Изображение выглядит как Цвет электрик

Автоматически созданное описание

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

(ДВФУ)

Институт математики и компьютерных технологий

Департамент программной инженерии и искусственного интеллекта

Решетнев Никита Ярославович

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Бакалаврская работа\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

вид ВКР

**РАЗРАБОТКА ОБРАЗОВАТЕЛЬНОЙ СИСТЕМЫ ДЛЯ ДАЛЬНЕВОСТОЧНОГО ЦЕНТРА МАТЕМАТИКИ. ПОДСИСТЕМА ДОКУМЕНТООБОРОТ**

по направлению подготовки (специальности) 09.03.04прогин

Программная инженерия

Владивосток

2024

Оглавление

[Введение 4](#_Toc169832477)

[1 Обзор существующих программных средств 7](#_Toc169832478)

[1.1 Общий функционал 7](#_Toc169832479)

[1.1.1 Составление договоров 7](#_Toc169832480)

[1.1.2 Отправка напоминаний об оплате 10](#_Toc169832481)

[1.2 Отличия платформы Мой класс 11](#_Toc169832482)

[1.3 Отличия платформы КонсультантПлюс 12](#_Toc169832483)

[1.4 Отличия платформы FreshDoc 12](#_Toc169832484)

[1.5 Отличие платформы DocZilla 13](#_Toc169832485)

[1.6 Отличие платформы 1С: Документооборот 13](#_Toc169832486)

[1.7 Сравнительный анализ 14](#_Toc169832487)

[1.8 Вывод 16](#_Toc169832488)

[2 Анализ и построение модели предметной области «Документооборот» 17](#_Toc169832489)

[2.1 Анализ предметной области «Документооборот» 17](#_Toc169832490)

[2.2 Объекты предметной области 17](#_Toc169832491)

[2.3 Задачи предметной области 21](#_Toc169832492)

[2.4 Методы решения задач 22](#_Toc169832493)

[2.5 Модель онтологии предметной области 25](#_Toc169832494)

[2.5.1 Определение вспомогательных терминов 25](#_Toc169832495)

[2.5.2 Модули онтологии 26](#_Toc169832496)

[3 Проектирование системы «Документооборот» 29](#_Toc169832497)

[3.1 Архитектурно-контекстная диаграмма системы 29](#_Toc169832498)

[3.2 Спецификация требований к системе 30](#_Toc169832499)

[3.2.1 Общее описание требований к системе 30](#_Toc169832500)

[3.2.2 Функциональные требования 31](#_Toc169832501)

[3.2.3 Нефункциональные требования 31](#_Toc169832502)

[3.2.4 Требования к удобству использования системы 31](#_Toc169832503)

[3.2.5 Требование к мобильности системы 32](#_Toc169832504)

[3.3 Критерии подтверждения 32](#_Toc169832505)

[3.4 Use Case 32](#_Toc169832506)

[3.4.1 Сценарии использования 34](#_Toc169832507)

[3.4.2 User Stories 37](#_Toc169832508)

[3.5 Проект верхнего уровня 39](#_Toc169832509)

[3.6 Диаграмма потоков данных 42](#_Toc169832510)

[3.7 Диаграмма состояний 44](#_Toc169832511)

[3.8 Проект базы данных 46](#_Toc169832512)

[3.8.1 Таблица users\_user 47](#_Toc169832513)

[3.8.2 Таблица contracts\_address 47](#_Toc169832514)

[3.8.3 Таблица university\_discipline 48](#_Toc169832515)

[3.8.4 Таблица university\_studygroup 49](#_Toc169832516)

[3.8.5 Таблица customers\_customer 50](#_Toc169832517)

[3.8.6 Таблица university\_student 51](#_Toc169832518)

[3.8.7 Таблица contracts\_contract 52](#_Toc169832519)

[3.9 Проект интерфейса программного средства 54](#_Toc169832520)

[3.9.1 Экран входа 54](#_Toc169832521)

[3.9.2 Экран договоров 54](#_Toc169832522)

[3.9.3 Экран просмотра заказчиков 58](#_Toc169832523)

[3.9.4 Экран просмотра учеников 60](#_Toc169832524)

[3.9.5 Экран просмотра дисциплин 62](#_Toc169832525)

[3.9.6 Экран просмотра учебных групп 62](#_Toc169832526)

[4 Реализация программного средства 64](#_Toc169832527)

[4.1 Инструменты разработки 64](#_Toc169832528)

[4.1.1 Клиентское приложение 64](#_Toc169832529)

[4.1.2 Серверное приложение 66](#_Toc169832530)

[4.1.3 Инфраструктура 67](#_Toc169832531)

[4.2 Тестирование 67](#_Toc169832532)

[4.2.1 Структурное тестирование 67](#_Toc169832533)

[4.2.2 Тестирование для подтверждения программной системы 70](#_Toc169832534)

[4.2.3 Набор тестов для проверки на всех тестовых ситуациях 74](#_Toc169832535)

[4.2.4 Нагрузочное тестирование 81](#_Toc169832536)

[4.2.5 Результаты тестирования 84](#_Toc169832537)

[Заключение 85](#_Toc169832538)

[Список литературы 86](#_Toc169832539)

# Введение

Дальневосточный центр математики предоставляет услуги по подготовке к ОГЭ и ЕГЭ по математике, информатике и физике, поступлению в ВУЗ, осуществляет подготовку к математическим олимпиадам. [1]

В центре математики проводятся занятия по математике для школьников, учеников и всех интересующихся, а также курсы по повышению квалификации для учителей. Дальневосточный центр математики проводит летние математические лагеря для детей. В центре математики проводятся семинары. [1]

В центр приходит много желающих с различными учебными целями такими как: подготовка к ЕГЭ и ОГЭ, подготовка к математическим олимпиадам, углубленное изучение школьной математики и др. [1]

Каждый ученик дальневосточного центра математики заключает договор с центром об оказании образовательных услуг. В договоре указывается подробная информация о курсе, времени прохождения, количеству занятий, цене за курс и прочие данные обучающегося.

Данный договор заполняется администратором вручную, а ученику выдается его бумажная копия. Данные о размере выплат, которые заказчик должен ежемесячно вносить в качестве оплаты образовательных услуг, вносятся в приложение «Мой класс» и доступны только администратору. По этой причине администратор вынужден самостоятельно напоминать заказчику об оплате.

Сейчас в Дальневосточном центре математики обучается около 180 обучающихся, при этом с каждым годом наблюдается прирост количества обучающихся. Поскольку количество учеников растёт, то администратору приходится составлять большое количество договоров на оказание услуг вручную, а также самостоятельно напоминать заказчику об оплате. Как следствие происходят такие ситуации как неправильное заполнение договора как со стороны администратора, так и со стороны обучающегося, кроме того, администратор может забыть напомнить заказчику об оплате, как следствие происходит просрочка платежа, что также негативно сказывается на работе администратора.

Заказчиком для выпускной квалификационной работы стал Дальневосточный центр математики в лице Первухина Михаила Александровича - директора центра. Требованием заказчика является разработка системы, которая бы предоставляла пользователям (администратору, ученику и родителю) решение проблем с договорами, тестированиями и с электронным дневником.

Разрабатываемое решение представляет из себя совокупность трех подсистем: подсистемы «Дневник», подсистемы «Договоры» и подсистемы «Адаптивное тестирование». В данной работе представлено описание подсистемы «Договоры».

В рамках выпускной квалификационной работы в качестве инструмента устранения проблемы составления договоров, а также напоминаний об оплате была поставлена цель - реализация системы автоматического составления договоров с напоминаниями об оплате.

Данная система будет решать проблему ручного составления договоров, а также ручного напоминания об оплате образовательных услуг учеников Дальневосточного центра математики. Это позволит уменьшить нагрузку на администратора, облегчить процедуру переоформления договоров и запись старых учеников на новые курсы, избегать ошибок при составлении договоров между учеником и Дальневосточным центром математики, а также позволит ученикам не забывать вносить оплату за обучение, получать новые интересующие их предложения.

Для решения проблемы необходимо решить следующие задачи:

1. проанализировать существующие программные системы для составления договоров с напоминаниями об оплате;
2. выполнить анализ предметной области составления договоров с напоминаниями об оплате, построить её модель, определить задачи, которые должна решать система по созданию договоров обучающихся в Дальневосточном центре математики;
3. разработать проект системы по составлению договоров с напоминаниями об оплате;
4. реализовать прототип системы по составлению договоров с напоминаниями об оплате обучающихся в Дальневосточном центре математики и провести его тестирование.

В первой главе проведён обзор существующих программных средств, произведён выбор оптимального решения.

Во второй выполнен анализ предметной области: были рассмотрены объекты и задачи, построена модель, определены методы решения задачи.

В третьей главе описывается проект создаваемой подсистемы «Договоры».

В четвертой описана реализация программной системы, а также приведены результаты тестирования.

В заключении приведены итоговые результаты выпускной квалификационной работы.

# 1 Обзор существующих программных средств

Одним из популярных средств для составления договоров на территории Российской Федерации являются такие платформы, как FreshDoc, DocZila, 1C: Документооборот, КонсультантПлюс, Мой Класс. Рассмотрим их общий функционал и отличительные особенности составления договоров внутри каждой системы.

## 1.1 Общий функционал

Каждая система включает в себя составление договоров, отправку напоминаний об оплате.

### 1.1.1 **Составление договоров**

Договором признается соглашение двух или нескольких лиц об установлении, изменении или прекращении гражданских прав и обязанностей.[2]

К договорам применяются правила о двух- и многосторонних сделках, предусмотренные главой 9 Гражданского Кодекса РФ, если иное не установлено Гражданским Кодексом.[2]

К обязательствам, возникшим из договора, применяются общие положения об обязательствах (статьи 307 - 419), если иное не предусмотрено правилами настоящей главы и правилами об отдельных видах договоров, содержащимися в Гражданском кодексе РФ.[2]

К договорам, заключаемым более чем двумя сторонами, общие положения о договоре применяются, если это не противоречит многостороннему характеру таких договоров.[2]

На рисунке 1 представлен пример составления договоров на платформе Мой класс. Администратор создаёт шаблон документа, после чего использует его для составления договоров для учащихся. Данные о сторонах договора берутся из базы данных учеников.

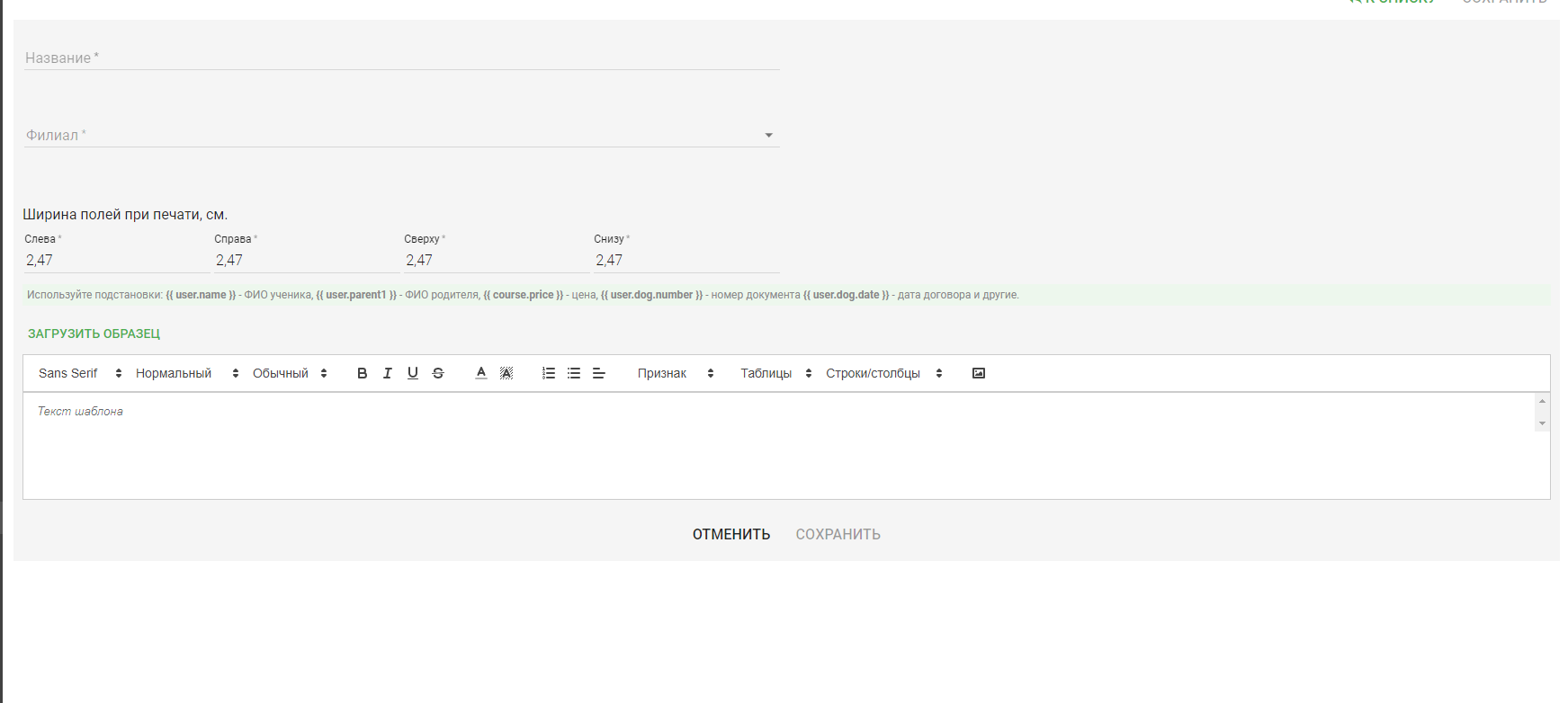


Рисунок 1 – Составление договоров на платформе Мой класс

На рисунке 2 представлен пример составления договоров на платформе КонсультантПлюс. Пользователь заполняет все данные, требуемые в договоре, после чего на выходе получается готовый договор.

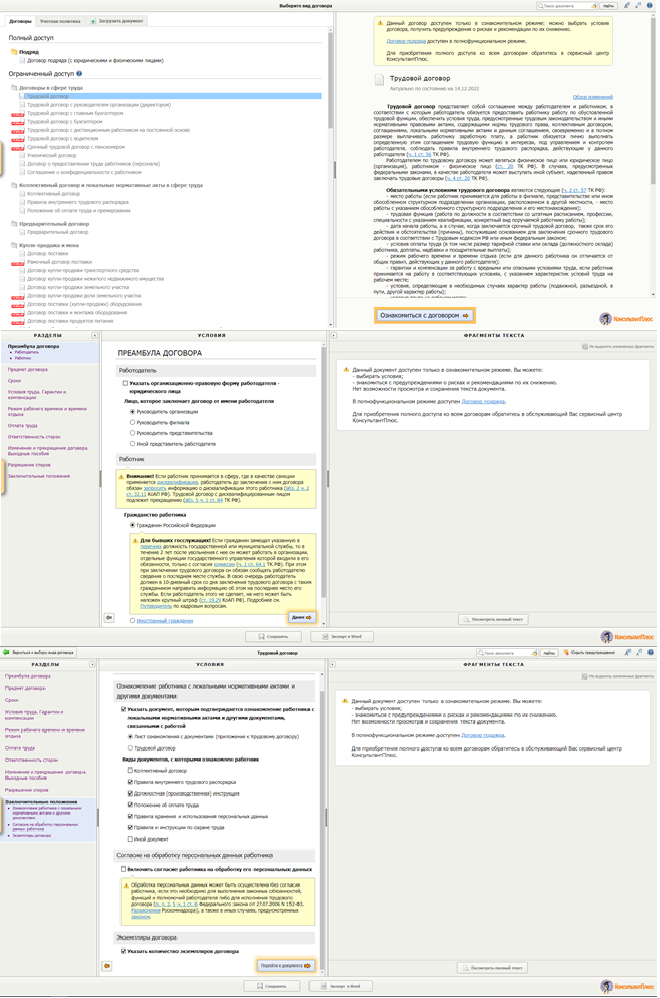


Рисунок 2 – Составления договоров на платформе КонсультантПлюс

На рисунке 3 представлен пример составления договоров на платформе FreshDoc. Пользователь заполняет новые данные, требуемые в договоре, либо берёт из базы уже заполняемые ранее данные о сторонах договора.

