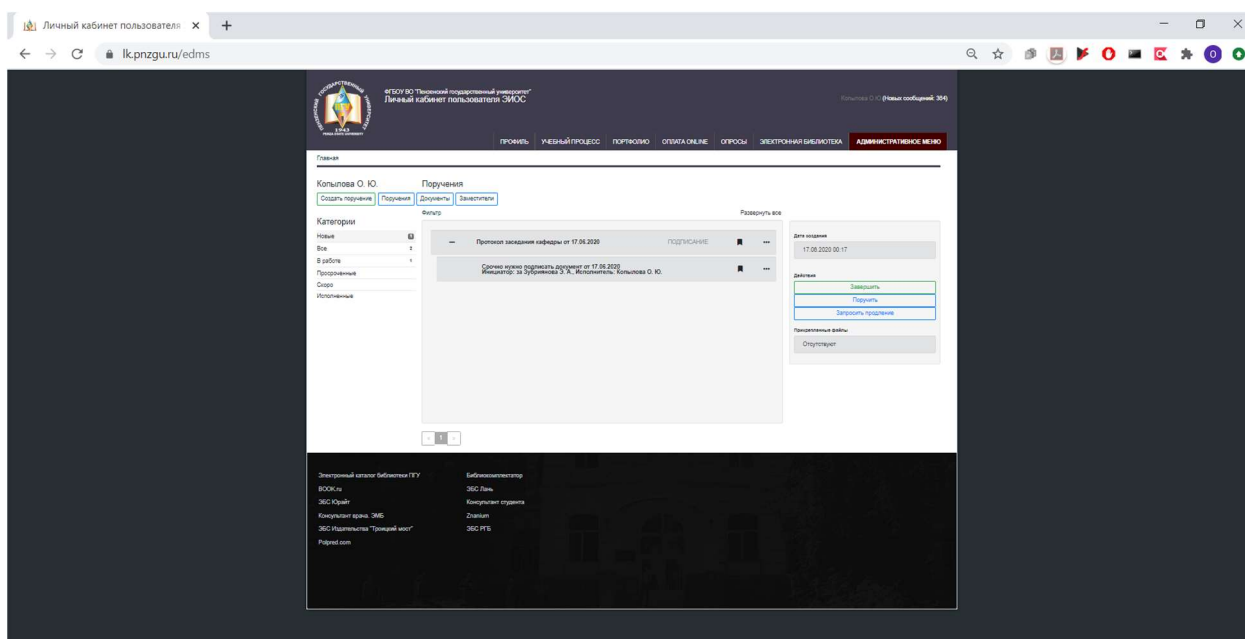


Рисунок Д.6 – Результат функционального теста №3 (сохранение подписания)

Личный кабинет пользователя

https://lk.pnzgu.ru/edms/517649557/create/assignment

70%

Личный кабинет пользователя ЭИОС

Копылова О.Ю. (Новых сообщений: 364)

ПРОФИЛЬ УЧЕБНЫЙ ПРОЦЕСС ПОРТФОЛИО ОПЛАТА ONLINE ОПРОСЫ ЭЛЕКТРОННАЯ БИБЛИОТЕКА АДМИНИСТРАТИВНОЕ МЕНЮ

Главная

Копылова О. Ю.

Поручения Документы Заместители

Кого замещает *

Копылова Ольга Юрьевна, Лаборатория разработки систем автоматизированного управления, Техник

Кто замещает *

Торопкин Роман Александрович, Лаборатория разработки систем автоматизированного управления, Техник

Тип замещения *

Помощник

Дата начала *

17.06.2020

Дата окончания *

28.06.2020

☐ Постоянное замещение

Сохранить

Электронный каталог библиотеки ПГУ Библиокомпьютер
BOOKli ЗБС Лань
ЗБС Юрайт Консультант студента
Консультант права СМБ Znanium

Рисунок Д.7 – Результат функционального теста №4 (заполнение формы создания замещения)

Личный кабинет пользователя

https://lk.pnzgu.ru/edms/517649557/assignments

70%

Личный кабинет пользователя ЭИОС

Копылова О.Ю. (Новых сообщений: 364)

ПРОФИЛЬ УЧЕБНЫЙ ПРОЦЕСС ПОРТФОЛИО ОПЛАТА ONLINE ОПРОСЫ ЭЛЕКТРОННАЯ БИБЛИОТЕКА АДМИНИСТРАТИВНОЕ МЕНЮ

Главная

Копылова О. Ю.

Создать замещение Поручения Документы Заместители

Фильтр

Мои заместители

ФИО (кто замещает)	Даты начала	Дата окончания	Тип	Статус
Торопкин Роман Александрович	17.06.2020	28.06.2020	Временное	Действует

Предыдущая 1 Следующая

Мои замещения

ФИО (кого замещает)	Даты начала	Дата окончания	Тип	Статус
Нет записей				

Предыдущая Следующая

Электронный каталог библиотеки ПГУ Библиокомпьютер

Рисунок Д.8 – Результат функционального теста №4 (сохранение замещения)

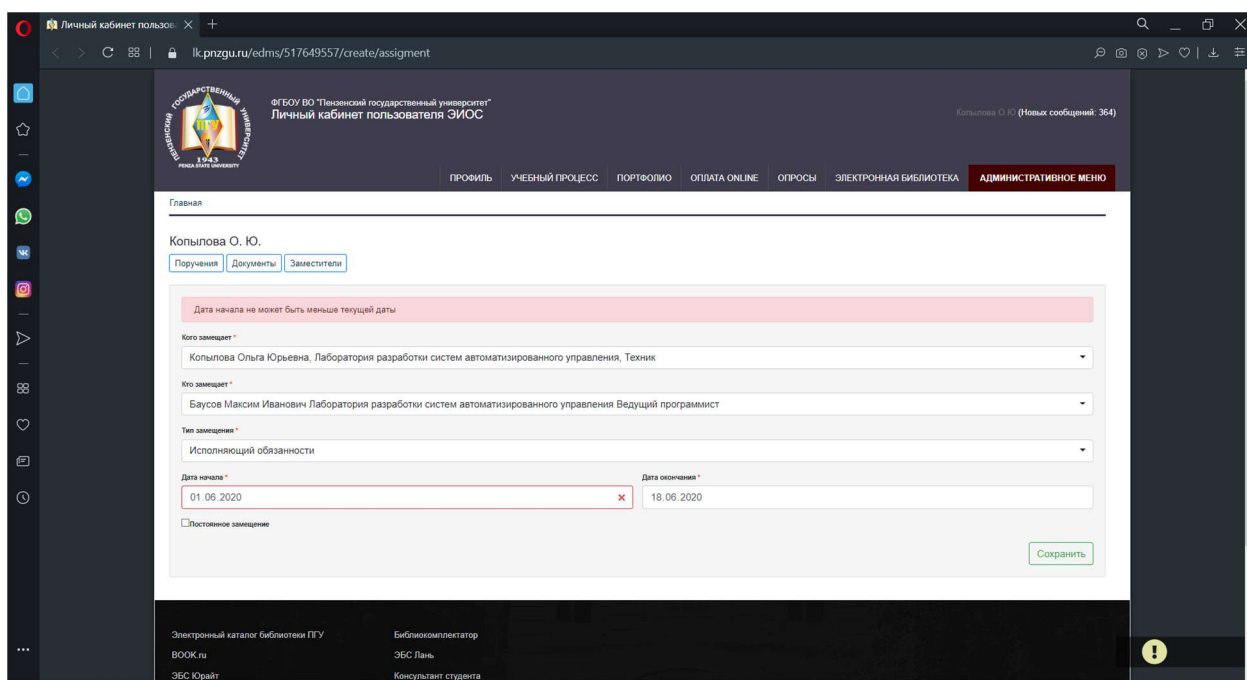


Рисунок Д.9 – Результат функционального теста №.5 (некорректные данные при создании замещения)

