

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

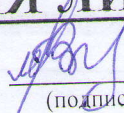
Специальность: 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»
Специализация: «Эксплуатация вычислительных машин, комплексов,
систем и сетей специального назначения»

ДИПЛОМНЫЙ ПРОЕКТ

на тему

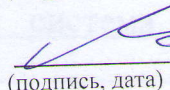
АНАЛИТИЧЕСКАЯ СИСТЕМА ПОДДЕРЖКИ
УПРАВЛЕНИЯ ЛИЧНЫМ СОСТАВОМ

Студент


(подпись, дата)

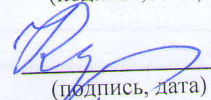
Вахрамеев Максим Александрович
(ФИО полностью)

Руководитель


(подпись, дата)

Митрохин Максим Александрович
(фамилия, инициалы)

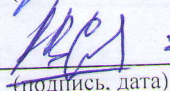
Нормоконтролёр


(подпись, дата)

22.06.19 А.В. Кучин
(фамилия, инициалы)

Рецензент

И.И. Степанов
(должность, место работы)


(подпись, дата)

22.06.19 А.В. Севастьянов
(фамилия, инициалы)

ООО «Бизнес-программы»

Работа допущена к защите (протокол зас. кафедры от 31.05.19 г. № 38-09-14)

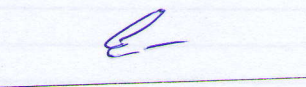
Заведующий кафедрой


(подпись)

М.А. Митрохин
(фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от
05.07.2019 № 2)

Секретарь ГЭК



Е.И. Гурин

Пенза 2019

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

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
«25» апреля 2019 г.

ЗАДАНИЕ
НА ДИПЛОМНЫЙ ПРОЕКТ

1. Студент Вахромеев Максим Александрович гр. 14В02 факультета Вычислительной Техники

2. Тема работы «Аналитическая система поддержки управления личным составом»

Тема утверждена приказом ПГУ № 554/0 от 29 "апреля" 2019 г.

3. Руководитель работы д.т.н. М.А. Митрохин

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

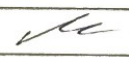
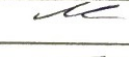
Разработать программное обеспечение для аналитики и контроля личного состава.

Исходные данные для разработки: сведения об успеваемости студентов учебных групп, личные данные студентов и преподавателей.

Выходные данные: результаты аналитической обработки данных в графическом виде.

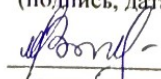
5. Перечень подлежащих разработке вопросов _____
1. Анализ предметной области. _____
 2. Обзор существующих средств аналитики. _____
 3. Постановка задачи на разработку. _____
 4. Разработка UML диаграмм и структуры базы данных. _____
 5. Разработка программного обеспечения. _____
 6. Разработка инструкции для оператора. _____
- _____
- _____
- _____

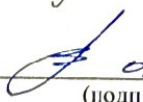
6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Исследовательская часть	60	10.05.2019	
2	Конструкторская часть	60	24.05.2019	
3	Технологический раздел	54	07.06.2019	
4	Анализ опасных и вредных факторов, воздействующих на оператора ЭВМ, и способы противодействия им	42	15.06.2019	

Дата выдачи задания " 29 " апреля 20 19 г.

Руководитель работы  _____
(подпись, дата)

Задание к исполнению принял студент  _____
(подпись, дата)

Работу к защите допустить
Декан факультета  01.08.18
(подпись, дата)

РЕФЕРАТ

Дипломный проект содержит 125 страниц, 65 рисунков, 5 приложений.
АНАЛИТИЧЕСКАЯ СИСТЕМА, ЭИОС ПГУ, ЭИОС, PHP, LARAVEL,
УПРАВЛЕНИЕ ЛИЧНЫМ СОСТАВОМ

Структура работы включает в себя введение, 3 главы, заключение, список используемых источников и приложения.

Цель работы: разработать аналитическую систему поддержки управления личным составом учебных подразделений.

Для достижения цели необходимо решить следующие задачи:

- изучить существующие решения;
- разработать UML-диаграммы;
- разработать схемы баз данных;
- разработать алгоритм функционирования модуля аналитики.

В результате проведенной работы был реализован проект, удовлетворяющий поставленной задаче: создано приложение на языке программирования PHP.

					ПГУ 1.090501.01.001 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Вахрамеев М.А.			Аналитическая система поддержки управления личным составом Пояснительная записка			
Провер.		Митрахин М.А.						
Н. Контр.		Кучин А.В.						
Утверд.		Митрахин М.А.						
						Лист	Лист	Листов
							4	125
						ФВТ гр. 14В02		

СОДЕРЖАНИЕ

Введение.....	7
1 Анализ предметной области	11
1.1 Формализация требований к системе	11
1.1.1 Постановка задачи	11
1.1.2 Обзор существующих решений.....	13
1.1.3 Информационные потребности пользователей	14
1.1.4 Требования, предъявляемые к системе.....	14
1.2 Конструкторская часть	17
1.2.1 Концепция Unified Modeling Language.....	17
1.2.2 Виды диаграмм UML.....	18
1.2.3 Диаграмма вариантов использования	18
1.2.4 Диаграмма классов.....	20
1.2.5 Диаграмма последовательности	21
1.2.6 Диаграмма деятельности	23
2 Разработка программного обеспечения	28
2.1 Выбор инструментов разработки	28
2.2 Алгоритмы работы приложения.....	31
2.3 Описание модулей	34
2.4 Проектирование базы данных системы	40
2.4.1 Создание инфологической модели базы данных.....	40
2.4.2 Создание даталогической модели базы данных	43
2.5 Тестирование	47
2.5.1 Методы тестирования.....	47
2.5.2 Unit-тесты	49

					<i>ПГУ 1.090501.01.001 ПЗ</i>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		<i>Вахрамеев М.А.</i>			<i>Аналитическая система поддержки управления личным составом Пояснительная записка</i>		
Провер.		<i>Митрохин М.А.</i>					
Н. Контр		<i>Кучин А.В.</i>					
Утверд.		<i>Митрохин М.А.</i>					
						Лит.	Лист
							5
						Листов	
						125	
						<i>ФВТ</i>	
						<i>гр. 14В02</i>	

2.5.3 Тесты в Laravel	50
2.5.4 Методика тестирования приложения.....	53
3 Анализ опасных и вредных факторов, воздействующих на оператора ЭВМ, и способы противодействия им	54
3.1 Введение.....	54
3.2 Аспекты производственной безопасности при работе на ПК	55
3.2.1 Психофизиологические факторы	55
3.2.2 Защита от излучений	57
3.2.3 Оборудование рабочих мест с ПК.....	58
3.2.4 Электробезопасность	59
3.2.5 Освещение рабочего места	60
Заключение	61
Список использованных источников	62
Приложение А. Текст программы	64
Приложение Б. Руководство оператора	99
Приложение В. Протокол тестирования.....	104
Приложение Г. Презентация	112
Приложение Д. Графическая часть	121

					<i>ПГУ 1.090501.01.001 ПЗ</i>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		<i>Вахрамеев М.А.</i>			<i>Аналитическая система поддержки управления личным составом Пояснительная записка</i>	Лист	Листов
Провер.		<i>Митрохин М.А.</i>					
						6	125
Н. Контр		<i>Кучин А.В.</i>				<i>ФВТ гр. 14В02</i>	
Утверд.		<i>Митрохин М.А.</i>					

Введение

Программное обеспечение для аналитики и контроля личного состава – сложный и всеобъемлющий программный комплекс. Затраты на разработку данных проектов подразумевают использование большого числа человеко-месяцев. По объему работ и направлениям аналитики данный раздел разработки программного обеспечения является одним из самых масштабных в области информационных технологий. Он предполагает использование знаний из разных сфер человеческой деятельности, а также из наук.

Использование программных средств для аналитики позволяет предугадать будущее состояние объекта или процесса, заранее спланировать меры предупреждения опасных событий, оценить эффективность работы. Развитие многих отраслей науки и промышленности происходили благодаря аналитическим расчетам на основе имеющихся данных.

Одной из отличительных черт автоматизированных систем аналитики, как программного обеспечения является работа с большими объемами данных. Персональные данные, успеваемость, результаты профессионально-должностной подготовки непосредственно каждого человека хранятся в специализированной базе данных, занимающей порой гигабайты места на жестких дисках. Применение систем управления базами данных в таких системах обусловлено следующими факторами:

Для хранения большого объема данных и его быстрой обработки требуется табличное представление. Именно системы управления базами данных позволяют достичь максимальной скорости и удобства хранения.

Современные системы управления базами данных спроектированы в максимально удобном виде для внедрения в программный код, что не вызывает дополнительных трудностей при разработке программного обеспечения.

Размещение ресурсов на диске осуществляется в иерархическом виде с применением операций шифрования и архивации. Также подразумевается хранение ресурсов на удаленных серверах, что вызовет дополнительные требования к хранению. Разработчикам программного обеспечения необходимо предусмотреть систему аутентификации и защиты информации в соответствии с требованиями руководящих документов. В настоящее время предполагается работа с разными версиями продукта и возможностью комфортного ведения параллельного процесса разработки. Для этого применяются системы контроля версий.

Для обработки большого объема данных в современном программном обеспечении используется обширный математический аппарат, позволяющий максимально быстро считать значения функций, определять зависимости и решать многие другие возникающие задачи, требующие математического подхода. Многие языки программирования имеют встроенные или созданные сообществом библиотеки для работы с математическими функциями.

Для отображения данных применяются специальные возможности графических библиотек языков программирования. Они позволяют строить графики, выводить изображения и видео-информацию на экран. Данные средства способствуют более наглядному представлению информации пользователю.

В конце прошлого и начале этого использование автоматизированных систем было очень дорогостоящим, поэтому лишь самые богатые компании и предприятия могли позволить себе установку данных систем. Для организаций часто встает вопрос покупки готового программного решения, использования минимального набора имеющихся офисных средств, либо разработка уникальной системы управления на заказ. При приобретении готового программного продукта требуется настройка программы под конкретные требования пользователя, дальнейшее техническое сопровождение, обучение персонала. Все это выражается в больших финансовых и временных затратах.

					ПГУ 1.090501.01.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

Набор офисных средств не всегда в полной мере может предоставить компании все ее требования. При разработке персональной системы можно учесть все требования компании и максимально улучшить автоматизацию действий.

Разрабатываемая аналитическая система контроля личного состава представляет собой замкнутую автоматизированную систему управления, так как в ней осуществлена обратная связь между выходом объекта управления и входом управляющего устройства. Таким образом создается замкнутый контур, обеспечивающий автоматический контроль над состоянием объекта управления. Основная задача системы – сбор данных о личном составе, поступающих с различных удаленных источников, дальнейшая аналитическая обработка и вывод результатов вычисления на экран оператора. Оператор имеет возможность не только наблюдать за поступающей информацией, но и подавать информацию об успехах каждого подчиненного, а также о критических ситуациях.

Данная аналитическая система взаимодействует с данными, предоставляемыми из электронной информационно-образовательной среды Пензенского государственного университета, которая обеспечивает доступ студентов к учебным планам, рабочим программам дисциплин, программам практик и итоговой аттестации. В этой среде фиксируется ход учебного процесса, заносятся данные о расписании, личном составе учебных групп, посещаемость занятий, результаты выполнения учебных заданий, текущий контроль. ЭИОС позволяет формировать электронное портфолио учащегося, в том числе сохранение курсовых работ, научных работ, рецензий и оценок со стороны участников образовательного процесса.

Данные, поступающие из электронной информационно-образовательной среды обрабатываются специально разработанным программным обеспечением, позволяющим производить аналитические операции и выводить их результаты на экран оператора. Это позволяет более качественно управлять личным составом учебной группы, принимать правильные управленческие решения.

В настоящее время все данные находятся в разных частях электронной информационно-образовательной среды, что затрудняет работу с ними и построение зависимостей между разными показателями. Постоянно растущее число данных о каждом учащемся, значительные временные затраты на поиск этих данных, а иногда и их недоступность вынуждают создать специализированную аналитическую систему, облегчающую поиск и увеличивающую эффективность управления личным составом. Она позволит в реальном времени отслеживать изменения в учебной деятельности личного состава, быстро реагировать на них и правильно планировать мероприятия по улучшению успеваемости.

Данный проект посвящен разработке аналитической системы контроля личного состава с модулем загрузки исходных данных, модулем аналитической обработки данных и модулем графического отображения результатов аналитических операций, которые автоматизируют работы по контролю.

1 Анализ предметной области

1.1 Формализация требований к системе

В специальном разделе пояснительной записки к дипломному проекту описаны основные этапы разработки программного продукта: постановка задачи, предварительные НИР и техническое задание. На этапе эскизного проектирования были определены требования к входным и выходным данным, составлен общий алгоритм работы программы. При техническом проектировании была определена архитектура аналитической системы управления личным составом, детализированы функциональные требования, разработаны основные алгоритмы работы, представлено описание конфигурации технических средств.

1.1.1 Постановка задачи

Управление личным составом учебной группы связано с оперированием большим объемом данных непосредственно о каждом учащемся. Сбор этих данных был изначально осуществлен при помощи средств электронной информационно-образовательной среды, но все эти данные представлены в множестве разделов. Этот факт затрудняет работу с данными, проведение сравнения показателей учебного процесса и аналитики. Человеку, на которого возложены обязанности по управлению учебной группой необходимо иметь полный доступ к ним, во время проводить анализ и принимать управленческие решения.

Со временем объем данных, поступающих из источника – электронной информационно-образовательной среды растет – это затрудняет поиск информации, что и так занимает не малое количество времени. Для доступа к данным об обучающихся необходимо иметь централизованное хранилище и возможность ведения аналитических операций.

Труд сотрудника, осуществляющего управление личным составом можно разделить на две категории – обеспечивающую и производительную [1]. Соотношение этих видов деятельности разное, но объем обеспечивающей деятельности остается не малой. Операции с электронными и бумажными документами относятся к обеспечивающей деятельности. Если уменьшить время на выполнение этих операций, то освободиться значительное количество времени для выполнения производительного труда.

Указанные выше факты определяют необходимость использования аналитической системы управления учебным процессом личным составом, имеющей централизованное хранилище данных, модуль аналитики и модуль графического отображения аналитических операций.

На примере организации управления учебным процессом личного состава учебных групп Военного-учебного центра при Пензенском государственном университете предлагается рассмотреть использование данной системы. Для оперативного доступа к информации об учебном процессе возникла необходимость создания централизованной системы аналитики. Кроме того, крайне важно иметь модуль графического отображения результатов аналитических операций для возможности принятия быстрых и правильных управленческих решений.

Именно эти проблемы должна решать аналитическая система управления учебным процессом личного состава.

1.1.2 Обзор существующих решений

Из условий поставленной задачи следует, что данная система является самостоятельным приложением. Вследствие этого в предварительных НИР были изучены имеющиеся примеры программного обеспечения аналогичного назначения. Специфика разрабатываемой системы, как правило, подразумевает разработку подобных приложений специализированными компаниями, нацеленными на конкретного потребителя услуг. Подобные системы были разработаны для нужд ВУЗов, например система Moodle – система управления курсами, представляющая собой свободное программное обеспечение для создания сайтов онлайн-обучения. Первая версия системы была представлена 20 августа 2002 года, последняя версия системы 3.6.1 выпущена 5 декабря 2018 года [2]. Подобная система для управления личным составом SIDPERS была разработана для нужд сухопутных войск армии США. Она предназначена для сбора, обработки, хранения и представления командованию информации по личному составу сухопутных войск как в мирное, так и в военное время. Данные собираются от источников тактического звена управления (рот и выше). Разработка системы была начата в 1970 году [3].

Создание системы управления учебным процессом – одна из наиболее важных задач, стоящих перед учебными заведениями. Разработка такой системы связана с непрерывно возрастающей потребностью в использовании потоков информации в учебном процессе. В связи с этим создаются различные образовательные электронные модули, производится подготовка профессорско-преподавательского состава. Данные системы позволяют отслеживать и решать несколько задач: полный учет выполнения учебных планов, учет посещаемости, формировать расписание, анализировать результаты выполнения лабораторных работ, курсовых работ.

При внедрении таких систем происходит решение трудностей и получение значимых результатов: обеспечение потребности в информации всех субъектов управления, целенаправленная организация потоков информации в учебном процессе, повышение эффективности анализа данных и принятия управленческих решений, значительное сокращение затрат рабочего времени на работу с документами. Успешный опыт функционирования такой системы необходимо рассматривать в контексте качества подготовки специалистов, формирования у выпускников профессиональных компетенций [4].

1.1.3 Информационные потребности пользователей

В результате анализа указанных выше проблем были выявлены следующие основные требования, предъявляемые пользователями к системе:

- наличие исходных данных из имеющихся источников;
- запись исходных данных в базу данных разрабатываемого программного обеспечения;
- просмотр исходных данных в табличном виде;
- аналитическая обработка данных в соответствии с заданными параметрами;
- графическое представление выполненных аналитических операций;
- дружественный пользователю интерфейс.

1.1.4 Требования, предъявляемые к системе

Исходя из поставленной задачи и проведенных предварительных научно-исследовательских работ, были сформулированы требования к разрабатываемому программному продукту.

Состав выполняемых функций

Разрабатываемое программное обеспечение должно быть разделено на модули. Каждый модуль должен выполнять набор специализированных функций.

Модуль работы с базой данных должен:

- 1) Выполнять загрузку исходных данных из системы управления базой данных.
- 2) Отображать полученные данные пользователю в табличном виде с разбиением на разделы.
- 3) Обеспечивать систематизацию и группировку данных для дальнейшей обработки.

Модуль аналитики должен:

- 1) Предоставлять пользователю выбор методов аналитики и сравнения в виде разработанного интерфейса аналитики.
- 2) Выбирать и группировать данные в соответствии с требованиями пользователя.
- 3) Производить математические вычислительные операции с данными.
- 4) Подготавливать данные для вывода на экран пользователя в заданном формате (в виде объекта, либо в json).

Модуль графического отображения результатов должен:

- 1) Получать рассчитанные аналитические данные.
- 2) Предоставлять пользователю вид отображения результатов (в виде текста, таблицы или графика).
- 3) Выводить данные на экран пользователя.
- 4) Отображать критические и положительные значения.

Требования к надежности

Для обеспечения стабильной и надежной работы аналитической системы необходимо:

- применять стандартные и надежные средства разработки и библиотеки;
- анализ корректности действий пользователя перед совершением операций;
- использование стабильной архитектуры сервера системы управления базой данных.

Требования к информационной и программной совместимости

В настоящее время программисту предоставляется большой выбор средств разработки и проектирования. В каждом конкретном случае происходит выбор наиболее удобного и подходящего средства.

Реализация модуля работы с базой данных выполнена при помощи средств СУБД MySQL, языка программирования PHP и фреймворка Laravel.

Реализация модуля аналитики выполнена средствами языка программирования PHP и его математических библиотек.

Реализация модуля графического отображения результатов выполнена средствами HTML и javascript с использованием библиотеки jQuery.

Для продления «жизни» программного обеспечения была обеспечена расширяемость – возможность подключения дополнительных плагинов, а также разработан API (программный интерфейс приложения), позволяющий взаимодействовать с другими приложениями.

На пользовательской ЭВМ должна быть установлена операционная система Microsoft Windows XP и старше, Linux с графической оболочкой или Mac OS. На компьютере должен быть установлен веб-браузер, поддерживающий HTML 5, CSS и JavaScript.

					ПГУ 1.090501.01.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		16

На серверной машине должна быть установлена операционная система Linux CentOS 7 или Linux Ubuntu 16.04 и старше. Для работоспособности аналитической системы должны быть установлены пакеты MySQL (mariadb), PHP версии 5.6 или выше, Apache 2 или nginx, Laravel.

1.2 Конструкторская часть

1.2.1 Концепция Unified Modeling Language

Unified Modeling Language (UML) – это графический язык моделирования и визуализации для объектного моделирования, описания параметров, конструирования и документирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур. Данный язык используется для визуального моделирования, спецификации и документирования элементов системы [5].

Основные принципы языка UML:

- создаваемая модель влияет на способы решения задачи;
- каждая модель может быть выражена на разных уровнях точности;
- ни одна модель не является достаточной.

На основе UML-технологии строится единая информационная модель. Нотация UML как язык визуального дизайна включает в себя большое количество компонентов графической нотации. Основными понятиями UML являются «класс», «атрибут», «операция» и «отношение». Основным концептуальным представлением является диаграмма.

Диаграмма в UML представляет собой графическое представление набора системных элементов, чаще всего изображаемых в виде связанного графа с вершинами (сущностями) и ребрами (связями). Диаграммы создаются для визуализации системы с разных точек зрения [6].

1.2.2 Виды диаграмм UML

UML диаграммы являются основным способом представления моделей.

Стандартом UML определяются следующие виды диаграмм:

- четыре типа представляют статическую структуру приложения;
- пять типов диаграмм представляют поведенческие аспекты системы;
- три представляют физические аспекты функционирования системы.

Далее будут представлены и описаны следующие диаграммы: диаграмма вариантов использования, диаграмма классов, диаграмма последовательности, диаграмма деятельности.