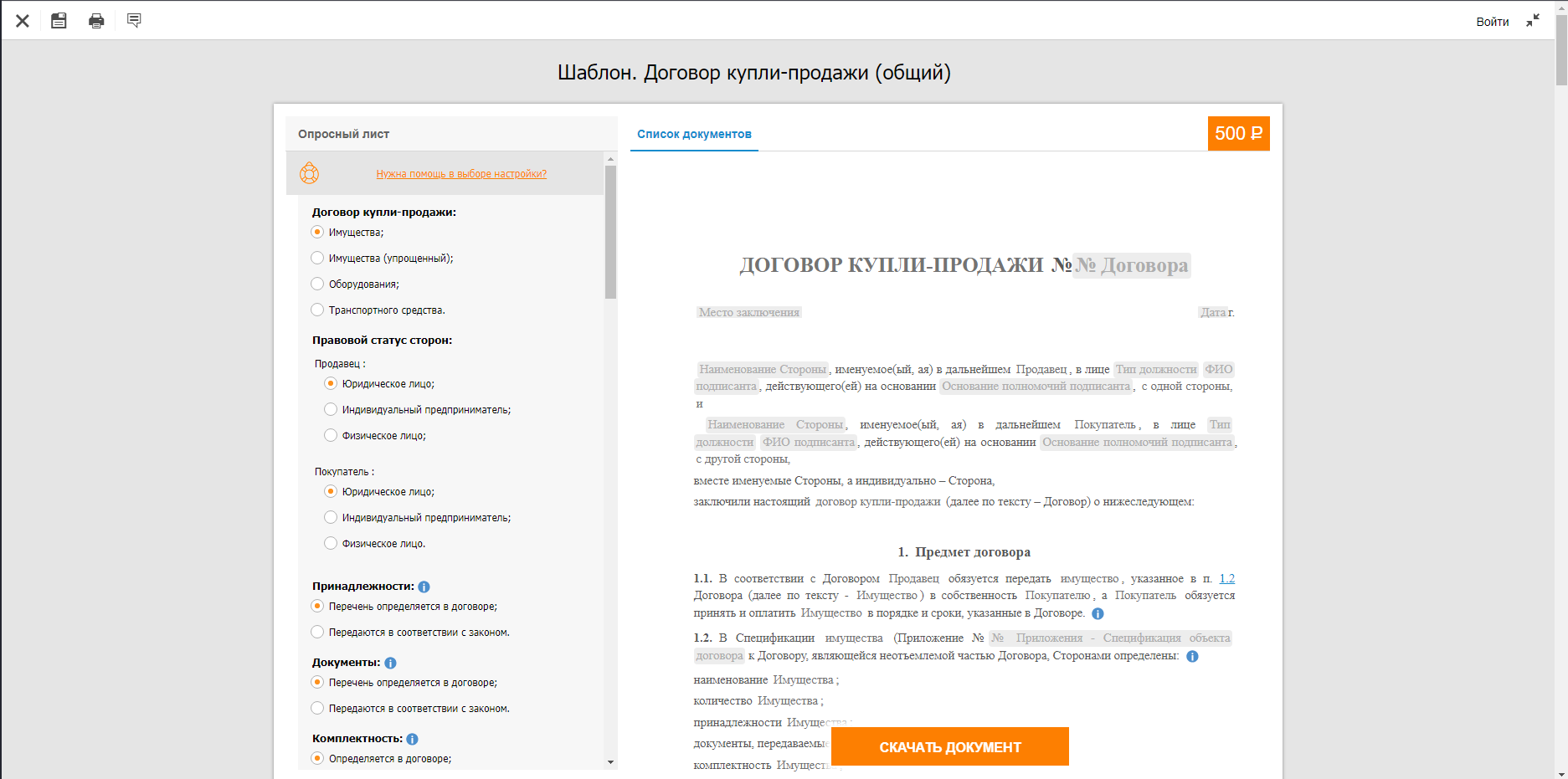


Рисунок 3 – Составления договоров на платформе FreshDoc

На рисунке 4 представлен пример составления договоров на платформе DocZilla. Пользователь заполняет новые данные, требуемые в договоре, либо берёт из базы уже заполняемые ранее данные о сторонах договора.

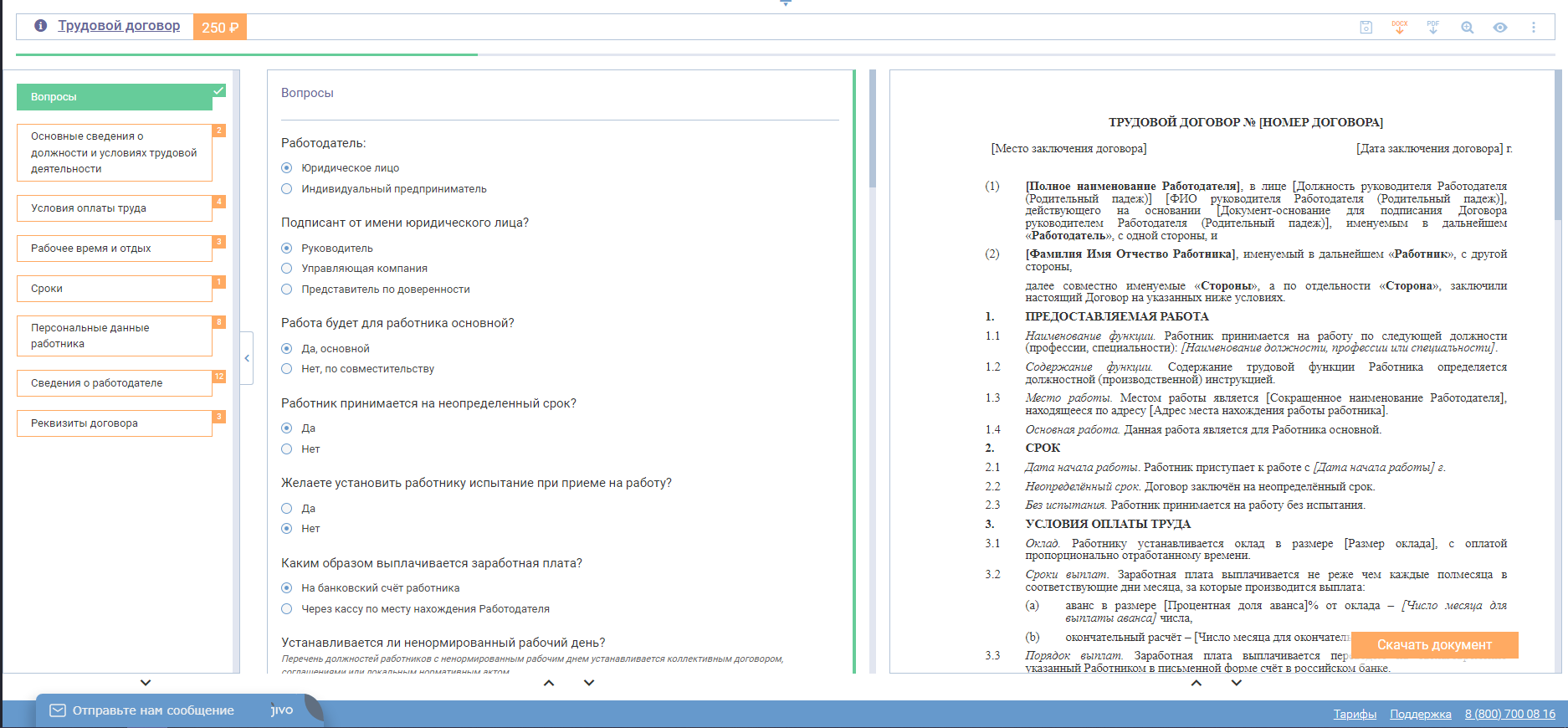


Рисунок 4 – Составления договоров на платформе DocZilla

На рисунке 5 представлен пример составления договоров на платформе 1С: Документооборот. Данные о сторонах договора берутся из базы данных учеников.

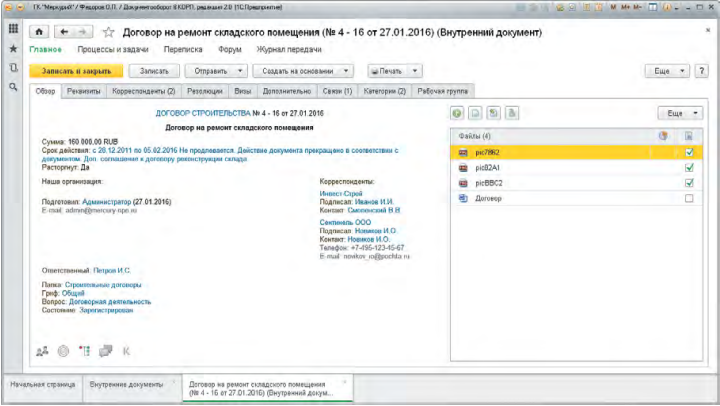


Рисунок 5 – Составления договоров на платформе 1С: Документооборот

### 1.1.2 Отправка напоминаний об оплате



Рисунок 6 – Отправка напоминания об оплате на платформе Мой класс

Уведомление – документ, содержащий какое-либо сообщение, ставящий в известность о чем-либо. [3]

На рисунке 6 представлен пример отправки напоминания об оплате на платформе Мой класс

На рисунке 7 представлен пример отправки напоминания об оплате на платформе 1С: Документооборот

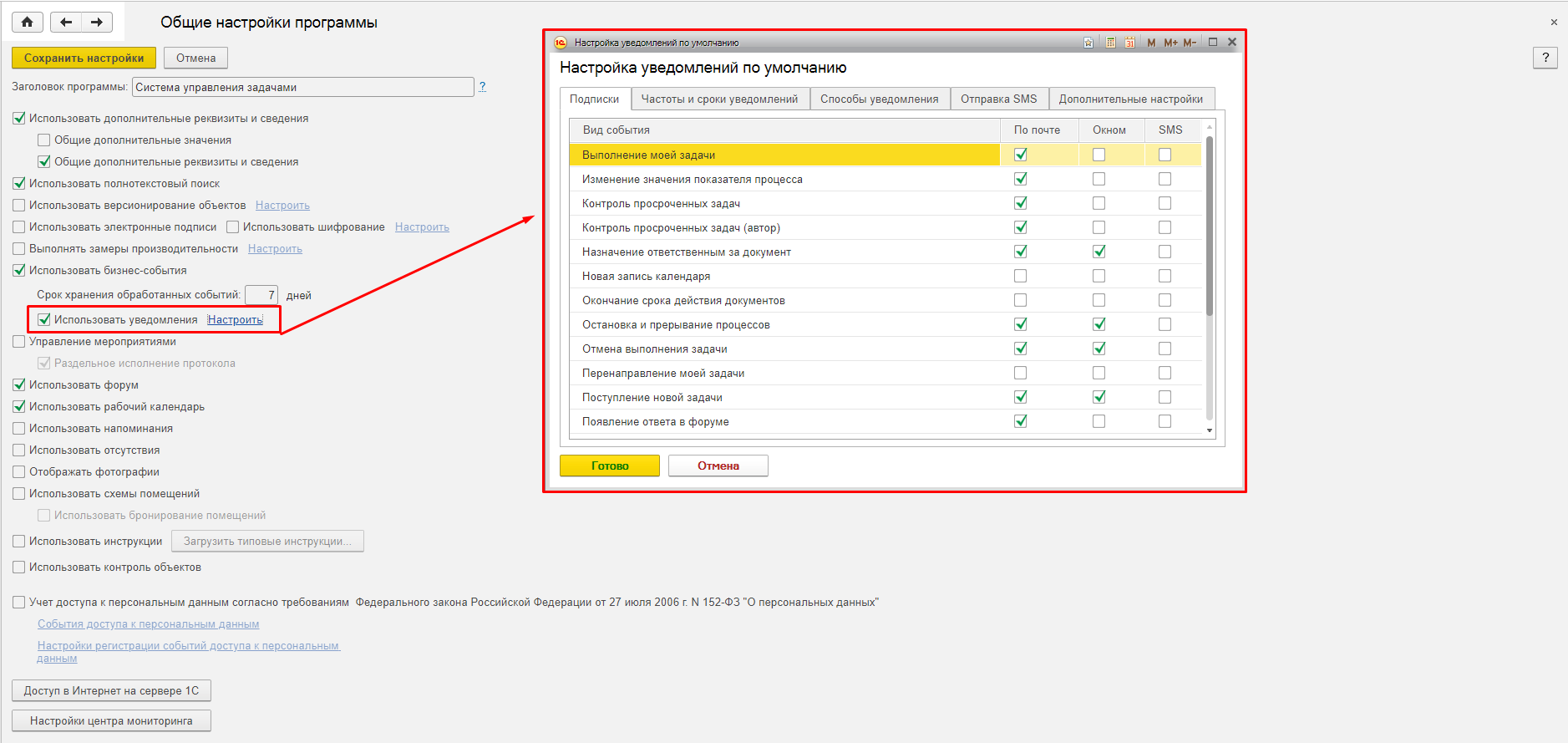


Рисунок 7 – Отправка напоминания об оплате на платформе 1C: Документооборот

## 1.2 Отличия платформы Мой класс

Отличительной чертой от других платформ является возможность подключения платформы «Мой класс» в систему образовательных учреждений, а также интеграция с сайтами. [4]

Более того, Мой класс является кроссплатформенным решением, другими словами, пользователь имеет одинаковый функционал на любого типа устройстве.

Главным минусом платформы «Мой класс» является условно-бесплатный способ распространения программного обеспечения.

Имеется малый набор шаблонов для составления договоров.

Отсутствует возможность просмотра составленных договоров.

## 1.3 Отличия платформы КонсультантПлюс

Главным отличием от других платформ является составление договора для конкретной ситуации, с юридически корректными формулировками и с учетом действующего законодательства. При изменениях законодательства сервис предупредит об этом и предложит обновить ранее сохраненный шаблон договора. [5]

Более того КонсультантПлюс предлагает более 50 шаблонов договоров и других документов организации. Также на платформе КонсультантПлюс по каждому договору есть возможность выбора условий. Это позволяет создать сотни вариантов одного и того же договора за счет включения в документ различных условий и их комбинаций. В итоге можно составить договор с учетом конкретной ситуации. [5]

Главным минусом платформы КонсультантПлюс является то, что все функции по составлению договоров являются платными.

Также минусом является полное отсутствие отправки напоминания об оплате договоров.

## 1.4 Отличия платформы FreshDoc

Главным плюсом платформы FreshDoc является доступ к документам по ссылке и их совместное использование: обработка, согласование, обсуждение. [6]

Более того FreshDoc умеет автоматически вести базу для заполнения документов. Достаточно однажды ввести данные для составления договоров, после этого эти данные можно будет использовать для составления других договоров.

Одним из плюсов является поддержка специалистов при составлении договоров и его дальнейшей поддержке.

Главным минусом платформы FreshDoc является коммерческий тип программного обеспечения. Есть 3 типа подписки: плата за один документ, основной тариф, в котором отсутствует создание собственных многовариантных шаблонов и загрузка документов из Word, и тариф профессионал, в котором эта возможность присутствует.

## 1.5 Отличие платформы DocZilla

Главным отличием DocZilla от других платформ является условно-бесплатное распространение программного обеспечения. На платформе расположено множество бесплатных шаблонов для составления договоров.

Также, плюсом является то, что платформа подсвечивает все изменения, которые были внесены в договор. Это помогает отслеживать изменения в действующем договоре.

Но, минусом является то, что также многие договоры являются платными, и их использование требует платной подписки на платформу.

## 1.6 Отличие платформы 1С: Документооборот



Рисунок 8 – Схема сопровождения договора

Главным отличием платформы 1С: Документооборот от других платформ является полное сопровождение договора от подготовки проекта договора до продления/расторжения договора. Схема сопровождения договора представлена на рисунке 8. [7]

Также, огромным плюсом платформы 1С: Документооборот является хранение всех присоединенных к договору файлов в базе данных:

1. черновики;
2. согласованные материалы;
3. скан-копии подписанных оригиналов;
4. вспомогательные файлы. [7]

Ещё один немаловажный плюс – учет этапов договора. Для каждого договора можно указать, какие этапы обработки он должен пройти и для каждого из этапов можно установить плановый срок прохождения. Отсюда по мере прохождения этапы вручную отмечаются как пройденные. Таким образом, для договора видно, на каком этапе он находится [7]

Также программа автоматически уведомляет ответственного за договор о приближении сроков этапов или просрочке [7]

Из минусов можно выделить дорогостоящую настройку программного обеспечения и обслуживание программного обеспечения.

## 1.7 Сравнительный анализ

Результаты сравнения рассмотренных платформ для составления договоров приведены в таблице 1

Таблица 1 – сравнение платформ для составления договоров

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Платформа  Функции | Мой класс | КонсультантПлюс | FreshDoc | DockZilla | 1С: Документооборот |
| Тип программного обеспечения | Условно бесплатный | Коммерческий | Коммерческий | Условно бесплатный | Коммерческий |
| Создание своих шаблонов договоров | + | - | - | - | + |
| База шаблонов договоров | - | + | + | + | + |
| Поддержка отправки уведомлений | + | - | - | - | + |
| Отслеживание изменений в договоре | - | - | - | + | + |

## 1.8 Вывод

Существуют много систем, позволяющих создавать договоры, однако, как следует из обзора существующих систем составления договоров, нет ни одной системы, сочетающих в себе бесплатный тип программного обеспечения, огромную базу шаблонов для договоров и систему отправки напоминаний об оплате. Вследствие чего, наиболее рациональным является решение о создании собственной системы составления договоров, в которую будут включены вышеописанные пункты.

# 2 Анализ и построение модели предметной области «Документооборот»

В данной главе выделяются основные понятия и информационные объекты предметной области, описывается модель предметной области.

## 2.1 Анализ предметной области «Документооборот»

Из представленного введения можно сделать вывод, что количество учеников в Дальневосточном центре математики значительно превышает количество администраторов, что затрудняет заполнение множества договоров и отслеживание оплат учеников. В данном случае система документооборота может стать эффективным инструментом минимизирования ошибок при заполнении разного рода документов и для ускорения работы администраторов при заполнении.

