

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет Математики и информатики
Кафедра Информационных и управляющих систем
Направление подготовки 09.03.01 – «Информатика и вычислительная техника»
Направленность (профиль) образовательной программы Автоматизированные
системы обработки информации и управления

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. Кафедрой
_____ А.В. Бушманов
« ____ » _____ 2023г

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка ПО для личного кабинета студента Амурского государственного университета «Распределение повышенной стипендии»

Исполнитель студент группы 953об	_____	Д.К. Сулейманов
	(подпись, дата)	
Руководитель доцент, канд. физ.-мат. наук	_____	В.В. Ерёмина
	(подпись, дата)	
Консультант по безопасности и экологичности доцент, канд. техн. наук	_____	А.Б. Булгаков
	(подпись, дата)	
Нормоконтроль инженер кафедры	_____	В.Н. Адаменко
	(подпись, дата)	

Благовещенск 2023

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. Кафедрой
_____ А.В. Бушманов
«_____» _____ 2023г

ЗАДАНИЕ

К выпускной квалификационной работе студента: Сулейманов Д.К.

1. Тема выпускной квалификационной работы: Разработка ПО для личного кабинета студента Амурского государственного университета «Распределение повышенной стипендии» (Утверждена приказом от 24.04.2023 №974-уч)

2. Срок сдачи студентом законченной работы _____

3. Содержание выпускной квалификационной работы: описание и анализ предметной области; инфологическое проектирование; логическое проектирование; физическое проектирование; разработка информационной базы данных; разработка программного модуля; описание программного интерфейса.

4. Перечень материалов приложения: Техническое задание

5. Дата выдачи задания: 07.09.2022

Руководитель выпускной квалификационной работы: _____

Еремина В.В. , канд. физ.-мат. наук, доцент

(фамилия, имя, отчество, должность, уч.степень, уч.звание)

Задание принял к исполнению (07.09.2022) _____

(подпись студента)

РЕФЕРАТ

Бакалаврская работа содержит 106 с., 55 рисунков, 24 таблицы, 17 источников.

РАЗРАБОТКА, ПРОЕКТИРОВАНИЕ, ER-ДИАГРАММА, ЛОГИЧЕСКАЯ МОДЕЛЬ БД, ФИЗИЧЕСКАЯ МОДЕЛЬ БД, ФУНКЦИОНАЛЬНАЯ ЗАВИСИМОСТЬ, UML, ACTIVE RECORD, RUBY ON RAILS, POSTGRESQL

Основной целью системы является автоматизация процесса распределения стипендий в Амурском государственном университете, увеличение прозрачности этого процесса и повышение его эффективности.

В данной выпускной квалификационной работе объектом исследования является автоматизированная система для распределения стипендий повышенных академических стипендий среди студентов Амурского государственного университета.

Предметом исследования является процесс разработки, реализации и внедрения данной системы, а также ее влияние на процесс распределения стипендий и мотивацию студентов к учебной и научной деятельности.

СОДЕРЖАНИЕ

Введение	9
1 Анализ предметной области	11
1.1 Обоснование актуальности темы квалификационного исследования	11
1.2 Положения о стипендии и структура организации	12
1.2.1 Структура университета	12
1.2.2 Положение о повышенных стипендий	15
1.2.3 Информационная структура университета	18
1.3 Обзор существующих методов решения рассматриваемой задачи	23
1.4 формулировка задачи исследования и общей методики ее решения	26
2 Системное проектирование АСОИУ	29
2.1 Средства разработки программного обеспечения	29
2.2 Разработка концепции архитектуры построения и платформы реализации информационных систем	31
2.3 Структура информационных систем, состав функциональных и обеспечивающих подсистем	36
2.4 Выбор и обоснование модели жизненного цикла	40
2.5 Требования к информационной безопасности для приложений «Ruby on Rails»	44
3 Программная реализация	45
3.1 Разработка структуры базы данных	45
3.1.1 Описание инфологической структуры	45
3.1.2 Описание логической структуры	50
3.1.3 Реализация физической структуры	56
3.2 Разработка клиентской части подсистемы	60
3.3 Разработка серверной части подсистемы	63

3.3.1	Взаимодействие серверной части с внешними системами	63
3.3.2	Взаимодействие с базой данных	65
3.4	Работа с программным обеспечением	66
3.4.1	Работа с модулем «Портфолио»	66
3.4.2	Работа с модулем «распределение стипендий»	70
4	Безопасность и экологичность	78
4.1	Безопасность	78
4.1.1	Опасные и вредные факторы на рабочем месте пользователя ПЭВМ	78
4.1.2	Освещение	82
4.1.3	Микроклимат	84
4.1.4	Требования к графическому интерфейсу	86
4.2	Экологичность	87
4.3	Чрезвычайные ситуации	88
4.3.1	Аварийные ситуации	88
4.3.2	Меры пожарной безопасности на рабочих местах	89
	Заключение	91
	Библиографические ссылки	92
	Библиографический список	93
	Приложение А	96

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей бакалаврской работе использованы ссылки на следующие стандарты и нормативные документы:

СП 2.2.3670-20. Санитарно-эпидемиологические требования к условиям труда.

СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания.

ГОСТ Р 50948-2001. Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности.

ГОСТ Р 50949-2001. Средства отображения информации индивидуального пользования. Методы измерений и оценки эргономических параметров и параметров безопасности.

ГОСТ 28406-89. Персональные электронные вычислительные машины. Интерфейсы видеомониторов. Общие требования

ГОСТ Р ИСО 1503-2014. Эргономика. Требования к пространственной ориентации и направлениям движения органов управления.

ГОСТ Р ИСО 9241-161-2016 Эргономика взаимодействия человек-система. Часть 161. Элементы графического пользовательского интерфейса.

ГОСТ 12.1.005-88 Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны

ГОСТ 12.0.003-2015 Система стандартов по безопасности труда. Опасные и вредные производственные факторы. Классификация.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ТЗ – Техническое задание

БД – База данных

ИС – Информационная система

АСОИУ – Автоматизированная система обработки информации и управления

СУБД – Система управления базами данных

АмГУ – Амурский государственный университет

RoR – Ruby on Rails

JS – JavaScript

API – Application Programming Interface

SQL – Structured Query Language

МЭШ – Московская электронная школа

ЭВМ – Электронно-вычислительная машина

ЯП – Язык программирования

УМУ – Учебно-методическое управление

CMS – Content management system

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

API (Application Programming Interface) – это набор программных инструкций и протоколов, предоставляемых разработчиками, для взаимодействия и интеграции между различными программами, сервисами или компонентами приложений. API определяет способы, как одна программа может взаимодействовать с другой, определяя доступные функции, данные и способы коммуникации между ними.

SQL (Structured Query Language) – это стандартный язык программирования, используемый для управления и взаимодействия с реляционными базами данных (РБД). SQL предоставляет набор команд и операторов, которые позволяют создавать, модифицировать и извлекать данные из базы данных. Он является декларативным языком, что означает, что пользователь описывает желаемый результат, а не конкретные шаги для его достижения.

IDEF1X (Integrated Definition for Information Modeling) – это методология моделирования данных, разработанная в рамках программы IDEF (Integrated Computer-Aided Manufacturing), созданной в США в конце 1970-х годов. IDEF1X предоставляет нотацию и правила для создания информационных моделей, которые служат основой для проектирования баз данных.

ВВЕДЕНИЕ

Цель преддипломной практики – разработать автоматизированную систему для распределения повышенных академических стипендий среди студентов Амурского государственного университета с учетом их достижений в ходе обучения в учебном семестре. Эта система будет представлена в виде модуля в личном кабинете университета и будет способствовать более справедливому распределению стипендий среди студентов на основе объективных критериев.

Для достижения поставленной цели проекта – создания автоматизированной системы для распределения стипендий повышенных академических стипендий среди студентов с учетом всех достижений, полученных в ходе обучения в учебном семестре в Амурском государственном университете - необходимо решить следующие задачи:

- исследование существующих способов и систем распределения стипендий в Амурском государственном университете. Это поможет определить преимущества и недостатки текущей системы, а также оценить лучшие практики;
- изучение требований и положений, связанных с распределением стипендий в университете. Это позволит определить основные критерии, на которые будет опираться новая система;
- разработка метода расчета стипендий, основанного на данных о нашей успеваемости и научной деятельности студентов. Это позволит создать более объективную систему распределения стипендий;
- создание удобного пользовательского интерфейса для системы, который будет легко понятен и удобен в использовании как для студентов, так и для комиссии. Это позволит сделать систему более привлекательной для пользователей;
- тестирование и отладка системы, включая учет возможных ошибок и нестандартных ситуаций. Это поможет убедиться в правильной работе системы и ее соответствии поставленным задачам.

Для решения поставленных задач можно использовать следующие методы:

– анализ существующих систем распределения стипендий в других университетах. Это позволит оценить преимущества и недостатки существующих решений и определить лучшие практики;

– интервьюирование администрации университета, преподавателей и студентов. Это позволит определить требования к системе распределения стипендий, выявить проблемы, связанные с текущей системой и оценить эффективность предлагаемой системы;

– моделирование системы распределения стипендий. Это позволит создать прототип системы и оценить ее работу на основе реалистичных данных;

– сравнительный анализ существующих интерфейсных решений для приложения. На основе сравнительного анализа выбрать лучшие практики для проекта, которые будут удобны и понятны для пользователей.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Обоснование актуальности темы квалификационного исследования.

В настоящее время разработка информационных и управляющих систем представляет собой эффективное решение для автоматизации множества задач, связанных с повторяющейся и монотонной деятельностью. В результате появления информационных систем были разработаны автоматизированные информационные системы (АИС), которые объединяют программные и аппаратные средства для автоматизации задач обработки, хранения и передачи различного вида информации.

Одним из основных принципов создания автоматизированной системы является системный подход, который предполагает рассмотрение предприятия или организации как единой целостной системы. Это включает анализ структурных элементов или бизнес-процессов, установление связей между ними и определение направлений информационных потоков, а также функций отдельных структурных подразделений.

Все эти задачи решаются путем постепенного совершенствования системы управления, повышения готовности используемого оборудования, улучшения условий труда операторов и обеспечения полной и достоверной информации о компонентах, используемых в деятельности, а также хранения информации в цифровом формате. Решение этих задач достигается через внедрение современных средств автоматизации в предприятия или организации.

Автоматизация бизнес-процессов является крайне важным аспектом для современных организаций, в том числе и для университетов. Автоматизация может значительно ускорить работу, сократить количество ошибок и повысить качество работы. В данном исследовании рассматривается разработка нового сервиса в личном кабинете Амурского Государственного Университета. Этот сервис предназначен для автоматизации распределения повышенной стипендии среди студентов.

Он позволяет хранить достижения студентов в цифровом виде в системе, что избавляет от необходимости собирать данные о своих наградах каждый раз, когда студент участвует в конкурсе на повышенную стипендию. Общая форма заявления, заполняемая по определенному алгоритму, автоматически генерируется для каждого студента, что исключает возможные ошибки и опечатки, связанные с человеческим фактором.

При выборе кандидатов на повышенную стипендию комиссия учитывает общий вес всех достижений и публикаций студентов. Однако, ручной анализ достижений каждого студента и составление рейтинговой таблицы занимают значительное количество времени. В связи с этим имеет смысл автоматизировать данный процесс. Автоматическая генерация рейтинговой таблицы с учетом суммарного веса всех достижений, внесенных студентами в систему, позволит комиссии только отобрать определенное количество кандидатов на повышенную стипендию.

Таким образом, данное исследование актуально с точки зрения создания информационной системы, которая автоматизирует распределение стипендий среди студентов, ускоряет процессы, сокращает количество ошибок и повышает эффективность работы университета. Кроме того, новая система может помочь университету в его усилиях в привлечении новых студентов и повышении качества образования.

1.2 Положении о стипендии и структура организации

1.2.1 Структура университета

Подсистема будет функционировать на базе федерального государственного образовательного учреждения высшего образования «Амурский государственного университета» (ФГБОУ ВО «АмГУ»). АмГУ является организацией высшего образования, осуществляющей в качестве основной цели ее деятельности образовательную деятельность по образовательным программам высшего образования и научную деятельность, созданной для осуществления образовательных, научных, социальных и иных функций некоммерческого характера. Структура данной учебной организации показана на рисунке 1, показана именно

та часть структуры, которая будет непосредственно взаимодействовать с системой.

В университете главенствующую роль играет ректор – высшая должностная особа образовательного учреждения. Ректор осуществляет текущее руководство деятельностью университета, за исключением вопросов, которые регулируются федеральными законами или уставом Университета и отнесены к компетенции Министерства, ученого совета Университета и других органов управления университетом. Он также действует от имени университета без необходимости предоставления доверенности и представляет университет в отношениях с юридическими и физическими лицами, а также органами государственной власти.

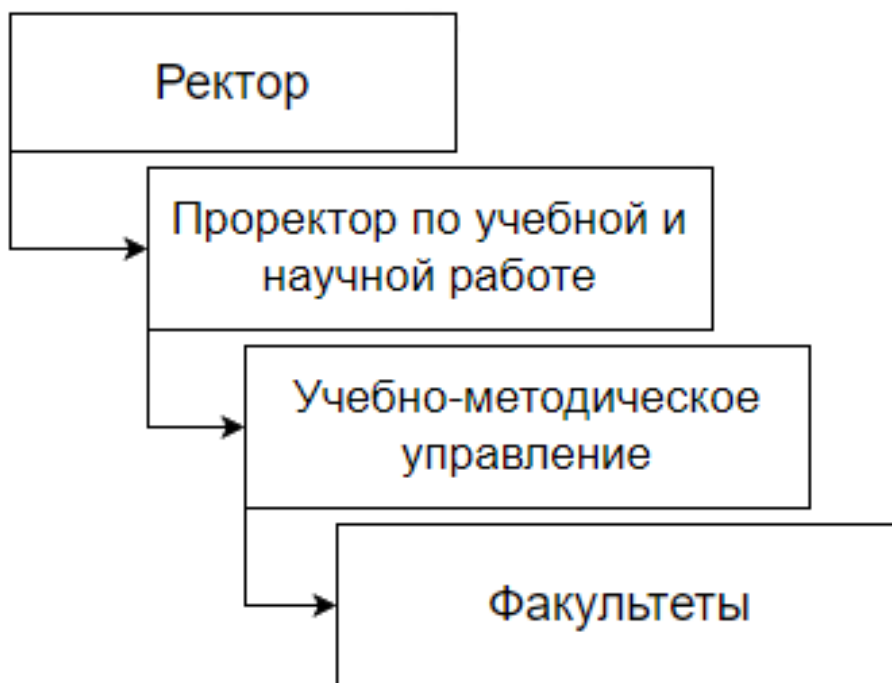


Рисунок 1 – Упрощенная структура университета

Поскольку ректору невозможно непосредственно контролировать все процессы в университете, задачи по учебной и научной работе возлагаются на проректора по учебной и научной работе. Проректор осуществляет надзор за следующими областями: организацией общей учебной и учебно-методической работы в университете, работой деканатов и кафедр в части организации и проведения учебной и учебно-методической работы, деятельностью учебного управления, а

также работой магистратуры и прочее.

Учебно-методическое управление (УМУ) находится в прямом подчинении проректора по учебной и научной работе. Главной целью УМУ является разработка и реализация мероприятий по повышению квалификации профессорско-преподавательского состава с учетом модернизации системы образования. Оно также обеспечивает условия для развития профессионального потенциала преподавателей, разрабатывает механизмы для повышения их мотивации и стимулирования профессионального роста.

Все вышеперечисленные функции проректора по учебной и научной работе и его подразделений направлены на обеспечение эффективной организации учебного процесса, повышение качества образования и научных исследований в университете.

В настоящее время обучение студентов в «АмГУ» осуществляется по сорока трем специальностям на десяти факультетах университета, факультеты показаны на рисунке 2.

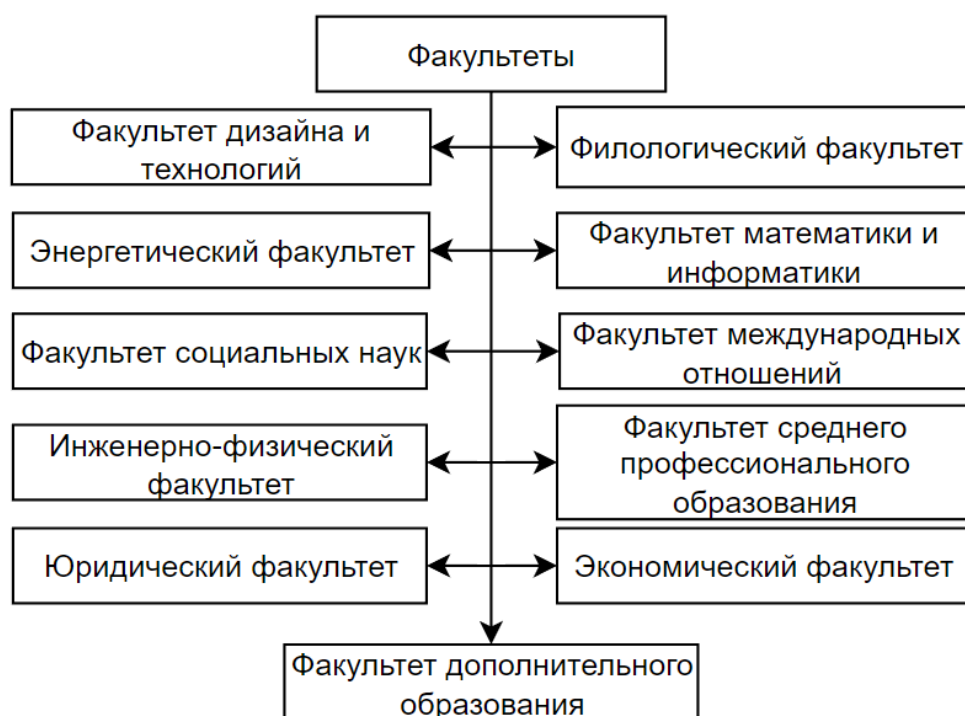


Рисунок 2 – Факультеты университета

В Амурском государственном университете (АмГУ) существует факультет дополнительного образования, который занимается подготовкой школьников

старших классов и абитуриентов к поступлению в университет, а также предоставляет возможности повышения квалификации и профессиональной переподготовки специалистов.

В рамках поощрения студентов в университете введены различные виды повышенных стипендий: именная стипендия, названная в честь М.Т. Луценко, повышенная государственная академическая стипендия, стипендия, названная в честь Н.Н. Муравьева-Амурского, президентская стипендия и стипендия ректора.

На каждом факультете присутствует заместитель декана по воспитательной работе. Они принимают участие в управлении учебно-воспитательной работой на факультете во время учебных и внеучебных мероприятий. Одной из их задач является формирование списка участников, претендующих на получение повышенных стипендий, а также организация работы стипендиальной комиссии факультета. Комиссия включает в себя декана факультета, заместителя декана и других представителей факультета, ответственных за рассмотрение заявок на стипендии и принятие решений относительно их назначения.

Такие меры способствуют поощрению студентов за их успехи и достижения, а также стимулируют активное участие в учебной и воспитательной деятельности в рамках университета.

Для понимания порядка проведения назначения повышенной академической стипендии, определение требующихся реквизитов и данных требуется обратиться к регламентируется положению от 15.06.2021 под номером 227-ОД «Положение о порядке назначения и выплаты, повышенных государственных академических стипендий студентам».

1.2.2 Положение о повышенных стипендий

Порядок и проведение назначения повышенной стипендии регламентируется положением от 15.06.2021 под номером 227-ОД «Положение о порядке назначения и выплаты, повышенных государственных академических стипендий студентам».

По положению стипендия – это денежная выплата, назначаемая обучаю-

щимся по основанной образовательной программе в целях стимулирования и поддержки освоения ими содержания соответствующих образовательных программ [1].

Студенты – лица, в установленном порядке зачисленные в высшее учебное заведение для обучения.

Обучение за счет средств федерального бюджета – обучение за счет средств субсидии на иные цели, финансируемые из федерального бюджета, далее – бесплатное обучение.

Численность студентов, получающих повышенную государственную академическую стипендию, не может составлять более 10 процентов от общего числа студентов, получающих государственную академическую стипендию.

По положению прием документов для конкурсного отбора осуществляется 2 раза в год (до 10 июля и до 10 февраля).

Повышенная стипендия назначается за достижения в следующих областях:

- учебной;
- научно-исследовательской;
- общественной;
- культурно-творческой;
- спортивной деятельности назначается,

студентам (программам бакалавриата, программам специалитета, программам магистратуры).

Приоритет при назначении повышенной стипендии получает студент, имеющий:

- достижение более высокого статуса и уровня;
- несколько достижений по виду деятельности;
- достижения по нескольким видам деятельности.

Для рассмотрения вопроса о выдвижении кандидатур на назначение повышенной стипендии в комиссию представляются следующие документы, подтверждающие достижения кандидата:

- ксерокопия зачетной книжки за два семестра, предшествующих назначе-

нию стипендии, заверенная в деканате с подтверждающей записью об от отсутствии пересдач зачетов и экзаменов для студентов (2 – 5) курсов обучения,

– заявление на участие в конкурсе, заполненная форма, образец показан на рисунке 3;

– подлинники документов, удостоверяющих достижения кандидатов (грамоты и дипломы призеров, сертификаты участников и иные документы, подтверждающие достижения и победы в различных видах деятельности и т.п.);

– заверенные копии документов, подтверждающие спортивные разряды и звания;

– документы, подтверждающие получение грантов на выполнение научно-исследовательских работ, участие в хоздоговорных работах;

– список научных трудов, публикаций, публичных выступлений, документальные свидетельства общественной деятельности, заверенные заместителем декана по научной работе.

Форма заявления на участие в конкурсе

АМГУ
Факультет
Заявление
Дата

на имя кого

Прошу допустить меня к участию в конкурсе на получение повышенной государственной академической стипендии за достижения в _____

_____ деятельности

(указать вид деятельности: учебная, научно-исследовательская, общественная, культурно-творческая, спортивная)

в ____ семестре 20__\20__ учебного года.

Получаю _____.

(указать вид стипендии: стипендия Президента РФ, стипендия Правительства РФ, именная стипендия и пр.)

С «Положением о порядке назначения и выплаты, повышенных государственных академических стипендий студентам» ознакомлен (-а).

К заявлению прилагаю подтверждающие документы с описанием: (копии свидетельств, грамот, дипломов, копии документов, подтверждающие разряды и звания, список научных трудов, публикаций, публичных выступлений, примеров общественной деятельности, список участия в научно-практических конференциях и иные документы, подтверждающие достижения, перечисленные в настоящем положении)

№	Наименование	Кол-во страниц
1		
2		
3		
4		
5		
...		
n		

Студент

(подпись)

(И.О. Фамилия)

Рисунок 3 – Форма заявления на участие в конкурсе

Для определения рейтинга студентов по достижениям используется балльная оценка, позволяющая сопоставить достижения студентов в комплексе. При равенстве баллов за достижения преимущество получает студент, имеющий более высокий рейтинг академической успеваемости, а при равенстве и академических рейтингов - студент, обучающийся на старших курсах. Все достижения студентов должны быть отражены в личном кабинете. Количество баллов за каждое достижение представлено в приложение 2 данного положения.

Порядок назначения повышенной стипендии регламентируется положением следующим образом. Формирование списка кандидатур на получение повышенных стипендий осуществляется в результате работы стипендиальной комиссии факультета, далее – комиссия. К рассмотрению комиссией принимаются документы кандидатов, представленные лично студентами. Заседания комиссии проводятся под руководством председателя комиссии и считаются правомочными при условии присутствия не менее 2/3 членов. Все решения комиссии принимаются открытым голосованием простым большинством голосов от числа присутствующих членов. Решение комиссии оформляется протоколом, который подписывает председатель и члены комиссии. Решение комиссии доносится до всех участников конкурса.

Во время проектирования подсистемы данное положение будет являться основным документом для определения функционала и атрибутов сервиса.

Поскольку подсистема будет представлять собой веб-приложение, полная стек разработка будет осуществляться с помощью фреймворка. Определимся в выборе фронт- и бэкенд части веб-сайта, веб-сервером и сервером базы данных.

1.2.3 Информационная структура университета

Организация уже обладает функционирующей информационной системой, состоящей из двух компонентов: публичной и корпоративной. Публичная часть системы, доступная по адресу amursu.ru, была разработана при помощи CMS-системы «1С-Битрикс». Корпоративная часть, расположенная по адресу cabinet.amursu.ru, была создана с использованием фреймворка «Ruby on Rails» и графического интерфейса «Semantic UI».

Модуль «Повышенная стипендия» будет связан исключительно с корпоративной частью информационной системы. Следовательно, выбор фреймворка, веб-сервера, базы данных и системы управления базами данных будет зависеть от уже используемых технологий в организации. В данном случае, в качестве бэкенд-части будет использован фреймворк «Ruby on Rails», веб-сервером – «Nginx», а сервером баз данных и системой управления базами данных – «PostgreSQL».

Таким образом, для реализации модуля "Повышенная стипендия" будут использованы следующие технологии: «Ruby on Rails» в качестве фреймворка, «Nginx» в качестве веб-сервера, и «PostgreSQL» в качестве сервера баз данных и системы управления базами данных.

«RoR», также известный как "Ruby и Rails", представляет собой фреймворк с многоуровневой архитектурой «Model-View-Controller» (MVC), показанной на рисунке 4. Он предназначен для разработки веб-приложений, которые могут использовать как реляционные, так и NoSQL базы данных, такие как «MySQL», «MariaDB», «PostgreSQL» или «MongoDB». Фреймворк написан на языке программирования Ruby.

«RoR» подходит для создания как обычных веб-сайтов, которые должны быть быстрыми, надежными и способными выдерживать высокую нагрузку, так и для веб-приложений с комплексной бизнес-логикой и динамическими веб-интерфейсами. «Ruby on Rails» является открытым программным обеспечением и распространяется под лицензией MIT.