1.2.3 Диаграмма вариантов использования

Проектируемая система представлена в виде множества сущностей (актеров), которые взаимодействуют с системой с помощью прецедентов. Актером называется лицо, взаимодействующее с системой извне. Каждый вариант использования определяет за собой готовый набор действий, совершаемых актером при взаимодействии с системой. В данной диаграмме не указывается способ и методы взаимодействия актеров с системой. На рисунке 1 представлена диаграмма вариантов использования.

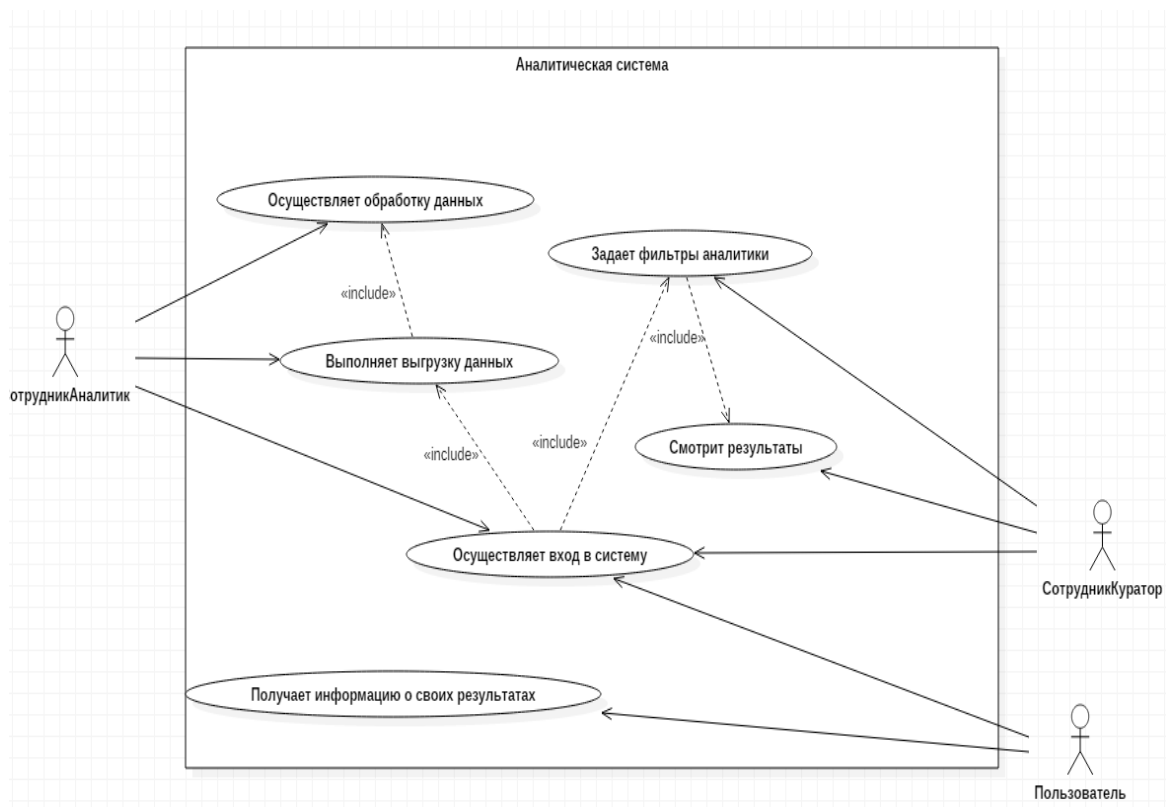


Рисунок 1 – Диаграмма вариантов использования

Актер «СотрудникКуратор» - специально обученный человек, который задает необходимые критерии аналитики и получающий результаты аналитики на экран. Он, как и все пользователи, проходит авторизацию в системе и имеет доступ к аналитическим данным. Он имеет возможность задавать специальные фильтры выборки.

Актер «СотрудникАналитик» - специально обученный человек, который осуществляет операции первичной выгрузки данных из СУБД. Помимо этого он имеет возможность осуществить первоначальную обработку данных.

Актер «СотрудникКуратор» - куратор учебной группы, который смотрит результаты аналитики, задает фильтры. На основании полученных данных он может принимать более точные управленческие решения.

1.2.4 Диаграмма классов

Диаграмма классов предназначена для статической структуры модели системы в терминологии классов ООП. На ней отражаются связи между сущностями системы, а также описывается внутренняя структура объектов.

На этой диаграмме не учитываются данные о временных аспектах функционирования аналитической системы. Диаграмма разрабатывается с учетом принципов объектно-ориентированного программирования и паттернов проектирования.

Между классами могут быть следующие отношения:

- ассоциация – произвольная семантическая связь классов, может иметь кратность (аналогично *ER*-диаграммам);
- обобщение – отношение между предком и потомком, интерпретация наследования;
- агрегация – включение одной сущностью других как составных частей, взаимосвязь между частью и целым;
- композиция – более сильный вариант агрегации, при котором часть не может существовать отдельно от целого;
- зависимость – семантическая связь, не являющаяся какой-либо из вышеперечисленных.

На рисунке 2 представлена диаграмма классов.

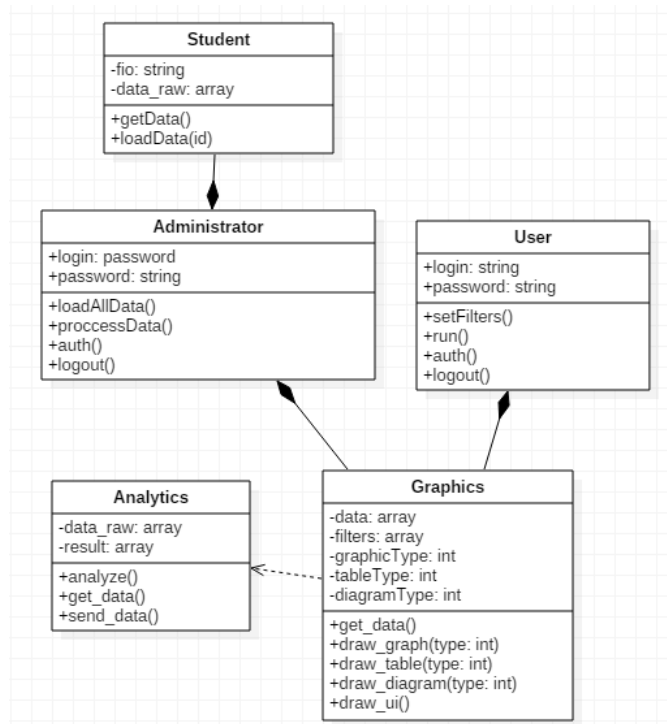


Рисунок 2 – Диаграмма классов

1.2.5 Диаграмма последовательности

Для моделирования взаимодействия объектов в языке UML используются диаграммы взаимодействия. Взаимодействия объектов можно рассматривать во времени. Взаимодействующие объекты обмениваются между собой информацией. Информация принимает форму законченных предложений.

Все действующие лица располагают в верхней части диаграммы и изображаются в виде прямоугольника, с отходящей от него в низ пунктирной линией. Эта линия называется линией жизни объекта, представляет собой фрагмент жизненного цикла объекта. Стрелки между линиями двух жизней объекта – это сообщения, передаваемые между действующим лицом и объектом для выполнения требуемых функций. Сообщения располагаются по порядку сверху вниз и помечаются именем сообщения [7].

На рисунках 3-5 представлены диаграммы последовательности.

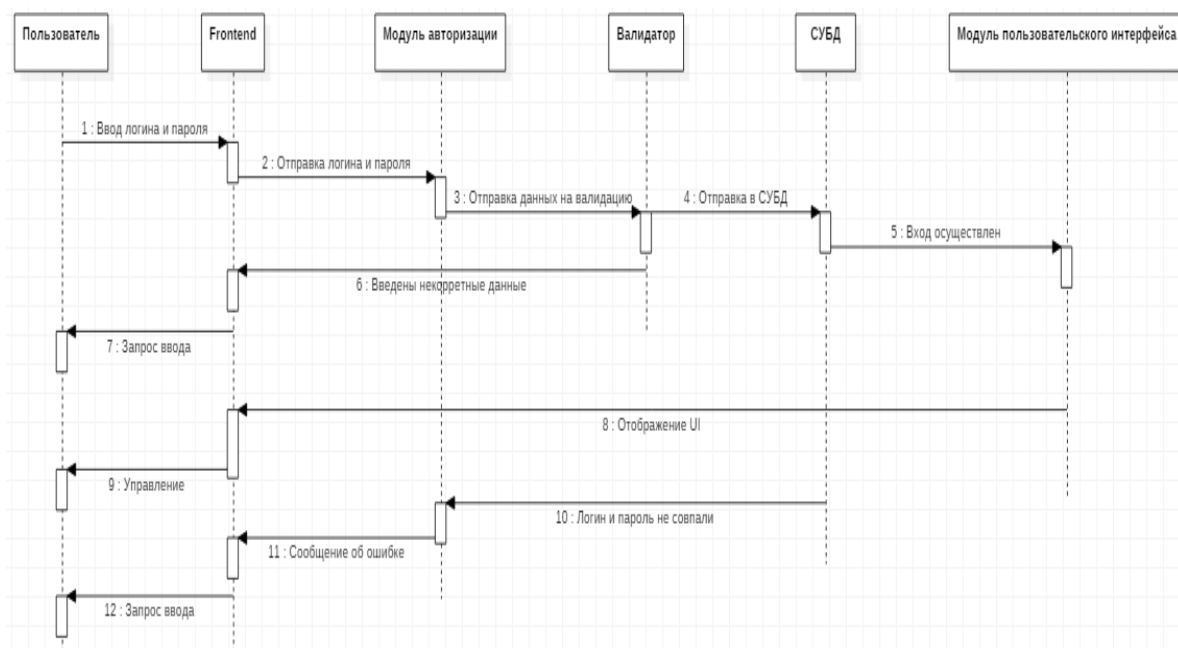


Рисунок 3 – Диаграмма последовательности работы пользователя

На рисунке 3 изображена диаграмма последовательности работы пользователя. Любой пользователь системы проходит авторизацию путем обращения к СУБД, в которой хранятся данные для аутентификации пользователей. Система сверяет введенные пользователем данные с данными из базы данных. В случае совпадения данных осуществляется вход в систему, иначе выводится сообщение об ошибке. Далее пользователь может обращаться к менеджеру запросов и выполнять действия в системе.

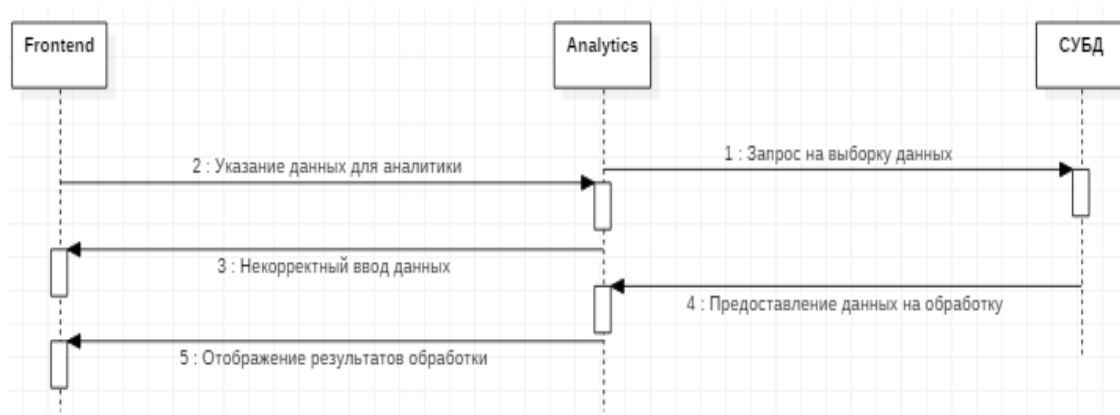


Рисунок 4 – Диаграмма последовательности работы модуля аналитики

На рисунке 4 представлена диаграмма последовательности работы модуля аналитики. Сервер обращается к СУБД при помощи менеджера подключений и получает необходимые данные. После обработки информации осуществляется ее дальнейшая передача. Менеджер событий отображает все совершаемые действия внутри модуля.

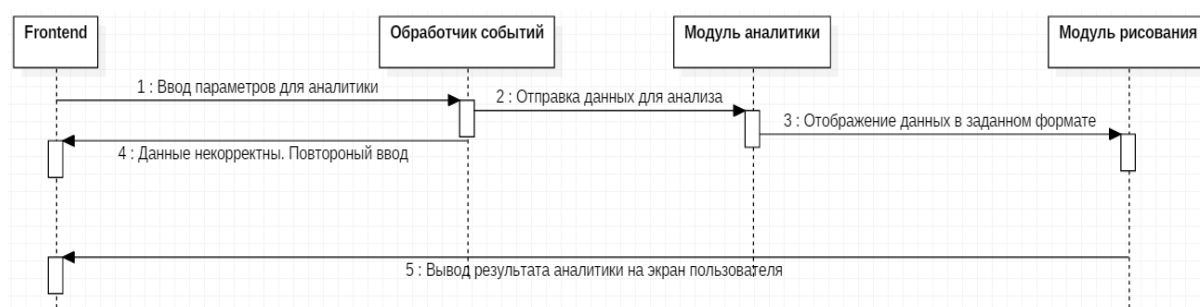


Рисунок 5 – Диаграмма последовательности работы модуля графики

На рисунке 5 представлена диаграмма последовательности работы модуля графики. После ввода параметров аналитики введенные данные отправляются в обработчик событий, который проверяет корректность введенных данных. В случае некорректного ввода, будет выведено сообщение об ошибке и предложен повторный ввод данных. Если обработчик событий не обнаружил ошибок, то данные отправляются на обработку модулю аналитики, а затем в модуле рисования они подготавливаются к выводу и отображаются на экране пользователя.

1.2.6 Диаграмма деятельности

Диаграмма деятельности представляет собой низкоуровневую UML-диаграмму.

Диаграммы деятельности состояются из ограниченного количества фигур, соединённых стрелками.

Основные фигуры и их назначение:

прямоугольники с закруглениями — действия;

- ромбы, прямоугольники — решения;
- широкие полосы — начало (разветвление) и окончание (схождение) ветвления действий;
- чёрный круг — начало процесса (начальный узел);
- чёрный круг с обводкой — окончание процесса (финальный узел).

Стрелки изображают от начала к концу действия и показывают потоки управления или потоки объектов (данных) [8].

На рисунках 6-8 представлены диаграммы деятельности.



Рисунок 6 – Диаграмма деятельности процесса авторизации

На рисунке 6 представлена диаграмма деятельности процесса авторизации. При входе в систему пользователь вводит логин и пароль. Если в системе есть пользователь с такими данными, то он успешно проходит авторизацию, иначе система возвращает пользователя к повторному вводу. При вводе невалидных символов система запрашивает повторный ввод.



Рисунок 7 – Диаграмма деятельности процесса запроса информации

На рисунке 7 представлена диаграмма деятельности процесса запроса информации. Пользователь «Администратор» получает исходные данные для обработки. Пользователь «Сотрудник» получает обработанные данные в соответствии с выбранными фильтрами.

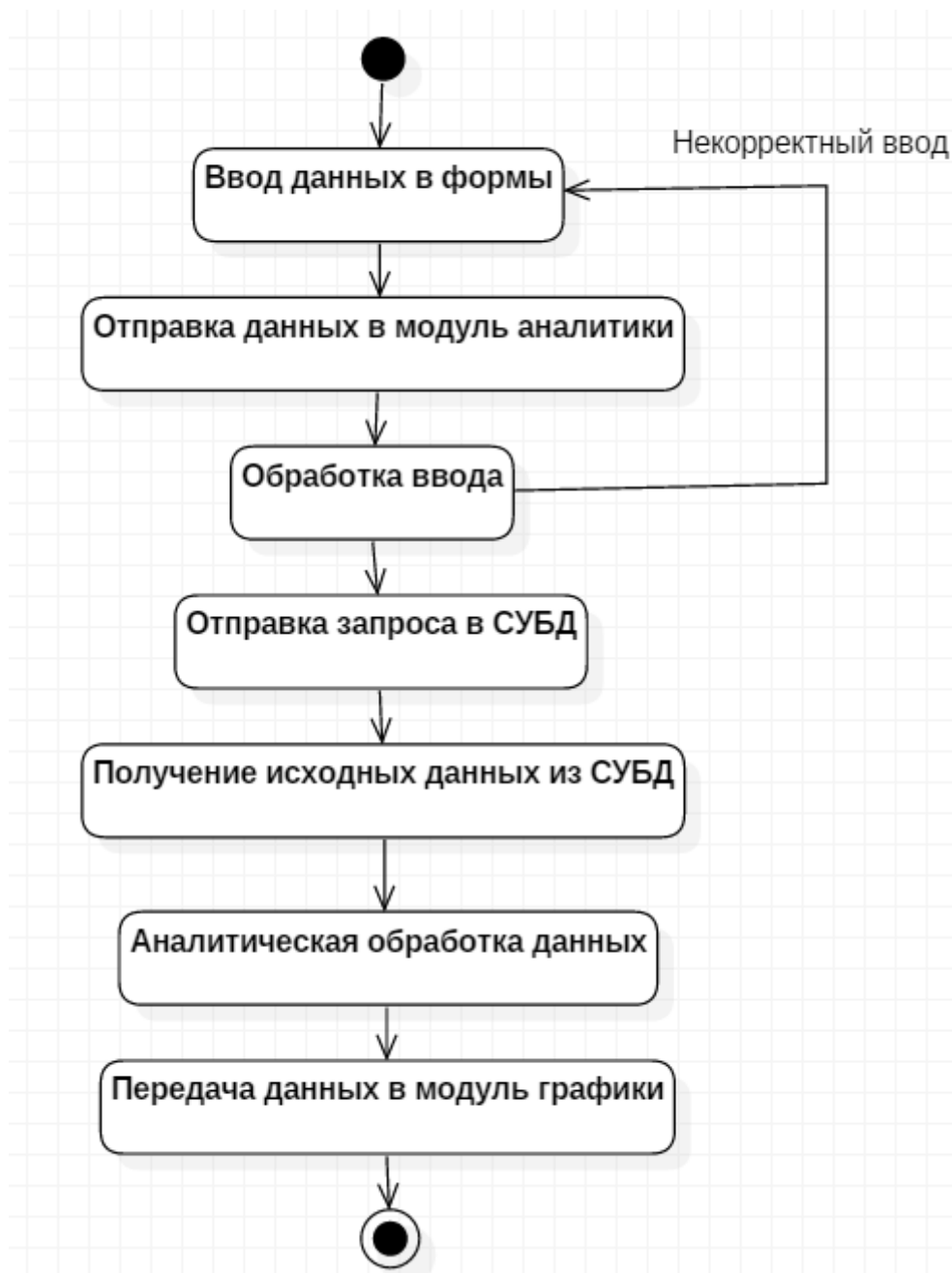


Рисунок 8 – Диаграмма деятельности процесса обработки данных

На рисунке 8 представлена диаграмма деятельности процесса обработки данных. Сервер обращается к СУБД с запросом на получении данных. При успешном выполнении запроса, данные подготавливаются к дальнейшему отображению для вывода на экран пользователя.

2 Разработка программного обеспечения

2.1 Выбор инструментов разработки

На основании поставленной задачи следует, что по исходным данным о студентах и информации о них необходимо создать базу данных, а также веб-интерфейс для работы с ней.

Для разработки программисту представлен широкий ряд средств разработки и проектирования. При выборе средств предпочтение, как правило, отдается тем, которые наиболее подходят для решения задачи.

При выборе средств разработки было проведено сравнение популярных средств разработки. Выбор уделялся непосредственно языку PHP и его фреймворкам, а также возможностям работы с СУБД (таблица 1).

Таблица 1 – Выбор средства и платформы разработки

Характеристика	Средства и платформы разработки			
	PHP 5.6	PHP 7.0	PHP 7.0 и Yii1	PHP 7.0 и Laravel
Прицип обработки кода	Интерпретатор	Интерпретатор	Интерпретатор	Интерпретатор
Язык	PHP	PHP	PHP	PHP
Средства доступа к базам данных	SQL, noSQL	SQL, noSQL	SQL, noSQL	SQL, noSQL
Платный	Нет	Нет	Нет	Нет

Наиболее удобным из всех был выбран PHP 7.0 с фреймворком Laravel. Данная связка позволяет воспользоваться новой версией языка программирования, а также современным фреймворком, который часто обновляется и имеет хороший показатель безопасности.

Для реализации серверной части приложения была выбрана операционная система Linux Ubuntu Server 16.04 с установленными пакетами Apache, MySQL, PHP 7.0, Composer и Laravel.

СУБД MySQL является наиболее распространенной и удобной для разработки веб-проектов. В открытом доступе представлено большое количество учебных материалов по освоению данной СУБД. При совместной работе с PHP она оставляет необходимый уровень быстродействия и доступа к хранимым данным.

MySQL – это многопоточная система управления базами данных. Данная СУБД характеризуется высокой скоростью работы, устойчивостью к атакам, легкостью в освоении и использовании. На данный момент MySQL находится в стадии разработки. Актуальная версия 8.0.15 от 1 февраля 2019 г. Исходный код предоставлен на github.

Широкое применение MySQL получила в реализации малых и средних приложений. Она устанавливается на Windows-системы, так и на Unix-подобные системы.