Ведение документооборота подразумевает процесс, включающий в себя составление и согласование, прием и передачу, отправку документов. Поэтому в узком смысле определение документооборота возможно как движение бумажных или электронных носителей от момента составления до получения и отправки на хранение в архив. Правильная организация этого процесса помогает обеспечить коммуникацию и информирование, необходима для формирования достоверной отчетности. [8]

Как следствие, можно сделать вывод о том, что создание системы документооборота является наиболее эффективным способом снижения нагрузки администраторов и минимизирования ошибок при заполнении.

## 2.2 Объекты предметной области

В рамках данной предметной области можно выделить следующие объекты:

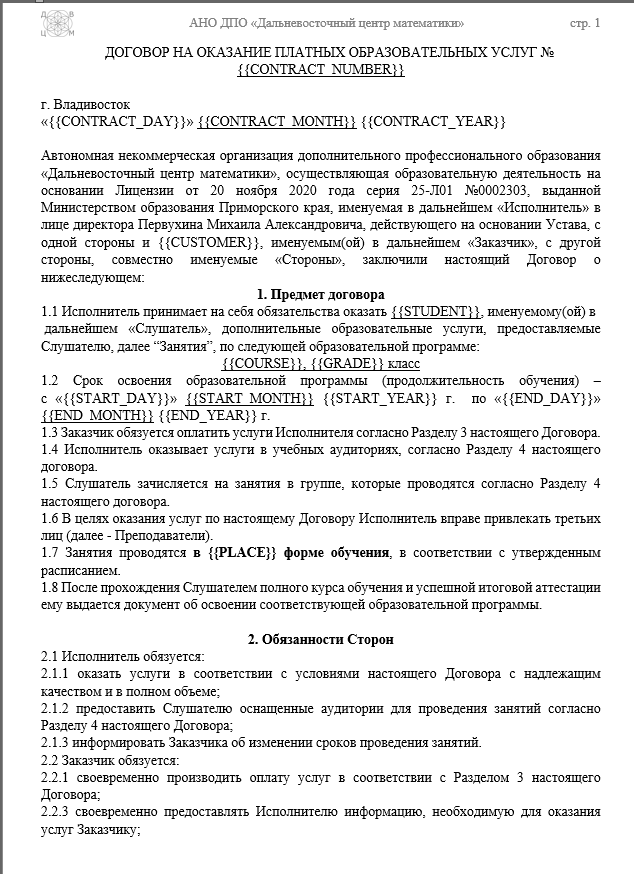


Рисунок 9 – Пример договора

1. договор – соглашение между собой двух или более сторон (субъектов), по какому-либо вопросу с целью установления, изменения или прекращения правовых отношений. Пример договора представлен на рисунке 9. Имеет следующие атрибуты:
2. место – место, где проводятся занятия в рамках определенного договора;
3. реквизиты заказчика – реквизиты заказчика, на которого заключается договор;
4. реквизиты ученика – реквизиты ученика, на которого заключается договор;
5. номер договора – уникальный номер договора. Используется для идентификации принадлежности договора к определенному ученику;
6. учебная группы – учебная группы, в которой будет проходить обучение в Дальневосточном центре математики;
7. сумма договора – итоговая сумма, которую ученик должен заплатить за своё обучение;
8. даты начала и конца договора – время действия договора.
9. программа обучения – созданный в рамках системы обучения документ, определяющий содержание и количество знаний, умений и навыков, предназначенных к обязательному усвоению по той или иной учебной дисциплине, распределение их по темам, разделам и периодам обучения.
10. название программы обучения;
11. количество занятий – числовое значение, находящее в диапазоне [1, +);
12. время занятия – время, в которое проводится одно занятие;
13. стоимость обучения – числовое значение, находящее в диапазоне [1, +);
14. классы, на которые рассчитана программа обучения.

Пример программы обучения представлен на рисунке 10

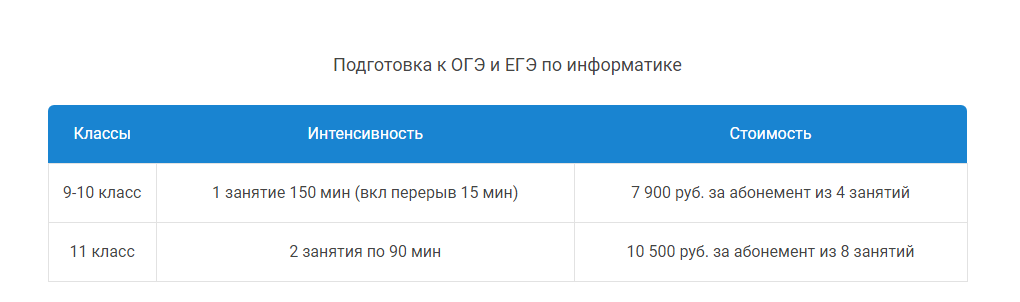


Рисунок 10 – Пример программы обучения

1. обучающийся – пользователь, зарегистрированный во внутренней системе Дальневосточного центра математики, имеет следующие атрибуты:
   1. ФИО – структура, состоящая из трёх строковых полей (фамилия, имя и отчество соответственно);
   2. номер телефона – номер телефона обучающегося;
2. заказчик – человек, который заключает договор с Дальневосточным центром математики, имеет следующие атрибуты:
   1. ФИО – структура, состоящая из трёх строковых полей (фамилия, имя и отчество соответственно);
   2. номер телефона – номер телефона ученика;
   3. email – электронная почта заказчика;
   4. паспортные данные – данные из паспорта заказчика, требуются для составления договора;
3. учебная группа – группа, в которой учатся ученики в Дальневосточном центре математики, имеет следующие атрибуты:
   1. название учебной группы;
   2. дата начала занятий – дата начала занятий определенной группы по определенной программе обучения
   3. дата конца занятий – дата конца занятий определенной группы по определенной программе обучения
   4. программа обучения - созданный в рамках системы обучения документ, определяющий содержание и количество знаний, умений и навыков, предназначенных к обязательному усвоению по той или иной учебной дисциплине, распределение их по темам, разделам и периодам обучения.
4. база договоров – база данных, где хранятся все договоры, которые были когда-либо составлены;
5. администратор системы – пользователь, зарегистрированный в системе Дальневосточного центра математики, с правами доступа, позволяющими создавать, удалять и редактировать информацию об учениках, просматривать данные о программах обучения, имеет следующие атрибуты:
   1. ФИО – структура, состоящая из трёх строковых полей (фамилия, имя и отчество соответственно);
   2. список договоров – список, состоящий из строк;
6. уведомление об оплате – короткое сообщение, призванное уведомить клиента об оплате договора. Имеет следующие атрибуты:
7. сумма оплаты – сумма, которая была выставлена администратором по договору;
8. почта заказчика – почта, на которую будет приходить уведомления об оплате;
9. тип уведомления – тип уведомления, который выбирает администратор для оплаты.

## 2.3 Задачи предметной области

В рамках данной предметной области требуется решить следующие задачи:

* + 1. Составление договоров вне зависимости от возраста обучающегося. Система должна составлять договоры вне зависимости от возраста обучающегося и отличать, когда договор составлен на обучающегося, а когда – на его родителей;
    2. Отправка уведомлений об оплате вне зависимости от типа устройства. Обучающийся или его родители должны получать уведомления на свои устройства вне зависимости от операционной системы и года выпуска этого устройства. Уведомление также должно приходить в сроки, установленные Дальневосточным центром математики;
    3. Упрощение процесса заполнения договоров для администратора. Система должна фиксировать процесс заполнения данных для договора, а также должна предоставлять механизмы для проверки правильности заполнения договоров. Например, может быть реализована проверка обязательных полей, форматов данных, логических связей и других условий. Если администратор заполнил неправильно определенное поле, то система должна выдать ошибку;

## 2.4 Методы решения задач

Для решения задач, которые нужны для эффективного документооборота предложены следующие варианты:

1. Варианты решения задачи составления договоров вне зависимости от возраста обучающегося:
2. Определение типа договора: В системе можно предусмотреть возможность выбора типа договора, который отражает, на кого именно будет составлен договор - на обучающегося или на его родителей. Это может быть реализовано путем добавления соответствующего поля или опции при создании нового договора;
3. Информационное сопровождение: во время заполнения договора система может предоставлять информационное сопровождение и указывать на необходимость заполнения определенных полей в зависимости от выбранного типа договора. Например, если договор составляется на обучающегося, то система может запросить информацию о его персональных данных, возрасте и других соответствующих атрибутах;
4. Автоматическая генерация договоров: можно рассмотреть возможность автоматической генерации договоров на основе предустановленных шаблонов, учитывающих особенности разных типов договоров. Это позволит упростить и ускорить процесс составления договоров и снизить вероятность ошибок при их заполнении.
5. Варианты решения задачи отправки уведомлений об оплате вне зависимости от типа устройства:
6. Многоформатные уведомления: разработать систему, способную отправлять уведомления об оплате в различных форматах, подходящих для разных типов устройств. Например, система может отправлять уведомления в виде SMS-сообщений, электронных писем, мобильных пуш-уведомлений или мгновенных сообщений через мессенджеры;
7. Адаптивный дизайн: обеспечить адаптивность уведомлений к различным устройствам и операционным системам. Уведомления должны автоматически подстраиваться под размеры экранов и возможности устройств, чтобы гарантировать их удобочитаемость и функциональность независимо от года выпуска или типа устройства;
8. Интеграция с различными каналами связи: разработать систему, способную интегрироваться с различными каналами связи, используемыми обучающимися и их родителями. Это может включать интеграцию с мобильными операторами для отправки SMS-сообщений, с почтовыми сервисами для отправки электронной почты, с платформами мгновенных сообщений и мессенджерами для отправки мгновенных уведомлений;
9. Система управления сроками: разработать систему управления сроками отправки уведомлений. Это позволит системе автоматически запланировать отправку уведомлений в определенное время или в определенные сроки, установленные Дальневосточным центром математики. Также можно предусмотреть механизмы повторных уведомлений или напоминаний в случае, если получатель не отреагировал на первоначальное уведомление;
10. Мониторинг и подтверждение доставки: реализовать механизмы мониторинга и подтверждения доставки уведомлений. Система должна отслеживать статус доставки каждого уведомления и предоставлять обратную связь о его успешной доставке или возникших проблемах. Это позволит контролировать процесс доставки уведомлений и принимать меры в случае неудачных попыток доставки.
11. Решение задачи упрощения процесса заполнения договоров для администратора:
12. Интуитивный пользовательский интерфейс: разработать интуитивно понятный и удобный пользовательский интерфейс, который будет направлять администратора при заполнении договоров. Интерфейс должен быть простым в использовании и логически структурированным, чтобы администратор мог легко перемещаться по полям и вносить соответствующую информацию;
13. Проверка обязательных полей: система должна проверять наличие и правильность заполнения обязательных полей. Например, если определенное поле является обязательным для заполнения, система должна выдавать ошибку или предупреждение, если администратор пропускает его или вводит некорректные данные;
14. Валидация форматов данных: реализовать механизмы валидации форматов данных для каждого поля в договоре. Система должна проверять, соответствуют ли введенные данные определенному формату (например, дата, номер телефона, электронная почта и т. д.). В случае неверного формата система должна предупреждать администратора об ошибке и требовать коррекции;
15. Логические связи и условия: если заполнение определенных полей зависит от логических связей или определенных условий, система должна обеспечивать их проверку. Например, если определенное поле требуется заполнить только при определенных условиях, система должна автоматически отображать или скрывать это поле и выдавать ошибку, если администратор нарушает логическую связь или условие;
16. Постепенное сохранение данных: система может предусмотреть механизм постепенного сохранения данных, чтобы предотвратить потерю введенной информации в случае сбоев или прерываний. это позволит администратору сохранить уже заполненные данные и продолжить заполнение договора в случае возникновения проблем;
17. Понятные сообщения об ошибках: система должна предоставлять понятные и информативные сообщения об ошибках, чтобы администратор понимал, в каком поле допущена ошибка и как ее исправить. Сообщения должны быть четкими и конкретными, чтобы ускорить процесс исправления ошибок.

## 2.5 Модель онтологии предметной области

Модель онтологии системы документооборота состоит из следующих моделей:

1. Обучающийся
2. Администратор
3. Договор
4. Программа обучения
5. Уведомление об оплате
6. Архивы документов

### 2.5.1 Определение вспомогательных терминов

день ≡ I[1; 31]

месяц ≡ I[1; 12]

год ≡ I[1900; ∞)

дата ≡ день ✕ месяц ✕ год

идентификаторы обучающихся ≡ I[0; 500000)

идентификаторы администраторов ≡ I[500000; 600000)

идентификаторы программ обучения ≡ I[600000; 700000)

идентификаторы учебных групп ≡ I[700000; 999999)

идентификаторы договоров ≡ I[1; ∞)

серии паспортов ≡ I[0; 9] ✕ I[0; 9] ✕ I[0; 9] ✕ I[0; 9]

номера паспортов ≡ I[0; 9] ✕ I[0; 9] ✕ I[0; 9] ✕ I[0; 9] ✕ I[0; 9] ✕ I[0; 9]

номера телефонов ≡ {7} ✕ {9} ✕ I[0; 9] ✕ I[0; 9] ✕ I[0; 9] ✕ I[0; 9] ✕ I[0; 9] ✕ I[0; 9] ✕ I[0; 9] ✕ I[0; 9] ✕ I[0; 9]

кем выдан паспорт ≡ {} N

прописка ≡ {} N

паспорта ≡ серии паспортов ✕ номера паспортов ✕ кем выданы паспорта ✕ дата ✕ прописка

фамилии ≡ {} N

имена ≡ {} N

отчества ≡ {} N

ФИО ≡ {} фамилии ✕ имена ✕ отчества

классы ≡ I[1; 11]

сравнение дат ≡ (λ (v: { x дата, дата }

(π (3, π(1,v)) > π (3, π(2,v))) ∨

((π (3, π(1,v)) = π (3, π(2,v))) (π (2, π(1,v)) > π (2, π(2,v)))) ∨

(((π (3, π(1,v)) = π (3, π(2,v))) ∧ (π (2, π(1,v)) = π (2, π(2,v))))

(π (1, π(1,v)) > π (1, π(2,v))))

) → {правда, ложь})

### 2.5.2 Модули онтологии

**Модуль «Заказчик»**

Идентификаторы заказчиков ≡ серии паспортов ✕ номера паспортов

**Сорт** заказчики: ({} идентификаторы заказчиков)

**Сорт** ФИО заказчика: (заказчики ФИО)

**Сорт** паспортные данные заказчика: (заказчики паспорта)

**Сорт** номер телефона: (заказчики → номера телефонов)

**Сорт** email: (заказчики → N)

**Модуль «Администратор»**

**Сорт** администраторы: ({} идентификаторы администраторов)

**Сорт** ФИО администратора: (администраторы ФИО)

**Модуль «Программа обучения»**

**Сорт** программы обучения: {} идентификаторы программ обучения

**Сорт** название программы обучения: (программы обучения → N \ ∅)

**Сорт** время занятия: (программы обучения → I[1; 90])

**Сорт** количество занятий: (программы обучения → I[1; +∞))

**Сорт** стоимость обучения: (программы обучения → I[1; ∞))

**Сорт** классы программы обучения: (программы обучения {}классы)

**Модуль «Обучающийся»**

**Сорт** обучающиеся: ({} идентификаторы обучающихся)

**Сорт** ФИО обучающегося: (обучающиеся ФИО)

**Сорт** номер телефона: (обучающиеся → номера телефонов)

**Модуль «Учебная группа»**

**Сорт** учебные группы: ({} идентификаторы учебных групп)

**Сорт** название учебной группы: (учебные группы N)

**Сорт** дата начала занятий: (учебные группы дата)

**Сорт** дата конца занятий: (учебные группы дата)

**Сорт** программа обучения: (учебные группы программа обучения)

**Ограничения:**

1. (v1: учебные группы) (v2: дата начала занятий(v1)) (v3: дата конца занятий(v1)) сравнение дат({v1, v2}) = правда

**Модуль «Договор»**

место обучения ≡ { г. Владивосток, ул. Русская, 46б, 2 этаж. , г. Владивосток, Океанский проспект, 87а.}

**Сорт** договоры: {} идентификаторы договоров

**Сорт** место: (договоры место обучения)

**Сорт** реквизиты заказчика: (договоры заказчики)

**Сорт** реквизиты ученика: (договоры ученики)

**Сорт** учебная группы: (договоры учебные группы)

**Сорт** сумма договора: (договоры I[1; ∞))

**Сорт** дата начала договора: (договоры дата)

**Сорт** дата конца договора: (договоры дата)

**Ограничения:**

1. (v1: договоры) (v2: дата начала договора(v1)) (v3: дата конца договора(v1)) сравнение дат({v1, v2}) = правда

**Модуль «Уведомление об оплате»**

типы уведомлений ≡ {уведомление о задолженности, уведомление об оплате, уведомление об остатке}

**Сорт** уведомления об оплате: {} N \ ∅

**Сорт** тип уведомления: (уведомления об оплате типы уведомления)

**Сорт** сумма оплаты: (уведомления об оплате I[1; ∞))

**Модуль «База договоров»**

**Сорт** база договоров: {} договоры

# 3 Проектирование системы «Документооборот»

Система документооборота предназначена для заключения договоров между учеником и Дальневосточным центром математики. Также предусматривается возможность уведомлений ученика об оплате договора.

