

Министерство науки и высшего образования Российской Федерации  
ФГБОУ ВО РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ  
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ  
(РГГМУ)

Институт Информационных систем и геотехнологий  
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

БАКАЛАВРСКАЯ РАБОТА

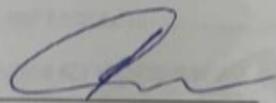
На тему “Автоматизация процесса выдачи направления в деканате”

Исполнитель Косабуко Владислав Игоревич

Руководитель кандидат технических наук, доцент – Колбина Ольга Николаевна

«К защите допускаю»

и.о. Заведующий кафедрой



(подпись)

К.Т.И.

(ученая степень, ученое звание)

Колбина Ольга Николаевна

(фамилия, имя, отчество)

«01» 06 2023 г.

Санкт-Петербург

2023г.

## Содержание

<b>ВВЕДЕНИЕ.....</b>	<b>4</b>
<b>Глава 1. Аналитическая часть .....</b>	<b>6</b>
<b>1.1 Анализ предметной области.....</b>	<b>6</b>
<b>1.2 Целевая аудитория.....</b>	<b>8</b>
<b>1.3 Техничко-экономическое обоснование .....</b>	<b>8</b>
<b>1.4 Сроки реализации проекта .....</b>	<b>13</b>
<b>1.5 Функциональные требования .....</b>	<b>15</b>
<b>1.6 Нефункциональные требования .....</b>	<b>15</b>
<b>1.7 Подходы SWOT, ISA, BPR, USM.....</b>	<b>16</b>
<b>1.7.1 Подход SWOT.....</b>	<b>16</b>
<b>1.7.2 Подход ISA .....</b>	<b>17</b>
<b>1.7.3 Подход BPR.....</b>	<b>18</b>
<b>1.7.4 Подход VSM.....</b>	<b>19</b>
<b>1.8 Техническое задание .....</b>	<b>20</b>
<b>Глава 2. Моделирование системы. ....</b>	<b>21</b>
<b>2.1 Выбор методологии проектирования .....</b>	<b>21</b>
<b>2.2 Моделирование предметной области .....</b>	<b>23</b>
<b>2.2.1 Диаграмма прецедентов.....</b>	<b>23</b>
<b>2.2.2 Диаграммы последовательностей .....</b>	<b>27</b>
<b>2.2.3 Диаграмма компонентов .....</b>	<b>31</b>
<b>2.2.4 Диаграмма развертывания .....</b>	<b>32</b>
<b>2.3 Схема базы данных .....</b>	<b>33</b>
<b>Глава 3. Реализация системы .....</b>	<b>36</b>
<b>3.1 Интерфейс системы .....</b>	<b>36</b>
<b>3.1.1 Графический интерфейс системы .....</b>	<b>36</b>
<b>3.1.2 Программный интерфейс системы .....</b>	<b>37</b>
<b>3.1.3 Аппаратный интерфейс системы.....</b>	<b>39</b>
<b>3.2 Окно Авторизации .....</b>	<b>39</b>
<b>3.3 Админское меню .....</b>	<b>40</b>

<b>3.4 Главное меню .....</b>	<b>42</b>
<b>3.5 Окно успеваемости.....</b>	<b>43</b>
<b>3.6 Окно печати ведомостей .....</b>	<b>44</b>
<b>3.7 Расчет надежности .....</b>	<b>46</b>
<b>3.7.1 Расчет надежности аппаратной части .....</b>	<b>46</b>
<b>3.7.2 Расчет надежности программного обеспечения .....</b>	<b>49</b>
<b>3.7.3 Расчет надежности системы в целом .....</b>	<b>50</b>
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>51</b>
<b>Список литературы и источников.....</b>	<b>54</b>
<b>Приложение А. Техническое задание на разработку информационной системы “Автоматизация процесса выдачи направления в деканате” ..</b>	<b>57</b>
<b>Приложение Б. Программный код excel_parser.py .....</b>	<b>63</b>

## ВВЕДЕНИЕ

Образование играет важную роль в жизни каждого человека, это система воспитания и обучения личности, а также совокупность знаний, умений и навыков. С начала XX века образование становилось все доступнее и доступнее, в наше время образование получает почти каждый человек в развитых странах. Сам процесс образования с годами менялся и усложнялся, и сейчас представляет собой сложный структурированный процесс, который как и многие другие процессы в нашей жизни требуют и подвергаются автоматизации.

Автоматизация процессов в образовании является одним из важнейших направлений развития современной информационной технологии. В современных условиях крайне важно обеспечить эффективное функционирование системы контроля и учета обучающихся, особенно в части выдачи ведомостей на сдачу задолженности и отслеживания успеваемости. Однако, традиционная ручная версия решения этой задачи требует большого количества времени и сил со стороны работников деканата, а также может стать причиной ошибок в данных и необходимости корректировки. В связи с этим, автоматизация выше перечисленных действий является актуальной задачей, которая будет рассмотрена в ходе практики и в ходе работы над дипломным проектом. В работе будет спроектирован проект и создана рабочая наглядная часть продукта автоматизации данного процесса.

**Объектом** исследования является взаимодействие студентов с деканатом.

**Предметом** исследования является вопрос автоматизации получения ведомости и отслеживания успеваемости.

Цель проекта: Автоматизировать процесс получения ведомости в деканате и отслеживания успеваемости для условного терминала, которое позволит студентам отслеживать свою успеваемость, а также оперативно получать ведомости на сдачу задолженностей во время дополнительных сессий.

Задачи проекта:

- Анализ предметной области
- Сроки реализации проекта и оценка материальных затрат
- Моделирование предметной области
- Реализация ИС

Методы проведения работы:

- Моделирование UML
- Проектирование баз данных
- Проектирование программы

Используемый инструментарий:

- Пакет Microsoft Office
- Draw.io
- SQLite
- Python
- Библиотеки Pandas, openpyxl, PyPDF2, pywin32, PyQt6
- QtDesigner
- PyCharm
- DBeaver

## **Глава 1. Аналитическая часть**

### **1.1 Анализ предметной области**

Предметная область – информационные системы в образовательном процессе.

Образование является важной частью жизни каждого человека, и как любой процесс нуждается в автоматизации, для достижения больших результатов, ускорения процесса и лучшего функционирования. Многие вещи уже были автоматизированы, например учет студентов, составление расписаний, дистанционное обучение и многое другое. Но все равно многие процессы до сих пор остаются не автоматизированными, например взаимодействие между студентами и деканатом, на что и направлен мой проект.

цв

Разрабатываемый продукт работает следующим образом:

Основной пользователь(студент) авторизуется на терминале в деканате используя свое ФИО и номер зачетной книжки и выбрать интересующий его вариант использования программы, в случае выбора функции просмотра успеваемости, студент должен получить список всех имеющихся у него задолженностей, в случае выбора функции печати ведомости, при удовлетворении условий сроков печати на принтере будет распечатана ведомость.

Для сотрудников деканата должно быть меню где они могут загружать данные об успеваемости студентов.

Проект будет использоваться студентами для сдачи задолженностей и должен существенно ускорить этот процесс.

Сейчас существует множество программных продуктов для автоматизации процессов учета и обучения в различных учебных заведениях, но все они направлены преимущественно на работу сотрудников деканата со

студентами, мой же проект нацелен в первую очередь на студентов и их взаимодействие с деканатом.

Процессы реализации приложения показаны в Таблице 1.

Таблица 1. Процессы реализации.

Процесс	Вход	Регламентирующий документ, правила и процедуры	Выход	Персонал, ПО, Оборудование
1.Предпроектный анализ	Предметная область, анализ объекта исследования, составление ТЗ	ТЗ, ГОСТ 34.602-2020, ГОСТ 19.701-90	Проведен предпроектный анализ, составленное ТЗ	ПК, разработчик
2.Проектный анализ	Анализ функциональных блоков, анализ способов реализации	ТЗ, ГОСТ 34.602-2020, ГОСТ 19.701-90	Готовый проектный анализ	ПК, разработчик
3.Проектирование	Модель, диаграммы	ТЗ, ГОСТ 34.602-2020, ГОСТ 19.701-90	Эскиз проекта	ПК, разработчик, draw.io
4.Реализация	Разработка алгоритмов и модулей	ТЗ, ГОСТ 34.602-2020, ГОСТ 19.701-90	Код приложения, интерфейс	ПК, разработчик

				к, pycharm, Qt designer
5.Отладка	Отладка модулей	ТЗ, ГОСТ 34.602-2020, ГОСТ 19.701-90	Рабочее приложение	ПК, разработчик, pycharm
6.Тестирование	Тестирование приложения	ТЗ, ГОСТ 34.602-2020, ГОСТ 19.701-90	Готовый продукт	ПК, разработчик
7.Реализация	Ввод в эксплуатацию	Руководство пользователя	Ввод в эксплуатацию	ПК, разработчик

Исходя из данных представленных в таблице прекрасно видны процессы предшествующие полноценной реализации проекта, видны цели и результаты каждого этапа реализации, а также документация, оборудование и персонал необходимые для реализации проекта.

## 1.2 Целевая аудитория

Целевая аудитория проекта – студенты и сотрудники деканата РГГМУ. Студенты выступают в роли непосредственно пользователей системы, персонал деканата в роли потенциального заказчика и обслуживающего персонала системы.

Кроме того, в проекте остается простор и возможности для расширения функционала, возможности использования в других учебных заведениях.

## 1.3 Техничко-экономическое обоснование

Для планирования процесса создания ПО АС, расчетов его стоимости необходимо определить трудоемкость разработки программных средств. Творческие элементы труда программистов могут быть определены либо на основе экспертных оценок опытных программистов, либо жесткими сроками разработки, в которых программист должен найти решение. Технические элементы труда программистов поддаются нормированию, но точность таких норм имеет большой разброс в зависимости от целого ряда факторов. На основе опытно-статистических данных различных организаций, занимающихся программированием задач для решения их на ЭВМ, выбран метод экспресс-оценки.

Для использования метода экспресс-оценки необходимо рассчитать каждый этап производства, указав его сроки и трудозатраты. Трудозатраты отражены на таблице 2.

Таблица 2. Трудозатраты.

№	Этап	Дата начала работ	Продолжительность	Дата окончания работ	Исполнитель	ЗП	Итого
1	Предпроектный анализ	01.01.2023	36	26.01.2023	Системный аналитик	3400 р. В день	122400 р.
1.1	Анализ предметной области	01.01.2023	14	06.01.2023	Системный аналитик	3400 р в день	47600 р.

1.2	Анализ объекта исследования	7.01.2023	14	17.01.2023	Системный аналитик	3400 р. В день	47600 р.
1.3	Составление ТЗ	18.01.2023	8	26.01.2023	Системный аналитик	3400 р. В день	27200 р.
2	Проектный анализ	27.01	30	6.02.2023	Системный аналитик	3400 р. В день	102000 р.
2.1	Анализ функциональных блоков	27.01.2023	15	1.02.2023	Системный аналитик	3400 р в день	51000 р.
2.2	Анализ способов реализации	2.02.2023	15	6.02.2023	Системный аналитик	3400 р в день	51000 р.
3	Проектирование	7.02.2023	40	25.02.2023	Системный аналитик	3400 Р. В день	136000 р.
3.1	UML	7.02.2023	25	13.02.2023	Системный аналитик	3400 Р. В день	85000 р.
3.2	Концептуальная модель	20.02.2023	15	25.02.2023	Системный аналитик	3400	51000 р.

						Р. В день	
4	Реализация	26.02. 2023	60	5.04.2 023	Програм мист Python	2100 р. В день	1260 00 р.
4. 1	Разработка алгоритмов	26.02. 2023	30	18.03. 2023	Програм мист Python	2100 р. В день	6300 0 р.
4. 2	Кодировани е модулей	19.03. 2023	30	5.04.2 023	Програм мист Python	2100 р. В день	6300 0 р.
5	Отладка	6.04.2 023	15	20.04. 2023	Програм мист Python	2100 р. В день	3150 0 р.
6.	Тестирован ие	21.04. 2023	40	15.05. 2023	Тестиров щик, програм мист python	2000/2 100 р. В день	8100 0 р.
6. 1	Альфа тестировани е	21.04. 2023	10	25.04. 2023	Тестиров щик	2000 р. В день	2000 0 р.
6. 2	Отчет по альфа тестировани ю	26.04. 2023	5	30.04. 2023	Тестиров щик	2000 р. В день	1000 0 р.

6.3	Ремонт	01.05.2023	10	5.05.2023	Программист Python	2100 р. В день	21000 р.
6.4	Повторное тестирование	6.05.2023	10	10.05.2023	Тестировщик	2000 р. В день	20000 р.
6.5	Отчет по итогам тестирования	11.05.2023	5	15.05.2023	Тестировщик	2000 р. В день	10000 р.
7	Оформление документации	16.05.2023	30	26.05.2023	Технический писатель	5100 р. В день	153000 р.
7.1	Работа с документами	16.05.2023	30	26.05.2023	Технический писатель	5100 р. В день	153000 р.
	<b>Стоимость работы сотрудников</b>						751900 р.
	<b>Приблизительная стоимость проекта</b>	3.10.2022	251	29.08.2023			~850000 р.

Стоимость работы в день для системного аналитика, программиста Python и тестировщика взяты с сайтов hh.ru при учете что рабочий день длится 8 часов.