MySQL поддерживает язык запросов SQL в стандарте ANSI SQL 99. В этой СУБД есть возможности поддержки неограниченного числа подключенных к работе пользователей, количества строк в таблицах до 50 миллионов, вложенных запросов, обеспечения целостности данных [9].

Широкой популярностью пользуется клон MySQL – mariadb, который был выбран при разработке данной системы. MariaDB поддерживает полностью совместимость с командами и API MySQL. MariaDB развивается более динамично, по сравнению с MySQL, и имеет больше возможностей при работе с памятью. Добавлено улучшение производительности InnoDB и движка обработки запросов, поддержка GIS, ликвидация таблиц, виртуальные и динамические колонки.

PHP – язык программирования, который используется WEB-сервером для генерации HTML кода. Данный язык разрабатывается для создания в первую очередь веб-приложений. Этот язык платформонезависимый, его установка возможна практически на любую операционную систему.

PHP версии 7 получил обновления, которые значительно увеличили производительность языка. В этой версии также была добавлена возможность группировки объявлений импорта, решена проблема Null-коалесцентного оператора, добавлены тесты PHPUnit, возможность обработки исключений в блоке try-catch, добавлена поддержка Unicode.

Laravel – это бесплатный фреймворк для PHP с открытым исходным кодом [10]. Он использует распространенную архитектурную модель MVC – Model View Controller – модель-представление-контроллер.

Этот веб-фреймворк позволяет упростить решение основных задач веб-разработки, например, таких как аутентификация, маршрутизация, сессии, кэширование. Вопросы безопасности также предусмотрены на минимально-необходимом уровне.

Пакеты в Laravel дают разработчику возможность модули в формате Composer сразу же к приложению в виде, готовом к использованию.

Eloquent ORM – это реализация шаблона проектирования ActiveRecord. Данный функционал дает разработчику возможность определять отношения между объектами базы данных.

Маршрутизация – Laravel сам автоматически генерирует с помощью именованных маршрутов конечные URL.

REST-контроллеры дают возможность быстрого написания API, без отвлечения на реализацию базовых методов.

Миграции – система управления версиями для баз данных. Изменения в коде приложения сами вносят изменения в структуру БД. Таким образом, процесс развертывания и обновления значительно ускоряется.

2.2 Алгоритмы работы приложения

На основе представленных выше моделей и диаграмм были разработаны алгоритмы работы приложения. Далее на рисунках 9-11 представлены некоторые основные алгоритмы работы приложения.



Рисунок 9 – Общий алгоритм работы приложения

При открытии в веб-браузере приложения пользователю необходимо пройти авторизацию при помощи ввода логина и пароля. При успешной авторизации происходит перенаправление пользователя на страницу его личного кабинета. Если данные введены некорректно или неверно, то пользователю повторно предлагается ввод данных. При помощи пользовательского интерфейса выбирается необходимый режим работы приложения. Далее в соответствии с этим выбором происходит обработка поступившего задания для приложения и его дальнейшее отображение на экране пользователя.

Алгоритм авторизации представлен на рисунке 10.

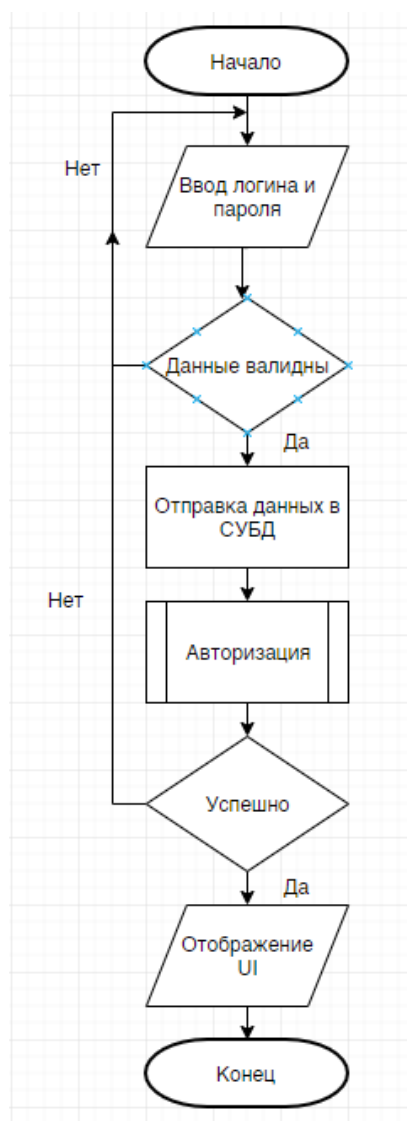


Рисунок 10 – Алгоритм авторизации

Пользователь вводит данные для входа в форму авторизации. Далее осуществляется валидация введенных данных. Если были введены некорректные данные, пользователю предлагается повторно осуществить ввод. Если данные введены корректно, то они отправляются в СУБД, где происходит сверка логина и пароля. Если логин соответствует паролю, то пользователь переходит на страницу с пользовательским интерфейсом. Иначе будет предложен повторный ввод данных для входа на сайт.

На рисунке 11 представлен алгоритм работы модуля аналитики.

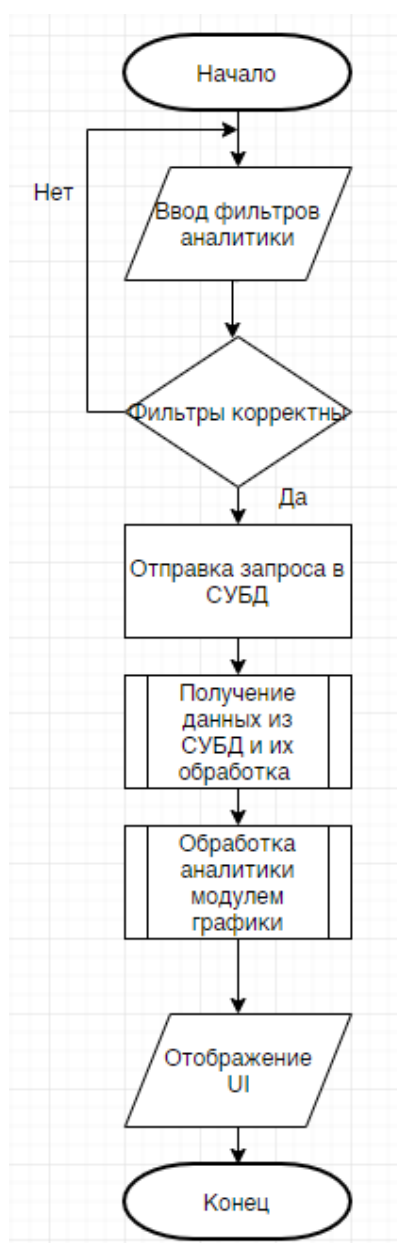


Рисунок 11 – Алгоритм работы модуля аналитики

Доступ к модулю аналитики имеют только сотрудники-аналитики. Пользователь выбирает предложенные системой методы фильтрации данных о студентах, а также способ отображения результата (таблица, график, диаграмма). В случае если фильтры заданы некорректно, пользователю будет предложен повторный ввод фильтров. Полученные фильтры используются в запросе к СУБД, при помощи которого отбираются исходные данные. Данные из СУБД далее поступают в функции обработки данных. После обработки осуществляется предварительная графическая обработка результатов. Далее графическая составляющая выводится на экран пользователя.

2.3 Описание модулей

На основе представленных выше моделей и алгоритмов были разработаны модули приложения.

Модуль авторизации.

Данный модуль позволяет пользователю авторизоваться в системе. Процесс авторизации осуществляется в несколько этапов:

1. Ввод логина и пароля на форме входа (рисунок 12).
2. Нажатие кнопки вход.

3. Валидация введенных данных. Валидация представляет собой набор функций осуществляющих проверку введенных символов на наличие SQL-инъекций, XSS-атак, проверки длины текста. В том случае, если некорректно введены данные, пользователь получает уведомление, которое отображается на экране (рисунок 13).

4. Отправка данных в СУБД. В системе управления базами данных происходит поиск по таблице users, где определяется совпадение полей login и password. Если пользователь с таким паролем найден, то авторизация пройдена, в переменную \$_SESSION записывается информации о сессии пользователя. В случае, если данные введены неправильно, выводится сообщение о ошибке и предлагается повторный ввод логина и пароля (рисунок 14).

5. Переход на страницу с пользовательским интерфейсом (рисунок 15).

The image shows a web form for logging in. The title 'Login' is at the top. There are two main input fields: 'E-Mail Address' containing 'admin1@gmail.com' and 'Password' which is masked with dots. Below the password field is a checkbox labeled 'Remember Me'. At the bottom of the form is a blue button labeled 'Login'.

Рисунок 12 – Форма входа

The screenshot shows a login form titled "Login". It has two input fields: "E-Mail Address" and "Password". The "E-Mail Address" field contains the text "admin". A red border highlights this field, and a tooltip message appears below it. The message, in Russian, states: "Адрес электронной почты должен содержать символ '@'. В адресе 'admin' отсутствует символ '@'." (The email address must contain the symbol '@'. The address 'admin' does not contain the symbol '@'). The "Password" field is filled with ten dots. Below the password field is a checkbox labeled "Remember Me". At the bottom is a blue "Login" button.

Рисунок 13 – Сообщение об ошибке при вводе

The screenshot shows the same login form titled "Login". The "E-Mail Address" field now contains "admin1@gmail.com". Below this field, a red error message reads: "These credentials do not match our records." The "Password" field is empty. The "Remember Me" checkbox and the "Login" button are still present.

Рисунок 14 – Данные введены неправильно

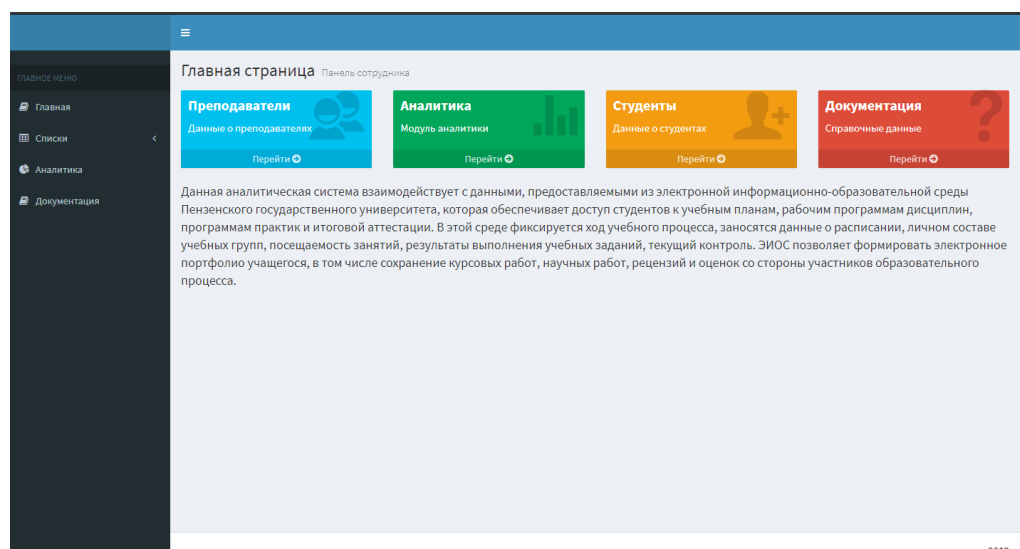


Рисунок 15 – Интерфейс приложения

Модуль аналитики.

Данный модуль осуществляет набор математических операций по работе с данными. Процесс работы модуля состоит из нескольких этапов:

1. В интерфейсе пользователя сотрудник-аналитик задает фильтры отбора данных (рисунок 16).
2. Проверка заданных фильтров. Если фильтры некорректны, то пользователю будет предложен их повторный выбор (рисунок 17).
3. Формирование запроса к СУБД. На основе выбранных фильтров создается набор условий и группировок для SQL запроса. Далее СУБД MySQL совершает выборку данных и отправляет их в модуль аналитики.
4. Полученные данные переводятся в формат JSON для возможности дальнейшей работы с ними в модуле графики.

Модуль графики.

Модуль графики включает в себя функции графического отображения информации на экран. Этапы работы графического модуля:

1. Получение данных из JSON. Перенос их в массив JavaScript.

2. Обработка значений, полученных от модуля аналитики.
3. Заполнение данных с учетом способа отображения.
4. Вывод изображения на экран пользователя (рисунок 18).

☐ Профиль обучения

Не учитывать

☐ Направление подготовки

Не учитывать

☐ Регион

Не учитывать

☐ Район

Не учитывать

☐ Город

Не учитывать

☐ Год обучения

Не учитывать

☐ Семестр

Не учитывать

☐ Оценка

Не учитывать

☒ Дисциплина

Не учитывать

Применить

Рисунок 16 – Фильтры отбора данных

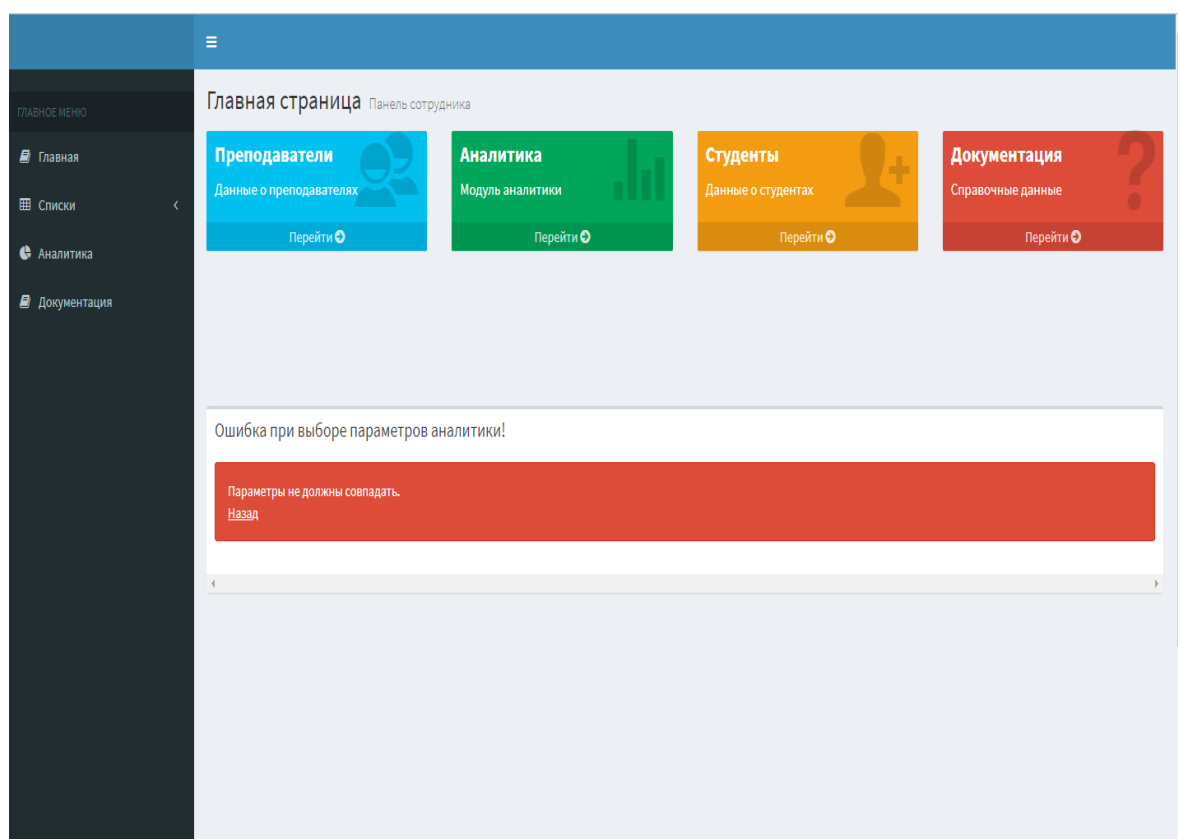


Рисунок 17 – Ошибка при выборе фильтров

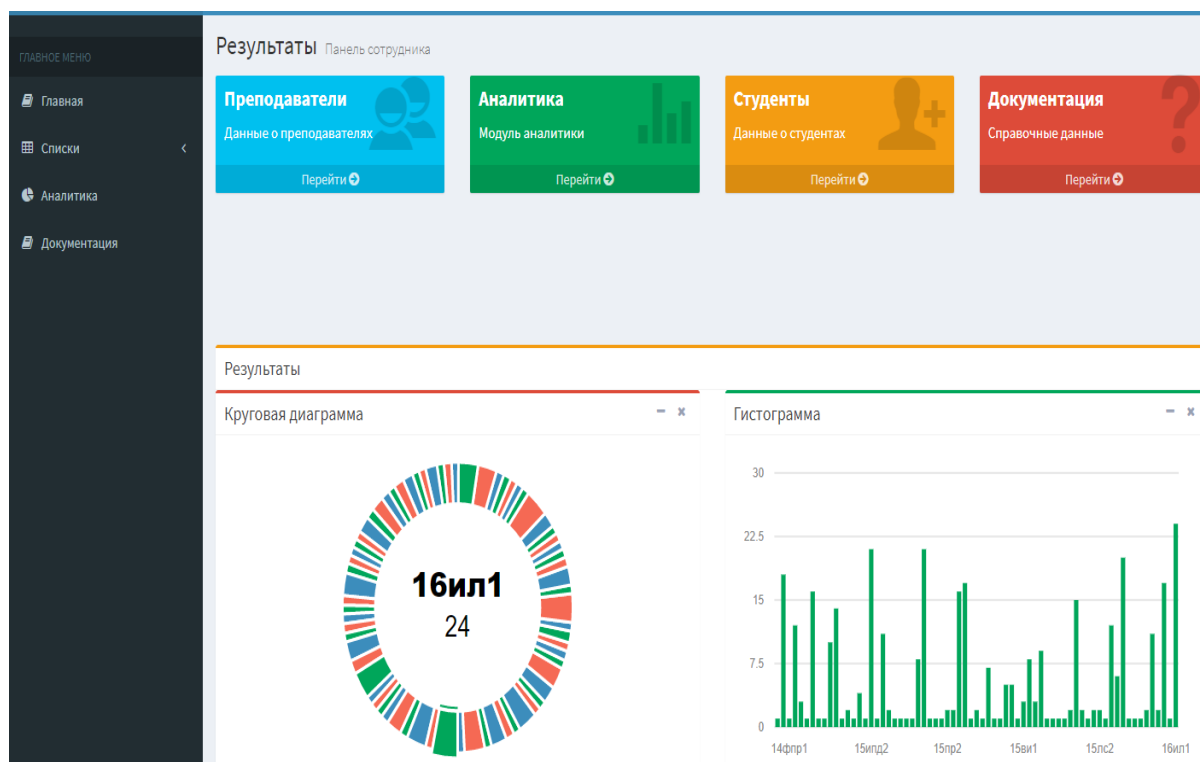


Рисунок 18 – Вывод результатов аналитики на экран

2.4 Проектирование базы данных системы

В базе данных аналитической системы будет храниться информация о учебной деятельности студентов. На основе анализа предметной области было определено, какую информацию необходимо поместить в базу данных.

На этапе проектирования БД будет определен состав и структура данных предметной области, которые должны быть в базе данных. Структура данных предметной области будет отражена с помощью информационно-логической модели. На основе данной модели реализация реляционной базы данных будет осуществляться значительно легче.

2.4.1 Создание инфологической модели базы данных

Информационно-логическая модель – это описание информационных аспектов предметной области и связей. Она не зависит от программного обеспечения СУБД и других выбранных средств разработки [11].

Наиболее распространенной методикой разработки моделей является ERD (Entity-Relationship Diagrams) – ER-модели. Такие диаграммы называются «Объект-отношение», либо «Сущность-связь». ER-модели удобнее всего разрабатывать в CASE-средствах (Computer Aided Software Engineering) [12].

Основными элементами ER-диаграммы являются:

- сущность – объект, имеющий существенное значение для предметной области. Изображается в виде прямоугольника с наименованием;
- связь – ассоциация между сущностями. На диаграмме обычно отображается в виде линии и надписанным над ней именем – глаголом, обозначающим действие;
- атрибут – характеристика или свойство сущности. Позволяет идентифицировать количественную или качественную характеристику сущности. Изображается в пределах прямоугольника-сущности.

					ПГУ 1.090501.01.001 ПЗ	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

Таким образом, были спроектированы следующие сущности (рисунок 19-23).

Пользователь
Логин Пароль Статус

Рисунок 19 – Сущность «Пользователь»

Студент
Фамилия Имя Отчество Дата рождения Пол Группа Факультет Профиль Направление Регион Район Город Улица Дом Квартира

Рисунок 20 – Сущность «Студент»

Дисциплина
Название

Рисунок 21 – Сущность «Дисциплина»

Оценка
Значение
Семестр
Год обучения

Рисунок 22 – Сущность «Оценка»

Преподаватель
Фамилия
Имя
Отчество
Дата рождения
Пол

Рисунок 23 – Сущность «Преподаватель»

Разработанная диаграмма является концептуальной, т.е. она не учитывает особенностей СУБД. На рисунке 24 представлена реляционная модель проектируемой базы данных.