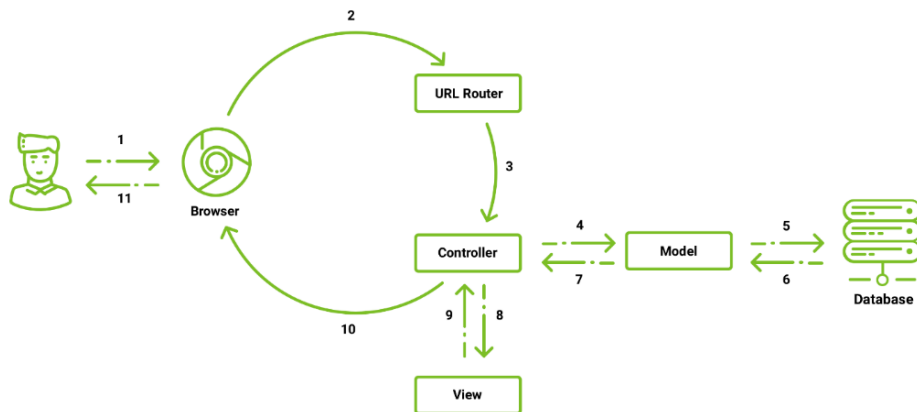


Рисунок 4 – Архитектурный шаблон Model-View-Controller

«Nginx»— это веб-сервер, который также может использоваться в качестве обратного прокси-сервера, почтового прокси-сервера и общего прокси-сервера для протоколов TCP/UDP. Упрощенная схема его работы представлена на рисунке 5.

Основная функциональность HTTP-сервера «Nginx» включает:

– обслуживание статических запросов, включая обработку индексных файлов, автоматическое формирование списка файлов и кэширование дескрипторов открытых файлов;

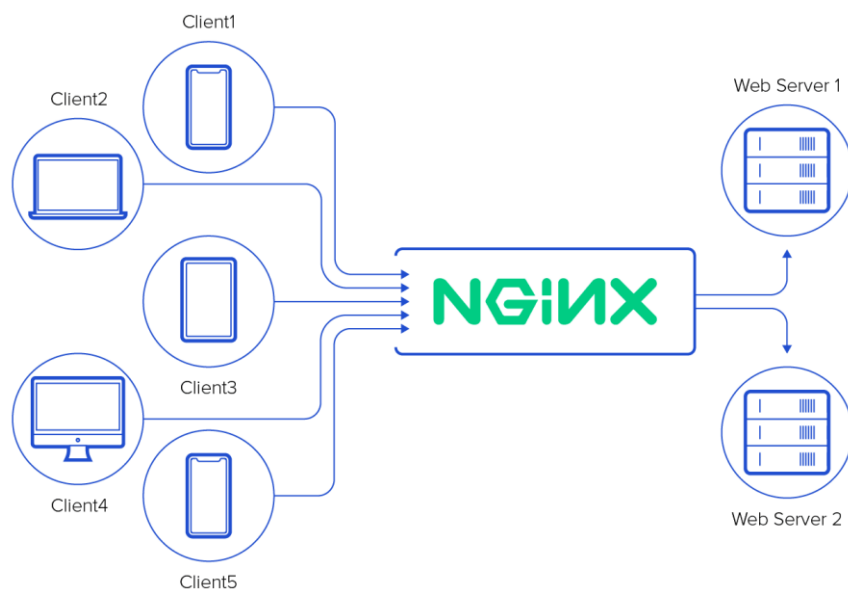


Рисунок 5 – Упрощенная схема работы «Nginx»

– реализация обратного проксирования с кэшированием, распределением нагрузки и обеспечением отказоустойчивости;

– поддержка ускоренной работы с серверами FastCGI, uwsgi, SCGI и memcached, включая кэширование, распределение нагрузки и обеспечение отказоустойчивости;

– модульность и наличие фильтров, таких как сжатие (gzip), поддержка докачки файлов (byte-ranges), передача данных в формате chunked, XSLT-фильтр, SSI-фильтр, возможность обработки нескольких подзапросов на одной странице, которые могут выполняться параллельно через прокси или FastCGI/uwsgi/SCGI;

– поддержка SSL и расширений TLS SNI;

– поддержка протокола HTTP/2 с возможностью приоритизации на основе весов и зависимостей;

– дополнительная функциональность в качестве почтового прокси-сервера, включающая перенаправление пользователя на IMAP- или POP3-сервер с использованием внешнего HTTP-сервера для аутентификации, а также проверку пользователя через внешний HTTP-сервер для аутентификации и перенаправление соединения на внутренний SMTP-сервер.

«PostgreSQL» – это свободно распространяемая система управления базами данных (СУБД) с объектно-реляционными возможностями (ORDBMS). Она является наиболее развитой открытой СУБД и пример ее использования показан на рисунке 6.

Преимущества «PostgreSQL» включают:

- поддержка баз данных неограниченного размера;
- мощные и надежные механизмы транзакций и репликации;
- возможность расширения с помощью встроенных языков программирования и поддержки загрузки модулей, совместимых с языком Си;
- использование наследования для более гибкой организации данных;
- легкая расширяемость системы.

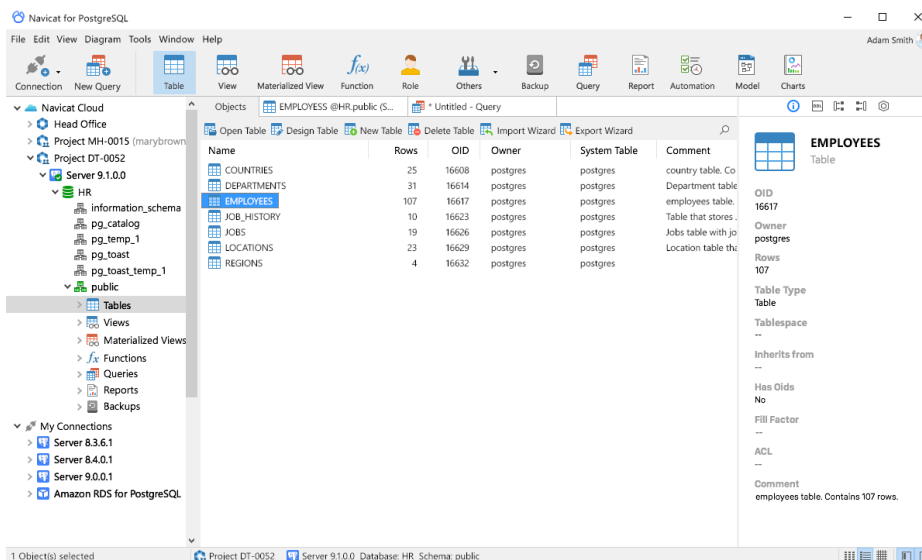


Рисунок 6 – СУБД «PostgreSQL»

Особенности «PostgreSQL»:

- нет ограничений на количество записей в таблице;
- нет ограничений на максимальный размер базы данных;
- нет ограничений на количество индексов в таблице;

- максимальный размер таблицы – 32 Тбайт;
- максимум полей в записи 250 – 1600 (в зависимости от типов полей);
- максимальный размер записи – 1,6 Тбайт;
- максимальный размер поля – 1 Гбайт.

Бэкенд фреймворк обеспечивает высокую производительность на уровне сервера и доступ к базе данных. Однако, поскольку веб-сайт предназначен для одновременного использования большого числа пользователей, возникает потребность в распределении нагрузки с сервера на персональные устройства клиентов. Для этого требуется возможность хранить часть состояния и данных системы на стороне персональных компьютеров пользователей и обновлять их только при необходимости, создавая иллюзию бесшовной работы системы. Этот подход к разработке веб-приложений известен как «Single Page Application» (SPA) – метод взаимодействия с пользовательским интерфейсом, при котором используется только один «HTML» документ. Взаимодействие с пользователем осуществляется через динамически загружаемые «HTML», «CSS» и «JavaScript» с использованием технологии «AJAX», как показано на рисунке 7. В ИС университета данный подход к организации внешнего интерфейса для работы с системой, совместно с созданием статичных страниц с использованием фреймворка «Ruby on Rails», успешно реализован.

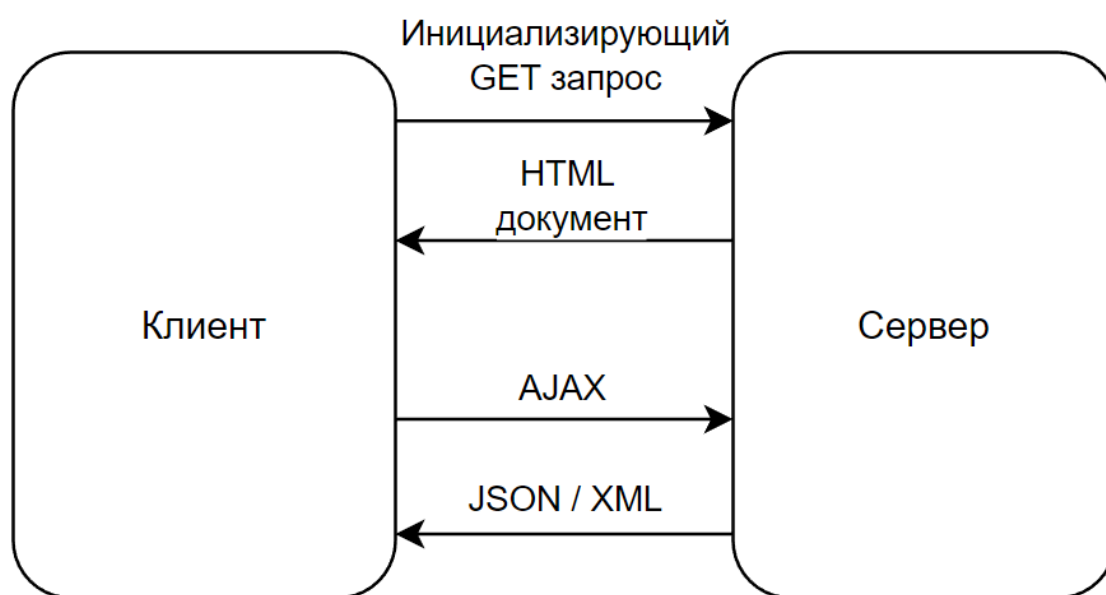


Рисунок 7 – Упрощенный принцип SPA

Для реализации SPA в информационной системе организации был выбран популярный фреймворк для фронтенда – «Vue.js».

«Vue.js» является прогрессивным фреймворком, предназначенным для разработки пользовательских интерфейсов. В отличие от монолитных фреймворков, «Vue.js» разработан таким образом, чтобы его можно было постепенно внедрять в проекты. Основной упор в «Vue.js» делается на решение задач уровня представления (view), что облегчает интеграцию с другими библиотеками и существующими проектами. Одновременно «Vue.js» полностью подходит для создания сложных одностраничных приложений, особенно в сочетании с современными инструментами и дополнительными библиотеками.

Одним из ключевых понятий в «Vue.js» являются компоненты. Эта абстракция позволяет создавать большие приложения из небольших "кусочков". Компоненты представляют собой объекты, которые могут быть повторно использованы. С точки зрения структуры интерфейса, практически любой пользовательский интерфейс можно представить в виде дерева компонентов, как показано на рисунке 8. В «Vue.js» компоненты являются экземплярами «Vue.js» с предустановленными опциями.

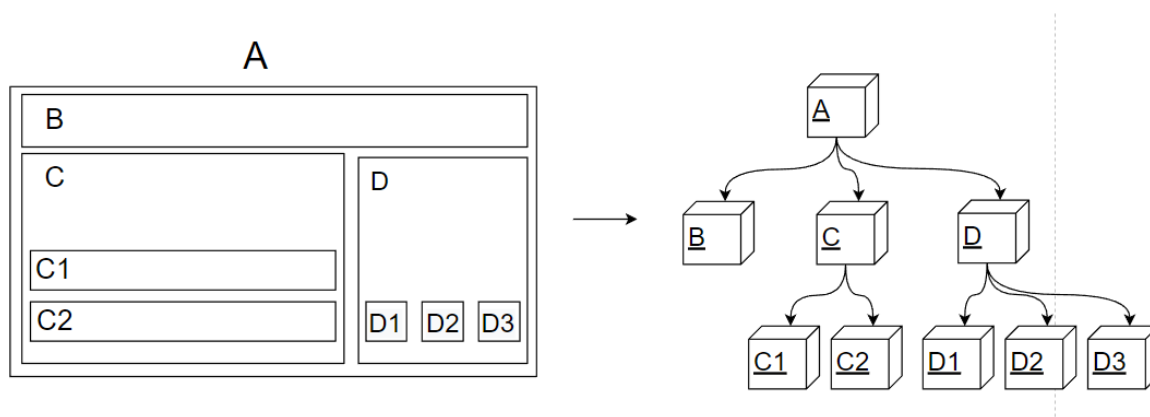


Рисунок 8 – Упрощенный принцип компонентного подхода

Указанный стек технологий является основным для информационных систем личного кабинета «АмГУ», и его следует учитывать при проектировании и разработке модуля, а также при добавлении дополнительных решений.

1.3 Обзор существующих методов решения рассматриваемой задачи

Решение, предоставляемое «1С Университет», предназначено для автома-

тизации основных процессов высшего учебного заведения, включая учет, хранение, обработку и анализ информации. Оно позволяет эффективно управлять такими процессами, как поступление в вуз, обучение, оплата за обучение, выпуск и трудоустройство выпускников, расчет и распределение нагрузки профессорско-преподавательского состава, а также деятельность учебно-методических отделов и деканатов. Решение также обеспечивает поддержку ФГОС-3 и уровневой системы подготовки, включая бакалавриат, специалитет и магистратуру, путем создания учебных планов и документов государственного образца, связанных с окончанием вуза.

Данное решение может быть применено для автоматизации рабочих мест сотрудников различных структурных подразделений вуза, включая приемную комиссию, деканаты, кафедры, учебно-методический отдел, бухгалтерию, студенческий отдел кадров и профсоюзный комитет.

Такое программное решение обладает значительными преимуществами, позволяя упростить и автоматизировать рутинные операции, повысить эффективность работы сотрудников и обеспечить более точный и надежный учет и анализ данных.



Рисунок 9 – «1С Университет ПРОФ»

Московская электронная школа (МЭШ) – это проект, созданный в Москве для обеспечения доступного и качественного образования с использованием информационных технологий, рисунок 10.

МЭШ предлагает широкий спектр образовательных программ для детей и

взрослых. В школе используются современные технологии, такие как онлайн-курсы, интерактивные уроки, видеолекции, вебинары и другие формы дистанционного обучения.

Программы МЭШ охватывают различные предметы и области знаний, включая математику, физику, информатику, робототехнику, иностранные языки и многое другое. Обучение осуществляется опытными преподавателями, которые разрабатывают и адаптируют образовательные материалы для онлайн-формата.

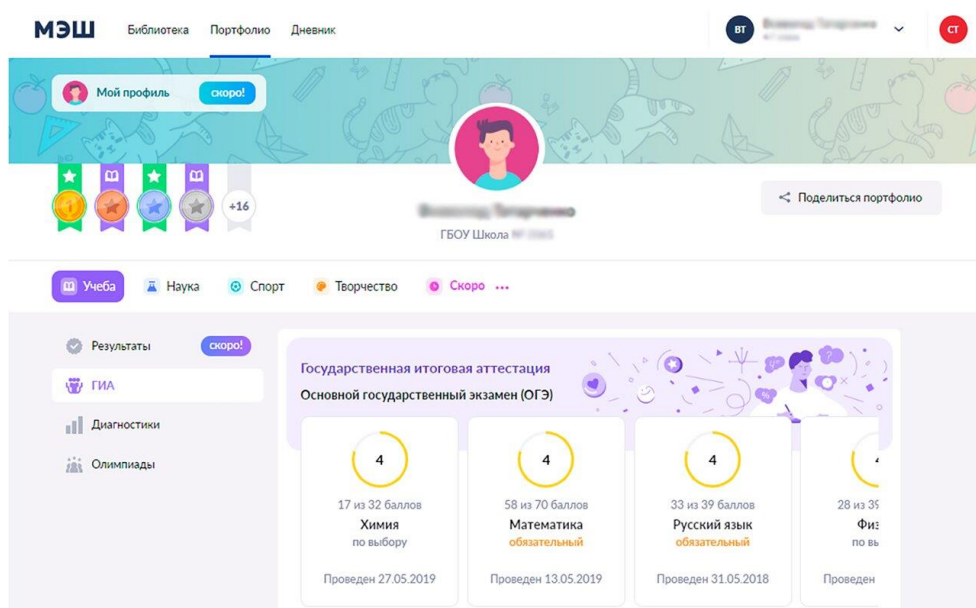


Рисунок 10 – Портфолио МЭШ

Московская электронная школа (МЭШ) предоставляет широкий спектр функциональности для управления и представления учебного процесса и достижений учащихся. Вот некоторые основные возможности портфолио системы МЭШ:

Управление учебными материалами: Портфолио МЭШ позволяет ученикам и учителям загружать и хранить различные учебные материалы, такие как задания, работы, проекты и презентации. Это обеспечивает удобный доступ к материалам и возможность оценки их качества.

Оценка и обратная связь: Учащиеся могут получать оценки и обратную связь от учителей по своим учебным материалам и достижениям. Это помогает им отслеживать свой прогресс и улучшать свои навыки.

Планирование и учет академических достижений: Портфолио МЭШ позволяет учащимся устанавливать учебные цели, составлять планы действий и отслеживать свои достижения в различных областях знаний. Они могут также включать свои успехи внеклассной деятельности, спорта и других областей.

1.4 формулировка задачи исследования и общей методики ее решения

Сформулировать задачу можно следующим образом, требуется программа, предназначенная для автоматизированной подачи заявлений студентами на конкурс получения повышенной академической стипендии. Заявление подается по заранее заполненной портфолио, находящиеся в личном кабинете «АмГУ». Для каждого студента будет возможность просмотреть таблицу первых 25, 50, 80 студентов, участвующих в конкурсе. Для человека человек с задачей, организации комиссии по повышенной академической стипендии, должна быть предусмотрена возможность выбора из рейтинга, составляемого из всех участников, студентов факультета, выбрать нужное количество обучающихся для выдачи повышенной стипендии. Разберем подробнее требования к программе со стороны основных действующих лиц.

Со стороны студента рассматриваются действия по внесению полученных достижений в учебной и вне учебной деятельности, научные работы, достижения в области спорта и в соревнованиях регионального и федерального уровня, а также просмотр рейтинга участников, претендующих на повышенную стипендию.

Требуется реализация следующих модель действий действующего лица, студента, в программе:

- переход в раздел «портфолио» в личном кабинете;
- добавление достижений, документов, научных работ и отзывов, если таковые имеются;
- редактирование внесенных ранее данных;
- просмотр внесенных данных, разделенных по разделам;
- переход в категорию «Повышенная стипендия» для просмотра рейтинга по факультету участника.

Для внесения данных в систему следует предусмотреть интерактивный элемент, после нажатия которого, должна появиться форма для заполнения следующих полей:

- поле с названием достижения;
- поле с категорией достижения;
- поле с видом достижения или научной публикации;
- поле с призовым местом, если таковое присутствует в достижении;
- поле с датой, указанной в достижении или дата публикации научной работы;
- поле описания достижения или научной работы;
- поле для добавления файла в формате pdf, подтверждающего достижение или же копия распечатанного достижения;
- ссылка на достижение, если такая возможность имеется.

После внесения достижений должна иметься возможность просмотра их в разделе «портфолио». Для удобства виды достижений должны быть разделены на разделы:

- общие сведения;
- портфолио достижений;
- портфолио документов;
- рейтинг.

В разделе «Повышенная стипендия», в личном кабинете, требуется возможность конкурсантом, просмотреть остальных участников своего факультета и свое место в рейтинге. Таблица отсортирована по принципу рейтинга, так первый студент в рейтинге претендует на первое место для получения стипендии. Для расчёта общей суммы баллов по внесенным данным конкурсанта в разделе «портфолио» требуется назначить вес для каждого достижения и научной работы баллы, которых указаны в приложении положения «ПУД СМК 145-2021».

Оператор, занимающийся выбором обучающихся для получения повышенной стипендии, должен иметь следующие возможности в системе:

- просмотр списка студентов на повышенную академическую стипендию;

- анализ и проверка каждого достижения (подразумевается все введенные данные в разделе «портфолио») на наличие ошибок или намеренно неправильно введенных данных;

- выбор участника в число тех, кто проходит по всем критериям и будет получать повышенную стипендию;

- отклонение участника от конкурса на повышенную академическую стипендию;

- формирование списка всех конкурсантов для получения повышенной стипендии;

При отклонении конкурсанта от участия, оператор должен описать по каким причинам данный участник был отстранен от участия на повышенную стипендию.

В случаи отстранения, конкурсант должен быть уведомлен об отстранении с описанием причины отказа о выдачи повышенной стипендии. Данная функция предназначена для тех случаев, когда имеется подозрение на намеренно введенные ложные данные с целью получения более высокого суммарного балла участником, поэтому данный конкурсант должен быть исключены из рейтинга для освобождения места в таблице для следующих участников.

После формирования списка студентов, оператор должен иметь возможность сформировать автоматически для каждого участника файлы для печати, состоящий из:

- автоматически заполненной формы заявления для участия в конкурсе;
- всех достижений, прикрепленных студентом в своей портфолио в формате pdf.

2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ АСОИУ

2.1 Средства разработки программного обеспечения

Для определения наиболее подходящего языка программирования в контексте веб-разработки из рассмотренных альтернатив, а именно «Python», «Ruby», «Node.js», «PHP», «Go» и «C#», проведен анализ по нескольким критериям, указанных в таблице 1. Каждый критерий был оценен по шкале от 1 до 5, где 1 представляет низкий уровень, а 5 – высокий уровень.

Критерии, учтенные в анализе, включают синтаксис и читаемость, производительность, экосистему и сообщество, масштабируемость, удобство разработки, безопасность и удобство для веб-разработки.

После проведения анализа, были подсчитаны суммы оценок для каждого языка, что позволяет сравнить их и определить предпочтительный язык программирования.

Таблица 1– Анализ языков программирования

Критерий	Python	Ruby	Node.js	PHP	GO	C#
1	2	3	4	5	6	7
Синтаксис и читаемость	4	5	4	3	4	4
Производительность	3	3	4	2	4	3
Экосистема и сообщество	5	5	4	5	4	4
Масштабируемость	5	4	5	4	5	4
Удобство разработки	5	4	5	5	4	4
Безопасность	4	5	2	2	4	4
Веб-разработка	5	5	5	5	3	4
Сумма оценок	31	31	29	26	28	27

Исходя из оценок, полученных по каждому критерию, можно сделать следующие выводы, «Ruby» и «Python» получили наивысшие суммарные оценки «31», что указывает на их превосходство по сравнению с другими языками. Так как система личного кабинета была разработана на языке «Ruby», то очевидным выбором для разработки подсистемы будет именно этого ЯП.

Для выбора фронтенд Framework'а проведем анализ самых популярных из них в виде таблицы 2 и определим какой из них следует выбрать для создания клиентской части подсистемы.

Таблица 2 – Анализ фронтенд Framework'ов

Критерии	Vue.js	Angular	Svelte	React
1	2	3	4	5
Популярность	4	5	3	5
Сообщество	4	5	3	5
Документация	5	4	3	4
Производительность	4	3	4	4
Гибкость	5	3	4	4
Обучаемость	4	3	4	3
Экосистема	4	4	2	5
Всего	30	27	23	30

Исходя из таблицы N, фреймворки «Vue.js» и «React» имеют одинаковую сумму баллов, равную 30-и. Оба фреймворка, таким образом, равнозначны по оценкам. Так как некоторые модули личного кабинета были уже написанные на «Vue.js», то для создания подсистемы следует выбрать именно «Vue.js».

В качестве СУБД для работы с базой данной системы были рассмотрены следующие системы: «PostgreSQL», «Microsoft SQL» «Server», «MySQL», таблица 3.

Таблица 3 – Анализ СУБД

Критерии	PostgreSQL	MS SQL Server	MySQL
1	2	3	4
Функциональность	5	4	4
Производительность	4	5	4
Масштабируемость	4	5	4
Сообщество	5	4	4
Поддержка	5	4	4
Стоимость	4	3	5
Всего	27	25	25

Исходя из анализа очевидным выбором СУБД для разработки подсистемы

будет «PostgreSQL», также дополнительным критерием выбора к данной СУБД является его использование в системе личного кабинета.

2.2 Разработка концепции архитектуры построения и платформы реализации информационных систем

UML – это стандартизированный язык моделирования, широко используемый для проектирования программных продуктов. UML предоставляет набор диаграмм, которые помогают визуализировать и описать различные аспекты системы. Ниже перечислены основные типы диаграмм UML, которые часто используются при проектировании программных продуктов:

Диаграмма вариантов использования, «Use Case Diagram». Диаграмма вариантов использования помогает описать функциональность системы из точки зрения ее актеров (пользователей). Она иллюстрирует взаимодействие между актерами и системой, а также показывает различные сценарии использования и их связи с функциональностью системы.

Диаграмма классов, «Class Diagram». Диаграмма классов представляет структуру системы, показывая классы, атрибуты, методы и связи между ними. Она помогает описать структуру данных и отношения между классами, а также иерархию наследования.

Диаграмма последовательности, «Sequence Diagram». Диаграмма последовательности показывает взаимодействие объектов в определенной последовательности. Она демонстрирует, как объекты обмениваются сообщениями друг с другом в рамках определенного сценария или функции системы.

Диаграмма состояний, «State Diagram». Диаграмма состояний описывает поведение объекта или системы в различных состояниях и переходах между ними. Она помогает моделировать дискретные состояния и условия, которые влияют на поведение системы в различных ситуациях.

Диаграмма компонентов, «Component Diagram». Диаграмма компонентов показывает физическую структуру системы и ее компоненты, такие как библиотеки, модули или сервисы, а также связи между ними. Она помогает понимать архитектуру системы и компонентное разделение.

Это только некоторые из основных диаграмм UML, и существуют и другие типы, такие как диаграмма развертывания, диаграмма активности и диаграмма объектов. Комбинируя различные диаграммы UML, разработчики могут создавать более полные и понятные модели системы, которые помогают в процессе проектирования программных продуктов. Все диаграммы будут выполнены в ПО «ArgoUML».

Начнем проектирование системы с диаграммы прецедентов, указанной на рисунке 11, определим основные действующие лица подсистемы и варианты использования.