Формула используемая для расчетов требуемого времени:

$$t_j^o = \frac{(3t_{min} + 2t_{max})}{5}$$

Где:

$t_j^o$  – ожидаемая продолжительность этапа(в днях)

$t_{min}$  – наименьшая по длительности оценка эксперта(в днях)

$t_{max}$  – наибольшая по длительности оценка эксперта(в днях)

## 1.4 Сроки реализации проекта

Диаграмма Ганта представлена на рисунке 1.

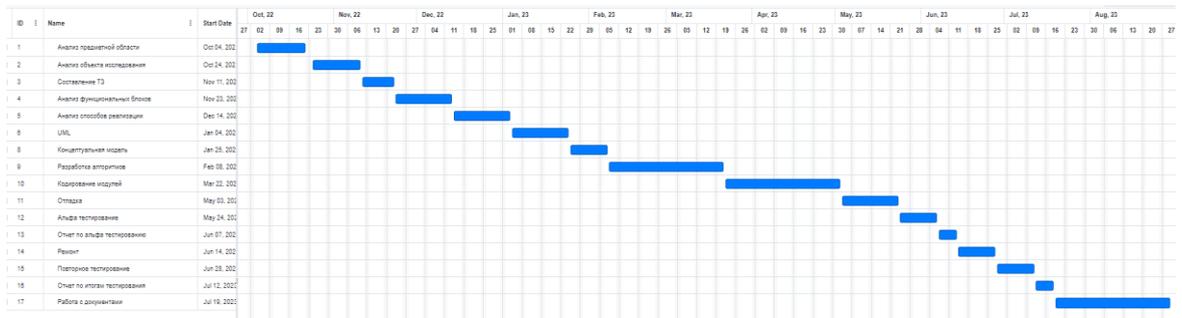


Рисунок 1. Диаграмма Ганта.

Диаграмма Ганта визуализирует сроки реализации проекта с обозначением задач, которые выполняются по ходу продвижения проекта.

На диаграмме отражены все этапы необходимые для реализации проекта:

- Анализ предметной области
- Анализ объекта исследования
- Составление ТЗ

- Анализ способов реализации
- UML
- Концептуальная модель
- Отладка
- Альфа тестирование
- Отчет по альфа тестированию
- Ремонт
- Повторное тестирование
- Отчет по повторному тестированию
- Работа с документами

Каждый этап реализации проекта выражен в рамках календарных дней с учетом выходных дней.

Как видно на диаграмме, для полноценной реализации проекта требуется 251 рабочий день.

В начале проекта проводится анализ предметной области для определения сущности проекта, его границ и первоначальные требования к функциональности, затем проводится анализ объекта исследования для установления его места в исследуемой области, затем составляется техническое задание для того чтобы готовый проект соответствовал ожиданиям заказчика.

После первого этапа, наступает этап Проектного анализа, в котором проводится анализ функциональных блоков и анализ способов реализации.

На этапе проектирования осуществляется UML-моделирование и составляется концептуальная модель проекта.

На этапе разработки осуществляется разработка алгоритмов и кодирование модулей.

После разработки осуществляется отладка.

Следующий этап – тестирование, это немаловажный этап в ходе которого проводится альфа тестирование, по итогам которого составляется отчет о альфа тестировании и осуществляется ремонт программы, после чего проводится повторное тестирование, по итогам которого снова составляется отчет и устраняются выявленные проблемы.

На последнем этапе разработки программы осуществляется работа с документацией.

На момент написания работы проект находится на стадии альфа-тестирования.

### **1.5 Функциональные требования**

Функциональные требования нужны для определения функциональности разрабатываемого программного обеспечения. В ходе анализа были выявлены следующие функциональные требования:

- Отслеживание успеваемости
- Печать ведомостей
- Загрузка актуальных данных об успеваемости сотрудниками деканата
- Выставление сроков доп.сессии для возможности печати ведомостей

### **1.6 Нефункциональные требования**

Нефункциональные требования описывают характеристики и рамки которыми должна обладать разрабатываемая система. Выявленные следующие нефункциональные требования:

- Приложение должно быть разработано на языке программирования python
- База данных должна быть на языке MySQL

- Приложения должно быть разработано для OS Windows
- Авторизация пользователя должна осуществляться по ФИО и номеру зачетной книжки

## 1.7 Подходы SWOT, ISA, BPR, USM

### 1.7.1 Подход SWOT

Подход SWOT требуется для идентификации, классификации, ранжированию и выбора проекта по разработке информационной системы так, чтобы они были увязаны с положительными и отрицательными факторами организации, а также различными возможностями и угрозами. Подход представлен в таблице 3.

Таблица 3. SWOT-анализ.

	Положительное влияние	Отрицательное влияние
Внутренняя среда	<ul style="list-style-type: none"> <li>• Востребованность</li> <li>• Актуальность</li> </ul>	<ul style="list-style-type: none"> <li>• Отсутствие прямого доступа к университетской бд</li> </ul>
Внешняя среда	<ul style="list-style-type: none"> <li>• Заинтересованность со стороны деканата</li> </ul>	<ul style="list-style-type: none"> <li>• Ограниченность использования</li> </ul>

Как видно в таблице, во внутренней среде положительное влияние оказывают востребованность и актуальность данной системы, в то время как отсутствие прямого доступа к университетской базе данных оказывает негативное влияние. Во внешней среде положительную роль играет заинтересованность со стороны деканата, ограниченность использования оказывает отрицательное влияние.

## 1.7.2 Подход ISA

Подход ISA являет собой таблицу, используемую для конструирования сложных инженерных продуктов и информационных систем. Подход представлен на таблице 4.

Таблица 4. ISA.

	А	В	С	Д	Е	Ф
1	Интерфейс	Пользователи	Монетизация	Контент	Общение	Обучение
2	Дата выпуска	Кол-во посещений	Покупка ИС	Точка окупаемости	Получение прибыли	Идея
3	БД	Сервер	Процессор	Программный код	Пользователь	Отладка
4	Пользователи	Информация	Программисты	Приложение	Юристы	Тех. поддержка
5	При покупке ИС	Во время использования ИС	В момент создания аккаунта	Во время обновления	При взломе ИС	При сбое в системе
6	Прибыль	Развитие	Создание конкуренции	Наличие спроса	Прогресс	Трудоустройство

На таблице отражены точки зрения(строки), принадлежащие пяти основным участникам разработки ИС:

1. Планировщик
2. Владелец
3. Проектировщик
4. Конструктор
5. Субподрядчик

В столбцах представлены шесть архитектурных моделей, с каждой из которых взаимодействует каждый вышеупомянутый участник:

1. Состав сущности
2. Функционирование сущности
3. Расположение сущности
4. Взаимодействующий с сущностью
5. Распределение событий
6. Мотивация

### 1.7.3 Подход BPR

Подход BPR подразумевает собой подход к планированию разработки используя BPR-методы, т.е. методы реинжиниринга бизнес-процессов.

Рассматриваемые бизнес-процессы в ходе реинжиниринга представлены на рисунках 2 и 3

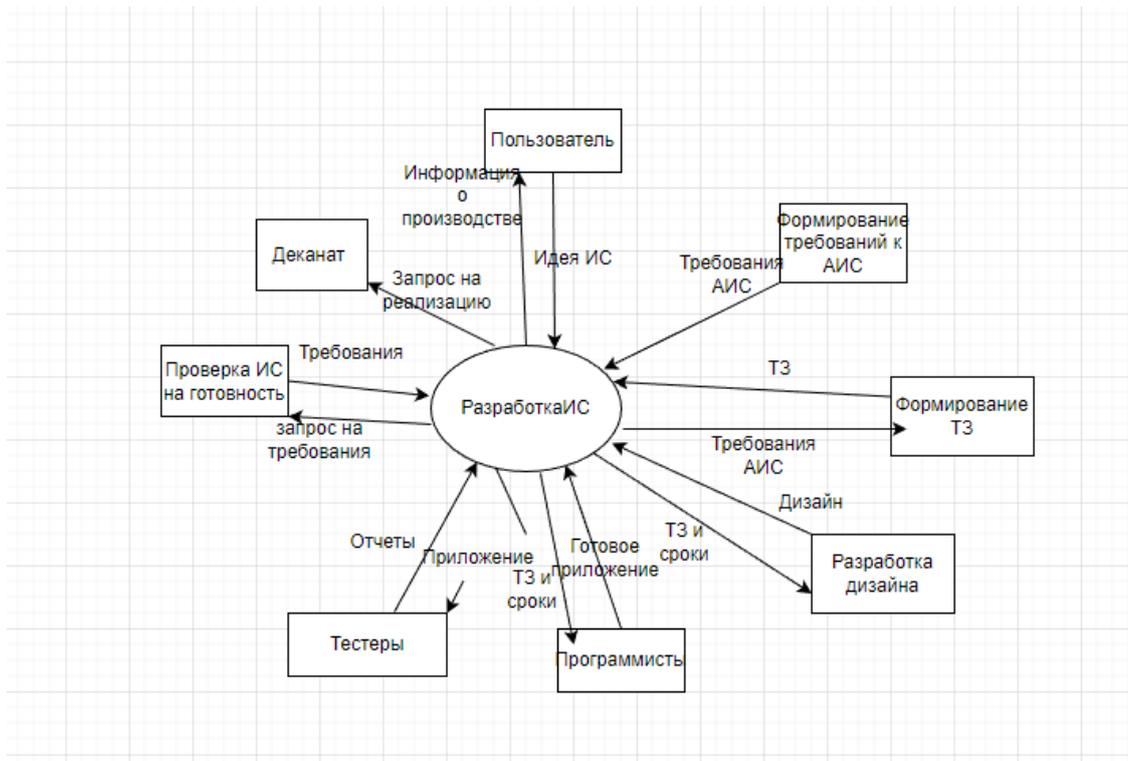


Рисунок 2. BPR.

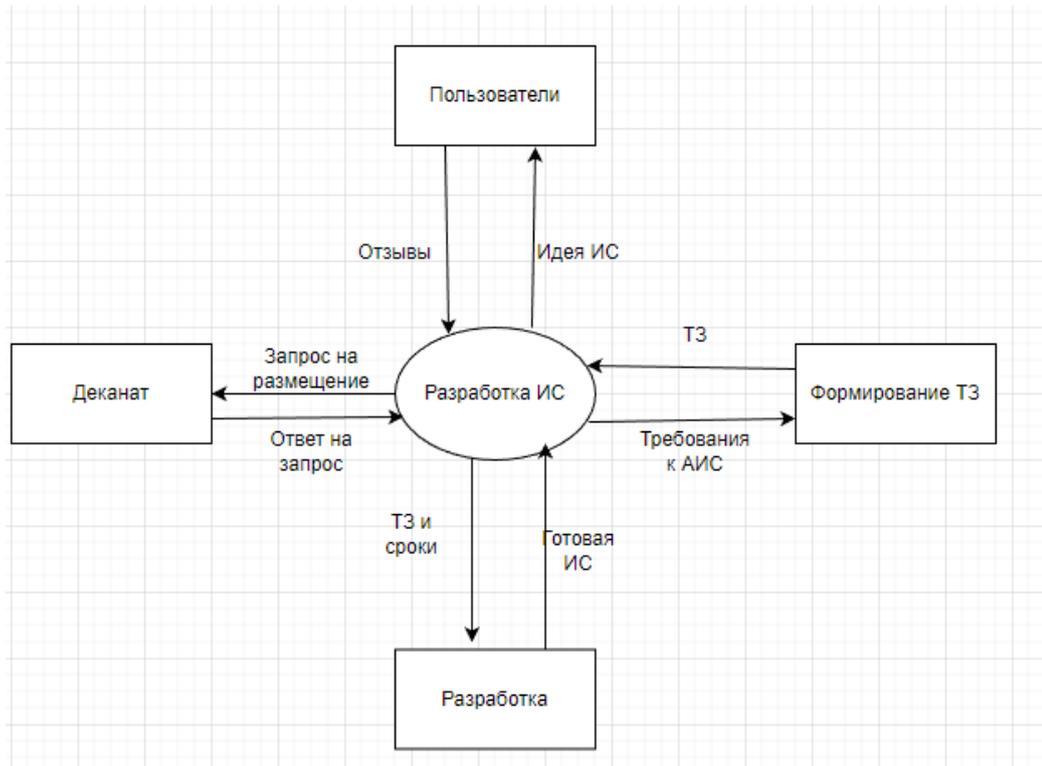


Рисунок 3. BPR.

По результатам реинжиниринга бизнес-процессов можно будет реализовать следующие изменения в системе:

- Реализовать более плотную связь с ИС: Университет
- Изменение интерфейса приложения
- Создать версии для других ОС
- Дополнить функционал программы

#### 1.7.4 Подход VSM

Подход VSM состоит в оценивании информационной емкости продукта, оценивании роли в отраслевой структуре, выявлении конкурентного преимущества, рассмотрении вопроса каким образом продукт создает новое направление и выявлению выгоды из проекта. Подход показан на таблице 5.

Таблица 5. VSM.