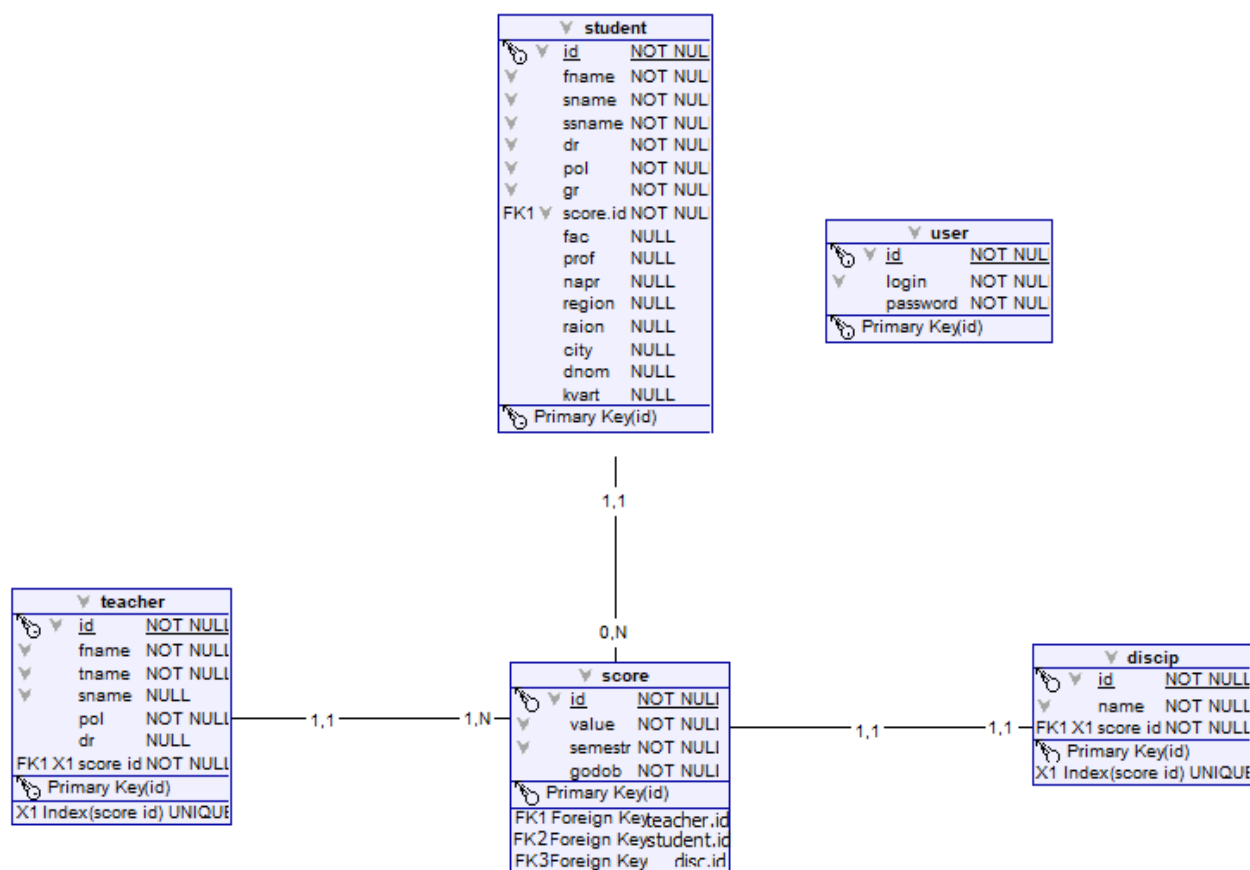


Рисунок 24 – Реляционная модель базы данных

2.4.2 Создание даталогической модели базы данных

Даталогическая модель – это описание структуры БД, на языке выбранной СУБД. В данной работе была выбрана СУБД MySQL. Требования, предъявляемые к системе можно полностью реализовать с использованием данной СУБД.

На рисунке 25 представлена физическая модель проектируемой базы данных.

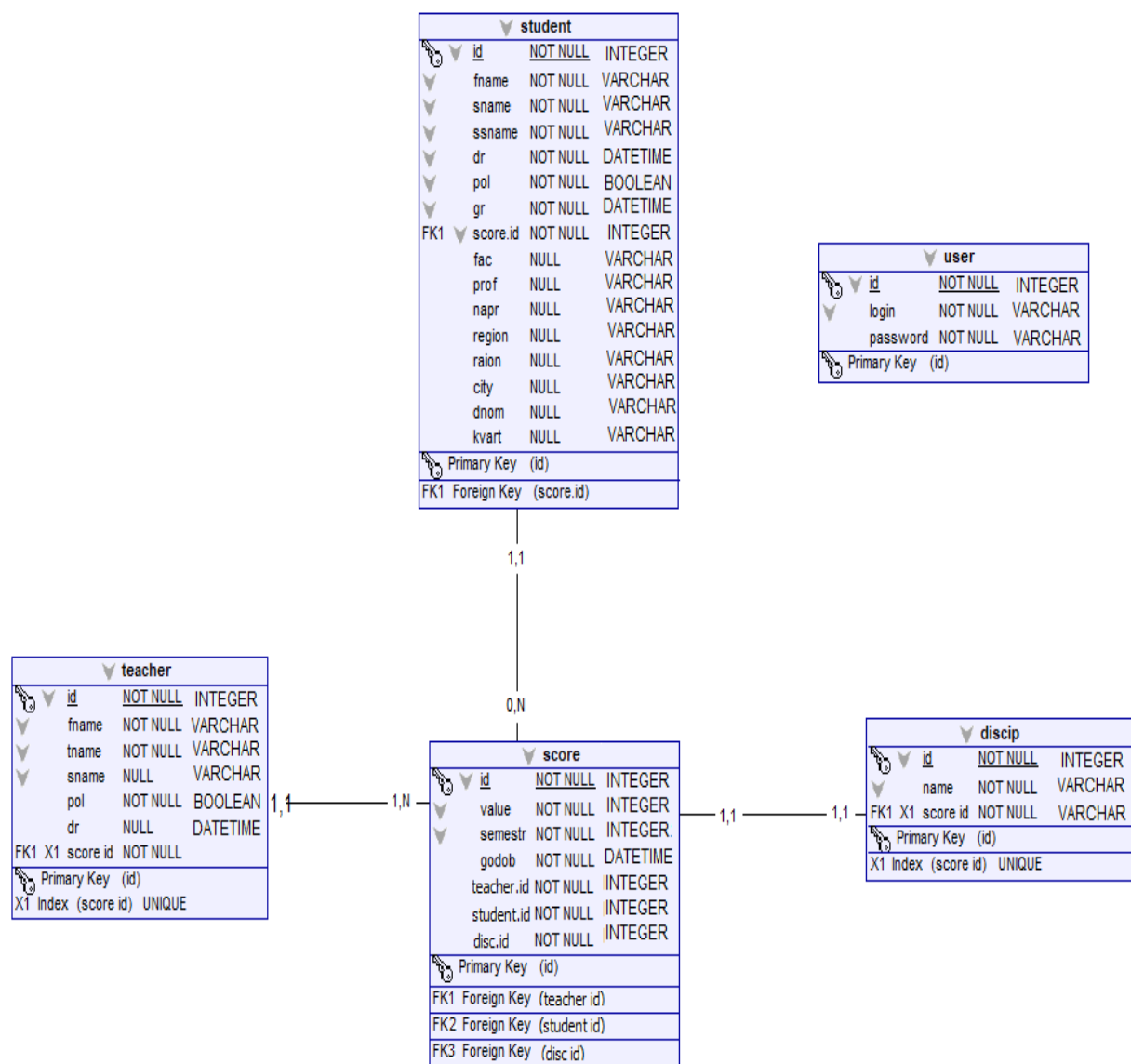


Рисунок 25 – Физическая модель базы данных

Результатом даталогического моделирования является схема базы данных. В ней находится описание всех полей таблиц с указанием их типов.

Схема базы данных представлена в таблице 2.

Таблица 2 - Элементы даталогической модели базы данных

Название таблицы	Описание таблицы	Название поля	Тип данных поля	Описание поля
1	2	3	4	5
student	Информация о студенте	id	integer	Уникальный номер студента
		fname	varchar	Фамилия студента
		sname	varchar	Имя студента
		ssname	varchar	Отчество студента
		dr	datetime	Дата рождения студента
		pol	boolean	Пол студента
		gr	integer	Группа студента
		fac	varchar	Факультет
		prof	varchar	Профиль
		napr	varchar	Направление
		region	varchar	Регион
		raion	varchar	Район
		city	varchar	Город
		dnom	varchar	Номер дома
		kvart	varchar	Квартира
teacher	Информация о преподавателе	id	integer	Уникальный номер преподавателя

Продолжение таблицы 2

1	2	3	4	5
		fname	varchar	Фамилия
		sname	varchar	Имя
		ssname	varchar	Отчество
		dr	datetime	Дата рождения
		pol	boolean	Пол
		dr	datetime	Дата рождения
		pol	boolean	Пол
user	Данные для входа в систему	id	integer	Уникальный номер пользователя
		login	varchar	Логин
		password	varchar	Пароль
discip	Данные о преподаваемой дисциплине	id	integer	Уникальный номер дисциплины
		name	varchar	Наименование
score	Данные об оценках	id	integer	Уникальный номер оценки
		value	integer	Значение
		semestr	integer	Семестр
		godob	datetime	Год обучения
		id_teacher	integer	Номер преподавателя
		id_student	integer	Номер студента

Продолжение таблицы 2

1	2	3	4	5
		id_disc	integer	Номер дисциплины

Таким образом была разработана физическая модель базы данных и схема базы данных. В результате проделанной работы были определены основные составляющие базы данных, выбраны средства разработки и платформа проектирования, разработаны структуры данных и схемы работы системы.

2.5 Тестирование

Тестирование программного обеспечения – важный и трудоемкий процесс, являющийся неотъемлемой частью разработки программного обеспечения.

Тестирование производится для промежуточного оценивания продуктов и программных средств. Оно, как правило, проводится на всем протяжении создания программного продукта и позволяет определить на сколько успешно разработчики справились со своей задачей [13].

Тесты составляют с учетом охвата максимального числа вариантов работы программы.

2.5.1 Методы тестирования

Статические методы тестирования используются при проведении инспектирования ПО (проверка программы соответствию заданным спецификациям, путем анализа представлений результатов проектирования на процессах жизненного цикла) и рассмотрении спецификаций компонентов без их выполнения [14].

					ПГУ 1.090501.01.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		47

Благодаря инспекции программ разработчиками и независимыми экспертами на различных этапах жизненного цикла проверяются и рассматриваются документы программ проектирования, что обеспечивает «взгляд со стороны». Этот прием позволяет на ранних этапах проектирования обнаружить дефекты и ошибки посредством многократных просмотров исходных кодов.

Динамическое тестирование ориентировано на проверку корректности ПС на множестве тестов, прогоняемых по ПС, в целях проверки и сбора данных на этапах ЖЦ и проведения измерения отдельных элементов тестирования для оценки характеристик качества, указанных в требованиях посредством выполнения системы на ЭВМ. Оно основывается на систематических, статистических, (вероятностных) и имитационных методах [15].

Выделяют две стратегии тестирования: методом «черного» ящика и методом «белого» ящика. Обе стратегии будут рассмотрены далее.

Метод «черного» ящика понимает под собой объект исследования, у которого неизвестным является внутреннее устройство. При тестировании изучение поведения системы возможно лишь при помощи манипулирования с входами и выходами. В настоящее время известны два вида «черных» ящиков. Первый – «черный» ящик, который рассматривается как конечный или бесконечный автомат, т.е. поведение такого «черного» ящика неизвестно. Второй вид – «черные» ящики, поведение которых можно наблюдать только в ходе эксперимента. В этом случае поведение «черного» ящика рассматривается в вероятностном смысле.

Метод «белого» ящика – это тестирование, которое основано на анализе внутренней структуры компонента или системы. Такое тестирование может проводиться на ранних этапах разработки, т.к. оно затрагивает только конкретные модули разрабатываемой системы.

В таблице 3 приведены некоторые виды и техники тестирования в контексте методов «белого» и «черного» ящиков.

					ПГУ 1.090501.01.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		48

Таблица 3 – Виды и техники тестирования

Вид тестирования	«Белый» ящик	«Черный» ящик
Статическое тестирование	Да	Нет
Динамическое тестирование	Изредка	Да
Ручное тестирование	Мало	Да
Автоматизированное тестирование	Да	Да
Модульное (компонентное) тестирование	Да	Нет
Интеграционное тестирование	Да	Да
Системное тестирование	Мало	Да
Дымовое тестирование	Мало	Да
Тестирование веб-приложений	Да	Да

2.5.2 Unit-тесты

Unit-тестирование (модульное тестирование) – это процесс в программировании, предназначенный для проверки работоспособности программного кода, а также его отдельных частей (модулей).

Данный подход к тестированию предполагает написание тестов для каждой функции программы, что способствует ускорению процесса тестирования без снижения его качества. За одну минуту можно выполнить тест несколько раз. Unit-тесты позволяют проводить тестирование точно для отдельных частей программного кода, не тратя время на разработку больших тестов, захватывающих не интересующие разработчика части кода.

Основные отличия модульного тестирования заключаются в том, что каждый тест фокусируется на отдельной небольшой части программного кода; тесты пишутся разработчиками; скорость выполнения тестов весьма высокая благодаря его фокусировке на отдельные части кода.

Unit-тесты подразделяются на одинокие (Solitary) и общительные (Sociable). Sociable тест – это тест, который в первую очередь полагается на существующие методы, которые уже находятся в тестируемом элементе.

Также этот тест может взаимодействовать с другими классами и вызывать их методы. Solitary тест – это тест, который использует дубли. Дубли – способы замены реального объекта для достижения цели тестирования.

Структура теста состоит из следующих частей: настройка тестовых данных; вызов метода; проверка результатов. После завершения тестирования возможно проведение рефакторинга кода для улучшения его качества.

2.5.3 Тесты в Laravel

В Laravel возможно создавать и проводить модульные и функциональные тесты. Тестирование осуществляется при использовании фреймворка PHPUnit.

Тестирование при помощи PHPUnit заключается в выполнении серии тестов, которые представляют собой фрагменты кода для автоматической проверки корректности модулей программного обеспечения. Каждый из них представляет собой отдельный Unit тест.

Для проверки работоспособности функции в коде используются модульные тесты, что также позволяет протестировать логическую часть кода. Для тестирования всего программного продукта используется метод функционального тестирования, в котором программное обеспечение рассматривается как единый «черный ящик».

Были разработаны тесты для некоторых функций разрабатываемой системы, что позволило быстро проверить их работоспособность и подтвердило отсутствие регрессии кода – появление ошибок в уже протестированных местах программы. Также это облегчило обнаружение и устранение имеющихся ошибок в коде.

PHPUnit – это фреймворк, применяющийся для тестирования приложений, написанных на языке программирования PHP. Он является экземпляром архитектуры xUnit для платформ модульного тестирования.

Для разрабатываемой системы были созданы автоматические тесты, которые проверяли исправную работу элементов системы авторизации и маршрутизации. На рисунке 26 представлен пример разработанного теста формы авторизации, а на рисунке 27 результат его выполнения в терминале.

```
public function test_user_can_view_a_login_form() {
    $response = $this->get('/login');
    $response->assertSuccessful();
    $response->assertViewIs('auth.login');
}

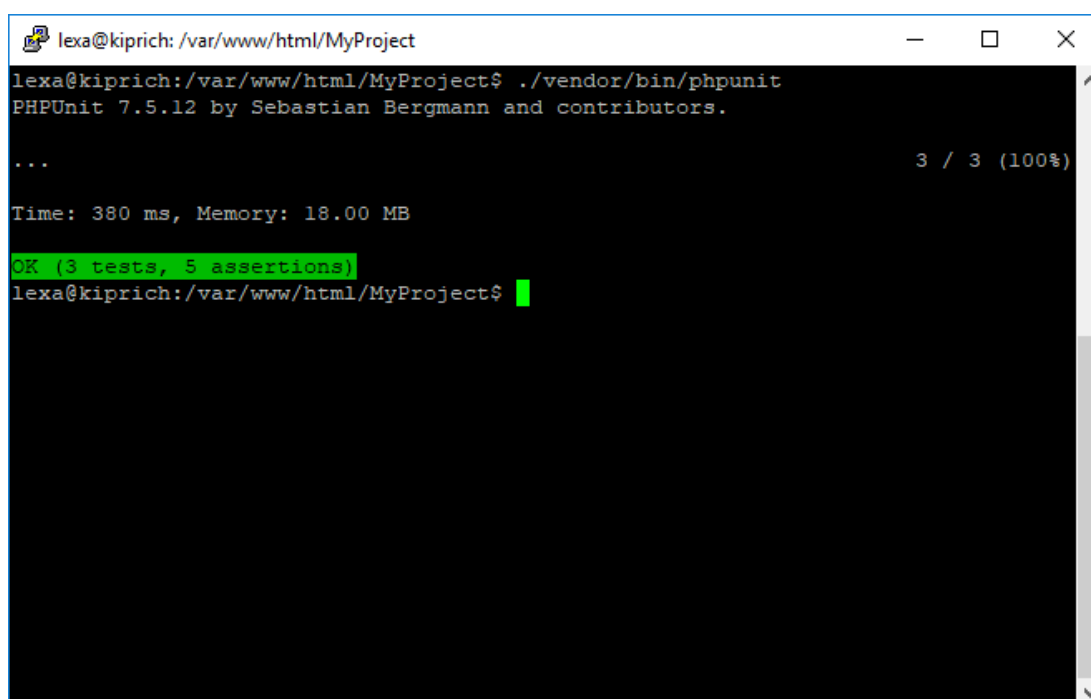
public function test_user_cannot_view_a_login_form_when_auth() {
    $user = factory(User::class)->make();
    $response = $this->actingAs($user)->get('/login');
    $response->assertRedirect('/home');
}

public function test_user_cannot_login_with_incorrect_password() {
    $user = factory(User::class)->create([
        'password' => bcrypt('PAROLNOTGOOD'),
    ]);

    $response = $this->from('/login')->post('/login', [
        'email' => $user->email,
        'password' => 'uehruger',
    ]);

    $response->assertRedirect('/login');
    $response->assertSessionHasErrors('email');
    $this->assertTrue(session()->hasOldInput('email'));
    $this->assertFalse(session()->hasOldInput('password'));
    $this->assertGuest();
}
```

Рисунок 26 – Тест системы авторизации



```
lexa@kiprich: /var/www/html/MyProject
lexa@kiprich:/var/www/html/MyProject$ ./vendor/bin/phpunit
PHPUnit 7.5.12 by Sebastian Bergmann and contributors.

...
3 / 3 (100%)

Time: 380 ms, Memory: 18.00 MB

OK (3 tests, 5 assertions)
lexa@kiprich:/var/www/html/MyProject$
```

Рисунок 27 – Результат тестирования

Данный тест включает в себя 3 функции, представленные на рисунке 26. Функция `test_user_can_view_a_login_form` предназначена для проверки наличия доступа к странице с формой авторизации у неавторизованного пользователя, путь к которой – `/login`. Если пользователь не авторизован в системе, то форма должна отобразиться с использованием вида `auth.login`.

Функция `test_user_cannot_view_a_login_form_when_auth` предназначена для проверки наличия доступа к странице с формой авторизации у авторизованного пользователя, путь к которой - `/login`. Если пользователь авторизован, то система автоматически осуществит перенаправление на главную страницу (`/home`).

Функция `test_user_cannot_login_with_incorrect_password` предназначена для проверка функционирования модуля авторизации. Тест осуществляет вход в систему с неверным паролем. В результате чего система определяет корректность введенных данных.

Использование фреймворка PHPUnit позволяет минимизировать время для написания тестов. Сами тесты создаются для конкретных функциональных блоков. Удобный интерфейс отображает результат тестирования на экране пользователя, а в случае выявления ошибок сообщает о них.

2.5.4 Методика тестирования приложения

С учетом рассмотренных выше способов и методов тестирования были предложены варианты тестирования при помощи Unit-тестов.

Цель тестирования – проверка работоспособности всего приложения. Результаты тестирования позволят определить корректность работы программного обеспечения.

Были определены тесты и их исполнители. Определена документация для отображения результатов тестирования. Проведена оценка ресурсов, необходимых для проведения тестов. Организован учет времени на выполнение отдельных частей тестирования ПО.

Модульное тестирование приложения проводится разработчиками при помощи встроенных средств фреймворка Laravel, а также путем ручного ввода данных. Протокол тестирования, содержащий список проведенных тестов и их результатов представлен в приложении В. По результатам тестирования можно определить на сколько корректно работает система и ее конкретные функциональные блоки.

3 Анализ опасных и вредных факторов, воздействующих на оператора ЭВМ, и способы противодействия им

3.1 Введение

Повсеместное распространение электронной вычислительной техники широко повлияло на условия производственной деятельности работников, а также внесло ряд изменений в организацию и порядок выполняемых работ на производстве. Труд работников умственного труда с каждым днем становится более напряженным и интенсивным, а для выполнения работ требуются большие затраты эмоциональной и физической энергии. Данные факты привели к увеличению роли защиты здоровья сотрудников. Процесс обеспечения безопасности рабочих должен облегчать труд человека и обеспечивать как можно больше комфортных и не угрожающих здоровью рабочих условий.