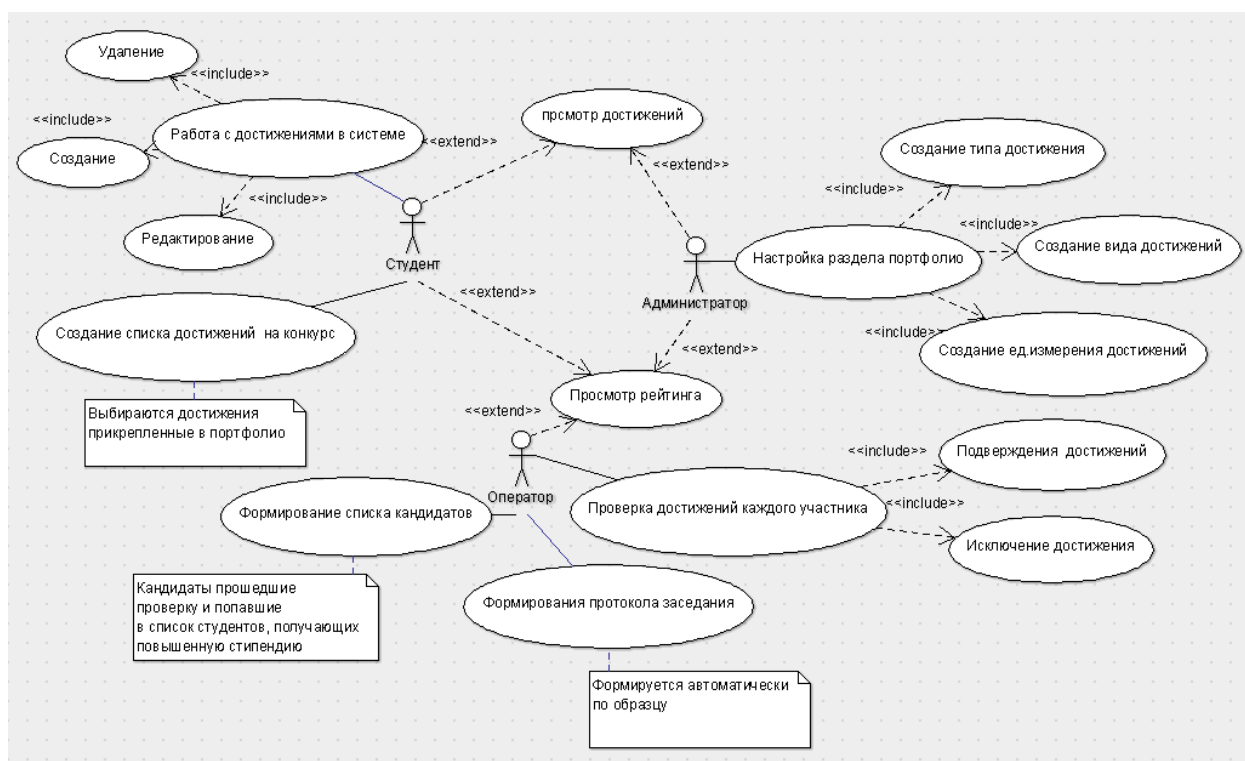


Рисунок 11 – Диаграмма прецедентов

В качестве действующих лиц (актеры) выступают:

- студент, прикрепляет, редактирует, удаляет достижения в портфолио, подает заявку на участие в конкурсе на повышенную стипендию;
- администратор, проводит создание, редактирование, удаление видов и типов достижений, проводит настройку единиц измерения достижений, администраторами могут выступать как IT отдел «АмГУ», так и сотрудники УМУ;
- оператор, к оператору относится диспетчер по работе с протоколом, и стипендиальная комиссия, основными функциями оператора выступают: созда-

ние и редактирование протокола, печать протокола, выбор студентов на повышенную стипендию.

Представленная на рисунке 12 диаграмма последовательности иллюстрирует процесс прикрепления нового достижения студента в его портфолио. Для успешного прикрепления достижения требуется выполнить несколько процессов.

Первоначально студент должен назвать достижение, которое планирует добавить в свое портфолио. Далее, студент должен выбрать тип деятельности из представленных в положении, определяющем доступные варианты. Кроме того, необходимо выбрать вид достижения также из предложенных в положении.

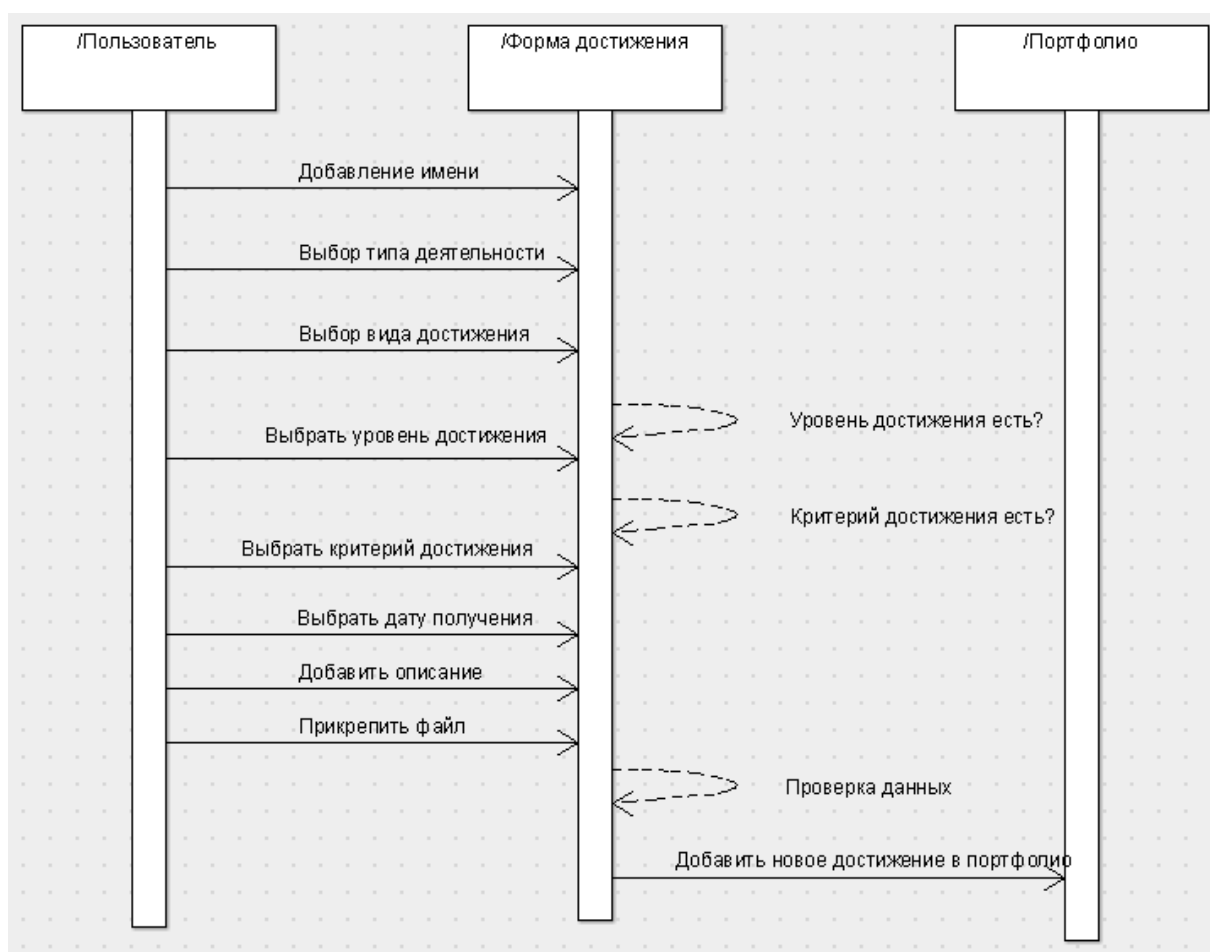


Рисунок 12 – Диаграмма последовательности создания достижения

В случае, если достижение имеет единицы измерения с уровнями, студент должен также выбрать соответствующий уровень достижения. Если уровень содержит дополнительные критерии, студенту требуется выбрать их из предложенных вариантов. Затем студент должен указать дату получения достижения, доба-

вить описание, если это необходимо, и прикрепить файл, связанный с достижением.

После завершения этих шагов происходит проверка введенных данных на корректность. В случае успешной проверки достижение прикрепляется к личному портфолио студента. Этот процесс представляет собой последовательную цепочку действий, которая гарантирует систематичность и структурированность при добавлении нового достижения в портфолио студента.

Важным аспектом сервиса является подача заявления с учетом достижений студента в портфолио, для этого следует разработать последовательность действий для создания электронного заявления. Данная последовательность представлена на рисунке 13.

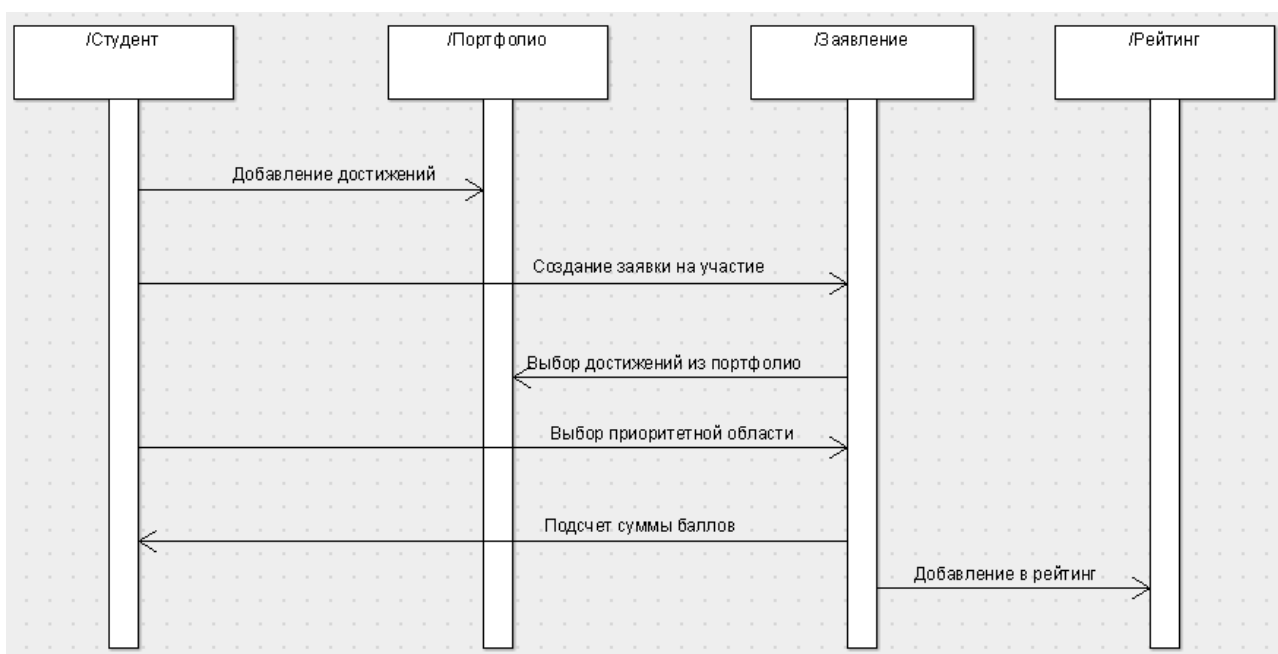


Рисунок 13 – Диаграмма последовательности создания заявки

Первоначально студенту предоставляется список достижений, содержащихся в его портфолио. Он должен проанализировать эти достижения и выбрать те, которые считает наиболее значимыми и релевантными для рассмотрения своего заявления комиссией. Этот выбор будет влиять на итоговую оценку его заявления.

Кроме выбора достижений, студенту необходимо указать приоритетный тип деятельности, который отражает его основные области интересов и активно-

сти. Это помогает комиссии лучше понять профиль студента и оценить его ценность и вклад в соответствующую сферу.

После того, как студент сделал выбор достижений и указал приоритетный тип деятельности, система подсчитывает суммарный балл, основываясь на заранее установленных критериях и весовых коэффициентах, связанных с каждым выбранным достижением.

После того, как студент подтвердит свой выбор и завершит ввод данных, его оценки и достижения будут учтены при формировании рейтинга его факультета.

На рисунке 14 показана диаграмма последовательности для создания протокола заседания стипендиальной комиссии.

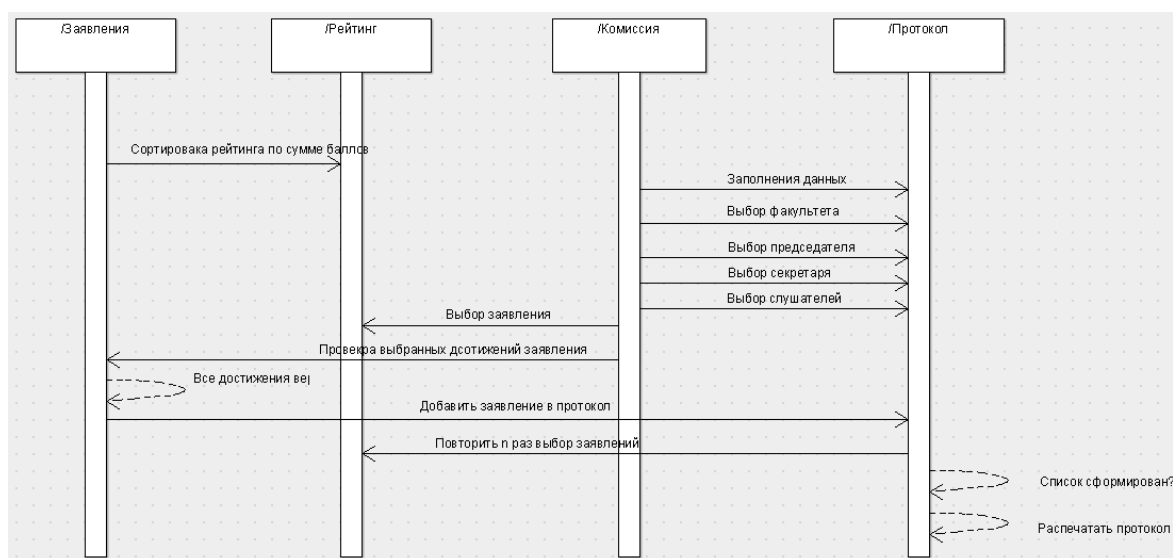


Рисунок 14 – Диаграмма последовательности создания

На диаграмме показаны основная последовательность действий при работе с протоколом. Так для начала создания протокола, оператору должна быть предоставлена рейтинговая таблица по факультету, для выбора студентов в кандидаты на повышенную стипендию. Оператору требуется внести основную информацию в протокол: выбрать председателя, секретаря и слушателей. После из рейтинга отобрать нужное количество студентов и прикрепить их к протоколу, перед этим проверив достижения каждого прикрепляемого студента. Как только нужное количество студентов выбрано и список согласован стипендиальной комиссией, можно распечатать протокол.

2.3 Структура информационных систем, состав функциональных и обеспечивающих подсистем

Диаграмма классов является инструментом для определения типов классов системы и статических связей между ними. Она также позволяет отобразить атрибуты классов, операции классов и ограничения, которые применяются к связям между классами.

При разработке системы, в которой требуется взаимодействие с базой данных, рекомендуется выбрать решение, которое предусматривает взаимодействие с БД с использованием принципа «ORM» (Object-Relational Mapping). Использование «ORM» позволяет упростить разработку, так как классы системы отражают сущности в базе данных. На рисунке 15 представлена диаграмма классов сущностей.

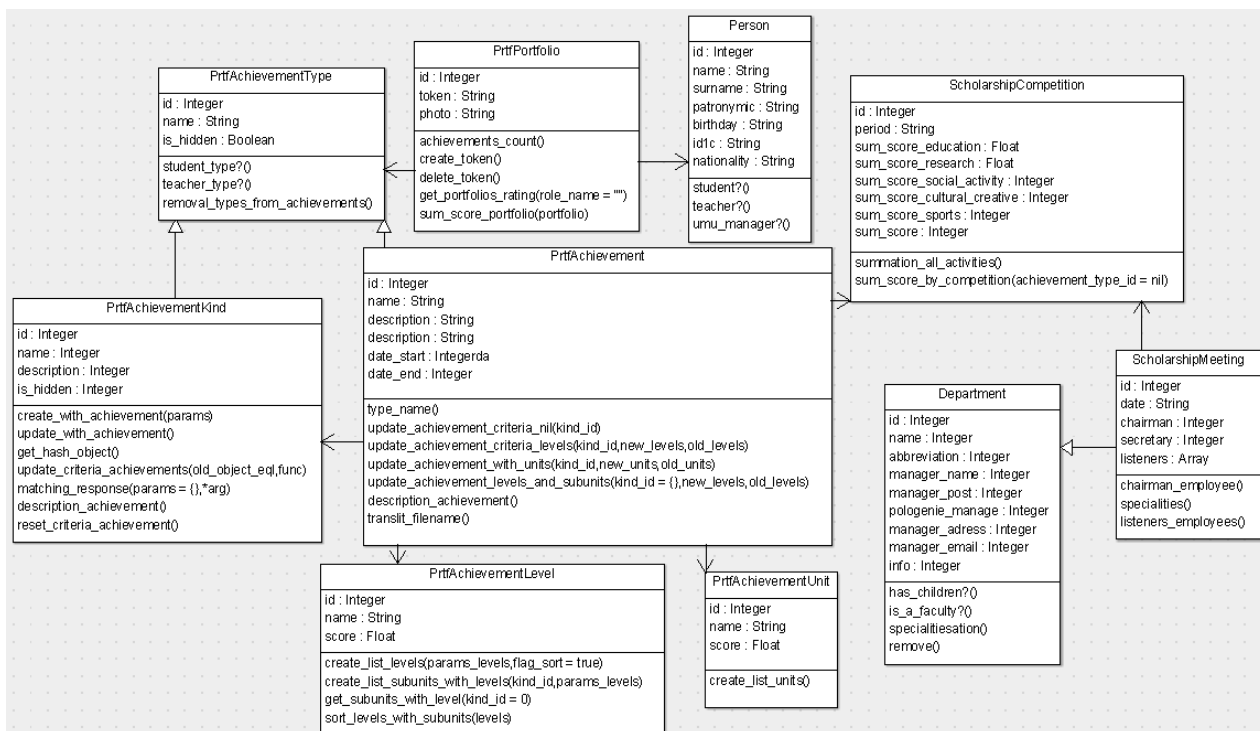


Рисунок 15 – Диаграмма классов подсистемы

Важно отметить, что все классы, указанные на диаграмме, будут наследоваться от общего родительского класса, например, в случае использования фреймворка «Ruby on Rails», этим родительским классом может быть класс «ActiveRecord», который предоставляет интерфейс методов для работы с базой данных.

Вспомогательный скрипт в рамках фреймворка «Rails» используется для автоматического создания структуры каталогов в приложении. Он является неотъемлемой частью инициализации нового проекта и обладает способностью самостоятельно определить расположение необходимых компонентов в этой структуре. Благодаря этой функциональности, пользователю не требуется вручную вводить или настраивать пути и расположение файлов, необходимых для правильной работы приложения.

Вспомогательный скрипт «Rails» умеет распознавать и создавать каталоги и подкаталоги, которые обычно присутствуют в типичном приложении. Это включает в себя каталоги для моделей, контроллеров, представлений, миграций базы данных, стилей, скриптов и других компонентов, необходимых для разработки и функционирования веб-приложения. Упрощенная структура каталога представлена на рисунке 16.

Каталог «cabinet» в организации компонентов приложения обладает подкаталогами, которые содержат представления «views» и помощники «helpers», контроллеры «controllers» и бэкэнд-бизнес-логику «models». Он играет важную роль в структуре и функционировании приложения, предоставляя место для размещения различных компонентов, необходимых для обработки веб-запросов и отображения данных.

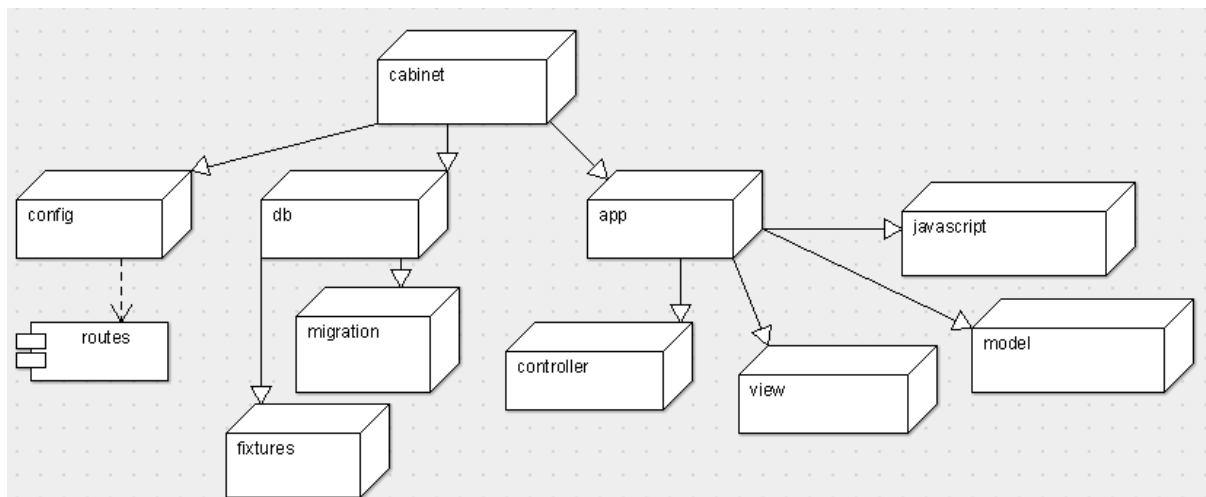


Рисунок 16 – Файловая структура проекта

Подкаталог «app/controllers» содержит классы контроллеров, которые являются частью архитектуры приложения и отвечают за обработку веб-запросов,

поступающих от пользователей. Они принимают запросы, взаимодействуют с моделями и представлениями, и возвращают соответствующий ответ.

Подкаталог «app/models» содержит классы моделей, которые моделируют и упаковывают данные, хранящиеся в базе данных приложения. Модели играют важную роль в описании бизнес-логики и операций над данными. Фреймворк «Rails» обеспечивает простоту и эффективность разработки в этой области, позволяя разработчикам легко взаимодействовать с базой данных и обрабатывать данные в своем приложении.

Подкаталог «app/view» содержит шаблоны отображения, которые заполняются данными из приложения, преобразуются в HTML и возвращаются в браузер пользователю. Здесь происходит форматирование и представление данных для отображения пользователю.

Каталог «config» содержит код конфигурации, необходимый для настройки приложения, включая конфигурацию базы данных, структуру среды выполнения «Rails» и маршрутизацию входящих веб-запросов. В этом каталоге также можно настроить поведение приложения для различных сред, таких как тестирование, разработка и развертывание.

Каталог «db» обычно содержит скрипты и инструменты для управления реляционной базой данных, используемой приложением. Здесь можно создавать, изменять и управлять структурой и содержимым базы данных, используя объекты моделей приложения. Этот каталог играет важную роль в поддержке и развитии приложения, связанной с базой данных.

Диаграмма развертывания – это тип диаграммы UML, которая иллюстрирует архитектуру выполнения системы, включая такие узлы, как аппаратные или программные среды выполнения, а также промежуточное программное обеспечение, которое их соединяет. Схема показана на рисунке 17.

По рисунку видно, что в данном случае используется архитектура клиент-сервер типа "тонкий клиент", показана на рисунке 18. Такая архитектура предполагает распределение функций между клиентами и серверами в распределенных вычислительных системах.

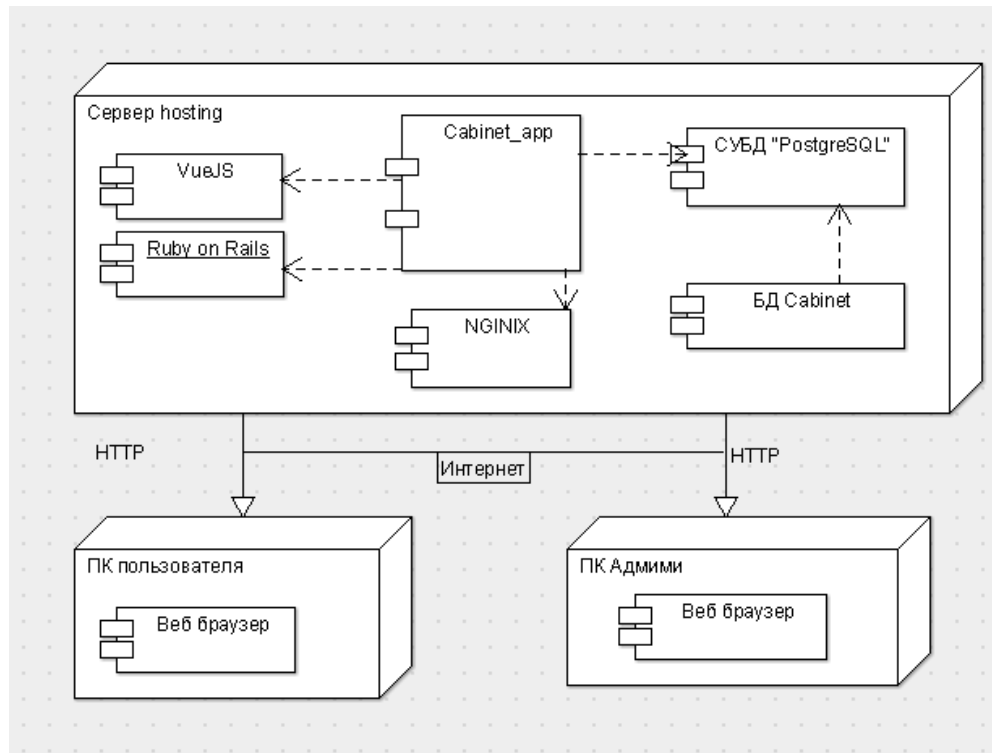


Рисунок 17 – Диаграмма развертывания

В тонком клиенте клиентские устройства имеют ограниченные вычислительные возможности и хранят минимальное количество данных и программного обеспечения. Вся вычислительная нагрузка и обработка данных происходят на серверах, которые находятся в центральной локации.