Ёмкость	Кол-во пользователей
Роль в отраслевой структуре	Улучшение образовательного процесса
Конкурентные преимущества	Отсутствие конкурентов
Новое направление	Упрощение и ускорение взаимодействия студентов с деканатом в рамках доп.сессии
Выгода	Улучшение образовательного процесса

Исходя из информации в таблице ёмкость системы оценивается количеством пользователей, роль в отраслевой структуре – улучшением образовательного процесса, отсутствие конкурентов на данный момент является конкурентным преимуществом, так же упрощение и ускорение взаимодействия студентов с деканатом в рамках дополнительной сессии является новым направлением в этой области, улучшение образовательного процесса это несомненная выгода от проекта.

### 1.8 Техническое задание

Техническое задание(ТЗ) – это основной и очень важный документ, определяющий требования и порядок создания информационной системы. В соответствии с ТЗ реализуется разработка ИС, ее приемка и ввод в работу.

Техническое задание находится в приложении А.

## Глава 2. Моделирование системы.

### 2.1 Выбор методологии проектирования

Главной целью проектирования информационной системы является создание максимально эффективной и функциональной системы, которая соответствует требованиям.

Проектирование информационной системы состоит из:

- Анализа бизнес-процессов
- Определения требований к системе
- Разработки архитектуры
- Разработки дизайна

Если грамотно разработать и реализовать информационную систему, она будет эффективнее и производительнее.

Исходя из вышесказанного встает вопрос о выборе методологии проектирования.

Существует множество методологий моделирования бизнес-процессов, разной степени популярности, и имеющие свои преимущества и недостатки в разных ситуациях.

Из самых распространенных выделяются методологии “IDEF”, “UML”, “BPMN” и “DFD”. Рассмотрим их подробнее:

**IDEF** – это группа методологий для решения задач по моделированию сложных систем. IDEF позволяет отображать и анализировать модели информационных систем во множестве различных разрезов. Для описания работы процессов в системе на разных уровнях детализации используются

различные типы IDEF-диаграмм, например такие как:

- IDEF0 – для анализа бизнес-процессов
- IDEF3 – для моделирования информационных потоков в системе
- IDEF1X – для моделирования данных для разработки семантических моделей данных

**UML** – это язык графического описания, который используется для моделирования объектов и бизнес-процессов при разработке ПО. UML использует графические изображения для создания понятной модели системы.

**BPMN**  – это нотация используемая для моделирования бизнес-процессов. BPMN позволяет создавать графические диаграммы, в которых довольно ясно отражены последовательность действий, роли в системе и потоки данных.

**DFD** – это методология для описания системы в виде графических диаграмм. DFD помогает определять источники и адресаты данных, потоки и хранилища данных, логические функции.

Выбирая методологию проектирования, надо взвесить все плюсы и минусы, учесть специфику разработки, предметную область. В ходе анализа было принято решение о использовании UML, ввиду того что на основании UML можно сгенерировать код, а также он обеспечивает стандартизированный подход к моделированию, который значительно упрощает процесс коммуникации между разработчиком и бизнес-проектировщиками. С помощью UML будут визуализированы такие аспекты системы как

структура, поведение, взаимодействие между объектами в системе, развертывание.

## 2.2 Моделирование предметной области

### 2.2.1 Диаграмма прецедентов

Диаграмма прецедентов представлена на рисунке 4.

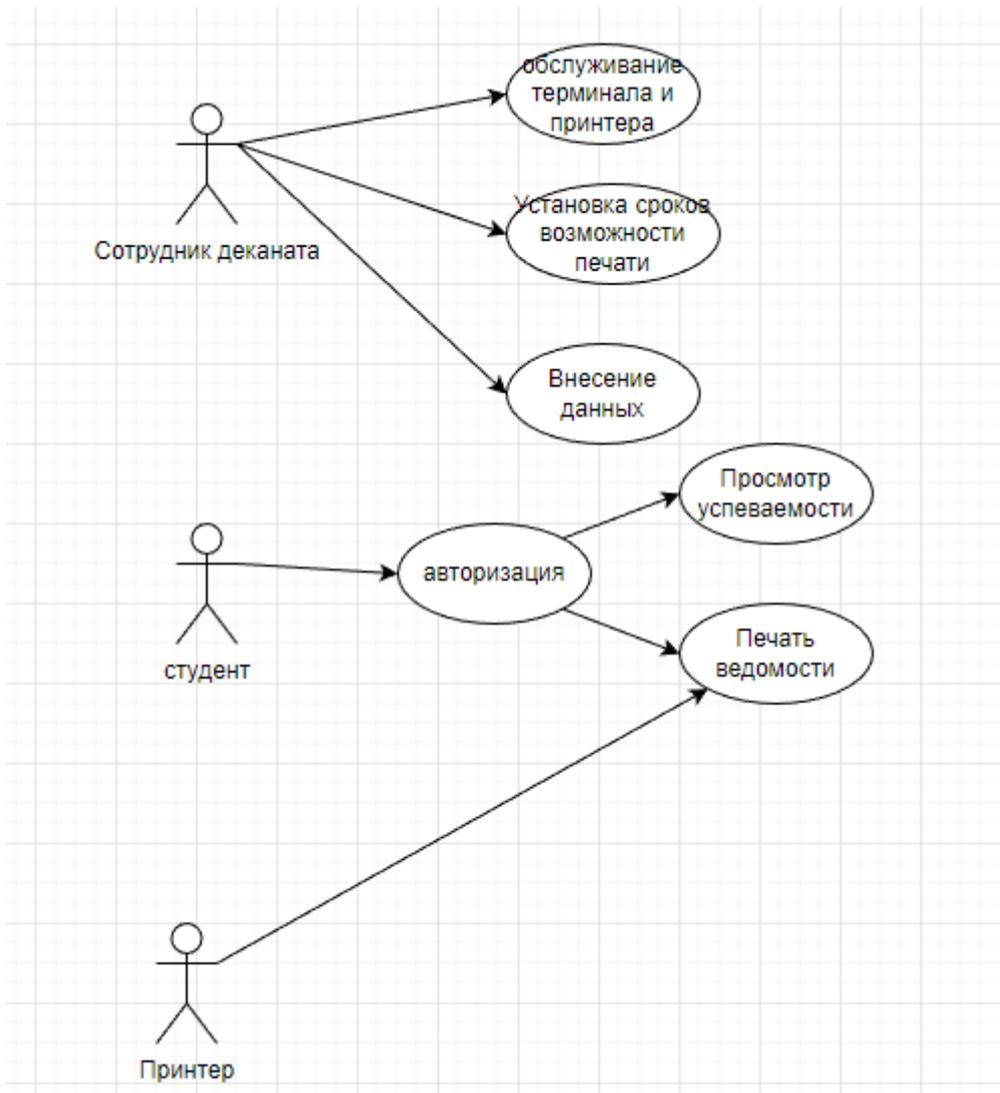


Рисунок 4. Диаграмма прецедентов.

На данной диаграмме представлены исполнители в лице студента, принтера и сотрудника деканата. На рисунках ниже представлены их задачи и соответствующие им бизнес-процессы.

Исполнитель	Задачи	Бизнес-процессы
Сотрудник деканата	Обслуживание терминала, установка сроков доп.сессии, внесение актуальных данных об успеваемости	Обслуживание терминала и принтера, установка сроков возможности печати, Внесение данных
	Использование терминала по назначению	Авторизация, Просмотр успеваемости, печать ведомости
Студент Принтер	Печать	Печать ведомости

Рисунок 5. Главнйй раздел выполнения ВИ.

В следующих рисунках представлены сценарии для процессов в диаграмме прецедентов.

На рисунке 6. представлен раздел выполнения варианта использования “Обслуживание терминала и принтера”

Главный раздел сценария выполнения варианта использования "обслуживание терминала и принтера"	
Вариант использования	Обслуживание терминала и принтера
Актеры	Сотрудник деканата
Краткое описание	Сотрудник деканата обслуживает терминал и принтер
Цель	Поддержание работоспособности
Тип	Зависимый
Типичный ход событий сценария выполнения варианта использования	
Действия актеров	Отклик системы
1.Сотрудник заправляет принтер 2.Сотрудник поддерживает работоспособность терминала	

Рисунок 6. Типичный ход сценария “Обслуживание терминала и принтера”.

На рисунке 7 представлен раздел выполнения варианта использования “Установка сроков возможности печати”

Главный раздел сценария выполнения варианта использования "Установка сроков возможности печати"	
Вариант использования	Установка сроков возможности печати
Актеры	Сотрудник деканата
Краткое описание	Сотрудник устанавливает сроки доп.сессии
Цель	Установка ограничения сроков печати
Тип	Зависимый
Типичный ход событий сценария выполнения варианта использования	
Действия актеров	Отклик системы
1.Сотрудник деканата устанавливает сроки в формате dd/mm/yy	2.Проверка формата введенных данных Исключение 1: Ошибка ввода формата данных 3.Сроки установлены
Исключение сценария выполнения варианта использования	
Исключение 1:	
Действия актеров	Отклик системы
	3.Ошибка формата ввода данных 4.Возвращение к окну ввода сроков доп.сессии

Рисунок 7. Типичный ход сценария “Установка сроков возможности печати”.

На рисунке 8 представлен раздел выполнения варианта использования “Внесение данных”

Главный раздел сценария выполнения варианта использования "Внесение данных"	
Вариант использования	Внесение данных
Актеры	Сотрудник деканата
Краткое описание	сотрудник деканата вносит файл с актуальными данными
Цель	Поддержка актуальности данных
Тип	Зависимый
Типичный ход событий сценария выполнения варианта использования	
Действия актеров	Отклик системы
1.Сотрудник деканата заносит файл содержащий актуальные данные успеваемости	2.Данные обновлены

Рисунок 8. Типичный ход сценария “Внесение данных”.

На рисунке 9 представлен раздел выполнения варианта использования “Авторизация”

Главный раздел сценария выполнения варианта использования "Авторизация"	
Вариант использования	Авторизация
Актеры	Студент
Краткое описание	Студент авторизуется в систему
Цель	Авторизироваться в систему
Тип	Зависимый
Типичный ход событий сценария выполнения варианта использования	
Действия актеров	Отклик системы
1. Студент вводит ФИО и номер Зач. Книжки	2. Проверяет данные 3. Осуществляет вход в систему Исключение 1
Исключение сценария выполнения варианта использования	
Исключение 1:	
Действия актеров	3. Данные не верны 4. Возвращает к меню авторизации

Рисунок 9. Типичный ход сценария “Авторизация”.

На рисунке 10 представлен раздел выполнения варианта использования “Просмотр успеваемости”

Главный раздел сценария выполнения варианта использования "Просмотр успеваемости"	
Вариант использования	Просмотр успеваемости
Актеры	Студент
Краткое описание	Студент отслеживает успеваемость
Цель	Просмотреть успеваемость
Тип	Зависимый
Типичный ход событий сценария выполнения варианта использования	
Действия актеров	Отклик системы
1. Студент выбирает функцию "Посмотреть успеваемость"	2. Выводится таблица содержащая информацию о предмете и оценке
Исключение сценария выполнения варианта использования	
Исключение 1:	
Действия актеров	3. Данные не верны 4. Возвращает к меню авторизации

Рисунок 10. Типичный ход сценария “Просмотр успеваемости”.

На рисунке 11 представлен раздел выполнения варианта использования “Печать ведомости”

Главный раздел сценария выполнения варианта использования "Печать ведомости"	
Вариант использования	Печать ведомости
Актеры	Студент, принтер
Краткое описание	Студент выбирает функцию печати ведомости, принтер печатает
Цель	Печать ведомости
Тип	Зависимый
Типичный ход событий сценария выполнения варианта использования	
Действия актеров	Отклик системы
1. Студент выбирает функцию "Распечатать ведомость"	2. Выводится окно в котором следует выбрать предмет, по которому имеется задолженность, для печать ведомости
3. Студент выбирает нужный ему предмет	4. Принтер распечатывает ведомость Исключение 1
Исключение сценария выполнения варианта использования	
Исключение 1:	
Действия актеров	
	4. Выводится окно о неисправности принтера/отсутствии чернил/тонера в принтере 5. Возвращает в главное меню

Рисунок 11. Типичный ход сценария “Печать ведомости”.

На вышеперечисленных рисунках показаны сценарии выполнения вариантов использования, где отражен ход событий, актеры, описания, цели и тип для каждого из сценариев. Разделы сценария довольно информативны чтобы описывать подробно каждый из них.

### 2.2.2 Диаграммы последовательностей

Диаграммы последовательностей моделируют взаимодействия между объектами в системе, позволяют увидеть как различные части системы взаимодействуют друг с другом для выполнения конкретных функций.

Диаграммы последовательностей отображены на рисунках 12-16.