Современные технологии не могут исключить разрушительного воздействия на здоровье работников, но, не смотря на это, сами пользователи ЭВМ могут приводить его к минимуму. При выполнении работ на компьютере требуется соблюдение методов безопасной работы, а незнание данных методов может стать причиной производственных травм, пренебрежение методами безопасной работы может принести большие неприятности.

Решение большей части проблем, вызванных работой на ЭВМ, заключается в корректной и правильной организации рабочего места работника. Помимо этого необходимо учитывать правила и технику безопасности при выполнении работ за персональным компьютером. Рабочее время сотрудника должно быть правильно распределено.

В данном разделе будет проведен анализ опасных и вредных факторов, воздействующих на оператора ЭВМ, а также способы и методы противодействия им.

					ПГУ 1.090501.01.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		54

3.2 Аспекты производственной безопасности при работе на ПК

На данный момент существуют специальные мероприятия по поддержанию безопасности ведения хозяйства, а также специализированные технические приспособления, которые повышают степень защиты. Применение данных методов показывает, что влияние опасных факторов на здоровье оператора значительно снижается.

Опасным называется производственный фактор, воздействие которого на работающего человека в определенных условиях приводит к травме или другому внезапному резкому ухудшению здоровья. Если же производственный фактор приводит к заболеванию или снижению трудоспособности, то его считают вредным. В зависимости от уровня и продолжительности воздействия вредный производственный фактор может стать опасным. Опасные и вредные производственные факторы подразделяются на четыре группы: физические, химические, биологические и психофизиологические [16].

3.2.1 Психофизиологические факторы

Психофизиологические факторы делятся на две группы:

1) Нервно-психические перегрузки. Данная группа разделяется на перенапряжение умственное, анализаторов, включает в себя монотонность труда и перегрузки эмоционального плана.

2) Физические перегрузки (статические и динамические).

Монотонность - психическое состояние человека, вызванное однообразием восприятий или действий. Это состояние характерно для работы оператора ПК, которая связана с процессом ввода однообразной информации. В связи с этим очень часто у сотрудника возможно возникновение рабочего стресса [17].

По статистике от 66 до 100 процентов работающих людей подвергаются рабочему стрессу. В группу риска занесены по большей части монотонные, изматывающие и спокойные виды деятельности.

Для уменьшения утомления, связанного с влиянием данного фактора предлагается применять комплексы специализированных упражнений, а также поддерживать оптимальные параметры режимов выполнения работ и дальнейшего отдыха в течение всего рабочего дня. При работе за персональным компьютером необходимо проводить перерывы длительностью 15 минут через каждые два часа работы. Если работа интенсивная, то – через каждый час. Рекомендуется выполнять два любых упражнения на гимнастику глаз.

Для предупреждения возникновения стресса при работе разработаны следующие правила:

- только половина рабочего времени должна проходить за экраном монитора;
- необходимо наличие 15-минутных перерывов при выполнении работ на ЭВМ;
- видеоадаптеры должны выбираться с частотой обновления экрана не менее 75 Гц, рекомендуется 85 или 100 Гц;
- при вводе данных с клавиатуры не рекомендуется зажимать ухом и плечом телефон.

Рабочая поза оператора очень сильно сказывается на результатах выполненных задач. Основные требования к рабочему месту при выполнении сидячей работы доведены в ГОСТ 12.2.032-78 «ССБТ.

Рабочее место при выполнении работ сидя. Общие эргономические требования» [17]. Для выполнения работы сидя рабочее место организуют только при легкой работе, которая не подразумевает передвижения работающего. Расположение и конструкция элементов рабочего места должна соответствовать физиологическим и психологическим требованиям и характеру работы сотрудника.

					ПГУ 1.090501.01.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		56

Для правильной организации рабочего места созданы специальные требования стандартов и технических условий. На рабочем месте человек проводит большую часть рабочего дня, поэтому его необходимо приспособить для двух или более позиций.

К обслуживанию ЭВМ, а также выполнению задач на ней допускаются лица, которые прошли медицинское освидетельствование при устройстве на работу. Далее медосмотры проводятся один раз в два года.

3.2.2 Защита от излучений

Основным источником проблем, связанных с охраной здоровья людей, использующих в своей работе персональные компьютеры, являются дисплеи (мониторы), особенно дисплеи с электронно-лучевыми трубками. Они представляют собой источники наиболее вредных излучений, неблагоприятно влияющих на здоровье операторов [17].

Персональные компьютеры во время работы создают электромагнитные излучения: рентгеновское, ультрафиолетовое, высокочастотное и низкочастотное.

Главную опасность в первую очередь представляет излучение монитора в диапазоне от 20 Гц до 300 МГц. Данное излучение распространяется во всех направлениях от монитора и оказывает воздействие на расстоянии до 5 метров. Было обнаружено, что электромагнитные поля с частотой 60 Гц могут вызывать изменения в клетках живых существ.

Мониторы излучают магнитные волны, которые по своей интенсивности не уступают уровням магнитных полей, что может обуславливать возникновение опухолей. Разработаны специальные стандарты для излучений очень-низко-частотных (ОНЧ)-магнитных полей, которым соответствует продукция многих поставщиков (Philips, Samsung). Современные мониторы не излучают переменных компонент, связанных с наличием систем вертикального и горизонтального отклонения луча.

					ПГУ 1.090501.01.001 ПЗ	Лист
						57
Изм.	Лист	№ докум.	Подпись	Дата		

При желании снижения уровня облучения переменными магнитными полями необходимо разместить мониторы так, чтобы расстояние от глаз пользователя до монитора составляло величину, равную расстоянию вытянутой руки.

3.2.3 Оборудование рабочих мест с ПК

Требования к организации и оборудованию рабочего места оператора ПК приведены в ГОСТ 12.2.032-78 [17]. Высота столешницы для операторов должна регулироваться в пределах 680-800 мм; если регулировка невозможна, то стоит придерживаться значения 725 мм.

Размеры поверхности стола для работа за ним на ПК должны быть от 800 до 1400 мм в ширину и 800-1000 мм в глубину.

При разработке стола необходимо учесть наличие пространства для ног с высотой от 600 мм и шириной от 500 мм. Глубина места под ноги на уровне колен должна составлять не менее 450 мм, а при вытянутых ногах – от 650 мм.

Рабочее сиденье оборудуется с учетом возможности использования рычагов регулировки по высоте и углам наклона как сиденья, так и спинки.

Подставка для ног должна быть оборудована на рабочем месте. Параметры подставки для ног должны быть не менее 300 мм в ширине, не менее 400 мм в глубине, высота может регулироваться в пределах до 150 мм.

Рекомендуется наличие изменения угла наклона подставки до 20 градусов. Поверхность должна быть сделана из рифленого материала, а также иметь бортик 5-10 мм.

Местоположение клавиатуры – на поверхности стола или на специальной рабочей подставке. Расстояние от края до клавиатуры 150-300 мм.

3.2.4 Электробезопасность

Компьютер является электрическим устройством, которое работает в сети переменного тока 220В.

Для предотвращения поражения электрическим током, а также возникновения пожара и выхода компьютера из строя необходимо соблюдать меры предосторожности: кабели питания должны быть подключены в исправные розетки; отсоединять провода от компьютера в процессе его работы; наносить краску на провода; выдергивать вилку за шнур; работать с оборудованием мокрыми руками; работать на машинах с поврежденным корпусом, проводкой.

Наиболее частыми электротравмами являются однополюсные (однофазные) прикосновения в изолированных и глухо заземленных сетях.

Для защиты жизни и здоровья человека при работе в компьютерном зале необходимо обеспечить надежное заземление проводов. Комплекс работ должен быть обычным, уровень допустимой величины не должен быть выше 90-100 Ом. Если корпус ЭВМ не заземлен, то прикосновение к нему может быть опасным для здоровья и нанести непоправимый вред. Заземленный ток не является опасным для человека.

3.2.5 Освещение рабочего места

Освещенность рабочего места оператора ПК должна быть не очень яркой и меньше, чем при работе с бумагами. При одновременной работе с бумагами и за компьютером необходимо использовать настольную лампу, которая будет направлена на документы. Рядом с монитором запрещено размещать предметы, которые могут создавать блики и отражать свет.

Для избегания бликов монитор должен быть расположен перпендикулярно окну. Не следует располагать монитор на фоне яркого окна – это оказывает дополнительную нагрузку на зрение. Наилучшее расположение окна – слева. Регулировку освещения рекомендуется производить при помощи штор или жалюзи.

На рабочей поверхности не рекомендуется присутствие резких теней. Вредны движущиеся тени, т.к. они могут привести к травмам. Тени можно смягчать за счет использования светильников со светорассеивающими стеклами.

В поле зрения не должно быть блескости – повышенной яркости светящихся поверхностей. Как правило блескость вызывает нарушение зрительных функций. Ее можно снизить при помощи уменьшения яркости источника света, либо при правильном выборе рациональных углов светильника.

Свечение со стороны экрана, а также частая смена заставок на нем может отрицательно воздействовать на зрение. Это необходимо учитывать при работе за ЭВМ.

					ПГУ 1.090501.01.001 ПЗ	Лист
						60
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

В процессе работы над дипломным проектом были разработаны следующие составные части аналитической системы управления учебным процессом личного состава: модуль аналитики, модуль графического отображения, модуль авторизации. Для этого была исследована предметная область и выявлены информационные потребности пользователя, осуществлен выбор средств разработки, спроектирована база данных системы, определены входные и выходные данные, разработан алгоритм функционирования системы и пользовательский интерфейс.

В технологическом разделе дипломного проекта были рассмотрены технологии, используемые при разработке, описаны их основные свойства, технические характеристики и область применения. Описаны основные методы тестирования, методика их применения при тестировании аналитической системы и результаты проведенного тестирования.

В разделе анализа опасных и вредных факторов, воздействующих на оператора ЭВМ, и способов противодействия им были рассмотрены вопросы безопасности жизнедеятельности при работе за ЭВМ. Данные вопросы имеют большое значение при анализе факторов и принципов охраны труда программистов и операторов ПК на предприятии, которые используют данную аналитическую систему.

На данный момент планируется внедрение и использование аналитической системы в рамках электронной информационно-образовательной среды Пензенского государственного университета.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Прока, Н.И. Экономика труда в организациях АПК [Текст] / Н.И. Прока – Москва: КолосС, 2009. – 440 с.
2. Moodle - система дистанционного обучения/ [Электронный ресурс] // Moodle - система дистанционного обучения - Режим доступа URL. <https://opentechnology.ru/products/moodle> [Дата обращения 30.04.2019 г.].
3. SIDPERS - Standard Installation/Division Personnel System / [Электронный ресурс] // SIDPERS - Standard Installation/Division Personnel System – Режим доступа URL. [https://www.acronymfinder.com/Standard-Installation%2FDivision-Personnel-System-\(SIDPERS\).html](https://www.acronymfinder.com/Standard-Installation%2FDivision-Personnel-System-(SIDPERS).html) [Дата обращения: 30.04.2019 г.].
4. Конобеева, А.Б. Формирование профессиональных компетенций в ВУЗе: сочетание универсальных профессиональных компетенций [Текст] / А.Б. Конобеева // Вестник МГУКИ. – 2010. - № 1 (33) – С. 182-185.
5. Лекция 2: Основные элементы языка UML / [Электронный ресурс] // НОУ ИНТУИТ | Лекция | Основные элементы языка UML – Режим доступа URL. <https://www.intuit.ru/studies/courses/32/32/lecture/1002> [Дата обращения: 20.04.2019 г.].
6. Язык UML. Руководство пользователя. Глава 7. Диаграммы / [Электронный ресурс] // Глава 7 – Режим доступа URL. http://alice.pnzgu.ru:8080/~zsa/sql/titan_zsa/uml_htm_gol/gl_07.htm [Дата обращения: 20.04.2019 г.].
7. Фаулер, М. UML. Основы [Текст] / М. Фаулер, К. Скотт. – СПб: Символ, 2006. – 184 с.
8. Розенберг, Д. Применение объектного моделирования с использованием UML и анализ прецедентов [Текст] / Д. Розенберг, К. Скотт. – М.: ДМК Пресс, 2002. – 124 с.

9. Фиайли, К. SQL [Текст] / К. Фиайли. – М.: ДМК Пресс, 2013. – 456 с.
10. Дронов, В.А. Laravel. Быстрая разработка современных динамических Web-сайтов на PHP, MySQL, HTML и CSS [Текст] / В.А. Дронов. – СПб: БХВ-Петербург, 2017. – 755 с.
11. Эмблер, С.В. Рефакторинг баз данных. Эволюционное проектирование [Текст] / С.В. Эмблер, П. Дж. Садаладж. - М.: Вильямс, 2007. – 380 с.
12. Гарсиа-Молина, Г. Системы баз данных. Полный курс [Текст] / Г. Гарсиа-Молина, Д. Ульман. - М.: Вильямс, 2004. – 572 с.
13. Бек, К. Экстремальное программирование. Разработка через тестирование [Текст] / К. Бек. – СПб: Питер, 2003. – 187 с.
14. Бейзер, Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем [Текст] / Б. Бейзер. – СПб: Питер, 2004. – 235 с.
15. Тестирование. Фундаментальная теория / [Электронный ресурс] // Тестирование. Фундаментальная теория. Хабр. – Режим доступа URL. <https://habr.com/ru/post/279535/> [Дата обращения: 22.04.2019 г.].
16. Охрана труда. Опасные и вредные производственные факторы / [Электронный ресурс] // Охрана труда. Опасные и вредные производственные факторы. – Режим доступа URL. http://www.gigiena-saratov.ru/aktyalnye_temy/usltrud/147277/ [Дата обращения: 15.04.2019 г.].
17. ГОСТ 12.2.032-78. Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования. – М.: Издательство стандартов, 1978. – 8 с.

ПРИЛОЖЕНИЕ А

(обязательное)

Текст программы

Ниже приводится листинг основных модулей программного кода аналитической системы поддержки управления личным составом.

Базовый контроллер страниц

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth');
    }

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Contracts\Support\Renderable
     */
    public function index()
    {
        return view('home');
    }
}
```

					ПГУ 1.090501.01.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		64

}

Контроллер модуля аналитики

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Exports\ExportUsers;

use App\Imports\ImportStudents;

use Illuminate\Routing\Controller as BaseController;

use Maatwebsite\Excel\Facades\Excel;

use App\Discipline;

use App\Score;

use App\Student;

use App\Teacher;

use Redirect;

use View;

class StatsController extends BaseController

{

public function setup() {

\$vals = Score::distinct()->get(['value'])->sort();

\$grs = Student::distinct()->get(['gr'])->sort();

\$facs = Student::distinct()->get(['fac'])->sort();

\$profs = Student::distinct()->get(['prof'])->sort();

\$naprs = Student::distinct()->get(['napr'])->sort();

\$regs = Student::distinct()->get(['region'])->sort();

\$rais = Student::distinct()->get(['raion'])->sort();

\$cs = Student::distinct()->get(['city'])->sort();

\$discs = Discipline::distinct()->get(['name'])->sort();

return \View::make("stats")->with(array(

'vals' => \$vals,

'grs' => \$grs,

'facs' => \$facs,

'profs' => \$profs,

'naprs' => \$naprs,

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подпись | Дата | | 65 |

```

        'regs' => $regs,
        'rais' => $rais,
        'cs' => $cs,
        'discs' => $discs,
    ));
}

public function draw(Request $request) {
    $input = $request->all();
    $p1 = array_count_values($input);
    $error = 0;
    if ($p1['Не учитывать'] < 12) {
        $error = 1;
        $terr = 'Выберите не более двух параметров.';
    }
    if ($p1['Не учитывать'] > 12) {
        $error = 2;
        $terr = 'Выберите параметр.';
    }
    if ($input[$input['optionsRadios']] != 'Не учитывать') {
        $error = 3;
        $terr = 'Параметры не должны совпадать.';
    }

    if ($error > 0) {
        return redirect('error')->with('terr', $terr);
    } else {

        if ($input['disc'] != 'Не учитывать') {
            $terr = 'Параметр задан неверно.';
            return redirect('error')->with('terr', $terr);
        }

        if ($input['score'] != 'Не учитывать') {
            $q_var = $input['score'];

```

```
$dataX = Score::where('value', '=', $q_var)->get();
```

```
if ($input['optionsRadios'] == 'gr' || $input['optionsRadios'] == 'fac' ||  
$input['optionsRadios'] == 'prof' || $input['optionsRadios'] == 'napr' ||  
$input['optionsRadios'] == 'region' || $input['optionsRadios'] == 'raion' ||  
$input['optionsRadios'] == 'pol' || $input['optionsRadios'] == 'city') {
```

```
$vr = $input['optionsRadios'];
```

```
$vr = 'students.' . $vr;
```

```
$subset = \DB::table('students')
```

```
->join('scores', function ($join) use ($q_var) {
```

```
    $join->on('students.id', '=', 'scores.id_student')
```

```
    ->where('scores.value', '=', $q_var);
```

```
})
```

```
->select($vr)
```

```
->get();
```

```
$subset = json_decode($subset, true);
```

```
// $subset = $dataX->map->only([$vr]);
```

```
$data = array();
```

```
for ($i = 0; $i < sizeof($subset); $i++) {
```

```
    $data[$subset[$i][str_replace('students.', '', $vr)]] = 0;
```

```
}
```

```
ksort($data);
```

```
for ($i = 0; $i < sizeof($subset); $i++) {
```

```
    $data[$subset[$i][str_replace('students.', '', $vr)]]++;
```

```
}
```

```
if ($vr == 'students.pol') {
```

```
    $data['Мужской'] = $data[1];
```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | 67 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

        unset($data[1]);
        $data['Женский'] = $data[0];
        unset($data[0]);
    }

    return \View::make("draw")->with(array(
        'data' => $data,
    ));
}

if ($input['optionsRadios'] == 'godob' || $input['optionsRadios'] == 'semestr') {

    $vr = $input['optionsRadios'];

    if ($vr == 'score') $vr = 'value';
    $subset = $dataX->map->only([$vr]);

    $data = array();
    foreach ($subset as $k) {
        $data[$k[$vr]] = 0;
    }

    foreach ($subset as $k) {
        $data[$k[$vr]]++;
    }

    return \View::make("draw")->with(array(
        'data' => $data,
    ));

}

if ($input['optionsRadios'] == 'dr') {
    $vr = $input['optionsRadios'];

```

```

$vr = 'students.' . $vr;
$subset = \DB::table('students')
->join('scores', function ($join) use ($q_var) {
    $join->on('students.id', '=', 'scores.id_student')
    ->where('scores.value', '=', $q_var);
})
->select($vr)
->get();
$subset = json_decode($subset, true);

$data = array();
for ($i = 0; $i < sizeof($subset); $i++) {
    $date = explode('-', $subset[$i][str_replace('students.', '', $vr)]);
    $y = $date[0];
    $data[strval($y)] = 0;
}
ksort($data);
for ($i = 0; $i < sizeof($subset); $i++) {
    $date = explode('-', $subset[$i][str_replace('students.', '', $vr)]);
    $y = $date[0];
    $data[strval($y)]++;
}

return \View::make("draw")->with(array(
    'data' => $data,
));
}

if ($input['optionsRadios'] == 'disc') {
    $ids = Discipline::where('id', '>', 0)->get();
    $ids = $ids->map->only(['id', 'name']);

    $vr = 'id_disc';
    $subset = $dataX->map->only([$vr]);

```

```

        $data = array();
        foreach ($subset as $k) {
            $data[$k[$vr]] = 0;
        }

        foreach ($subset as $k) {
            $data[$k[$vr]]++;
        }

        foreach ($data as $k => $d) {
            $name = Discipline::find($k)->name;
            $data[$name] = $data[$k];
            unset($data[$k]);
        }
        return \View::make("draw")->with(array(
            'data' => $data,
        ));
    }
}

if ($input['semestr'] != 'Не учитывать') {
    $q_var = $input['semestr'];
    if ($q_var == 'Зимний') $q_var = 1;
    else $q_var = 2;

    $dataX = Score::where('semestr', '=', $q_var)->get();

    if ($input['optionsRadios'] == 'gr' || $input['optionsRadios'] == 'fac' ||
        $input['optionsRadios'] == 'prof' || $input['optionsRadios'] == 'napr' ||
        $input['optionsRadios'] == 'region' || $input['optionsRadios'] == 'raion' ||
        $input['optionsRadios'] == 'pol' || $input['optionsRadios'] == 'city') {

        $vr = $input['optionsRadios'];
    }
}

```

```

$vr = 'students.' . $vr;
$subset = \DB::table('students')
->join('scores', function ($join) use ($q_var) {
    $join->on('students.id', '=', 'scores.id_student')
    ->where('scores.semestr', '=', $q_var);
})
->select($vr)
->get();

$subset = json_decode($subset, true);
//$subset = $dataX->map->only([$vr]);

$data = array();
for ($i = 0; $i < sizeof($subset); $i++) {
    $data[$subset[$i][str_replace('students.', '', $vr)]] = 0;
}
ksort($data);

for ($i = 0; $i < sizeof($subset); $i++) {
    $data[$subset[$i][str_replace('students.', '', $vr)]]++;
}

if ($vr == 'students.pol') {
    $data['Мужской'] = $data[1];
    unset($data[1]);
    $data['Женский'] = $data[0];
    unset($data[0]);
}

return \View::make("draw")->with(array(
    'data' => $data,
));
}