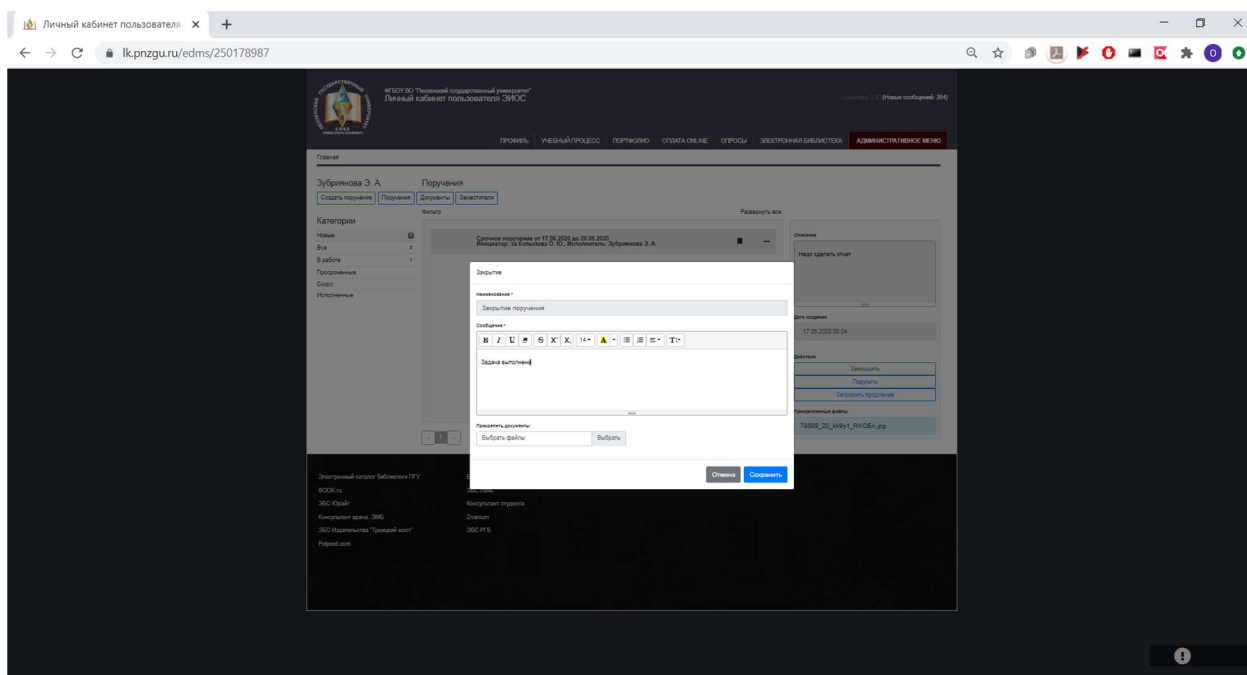


Рисунок Д.10 – Результат функционального теста №.6 (создание запроса на завершение)

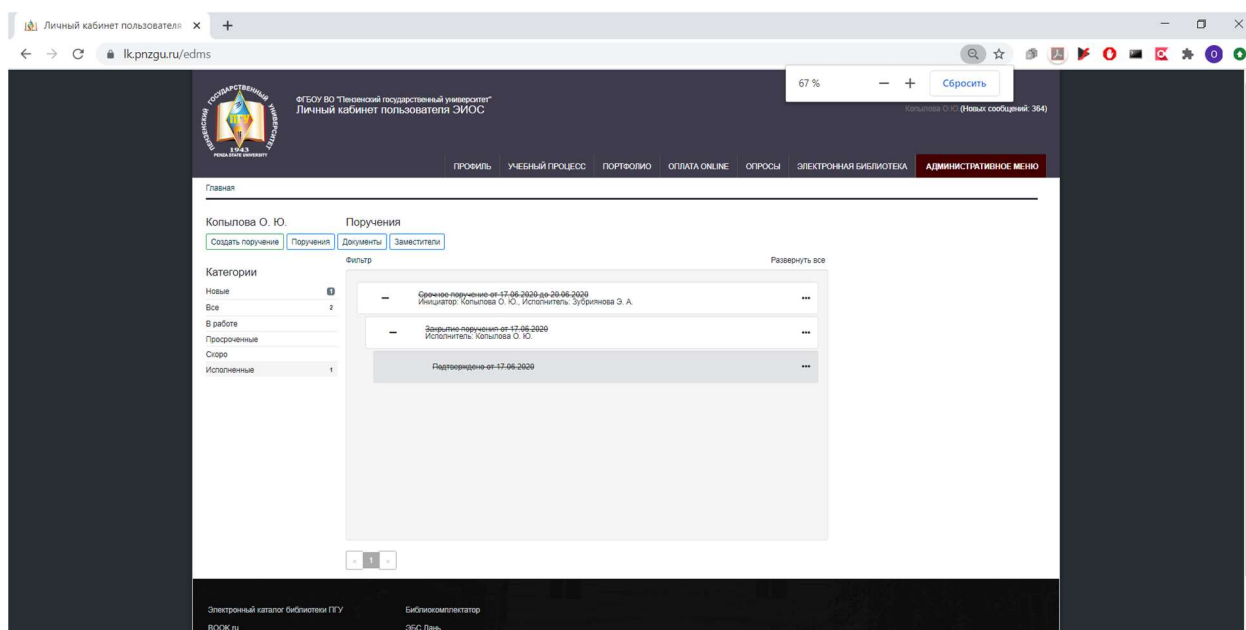


Рисунок Д.11 – Результат функционального теста №.7 (подтверждение запроса на завершение)

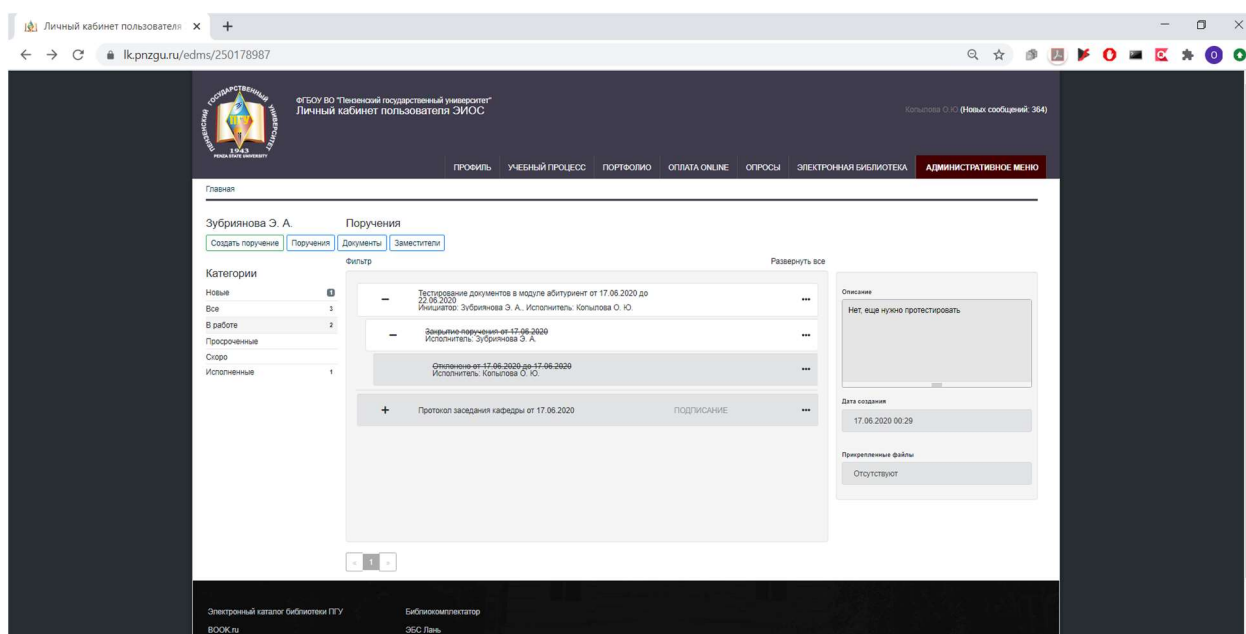


Рисунок Д.12 – Результат функционального теста №.8 (отклонение запроса на завершение)

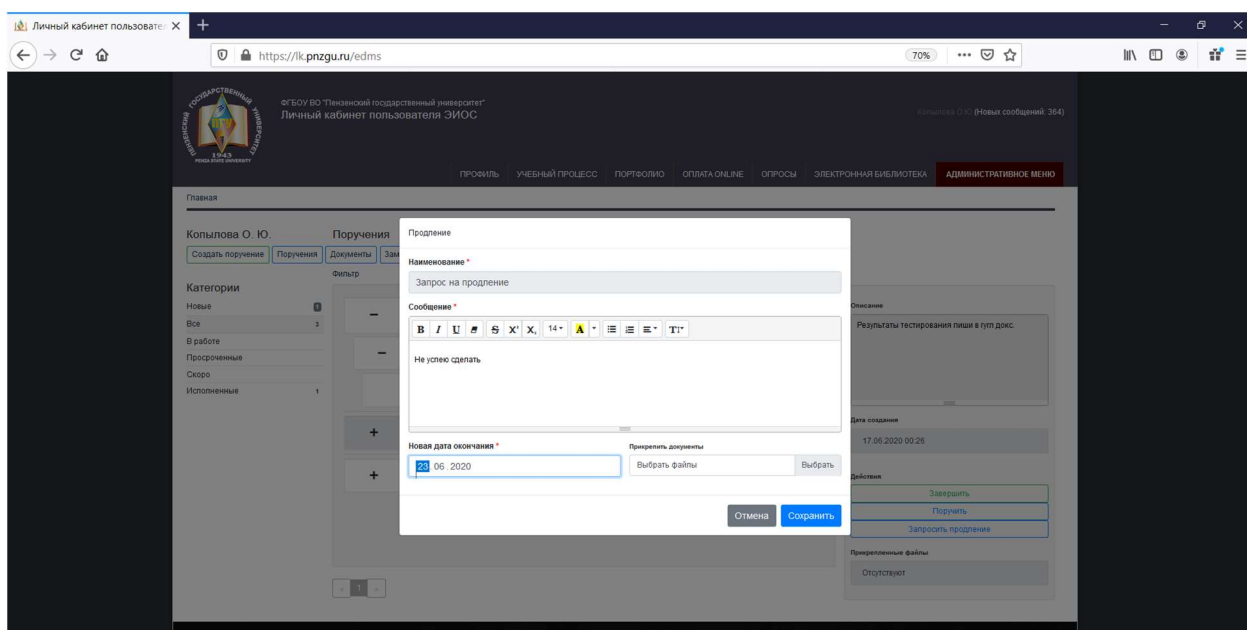


Рисунок Д.13 – Результат функционального теста №.9 (создание запроса на продление)