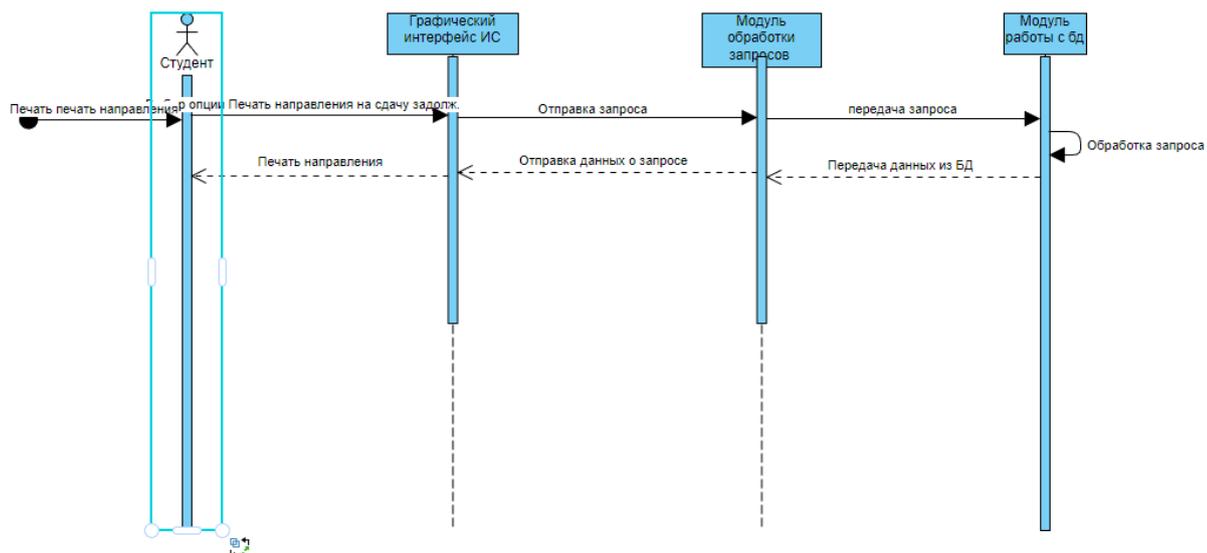


Рисунок 12. Диаграмма последовательности при выполнении функции “Печать направления”.

На рисунке 12 отображена диаграмма последовательности при выполнении функции “Печать ведомости”. На ней отображены актеры (Студент), объекты системы соответствующие процессу, активационные планки отражают какие компоненты системы задействованы в ходе процесса, также отражены синхронные, например “Выбор опции просмотр долгов” или “Отправка запроса” и возвращаемые сообщения такие как “Вывод информации о долгах”.

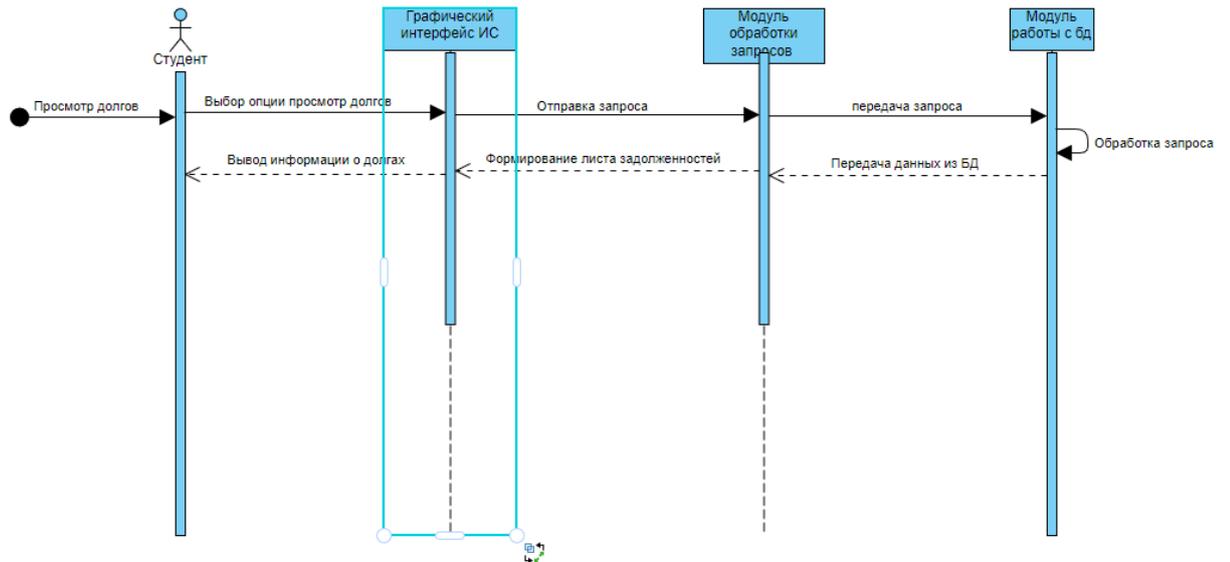


Рисунок 13. Диаграмма последовательности при выполнении функции “Просмотр долгов”.

На рисунке 13 отображена диаграмма последовательности при выполнении функции “Просмотр долгов”. На ней отображены актеры (Студент), объекты системы соответствующие процессу, активационные планки отражают какие компоненты системы задействованы в ходе процесса, также отражены синхронные, к примеру “Выбор опции просмотр долгов” и возвращаемые сообщения, такие как “Вывод информации о долгах”.

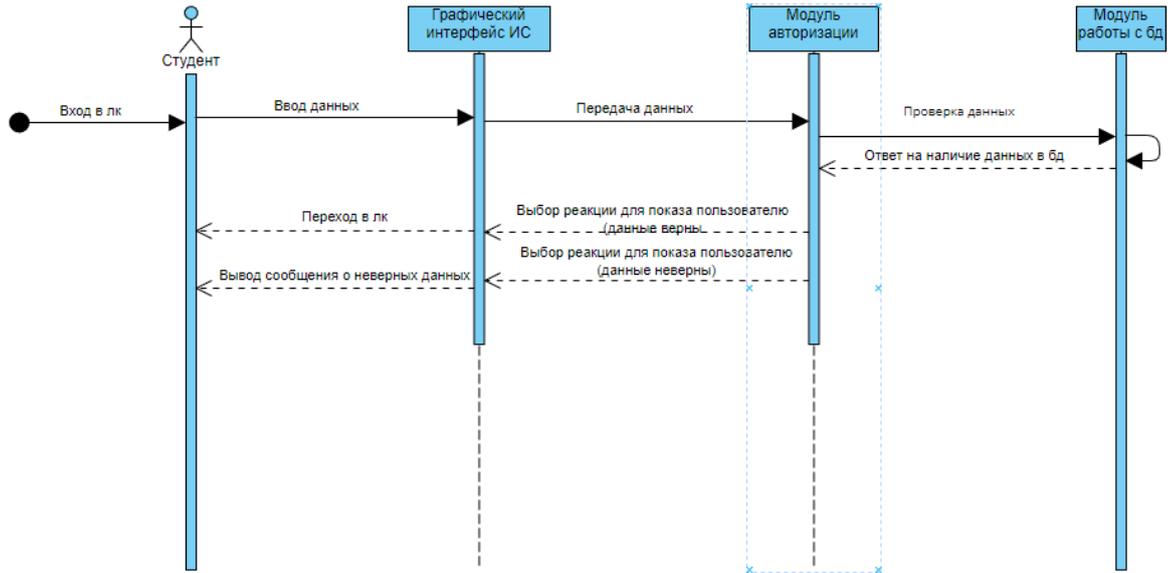


Рисунок 14. Диаграмма последовательности при выполнении функции “Вход в лк”.

На рисунке 14 отображена диаграмма последовательности при выполнении функции “Вход в лк”. На ней отображены актеры (Студент), объекты системы соответствующие процессу, активационные планки отражают какие компоненты системы задействованы в ходе процесса, также отражены синхронные, например “Ввод данных” и возвращаемые сообщения, например “Переход в лк”.

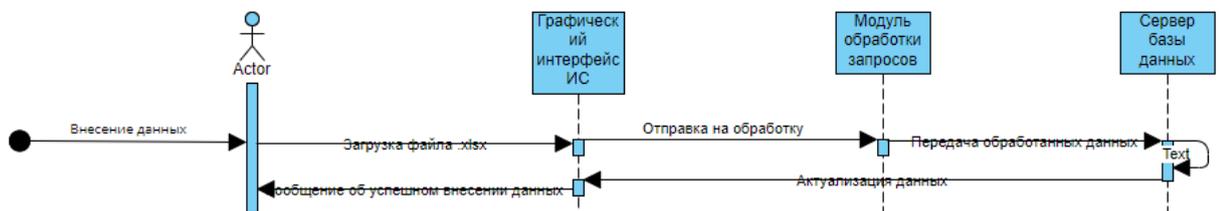


Рисунок 15. Диаграмма последовательности при выполнении функции “Внесение данных”.

На рисунке 15 отображена диаграмма последовательности при выполнении функции “Внесение данных”. На ней отображены актеры (Сотрудник деканата), объекты системы соответствующие процессу, активационные планки отражают какие компоненты системы задействованы в ходе процесса, также отражены синхронные, например “Загрузка файла .xlsx” и возвращаемые сообщения, например “Актуализация данных”.

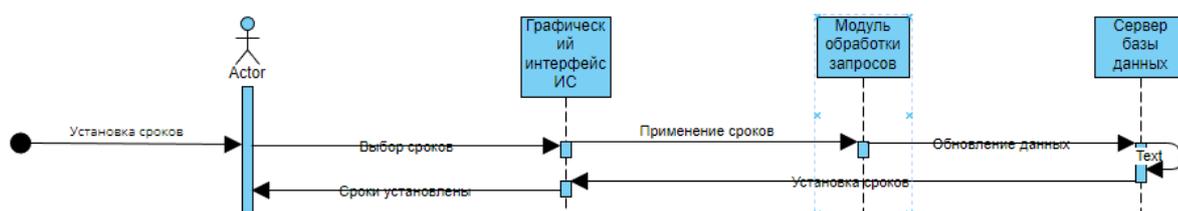


Рисунок 16. Диаграмма последовательности при выполнении функции “Установка сроков”.

На рисунке 16 отображена диаграмма последовательности при выполнении функции “Установка сроков”. На ней отображены актеры (Сотрудник деканата), объекты системы соответствующие процессу, активационные планки отражают какие компоненты системы задействованы в ходе процесса, также отражены синхронные, например “Выбор сроков” и возвращаемые сообщения, например “Установка сроков”.

### 2.2.3 Диаграмма компонентов

Диаграмма компонентов используется для визуализации организации компонентов системы. Компонентами системы могут являться программные компоненты, как например база данных или интерфейс системы, терминал и т.д.

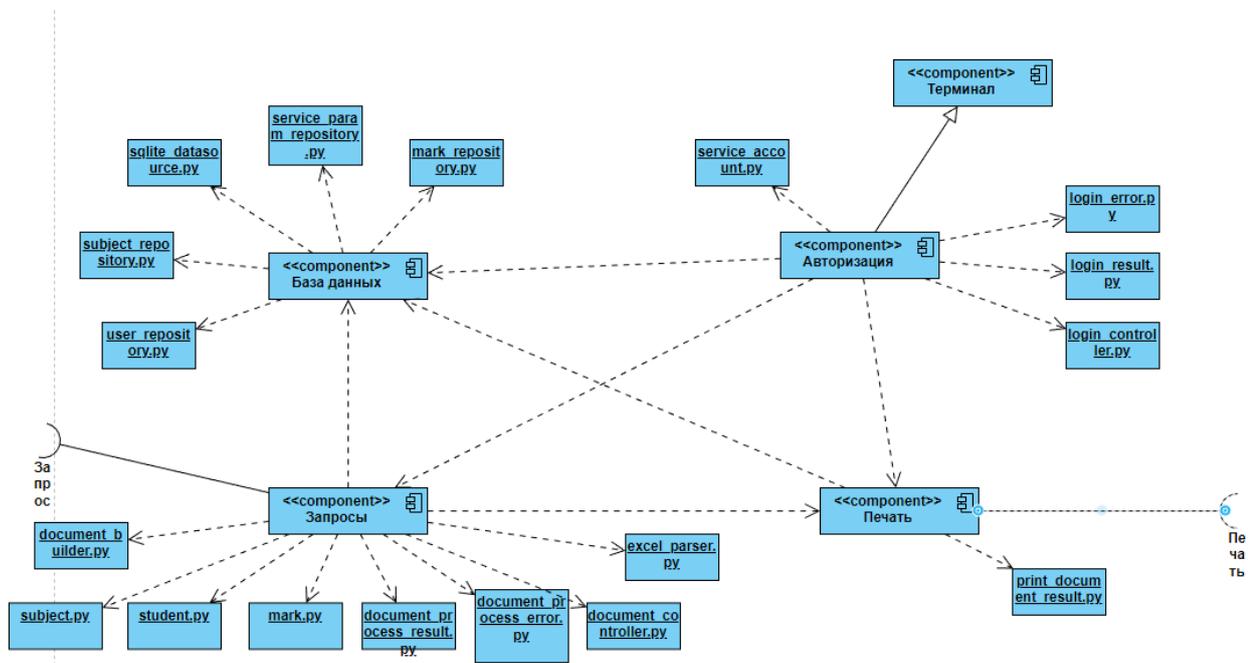


Рисунок 17. Диаграмма компонентов.

На рисунке видна организация и взаимодействие компонентов системы между собой, с указанием их зависимостей, и соединений. Основными компонентами системы являются: терминал, авторизация, база данных, запросы и печать.

#### 2.2.4 Диаграмма развертывания

Диаграмма развертывания – это диаграмма, которая отражает архитектуру исполнения системы. Она включает в себя аппаратные или же программные узлы исполнения. Диаграмма развертывания используется для визуализации физического аппаратного и программного обеспечения системы, дает понимание о том как конкретно будет физически развернута система.