```

```

if ($input['optionsRadios'] == 'godob' || $input['optionsRadios'] == 'score') {

    $vr = $input['optionsRadios'];

    if ($vr == 'score') $vr = 'value';
    $subset = $dataX->map->only([$vr]);

    $data = array();
    foreach ($subset as $k) {
        $data[$k[$vr]] = 0;
    }

    foreach ($subset as $k) {
        $data[$k[$vr]]++;
    }

    return \View::make("draw")->with(array(
        'data' => $data,
    ));

}

if ($input['optionsRadios'] == 'dr') {
    $vr = $input['optionsRadios'];

    $vr = 'students.' . $vr;
    $subset = \DB::table('students')
->join('scores', function ($join) use ($q_var) {
        $join->on('students.id', '=', 'scores.id_student')
        ->where('scores.semestr', '=', $q_var);
    })
->select($vr)
->get();
    $subset = json_decode($subset, true);
}

```

```

        if ($input['optionsRadios'] == 'godob' || $input['optionsRadios'] == 'semestr' ||
        $input['optionsRadios'] == 'score') {

```

```

            $vr = $input['optionsRadios'];

```

```

            if ($vr == 'score') $vr = 'value';

```

```

            $vr = 'scores.' . $vr;

```

```

            $subset = \DB::table('students')

```

```

            ->join('scores', function ($join) use ($q_var) {

```

```

                $join->on('students.id', '=', 'scores.id_student')

```

```

                ->where('students.city', '=', $q_var);

```

```

            })

```

```

            ->select($vr)

```

```

            ->get();

```

```

            $subset = json_decode($subset, true);

```

```

            $data = array();

```

```

            for ($i = 0; $i < sizeof($subset); $i++) {

```

```

                $data[$subset[$i][str_replace('scores.', '', $vr)]] = 0;

```

```

            }

```

```

            ksort($data);

```

```

            for ($i = 0; $i < sizeof($subset); $i++) {

```

```

                $data[$subset[$i][str_replace('scores.', '', $vr)]]++;

```

```

            }

```

```

            return \View::make("draw")->with(array(

```

```

                'data' => $data,

```

```

            ));

```

```

        }

```

```

if ($input['optionsRadios'] == 'dr') {
    $vr = $input['optionsRadios'];
    $subset = $dataX->map->only([$vr]);

    $data = array();
    foreach ($subset as $k) {
        $date = explode('-', $k[$vr]);
        $data[$date[0]] = 0;
    }

    $data = array();

    for ($i = 0; $i < sizeof($subset); $i++) {
        $data[$subset[$i][str_replace('scores.', '', $vr)]] = 0;
    }
    ksort($data);

    for ($i = 0; $i < sizeof($subset); $i++) {
        $data[$subset[$i][str_replace('scores.', '', $vr)]]++;
    }

    return \View::make("draw")->with(array(
        'data' => $data,
    ));
}

if ($input['optionsRadios'] == 'dr') {
    $vr = $input['optionsRadios'];
    $subset = $dataX->map->only([$vr]);

    $data = array();
    foreach ($subset as $k) {
        $date = explode('-', $k[$vr]);

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подпись | Дата | | 74 |

```

        $data[$date[0]] = 0;
    }

    foreach ($subset as $k) {
        $date = explode('-', $k[$vr]);
        $data[$date[0]]++;
    }

    return \View::make("draw")->with(array(
        'data' => $data,
    ));
}

if ($input['optionsRadios'] == 'disc') {
    $ids = Discipline::where('id', '>', 0)->get();
    $ids = $ids->map->only(['id', 'name']);

    $vr = 'scores.id_disc';
    $subset = \DB::table('students')
    ->join('scores', function ($join) use ($q_var) {
        $join->on('students.id', '=', 'scores.id_student')
        ->where('students.region', '=', $q_var);
    })
    ->select($vr)
    ->get();

    $subset = json_decode($subset, true);

    $data = array();

    for ($i = 0; $i < sizeof($subset); $i++) {
        $data[$subset[$i][str_replace('scores.', '', $vr)]] = 0;
    }
}

```

```

ksort($data);

for ($i = 0; $i < sizeof($subset); $i++) {
    $data[$subset[$i][str_replace('scores.', '', $vr)]]++;
}

foreach ($data as $k => $d) {
    $name = Discipline::find($k)->name;
    $data[$name] = $data[$k];
    unset($data[$k]);
}

return \View::make("draw")->with(array(
    'data' => $data,
));

}

}

if ($input['napr'] != 'Не учитывать') {
    $q_var = $input['napr'];

    $dataX = Student::where('napr', '=', $q_var)->get();

    if ($input['optionsRadios'] == 'gr' || $input['optionsRadios'] == 'fac' ||
        $input['optionsRadios'] == 'prof' || $input['optionsRadios'] == 'region' ||
        $input['optionsRadios'] == 'raion' || $input['optionsRadios'] == 'city' ||
        $input['optionsRadios'] == 'pol') {

        $vr = $input['optionsRadios'];
        $subset = $dataX->map->only([$vr]);

        $data = array();
        foreach ($subset as $k) {
            $data[$k[$vr]] = 0;
        }
    }
}

```

```

foreach ($subset as $k) {
    $data[$k[$v]]++;
}

if ($v == 'pol') {
    $data['Мужской'] = $data[1];
    unset($data[1]);
    $data['Женский'] = $data[0];
    unset($data[0]);
}
return \View::make("draw")->with(array(
    'data' => $data,
));
}

```

```

if ($input['optionsRadios'] == 'godob' || $input['optionsRadios'] == 'semestr' ||
$input['optionsRadios'] == 'score') {

```

```

    $v = $input['optionsRadios'];

```

```

    if ($v == 'score') $v = 'value';

```

```

    $v = 'scores.' . $v;

```

```

    $subset = \DB::table('students')

```

```

->join('scores', function ($join) use ($q_var) {

```

```

    $join->on('students.id', '=', 'scores.id_student')

```

```

    ->where('students.napr', '=', $q_var);

```

```

})

```

```

->select($v)

```

```

->get();

```

```

    $subset = json_decode($subset, true);

```

```

    $data = array();

```

```

for ($i = 0; $i < sizeof($subset); $i++) {
    $data[$subset[$i][str_replace('scores.', '', $vr)]] = 0;
}
ksort($data);

for ($i = 0; $i < sizeof($subset); $i++) {
    $data[$subset[$i][str_replace('scores.', '', $vr)]]++;
}

return \View::make("draw")->with(array(
    'data' => $data,
));
}

if ($input['optionsRadios'] == 'dr') {
    $vr = $input['optionsRadios'];
    $subset = $dataX->map->only([$vr]);

    $data = array();
    foreach ($subset as $k) {
        $date = explode('-', $k[$vr]);
        $data[$date[0]] = 0;
    }

    foreach ($subset as $k) {
        $date = explode('-', $k[$vr]);
        $data[$date[0]]++;
    }

    return \View::make("draw")->with(array(
        'data' => $data,
    ));
}

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подпись | Дата | | 78 |

```

}

if ($input['optionsRadios'] == 'disc') {
    $ids = Discipline::where('id', '>', 0)->get();
    $ids = $ids->map->only(['id', 'name']);

    $vr = 'scores.id_disc';
    $subset = \DB::table('students')
        ->join('scores', function ($join) use ($q_var) {
            $join->on('students.id', '=', 'scores.id_student')
            ->where('students.napr', '=', $q_var);
        })
        ->select($vr)
        ->get();

    $subset = json_decode($subset, true);

    $data = array();

    for ($i = 0; $i < sizeof($subset); $i++) {
        $data[$subset[$i][str_replace('scores.', '', $vr)]] = 0;
    }
    ksort($data);

    for ($i = 0; $i < sizeof($subset); $i++) {
        $data[$subset[$i][str_replace('scores.', '', $vr)]]++;
    }

    foreach ($data as $k => $d) {
        $name = Discipline::find($k)->name;
        $data[$name] = $data[$k];
        unset($data[$k]);
    }

    return \View::make("draw")->with(array(
        'data' => $data,

```

```

    ));

    }

}

if ($input['prof'] != 'Не учитывать') {
    $q_var = $input['prof'];

    $dataX = Student::where('prof', '=', $q_var)->get();

    if ($input['optionsRadios'] == 'gr' || $input['optionsRadios'] == 'fac' ||
        $input['optionsRadios'] == 'napr' || $input['optionsRadios'] == 'region' ||
        $input['optionsRadios'] == 'raion' || $input['optionsRadios'] == 'city' ||
        $input['optionsRadios'] == 'pol') {

        $vr = $input['optionsRadios'];
        $subset = $dataX->map->only([$vr]);

        $data = array();
        foreach ($subset as $k) {
            $data[$k[$vr]] = 0;
        }

        foreach ($subset as $k) {
            $data[$k[$vr]]++;
        }

        if ($vr == 'pol') {
            $data['Мужской'] = $data[1];
            unset($data[1]);
            $data['Женский'] = $data[0];
            unset($data[0]);
        }

        return \View::make("draw")->with(array(
            'data' => $data,

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | 80 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```
));  
}
```

```
if ($input['optionsRadios'] == 'godob' || $input['optionsRadios'] == 'semestr' ||  
$input['optionsRadios'] == 'score') {
```

```
$vr = $input['optionsRadios'];
```

```
if ($vr == 'score') $vr = 'value';
```

```
$vr = 'scores.' . $vr;
```

```
$subset = \DB::table('students')
```

```
->join('scores', function ($join) use ($q_var) {
```

```
    $join->on('students.id', '=', 'scores.id_student')
```

```
    ->where('students.prof', '=', $q_var);
```

```
})
```

```
->select($vr)
```

```
->get();
```

```
$subset = json_decode($subset, true);
```

```
$data = array();
```

```
for ($i = 0; $i < sizeof($subset); $i++) {
```

```
    $data[$subset[$i][str_replace('scores.', '', $vr)]] = 0;
```

```
}
```

```
ksort($data);
```

```
for ($i = 0; $i < sizeof($subset); $i++) {
```

```
    $data[$subset[$i][str_replace('scores.', '', $vr)]]++;
```

```
}
```

```
return \View::make("draw")->with(array(
```

```
    'data' => $data,
```

```
));
```

```

}

if ($input['optionsRadios'] == 'dr') {
    $vr = $input['optionsRadios'];
    $subset = $dataX->map->only([$vr]);

    $data = array();
    foreach ($subset as $k) {
        $date = explode('-', $k[$vr]);
        $data[$date[0]] = 0;
    }

    foreach ($subset as $k) {
        $date = explode('-', $k[$vr]);
        $data[$date[0]]++;
    }

    return \View::make("draw")->with(array(
        'data' => $data,
    ));
}

if ($input['optionsRadios'] == 'disc') {
    $ids = Discipline::where('id', '>', 0)->get();
    $ids = $ids->map->only(['id', 'name']);

    $vr = 'scores.id_disc';
    $subset = \DB::table('students')
->join('scores', function ($join) use ($q_var) {
        $join->on('students.id', '=', 'scores.id_student')
->where('students.prof', '=', $q_var);
    })
->select($vr)

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГЧ 1.090501.01.001 ПЗ | л/сч |
| | | | | | | |
| Изм. | л/сч | № докум. | Подпись | Дата | | 82 |

```
->get();
```

```
$subset = json_decode($subset, true);
```

```
$data = array();
```

```
for ($i = 0; $i < sizeof($subset); $i++) {
```

```
    $data[$subset[$i][str_replace('scores.', '', $vr)]] = 0;
```

```
}
```

```
ksort($data);
```

```
for ($i = 0; $i < sizeof($subset); $i++) {
```

```
    $data[$subset[$i][str_replace('scores.', '', $vr)]]++;
```

```
}
```

```
foreach ($data as $k => $d) {
```

```
    $name = Discipline::find($k)->name;
```

```
    $data[$name] = $data[$k];
```

```
    unset($data[$k]);
```

```
}
```

```
return \View::make("draw")->with(array(
```

```
    'data' => $data,
```

```
));
```

```
}
```

```
}
```

```
if ($input['fac'] != 'Не учитывать') {
```

```
    $q_var = $input['fac'];
```

```
$dataX = Student::where('fac', '=', $q_var)->get();
```

```
if ($input['optionsRadios'] == 'gr' || $input['optionsRadios'] == 'prof' ||
```

```
$input['optionsRadios'] == 'napr' || $input['optionsRadios'] == 'region' ||
```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подпись | Дата | | 83 |

```
$input['optionsRadios'] == 'raion' || $input['optionsRadios'] == 'city' ||
$input['optionsRadios'] == 'pol') {
```

```
$vr = $input['optionsRadios'];
$subset = $dataX->map->only([$vr]);
```

```
$data = array();
foreach ($subset as $k) {
    $data[$k[$vr]] = 0;
}
```

```
foreach ($subset as $k) {
    $data[$k[$vr]]++;
}
```

```
if ($vr == 'pol') {
    $data['Мужской'] = $data[1];
    unset($data[1]);
    $data['Женский'] = $data[0];
    unset($data[0]);
}
return \View::make("draw")->with(array(
    'data' => $data,
));
}
```

```
if ($input['optionsRadios'] == 'godob' || $input['optionsRadios'] == 'semestr' ||
$input['optionsRadios'] == 'score') {
```

```
$vr = $input['optionsRadios'];
```

```
if ($vr == 'score') $vr = 'value';
```

```
$vr = 'scores.' . $vr;
```

```
$subset = \DB::table('students')
```

```
->join('scores', function ($join) use ($q_var) {
```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подпись | Дата | | 84 |

```

        $join->on('students.id', '=', 'scores.id_student')
        ->where('students.fac', '=', $q_var);
    })
    ->select($vr)
    ->get();

    $subset = json_decode($subset, true);

    $data = array();

    for ($i = 0; $i < sizeof($subset); $i++) {
        $data[$subset[$i][str_replace('scores.', '', $vr)]] = 0;
    }
    ksort($data);

    for ($i = 0; $i < sizeof($subset); $i++) {
        $data[$subset[$i][str_replace('scores.', '', $vr)]]++;
    }

    return \View::make("draw")->with(array(
        'data' => $data,
    ));
}

    $subset = json_decode($subset, true);

    $data = array();

    for ($i = 0; $i < sizeof($subset); $i++) {
        $data[$subset[$i][str_replace('scores.', '', $vr)]] = 0;
    }
    ksort($data);

    for ($i = 0; $i < sizeof($subset); $i++) {
        $data[$subset[$i][str_replace('scores.', '', $vr)]]++;
    }

```

```

    }

    foreach ($data as $k => $d) {
        $name = Discipline::find($k)->name;
        $data[$name] = $data[$k];
        unset($data[$k]);
    }
    return \View::make("draw")->with(array(
        'data' => $data,
    ));
}

```

```

    }
}

}
}

```

Шаблон модуля вывода данных

@extends('main')

<link rel="stylesheet" href="bower_components/bootstrap/dist/css/bootstrap.min.css">

<!-- Font Awesome -->

<link rel="stylesheet" href="bower_components/font-awesome/css/font-awesome.min.css">

<!-- Ionicons -->

<link rel="stylesheet" href="bower_components/Ionicons/css/ionicons.min.css">

<!-- Theme style -->

<link rel="stylesheet" href="css/AdminLTE.min.css">

<!-- AdminLTE Skins. Choose a skin from the css/skins

folder instead of downloading all of them to reduce the load. -->

<link rel="stylesheet" href="css/skins/_all-skins.min.css">

<!-- Morris chart -->

<link rel="stylesheet" href="bower_components/morris.js/morris.css">

<!-- jvectormap -->

<link rel="stylesheet" href="bower_components/jvectormap/jquery-jvectormap.css">

<!-- Date Picker -->

<link rel="stylesheet" href="bower_components/bootstrap-datepicker/dist/css/bootstrap-datepicker.min.css">

<!-- Daterange picker -->

<link rel="stylesheet" href="bower_components/bootstrap-daterangepicker/daterangepicker.css">

<!-- bootstrap wysihtml5 - text editor -->

<link rel="stylesheet" href="plugins/bootstrap-wysihtml5/bootstrap3-wysihtml5.min.css">

<!-- DataTables -->

<link rel="stylesheet" href="bower_components/datatables.net-bs/css/dataTables.bootstrap.min.css">

<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->

<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

<!-- Google Font -->

<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,700,300italic,400italic,600italic">

@section('content')

<div class="wrapper">

<header class="main-header">

<!-- Logo -->

<!-- Header Navbar: style can be found in header.less -->

<nav class="navbar navbar-static-top">

<!-- Sidebar toggle button-->

Навигация

</nav>

</header>

<!-- Left side column. contains the logo and sidebar -->

<aside class="main-sidebar">

<!-- sidebar: style can be found in sidebar.less -->

<section class="sidebar">

<!-- Sidebar user panel -->

<div class="user-panel">

</div>

<!-- sidebar menu: : style can be found in sidebar.less -->

<ul class="sidebar-menu" data-widget="tree">

<li class="header">ГЛАВНОЕ МЕНЮ

<i class="fa fa-book"></i> Главная

<li class="treeview">


```

        <i class="fa fa-table"></i> <span>Списки</span>
        <span class="pull-right-container">
            <i class="fa fa-angle-left pull-right"></i>
        </span>
    </a>
    <ul class="treeview-menu">
        <li><a href="#"><i class="fa fa-circle-o"></i>Преподаватели</a></li>
        <li><a href="#"><i class="fa fa-circle-o"></i>Студенты</a></li>
    </ul>
</li>
<li><a href="#"><i class="fa fa-pie-chart"></i> <span>Аналитика</span></a></li>
<li><a href="#"><i class="fa fa-book"></i> <span>Документация</span></a></li>
</ul>
</section>
<!-- /.sidebar -->
</aside>

<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <h1>
            Результаты
            <small>Панель сотрудника</small>
        </h1>
    </section>

    <!-- Main content -->
    <section class="content">
        <!-- Small boxes (Stat box) -->
        <div class="row">
            <div class="col-lg-3 col-xs-6">
                <!-- small box -->
                <div class="small-box bg-aqua">
                    <div class="inner">

```

<h3 style="font-size: 20px;">Преподаватели</h3>

<p>Данные о преподавателях</p>

</div>

<div class="icon">

<i class="ion ion-android-contacts"></i>

</div>

Перейти <i class="fa fa-arrow-circle-right"></i>

</div>

</div>

<!-- ./col -->

<div class="col-lg-3 col-xs-6">

<!-- small box -->

<div class="small-box bg-green">

<div class="inner">

<h3 style="font-size: 20px;">Аналитика</h3>

<p>Модуль аналитики</p>

</div>

<div class="icon">

<i class="ion ion-stats-bars"></i>

</div>

Перейти <i class="fa fa-arrow-circle-right"></i>

</div>

</div>

<!-- ./col -->

<div class="col-lg-3 col-xs-6">

<!-- small box -->

<div class="small-box bg-yellow">

<div class="inner">

<h3 style="font-size: 20px;">Студенты</h3>

<p>Данные о студентах</p>

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | 90 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

</div>
<div class="icon">
  <i class="ion ion-person-add"></i>
</div>
<a href="#" class="small-box-footer">Перейти <i class="fa fa-arrow-circle-
right"></i></a>
</div>
</div>
<!-- ./col -->
<div class="col-lg-3 col-xs-6">
  <!-- small box -->
  <div class="small-box bg-red">
    <div class="inner">
      <h3 style="font-size: 20px;">Документация</h3>

      <p>Справочные данные</p>
    </div>
    <div class="icon">
      <i class="ion ion-help"></i>
    </div>
    <a href="#" class="small-box-footer">Перейти <i class="fa fa-arrow-circle-
right"></i></a>
  </div>
</div>
<!-- ./col -->
</div>
<!-- /.row -->
<!-- Main row -->
<div class="row">
</div>
<!-- /.row (main row) -->

</section>

<section class="content">

```

```

<div class="box box-warning">
  <div class="box-header with-border">
    <h3 class="box-title">Результаты</h3>
  </div>
  <!-- /.box-header -->

```

```

<div class="row">
  <div class="col-md-6">
    <!-- DONUT CHART -->
    <div class="box box-danger">
      <div class="box-header with-border">
        <h3 class="box-title">Круговая диаграмма</h3>

        <div class="box-tools pull-right">
          <button type="button" class="btn btn-box-tool" data-widget="collapse"><i class="fa fa-minus"></i></button>
          <button type="button" class="btn btn-box-tool" data-widget="remove"><i class="fa fa-times"></i></button>
        </div>
      </div>
      <div class="box-body chart-responsive">
        <div class="chart" id="sales-chart" style="height: 300px; position: relative;"></div>
      </div>
    <!-- /.box-body -->
  </div>
  <!-- /.box -->
</div>

<!-- /.col (LEFT) -->
<div class="col-md-6">
  <!-- BAR CHART -->

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | 92 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

<div class="box box-success">
  <div class="box-header with-border">
    <h3 class="box-title">Гистограмма</h3>

    <div class="box-tools pull-right">
      <button type="button" class="btn btn-box-tool" data-widget="collapse"><i class="fa
fa-minus"></i>
    </button>
      <button type="button" class="btn btn-box-tool" data-widget="remove"><i class="fa
fa-times"></i></button>
    </div>
  </div>
  <div class="box-body chart-responsive">
    <div class="chart" id="bar-chart" style="height: 300px;"></div>
  </div>
<!-- /.box-body -->
</div>
<!-- /.box -->

</div>
<!-- /.col (RIGHT) -->
</div>
</section>

<!-- /.content -->
</div>
<!-- /.content-wrapper -->
<footer class="main-footer">
  <div class="pull-right hidden-xs">
    <b>2019</b>
  </div>
  <strong></strong>
</footer>

<!-- Control Sidebar -->