## 3.1 Архитектурно-контекстная диаграмма системы

Архитектурно-контекстная диаграмма (АКД) – это графическое представление системы и внешних компонентов, взаимодействующих с системой. Эти компоненты связаны с системой через интерфейсы.

Архитектурно-контекстная диаграмма системы документооборота для Дальневосточного центра математики представлена на рисунке 13.

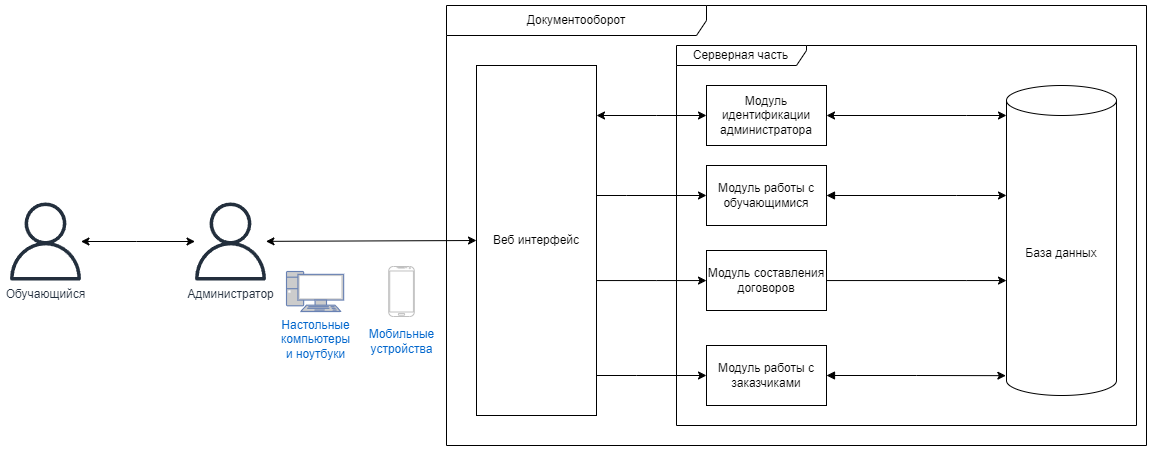


Рисунок 13 – Архитектурно-контекстная диаграмма

Взаимодействие с системой документооборота происходит посредством веб-интерфейса на устройстве пользователей, здесь же отображается вся информация, предоставляемая самой системой.

Модуль идентификации администратора отвечает за проверку наличия записей об администраторе в базе данных, а также определения, что пользователь действительно является администратором.

Модуль работы с обучающимися отвечает за следующие функции:

1. Добавления новых обучающихся в базу данных приложения: модуль просит администратора ввести данные об обучающемся, после чего заносит их в базу данных для дальнейшей работы с ними.
2. Просмотр данных обучающегося: модуль способен показать все данные о каждом обучающемся, которые могут быть использованы администратором для осведомления обучающихся.
3. Изменение данных обучающегося: модуль способен изменять данные обучающихся. Требуется для актуализации данных.

Модуль работы с заказчиками отвечает за следующие функции:

1. Добавления новых заказчиков в базу данных приложения: модуль просит администратора ввести данные заказчика, после чего заносит их в базу данных для дальнейшей работы с ними.
2. Просмотр данных о заказчике: модуль способен показать все данные о каждом заказчике, которые могут быть использованы администратором для осведомления заказчиков;
3. Изменение данных заказчика: модуль способен изменять данные заказчиков. Требуется для актуализации данных.

Модуль составления договоров предназначен для полноценной работой с договорами (создание договоров, просмотр договоров, просмотр конкретного договора, отправка уведомлений об оплате)

## 3.2 Спецификация требований к системе

К данной системе представлены следующие требования.

### 3.2.1 Общее описание требований к системе

Главная задача системы – автоматизировать документооборот в Дальневосточном центре математики.

### 3.2.2 Функциональные требования

Система должна представлять администратору следующие функции:

* Аутентификация администратора;
* Добавление обучающегося в систему;
* Просмотр списка обучающихся в системе;
* Изменение данных обучающегося в системе;
* Просмотр данных конкретного обучающегося в системе;
* Добавление заказчика в систему;
* Просмотр списка заказчиков в системе;
* Изменение данных заказчика в системе;
* Просмотр данных конкретного заказчика в системе;
* Просмотр заключенных договоров в системе;
* Создание договора с обучающимся и заказчиком;
* Просмотр конкретного договора с обучающимся и заказчиком в системе;
* Отправка уведомлений о задолженности заказчику на устройства.

### 3.2.3 Нефункциональные требования

1. Интуитивно понятный интерфейс;
2. Быстрая работа системы;
3. Надежность системы без потери данных;
4. Отказоустойчивость системы.

### 3.2.4 Требования к удобству использования системы

* Система должна быть предназначена для пользователей ПК и мобильных устройств, владеющих русским языком;
* Система должна выдавать всю информацию на русском языке;
* Система должна выдавать сообщения об ошибках, неправильно введенных данных;
* Тексты сообщений и подсказки должны быть понятными и дружелюбными;
* Запуск системы должен быть как можно более простой для пользователя.

### 3.2.5 Требование к мобильности системы

Система должна быть работать одинаково в современных браузерах – Google Chrome, Mozilla Firefox, Opera. Система должна иметь адаптивный дизайн для мобильных устройств.

## 3.3 Критерии подтверждения

Система будет считаться успешно реализованной если будут соблюдены следующие критерии:

* реализованы все требования, указанные в п. 0;
* сообщения об ошибках понятны пользователям;

## 3.4 Use Case

**Use Case** (вариант использования) — техника описания взаимодествия пользователья с системой. С помощью вариантов использования могут быть описаны требования к взаимодействию, пользовательские требования.

Взаимодействие **обучающегося** с системой состоит в том, что он может записаться на любой понравившийся ему курс.

Взаимодействие администратора с системой состоит в следующем:

* Просмотр списка составленных договоров
* Создание нового договора
* Просмотр данных об обучающемся
* Внесение данных о новом обучающемся
* Просмотр всех обучающихся
* Отправка уведомлений с выбором типа уведомлений

На рисунке 14 представлена диаграмма вариантов использования системы адаптивного тестирования для дальневосточного центра математики.

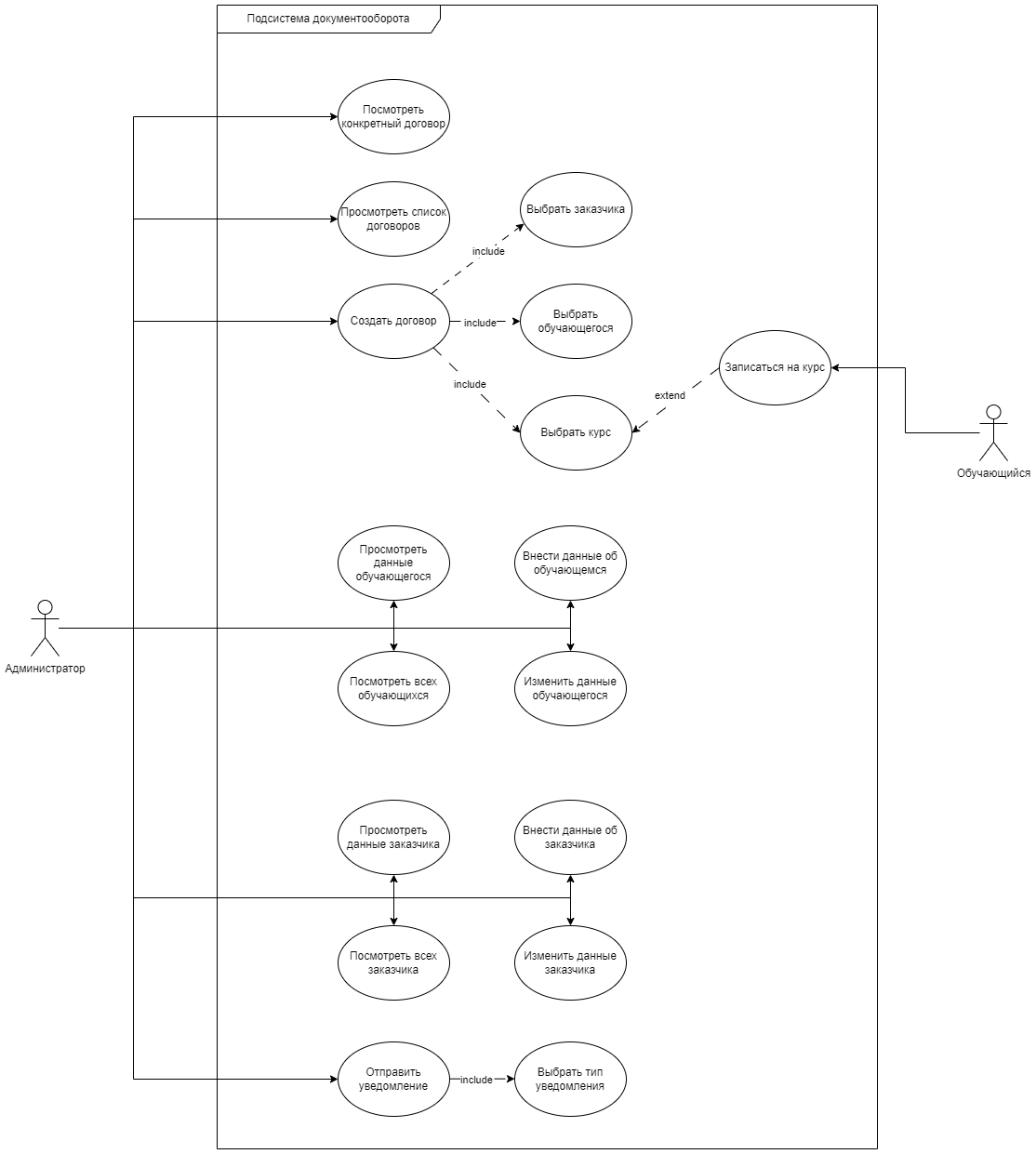


Рисунок 14 – Use Case диаграмма

### 3.4.1 Сценарии использования

**Варианты использования** (Сценарии использования) – это история о том, как система и пользователи могут взаимодействуют. Чаще всего части сценария использования выражается диаграммами, например use case. Но это не всегда дает понять полную последовательность действий и альтернативные сценарии.

Иными словами, **вариант использования** – это спецификации набора действий, выполняемых системой, который дает заметный результат и который, значим для одного или нескольких субъектов или других заинтересованных сторон системы.

Таблица 2 – Вариант использования 1. Регистрация обучающегося.

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор**  **Варианта использования** | Вариант использования – 1 | Название варианта использования | Регистрация обучающегося |
| **Автор** | Решетнев Никита | Автор последнего обновления | Решетнев Никита |
| **Дата создания** | 28.12.2023 | Дата последнего обновления | 28.12.2023 |
| **Действующее лицо** | Администратор, регистрирующий нового обучающегося | | |
| **Описание** | Администратор заходит в систему с целью регистрации нового обучающегося, который решил учиться в Дальневосточном центре математики. | | |
| **Предварительные условия** | 1. Система в исправном состоянии; 2. Об обучающемся нет информации в системе Дальневосточного центра математики; 3. У администратора стабильное интернет-соединение. | | |
| **Выходные условия** | Система возвращает созданный аккаунт обучающегося и записывает его в базу данных системы. | | |
| **Нормальное направление развития варианта использования** | 1. Пользователь проходит идентификацию. 2. Система идентифицирует пользователя как администратора. 3. Пользователь выбирает модуль внесения данных об обучающемся. 4. Пользователь вводит данные обучающегося. 5. Система обрабатывает данные обучающегося. | | |
|  | 1. Система выдает пользователю окно с подтверждением действия. | | |

*Окончание таблицы 2*

|  |  |
| --- | --- |
|  | 1. Пользователь подтверждает действие.   Система сохраняет нового обучающегося. |
| **Альтернативное направление варианта использования** | 1а   1. Неверный логин или пароль. 2. Слишком много попыток входа, повторите позже.   4а   1. Пользователь решает не вносить обучающегося и нажимает кнопку «Сохранить обучающегося». Система выдает ошибку о незаполненных полях.   5а   1. Пользователь ввел некорректные данные обучающегося. Система выдает ошибку о некорректных данных и возвращает пользователя к пункту 4 нормального направления.   6а  1. Пользователь решает не вносить обучающегося и нажимает кнопку «Отмена». Система возвращает пользователя к пункту 4 нормального направления. |
| **Исключения** | **\***e1   1. Отказ любого рода системы 2. Система выводит сообщение: «О сбое системы»   \*e2   1. Сбой питания: 2. Система выводит сообщение: «О сбое системы» |
| **Приоритет** | Высокий |
| **Частота использования** | Используется в каждом случае работы с системой |
| **Предложения** | Система должна автоматически переносить пользователя в модуль составления договоров |

\**О сбое системы* = «В системе произошла ошибка, попробуйте снова»

Таблица 3 – Вариант использования 2. Создание нового договора.

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор**  **Варианта использования** | Вариант использования - 2 | Название варианта использования | Создание нового договора |
| **Автор** | Решетнев Никита | Автор последнего обновления | Решетнев Никита |
| **Дата создания** | 28.12.2023 | Дата последнего обновления | 28.12.2023 |
| **Действующее лицо** | Администратор, который хочет создать новый договор для обучающегося | | |
| **Описание** | Пользователь заходит в программу, выбирает обучающегося и курс, создает новый договор. | | |

*Окончание таблицы 3*

|  |  |
| --- | --- |
| **Предварительные условия** | 1. Обучающийся должен быть зарегистрирован в системе Дальневосточного центра математики 2. Система в исправном состоянии. 3. Модуль создания договоров работает исправно. |
| **Выходные условия** | Система выдаёт созданный договор для обучающегося по определенной программе |
| **Нормальное направление развития варианта использования** | * 1. Пользователь проходит идентификацию.   2. Система идентифицирует пользователя как администратора.   3. Пользователь выбирает модуль создания договоров.   4. Пользователь выбирает обучающегося из списка обучающихся   5. Пользователь выбирает курс, на который нужно зачислить обучающегося и к которому нужно создать договор.   6. Пользователь вводит срок действия договора, цену договора   7. Система обрабатывает введенные данные.   8. Система выдает пользователю окно с подтверждением действия.   9. Пользователь подтверждает действие.   Система создает новый договор и сохраняет его в базу данных |
| **Альтернативное направление варианта использования** | 1а   1. Неверный логин или пароль. 2. Слишком много попыток входа, повторите позже.   6a   1. Пользователь вводит некорректные данные. Система выдает ошибку о некорректных данных и возвращает пользователя к пункту 6 нормального направления .   9а  1. Пользователь решает не создавать договор и нажимает кнопку «Отмена». Система возвращает пользователя к пункту 6 нормального направления. |
| **Исключения** | **\***e1   1. Отказ любого рода системы. 2. Система выводит сообщение: «О сбое системы».   \*e2   1. Сбой питания. 2. Система выводит сообщение: «О сбое системы». |
| **Приоритет** | Высокий |
| **Частота использования** | Используется в каждом случае выбора пользователем модуля создания новых договоров. |

\**О сбое системы* = «В системе произошла ошибка, попробуйте снова»

### 3.4.2 User Stories

**Пользовательские истории (User Story)** — описание требований системы, которые сформулированы в виде предложений на языке пользователя. Они используются гибкими методологиями разработки программного обеспечения, например для Agile, для спецификации требований. Ниже представлен список user story для подсистемы анализа текста задач:

1. Как пользователь, я хочу иметь возможность работать с документами офлайн, чтобы не зависеть от интернета;
2. Как пользователь, я хочу иметь возможность сортировать документы по курсам, на которые записаны обучающиеся;
3. Как пользователь, я хочу, чтобы система предоставляла удобный редактор данных обучающихся, чтобы править (добавлять, изменять) данные обучающихся в зависимости от изменений данных;
4. Как пользователь, я хочу, чтобы система автоматически, после создания обучающегося, открывала модуль создания договоров и вносила данные об обучающемся;
5. Как пользователь, я хочу, чтобы система умела сохранять созданные договоры в удобные для дальнейшего редактирования и просмотра форматы (docx, pdf, odt);
6. Как пользователь, я хочу, чтобы система автоматически, после создания договора, отправляла его на печать на принтер;
7. Как пользователь, я хочу, чтобы система автоматически отправляла уведомления нужного типа обучающимся;
8. Как пользователь, я хочу, чтобы система имела различные способы отправки уведомлений для обучающегося (почта, WhatsApp, SMS сообщение);
9. Как пользователь, я хочу, чтобы у системы был понятный и интуитивный интерфейс.

Для грамотного расставления приоритетов для требований была использована таблица, представленная в таблице 4, где «Must do» – задачи, которые должны быть выполнены обязательно, «Should do» – задачи, которые также обязательно выполнить, но они менее важны, «Could do» – задачи, которые также надо выполнить, но, если они не будут выполнены, система работать хуже не будет.