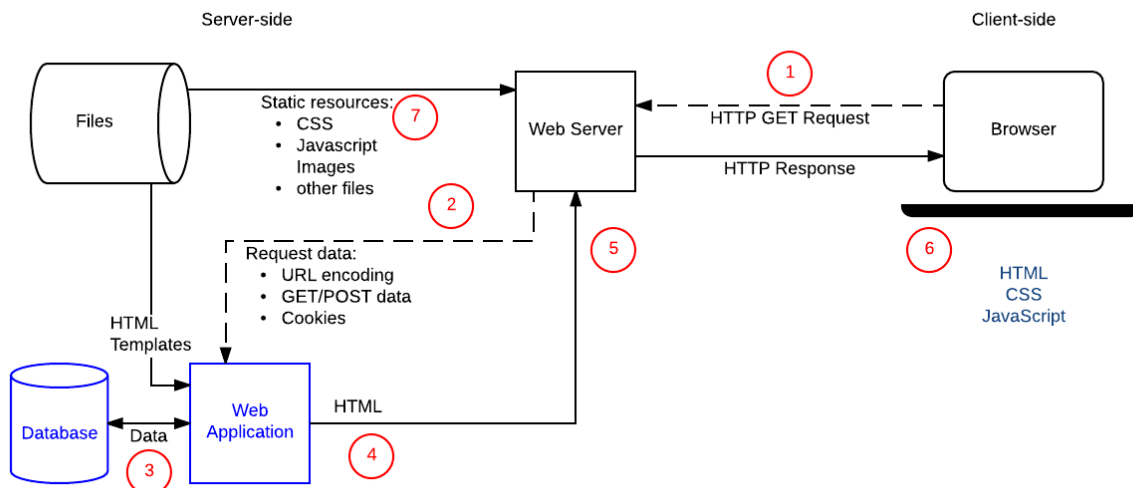


Рисунок 18 – Клиент серверная архитектура

Преимущества такой архитектуры заключаются в централизации хранения данных и программного обеспечения, что упрощает управление и обновление системы. Клиентские устройства могут быть более доступными и простыми в

использовании, так как не требуют высокой вычислительной мощности и большого объема памяти. Кроме того, такой подход может повысить безопасность, так как данные хранятся на серверах, что уменьшает риск потери информации в случае утери или кражи устройства.

Однако тонкий клиент также имеет некоторые недостатки. Ограниченные возможности клиентских устройств могут привести к снижению производительности системы, если серверы не справляются с большой нагрузкой запросов. Реализация такой архитектуры может быть сложной и требовать дополнительных усилий для обеспечения стабильной работы системы.

2.4 Выбор и обоснование модели жизненного цикла

Существует несколько методик разработки программного обеспечения, каждая из которых предлагает свой подход к управлению процессом разработки. Ниже приведены некоторые из самых популярных методик.

«Spiral Model», спиральная модель, представленная на рисунке 19. Спиральная модель объединяет элементы каскадной модели и итеративного подхода. Она предлагает циклический процесс разработки, в котором каждый цикл состоит из четырех основ фаз: определение целей и альтернатив, оценка рисков, разработка и тестирование.

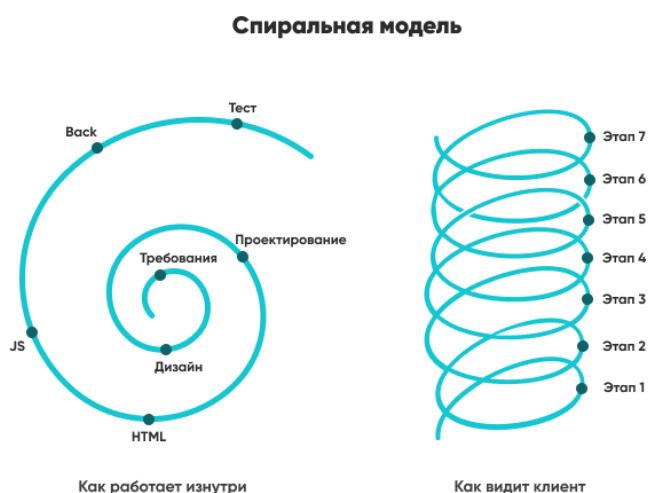


Рисунок 19 – Спиральная модель

Каждый цикл начинается с небольшого прототипа или версии продукта, которая затем анализируется и оценивается перед переходом к следующему

циклу.

Каскадная модель «Waterfall Model», рисунок 20. Каскадная модель представляет собой линейный подход к разработке программного обеспечения, где каждая фаза проекта выполняется последовательно. Этапы разработки включают сбор требований, проектирование, разработку, тестирование и развертывание. Каждая фаза завершается перед переходом к следующей, и изменения в требованиях после начала разработки могут быть затруднительными.



Рисунок 20 – Каскадная модель

«Agile-методологии», такие как «Scrum» и «Kanban», являются итеративными и инкрементальными подходами к разработке программного обеспечения. Они ставят акцент на гибкость, коммуникацию и быстрое реагирование на изменения требований. Разработка происходит в коротких циклах, называемых спринтами, где команда разработчиков работает над определенными задачами в течение ограниченного времени.

«DevOps» – это философия и методология, объединяющая разработку и операции в единый процесс, рисунок 21. «DevOps» стремится к автоматизации и сглаживанию разрыва между командами разработчиков и командами операций, что позволяет более быстро и надежно доставлять программное обеспечение в эксплуатацию.



Рисунок 21 – DevOps методология

После анализа самых популярных моделей жизненного цикла разработки программного продукта было принято решение остановиться на спиральной модели. Выбор данной модели обусловлен ее способностью эффективно управлять изменениями требований к продукту, а также обеспечивать оперативную генерацию результатов на каждой итерации и оценку работы программного продукта относительно прототипа.

Спиральная модель также хорошо подходит для расширения функционала, поскольку она позволяет гибко вносить изменения в процесс разработки новых версий прототипа. Это особенно ценно в области программной разработки, где требования и технические аспекты могут изменяться со временем.

2.5 Требования к информационной безопасности для приложений «Ruby on Rails»

«Ruby on Rails» служит широко распространенной средой веб-разработки, известной своей дружелюбностью к новичкам. Однако, как и любая экосистема разработки, она требует защиты от различных векторов атак. Ниже представлены распространенные уязвимости в «RoR».

«XSS/Cross-Site Scripting»: «XSS-атаки» выделяются как преобладающая брешь в безопасности в проектах «Ruby on Rails», способная полностью скомпрометировать веб-приложения. Используя несколько точек входа, злоумышленники внедряют в проект вредоносный код. Атаки «XSS» могут исходить из

страниц результатов поиска, сообщений, комментариев, обзоров и подобных источников. Внедренный вредоносный код остается встроенным в приложение, что делает его доступным для пользователей.

«CSRF/подделка межсайтовых запросов»: «CSRF» или подделка межсайтовых запросов, пример которой показан на рисунке 22. Использование метода «match» в файле «route.rb» определяет систему маршрутизации на веб-сайте. Этот метод помогает связать определенные действия с различными методами «HTTP-запроса», включая «GET», «POST», «PATCH», «DELETE» и т. д.

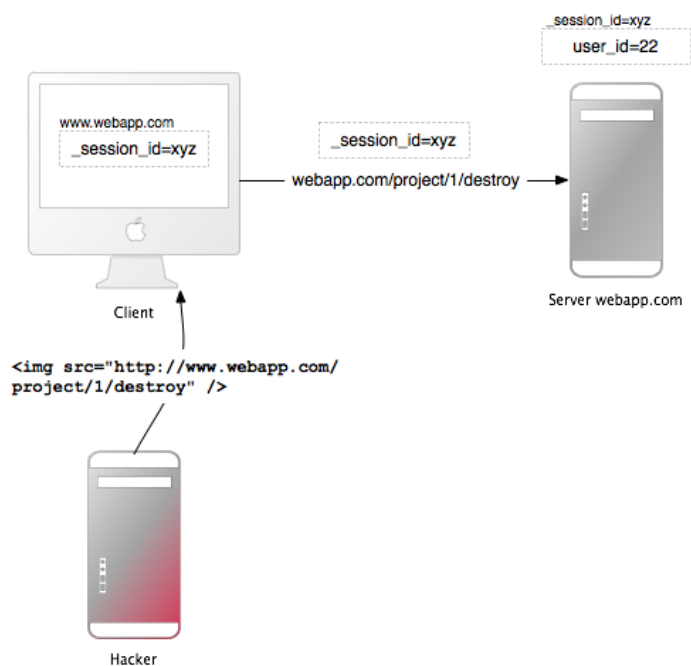


Рисунок 22 – «CSRF»

Инструменты сканирования безопасности «Rails» постоянно рекомендуют использовать альтернативные методы «HTTP» для передачи параметров и мониторинга ответов сервера. Сообщество разработчиков «RoR» разработало защитный механизм, называемый аутентификацией по токenu, для противодействия таким атакам.

SQL-инъекции. Злоумышленники часто используют SQL-инъекции для определения путей передачи непроверенных данных. SQL-инъекции не только предоставляют несанкционированный доступ к базе данных, но и раскрывают конфиденциальную информацию. Хакеры часто полагаются на SQL-инъекции, как показано на рисунке 23, для быстрого поиска определенных данных, что об-

легчает быстрый поиск нужных записей. Кроме того, хакеры используют возможность внедрения вредоносного кода в операторы SQL.

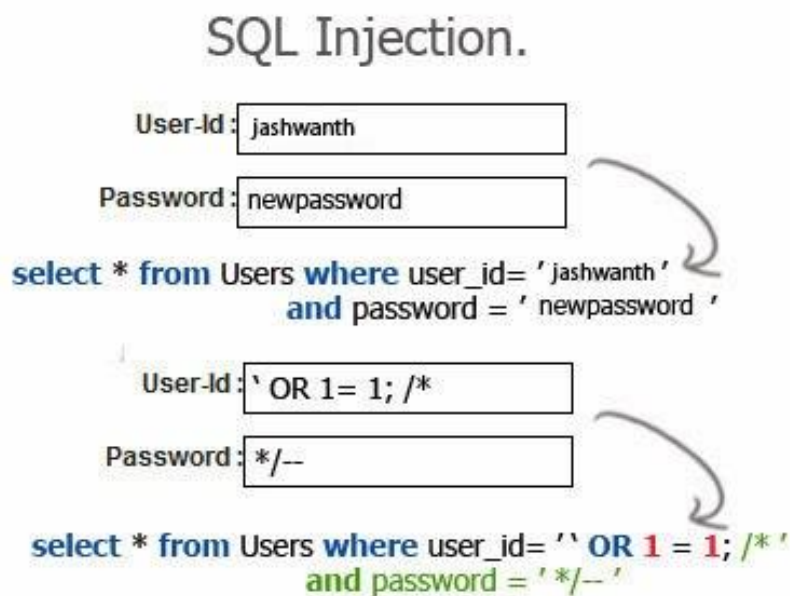


Рисунок 23 – SQL Инъекция

«Clickjacking», автоматическая сетевая атака, которая перенаправляет пользователей на другие страницы, не затрагивая целевой сайт, представляет незначительную угрозу. Хакеры часто используют такие атаки для направления трафика на внешние ресурсы. Среда разработки RoR включает механизм предотвращения таких перенаправлений. Этого можно добиться, добавив HTTP-заголовков «X-Frame-Options: SAMEORIGIN» на соответствующие страницы.

«RoR» включает в себя основные механизмы и методы защиты для обеспечения надежной безопасности. Придерживаясь кратких рекомендаций по безопасности и «Rails-way», лучших практик при разработке с использованием «Ruby on Rails», можно быть уверенным в использовании преимуществ инфраструктуры «RoR» для веб-разработки.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

3.1 Разработка структуры базы данных

3.1.1 Описание инфологической структуры

Первым этапом проектирования необходимо определить перечень сущностей необходимых для функционирования сервиса, выбран следующий набор:

- сущность «people» («Пользователи») хранит в себе основную информацию об пользователях системы;

- сущность «prtf_portfolios» («Портфолио») хранит информацию об портфолио пользователя личного кабинета;

- сущность «prtf_achievement_types» («Типы достижений») хранит информацию об типах деятельности;

- сущность «prtf_achievement_kinds» («Виды достижений») хранит в себе информацию об видах достижений каждого типа деятельности;

- сущность «prtf_achievements» («Достижения») хранит в себе информацию об достижениях;

- сущность «prtf_achievement_levels» («Уровни достижений») хранит в себе информацию об уровнях вида достижения, уровень может дополнительно делиться на критерии;

- сущность «prtf_achievement_units» («Критерии достижений») хранит в себе информацию об критериях оценки достижений;

- сущность «scholarship_competitions» («Заявление на стипендию») хранит в себе информацию об заявлениях студентов на участие в конкурсе на повышенную стипендию;

- сущность «scholarship_meetings» («Стипендиальная комиссия») хранит в себе информацию об заявлениях студентов на участие в конкурсе на повышенную стипендию;

- сущность «department» («Отделы») хранит в себе информацию об отделах университета, например, факультеты и кафедры.

Следующим этапом сформируем спецификации столбцов каждой таб-

лицы. Спецификация атрибутов таблицы «Пользователи» представлены на таблице 4.

Таблица 4 – Спецификация атрибутов таблицы «Пользователи»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
<u>ИД</u>	Число, однозначно определяющее пользователя	bigint	> 0	234 567
Имя	Имя	string	–	Даниил
Фамилия	Фамилия	string	–	Данилов
Отчество	Отчество	string	–	Даниилович
Дата рождения	День рождение	date	≤ now	12.12.1912
ИД1С	Идентификатор, однозначно определяющий пользователя в системе 1С Университет	string	id000001 – id999999 ab000001 – ab999999	id009473
Адрес регистрации	Адрес	string	–	675000, РОССИЯ, Благовещенск г, Театральная 1
национальность	Гражданство	string	–	Россия
пол	Пол	string	–	Мужской
Семейное положение	Семейное положение	string	–	Холост
Номер СНИЛС	Номер СНИЛС	string	–	0000 0000 0000
фото	Фото	string	–	Фото
Успехи студента	Портфолио студента	text	–	–
диссертации	Научные работы	text	–	–
Персональная информация	Персональная информация	text	–	–

Теперь сформируем спецификацию для сущности в «Портфолио». Спецификация атрибутов представлена на таблице 5.

Таблица 5 – Спецификация атрибутов таблицы «Портфолио»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
ИД	Число, однозначно определяющее портфолио	bigint	> 0	12
фото	Фото	string	–	Фото
токен	Уникальный токен портфолио	string	–	3131dadaewq331

Продолжим формирование спецификации, рассмотрим сущность «Типы достижений». Спецификация атрибутов представлена на таблице 6.

Таблица 6 – Спецификация атрибутов таблицы «Типы достижений»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
ИД	Число, однозначно определяющее тип достижения	bigint	> 0	8
Название	Название типа достижения, деятельности	string	–	За достижения в общественной деятельности
Группа	Для какой группы людей предназначен данный тип достижения	integer	1 или 3	1

Продолжим формирование спецификации, рассмотрим сущность «Виды достижений». Спецификация атрибутов представлена на таблице 7.

Таблица 7 – Спецификация атрибутов таблицы «Виды достижений»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
ИД	Число, однозначно определяющее вид достижения	bigint	> 0	8
Название	Название вида достижения	string	–	Участие в Федеральном интернет экзамене бакалавров
Описание	Описание, комментарий к виду достижения	string	–	Служебная записка, подтверждающая членство в отряде.
Скрыт	Отвечает за сокрытие вида достижения для пользователей	boolean	True или False	False

Продолжим формирование спецификации, рассмотрим сущность «Достижения». Спецификация атрибутов представлена на таблице 8.

Таблица 8 – Спецификация атрибутов таблицы «Достижения»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
ИД	Число, однозначно определяющее достижение	bigint	> 0	2854

1	2	3	4	5
Название	Название достижения	string	3 – 128	Грамота
Описание	Описание к достижению	text	–	Грамота за олимпиаду по математике
Дата получения	Дата получения достижения	string	–	2023-05-23

Продолжим формирование спецификации, рассмотрим сущность «Уровни достижения». Спецификация атрибутов представлена на таблице 9.

Таблица 9 – Спецификация атрибутов таблицы «Уровни достижений»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
ИД	Число, однозначно определяющее уровень	bigint	> 0	3
Название	Название уровня	string	–	международный
баллы	Количество баллов за уровень	float	–	3.5

Продолжим формирование спецификации, рассмотрим сущность «Критерий достижения». Спецификация атрибутов представлена на таблице 10.

Таблица 10 – Спецификация атрибутов таблицы «Критерий достижений»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
ИД	Число, однозначно определяющее уровень	bigint	> 0	3
Название	Название критерия	string	–	1 место
баллы	Количество баллов за критерий	float	–	4.5

Продолжим формирование спецификации, рассмотрим сущность «Заявление на стипендию». Спецификация атрибутов представлена на таблице 11.

Таблица 11 – Спецификация атрибутов таблицы «Заявление на стипендию»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
ИД	Число, однозначно определяющее заявление	bigint	> 0	31
Дата подачи	Дата подача заявления	string	–	2023-05-23

Продолжение таблицы 11

1	2	3	4	5
Сумма баллов за учебную деятельность	Сумма баллов за выбранные достижения по учебной деятельности	float	–	45
Сумма баллов за научно-исследовательскую деятельность	Сумма баллов за выбранные достижения научно-исследовательской деятельности	float	–	22
Сумма баллов за общественную деятельность	Сумма баллов за выбранные достижения общественной деятельности	float	–	12
Сумма баллов за культурно-творческую деятельность	Сумма баллов за выбранные достижения культурно-творческой деятельности	float	–	43
Сумма баллов за спортивную деятельность	Сумма баллов за выбранные достижения спортивной деятельности	float	–	21
Общая сумма баллов	Сумма баллов за все типы деятельности	float	–	78
Приоритетный тип деятельности	Выбранный приоритетный тип деятельности	bigint	–	3

Продолжим формирование спецификации, рассмотрим сущность «Стипендиальная комиссия». Спецификация атрибутов представлена на таблице 12.

Таблица 12 – Спецификация атрибутов таблицы «Стипендиальная комиссия»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5
ИД	Число, однозначно определяющее заявление	bigint	> 0	31
Дата проведения	Дата заседания комиссии	string	–	2023-05-23
Председатель	Председатель комиссии	bigint	–	231
Секретарь	Секретарь комиссии	bigint	–	123
Слушатели	Массив слушателей	bigint	–	[321, 234, 122]

Продолжим формирование спецификации, рассмотрим сущность «Отделы». Спецификация атрибутов представлена на таблице 13.

Таблица 13 – Спецификация атрибутов таблицы «Отделы»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
1	2	3	4	5

1	2	3	4	5
ИД	Число, однозначно определяющее заявление	bigint	> 0	31
ИД1С	Идентификатор, однозначно определяющий отдел в системе 1С Университет	string	–	Id000321
Название	Название отдела	string	–	Факультет математики и информатики
Аббревиатура	Аббревиатура отдела	string	–	ФМиИ
Адрес	Адрес расположения отдела	string	–	Россия, Амурская область, г. Благовещенск, ул. Игнатъевское шоссе, д. 21, корпус 8, каб. 208
Номер	Номер отдела	string	–	
Почта	Электронная почта отдела	string	–	jf@amursu.ru
ФИО ответственного	ФИО ответственного	string	–	Данилов Даниил Данилович
Пост ответственного	Пост ответственного	string	–	Декан
Информация	Информация по отделу	text	–	В настоящее время на нем обучается более 500 студентов - по дневной и более 450 ...

Заключительном этапе при инфологическом проектировании является создания модели сущность-связь в нотации Мартина, Чена или IDEF1X. Для построения модели БД в данной курсовой работе была выбрана нотация Мартина, ER-диаграмма показана на рисунке 24.

Инфологическая схема базы данных будет реализована с использование свободно распространяемого программного обеспечения «draw.io».

3.1.2 Описание логической структуры

Логическое проектирование базы данных (ЛПБД) – это этап разработки базы данных, на котором определяются ее структура и связи между данными.

Целью логического проектирования является создание схемы базы дан-

ных, которая будет эффективно хранить и обрабатывать информацию, отражая требования и бизнес-правила организации.

Связь «Пользователи» – «Портфолио». Рассмотрим две сущности: «Пользователи» и «Портфолио». Между ними установлена связь типа «один-к-одному». Одному пользователю может соответствовать только одна портфолио в БД, и в то же время у портфолио может быть только один пользователь, связь показана на рисунке 25. Исходя из данной информации добавим внешний ключ к сущности «Портфолио», чтобы связать данные сущности.

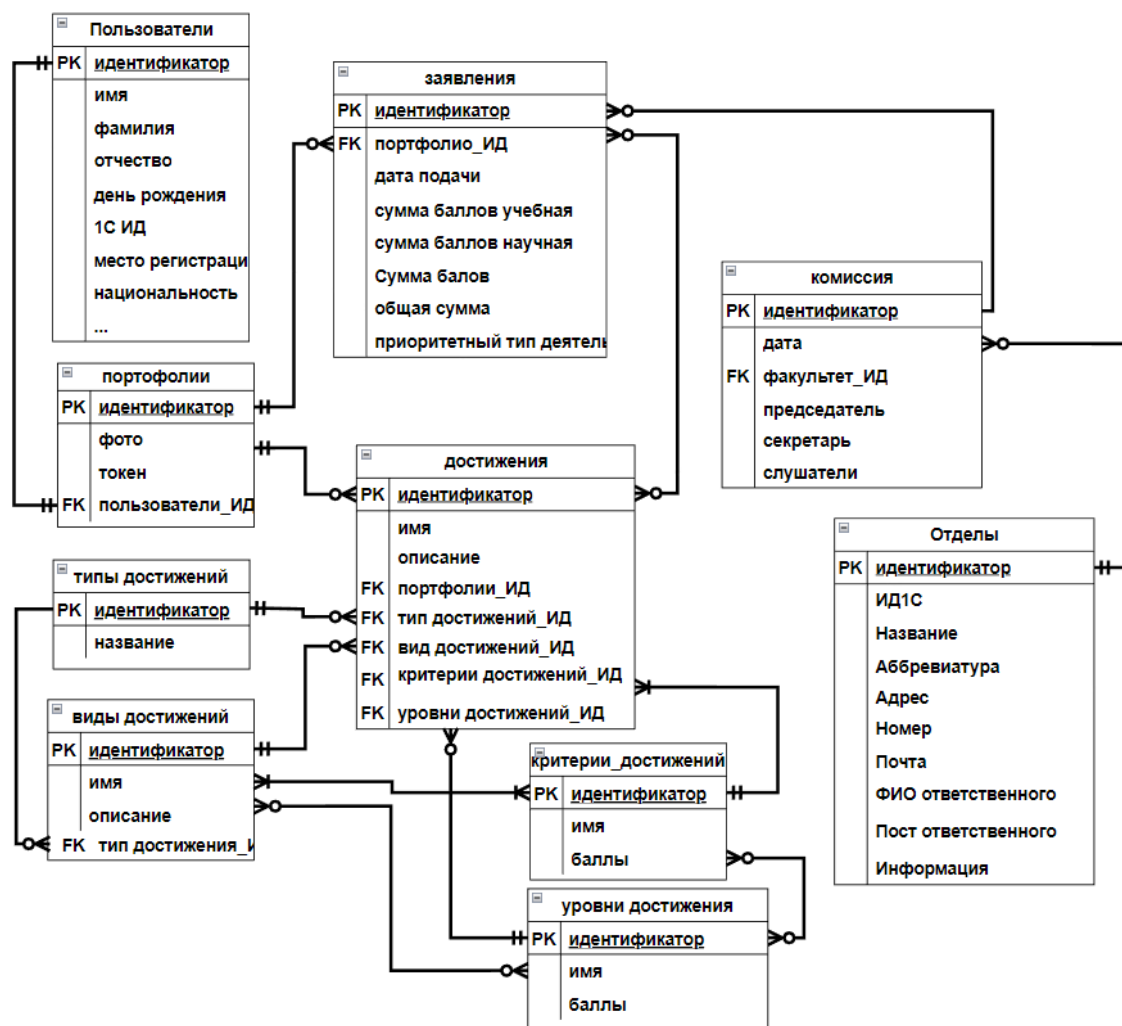


Рисунок 24 – Инфологическая схема БД в нотации Мартина



Рисунок 25 – Связь «Пользователь» – «Портфолио»

Связь «Портфолио» – «Достижения». Рассмотрим две сущности: «Портфолио» и «Достижения». Между ними установлена связь типа «один-ко-многим». Одному портфолио может соответствовать несколько достижений в БД, и в то же время у одного достижения может быть только одна портфолио, связь показана на рисунке 26. Добавим внешний ключ в сущность «Достижения», так как он является дочерним по отношению к сущности «Портфолио».



Рисунок 26 – Связь «Портфолио» – «Достижения»

Связь «Типы достижений» – «Достижения». Рассмотрим две сущности: «Типы достижений» и «Достижения». Между ними установлена связь типа «один-ко-многим». Одному типу достижений может соответствовать несколько достижений в БД, и в то же время у одного достижения может быть только один тип достижения, связь показана на рисунке 27. Добавим внешний ключ в сущность «Достижения», так как он является дочерним по отношению к сущности «Типы достижений».

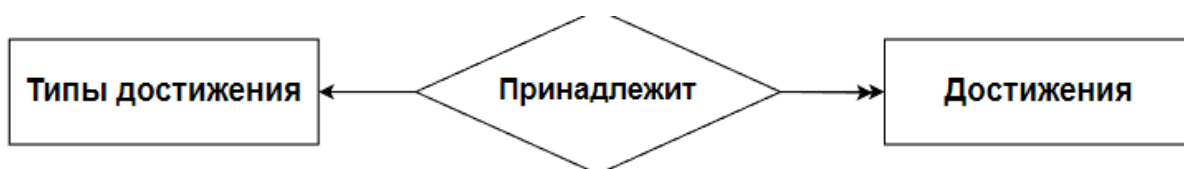


Рисунок 27 – Связь «Типы достижений» – «Достижения»

Связь «Уровни достижений» – «Критерии достижений». Рассмотрим две сущности: «Уровни достижений» и «Критерии достижений». Между ними уста-

новлена связь типа «многие-ко-многим».

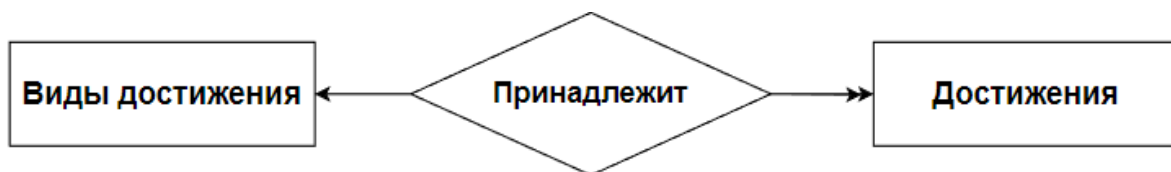


Рисунок 28 – Связь «Виды достижений» – «Достижения»

Одному уровню достижения может соответствовать несколько критериев достижений в БД, и в то же время у одного критерия достижения может быть несколько уровней достижения, связь показана на рисунке 29.



Рисунок 29 – Связь «Уровни достижений» – «Критерии достижений»

Связь «Вид достижения» – «Критерии достижений». Рассмотрим две сущности: «Вид достижения» и «Критерии достижений». Между ними установлена связь типа «многие-ко-многим». Одному виду достижения может соответствовать несколько критериев достижений в БД, и в то же время у одного вида достижения может быть несколько критериев достижения, связь показана на рисунке 30.