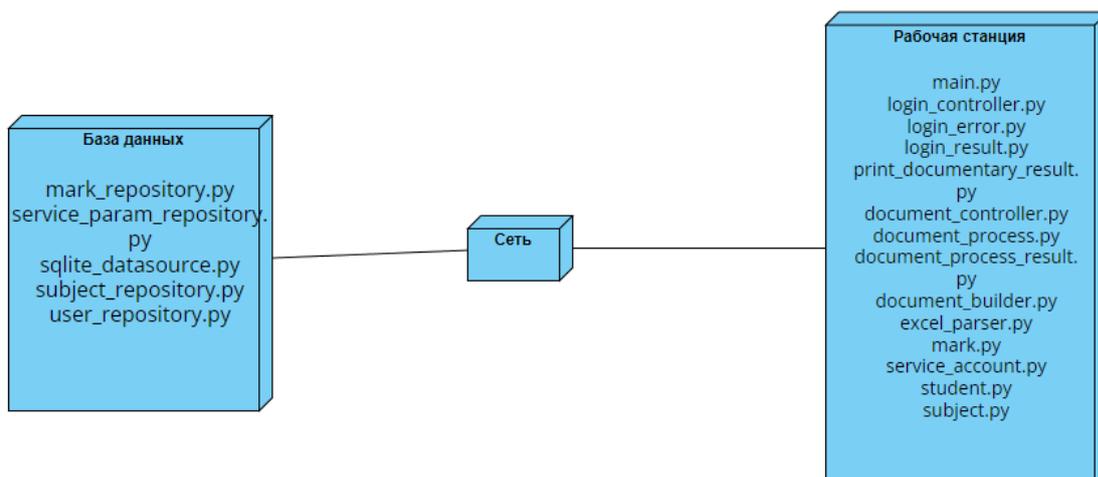


Рисунок 18. Диаграммы развертывания.

На диаграмме видно два основных компонента системы – базу данных и рабочую станцию(терминал) объединенных в одну сеть, у каждого компонента указаны соответствующие им программные компоненты.

### 2.3 Схема базы данных

Схема базы данных представляет собой логическую конфигурацию базы данных. Схема показывает как связаны между собой таблицы, представления из которых состоит база данных.

Для любой информационной системы в которой хранятся и обрабатываются большие объемы данных крайне важно изначально грамотно спроектировать схему базы данных, ибо в дальнейшем неправильно составленная схема приведет к тому что надо будет потратить много усилий для исправлений ошибок и последствий ее распространения.

Схема выполнена с помощью языка моделирования IDEF1X, которая используется для разработки семантики моделей данных, формирования

графических представлений информационных моделей, которые отражают структуру и семантику информации внутри среды или системы.

Схема базы данных отображена на рисунке 19.

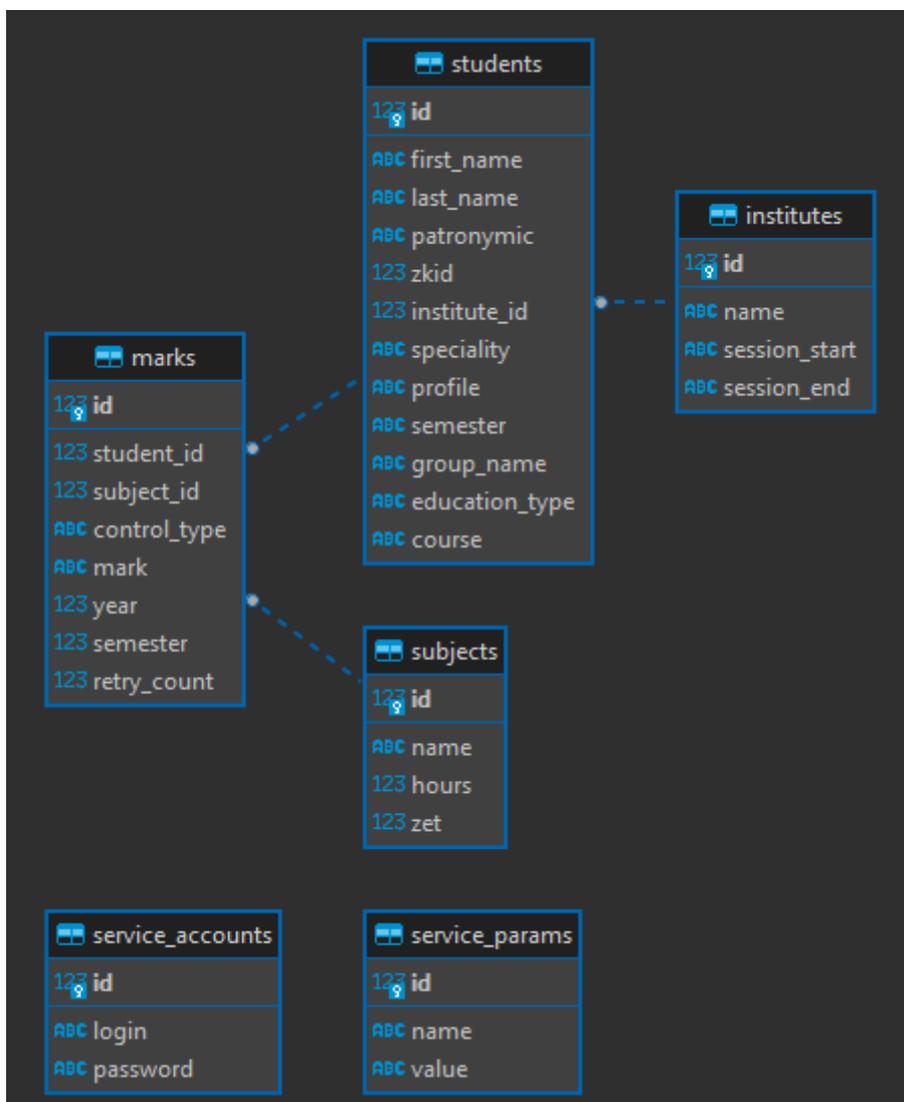


Рисунок 19. Схема базы данных.

На схеме отражены 5 основных таблиц и их связи между собой, каждая таблица содержит в себе столбцы в которых содержится необходимая информация. В таблице “marks” содержится информация об id студента, которое генерируется по мере добавления студентов в базу данных, id дисциплины которая генерируется при добавлении дисциплин в базу данных, а также информацию о виде контроля, оценке, году и номеру семестра в котором проводилась аттестация по определенной дисциплине. В таблице

“subjects” содержится информация о самих дисциплинах, их id, наименование, и о часах выделенных на предмет. В таблице “students” содержится информация о студентах, их ФИО, номер зачетной книжки, их факультете, направлении обучения, профилю, номеру группы, виду обучения. В таблице “institutes” содержится информация о факультетах и хранится информация о сроках дополнительной сессии. В таблице “service\_accounts” содержатся данные с учетными записями для обслуживающего персонала. В таблице “service\_params” содержатся учетные записи студентов.

## **Глава 3. Реализация системы**

### **3.1 Интерфейс системы**

Интерфейс – это граница между пользователем и программой, операционной системой, или же способ взаимодействия приложений между собой.

Для чего нужен интерфейс:

- Ввод и отображение информации
- Управление отдельными приложениями
- Обмен данными с другими устройствами
- Взаимодействие с системой

В рамках проекта будет рассмотрен графический интерфейс, программный, аппаратный и аппаратно-программный интерфейс системы.

#### **3.1.1 Графический интерфейс системы**

Графический пользовательский интерфейс(GUI) – это то что видит пользователь системы при взаимодействии с ней. Это могут быть разные окна, кнопки и т.д.

Графический интерфейс делает работу с программой более наглядной и понятной, вместо инициализации скриптов через интерпритаторы или команду строку достаточно нажать нужную кнопку. Графический интерфейс состоит из множества компонентов:

- Окна
- Меню
- Кнопки и интерактивные элементы
- Указатели
- Манипуляции

Хороший графический интерфейс должен соответствовать определенным требованиям, таким как:

- Наглядность
- Понятность
- Удобство
- Эффективность
- Обратная связь
- Последовательность
- Дизайн

Руководствуясь требованиями к графическому интерфейсу в ходе работ над проектом, с помощью библиотеки PyQt6 и среде разработки графических интерфейсов Qt Designer был разработан интерфейс, пример которого отражен на рисунке 20.

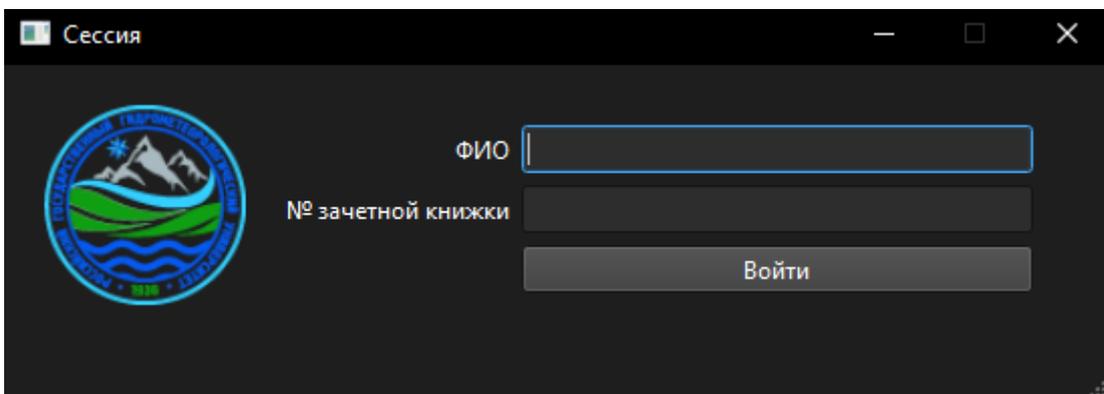


Рисунок 20. Графический интерфейс меню авторизации.

На рисунке 20 видны такие компоненты как само окно, меню авторизации, кнопка войти, интерактивные элементы в виде строк ввода

Указатели в виде мигающей вертикальной черты в строке ввода, так же с окном возможно провести определенные манипуляции, окно можно свернуть, открыть на весь экран или вовсе закрыть.

### 3.1.2 Программный интерфейс системы

Программный интерфейс системы (API) – это набор способ и правил, по которым различные программы взаимодействуют между собой и обмениваются данными.

API значительно ускоряет процесс разработки, он позволяет использовать уже готовые функции. Многие API предоставлены операционной системой, которая дает следующие возможности:

- Работать с файловой системой
- Отрисовывать графику
- Хранить данные
- Использовать сетевые возможности
- Воспроизводить аудио, видео и т.д

Помимо системных API есть и сторонние, они могут быть представлены в виде разнообразного ПО либо же библиотек, что значительно ускоряет процесс разработки используя уже готовые решения.

В ходе проектирования и разработки программы был сделан выбор в пользу использования языка программирования Python, за счет его удобства и наличия удобных библиотек, необходимых для реализации программы.

В ходе разработки были использованы следующие библиотеки:

- Pandas
- openpyxl
- PyPDF2
- Pywin32
- PyQt6

Библиотека Pandas используется для обработки и анализа данных, предоставляет специальные структуры данных и операции для обработки

таблиц и временных рядов. В разрабатываемой системе Pandas используется для обработки данных выгрузки из 1С:Университет в формате .xlsx.

Библиотека openpyxl используется для чтения и редактирования файлов с расширениями .xlsx, .xlsm, .xltx и .xltm.

Библиотека PyPDF2 это библиотека для работы с файлами PDF. В разрабатываемой программе эта библиотека используется для формирования ведомости на печать в виде файла с расширением .pdf.

Библиотека Pywin32 позволяет получить доступ к Component Object Model (COM) и управлять приложениями Microsoft через Python. В разрабатываемой информационной системе используется в модуле печати для считывания файлов .xlsx.

Библиотека PyQt6 используется для создания графического интерфейса с помощью инструментария Qt Designer.

### **3.1.3 Аппаратный интерфейс системы**

Аппаратный интерфейс – это система шин, разъемов, портов и согласующих устройств, обеспечивающих связь всех частей системы между собой.

В разрабатываемой информационной системе аппаратный интерфейс представлен связью терминал-принтер, принтер подключается к терминалу кабелем USB 2.0/3.0.

## **3.2 Окно Авторизации**

Меню авторизации встречает пользователя при запуске программы, и по сути является “воротами” в систему и к ее функционалу.

Графический интерфейс окна был выполнен в Qt Designer.

На рисунке 21 видны такие элементы как само окно, поле ввода для ФИО (выполняет роль логина) и номера зачетной книжки (выполняет роль пароля),

а также кнопка “Войти” после нажатия которой осуществляется вход в учетную запись и переход к следующему меню, в зависимости от введенных данных. Данные об учетных записях администраторов и обычных пользователей-студентов хранятся в отдельных таблицах в базе данных.