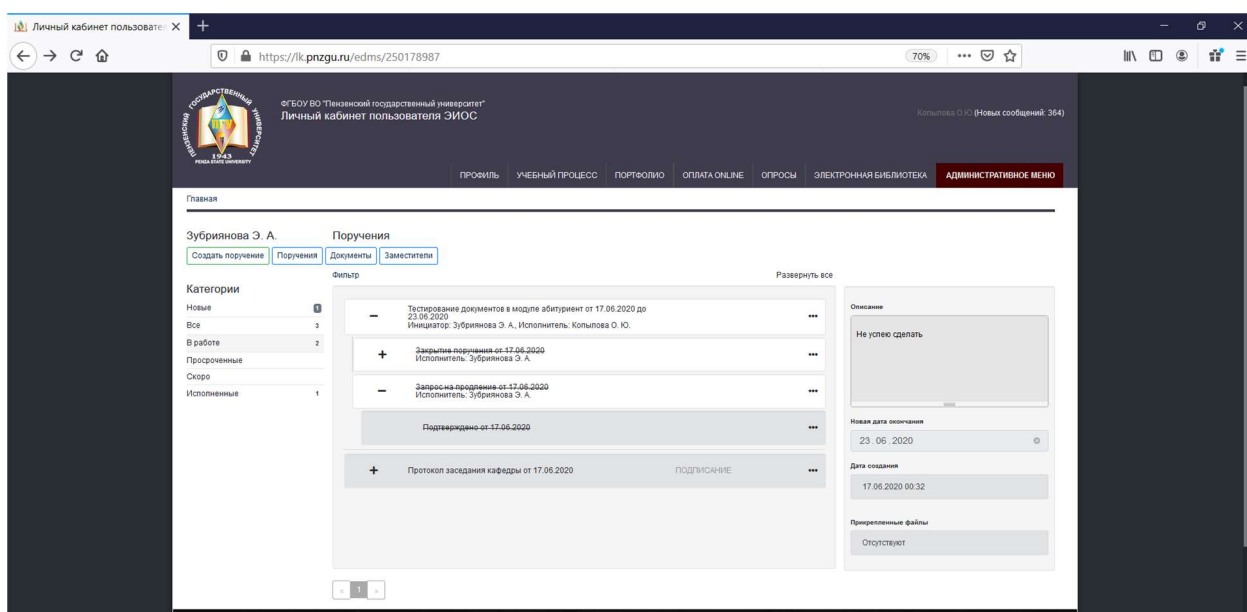


Рисунок Д.14 – Результат функционального теста №.10 (подтверждение продления выполнения)

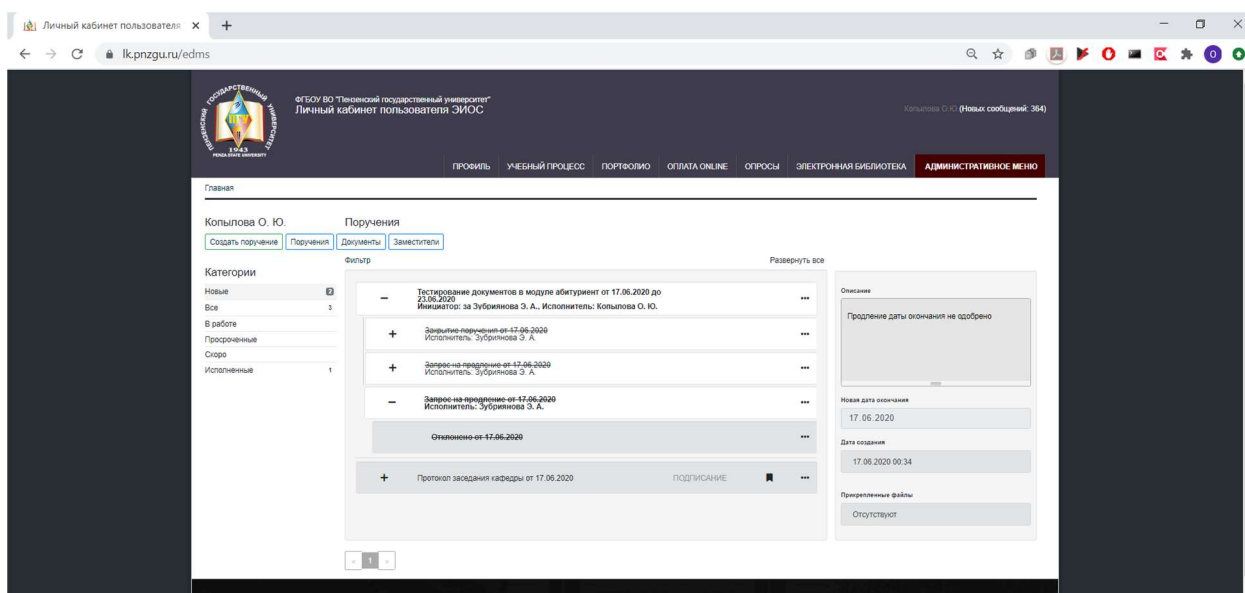


Рисунок Д.15 – Результат функционального теста №.11 (отклонение продления выполнения)

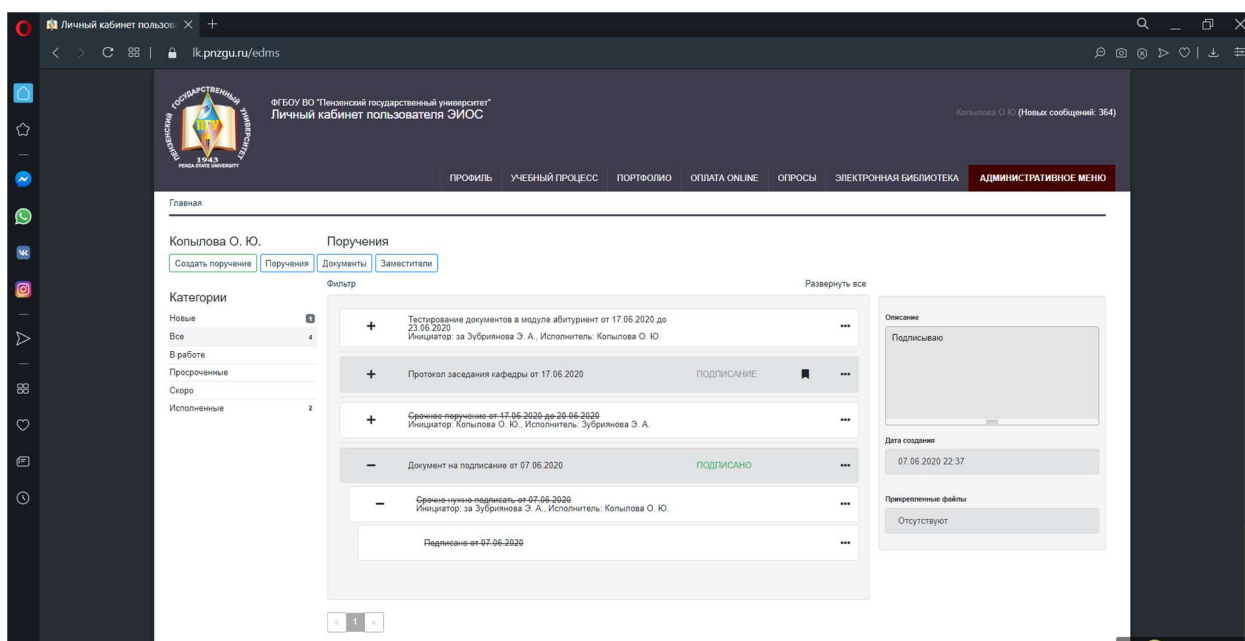


Рисунок Д.16 – Результат функционального теста №.12 (подтверждение подписания документа)