Связь «Вид достижения» – «Уровни достижений». Рассмотрим две сущности: «Вид достижения» и «Критерии достижений». Между ними установлена связь типа «многие-ко-многим». Одному уровню достижения может соответствовать несколько видов достижений в БД, и в то же время у одного уровня достижения может быть несколько видов достижений, связь показана на рисунке 31.



Рисунок 30 – Связь «Вид достижения» – «Критерии достижений»

Так как представление связи, представленные на рисунке 30 и 31, типа «многие-ко-многим», ее следует разрешить путем добавления связывающей

сущности «Виды достижений_Критерии_Уровни», в данную сущность добавим внешние ключи сущности «Уровни достижений», «Критерии достижений» и «Виды достижений».



Рисунок 31 – Связь «Вид достижения» – «Уровни достижений»

Данное решение обусловлено тем, что у одного вида может быть, как и множество только уровней, так и множество только критериев. Но также может быть и множество уровней в свою очередь у уровней может быть множество дополнительных критериев.

Связь «Заявление на стипендию» – «Достижения». Рассмотрим две сущности: «Заявление на стипендию» и «Достижения». Между ними установлена связь типа «многие-ко-многим». Одному достижению может соответствовать несколько заявлений в БД, и в то же время у одного заявления может быть несколько достижений, связь показана на рисунке 32. Для решения связи «многие-ко-многим» добавим новую сущность «Заявление_ Достижения» и помести в данную сущность внешние ключи этих двух таблиц.



Рисунок 32 – Связь «Заявление на стипендию» – «Достижения»

Связь «Стипендиальная комиссия» – «Заявление на стипендию». Рассмотрим две сущности: «Стипендиальная комиссия» и «Заявление на стипендию». Между ними установлена связь типа «один-ко-многим». Одной комиссии может соответствовать несколько заявлений в БД, и в то же время у одного заявления может быть только одна комиссия, связь показана на рисунке 33. Добавим внешний ключ в сущность «Заявление на стипендию», так как он является дочерним по отношению к сущности «Стипендиальная комиссия».



Рисунок 33 – Связь «Стипендиальная комиссия» – «Заявление на стипендию»

Связь «Отделы» – «Стипендиальная комиссия». Рассмотрим две сущности: «Отделы» и «Стипендиальная комиссия». Между ними установлена связь типа «один-ко-многим». Одному отделу может соответствовать множество комиссий в БД, и в то же время у одной комиссии может быть только один отдел, связь показана на рисунке 34. Добавим внешний ключ в сущность «Стипендиальная комиссия», так как он является дочерним по отношению к сущности «Отделы».



Рисунок 34 – Связь «Отделы» – «Стипендиальная комиссия»

Сущность «Достижения» связана с сущностями «Уровни достижения», «Критерии достижения» связаны отношением «один-ко-многим». Поэтому добавим в сущность «Достижения» внешние ключи перечисленных сущностей. Связи показаны на рисунке 35.

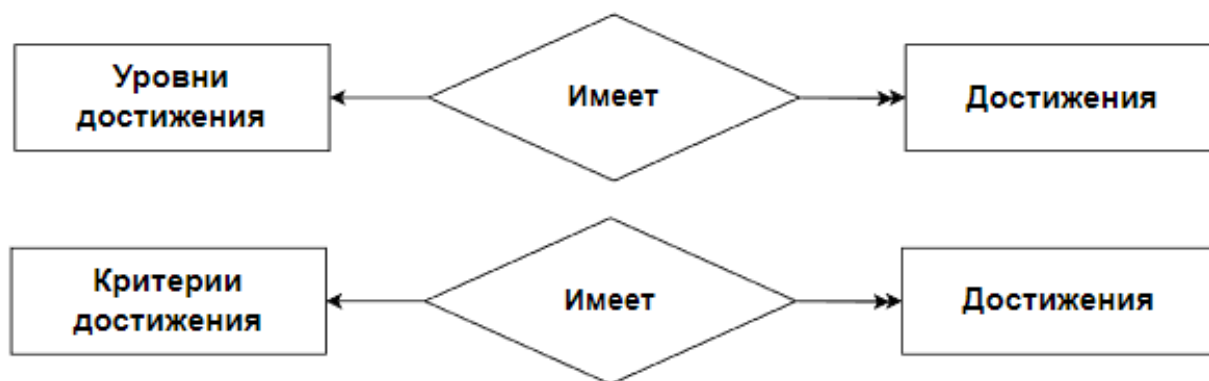


Рисунок 35 – Связи сущностью «Достижения»

Связь «Портфолио» – «Заявление на стипендию». Рассмотрим две сущности: «Портфолио» и «Заявление на стипендию». Между ними установлена связь типа «один-ко-многим». Одна портфолио может иметь множество заявлений в БД, и в то же время у одного заявления может быть только одна портфолио, связь

показана на рисунке 36. Добавим внешний ключ в сущность «Заявление на стипендию», так как он является дочерним по отношению к сущности «Портфолио».



Рисунок 36 – Связь «Портфолио» – «Заявление на стипендию»

Все отношения, указанные в таблицах (4 – 13), приведены к первой, второй и третьей нормальной форме, удовлетворяя следующим условиям:

- каждый атрибут должен быть атомарным, то есть не может быть разделен на более мелкие части;
- каждый атрибут в таблице должен зависеть только от первичного ключа;
- нет транзитивных зависимостей между атрибутами, то есть если атрибут А зависит от В, а В зависит от С, то А должен зависеть только от В, а не от С.

Завершим этап реализации логической структуры БД, путем построения схемы в формате «IDEF1X», схема показана на рисунке 37.

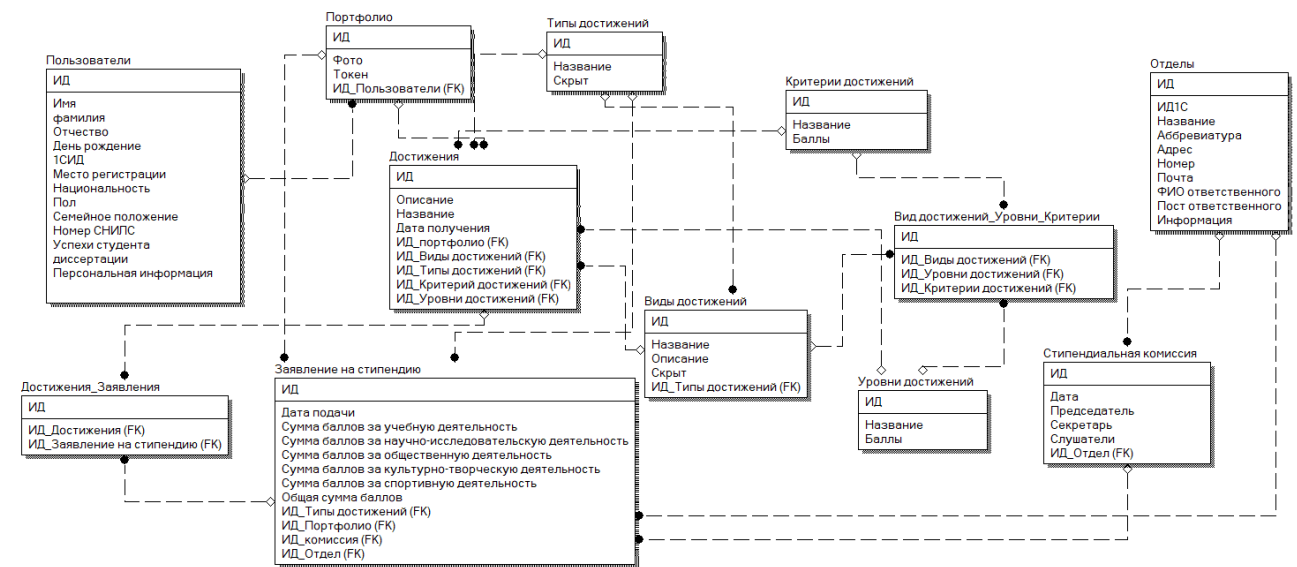


Рисунок 37 – Логическая схема IDEF1X

3.1.3 Реализация физической структуры

На основании составленных таблицы атрибутов были установлены необходимые связи между сущностями, разрешена связь «многие-ко-многим», а также определены необходимые типы данных. Диаграмма, представляющая

связи между сущностями, теперь может быть использована для физического проектирования базы данных. Физическое проектирование БД представляет собой процесс создания физической структуры БД на основе логического моделирования. В таблицах (14 – 25) приведены физические представления атрибутов сущностей.

Физическое представление атрибутов включает в себя выбор оптимальных типов данных для хранения значений атрибутов, определение ограничений целостности и индексирование данных для обеспечения эффективного доступа и запросов к БД.

Таблица 14 – Физическая структура данных сущности «Пользователи»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
1	2	3	4	5
ИД	Число	>0	bigint	Primary key
Имя	Строка	Не пустое	String	–
Фамилия	Строка	Не пустое	String	–
Отчество	Строка	Не пустое	String	–
Дата рождения	Дата	> 01.01.1940	Date	–
ИДІС	Строка	Не пустое	String	–
Адрес регистрации	Строка	Не пустое	String	–
национальность	Строка	Не пустое	String	–
пол	Строка	Не пустое	String	–
Семейное положение	Строка	Не пустое	String	–
Номер СНИЛС	Строка	Не пустое	String	–
фото	Строка	Не пустое	String	–
Успехи студента	Текст	–	Text	–
диссертации	Текст	–	Text	–
Персональная информация	Текст	–	Text	–

Таблица 15 – Физическая структура данных сущности «Портфолио»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
1	2	3	4	5
ИД	Число	>0	bigint	Primary key
фото	Строка	–	String	–
токен	Строка	Не пустое	String	–
ид пользователь	Число	Не пустое	bigint	Foreign key

Таблица 16 – Физическая структура данных сущности «Типы достижений»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
1	2	3	4	5
ИД	Число	>0	bigint	Primary key
Название	Строка	–	String	–
Группа	Число	1 или 5	integer	–

Таблица 17 – Физическая структура данных сущности «Типы достижений»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
1	2	3	4	5
ИД	Число	>0	bigint	Primary key
Название	Строка	–	String	–
Описание	Число	1 или 5	integer	–
Скрыт	Логическое	True или False	Boolean	–
ИД_типы достижений	Число	>0	bigint	Foreign key

Таблица 18 – Физическая структура данных сущности «Типы достижений»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
1	2	3	4	5
ИД	Число	>0	bigint	Primary key
Название	Строка	–	String	–
Описание	Число	1 или 5	integer	–
Дата получения	Строка	–	String	–
ИД_Типы достижений	Число	>0	bigint	Foreign key
ИД_Виды достижений	Число	>0	bigint	Foreign key
ИД_Критерий достижения	Число	>0	bigint	Foreign key
ИД_Уровень достижения	Число	>0	bigint	Foreign key

Таблица 19 – Физическая структура данных сущности «Уровень достижений»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
1	2	3	4	5
ИД	Число	>0	bigint	Primary key
Название	Строка	–	String	–
баллы	Число	> 0	float	–

Таблица 20 – Физическая структура данных сущности «Критерий достижений»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
1	2	3	4	5

1	2	3	4	5
ИД	Число	>0	bigint	Primary key
Название	Строка	–	String	–
баллы	Число	> 0	float	–

Таблица 21 – Физическая структура данных сущности «Заявление на стипендию»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
1	2	3	4	5
ИД	Число	>0	bigint	Primary key
Дата подачи	Строка	–	String	–
Сумма баллов за учебную деятельность	Число	> 0	float	–
Сумма баллов за научно-исследовательскую деятельность	Число	> 0	float	–
Сумма баллов за общественную деятельность	Число	> 0	float	–
Сумма баллов за культурно-творческую деятельность	Число	> 0	float	–
Сумма баллов за спортивную деятельность	Число	> 0	float	–
Общая сумма баллов	Число	> 0	float	–
Приоритетный тип деятельности	Число	> 0	float	–
ИД_ Стипендиальная комиссия	Число	> 0	bigint	Foreign key
ИД_ Портфолио	Число	> 0	bigint	Foreign key

Таблица 22 – Физическая структура данных сущности «Стипендиальная комиссия»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
1	2	3	4	5
ИД	Число	>0	bigint	Primary key
Дата проведения	Строка	–	String	–
Председатель	Число	> 0	bigint	–
Секретарь	Число	> 0	bigint	–
Слушатели	Массив чисел	–	bigint	–
ИД_отдел	Число	> 0	bigint	Foreign key

Таблица 23 – Физическая структура данных сущности «Отдел»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
1	2	3	4	5
ИД	Число	>0	bigint	Primary key
ИД1С	Строка	–	String	–
Название	Строка	–	String	–
Аббревиатура	Строка	–	String	–
Адрес	Строка	–	String	–
Номер	Строка	–	String	–
Почта	Строка	–	String	–
ФИО ответственного	Строка	–	String	–
Пост ответственного	Строка	–	String	–
Информация	Строка	–	String	–

Таблица 24 – Физическая структура данных сущности «Виды достижений_Уровни_Критерии»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
1	2	3	4	5
ИД	Число	>0	bigint	Primary key
ИД_Виды достижений	Число	>0	bigint	Foreign key
ИД_Уровни	Число	>0	bigint	Foreign key
ИД_Критерии	Число	>0	bigint	Foreign key

Таблица 25 – Физическая структура данных сущности «Достижения_Заявления»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
1	2	3	4	5
ИД_Достижения	Число	>0	bigint	Foreign key
ИД_Заявления	Число	>0	bigint	Foreign key

Завершим этап реализации физической структуры БД, путем построения схемы в формате «IDEF1X», схема показана на рисунке 38.

После проведения и выполнения реализации трех этапов проектирования БД, можно приступить к реализации человеко-машинного интерфейса для удобной работы пользователей с подсистемой.

3.2 Разработка клиентской части подсистемы

Для удобства пользователя в качестве интерфейса системы будут использованы «Реактивные» элементы, позволяющие ускорить работу системы пере-

неся часть вычислений на рабочую машину пользователя, и организовать красивый и удобный интерфейс пользователя. Для данных целей будет использован язык JavaScript. Для разработки интерфейса программного продукта был выбран «Vue.js» за простоту в обучении и полноту документации при разработке.

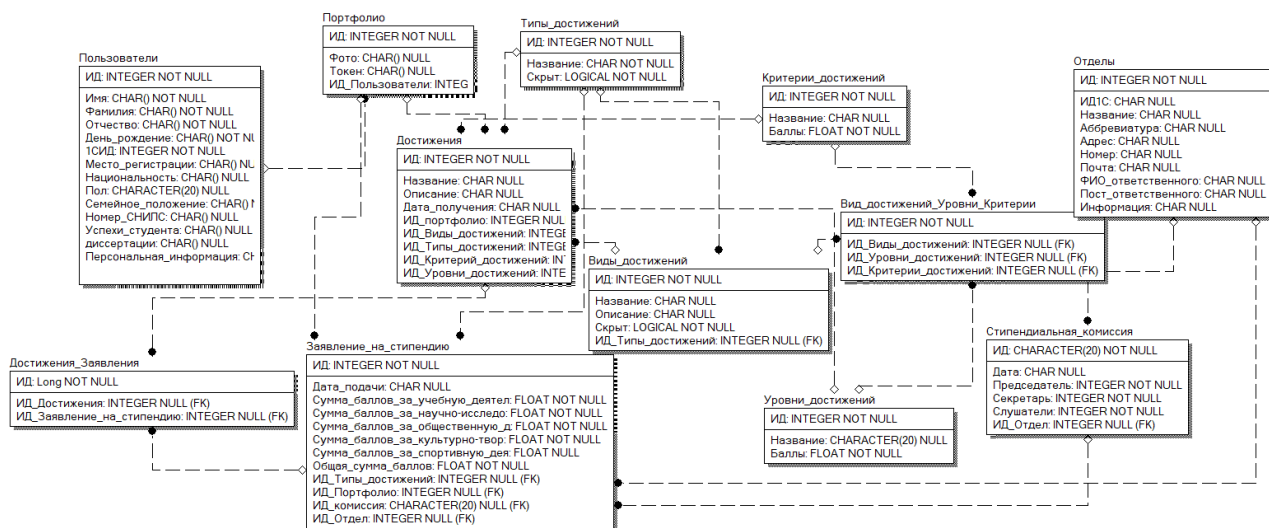


Рисунок 38 – Физическая схема БД

В контексте «Vue.js», реактивность описывает способность фреймворка отслеживать и автоматически обновлять компоненты в ответ на изменение данных.

«Vue.js» использует систему реактивности, которая позволяет связывать данные с шаблонами компонентов. Когда данные изменяются, «Vue.js» автоматически обнаруживает эти изменения и обновляет соответствующие части DOM, чтобы отобразить новое состояние данных.

Основными механизмами реактивности в «Vue.js» являются следующие:

- объекты «Vue», созданный ранее экземпляр «Vue» с помощью конструктора «Vue» и переданный ему объект данных. «Vue» отслеживает изменения в этом объекте и автоматически обновляет соответствующие компоненты;
- «computed-свойства», определяемое вычисляемое свойства внутри компонента «Vue». Эти свойства вычисляются на основе зависимостей, таких как другие данные или вычисляемые свойства, и автоматически обновляются, когда изменяются эти зависимости;
- «watchers», использование «наблюдателей» внутри компонентов «Vue»,

для отслеживания изменений определенных данных и выполнений определенных действия при их изменении.

Создание всей клиентской части проходила с использованием компонентного подхода «Vue». Компонентный подход в «Vue.js» – это способ организации приложения путем разделения его на небольшие и переиспользуемые компоненты. Компоненты в «Vue.js» представляют собой независимые и изолированные блоки функциональности, которые могут содержать шаблон «HTML», логику «JavaScript» и стили «CSS».

Так как разрабатывается большой программный продукт с использованием множества различных компонентов различной вложенности, было принято решение использовать в данном проекте централизованное хранилище данных.

«Vuex» – это шаблон управления состоянием и библиотека для приложений «Vue.js». Она предоставляет централизованное хранилище данных для всех компонентов приложения с правилами, которые гарантируют, что состояние может изменяться только в предсказуемом и контролируемом манере.

«Vuex» помогает справиться с общим управлением состоянием за счет дополнительных концепций и шаблонов. Это компромисс между краткосрочной и долгосрочной продуктивностью. Основную концепцию «Vuex» можно увидеть на рисунке 39.

Для работы и получения данных с серверной части было решено использовать популярную JS библиотеку «Axios». Данное решение является одним из самых популярных для связывания серверной и клиентской части системы.

Основные преимущества использования библиотеки «Axios»:

- простой API для выполнения HTTP-запросов;
- поддержка обещаний и асинхронности, что упрощает работу с асинхронным кодом;
- межплатформенная совместимость – работает как в браузере, так и на сервере.
- возможность использования интерцепторов запросов и ответов для манипуляции данными;

– встроенная обработка ошибок для удобного обнаружения и обработки ошибочных состояний.

Все эти преимущества делают «Axios» удобным и мощным инструментом для работы с сетевыми запросами в приложениях «Vue.js» и других JavaScript-проектах.

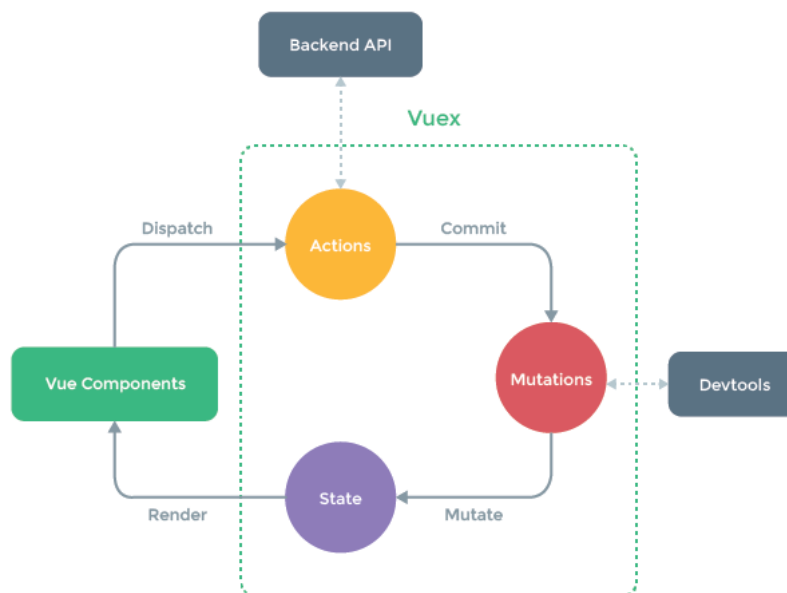


Рисунок 39 – Разработка на «Vue» с использованием «Vuex»

В заключение отмечается, что в ходе разработки клиентской части приложения были тщательно изучены и применены передовые практики, связанные с использованием фреймворка «Vue.js». В основу разработки интерфейса был положен принцип современного дизайна, который обеспечивает эстетическое восприятие и в то же время облегчает понимание и интуитивную навигацию при работе с программным обеспечением. Оформление пользовательского интерфейса программы будет детально представлено в последующем разделе, сопровождаемое описанием функциональности программного обеспечения.

3.3 Разработка серверной части подсистемы

3.3.1 Взаимодействие серверной части с внешними системами

Взаимодействие серверной части с внешними системами будет происходить с использованием API – механизмы, которые позволяют двум программным компонентам взаимодействовать друг с другом, используя набор определений и протоколов. API – (Application Programming Interface), что значит программный

интерфейс приложения. В контексте API слово «приложение» относится к любому ПО с определенной функцией. Интерфейс можно рассматривать как сервисный контракт между двумя приложениями. Этот контракт определяет, как они взаимодействуют друг с другом, используя запросы и ответы. Документация API содержит информацию о том, как разработчики должны структурировать эти запросы и ответы.

«REST API». На сегодняшний день это самые популярные и гибкие API-интерфейсы в Интернете. Клиент отправляет запросы на сервер в виде данных. Сервер использует этот клиентский ввод для запуска внутренних функций и возвращает выходные данные обратно клиенту.

В данном разрабатываемом программном продукте используется именно REST API, упрощенная схема работы «REST API» представлена на рисунке 40.

«REST» – это (Representational State Transfer), т. е. передача репрезентативного состояния. «REST» определяет набор функций, таких как «GET», «PUT», «DELETE» и т. д., которые клиенты могут использовать для доступа к данным сервера. Клиенты и серверы обмениваются данными по протоколу HTTP.

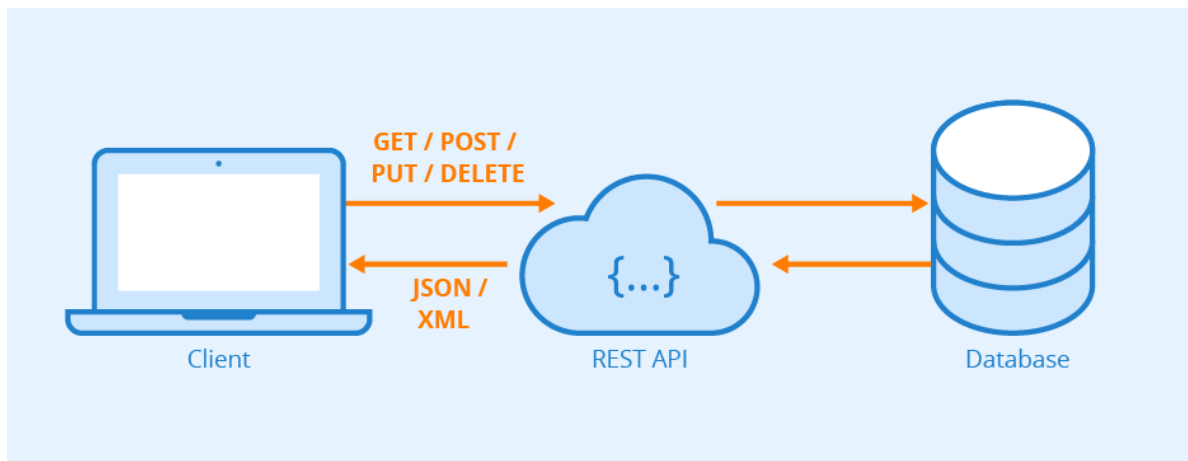


Рисунок 40 – Действие «REST API»

Главной особенностью «REST API» является то, что такая передача выполняется без сохранения состояния. Без сохранения состояния означает, что серверы не сохраняют клиентские данные между запросами. Клиентские запросы к серверу аналогичны URL-адресам, которые вы вводите в браузере для посещения веб-сайта. Ответ от сервера представляет собой простые данные без типичного графического отображения веб-страницы.

Выбор именно данного способа взаимодействия с внешними системами обусловлен преимуществами: интеграции, инновации, расширение, простоте обслуживания API. Именно благодаря API клиентская часть сможет получать и отправлять данные на серверную часть приложения.

Для разработки API на «RoR» требуется использование контроллеров в связке с маршрутами. Контроллеры - это компоненты, которые обрабатывают запросы и формируют ответы.

Маршруты определяют связь между URL-адресами и действиями в контроллерах. Они указывают, какие запросы должны быть отправлены на какие действия контроллера. Маршрутизация позволяет приложению определить, какой контроллер и какое действие должны быть вызваны для обработки запроса.

При разработке API на «Rails» потребуется создать контроллеры для обработки различных запросов. Следует определить действия в контроллерах, которые соответствуют определенным операциям, API которых должно поддерживать. Затем требуется настраивать маршруты, указывая, какие URL-адреса должны быть связаны с определенными действиями контроллеров

3.3.2 Взаимодействие с базой данных

В «Ruby on Rails» связь между объектами и базой данных осуществляется с использованием «ORM», который называется «ActiveRecord». Он обеспечивает отображение полей таблицы в поля объекта, проводит валидацию объектов перед сохранением и генерирует код для установления связей между объектами. Для создания новой модели достаточно унаследоваться от класса «ActiveRecord::Base». По умолчанию «ActiveRecord» будет работать с таблицей, названной так же, как класс, но во множественном числе. Доступ ко всем полям таблицы можно получить с помощью методов, имеющих такие же названия.

«ActiveRecord» в «Ruby on Rails» предоставляет мощные инструменты для работы с базой данных. Он следует принципу «Convention over Configuration», что означает, что большая часть конфигурации уже предопределена, и не требуется писать много кода, чтобы разрабатываемая программа работала.

С помощью «ActiveRecord» можно легко осуществлять операции с дан-

ными, включая поиск, создание и удаление записей. Например, методы «find(id)», «all» и «find_by_...» предоставляют простой способ получения данных из таблицы. Также можно использовать метод «create» для создания новых записей и «destroy» для удаления существующих записей.