Таблица 4 – Таблица распределения приоритетов (Must do/Should do/Could do)

|  |  |  |
| --- | --- | --- |
| **Must do** | **Should do** | **Could do** |
| **Пользователь** | | |
| Как пользователь, я хочу иметь возможность сортировать документы по курсам, на которые записаны обучающиеся | Как пользователь, я хочу, чтобы система имела различные способы отправки уведомлений для обучающегося (почта, WhatsApp, SMS сообщение) | Как пользователь, я хочу, чтобы у системы был понятный и интуитивный интерфейс. |
| Как пользователь, я хочу, чтобы система автоматически отправляла уведомления нужного типа обучающимся | Как пользователь, я хочу, чтобы система автоматически, после создания договора, отправляла его на печать на принтер | Как пользователь, я хочу, чтобы система предоставляла удобный редактор данных обучающихся, чтобы править (добавлять, изменять) данные обучающихся в зависимости от изменений данных |
| Как пользователь, я хочу, чтобы система умела сохранять созданные договоры в удобные для дальнейшего редактирования и просмотра форматы (docx, pdf, odt) | Как пользователь, я хочу, чтобы система автоматически, после создания обучающегося, открывала модуль создания договоров и вносила данные об обучающемся |  |
| Как пользователь, я хочу иметь возможность работать с документами офлайн, чтобы не зависеть от интернета |  |  |

## 3.5 Проект верхнего уровня

Проект верхнего уровня системы документооборота для Дальневосточного центра математики представлена на рисунке 15.

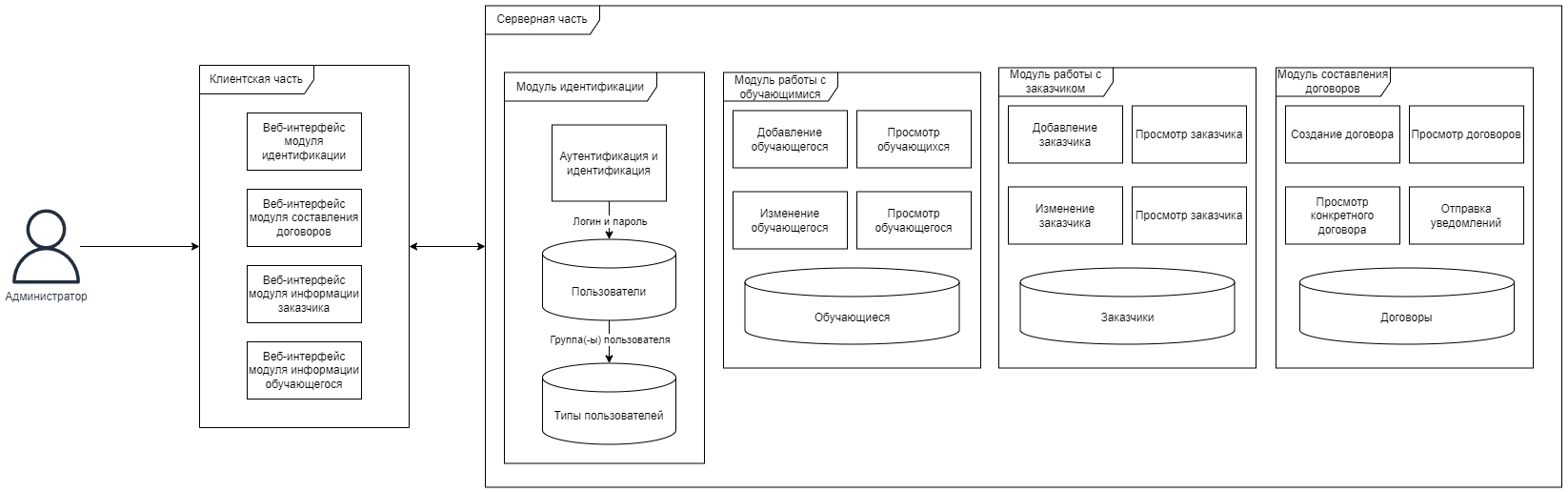


Рисунок 15– Проект верхнего уровня

Взаимодействие с системой документооборота происходит посредством веб-интерфейса на устройстве пользователей, здесь же отображается вся информация, предоставляемая самой системой.

Модуль идентификации администратора отвечает за проверку наличия записей об администраторе в базе данных, а также определения, что пользователь действительно является администратором.

Модуль работы с обучающимися отвечает за следующие функции:

1. Добавления новых обучающихся в базу данных приложения: модуль просит администратора ввести данные об обучающемся, после чего заносит их в базу данных для дальнейшей работы с ними.
2. Просмотр данных обучающегося: модуль способен показать все данные о каждом обучающемся, которые могут быть использованы администратором для осведомления обучающихся.
3. Изменение данных обучающегося: модуль способен изменять данные обучающихся. Требуется для актуализации данных.

Модуль работы с заказчиками отвечает за следующие функции:

1. Добавления новых заказчиков в базу данных приложения: модуль просит администратора ввести данные заказчика, после чего заносит их в базу данных для дальнейшей работы с ними.
2. Просмотр данных о заказчике: модуль способен показать все данные о каждом заказчике, которые могут быть использованы администратором для осведомления заказчиков;
3. Изменение данных заказчика: модуль способен изменять данные заказчиков. Требуется для актуализации данных.

Модуль составления договоров предназначен для полноценной работой с договорами (создание договоров, просмотр договоров, просмотр конкретного договора).

## 3.6 Диаграмма потоков данных

Диаграммы потоков данных (ДПД) используются для графического представления потока данных в информационной системе. ДПД описывает процессы, которые задействованы в системе для передачи данных из входных данных в базу данных и генерации отчетов. ДПД графически представляет функции или процессы, которые захватывают, манипулируют, хранят и распределяют данные между системой и ее средой, а также между компонентами системы.

Диаграмма потоков данных для системы документооборота для Дальневосточного центра математики представлена на рисунке 16

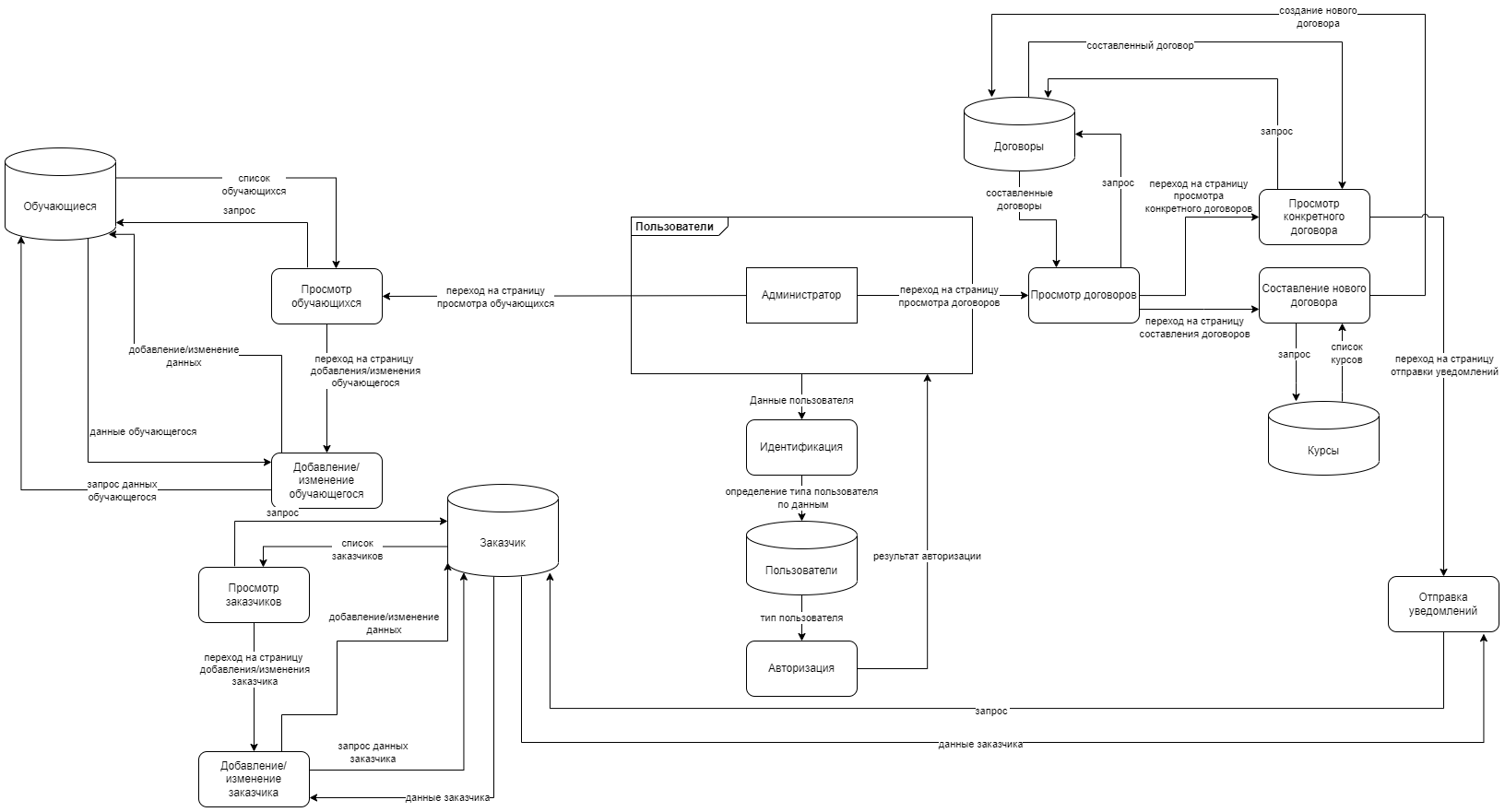


Рисунок 16 – Диаграмма потоков данных

## 3.7 Диаграмма состояний

Диаграмма состояний (ДС) используется для представления состояния системы или ее части в конечные моменты времени. Это поведенческая диаграмма демонстрирует поведение системы с использованием конечных переходов состояний. Проще говоря, ДС используется для моделирования динамического поведения системы в зависимости от времени и изменения внешних раздражителей.

Данная диаграмма используется для указания событий, ответственных за изменение состояния (не показывая, какие процессы вызывают эти события). Также, это помогает нам понять реакцию системы на раздражители.

Диаграмма состояний для системы документооборота для Дальневосточного центра математики представлена на рисунке 17.

На данной диаграмме выделены основные доступные действия системы документооборота, такие как:

* Просмотр договоров;
* Создание договора для заказчика;
* Внесение данных о новом обучающемся Дальневосточного центра математики;
* Просмотр данных обучающихся Дальневосточного центра математики;
* Редактирование данных обучающегося Дальневосточного центра математики;
* Внесение данных о новом заказчике Дальневосточного центра математики;
* Просмотр данных заказчиков Дальневосточного центра математики;
* Редактирование данных заказчиков Дальневосточного центра математики;
* Отправка уведомлений заказчикам.

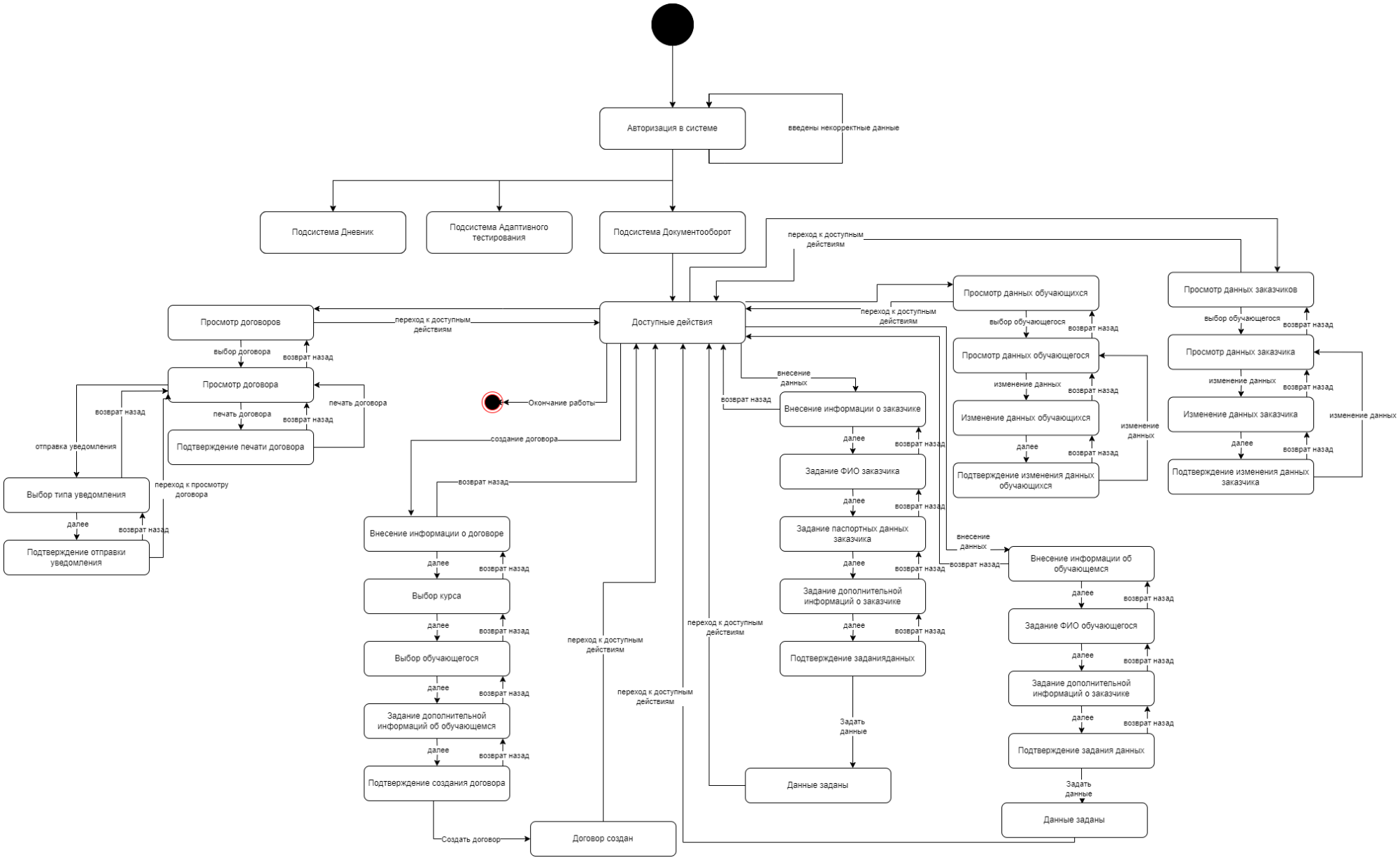


Рисунок 17 – Диаграмма состояний

## 3.8 Проект базы данных

Проект базы данных для системы документооборота для Дальневосточного центра математики представлен на рисунке 18.

Изображение выглядит как текст, диаграмма, Параллельный, Шрифт

Автоматически созданное описание

Рисунок 18 – Проект базы данных

### 3.8.1 Таблица users\_user

Используется для хранения информации об администраторах. Состоит из следующих столбцов:

1. id (обязательный) – идентификатор пользователя, используется для однозначной идентификации заказа в базе данных:
2. тип значения – целое число;
3. диапазон допустимых значений – [1; 264 – 1];
4. username (обязательный) – логин администратора, используется для аутентификации администратора в системе:
5. тип значения – строка;
6. диапазон допустимых значений – от 5 до 50 символов;
7. password (обязательный) – пароль администратора, используется для аутентификации администратора в системе:
8. тип значения — строка;
9. диапазон допустимых значений – до 128 символов;
10. last\_name (обязательный) — фамилия администратора, используется для идентификации администратора:
11. тип значения — строка;
12. диапазон допустимых значений — от 1 до 100 символов;
13. first\_name (обязательный) — имя администратора, используется для идентификации администратора:
14. тип значения — строка;
15. диапазон допустимых значений — от 1 до 100 символов;

### 3.8.2 Таблица contracts\_address

Используется для хранения информации о местах обучения. Состоит из следующих столбцов:

1. id (обязательный) – идентификатор места обучения, используется для однозначной идентификации места обучения в базе данных:
   1. тип значения – целое число;
   2. диапазон допустимых значений – [1; 264 – 1];
2. address (обязательный) – адрес места обучения, используется для хранения адресной информации:
   1. тип значения – строка;
   2. диапазон допустимых значений – от 5 до 255 символов.

### 3.8.3 Таблица university\_discipline

Используется для хранения информации о курсах, которые изучают в Дальневосточном центре математики. Состоит из следующих столбцов:

1. id (обязательный) – идентификатор курса, используется для однозначной идентификации курса в базе данных:
   1. тип значения – целое число;
   2. диапазон допустимых значений – [1; 264 – 1];
2. discipline\_name (обязательный) – название курса, используется для идентификации курса:
   1. тип значения – строка;
   2. диапазон допустимых значений – от 1 до 255 символов;
3. discipline\_price (обязательный) – стоимость курса, используется для указания стоимости курса:
   1. тип значения – десятичное число;
   2. диапазон допустимых значений – от 0 до 1 000 000.00;
4. lessons\_per\_week (обязательный) – количество уроков в неделю в курсе, используется для указания частоты занятий:
   1. тип значения – целое число;
   2. диапазон допустимых значений – от 1 до 14;
5. lesson\_time (обязательный) – время урока в курсе, используется для указания длительности каждого урока:
   1. тип значения – целое число (минуты);
   2. диапазон допустимых значений – от 30 до 180 минут.