```

```

<aside class="control-sidebar control-sidebar-dark">
  <!-- Create the tabs -->
  <ul class="nav nav-tabs nav-justified control-sidebar-tabs">
    <li><a href="#control-sidebar-home-tab" data-toggle="tab"><i class="fa fa-home"></i></a></li>
    <li><a href="#control-sidebar-settings-tab" data-toggle="tab"><i class="fa fa-gears"></i></a></li>
  </ul>
  <!-- Tab panes -->
  <div class="tab-content">
    <!-- Home tab content -->
    <div class="tab-pane" id="control-sidebar-home-tab">
      <h3 class="control-sidebar-heading">Recent Activity</h3>

      </div>
    <!-- /.tab-pane -->
    <!-- Stats tab content -->
    <div class="tab-pane" id="control-sidebar-stats-tab">Stats Tab Content</div>
    <!-- /.tab-pane -->
    <!-- Settings tab content -->
    <div class="tab-pane" id="control-sidebar-settings-tab">
      <form method="post">
        <h3 class="control-sidebar-heading">General Settings</h3>

        <div class="form-group">
          <label class="control-sidebar-subheading">
            Report panel usage
            <input type="checkbox" class="pull-right" checked>
          </label>

          <p>
            Some information about this general settings option
          </p>

```

```

</div>

<!-- /.form-group -->

<div class="form-group">
  <label class="control-sidebar-subheading">
    Allow mail redirect
    <input type="checkbox" class="pull-right" checked>
  </label>

  <p>
    Other sets of options are available
  </p>
</div>

<!-- /.form-group -->

<div class="form-group">
  <label class="control-sidebar-subheading">
    Expose author name in posts
    <input type="checkbox" class="pull-right" checked>
  </label>

  <p>
    Allow the user to show his name in blog posts
  </p>
</div>

<!-- /.form-group -->

<h3 class="control-sidebar-heading">Chat Settings</h3>

<div class="form-group">
  <label class="control-sidebar-subheading">
    Show me as online
    <input type="checkbox" class="pull-right" checked>
  </label>
</div>

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подпись | Дата | | 95 |

<!-- /.form-group -->

<div class="form-group">

<label class="control-sidebar-subheading">

Turn off notifications

<input type="checkbox" class="pull-right">

</label>

</div>

<!-- /.form-group -->

<div class="form-group">

<label class="control-sidebar-subheading">

Delete chat history

<i class="fa fa-trash-o"></i>

</label>

</div>

<!-- /.form-group -->

</form>

</div>

<!-- /.tab-pane -->

</div>

</aside>

<!-- /.control-sidebar -->

<!-- Add the sidebar's background. This div must be placed

immediately after the control sidebar -->

<div class="control-sidebar-bg"></div>

</div>

<!-- ./wrapper -->

<!-- jQuery 3 -->

<script src="bower_components/jquery/dist/jquery.min.js"></script>

<!-- jQuery UI 1.11.4 -->

<script src="bower_components/jquery-ui/jquery-ui.min.js"></script>

<!-- Resolve conflict in jQuery UI tooltip with Bootstrap tooltip -->

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | 96 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

<script>
    $.widget.bridge('uibutton', $.ui.button);
</script>
<!-- Bootstrap 3.3.7 -->
<script src="bower_components/bootstrap/dist/js/bootstrap.min.js"></script>
<!-- Morris.js charts -->
<script src="bower_components/raphael/raphael.min.js"></script>
<script src="bower_components/morris.js/morris.min.js"></script>
<!-- Sparkline -->
<script src="bower_components/jquery-sparkline/dist/jquery.sparkline.min.js"></script>
<!-- jvectormap -->
<script src="plugins/jvectormap/jquery-jvectormap-1.2.2.min.js"></script>
<script src="plugins/jvectormap/jquery-jvectormap-world-mill-en.js"></script>
<!-- jQuery Knob Chart -->
<script src="bower_components/jquery-knob/dist/jquery.knob.min.js"></script>
<!-- daterangepicker -->
<script src="bower_components/moment/min/moment.min.js"></script>
<script src="bower_components/bootstrap-daterangepicker/daterangepicker.js"></script>
<script src="bower_components/datatables.net/js/jquery.dataTables.min.js"></script>
<script src="bower_components/datatables.net-bs/js/dataTables.bootstrap.min.js"></script>
<!-- datepicker -->
<script
                                src="bower_components/bootstrap-datepicker/dist/js/bootstrap-
                                datepicker.min.js"></script>
<!-- Bootstrap WYSIHTML5 -->
<script src="plugins/bootstrap-wysihtml5/bootstrap3-wysihtml5.all.min.js"></script>
<!-- Slimscroll -->
<script src="bower_components/jquery-slimscroll/jquery.slimscroll.min.js"></script>
<!-- FastClick -->
<script src="bower_components/fastclick/lib/fastclick.js"></script>
<!-- AdminLTE App -->
<script src="js/adminlte.min.js"></script>
<!-- AdminLTE dashboard demo (This is only for demo purposes) -->
<script src="js/pages/dashboard.js"></script>
<!-- AdminLTE for demo purposes -->
<script src="js/demo.js"></script>

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГЧ 1.090501.01.001 ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 97 |

```

<script>
$(function () {
    "use strict";

    //DONUT CHART
    var donut = new Morris.Donut({
        element: 'sales-chart',
        resize: true,
        colors: ["#3c8dbc", "#f56954", "#00a65a"],
        data: [
            @foreach ($data as $key => $d)
                { label: '{{ $key }}', value: {{ $d }} },
            @endforeach
        ],
        hideHover: 'auto'
    });

    //BAR CHART
    var bar = new Morris.Bar({
        element: 'bar-chart',
        resize: true,
        data: [
            @foreach ($data as $key => $d)
                { y: '{{ $key }}', a: {{ $d }} },
            @endforeach
        ],
        barColors: ['#00a65a', '#f56954'],
        xkey: 'y',
        ykeys: ['a'],
        labels: ['студентов'],
        hideHover: 'auto'
    });
});
</script>
@endsection

```

ПРИЛОЖЕНИЕ Б

(справочное)

Руководство оператора

1 Назначение программы

Данное программное обеспечение аналитической системы поддержки управления личным составом обеспечивает:

- аутентификацию сотрудников в системе;
- получение сведений о преподавателях;
- получение сведений о студентах;
- установку фильтров аналитики;
- проведение аналитики;
- вывод результатов аналитики на экран.

2 Условия выполнения программы

Программное обеспечение предназначено для запуска на персональных компьютерах с процессором не ниже Intel Core 2 Duo под управлением операционных систем Window, Linux, OS X. На компьютере должен быть установлен браузер Google Chrome версии 26 или выше, Firefox версии 19 или выше. Серверная часть должна быть установлена на ЭВМ с установленной на ней ОС Linux Ubuntu версии не ниже 16.04.

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подпись | Дата | | 99 |

3 Выполнение программы

Для начала работы программы следует открыть браузер и в адресной строке ввести адрес программного обеспечения - <http://192.168.1.231>.

В верхней и левой части окна находится главное меню приложения (рисунок Б.1). Выбор команд меню задает выполнение необходимых действий. Под главным меню на главной странице расположен текст описания программы.

Главное меню состоит из следующих пунктов:

- «Главная» - переход на главную страницу;
- «Списки» - открывает список преподавателей или студентов;
- «Аналитика» - загружает модуль аналитики;
- «Документация» - предлагает на скачивание файл документации к ПО.

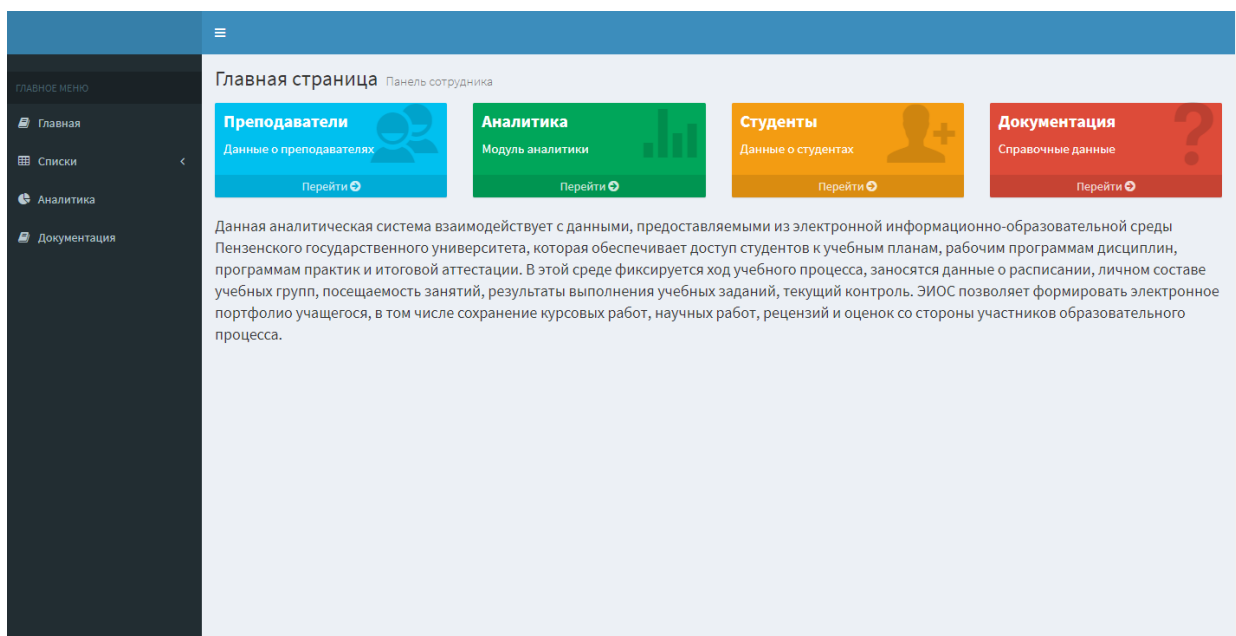


Рисунок Б.1 – Главная страница приложения

На странице «Преподаватели» (рисунок Б.2) на экран пользователя выводится список преподавателей, зарегистрированных в системе.

| Фамилия | Имя | Отчество | Пол | Дата рождения |
|------------|----------|------------|---------|---------------|
| Давыдов | Тимур | Дмитриевич | Мужской | 14/12/1980 |
| Маслов | Болеслав | Иванович | Мужской | 30/01/1959 |
| Фёдоров | Гордей | Андреевич | Мужской | 11/01/1980 |
| Пономарёва | Нонна | Максимовна | Женский | 15/07/1972 |
| Терентьев | Тимур | Евгеньевич | Мужской | 19/07/1986 |
| Юдин | Лев | Иванович | Мужской | 26/09/1970 |
| Соболев | Марат | Дмитриевич | Мужской | 02/06/1957 |
| Хохлова | Зинаида | Дмитриевна | Женский | 12/04/1996 |

Рисунок Б.2 – Список преподавателей

На странице «Студенты» (рисунок Б.3) на экран пользователя выводится список студентов, зарегистрированных в системе.

| Фамилия | Имя | Отчество | Пол | Дата рождения | Группа | Факультет | Профиль подготовки | Направление подготовки | Регион |
|--------------|---------|--------------|---------|---------------|--------|-----------------------------------|--|---|-------------|
| Абдрахманова | Альбина | Рафаэлевна | Женский | 22/09/1998 | 15ипо1 | Историко-филологический факультет | Дошкольное образование | 44.03.01 Педагогическое образование | |
| Абсаямова | Дмитрий | Сергеевич | Мужской | 24/10/1997 | 15юю3 | Юридический факультет | Хирургия 14.01.17 | 31.06.01 Клиническая медицина | Пензенс обл |
| Аверкин | Сергей | Вячеславович | Мужской | 16/04/1997 | 15ипи1 | Историко-филологический факультет | Теория и методика профессионального образования 13.00.08 | 44.06.01 Образование и педагогические науки | Пензенс обл |

Рисунок Б.3 – Список студентов

Главная страница модуля «Аналитика» содержит пункты выбора фильтров (рисунок Б.3). Пользователю предоставляется выбрать два набора данных для сравнения. Для создания графического отображения необходимо нажать кнопку «Применить» (Рисунок Б.4).

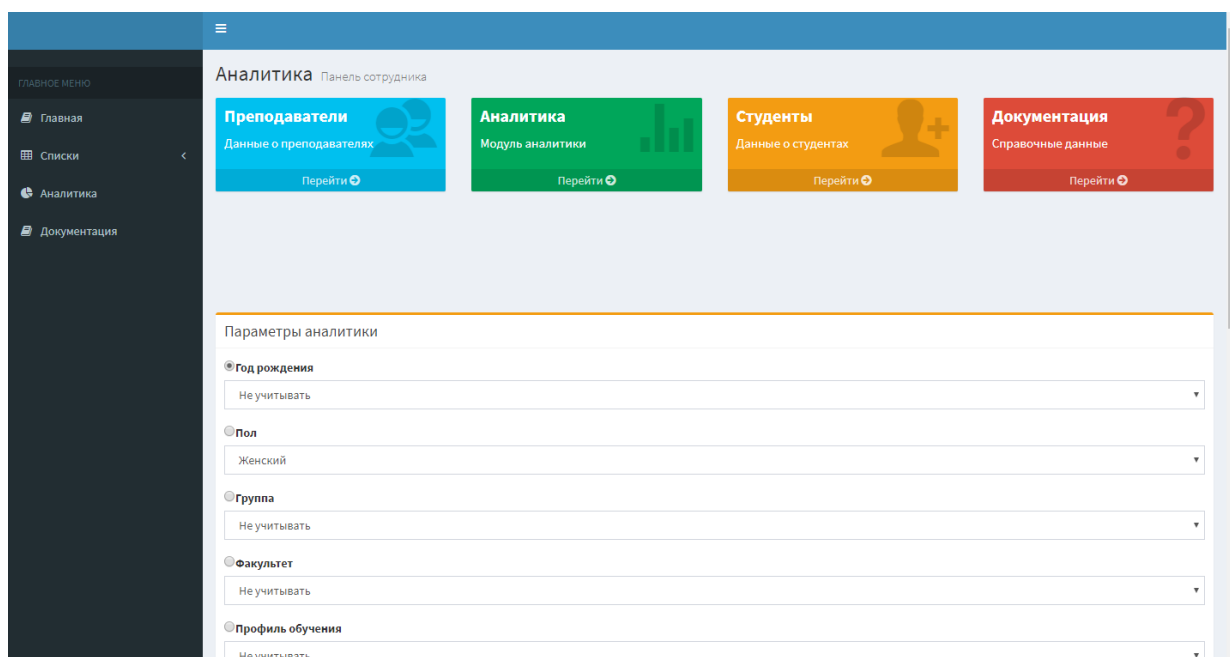


Рисунок Б.3 – Главное окно модуля аналитики

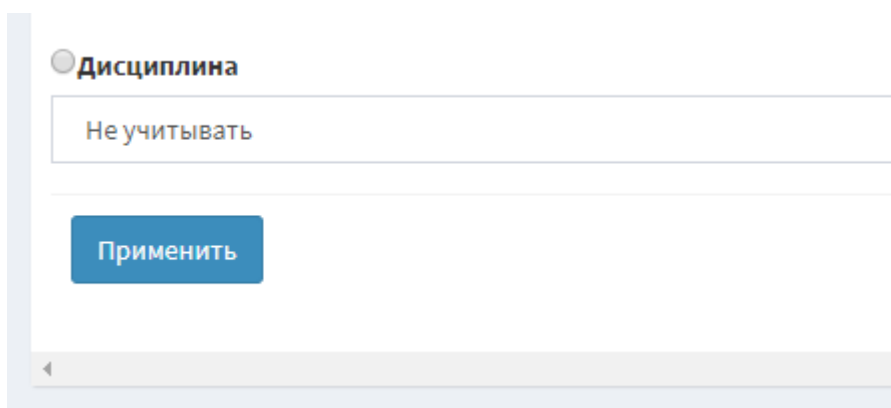


Рисунок Б.4 – Кнопка «Применить»

Результаты работы модуля представлены на странице отображения результатов в виде круговой диаграммы и гистограммы (Рисунок Б.5).

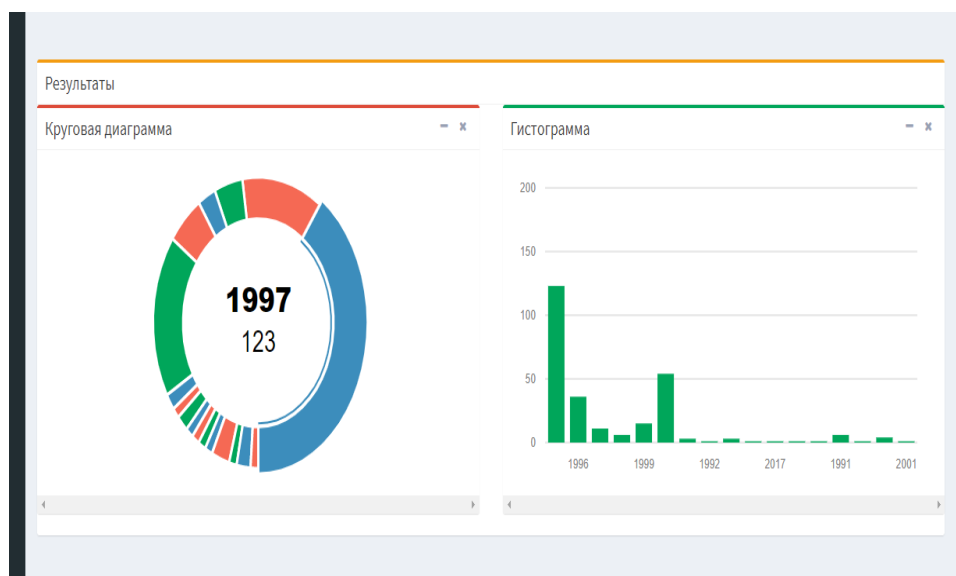


Рисунок Б.5 – Результат работы модуля

Для работы в системе пользователь должен быть авторизован (Рисунок Б.6).

The screenshot shows a login form titled "Login". It contains two input fields: "E-Mail Address" and "Password". Below these fields is a checkbox labeled "Remember Me". At the bottom of the form is a blue button labeled "Login".

Рисунок Б.6 – Форма авторизации

ПРИЛОЖЕНИЕ В

(обязательное)

Протокол тестирования

1.1 Корректная авторизация в приложении

В ходе тестирования в форму входа (рисунок В.1) были введены корректные данные для входа: email – admin@gmail.com, пароль – admin. Данные введены верно, пользователь был перенаправлен на главную страницу (рисунок В.2).

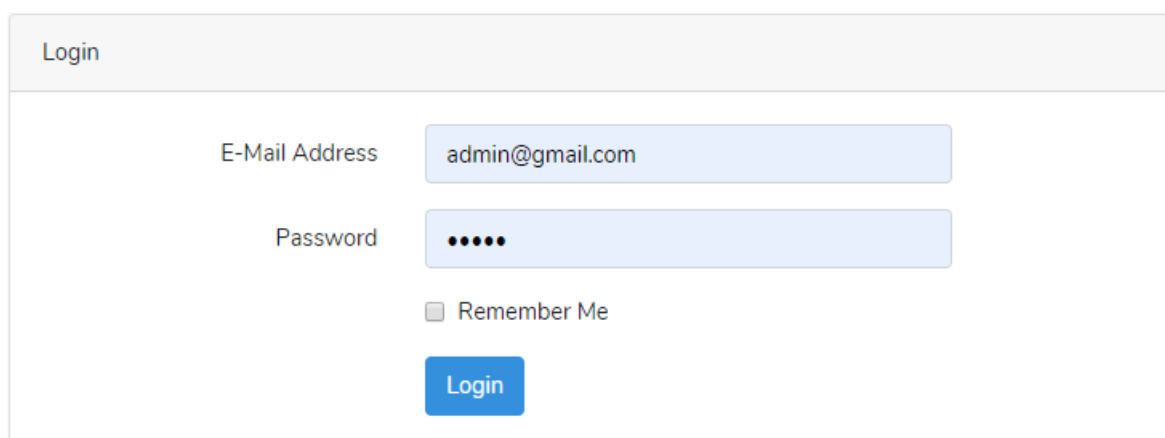


Рисунок В.1 – Форма входа

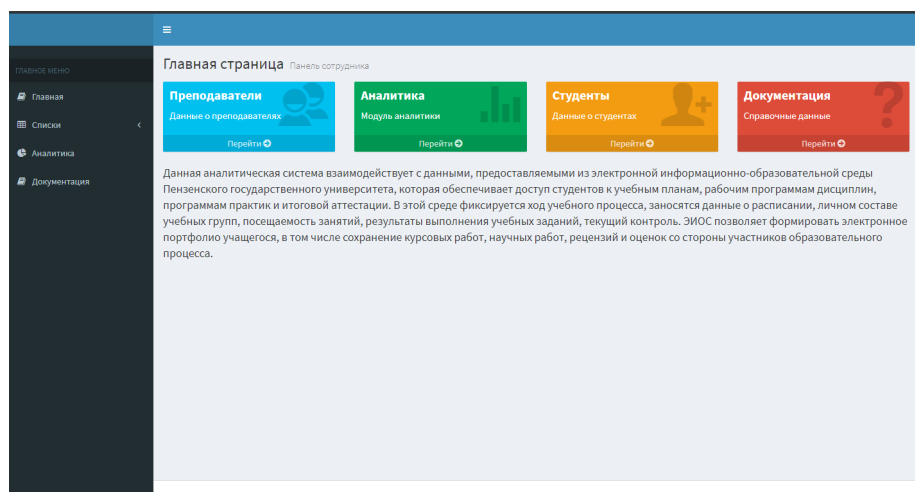


Рисунок В.2 – Главная страница

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | 104 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

1.2 Некорректная авторизация в приложении

В ходе тестирования в форму входа (рисунок В.3) были введены корректные данные для входа: email – admin1@gmail.com, пароль – admin1. Данные введены неверно, пользователь получил сообщение об ошибке (рисунок В.4).

The screenshot shows a login form titled "Login". It contains two input fields: "E-Mail Address" with the value "admin1@gmail.com" and "Password" with masked characters "•••••". Below the password field is a checkbox labeled "Remember Me" which is unchecked. At the bottom is a blue "Login" button.

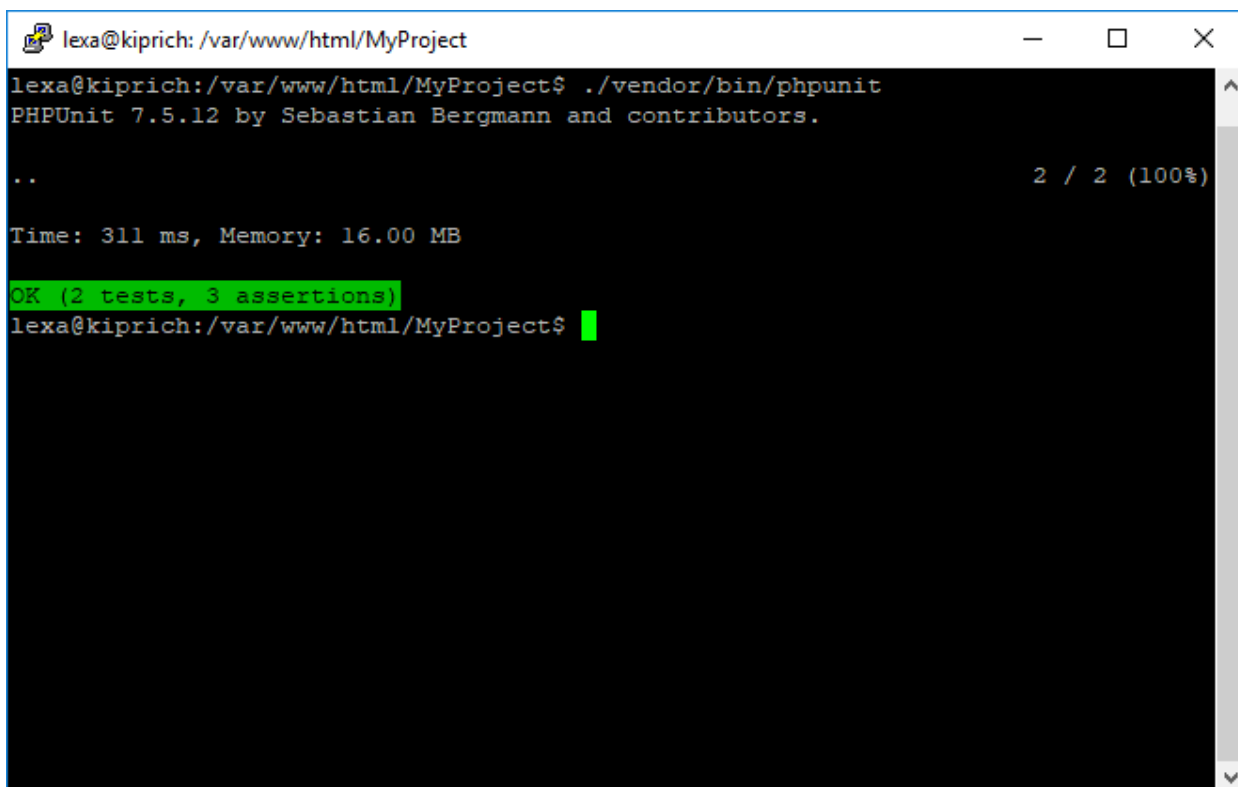
Рисунок В.3 – Форма входа

The screenshot shows the same login form as Figure B.3, but with an error message. The "E-Mail Address" field, which contains "admin1@gmail.com", is highlighted with a red border. Below the field, the text "These credentials do not match our records." is displayed in red. The "Password" field is empty, the "Remember Me" checkbox is unchecked, and the "Login" button is still present.

Рисунок В.4 – Ошибка неверного ввода данных

1.3 Тестирование доступа к форме авторизации

При помощи автоматизированного теста был проверен доступ к форме авторизации (рисунок В.5).