Одна из главных преимуществ «ActiveRecord» заключается в том, что он автоматически отображает поля таблицы в поля объекта, что делает работу с данными удобной и интуитивно понятной. Вы можете обращаться к полям таблицы с помощью методов, имеющих такие же названия.

В целом, «ActiveRecord» является мощным инструментом, который значительно упрощает работу с базой данных в «Ruby on Rails», позволяя разработчикам сосредоточиться на бизнес-логике и функциональности своих приложений.

3.4 Работа с программным обеспечением

3.4.1 Работа с модулем «Портфолио»

При входе пользователя в раздел «Портфолио» появляется главное окно, которое иллюстрируется на рисунке 41. Главное окно состоит из заголовка и тела, где заголовок включает аватар и полное имя пользователя. В правом верхнем углу заголовка размещена кнопка, позволяющая перейти на страницу старой портфолио. Ниже кнопки находится вторая кнопка, при нажатии на которую открывается форма для заполнения достижений.

Дополнительно, в нижнем правом углу главного окна находится элемент, обеспечивающий возможность переключения между вкладками. Этот элемент упрощает навигацию и позволяет пользователю быстро переключаться между различными контекстами и содержимым, представленными в рамках портфолио.

Первая вкладка в разделе «Портфолио» под названием «Общие сведения» предназначена для предоставления основной информации о пользователе в рамках системы личного кабинета «АмГУ». Следует отметить, что данная вкладка имеет информационный характер и служит для осведомления пользователя о его учебной и профессиональной истории.

На данной вкладке пользователь может ознакомиться с различными аспектами своего образования, включая, но не ограничиваясь, информацией о теку-

щем учебном процессе, предыдущих образовательных достижениях, а также других связанных с обучением данных.

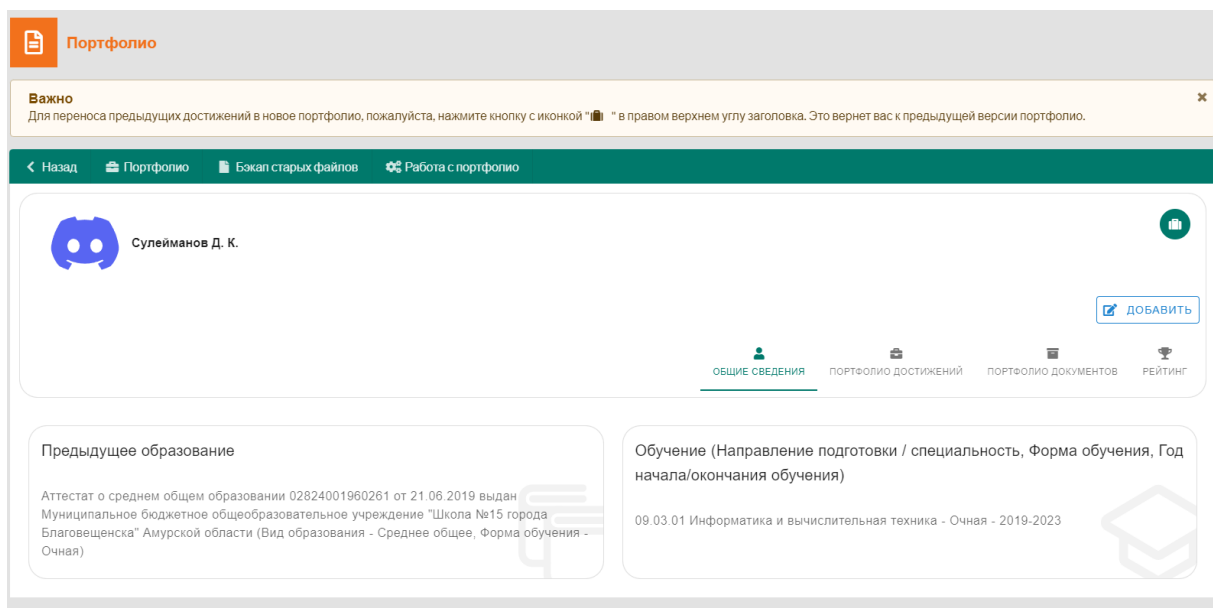


Рисунок 41 – Главный вид портфолио

На рисунке 42 изображена вкладка «Портфолио достижений» в системе, где представлен список достижений в виде карточек с краткой информацией. Эта вкладка предоставляет пользователю возможность ознакомиться с его достижениями в удобной форме.

Каждая карточка достижения содержит сжатую информацию о конкретном достижении, включая его описание и другие соответствующие детали. Кроме того, в рамках данной вкладки предусмотрены функции удаления и редактирования достижений, позволяющие пользователю управлять своим портфолио в соответствии с его потребностями и изменениями в профессиональной траектории.

Функция удаления достижений позволяет пользователю удалить конкретное достижение из своего портфолио, если оно стало неактуальным или больше не соответствует его целям и интересам. Таким образом, пользователь имеет возможность актуализировать и поддерживать актуальность своего портфолио в соответствии с текущими потребностями и стратегией.

Функция редактирования достижений позволяет пользователю изменять информацию о своих достижениях, например, добавлять новые детали, обнов-

лять описания или вносить другие необходимые изменения.

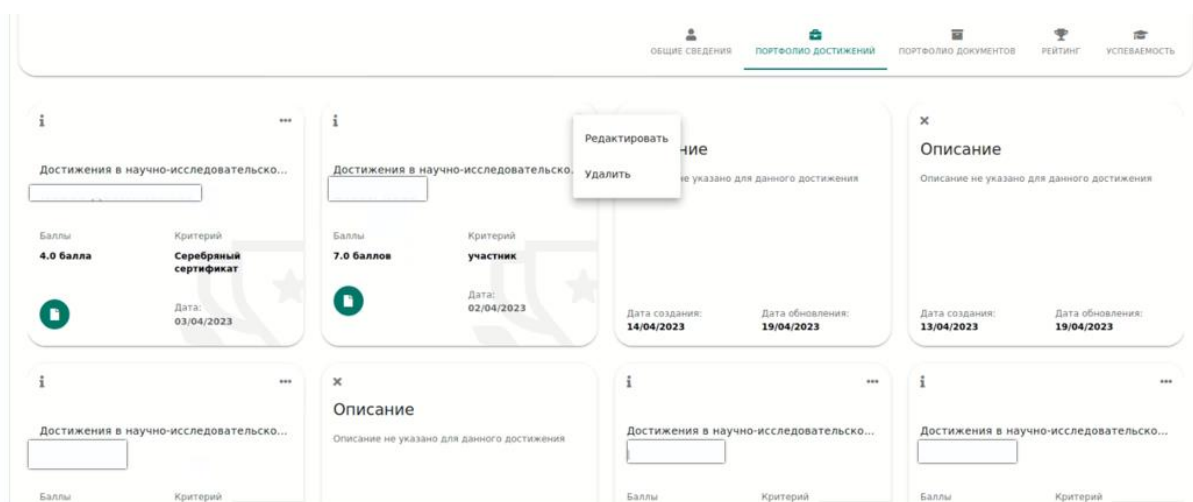


Рисунок 42 – Список достижений

Это предоставляет гибкость и возможность актуализировать свое портфолио в соответствии с развивающимися профессиональными целями и достижениями.

На рисунке 43 представлена форма, предназначенная для внесения информации о достижениях в портфолио. Эта форма обеспечивает пользователя возможность ввода различных данных, необходимых для полного описания достижения.

Рисунок 43 – Форма добавления достижения

В рамках данной формы пользователь должен указать следующую информацию: название достижения, которое является кратким описанием его сути или содержания. Кроме того, пользователю предоставляется возможность выбрать тип достижения из предложенного набора опций, отражающих различные категории достижений.

Также в форме предусмотрено поле выбора вида достижения, где пользователь может указать более конкретные аспекты или подкатегории своего достижения. Здесь пользователь может выбрать уровень или критерий, соответствующий его достижению, чтобы более точно определить его значение и значимость.

Пользователю также предоставляется возможность выбрать дату, связанную с достижением, чтобы указать, когда оно было достигнуто или официально признано.

Помимо указания даты и основной информации о достижении, пользователь имеет возможность внести дополнительное описание по своему усмотрению. Это поле позволяет раскрыть более подробные сведения о характеристиках достижения, его контексте и влиянии.

Кроме того, форма предоставляет пользователю возможность прикрепить файл, подтверждающий достижение. Это может быть, например, сертификат, диплом, свидетельство или другой документ, который подтверждает достижение и его достоверность. Прикрепление такого файла способствует более надежному документированию достижения и предоставляет дополнительную подтверждающую информацию.

Предусмотрена еще одна вкладка «Рейтинг», данная вкладка является информационной и позволяет увидеть рейтинг портфолио по всему университету, узнать личное место, и место интересующих студентов и их сумму баллов за все достижения.

Существует также вкладка «Рейтинг» в системе портфолио, которая представляет собой информационный раздел, представленный на рисунке 44, позволяющий пользователям узнать рейтинг своего портфолио в контексте всего университета. Эта вкладка обеспечивает пользователя информацией о личном месте

в рейтинге, а также позволяет ознакомиться с местом и суммой баллов других интересных студентов на основе их достижений.

Рейтинг по университету						
МЕСТО	ФИО	ФАКУЛЬТЕТ	ГРУППА	КОЛИЧЕСТВО	БАЛЛЫ	
☰ 2790	Сулейманов Д.К.	☰ ФМИИ	☰ 953-об	☰ 1	★ 0	
☰ 1	██████████	☰ ФМИИ	☰ 257-ом	☰ 6	★ 34	
☰ 2	██████████	☰ ФДИТ	☰ 281-об	☰ 5	★ 0	
☰ 3	██████████	☰ ФДИТ	☰ 084-об	☰ 2	★ 0	
☰ 4	██████████	☰ ФМИИ	☰ 257-об	☰ 0	★ 0	
☰ 5	██████████	☰ ФДИТ	☰ 281-об	☰ 0	★ 0	
☰ 6	██████████	☰ ЭФ	☰ 142-об1	☰ 1	★ 0	
☰ 7	██████████	☰ ФДИТ	☰ 289-об	☰ 0	★ 0	
☰ 8	██████████	☰ ФМИИ	☰ 257-ом	☰ 6	★ 0	
☰ 9	██████████	☰ ЭФ	☰ 242-ом1	☰ 0	★ 0	
☰ 10	██████████	☰ ФСН	☰ 266-ос	☰ 1	★ 0	
☰ 11	██████████	☰ ФМИИ	☰ 255-об	☰ 0	★ 0	

Рисунок 44 – Рейтинг портфолио по университету

Рейтинг портфолио отражает относительное положение пользователя в сравнении с другими студентами университета. Это предоставляет пользователю возможность оценить свой успех и продвижение в своих достижениях относительно своих коллег. Отображение личного места и суммы баллов, набранных за все достижения, позволяет студенту более ясно оценить свой академический прогресс и свою общую производительность.

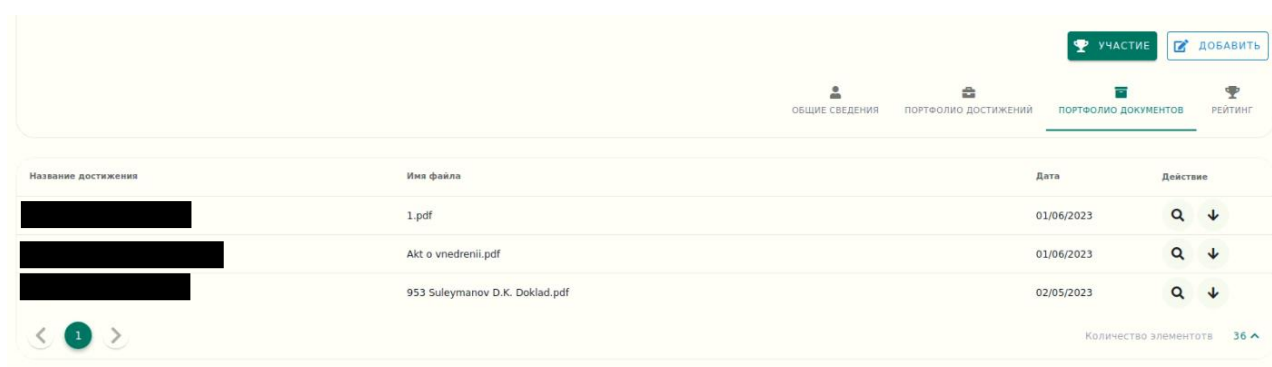
На рисунке 45 представлена вкладка «Портфолио документов», которая содержит все файлы, прикрепленные к портфолио студента. Список представлен в виде таблицы, где каждая строка представляет собой конкретное достижение, а каждый столбец содержит информацию о названии достижения и имени файла, которому оно прикреплено.

Студенты могут легко просматривать и управлять своими достижениями, используя эту вкладку. Доступна возможность просмотра загруженного файла или его скачивания, что делает использование портфолио более удобным и функциональным.

3.4.2 Работа с модулем «распределение стипендий»

Подсистему для личного кабинета «АмГУ», которая позволяет составить

протокол заседания стипендиальной комиссии по вопросу повышенной академической стипендии для студентов. Для этого требуется реализовать возможность выбора студентом достижений, которые будут использоваться при расчете баллов и определения места в рейтинге. Просмотр рейтинга по факультету для студентов. Заполнения данных протокола по шаблону, представленному в положении. Возможность просмотра достижений у каждого студента и удаление достижений, не являющихся корректным.



Название достижения	Имя файла	Дата	Действие
[REDACTED]	1.pdf	01/06/2023	🔍 ↓
[REDACTED]	Akt o vnedrenii.pdf	01/06/2023	🔍 ↓
[REDACTED]	953 Suleymanov D.K. Doklad.pdf	02/05/2023	🔍 ↓

Рисунок 45 – Список прикрепленных документов

Для дальнейшей разработки подсистемы будет использоваться интерактивная среда разработки «RubyMine» установленная на операционную систему Ubuntu 20.04, которая заранее была установлена на ПК с использованием загрузочного USB-носителя. Также потребуется RVM, что представляет собой программную платформу для «Unix»-подобных операционных систем, предназначенную для управления несколькими установками «Ruby» на одном устройстве.

При разработке было решено выбрать шаблон проектирования MVC, так как этот шаблон позволяет создавать приложения, в котором различные аспекты разделены, но достаточно тесно взаимодействуют друг с другом. Эта схема указывает расположение каждого вида логики в приложении. Логика пользовательского интерфейса относится к представлению. Логика ввода относится к контроллеру. Бизнес-логика размещается в модели. Это разделение позволяет работать со сложными структурами при создании приложения, так как обеспечивает одновременную реализацию только одного аспекта.

На рисунке 46 изображена веб-страница с меню выбора вкладок и кнопкой «Участие». Кнопка расположена в правом углу меню и выделена по сравнению

с остальным интерфейсом, привлекая внимание пользователя. Внешний вид кнопки изменяется в зависимости от состояния заявки. Если заявка подана, то кнопка будет иметь зеленоватый цвет, как и представлено на рисунке, в случае если заявка еще не была представлена, кнопка будет иметь иной стиль, вместо заливки будет использована обводка кнопки.

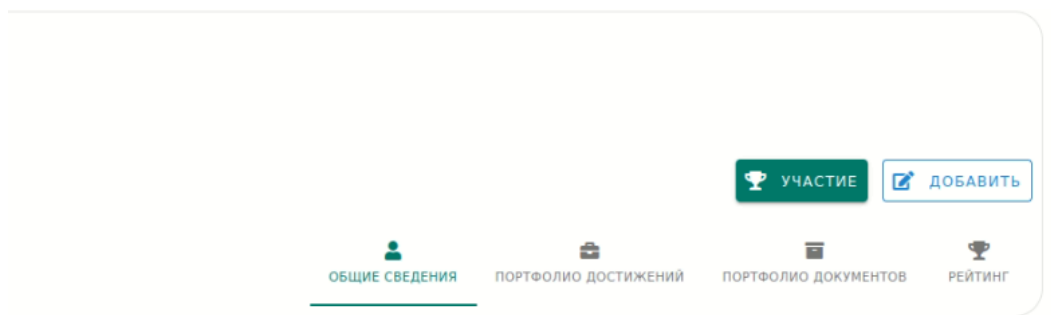


Рисунок 46 – Кнопка «Участие»

После нажатия на кнопку «Участие» справа от интерфейса появляется выдвижная боковая панель, которая представлена на рисунке 47. Данная боковая панель является интерактивным элементом, содержащим список достижений, доступных пользователю для выбора. Она предоставляет удобный и организованный способ представления достижений, сгруппированных по типу.

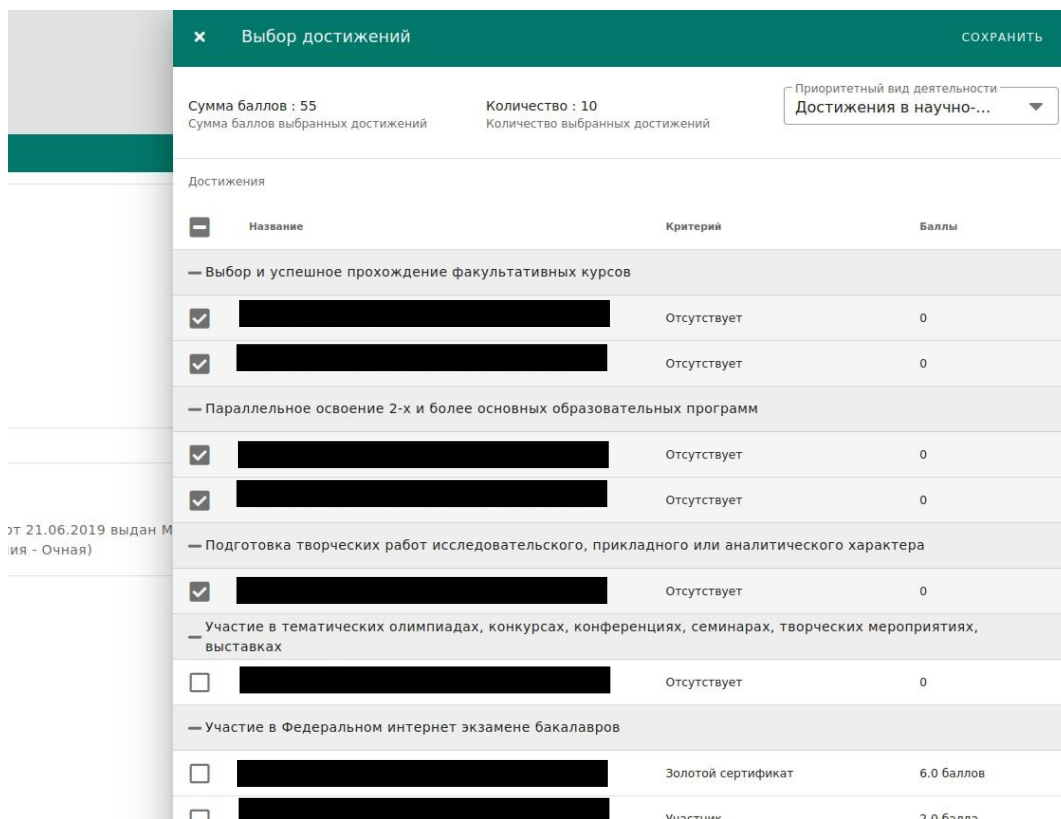


Рисунок 47 – Боковая панель для выбора достижений

Боковая панель представлена в виде выдвигаемого окна, которое при активации кнопки «Участие» открывается справа от основного интерфейса. Ее дизайн и структура обеспечивают удобство навигации и выбора достижений для пользователя.

Внутри боковой панели достижения представлены в виде таблицы, содержащей следующие столбцы. Название: указывает на наименование каждого отдельного достижения, доступного для выбора. Название предоставляет информацию о сути достижения. Критерий: отражает критерий, по которому данное достижение оценивается. Баллы: представляет количество баллов, назначенных каждому достижению в соответствии с его значимостью или сложностью. Баллы могут использоваться для расчета общего рейтинга студента.

Боковая панель организована таким образом, чтобы достижения были легко доступны и удобно классифицированы по виду деятельности. Это позволяет пользователям быстро просматривать список и выбирать достижения, которые наиболее соответствуют их интересам и критериям для участия в конкурсе или получения повышенной стипендии.

После выбора списка достижений и приоритетной деятельности, студент включается в рейтинг своего факультета, что позволяет отслеживать его текущую позицию. Рейтинговая таблица, представленная на рисунке 48, обеспечивает возможность студентам и другим заинтересованным лицам ознакомиться с этим рейтингом. Дополнительно, предусмотрены фильтрация списка по группе и система поиска, позволяющая осуществить поиск по информации в таблице.

На рисунке 49 представлена рейтинговая таблица, используемая комиссией для определения списка студентов, претендующих на получение повышенной стипендии. Данная таблица имеет структурированный формат и содержит информацию о студентах, их достижениях и оценках, а также дате создания заявки. Рейтинг формируется по факультетам и предоставляет основные данные, необходимые для принятия решения комиссией. В таблице присутствуют следующие столбцы:

– место, указывает позицию студента в рейтинге на основе его достижений

и оценок, студенты располагаются в порядке убывания их общего рейтинга;

- ФИО, содержит полные имена и фамилии студентов, представленных в таблице;

Рейтинг ФМИИ

Выбрать группу Поиск

МЕСТО	ФИО	ГРУППА	БАЛЛЫ	КОЛИЧЕСТВО	ДАТА
# 1	██████████	055-06	84	0	16/05/2023
# 2	██████████ М.	955-06	80	0	16/05/2023
# 3	██████████	053-06	75	0	16/05/2023
# 4	██████████	157-06	68	0	16/05/2023
# 5	██████████ В.	157-06	67	0	16/05/2023
# 6	██████████	852-06	66	0	16/05/2023
# 7	██████████ Е.	155-06	66	0	16/05/2023
# 8	██████████	953-06	61	5	16/05/2023
# 9	██████████	955-06	61	0	16/05/2023
# 10	██████████ А.	064-0м	56	0	16/05/2023
# 11	██████████	157-06	46	0	16/05/2023
# 12	██████████ А.А.	155-06	45	0	16/05/2023

Рисунок 48 – Рейтинговая таблица для студента

- группа, отображает группу, к которой принадлежит каждый студент;
- приоритетный тип деятельности, указывает на основную область, в которой студент достиг значительных результатов, может включать учебную деятельность, научно-исследовательскую работу, культурно-творческие достижения и спортивные достижения;

РЕЙТИНГ ПРОТОКОЛ ЗАСЕДАНИЯ

Рейтинг студентов ФМИИ

Место	ФИО	Группа	Приоритетный тип деятельности	Учебная	Научно-исследовательская	Общественная	Культурно-творческая	Спортивная	Суммарный балл	Количество достижений	Дата создания	Действия
<input type="checkbox"/> 1	██████████	055-06	За достижения в общественной деятельности	0	28	28	23	5	84	0	16/05/2023	
<input type="checkbox"/> 2	██████████	955-06	Достижения в научно-исследовательской деятельности	0	15	20	25	20	80	0	16/05/2023	
<input type="checkbox"/> 3	██████████	053-06	За достижения в общественной деятельности	0	11	24	26	14	75	0	16/05/2023	
<input type="checkbox"/> 4	██████████	157-06	Достижения в научно-исследовательской деятельности	0	12	10	21	25	68	0	16/05/2023	
<input type="checkbox"/> 5	██████████	157-06	Достижения в научно-исследовательской деятельности	0	23	24	15	5	67	0	16/05/2023	
<input type="checkbox"/> 6	██████████	852-06	За достижения в общественной деятельности	0	22	13	15	16	66	0	16/05/2023	
<input type="checkbox"/> 7	██████████	155-06	Достижения в учебной деятельности	0	7	29	23	7	66	0	16/05/2023	
<input type="checkbox"/> 8	██████████	953-06	Достижения в научно-исследовательской деятельности	0	61	0	0	0	61	6	16/05/2023	
<input type="checkbox"/> 9	██████████	955-06	За достижения в общественной деятельности	0	28	9	19	5	61	0	16/05/2023	
<input type="checkbox"/> 10	██████████	064-0м	Достижения в научно-исследовательской деятельности	0	17	5	8	26	56	0	16/05/2023	
<input type="checkbox"/> 11	██████████	157-06	Достижения в учебной деятельности	0	17	4	4	21	46	0	16/05/2023	
<input type="checkbox"/> 12	██████████	155-06	Достижения в научно-исследовательской деятельности	0	18	4	3	20	45	0	16/05/2023	

Рисунок 49 – Рейтинговая таблица для комиссии

- баллы за учебную деятельность, отражают оценку успеваемости студента в академической сфере;
- баллы за научно-исследовательскую деятельность, отражают оценку вклада студента в научные исследования и проекты;
- баллы за культурно-творческую деятельность, отражают оценку достижений студента в области культуры, искусства или других творческих областях;
- баллы за спортивную деятельность, отражают оценку достижений студента в спорте;
- суммарное количество баллов, показывает общий рейтинг студента, рассчитанный на основе суммы баллов за различные типы деятельности;
- дата создания заявки, указывает на дату, когда студент подал заявку на участие в конкурсе на повышенную стипендию;
- в последнем столбце таблицы представлена кнопка, позволяющая комиссии просмотреть достижения, выбранные.

На рисунке 50 представлена боковая панель, специально разработанная для просмотра достижений студента.

Список достижений							
Сумма баллов : 55							
Сумма баллов достижений							
Достижения FullName							
Название	Тип	Вид	Критерий	Баллы	файл	Действия	
[Redacted]	Достижения в научно-исследовательской деятельности	Финалист Национального межвузовского чемпионата "МОЛОДЫЕ ПРОФЕССИОНАЛЫ (WORLD SKILLS RUSSIA)"	3 место	11.0 баллов	Файл	[Иконка удаления]	
[Redacted]	Достижения в научно-исследовательской деятельности	Финалист Национального межвузовского чемпионата "МОЛОДЫЕ ПРОФЕССИОНАЛЫ (WORLD SKILLS RUSSIA)"	3 место	11.0 баллов	Файл	[Иконка удаления]	
[Redacted]	Достижения в научно-исследовательской деятельности	Финалист Национального межвузовского чемпионата "МОЛОДЫЕ ПРОФЕССИОНАЛЫ (WORLD SKILLS RUSSIA)"	3 место	11.0 баллов	Файл	[Иконка удаления]	
[Redacted]	Достижения в научно-исследовательской деятельности	Финалист Национального межвузовского чемпионата "МОЛОДЫЕ ПРОФЕССИОНАЛЫ (WORLD SKILLS RUSSIA)"	3 место	11.0 баллов	Файл	[Иконка удаления]	
[Redacted]	Достижения в научно-исследовательской деятельности	Финалист Национального межвузовского чемпионата "МОЛОДЫЕ ПРОФЕССИОНАЛЫ (WORLD SKILLS RUSSIA)"	3 место	11.0 баллов	Файл	[Иконка удаления]	

Рисунок 50 – Боковая панель с достижениями студента

В верхней части панели находится заголовок, в котором указана общая сумма баллов, отражающая накопленные достижения студента. Ниже заголовка располагается таблица, содержащая описание каждого достижения.