### 3.8.4 Таблица university\_studygroup

Используется для хранения информации об учебных группах в Дальневосточном центре математики. Состоит из следующих столбцов:

1. id (обязательный) – идентификатор курса, используется для однозначной идентификации курса в базе данных:
   1. тип значения – целое число;
   2. диапазон допустимых значений – [1; 264 – 1];
2. name (обязательный) – название курса, используется для идентификации курса:
   1. тип значения – строка;
   2. диапазон допустимых значений – от 1 до 255 символов;
3. education\_date\_start (обязательный) – дата начала обучения, используется для указания даты начала курса:
   1. тип значения – дата;
   2. диапазон допустимых значений – формат даты 'YYYY-MM-DD';
4. education\_date\_end (обязательный) – дата окончания обучения, используется для указания даты завершения курса:
   1. тип значения – дата;
   2. диапазон допустимых значений – формат даты 'YYYY-MM-DD';
5. discipline\_id (обязательный) – идентификатор дисциплины, используется для связи курса с определенной дисциплиной:
   1. тип значения – целое число;
   2. диапазон допустимых значений – [1; 264 – 1];
6. grade (обязательный) – класс, используется для указания класса, для которого предназначен курс:
   1. тип значения – целое число;
   2. диапазон допустимых значений – от 1 до 11.

### 3.8.5 Таблица customers\_customer

Используется для хранения информации о заказчиках в Дальневосточном центре математики. Состоит из следующих столбцов:

1. id (обязательный) – идентификатор клиента, используется для однозначной идентификации клиента в базе данных:
   1. тип значения – целое число (bigint);
   2. диапазон допустимых значений – [1; 264 – 1];
2. name (обязательный) – имя клиента, используется для идентификации клиента:
   1. тип значения – строка;
   2. диапазон допустимых значений – до 256 символов;
3. phone\_number (обязательный) – номер телефона клиента, используется для связи с клиентом:
   1. тип значения – строка;
   2. диапазон допустимых значений – до 32 символов;
4. passport\_number (обязательный) – номер паспорта клиента, используется для идентификации личности клиента:
   1. тип значения – строка;
   2. диапазон допустимых значений – 11 символов. Только цифры;
5. passport\_issuance (обязательный) – место выдачи паспорта, используется для указания органа, выдавшего паспорт:
   1. тип значения – строка;
   2. диапазон допустимых значений – до 300 символов;
6. passport\_issue\_date (обязательный) – дата выдачи паспорта, используется для указания даты выдачи паспорта:
   1. тип значения – дата;
   2. диапазон допустимых значений – формат даты 'YYYY-MM-DD';
7. passport\_division\_code (обязательный) – код подразделения, выдавшего паспорт, используется для идентификации органа выдачи паспорта:
   1. тип значения – строка;
   2. диапазон допустимых значений – до 7 символов;
8. passport\_registration (обязательный) – регистрация по месту жительства, используется для указания адреса регистрации клиента:
   1. тип значения – строка;
   2. диапазон допустимых значений – до 500 символов;
9. email (обязательный) – электронная почта клиента, используется для связи с клиентом:
   1. тип значения – строка;
   2. диапазон допустимых значений – до 256 символов.

### 3.8.6 Таблица university\_student

Используется для хранения информации об учениках в Дальневосточном центре математики. Состоит из следующих столбцов:

1. id (обязательный) – идентификатор клиента, используется для однозначной идентификации клиента в базе данных:
   1. тип значения – целое число;
   2. диапазон допустимых значений – [1; 264 – 1];
2. name (обязательный) – имя клиента, используется для идентификации клиента:
   1. тип значения – строка;
   2. диапазон допустимых значений – до 64 символов;
3. phone\_number (обязательный) – номер телефона клиента, используется для связи с клиентом:
   1. тип значения – строка;
   2. диапазон допустимых значений – до 32 символов.

### 3.8.7 Таблица contracts\_contract

Используется для хранения информации о договорах в Дальневосточном центре математики. Состоит из следующих столбцов:

1. id (обязательный) – идентификатор договора, используется для однозначной идентификации договора в базе данных:
   1. тип значения – целое число;
   2. диапазон допустимых значений – [1; 264 – 1];
2. contract\_creation\_date (обязательный) – дата создания договора, используется для указания даты заключения договора:
   1. тип значения – дата;
   2. диапазон допустимых значений – формат даты 'YYYY-MM-DD';
3. contract\_expiration\_date (обязательный) – дата окончания договора, используется для указания даты окончания действия договора:
   1. тип значения – дата;
   2. диапазон допустимых значений – формат даты 'YYYY-MM-DD';
4. contract\_price (обязательный) – стоимость договора, используется для указания стоимости обучения по договору:
   1. тип значения – целое число;
   2. диапазон допустимых значений – от 0 до 1 000 000;
5. creator\_id (обязательный) – идентификатор создателя договора, используется для указания пользователя, создавшего договор:
   1. тип значения – целое число;
   2. диапазон допустимых значений – [1; 264 – 1];
6. customer\_id (обязательный) – идентификатор клиента, используется для указания клиента, заключившего договор:
   1. тип значения – целое число;
   2. диапазон допустимых значений – [1; 264 – 1];
7. student\_id (обязательный) – идентификатор ученика, используется для указания ученика, который будет обучаться по договору:
   1. тип значения – целое число;
   2. диапазон допустимых значений – [1; 264 – 1];
8. study\_group\_id (обязательный) – идентификатор учебной группы, используется для указания группы, в которой будет обучаться ученик:
   1. тип значения – целое число;
   2. диапазон допустимых значений – [1; 264 – 1];
9. address\_id (обязательный) – идентификатор адреса, используется для указания места проведения занятий:
   1. тип значения – целое число;
   2. диапазон допустимых значений – [1; 264 – 1];
10. is\_online (обязательный) – формат обучения, используется для указания, является ли обучение онлайн:
    1. тип значения – логическое;
    2. допустимые значения – true (онлайн) или false (оффлайн).

## 3.9 Проект интерфейса программного средства

### 3.9.1 Экран входа

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 19 – Экран входа в систему

При первом запуске веб приложения, администратор увидит экран идентификации (рисунок 19). Здесь администратор должен войти в систему под своим логином и паролем. Для входа необходимо заполнить поля «Логин» и «Пароль». При вводе неверных данных или отсутствии Интернет соединения появится сообщение об ошибке. После успешного входа администратор перейдёт на экран просмотра договоров.

### 3.9.2 Экран договоров

Экран договоров (рисунок 20) позволяет посмотреть договоры, заключить новый договор, посмотреть существующие договоры, отправить уведомления и сохранить в формате word. Начальный экран договоров состоит из:

1. заголовки;
2. элемент «Профиль»;
3. таблица с существующими договорами;
4. кнопки работы с договорами.

Заголовки позволяют выполнять навигацию между различными экранами веб приложения.

При нажатии на элемент профиля (рисунок 21) выпадает элемент, в котором написан логин администратора, зашедшего в приложение, и кнопка «Выход». При нажатии на кнопку «Выход» администратор перейдёт на экран входа (рисунок 19).

При нажатии на кнопку «Создать новый договор» открывается окно создания нового договора для заказчика (рисунок 22). В окне есть две кнопки: «Назад» и «Сохранить». Кнопка назад закрывает окно и возвращает администратора к состоянию, изображенному на рисунке 20.

Также, в данном окне можно выбрать из раскрывающихся списков заказчика, учеников, учебную группу, место обучения. Также, администратор выбирает дату начала и конца договора, и выбирает: онлайн или оффлайн обучение будет.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 20 – Экран «Контракты»

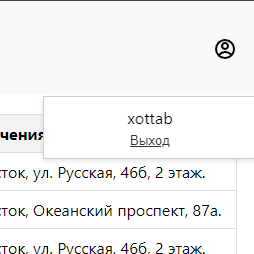


Рисунок 21 – Элемент «Профиль»

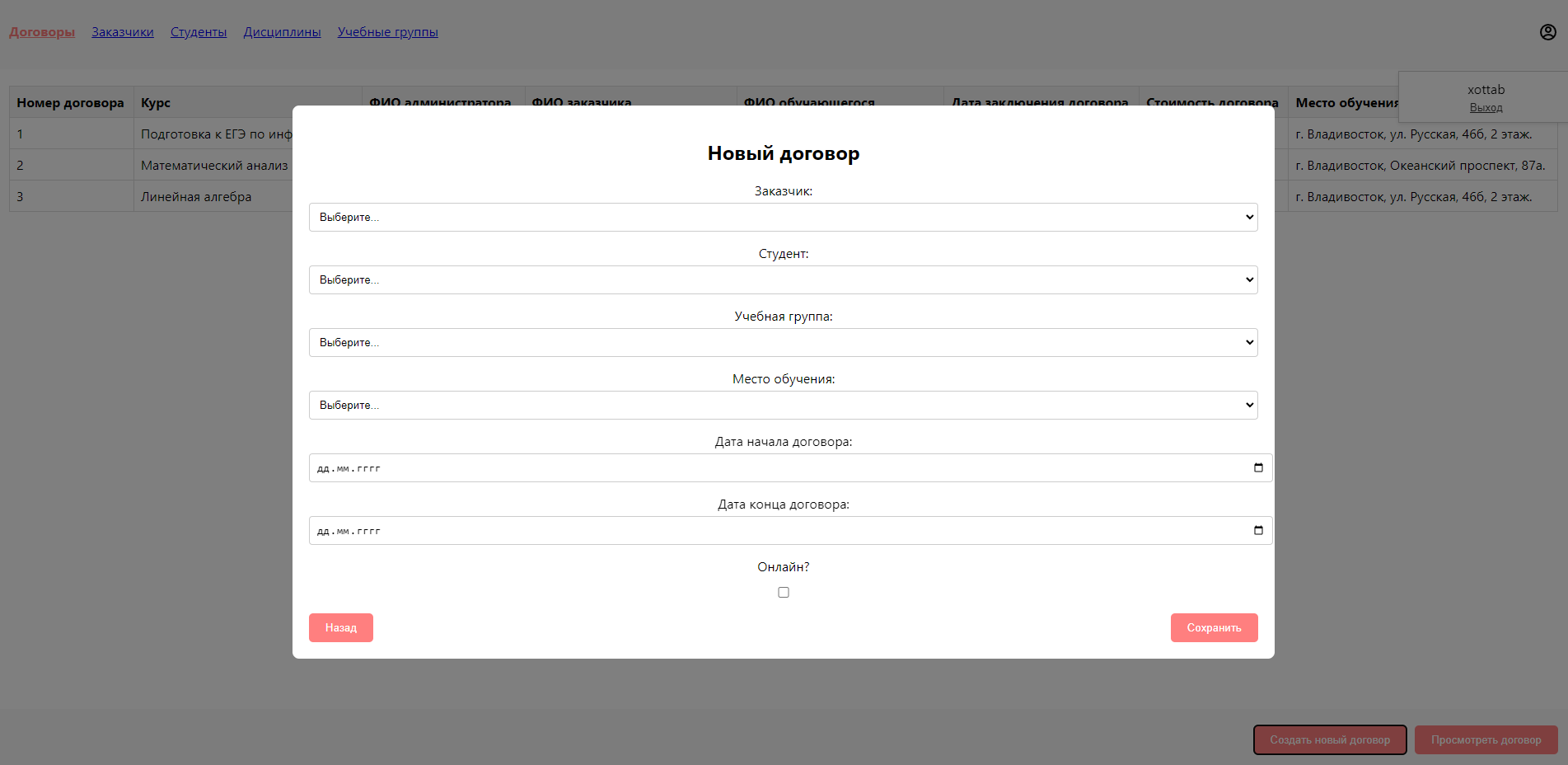


Рисунок 22 – Экран договоров. Окно создания нового договора

После нажатия на кнопку «Сохранить», введенные данные валидируются. Если возникла ошибка при заполнении – то у поля, в котором возникла ошибка – появится уведомление, которое подскажет, в чём проблема. Если всё хорошо – то договор сохраняется и окно закрывается.

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Автоматически созданное описание

Рисунок 23 – Экран договоров. Окно просмотра договора

При нажатии на один из договоров кнопку «Посмотреть договор» в экране договоров открывается окно просмотра договора, изображенное на рисунке 23.

В данном окне есть поле «Информация», в котором написана вся основная информация о договоре, ученике и заказчике. Также, данное окно имеет три кнопки:

1. Кнопка «Назад» – возвращает администратора к состоянию, изображенному на рисунке 20;
2. Кнопка «Сохранить в Word» – отправляет запрос на сервер о создании word файла. После создания word файла на сервере он отправляется на компьютер администратора, который может его сохранить. После чего окно закрывается;
3. Кнопка «Отправить уведомление» - отправляет запрос на сервер об отправке уведомления о задолженности на почту. Сервер отправляет с указанной почты сообщение, изображенное на рисунке 24. После чего окно закрывается.

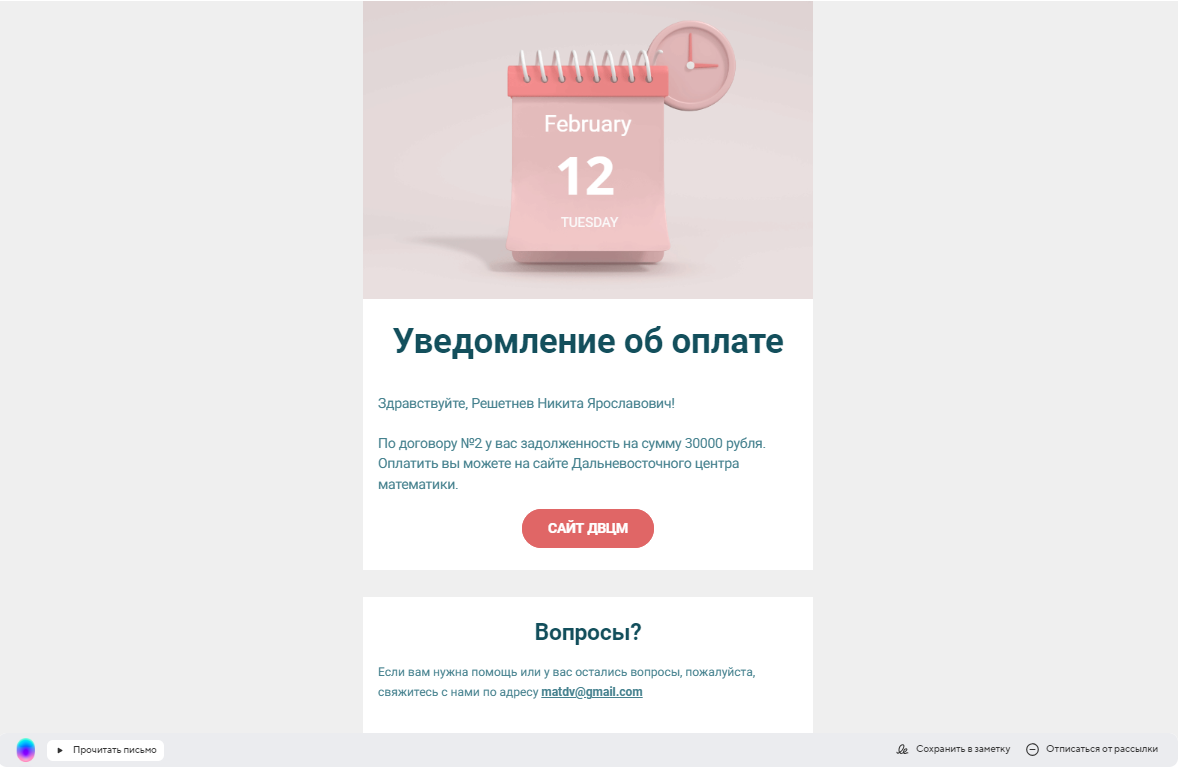


Рисунок 24 – Пример уведомления о задолженности

### 3.9.3 Экран просмотра заказчиков

Экран просмотра заказчиков (рисунок 25) позволяет просмотреть данные всех заказчиков, находящихся в базе данных.

Изображение выглядит как текст, программное обеспечение, веб-страница, снимок экрана

Автоматически созданное описание

Рисунок 25 – Экран просмотра заказчиков

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, число

Автоматически созданное описание

Рисунок 26 – Экран просмотра заказчиков. Окно создания заказчика

При нажатии на кнопку «Создать нового заказчика» открывается окно, изображенное на рисунке 26, в которое можно ввести все персональные данные заказчика. В окне есть две кнопки: «Назад» и «Сохранить». Кнопка назад закрывает окно и возвращает администратора к состоянию, изображенному на рисунке 25.

После нажатия на кнопку «Сохранить», введенные данные валидируются. Если возникла ошибка при заполнении – то у поля, в котором возникла ошибка – появится уведомление, которое подскажет, в чём проблема. Если всё хорошо – то заказчик сохраняется и окно закрывается.

При нажатии на заказчика и на кнопку «Изменить данные заказа» в экране просмотра заказчиков открывается окно редактирования заказчика (рисунок 27). В нём можно изменить персональные данные заказчика.

В окне есть две кнопки: «Назад» и «Сохранить». Кнопка назад закрывает окно и возвращает администратора к состоянию, изображенному на рисунке 25.

После изменения полей, введенные данные валидируются. И если ошибок в заполнении не было – то у данного заказчика изменятся персональные данные, и администратор возвращается к состоянию, изображенному на рисунке 25.