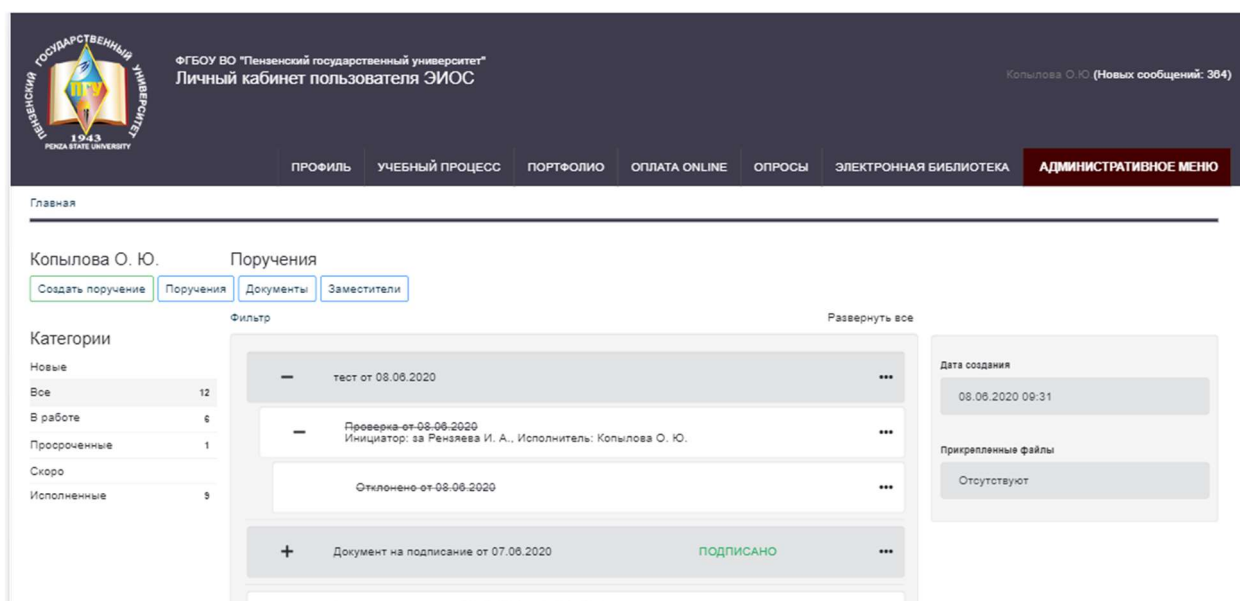


Рисунок Д.17 – Результат функционального теста №.13 (отклонение подписания документа)

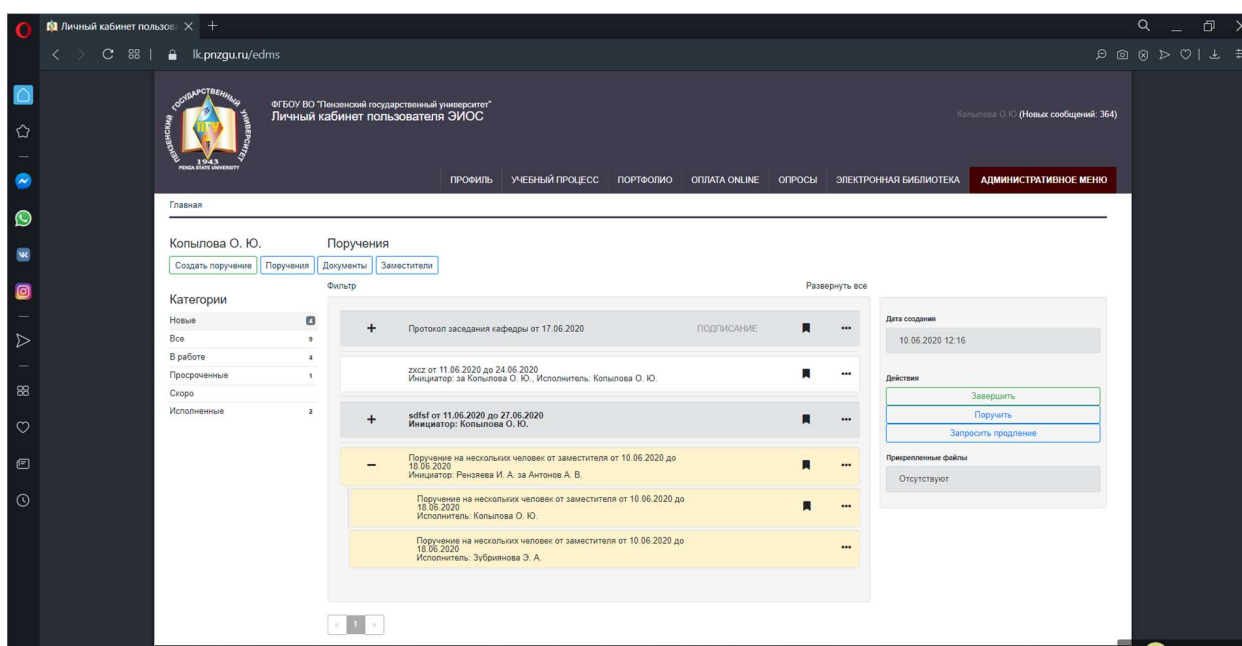


Рисунок Д.18 – Результат теста покрытия операторов №.1 (просмотр от имени одного из исполнителей)

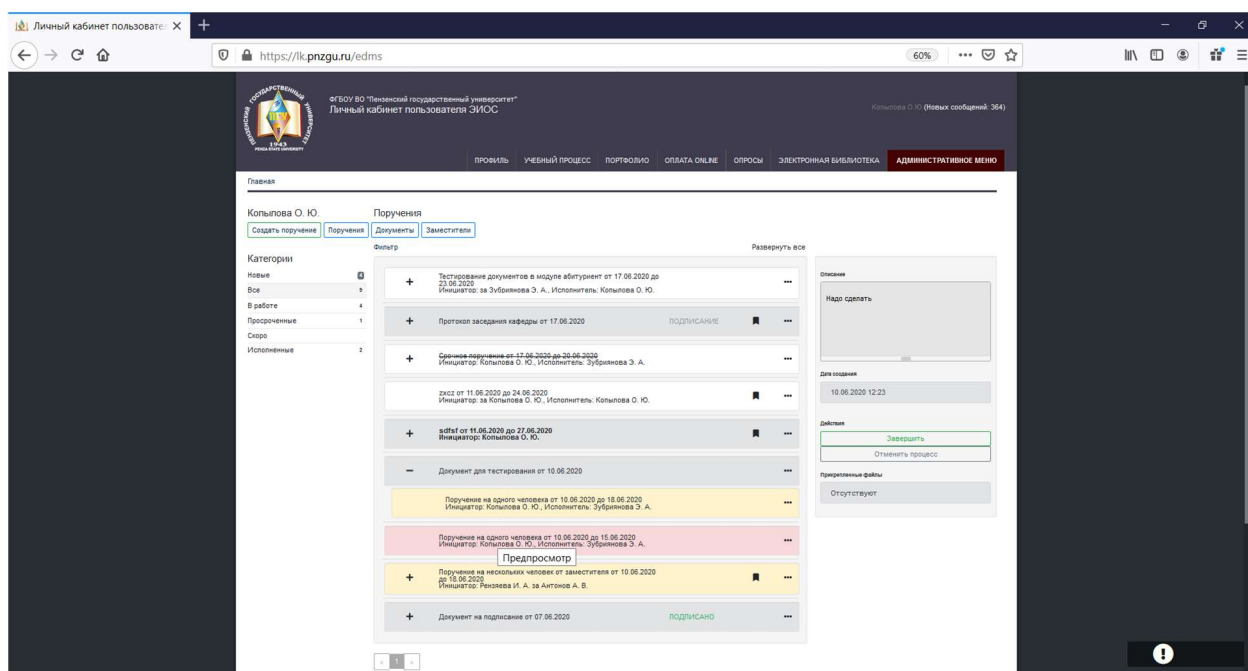


Рисунок Д.19 – Результат теста покрытия операторов №.1 (просмотр от имени инициатора)

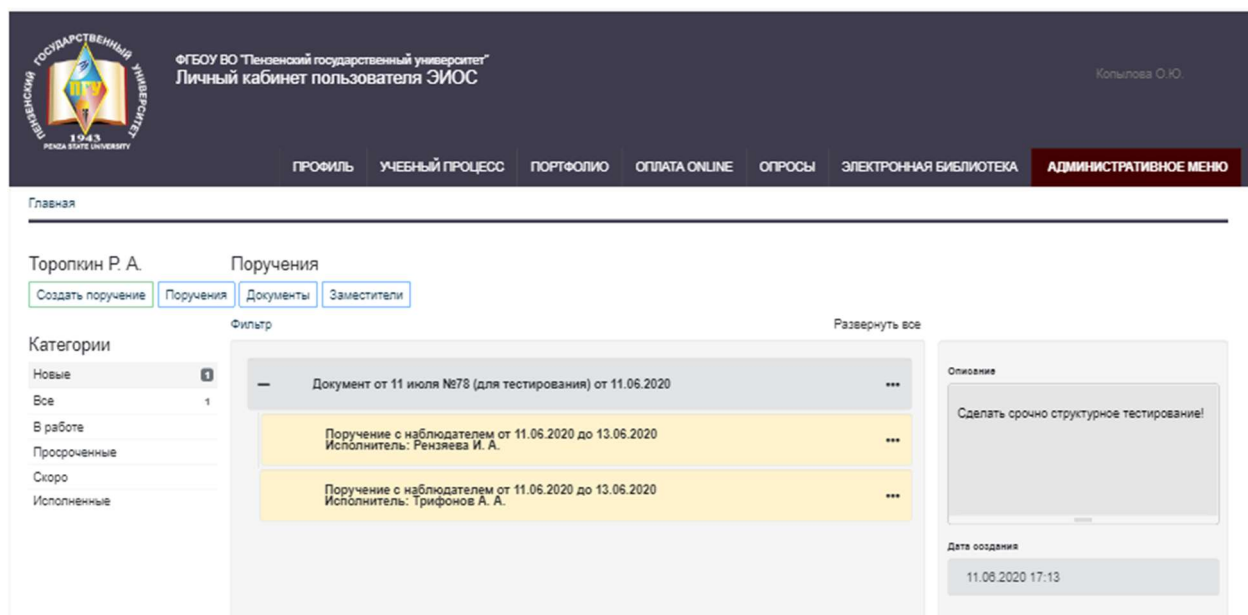


Рисунок Д.20 – Результат теста покрытия операторов №.2 (просмотр от имени наблюдателя)