Для управления достижением предусмотрены две колонки: Прикрепленный файл: предусмотрена возможность просмотра прикрепленного файла у каждого достижения. Кнопки удаления: для удобства управления достижениями, в таблице предусмотрены соответствующие кнопки удаления.

Боковая панель обеспечивает удобный и структурированный способ просмотра достижений студента. Заголовок с общей суммой баллов позволяет быстро оценить общую успешность студента, а таблица достижений предоставляет подробную информацию о каждом достижении, включая возможность просмотра прикрепленных файлов и удаления неактуальных записей.

Форма, представленная на рисунке 51, разработана для составления протокола. Она предлагает удобный и структурированный способ заполнения необходимых данных согласно требованиям, изложенным в положении.

№	Ф.И.О студента курс, группа	Приоритетный вид деятельности	Учебная	Научно-исследовательская	Общественная	Культурно-творческая	Спортивная	Итоговый балл по всем видам деятельности	Голосование	Действие
# 1	Сулейманов Д.К 953-об	Достижения в научно-исследовательской деятельности	0	18	0	0	0	0	👍 1 🗳️ 0	🗑️ 👁️

Рисунок 51 – Форма протокола

Первым шагом в заполнении формы является выбор факультета из выпадающего списка. После этого следует указать председателя комиссии и секретаря, которые будут ответственными за составление протокола.

Следующим этапом является указание даты при помощи выпадающего меню, что обеспечивает точность и удобство в выборе нужного дня, месяца и года.

Далее необходимо заполнить список присутствующих. Это позволяет учесть всех участников, присутствующих в комиссии.

Под секцией «присутствовали» добавлена таблица, в которой указаны выбранные студенты для получения повышенной стипендии. Данная таблица соответствует шаблону, установленному в положении, и позволяет четко и структурированно представить список студентов, имеющих право на повышенную стипендию.

В данной таблице столбцы с кнопками «за» и «против» представлены для участия комиссии в процессе принятия решения по каждому заявлению, указанному в протоколе. При использовании этих кнопок, комиссия может открыто высказывать своё мнение и голосовать за или против данного заявления.

Кнопка «за» предназначена для выражения поддержки заявления, в то время как кнопка «против» используется для выражения негативного отношения к заявлению. Этот процесс публичного голосования позволяет комиссии принимать решения на основе коллективного мнения и предпочтений.

В конце таблицы имеется кнопка, которая позволяет удалить кандидата из протокола. Это может быть полезно в случае, если комиссия приняла решение исключить определенного кандидата из рассмотрения или протокола.

Также присутствует ещё одна кнопка, которая открывает боковое меню со всеми достижениями студента. При нажатии на эту кнопку отображается дополнительная информация о достижениях и успехах данного студента, которая может быть полезна комиссии при оценке его кандидатуры.

4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

В контексте соответствия нормативным документам и стандартам, следует учесть требования, установленные в отношении рабочей среды, эргономики и электробезопасности. Это включает в себя определение оптимальных параметров рабочего места, таких как освещение, шумовой фон, температурный режим и вентиляция, согласно соответствующим регулятивным документам. Необходимо также обеспечить удобство расположения оборудования, чтобы минимизировать физическую нагрузку на сотрудников и предотвратить возникновение повреждений опорно-двигательного аппарата или напряжения глаз.

При обеспечении здоровья и безопасности сотрудников при работе с ЭВМ следует учесть факторы, такие как эргономические характеристики клавиатуры, мыши и других устройств ввода-вывода, а также параметры дисплеев, такие как разрешение, размер и яркость. Также важно предоставить возможность периодической физической активности и отдыха, включая паузы для глаз и регулярные перерывы в работе.

4.1 Безопасность

4.1.1 Опасные и вредные факторы на рабочем месте пользователя ПЭВМ

В соответствии с ГОСТ 12.0.003-2015, при работе с персональными электронно-вычислительными машинами (ПЭВМ), существуют опасные и вредные факторы, которые могут оказывать негативное воздействие на здоровье пользователей. Эти факторы включают электростатические поля, риск поражения электрическим током, электромагнитное излучение, экстремальные температуры воздуха в рабочей зоне, монотонность трудового процесса, повышенный уровень шума, несоответствие естественному освещению и другие [2].

С целью предотвращения или снижения воздействия указанных вредных факторов на пользователей ПЭВМ, были разработаны требования, касающиеся помещений, освещения, уровня шума и организации рабочих мест. Также сформулированы рекомендации, которые следует учитывать пользователям ПЭВМ.

4.1.2 Организация рабочего места

Исследования и практический опыт показывают, что правильная организация рабочего пространства играет значительную роль в работоспособности, комфорте и благополучии сотрудников. Важность этого аспекта связана с несколькими факторами.

Чтобы избежать негативных факторов, сохранить здоровье и повысить эффективность работы операторов необходимо следовать следующим рекомендациям [3]:

- высота поверхности стола, предназначенной для работы, должна быть регулируема в пределах (680 – 800) мм для взрослых пользователей. В случае отсутствия такой возможности, высота рабочей поверхности должна составлять 725 мм;

- рабочий стол должен обеспечивать достаточное пространство для ног. Высота пространства для ног должна быть не менее 600 мм, ширина не менее 500 мм, глубина на уровне колен не менее 450 мм, а на уровне вытянутых ног не менее 650 мм;

- поверхность сиденья должна иметь ширину и глубину не менее 400 мм, а также быть регулируемой в пределах (400 – 550) мм по высоте и с углами наклона вперед до 15 градусов и назад до 5 градусов. Угол наклона спинки в вертикальной плоскости должен быть в пределах ± 30 градусов;

- сиденье должно быть оснащено стационарными или съемными подлокотниками длиной не менее 250 мм и шириной (50 – 70) мм. Подлокотники должны быть регулируемы над сиденьем в пределах (230 ± 30) мм и иметь внутреннее расстояние между ними в пределах (350 – 500) мм;

- рабочее место пользователя ПЭВМ должно быть оборудовано подставкой для ног, которая имеет ширину не менее 300 мм, глубину не менее 400 мм, регулируется по высоте в пределах 150 мм и имеет возможность регулировки угла наклона опорной поверхности подставки до 20 градусов.

- клавиатура должна быть расположена на поверхности стола на расстоянии 100 мм от края, обращенного к пользователю, или на специальной рабочей поверхности, отделенной от основного стола и регулируемой по высоте;

– 300 мм от края, обращенного к пользователю или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

Рабочий стол должен быть спроектирован таким образом, чтобы на его поверхности можно было оптимально разместить все необходимое оборудование, учитывая его количество и особенности, такие как размеры монитора, компьютера, клавиатуры, мышь и других элементов. Это также должно учитывать характер работы, которую выполняет человек, чтобы создать наиболее комфортные условия для выполнения задач, рисунок 52

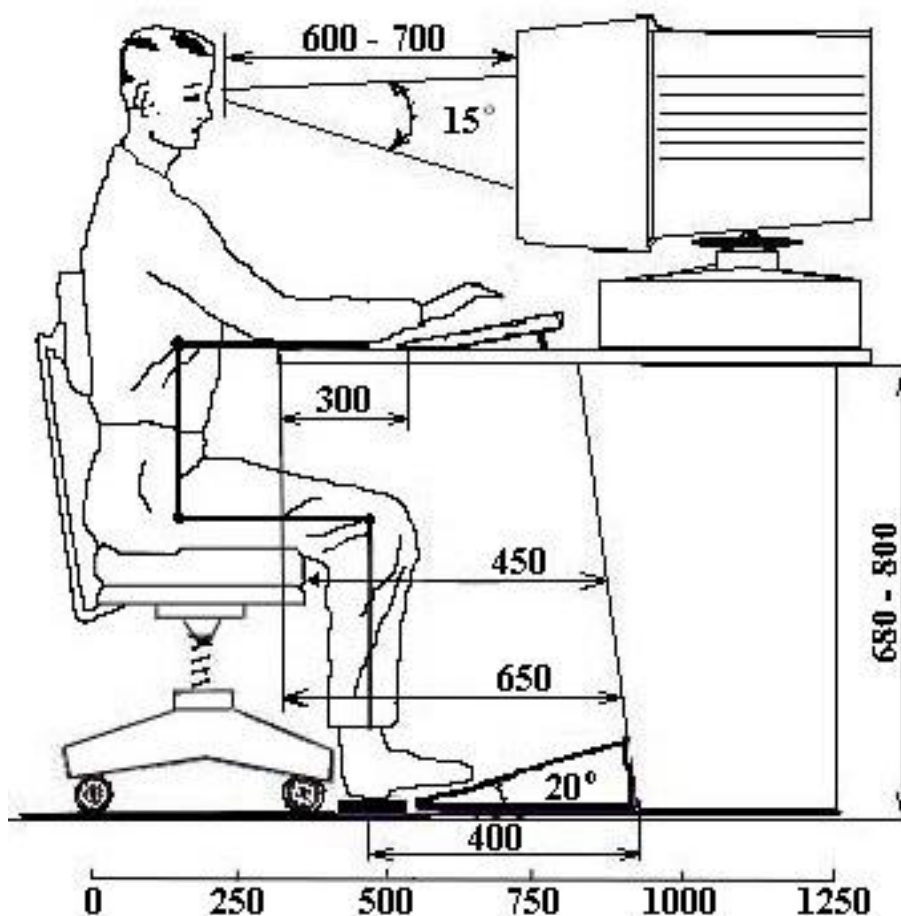


Рисунок 52 – Пример организация рабочего места оператора ПЭВМ

На одно рабочее место с ПЭВМ, на котором нет периферийного оборудования и установлен ЖК монитор, требуется 4,5 м², в противном случае, одно рабочее место с ПЭВМ должно занимать 6 м², так же при размещении рабочих мест с ПЭВМ расстояние между рабочими столами: в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора, должно быть не менее 2 м; а расстояние между боковыми поверхностями видеомониторов – не менее

1,2 м, как показано на рисунке 53.

Для осуществления самостоятельной работы по обслуживанию сервера применяются следующие требования и процедуры, сформулированные согласно научным и профессиональным принципам:

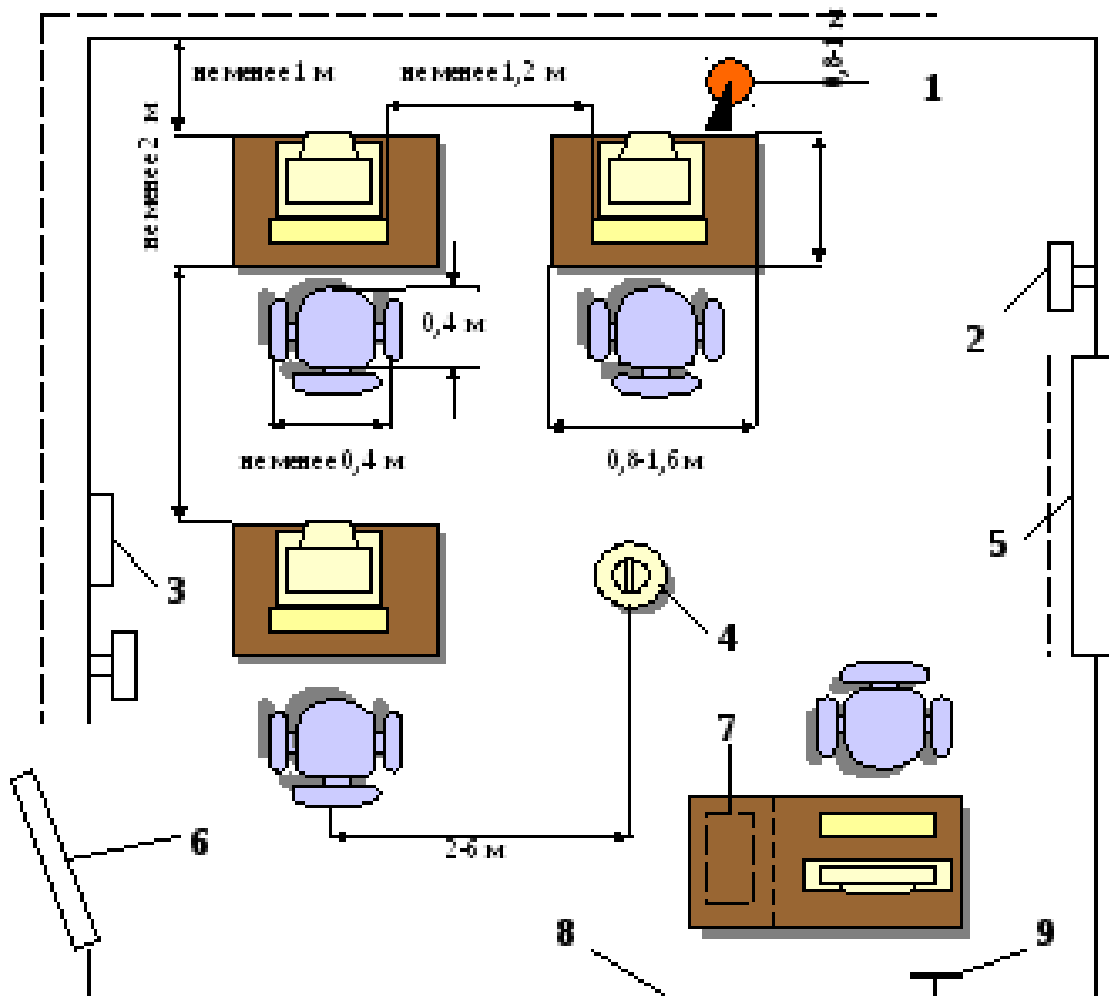


Рисунок 53 – Пример организации рабочего пространства

– работники, занимающиеся обслуживанием сервера, должны быть не моложе 18 лет, чтобы обеспечить сознательность и ответственность в выполнении своих обязанностей;

– при поступлении на работу требуется прохождение предварительного медицинского осмотра с целью оценки состояния здоровья работника и его пригодности для выполнения работы, связанной с обслуживанием сервера. Это позволяет определить, соответствует ли работник требованиям и нормам, предъявляемым к данной должности;

– для наблюдения за состоянием здоровья работников и своевременного

выявления ранних признаков негативного воздействия вредных производственных факторов на их здоровье требуется проведение периодических медицинских осмотров в процессе трудовой деятельности;

– вводный инструктаж, первичный инструктаж на рабочем месте и стажировка в течение (2 – 14) смен под руководством определенного лица, назначенного приказом по предприятию или распоряжением по подразделению, обеспечивают ознакомление работника с основными принципами работы и безопасности, а также практическое обучение навыкам, необходимым для эффективного выполнения обязанностей.

– работники должны успешно пройти обучение по безопасным приемам и методам выполнения работ, а также по оказанию первой помощи при несчастных случаях. Это гарантирует, что работники будут обладать необходимыми знаниями и умениями для обеспечения своей безопасности и безопасности окружающих при выполнении работ;

– работники должны пройти обучение и успешно пройти проверку знаний по электробезопасности, включающую требования и меры предосторожности при работе с электрическим оборудованием, чтобы минимизировать риск возникновения электрических травм и несчастных случаев.

Вышеуказанные требования и процедуры направлены на обеспечение безопасной и эффективной работы при обслуживании сервера. Они основываются на актуальных научных данных и профессиональных рекомендациях в данной области.

4.1.2 Освещение

Достаточное освещение является важным требованием для помещения, в котором можно установить ПЭВМ. Достаточное освещение играет решающую роль в повышении производительности труда, способствуя обучению зрительной эргономике. И наоборот, недостаточное освещение может привести к быстрой утомляемости, снижению внимания при работе за компьютером, ухудшению зрения, повышенной раздражительности из-за чрезмерной концентрации внимания. Для решения этих проблем могут использоваться различные типы освеще-

ния, в том числе естественное, искусственное, комбинированное и аварийное.

Естественное освещение естественным образом возникает во всех рабочих средах, где люди используют ПЭВМ. Конкретный тип естественного освещения зависит от расположения комнаты по отношению к источникам света, таким как окна. Это естественное освещение может быть классифицировано как боковое освещение, верхнее освещение или их комбинация.

Искусственное освещение в основном служит заменой недостаточного естественного освещения, особенно в ночное время или в помещениях с недостаточным дневным освещением. В таких случаях становится важным обеспечить повышенную освещенность всей рабочей зоны. Размещение источников света должно быть тщательно продумано и адаптировано к конкретным местам на рабочем месте, чтобы оптимизировать условия освещения.

Совмещенное освещение применяется в случае, когда естественное освещение недостаточно для обеспечения требуемого уровня освещенности. Такой тип освещения часто используется при выполнении точных работ, где требуется высокая точность и ясность видения.

Аварийное освещение предназначено для использования в случае аварийного отключения общего освещения. Оно обеспечивает минимальное освещение для обеспечения безопасности и возможности эвакуации людей.

В соответствии с СанПиНом 1.2.3685 предусмотрены четкие требования к освещению на рабочих местах, оснащенных ПЭВМ. Эти требования описывают оптимальные уровни освещенности, которые необходимы для обеспечения комфортного и безопасного трудового процесса ПЭВМ. Согласно данным требованиям равномерность освещенности должна быть не менее 0,6 в основных помещениях (в учебных кабинетах черчения и рисования – не менее 0,7; на ледовых аренах – не менее 0,5; для спортивных залов разного назначения в физкультурно-оздоровительных организациях – 0,7), в прочих вспомогательных помещениях – не менее 0,4. Для обучающихся с нарушениями зрения учебные помещения и читальные залы оборудуются комбинированной системой общего искусственного и местного освещения [4].

Необходимо обратить особое внимание на общие требования к организации рабочих мест с ПЭВМ с учетом естественного освещения. Согласно указаниям, рабочие места с ПЭВМ должны быть размещены относительно источников естественного освещения таким образом, чтобы световой поток падал сбоку, с преобладанием слева, согласно рисунку 54.

Совокупная уровень освещенности, создаваемый общим и местным освещением, должен соответствовать следующим значениям в зависимости от группы обучающихся: для обучающихся с высокой степенью осложненной близорукости и высокой степенью дальновидности – 1000 люкс; для обучающихся с поражением сетчатки и зрительного нерва (без светобоязни) – от 1000 до 1500 люкс; для обучающихся со светобоязнью - не более 500 люкс.

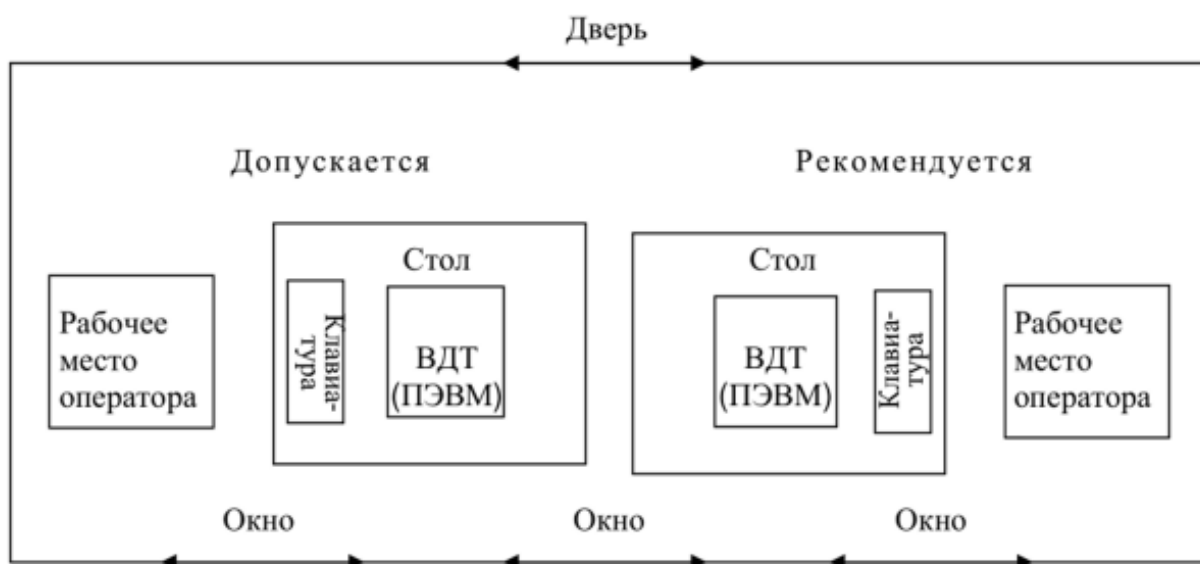


Рисунок 54 – Схема расположения рабочих мест относительно светопроемов

Для обеспечения необходимого уровня освещения рекомендуется использовать люминесцентные лампы, обладающие высокой светоотдачей и спектральным составом, близким к естественному. Освещенность на поверхности рабочего стола должна находиться в диапазоне от 300 до 500 люкс.

При внутренней отделке помещений, где расположены ПЭВМ, следует использовать материалы с диффузно-отражающими свойствами. Коэффициент отражения для потолка должен составлять от 0,7 до 0,8, для стен – от 0,5 до 0,6, а для пола – от 0,3 до 0,5.

4.1.3 Микроклимат

Микроклимат в производственных помещениях представляет собой комплекс нормированных факторов, таких как температура, влажность, тепловое излучение и другие, которые оказывают воздействие на теплообмен организма человека и существенно влияют на его самочувствие, работоспособность, здоровье и производительность труда. Поэтому соблюдение оптимальных параметров микроклимата на рабочих местах является важной задачей в области охраны труда с целью обеспечения безопасности и комфорта работников.

При работе ПЭВМ значительный вклад в формирование микроклимата производственного помещения вносит само устройство. ПЭВМ выделяет тепло, что приводит к повышению температуры окружающей среды и организма человека, что, в свою очередь, может отрицательно сказываться на работоспособности и производительности. Поддержание оптимального уровня температуры в рабочем помещении позволяет создать безопасные и комфортные условия для работы с ПЭВМ.

Для обеспечения оптимального микроклимата в помещении применяются специальные вентиляционные системы. Вентиляционная система представляет собой комплекс технических мероприятий, направленных на обновление воздуха в помещении и поддержание оптимальных метеорологических параметров. Главная цель такой системы заключается в подаче свежего воздуха извне. Для достижения максимального комфорта часто применяется естественная вентиляция, а в теплое время года может быть установлена система кондиционирования для полной регулировки микроклимата в рабочем помещении.

Системы кондиционирования используются для поддержания постоянной температуры, влажности и очистки воздуха от вредных веществ. Они успешно решают проблему задержки углекислого газа в помещении.

Для обеспечения стабильной и равномерной температуры воздуха в рабочих помещениях в холодный период года применяются системы отопления. Расчет таких систем осуществляется с учетом потерь тепла через ограждающие конструкции здания и нагрева холодного воздуха, проникающего в помещение. Существуют различные типы систем отопления, включая водяные, паровые, воз-

душные и комбинированные системы.

Системы водяного отопления являются наиболее эффективными санитарно-гигиенически, а также привлекательными с точки зрения их надежности и возможности регулировки температуры в широком диапазоне. Они широко применяются в помещениях, где находятся рабочие места с компьютерами. В холодный период года рекомендуется поддерживать температуру в помещении не выше (22 – 24) °С, а в теплый период – в пределах (20 – 25) °С. Относительная влажность воздуха должна находиться в диапазоне (40 – 60) процентов, а скорость движения воздуха не должна превышать 0,1 метра в секунду.

4.1.4 Требования к графическому интерфейсу

В процессе разработки данной системы применялись компоненты, описанные в ГОСТ Р ИСО 9241-161-2016, которые определяют различные методы ввода данных и взаимодействия с графическими элементами пользовательского интерфейса. В соответствии с указанным стандартом, доступны следующие методы ввода: ввод данных с использованием клавиатуры, использование компьютерной мыши, ручки, распознавание жестов, отслеживание положения глазного яблока, а также речевой ввод с использованием голосовых команд и распознавание речи.

При проектировании разработчики должны уделять особое внимание трем составляющим интерфейса пользователя: интерактивность, информативность, декоративность, рисунок 55 [5].

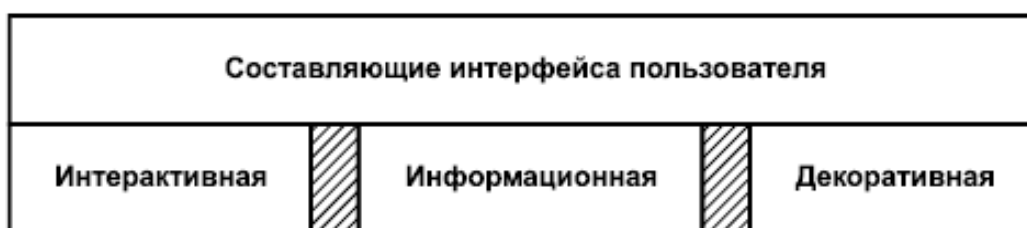


Рисунок 55 – Три основных составляющих интерфейса

При наличии нескольких методов ввода пользователю предоставляется возможность использовать все доступные методы взаимодействия. Однако в данной работе был выбран классический подход, основанный на применении клавиатуры и мыши для взаимодействия с системой.

Элементы интерфейса пользователя должны соответствовать требованиям,

установленным для обеспечения доступности. Это подразумевает соответствие всем применимым требованиям и рекомендациям, которые обеспечивают доступность программного обеспечения для всех пользователей, включая тех, у которых есть ограничения или потребности в использовании вспомогательных технологий.

Важно создавать программное обеспечение, которое учитывает потребности разнообразных пользователей и обеспечивает равный доступ к информации и функциональности. Это помогает создать более инклюзивное и доступное пользовательское взаимодействие.

Веб-приложение, разработанное с соблюдением принципов классического пользовательского интерфейса, включает элементы и принципы, определенные в ГОСТ Р ИСО 9241-161-2016. Все методы взаимодействия с системой соответствуют современным принципам пользовательского интерфейса и основаны на широко применяемых средствах ввода.