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Автоматически созданное описание

Рисунок 27 – Экран просмотра заказчиков. Окно изменения заказчика

### 3.9.4 Экран просмотра учеников

Экран просмотра учеников (рисунок 28) позволяет просмотреть данные всех учеников, находящихся в базе данных.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 28 – Экран просмотра учеников

При нажатии на кнопку «Создать нового ученика» открывается окно, изображенное на рисунке 29, в которое можно ввести все персональные данные ученика. В окне есть две кнопки: «Назад» и «Сохранить». Кнопка назад закрывает окно и возвращает администратора к состоянию, изображенному на рисунке 28.

После нажатия на кнопку «Сохранить», введенные данные валидируются. Если возникла ошибка при заполнении – то у поля, в котором возникла ошибка – появится уведомление, которое подскажет, в чём проблема. Если всё хорошо – то ученик сохраняется и окно закрывается.

При нажатии на ученика и на кнопку «Изменить данные ученика» в экране просмотра учеников открывается окно редактирования ученика (рисунок 30). В нём можно изменить персональные данные ученика.

В окне есть две кнопки: «Назад» и «Сохранить». Кнопка назад закрывает окно и возвращает администратора к состоянию, изображенному на рисунке 28.

После изменения полей, введенные данные валидируются. И если ошибок в заполнении не было – то у данного ученика изменятся персональные данные, и администратор возвращается к состоянию, изображенному на рисунке 28.

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 29 – Экран просмотра учеников. Окно создания ученика

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 30 – Экран просмотра учеников. Окно изменения ученика

### 3.9.5 Экран просмотра дисциплин

Экран просмотра дисциплин (рисунок 31) позволяет просмотреть данные всех дисциплин, находящихся в базе данных.

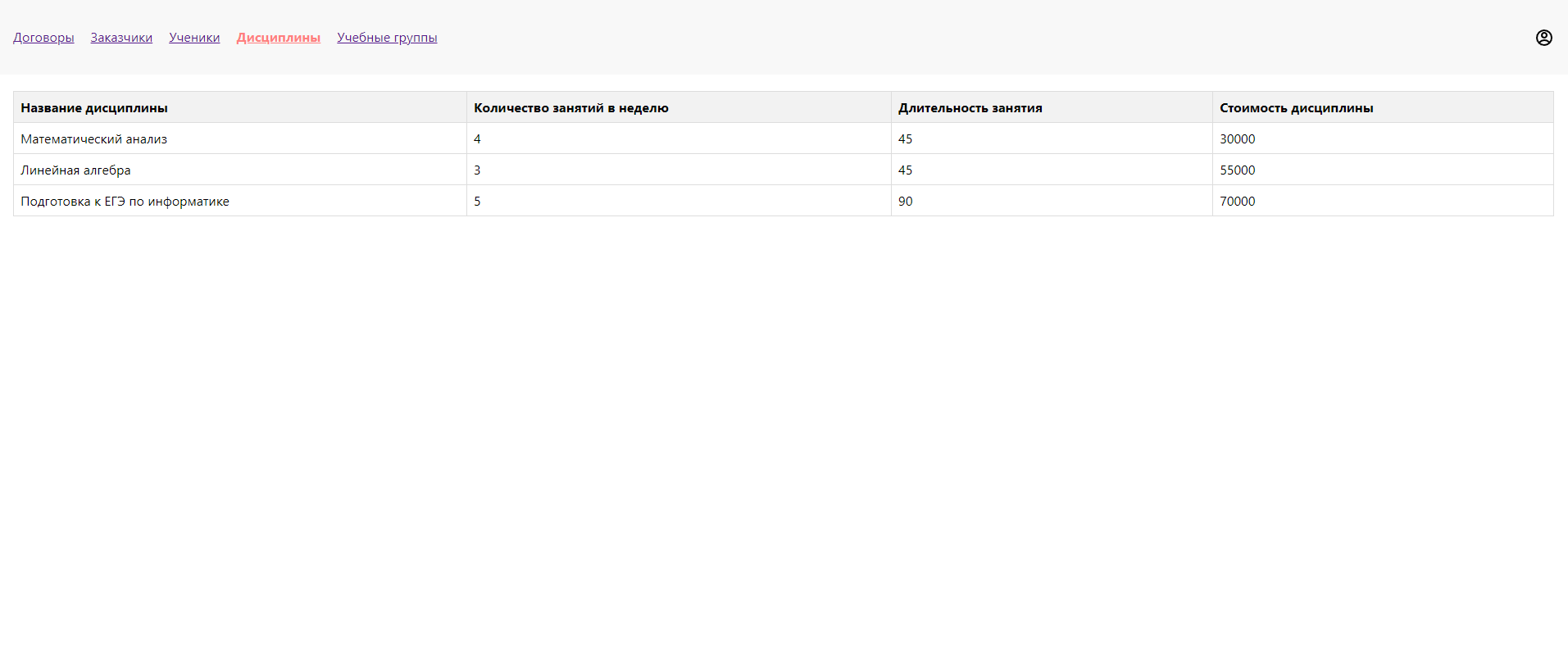


Рисунок 31 – Экран просмотра дисциплин

### 3.9.6 Экран просмотра учебных групп

Экран просмотра учебных групп (рисунок 32) позволяет просмотреть данные всех учебных групп, находящихся в базе данных.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 32 – Экран просмотра учебных групп

# 4 Реализация программного средства

На этапе реализации системы документооборота для Дальневосточного центра математики на практике были использованы спецификации, полученные на этапе проектирования, а также были выбраны средства реализации в соответствии со спецификациями.

## 4.1 Инструменты разработки

**Протокол взаимодействия подсистем [10].** HTTP (протокол передачи гипертекста) — протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате «HTML», в настоящий момент используется для передачи произвольных данных. В рамках выпускной квалификационной работы используется формат Json.

**Система контроля версий [11].** Git — распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано.

### 4.1.1 Клиентское приложение

**Разработка интерфейса [12].** Figma — онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени. Сервис бесплатный. Встроенные готовые шаблоны мобильных компонент отсутствуют, но их огромное количество в качестве плагинов от сторонних пользователей. Для тестирования на реальном устройстве существует мобильная версия приложения.

**Платформа.** Web-браузер – прикладное программное обеспечение для просмотра страниц, содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями; а также для решения других задач. В глобальной сети браузеры используют для запроса, обработки, манипулирования и отображения содержания веб-сайтов.

**Среда разработки [13].** JetBrains WebStorm IDE – специализированная среда разработки для JavaScript и TypeScript от компании JetBrains.

**Язык программирования [14].** JavaScript — это кроссплатформенный, интерпретируемый язык программирования, который используется для создания интерактивных веб-страниц и приложений. Функционал JavaScript может улучшить удобство взаимодействия пользователя с веб-сайтом: от обновления ленты новостей в социальных сетях и до отображения анимации и интерактивных карт. JavaScript был создан в 1995 году Бренданом Эйхом, когда тот еще был инженером в Netscape. JavaScript имеет динамическую слабую типизацию и может использоваться как для разработки на стороне клиента, так и для разработки на стороне сервера.

**Система сборки проекта [15].** Yarn - один из основных менеджеров пакетов JavaScript, разработанный в 2016 году Facebook (ныне Meta) для среды выполнения Node.js JavaScript . Альтернатива диспетчеру пакетов npm, Yarn была создана в результате сотрудничества Facebook, Exponent (ныне Expo.dev), Google и Tilde (компания, разработавшая Ember.js), чтобы решить проблемы согласованности, безопасности и производительности с большими кодовыми базами.

**Библиотеки [16, 17].**

keycloak-js – Keycloak поставляется с клиентской библиотекой JavaScript под названием keycloak-js, которую можно использовать для защиты веб-приложений. Адаптер также имеет встроенную поддержку приложений Cordova.

React (иногда React.js или ReactJS) — JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций.

### 4.1.2 Серверное приложение

**Платформа [18].** Веб-сервер — это специальная программа, которая принимает запросы пользователей, обрабатывает их и отправляет ответ обратно по протоколу прикладного уровня HTTP.

**Среда разработки [19].** PyCharm — кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains.

**Язык программирования [20].** Python — это интерпретируемый, интерактивный, объектно-ориентированный и высокоуровневый язык программирования общего назначения c динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода, а также на обеспечение переносимости написанных на нем программ. Задумка по реализации языка появилась в конце 1980-х годов, а разработка его реализации началась в 1989 году сотрудником голландского института CWI Гвидо ван Россумом.

**База данных [21].** PostgreSQL – это объектно-реляционная система управления базами данных (СУБД), основанная на POSTGRES версии 4.2, разработанная на факультете компьютерных наук Калифорнийского университета в Беркли. POSTGRES впервые применила многие концепции, которые стали доступны в некоторых коммерческих системах баз данных гораздо позже.

**Система управления пакетами [22].** pip – это стандартный инструмент для установки пакетов из PyPI. Pip доступен по умолчанию в большинстве установок Python через стандартный библиотечный пакет ensurepip.

**Библиотеки [23].** Django REST Framework (DRF) - это быстрый и стандартный способ создания REST API. Он имеет удобный для просмотра графический интерфейс, множество встроенных форм аутентификации и поддерживает множество типов сериализации. Технически не нужно использовать это для создания API (вместо этого желательно отправлять полезную нагрузку с запросами и ответами), но рекомендуется использовать DRF, поскольку он стандартизирован.

### 4.1.3 Инфраструктура

**Развертывание [24].** Docker – программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.

## 4.2 Тестирование

### 4.2.1 Структурное тестирование

В таблице 5 представлен набор тестов для структурного тестирования программной системы.

Таблица 5 – Набор тестов для структурного тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Тестовая ситуация** | **Начальное состояние системы** | **Действия** | **Ожидаемый результат** |
| 1 | Вход в систему | Экран авторизации | Пользователь вводит данные учетной записи и нажимает на кнопку «Авторизоваться» | Переход на экран администратора, |
| 2 | Выход из системы из экрана договоров | Экран договоров | Администратор нажимает на профиль, затем на кнопку «Выход» | Переход на экран авторизации |
| 3 | Выход из системы из экрана просмотра заказчиков | Экран просмотра заказчиков | Администратор нажимает на профиль, затем на кнопку «Выход» | Переход на экран авторизации |
| 4 | Выход из системы из экрана просмотра учеников | Экран просмотра учеников | Администратор нажимает на профиль, затем на кнопку «Выход» | Переход на экран авторизации |
| 5 | Выход из системы из экрана просмотра дисциплин | Экран просмотра дисциплин | Администратор нажимает на профиль, затем на кнопку «Выход» | Переход на экран авторизации |

*Продолжение таблицы 5*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Тестовая ситуация** | **Начальное состояние системы** | **Действия** | **Ожидаемый результат** |
| 6 | Выход из системы из экрана  просмотра учебных групп | Экран просмотра учебных групп | Администратор нажимает на профиль, затем на  кнопку «Выход» | Переход на экран авторизации |
| 7 | Создание нового договора | Экран договоров | Администратор нажимает на кнопку «Создать новый договор» | Открывается окно создания договора |
| 8 | Сохранение нового договора | Экран договоров, окно создания договора открыто, данные введены | Администратор нажимает на кнопку «Сохранить» | Договор сохраняется, окно закрывается |
| 9 | Закрытие окна создания договора | Экран договоров, окно создания договора открыто | Администратор нажимает на кнопку «Назад» | Окно закрывается |
| 10 | Просмотр существующего договора | Экран договоров, договор выбран | Администратор нажимает на кнопку «Посмотреть договор» | Открывается окно просмотра договора |
| 11 | Возврат к экрану договоров | Экран договоров, окно просмотра договора открыто | Администратор нажимает на кнопку «Назад» | Закрывается окно просмотра договора |
| 12 | Сохранение документа в Word | Экран договоров, окно просмотра договора открыто | Администратор нажимает на кнопку «Сохранить в Word» | Скачивается договор, окно закрывается |
| 13 | Отправка уведомлений | Экран договоров, окно просмотра договора открыто | Администратор нажимает на кнопку «Отправить уведомление» | Уведомление отправляется, окно закрывается |
| 14 | Создание нового заказчика | Экран просмотра заказчиков | Администратор нажимает на кнопку «Создать нового заказчика» | Открывается окно создания заказчика |
| 15 | Сохранение нового заказчика | Экран просмотра заказчиков, окно создания заказчиков открыто, данные введены | Администратор нажимает на кнопку «Сохранить» | Заказчик сохраняется, окно закрывается |

*Окончание таблицы 5*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Тестовая ситуация** | **Начальное состояние системы** | **Действия** | **Ожидаемый результат** |
| 16 | Закрытие окна создания заказчика | Экран просмотра заказчиков, окно создания заказчика открыто | Администратор нажимает на кнопку «Назад» | Окно закрывается |
| 17 | Изменение данных заказчика | Экран просмотра заказчиков, заказчик выбран | Администратор нажимает на кнопку «Изменить данные заказчика» | Открывается окно изменения заказчика |
| 18 | Сохранение измененных данных заказчика | Экран просмотра заказчиков, окно изменения заказчика открыто | Администратор нажимает на кнопку «Сохранить» | Заказчик сохраняется, окно закрывается |
| 19 | Отменить изменения | Экран просмотра заказчиков, окно изменения заказчика открыто | Администратор нажимает на кнопку «Назад» | Окно закрывается |
| 20 | Создание нового ученика | Экран просмотра учеников | Администратор нажимает на кнопку «Создать нового ученика» | Открывается окно создания ученика |
| 21 | Сохранение нового ученика | Экран просмотра учеников, окно создания ученика открыто, данные введены | Администратор нажимает на кнопку «Сохранить» | Ученик сохраняется, окно закрывается |
| 22 | Закрытие окна создания ученика | Экран просмотра учеников, окно создания ученика открыто | Администратор нажимает на кнопку «Назад» | Окно закрывается |
| 23 | Изменение данных ученика | Экран просмотра учеников, ученик выбран | Администратор нажимает на кнопку «Изменить данные ученика» | Открывается окно изменения ученика |
| 24 | Сохранение измененных данных ученика | Экран просмотра учеников, окно изменения ученика открыто | Администратор нажимает на кнопку «Сохранить» | Ученик сохраняется, окно закрывается |
| 25 | Отменить изменения | Экран просмотра учеников, окно изменения ученика открыто | Администратор нажимает на кнопку «Назад» | Окно закрывается |

### 4.2.2 Тестирование для подтверждения программной системы

В таблице 6 представлено разбиение входных и выходных данных на классы эквивалентности.

Таблица 6 – Классы эквивалентности входных и выходных данных

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Входные/выходные данные** | **Класс эквивалентности** | **Тестовая ситуация** |
| Вход в систему | Логин | 1 | Учетной записи не существует |
| 2 | Учетная запись существует |
| Пароль | 3 | Неверный пароль |
| 4 | Верный пароль |
| Результат | 5 | Неуспешная идентификация или аутентификация |
| 6 | Успешная идентификация и аутентификация |
| Создание нового договора | Заказчик | 7 | Заказчик выбран |
| 8 | Заказчик не выбран |
| Студент | 9 | Студент выбран |
| 10 | Студент не выбран |
| Учебная группа | 11 | Учебная группа выбрана |
| 12 | Учебная группа не выбран |
| Место обучения | 13 | Место обучения выбрано |
| 14 | Место обучения не выбрано |
| Дата начала договора: | 15 | Дата начала договора выбрана |
| 16 | Дата начала договора не выбрана |
| Дата конца договора: | 17 | Дата конца договора выбрана |
| 18 | Дата конца договора не выбрана |

*Продолжение таблицы 6*

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Входные/выходные данные** | **Класс эквивалентности** | **Тестовая ситуация** |
| Создание нового договора |  | 19 | Дата конца договора раньше, чем дата начала договора |
| Просмотр договора | Договор | 20 | Договор выбран |
| 21 | Договор не выбран |
| 22 | Договор сохранен в Word |
| 23 | Уведомление отправлено |
| Создание нового заказчика | ФИО заказчика | 24 | Указано ФИО |
| 25 | Пустое ФИО |
| 26 | Невалидное ФИО |
| Номер телефона | 27 | Указан номер телефона |
| 28 | Пустой номер телефона |
| 29 | Невалидный номер телефона |
| Email | 30 | Указан email |
| 31 | Пустой email |
| 32 | Невалидный email |
| Дата выдачи | 33 | Указана дата выдачи |
| 34 | Пустая дата выдачи |
| 35 | Невалидная дата выдачи |
| Место выдачи паспорта | 36 | Указано место выдачи паспорта |
| 37 | Пустое место выдачи паспорта |

*Продолжение таблицы 6*

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Входные/выходные данные** | **Класс эквивалентности** | **Тестовая ситуация** |
| Создание нового заказчика | Регистрация | 38 | Указана регистрация |
| 39 | Пустая регистрация |
| Создание нового заказчика | Серия и номер паспорта | 40 | Указаны серия и номер паспорта |
| 41 | Пустые серия и номер паспорта |
| 42 | Невалидные серия и номер паспорта |
| 43 | Серия и номер паспорта уже есть |
| Код подразделения | 44 | Указан код подразделения |
| 45 | Пустой код подразделения |
| 46 | Невалидный код подразделения |
| Изменение заказчика | Заказчик | 47 | Заказчик выбран |
| 48 | Заказчик не выбран |
| ФИО заказчика | 49 | Указано ФИО |
| 50 | Пустое ФИО |
| 51 | Невалидное ФИО |
| Номер телефона | 52 | Указан номер телефона |
| 53 | Пустой номер телефона |
| 54 | Невалидный номер телефона |
| Email | 55 | Указан email |
| 56 | Пустой email |

*Продолжение таблицы 6*

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Входные/выходные данные** | **Класс эквивалентности** | **Тестовая ситуация** |
| Изменение заказчика |  | 57 | Невалидный email |
| Дата выдачи | 58 | Указана дата выдачи |
| 59 | Пустая дата выдачи |
| 60 | Невалидная дата выдачи |
| Место выдачи паспорта | 61 | Указано место выдачи паспорта |
| 62 | Пустое место выдачи паспорта |
| Регистрация | 63 | Указана регистрация |
| 64 | Пустая регистрация |
| Серия и номер паспорта | 65 | Указаны серия и номер паспорта |
| 66 | Пустые серия и номер паспорта |
| 67 | Невалидные серия и номер паспорта |
| 68 | Серия и номер паспорта уже есть |
| Код подразделения | 69 | Указан код подразделения |
| 70 | Пустой код подразделения |
| 71 | Невалидный код подразделения |
| Создание ученика | ФИО ученика | 72 | Указано ФИО |
| 73 | Пустое ФИО |
| 74 | Невалидное ФИО |
| Номер телефона | 75 | Указан номер телефона |

*Окончание таблицы 6*

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Входные/выходные данные** | **Класс эквивалентности** | **Тестовая ситуация** |
|  |  | 76 | Пустой номер телефона |
| 77 | Невалидный номер телефона |
| Изменение ученика | Ученик | 78 | Ученик выбран |
| 79 | Ученик не выбран |
| ФИО ученика | 80 | Указано ФИО |
| 81 | Пустое ФИО |
| 82 | Невалидное ФИО |
| Номер телефона | 83 | Указан номер телефона |
| 84 | Пустой номер телефона |
| 85 | Невалидный номер телефона |

**4.2.3 Набор тестов для проверки на всех тестовых ситуациях**

Тестирование по стратегии черного ящика заключается в выборе соответствующих данных в соответствии с функциональными возможностями системы и ее тестирования в отношении функциональных спецификаций для того, чтобы проверить наличие нормального и ненормального поведения системы. Набор тестов для проверки всех тестовых ситуациях представлен в таблице 7.