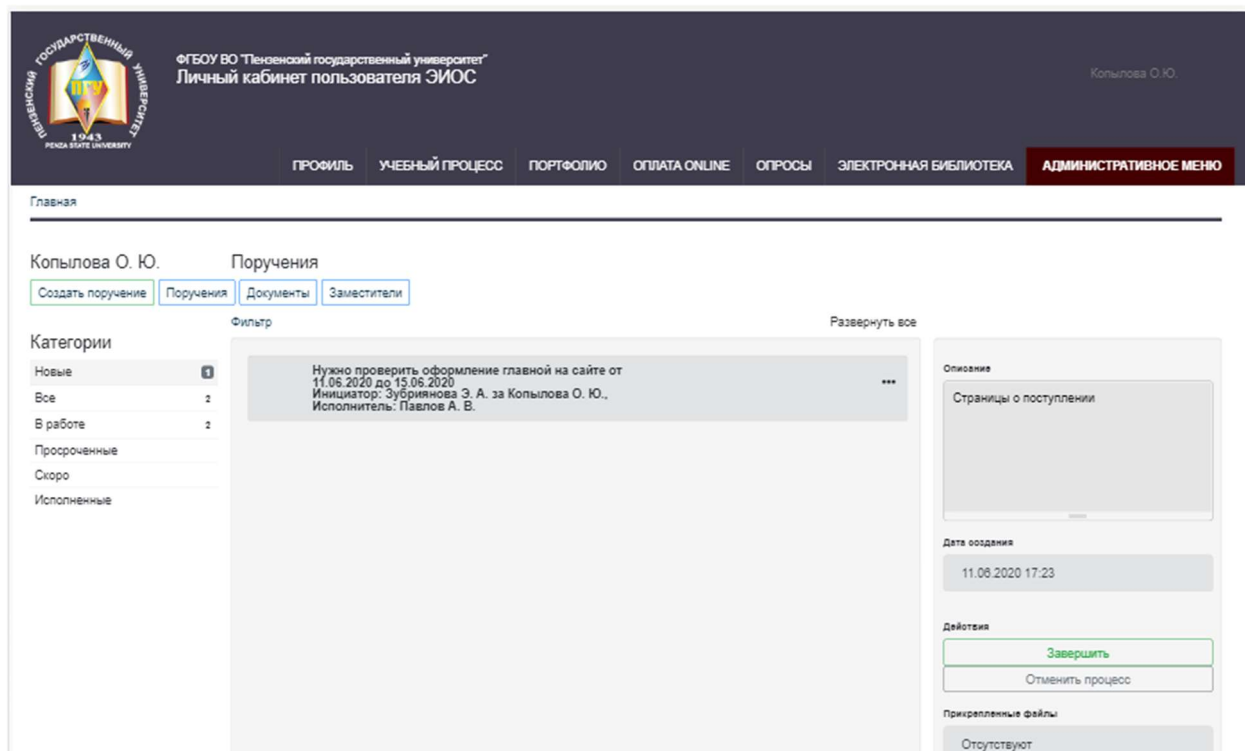


Рисунок Д.21 – Результат теста покрытия операторов №.3 (просмотр от имени инициатора)

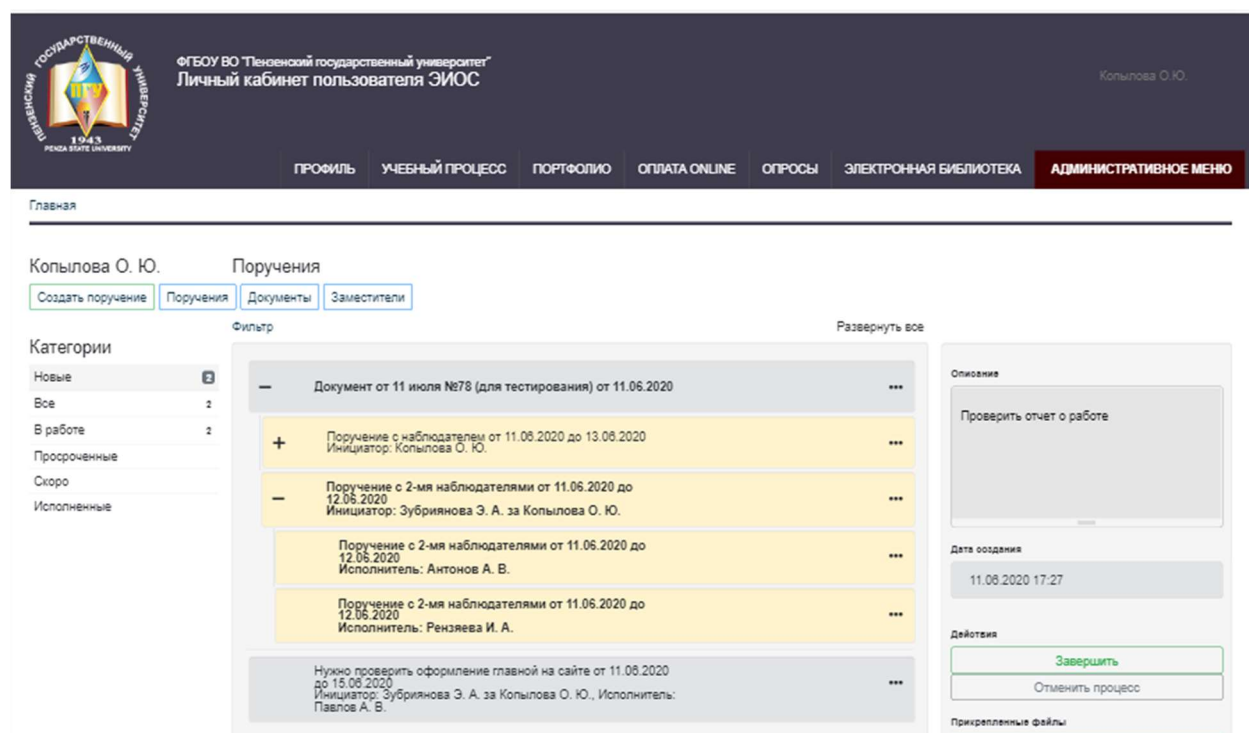


Рисунок Д.22 – Результат теста покрытия операторов №.4 (просмотр от имени инициатора)

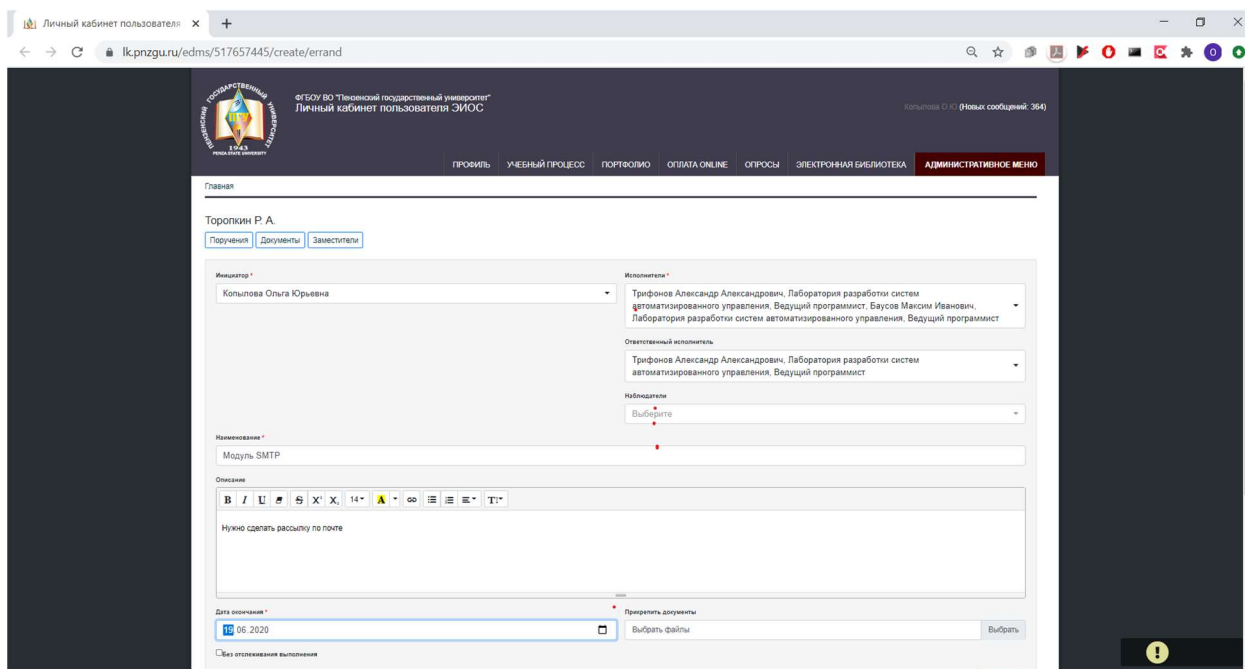


Рисунок Д.23 – Результат теста покрытия маршрутов №.1 (создание поручения)