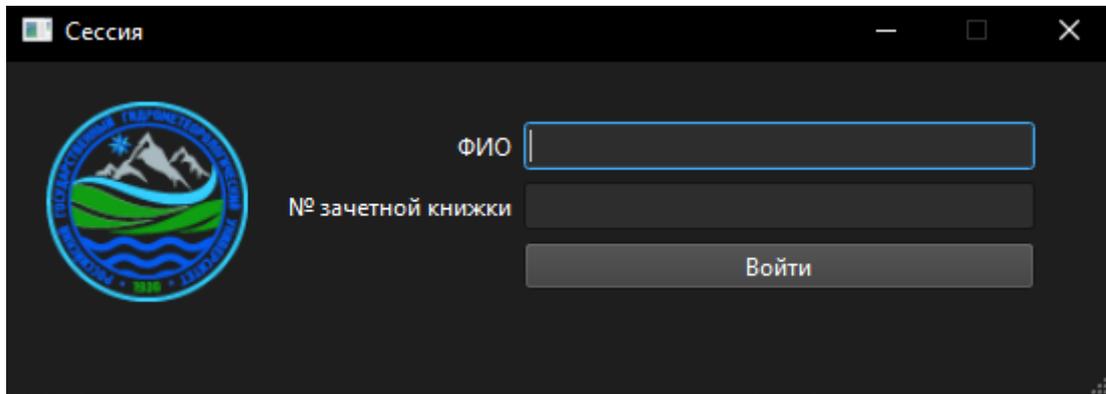


Рисунок 21. Графический интерфейс меню авторизации.

За работу окна и функционал отвечают следующие Python скрипты:

1. main.py
2. login\_controller.py
3. login\_error.py
4. login\_result.py
5. service\_account.py
6. student.py

### 3.3 Админское меню

Админское меню – это главная рабочая панель администратора приложения. С помощью админского меню осуществляется работа администратора, добавляется и редактируется контент и информация, осуществляются необходимые настройки программы, в пределах компетенции регулирует работу приложения.

В рамках разрабатываемого проекта в админском меню осуществляются следующие функции:

- Загрузка актуальных данных об успеваемости (загрузка выгрузки из 1С:Университет в формате .xlsx)
- Установка сроков дополнительной сессии для каждого факультета

Админское меню показано на рисунке 22.

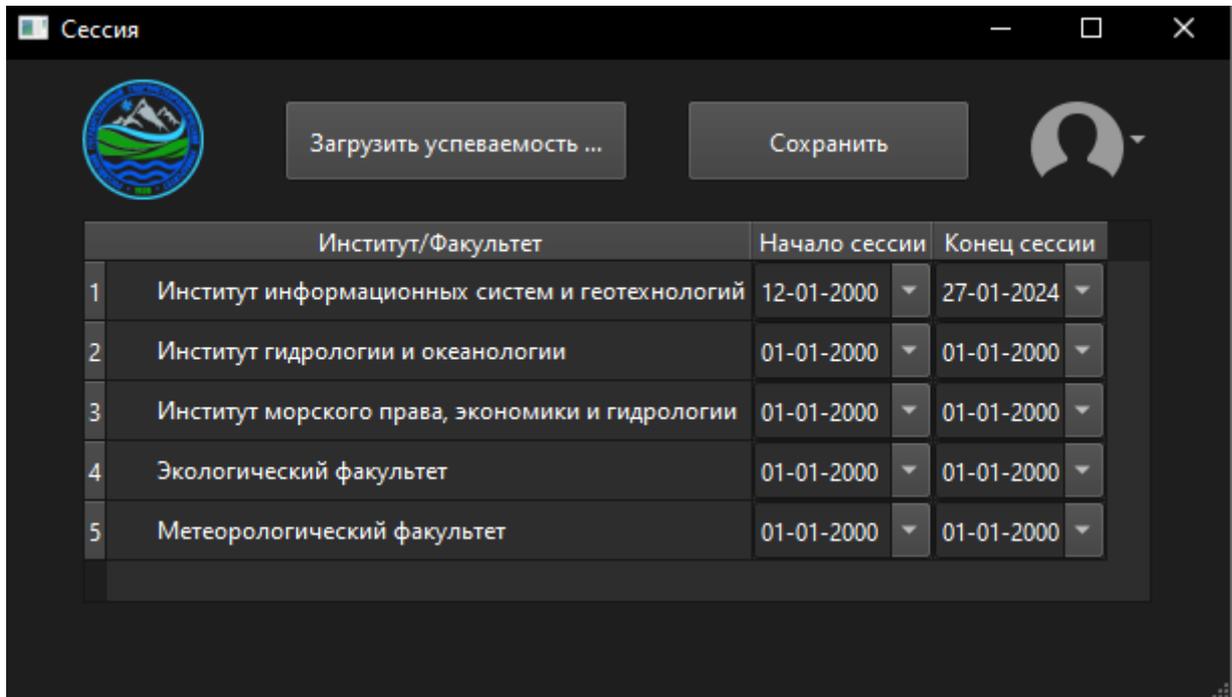


Рисунок 22. Графический интерфейс админского меню.

На рисунке видна информация о залогиненном пользователе, интерактивные поля выбора для даты начала сессии и конца сессии (на рисунке 23 отображено более подробно), кнопка “Сохранить” которая подтверждает даты сессии и устанавливает сроки возможной печати, также видна кнопка “Загрузить успеваемость ...” при нажатии которой появляется возможность выбрать файл формата .xlsx, содержащий данные об успеваемости студентов, после чего происходит парсинг информации и размещение её в базе данных (программный код excel\_parser.py находится в приложении Б). Еще имеется кнопка “Выйти”, при нажатии которой осуществляется выход к предыдущему меню – меню авторизации.

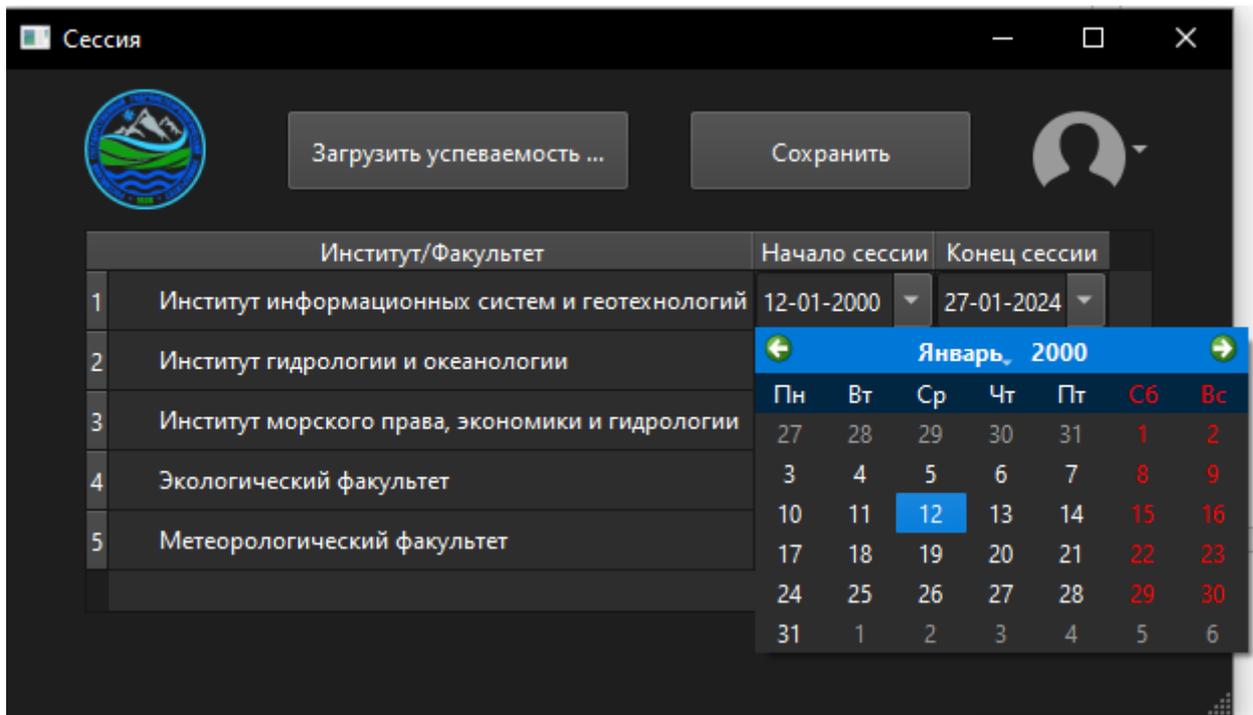


Рисунок 23. Графический интерфейс админского меню, интерактивное окно выбора даты.

За работу окна и функционал отвечают следующие скрипты Python:

1. main.py
2. service\_param\_repository.py
3. excel\_parser.py

### 3.4 Главное меню

Главное меню – это меню с которым непосредственно взаимодействует обычный пользователь системы, в него он попадает после авторизации в систему.

В главном меню разрабатываемой системы пользователь видит информацию о авторизованном пользователе (при нажатии на иконку пользователя в правом верхнем углу), две кнопки вызывающих основной функционал программы, кнопка “Успеваемость” и кнопка “Печать ведомостей”, при

использовании которых выводит окно успеваемости и окно печати ведомостей соответственно, также аналогично как и в админском меню присутствует кнопка “Выйти”, при активации которой осуществляется выход из учетной записи и возврат к меню авторизации. Главное меню показано на рисунке 24.

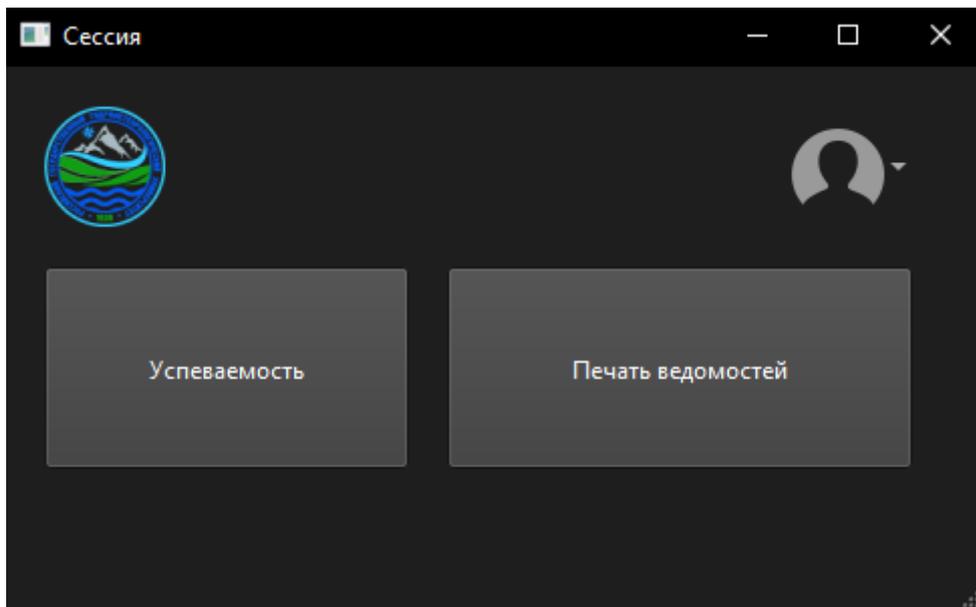


Рисунок 24. Графический интерфейс главного меню.

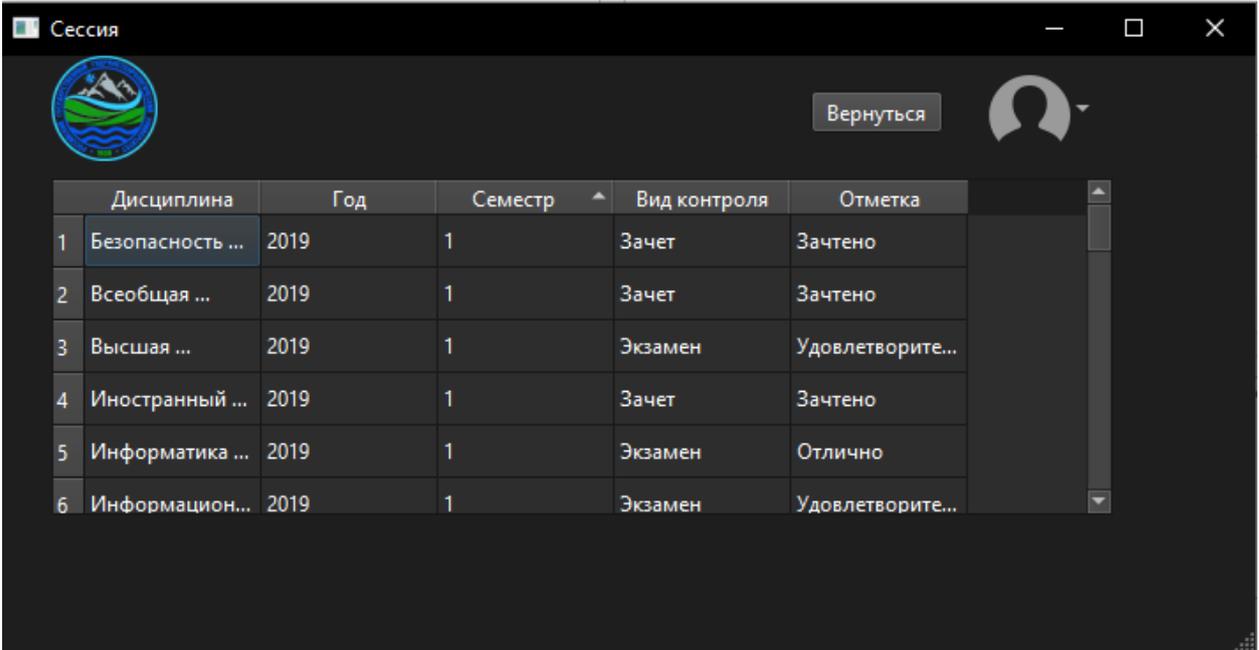
За работу окна и функционал отвечают следующие скрипты Python:

1. main.py
2. login\_result.py
3. student.py

### 3.5 Окно успеваемости

Окно успеваемости – окно вызываемое активацией кнопки “Успеваемость” в главном меню. В окне успеваемости выводится полная интерактивная таблица дисциплин пройденных студентом в процессе обучения, таблица содержит в себе информацию о наименовании дисциплины, году аттестации по этой дисциплине, семестре в котором проходило обучение, виде контроля и отметке. Помимо таблицы в окне присутствуют информация о авторизовавшемся пользователе, а также интерактивные кнопки “Вернуться”

для возвращения к главному меню и “Выйти”, для выхода из учетной записи и возвращения к меню авторизации. Окно успеваемости показано на рисунке 24.