4.2 Экологичность

ПЭВМ включают в себя множество компонентов, содержащих токсичные вещества, которые представляют угрозу для человека и окружающей среды. К таким веществам относятся следующие примеры [6]:

- ртуть, содержащаяся в подсветке ЖК-мониторов, является опасным веществом, способным нанести вред мозгу и нервной системе;
- щелочи, содержащиеся в щелочных аккумуляторах и источниках бесперебойного питания, могут прожигать слизистые оболочки и кожу, представляя опасность для здоровья;
- никель и цинк, присутствующие в материнских платах и батареях питания ноутбуков, могут вызывать дерматит и иметь негативное воздействие на организм;
- поливинилхлорид, содержащийся в кабелях, используемых для подключения электронных устройств, может наносить вред нервной системе и быть причиной развития раковых заболеваний.

Из-за наличия этих вредных веществ, корректная утилизация ПЭВМ тре-

бует применения специальных комплексных методов. Данный комплекс мероприятий включает следующие этапы: сортировка металлических и неметаллических частей ПЭВМ, металлические части направляются на переплавку для последующего использования в производстве, неметаллические компоненты ПЭВМ подлежат специальной утилизации.

В настоящее время идет разработка и внедрение малоотходных технологий в различных промышленных секторах. Однако полный переход ведущих отраслей промышленности на безотходную технологию представляет собой сложную задачу, требующую решения технологических, конструкторских и организационных проблем. Необходимо приступить к решению этого комплекса задач, чтобы достичь полной безопасности при утилизации ПЭВМ и минимизировать их негативное влияние на окружающую среду.

4.3 Чрезвычайные ситуации

4.3.1 Аварийные ситуации

В процессе работы возможны различные аварийные ситуации, включающие обрыв проводов питания, неисправность заземления, повреждение электрооборудования, инженерных коммуникаций, а также повреждение конструктивных элементов здания или помещения.

В случае возникновения любой чрезвычайной ситуации, упомянутой выше, а также при наблюдении ухудшения самочувствия, следует незамедлительно принять следующие меры: остановить производственные работы, при необходимости обеспечить отключение электроэнергии, в случае наличия пострадавших обеспечить оказание первой помощи, гарантировать открытие аварийных выходов и эвакуацию персонала, сообщить руководителю работ о принятых мерах и действовать в соответствии с полученными указаниями, подать отчет оперативному дежурному о принятых мерах.

Эффективное управление аварийными ситуациями в процессе работы является неотъемлемой частью обеспечения безопасности персонала и сохранения инфраструктуры. Соблюдение указанных мер позволяет минимизировать риски, связанные с аварийными ситуациями, и обеспечить своевременные действия для

предотвращения возможных негативных последствий.

В случае несчастного случая или происшествия, сотрудник, присутствующий вблизи, обязан немедленно предоставить первую помощь и сообщить об этом оперативному дежурному или начальнику отдела. Если человек оказался под воздействием электрического напряжения, необходимо немедленно отключить источник питания и освободить его от электрического тока. В случае, когда невозможно отключить электроустановку, следует принять другие меры для освобождения пострадавшего.

Для освобождения пострадавшего от электрического тока можно использовать средства защиты, такие как канат, палка или другой сухой предмет, который не проводит электричество. Также можно попытаться оттянуть пострадавшего за сухую одежду, избегая контакта с металлическими предметами или незащищенными участками тела. Для собственной безопасности оказывающего помощь рекомендуется использовать диэлектрические перчатки для изоляции рук. В случае их отсутствия можно обмотать руки сухой одеждой.

Принятие данных мер позволяет обезопасить пострадавшего и минимизировать риск дальнейших повреждений. Необходимо придерживаться научных принципов и рекомендаций в области безопасности при оказании помощи, сосредотачиваясь на предупреждении дополнительных травм и сохранении жизни пострадавшего.

4.3.2 Меры пожарной безопасности на рабочих местах

При проектировании размещения технологического и прочего оборудования необходимо уделять особое внимание обеспечению свободного доступа к путям эвакуации и выходам в случае чрезвычайных ситуаций [7]. Важно обеспечить такую установку персонального компьютера и монитора, которая исключает возможность их падения. Необходимо избегать следующих методов установки компьютерного оборудования:

– размещение в нишах мебельных «стенок», тумбочках и аналогичных конструкциях, которые могут ограничивать доступ и создавать препятствия при необходимости эвакуации;

- установка вблизи электронагревательных приборов и горючих материалов, таких как тюли, занавески, гардины, декоративные элементы и новогодние елки. Расстояние между компьютером и такими предметами должно быть не менее 1-го метра;

- размещение компьютера ближе чем на 0,7 метра от проходов, путей передвижения и маршрутов эвакуации людей.

Перед началом использования персонального компьютера рекомендуется выполнить следующие меры безопасности:

- осмотреть место установки компьютера и монитора с учетом аспектов безопасности, чтобы убедиться, что все требования безопасности соблюдены;

- проверить внешнее состояние компьютера, монитора, электрического шнура и розетки, чтобы убедиться в их исправности. Если корпус, электрический шнур, розетка или задняя крышка повреждены, использование компьютера не рекомендуется;

- удалить горючие материалы, такие как салфетки, накидки, книги, газеты, декоративные элементы и прочее, которые находятся на, над или около компьютера и монитора;

- убедиться, что вентиляционные отверстия в задней крышке компьютера и монитора не заблокированы предметами;

- обеспечить наличие огнезащитной ткани или огнетушителя рядом с компьютером;

- соблюдение указанных мер безопасности при работе с компьютерной техникой поможет снизить риск возникновения пожара и создать безопасные условия для эксплуатации.

ЗАКЛЮЧЕНИЕ

В результате проведения анализа предметной области была осуществлена тщательная экспертиза организационной структуры предприятия и проведен анализ имеющихся сред разработки и конкурентов. Этот этап исследования позволил достичь поставленных целей и определить окончательный вид программного продукта.

Проектирование системы включало подробную разработку архитектуры системы и выделение необходимых модулей для последующей реализации. Также проведен анализ различных моделей жизненного цикла и архитектуры проекта. Благодаря выбранной модели жизненного цикла была обеспечена эффективная разработка системы с соблюдением принципов параллельного тестирования и разработки модулей.

Разработка базы данных, соответствующей требованиям заказчика, учитывала полноту и точность данных, необходимых для безошибочного функционирования продукта. Также был разработан план реализации модулей, отвечающих за отображение и взаимодействие данных. Полученные результаты на данном этапе послужили основой для создания схемы взаимно-действия моделей системы, что упростило процесс разработки программного продукта.

В рамках предметной области была разработана подсистема для личного кабинета, предназначенного для распределения повышенных стипендий среди студентов Амурского государственного университета. Данная подсистема является инновационным решением, позволяющим эффективно и справедливо распределять стипендии и мотивировать студентов на достижение новых высот.

Она учитывает выбранные индивидуальные достижения, которые студенты могут прикреплять в разделе личного кабинета, известном как «портфолио», такие как, академические достижения, участие в проектах и научных конференциях. Общий алгоритм распределения стипендий был разработан с учетом всех этих факторов, что позволяет обеспечить справедливость и прозрачность процесса.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1 Положение о порядке назначения и выплаты, повышенных государственных академических стипендий студентов [Электронный ресурс]: положение, утвержденное приказом ректора от 15.06.2021 №227-ОД. Доступ из сведений об образовательной организации сайта «АмГУ».

2 internet-law.ru: Информационный портал [Электронный ресурс] – Режим доступа: – 07.06.2023.

3 Пособие по безопасной работе на персональных компьютерах [Текст] / разраб. В. К. Шумилин. - М. : НЦ ЭНАС, 2005. - 28 с.

4 rospotrebnadzor.ru: федеральная служба по надзору в сфере защиты прав потребителей и благополучия человека [Электронный ресурс] – Режим доступа: <https://clck.ru/Wgw5x> – 07.06.2023.

5 meganorm.ru: Информационный портал [Электронный ресурс] – Режим доступа: <https://meganorm.ru/Data2/1/4293750/4293750544.pdf> – 07.06.2023.

6 Экология компьютерной техники [Электронный ресурс]: учебное пособие / И. Г. Гетия, В. К. Шумилин [и др.]. – Москва: Московский государственный университет приборостроения и информатики, 2007. – 25 с. – Режим доступа: https://studopedia.ru/14_131205_ekologiya-kompyuternoy-tehniki.html – 07.06.2023.

7 СП 1.13130.2020 Система противопожарной защиты. Эвакуационные пути и выходы. Введ. 19-09-2020

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Асалханов, П. Г. Web-программирование: JavaScript: учебное пособие / П. Г. Асалханов. – Иркутск: Иркутский ГАУ, 2020. – 123 с.

2 Батенькина, О. В. Юзабилити информационных систем : учебное пособие / О. В. Батенькина, О. Н. Ткаченко. – Омск : ОмГТУ, 2015. – 144 с. – ISBN 978-5-8149-2095-9. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/149059> (дата обращения: 07.03.2023).

3 Гутман, Г. Н. Объектно-реляционная СУБД PostgreSQL: учебное пособие / Г. Н. Гутман. – Самара: Самарский государственный технический университет, ЭБС АСВ, 2016. – 125 с.

4 ГОСТ Р 7.0.100-2018. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления = System of standards on information, librarianship and publishing. Bibliographic record. Bibliographic description. General requirements and rules : национальный стандарт Российской Федерации : издание официальное : утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 3 декабря 2018 г. N 1050-ст : введен впервые : дата введения 2019-07-01 / разработан Федеральным государственным унитарным предприятием "Информационное телеграфное агентство России (ИТАР-ТАСС)", филиал "Российская книжная палата", Федеральным государственным бюджетным учреждением "Российская государственная библиотека", Федеральным государственным бюджетным учреждением "Российская национальная библиотека" - М.: Стандартинформ, 2018. – Текст : непосредственный.

5 ГОСТ 19.101-77. Единая система программной документации (ЕСПД). Виды программ и программных документов = Unified system for program documentation. Types of programs and program documents : межгосударственный стандарт : издание официальное : утвержден и введен в действие Постановлением Государственного комитета стандартов Совета Министров СССР от 20 мая

1977 г. N 1268 : введен впервые : дата введения 1980-01-01 / Электронный текст документа подготовлен ЗАО "Кодекс" и сверен по: официальное издание Единая система программной документации: Сб. ГОСТов . - М.: Стандартиформ, 2010. - Текст : электронный.

6 ГОСТ 19.102-77. Единая система программной документации (ЕСПД). Стадии разработки = Unified system for program documentation. Development stages : межгосударственный стандарт : издание официальное : утвержден и введен в действие Постановлением Государственного комитета стандартов Совета Министров СССР от 20 мая 1977 г. N 1268 : введен впервые : дата введения 1980-01-01 / Электронный текст документа подготовлен ЗАО "Кодекс" и сверен по: официальное издание Единая система программной документации: Сб. ГОСТов . – М.: Стандартиформ, 2010. – Текст : электронный.

7 ГОСТ 19.201-78. Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению = Unified system for program documentation. Technical specification for development. Requirements for contents and form of presentation : межгосударственный стандарт : издание официальное : утвержден и введен в действие Постановлением Государственного комитета СССР по стандартам от 18 декабря 1978 г. N 3351 : введен впервые : дата введения 1980-01-01 / Электронный текст документа подготовлен АО "Кодекс" и сверен по: официальное издание Единая система программной документации: Сборник национальных стандартов. - М.: Стандартиформ, 2010. - Текст : электронный.

8 Инновации и научно-техническое творчество молодежи. 20-21 апреля 2022 г : материалы конференции / RU. – Новосибирск : СибГУТИ, 2022. – 1323 с. – ISBN 978-5-91434-072-5. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/257177> (дата обращения: 07.03.2023).

9 Маран, М. М. Программная инженерия : учебное пособие для вузов / М. М. Маран. – 3-е изд., стер. – Санкт-Петербург : Лань, 2022. – 196 с. – ISBN 978-5-8114-9323-4. – Текст : электронный // Лань : электронно-библиотечная система.

– URL: <https://e.lanbook.com/book/189470> (дата обращения: 07.03.2023).

10 Современные технологии разработки программного обеспечения : учебно-методическое пособие / составитель Н. А. Федыкова. – Брянск : Брянский ГАУ, 2022. – 58 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/305087> (дата обращения: 07.03.2023).

11 Соловцова Л.А., Назаренко Н.В. Базы данных: Методические указания по выполнению курсовой работы. – М: АмГУ, 2016. – 62 с.

12 Стандарт организации. Оформление выпускных квалификационных и курсовых работ (проектов) [Электронный ресурс] / АмГУ; разработ. Л. А. Проказина, Н. А. Чалкина, С. Г. Самохвалова. – Введ. с 05.04.2018. - Благовещенск: [б. и.], 2018. - 75 с. URL: https://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/9702.pdf (дата обращения 05.03.2023).

13 Хартл, М. Ruby on Rails для начинающих / М. Хартл. – Москва : ДМК Пресс, 2017. – 572 с. – ISBN 978-5-97060-429-8. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/90110> (дата обращения: 07.03.2023).

14 Шёниг, Г. –. PostgreSQL 11. Мастерство разработки / Г. -. Шёниг ; перевод с английского А. А. Слинкина. – Москва : ДМК Пресс, 2020. – 352 с. – ISBN 978-5-97060-671-1. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/131714> (дата обращения: 07.03.2023).

15 Янцев, В. В. JavaScript. Как писать программы: учебное пособие для вузов / В. В. Янцев. – Санкт-Петербург: Лань, 2022. – 200 с.

16 Янцев, В. В. JavaScript. Готовые программы: учебное пособие для вузов / В. В. Янцев. – Санкт-Петербург: Лань, 2021. – 200 с.

17 Rails guides. [Электронный ресурс]. URL: <https://guides.rubyonrails.org> (дата обращения 13.03.2023).

ПРИЛОЖЕНИЕ А

Техническое задание

1 Введение

1.1 Наименование программы

Наименование программного модуля – «Распределение повышенной стипендии»

1.2 Краткая характеристика области применения

Данная программа предназначена для автоматизации процесса подачи заявлений на повышенную стипендию студентами в учебном заведении. Она обеспечивает удобство процесса и уменьшает вероятность ошибок. Программа также предоставляет функционал для просмотра таблицы первых 25, 50, 80 студентов, претендующих на повышенную стипендию по каждому факультету, а также возможность выбора нужного количества обучающихся для выдачи повышенной стипендии из таблицы участников на конкурс. Администратор программы имеет доступ к разделу портфолио, где может настроить и добавить виды и типы достижений, учитываемых при конкурсе на повышенную стипендию, что обеспечивает более объективный и справедливый конкурс.

2 Основания для разработки

Основанием для разработки служит выпускная квалификационная работа студента «Амурского Государственного Университета» по закреплённой теме: «Разработка ПО для личного кабинета студента «Амурского Государственного Университета», «Распределение повышенной стипендии».

3 Назначение разработки

Модуль будет использоваться в Амурском государственном университете несколькими группами пользователей: студент, заместитель декана по внеучебной работе, администратор системы, сотрудник учебно-методического управления.

3.1 Функциональное назначение

Данная автоматизированная программа организации подачи заявлений

Продолжение ПРИЛОЖЕНИЯ А

на повышенную стипендию предназначена для упрощения и улучшения процесса получения повышенной стипендии студентами, а также обеспечения более справедливого конкурса. В рамках данной программы студенты имеют возможность заполнять свою портфолио в системе университета и следить за рейтинговой таблицей студентов, участвующих в конкурсе на получение повышенной стипендии.

Зам декана по воспитательной работе факультета, далее именуемый оператор или зам декана ПВР, имеют возможность отбирать студентов из рейтинговой таблицы для назначения повышенной стипендии. Они также могут проверять заполненные студентом данные и достижения и в случае некорректно или ложно заполненных данных, удалить неверно заполненные данные кандидата на повышенную стипендию.

Администратор и сотрудник Управления менеджмента учебного заведения имеют возможность заполнения портфолио, назначения и удаления студентов из кандидатов на повышенную стипендию, а также добавление, редактирование и удаление видов и типов достижений в системе. Кроме того, они могут добавлять места за участие и назначать им баллы, а также добавлять и удалять единицы измерения достижения.

Таким образом, данная автоматизированная программа организации подачи заявлений на повышенную стипендию обеспечивает более удобный, прозрачный и справедливый процесс получения повышенной стипендии студентами, а также организаторами конкурса.

3.2 Эксплуатационное назначение

Эта автоматизированная программа предназначена для упрощения и улучшения процесса получения повышенной стипендии студентами, а также организаторам конкурса. Она обеспечивает возможность подачи заявлений через заранее заполненную портфолио в личном кабинете, что значительно повышает удобство процесса и уменьшает вероятность ошибок. Программа также предо-

Продолжение ПРИЛОЖЕНИЯ А

ставляет функционал для просмотра таблицы первых 25, 50, 80 студентов, претендующих на повышенную стипендию по каждому факультету, а также возможность выбора нужного количества обучающихся для выдачи повышенной стипендии из таблицы участников на конкурс.

Администратор программы имеет доступ к разделу портфолио, где может настроить и добавить виды и типы достижений, учитываемых при конкурсе на повышенную стипендию, что обеспечивает более объективный и справедливый конкурс.

4 Требования к программе или программному изделию

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

Для студента доступны следующие функции:

– Добавление достижений и документов - студент может внести данные о своих достижениях, документах, научных работах и отзывах в разделе "портфолио". Для этого пользователю необходимо перейти в соответствующий раздел и нажать кнопку "добавить достижение". После этого в появившейся форме пользователь должен заполнить следующие поля:

- название достижения (обязательное поле);
- категория достижения (обязательное поле);
- вид достижения (обязательное поле);
- призовое место (обязательное поле);
- дата, указанная в достижении (обязательное поле);
- описание достижения (необязательное поле);
- файл в формате pdf подтверждающий достижение, копия распечатанного достижения, ссылка на достижение (если такая возможность имеется), (обязательное поле).

– Просмотр добавленных данных - после внесения достижений студент может просмотреть их в разделе "портфолио", где каждой вкладке относится

Продолжение ПРИЛОЖЕНИЯ А

определенная категория достижений: общие сведения, портфолио достижений, портфолио документов, портфолио работ, портфолио отзывов, успеваемость.

– Редактирование и удаление данных - студент может в любой момент изменять или удалять свои данные в разделе "портфолио", если это необходимо.

– Просмотр таблицы рейтинга студентов – в разделе "повышенная стипендия" студент может просмотреть таблицу рейтинга студентов выбранного факультета, участвующих в конкурсе на повышенную стипендию. Это позволяет студентам узнать о своем месте в рейтинге и возможности получения повышенной стипендии. Таблица содержит данные студентов, участвующих в конкурсе на повышенную стипендию. Рейтинг отсортирован по шансу получения стипендии, где первый студент имеет самый высокий шанс. Расчёт производится на основе данных, представленных в разделе "портфолио". Баллы, указанные в положении ПУД СМК 145-2021, назначаются в зависимости от достижений и других параметров.

Оператор, занимающийся выбором обучающихся для получения повышенной стипендии, имеет следующие возможности:

- просмотр списка студентов;
- просмотр и проверка каждого достижения (подразумевается все введенные данные в разделе «портфолио») на наличие ошибок или намеренно неправильно введенных данных;
- выбор студента в число тех, кто будет получать повышенную стипендию;
- удаление неверно заполненных достижений участником;
- формирование списка всех студентов для получения повышенной стипендии;

При отклонении обучающегося от участия, оператору требуется описать по каким причинам данный участник был отклонён, после чего учащийся получит уведомление с описанием причины отказа о выдаче повышенной стипендии. Данная функция предназначена для тех случаев, когда имеются шансы получить

Продолжение ПРИЛОЖЕНИЯ А

стипендию студентом, но из-за некорректных или намерено ложных внесенных данных в портфолио, должны быть исключены из рейтинга для освобождения места в таблице для следующих участников.

После формирования списка студентов, оператор может сформировать автоматически для каждого участника файлы для печати, состоящий из:

- автоматически заполненной формы заявления для участия в конкурсе;
- всех достижений, прикрепленных студентом в своей портфолио в формате pdf;
- автоматическое формирование протокола заседания по форме указанном в положении.

Администратор программы имеет доступ к разделу портфолио, где может настроить и добавить виды и типы достижений, учитываемых при конкурсе на повышенную стипендию. Он также может просмотреть таблицу рейтинга студентов выбранного факультета, участвующих в конкурсе на повышенную стипендию, чтобы определить, кто получит повышенную стипендию. Администратор должен иметь доступ к функционалу резервного копирования данных и восстановления в случае сбоя системы. Кроме того, администратор должен иметь возможность управлять доступом к конфиденциальным данным студентов и защищать их паролем. При необходимости администратор должен иметь возможность вносить изменения в код программы для обеспечения ее безопасности и надежности.

4.1.2 Требования к организации входных и выходных данных

Данные о портфолио и студентах будет хранится в базе данных, далее БД, «PostgreSQL». Доступ к БД будет реализован через «Человеко-машинный интерфейс» при помощи архитектуры запросов RESTful с использованием фреймворк «VueJs». Пользователь при помощи интерфейса будет слать запрос на сервер, где находится БД и СУБД, и получать ответ от сервера в виде данных, которые сразу же отображаются в представлении.

Продолжение ПРИЛОЖЕНИЯ А

Проверка введенных данных изначально происходит на клиенте где проверяется:

- заполнены ли все поля с атрибутом «Обязательные»;
- корректность введенных данных;
- ограничения на количество символов в поле ввода;

Окончательная проверка происходит на сервере, и она во много повторяет проверку на клиенте, для предотвращения обхода проверки на клиенте, помимо этого на сервере проверяется информация на:

- уникальность созданной записи для определенного пользователя;
- название файла;
- размер файла (не больше 8 Мбайт).

4.1.3 Требования к временным характеристикам

После изменения данных в БД, в пользовательском интерфейсе информация должна обновиться с задержкой от 3 до 8 секунд, при хорошем интернет соединении. В случае если информация не обновляется больше 20 секунд, считать, что данные не были внесены в базу данных.

4.2 Требования к надежности

Данная автоматизированная программа организации подачи заявлений на повышенную стипендию должна быть надежной и защищенной. Доступ к конфиденциальным данным студентов должен быть ограничен и защищен паролем. Система должна иметь функционал резервного копирования данных и восстановления в случае сбоя системы. Кроме того, программа должна быть защищена от взлома и атак на сервер. Для этого должны быть установлены соответствующие меры безопасности, такие как шифрование данных и использование современных алгоритмов защиты информации.

4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы.

В связи с тем, что в базе данных хранятся личные данные пользователей. Надежное (устойчивое) функционирование программы должно быть обеспечено

Продолжение ПРИЛОЖЕНИЯ А

выполнением заказчиком совокупности организационно-технических мер и мероприятий, перечень которых приведен ниже:

- организация хранения информации в виде Raid 10;
- организацией бесперебойного питания технических средств;
- использованием лицензионного программного обеспечения;
- регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
- выполнением требований ГОСТ Р 53114-2008 Защита информации. Обеспечение информационной безопасности в организации;
- регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

4.2.2 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 10 минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать 1 день.

4.3 Условия эксплуатации

Система размещается на сервере вуза с устойчивой связью доступом в интернет, пропускной мощностью канала не менее 1000/1000 (Mbps), с описанным DNS и DHCP протоколами, а также связью с системой 1С Университет.

4.3.1 Климатические условия эксплуатации

Специальные условия не требуются.

4.3.2 Требования к видам обслуживания

Продолжение ПРИЛОЖЕНИЯ А

Не реже чем в 6 месяцев производить обновление системы и связанных с ней компонентов.

4.3.3 Требования к численности и квалификации персонала

При установке, настройке и сопровождении системы необходим системный администратор, для организации бесперебойной работы сервера и устранения возникших технических проблем во время работы системы. В процессе эксплуатации с системой работает пользователь (студент, оператор).

Системный администратор должен иметь высшее профильное образование. В перечень задач, выполняемых системным администратором, должны входить

- развертывание и обновление системы;
- настройка сетевого оборудования.

Пользователь программы должен обладать практическими навыками работы с графическим пользовательским интерфейсом, знать русский язык.

4.4 Требования к составу и параметрам технических средств

Состав технических средств:

- Сервер предприятия;
- Дисковое пространство, организованное в формате Raid 10, объемом не менее 512 гб;
- Доступ к сети интернет и пропускной мощностью канала не менее 1000/1000 (Mbps).

4.5 Требования к информационной и программной совместимости

Для корректной работы системы необходимо программное обеспечение:

- Ruby 3.0.3 и выше;
- Rails 6.0.3.5 и выше;
- Vue 2 и выше;
- Vuetify 2 и выше;
- Axios 0.27.2 и выше;
- Iodash 4.17.21 и выше.

Продолжение ПРИЛОЖЕНИЯ А

4.6 Требование к маркировке и упаковке

Программное обеспечение распространяется по средство системы контроля версий GitLab.

Специальных требований к маркировке не предъявляется

4.7 Требования к транспортированию и хранению

Специальных требований не предъявляется.

4.8 Специальные требования

Специальных требований не предъявляется.

4.9 Требования к программной документации

Предварительный состав программной документации:

- техническое задание;
- руководство пользователя.

5 Техничко-экономические показатели

Модуль разрабатывается для информационной системы ФГБОУ ВО «АмГУ» и не предназначен для извлечения прибыли. Достоинством данного программного продукта является возможность полной интеграции в уже существующую информационную сеть.

6 Стадии и этапы разработки

Разработка должна быть проведена в четыре стадии:

1. анализ предметной области;
2. техническое задание;
3. технический (и рабочий) проекты;
4. внедрение.

На стадии «Анализ предметной области» должен быть выявлены потребности заказчика и определены задачи которые решает программа, её состав и функции.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

Продолжение ПРИЛОЖЕНИЯ А

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

- разработка программы для распределения повышенной стипендии;
- разработка программной документации для сопровождения;
- прохождения ряда программных испытаний;
- прохождения испытаний в реальных условиях

На стадии «Внедрение» должен быть выполнен этап разработки «Подготовка и передача программы».

Содержание работ по этапам:

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- постановка задачи;
- определение и уточнение требований к техническим средствам;
- определение требований к программе;
- определение стадий, этапов и сроков разработки программы и документации на нее;
- согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах заказчика.

7 Порядок контроля и приемки

При приемке проводятся приемо-сдаточные испытания.

Ход проведения приемо-сдаточных испытаний заказчик и исполнитель документируют в протоколе испытаний.

Продолжение ПРИЛОЖЕНИЯ А

На основании протокола испытаний исполнитель совместно с заказчиком подписывают акт приемки-сдачи программы в эксплуатацию.

7.1 Исполнители

Руководитель разработки

Преподаватель

Исполнитель

Студент группы

Еремина В.В.

Сулейманов Д.К.

953-об