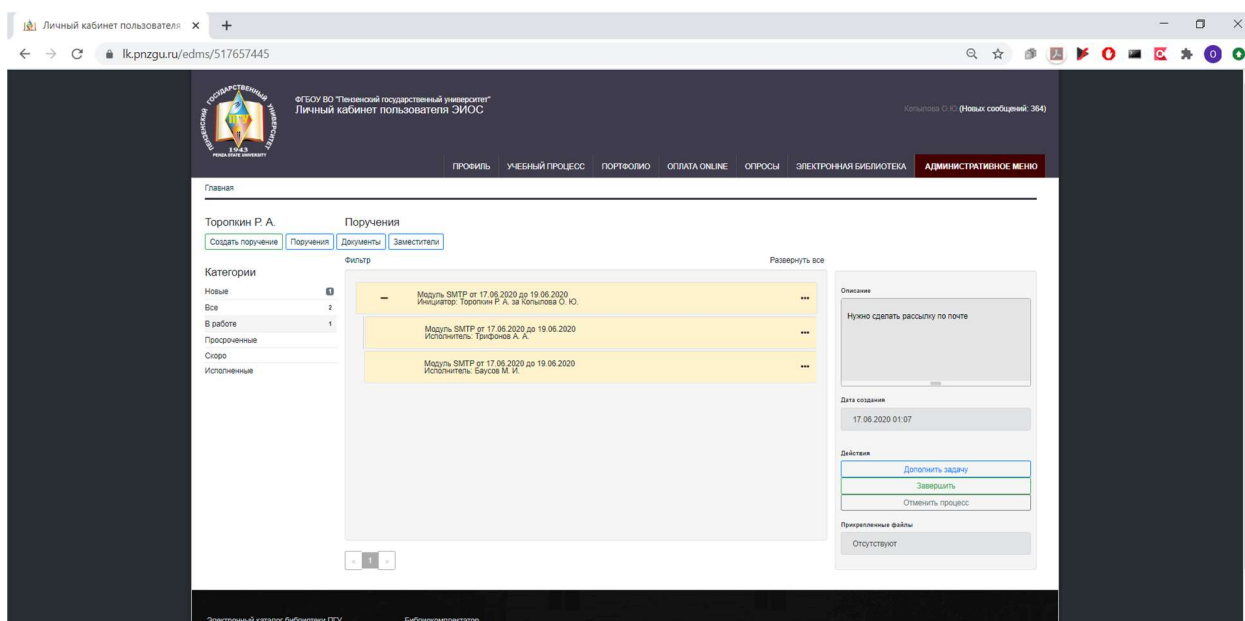


Рисунок Д.24 – Результат теста покрытия маршрутов №.1 (просмотр от имени инициатора)

Личный кабинет пользо... | Личный кабинет пользо... | +

lk.pnzgu.ru/edms/517657445/create/errand

ПРОФИЛЬ | УЧЕБНЫЙ ПРОЦЕСС | ПОРТФОЛИО | ОПЛАТА ONLINE | ОПРОСЫ | ЭЛЕКТРОННАЯ БИБЛИОТЕКА | АДМИНИСТРАТИВНОЕ МЕНЮ

Главная

Торопкин Р. А.

Поручения | Документы | Заместители

Инициатор *

Торопкин Роман Александрович

Исполнитель *

Павлов Антон Владиславович, Лаборатория интернет-технологий, Заведующий лабораторией учебной, Рензеева Ирина Александровна, Лаборатория разработки систем автоматизированного управления, Ведущий программист

Ответственный исполнитель

Павлов Антон Владиславович, Лаборатория интернет-технологий, Заведующий лабораторией учебной

Наблюдатели

Выберите

Наименование *

Тестирование мобильного приложения

Описание

В | I | U | S | X | 14 | A | B | T

Дата окончания *

ДД.ММ.ГГГГ

Прислать документы

Выбрать файлы

Выбрать

☐ Без отслеживания выполнения

Сохранить

Рисунок Д.25 – Результат теста покрытия маршрутов №.2 (создание поручения)

Личный кабинет пользователя | +

lk.pnzgu.ru/edms/30918842

ГОУ ВПО "Тюменский государственный университет" | Личный кабинет пользователя ЗИОС

Копылова О.Ю. (Новых сообщений: 364)

ПРОФИЛЬ | УЧЕБНЫЙ ПРОЦЕСС | ПОРТФОЛИО | ОПЛАТА ONLINE | ОПРОСЫ | ЭЛЕКТРОННАЯ БИБЛИОТЕКА | АДМИНИСТРАТИВНОЕ МЕНЮ

Главная

Павлов А. В.

Создать поручение | Поручения | Документы | Заместители

Категории

Новые

Все

В работе

Прозроченные

Скоро

Исполненные

Поручения

Файлы

Развернуть все

Инициатор	Исполнитель	Дата создания	Действие
Тестирование мобильного приложения от 17.06.2020 до 25.06.2020	Инициатор: Торопкин Р. А.	17.06.2020 01:17	Завершить
Тестирование мобильного приложения от 17.06.2020 до 25.06.2020	Исполнитель: Павлов А. В.		Получить
Тестирование мобильного приложения от 17.06.2020 до 25.06.2020	Исполнитель: Рензеева И. А.		Запросить продление

Присланные файлы

Отсутствуют

Рисунок Д.26 – Результат теста покрытия маршрутов №.2 (просмотр от имени исполнителя)

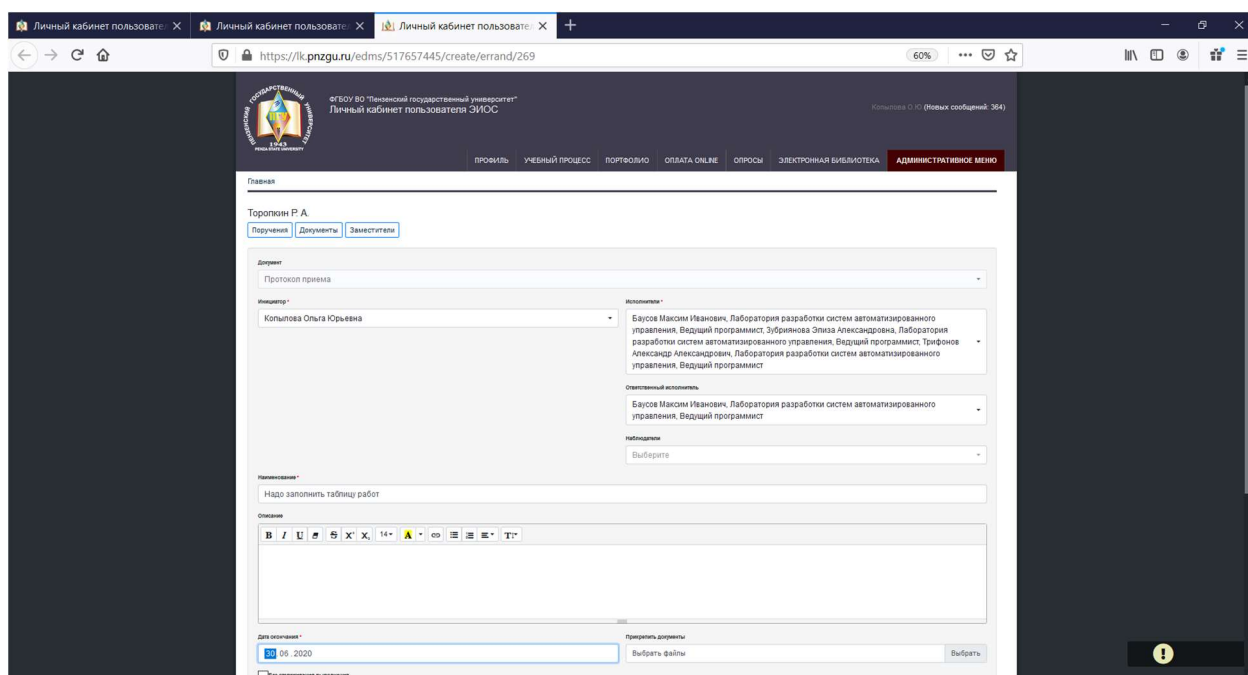


Рисунок Д.27 – Результат теста покрытия маршрутов №.3 (создание поручения)