Таблица 7 – Набор тестов по методу черного ящика

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Покрываемые классы эквивалентности** | **Входные данные** | **Выходные данные** |
| Вход в систему | 1, 3, 5 | Логин = «»  Пароль = «» | Неуспешная идентификация или аутентификация |

*Продолжение таблицы 7*

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Покрываемые классы эквивалентности** | **Входные данные** | **Выходные данные** |
| Вход в систему | 1, 4, 5 | Логин = «test»  Пароль = «test»  (Не найдена учетная запись) | Неуспешная идентификация или аутентификация |
| 2, 3, 5 | Логин = «xottab»  Пароль = «123323»  (Неверный пароль) | Неуспешная идентификация или аутентификация |
| 2, 4, 6 | Логин = «xottab»  Пароль = «12341» | Успешная идентификация и аутентификация |
| Создание договора | 8, 10, 12, 14, 16, 18 | Заказчик = «»  Студент = «»  Учебная группа = «»  Место обучения = «»  Дата начала договора = «»  Дата конца договора = «» | Договор не создан |
| 7, 9, 11, 13, 15, 19 | Заказчик = «1»  Студент = «1»  Учебная группа = «1»  Место обучения = «1»  Дата начала договора = «11.11.2011»  Дата конца договора = «11.11.2010» | Договор не создан |
| 7, 9, 11, 13, 15, 17 | Заказчик = «1»  Студент = «1»  Учебная группа = «1»  Место обучения = «1»  Дата начала договора = «11.11.2011»  Дата конца договора = «11.11.2012» | Договор создан |
| Просмотр договора | 21 | Договор = «» | Пожалуйста, выберите договор |
| 20, 22, 23 | Договор = «  Заказчик = «1»  Студент = «1»  Учебная группа = «1»  Место обучения = «1»  Дата начала договора = «11.11.2011»  Дата конца договора = «11.11.2012»  » | Договор выбран |

*Продолжение таблицы 7*

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Покрываемые классы эквивалентности** | **Входные данные** | **Выходные данные** |
| Создание нового заказчика | 25, 28, 31, 34, 37, 39, 41, 45, | ФИО заказчика = «»  Номер телефона = «»  Email = «»  Серия и номер паспорта = «» | Заказчик не создан |
|  | Место выдачи паспорта = «»  Регистрация = «»  Дата выдачи = «»  Код подразделения = «» |  |
| 26, 27, 30, 33, 36, 38, 40, 44 | ФИО заказчика = «ААА»  Номер телефона = «+7999999999»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-007» | Заказчик не создан |
| 24, 29, 30, 33, 36, 38, 40, 44 | ФИО заказчика = «Иванов»  Номер телефона = «-vqwerty»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-007» | Заказчик не создан |
| 24, 27, 32, 33, 36, 38, 40, 44 | ФИО заказчика = «Иванов»  Номер телефона = «+7999999999»  Email = «qwerty@»  Серия и номер паспорта = «4444 444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-007» | Заказчик не создан |

*Продолжение таблицы 7*

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Покрываемые классы эквивалентности** | **Входные данные** | **Выходные данные** |
| Создание нового заказчика | 24, 27, 30, 35, 36, 38, 40, 44 | ФИО заказчика = «Иванов»  Номер телефона = «+7999999999»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2077»  Код подразделения = «270-007» | Заказчик не создан |
| 24, 27, 30, 33, 36, 38, 42, 44 | ФИО заказчика = «Иванов»  Номер телефона = «+7999999999»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-007» | Заказчик не создан |
| 24, 27, 30, 33, 36, 38, 43, 44 | ФИО заказчика = «Иванов»  Номер телефона = «+7999999999»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-007»  [База данных]  Серия и номер паспорта = «4444 444444» | Заказчик не создан |
| 24, 27, 30, 33, 36, 38, 40, 46 | ФИО заказчика = «Иванов»  Номер телефона = «+7999999999»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444» | Заказчик не создан |

*Продолжение таблицы 7*

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Покрываемые классы эквивалентности** | **Входные данные** | **Выходные данные** |
| Создание нового заказчика |  | Место выдачи паспорта =  «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-0077» |  | |
| 24, 27, 30, 33, 36, 38, 40, 44 | ФИО заказчика = «Иванов»  Номер телефона = «+7999999999»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-007» | Заказчик создан | |
| Изменение заказчика | 50, 53, 56, 59, 62, 64, 66, 70 | ФИО заказчика = «»  Номер телефона = «»  Email = «»  Серия и номер паспорта = «»  Место выдачи паспорта = «»  Регистрация = «»  Дата выдачи = «»  Код подразделения = «» | Заказчик не изменен | |
| 51, 52, 55, 58, 61, 63, 65, 69 | ФИО заказчика = «ААА»  Номер телефона = «+7999999999»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-007» | Заказчик не изменен | |
| 49, 54, 55, 58, 61, 63, 65, 69 | ФИО заказчика = «Иванов»  Номер телефона = «-vqwerty»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444» | Заказчик не изменен | |

*Продолжение таблицы 7*

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Покрываемые классы эквивалентности** | **Входные данные** | **Выходные данные** |
| Изменение заказчика | 49, 52, 57, 58, 61, 63, 65, 69 | Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-007»  ФИО заказчика = «Иванов»  Номер телефона = «+7999999999»  Email = «qwerty@»  Серия и номер паспорта = «4444 444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-007» | Заказчик не изменен |
| 49, 52, 55, 60, 61, 63, 65, 69 | ФИО заказчика = «Иванов»  Номер телефона = «+7999999999»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2077»  Код подразделения = «270-007» | Заказчик не изменен |
| 49, 52, 55, 58, 61, 63, 67, 69 | ФИО заказчика = «Иванов»  Номер телефона = «+7999999999»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-007» | Заказчик не изменен |
| 49, 52, 55, 58, 61, 63, 68, 69 | ФИО заказчика = «Иванов»  Номер телефона = «+7999999999»  Email = «qwerty@q.ru» | Заказчик не изменен |

*Продолжение таблицы 7*

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Покрываемые классы эквивалентности** | **Входные данные** | **Выходные данные** |
| Изменение заказчика |  | Серия и номер паспорта = «4444 444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-007»  [База данных]  Серия и номер паспорта = «4444 444444» | Заказчик не изменен |
| 49, 52, 55, 58, 61, 63, 65, 71 | ФИО заказчика = «Иванов»  Номер телефона = «+7999999999»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-0077» | Заказчик не изменен |
| 49, 52, 55, 58, 61, 63, 65, 69 | ФИО заказчика = «Иванов»  Номер телефона = «+7999999999»  Email = «qwerty@q.ru»  Серия и номер паспорта = «4444 444444»  Место выдачи паспорта = «Yes»  Регистрация = «Yes»  Дата выдачи = «11.11.2011»  Код подразделения = «270-007» | Заказчик изменен |
| Создание ученика | 73, 76 | ФИО ученика = «»  Номер телефона = «» | Ученик не создан |
| 74, 75 | ФИО ученика = «ААА»  Номер телефона = «+79999999999» | Ученик не создан |
| 72, 77 | ФИО ученика = «Иванов»  Номер телефона = «-vsss» | Ученик не создан |
| 72, 75 | ФИО ученика = «Иванов» | Ученик создан |

*Окончание таблицы 7*

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Покрываемые классы эквивалентности** | **Входные данные** | **Выходные данные** |
|  |  | Номер телефона = «+79999999999» |  |
| Изменение ученика | 81, 84 | ФИО ученика = «»  Номер телефона = «» | Ученик не создан |
| 82, 83 | ФИО ученика = «ААА»  Номер телефона = «+79999999999» | Ученик не создан |
| 80, 85 | ФИО ученика = «Иванов»  Номер телефона = «-vsss» | Ученик не создан |
| 80, 83 | ФИО ученика = «Иванов»  Номер телефона = «+79999999999» | Ученик создан |

### 4.2.4 Нагрузочное тестирование

В рамках тестирования разработанного сервиса было проведено нагрузочное тестирование функций вывода договоров, просмотра заказчиков и учеников. Алгоритм тестирования:

* 1. Очищается база данных;
  2. В базу данных добавляется набор заказчиков, учеников и договоров;
  3. Осуществляется запрос на вывод договоров;
  4. Замеряется время ответа сервера;
  5. Осуществляется запрос на вывод заказчиков;
  6. Замеряется время ответа сервера;
  7. Осуществляется запрос на вывод учеников;
  8. Замеряется время ответа сервера;
  9. Полученные результаты суммируются
  10. Пункты 2-9 повторяются для наборов от 10 до 10000 пользователей;

**Параметры тестового стенда:**

* 1. Операционная система: Windows
  2. Процессор:
     1. Тактовая частота: до 4,3 Ghz
     2. Общее количество ядер: 12
  3. Объём оперативной памяти: 32 ГБ
  4. Накопитель:
     1. Объём: 2 ТБ
     2. Максимальная скорость чтения/записи: 5000 мегабайт в секунду

На рисунке 33 представлен график зависимости времени ответа сервера от количества одновременного проходящих тестирование и запрашивающих аналитику пользователей.



Рисунок 33 – График зависимости времени обработки команд от количества записей в базе данных

Время ответа нарастает линейно и не является оптимальным. В связи с тем, что модель договора в ORM Django была построена с подвязыванием учеников и заказчиков, с большим количеством записей будет возрастать большее время обработки запросов.

Для решения данной проблемы был настроен триггер для переноса всех договоров в архивную таблицу базы данных. Данный триггер будет работать каждый месяц, что обеспечит малые размеры (не более 1000 записей в таблице) основной таблицы. Код функции, которая перемещает данные из одной таблицы в другую представлен на листинге 4.2.4.1. Также, на листинге 4.2.4.2 представлен триггер, который выполняет функцию переноса раз в месяц.

Листинг 4.2.4.1 – функция переноса данных

|  |
| --- |
| CREATE OR REPLACE FUNCTION archive\_contracts()  RETURNS VOID AS $$  BEGIN  INSERT INTO contracts\_archive  SELECT \* FROM contracts;  TRUNCATE TABLE contracts;  END;  $$ LANGUAGE plpgsql; |

Листинг 4.2.4.2 – триггер запуска функции каждый месяц

|  |
| --- |
| INSERT INTO pgagent.pga\_job (jobname, jobdesc, jobhostagent, jobenabled)  VALUES ('Monthly Archive Contracts', 'Archive contracts monthly', '', true);  SELECT jobid FROM pgagent.pga\_job WHERE jobname = 'Monthly Archive Contracts';  INSERT INTO pgagent.pga\_jobstep (jstjobid, jstname, jstdesc, jstcode, jstkind, jstenabled)  VALUES (<jobid>, 'Archive Step', 'Step to archive contracts', 'CALL archive\_contracts();', 'sql', true);  INSERT INTO pgagent.pga\_schedule (jscjobid, jscname, jscdesc, jscenabled, jscminutes, jschours, jscdays, jscmonths, jscweekdays, jscstart)  VALUES (<jobid>, 'Monthly Schedule', 'Run monthly', true, '0', '0', '1', '\*', '\*', now()); |

### 4.2.5 Результаты тестирования

* 1. Было введено 25 тестовых случаев структурного тестирования;

1. Было выделено 85 классов эквивалентностей входных/выходных данных этих функций;
2. Было покрыто 85 классов эквивалентностей входных/выходных данных;
3. Результаты нагрузочного тестирования свидетельствуют о том, что производительность системы улучшилась. После внедрения триггера время обработки запросов уменьшилось за счёт малого объема таблицы.

**Заключение**

В ходе выполнения выпускной квалификационной работы были получены следующие результаты:

1. проанализированы существующие программные системы для составления договоров с напоминаниями об оплате;
2. проанализирована предметная область «Документооборот» и построена ее модель;
3. разработан проект подсистемы «Документооборот»;
4. реализован прототип подсистемы «Документооборот» и проведено его тестирование.

Цель достигнута. Все задачи, поставленные для достижения цели выпускной квалификационной работы были выполнены.

# Список литературы

1. Дальневосточный центр математики [Электронный ресурс]. URL: https://matdv.ru/ (дата обращения 15.12.2022);
2. КонсультантПлюс [Электронный ресурс]. URL: https://www.consultant.ru/document/cons\_doc\_LAW\_5142/80d946bb2e2c22c74d20a8bdb37d0f0c034aa6bc/ (дата обращения 15.12.2022);
3. Онлайн словарь-энциклопедия Academic [Электронный ресурс]. URL: https://noun\_ru.academic.ru/10608/уведомление (дата обращения 15.12.2022);
4. Мой класс – CRM для школ и детских центров [Электронный ресурс]. URL: https://moyklass.com/ (дата обращения 15.12.2022);
5. Конструктор договоров \ КонсультантПлюс [Электронный ресурс]. URL: https://www.consultant.ru/about/kd/ (дата обращения 15.12.2022);
6. Конструктор документов FreshDoc [Электронный ресурс]. URL: https://www.freshdoc.ru/ (дата обращения 15.12.2022);
7. Документация по 1С: Документооборот [Электронный ресурс]. URL: https://v8.1c.ru/doc8/rabota-s-dogovornymi-dokumentami/ (дата обращения 15.12.2022);
8. Документооборот в организации: правила и основные этапы [Электронный ресурс]. URL: https://www.kdelo.ru/art/385536-dokumentooborot-v-organizatsii-19-m9 (дата обращения 15.12.2022);
9. Артемьева, И. Л. Методы системного анализа и моделирования / И. Л. Артемьева // Дальневосточный Федеральный Университет. –2019. –С. 12-20;
10. HTTP [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/HTTP (дата обращения 07.06.2024);
11. Git [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Git (дата обращения 07.06.2024);
12. Figma [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Figma (дата обращения 07.06.2024);
13. Webstorm [Электронный ресурс]. URL: https://www.jetbrains.com/webstorm/ (дата обращения 07.06.2024);
14. JavaScript [Электронный ресурс]. URL: https://ravesli.com/intro-javascript/ (дата обращения 07.06.2024);
15. Yarn [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Yarn\_(package\_manager) (дата обращения 07.06.2024);
16. keycloak-js [Электронный ресурс]. URL: https://www.keycloak.org/docs/latest/securing\_apps/ (дата обращения 07.06.2024);
17. React [Электронный ресурс]. URL: https://react.dev/ (дата обращения 07.06.2024);
18. Веб-сервер [Электронный ресурс]. URL: https://doka-guide.vercel.app/tools/web-server/ (дата обращения 07.06.2024);
19. PyCharm [Электронный ресурс]. URL: https://www.jetbrains.com/pycharm/ (дата обращения 07.06.2024);
20. Python [Электронный ресурс]. URL: https://ravesli.com/intro-python/ (дата обращения 07.06.2024);
21. PostgreSQL [Электронный ресурс]. URL: https://www.postgresql.org/docs/current/intro-whatis.html (дата обращения 07.06.2024);
22. pip [Электронный ресурс]. URL: https://packaging.python.org/en/latest/guides/tool-recommendations/ (дата обращения 07.06.2024);
23. Django Rest Framework [Электронный ресурс]. URL: https://williamqliu.github.io/2015/10/03/django-rest-framework.html (дата обращения 07.06.2024);
24. Docker [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Docker (дата обращения 07.06.2024).