	Дисциплина	Год	Семестр	Вид контроля	Отметка
1	Безопасность ...	2019	1	Зачет	Зачтено
2	Всеобщая ...	2019	1	Зачет	Зачтено
3	Высшая ...	2019	1	Экзамен	Удовлетворите...
4	Иностранный ...	2019	1	Зачет	Зачтено
5	Информатика ...	2019	1	Экзамен	Отлично
6	Информацион...	2019	1	Экзамен	Удовлетворите...

Рисунок 25. Графический интерфейс меню с таблицей успеваемости.

За работу окна и функционал отвечают следующие скрипты Python:

1. main.py
2. student.py
3. subject.py
4. mark.py
5. user\_repository.py
6. mark\_repository.py
7. subject\_repository.py

### 3.6 Окно печати ведомостей

В окне печати ведомости, в отличие от окна успеваемости, в таблицу выводятся не все дисциплины, а только те по которым имеются непогашенные задолженности. Щелкнув по предмету в таблице можно нажать на кнопку “Печать”, после чего инициализируется принтер и

осуществляется печать сформировавшейся ведомости. Помимо кнопки “Печать” имеются кнопки “Вернуться” и “Выйти” исполняющие те же функции что были описаны в пункте 3.5. Окно печати ведомостей отображено на рисунке 25.

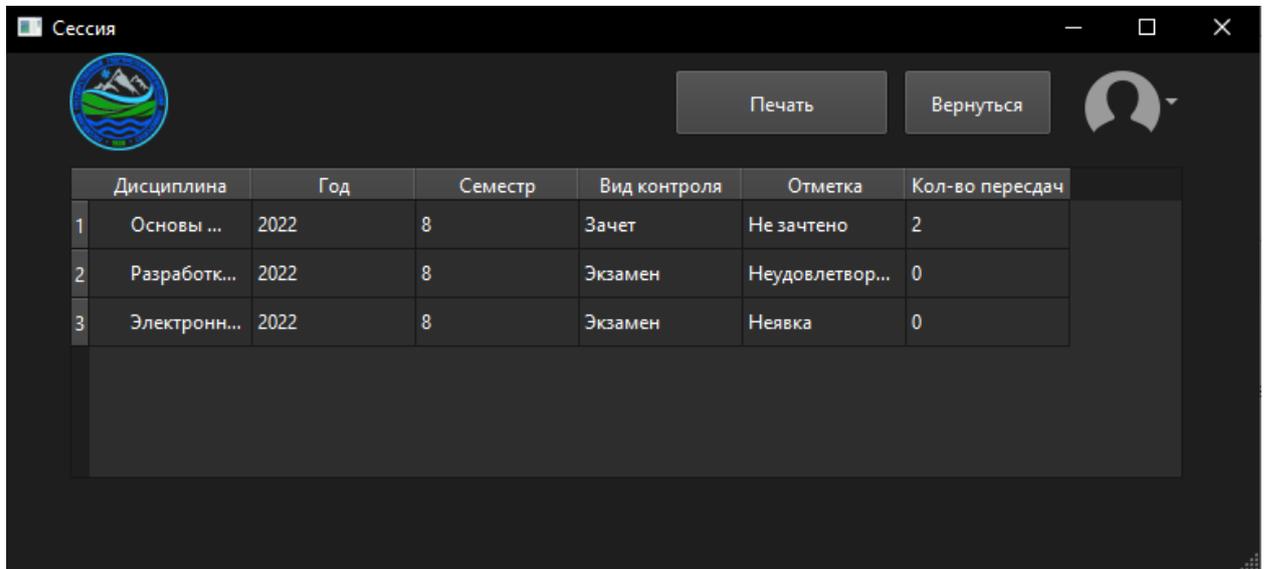


Рисунок 26. Графический интерфейс меню печати.

За работу окна и функционал отвечают следующие скрипты Python:

1. main.py
2. student.py
3. subject.py
4. mark.py
5. user\_repository.py
6. mark\_repository.py
7. subject\_repository.py
8. document\_controller.py
9. document\_process\_error.py
10. document\_process\_result.py
11. drint\_document\_result.py

12.document\_builder.py

### 3.7 Расчет надежности

#### 3.7.1 Расчет надежности аппаратной части

Аппаратная часть включает в себя следующее оборудование:

- Терминал
- Принтер
- Сервер базы данных

Исходя из данных устройства аппаратной, отраженной на рис. 26, отказ одного элемента цепи приведет к нарушению функционированию системы. В таком случае для расчета надежности используется следующая формула:

$$P_{\text{аппч}} = P_{\text{серв}} * P_{\text{терм}} * P_{\text{принт}}$$



Рисунок 27. Схема аппаратной части.

При расчете надежности стоит учитывать что:

- Отказы устройств могут являться случайными и независимыми;
- Устройства имеют одинаковые технические параметры.

Аппаратные части сервера и интенсивность их отказов представлены на таблице 6. Схема элементов сервера продемонстрирована на рисунке 26.

Таблица 6. Аппаратные части сервера и их интенсивность отказа.

Наименование элемента	Интенсивность отказа
Память сервера	$3.2 \times 10^{-8} \text{ч}^{-1}$
Жёсткий диск сервера	$8.7 \times 10^{-7} \text{ч}^{-1}$
Сетевая карта	$1.9 \times 10^{-6} \text{ч}^{-1}$
Котроллер RAID массива	$6.2 \times 10^{-7} \text{ч}^{-1}$
Блок питания	$2.0 \times 10^{-7} \text{ч}^{-1}$
Материнская плата	$4.2 \times 10^{-8} \text{ч}^{-1}$
Процессор с видео ядром	$3.0 \times 10^{-7} \text{ч}^{-1}$



Рисунок 28. Схема элементов сервера.

Вероятность безотказной работы рассчитывается по следующей формуле:

$$P(t) = e^{-\lambda_{\text{общ}} * t}, \text{ где}$$

- $t$  – время работы на отказ;
- $\lambda_{\text{общ}}$  - сумма интенсивности отказа всех элементов.

$$\lambda_{\text{общ}} = 3.2 * 10^{-8} + 8.7 * 10^{-7} + 1.9 * 10^{-6} + 6.2 * 10^{-7} + 2.0 * 10^{-7}$$

$$+ 4.2 * 10^{-8} + 3.0 * 10^{-7} = 3.964 * 10^{-6};$$

$$T = 1/\lambda_{\text{общ}} = \frac{1}{4.074 * 10^{-6}} = 2.522 * 10^5 \text{ - период времени наработки на отказ}$$

Исходя из расчетов выше, можно найти вероятность безотказной работы сервера:

$$P_{\text{серв}}(t) = e^{-\lambda_{\text{общ}} * t} = e^{-4.074 * 10^{-6} * 5000} \approx 0.98\%$$

Итого вероятность безотказной работы сервера равна 0.98%

Основные элементы терминала и интенсивность их отказа представлены в таблице 7. Схема основных элементов устройства терминала отражена на рисунке 27.

Таблица 7. Аппаратные части терминала и их интенсивность отказа.

Наименование элемента	Интенсивность отказа
Источник питания	$1.9 \times 10^{-7} \text{ч}^{-1}$
Накопитель	$7.2 \times 10^{-7} \text{ч}^{-1}$
Материнская плата устройства	$4.0 \times 10^{-8} \text{ч}^{-1}$
Сетевая карта	$1.0 \times 10^{-7} \text{ч}^{-1}$
Процессор устройства	$3.9 \times 10^{-7} \text{ч}^{-1}$

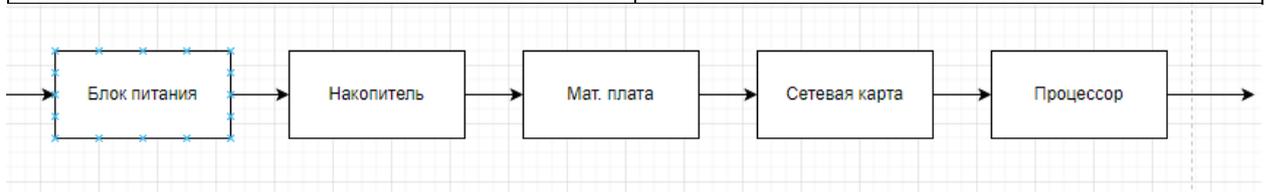


Рисунок 29. Схема элементов терминала.

Расчет интенсивности отказа терминала:

$$\lambda_{\text{общ}} = 1.9 * 10^{-7} + 7.2 * 10^{-7} + 4.0 * 10^{-8} + 1.0 * 10^{-7} + 3.9 * 10^{-7}$$

$$= 1.44 * 10^{-6}$$

Расчет наработки на отказ:

$$T = 1/\lambda_{\text{общ}} = \frac{1}{1.44 * 10^{-6}} = 6.944 * 10^5 \text{ часов}$$

Расчет вероятности безотказной работы терминала:

$$P_{\text{терм}}(t) = e^{-\lambda_{\text{общ}} * t} = e^{-1.44 * 10^{-6} * 5000} \approx 0.992\%$$

Вероятность безотказной работы принтера (Brother DCP-1510 series) была взята из технической документации к принтеру:  $P_{\text{принт}} = 0.578\%$

Имея вероятность безотказной работы всех элементов цепи, становится возможным найти вероятность безотказной работы всей аппаратной части:

$$P_{\text{аппч}}(t) = 0.98 * 0.992 * 0.578 \approx 0.56$$

### 3.7.2 Расчет надежности программного обеспечения

Программное обеспечения можно разбить на 2 модуля:

- ОС Windows
- Программный код

Схема ПО представлена на рис. 28.

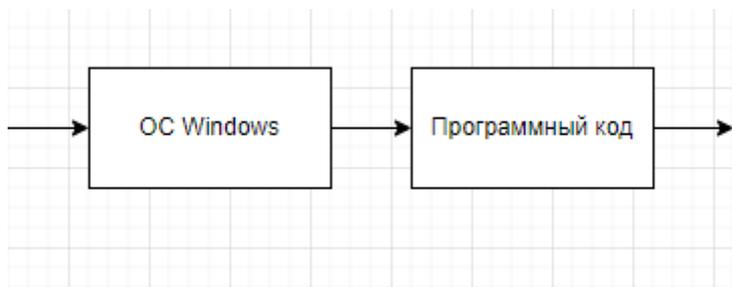


Рисунок 30. Общая схема ПО.

Для расчета безотказной работы используется следующая формула:

$$P_{\text{по}} = P_{\text{ос}} * P_{\text{пк}}, \text{ где}$$

$P_{oc}$  – вероятность безотказной работы ОС Windows. По заявлениям разработчика  $P_{oc} = 0.95$ .

$P_{пк}$  – вероятность безотказной работы программного кода. Расчет надежности программного кода происходит из количества написанного кода и корреляции между средним количеством ошибок на тысячу строк кода. В итоге  $P_{пк} \approx 0.99$ .

В итоге вероятность безотказной работы программного обеспечения равна:

$$P_{по} = P_{oc} * P_{пк} = 0.95 * 0.99 = 0.9405 \approx 0.94$$

### 3.7.3 Расчет надежности системы в целом

Исходя из расчетов в предыдущих подпунктах можно оценить надежность всей системы.

Расчет надежности системы производится по следующей формуле:

$$P_{сис} = P_{по} * P_{аппч} = 0.94 * 0.56 = 0.5264$$

Исходя из расчетов самым слабым звеном системы является принтер Brother DCP-1510 series, который легко подвергается замене.

## ЗАКЛЮЧЕНИЕ

Автоматизация процессов в образовании является одним из востребованных и актуальных направлений автоматизации. Активная автоматизация и переход на цифровые технологии позитивно складываются на всем процессе образования, на всех его этапах и стадиях. В рамках работы над проектом, направленным на автоматизацию одного из аспектов в процессе образования были поставлены следующие цели:

- Анализ предметной области
- Сроки реализации проекта и оценка материальных затрат
- Моделирование предметной области
- Реализация ИС

Анализ предметной области, сроки реализации проекта и оценка материальных затрат были проведены в первой главе работы, моделирование предметной области во второй, частичная реализация информационной системы отражена в третьей главе.