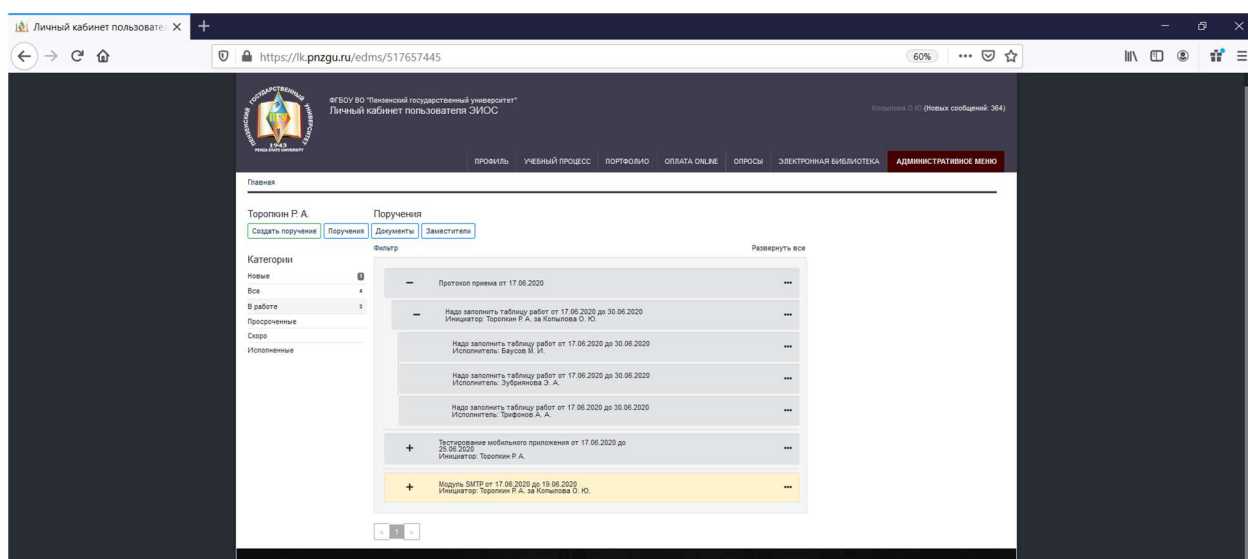


Рисунок Д.28 – Результат теста покрытия маршрутов №.3 (просмотр от имени инициатора)

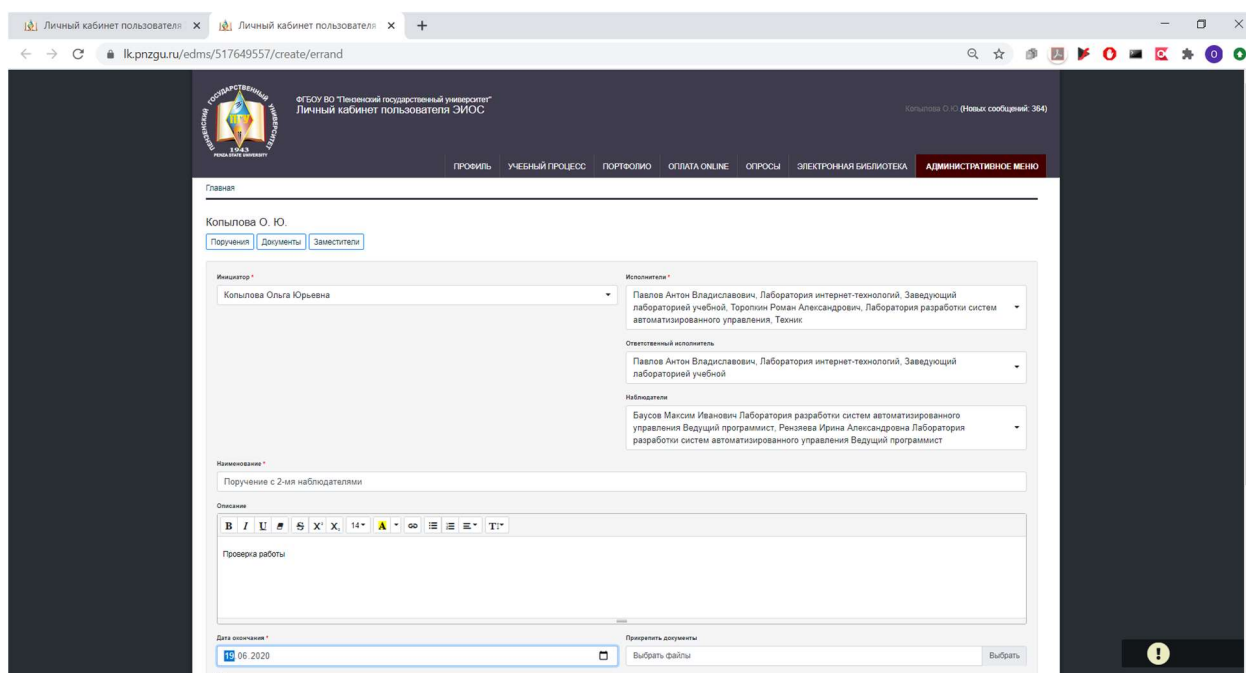


Рисунок Д.29 – Результат теста покрытия маршрутов №.4 (создание поручения)

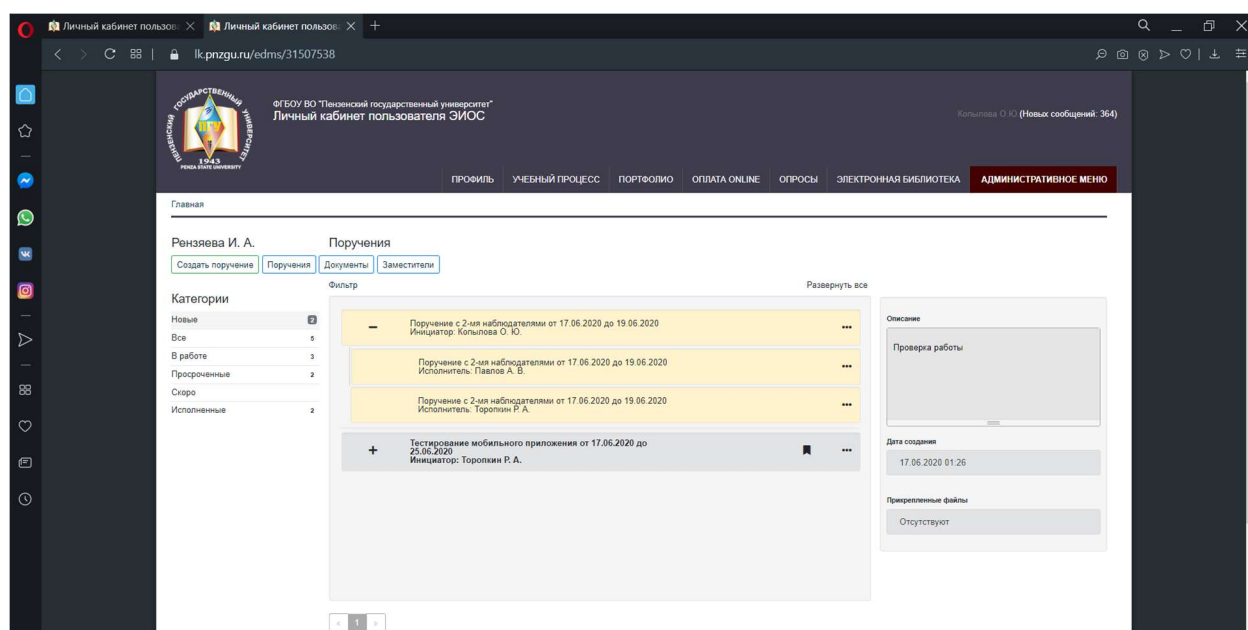


Рисунок Д.30 – Результат теста покрытия маршрутов №.4 (просмотр от имени наблюдателя)

Приложение Е
БЛОК-СХЕМА МЕТОДА «СОЗДАТЬ ПОРУЧЕНИЕ»
(рекомендуемое)