```
lexa@kiprich: /var/www/html/MyProject
lexa@kiprich:/var/www/html/MyProject$ ./vendor/bin/phpunit
PHPUnit 7.5.12 by Sebastian Bergmann and contributors.

..                                                                2 / 2 (100%)

Time: 311 ms, Memory: 16.00 MB
OK (2 tests, 3 assertions)
lexa@kiprich:/var/www/html/MyProject$
```

Рисунок В.5 – Тестирование доступа к форме авторизации

1.4 Тестирование перенаправления авторизованного пользователя

При помощи автоматизированного теста было проверено наличие переадресации пользователя со страницы авторизации на главную, если он уже авторизован в системе (рисунок В.6).

```
lexa@kiprich: /var/www/html/MyProject
lexa@kiprich:/var/www/html/MyProject$ ./vendor/bin/phpunit
PHPUnit 7.5.12 by Sebastian Bergmann and contributors.

... 3 / 3 (100%)

Time: 380 ms, Memory: 18.00 MB

OK (3 tests, 5 assertions)
lexa@kiprich:/var/www/html/MyProject$
```

Рисунок В.6 – Тестирование переадресации

1.5 Тестирование входа с некорректным паролем

При помощи автоматизированного теста была осуществлена авторизации в системе с неверным паролем (рисунок В.7).

```
lexa@kiprich: /var/www/html/MyProject
lexa@kiprich:/var/www/html/MyProject$ ./vendor/bin/phpunit
PHPUnit 7.5.12 by Sebastian Bergmann and contributors.

.... 4 / 4 (100%)

Time: 446 ms, Memory: 20.00 MB

OK (4 tests, 12 assertions)
lexa@kiprich:/var/www/html/MyProject$
```

Рисунок В.7 – Авторизация с неверным паролем

1.6 Тестирование модуля аналитики. Ввод 2 параметров

Для проведения тестирования были заданы два параметра: «Год рождения» – 1999 и «Факультет» - для сортировки (рисунок В.8).

Параметры аналитики

☐ Год рождения

1999

☐ Пол

Не учитывать

☐ Группа

Не учитывать

☒ Факультет

Не учитывать

Рисунок В.8 – Ввод фильтров аналитики

После нажатия кнопки «Применить» результат аналитики был выведен на экран (рисунок В.9).

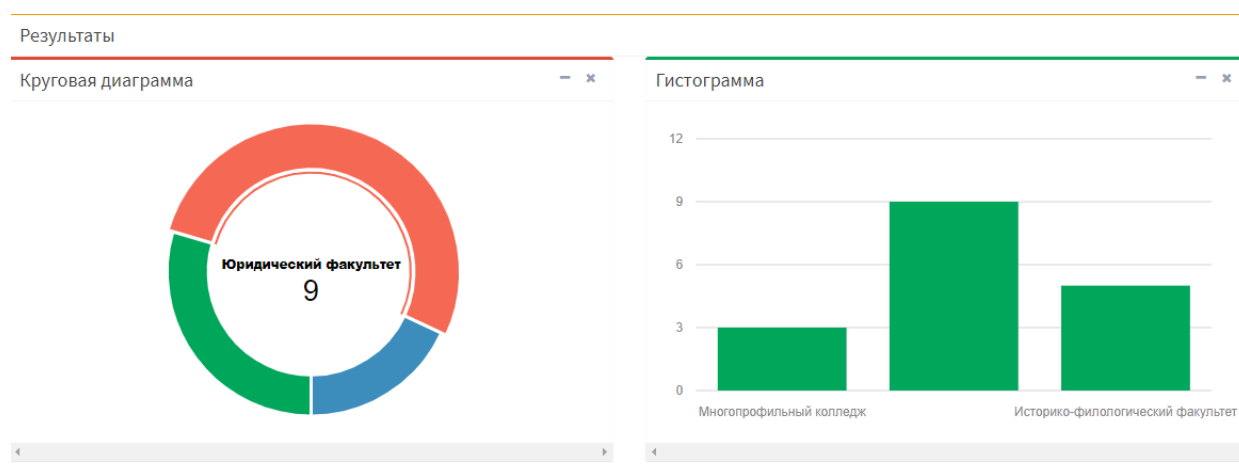


Рисунок В.9 – Результат аналитики

1.7 Тестирование модуля аналитики. Ввод 1 параметра

Для проведения тестирования был задан параметр «Год рождения» - для сортировки (рисунок В.10).

Параметры аналитики

☒ Год рождения

Не учитывать

☐ Пол

Не учитывать

☐ Группа

Рисунок В.10 – Выбор фильтров аналитики

После нажатия кнопки «Применить» приложение вывело сообщение об ошибке (рисунок В.11).

Ошибка при выборе параметров аналитики!

Выберите параметр.

[Назад](#)

Рисунок В.11 – Окно ошибки

1.8 Тестирование модуля аналитики. Ввод 3 параметров

Для проведения тестирования были заданы два параметра: «Год рождения» – 1999, «Пол» - мужской и «Факультет» - для сортировки (рисунок В.12).

Параметры аналитики

Год рождения

1997

Пол

Мужской

Группа

Не учитывать

Факультет

Не учитывать

Рисунок В.12 – Ввод 3 параметров аналитики

После нажатия кнопки «Применить» приложение вывело сообщение об ошибке (рисунок В.13).

Ошибка при выборе параметров аналитики!

Выберите не более двух параметров.

Назад

Рисунок В.13 – Сообщение об ошибке

1.9 Тестирование модуля аналитики. Ввод некорректных параметров

Для проведения тестирования были заданы два параметра: «Дисциплина» – Математика и «Регион» - для сортировки (рисунок В.14).

☒ **Регион**

Не учитывать

☐ **Район**

Не учитывать

☐ **Город**

Не учитывать

☐ **Год обучения**

Не учитывать

☐ **Семестр**

Не учитывать

☐ **Оценка**

Не учитывать

☐ **Дисциплина**

Математика

Рисунок В.14 – Ввод параметров аналитики

После нажатия кнопки «Применить» приложение вывело сообщение об ошибке (рисунок В.15).

Ошибка при выборе параметров аналитики!

Параметр задан неверно.
[Назад](#)

Рисунок В.15 – Сообщение об ошибке

ПРИЛОЖЕНИЕ Г

(обязательное)

Презентация

ФГБОУ ВО «ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА СПЕЦИАЛИСТА
на тему

Аналитическая система поддержки управления личным составом



Пенза, 2019

Выполнил:
студент 5-го курса гр. 14ВО2
Вахромеев Максим Александрович

Руководитель:
д.т.н., зав. каф. ВТ
Митрохин Максим Александрович

Рисунок Г.1 – Слайд 1

Разработка аналитической системы

Необходима аналитическая система для поддержки управления личным составом, способная увеличить эффективность работы сотрудников

Функции системы:

- учет данных о преподавателях и студентах
- учет данных об успеваемости
- составление аналитических запросов
- графическое отображение результатов аналитики

Рисунок Г.2 – Слайд 2

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 112 |

Постановка задачи на разработку

Клиентское приложение:

- модуль авторизации
- просмотр списка студентов и преподавателей
- форма создание запроса с указанием фильтров
- вывод результатов выполнения запроса

Серверная часть:

- обработка запросов к базе данных
- разграничение прав доступа
- выполнение аналитических задач

Рисунок Г.3 – Слайд 3

Выбор средств и платформы разработки

| Характеристика | Средства и платформы разработки | | | |
|---------------------------------|---------------------------------|-------------------|-------------------|-------------------|
| | PHP 5.6 | PHP 7 | PHP 7 и Yii1 | PHP 7 и Laravel |
| Принцип обработки кода | «Интерпретатор» | «Интерпретатор» | «Интерпретатор» | «Интерпретатор» |
| Язык | PHP | PHP | PHP | PHP |
| Средства доступа к базам данных | SQL, <u>noSQL</u> | SQL, <u>noSQL</u> | SQL, <u>noSQL</u> | SQL, <u>noSQL</u> |
| Платный | Нет | Нет | Нет | Нет |

Рисунок Г.4 – Слайд 4

Средства разработки

- Язык программирования PHP 7
- СУБД MySQL ([mariadb](#))
- Фреймворк Laravel

Все используемые средства являются продуктами с открытым исходным кодом, бесплатны для коммерческого использования, имеют активные сообщества пользователей



Рисунок Г.5 – Слайд 5

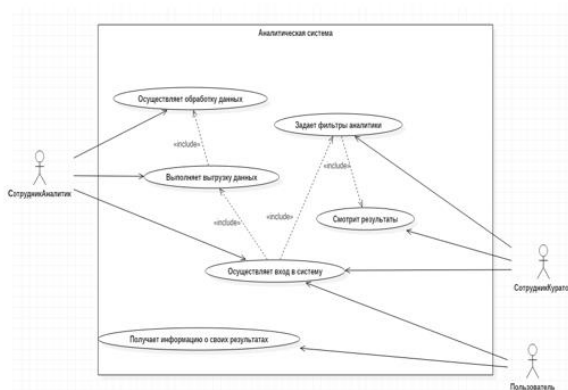


Рисунок Г.6 – Слайд 6

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090501.01.001 ПЗ

Лист

114

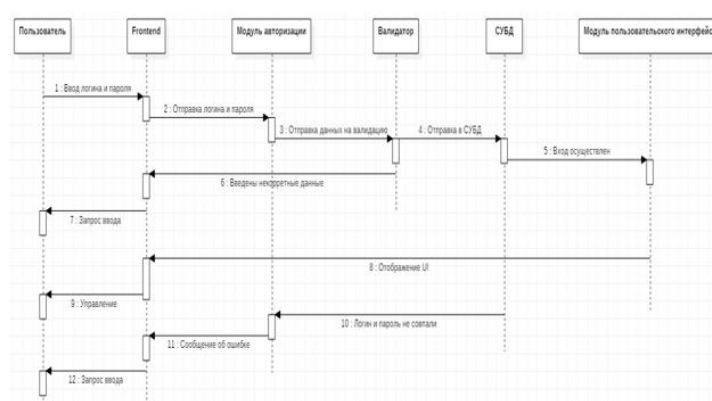


Рисунок Г.7 – Слайд 7



Рисунок Г.8 – Слайд 8



Рисунок Г.9 – Слайд 9

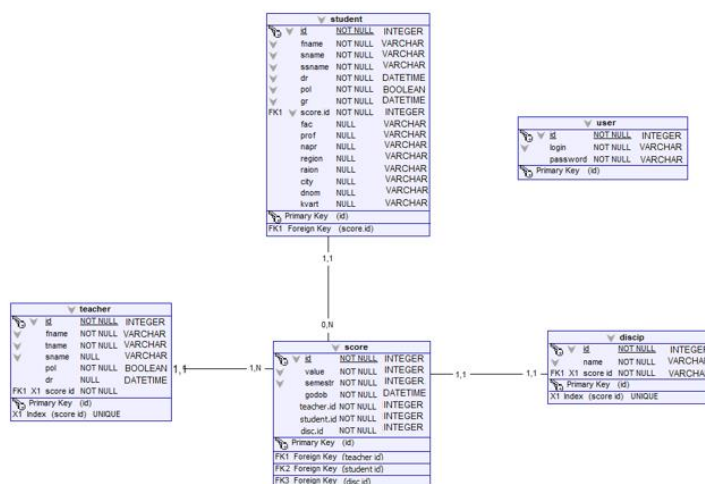
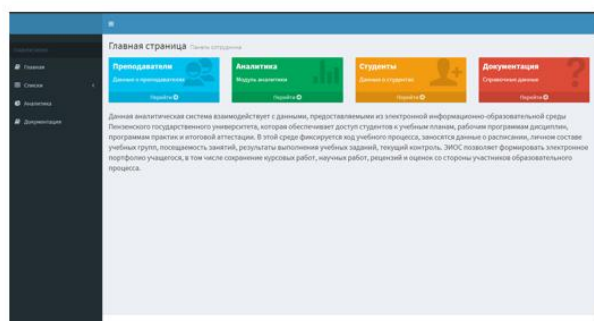


Рисунок Г.10 – Слайд 10

Интерфейс клиентского приложения



Главная страница

Login

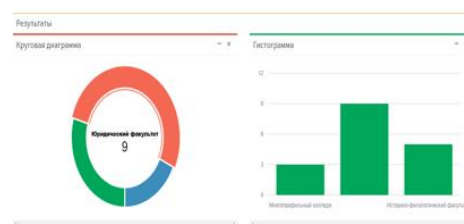
E-Mail Address: admin@gmail.com

Password: *****

☐ Remember Me

Login

Форма входа



Результаты аналитики

Рисунок Г.11 – Слайд 11

Фильтры аналитики

Параметры аналитики

☐ Год рождения

1999

☐ Пол

Не учитывать

☐ Группа

Не учитывать

☐ Факультет

Не учитывать

Ввод фильтров аналитики

Ошибка при выборе параметров аналитики!

Параметр задан неверно.
[Назад](#)

Сообщение об ошибке

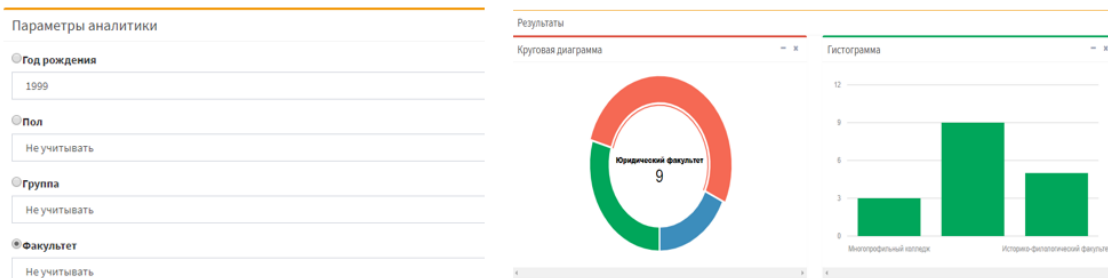
Рисунок Г.12 – Слайд 12

Тестирование

- Авторизация пользователя системы
- Ввод фильтров аналитики
- Unit-тестирование в Laravel

Рисунок Г.13 – Слайд 13

Тестирование

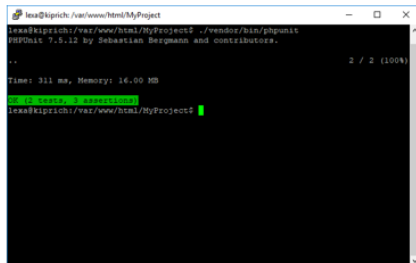


Ввод фильтров

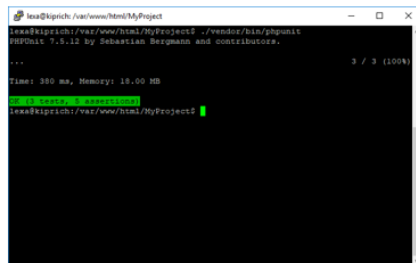
Результат аналитики

Рисунок Г.14 – Слайд 14

Тестирование



Тестирование формы авторизации



Тестирование переадресации

Рисунок Г.15 – Слайд 15

Выводы

- Проведен анализ предметной области
- Спроектирована аналитическая система с использованием методологии UML 2.0
- Разработана система с использованием средств с открытым исходным кодом
- Разработана математическая модель модуля аналитики, этот функционал добавлен в аналитическую систему
- Разработанная система протестирована, весь реализованный функционал работает корректно
- Получившийся интерфейс понятен и не требует высокой квалификации персонала
- Разработанная система проста во внедрении, в том числе в существующую инфраструктуру

Рисунок Г.16 – Слайд 16

Спасибо за внимание!

Рисунок Г.17 – Слайд 17

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.01.001 ПЗ | Лист |
| | | | | | | 120 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

ПРИЛОЖЕНИЕ Д
(обязательное)

Графическая часть

В данном приложении представлены разработанные UML-диаграммы.

| | | | | | | |
|-------------|-------------|-----------------|----------------|-------------|-------------------------------|-------------|
| | | | | | <i>ПГУ 1.090501.01.001 ПЗ</i> | <i>Лист</i> |
| | | | | | | |
| <i>Изм.</i> | <i>Лист</i> | <i>№ докум.</i> | <i>Подпись</i> | <i>Дата</i> | | <i>121</i> |