В первой главе был произведен анализ предметной области, что позволило выявить сущность системы, определить первоначальные минимальные требования к функциональности системы, а также определить границы системы. Был произведен анализ целевой аудитории, который позволил выявить категорию потребителей системы, были выявлены функциональные и нефункциональные требования, чтобы точно знать какие требования должны быть обязательно реализованы в системе. Был осуществлен расчет технико-экономических обоснований и сроков реализации проекта, для того чтобы представлять примерную стоимость проекта учитывая затраты на разработку, зарплату разработчикам и стоимость оборудования. Был осуществлен анализ проекта по нескольким

подходам для того чтобы выявить слабые и сильные стороны проекта, конкурентные преимущества, возможности дальнейшего развития системы, а также выявления информационных потоков данных и того как примерно работает система.

Во второй главе осуществлялся выбор методологии проектирования и моделирование предметной области. В ходе сравнения различных нотаций и методологий проектирования был сделан выбор в пользу методологии UML, на основе которой осуществлялось моделирование системы, была составлена диаграмма прецедентов, где для каждого из отраженных действий был расписан ход сценария, были составлены диаграммы последовательностей, отражающие взаимодействие объектов в системе, также были реализованы диаграмма компонентов и диаграмма развертывания для визуализации организации компонентов системы и отражения архитектуры развертывания системы соответственно. В конце была приведена схема базы данных, которая отражает логическую конфигурацию базы данных, отражает связь таблиц с данными в будущей информационной системе.

В третьей главе была представлена реализация проекта на текущий момент, описан принцип работы, представлены компоненты программы. Был подробно описан графический, программный и аппаратный интерфейс системы, указаны средства и библиотеки используемые при разработке информационной системы, подробно описаны функции, фоновые процессы системы для каждого из окон программы, а так же указаны Python скрипты отвечающие за процессы и работу окон, также был произведен расчет надежности программы.

Разрабатываемая информационная система “Автоматизация получения ведомости в деканате” существенно оптимизирует взаимодействие студентов с деканатом, позволяя студентам самостоятельно отслеживать свою успеваемость и получать ведомости на сдачу задолженности минуя походы в деканат. Информационная система имеет простой и интуитивно понятный

интерфейс взаимодействия с пользователем, предоставляет пользователю данные об успеваемости в виде понятной и детальной таблицы, дает возможность печатать ведомости и соответствует всем заявленным требованиям.

В итоге, в процессе работы над проектом был произведен практически полный комплекс мероприятий по подготовке к разработке и непосредственно самой разработке информационной системы. На данный момент программа находится на этапе альфа-тестирования, после проведения которого остаются этапы составления отчета по альфа-тестированию, ремонту, повторному тестированию и составлению по нему отчета и наконец работе с документацией и вводом в действие. Программа находится в рабочем состоянии и позволяет выполнять все возложенные на нее задачи.

### Список литературы и источников

1. ГОСТ 34.602-2020 . МГС. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. – М.:Изд-во стандартов, 2021 – 30с.
2. ГОСТ 7.53—2001. Издания. Международная стандартная нумерация книг. — Взамен ГОСТ 7.53—86; введ. 2002—07—01. — Минск: Межгос. совет по стандартизации, метрологии и сертификации; М.: Изд-во стандартов, сор. 2002. — 3 с.
3. ГОСТ 19.701-90 (ИСО 5807-85). ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. – М.:Изд-во стандартов, 1991. – 26 с.
4. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание. Общие требования и правила составления.
5. ГОСТ 7.32-2001 «Отчет о научно-исследовательской работе. Структура и правила оформления».
6. Сайт поиска работы HeadHunter [Электронный ресурс]. – Режим доступа: <https://spb.hh.ru/?hhtmFrom=article>
7. Visual paradigm online. Online Productivity suite [Электронный ресурс]. – Режим доступа: [online.visual-paradigm.com](http://online.visual-paradigm.com)
8. Хабр. Сообщество IT-специалистов [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/flows/develop/>
9. Служба поддержки Microsoft [Электронный ресурс]. – Режим доступа: <https://support.microsoft.com/ru-ru>
10. Openpyxl a python library to read [Электронный ресурс]. –Режим доступа: <https://openpyxl.readthedocs.io/en/stable/>
11. PDFtoPrinter: Command-line PDF printing [Электронный ресурс]. – Режим доступа: <http://www.columbia.edu/~em36/pdfprinter.html>

- 12.Pandas [Электронный ресурс]. – Режим доступа: <https://pandas.pydata.org>
- 13.PyQt6. Python bindings for the Qt cross platform application toolkit [Электронный ресурс]. – Режим доступа: <https://pypi.org/project/PyQt6/>
- 14.Гультияев, А.К. Проектирование и дизайн пользовательского интерфейса[Текст] / А.К. Гультияев - СПб.: КОРОНАПринт, 2000. - 349 с.
- 15.Вендров, А.М. Современные методы и средства проектирования информационных систем [Текст] / А.М. Вендров - М.: Финансы и статистика, 2008. – 65 с.
- 16.Илюшечкин, В.М. Основы использования и проектирования баз данных[Текст] / В.М. Илюшечкин - М.: Издательство Юрайт 2010. - 213с.
- 17.Карпова, Т.С. Базы данных: модели, разработка, реализация[Текст] / Т.С. Карпова – СПб.: Питер, 2002. – 304с.
- 18.Антонов А.В. Системный анализ. Методология. Построение модели: Учеб. пособие. - Обнинск: ИАТЭ, 2007. - 272 с
- 19.Эйри, Джоунс Функции SQL. Справочник программиста / Джоунс Эйри. - М.: Диалектика / Вильямс, 2013. - 0 с
- 20.Ларман, Крэг Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку / Крэг Ларман. - М.: Вильямс, 2013. - 736 с.
- 21.Грекул, В. И. Управление внедрением информационных систем / В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина. - Москва: РГГУ, 2014. - 224 с.
- 22.Новицкий Н. И., Горюшкин А. А., Кривенков А. В. Техничко-экономические показатели работы предприятий; ТетраСистемс - Москва, 2010. - 272 с.
- 23.Векторные диаграммы в схемах релейной защиты и автоматики. - М.: НЦ ЭНАС, 2021. - 128 с.
- 24.Мишин А. В. Информационные технологии в профессиональной

25. деятельности: учебное пособие / А. В. Мишин, Л. Е. Мистров, Д. В. Картавец.
- 26.- Москва : Российская академия правосудия, 2011 - 311 с.
27. Силич В. А. Реинжиниринг бизнес-процессов : учеб. пособие / В. А.
28. Силич, М. П. Силич. - Томск : ТУСУР, 2014 - 199 с.
29. Леоненков, А.В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose / А.В. Леоненков - Москва: БИНОМ, 2006. - 319 с.

## **Приложение А. Техническое задание на разработку информационной системы “Автоматизация процесса выдачи направления в деканате”**

### **1. Общие сведения**

#### **1.1 Наименование системы**

**1.1.1 Полное наименование системы** – “Автоматизация процесса выдачи ведомости в деканате”

**1.1.2 Краткое наименование системы** – “АПВВД”

#### **1.2 Основания для проведения работ**

Работа выполняется на основании допуска к сдаче ВКР в срок до ... мая

#### **1.3 Наименование организаций Заказчика и Разработчика**

**1.3.1 Заказчик** - РГГМУ

**1.3.2 Разработчик** – Косабуко В.И.

#### **1.4 Сроки**

Планируется завершить разработку программы за 10 дней до даты сдачи ВКР (ориентировочно до 26-27 мая) на основании того что за 10 дней до сдачи вносить корректировки в работу запрещается

#### **1.5 Источники финансирования**

Отсутствуют

## **1.6 Порядок оформления и предъявления заказчику результатов работ**

Работы по созданию программы ведутся поэтапно. По окончании работ в указанный срок разработчик сдает готовую систему.

## **2. Назначение и цели создания системы**

### **2.1 Назначение системы**

Система предназначена для автоматизации взаимодействия студентов с деканатом, в том числе и период дополнительной сессии.

### **2.2 Цели создания системы**

Создать возможность студентам отслеживать свою успеваемость минуя очередь в деканат, а также возможность самостоятельно печатать ведомость на сдачу задолженности в период доп.сессии

## **3. Характеристика объектов автоматизации**

Автоматизация получения ведомости

Автоматизация отслеживания успеваемости

## **4. Требования к системе**

### **4.1 Требования к системе в целом**

#### **4.1.1 Требования к структуре и функционированию системы**

Система должна быть централизованной, т.е. все данные должны располагаться в центральном хранилище.

Авторизация в систему студентом должна осуществляться по ФИО и номеру зачетной книжки.

Обслуживающий персонал(сотрудники деканата) осуществляют авторизацию в системе по специальной учетной записи

Обслуживающий персонал должен иметь возможность:

- Загружать актуальные данные об успеваемости студентов
- Выставлять сроки возможной печати ведомостей на сдачу задолженностей (сроки доп.сессии) в формате с dd/mm/yy по dd/mm/yy

Студент (основной пользователь системы) после авторизации должен иметь выбор между функциями:

- Просмотр долгов
- Печать ведомости

#### **4.1.2 Требования к численности и квалификации персонала системы и режиму его работы**

##### **4.1.2.1 Требования к численности персонала**

Будет достаточно одного человека

##### **4.1.2.2 Требования к квалификации персонала**

Квалификация не требуется

##### **4.1.2.3 Требования к режиму работы персонала**

Стандартный режим работы деканата

##### **4.1.3 Требования к материальной базе**

Терминал (ПК) и принтер

##### **4.1.4 Требования к эргономике и технической эстетике**

Система должна иметь минималистичный, простой и понятный интерфейс, соответствующий следующим требованиям:

- Интерфейс должен иметь русскоязычную локализацию
- Должен использоваться понятный читаемый шрифт
- При возникновении ошибок при печати/авторизации должно выводиться сообщение об ошибке

#### **4.1.5 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы**

Те же требования что и к терминалу и принтеру

#### **4.1.6 Требования к информационной безопасности**

БД должна быть защищена от несанкционированного доступа

#### **4.1.7 Требования по сохранности информации при авариях**

В случае аварии необходимо обеспечить сохранность данных

#### **4.1.8 Требования по стандартизации и унификации**

Разработка системы должна осуществляться с использованием методологий функционального моделирования: IDEF0, DFD и информационного моделирования IDEF1X.

Для работы с БД должен использоваться язык запросов SQL.

Система должна быть написана на языке Python.

#### **4.1.9 Требования к ПО**

Операционная система Windows

### **5. Состав и содержание работ по созданию системы**

Работы по созданию системы выполняются следующим образом:

Предпроектный анализ –

Проектный анализ –

Проектирование –

Реализация –

Отладка –

Тестирование –

Реализация –

## **6. Порядок контроля и приемки системы**

В качестве контроля и приемки системы планируется провести 2 тестирования: первичное альфа-тестирование, по итогам которого необходимо составить отчет и внести соответствующие правки в работу программы, устранить баги и недочеты, после чего провести повторное тестирование с целью удостовериться в устранении ошибок работы программы, по итогам которого вновь пишется отчет, и при необходимости вновь производится отладка.

## **7. Требования к составу и содержанию работ по подготовке объекта к вводу системы в действие**

Перечень мероприятий по вводу системы в эксплуатацию:

- 1) Установка терминала (ПК) и принтера
- 2) Установка требуемого ПО на ПК
- 3) Установка системы на ПК

## **8. Требования к документации**

- 1) Схема функциональной структуры
- 2) Общее описание системы
- 3) Инструкция
- 4) Руководство пользователя
- 5) ГОСТ 19.701-90
- 6) ГОСТ 34.201-2020

## **9.Источники разработки**

Настоящее техническое задание разработано на основе следующих документов и информационных материалов:

ГОСТ 24.701-86 «Надежность автоматизированных систем управления».

ГОСТ 15150-69 «Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды».

ГОСТ 34.602-2020 «Комплекс стандартов на автоматизированные системы».

**Приложение Б. Программный код excel\_parser.py**

```
import pandas
import re
from model.mark import Mark
from model.student import Student
from model.subject import Subject

def parse(excel_document):
    data = pandas.read_excel(excel_document)
    student = __parse_name(data.iloc[0][1])
    index = -1
    current_year = 0
    current_semester = 0
    marks = []

    for row_data in data.iterrows():
        index = index + 1

        if index < 5: # skip header
            continue

        row = row_data[1]
        first_column = row[1]

        year = __parse_year(first_column)

        if year:
```

```

    current_year = year
    continue

    if __is_semester(first_column):
        current_semester = current_semester + 1
        continue

    mark = __parse_mark(row, student, current_year, current_semester)
    marks.append(mark)

return student, marks

def __parse_name(line):
    match = re.search(r'(\S*) (\S*) (\S*); зачетная книга (\d*)', line)

    if not match:
        raise Exception(f'Не получилось извлечь имя и номер ЗК из строки:
"{line}")

    last_name = match.group(1)
    first_name = match.group(2)
    patronymic = match.group(3)
    zkid = match.group(4)

    return Student(first_name, last_name, patronymic, zkid)

def __parse_year(line):
    if ', Рабочий план ' in line:
        match = re.search(r'(\d*) - \d*, .*Рабочий план ', line)

```

```
    if not match:
        raise Exception(f'Не удалось извлечь год обучения из строки:
"{line}")

    return match.group(1)

return None

def __is_semester(line):
    return ' семестр' in line

def __parse_mark(row, student, year, semester):
    name = row[1]
    control_type = row[4]
    mark = row[6]

    subject = Subject(name)
    return Mark(student, subject, control_type, mark, year, semester)
```