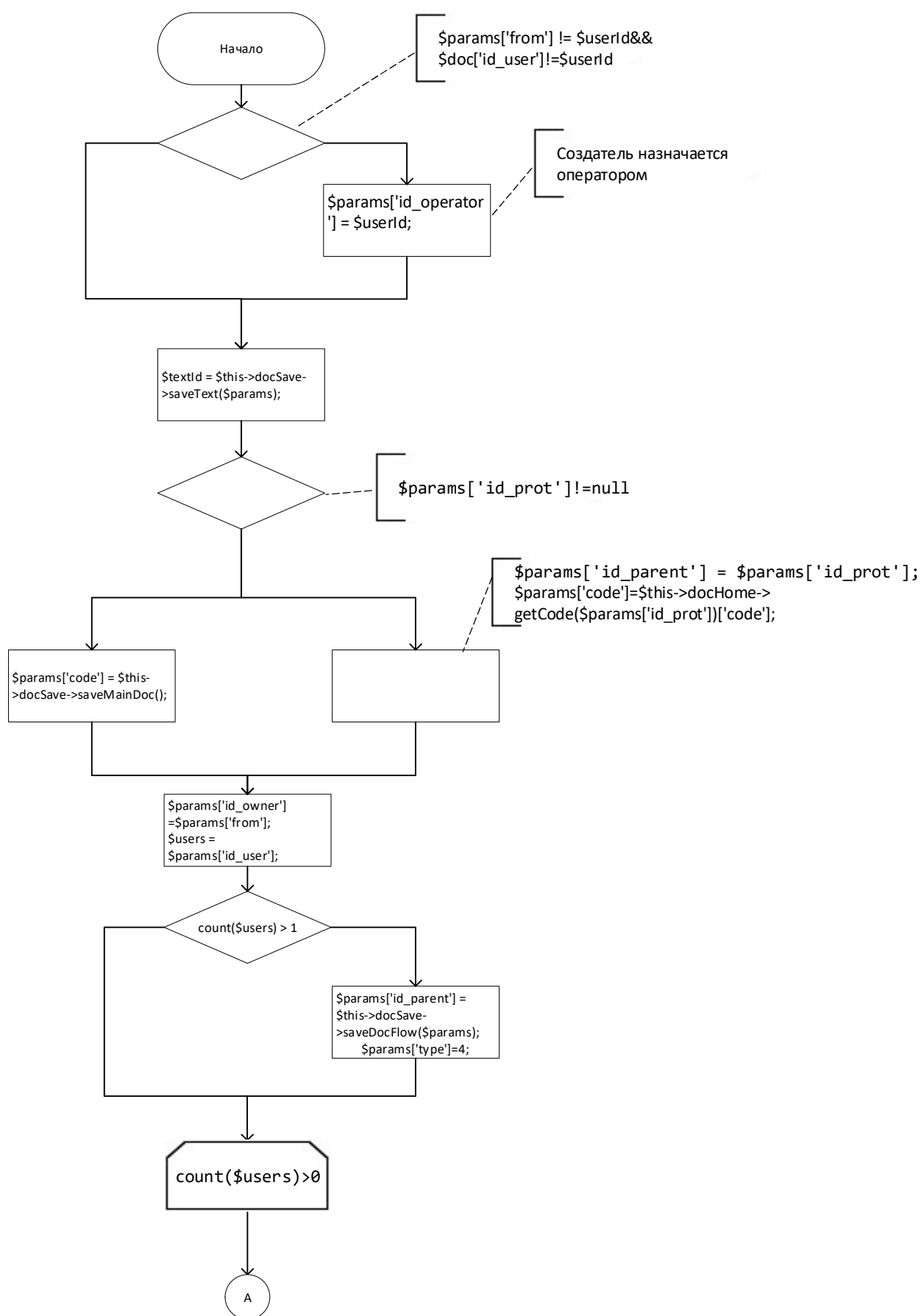


Рисунок Е.1 – Блок-схема (часть 1)

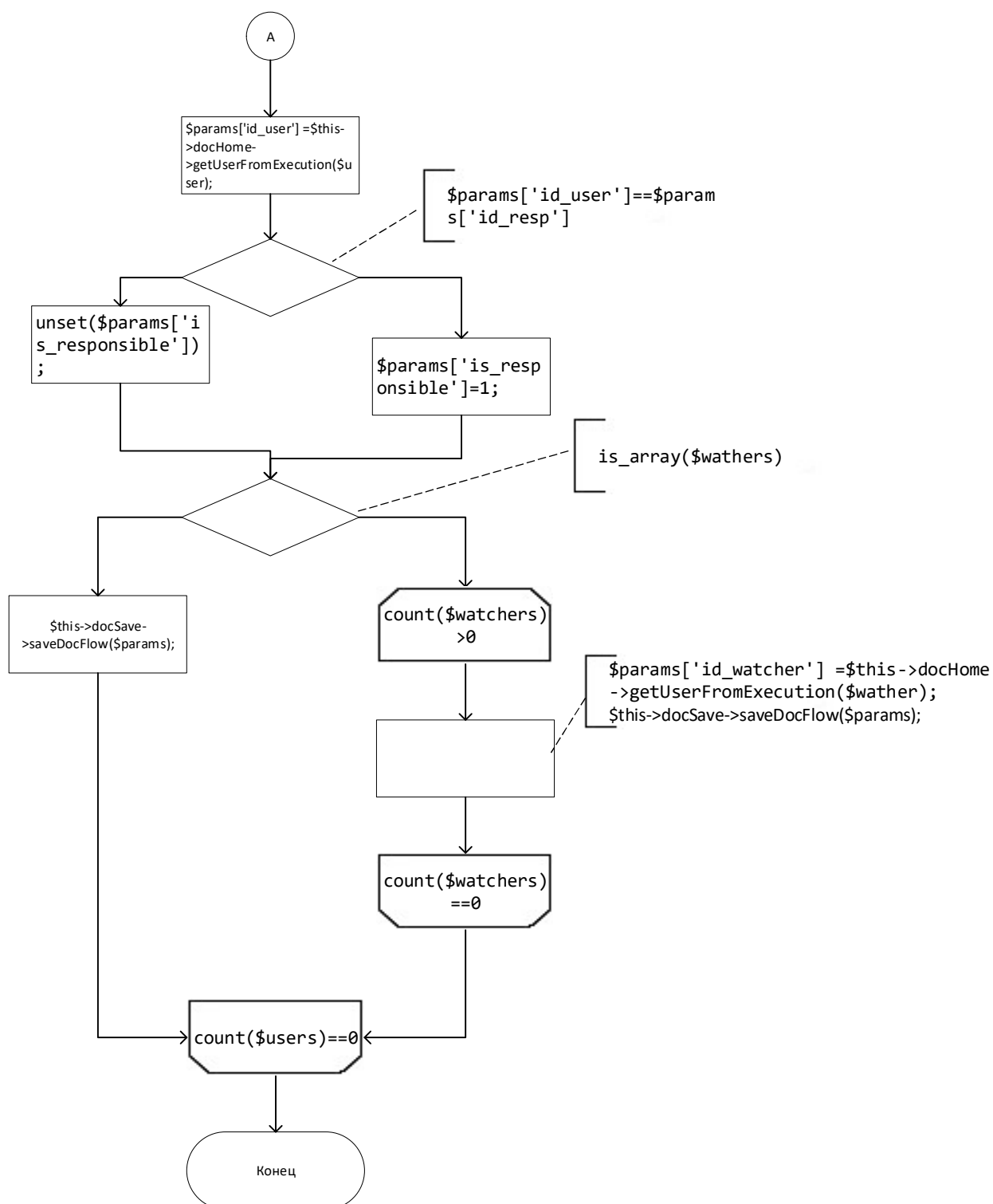


Рисунок Е.2 – Блок-схема (часть 2)

Приложение Ж
ГРАФ ПОТОКА ДАННЫХ
(рекомендуемое)

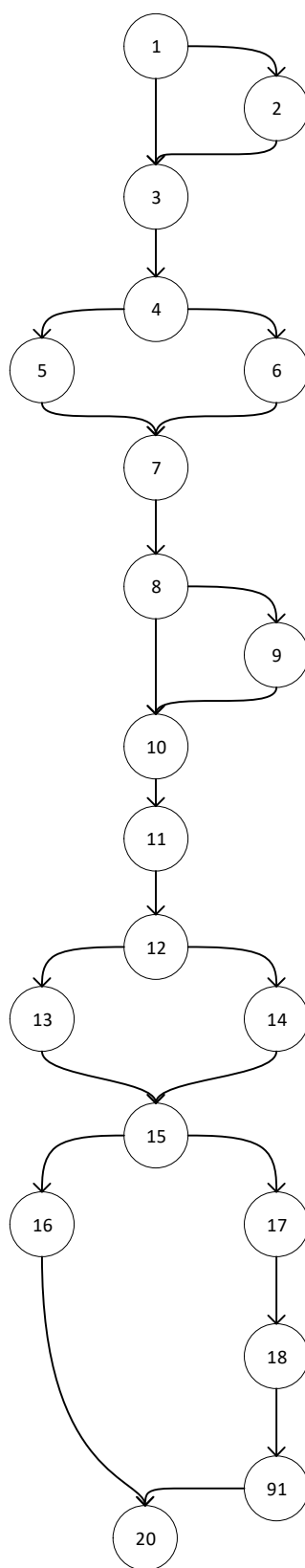


Рисунок Ж.1 – Граф потока данных

Приложение И
АКТ О ВНЕДРЕНИИ
(рекомендуемое)



МИНОБРАЗОВАНИЯ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Пензенский государственный университет»
(ФГБОУ ВО «ПГУ»)

ул. Красная, д. 40, г. Пенза, Россия, 440026
Тел/факс: (841-2) 56-51-22, e-mail: cnit@pnzgu.ru, <http://www.pnzgu.ru>
ОКПО 02069042, ОГРН 1025801440620, ИНН/КПП 5837003736/583701001

№

Г

Г

Г

Г

АКТ

о внедрении web-приложения для контроля исполнения поручений в
системе электронного документооборота вуза

Настоящий акт свидетельствует о том, что «Web-приложение для
контроля исполнения поручений в системе электронного документооборота
вуза» автоматизированной системы ЭИОС ПГУ, разработанная Копыловой
Ольгой Юрьевной, внедрена в работу ФГБОУ ВО «Пензенский
государственный университет». Внедрение происходило с 12.05.20 по
22.05.20.

В ходе тестовой эксплуатации подсистемы ЭИОС ПГУ подтверждено,
что она соответствует требованиям технического задания и позволяет
осуществлять контроль исполнения поручений сотрудниками ПГУ.

Проректор по цифровизации

И.Г. Кревский

Личную подпись И.Г. Кревский
ЗАВЕРЯЮ
Специалист по кадрам
«___» _____ 20__ г.



Рисунок И.1 – Акт о внедрении web-приложения